performing a wavelet transform on an input digital image. The resulting wavelet components are compared to a threshold value; coefficients falling below the threshold are discarded. The remaining coefficients are quantized. The quantized coefficients are then compressed using an entropy encoding technique, such as arithmetic, run length, or Huffman encoding, or a combination of Huffman and run length encoding. The wavelet transform can be an integer reversible wavelet transform derived using a lifting scheme or correction method, while the quantization scheme can be sub-band oriented. To further enhance the speed of the compression scheme, input color image pixels can be reduced using a color table. In addition, color pixels can be transformed between color spaces prior to wavelet transformation.

According to another aspect of the invention, a corresponding method of decompression is provided.

According to another aspect of the present invention, a compression method is provided that allows user selected portions of an image to compressed to different image qualities, whereby permitting non-uniform image compression.

According to another aspect of the present invention, a compression method is provided that permits compression quality to be based on image specific parameters.

According to another aspect of the present invention, a method of compressing images using a "split and merge" technique is provided.

According to further aspect of the present invention, an image compression system includes a compressor configured to generate a compressed image based on an integer wavelet transform derived using either a lifting scheme or correction method. The compressor can be implemented using one or more electronic components, such as application specific integrated circuits (ASICs), microprocessors, discrete logic components, or any combination of the aforementioned.

According to another aspect of the present invention, a corresponding image decompression system is provided.

Brief Description of the Drawings

The invention is pointed out with particularity in the appended claims. However, other features of the invention will become more apparent, and the invention will be best understood by referring to the following detailed description in conjunction with the accompanying drawings, in which:

FIG. 1 illustrates a flow diagram for a method of compressing an image that is in accordance with an embodiment of the present invention;

FIGS. 2-4 depict wavelet coefficients for various levels of decomposition;

FIG. 5 illustrates a flow diagram of a method of decompressing an image that has been compressed using the method of FIG. 1;

FIG. 6 is a block diagram of a system that can incorporate a software program implementing any of the methods shown in FIGS. 1, 5, and 8-13 in accordance with a second embodiment of the present invention;

FIG. 7 is a block diagram of a system for compressing and decompressing an image in accordance with another embodiment of the present invention;

FIG. 8 illustrates a flow diagram of a method compressing an image that is in accordance with a further embodiment of the present invention;

FIG. 9 illustrates a flow diagram of a method for decompressing an image that has been compressed according to the method of FIG. 8;

FIG. 10 illustrates a flow diagram of a method of compressing an image in accordance with a further embodiment of the present invention;

FIG. 11 illustrates a flow diagram of a method of decompressing an image that has been compressed according to the method of FIG. 10;

FIG. 12 illustrates a flow diagram of a method of compressing an image that is in accordance with a further embodiment of the present invention; and

FIG. 13 illustrates a flow diagram of a method for decompressing an image that has been compressed according to the method of FIG. 12.

Detailed Description of the Preferred Embodiments

Referring now to the drawings, and in particular to FIG. 1, there is shown a flow diagram of a method for compressing an image that conforms to a first embodiment of the invention. In step 20, a digital image is received from an image source. The digital image consists of a matrix of values representing an array of pixels. Specifically, the array of pixels represents a still image or a frame from a video image. In step 22, the image is optionally displayed on an appropriate viewing device, such as a computer or video display unit having a flat panel or cathode ray tube (CRT). Next, in step 24, color and wavelet transformations of the image take place. The image transformations involved in this step include color transform for color images only, and wavelet transform for both gray level images and color images. In step 26, the values representing the transformed images are quantized and compared to thresholds. Values falling outside the threshold are discarded. In step 28, the remaining quantized values are encoded to remove redundant information, creating a compressed image file. Next, in step 30 the compressed image file is generated as output.

Referring to the color transformation of step 24, digital color images are typically based on an RGB color model, such as is commonly used with TIFF or BMP images. In order to get a higher compression ratio, the

RGB pixels are transformed to other color models, such as YIQ or YUV models. The method can convert RGB inputs into YIQ or YUV color spaces according to the following relationships.

RGB to YIQ:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.596 & -0.275 & 0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

RBG to YUV:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.148 & -0.289 & 0.439 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

In the YIQ color space, there is one luminescence (Y) and two color planes (I, Q). The Y component is critical, while the I-Q components are less sensitive to error introduced by data compression.

The wavelet transform (also referred to as wavelet decomposition) operates on the converted color space signals. The purpose of the wavelet transform is to represent the original image by a different basis to achieve the objective of decorrelation. There are many different wavelet transforms that can be used in this step. For instance, the reversible integer wavelet transform described herein below is a preferred wavelet transform. However, to develop a better understanding of the preferred transform, the following alternative wavelet transform is first described.

Let $C^0 = [C^0_{jk}]$ ($j - 0, \ldots, M-1$; $k = 0, \ldots, N-1$) represent the original, uncompressed image, where M and N are integers which have the common factor 2^L (L is a positive integer). A one-level wavelet decomposition, where L = 1, results in the four coefficient quadrants as shown in Figure 2. Each quadrant represents a set of wavelet coefficients.

Quadrant C^1 represents the blurred image of the original image C^0, where $C^1 = [C^1_{jk}]\,(j=0,\dots,\frac{M}{2}-1;\ k=0,\dots,\frac{N}{2}-1)$. HD^1 represents the horizontal high frequency part of C^0, while VD^1 represents the vertical high frequency part of C^0, and DD^1 represents the diagonal high frequency part of C^0. The decomposition can be iteratively repeated L times to obtain different levels of decomposition. For example, for L = 2, C^0 is set to equal C^1. The iterative formula for computing a decomposition is given as follows:

(1) Let $\overline{C}^0 = rC^0$, r>0 is a factor which can be changed for different needs.

(2) Transform for image columns:

For k=0,...,N-1, calculate

$$\begin{cases} \tilde{d}^1_{0k} = \dfrac{\overline{C}^0_{1k} - \overline{C}^0_{0K}}{2}, \\ \tilde{d}^1_{jk} = \dfrac{1}{4}\left(\overline{c}^{-0}_{2j-1,k} - 2\overline{c}^{-0}_{2j,k} + \overline{c}_{2j+1,k}\right), j=1,\dots,\dfrac{M}{2}-1. \end{cases} \quad (3.1.1)$$

For k=0,...,N-1, calculate

$$\begin{cases} \tilde{C}^1_{0k} = \overline{C}^0_{1,k} - \dfrac{\tilde{d}_{ok} + \tilde{d}_{1,k}}{2}, \\ \tilde{c}^1_{jk} = \overline{C}_{2j+1,k} - \dfrac{\tilde{d}^1_{jk} + \tilde{d}_{j+1,k}}{2}, j=1,\dots,\dfrac{M}{2}, \\ \dfrac{\tilde{C}^{-1}_{N-2}}{2}, k = \tilde{C}^{-0}_{N-1,k} - \dfrac{\tilde{d}_{N-2}}{2}, k. \end{cases} \quad (3.1.2)$$

(3) Transform for rows:

For j = 0, . . . M/2 - 1, computing

$$\begin{cases} hd^1_{j,0} = \dfrac{\tilde{c}_{j1} - \tilde{c}^1_{j0}}{2}, \\ hd^1_{jk} = \dfrac{1}{4}\left(\tilde{c}^1_{j,2k-1} - 2\tilde{c}_{j,2k+1}, + \tilde{c}_{j,2k+1}\right), k=1 \ldots \dfrac{N}{2}-1. \end{cases} \quad (3.1.3)$$

and

$$\begin{cases} c^1_{jo} = \tilde{c}^1_{j,1} - \dfrac{hd^1_{jo} + hd^1_{j1}}{2}, \\ c^1_{jk} = c^1_{j,2k+1} - \dfrac{hd^1_{j,k} + hd^1_{j,k+1},}{2}, \ldots, \dfrac{M}{2}-2, \\ c_{j,\frac{N-2}{2}} = \tilde{c}^1_{j,N-1} hd_{j,\frac{N-2}{2}}, \end{cases} \quad (3.1.4)$$

For j = 0, . . . , M/2 - 1, computing

$$\begin{cases} dd^1_{j,0} = \dfrac{\tilde{d}^1_{j,1} - \tilde{d}^1_{j0}}{2}, \\ dd^1_{jk} = \dfrac{1}{4}\left(\tilde{d}^1_{j,2k-1} - 2\tilde{d}_{j,2k} + \tilde{d}^1_{j,2k+1}\right), k=1 \ldots \dfrac{N}{2}-1. \end{cases} \quad (3.1.5)$$

and

$$\begin{cases} vd^1_{jo} = \tilde{d}^1_{j,1} - \dfrac{dd^1_{jo} + dd^1_{j1}}{2}, \\ vd^1_{jk} = \tilde{d}^1_{j,2k+1} - \dfrac{dd^1_{j,k} - dd^1_{j,k+1}}{2}, k=1, \ldots, \dfrac{M}{2}-2, \\ vd_{j,\frac{N-2}{2}} = \tilde{d}_{N-1,k} - dd^1_{j,\frac{N-2}{2}}, \end{cases} \quad (3.1.6)$$

(4) $C^1 = [c^1_{j,k}]$, $HD^1 = [hd^1_{j,k}]$, $VD^1 [vd^1_{j,k}]$ and $DD = [dd^1_{j,k}]$, $j=0, \ldots, \frac{M}{2}-1$ $k=0, \ldots, \frac{M}{2}-1$.

Remark: If it is necessary, we also can use matrix multiply Wavelet Coefficient Image of 1 levels$=W_1 C^{\circ} W_1^T$.

Here, W^1 is the transform matrix for l level wavelet decomposition.

FIG. 3 depicts a three-level wavelet decomposition, where L = 3.

In step 26, the first loss in accuracy occurs. Both thresholding and quantization reduce accuracy with which the wavelet coefficients are represented. In step 26, the wavelet coefficients are matched against threshold values, and if the values are less than the established threshold values specified, then the resultant value is set to zero.

An important feature of the invention is that the wavelet coefficients are then quantized to a number of levels depending upon which quadrant is being processed, and the desired compression or quality factor. This can be very important in image compression, as it tends to make many coefficients zeros, especially those for high spatial frequencies, which reduces the size of a compressed image.

A multilevel uniform thresholding method can be used as described below.

Let $T = (t_1, \ldots, t_L, t_{L+1})$ be the chosen thresholds, where t_l is the threshold for l the (l=I, ..., L) level and t_{L+1} is a threshold for blurred image C^L. Thresholding sets every entry in the blocks C^L, HD^l, VD^l and DD^l (l = I, ... L) to be zero if its absolute value is not greater than the corresponding threshold.

For color images, three threshold vectors which correspond three different color planes, such as y, I and Q, are used.

The step of quantization essentially scales the wavelet coefficients and truncates them to a predetermined set of integer values. The quantization table shown in Table 1 can be used.

q^1_{HD} q^1_{VD} q^1_{DD}	q^2_{HD} q^2_{VD} q^2_{DD}	... ...	q^L_{HD} q^L_{VD} q^L_{DD}	q_c^{L+1}

TABLE 1

In Table 1, the entries q^l_{HD} are quantization factors for blocks HD^l (l = I, ... L), q^l_{VD} and q^l_{DD} for blocks VD^l and DD^l (l = I, ..., L) respectively, and the factor q_c^{L+1} is for the most blurred image C^L. The factors can be integers between 0 and 255. The quantization scheme for the block HD^l (l = I, ..., L) is

$$\overline{hd}^l_{j,k} = round\ \frac{hd^l_{j,k} \cdot q^l_{HD}}{\max^l_{HD}}, j=0,\ldots,\frac{M}{2^l}-1; k=0,\ldots,\frac{N}{2^l}-1. \quad (3.2.1)$$

Here, $\overline{hd}^l_{j,k}(j=0,\ldots,\frac{M}{2^l}-1;\ ,k=0,\ldots,\frac{N}{2^l}-1)$ *are quantized wavelet coefficients of block* HD^l $(l=1,\ldots,L)$

$$\max^l_{HD} = \max_{\substack{0\le j\le (M/2^l-1)\\ 0\le k\le (N/2^l-1)}} (|hd^l_{j,k}|),$$

and the function round*(x)* gives the nearest integer of x. Equation (3.2.1) is used for quantization of the other blocks (quadrants).

For color images, there are three separate quantization tables for the different color bands.

In step 28, entropy compression is applied to the resultant coefficients using either Arithmetic, Run Length, or Huffman, or Huffman and Run Length combined. The compression algorithm can be selected at run-time by the user, based on the desired compression ratio and the amount of time required to get the selected level of compression. The encoding step includes the entropy compression as well as coefficient rearranging.

An alternative process to that shown in FIG. 1 includes an optional down sampling of the IQ color planes. This down sampling may be done once or twice to produce two image planes either one-fourth or one-sixteenth the size of the original plane. If the down sampling is done, it will be accomplished prior to the wavelet transform of step 24. The down sampling reduces the compression time and size of the image file.

FIG. 5 shows a corresponding method for decompressing an image compressed using the method of FIG. 1. In step 40, the compressed image file is input. In step 42, the image is decoded. Next, in step 44 the values are de-quantized. Next, in step 46 inverse color and wavelet transformations are performed on the de-quantized data. In step 48, optional image post-processing takes place to refine the decompressed image. In step 50, the decompressed image is displayed.

The decoding of step 42 is the inverse operation of the encoding of step 28. Similarly, it can be divided into two parts: Entropy decoding (Huffman or arithmetic), and coefficient rearranging.

The decoding step produces quantized wavelet coefficients in 3*L+1 blocks. Dequantizing (step 44) uses the same quantization table as quantizing (Table 1), and the scheme as follows: for l = I, ..., L

$$\underline{hd}^{l}_{j,k} = \frac{\overline{hd}^{l}_{j,k} \cdot \max^{l}_{HD}}{q_{HD}}, j=0,\dots,\frac{M}{2^{l}}-1;\ k=0,\dots,\frac{N}{2^{l}}-1. \quad \textbf{(4.2.1)}$$

Equation (4.2.1) produces the approximate coefficients for the blocks HD^l (l = I, ..., L), which are shown in FIG. 3. The dequantizing scheme for other blocks is similar to 4.1.2).

In step 46, the inverse wavelet transform, also referred to as wavelet reconstruction, is performed prior to the inverse color transformation. FIG. 4 depicts a one-level wavelet reconstruction.

The wavelet reconstruction can be iteratively performed for various levels of decomposition, according to the following equations.

(1) Inverse transform for rows:

For j=0, ..., $\frac{M}{2}-1$, calculate

$$\begin{cases} \tilde{d}^1_{j,1} = \upsilon d^1_{j,0} + \frac{dd^1_{j,0}+dd^1_{j,1}}{2} \\ \tilde{d}^1_{j,2k+1} = \upsilon d^1_{jk} + \frac{dd^1_{j,k}-dd^1_{j,k+1}}{2}, k=1,\ldots,\frac{N}{2}-2, \\ \tilde{d}_{N-1,k} = \upsilon d_{j,\frac{N-1}{2}} + dd^1_{j,\frac{N-2}{2}}. \end{cases} \quad (4.3.1)$$

and

$$\begin{cases} \tilde{d}^1_{j,0} = d^1_{j,1} - 2dd^1_{j,0}, \\ \tilde{d}^1_{j,2k} = \frac{\tilde{d}^1_{j,2k-1}+\tilde{d}^1_{j,2k+1}}{2} - 2dd^1_{j,k}, \; k=1,\ldots,\frac{N}{2}-1. \end{cases} \quad (4.3.2)$$

For j=0, ..., M/2 - 1, calculate

$$\begin{cases} \tilde{c}^1_{j,1} = c^1_{j0} + \frac{hd^1_{j0}+hd^1_{j1}}{2} \\ \tilde{c}^1_{j,2k+1} = c^1_{jk} + \frac{hd^1_{j,k}-hd^1_{j,k+1}}{2}, k=1,\ldots,\frac{N}{2}-2, \\ \tilde{c}_{j,N-1} = c_{j,\frac{N-2}{2}} + hd^1_{j,\frac{N-2}{2}}. \end{cases} \quad (4.3.3)$$

and

$$\begin{cases} \tilde{c}^1_{j,0} = c^1_{j,1} - 2hd^1_{j,0}, \\ \tilde{c}^1_{j,2k} = \frac{1}{2}(\tilde{c}^1_{j,2k-1}+\tilde{c}^1_{j,2k+1}) - 2hd^1_{jk}, \; k=1\ldots,\frac{N}{2}-1. \end{cases} \quad (4.3.4)$$

(2) Inverse transform for column:

For k=0, ..., N - 1, calculate

and

$$\begin{cases} \bar{c}^0_{1,k} = \tilde{c}^1_{0k} + \dfrac{\tilde{d}^1_{0k} + \tilde{d}^1_{1,k}}{2}, \\ \bar{c}^0_{2j+1,k} = \tilde{c}^1_{jk} + \dfrac{\tilde{d}^1_{j,k} - \tilde{d}^1_{j+1,k}}{2}, j=1,\ldots,\dfrac{M}{2}-2, \\ \bar{c}^0_{N-1,k} = \tilde{c}^1_{\frac{N-2}{2},k} + \tilde{d}^1_{\frac{N-2}{2},k}. \end{cases} \quad \mathbf{(4.3.5)}$$

$$\begin{cases} \bar{c}^0_{0k} = c^0_{1k} - 2\tilde{d}^1_{0k}, \\ \bar{c}^0_{2j,k} = \dfrac{1}{2}(\bar{c}^0_{2j-1,k} + \bar{c}^0_{2j+1k},) - 2\tilde{d}^1_{jk}, \; j=1,\ldots,\dfrac{M}{2}-1. \end{cases} \quad \mathbf{(4.3.6)}$$

(3) $$c^0_{j,k} = \tilde{c}^0_{j,k} / r, \; j=0,\ldots,M-1; \; k=0,\ldots,N-1.\, C^0 = [c^0_{j,k}]^{N}_{1} x^{N}_{2}.$$

Following the inverse wavelet transformation, an inverse color transform is performed. Equations (5)-(6) give the inverse transforms for the YIQ and YUV color spaces.

For YIQ to RGB:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad (5)$$

For YUV to RGB:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.581 \\ 1.000 & 2.032 & 0.000 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \quad (6)$$

In step 48, a user can optionally apply image filtering to improve the image quality. Filters are known in the art for sharpening, smoothing and brightening images. Users can choose any number of processing filters at compression time. Information defining the selected filters can be stored in the coded image file, in a form such as a one byte flag in a file header. In addition to optionally applying the filters, the method can also be implemented to automatically detect and apply the selected filters following decompression.

To sharpen an image, a filter is used that weights the eight pixels adjacent to the current pixel, as well as the current pixel, by one or more predetermined values. The weighted values of the nine pixels are then summed to derive a new value for the current pixel. For example, the surrounding eight pixel values can be weighted by the value -35/800, while the current pixel is weighted by 1.35. The sharpening filter is applied to every pixel in the image.

To smooth images, for every pixel, the average of the pixel and the eight adjacent pixels is calculated. Then the pixel value and the average is compared. The smaller of the two replaces the original pixel and is output as the smoothed pixel value.

To brighten images, the weighted sum of each pixel and the correspond eight adjacent pixels is calculated. For example, each of the adjacent pixels can be multiplied by the value 1/90 and the summed with the current pixel to obtain a brighten current pixel.

Another filter that can be used is one that adds a random value between [-12, 12] to each of the pixels in the image.

In FIG. 6 there is displayed a preferred hardware platform that can execute software for implementing an embodiment of the present invention. The computer system of FIG. 3 includes a CPU 62, a main memory 64, an I/O subsystem 66, and a display 68, all coupled to a CPU bus 70. The I/O subsystem 66 communicates with peripheral devices that include an image source 72, an image storage device 74, and a mass storage memory 76. Although shown as three separate devices, peripherals 72-76 can be implemented using a single memory device, such as a hard disk drive commonly found in computers.

The image source 72 may be a digital still image or video source, such as a CD-ROM drive, scanner, or network connection. In addition, the image source 85 can include analog video sources, such as a video camera, VCR, television broadcast or cable receiver. The analog video signals would be converted to a digital form by the image source 85 using conventional conversion techniques. Alternatively, an image source 72 can include a video camera and communications systems for transmitting real-time video to the I/O subsystem 66.

The image storage 74 can be a computer disk, such as a that used by a hard drive, or a portable memory medium, such as a floppy or ZIP disk, or a read/write optical CD.

In operation, a computer program, which implements aspects of the invention, is retrieved from the mass storage memory 76 into the main memory 64 for execution by the CPU 62. Upon execution of the compression aspect of the invention, the compressed image file can be stored in the image storage 74; while upon execution of the decompression aspect of the invention, the decompressed image can be viewed on the display 68. Operating under the control of the computer program, the CPU 62 can process images according to the methods set forth herein, as shown in FIGS. 1-2 and 6-10.

FIG. 7 illustrates an alternative hardware platform implementing a system in accordance with a further embodiment of the present invention. System 80 can be implemented using a variety of different hardware components, such as ASIC (Application Specific Integrated Circuits), or a combination of discrete digital components, such as microprocessors, standard logic components, and other programmable logic devices. The system 80 includes a compression system 81 and a decompression system 82. The compression system 81 can be configured to perform any one or combination of the compression methods set forth in FIGS. 1, 8, 10, and 12;

while the decompression system can be configured to perform any one or combination of the decompression methods set forth in FIGS. 5, 9, 11, and 13.

An image source 85 provides digital pixel values to a color converter 84. The image source 85 can provide the same functionality as described earlier for the image source 72 of FIG. 6.

The color converter 84 performs a color space transformation on the input pixels, such as any of those described herein for FIG. 1. The converter functionality can be provided by conventional integrated circuits that are readily available from various manufacturers. Compressor 86 compresses the transformed pixels, removing redundant data. The compressed image file generated by the compressor 86 can be transferred directly to the decompression system 82 over a transmission medium 91. The transmission medium 91 can be a radio-link, computer network, cable television network, or satellite link. Alternatively, the compressor 86 can transmit its output to a portable storage medium 92, such as an optical, floppy, or ZIP disk; or to a mass storage device 94 such as a computer hard disk or archival system.

The decompressor 88 expands the compressed image file by applying an inverse wavelet transformation, as well as de-quantization and de-encoding functions. The decompressed data is then passed to an inverse color converter 90 that applies an inverse color space transformation to generate pixel values in a color space and format appropriate for the image display 89. Standard electronic components are readily available for performing the function of the inverse color converter 90.

FIG. 8 illustrates a flow diagram of a method of compressing an image in accordance with an alternative embodiment of the present invention. In step 100, a digital image is input. In step 102, a color space transformation is performed on the input image pixels.

In step 104, the pixels are subjected to a wavelet transformation. In step 106, sub-band quantization is performed on the wavelet coefficients. Next, in step 108 the quantized sub-bands are respectively entropy encoded. In step 110, the coded image file is output.

Sub-band oriented quantization and entropy coding are well suited for wavelet-based image compression. The main idea is to take the advantage of different quantizations at different sub-bands (wavelet quadrant) and encode each band accordingly. Quadrants having a high variance in wavelet values can be allocated a finer mesh size for quantization, while those quadrants with smaller variances will be assigned fewer levels of quantization. That is, the number of bits one wishes to allocate to the output could be varied by quadrant. Those quadrants with large variances will utilize more bits, while those with low variants will utilize fewer bits. In this way, the number of bits resulting from quantization will remain the same, but their allocation will differ depending upon the nature of the image. This technique greatly improves image quality while maintaining a high compression ratio.

FIG. 9 illustrates a flow diagram of a method of decompressing an image compressed according to the methods shown in FIG. 8. Step 120, the compressed file is input. In step 122, the input image is entropy decoded. In step 124, de-quantization is performed on the decoded image file. Next, in step 126, an inverse wavelet transform is performed on the image. In step 128, an inverse color transformation is performed. In step 130, post-processing altering is optionally performed. In step 132, the decompressed image file is then displayed.

FIG. 10 illustrates a flow diagram of a method of compressing an image in accordance with another embodiment of the present invention. This method performs color-bit depth compression, which essentially reduces the number of colors in the image to achieve compression. In step 140, the image is input with its original color. For example, each color pixel could be represented by a standard 24-bit value. Next, in step 142, a color table is created corresponding to the image. The color table is a set of quantized color values. The quantized color values represent a smaller number of colors with correspondingly fewer bits. Each of the input pixels is mapped to the color table. In step 144, an index is calculated for each pixel in the image by dithering the pixel values. Dithering is accomplished by weighting pixels adjacent to the current pixel in a frame and then arithmetically combining the weighted values with the current pixel value to produce the index, which then represents the current pixel. The dithering process is repeated for each pixel in a frame. In step 146, the indexes are wavelet transformed. In step 148, the wavelet coefficients are entropy coded. In step 150, the coded image file is output.

FIG. 11 illustrates a flow diagram of a method of decompressing an image that has been compressed according to the method shown in FIG. 10. In step 160, a compressed image file is received. Next, in step 162, the image file is entropy decoded. In step 164, an inverse wavelet transform is applied to the decoded data. Next, in step 166, post-processing filtering of the image is optionally applied. Next, in step 168, the decompressed image is displayed.

FIG. 12 illustrates another method of compressing an image in accordance with another embodiment of the present invention. In this method, a user can selective vary compression parameters (step 173) to obtain a lossless or near-lossless compressed image at a desired compression ratio. In step 170, the image is input. In step 172, an integer color transform is performed on the input image. In step 173, compression parameters are selected by the user using a software interface. These parameters can include those described herein below in the subsection title "Peak Signal to Noise Ratio (PSNR) Controlled Compression". In step 174, an integer wavelet transform is performed on the color transformed pixels. In step 176, the wavelet coefficients are entropy coded. Next, in step 178, the compressed image file is then output from the system.

The integer color transformation of step 172 is an integer reversible transform which can be used in color image compression to reduce processing time and image size. Step 172 transforms RGB color components to a set of color components Y-Nb-Nr, which are known.

The RGB to Y-Nb-Nr transform is given by the equations:

$$Y = G + Int(R/2 + B/2),$$
$$Nb = B - Int(Y/2),$$
$$Nr = R - Int(Y/2).$$

The integer wavelet transform of step 174 is described below in detail.

FIG. 13 illustrates a method of decompressing an image file that has been compressed according to the method shown in FIG. 12. In step 180, a compressed image file is input. In step 182, the image is entropy decoded. Next, in step 184, an inverse integer wavelet transform is performed on the decoded data. In step 186, an inverse integer color transform is performed. Next, in step 188 optional post-processing filtering is performed on the image. Next, in step 190, the decompressed image is displayed.

The Y-Nb-Nr to RGB transform of step 186 is given by the equations:

$$R = Nr + Int(Y/2),$$
$$B = Nb + Int(Y/2),$$
$$G = Y - Int\ (R/2 + B/2)$$

The inverse integer wavelet transform of step 184 is described in detail below.

Reversible Integer Wavelet Transform

This method allows a series of transformations which are very close to the corresponding biorthogonal wavelet transforms or some non-orthogonal wavelet transforms, but can be calculated with only integer addition and bit-shift operations. In addition, the integer wavelet transforms created disclosed herein possess a property of precision preservation (PPP). This property is very useful for conserving memory in both compression and decompression, and speed up the whole procedure in some applications. Two general methods from which one can get the integer wavelet transform desired are disclosed.

Basic Integer Wavelet Transformations

Two examples are provided as the starting point for the unique method. For the sake of convenience, length, and simplicity, presented is only the algorithm for a one level decomposition and reconstruction and only for a one dimensional signal. The extension to two dimensions is immediate as the rows and columns can be treated into a sequence of one dimensional signals. For the following examples, assume that $\{c_n^0\}_{n=0}^{N-1}$ is the original signal where the superscript indicates level and the subscript indicates a particular point in the signal. Also, $\{c_n^1\}_{n=0}^{M_1-1}$ and $\{d_n^1\}_{n=0}^{M_1-1}$ are its decomposition parts at the first level. Here

$$N_1 = \begin{cases} \frac{N}{2}, & \text{if } N \text{ is an even number,} \\ \frac{N+1}{2}, & \text{if } N \text{ is an odd number;} \end{cases}$$

$$M_1 = N - N_1$$

$\{c_n^1\}_{n=0}^{N_1-1}$ and $\{d_n^1\}_{n=0}^{M_1-1}$ are its low frequency (l) part and high frequency (h) part, respectively. For multi-levels, we just create $\{c_n^1\}_{n=0}^{N_1-1}$ as $\{c_n^0\}_{n=0}^{N_1}$ and repeat the procedure again.

Example 1: A (2,2)-wavelet transform by integer calculation.

This transformation is similar to a variation of the Haar wavelet transform which uses low and high pass analysis (decomposition) filters given as:

n	0	1
$\tilde{h}_n$	1/2	1/2
$\tilde{g}_n$	1/2	-1/2

(1) Compute

$$d_k^1 = c_{2k}^0 - c_{2k+1}^0, k=0, \ldots, M_1-1. \quad (2.1)$$

(2) Compute

$$c_k^1 = Int\left(\frac{d_k^1}{2}\right) + c_{2k+1}^0, \quad k=0,\ldots,N_1-2,$$

$$c_{N_1-1}^1 = \begin{cases} Int\left(\frac{d_{M_1-1}^1}{2}\right) + c_{2k+1}^0, & \text{if } N \text{ is an even number,} \\ c_{N-1}^0, & \text{if } N \text{ is an odd number.} \end{cases} \tag{2.2}$$

Here, Int(x) is an arbitrary rounding function which may have different interpretations. For example, Int(x) can be the integer which is nearest to x, or Int(x) may be any integer which satisfies $x-1<Int(x)\leq x$, etc. It is easy to see that all entries in both $\{c_n^1\}_{n=0}^{N_1-1}$ and $\{d_n^1\}_{n=0}^{M_1-1}$ are integers.

From (2.1)-(2.2), we can easily get the following integer reconstruction algorithm:

(b) Reconstruction

(1) If N is an even number, compute:

$$c_{2k+1}^0 = c_k^1 - Int\left(\frac{d_k^1}{2}\right), \quad k=0,\ldots,N_1-1; \tag{2.3}$$

or, if N is an odd number, we have

$$c_{2k+1}^0 = c_k^1 - Int\left(\frac{d_k^1}{2}\right). \quad k=0,\ldots,N_1-2;$$
$$c_{N-1}^0 = c_{N_1}^1. \tag{2.4}$$

(2) Compute

$$c_{2k}^0 = d_k^1 + c_{2k+1}^0 \quad k=0,\ldots,M_1-1. \tag{2.5}$$

Remark. Since (2.1)-(2.6) are not linear because of the rounding operation Int(x), this means the transformation order becomes significant. For instance, if the decomposition was applied first to the columns and then to the rows, the inverse transformation must be applied first to the rows and then to the columns.

Example 2: Lazy wavelet transform.
The lazy wavelet transform is used to illustrate an important concept. The corresponding inverse transform is nothing else but sub-sampling the even and odd indexed samples. Decomposition and reconstruction can use the same formula as follows:

$$c_k^1 = c_{2k}^0,\ k=0,\dots,N_1-1;$$
$$d_k^1 = c_{2k+1}^0,\ k=0,\dots,M_1-1.$$

Examples 1 and 2 are not good transforms for image compression, but they are simple. Much better transforms can be achieved from these two. As suggested above, they are considered only as a starting point for the integer, reversible, wavelet transform algorithm of the disclosed invention.

It is noted that there is another interesting property in the above two transforms which may not be easily seen. If the values of the signal pixels are represented by a finite number of bits, say one bit or one byte, the same number of bits can be used to represent the result of the forward transform within the computer itself because of the complementary code property. While, from the reconstruction algorithm, the computer will get back the exact original signal through the same complementary code property. This property is called a *Property of Precision Preservation (PPP)* for these wavelets.

It is known that the general values for the high frequency wavelet coefficients are small, and all higher levels of the decomposition also provide generally small values in the high frequency band. This allows the preservation of precision during the computational stage of the wavelet coefficients. Now, the complementary code property, the other aspect of the *PPP* property is a well known characteristic of integer arithmetic as done by the computer. Consider the computation of the difference of two integers given as $c = b - a$ and the inverse computation of $a = b - c$. The nature of the computation within the computer can be specified as follows:

$$c_m = \begin{cases} b-a \text{ if } -2^{q-1} \le b-a < 2^{q-1}-1 \\ -2^q+b-a \text{ if } b-a \ge 2^{q-1} \\ 2^q+b-a \text{ if } b-a < -2^{q-1} \end{cases}$$

and the inverse is

$$a_m = \begin{cases} b-c_m \text{ if } -2^{q-1} \le b-a < 2^{q-1}-1 \\ -2^q+b-c_m \text{ if } b-c_m \ge 2^{q-1} \\ 2^q+b-c_m \text{ if } b-c_m < -2^{q-1} \end{cases}$$

where the m subscript indicates the internal representation, and the range of the integers a, b, c is $[-2^{q-1}, 2^{q-1}-1]$. The internal representation of c_m when it is outside the range, its appearance is as a two's complement number, so the representation may not be the same as the external representation of c. However, the same complementary code for the a_m will cause the internal representation to be identical to the external representation of a. For example, if we let b = 2 (00000010) and a = -127 (10000001) then c_m has the internal binary value of (10000001) when q=4. With a value of -127 for c_m the inverse value for a_m will just be *a*.

In fact, for Example 2, this property is obviously true. While for Example 1, if the range of the pixel values is within a finite number of bits, say q, we can only use q bits as the working unit, which means the value of transform coefficients will also be within the interval with length 2^q, say $[-2^{q-1}, 2^{q-1} - 1]$. Due to the nature of computation on a machine, most machines will implement (2.1)-(2.2) automatically as follows (the complementary code property):

$$d_k^1=\begin{cases} c_{2k}^0-c_{2k+1}^0, & if\ -2^{q-1}\le c_{2k}^0-c_{2k+1}^0<2^{q-1}, \\ c_{2k}^0-c_{2k+1}^0-2^q, & if\ c_{2k}^0-c_{2k+1}^0\ge 2^{q-1}, \\ 2^q+(c_{2k}^0-c_{2k+1}^0), & if\ c_{2k}^0-c_{2k+1}^0<-2^{q-1}. \end{cases} \tag{2.6}$$

$$c_k^1=\begin{cases} Int(\frac{d_k^1}{2})+c_{2k+1}^0, & if\ -2^{q-1}\le Int(\frac{d_k^1}{2})+c_{2k+1}^0<2^{q-1}, \\ Int(\frac{d_k^1}{2})+c_{2k+1}^0-2^q, & if\ Int(\frac{d_k^1}{2})+c_{2k+1}^0\ge 2^{q-1}, \\ Int(\frac{d_k^1}{2})+c_{2k+1}^0+2^q, & if\ Int(\frac{d_k^1}{2})+c_{2k+1}^0<-2^{q-1}. \end{cases} \tag{2.7}$$

While the reconstruction algorithm (2.3) and (2.5) will be implemented by the computer itself as

$$c_{2k+1}^0=\begin{cases} c_k^1-Int(\frac{d_k^1}{2}), & if\ -2^{q-1}\le c_k^1-Int(\frac{d_k^1}{2})<2^q, \\ 2^q+\left(c_k^1-Int(\frac{d_k^1}{2})\right), & if\ c_k^1-Int(\frac{d_k^1}{2})<-2^{q-1}, \\ \left(c_k^1-Int(\frac{d_k^1}{2})\right)-2^q, & if\ c_k^1-Int(\frac{d_k^1}{2})>-2^{q-1}. \end{cases} \tag{2.8}$$

$$c_{2k}^0=\begin{cases} d_k^1+c_{2k+1}^0, & if\ -2^{q-1}\le d_k^1+c_{2k+1}^0<2^{q-1}, \\ d_k^1+c_{2k+1}^0+2^q, & if\ d_k^1+c_{2k+1}^0<-2^{q-1}, \\ d_k^1+c_{2k+1}^0-2^q, & if\ d_k^1+c_{2k+1}^0\ge -2^{q-1}. \end{cases} \tag{2.9}$$

It is obvious that (2.8)-(2.9) are just the reverse of (2.6)-(2.7). It is also easy to see that if we properly take advantage of the bound in the coefficient size mentioned above, the algorithm can be implemented using a minimal amount of storage.

The following are examples which give motivation for our new approach.

Example 3: A (2.6) wavelet transform by integer calculation (2).

This transformation is similar to using the following analysis filters:

n	-2	-1	0	1	2	3
$\tilde{h}_n$	0	0	1/2	1/2	0	0
$\tilde{g}_n$	-1/16	-1/16	1/2	-1/2	1/16	1/16

(a) Decomposition

Decomposition starts with Example 1 at step (1) and (2), and then upgrades the high frequency component at step (3):

(1) Compute

$$d_k^{1,0}=c_{2k}^0-c_{2k+1}^0,\ k=0,\dots,M_1-1.$$

(2) Compute

$$c_k^1=Int(\frac{d_k^{1,0}}{2})+c_{2k+1}^0,\ \ k=0,\dots,N_1-2,$$

$$c_{N_1-1}^1=\begin{cases}Int(\frac{d_{M_1-1}^{1,0}}{2}),+c_{N-1}^0, & if\ N\ is\ an\ even\ number,\\ c_{N-1}^0, & if\ N\ is\ an\ odd\ number;\end{cases}$$

(3) Compute

$$\begin{cases}d_0^1=Int(\frac{c_0-c_1^1}{4}),+d_0^{1,0}\\ d_k^1,=Int(\frac{c_{k-1}^1-c_{k+1}^1}{4})-d_k^{1,0},k=1,\dots,M_1-2,\end{cases}$$

and then, if N is even, calculate

$$d^{1}_{M_1-1}=Int(\frac{c^{1}_{N_1-2}-c^{1}_{N_1-1}}{4})-d^{1,0}_{M_1-1},$$

else, calculate

$$d^{1}_{M_1-1}=Int(\frac{c^{1}_{N_1-3}-c^{1}_{N_1-1}}{4})-d^{1,0}_{M_1-1}.$$

(b) Reconstruction

The reconstruction algorithm is identical to the decomposition algorithm, except it is now running "backwards".

(1) Compute

$$\begin{cases} d^{1,0}_0=Int(\frac{c^1_0-c^1_1}{4})-d^1_0 \\ d^{1,0}_k=Int(\frac{c^1_{k-1}-c^1_{k+1}}{4})-d^1_k, k=1,\ldots,M_1-2, \end{cases}$$

and then, if N is even, calculate

$$d^{1,0}_{M_1-1}=Int(\frac{c^{1}_{N_1-2}-c^{1}_{N_1-1}}{4})-d^{1,1}_{M_1-1},$$

else calculate

$$d^{1,0}_{M_1-1}=Int(\frac{c^{1}_{N_1-3}-c^{1}_{N_1-1}}{4})-d^{1,1}_{M_1-1},$$

(2) If N is an even number, compute

$$c^0_{2k+1}=c^1_k-Int(\frac{d^{1,0}_k}{2}),\ k=0,\ldots,N_1-1;$$

or, if N is an odd number, we have

$$c^0_{2k+1}=c^1_k-Int(\frac{d^{1,0}_k}{2}),\ k=0,\ldots,N_1-2;$$

$$c^0_{N-1}=c^1_{N_1}.$$

(3) Compute

$$c^0_{2k}=d^{1,0}_k+c^0_{2k+1},\ k=0,\dots,M_1-1.$$

We see in step (2)-(3) above, that they are just the same as shown for the reconstruction of the (2.2)-wavelet transform (Example 1).

Example 4: A (1,3)-wavelet transform by integer calculation.

The following nonlinear transform is a variation of the transform which uses biorthogonal analysis filters:

n	-1	0	1
$\tilde{h}_n$	1	0	0
$\tilde{g}_n$	1/4	-1/2	1/4

(a) Decomposition

This decomposition starts with the *Lazy wavelet* at step (1) and upgrades the high frequency component at step (2):

(1) Set

$$c^1_k=c^0_{2k},\ k=0,\dots,N_1-1;$$

$$d^1_k=c^0_{2k+1},\ k=0,\dots,M_1-1.$$

(2) If N is an even number, calculate

$$\begin{cases} d^1_k=Int\left(\dfrac{c^1_k-c^1_{k+1}}{2}\right)-d^{1,0}_k,\ k=0,\dots,M_1-2, \\ d^1_{M_1-1},=c^1_{N_1-1}-d^{1,0}_{M_1-1}. \end{cases}$$

Otherwise, if N is an odd number, calculate

$$d_k^1 = Int\left(\frac{c_{2k}^0 + c_{2k+2}^0}{2}\right) - c_{2k+1},\ k=0,\dots,M_1-1.$$

(b) Reconstruction

(1) Set

$$c_{2k}^0 = c_k^1,\ k=0,\dots,N_1-1;$$

(2) If N is an even number, calculate

$$\begin{cases} c_{2k+1}^0 = Int\left(\frac{c_{2k}^0 + c_{2k+2}^0}{2}\right) - d_k^1,\ k=0,\dots,M_1-2, \\ c_{N-1}^0, = c_{N-2}^0 - d_{M_1-1}^1. \end{cases}$$

Otherwise, if N is an odd number, calculate

$$c_{2k+1}^0 = Int\left(\frac{c_{2k}^0 + c_{2k+2}^0}{2}\right) - d_k^1,\ k=0,\dots,M_1-1.$$

Example 5: (5,3)-wavelet transform by integer calculation.

This transformation is also similar in function to using the biorthogonal analysis filters. It is given by

n	-2	-1	0	1	2
$\tilde{h}_n$	-1/8	1/4	3/4	1/4	-1/8
$\tilde{g}_n$	1/4	-1/2	1/4	0	0

(a) Decomposition

This decomposition starts with Example 3 at step (1) and upgrade low frequency components at step (2):

(1) Set

$$c_k^{1,0}=c_{2k}^0,\ k=0,\dots,N_1-1;$$

If N is an even number, calculate

$$\begin{cases} d_k^1=Int\left(\frac{c_{2k}^0+c_{2k+2}^0}{2}\right)-c_{2k+1}^0,\ k=0,\dots,M_1-2, \\ d_{M_1-1}^1=c_{N-2}^0-c_{N-1}^1. \end{cases}$$

Otherwise, if N is an odd number, calculate

$$d_k^1=Int\left(\frac{c_{2k}^0+c_{2k+2}^0}{2}\right)-c_{2k+1},\ k=0,\dots,M_1-1.$$

(2) If N is an even number, compute

$$\begin{cases} c_0^1=c_0^{1,0}-Int\left(\frac{d_0^1}{2}\right), \\ c_k^1=c_k^{1,0}-Int\left(\frac{d_{k-1}^1+d_k^1}{4}\right),\ k=1,\dots,N_1-2, \\ c_{N-1}^1=c_{N_1-2}^{1,0}-Int\left(\frac{d_{N_1-2}^1+d_{N_1-1}^1}{4}\right). \end{cases}$$

Otherwise, if N is an odd number, calculate

$$\begin{cases} c_0^1=c_0^{1,0}-Int\left(\frac{d_0^1}{2}\right), \\ c_k^1=c_k^{1,0}-\frac{d_{k-1}^1+d_k^1}{4},\ k=1,\dots,N_1-2, \\ c_{N_1-1}^1=c_{N_1-1}^{1,0}-Int\left(\frac{d_{M_1-1}^1}{2}\right). \end{cases}$$

(b) Reconstruction

(1) Compute

$$c_0^0=c_0^1+Int(\frac{d_0^1}{2}),$$

$$c_{2K}^0=c_k^1+Int(\frac{d_{k-1}^1+d_k^1}{4}),\ k=1,\ldots,N_1-2,$$

Then, if N is even, calculate

$$c_{N-2}^0=c_{N_1-1}^1+Int(\frac{d_{N_1-2}^1+d_{N_1-1}^1}{4}).$$

else calculate

$$c_{N-1}^0=c_{N_1-1}^1+Int(\frac{d_{M_1-1}^1}{2}).$$

(2) Compute

$$c_{2k+1}^0=Int(\frac{c_{2k}^0+c_{2k+2}^0}{2})-d_k^1,\ k=0,\ldots,M_1-2,$$

Then, if N is even, calculate

$$c_{N-1}^0=c_{N-2}^0-d_{M_1-1}^1.$$

The *PPP* property for Examples 1-2 mentioned at the end of the previous section is also applicable for these three examples. It is obvious these three transformations are not really linear, but they are similar to the one using the corresponding filters given above. Especially, the filters in Example 3 and Example 5 belong to, with minor modification, the group of the best biorthogonal filters for image compression.

Also, from the above three examples, we can note that if we begin with integer (linear or nonlinear) wavelet transformations and then use some proper upgrading formulas, we can get other, much better integer, wavelet transformations for image compression.

<u>Lifting Scheme and Integer Biorthogonal Filtering</u>

The *Lifting scheme*, discussed by W. Sweldens in "The Lifting Scheme: A Custom-Designed Construction of Biorthogonal Wavelet", Applied and Computational Harmonic Analysis, Vol. 3, No. 2, April 1996, is a recently developed approach for constructing biorthogonal wavelets with compact support. It can be used, with minor modifications, to create integer biorthogonal wavelet transformations. The following is an adaptation of the lifting scheme.

Definition 1. The set of filters {h, h, g, g}, a set of biorthogonal filters if the following formula is satisfied:

$$\forall \omega \in \mathbf{R} : \tilde{m}(\omega)\overline{M'(\omega)} = 1 .$$

where $$m(\omega) = \begin{bmatrix} h(\omega) & h(\omega+\pi) \\ g(\omega) & g(\omega+\pi) \end{bmatrix},$$

and $$h(\omega) = \sum_k h_k e^{-k\omega} \; and \; g(\omega) = \sum_k g_k e^{-k\omega},$$

and *similarly for* $\tilde{m}(\omega)$, $\tilde{h}(\omega)$ *and* $\tilde{g}(\omega)$.

The following lemma is the main result of the lifting scheme [1] reported as corollary 6 in that paper.

Lemma 1. Take an initial set of finite biorthogonal filters {h, $\tilde{h}^o$, g^o, $\tilde{g}$}, then a new set of finite biorthogonal filters {h, $\tilde{h}$, g, $\tilde{g}$} can be found as

$$\tilde{h}(\omega) = \tilde{h}^0(\omega) + \tilde{g}(\omega)\overline{s(2\omega)}$$
$$g(\omega) = g^0(\omega) - h(\omega)s(2\omega) .$$

Similarly if we take {h^o, $\tilde{h}$, g, $\tilde{g}^o$} as an initial set of biorthogonal filters, a new set of biorthogonal filters {h, $\tilde{h}$, g, $\tilde{g}$} can be found as can be found as

$$h(\omega)=h^0(\omega)+g(\omega)\overline{s(2\omega)}$$
$$\tilde{g}(\omega)=\tilde{g}^0(\omega)-\tilde{h}(\omega)\tilde{s}(2\omega).$$

Here s(ω) is a trigonometric polynomial and the corresponding filter s is finite, and so is $\tilde{s}(\omega)$. Actually, regarding the filters (4.1) is equivalent to

$$\tilde{h}_k=\tilde{h}_k^0+\sum_l \tilde{g}_{k+2l}\tilde{s}_l$$
$$g_k=g_k^0-\sum_l h_{k-2l}\tilde{s}_l$$

or

$$h_k=h_k^0+\sum_l g_{k+2l}\tilde{s}_l$$
$$\tilde{g}_k=\tilde{g}_k^0-\sum_l \tilde{h}_{k-2l}\tilde{s}_l \qquad (4.26)$$

Next we use the lifting scheme with minor modifications to create an integer, nonlinear, quasi-biorthogonal, wavelet algorithm. Suppose $[C_n^0]$ is a original signal, $[C_n^1]$ and $[d_n^1]$ are again its low and high frequency decomposition parts, obtained by using the filters {h, $\tilde{h}$, g, $\tilde{g}$}.

If we use filters {$\tilde{h}$, $\tilde{g}$} for decomposition (analysis), the corresponding decomposition algorithm is

$$\begin{cases} C_k^1=\alpha_c\sum_n C_n^0\tilde{h}_{n-2k}, \\ d_k^1=\alpha_d\sum_n C_n^0\tilde{g}_{n-2k}. \end{cases}$$

While the reconstruction algorithm will be

$$c_n^0=2\sum_k\left(\frac{c_k^1 h_{n-2k}}{\alpha_c}+\frac{d_k^1 g_{n-2k}}{\alpha_d}\right),$$

related to the synthesis filter {h,g}. Here, parameters α_c and α_d are positive constants with $\alpha_c \bullet \alpha_d=2$. For example, in the situation of regular biorthogonal decomposition and reconstruction, $\alpha_c=\alpha_d=\sqrt{2}$; and for Example 1 through Example 5 above, $\alpha_c=1$ and $\alpha_d=2$.

If the set of filters {h, $\tilde{h}$, g, $\tilde{g}$} is from {h, $\tilde{h}^0$, g^0, $\tilde{g}$} by (4.2b), then decomposition can be accomplished as follows:

1. Calculate

$$\begin{cases} c_k^{1,0}=\alpha_c\sum_n c_n^0\tilde{h}_{n-2k}^0, \\ d_k^1=\alpha_d\sum_n c_n^0\tilde{g}_{n-2k}. \end{cases}$$

2. Calculate

$$c_k^1=c_k^{1,0}+\frac{\alpha_c}{\alpha_d}\sum_l d_{k-l}^1 s_l. \qquad (4.4)$$

The relative reconstruction scheme will be:

1. Calculate

$$c_k^{1,0}=c_k^1\frac{\alpha_c}{\alpha_d}\sum_l d_{k-l}^1 s_l \qquad (4.5)$$

2. Calculate

$$c_n^0=2\sum_k\left(\frac{c_k^{1,0}h_{n-2k}}{\alpha_c}+\frac{d_k^1 g_{n-2k}^0}{\alpha_d}\right). \qquad (4.6)$$

Here, equations (4.3) and (4.6) are just the wavelet (inverse) transforms using biorthogonal filters {h, $\tilde{h}^0$, g^0,

$\tilde{g}$}. While (4.4) and (4.5) are forward and backward upgrading formulas.

Similarly if the set of filters {h, $\tilde{h}$, g, $\tilde{g}$} is from the initial set of filters {h^0, $\tilde{h}$, g, $\tilde{g}^0$} by using (4.2b), the relative decomposition is:

1. Calculate

$$\begin{cases} c_k^1 = \alpha_c \sum_n c_n^0 h_{n-2k} \\ d_k^{1,0} = \alpha_d \sum_n c_n^0 \overline{g}_{n-2k}^0. \end{cases}$$

2. Calculate

$$d_k^1 = d_k^{1,0} \frac{\alpha_c}{\alpha_d} \sum_1 c_{k-1}^1 s_1$$

The reconstruction scheme is:

1. Calculate

1. Calculate

$$d_k^{1,0} = d_k^1 \frac{\alpha_c}{\alpha_d} \sum_1 c_{k-1}^1 s_1$$

2. Calculate

$$c_n^0 = 2\sum_k \left(\frac{c_k^{1,0} h_{n-2k}}{\alpha_c} + \frac{d_k^1 g_{n-2k}^0}{\alpha_d} \right).$$

<u>Corollary 4.1.</u> *Suppose biorthoganal filters* $\{h, \tilde{h}, g, \tilde{g}\}$ *are from initial filters* $\{h, \tilde{h}^0, g^0, \tilde{g}\}$ *by the lifting scheme* (4.1*a*) *or* (4.2*a*). *If the decomposition and reconstruction by filters* $\{h, \tilde{h}^0, g^0, \tilde{g}\}$ *can be accomplished only by integer calculation, such as Example* 2, *we also can create a corresponding integer wavelet decomposition and reconstruction scheme which is very "close" to the original one by using filters* $\{h, \tilde{h}, g, \tilde{g},\}$. *Here the word "close" means that the difference of the two decompostion schemes is just some rounding error, and this rounding error will l corrected by the integer reconstruction scheme.*

In fact, if $\{c_k^{1,0}\}$ *and* $\{d_k^1\}$ *are integer after* (4.3), *we can calculate* $\{c_k^1\}$ *by*

$$c_k^1 = c_k^{1,0} + Int\left(\frac{\alpha_c}{\alpha_d} \sum d_{k-l} s_l\right)$$

instead of (4.4). Here Int(x), as described in Section 2, is an arbitrary rounding up function which satisfies $x-1 \le Int(x) \le x+1$. It is obvious that (4.7) is very close to (4.4), and the exact reconstruction scheme can easily be obtained from

$$c_k^{1,0} = c_k^1 - Int\left(\frac{\alpha_c}{\alpha_d} \sum_l d_{k-l} s_l\right)$$

and (4.6). There will be a similar result, if the set of biorthogonal filters $\{h, \tilde{h}, g, \tilde{g}\}$ is obtained from the initial set of filters $\{\dot{h}, \tilde{h}, \dot{g}, \tilde{g}^0\}$ by using (4.2b).

Except for the example shown in the *Lazy wavelet* (Example 2), most standard biorthogonal wavelet forms cannot be performed directly by integer, even for one of the simplest wavelets, the *Harr wavelet*. However, if the parameters α_c, and α_d are properly chosen and the transform algorithms, such as Example 1 and Example 3, are slightly changed, a variation of the original biorthogonal wavelet transforms with respect to the set of filters {h, h, g, g} is created. On the other hand, the parameters should be also chosen carefully to guarantee that only addition and shift operations are needed by the algorithm.

If the set of filters {h, $\tilde{h}$, g, $\tilde{g}$} is obtained from a set of filters {$\mathring{h}$, $\tilde{h}$, g, $\tilde{\mathring{g}}$} by the lifting scheme, and the set {$\mathring{h}$, $\tilde{h}$, g, $\tilde{\mathring{g}}$} is also obtained from a filter set {$\mathring{h}$, $\tilde{\mathring{h}}$, $\mathring{g}$, $\tilde{\mathring{g}}$}, one can repeatedly use Corollary 1 to get a "close" integer wavelet transformation.

The Correction Method for Creating Integer Wavelet Transforms

Another approach for obtaining integer wavelets is using the so-called *Correction method*. The motivation of this method is from the S+P transform. The lifting scheme for generating biorthogonal wavelets can be considered as a special case of the correction method. From this can be derived complicated filters with fast decomposition and reconstruction algorithms.

Assuming a simple integer wavelet transform, such as Examples 1 through 3, the decomposition and reconstruction scheme of which can be formulated as follows:

Decomposition

$$c_1^{1,0}=df_c\left(\left\{c_n^0\right\}\right)$$
$$d_1^{1,0}=df_d\left(\left\{c_n^0\right\}\right) \tag{5.1}$$

Reconstruction

$$c_n^0=rf\left(\left\{c_1^{1,0}\right\},\left\{d_k^{1,0}\right\}\right) \tag{5.2}$$

Here, (5.1) and (5.2) can be the same as (4.3) and (4.6) or other algorithms.

In general, after the above decomposition, one may not be satisfied with the result. There may still be some correlation among the high pass components because of the aliasing from the low pass components, or the low pass components do not carry enough of the expected information from the original signal. Hence, one could make an improvement by putting some correction part on the high pass components or low pass components. There are many ways to accomplish this. However, for the sake of the integer calculation, it is preferable to use following correction method. To make a correction for the high pass part, the corresponding formula would be:

$$d_2^1 = d_k^{1,0} - Int(dc_{2,k}^1) k = \cdots, 0, 1, 2 \cdots \qquad (5.3)$$

Here, dc_k^1 is a correction quantity for d_k^1

$$d_k^1 = \sum_{j=1}^{S_1} \sigma_i c_{k+1}^{1,0} + \sum_{j=1}^{1} \tau_j d_{k+j}^{1,0}, k = \cdots, 0, 1, 2, \cdots \qquad (5.4)$$

and $\{\sigma_i\}_{j=s_0}^{s_l}$ and $\{\tau_j\}_{j=1}^{T}$ are given parameters which have been chosen for the user's purpose such as reducing the redundancy among high pass components or some other special requirement. To preserve the integer calculation, any entries in both $\{\sigma_i\}_{i=s_0}^{s_l}$ and $\{\tau_j\}_{j=1}^{T}$ should be rational numbers with denominators being powers of 2.

From (5.1), (5.3) and (5.4), it is easy to see the perfect reconstruction algorithm can be

$$d_k^{1,0} = d_k^1 + Int(dc_k), k = \cdots, m, m-1, m-2 \cdots, \qquad (5.5)$$

combined with (5.2).

As mentioned above, the Lifting scheme is a special condition of the correction method. Examples 3 through 5 can also be considered as the examples of this method. We next give an example of the Correction method which cannot be included in the group of Lifting scheme, and also which does not result in a closed form of compact support for biorthogonal filters.

Example 6: S+P transform, which is similar to using following analysis filters.

n	-2	-1	0	1	2	3
$\tilde{h}_n$	0	0	1/2	1/2	0	0
$\tilde{g}_n$	-1/16	-1/16	15/32	-17/32	7/32	-1/32

While the synthesis filters do not have compact support, the S+P transform can be implemented as follows:

(a) Decomposition

(1) Take the decomposition step of Example 1, that is, compute

$$d_k^{1,0} = c_{2k}^0 - c_{2k+1}^0,\ k=0,1,\cdots,M_1-1;$$

and

$$c_k^1 = Int\left(\frac{d_k^{1,0}}{2}\right) + c_{2k+1}^0,\ k=0,\cdots,N_1-2,$$

$$c_{n_1-1}^1 = \begin{cases} Int\left(\dfrac{d_{M_1-1}^{1,0}}{2}\right) + c_{N-1}^0, \\ c_{n-1}^0 \end{cases}$$

(2) Correction Step: Define $S_0 = -1$, $S_1 = 1$, $T = 1$ and

$$\sigma_{-1} = -\frac{1}{4},\ \sigma_0 = -\frac{1}{6},\ \sigma_1 = \frac{1}{6};$$

$$\tau_1 = \frac{1}{4}.$$

and now compute

$$d_0^1 = d_0^{1,0} - Int\left(\frac{c_0^1 - c_1^1}{4}\right);$$

$$d_k^1 = d_k^{1,0} - Int\left(\frac{2c_{k-1}^1 + c_k^1 - 3c_{k+1}^1 - 2d_{k+1}^{1,0}}{8}\right), k=1,\cdots,M_1-2;$$

$$d_{M_1-1}^1 = d_{M_1-1}^{1,0} - Int\left(\frac{c_{M_1-2}^1 - c_{M_1-1}^1}{4}\right).$$

(b) Reconstruction

(1) Compute

$$d_{M_1-1}^{1,0} = d_{M_1-1}^1 + Int\left(\frac{c_{M_1-2}^1 - c_{M_1-1}^1}{4}\right)$$

$$d_k^{1,0} = d_k^1 + Int\left(\frac{2c_{k-1}^1 + c_k^1 - c_{k+1}^1 - 2d_{k+1}^{1,0}}{8}\right), k=M_1-2,\cdots,1;$$

$$d_0^{1,0} = d_0^1 + Int\left(\frac{c_0^1 - c_1^1}{4}\right).$$

(2) If N is an even number, compute

$$c_{2k+1}^0 = c_k^1 - Int\left(\frac{d_k^1}{2}\right), k=0,\cdots,N_1-1$$

or, if N is an odd number, we have

$$c_{2k+1}^0 = c_k^1 - Int\left(\frac{d_k^1}{2}\right), k=0,\cdots,N_1-2,$$

$$c_{N-1}^0 = c_{N_1}^1.$$

(3) Compute

$$c_{2k}^0 = d_k^1 + c_{2k+1}^0, k=0,\cdots,M_1-1.$$

<u>Boundary Conditions</u>

There are two issues dealing with boundary filtering if the Lifting scheme or the Correction method is used to generate the integer wavelet transformations. The first is how to process the boundaries which occur in the start-up wavelet transformations. The second is how to deal with the boundaries in the deductive formula. If the boundaries in the start-up wavelet transform have already been established, then those in the upgrading formula are relatively easy to establish. For the *Lifting scheme*, the boundaries in both steps should be processed in the same way. While, for the Correction method, according to (5.3)-(5.4), one has more choices to process boundaries in the second step. Therefore, the process by which the boundaries in the start-up wavelet transformations are established is discussed. Assume compact supported biorthogonal wavelets.

Suppose the original signal is $\{C_n^0\}_{n=0}^{N}$. For creating integer biorthogonal wavelet transformations, use the following symmetric extension:

(1) If current biorthogonal filters have even length, the boundaries of the signal are extended as $C_{-k}^0 = C_{k-1}^0,\ k=1,2,\cdots;$

(2) If the filters have odd length, the following extension is performed
$C_{-k}^0 = C_k^0,\ k=1,2,\cdots.$

Examples 1 through 5 use the boundaries given above. In Example 6, the start up wavelet transform uses the above boundaries but in the upgrading step, another boundary filtering is used. In addition, for arbitrarily sized images or signals, one can use the same technique described in the above examples to deal with this condition.

As mentioned earlier, for many applications, lossless image compression is more important than lossy compression. The integer wavelet transforms described above provide the opportunity to compress without loss. It is also obvious that the integer wavelet algorithms can be used wherever ordinary wavelets are used, especially in signal and image compression. However, for most computers, the integer wavelet transform is much faster than other wavelets and it uses much less memory.

Peak Signal to Noise Ratio (PSNR) Controlled Compression

Peak Signal to Noise Ratio (PSNR) is a widely used quality measurement. PSNR controlled compression allows users to choose their desired PSNR for the compressed image. In each of the compression methods set forth herein, a user can selectively set the PSNR and the desired compression ratio, as well as the initial quantization and threshold levels for each quadrant of wavelet coefficients, to obtain the desired image quality.

For example, the wavelet map of FIG. 3 shows a total of 10 regions (quadrants). Each of these ten quadrants can have two additional parameters associated with them. The parameters define the quantization and threshold values for that particular quadrant. Since there are three planes for color (only one for gray level) the maximum number of parameters that the user can control is 60 -- 10 for quantization and 10 for thresholding for each of the three color layers. In the case of a gray level image, there are only 20 parameters.

If a compression ratio, or a quality factor which indirectly defines a compression ratio, is specified, then the user wants the compression ratio to remain identical over the changes in the parameters. In order to accomplish this, two parameters are monitored: the compression ratio and PSNR (peak signal to noise ratio). The PSNR is defined as PSNR = 20 $\log_{10}$ (X/MSE), where the X is the average absolute value of the pixels in the compressed image and MSE is the mean squared error measured between the compressed and original image. Holding the compression ratio constant, the PSNR needs to increase to improve image quality. The way to increase the PSNR is to reduce the MSE.

An iterative method can be used to adjust parameters to achieve the desired PSNR. The step are as follows:

(a) Pick an initial parameter setting P_0;

(b) Quantize the wavelet coefficients with P_0 and calculate the corresponding PSNR;

(c) If the PSNR is close to the desired one, stop and output the coded file; otherwise, get an adjusted vector ΔP_0 and set $P_0 \leftarrow P_0 + \Delta P_0$, go to step (b).

Progressive Decomposition

Progressive decompression allows users to decode images at varying degrees of resolution, starting from the lowest resolution and progressing to the highest resolution. The advantage of this feature is that users can download small pieces of the coded file and view the image at lower resolution to determine if they want to download the whole image. Progressive decomposition can be used with any of the decompression methods previously disclosed herein. Progressive decomposition is accomplished according to the following steps:

(a) Input the lowest bandpass component C^1 of the coded file and reconstruct the lowest resolution image I^0;

(b)Display image I^0;

(c) If the user is not satisfied with the image quality or the resolution is big enough for stop; otherwise, go to step (d);

(d) Input the lowest three band-pass components HD^1, VD^1, and DD^1 successively in the current image file. Reconstruct the new image I^1 from C^1, HD^1, VD^1, and DD^1. Let $I^0 = I^1$; go to step (b).

Image Map Editor

The image map editor creates an image map over a compressed image file. This permits an image compressed according to one of the methods set forth herein to be easily integrated into a web page using an http link. A user selects one or several areas of compressed image, assigns one or more http links to the areas. The image map editor calculates the coordinates of the areas and outputs the HTML associate with the image. The user can add such information into program source code. Following is an example of such image map:

```
<EMBED SRC="cow.cod" type="image/cis-cod"
WIDTH="257" poly= "44, 45, 103, 78, 103, 86, 54,
86, 54, 78",
href="http://www.infinop.com"></EMBED>
```

Non-Uniform Image Compression

The present invention allows a user to perform non-uniform image compression. Essentially, non-uniform compression is accomplished by dividing an image into one or more rectangles, each representing a matrix of image pixels. Each rectangle can be compressed by any of the methods disclosed herein.

For instance, referring to the compression method of FIG. 8, integrating the non-uniform compression feature with the method allows a user to partition the image into several parts with different interests. The user can then compress these areas with different image and/or compression qualities. The parts can have any shape.

The non-uniform compression feature can be incorporated in to the method of FIG. 8 as follows. Steps 100-102 are performed. Then, the user creates bitmap matrices defining the partitioned areas. Each area is then wavelet transformed. Different quantizations are then applied to the different areas according to the transformed matrices obtained above.

Split and Merge Wavelet Algorithm for Big Image Compression

This algorithm allows users to compress large images by partitioning them into smaller pieces. The key is to divide the original image into several smaller pieces and compress/decompress them separately by using overlap and de-overlap technique. With this technique, the individually compressed pieces are equivalent to compressed whole image. The user does not see any edge effects in the decompressed image, which normally occur with conventional split and merge methods.

Also, with this algorithm, users can selectively decompress the whole image or choose a specific part to decompress according to an image map created during the compression phase. The algorithm is preferably implemented as a software program executing on a general purpose computer.

There are two ways to compress an image by splitting it: automatically or interactively. The automatic approach is transparent to users since the algorithm will automatically split to the image according to the characteristics of the computer used to perform the compression. Using the automated method, the algorithm first detects the size of the source image and the memory size of the host computer. Next, the image is split into several pieces with a predetermined number of pixels overlapping according to the image size and computer's memory. Overlapping pixels are those that appear in more than one piece of the split image.

Each piece of image is compressed in order according to any of the methods disclosed herein from the image resource.

The split image is decompressed as follows. First, the headers of the compressed image pieces are read to determine their order and compression parameters, such as quantization thresholds and decomposition levels. Next, each piece of the image is decompressed and deoverlapped. Merge all pieces together in the proper place for display.

Using the interactive method, a user can indicate how many blocks they want to divide the image into and how many pixels they want for overlap. To compress an image according to this approach, the size of the source image is first detected. Then, the user's choice for the number of blocks and number of overlapping pixels is entered. Next, the image is divided into the pieces according to the user's choice and the size of the image. Finally, the individual pieces are compressed according to one of the methods disclosed herein.

The interactively split image is decompressed as follows. First, the header of the coded image is read. Next, an image map is displayed for the user to look at what the image context is about. The user can then choose to display entire image or a specific piece of image. If user chooses to display a single piece of image, the algorithm finds the position of this coded piece and decompresses it. If the user instead chooses to display the entire image, the algorithm decompress each piece of image and de-overlaps it. All pieces are then merged together in the appropriate display location.

Example A, below, shows further technical details related to the present invention.

While specific embodiments of the present invention have been shown and described, it will be apparent to this skilled in the art that the disclosed invention may be modified in numerous ways and may assume many embodiments other than the preferred form specifically set out and described above. Accordingly, it is intended by the appended claims to cover all modifications of the invention which fall within the true spirit and scope of the invention.

EXAMPLE A

1.0 Quality Compression Optimization

The following sections deal with the improvement of the compressed image quality. As stated in the report introduction, we believe this is the last significant major problem to be addressed in the still imagery compression system.

1.1 Introduction

The issue is the mechanism which can be used to improve the quality of the image based upon the nature of the parameters used in the compression process. We think this is an important issue, and in our experimentation, we have found fairly striking results. For example, if we hold the compression ratio constant, we can compress an image and obtain on a subjective scale of C, where

- A is no observable defect,
- B is observable but not noticeable,
- C is quite noticeable, but not distracting from the image,
- D is very noticeable and detracts from the image,
- E is unacceptable.

If we now change the parameters by a hand optimization, we find we can change the subjective evaluation from a C to a B with the compression ratio left alone. We believe this is significant, in that it guarantees a compression system which is tailored for each image independently, rather than have each image compressed by the same set of parameters irrespective of the image content.

To review the process of compression, consider the following figure, Figure 49.

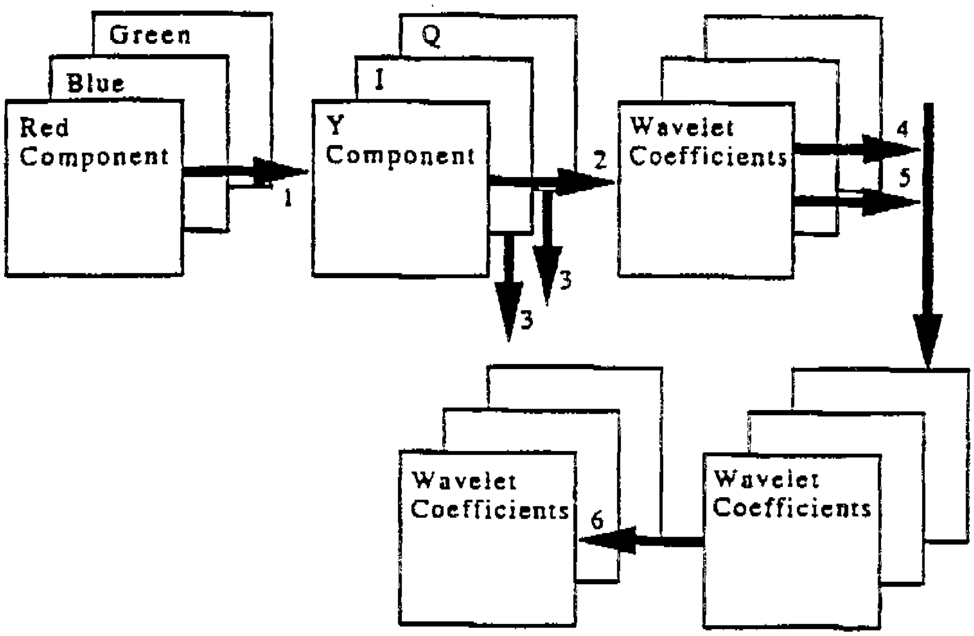


Figure 49
RGB Image in 24 Bit Color Depth Showing Transform to YIQ

Figure 49 shows the relationship between the RGB and YIQ color representation and the processes which take place in the algorithm as this data is transformed into the final lossy set of wavelet coefficients.. The processes which apply to the imagery are shown as heavy arrows with the process identifier shown as part of the arrow. If no process number is present, then the arrow is just a passive link between processes and data. The important point to note is that in color imagery, we deal with three images (one luminescence, and two color planes). The Y component is critical, while the IQ components are less sensitive to error introduced by the compression system.

The processes shown in Figure 49 are as follows:

- (1) Transform the RGB format into YIQ format, using long integers. This format is only an approximation of the YIQ format.

- (2) Transform the YIQ planes into a wavelet decomposition using our own integer wavelet transform. This produces the result shown in Figure 50, but for each plane.

- (3) We have shown an alternative process which is an optional down sampling for the IQ color planes. This down sampling may be done once or twice to produce two image planes either one fourth or one sixteenth the size of the original plane. If this process is to be done, it will be accomplished prior to the wavelet transform of process (2).

- (4) Here the first step in the loss occurs. The wavelet coefficients are now quantized to a number of levels depending upon which quadrant is being processed, and the desired compression or quality factor.

- (5) Simultaneously with step (4), the wavelet coefficients are being matched against threshold values, and if the values are less than the established threshold values specified, then the resultant value is set to zero.

- (6) The last step in the process is to entropy compress the resultant coefficients using either Arithmetic, Run Length, or Huffman, of Huffman and Run Length combined. The key issue is the amount of compression desired against the invested time required to get that level of compression.

The issue now, irrespective of the down sampling or not of the IQ components, is the fact that we process the three planes into five levels in a decomposition as shown in Figure 50. From this figure, one can see a total of 16 regions (quadrants) which are defined by the numbers 1 through 16. Each of these sixteen quadrants have two additional parameters associated with them. The parameters define the quantization and threshold values for that particular quadrant. Since there are three planes for color (only one for gray level) the maximum number of parameters that the user can control is 96 -- 16 for quantization and 16 for thresholding for each of the three color layers. In the case of a gray level image, there are only parameters for one layer.

Our experience has shown the parameters for an image are very sensitive in some cases, and not in other cases. In order to measure this sensitivity, we generated the variance of the wavelet coefficients in the 16 quadrants. Table 1 provides these values for each of the three planes.

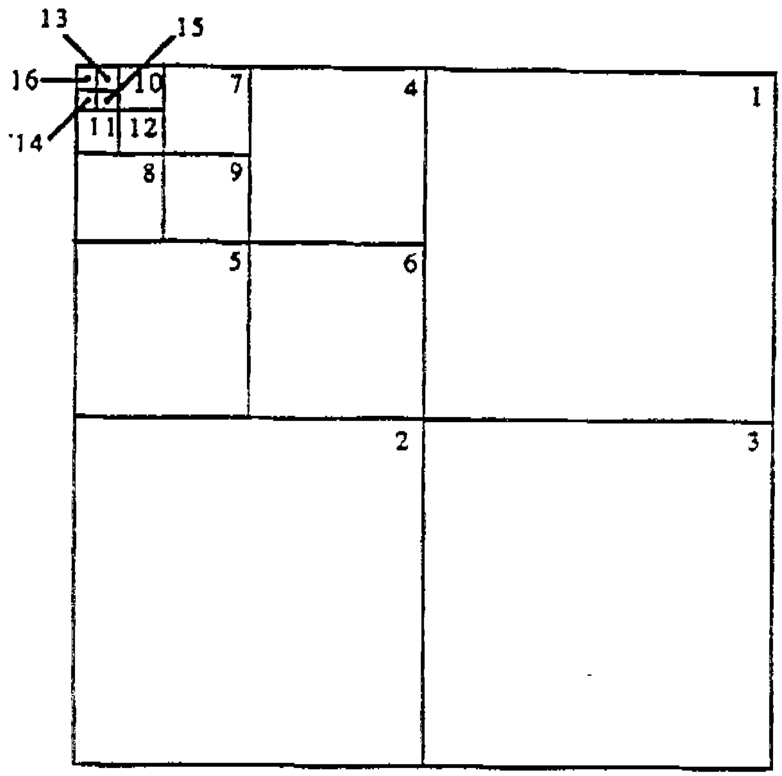


Figure 50
Decomposition of a Plane into Five Levels

Image	Lenna Wavelet Variance			Tulips Wavelet Variance		
Quadrant	Y	I	Q	Y	I	Q
1	13	4.7	1.4	164.2	27.3	6.6
2	6	3.2	1.1	134.6	24.9	5.0
3	1	1.3	0.6	28.9	4.6	1.1
4	56	5.8	1.7	276.1	57.7	12.9
5	23	4.5	1.1	274.3	67.3	12.4
6	10	2.8	0.8	124.8	22.7	5.0
7	128	11.4	2.6	225.7	73.2	13.7
8	55	7.4	1.1	298.1	107.3	16.2
9	37	3.3	1.0	114.2	35.5	6.8
10	253	27.5	4.1	174.8	73.9	10.7
11	75	11.8	1.5	285.1	128.2	14.9
12	64	6.5	1.1	107.9	47.8	6.7
13	499	41.7	10.1	135.4	75.6	8.7
14	138	11.2	2.5	245.0	144.0	11.2
15	156	12.2	2.0	89.3	49.1	5.1
16	1,4161	1,281.8	295.6	7,408.6	690.4	77.9

Table 1
Wavelet Coefficient Variance by Numbered Quadrant for Two Images

The clear information to be gained by the numbers in Table 1 is the images are quite different, and the quantization or threshold levels set for all images will only be an approximate solution at best. The optimum solution would be to have the quantization and threshold values set according to the variance values. Such settings could be found in a table with various ranges, and for each such range, the parameters of interest could be defined. This would give a more optimal solution, but still not the optimal solution. In order to get optimality, one would need to search over the variable space using the near optimal settings to find the actual best values for the parameters.

As the values in Table 1 grow, the implication is a finer mesh size for quantization. That is, the number of bits one wishes to allocate to the output could be varied by quadrant. Those quadrants with large variances will utilize more bits, while those with low variants will utilize fewer bits. In this way, the number of bits resulting from quantization will remain the same, but their allocation will differ depending upon the nature of the image.

1.2 Approach

The solution to the problem posed in the previous section is to determine how the ninety-six parameters interact with one another. The problem is a bit more sophisticated than just measuring parameters, however. The issue is with each parameter change, the compression ratio will change. If a compression ratio, or a quality factor which indirectly defines a compression ratio, is specified, then the user wants the compression ratio to remain identical over the changes in the parameters. In order to accomplish this, there are two parameters which we must monitor: PSNR (peak signal to noise ratio which is defined to be PSNR = 20 $\log_{10}$ (X/MSE) where the X is the average absolute value of the pixels in the after image and MSE is the mean squared error measured between the before and after image) and the compression ratio. The compression ratio must be held constant, and the PSNR needs to increase, and the way to increase the PSNR is to reduce the MSE.

The difficulty with this system as described is in many cases, small changes in the parameters introduce significant changes in the MSE. Also, we believe, the parameters are not independent. We have also seen images where the parameters can be changed in one way, then altered, and the results are exactly the same. This indicates the optimal value is not a single point, but rather something like a plane with little slope.

1.3 Status

We have established the rules under which the optimal solution will need to exist, and are at the moment writing software to measure the variance within the Lightning Strike environment. Once this is done, we will being examining many images to see how close we can determine the optimal parameters for a defined set of variances.

CIS-2 Image Compression Algorithm

HONGYANG CHAO

Part I: Brief Review of LSIC 3.0

1. Introduction

CIS-2 (temporary name), which has been being used in **Lightning Strike 3.0** image compression software, is a wavelet based image compression algorithm. CIS-2 has following inventions:

- Integer reversible wavelet algorithm with Property of Precision Preservation;
- Subband oriented quantization and related entropy coding;
- Wavelet lossless compression for color and gray images;
- Progressive transmission algorithm for color bit compression;
- Progressive transmission and decompression algorithms;
- Non-uniform image compression algorithm;
- Quality based wavelet coefficient quantization tables;
- Attached optional post-processing filters;
- Image map editor;
- Optional peak signal noise ratio controlled compression;
- Special split and merge wavelet compression algorithm for very big image compression without any boundary effects ;
- Image dependent parameter optimization .

2. Main steps of the algorithm

In Lsic 3.0, three kinds of different image compression methods are included:

Method 1: Quality controlled wavelet based compression

Method 2: Color bit depth compression

Method 3: Wavelet lossless compression

Section 2.1-2.3 will give brief description of above method. The details will discuss later.

2.1. Main step of method 1

Figure 1 and Figure 2 give the flow charts of the image compression and decompression of method1 respectively. Every step in both compression and decompression has lot of details, which will be described later.

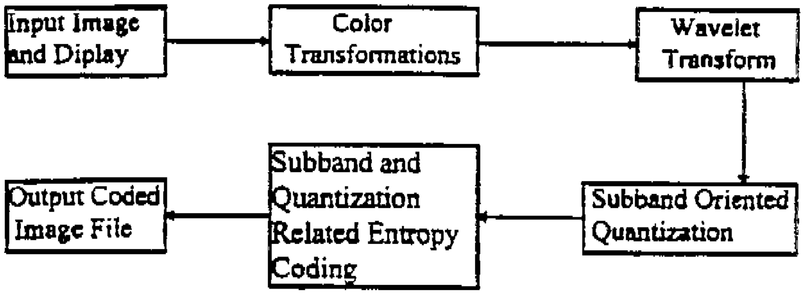


Figure 1: Compression flow chart for Method 1

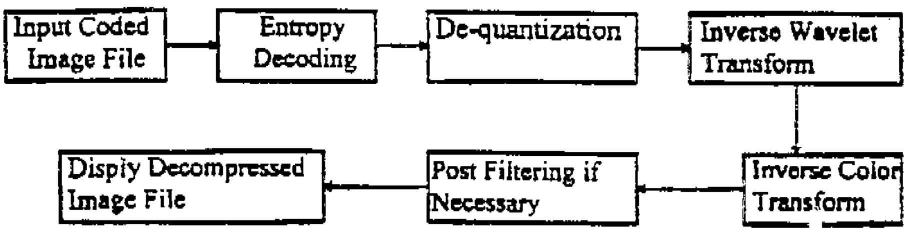


Figure 2: Decompression flow chart for Method 1

2.2. Main step of method 2

This method based on using less number of colors to approximately represent original images. Following figure gives the main step of the algorithm for the compression and decompression.

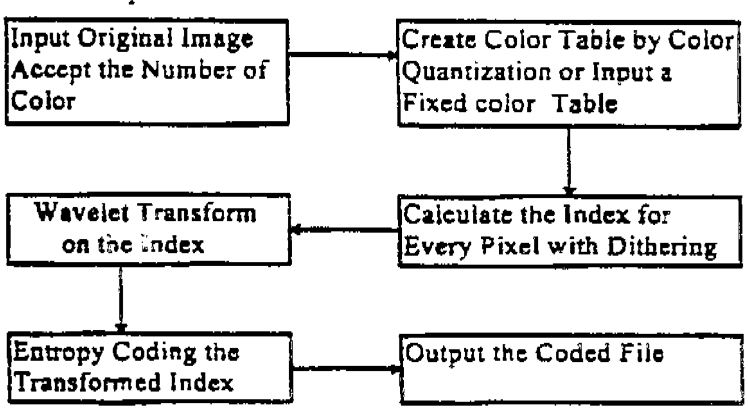


Figure 3: Compression flow chart for Method 2

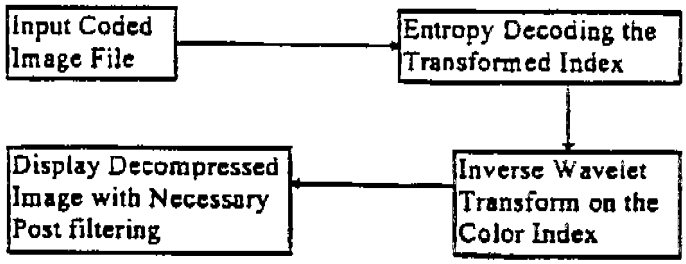


Figure 4: Decompression flow chart for Method 2

2.3. Main steps of method 3

The main steps of method 3 is almost as same as method 1. However, at the every step we use different methods. Following are its compression and decompression flow charts:

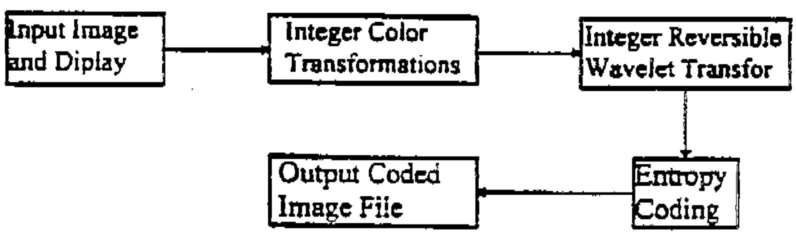


Figure 5: Compression flow chart for Method 3

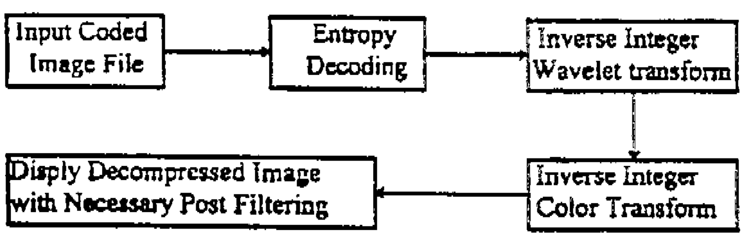


Figure 6: Decompression flow chart for Method 3

3. Brief Description for Other Features

Most Features (or inventions) are included in above three compression methods. Following is the brief introduction for some inventions list above.

3.1. Progressive decompression

This algorithm allows users to decode images from the lowest resolution to highest resolution. The advantage of this feature is that users can download small piece of the coded file and view the image at lower resolution to determine if they want to download the whole image.

Steps:

(a) Input the lowest pass component LL^0 of the coded file and reconstruct the lowest resolution image I^0;

(b) Display image I^0. If the user doesn't like it or the resolution is big enough for , stop; otherwise, go to next step;

(c) Input the lowest three band-pass components HL^0, LH^0 and HH^0 successively in the current coded file. Reconstruct the new image I^1 from LL^0, HL^0, LH^0 and HH^0. Let $I^0 = I^1$, go to step (b).

3.2. Non-uniform image compression

The algorithm allows users to divide the image into several parts with different interests and compress these areas with different qualities. The areas can have any shape. This algorithm is only available for method 1.

Original image goes though all of the procedure except quantization part, which follows the steps below:

(a) Creating the bitmap matrices related to the areas chosen by the user;

(b) Wavelet transform to every bitmap matrix;

(c) Different quantization in different areas according to the transformed matrices obtained above step.

3.3. Peak Signal Noise Ratio (PSNR) controlled compression

Peak Signal Noise Ratio (PSNR) is an image quality measurement used by most professional people. PSNR controlled compression allows users to choose their desired PSNR for the compressed image.

The related algorithm is an iterated system:

(a) Picking an initial parameter setting P_0;

(b) Quantize the wavelet coefficients with P_0 and calculate the corresponding PSNR;

(c) If the PSNR is close to desired one, stop and output the coded file; otherwise, get an adjusted vector ΔP_0 and set $P_0 \leftarrow P_0 + \Delta P_0$, go to step (b);

3.4. Attached optional post-processing filters

Users can choose any number of following processing filters at their compressing time. The desired results can stored in the coded image file, and, anyone who decompress the coded file will see the same result immediately.

- Sharpening images
- Smoothing images
- Improving the visual quality
- Brightening the images

3.5. Image map editor

Image Map Editor create a image map over Lsic3.0 compressed image file. User selects one or several areas of compressed image, assigns the http links to the areas, then, Image Map Editor calculates the coordinates of the areas and outputs a HTML associate with the image. User can add such information into his/her source code.

Following is an example of such image map:

<EMBED SRC="cow.cod" type = "image/cis-cod" WIDTH= "257" poly= "44, 45, 103, 78, 103, 86, 54, 86, 54, 78", href= "http://www.infinop.com"></EMBED>

3.6. Split and merge wavelet algorithm for very big image compression

This algorithm allows users to compress very big image by an ordinary machine. The key is to divide the original image into several smaller pieces and compress/decompress them separately by using overlap and dis-overlap technique. With this technique, the compression/decompression piece by piece is equal to compress/decompress the whole image together, which means users won't see any edge effect at decompressed image which appears at general split method.

Also, with this algorithm, users can either decompress the whole image or choose the specific part to decompress according to a image map we create for the division .

3.7. Image dependent optimized parameter setting

This algorithm allows user to get the best (or almost best) image quality at the desired compression ratio by choosing image related parameter setting.

3.8. Integer reversible wavelet algorithm with PPP property

See attached unpublished paper titled as "An Approach to Fast Integer Reversible Wavelet Transformations for Image Compression"

3.9. Integer color transformation

This algorithm is a integer reversible transform which has been used in lossless color image compression (Method 3) for Lsic 3.0.

The algorithm transform RGB color components to a new set of color components Y-Nb-Nr:

Forward transform RGB to Y-Nb-Nr:

$$Y = G + \mathrm{Int}\left(\tfrac{R+B}{2}\right),$$
$$Nb = B - \mathrm{Int}\left(\tfrac{Y}{2}\right),$$
$$Nr = R - \mathrm{Int}\left(\tfrac{Y}{2}\right).$$

Inverse transform Y-Nb-Nr to RGB:

$$R = Nr + \mathrm{Int}\left(\tfrac{Y}{2}\right),$$
$$B = Nb + \mathrm{Int}\left(\tfrac{Y}{2}\right),$$
$$G = Y - \mathrm{Int}\left(\tfrac{R+B}{2}\right).$$

3.10. Subband related Quantization and entropy coding

This entropy coding method is just designed for wavelet based image compression. The main idea is to take the advantage of different quantization at different subbands and encode each band according to its content. This method reduce the coding cost greatly.

CIS-1 Image Compression Algorithm

HONGYANG CHAO*
Computer and Information Science Inc.

1. Introduction

CIS-1, which has been being used in **Lightning Strike** image compression software, is a wavelet based image compression algorithm. CIS-1 has following advantages:

- Reach almost optimal compression ratio;
- Keep the major characteristics as more as possible. In other words, it reduce insignificant components gradually according to human visual system, so that people can still accept the image quality at the extremely high compression ratio;
- Fast.

2. Main steps of the algorithm

Figure 1 and Figure 2 give the flow charts of the image compression and decompression respectively. Every step in both compression and decompression has lot of details, which will be described later.

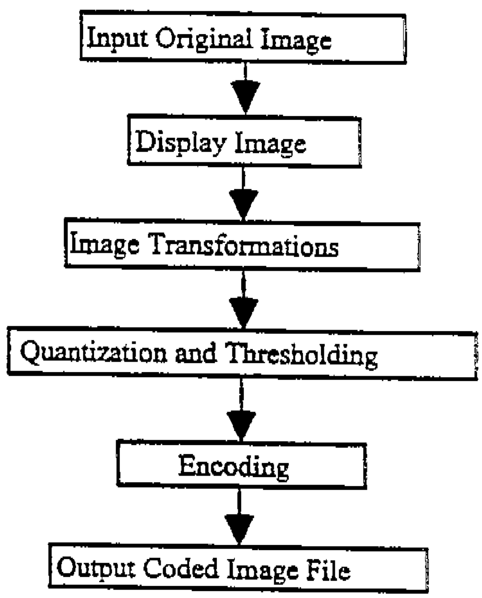


Figure 1: Image compression

*

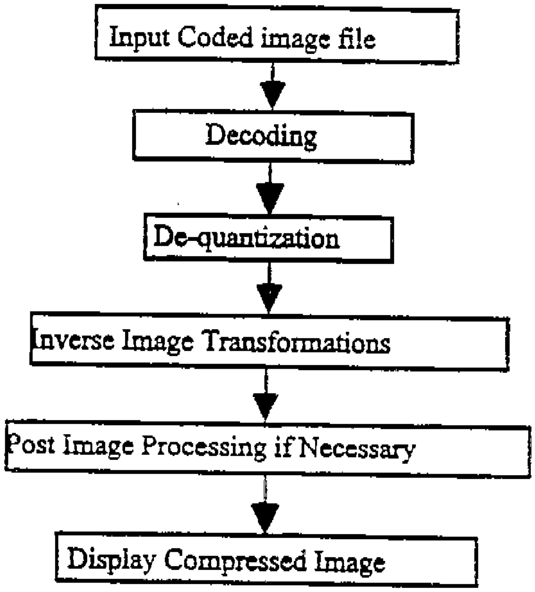


Figure 2: Image decompression

3. Image Compression

For the compression, we only have to describe three parts: image transformations, quantization and thresholding, and entropy coding.

3.1. Image Transformations

The image transformations involved in this algorithm include color transform (for color images) and wavelet decomposition (for both gray level images and color images).

(a) Color transform

In general, input color images are based on RGB color model, such as TIFF or BMP images. In order to get high compression ratio, it is better to change RGB color model to other color models, such as YIQ or YUV models.

RGB to YIQ:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

RGB to YUV:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.439 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

(b) Wavelet decomposition

The purpose of wavelet transform is to represent the original image by different basis to achieve the objective of decorrelation. There are a lot of wavelets which can be used in this step. In CIS-1, we use a wavelet which results in the following algorithm: Suppose $C^0 = [c^0_{jk}]_{M \times N}$ $(j = 0, \cdots, M-1;\ \ k = 0, \cdots, N-1)$ is original image, where M and N are integers which have the common factor 2^L (L is a positive integer). After one-level wavelet decomposition, we will get four parts as shown in figure 3.

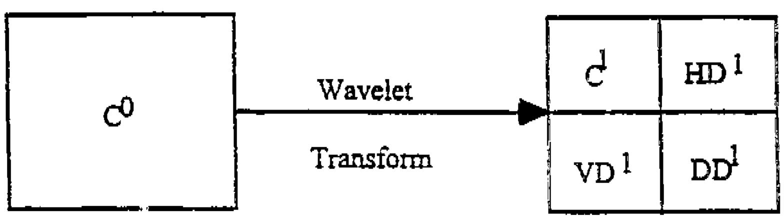


Figure 3. Wavelet image decomposition

We call $\mathbf{C}^1 = [c^1_{jk}]$ $(j = 0, \cdots, \frac{M}{2}-1;\ k = 0, \cdots, \frac{N}{2}-1)$ the blurred image of $\mathbf{C}^0$, $\mathbf{HD}^1$ the horizontal high frequency part of $\mathbf{C}^0$, $\mathbf{VD}^1$ the vertical high frequency part, and $\mathbf{DD}^1$ the diagonal ones. Setting $\mathbf{C}^0 = \mathbf{C}^1$, we can repeat the same procedure L times or until the size of the new blurred image $\mathbf{C}^1$ is small enough. Therefore, we only have to give the algorithm for one level decomposition:

(1) Let $\overline{\mathbf{C}}^0 = r\mathbf{C}^0$, $r > 0$ is a factor which can be changed for different needs.

(2) Transform for image columns:

- For $k = 0, \cdots, N-1$, calculate

$$\begin{cases} \tilde{d}^1_{0k} = \dfrac{\bar{c}^0_{1k} - \bar{c}^0_{0k}}{2}, \\ \tilde{d}^1_{jk} = \dfrac{1}{4}(\bar{c}^0_{2j-1,k} - 2\bar{c}^0_{2j,k} + \bar{c}^0_{2j+1,k}), \quad j = 1, \cdots, \frac{M}{2} - 1. \end{cases} \tag{3.1.1}$$

- For $k = 0, \cdots, N-1$, calculate

$$\begin{cases} \tilde{c}^1_{0k} = \tilde{c}^0_{1,k} - \dfrac{\tilde{d}^1_{0k} + \tilde{d}^1_{1,k}}{2}, \\ \tilde{c}^1_{jk} = \tilde{c}^0_{2j+1,k} - \dfrac{\tilde{d}^1_{jk} + \tilde{d}^1_{j+1,k}}{2}, \ j = 1, \cdots, \frac{M}{2} - 2, \\ \tilde{c}^1_{\frac{M}{2}-1,k} = \tilde{c}^0_{N-1,k} - \tilde{d}^1_{\frac{M}{2}-1,k}. \end{cases} \quad (3.1.2)$$

(3) Transform for rows:

- For $j = 0, \cdots, \frac{M}{2} - 1$, computing

$$\begin{cases} hd^1_{j,0} = \dfrac{\tilde{c}^1_{j1} - \tilde{c}^1_{j0}}{2}, \\ hd^1_{jk} = \dfrac{1}{4}(\tilde{c}^1_{j,2k-1} - 2\tilde{c}^1_{j,2k} + \tilde{c}^1_{j,2k+1}), \ k = 1, \cdots \frac{N}{2} - 1. \end{cases} \quad (3.1.3)$$

and

$$\begin{cases} c^1_{j0} = \tilde{c}^1_{j,1} - \dfrac{hd^1_{j0} + hd^1_{j1}}{2}, \\ c^1_{jk} = \tilde{c}^1_{j,2k+1} - \dfrac{hd^1_{j,k} + hd^1_{j,k+1}}{2}, \ k = 1, \cdots, \frac{M}{2} - 2, \\ c^1_{j,\frac{N}{2}-1} = \tilde{c}^1_{j,N-1} - hd^1_{j,\frac{N}{2}-1}. \end{cases} \quad (3.1.4)$$

- For $j = 0, \cdots, \frac{M}{2} - 1$, computing

$$\begin{cases} dd^1_{j,0} = \dfrac{\tilde{d}^1_{j,1} - \tilde{d}^1_{j,0}}{2}, \\ dd^1_{j,k} = \dfrac{1}{4}(\tilde{d}^1_{j,2k-1} - 2\tilde{d}^1_{j,2k} + \tilde{d}^1_{j,2k+1}), \ k = 1, \cdots, \frac{N}{2} - 1. \end{cases} \quad (3.1.5)$$

and

$$\begin{cases} vd^1_{j0} = \tilde{d}^1_{j,1} - \dfrac{dd^1_{j0} + dd^1_{j1}}{2}, \\ vd^1_{jk} = \tilde{d}^1_{j,2k+1} - \dfrac{dd^1_{j,k} - dd^1_{j,k+1}}{2}, \ k = 1, \cdots, \frac{M}{2} - 2, \\ vd^1_{j,\frac{N}{2}-1} = \tilde{d}^1_{N-1,k} - dd^1_{j,\frac{N}{2}-1}. \end{cases} \quad (3.1.6)$$

(4) $\mathbf{C}^1 = \left[c^1_{j,k}\right]$, $\mathbf{HD}^1 = \left[hd^1_{j,k}\right]$, $\mathbf{VD}^1 = \left[vd^1_{j,k}\right]$ and $\mathbf{DD}^1 = \left[dd^1_{j,k}\right]$, $j = 0, \cdots, \frac{M}{2} - 1$; $k = 0, \cdots, \frac{M}{2} - 1$.

Remark: If it is necessary, we also can use matrix multiply

$$\text{Wavalet Coefficient Image of } l \text{ levels} = \mathbf{W}_l \mathbf{C}^0 \mathbf{W}_l^T.$$

Here, W_l is the transform matrix for l level wavelet decomposition.

3.2. Thresholding and Quantization

Both thresholding and Quantization allow us to reduce accuracy with which the wavelet coefficients are represented when converting the wavelet decomposition to an integer representation. This can be very important in image compression, as it tends to make many coefficients zeros--especially those for high spatial frequencies.

After L level, for example L=3, wavelet decomposition, we get the wavelet coefficients of the original image as plotted in Figure 4:

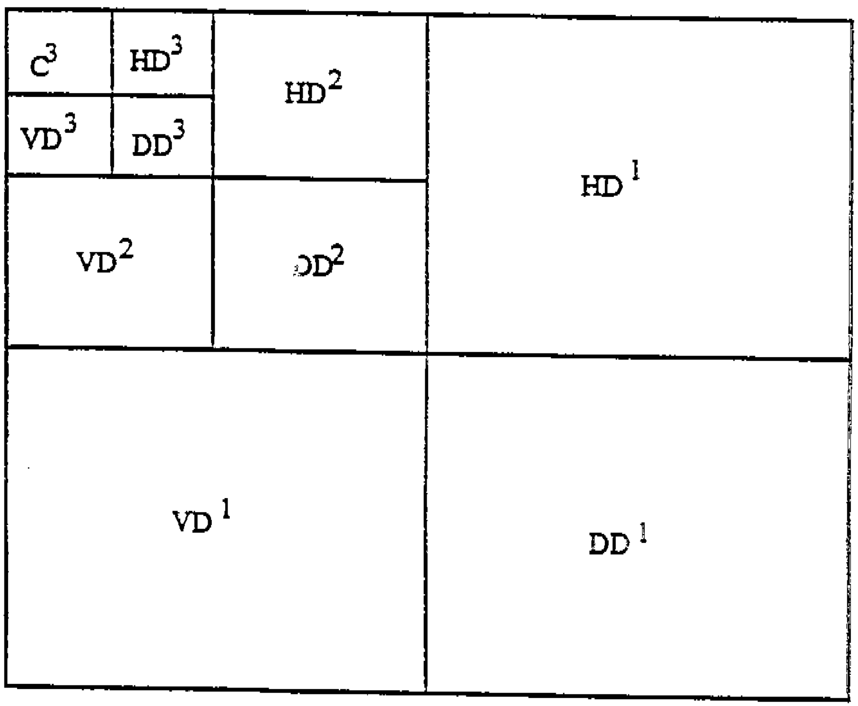


Figure 4. L=3 wavelet coefficients distribution

(a) Thresholding

In algorithm CIS-1, we use multilevel uniform thresholding method: Let

$$\mathbf{T} = (t_1, \cdots, t_L, t_{L+1})$$

be the chosen thresholds, where t_l is the threshold for l th $(l = 1, \cdots, L)$ level and t_{L+1} is a threshold for blurred image $\mathbf{C}^L$. Thresholding is to set every entry in the blocks $\mathbf{C}^L$,

$\mathbf{HD}^l$, $\mathbf{VD}^l$ and $\mathbf{DD}^l$ $(l=1,\cdots,L)$ to be zeros if its absolute value is not greater than the corresponding threshold.

Remark. For color image, we can have three threshold vectors which correspond three different color bands, such as Y, I and Q.

(b) Quantization

Quantization is to scale the wavelet coefficients and truncate them to integer values. In CIS-1, we use the quantization table shown in Table 1 to implement it.

q^1_{HD} q^1_{VD} q^1_{DD}	q^2_{HD} q^2_{VD} q^2_{DD}	••• ••• •••	q^L_{HD} q^L_{VD} q^L_{DD}	q_C^{L+1}

Table 1. Quantization table

Here, the entries q^l_{HD} are quantization factors for blocks $\mathbf{HD}^l$ $(l=1,\cdots,L)$, q^l_{VD} and q^l_{DD} for blocks $\mathbf{VD}^l$ and $\mathbf{DD}^l$ $(l=1,\cdots,L)$ respectively, and the factor q_C^{L+1} is for the most blurred image $\mathbf{C}^L$. All the factors are integers between 0 and 255. The quantization scheme for the block $\mathbf{HD}^l$ $(l=1,\cdots,L)$ is

$$\overline{hd}^l_{j,k} = round(\frac{hd^l_{j,k} \bullet q^l_{HD}}{\max^l_{HD}}),\ j=0,\cdots,\tfrac{M}{2^l}-1;\quad k=0,\cdots,\tfrac{N}{2^l}-1. \quad (3.2.1)$$

Here, $\overline{hd}^l_{j,k}$ $(j=0,\cdots,\frac{M}{2^l}-1;\quad k=0,\cdots,\frac{N}{2^l}-1)$ are quantized wavelet coefficients in block $\mathbf{HD}^l$ $(l=1,\cdots,L)$,

$$\max\nolimits^l_{HD} = \max_{\substack{0\le j\le (M/2^l-1)\\ 0\le k\le (N/2^l-1)}} \left(\left|hd^l_{j,k}\right|\right),$$

and the function $round(x)$ gives the nearest integer of x. The scheme of quantization for other blocks are the similar to (3.2.1).

Remark. For color image, as the same as thresholding, we can have three separate quantization tables for different color bands.

3.3. Entropy Coding

Here, the encoding means the lossless compression for the wavelet coefficients. It is divided into two parts: Coefficient arrangement and entropy coding (Huffman or arithmetic).

4. Decompression

4.1 Decoding

Decoding, just as encoding, can be divided into two parts: Entropy decoding (Huffman or arithmetic), and coefficient rearranging.

4.2 Dequantization

After Decoding, we get quantized wavelet coefficients in $3*L+1$ Blocks. Dequantizing uses the same quantization table as quantizing, and the scheme as follow: for $l=1,\cdots,L$

$$\widetilde{hd}^l_{j,k} = \frac{\overline{hd}^l_{j,k} \bullet \max^l_{HD}}{q^l_{HD}}, \ j=0,\cdots,\tfrac{M}{2^l}-1; \quad k=0,\cdots,\tfrac{N}{2^l}-1. \qquad (4.2.1)$$

(4.2.1) allows us to get the approximate coefficients for the blocks $\mathbf{HD}^l$ $(l=1,\cdots,L)$, which is shown in Figure 4 . The dequantizing scheme for other blocks are similar to (4.1.2).

4.3 Inverse Image Transformations

(a) Wavelet reconstruction

We are going to describe the algorithm for one-level reconstruction which is plotted in Figure 5.

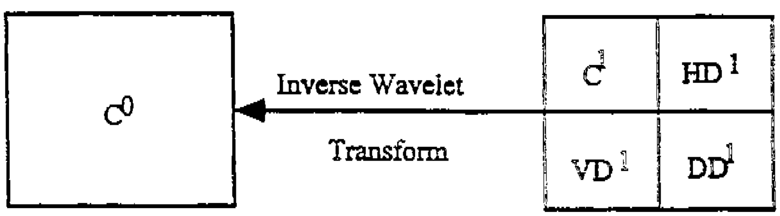


Figure 5. One-level wavelet reconstruction

(1) Inverse transform for rows:

o For $j=0,\cdots,\frac{M}{2}-1$, calculate

$$\begin{cases} \tilde{d}^1_{j,1} = vd^1_{j,0} + \dfrac{dd^1_{j,0}+dd^1_{j,1}}{2} \\ \tilde{d}^1_{j,2k+1} = vd^1_{jk} + \dfrac{dd^1_{j,k}-dd^1_{j,k+1}}{2}, \quad k=1,\cdots,\frac{N}{2}-2, \\ \tilde{d}^1_{N-1,k} = vd^1_{j,\frac{N-2}{2}} + dd^1_{j,\frac{N-2}{2}}. \end{cases} \qquad (4.3.1)$$

and

$$\begin{cases} \tilde{d}^1_{j,0} = \tilde{d}^1_{j,1} - 2dd^1_{j,0}, \\ \tilde{d}^1_{j,2k} = \dfrac{(\tilde{d}^1_{j,2k-1} + \tilde{d}^1_{j,2k+1})}{2} - 2dd^1_{j,k}, \quad k = 1,\cdots,\frac{N}{2}-1. \end{cases} \tag{4.3.2}$$

- For $j = 0,\cdots,\frac{M}{2}-1$, calculate

$$\begin{cases} \tilde{c}^1_{j,1} = c^1_{j0} + \dfrac{hd^1_{j0} + hd^1_{j1}}{2}, \\ \tilde{c}^1_{j,2k+1} = c^1_{jk} + \dfrac{hd^1_{j,k} + hd^1_{j,k+1}}{2}, \quad k = 1,\cdots,\frac{N}{2}-2, \\ \tilde{c}^1_{j,N-1} = c^1_{j,\frac{N}{2}-1} + hd^1_{j,\frac{N}{2}-1}. \end{cases} \tag{4.3.3}$$

and

$$\begin{cases} \tilde{c}^1_{j0} = \tilde{c}^1_{j1} - 2hd^1_{j,0}, \\ \tilde{c}^1_{j,2k} = \dfrac{1}{2}(\tilde{c}^1_{j,2k-1} + \tilde{c}^1_{j,2k+1}) - 2hd^1_{jk}, \quad k = 1,\cdots\frac{N}{2}-1. \end{cases} \tag{4.3.4}$$

(2) Inverse transform for column:

- For $k = 0,\cdots,N-1$, calculate

$$\begin{cases} \bar{c}^0_{1,k} = \bar{c}^1_{0k} + \dfrac{\tilde{d}^1_{0k} + \tilde{d}^1_{1k}}{2}, \\ \bar{c}^0_{2j+1,k} = \bar{c}^1_{jk} + \dfrac{\tilde{d}^1_{jk} + \tilde{d}^1_{j+1,k}}{2}, \quad j = 1,\cdots,\frac{M}{2}-2, \\ \bar{c}^0_{N-1,k} = \bar{c}^1_{\frac{M}{2}-1,k} + \tilde{d}^1_{\frac{M}{2}-1,k}. \end{cases} \tag{4.3.5}$$

and

$$\begin{cases} \bar{c}^0_{0k} = \bar{c}^0_{1k} - 2\tilde{d}^1_{0k}, \\ \bar{c}^0_{2j,k} = \dfrac{1}{2}(\bar{c}^0_{2j-1,k} + \bar{c}^0_{2j+1,k}) - 2\tilde{d}^1_{jk}, \quad j = 1,\cdots,\frac{M}{2}-1 \end{cases} \tag{4.3.6}$$

(3) $c^0_{j,k} = \bar{c}^0_{j,k} / r, \quad j = 0,\cdots,M-1; \quad k = 0,\cdots,N-1$. $\mathrm{C}^0 = \left[c^0_{j,k}\right]_{\frac{M}{2}\times\frac{N}{2}}$.

(b) Inverse color transform

For color image, we have to do inverse color transform

- YIQ to RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}.$$

$$\begin{cases}\tilde{d}^1_{j,0} = \tilde{d}^1_{j,1} - 2dd^1_{j,0}, \\ \tilde{d}^1_{j,2k} = \dfrac{(\tilde{d}^1_{j,2k-1} + \tilde{d}^1_{j,2k+1})}{2} - 2dd^1_{j,k}, \quad k = 1,\cdots,\frac{N}{2}-1.\end{cases} \tag{4.3.2}$$

- For $j = 0,\cdots,\frac{M}{2}-1$, calculate

$$\begin{cases}\tilde{c}^1_{j,1} = c^1_{j0} + \dfrac{hd^1_{j0} + hd^1_{j1}}{2}, \\ \tilde{c}^1_{j,2k+1} = c^1_{jk} + \dfrac{hd^1_{j,k} + hd^1_{j,k+1}}{2}, \quad k = 1,\cdots,\frac{N}{2}-2, \\ \tilde{c}^1_{j,N-1} = c^1_{j,\frac{N-1}{2}} + hd^1_{j,\frac{N-1}{2}}.\end{cases} \tag{4.3.3}$$

and

$$\begin{cases}\tilde{c}^1_{j0} = \tilde{c}^1_{j1} - 2hd^1_{j,0}, \\ \tilde{c}^1_{j,2k} = \dfrac{1}{2}(\tilde{c}^1_{j,2k-1} + \tilde{c}^1_{j,2k+1}) - 2hd^1_{jk}, \quad k = 1,\cdots\frac{N}{2}-1.\end{cases} \tag{4.3.4}$$

(2) Inverse transform for column:

- For $k = 0,\cdots,N-1$, calculate

$$\begin{cases}\tilde{c}^0_{1,k} = \tilde{c}^1_{0k} + \dfrac{\tilde{d}^1_{0,k} + \tilde{d}^1_{1,k}}{2}, \\ \tilde{c}^0_{2j+1,k} = \tilde{c}^1_{jk} + \dfrac{\tilde{d}^1_{jk} + \tilde{d}^1_{j+1,k}}{2}, \quad j = 1,\cdots,\frac{M}{2}-2, \\ \tilde{c}^0_{N-1,k} = \tilde{c}^1_{\frac{N-1}{2},k} + \tilde{d}^1_{\frac{N-1}{2},k}.\end{cases} \tag{4.3.5}$$

and

$$\begin{cases}\tilde{c}^0_{0k} = \tilde{c}^0_{1k} - 2\tilde{d}^1_{0k}, \\ \tilde{c}^0_{2j,k} = \dfrac{1}{2}(\tilde{c}^0_{2j-1,k} + \tilde{c}^0_{2j+1,k}) - 2\tilde{d}^1_{jk}, \quad j = 1,\cdots,\frac{M}{2}-1\end{cases} \tag{4.3.6}$$

(3) $c^0_{j,k} = \tilde{c}^0_{j,k} / r, \quad j = 0,\cdots,M-1; \quad k = 0,\cdots,N-1$. $\mathbf{C}^0 = \left[c^0_{j,k}\right]_{\frac{M}{2}\times\frac{N}{2}}$.

(b) Inverse color transform

For color image, we have to do inverse color transform

- YIQ to RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}.$$

- YUV to RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.581 \\ 1.000 & 2.032 & 0.000 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

4.4 Necessary Post Image Processing

(a) Color Quatization

An Approach to Fast Integer Reversible Wavelet Transforms for Image Compression*

Hongyang Chao**

Computer and Information Science Inc.

3401 E. University, Suite 104, Denton, TX 76208

AND

Dept. of Computer Science, Zhongshan Univ.

Guangzhou 510275, P.R. China

Paul Fisher

Computer and Information Science Inc.

3401 E. University, Suite 104, Denton, TX 76208

AND

Dept. of Computer Science, Univ. of North Texas

Denton, TX 76205

Abstract

In this paper, we propose a general method for creating integer wavelet transforms which can be used in both lossless (reversible) and lossy compression of signals and images with arbitrary size. This method allows us to get a series of transformations which are very close to the corresponding biorthogonal wavelet transforms or some non-orthogonal wavelet transforms, but can be calculated with only integer addition and bit-shift operations. In addition, the integer wavelet transforms created in this paper possess a property of precision preservation (PPP). This property is very useful for conserving memory in both compression and decompression, and speed up the whole procedure in some applications. The motivation of this paper comes from the lifting scheme [1] and S+P transform [3].

* This work has been partially supported by the US Navy under SBIR Contract N00039-94-C-0013.

** The author is partially supported by the National Science Foundation of P. R. China and the Science Foundation of Zhongshan University.

1. Introduction

The wavelet transform has proven to be one of the most powerful tools in the field of image compression. Theoretically, the wavelet transformation is lossless, but since all computers have only finite precision, most of transformations are lossy in practice, even when we use floating point calculations. On the other hand, integer calculations are much faster than floating point for virtually all computers; and integer computations are much easier to implement in hardware which is more important in some applications. The memory utilization of integers is also a positive consideration. The difficulty is, if we directly use integers in the wavelet transform and its inverse without some proper considerations, it will cause the loss of accuracy. For some important image applications, the user wants to have complete control of the precision in which the image pixels are represented during the compression process, and thus prefers to have the image compressed from lossless to lossy.

Lossless compression is also very important for images found in such applications as medical and space science. In such situations, the designer of the compression algorithm must be very careful to avoid discarding any information that may be required or even useful at some later point From the academic point of view, it is also very interesting to have a compression scheme which has very fast performance, and which can exactly reconstruct the image when necessary. In addition, it is also very useful to have some wavelet transforms which exhibit the property of precision preservation (PPP), which can be utilized in the computer which has limited precision and limited memory without losing any precision during the computation.

In this paper, we are going to describe two general methods from which one can get the integer wavelet transform desired. All of the wavelet transforms from the methods given in this paper possess the property of precision preservation (PPP). We draw on the work of several other authors who have already contributed to this area [2-3], where some specific examples were developed. However, this paper presents a more general method which allows one to see several new results as well as those presented and acknowledged prior to this work.

This paper is organized as follows: Section 2 and 3 give some examples of integer wavelet transforms. The examples in section 2 are the starting point for our approach, and the examples in Section 3 show the steps and motivation of our general method. Section 4 indicates how one can use the lifting technique to create an integer biorthogonal wavelet transform. The Correction technique to generate more general integer wavelet transforms is described in Section 5. Section 6 describes how to process boundaries in order to apply the integer calculation in finite sized images or signals. In Section 7, we prove the integer wavelet transforms developed by both the lifting and correction method possess the property of precision preservation (PPP). Some example images are also shown in this section. The last section, Section 8, provides the conclusion to this paper.

2. Basic integer wavelet transformations

We provide the following two examples as the starting point for our new method. For the sake of convenience, length, and simplicity, we only discuss the algorithm for a one level decomposition and reconstruction and only for a one dimensional signal. The extension to two dimensions is immediate as the rows and columns can be treated into a sequence of one dimensional signals. For the following examples, assume that $\{c_n^0\}_{n=0}^{N-1}$ is the original signal where the superscript indicates level and the subscript indicates a particular point in the signal. Also, $\{c_n^1\}_{n=0}^{N_1-1}$ and $\{d_n^1\}_{n=0}^{M_1-1}$ are its decomposition parts at the first level. Here

$$\begin{cases} N_1 = \begin{cases} \frac{N}{2}, & \text{if } N \text{ is an even number,} \\ \frac{N+1}{2}, & \text{if } N \text{ is an odd number;} \end{cases} \\ M_1 = N - N_1. \end{cases}$$

$\{c_n^1\}_{n=0}^{N_1-1}$ and $\{d_n^1\}_{n=0}^{M_1-1}$ are its low frequency (l) part and high frequency (h) part, respectively. For multi-levels, we just treat $\{c_n^1\}_{n=0}^{N_1-1}$ as $\{c_n^0\}_{n=0}^{N-1}$ and repeat the procedure again.

<u>Example 1</u>: A (2,2)-wavelet transform by integer calculation.

This transformation is similar to a variation of the Haar wavelet transform which uses low and high pass analysis (decomposition) filters given as:

n	0	1
$\tilde{h}_n$	1/2	1/2
$\tilde{g}_n$	1/2	-1/2

(1) Compute

$$d_k^1 = c_{2k}^0 - c_{2k+1}^0, \quad k = 0, \cdots, M_1 - 1. \tag{2.1}$$

(2) Compute

$$c_k^1 = \mathrm{Int}(\frac{d_k^1}{2}) + c_{2k+1}^0, \quad k = 0, \cdots, N_1 - 2,$$

$$c_{N_1-1}^1 = \begin{cases} \mathrm{Int}(\frac{d_{N_1-1}^1}{2}) + c_{N-1}^0, & \text{if } N \text{ is an even number,} \\ c_{N-1}^0, & \text{if } N \text{ is an odd number.} \end{cases} \tag{2.2}$$

Here, $\mathrm{Int}(x)$ is an arbitrary rounding function which may have different interpretations. For example, $\mathrm{Int}(x)$ can be the integer which is nearest to x, or $\mathrm{Int}(x)$ may be any integer which satisfies $x - 1 < \mathrm{Int}(x) \le x$, etc. It is easy to see that all entries in both $\{c_n^1\}_{n=0}^{N_1-1}$ and $\{d_n^1\}_{n=0}^{M_1-1}$ are integers.

From (2.1)-(2.2), we can easily get the following integer reconstruction algorithm:

(b) Reconstruction

(1) If N is an even number, compute

$$c_{2k+1}^{0} = c_k^{1} - \mathrm{Int}(\frac{d_k^{1}}{2}), \quad k = 0, \cdots, N_1 - 1 \ ; \tag{2.3}$$

or, if N is an odd number, we have

$$\begin{aligned} c_{2k+1}^{0} &= c_k^{1} - \mathrm{Int}(\frac{d_k^{1}}{2}), \quad k = 0, \cdots, N_1 - 2, \\ c_{N-1}^{0} &= c_{N_1}^{1}. \end{aligned} \tag{2.4}$$

(2) Compute

$$c_{2k}^{0} = d_k^{1} + c_{2k+1}^{0}, \quad k = 0, \cdots, M_1 - 1. \tag{2.5}$$

<u>Remark</u> Since (2.1)-(2.6) are not linear because of the rounding operation $\mathrm{Int}(x)$, this means the transformation order becomes significant. For instance, if the decomposition was applied first to the columns and then to the rows, the inverse transformation must be applied first to the rows and then to the columns.

<u>Example 2</u>: Lazy wavelet transform.

The Lazy wavelet transform *does not do anything*. However, this illustrates an important concept. The corresponding inverse transform is nothing else but sub-sampling the even and odd indexed samples. Decomposition and reconstruction can use same formula as follows:

$$\begin{aligned} c_k^{1} &= c_{2k}^{0}, \ k = 0, \cdots, N_1 - 1; \\ d_k^{1} &= c_{2k+1}^{0}, \ k = 0, \cdots, M_1 - 1. \end{aligned}$$

Examples 1 and 2 are not good transforms for image compression, but they are simple. Much better transforms can be achieved from these two. As suggested above, we consider them only as a starting point for our integer, reversible, wavelet transform algorithm.

We must mention that there is another interesting property in the above two transforms which may not be easily seen. If the values of the signal pixels are represented by a finite number of bits, say one bit or one byte, we can still use the same number of bits to represent the result of the forward transform within the computer itself because of the complementary code property. While, from the reconstruction algorithm, the computer will get back the exact original signal through the same complementary code property. We call this property *a Property of Precision Preservation (PPP)* for these wavelets.

It is known that the general values of the high frequency wavelet coefficients are small, and all higher levels in the decomposition also provide generally small values in the high frequency band. This allows the preservation of precision during the computational stage of the wavelet coefficients. Now, the complementary code property, the other aspect of the *PPP* property is a well known characteristic of integer arithmetic as done by the computer. Consider the computation of the difference of two integers given as $c = b\text{-}a$ and the inverse computation of $a = b - c$. The nature of the computation within the computer can be specified as follows:

$$c_m = \begin{cases} b-a & \text{if } -2^{q-1} \le b-a < 2^{q-1}-1 \\ -2^q + b - a & \text{if } b-a \ge 2^{q-1} \\ 2^q + b - a & \text{if } b-a < -2^{q-1} \end{cases}$$

and the inverse is

$$a_m = \begin{cases} b-c_m & \text{if } -2^{q-1} \le b-c_m < 2^{q-1}-1 \\ -2^q + b - c_m & \text{if } b-c_m \ge 2^{q-1} \\ 2^q + b - c_m & \text{if } b-c_m < -2^{q-1} \end{cases}$$

where the m subscript indicates the internal representation, and the range of the integers a, b, c is $\left[-2^{q-1},\ 2^{q-1}-1\right]$. The internal representation of c_m when it is outside the range, its appearance is as a two's complement number, so the representation may not be the same as the external representation of c. However, the same complementary code for the a_m will cause the internal representation to be identical to the external representation of a. For example, if we let $b=2$ (00000010) and $a=-127$ (10000001) then c_m has the internal binary value of (10000001) when $q=4$. With a value of -127 for c_m, the inverse value for a_m will just be a.

In fact, for Example 2, this property is obviously true. While for Example 1, if the range of the pixel values is within a finite number of bits, say q, we can only use q bits as the working unit, which means the value of transform coefficients will also be within the interval with length 2^q, say $\left[-2^{q-1},\ 2^{q-1}-1\right]$. Due to the nature of computation on a machine, most machines will implement (2.1)-(2.2) automatically as follows (the complementary code property):

$$d_k^1 = \begin{cases} c_{2k}^0 - c_{2k+1}^0, & \text{if } -2^{q-1} \le c_{2k}^0 - c_{2k+1}^0 < 2^{q-1}, \\ c_{2k}^0 - c_{2k+1}^0 - 2^q, & \text{if } c_{2k}^0 - c_{2k+1}^0 \ge 2^{q-1}, \\ 2^q + \left(c_{2k}^0 - c_{2k+1}^0\right), & \text{if } c_{2k}^0 - c_{2k+1}^0 < -2^{q-1}. \end{cases} \quad (2.6)$$

$$c_k^1 = \begin{cases} \mathrm{Int}(\frac{d_k^1}{2}) + c_{2k+1}^0, & \text{if } -2^{q-1} \le \mathrm{Int}(\frac{d_k^1}{2}) + c_{2k+1}^0 < 2^{q-1}, \\ \mathrm{Int}(\frac{d_k^1}{2}) + c_{2k+1}^0 - 2^q, & \text{if } \mathrm{Int}(\frac{d_k^1}{2}) + c_{2k+1}^0 \ge 2^{q-1}, \\ \mathrm{Int}(\frac{d_k^1}{2}) + c_{2k+1}^0 + 2^q, & \text{if } \mathrm{Int}(\frac{d_k^1}{2}) + c_{2k+1}^0 < -2^{q-1}. \end{cases} \tag{2.7}$$

While the reconstruction algorithm (2.3) and (2.5) will be implemented by the computer itself as

$$c_{2k+1}^0 = \begin{cases} c_k^1 - \mathrm{Int}(\frac{d_k^1}{2}), & \text{if } -2^{q-1} \le c_k^1 - \mathrm{Int}(\frac{d_k^1}{2}) < 2^q, \\ 2^q + \left(c_k^1 - \mathrm{Int}(\frac{d_k^1}{2})\right), & \text{if } c_k^1 - \mathrm{Int}(\frac{d_k^1}{2}) < -2^{q-1}, \\ \left(c_k^1 - \mathrm{Int}(\frac{d_k^1}{2})\right) - 2^q, & \text{if } c_k^1 - \mathrm{Int}(\frac{d_k^1}{2}) > 2^{q-1}. \end{cases} \tag{2.8}$$

$$c_{2k}^0 = \begin{cases} d_k^1 + c_{2k+1}^0, & \text{if } -2^{q-1} \le d_k^1 + c_{2k+1}^0 < 2^{q-1}, \\ d_k^1 + c_{2k+1}^0 + 2^q, & \text{if } d_k^1 + c_{2k+1}^0 < -2^{q-1}, \\ d_k^1 + c_{2k+1}^0 - 2^q, & \text{if } d_k^1 + c_{2k+1}^0 \ge 2^{q-1}. \end{cases} \tag{2.9}$$

It is obvious that (2.8)-(2.9) are just the reverse of (2.6)-(2.7). It is also easy to see that if we properly take advantage of the bound in the coefficient size mentioned above, the algorithm can be implemented using a minimal amount of storage.

3. More Examples and Additional Analysis

In this section we are going to give more examples which will give some motivation for our new approach.

Example 3: A (2,6)-wavelet transform by integer calculation [2].

This transformation is similar to using following analysis filters

n	-2	-1	0	1	2	3
h_n	0	0	1/2	1/2	0	0
$\tilde{g}_n$	-1/16	-1/16	1/2	-1/2	1/16	1/16

(a) Decomposition

Decomposition starts with Example 1 at step (1) and (2), and then upgrades the high frequency component at step (3):

(1) Compute

$$d_k^{1,0} = c_{2k}^0 - c_{2k+1}^0, \quad k = 0, \cdots, M_1 - 1.$$

(2) Compute

$$c_k^1 = \text{Int}(\frac{d_k^{1,0}}{2}) + c_{2k+1}^0, \quad k = 0, \cdots, N_1 - 2,$$

$$c_{N_1-1}^1 = \begin{cases} \text{Int}(\frac{d_{M_1-1}^{1,0}}{2}) + c_{N-1}^0, & \text{if } N \text{ is an even number,} \\ c_{N-1}^0, & \text{if } N \text{ is an odd number.} \end{cases}$$

(3) Compute

$$\begin{cases} d_0^1 = \text{Int}(\frac{c_0^1 - c_1^1}{4}) - d_0^{1,0} \\ d_k^1 = \text{Int}(\frac{c_{k-1}^1 - c_{k+1}^1}{4}) - d_k^{1,0}, \quad k = 1, \cdots, M_1 - 2. \end{cases}$$

and then, if N is even, calculate

$$d_{M_1-1}^1 = \text{Int}(\frac{c_{N_1-2}^1 - c_{N_1-1}^1}{4}) - d_{M_1-1}^{1,0},$$

else, calculate

$$d_{M_1-1}^1 = \text{Int}(\frac{c_{N_1-3}^1 - c_{N_1-1}^1}{4}) - d_{M_1-1}^{1,0}.$$

(b) Reconstruction

The reconstruction algorithm is identical to the decomposition algorithm, except it is now running "backwards".

(1) Compute

$$\begin{cases} d_0^{1,0} = \text{Int}(\frac{c_0^1 - c_1^1}{4}) - d_0^1 \\ d_k^{1,0} = \text{Int}(\frac{c_{k-1}^1 - c_{k+1}^1}{4}) - d_k^1, \quad k = 1, \cdots, M_1 - 2, \end{cases}$$

and then, if N is even, calculate

$$d_{M_1-1}^{1,0} = \text{Int}(\frac{c_{N_1-2}^1 - c_{N_1-1}^1}{4}) - d_{M_1-1}^{1,1},$$

else, calculate

$$d_{M_1-1}^{1,0} = \text{Int}(\frac{c_{N_1-3}^1 - c_{N_1-1}^1}{4}) - d_{M_1-1}^{1,1}.$$

(2) If N is an even number, compute

$$c_{2k+1}^0 = c_k^1 - \text{Int}(\frac{d_k^{1,0}}{2}), \quad k = 0, \cdots, N_1 - 1 \;;$$

or, if N is an odd number, we have

$$c^0_{2k+1} = c^1_k - \mathrm{Int}(\frac{d^{1,0}_k}{2}), \quad k = 0,\cdots,N_1 - 2,$$

$$c^0_{N-1} = c^1_{N_1}.$$

(3) Compute

$$c^0_{2k} = d^{1,0}_k + c^0_{2k+1}, \quad k = 0,\cdots,M_1 - 1.$$

We see in step (2)-(3) above, that they are just the same as shown for the reconstruction of the (2,2)-wavelet transform (Example 1).

Example 4: A (1,3)-wavelet transform by integer calculation.

The following nonlinear transform is a variation of the transform which uses biorthogonal analysis filters:

n	-1	0	1
$\tilde{h}_n$	1	0	0
$\tilde{g}_n$	1/4	-1/2	1/4

(a) Decomposition

This decomposition starts with the *Lazy wavelet* at step (1) and upgrades the high frequency component at step (2):

(1) Set
$$c^1_k = c^0_{2k},\ k = 0,\cdots,N_1 - 1;$$
$$d^1_k = c^0_{2k+1},\ k = 0,\cdots,M_1 - 1.$$

(2) If N is an even number, calculate

$$\begin{cases} d^1_k = \mathrm{Int}(\dfrac{c^1_k + c^1_{k+1}}{2}) - d^{1,0}_k, & k = 0,\cdots,M_1 - 2, \\ d^1_{M_1-1} = c^1_{N_1-1} - d^{1,0}_{M_1-1}. \end{cases}$$

Otherwise, if N is an odd number, calculate

$$d^1_k = \mathrm{Int}(\frac{c^0_{2k} + c^0_{2k+2}}{2}) - c^0_{2k+1}, \quad k = 0,\cdots,M_1 - 1.$$

(b) Reconstruction

(1) Set
$$c^0_{2k} = c^1_k, \quad k=0,\cdots,N_1 - 1;$$

(2) If N is an even number, calculate

$$\begin{cases} c^0_{2k+1} = \mathrm{Int}(\dfrac{c^0_{2k} + c^0_{2k+2}}{2}) - d^1_k, & k = 0,\cdots,M_1 - 2, \\ c^0_{N-1} = c^0_{N-2} - d^1_{M_1-1}. \end{cases}$$

Otherwise, if N is an odd number, calculate

$$c_{2k+1}^0 = \mathrm{Int}(\frac{c_{2k}^0 + c_{2k+2}^0}{2}) - d_k^1, \quad k = 0, \cdots, M_1 - 1.$$

Example 5: A (5,3)-wavelet transform by integer calculation.

This transformation is also similar in function to using the biorthogonal analysis filters. It is given by

n	-2	-1	0	1	2
$\tilde{h}_n$	-1/8	1/4	3/4	1/4	-1/8
$\tilde{g}_n$	1/4	-1/2	1/4	0	0

(a) Decomposition

This decomposition starts with Example 3 at step (1) and upgrade low frequency components at step (2):

(1) Set $c_k^{1,0} = c_{2k}^0$, $k=0, \cdots, N_1 - 1$;

If N is an even number, calculate

$$\begin{cases} d_k^1 = \mathrm{Int}(\frac{c_{2k}^0 + c_{2k+2}^0}{2}) - c_{2k+1}^0, \quad k = 0, \cdots, M_1 - 2, \\ d_{M_1-1}^1 = c_{N-2}^0 - c_{N-1}^0. \end{cases}$$

Otherwise, if N is an odd number, calculate

$$d_k^1 = \mathrm{Int}(\frac{c_{2k}^0 + c_{2k+2}^0}{2}) - c_{2k+1}^0, \quad k = 0, \cdots, M_1 - 1.$$

(2) If N is an even number, compute

$$\begin{cases} c_0^1 = c_0^{1,0} - \mathrm{Int}(\frac{d_0^1}{2}), \\ c_k^1 = c_k^{1,0} - \mathrm{Int}(\frac{d_{k-1}^1 + d_k^1}{4}), \quad k = 1, \cdots, N_1 - 2, \\ c_{N_1-1}^1 = c_{N_1-2}^{1,0} - \mathrm{Int}(\frac{d_{N_1-2}^1 + d_{N_1-2}^1}{4}). \end{cases}$$

Otherwise, if N is an odd number, calculate

$$\begin{cases} c_0^1 = c_0^{1,0} - \mathrm{Int}(\frac{d_0^1}{2}), \\ c_k^1 = c_k^{1,0} - \mathrm{Int}(\frac{d_{k-1}^1 + d_k^1}{4}), \ k = 1, \cdots, N_1 - 2, \\ c_{N_1-1}^1 = c_{N_1-1}^{1,0} - \mathrm{Int}(\frac{d_{M_1-1}^1}{2}). \end{cases}$$

(b) Reconstruction

(1) Compute

$$c_0^0 = c_0^1 + \mathrm{Int}(\frac{d_0^1}{2}),$$

$$c_{2k}^0 = c_k^1 + \mathrm{Int}(\frac{d_{k-1}^1 + d_k^1}{4}), \ k = 1, \cdots, N_1 - 2.$$

Then, if N is even, calculate

$$c_{N-2}^0 = c_{N_1-1}^1 + \mathrm{Int}(\frac{d_{N_1-2}^1 + d_{N_1-1}^1}{4}),$$

else calculate

$$c_{N-1}^0 = c_{N_1-1}^1 + \mathrm{Int}(\frac{d_{M_1-1}^1}{2}).$$

(2) Compute

$$c_{2k+1}^0 = \mathrm{Int}(\frac{c_{2k}^0 + c_{2k+2}^0}{2}) - d_k^1, \ k = 0, \cdots, M_1 - 2.$$

Then, if N is even, calculate

$$c_{N-1}^0 = c_{N-2}^0 - d_{M_1-1}^1.$$

The *PPP* property for Example 1-2 mentioned at the end of the previous section is also applicable for these three examples. It is obvious these three transformations are not really linear, but they are similar to the one using the corresponding filters given above. Especially, the filters in Example 3 and Example 5 belong to, with minor modification, the group of the best biorthogonal filters for image compression according to both our experience and the conclusion of [4].

Also, from the above three examples, we can note that if we begin with integer (linear or nonlinear) wavelet transformations and then use some proper upgrading formulas, we can get other, much better integer, wavelet transformations for image compression. Now, the key problem is: What kind of deductive formulas should be used? We provide an answer to this question in the following two sections, Section 4 and Section 5.

4. Lifting Scheme and Integer Biorthogonal Filtering

The *Lifting scheme*, discovered by Sweldens [1], is a new approach for constructing biorthogonal wavelets with compact support. However, the most interesting part of this method for us is: it can be used, with minor modification, to create integer biorthogonal wavelet transformations. The following is an adaptation of the technique of [1].

Definition 1. The set of filters $\{h,\ \tilde{h},\ g,\ \tilde{g}\}$ is a set of *biorthogonal filters* if the following formula is satisfied:

$$\forall \omega \in \mathbf{R}:\ \tilde{m}(\omega)\overline{m^{t}(\omega)} = 1,$$

where

$$m(\omega) = \begin{bmatrix} h(\omega) & h(\omega+\pi) \\ g(\omega) & g(\omega+\pi) \end{bmatrix},$$

and

$$h(\omega) = \sum_k h_k e^{-ik\omega} \text{ and } g(\omega) = \sum_k g_k e^{-ik\omega},$$

and similarly for

$$\tilde{m}(\omega),\ \tilde{h}(\omega) \text{ and } \tilde{g}(\omega).$$

The following lemma is the main result of the *lifting scheme* [1] reported as corollary 6 in that paper.

Lemma 1. Take an initial set of finite biorthogonal filters $\{h,\ \tilde{h}^0,\ g^0,\ \tilde{g}\}$, then a new set of finite biorthogonal filters $\{h,\ \tilde{h},\ g,\ \tilde{g}\}$ can be found as

$$\begin{aligned} \tilde{h}(\omega) &= \tilde{h}^0(\omega) + \tilde{g}(\omega)\overline{s(2\omega)} \\ g(\omega) &= g^0(\omega) - h(\omega)s(2\omega). \end{aligned} \tag{4.1a}$$

Similarly, if we take $\{h^0,\ \tilde{h},\ g,\ \tilde{g}^0\}$ as an initial set of biorthogonal filters, a new set of finite biorthogonal filters $\{h,\ \tilde{h},\ g,\ \tilde{g}\}$ can be found as

$$\begin{aligned} h(\omega) &= h^0(\omega) + g(\omega)\overline{\tilde{s}(2\omega)} \\ \tilde{g}(\omega) &= \tilde{g}^0(\omega) - \tilde{h}(\omega)\tilde{s}(2\omega). \end{aligned} \tag{4.1b}$$

Here, $s(\omega)$ is a trigonometric polynomial and the corresponding filter s is finite, and so is $\tilde{s}(\omega)$. Actually, regarding the filters, (4.1) is equivalent to

$$\begin{aligned} \tilde{h}_k &= \tilde{h}_k^0 + \sum_l \tilde{g}_{k+2l} s_l \\ g_k &= g_k^0 - \sum_l h_{k-2l} s_l. \end{aligned} \tag{4.2a}$$

or

$$\begin{aligned} h_k &= h_k^0 + \sum_l g_{k+2l} \tilde{s}_l \\ \tilde{g}_k &= \tilde{g}_k^0 - \sum_l \tilde{h}_{k-2l} \tilde{s}_l. \end{aligned} \tag{4.2b}$$

Next, we use the lifting scheme with minor modification to create an integer, nonlinear, quasi-biorthogonal, wavelet algorithm. Suppose $\{c_n^0\}$ is a original signal, $\{c_n^1\}$ and $\{d_n^1\}$ are again its low and high frequency decomposition parts, obtained by using the filters $\{h,\ \tilde{h},\ g,\ \tilde{g}\}$.

If we use filters $\{\tilde{h}, \tilde{g}\}$ for decomposition (analysis), the corresponding decomposition algorithm is

$$\begin{cases} c_k^1 = \alpha_c \sum_n c_n^0 \tilde{h}_{n-2k}, \\ d_k^1 = \alpha_d \sum_n c_n^0 \tilde{g}_{n-2k}. \end{cases}$$

While the reconstruction algorithm will be

$$c_n^0 = 2\sum_k \left(\frac{c_k^1 h_{n-2k}}{\alpha_c} + \frac{d_k^1 g_{n-2k}}{\alpha_d} \right),$$

related to the synthesis filter $\{h, g\}$. Here, parameters α_c and α_d are positive constants with $\alpha_c \bullet \alpha_d = 2$. For example, in the situation of regular biorthogonal decomposition and reconstruction, $\alpha_c = \alpha_d = \sqrt{2}$; and for Example 1 through Example 5 above, $\alpha_c = 1$ and $\alpha_d = 2$.

If the set of filters $\{h, \tilde{h}, g, \tilde{g}\}$ is from $\{h, \tilde{h}^0, g^0, \tilde{g}\}$ by (4.2b), then the decomposition can be accomplished as follows:

1. Calculate
$$\begin{cases} c_k^{1,0} = \alpha_c \sum_n c_n^0 \tilde{h}_{n-2k}^0, \\ d_k^1 = \alpha_d \sum_n c_n^0 \tilde{g}_{n-2k}. \end{cases} \quad (4.3)$$

2. Calculate
$$c_k^1 = c_k^{1,0} + \frac{\alpha_c}{\alpha_d} \sum_l d_{k-l}^1 s_l. \quad (4.4)$$

The relative reconstruction scheme will be:

1. Calculate
$$c_k^{1,0} = c_k^1 - \frac{\alpha_c}{\alpha_d} \sum_l d_{k-l}^1 s_l, \quad (4.5)$$

2. Calculate
$$c_n^0 = 2\sum_k \left(\frac{c_k^{1,0} h_{n-2k}}{\alpha_c} + \frac{d_k^1 g_{n-2k}^0}{\alpha_d} \right). \quad (4.6)$$

Here, equations (4.3) and (4.6) are just the wavelet (inverse) transforms using biorthogonal filters $\{h, \tilde{h}^0, g^0, \tilde{g}\}$. While (4.4) and (4.5) are forward and backward upgrading formulas.

Similarly, if the set of filters $\{h, \tilde{h}, g, \tilde{g}\}$ is from the initial set of filters $\{h^0, \tilde{h}, g, \tilde{g}^0\}$ by using (4.2b), the relative decomposition is:

1. Calculate
$$\begin{cases} c_k^1 = \alpha_c \sum_n c_n^0 \tilde{h}_{n-2k}, \\ d_k^{1,0} = \alpha_d \sum_n c_n^0 \tilde{g}_{n-2k}^0. \end{cases}$$

2. Calculate $$d_k^1 = d_k^{1,0} - \frac{\alpha_d}{\alpha_c}\sum_l c_{k+l}^1 s_l .$$

The reconstruction scheme is:

1. Calculate $$d_k^{1,0} = d_k^1 + \frac{\alpha_d}{\alpha_c}\sum_l c_{k+l}^1 s_l$$

2. Calculate $$c_n^0 = 2\sum_k\left(\frac{c_k^1 h_{n-2k}^0}{\alpha_c} + \frac{d_k^1 g_{n-2k}}{\alpha_d}\right)$$

For the sake of clarity, we haven't considered the boundary situation, but we will address this later.

Corollary 4.1. Suppose biorthogonal filters $\{h, \tilde{h}, g, \tilde{g}\}$ are from initial filters $\{h, \tilde{h}^0, g^0, \tilde{g}\}$ by the lifting scheme (4.1a) or (4.2a). If the decomposition and reconstruction by filters $\{h, \tilde{h}^0, g^0, \tilde{g}\}$ can be accomplished only by integer calculation, such as Example 2, we also can create a corresponding integer wavelet decomposition and reconstruction scheme which is very "close" to the original one by using filters $\{h, \tilde{h}, g, \tilde{g}\}$. Here the word "close" means that the difference of the two decomposition schemes is just some rounding error, and this rounding error will be corrected by the integer reconstruction scheme.

In fact, if $\{c_k^{1,0}\}$ and $\{d_k^1\}$ are integer after (4.3), we can calculate $\{c_k^1\}$ by

$$c_k^1 = c_k^{1,0} + \mathrm{Int}\left(\frac{\alpha_c}{\alpha_d}\sum_l d_{k-l}^1 s_l\right). \tag{4.7}$$

instead of (4.4). Here $\mathrm{Int}(x)$, as described in Section 2, is an arbitrary rounding up function which satisfies $x-1 \le \mathrm{Int}(x) \le x+1$. It is obvious that (4.7) is very "close" to (4.4), and the exact reconstruction scheme can easily be obtained from

$$c_k^{1,0} = c_k^1 - \mathrm{Int}\left(\frac{\alpha_c}{\alpha_d}\sum_l d_{k-l}^1 s_l\right) \tag{4.8}$$

and (4.6). There will be a similar result, if the set of biorthogonal filters $\{h, \tilde{h}, g, \tilde{g}\}$ is obtained from the initial set of filters $\{h^0, \tilde{h}, g, \tilde{g}^0\}$ by using (4.2b).

We can now note, except for the example shown in the Lazy *wavelet*, (Example 2) most standard biorthogonal wavelet transforms cannot be performed directly by integer, even for one of the simplest wavelets, the *Haar wavelet*. However, if we properly choose the parameters α_c and α_d, and slightly change the transform algorithms, such as Example 1 and Example 3, we can have a variation of the original biorthogonal wavelet transforms with respect to the set of filters

$\{h, \tilde{h}^0, g^0, \tilde{g}\}$ (or $\{h^0, \tilde{h}, g, \tilde{g}^0\}$). On the other hand, the parameters $\{s_i\}$ should be also chosen carefully to guarantee that only addition and shift operations are needed by the algorithm.

Another observation: if the set of filters $\{h, \tilde{h}, g, \tilde{g}\}$ is obtained from a set of filters $\{h^0, \tilde{h}, g, \tilde{g}^0\}$ by the lifting scheme, and the set $\{h^0, \tilde{h}, g, \tilde{g}^0\}$ is also obtained from a filter set $\{h^0, \tilde{h}^0, g^0, \tilde{g}^0\}$, we can repeatedly use Corollary 1 to get a "close" integer wavelet transformation.

5. The Correction Method for Creating Integer Wavelet Transforms

In this section, we will describe another approach for obtaining integer wavelets by using the so called *Correction method*. The motivation of this method is from the S+P transform, and we will now generalize this approach. Actually, the lifting scheme for generating biorthogonal wavelets can be considered as a special case of the correction method. From this method we can get some even complicated filters with fast decomposition and reconstruction algorithm.

Suppose that we already have a simple integer wavelet transform, such as Examples 1 through 3, the decomposition and reconstruction scheme of which can be formulated as follows:

Decomposition
$$c_k^{1,0} = df_c\left(\{c_n^0\}\right) \qquad d_k^{1,0} = df_d\left(\{c_n^0\}\right) \tag{5.1}$$

Reconstruction
$$c_n^0 = rf\left(\{c_k^{1,0}\},\{d_k^{1,0}\}\right) \tag{5.2}$$

Here, (5.1) and (5.2) can be the same as (4.3) and (4.6) or other algorithms.

In general, after the above decomposition, one may not be satisfied with the result. There may still be some correlation among the highpass components because of the aliasing from the lowpass components, or the lowpass components do not carry enough of the expected information from the original signal. Hence, we could make an improvement by putting some correction part on the highpass components or lowpass components. There are many ways to accomplish this. However, for the sake of the integer calculation, we prefer to use following correction method. For example, if we want to make a correction for the highpass part, the corresponding formula would be:

$$d_k^1 = d_k^{1,0} - \mathrm{Int}\left(dc_k^1\right) \; k = \cdots, 0, 1, 2, \cdots. \tag{5.3}$$

Here, dc_k^1 is a correction quantity for d_k^1

$$dc_k^1 = \sum_{i=S_0}^{S_1} \sigma_i c_{k+i}^{1,0} + \sum_{j=1}^{T} \tau_j d_{k+j}^{1,0}, \; k = \cdots, 0, 1, 2, \cdots \tag{5.4}$$

and, $\{\sigma_i\}_{i=S_0}^{S_1}$ and $\{\tau_j\}_{j=1}^{T}$ are given parameters which have been chosen for the user's purpose, such as reducing the redundancy among highpass components or some other special requirement. We are not going to discuss how to choose these parameters, but one can refer to the references [3, 5, 6] for clarification of this process. The only thing we need to mention is, for the sake of the integer calculation, any entries in both $\{\sigma_i\}_{i=S_0}^{S_1}$ and $\{\tau_j\}_{j=1}^{T}$ should be rational numbers with denominators being powers of 2.

From (5.1), (5.3) and (5.4), it is easy to see the perfect reconstruction algorithm can be

$$d_k^{1,0} = d_k^1 + \mathrm{Int}(dc_k),\ k = \cdots,\ m,\ m-1,\ m-2\cdots, \tag{5.5}$$

combined with (5.2).

As mentioned above, the Lifting scheme is a special condition of the Correction method. Examples 3 through 5 can also be considered as the examples of this method. We next give an example of the Correction method which cannot be included in the group of Lifting scheme, and also which does not result in a closed form of compact support for biorthogonal filters.

Example 6 S+P transform [3], which is similar to using following analysis filters

n	-2	-1	0	1	2	3
h_n	0	0	1/2	1/2	0	0
$\tilde{g}_n$	-1/16	-1/16	15/32	-17/32	7/32	-1/32

While, the synthesis filters do not have compact support. However, the S+P transform can be implemented as follows:

(a) Decomposition

(1) Take the decomposition step of Example 1, that is, compute

$$d_k^{1,0} = c_{2k}^0 - c_{2k+1}^0, \quad k = 0,\ 1,\cdots, M_1 - 1;$$

and

$$c_k^1 = \mathrm{Int}(\frac{d_k^{1,0}}{2}) + c_{2k+1}^0, \quad k = 0,\cdots, N_1 - 2,$$

$$c_{M_1-1}^1 = \begin{cases} \mathrm{Int}(\frac{d_{M_1-1}^{1,0}}{2}) + c_{N-1}^0, & \text{if } N \text{ is an even number,} \\ c_{N-1}^0, & \text{if } N \text{ is an odd number.} \end{cases}$$

(2) Correction Step: Define $S_0 = -1,\ S_1 = 1,\ T = 1$ and

$$\sigma_{-1} = -\tfrac{1}{4},\ \sigma_0 = -\tfrac{1}{4},\ \sigma_1 = \tfrac{1}{8};$$
$$\tau_1 = \tfrac{1}{2}.$$

and now compute

In fact, the quantity b_k would have the same value in both (4.7) and (4.8) if we calculate it in the same way. On the other hand, if the working unit for b_k is q bits, the machine will give b_k another value, say $\tilde{b}_k$ ($-2^{q-1} \le \tilde{b}_k < 2^{q-1}$), where $\tilde{b}_k$ is not equal to b_k in the sense of mathematics if the value of b_k is beyond the interval $[-2^{q-1}, 2^{q-1}-1]$. However, $\tilde{b}_k$ will be the same in both (4.7) and (4.8). Therefore, the machine will automatically implement (7.1) and (7.2) as

$$c_k^1 = \begin{cases} c_k^{1,0} + \tilde{b}_k, & \text{if } -2^{q-1} \le c_k^{1,0} + \tilde{b}_k < 2^{q-1}, \\ c_k^{1,0} + \tilde{b}_k - 2^q, & \text{if } c_k^{1,0} + \tilde{b}_k \ge 2^{q-1}, \\ 2^q + c_k^{1,0} + \tilde{b}_k & \text{if } c_k^{1,0} + \tilde{b}_k < -2^{q-1}. \end{cases} \tag{7.1m}$$

and

$$c_k^{1,0} = \begin{cases} c_k^1 - \tilde{b}_k, & \text{if } -2^{q-1} \le c_k^1 - \tilde{b}_k < 2^{q-1}, \\ 2^q + c_k^1 - \tilde{b}_k, & \text{if } c_k^1 - \tilde{b}_k < -2^{q-1}, \\ c_k^1 - \tilde{b}_k - 2^q, & \text{if } c_k^1 - \tilde{b}_k \ge 2^{q-1}. \end{cases} \tag{7.2m}$$

It is easy to see that (7.2m) is just the backward operation of (7.1m), which provides the evidence that the conclusion of this lemma is correct.

It should be mentioned that the coefficients $\{c_k^1\}$ obtained by (4.3) and (7.1m) might not be the "real" wavelet coefficients using common sense. However, if we still use the working unit with q bits precision at the reconstruction step, (7.2m) and (4.6) will give the exact original signal back. On the other hand, the coefficients $\{c_k^1\}$ still keep the most continuity of the "real" wavelet coefficients. Therefore, when we repeat the decomposition step on $\{c_k^1\}$, most small coefficients in its high frequency part $\{d_k^2\}$ will be almost the same as the "real" coefficients (within some rounding error), which allows us to still take advantage of the "real" wavelet transform in image compression.

A similar argument can show the same *PPP* property will hold for the integer wavelet transforms generated by the Correction method in Section 5.

As we mentioned before, for many applications, the lossless image compression is as important as lossy compression. The integer wavelet transforms give the opportunity to compress without loss. It is also obvious that the integer wavelet algorithms can be used wherever ordinary wavelets are used, especially in signal and image compression. However, for most computers, the integer wavelet transform is much faster than the ordinary one and it uses much less memory. The following are some applications illustrating these types of transforms.

$$
\begin{cases}
d_0^1 = d_0^{1,0} - \mathrm{Int}(\frac{c_0^1 - c_1^1}{4}); \\
d_k^1 = d_k^{1,0} - \mathrm{Int}(\frac{2c_{k-1}^1 + c_k^1 - 3c_{k+1}^1 - 2d_{k+1}^{1,0}}{8}), \quad k = 1, \cdots, M_1 - 2; \\
d_{M_1-1}^1 = d_{M_1-1}^{1,0} - \mathrm{Int}(\frac{c_{M_1-2}^1 - c_{M_1-1}^1}{4}).
\end{cases}
$$

(b) Reconstruction

(1) Compute

$$
\begin{cases}
d_{M_1-1}^{1,0} = d_{M_1-1}^1 + \mathrm{Int}(\frac{c_{M_1-2}^1 - c_{M_1-1}^1}{4}); \\
d_k^{1,0} = d_k^1 + \mathrm{Int}(\frac{2c_{k-1}^1 + c_k^1 - c_{k+1}^1 - 2d_{k+1}^{1,0}}{8}), k = M_1 - 2, \cdots, 1; \\
d_0^{1,0} = d_0^1 + \mathrm{Int}(\frac{c_0^1 - c_1^1}{4}).
\end{cases}
$$

(2) If N is an even number, compute

$$c_{2k+1}^0 = c_k^1 - \mathrm{Int}(\frac{d_k^1}{2}), \quad k = 0, \cdots, N_1 - 1$$

or, if N is an odd number, we have

$$c_{2k+1}^0 = c_k^1 - \mathrm{Int}(\frac{d_k^1}{2}), \quad k = 0, \cdots, N_1 - 2,$$

$$c_{N-1}^0 = c_{N_1}^1.$$

(3) Compute

$$c_{2k}^0 = d_k^1 + c_{2k+1}^0, \quad k = 0, \cdots, M_1 - 1.$$

6. Boundary Conditions

In the previous two sections, we did not show how to get the integer, wavelet transform at the boundaries of signals. However, for all of the examples given above, the boundaries have been considered. There are two issues dealing with boundary filtering if we use the *Lifting scheme* or the *Correction method* to generate the integer wavelet transformations. The first is how to process the boundaries which occur in the start-up wavelet transformations. The second is how to deal with the boundaries in the deductive formula. If the boundaries in the start-up wavelet transform have already been established, then those in the upgrading formula are easy to establish. In fact, for the *Lifting scheme*, the boundaries in both steps should be processed in the same way. While, for the Correction *method*, it is easy to see from (5.3)-(5.4) that one has more choices to process boundaries in the second step. Therefore, the only thing we need to discuss here is the process by which the boundaries in the start up wavelet transformations are established. Assume we begin with compact supported biorthogonal wavelets.

Suppose the original signal is $\{c_n^0\}_{n=0}^N$. For creating integer biorthogonal wavelet transformations we can use the following symmetric extension [7]:

(1). If current biorthogonal filters have even length, we extend the boundaries of the signal as $c_{-k}^0 = c_{k-1}^0$, $k = 1,2,\cdots$;

(2). If the filters have odd length, we do the extension as $c_{-k}^0 = c_k^0$, $k = 1,2,\cdots$

Example 1 through 5 use the boundaries give above. In Example 6, the start up wavelet transform uses the above boundaries, but in the upgrading step, another boundary filtering is used. In addition, for arbitrarily sized images or signals, one can use the same technique which we described in the above examples to deal with this condition.

7. Some Applications

Before talking about any applications of the integer wavelet transform given above, we first prove that a nice *property of precision preservation (PPP)*, which is similar to the one mentioned in Section 2, holds for both the Lifting and Correction upgrading technique. This property is very important for many applications.

Lemma 7.1 Suppose that our integer wavelet transform starts with a pair of biorthogonal filters with the *PPP* property discussed in Section 2, that is, (4.3) and (4.6) possess this property. Then, the same property will be preserved in the whole algorithm if we adopt the Lifting scheme to be the upgrading formula.

In other words, Lemma 7.1 states if we only use the working units with the same precision as the original signal or image to calculate the wavelet transform developed in Section 4, the equations (4.8) and (4.6) are still the backward operations of the equations (4.3) and (4.7).

Proof. Assume that we only use q bits to represent images or signals, say, the range of the pixel values is within $[-2^{q-1}, 2^{q-1}-1]$. According to the hypothesis of the lemma, the equations (4.3) and its inverse (4.6) have the *PPP* property. Therefore, what we have to verify here is that the equation (4.7) and its inverse (4.8) can preserve the same property. We rewrite (4.7) and (4.8) as follow:

$$c_k^1 = c_k^{1,0} + b_k \,, \tag{7.1}$$

and its inverse

$$c_k^{1,0} = c_k^1 - b_k \,. \tag{7.2}$$

Here,

$$b_k = \mathrm{Int}\left(\frac{\alpha_r}{\alpha_d}\sum_i d_{k-i}^1 s_i\right).$$

Application 1. Lossless image compression

As mentioned at the beginning of this paper, the integer wavelet transformation established by the techniques described in this paper can always be used for lossless image compression because of the reversible ability. Especially, we can use the *PPP* property discussed in Lemma 7.1. We have used this wavelet lossless technology (WLT) for gray scale lossless image compression, and we have tried several images. For most natural images, the size of wavelet lossless compressed images is much smaller than corresponding GIF images. Figure 1 through 4 give some examples. Figure 1 is a standard image for compression, Figure 2 and 4 are X-ray images and Figure 3 is a two-value image but we treat it as a 8 bit gray scale image in order to compare with the GIF format. In fact, if we convert Figure 2 to a binary image, better result can be obtained by the JBIG technique.

(512x512)
Figure 1. Compression Ratio
WLT: 1.9:1/GIF: 1.05:1

(512x512)
Figure 2. Compression Ratio:
WLT 4.5:1/GIF: 2.72:1

"Visioneer may have come up with on
against the paper blizzard. . .gets pile:
way to others throughout your compa

Figure 3. Compression Ratio: WLT: 20.8/ GIF 17.8 (152x794)

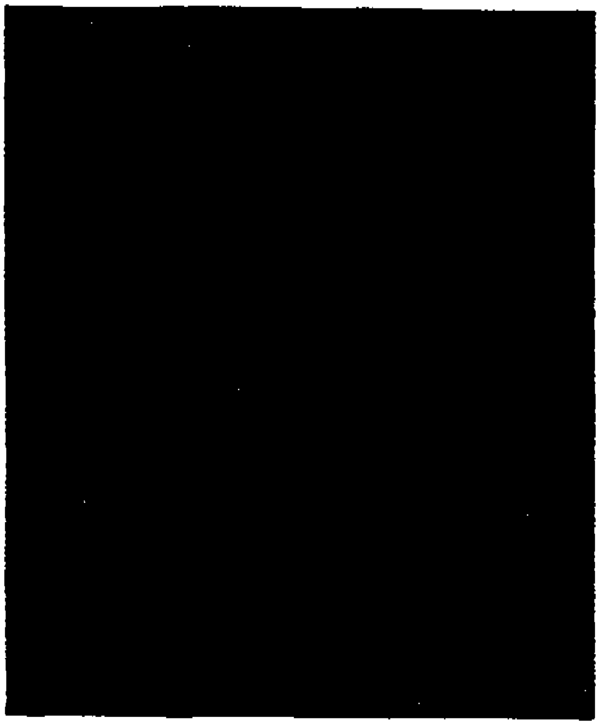


Figure 4. Compression Ratio: WLT 3.8:1/GIF 1.98:1 (1232x1024)

Application 2. Large scale medical image compression

Usually, 12 bits are used to represent one pixel in medical images. In this situation, the values of the pixels vary from 0 to 4095. Such images require careful treatment when a transform coding method is used for compression. If we use ordinary biorthogonal wavelets, the range of the transform coefficients will expand to $[-2^{16}, 2^{16}]$ when five levels of transform are used. Therefore, a longer working unit has to be employed, which consumes significant computer resources. However, the integer wavelet technique developed in this paper will solve this problem. For example, if we use the transforms given in Example 3, 5 and 6, the values of transform coefficients will be limited to the range of $[-2^{13}, 2^{13}]$. Even if we do not use the *PPP* property for these wavelets, *16* bits for the working unit is sufficient for all computations.

8. Conclusion

This paper has shown the processes necessary in order to obtain a non-linear, integer, biorthogonal, or non-biorthogonal reversible wavelet transform suitable for signal or image processing. We have shown how such a transform can be obtained either using the Lifting method, or the Correction method. For example, all interpolation wavelets can be modified to be corresponding integer wavelets without loosing any properties of original wavelets. In addition, we have shown under certain conditions, the precision of the transform computation on the computer can remain at the same precision of the data, thus reducing the need for additional computer memory during the transform computation. These are extremely powerful techniques when the target data are large images, or the requirements establish a need for speed.

Although this paper establishes the structure for the integer transform based upon the biorthogonal wavelet or some non-biorthogonal wavelet, we do not imply the examples in this paper are necessarily the best wavelets for any particular application. However, we do claim if one is going to use such a technique, the ideas suggested in this paper will provide the best implementation.

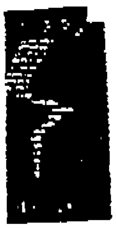
References

1. Wim Sweldens, *The lifting scheme: A custom-design construction of biorthogonal wavelets*, Applied and Computational Harmonic Analysis, Vol. 3, No. 2, April 1996.
2. A. Zandi, J. Allen, E. Schwartz and M. Boliek, *CREW: Compression with reversible embedded wavelets*, in IEEE Data Compression Conference, (Snowbird, Utah), pp.212-221, March 1995.
3. Amir Said, *An image multiresolution representation for lossless and lossy compression*, Submitted to the IEEE Transactions on Image Processing.
4. J. Villasenor, B. Belzer, and J. Liao, *Wavelet filter evaluation for image compression*, IEEE Trans. Image Processing, Vol. 4, pp.1053-1060.
5. G.R. Kuduvalli and R.M. Rangayyan, *Performance analysis of reversible image compression techniques for high-resolution digital teleradiology*, IEEE Trans. Med. Imaging, Vol. 11, pp. 430-445, Sept. 1992.
6. W.H. Press, B.P Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes: the Art of Science Programming*, Cambridge University Press, Cambridge, New York, 1986.
7. G. Strang and Truong Nguyen, *Wavelets and filter banks*, Wellesley-Cambridge Press, 1996.

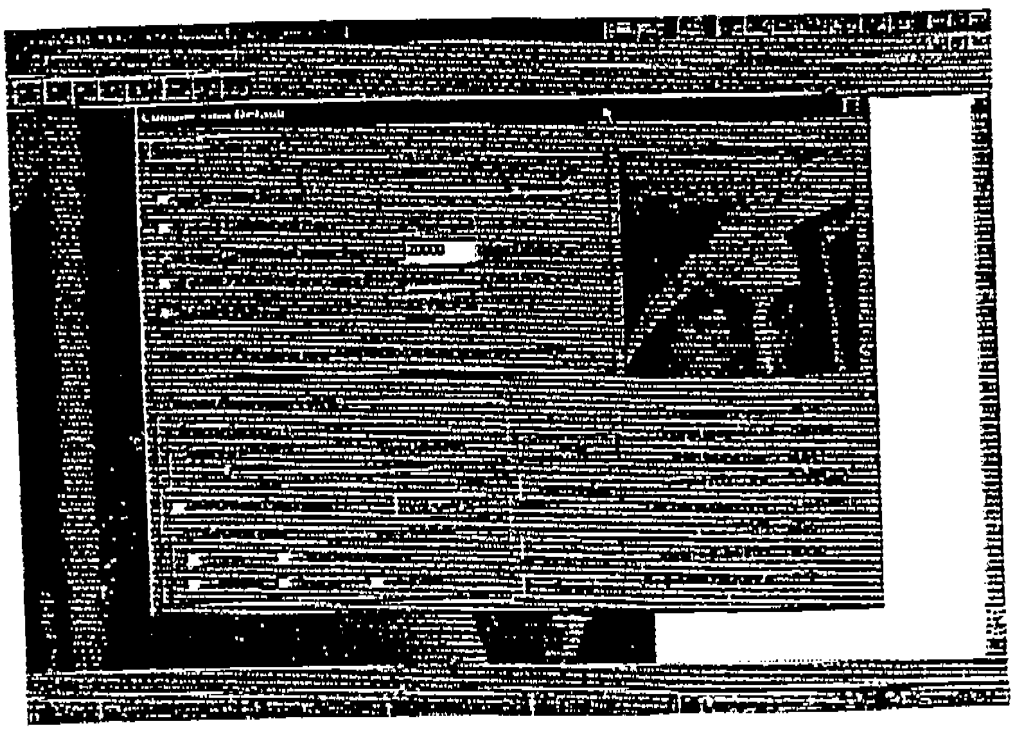
INFINITRON

Windows Image Compressor V3.0

Lightning Strike



Announcing, Lightning Strike™ Image Compressor (LSIC) version 3.0, a Windows 95 tool that compresses still images from 50:1 to 200:1 using INFINITRON's proprietary wavelet technology. LSIC is a versatile, easy to use tool for Web and Graphic designers that can handle a wide variety of digital image formats, and it includes filters and convenient web tools. Images can be compressed 3 to 5 times more than JPEG, while maintaining similar or better image fidelity. Images can be viewed in 2 to 4 seconds over the Internet rather than 10 to 20 seconds for images compressed under JPEG. This has enormous benefits for reducing bottlenecks on corporate networks and the web, and in addition, requires less storage space.



Lightning Strike Features

Compression. Images can be compressed as high as 200:1 using wavelet technology.

Compression Control. An EASY mode allows the user to compress images with minimal input, requiring only a decision between more quality or more compression. An ADVANCED mode enables the user to select; 1) image file size, 2) compression ratio, 3) PSNR, or 4) master level. Web designers will like the one step process to control the size of their image files, thus insuring the speed an image may be viewed on a browser.

Non Uniform Compression. Regions of an image can be selected for less compression to preserve a higher image quality while the rest of the image is compressed to the specified compression ratio. In this way important parts of a picture maintain crucial details while the over all picture file can be made as small as possible

Post Reconstruction Filters. Filters are available to enhance the reconstructed image. At compression time the user can preset a control to have these filters operate automatically during reconstruction. The filters include; quality improvement, sharpen (edge enhancement), smoothing, and brighten.

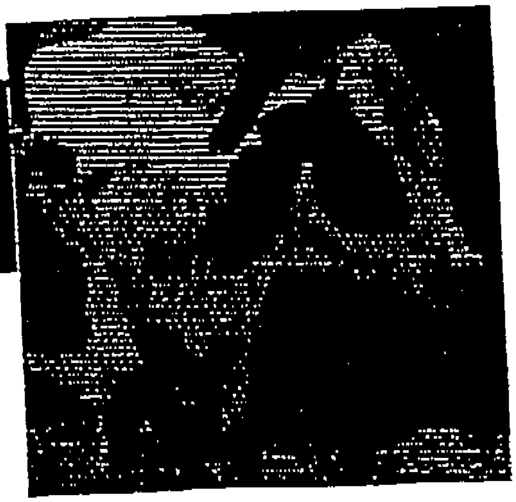
Transparencies. The user will have the ability to set pixels transparent so that a color in the background (already on the page) can be seen through the picture. This is useful for creative web site developers.

Progressive Compression. An image can be compressed so that when it is viewed it will appear quickly, first with low resolution, and then progressively building up in detail as it is downloaded. This insures the viewer does not lose interest while the image is downloaded.

INFINITRON

Windows Image Compressor V3.0

Lightning Strike



Cow, Bitmap, No Compression

Cow, JPEG 85:1 Compression

Cow, Lightning Strike, 85:1 compression

Applications Lightning Strike Image Compression

- Images on the Web
- Photo Stock
- Data Warehousing
- Catalogues
- Video Games
- CDs (Encyclopedias, Museums, Science, and Medicine)
- Archives (Art, History, Genealogy)
- Medical Imaging

Performance

Encode time typically less than 3 seconds for a 320 X 240 pixel, 24 bit color image on a 133 MHz. Pentium with 16 MB RAM.

Minimum Recommended System

Windows 95/NT OS
Pentium 100 MHz. 8 MB RAM
2 MB for program files
10 MB plus to swap image files

Ancillary INFINITRON Products

- Netscape Navigator Plug-in
- Java Applet
- ActiveX Control
- Web Site Image Converter
- Lightning Strike SDK
- GML Banner Generator

About INFINITRON, Inc.

Founded in 1992, INFINITRON, Inc. is a private company that specializes in the design and marketing of high quality image and video compression solutions for a wide array of markets. INFINITRON is based in Vancouver, BC with labs in Regina, Saskatchewan and Denton Texas.

Download a FREE demo version of Lightning Strike Windows Compressor from:
www.infinitron.com

INFINITRON USA Office 3401 East University, #104, Denton, TX, 76208
Tel: 817.484.1165 FAX: 817.484.0588

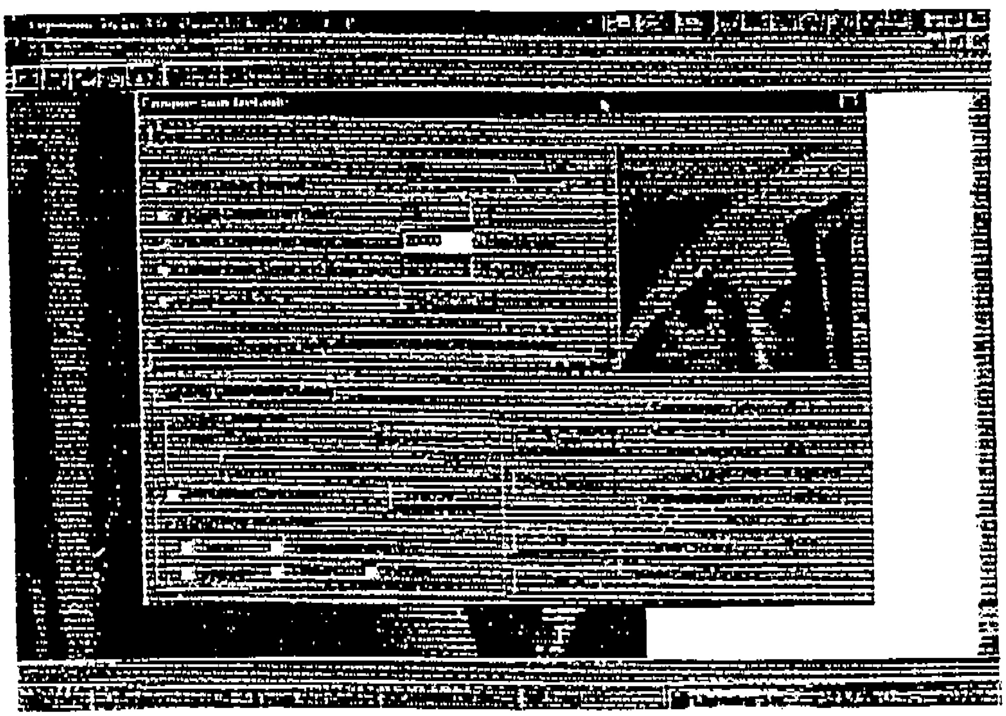
INFINITRON Canada Office 10th Flr 1199 W. Hastings, Vancouver, BC, V6E 3T5
Tel: 604.688.9789 FAX: 604.688.9798

INFINITRON

Windows Image Compressor V3.0

Lightning Strike

Announcing, Lightning Strike Image Compressor (TM) (LSIC) version 3.0, a Windows 95 tool that compresses still images from 50:1 to 200:1 using Infinitron's proprietary Wavelet technology. LSIC is a versatile, easy to use tool for Web and Graphic designers that can handle a wide variety of digital image formats and includes filters, and convenient web tools. Images can be compressed to files 4 times smaller than JPEG, while maintaining similar or better image fidelity. Images can be viewed in 2 to 5 seconds over the internet rather than 10 to 20 seconds for images compressed under JPEG. This has enormous benefits for reducing bottlenecks on corporate networks and the web, and in addition, requires less storage space.



Lightning Strike Features

Compression. Images can be compressed to ~~over~~ ~~100~~:1 using Wavelet Technology.

Compression Control. An EASY mode allows the user to compress images with minimal input, requiring only a decision between more quality or more compression. An ADVANCED mode enables the user to select; 1) image file size, 2) compression ratio, 3) PSNR, or 4) master level. Web designers will like the one step process to control the size of their image files, thus insuring the speed an image may be viewed on a browser.

Non Uniform Compression. Regions of an image can be selected for less compression to preserve a higher image quality while the rest of the image is compressed to the specified compression ratio. In this way important parts of a picture maintain crucial details while the over all picture file can be made as small as possible

Post Reconstruction Filters. Filters are available to enhance the reconstructed image. At compression time the user can preset a control to have these filters operate automatically during reconstruction. The filters include: quality improvement, sharpen (edge enhancement), smoothing, and brighten.

Transparencies. The user will have the ability to set pixels transparent so that a color in the background (already on the page) can be seen through the picture. This is useful for creative web site developers.

Progressive Compression. An image can be compressed so that when it is viewed it will appear quickly, first with low resolution, and then progressively building up in detail as it is downloaded. This insures the viewer does not loose interest while the image is downloaded.

INFINITRON

Windows Image Compressor V3.0

Lightning Strike

Lenna .bmp, No Compression

Lenna, JPEG 115:1 Compression

About INFINITRON, Inc.
Founded in 1992, INFINITRON, Inc. is a private company that specializes in the design and marketing of high quality image and video compression solutions for a wide array of markets. INFINITRON is based in Vancouver, BC with labs in Regina, Saskatchewan and Denton Texas.

Applications Lightning Strike Image Compression

- Images on the Web
- Photo Stock,
- Data Warehousing
- Cataloques.
- Video Games
- CDs (Encyclopedias, Museums, Science, and Medicine)
- Archives (Art, History, Geneology)
- Medical Imaging

Performance
Encode time typically less than 3 seconds for a 320 X 240 pixel, 24 bit color image on a 133 MHz. Pentium with 16 MB RAM.

Minimum Recommended System
Windows 95/NT OS
Pentium 100 MHz, 8 MB RAM
2 MB for program files
10 MB plus to swap image files

Ancillary INFINITRON Products

- Netscape Navigator Plug-in
- Java Applet
- ActiveX Control
- Web Site Image Converter
- Lightning Strike SDK
- GML Banner Generator

Download a FREE demo version of Lightning Strike Windows Compressor from:
www.infinitron.com

INFINITRON USA Office 3401 East University, #104, Denton, TX. 76208
Tel: 817.484.1165 FAX: 817.484.0588

INFINITRON Canada Office 10th Flr 1199 W. Hastings, Vancouver, BC, V6E 3T5
Tel: 604.688.9789 FAX: 604.688.9798

INFINITRON
Lightning Strike

Product Fact Sheet
Windows Compressor

Compressor Version 3.0

The Lightning Strike Compressor is a Windows tool that compresses still images from a wide variety of digital image formats using Infinitron's proprietary Wavelet algorithm. Images can be compressed to files 5 times smaller than JPEG, while maintaining similar or better image fidelity. Images can be viewed in 1 or 2 seconds over the internet rather than 10 to 20 seconds for images compressed under JPEG. This has enormous benefits for transmitting over corporate networks or the web, and in addition, saves space required for storing all those images.

Lightning Strike is a collection of tools in a user friendly environment. Two levels of user control are offered, one quick and easy for most applications, the other a master level for the advanced user who wishes to control parameters to maximize image quality.

The compression approach used by Lightning Strike is based upon integer wavelets. This technology is acknowledged by leading experts as a superior compression technique as compared to discrete cosine transform used in JPEG.

Lightning Strike Windows Compressor Features

Image Compression Options and Control

Compression Technique Options
Both Infinitron's Wavelet Compression and other frequently used compression methods are included in the product so users need only have Lightning Strike on their work station to perform all image compressions. Images can be compressed to Wavelet, JPEG, PNG, and GIF.

Compression Quality Versus Speed Options
The user can select one of two encoding processes that trade quality for speed of compression and ease of use. With the "Advanced" option selected, the optimum compression parameters are set by the user to give the best possible images for selected compression ratio. With "Easy" selected you get the fastest compression without having to know details of parameter selection.

Compression Ratio Control
The compressed image file size or compression ratio may be specified rather than the quality factor. This enables a web designer to control the size of their image files or the speed an image may be viewed, in a one step process.

Region of Interest Focusing
Regions of an image can be selected for less compression to preserve a higher image quality while the rest of the image is compressed to the specified compression ratio. In this way important parts of a picture maintain crucial details while the over all picture file can be made as small as possible. This is also known as Non-Uniform Compression.

INFINITRON
Lightning Strike

Product Fact Sheet
Windows Compressor

Split and Merge

Very large images, which could not otherwise be compressed due to their large size, can be split into smaller images and compressed individually. This process has the side advantage of using RAM more effectively speeding time for compression. The split images can be reassembled using the merge aspect of the feature.

Post Reconstruction Filters

Filters are available to enhance the reconstructed image. At compression time the user can preset a control to have these filters operate automatically during reconstruction. The filters include; quality improvement, sharpen (edge enhancement), smoothing, and brighten.

Transparencies

The user will have the ability to set pixels transparent so that a color in the background (already on the page) can be seen through the picture. This is useful for creative web site developers. This gives the ability to display pictures other than the rectangular shape allotted on the web page, i.e. circles, polygons etc.. Also, designers often use this feature for shadowing, letters and objects.

Progressive Decompression

An image can be compressed so that when it is viewed it will appear quickly, first with low resolution, and then progressively building up in detail as it is downloaded. This insures the viewer does not loose interest while the image is downloaded.

INFINITRON
Lightning Strike

Product Fact Sheet
Windows Compressor

Image Comparisons

Picture of Lena, with no Compression (512 X 512 Image)

INFINITRON
Lightning Strike

Product Fact Sheet
Windows Compressor

Picture of Lena Compressed 100:1 with Lightning Strike

INFINITRON
Lightning Strike

Product Fact Sheet
Windows Compressor

File Functions

PNG File Structure
The compressed images are stored in file format compliant with the PNG standard. In the future this file format will replace the GIF format used today.

Batch Compress
The user is able to compress many images at once by adding or deleting image files (or paths) to a list box.

Image Statistics
The compressor stores image statistics on each compressed image which may be viewed by the user. The following information is provided: image dimensions, compression ratio, file sizes, MSE, PSNR, maximum pixel difference, compression and decompression times.

Performance and System Requirements

Encoding and Decode Time.
The typical time to encode or decode a 320X240, 24 bit color image is 1 second on a Pentium running at 133 MHz with 16 Meg RAM.

Minimum Recommended System
The minimum system requirements for an IBM PC Compatible are:

Hard Disc Drive	2 Mbytes free for program files. 10 Mb plus to swap image files.
Operating System	MS Windows 3.1(Win32)/ 95/ NT
RAM	8 Mbytes

This software is also available on the Apple MAC, Solaris, and UNIX platforms.

INFINITRON
Lightning Strike

Product Fact Sheet
Windows Compressor

Ancillary Infinitron Products

Netscape Navigator Plug-in
Netscape plug-ins are available for the Mac68k, Mac PPC, Widows 3.1 and 95/NT.

Java Applet
Java Applets are available for the Mac68k, Mac PPC, Widows 3.1 and 95/NT.

ActiveX Control
The Lightning Strike decompression software is available for such applications as Microsoft's Internet Explorer, as an ActiveX control.

Web Site Image Converter
This utility will automate the conversion of web pages from JPEG to the Lightning Strike format. The utility searches an HTML file and replicates it replacing any JPEG image tags with Lightning Strike tags and converting the JPEG image files to Lightning Strike. The utility can follow link tags to recursively convert and replicate an entire web site or sub-section of a web site to the Lightning Strike format. This utility will be available for Windows NT and most flavors of the UNIX operating system.

Lightning Strike Software Developers Kit
Using the SDK, a developer can integrate the highly efficient Lightning Strike module libraries into their own applications.

Download a FREE demo version of Lightning Strike Windows Compressor from: www.infinitron.com

Infinitron USA Office 3401 East University, #104, Denton, TX, 76208
Tel: 817.484.1165 FAX: 817.484.0588

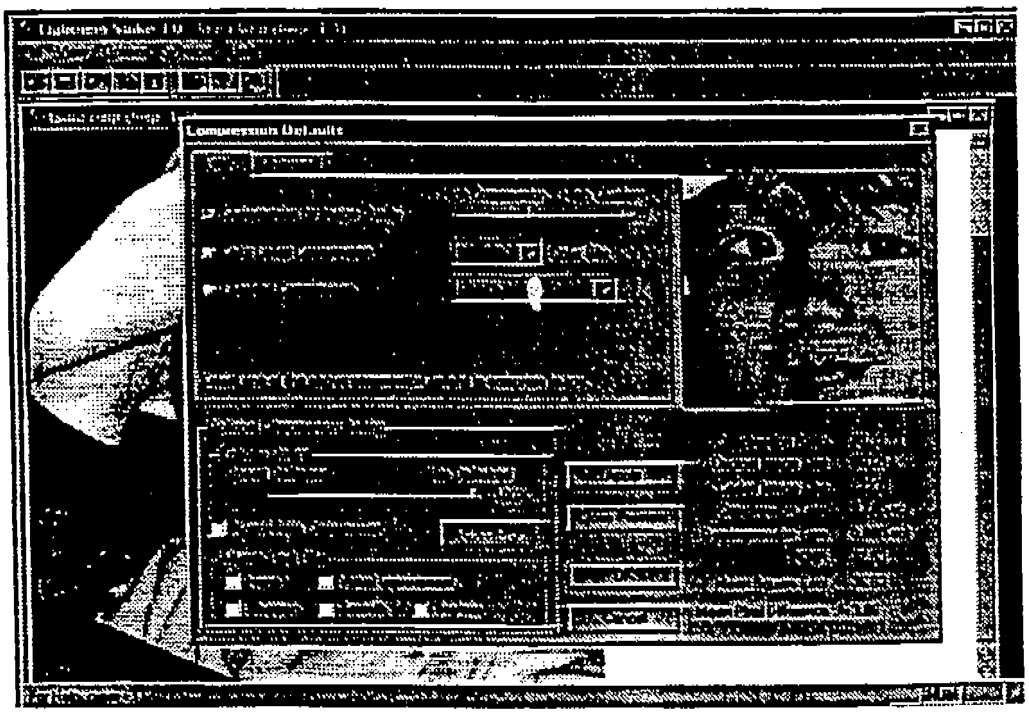
Infinitron Canada Office 10th Flr 1199 W. Hastings, Vancouver, BC, V6E 3T5
Tel: 604.688.9789 FAX: 604.688.9789

INFINITRON

Windows Image Compressor V3.0

Lightning Strike

Announcing, Lightning Strike™ Image Compressor (LSIC) version 3.0, a Windows 95 tool that compresses still images from 20:1 to 200:1 using INFINITRON's proprietary wavelet technology. LSIC is a versatile, easy to use tool for Web and Graphic designers that can handle a wide variety of digital image formats, and it includes filters and convenient web tools. Images can be compressed at ratios well in excess of present JPEG ratios while maintaining comparable image fidelity. This translates to much shorter image down load time on the web. This has enormous benefits for reducing bottlenecks on corporate networks and the web, and in addition, requires less storage space.



Lightning Strike

Lightning Strike Features

Compression. Uses a proprietary integer wavelet.

Compression Control. An EASY mode allows the user to compress images with minimal input, requiring only a decision between quality and compression. An ADVANCED mode enables the user to select: 1) image file size, 2) compression ratio, 3) PSNR, or 4) master level for professionals where every parameter can be altered. We also provide the highest, wavelet lossless compression for users wishing this capability. Web designers will like the one step process to control the size of their image files allowing control over the delivery time of an image over a network.

Non Uniform Compression. Regions can be selected for less compression to preserve image quality while the rest of the image is compressed to the specified compression ratio. In this way, important parts of a picture maintain

as small as possible

Post Reconstruction Filters. Filters are available to enhance the reconstructed image. At compression time the user can preset a control to have these filters operate automatically during reconstruction. The filters include: visual quality improvement, sharpen (edge enhancement), smoothing, and brighten.

Transparencies. The user will have the ability to set pixels transparent, so that a color in the background (already on the page) can be seen through the picture. This is useful for creative web site developers.

Progressive Compression. An image can be compressed so that when it is viewed it will appear quickly, first with low resolution, and then progressively building up in detail as it is downloaded. This insures the viewer does

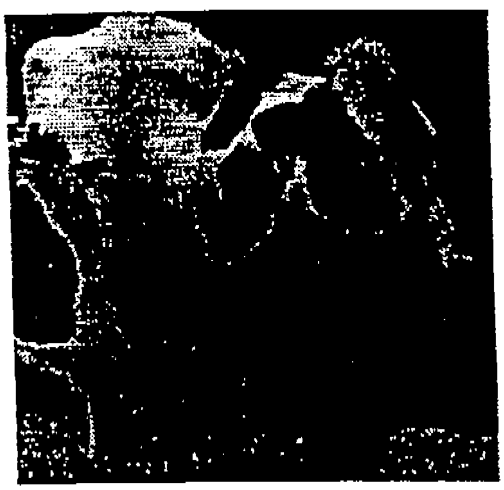
INFINITRON

Windows Image Compressor V3.0

Lightning Strike

Cow, Bitmap, No Compression

Cow, JPEG 85:1 Compression



Cow, Lightning Strike, 85:1 compression

Applications for Lightning Strike Image Compression

- Images on the Web
- Photo Stock
- Data Warehousing
- Catalogues
- Video Games
- CDs (Encyclopedias, Museums, Science, and Medicine)
- Archives (Art, History, Genealogy)
- Medical Imaging

Performance

Encode time typically less than 2.5 seconds for a 640 X 480 pixel, 24 bit color image on a 133 MHz Pentium with 16 MB RAM. Decode time is less than .75 seconds.

Minimum Recommended System

Windows 95/NT OS
Pentium 100 MHz, 8 MB RAM
2 MB for program files
10 MB plus to swap image files

Ancillary INFINITRON Products

- Netscape Navigator Plug-in
- Java Applet
- ActiveX Control
- Web Site Image Converter
- Lightning Strike SDK
- GML Banner Animation/Compression
- Black and White Image Compression

About INFINITRON, Inc.

Founded in 1992, INFINITRON, Inc. is a private company that specializes in the design and marketing of high quality image and video compression solutions for a wide array of markets. INFINITRON is based in Vancouver, BC with offices in Regina, Saskatchewan and Denton Texas.

Download a FREE demo version of Lightning Strike Windows Compressor from: www.infinitron.com

INFINITRON USA Office 3401 East University, #104, Denton, TX 76208
Tel: 817.484.1165 FAX: 817.484.0586

INFINITRON 10th Flr 1199 W. Hastings, Vancouver, BC, V6E 3T5

What is claimed is:

1. A method for compressing an image, comprising the steps of:

performing a wavelet transformation of the image;

quantizing the wavelet transformed image;

applying entropy coding to the quantized image; and

outputting a file that includes the entropy coded image.

2. The method of claim 1, further comprising the following step:

performing a color transformation of the image.

3. The method of claim 1, further comprising the following step:

performing the wavelet transformation using an integer wavelet transform.

4. The method of claim 3, further comprising:

deriving the integer wavelet transform using a lifting scheme.

5. The method of claim 3, further comprising:

deriving the integer wavelet transform using a correction method.

6. The method of claim 1, wherein the step of quantizing includes the sub-step of:

processing the wavelet transformed image using sub-band oriented quantization.

7. The method of claim 1, further comprising:

comparing the wavelet transformed image to at least one predetermined threshold value.

8. A method for wavelet-based image compression using reduced color components, comprising the steps of:

creating a color table for an input image having a plurality of pixels;

calculating an index for each of the pixels, whereby generating a plurality of indices;

performing a wavelet transformation on the indices;

applying entropy coding on the transformed indices; and

outputting a file that includes the entropy coded indices.

9. The method of claim 8, further comprising: dithering the pixels to generate the indices.

10. The method of claim 8, further comprising: partitioning a large image into a plurality of small images to produce the input image.

11. The method of claim 10, wherein the large image is selectively partitioned.

12. An image processing system, comprising:

means for performing a wavelet transformation on an input image;

means for quantizing the wavelet transformed image;

means for entropy coding to the quantized image; and

means for outputting the entropy coded image.

13. The image processing system of claim 12, further comprising:

means for receiving the entropy coded image;

means for entropy decoding the received image;

means for de-quantizing the decoded image; and means for performing an inverse wavelet transformation on the de-quantized image to produce an output image.

14. The image processing system of claim 12, further comprising:

means for displaying the output image.

15. The image processing system of claim 12, further comprising:

means for transmitting the entropy encoded image over a communications medium.

16. An image compression system, comprising:

a compressor configured to generate a compressed image based on an integer wavelet transform derived using a technique selected from a lifting scheme and a correction method.

17. The image compression system of claim 16, wherein the compressor quantizes a wavelet transformed image to produce the compressed image.

18. The image compression system of claim 16, wherein the compressor entropy encodes a quantized image to produce the compressed image.

19. The image compression system of claim 16, wherein the compressor performs a color transformation to produce the compressed image.

20. An image decompression system, comprising:

a decompressor configured to generate a decompressed image based on an integer inverse wavelet transform derived using a technique selected from a lifting scheme and a correction method.

21. A computer-readable memory storing a computer program for directing a computer system to perform image compression, wherein the computer program implements steps for performing a wavelet transformation of an input image, quantizing the wavelet transformed image, applying entropy coding to the quantized image, and outputting a file that includes the entropy coded image.

22. A method of compressing a data file, comprising the steps of:

performing a wavelet transformation of the data file to provide a series of wavelet coefficients;

quantizing those wavelet coefficients which fall above a predetermined threshold value to provide a quantized series of wavelet coefficients; and

compressing the quantized series of wavelet coefficients to provide a compressed data file.

23. The method of claim 22 wherein the compressing step comprises the step of applying an entropy coding to the quantized series of wavelet coefficients.

24. The method of claim 23 wherein the entropy coding is selected from the group of arithmetic, Huffman, run length and Huffman run length combined.

25. The method of claim 23 further comprising the step of performing a color transformation of the data file prior to the wavelet transformation step.

26. The method of claim 25 wherein the quantizing step comprises sub-band orientation quantization.

27. The method of claim 26 wherein the wavelet transformation step comprises integer wavelet transformation.

28. The method of claim 22 further comprising the step of filtering the data file prior to the wavelet transformation step.

29. The method of claim 27 wherein the integer wavelet transformation comprises biorthogonal filter method.

30. The method of claim 27 wherein the integer wavelet transformation comprises the correction method.

31. A compressed data file comprising a wavelet transformation of a data file having a series of compressed, quantized wavelet coefficients, the quantized wavelet coefficients having a value above a predetermined threshold value to provide a quantized series of wavelet coefficients.

32. A program for compressing a data file comprising:

a routine for performing a wavelet transformation of the data file to provide a series of wavelet coefficients;

a routine for quantizing those wavelet coefficients which fall above a predetermined threshold value to provide a quantized series of wavelet coefficients; and

a routine for compressing the quantized series of wavelet coefficients to provide a compressed data file.

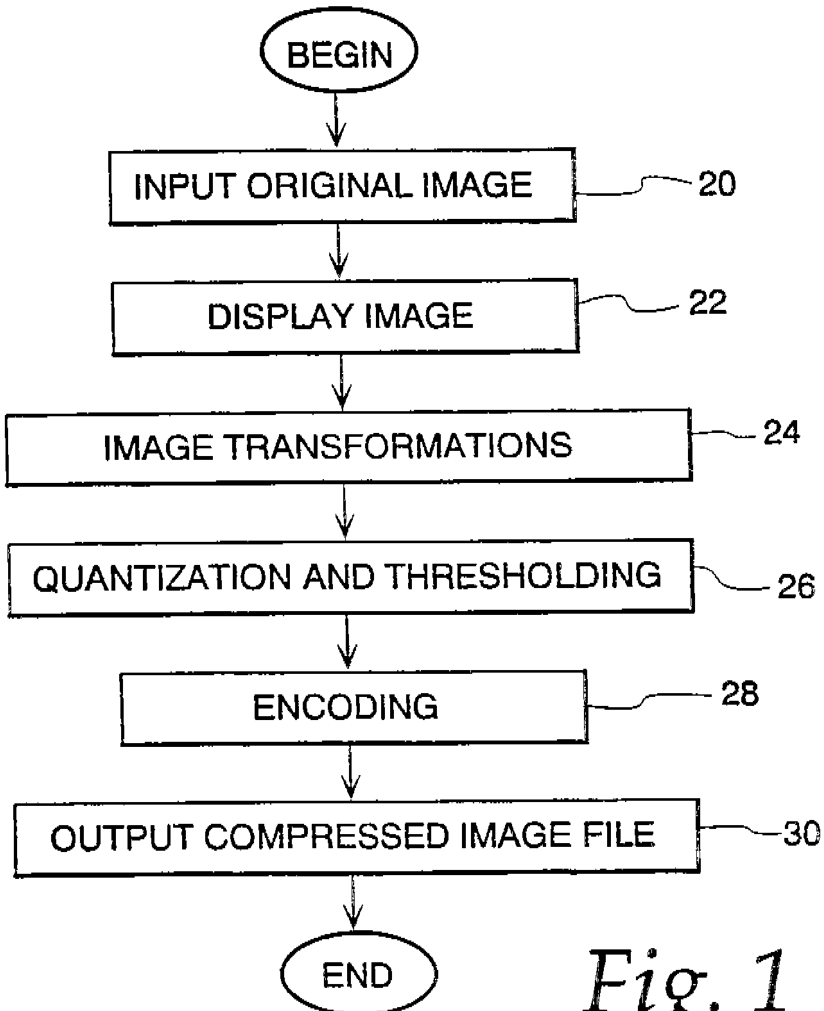

BEGIN
INPUT ORIGINAL IMAGE
20
DISPLAY IMAGE
22
IMAGE TRANSFORMATIONS
24
QUANTIZATION AND THRESHOLDING
26
ENCODING
28
OUTPUT COMPRESSED IMAGE FILE
30
END

Fig. 1

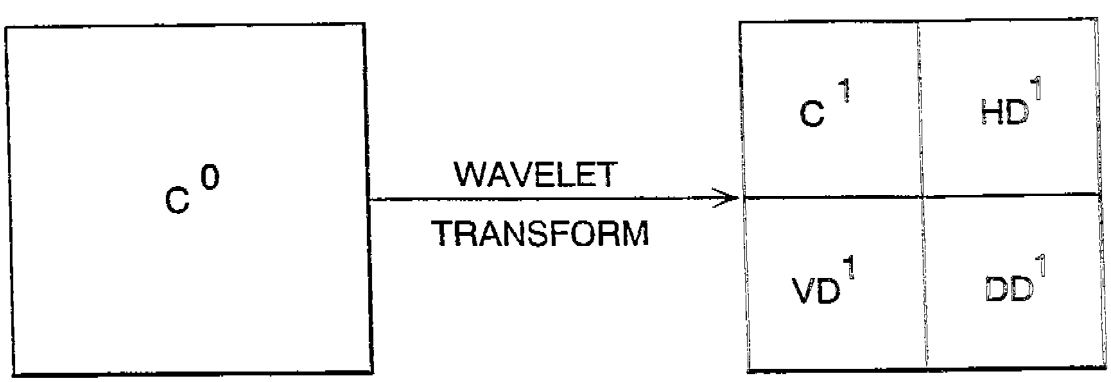
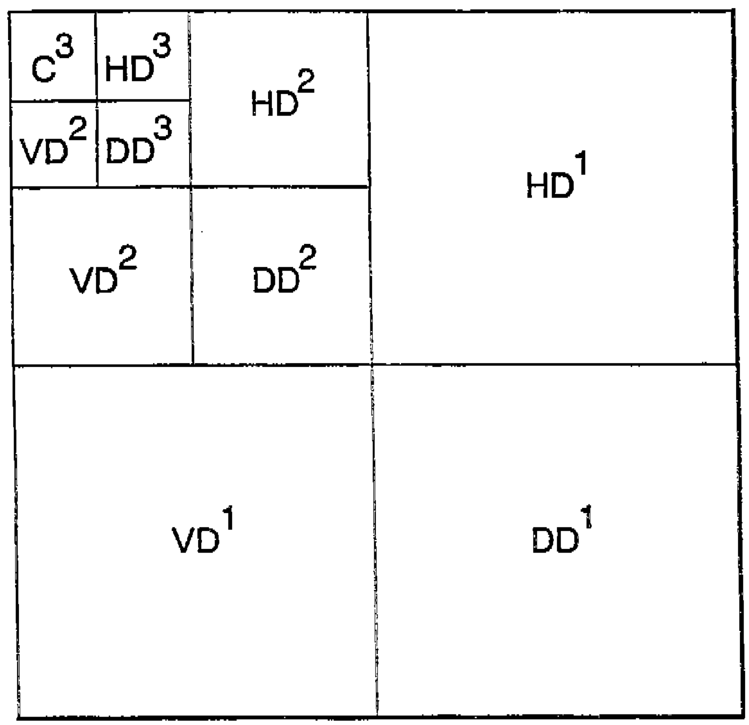
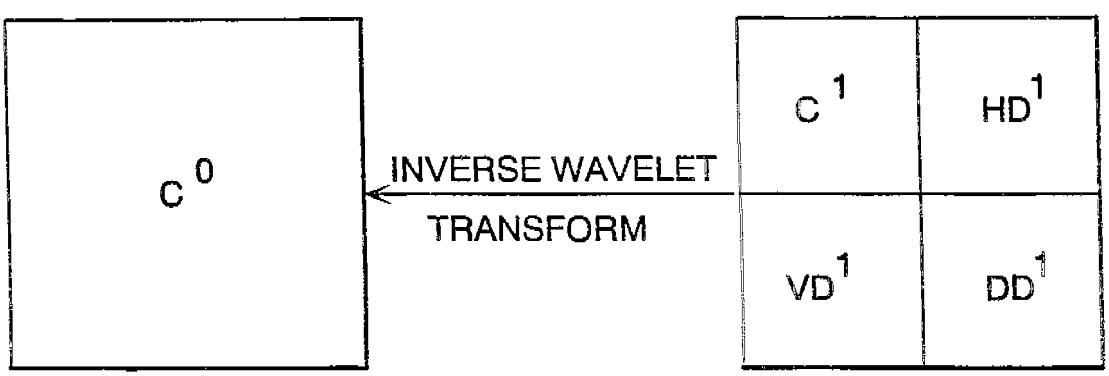

C 0
WAVELET
TRANSFORM
C 1
HD 1
VD 1
DD 1

Fig. 2



C^3
HD^3
HD^2
VD^2
DD^3
HD^1
VD^2
DD^2
VD^1
DD^1

Fig. 3



C^0
INVERSE WAVELET TRANSFORM
C^1
HD^1
VD^1
DD^1

Fig. 4

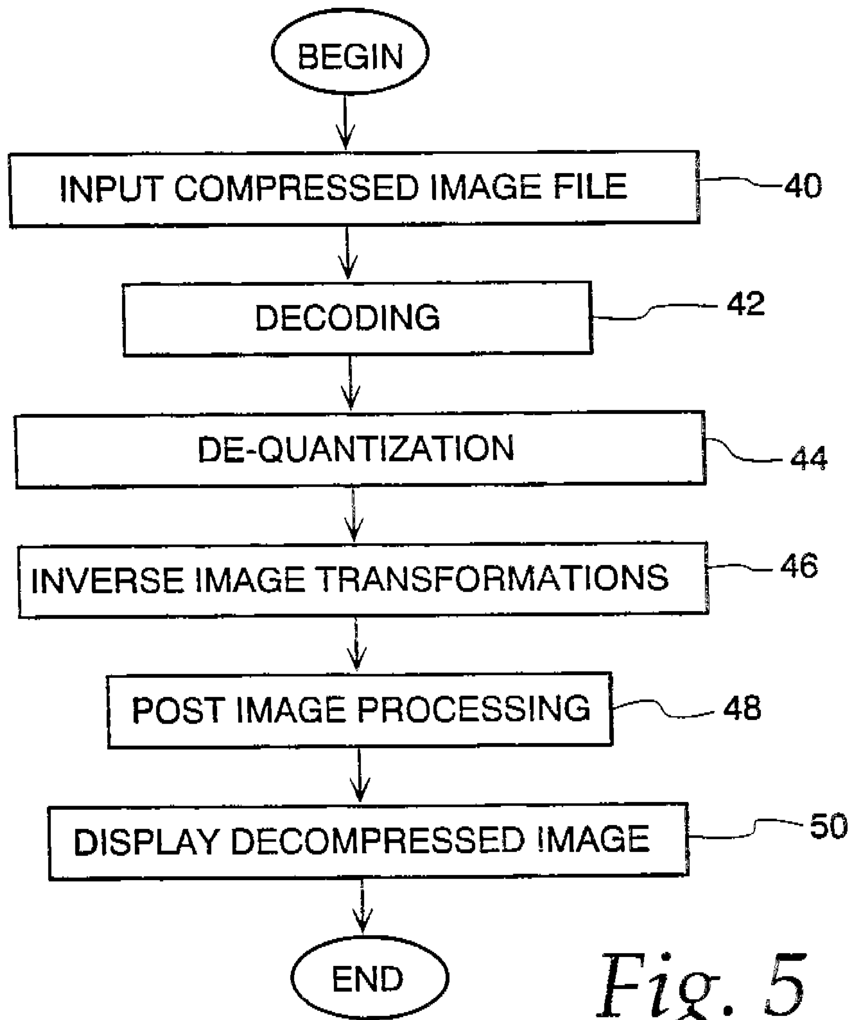

BEGIN
INPUT COMPRESSED IMAGE FILE
40
DECODING
42
DE-QUANTIZATION
44
INVERSE IMAGE TRANSFORMATIONS
46
POST IMAGE PROCESSING
48
DISPLAY DECOMPRESSED IMAGE
50
END

Fig. 5

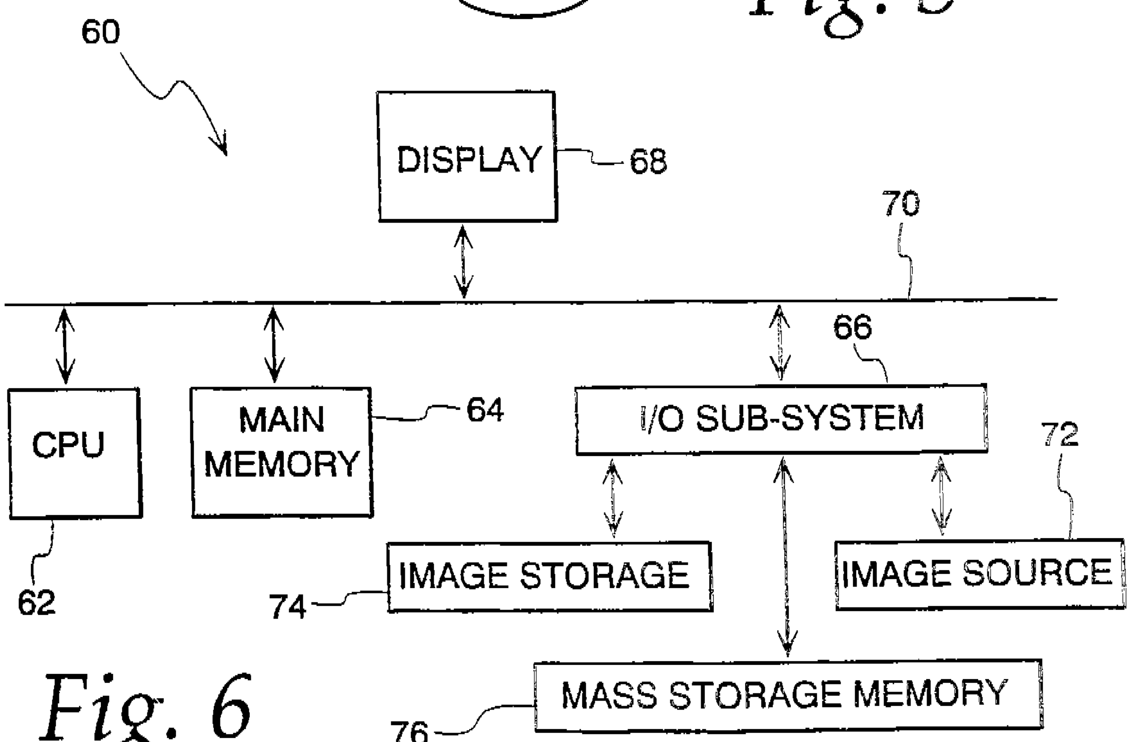

60
DISPLAY
68
70
66
CPU
MAIN MEMORY
64
I/O SUB-SYSTEM
72
62
IMAGE STORAGE
74
IMAGE SOURCE
MASS STORAGE MEMORY
76

Fig. 6

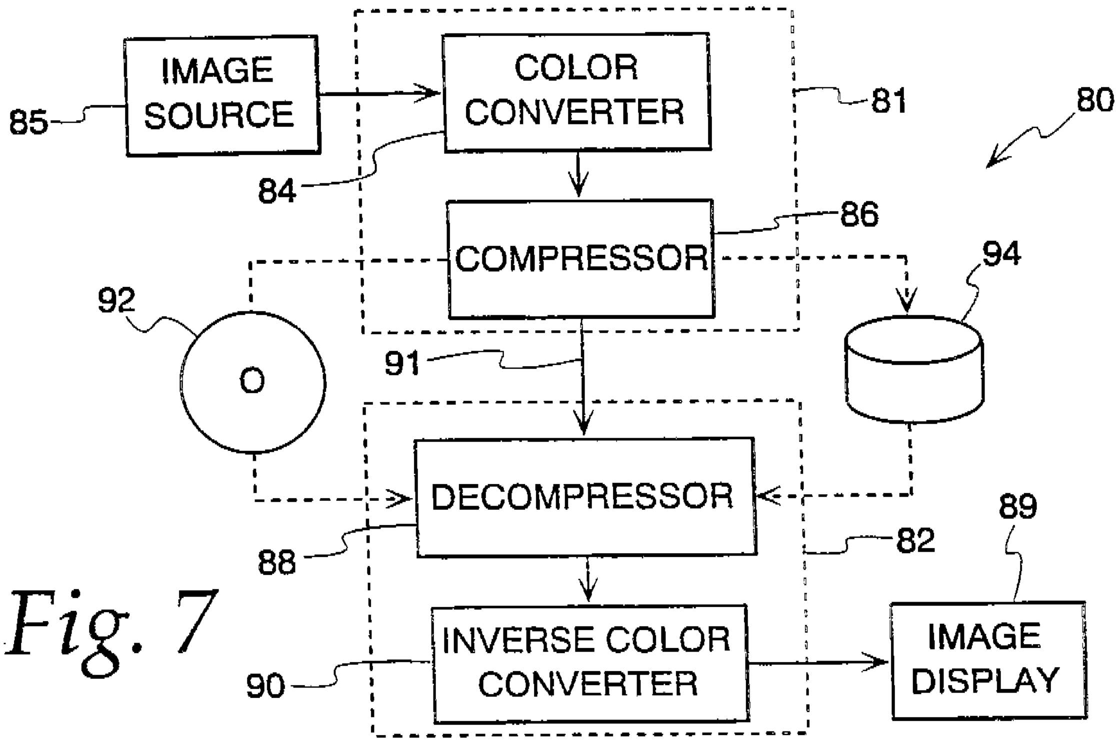


IMAGE SOURCE
COLOR CONVERTER
COMPRESSOR
DECOMPRESSOR
INVERSE COLOR CONVERTER
IMAGE DISPLAY
O
80
81
82
84
85
86
88
89
90
91
92
94

Fig. 7

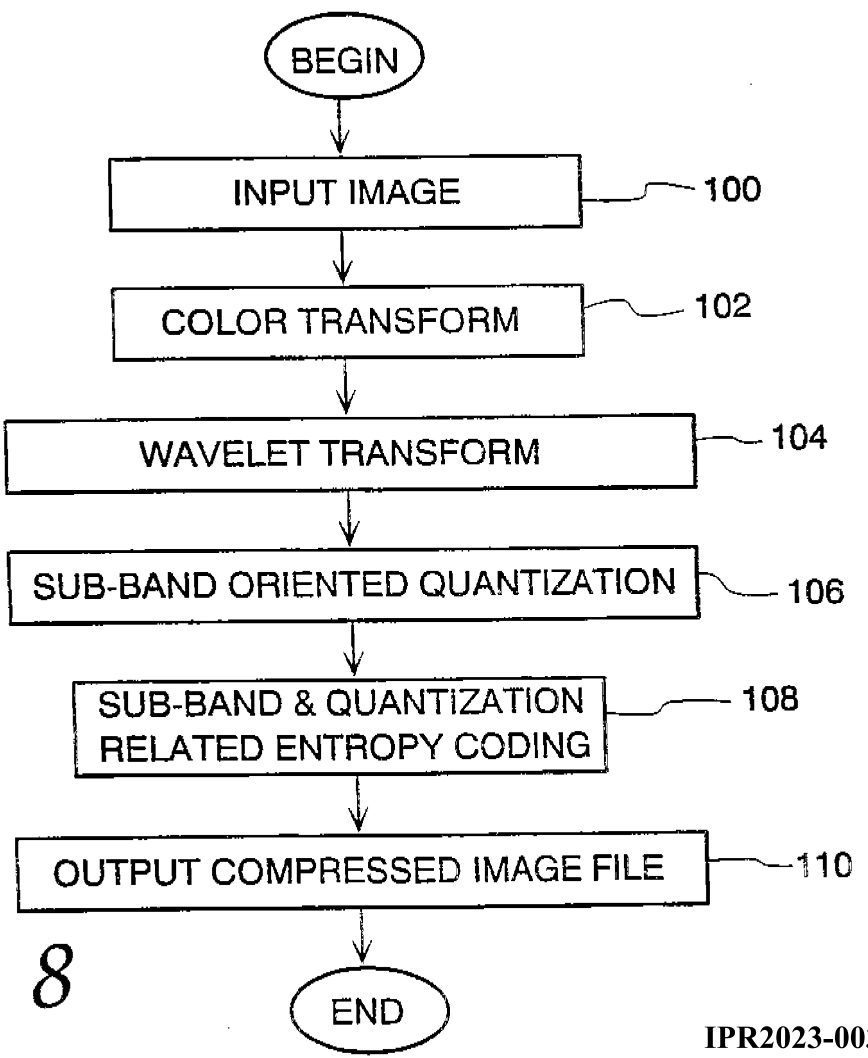

BEGIN
INPUT IMAGE
100
COLOR TRANSFORM
102
WAVELET TRANSFORM
104
SUB-BAND ORIENTED QUANTIZATION
106
SUB-BAND & QUANTIZATION RELATED ENTROPY CODING
108
OUTPUT COMPRESSED IMAGE FILE
110
END

Fig. 8

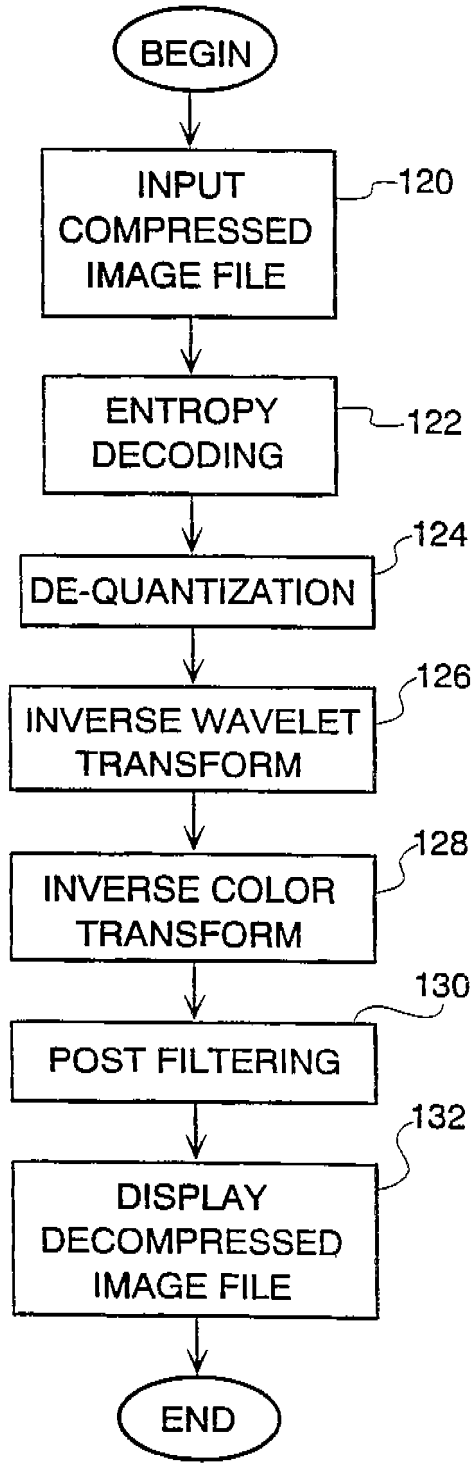

BEGIN
INPUT COMPRESSED IMAGE FILE
120
ENTROPY DECODING
122
DE-QUANTIZATION
124
INVERSE WAVELET TRANSFORM
126
INVERSE COLOR TRANSFORM
128
POST FILTERING
130
DISPLAY DECOMPRESSED IMAGE FILE
132
END

Fig. 9

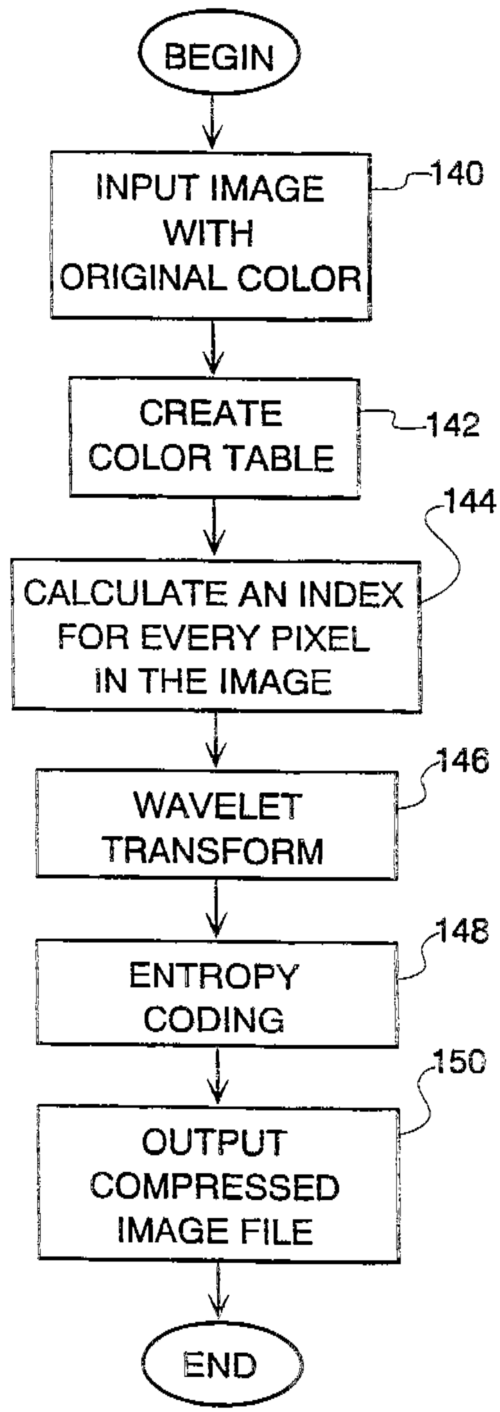

BEGIN
INPUT IMAGE WITH ORIGINAL COLOR
140
CREATE COLOR TABLE
142
CALCULATE AN INDEX FOR EVERY PIXEL IN THE IMAGE
144
WAVELET TRANSFORM
146
ENTROPY CODING
148
OUTPUT COMPRESSED IMAGE FILE
150
END

Fig. 10

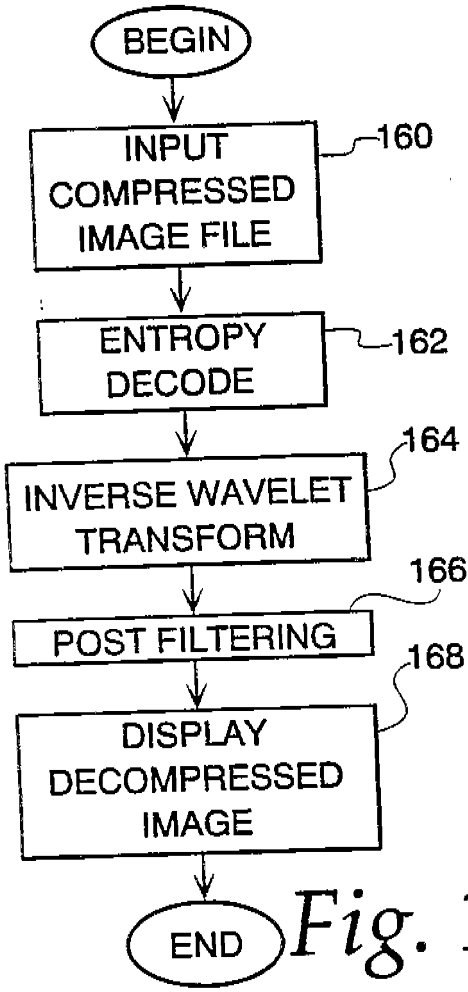


Fig. 11

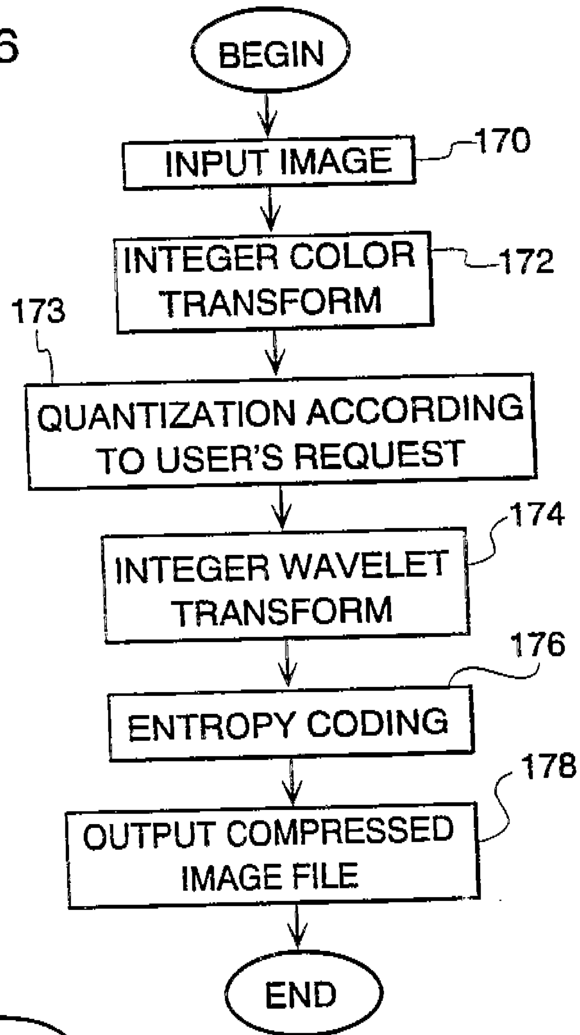


Fig. 12

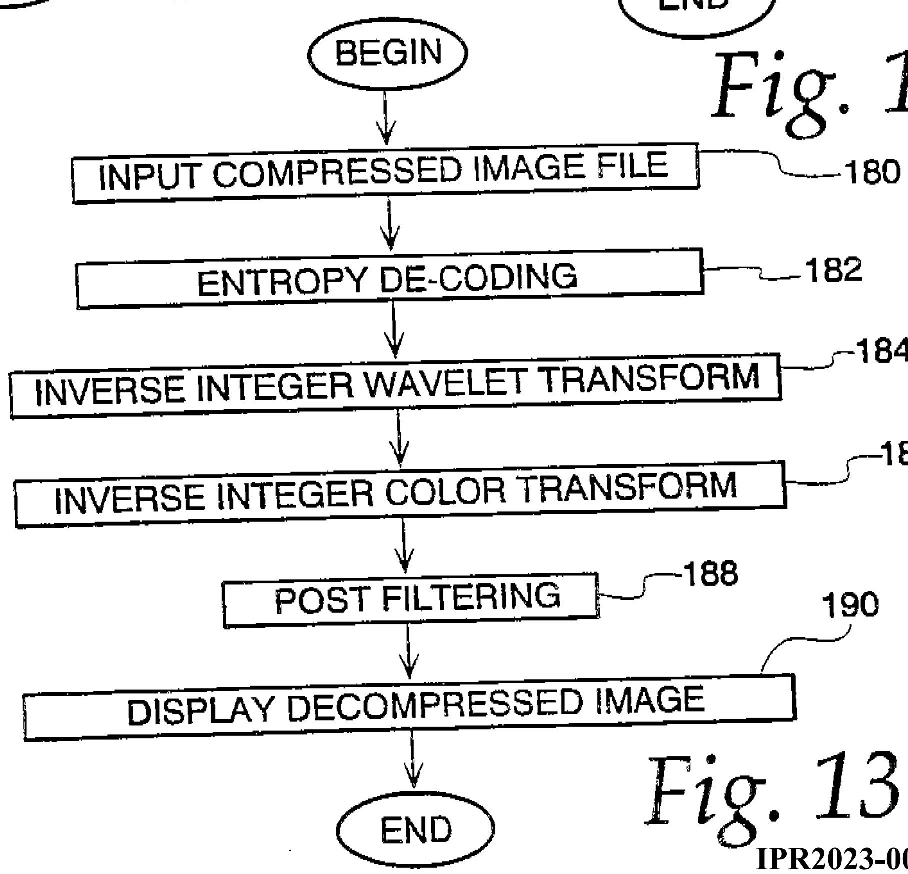


Fig. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/04700

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06K 9/00
US CL :Please See Extra Sheet.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 382/232, 233, 236, 238, 239, 240, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,495,292 A (ZHANG et al) 27 February 1996, col. 3, lines 1-68.	1-32
Y	US 5,414,780 A (CARNAHAN) 09 May 1995, col. 4, lines 1-68.	1-32

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
09 JUNE 1998	24 AUG 1998

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	LEO BOUDREAU

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

382/232, 233, 236, 238, 239, 240, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 13/38, 15/17

A1

(11) International Publication Number: WO 98/43177

(43) International Publication Date: 1 October 1998 (01.10.98)

(21) International Application Number: PCT/US98/05304

(22) International Filing Date: 19 March 1998 (19.03.98)

(30) Priority Data:

60/041,366	25 March 1997 (25.03.97)	US
08/925,275	8 September 1997 (08.09.97)	US

(71) Applicant: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, P.O. Box 58119, Santa Clara, CA 95052-8119 (US).

(72) Inventors: TSO, Michael, Man-Hak; 5744 S.E. Preston Court, Hillsboro, OR 97123 (US). WILLIS, Thomas, G.; 619 S.W. Arboretum Circle, Portland, OR 97221 (US). RICHARDSON, John, W.; 2748 N.E. 19th Avenue, Portland, OR 97212 (US). KNAUERHASE, Robert, Conrad; 4926 S.W. Corbett Avenue #108, Portland, OR 97201 (US). MACIELINSKI, Damien; 415 S.W. 121st Place, Portland, OR 97225 (US).

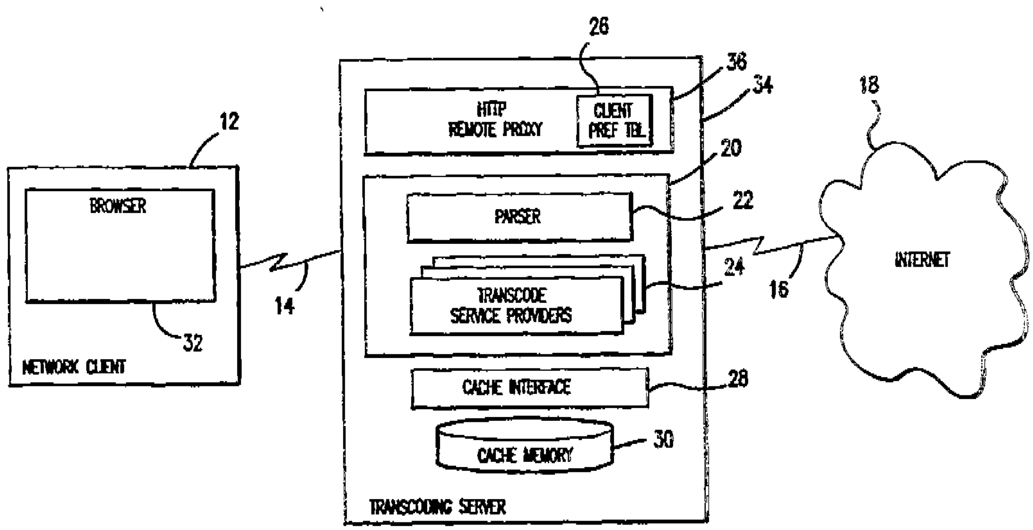
(74) Agents: ALTMILLER, John, C. et al.; Kenyon & Kenyon, 1025 Connecticut Avenue, N.W., Washington, DC 20036 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SYSTEM FOR DYNAMICALLY TRANSCODING DATA TRANSMITTED BETWEEN COMPUTERS



(57) Abstract

A system for dynamically transcoding data transmitted between computers is implemented in an apparatus for use in transmitting data between a network server (10) and a network client (12) over a communications link (14). The apparatus includes a parser (22) coupled to a transcode service provider (24). The parser (22) is configured to selectively invoke the transcode service provider (24) in response to a predetermined selection criterion.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

SYSTEM FOR DYNAMICALLY TRANSCODING DATA TRANSMITTED BETWEEN COMPUTERS

Background of the Invention

This application claims the benefit of U.S. Provisional Application No. 60/041,366, filed March 25, 1997.

Field of the Invention

The present invention relates generally to the field of data communications for personal computers (PCs), and in particular to a system for dynamically transcoding data transmitted between two computers over a communications link.

Related Art

The Internet is quickly becoming the preferred data communications medium for a broad class of computer users ranging from private individuals to large multi-national corporations. Such users now routinely employ the Internet to access information, distribute information, correspond electronically, and even conduct personal conferencing. An ever-growing number of individuals, organizations and businesses have established a presence on the Internet through "web pages" on the World-Wide Web (WWW).

For a wide variety of reasons, it may be desirable to manipulate data transmitted between a local client computer and a network server computer. For example, in certain instances it may be advantageous to dynamically add, modify or delete content retrieved from an Internet server computer before that content is provided to a client computer. Conversely, it may be advantageous to modify a content request from a client computer prior to transmitting the request to an Internet server computer. While such dynamic manipulation of requests and responses is desirable, it is impractical to expect the expansive Internet infrastructure to quickly change to accommodate such a new capability. For this reason, it is desirable to implement such new capabilities in a way that does not require changes to either existing client computers or Internet server computers.

It is known to deploy a proxy server, or network proxy, as an intermediary between one or more client computers and an external network such as the Internet. Network proxies are described generally in Ian S. Graham, HTML Source Book: A Complete Guide to HTML 3.0 403 (2d ed. 1996). One common application for a proxy server is as a so-called "firewall," wherein the proxy server is responsible for all communications with the outside world. In other words, local devices are not permitted to communicate directly with external network computers, such as Internet servers. Instead, each local device directs requests for network-resident data to the proxy server. When the proxy server receives such a request, it forwards the request to the appropriate external computer, receives the response from the external computer, and then forwards the response to the local device. The external computer thus has no knowledge of the local devices. In this way, the local devices are protected from potential dangers such as unauthorized access.

Existing proxy servers do not manipulate the data passing through them. In essence, proxy servers are merely blind conduits for requests and responses. This limitation of existing proxy servers restricts these devices from being used to full advantage when facilitating communications between local devices and network devices. There is therefore a need for a so-called "smart" proxy capable of examining the data passing through it, whether it be a request intended for an external network device or network content being returned to a local device, and dynamically acting upon that data. Such a device can be used to transparently provide a wide range of services that were heretofore impossible without modifying existing Internet infrastructure.

Summary of the Invention

Embodiments of the present invention relate to devices, systems and methods for transcoding information transmitted between computers, such as a network server computer and a network client computer.

According to one embodiment, an apparatus for use in transmitting data between a network server and a network client over a communications link includes a parser coupled to a transcode service provider. The parser is configured to selectively invoke the transcode service provider in response to a predetermined selection criterion.

Brief Description of the Drawings

Fig. 1 is a schematic diagram illustrating an environment in which embodiments of the present invention may be applied.

Fig. 2 is a schematic diagram illustrating a transcoder module according to an embodiment of the present invention.

Fig. 3 is a schematic diagram illustrating an embodiment of the present invention for a non-enabled network client.

Fig. 4 is a schematic diagram illustrating an example of a user interface for providing a non-enabled network client with control over transcoding functionality.

Fig. 5 is a schematic diagram illustrating an embodiment of the present invention for an enabled network client.

Fig. 6 is a schematic diagram illustrating a network client with transcoding functionality integrated in a browser according to an embodiment of the present invention.

Figs. 7-9 are flow charts illustrating logic for presenting a requested URL object to a network client according to an embodiment of the present invention.

Detailed Description

Embodiments of the present invention provide the ability to dynamically transcode information transmitted between, for example, a network server computer and a network client computer. As used herein, the term "transcode" applies to virtually any manipulation of data including, but not limited to, adding, modifying or deleting data.

Referring now to **Fig. 1**, which illustrates an environment in which embodiments of the present invention may be advantageously applied, a network server **10** manages the transfer of data from the Internet **18** to a network client **12**. Network client **12** may be any computer having suitable data communications capability.

Network client **12** communicates requests for information to, and receives information from, network server **10** over a client/server communications link **14**. Client/server communications link **14** may comprise, for example, a so-called "slow network" using, for example, POTS (Plain Old Telephone System) dial-up technology or wireless connections. Alternatively, client/server communications link **14** may comprise a so-called "fast network," such as a LAN or WAN (Wide Area Network), which is capable of operating at much higher speeds than are possible with slow networks. Combinations of these access methods are also possible. For example, network client **12** may use a POTS or wireless dial-up connection to a modem bank maintained by an ISP (Internet Service Provider), which is in turn connected to network server **10** over a LAN. Network server **10** communicates with computers resident on Internet **18** through server/network communications link **16**, which may comprise any suitable communications medium known in the art.

According to a first general embodiment of the present invention, illustrated schematically in **Fig. 2,** a transcoder **20** includes a parser **22** and a plurality of transcode service providers **24**. Parser **22** is configured to act upon data received by transcoder **20**, such as a request for a network object generated by a client device or a reply to such a request provided by a content server device. In this particular embodiment, parser **22** is responsible for selectively invoking one or more of transcode service providers **24** based upon a predetermined selection criterion.

Transcoder **20** may be implemented, for example, as a software module installed in a network proxy, in a client device, in a network server device, or in a content server device. In one particular implementation, illustrated in **Fig. 3**, transcoder **20** is installed in a remote transcoding server **34** arranged between network client **12** and Internet **18**. Transcoding server **34** may comprise, or be a part of, a network server, a stand-alone computer in communication with a network server, or a distributed system of computers. Remote transcoding server **34** may be coupled, for example, to an ISP's network, a corporate network, or anywhere on Internet **18**, and may provide multiple users (i.e., clients) with a means to obtain content on Internet **18**.

In the particular embodiment illustrated in **Fig. 3**, transcoding server **34** includes an HTTP (HyperText Transfer Protocol) remote proxy **36**, capable of accessing Internet **18** over server/network communications link **16**. HTTP remote proxy **36** differs from known network proxies, which generally are little more than a conduit for requests to, and replies from, external Internet resources, in that it is capable not only of examining such requests and replies, but also of acting upon commands in the requests by, for example, determining whether or not to transcode content. Moreover, using transcoder **20**, HTTP remote proxy **36** is capable of changing content received from Internet **18** prior to returning it to a requesting network client **12**, as is explained further below.

Looking more closely at the embodiment in **Fig. 3**, transcoder **20** is coupled to HTTP remote proxy **36**. Parser **22** manages the transcoding of data to be transmitted from transcoding server **34** to network client **12**. To this end, parser **22** controls transcode service providers **24** to selectively transcode content based on a predetermined selection criterion. For example, one or more transcode service providers **24** may provide the capability to compress and/or scale different types of data content, such as image, video, or HTML (HyperText Markup Language). Such uses are described further in co-pending U.S. patent applications Serial No. 08/772,164 entitled "System for Enhancing Data Access Over a Communications Link," filed on December 20, 1996, and Serial No. 08/799,654 entitled "Method and Apparatus for Scaling Image Data," filed on February 11, 1997, both of which are assigned to Intel Corporation. For purposes of illustrating certain features of the present invention, a number of embodiments are described below in terms of content scaling/compression; however, as is explained, transcode service providers **24** may provide a wide variety of transcoding functions.

As shown in **Fig. 3**, transcoding server **34** may also include a server-side cache memory **30** managed by a server-side cache interface **28**. Server-side cache memory **30** may be used to store both original and transcoded versions of content for later transmission to network client **12** without the need to re-retrieve the content from Internet **18** or to re-transcode the content.

Transcoding server **34** is coupled to network client **12** by client/server communications link **14**. Network client **12** includes a browser **32**, such as the Netscape Navigator v.3.0 browser (although the invention is not limited in this respect), which manages the presentation of data to a user. In this embodiment, network client **12** is "non-enabled," meaning no specialized transcoding software is preloaded on network client **12**.

Parser **22** may comprise a relatively simple, uniform interface to

HTTP remote proxy **36**, and may provide an API (Application Programming Interface) for transcoding data received by HTTP remote proxy **36**. Parser **22** manages one or more transcode service providers **24** that are accessed through a common SPI (Service Provider Interface). In this particular embodiment, parser **22** is designed in compliance with the Windows Open Systems Architecture (WOSA), and may be implemented as a Win32 DLL (Dynamic Link Library). The WOSA architecture, described in Readings on Microsoft Windows and WOSA (Microsoft Corp. 1995), enables additional transcode service providers **24** to be dynamically added to the system to provide new features and/or better transcoding algorithms, while at the same time not requiring changing or retesting other software components in the system. This feature is especially beneficial where transcoding server **34** also interacts with "enabled" network clients equipped with specialized transcoding software. It should be noted that some of the features of parser **22** described below may be inapplicable to the non-enabled client embodiment of **Fig. 3**; however, transcoding server **34** may advantageously be configured flexibly enough to process requests from both non-enabled and enabled network clients.

Like parser **22**, server-side cache interface **28** may be modeled after a standard Get/Set interface. Server-side cache memory **30** essentially "owns" all cached objects, in that it manages the properties and storage of the objects and may invalidate any non-locked object at any time; however, the actual format of any given cached object is known only by parser **22** and its associated transcode service providers **24**. Thus, for data integrity and transcoding efficiency purposes, all access to server-side cache memory **30** in this embodiment is through parser **22**.

Server-side cache interface **28** may include the following calls:

```
CreateEntry(URL, &Entry, ...);
GetEntry(URL, &Entry);
CreateStream(Entry, &StreamEntry, ...);
GetStream(Entry, &StreamEntry, ...);
CloseEntry(Entry);
CloseStreamEntry(StreamEntry);
```

GetProperties(Entry, &Properties, ...);
SetProperties(Entry, &Properties, ...);
Read(StreamEntry, &OutStream, ...);
Write(StreamEntry, &InStream, ...).

Unlike most cache memories, server-side cache interface **28** and server-side cache memory **30** enable maintenance of multiple representations of a given cached object, with descriptive information about each representation included in server-side cache memory **30**. In addition, server-side cache interface **28** and server-side cache memory **30** serve as a synchronization point for multi-threaded accesses to cached objects. It should be noted that the illustrated embodiment does not require any particular configuration for server-side cache interface **28** and/or server-side cache memory **30**. Indeed, functionality attributed to these components in the various embodiments described herein may be readily implemented in other system components.

The CreateEntry() call creates and returns a cache entry for a specified hypertext object. This call also creates an entry stream for an original version of the hypertext object. Similarly, the GetEntry() call obtains a cache entry for a hypertext object already existing in cache memory **30**. Both the CreateEntry() and GetEntry() calls set locks on associated cached objects until a CloseEntry() call is invoked. Once a lock is set, the cached object will not be replaced or invalidated by cache interface **28**, permitting one or more transcode service providers **24** to safely perform any required cache operations, such as object retrieval and/or storage.

After a cache entry is created or opened by a CreateEntry() or GetEntry() call, the CreateStream() or GetStream() calls may respectively create or open an extra stream entry for the cached object. Each extra stream entry is associated with a different transcoded version of the hypertext object, which may be retrieved or appended to by one of transcode service providers **24**. Stream-based processing of cached objects makes it possible for transcoding server **34** to begin transmitting a transcoded version of a hypertext object to a requesting network client

12 even while transcode service provider 24 is appending additional transcoded content to that same version. Advantages of this stream-based processing include reducing user latency through incremental painting of objects and avoiding unnecessary idle time on client/server communications link 14, thereby providing users with a more responsive "feel."

The GetProperties() and SetProperties() calls retrieve and store information about cached objects, including information maintained by transcode service provider 24 used to determine transcoding properties and transcoding status of a cached object. Transcode service provider 24 may use such information, for example, to determine current compression progress for scaled data access and staged refinements.

The Read() call reads data from a specified cached object data stream. For example, transcode service provider 24 may invoke this call and tunnel stream data through HTTP remote proxy 36 directly to network client 12. The Write() call caches data from a new HTTP data stream. This call will append an incoming data stream received from, for example, a Web server or transcode service provider 24, to an opened cache stream which may be concurrently read using the Read() call.

In the present embodiment, parser 22 includes the following calls:

GetObject(URL, InParams, &OutParams, &OutStream, ...);
GetScaledObject(URL, InParams, &OutParams, &OutStream, Stage, ...);
PutObject(URL, InParamStruct, &InStream, &OutParams, &OutStream, ...).

As detailed below, parser 22 uses these calls to manage the provision of requested content to network client 12.

The GetObject() call is used to service non-enabled client requests, and returns a non-transcoded (i.e., original) version of a specified hypertext object. In this embodiment, transcoding server 34 assumes that each HTTP request has a unique thread that may be blocked until the request is satisfied. Accordingly, the

GetObject() call will block until it either returns the requested data stream or indicates failure with a cause (e.g., object does not exist). This ability to return a so-called standard hypertext object is advantageous for compatibility reasons, enabling embodiments of the present invention to be used with existing browsers that do not include support for certain transcoding functionality (e.g., advanced data compression), and enabling users to selectively retrieve non-transcoded versions.

The GetScaledObject() call is similar to GetObject(), and is also used to request an object from server-side cache memory **30**; however, it adds support for requesting a particular version of that object, such as a high-quality rendition. Unlike traditional caching proxies, transcode service providers **24** can use server-side cache memory **30** to store several different versions of an object to support clients with different communications and/or presentation capabilities. Thus, an additional "Stage" parameter may be used to indicate which version of the cached object is to be returned to network client **12**. Where transcode service provider **24** is configured to scale network content, it may use this parameter to request a version of a cached object having, for example, a default scaled quality, a refinement to a better-quality version, or the original non-scaled version.

In this embodiment, when network client **12** requests a hypertext object, HTTP remote proxy **36** uses either the GetObject() or GetScaledObject() call (depending on if network client **12** is capable of receiving scaled/transcoded datatypes) to retrieve the hypertext object from parser **22**. If the hypertext object is not found, parser **22** uses the CreateEntry() call to create an entry (in effect, a placeholder) in server-side cache memory **30** for the new object. The new entry is returned to HTTP remote proxy **36**, which requests the hypertext object from Internet **18**. As a data stream for the hypertext object is returned, HTTP remote proxy **36** calls parser **22** using the PutObject() call, passing into this call the new entry and the handle to the data stream to be placed into the entry. Parser **22** selects an appropriate transcode service provider **24** based, for example, on the content type of the data stream. In this context, the term content type encompasses a datatype, an HTTP MIME (Multipurpose Internet Mail Extensions) type, a content format, and so on. The selected transcode service provider **24** uses a separate thread to read the incoming data stream, transcode it, and place it within the entry of server-side cache memory **30**. The current thread immediately returns to HTTP remote proxy **36**, which once again calls GetScaledObject() (or GetObject()). This case will always result in a cache hit. This thread then works simultaneously with the separate thread in the PutObject() to tunnel data (either original or transcoded) from transcoding server **34** to network client **12**.

Multiple-thread processing improves the efficiency of the present embodiment by not waiting for a hypertext object to be received in its entirety by HTTP remote proxy **36**, or added in its entirety to server-side cache memory **30**, before beginning to send the object to network client **12**. Another benefit of multiple-thread processing is that parser **22** may efficiently process requests for the same hypertext object from multiple network clients **12**. The hypertext object need only be retrieved from Internet **18** once, and appropriate versions may be transmitted to such multiple network clients **12** concurrently. It should be noted, however, that embodiments of the present invention may be implemented without multiple-thread processing.

As noted above, parser **22** may selectively invoke one of transcode service providers **24** based upon satisfaction of a predetermined selection criterion. Such selection criterion may comprise, for example, information contained in a header portion of a data packet received by transcoding server **34**, such as a MIME type, a URL (Uniform Resource Locator), a last modified time indicator and so on. Alternatively, the predetermined selection criterion may comprise information contained in a data portion of such a data packet, such as particular content, key words, structures (for example, heading levels), and so on. Still further, the predetermined selection criterion may comprise a condition of the device on which transcoding server **34** is installed (for example, a current processing load), a condition of a device to which transcoding server **34** is coupled, or a condition of a communications link. Transcoding server **34** may provide the ability to dynamically update such predetermined selection criteria.

The following discussion provides still more examples of the types of information which may be used to dictate which of transcode service providers **24** are invoked. It should be noted, however, that these examples are provided by way of illustration only, and are not intended to limit in any way the scope of the invention claimed herein. The predetermined selection criterion may comprise: (1) network client **12**, such as a display dimension, resolution, number of colors, processor type, memory/disk configuration, modem or network interface type, installed add-in boards (for example, hardware compression/decompression), software configuration (for example, availability of pre-installed software decompression modules), physical location/proximity (for example, as determined by a telephone area code), and user identity; (2) characteristics of transcoding server **34** or some other network server, including system load and identification information (for example, the owner of the server); (3) content characteristics, such as its data type, type of encoding/compression, size, and dimension; (4) network characteristics, including best-case, worst-case and average latency, bandwidth and/or error rates (for example, for wireless communications) between network client **12** and a proxy, and/or between a proxy and a server (this may be predetermined for guaranteed bandwidth links like ATM (Asynchronous Transfer Mode), or dynamically measured/predicted for so-called "best effort" links like many IP (Internet Protocol) links); (5) proxy characteristics, including system load, available storage, physical location/proximity, and identity (owner); (6) user preferences, including preferred content quality/speed tradeoff, language, content rating, exclusion list, inclusion list, data type-specific preferences (for example, "never download" images), include/exclude advertising, amount of advertising desired, offensive language removal, whether the user's defined or learned preferences may be disclosed (and to whom), custom rules or programs for filtering/transcoding/processing data, and shared preferences with either another user or a group of users (any of the foregoing user preferences may be explicitly defined or system predicated, such as based on usage statistics compiled over time); (7) group preferences, including results from collaborative rating systems, whether manual (for example, a prior user manually assigned a rating to a Web page after viewing it) or automatic (for example, given a large number of users who accessed a link on a given page, the probability of any given user subsequently following that link); (8) content provider preferences, including the degree of alteration desired for its content, the prioritization for download and display of different content types, cache restriction or prioritization parameters such as update frequency or replacement preferences, the types of users to target, rules or programs to run for customizing content (for example, news or advertising, custom language translation software) based on user or client characteristics, desire for receiving certain types of user or group data collected (for example, demographics or access patterns), and type of payment/reward offered in exchange for such information; and (9) other preferences, including software vendor rules or programs for dynamically checking content created or distributed using unauthorized software and companies' desire to enforce correct usage of certain types of content (for example, trademarks and logos).

Applying the above-listed selection criteria, or combinations thereof, embodiments of the present invention may be used to provide a virtually limitless range of dynamic transcoding services. For example, client and/or proxy physical proximity, in combination with demographic data, may be used for extremely targeted advertising. Such advertising may be added to any content passing through a proxy, for example, or some other mechanism. This can in turn be tailored even further based upon the user's willingness to tolerate advertising or share demographic information, as well as the advertiser's ability/willingness to subsidize or otherwise reward the user for participation.

Embodiments of the present invention may be advantageously used to reduce the amount of data that is transmitted to network client **12**, thereby promoting faster downloading and rendering of content. Suitable transcoding techniques include lossy compression and transcoding to a more efficient (and perhaps not widely supported) format specifically for the transmission. Similarly, HTTP remote proxy **36** may be configured to "predigest" Web sites or groups of sites to produce extremely condensed overviews of large amounts of content (for example, a tree structure, pages with only first-level or first- and second-level headings, thumbnails of pages, or only parts of a page or site that have changed since the user's last visit). Such applications may be especially advantageous for poorly-connected or computationally limited devices such as PDAs (Personal Digital Assistant), since this predigestion can be performed on a well-connected proxy server with an abundance of computational power, and the concise result can be easily downloaded and rendered on the more limited device.

Embodiments of the present invention may alternatively be used for dynamic translation of data, such as Web pages, to a user's native language (determined by user preference or automatically by the physical location of network client **12** or transcoding server **34**). Such a capability greatly simplifies the task of making content truly global, as well as reduces storage and maintenance required at the content provider (that is, only one copy of the content need be maintained, rather than different copies for each of a plurality of different languages).

Embodiments of the present invention may be used to block certain types of content or to automatically censor offensive language (similar to a "beep" used for television broadcasts). Only the particular offensive parts of the content (for example, obscene words) may be removed, or entire Web sites may be blocked. Similarly, transcoding server **34** may be configured to scan content for certain words or phrases to ensure that trademarks or logos are used correctly (for example, as a source identifier rather than a generic product designation). This feature may be offered as a service to companies or organizations, who would supply a list of words or phrases to flag. A similar capability could be used to automatically insert links into the content upon detection of certain words or phrases. For example, Intel Corporation might want to automatically add a link to its corporate Website whenever the name "Intel" is used in a Web page. Using an embodiment of the present invention, such links can be dynamically added to the content before it is displayed to a user. In a similar vein, an embodiment of the present invention may be used to scan for content that was created or distributed using unlicensed software. This feature may be implemented using special keys (binary bit patterns) embedded in the content or headers put in by the content creation or distribution software. The scanning logic and logic for taking a predetermined responsive action, such as denying service or posting a warning, may optionally be supplied by the vendor of the software in question or configured into transcoding server **34**.

Embodiments of the present invention may also be used to scan content for computer viruses prior to sending such content to network client **12**. For example, an existing virus scanning routine may be installed on transcoding server **34**, possibly as a plug-in module Transcoding server **34** may then be configured to invoke the virus scanning routine to ensure any content transmitted to network client **12** is free of viruses. A significant advantage provided by such an embodiment is that virus scanning software need only be maintained on transcoding server **34**, rather than on a plurality of network clients **12**. In this way, the benefit of upgrades to the virus checking software may be efficiently and timely provided to large numbers of users, thus avoiding the problem of any particular user relying on outdated virus scanning software.

Embodiments of the present invention may also be used to produce custom content on demand in accordance with user-specific preferences and/or associations with collaborative rating systems. In a variation on such an embodiment, transcoding server **34** can collect preferences and append them as part of a client request transmitted to a content provider so that the dynamic content generation can be done at the content server. Likewise, a proxy provider (for example, an Internet Service Provider (ISP)), can collect and make available to content providers information such as user preferences and data access statistics, as well as content provider specific statistics (for example, how many users from a given region or a given profile accessed a particular Web site, and at what time, in the past month). Such information may be used for applications such as targeted advertising.

Embodiments of the present invention may further be used to automatically check the validity of links in an object, and correct or remove invalid links, prior to transmitting the object to network client **12**. This capability may be provided, for example, as a service to content providers who may not have the most up-to-date information on Websites they are linked to which have moved or been deleted.

To further illustrate the general operation of the embodiment illustrated in **Fig. 3**, assume a user of network client **12** wishes to access a particular Web page, or URL (Uniform Resource Locator), on Internet **18**. Further assume that the desired URL resides on, or is accessible through, transcoding server **34**. Network client **12**, via browser **32**, transmits an HTTP request for the hypertext object to transcoding server **34** over client/server communications link **14**. Where browser **32** normally accesses Internet **18** through a proxy, browser **32** is configured to pass user requests through transcoding server **34** via browser's **32** standard proxy configuration procedures. As is well known in the art, browser **32** may actually transmit a plurality of additional HTTP requests corresponding to each of various distinct hypertext objects that may be embedded in the Web page. In such a case, transcoding server **34** may process each such request in the manner described below.

According to this embodiment, HTTP remote proxy **36** is capable of distinguishing between a non-enabled network client **12** and an enabled network client **12**. This may be accomplished, for example, using a private protocol to transmit content requests from an enabled network client to transcoding server **34**, so that the use of some other communications protocol indicates network client **12** is non-enabled. This method of sending a private protocol in each request to HTTP remote proxy **36** is an improvement over a registration type process. The overhead involved in making the enabled/non-enabled determination on a per request basis is relatively small, while providing a significant advantage because it addresses the situation for HTTP remote proxy **36** where a first network client disconnects and a second network client, likely with different communications and/or presentation capabilities, reconnects using the same IP address.

Upon determining that network client **12** is non-enabled, HTTP remote proxy **36** may record the IP address of network client **12** in a client preference table **26** maintained in a local data store (client preference table **26** may improve performance of this or other embodiments, but is not required). HTTP remote proxy **36** then passes the hypertext object to parser **22**. HTTP remote proxy **36** may also inform parser **22** of any applicable user preferences (e.g., from client preference table **26**). Upon being invoked, parser **22** first calls cache interface **28** with the requested hypertext object to determine whether a copy of the required version already resides in server-side cache memory **30**. For purposes of illustration, assume no entry exists in server-side cache memory **30** for the requested hypertext object. HTTP remote proxy **36** then invokes a call to retrieve the hypertext object from Internet **18** over server/network communications link **16**. Assuming the requested hypertext object is found, HTTP remote proxy **36** begins receiving an HTTP data stream representing the hypertext object. HTTP remote proxy **36** passes the handle for this incoming data stream to parser **22**.

Parser **22** dynamically determines whether the data stream satisfies any applicable predetermined selection criteria. For example, where transcode service providers **24** are configured to scale data of different types, parser **22** may determine the content type for the data stream (e.g., image/jpeg, image/gif, video/mpeg) by interrogating a MIME type in the content-type header record that appears at the beginning of the incoming HTTP data stream. If parser **22** detects a match for a predetermined selection criterion, the HTTP stream handle is given to the appropriate transcode service provider **24**. Transcode service provider **24** then transcodes the data stream appropriately, and HTTP remote proxy **26** transmits the transcoded data stream to network client **12**.

A non-enabled network client **12** may optionally be provided with the ability to actively control aspects of the transcoding process, or indeed whether or not to transcode requested content at all. To provide this ability, HTTP remote proxy **36** may embed additional instructions at the beginning of the HTML header for the requested URL prior to transmitting the associated data stream to network client **12**. These embedded instructions may be implemented, for example, as JavaScript codes, VB Script codes or Java Applet codes. As browser **32** of network client **12** receives the data stream, the embedded instructions will automatically execute so long as browser **32** is equipped to support them. For example, if the embedded instructions are implemented as JavaScript codes, browser **32** may be a JavaScript-enabled browser such as a Netscape Navigator v.2.0 or above browser, or an Internet Explorer v.3.0 or above browser. If browser **32** is not equipped for such HTML scripting, the embedded instructions will not interfere with the browser's **32** normal processing, as such browsers **32** are typically configured to ignore any data they cannot interpret.

The embedded instructions transmitted to network client **12** may enable the user to manipulate some of the transcoding capabilities of transcoding server **34**. As illustrated in **Fig. 4**, the embedded instructions may drive a user interface in the form of a pop-up window **40** that is displayed at the top of a browser window **38**. Pop-up window **40** includes a three-state switch **42** having "ON," "OFF" and "AUTO" settings, and may also include a hypertext link **44** which the user may follow to download specialized client software supporting, for example, more sophisticated transcoding functionality (i.e., become "enabled"). The initial setting of three-state switch **42** may be based upon a prior determination by HTTP remote proxy **36** as to whether network client **12** has an established preference for reception of transcoded content. If so, three-state switch **42** may be set to "ON;" if not, three-state switch **42** may be set to "OFF." A goal of this feature is to provide the user with some means for communicating a preference to HTTP remote proxy **36** with regard to aspects of particular transcoding features, such as a content quality/latency tradeoff where the transcoding comprises data compression/scaling. Persons skilled in the art will recognize that many other means for providing this capability are possible, and such other means could enable the user to communicate preferences beyond simply a yes/no indication for transcoding.

In the illustrated in **Fig. 4**, pop-up window **40** enables the user to change his or her preference as to whether transcoded or original content is desired, and communicates such changes to HTTP remote proxy **36**. Pop-up window **40** may or may not interact with browser **32**, meaning the user's preference will only take effect after setting three-state switch **42** and clicking on the browser's **32** "RELOAD" button **46** to cause browser **32** to request the (transcoded or un-transcoded) content for presentation to the user. Subsequent pages in the current session may then be rendered in accordance with the new setting of three-state switch **42** without further user intervention. Upon receipt, HTTP remote proxy **36** may update user preference table **26** accordingly. As an alternative, pop-up window **40** may be configured to automatically invoke the "RELOAD" operation when the user indicates a change (such as by flipping three-state switch **42**). Where browser **32** is a JavaScript-enabled browser, JavaScript instructions inserted by HTTP remote proxy **36** in the HTML document may "POST" the state of three-state switch **42** to HTTP remote proxy **36** and also cause browser **32** to "RELOAD" the current URL.

It is possible to allow a non-enabled network client **12** to save the state of three-state switch **42** on network client **12** across multiple sessions of browser **32** using what is known in the art as a "cookie." In other words, a cookie may be used to store the state of three-state switch **42** persistently. When a new session of browser **32** is initiated by a user, this state information may be read from network client **12** and "POSTed" by the JavaScript code (inserted at the beginning of the HTML document) to HTTP remote proxy **36** before any content for the requested hypertext object is actually sent to network client **12**. This will allow HTTP remote proxy **36** to update user preference table **26** with the correct state of three-state switch **42**, and hence send correctly-transcoded content to network client **12**. In such an embodiment, the state information may be "POSTed" to HTTP remote proxy **36** each time a given URL is requested by browser **32**. This will allow network client **12** to receive the correctly-transcoded content even if the HTTP remote proxy **36** to which it is coupled changes due to, for example, a change in geographical location of network client **12** or network load-balancing procedures.

The embodiment shown in **Fig. 3** may also be used for network clients **12** that already access Internet **18** through a standard proxy. JavaScript-enabled browsers **32** may query the local IP address of network client **12** and "POST" this information to HTTP remote proxy **36**. The HTTP header of this "POST" message will contain the IP address of the standard proxy, which will now be different from the IP address of network client **12** (which is included in the contents of the message). A comparison of the two IP addresses will determine whether network client **12** resides behind a standard proxy. HTTP remote proxy may then use this information to update transcoding information about network client **12** in user preference table **26**.

According to another embodiment of the present invention, illustrated in **Fig.** 5, network client **12** may be "enabled," containing specialized software to support, for example, more sophisticated transcoding features than are provided by the above-described embodiments, or to perform some or all of the transcoding functions on the client side. As illustrated, network client **12** includes an HTTP local proxy **48** coupled to a client-side parser **50** which, similar to parser **22** of transcoding server **34**, controls one or more client-side transcode service providers **52**. Each transcode service provider **52** may be configured, for example, to transcode content before it is rendered to a user or to perform a counterpart transcoding function (e.g., decoding, decompression) with respect to a function performed by a corresponding transcode service provider **24** of transcoding server **34**. As in transcoding server **34**, network client **12** may include a client-side cache memory **56** managed by a client-side cache interface **54**. Client-side cache interface **54** may be an already-existing facility supported by the operating system, such as WININET. Using an existing caching facility reduces the amount of software that is to be downloaded to network client **12** to implement this embodiment, and also allows other applications, such as disconnected browsers, to share client-side cache memory **56**.

HTTP local proxy **48**, client-side parser **50** and client-side transcode service providers **52** (collectively, the client software) may be downloaded to network client **12** on demand, such as by clicking on hypertext link **44** presented by pop-up window **38** illustrated in **Fig. 4**. Alternatively, the client software could be distributed to users on a portable storage medium, such as a diskette or CD-ROM, or it may be preloaded on an off-the-shelf personal computer. In the embodiment of **Fig. 5**, the client software is separate from browser **32**; however, in yet another embodiment the client software may be integrated in browser **32** (see **Fig. 6**).

The enabled client embodiments provide network client **12** with expanded flexibility for rendering hypertext objects. As in the non-enabled client embodiments described above, enabled network client **12** may receive a transcoded data stream from HTTP remote proxy **36** in a format that is already supported by the standard internal rendering software of browser **32** (e.g., JPG, GIF). This would be the case where, for example, the transcoding process involved adding or deleting text to the hypertext object. In addition, HTTP remote proxy **36** may transcode a hypertext object to a data stream having a new MIME type, such as where the transcoding process comprised scaling or data compression, in which case a client-side transcode service provider **52** could be provided to convert the data stream back to a MIME type supported by browser **32**. For example, HTTP remote proxy **36** could transmit a file compressed using a non-standard, not well-supported but leading-edge compression algorithm to network client **12**, and client-side transcode service provider **52** could uncompress the file back to its original format. This approach has the benefit of relieving HTTP local proxy **48** from having to provide a user interface, and eliminates restrictions imposed by limitations as to the data types supported by browser **32**. In this way, the transcoding process can remain transparent to users, browsers and Web servers even when it involves changing content to different datatypes.

Yet another possibility is that enabled network client **12** includes one or more add-ins **46** specifically configured to transcode, render or playback content received by network client **12**. Add-ins **46** may be implemented, for example, using Netscape plug-ins or ActiveX controls. Moreover, add-ins **46** may be installed as part of the client software, as illustrated in **Fig. 5**, or integrated with browser **32**. Such add-ins **46** are beneficial in that they generally may be configured to permit a user to click on a specific object to obtain a different version (e.g., higher quality) representation. Add-ins **46** are also beneficial in that they appear to a user to be well-integrated with browser **32**, and are easily upgradeable. Combinations of the above-described presentation facilities are also possible.

In an advantageous optional application of add-ins **46**, network client **12** may be configured to request that an appropriate add-in **46** be downloaded from HTTP remote proxy **36** in the event that network client **12** determines it is unable to transcode a particular received data stream. HTTP remote proxy **36** could then download the necessary add-in **46** or, alternatively, resend the data stream in a different format. This facility provides for automatic extension of the system, ensuring that client software is as current as possible.

In the embodiment of **Fig. 5**, browser **32** is configured to send all HTTP requests through HTTP local proxy **48**, thus allowing HTTP local proxy **48** to improve retrieval and rendering of requested hypertext objects. For example, when HTTP local proxy **48** receives an HTTP request from browser **32** for a hypertext object associated with a Web page, it passes the URL to client-side cache interface **54** to check whether a copy of the hypertext object already exists in client-side cache memory **56**. If the hypertext object is cached, HTTP local proxy **48** passes the cached object to browser **32** for rendering. If the requested hypertext object is not cached, HTTP local proxy **48** transmits an HTTP request to transcoding server **34** for processing. HTTP local proxy **48** may use a custom Get() request for this purpose to enable transcoding server **34** to identify network client **12** as enabled. Performing the processing described above with reference to other embodiments, transcoding server **34** will return a data stream for the hypertext object to HTTP local proxy **48**.

To further illustrate the features and benefits of embodiments of the present invention, the flow charts provided in **Figs. 7-9** illustrate the logic for an embodiment of a method by which an enabled network client may render a hypertext object resident on the Internet. The flow charts are not intended to be comprehensive of all processing that is performed, but rather are intended to describe the overall flow of the method. Detailed descriptions of the various processes have been provided above with reference to various disclosed embodiments. Where practical, the following description includes reference numbers for previously-described structural elements, although the method is not limited to those structures.

Referring now to **Fig. 7**, processing begins when a user on network client **12** requests a hypertext object from browser **32 (Step 100)**. This could be in the form of a request for a specific Web page, in which case a plurality of hypertext objects will likely be displayed to the user, or in the form of a click on an image already being displayed to the user. Browser **32** may be configured to pass all HTTP requests through HTTP local proxy **48**, so HTTP local proxy **48** may intercept the HTTP(URL) request from browser **32** (**Step 110**).

In this particular embodiment, HTTP local proxy **48** first checks whether the requested hypertext object exists in client-side cache memory **56** (**Step 120**). To do this, HTTP local proxy **48** may invoke client-side parser **50** using a GetScaledObject(URL) call, which in turn issues a GetEntry call to client-side cache interface **54** to open a stream for the cached object. This effectively "retrieves" the cached object from client-side cache memory **56** if it exists (**Step 140**). HTTP local proxy **48** then passes the stream to browser **32**, which displays the cached object to the user (**Step 150**).

Referring now to **Fig. 8**, if the requested URL object is not found in client-side cache memory **56**, HTTP local proxy **48** transmits a request for the object to transcoding server **34** using, for example, a Post of a GetStage(URL, Stage=0) call (**Step 160**). Upon receipt of this call, HTTP remote proxy **36** invokes parser **22**, which in turn issues a GetScaledObject() call to server-side cache interface **28** to determine whether a non-transcoded version of the requested hypertext object already exists in the server-side cache memory **30** (**Step 170**). If the hypertext object is cached, server-side cache interface **28** issues a GetEntry call to open a stream for the cached object (**Step 200**). In addition, parser **22** may issue a GetProperties(URL, ...) call to server-side cache interface **28** to retrieve information about the transcoding properties and transcoded status (such as the refinement level) of the cached object.

If parser **22** determines that the requested hypertext object does not exist in the server-side cache memory **30**, HTTP remote proxy **36** issues an HTTP request to retrieve the hypertext object from Internet **18** (**Step 190**). If the object is not found, HTTP remote proxy **36** returns an error to network client **12** which browser **32** will communicate to the user (**Step 220**); if the object is found, HTTP remote proxy **36** passes the handle for the incoming data stream to parser **22**, which in turn initiates caching of an original version of the retrieved hypertext object (**Step 230**).

Referring now to **Fig. 9**, once the requested hypertext object has started to be obtained, parser **22** determines whether (and how) to transcode the object before transmitting it to network client **12** (**Step 240**). Both this decision-making process and exemplary transcoding processes are described in detail above. For purposes of the present illustration, assume parser **22** determined that transcoding was appropriate and therefore generated a transcoded version of the requested hypertext object (**Step 250**). HTTP remote proxy **36** transmits a data stream for the transcoded hypertext object to network client **12** (**Step 260**). Upon receipt, HTTP local proxy **48** initiates caching of the transcoded hypertext object (**Step 270**). In addition, client-side parser **50** determines whether any further processing is required before the hypertext object is rendered (e.g., a new MIME type has been established by transcoding server **34) (Step 280**).

If no additional transcoding is required, HTTP local proxy **48** passes the handle for the received data stream to browser **32** for display to the user (**Step 290**). If additional transcoding is required, client-side parser **50** passes the handle to an appropriate transcode service provider **52** (**Step 300**). The result of this latter processing may be a hypertext object which browser **32** can readily display to the user (**Step 320**), or the result may be a hypertext object having a non-standard MIME type, in which case browser **32** may invoke add-in **46** to display the object (**Step 330**).

According to another embodiment of the present invention, additional data or programs need not necessarily be inserted as part of a response to a client request. Rather, data and programs may be transparently "pushed" to network client **12** without the user or the browser **32** software's detection or intervention. One advantage of this approach is that transcoding server **34** is able to detect when client/server communications link **14** is underutilized, and can thus push data to network client **12** with limited risk of interfering with other transactions. An especially advantageous implementation uses at least a local proxy, which could issue its own requests (rather than being user-driven) to content providers or networked proxy servers, or receive unsolicited data pushed to it from the network. The local proxy may store the data in a client-side cache, install it as a program, or prompt the user to take some further action. Many potential uses for such an embodiment are possible. For example, an advertiser of software products or music can preload network client **12** with trial versions of products before prompting the user with an advertisement, thus enabling instant playback capability without the user having to wait for a demo to be downloaded (and possibly losing interest in the meantime).

A number of different configurations are possible for implementing embodiments of the present invention. In a first configuration, the only additional apparatus required is a remote proxy. That is, no new software needs to be installed on network client **12**. The remote proxy may reside anywhere on a suitable network, such as the Internet, including at particular content provider sites. Alternatively, the remote proxy may be located at ISP local POPs (Point of Presence), for example, if location-specific characteristics are to be used as predetermined selection criteria. Of course, such information can be gathered by other methods as well, such as user-preference settings or assigning location-specific domain names to proxies. In a second configuration, a new piece of client software acting as a local proxy may be installed, for example, on a client device. The user would then point the client application's proxy to the local host. Combinations of these exemplary configurations are likewise possible, as well as simultaneously having multiple modes active (for example, a local proxy acting as a pass-through for some requests and a non-pass-through for others that require the use of a remote proxy).

Where network client **12** connects to a remote proxy over a relatively slow communications link, it may be particularly advantageous to implement transcoding and link validity checking on remote proxies. Combinations of remote and local proxies can sometimes give more efficient implementations of certain applications, such as automatic data/program download and interactively displaying predigested content. Other applications, such as translation and trademark enforcement, can be done efficiently on local proxies alone, but may be more advantageously done on remote proxies because the results can be cached for use by others, thereby saving resources for future requests. Still other applications, such as clickstream analysis, are generally better implemented on a local proxy because there are more resources available locally to the individual user, and also for privacy reasons.

In view of the foregoing description, it should be apparent that it is possible for there to be more than one so-called "smart" proxy arranged between a client device and a content server device. If left unchecked, such a condition can result in content being altered excessively (for example, too many ads inserted, multiple lossy compressions resulting in indecipherable images). To address this problem, an embodiment of the present invention may use a special proxy-to-proxy protocol that extends the existing request/response structures to indicate whether and what sort of transcoding has already been performed on the content. Such a specialized protocol, in addition to other proxy-to-proxy messages which may be implemented on an as-needed basis, enables multiple proxies to work collaboratively, yet still transparently to users, client software, existing "standard" proxies and content servers.

According to yet another embodiment of the present invention, a proxy server may be used to provide certain Internet proxy or server users with so-called "VIP" treatment, identifying users who are entitled (either through payment or based on some other selection criterion, such as extent of usage) to have a higher priority when competing with other users for proxy resources. By contrast, with existing Internet proxies and servers, users are serviced either on a random or first-come/first-served basis.

In one particular implementation of such an embodiment, transcoding server 34 may be configured to extract user IP addresses from requests it processes and maintain information such as how frequently, or for what duration, a user is browsing a particular Web site. Such information could be used to determine "frequent browser miles" at particular Web sites. Users can then be rewarded with faster response times for subsequent visits to the site, or the site owner could choose to reward the user with improved performance on all sites reached through the same proxy. Still another possibility is that users may pay for such preferred service, being assigned a password which may be provided to transcoding server 34. Yet another possibility is that a Web site owner can pay a proxy provider to improve the performance of all users while visiting the owner's site.

In another particular implementation, information identifying users to be given VIP treatment may be passed to transcoding server 34 in the form of a Web page. Upon receipt of such a Web page, the proxy may subsequently allow servicing threads to perform work for requests generated by VIP users first. To do this, transcoding server 34 may boost thread scheduling priorities (within the operating system) for the VIP service, while ensuring there is no starvation of any thread (that is, no user should be denied access entirely by VIP users). In addition, transcoding server 34 may permit preferential caching for particular Web sites and more aggressive pre-fetching for VIP users. Still further, transcoding server 34 may use more resource-intensive compression algorithms, for example, to provide better quality content for the same latency at the expense of slowing down access for non-VIP users.

It is possible that certain content providers or users will not wish to have their content dynamically altered in any manner. Accordingly, embodiments of the present invention may be implemented in such a way that either content providers or users are given the capability to override any potentially content-altering service.

This may be accomplished, for example, using a pass-through technique triggered by a special tag embedded within the content.

As the foregoing description demonstrates, embodiments of the present invention may be used to provide a system for improving the communications capabilities of computers accessing networks such as the Internet. Embodiments of the invention may be advantageously applied to computers having limited communications bandwidth available, such as mobile computers or personal computers accessing a network over a modem connection. The unique features of such embodiments enhance the ability of these computers to access data on the network in a timely fashion with reduced user-visible latencies, thereby enabling content authors to produce rich content without fear that only users with highly-sophisticated data communications and display capabilities are able to enjoy it. Embodiments of the present invention may also be advantageously used for purposes other than, or in addition to, reducing latency. Such purposes include, for example, converting color images to greyscale images for users lacking a color display; filtering and/or deleting undesired content, such as pornography; adding content, such as advertising; and language translation.

Although the present invention has been described with reference to embodiments for accessing data from the Internet, persons skilled in the art will recognize that it is equally applicable to other networking environments. For example, embodiments of the present invention may be used to enhance data communications between a network client computer and an "intranet." An intranet typically is a secure corporate network modeled after the Internet architecture, and generally includes mechanisms for communicating with external networks such as the Internet.

The foregoing is a detailed description of particular embodiments of the present invention. The invention embraces all alternatives, modifications and variations that fall within the letter and spirit of the claims, as well as all equivalents of the claimed subject matter. For example, some or all of the features described above as being provided by a remote proxy may be implemented in a content server. Likewise, some or all of the features described above as being provided by a local proxy may be implemented in a browser application. Persons skilled in the art will recognize from the foregoing detailed description that many other alternatives, modifications and variations are possible.

What Is Claimed Is:

1. An apparatus for use in transmitting data between a network server and a network client over a communications link, said apparatus comprising a parser coupled to a transcode service provider, said parser being configured to selectively invoke said transcode service provider in response to a predetermined selection criterion.

2. The apparatus of claim 1, wherein said predetermined selection criterion comprises a characteristic of the data being transmitted.

3. The apparatus of claim 1, wherein said predetermined selection criterion comprises a characteristic of the communications link.

4. The apparatus of claim 1, wherein said predetermined selection criterion comprises a characteristic of the network server.

5. The apparatus of claim 1, wherein said predetermined selection criterion comprises a characteristic of the network client.

6. The apparatus of claim 1, wherein said predetermined selection criterion comprises a user preference.

7. The apparatus of claim 1, wherein data is transmitted from the network server to the network client in response to a request by the network client, said predetermined selection criterion being included in said request.

8. A method for providing a network client with a data object residing on a network server, wherein the network client and the network server are coupled by a communications link, said method comprising the steps of:

receiving a data object from the network server;

selectively transcoding the data object according to a predetermined selection criterion; and

providing the data object to the network client.

9. The method of claim 8, wherein said transcoding step further comprises comparing a characteristic of the received data object to the predetermined selection criterion.

10. The method of claim 8, wherein said step of selectively transcoding the data object further comprises determining whether the data object includes content created with an unregistered software product.

11. The method of claim 10, wherein said step of selectively transcoding the data object further comprises adding a message to the data object corresponding to said detection of content created with an unregistered software product.

12. The method of claim 8, wherein said step of selectively transcoding the data object comprises compressing a portion of the data object.

13. The method of claim 8, wherein said step of selectively transcoding the data object comprises translating a portion of the data object from a first language to a second language.

14. The method of claim 8, wherein said step of selectively transcoding the data object further comprises determining whether the data object includes offensive content.

15. The method of claim 14, wherein said step of selectively transcoding the data object further comprises modifying the data object to prevent offensive content from being rendered by the network client.

16. The method of claim 8, wherein said step of selectively transcoding the data object further comprises adding advertising information into the data object.

17. The method of claim 16, wherein said advertising information is selected in accordance with user profile information.

18. The method of claim 8, wherein said step of selectively transcoding the data object further comprises determining whether the data object includes a link to a second data object.

19. The method of claim 18, further comprising the step of validating the link to a second data object.

20. The method of claim 19, wherein said step of selectively transcoding the data object further comprises correcting an invalid link.

21. The method of claim 8, wherein said step of selectively transcoding the data object further comprises communicating information relating to said transcoding to the network server.

22. The method of claim 8, wherein said step of selectively transcoding the data object further comprises determining whether the network client is preconfigured to receive preferential treatment of requests.

23. A set of instructions residing on a storage medium for execution by a computer, the computer being coupled to a device for rendering a data object to a user, said set of instructions comprising instructions for:

parsing a data object to be rendered to detect content corresponding to a predetermined selection criterion;

selectively transcoding the data object in response to said detection prior to rendering the data object.

24. The set of instructions of claim 23, wherein the storage medium comprises a magnetic storage device.

25. The set of instructions of claim 23, wherein the storage medium comprises a memory installed in a computer.

1/9

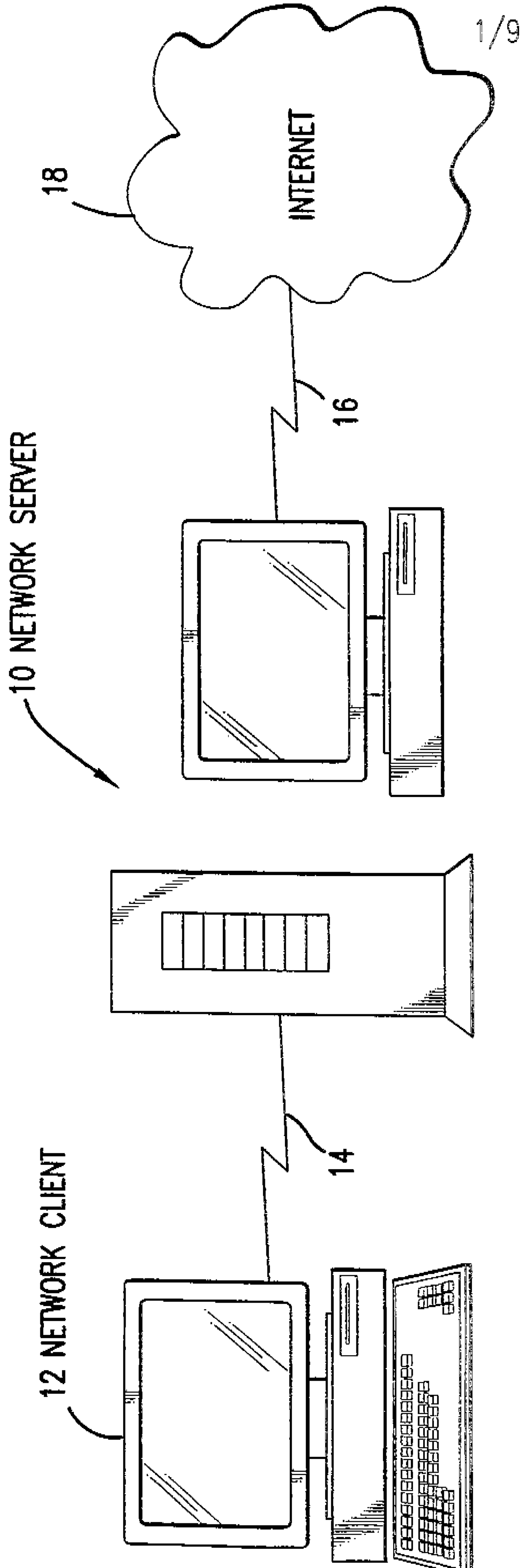


FIG.1

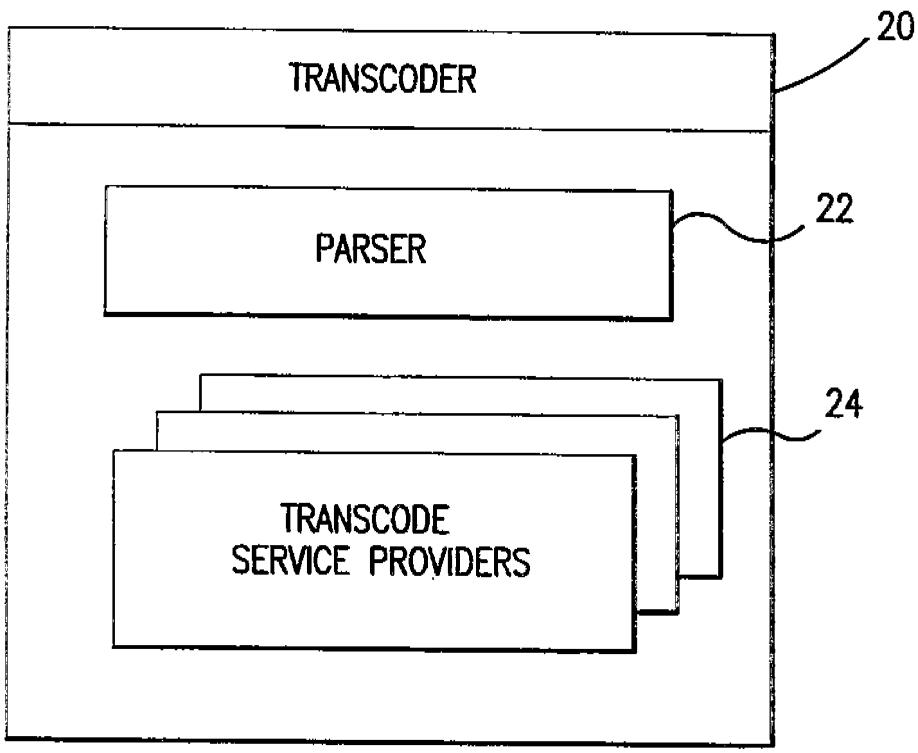


FIG.2

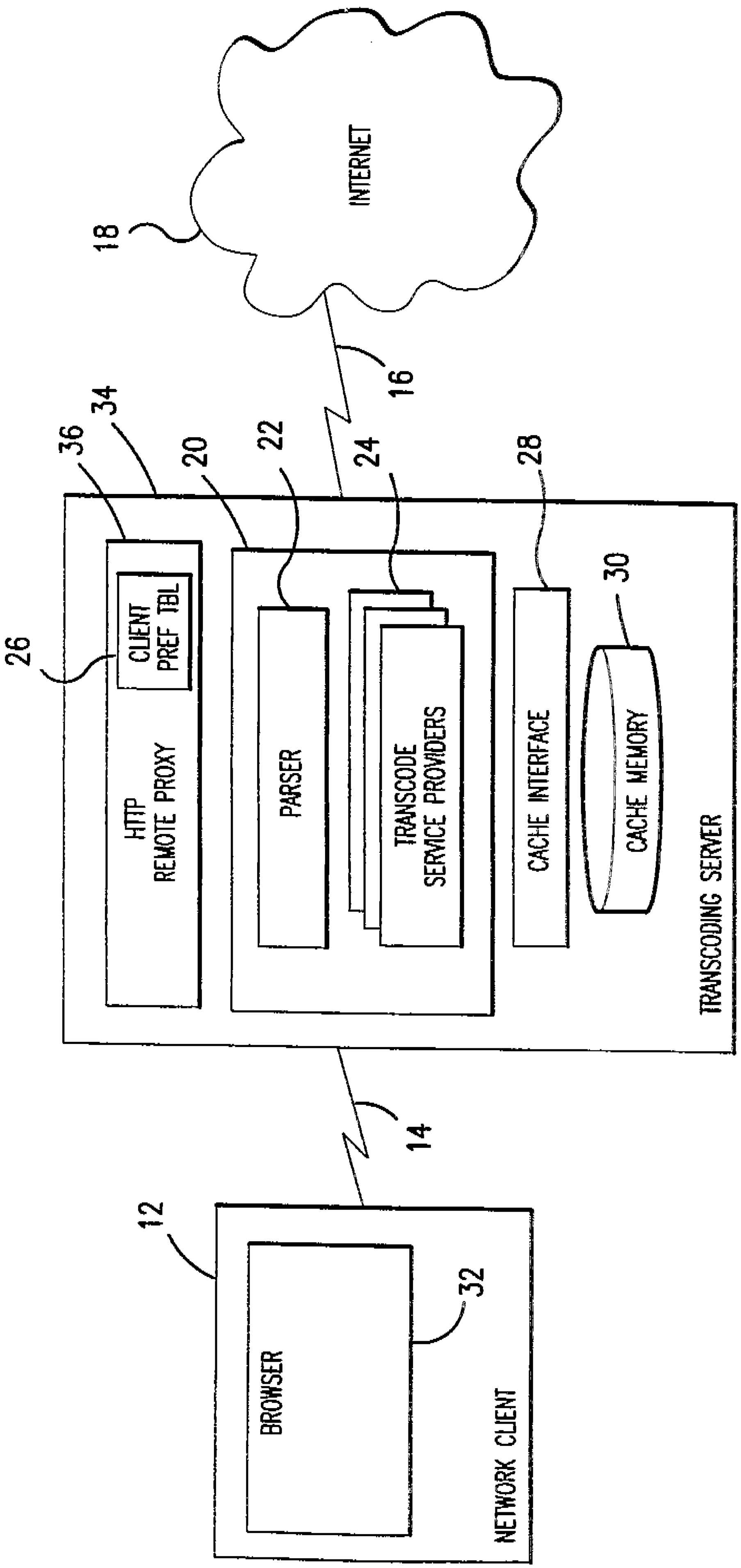

INTERNET
18
16
34
36
20
22
24
28
26
CLIENT PREF TBL
HTTP REMOTE PROXY
PARSER
TRANSCODE SERVICE PROVIDERS
CACHE INTERFACE
30
CACHE MEMORY
TRANSCODING SERVER
14
12
BROWSER
32
NETWORK CLIENT

FIG.3

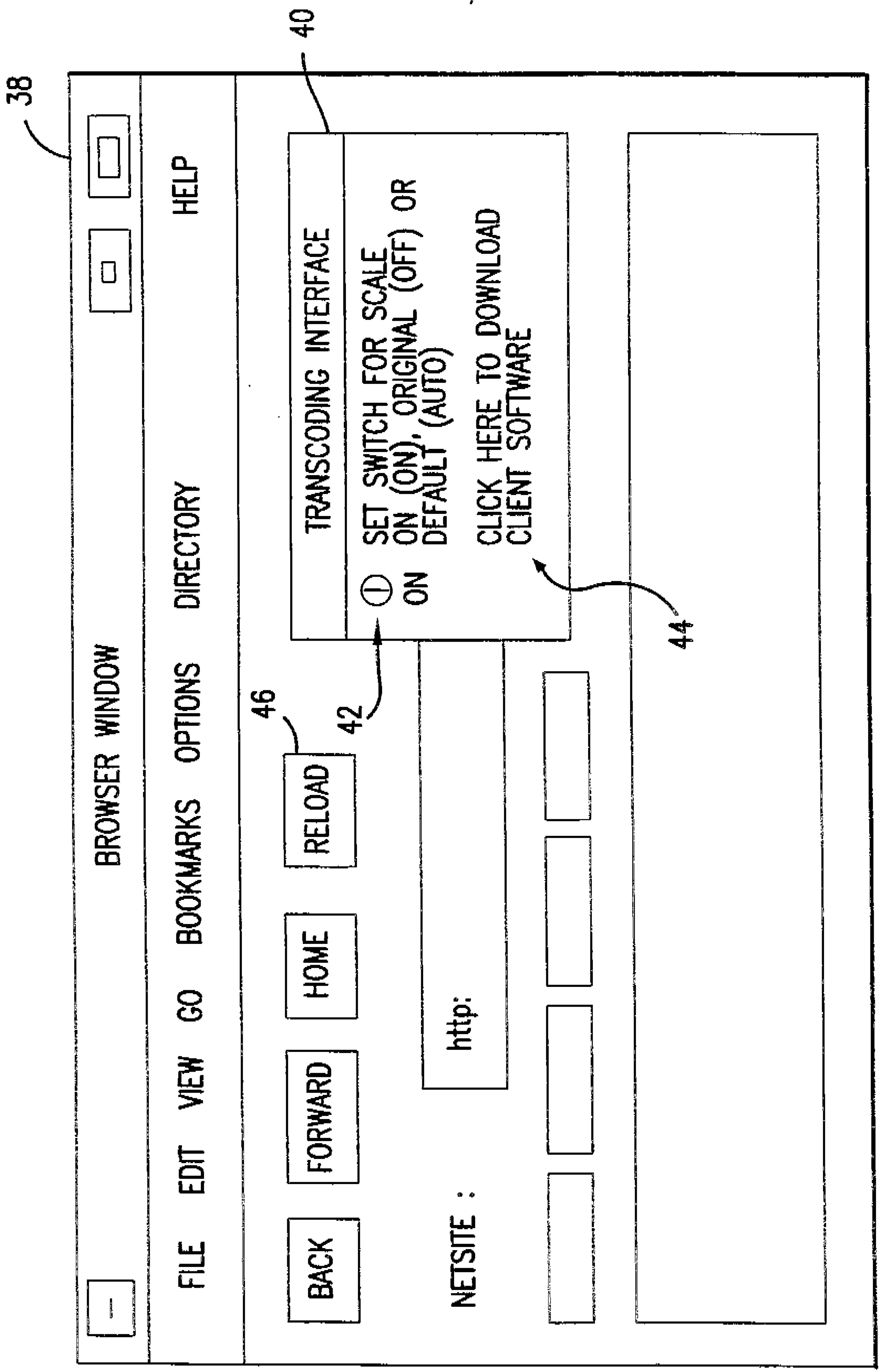

38
40
BROWSER WINDOW
FILE EDIT VIEW GO BOOKMARKS OPTIONS DIRECTORY HELP
BACK
FORWARD
HOME
RELOAD
46
42
TRANSCODING INTERFACE
ON
SET SWITCH FOR SCALE ON (ON), ORIGINAL (OFF) OR DEFAULT (AUTO)
CLICK HERE TO DOWNLOAD CLIENT SOFTWARE
44
NETSITE :
http:

FIG.4

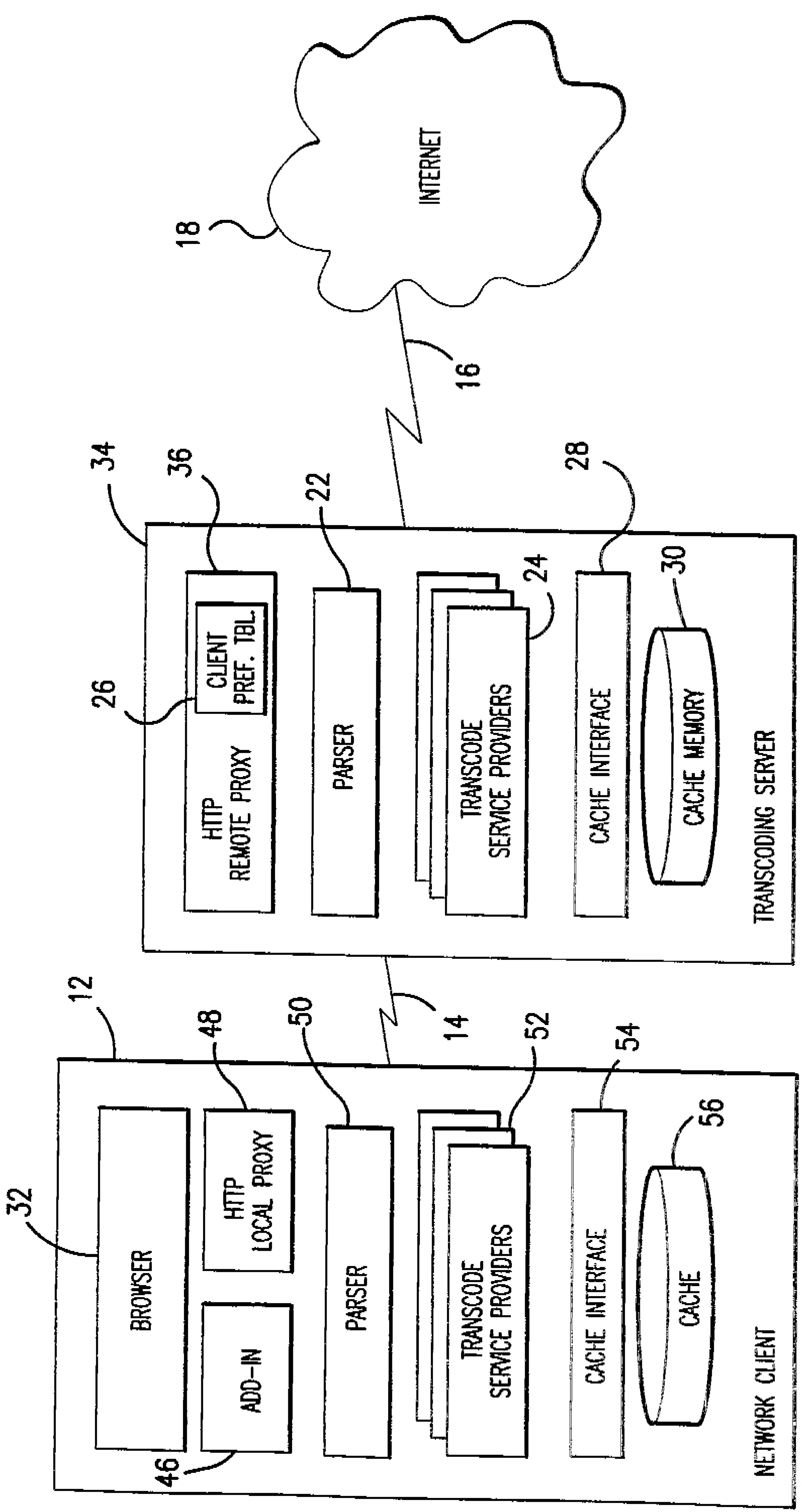

INTERNET
18
16
34
36
26
CLIENT PREF. TBL.
HTTP REMOTE PROXY
22
PARSER
24
TRANSCODE SERVICE PROVIDERS
28
CACHE INTERFACE
30
CACHE MEMORY
TRANSCODING SERVER
12
14
32
BROWSER
46
ADD-IN
48
HTTP LOCAL PROXY
50
PARSER
52
TRANSCODE SERVICE PROVIDERS
54
CACHE INTERFACE
56
CACHE
NETWORK CLIENT

FIG.5

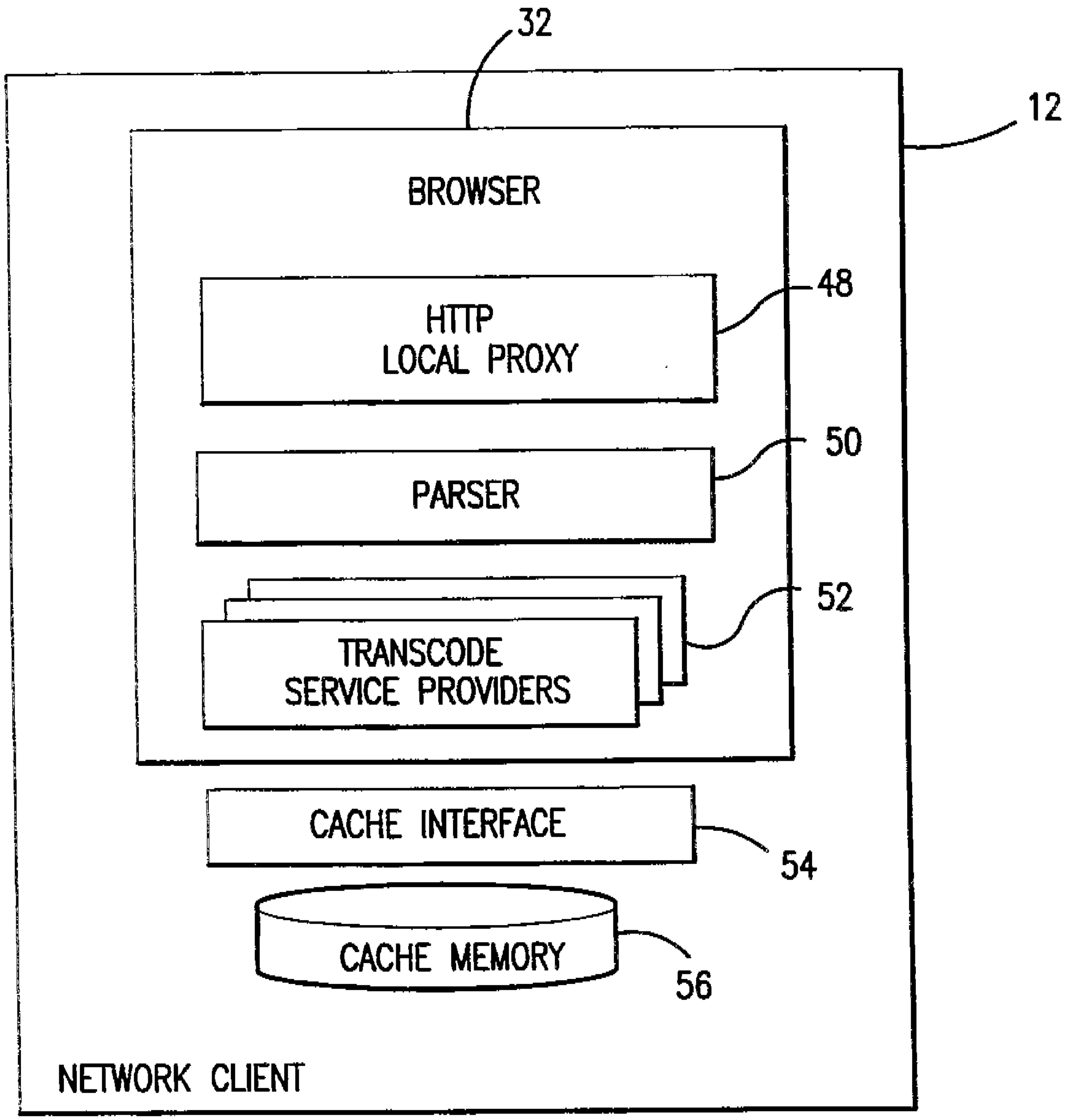

32
12
BROWSER
HTTP
LOCAL PROXY
48
PARSER
50
TRANSCODE
SERVICE PROVIDERS
52
CACHE INTERFACE
54
CACHE MEMORY
56
NETWORK CLIENT

FIG.6

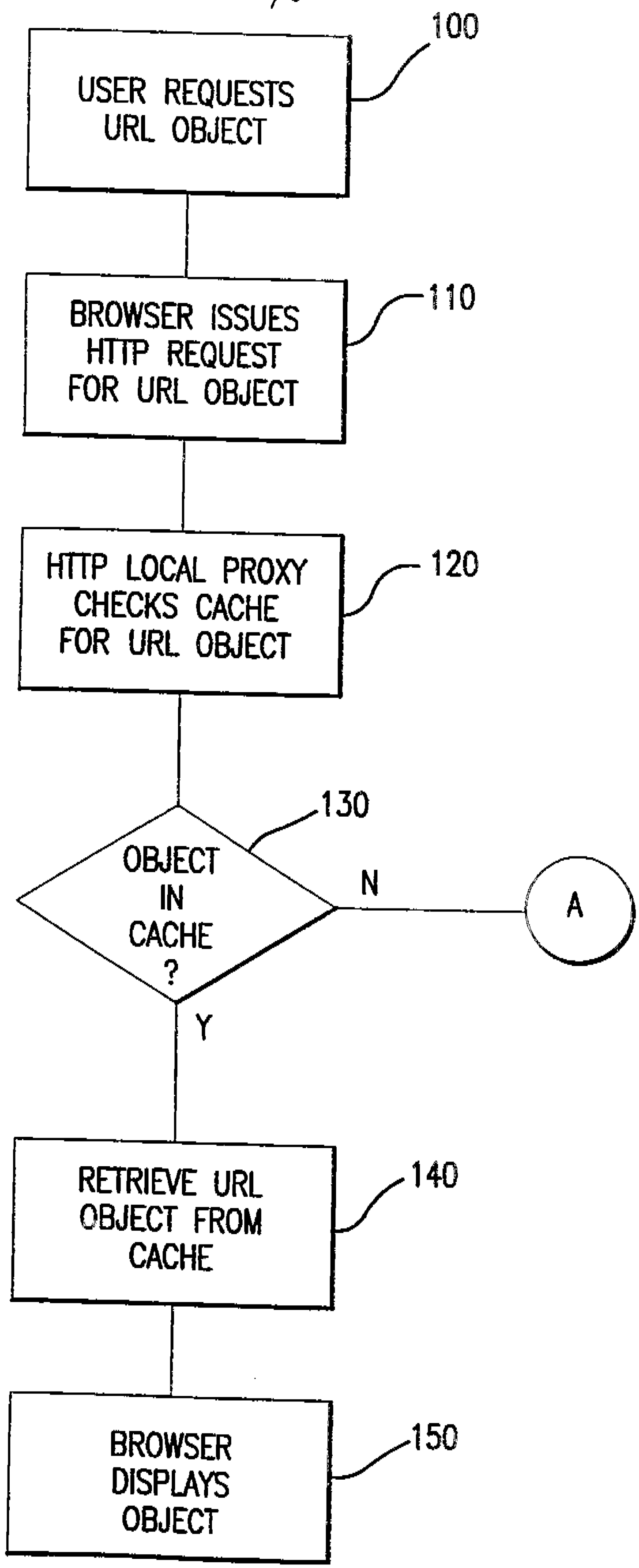

USER REQUESTS URL OBJECT
100
BROWSER ISSUES HTTP REQUEST FOR URL OBJECT
110
HTTP LOCAL PROXY CHECKS CACHE FOR URL OBJECT
120
OBJECT IN CACHE ?
130
N
A
Y
RETRIEVE URL OBJECT FROM CACHE
140
BROWSER DISPLAYS OBJECT
150

FIG.7

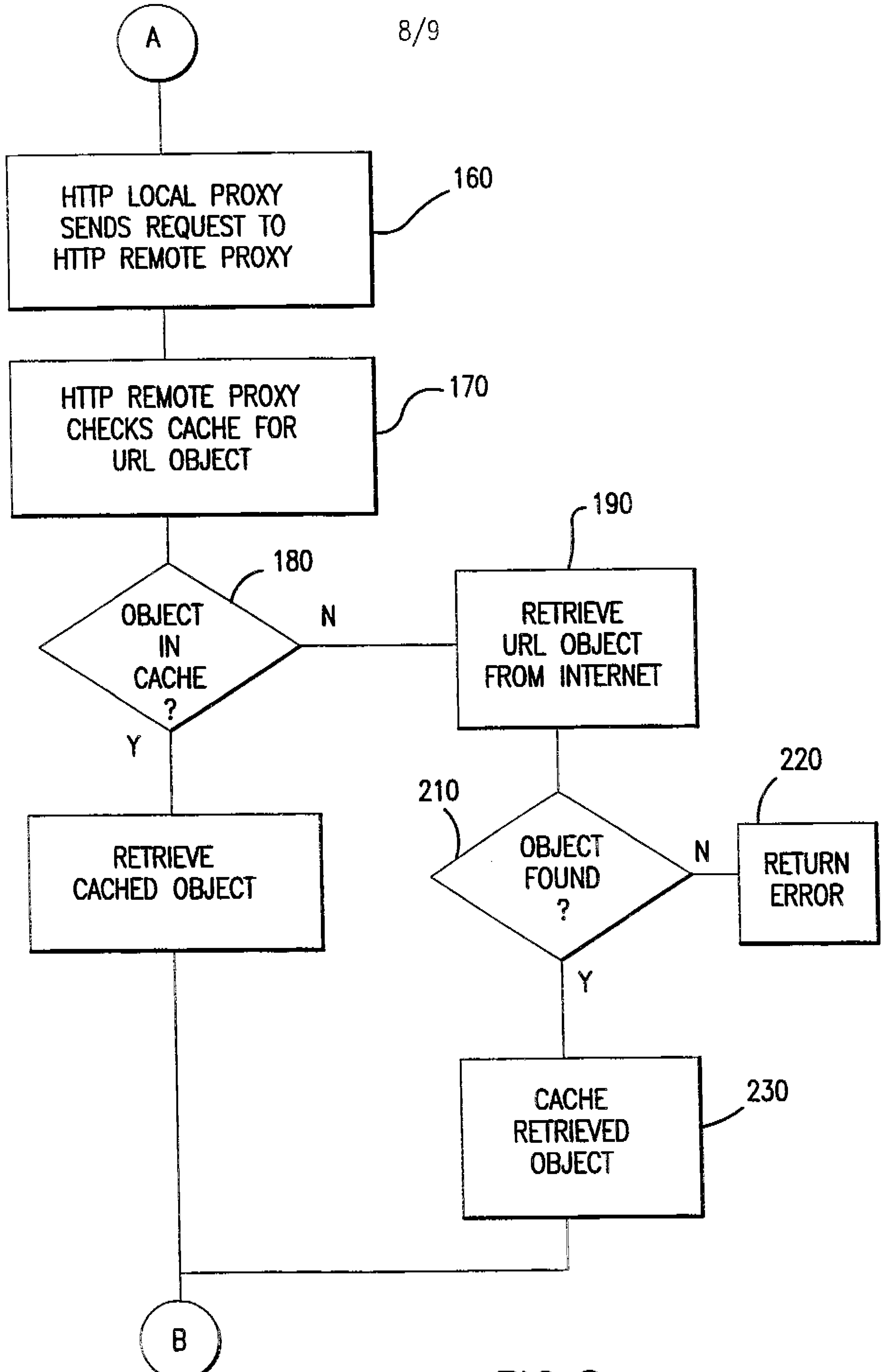

A
HTTP LOCAL PROXY SENDS REQUEST TO HTTP REMOTE PROXY
160
HTTP REMOTE PROXY CHECKS CACHE FOR URL OBJECT
170
OBJECT IN CACHE ?
180
N
Y
RETRIEVE URL OBJECT FROM INTERNET
190
RETRIEVE CACHED OBJECT
OBJECT FOUND ?
210
N
Y
RETURN ERROR
220
CACHE RETRIEVED OBJECT
230
B

FIG.8

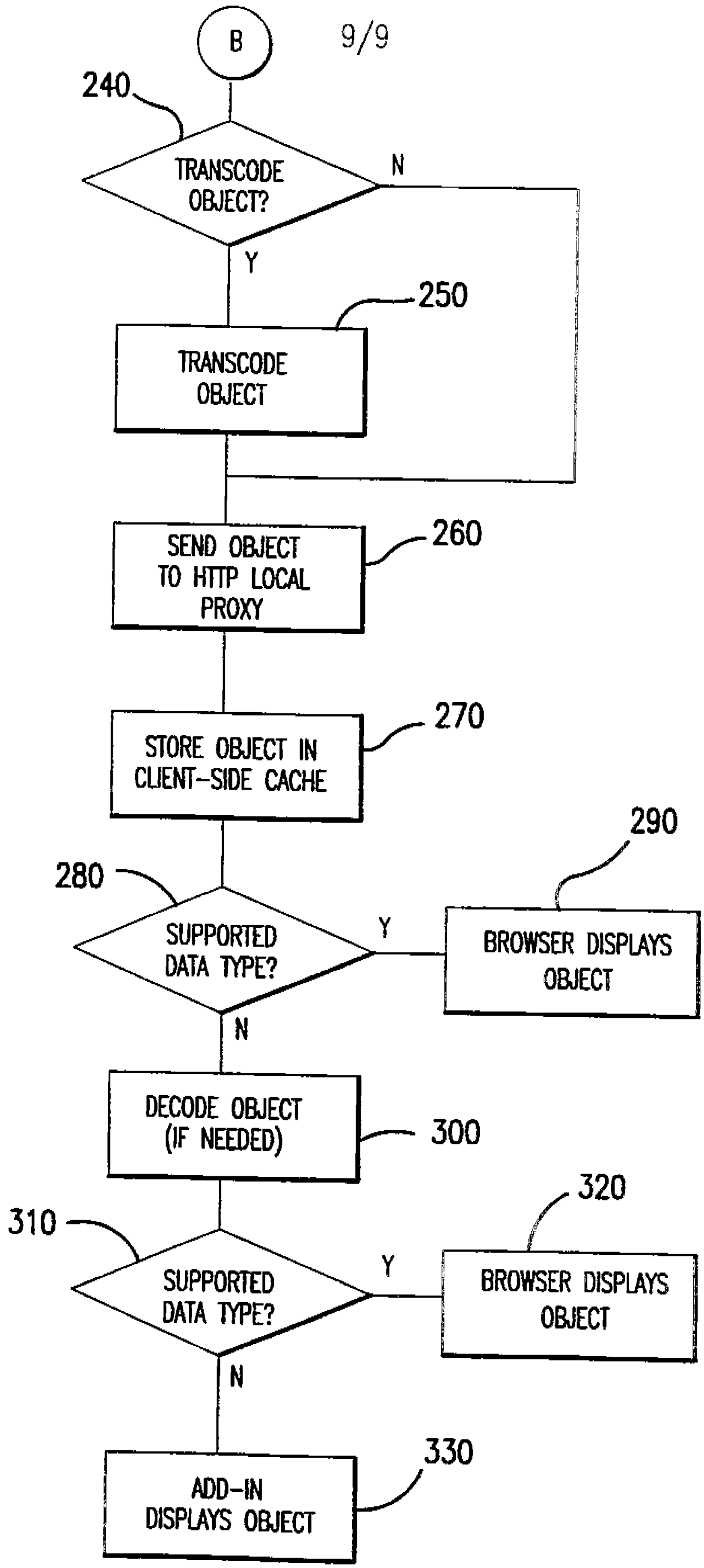

B
240
TRANSCODE OBJECT?
N
Y
250
TRANSCODE OBJECT
260
SEND OBJECT TO HTTP LOCAL PROXY
270
STORE OBJECT IN CLIENT-SIDE CACHE
280
SUPPORTED DATA TYPE?
Y
290
BROWSER DISPLAYS OBJECT
N
DECODE OBJECT (IF NEEDED)
300
310
SUPPORTED DATA TYPE?
Y
320
BROWSER DISPLAYS OBJECT
N
330
ADD-IN DISPLAYS OBJECT

FIG.9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/05304

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 13/38, 15/17
US CL :395/200.76, 200.48, 200.59; 345/335
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/200.76, 200.48, 200.59, 200.33, 200.47, 200.32, 200.58, 200.77; 345/335

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

USPTO APS
search terms: WWW, URL, HTML, HTTP, parser, configure, data object, selection criteria, forward

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,544,320 A (KONRAD) 06 August 1996 Abstract; Fig. 2; C16:14-C17:18	1-25
Y, P	US 5,706,434 A (KREMEN et al.) 06 January 1998 Abstract; Fig. 1, 5, 6; C5:21-59; C10:23-C11:50	1-25
A, P	US 5,724,556 A (SOUDER et al.) 03 March 1998 Abstract; Fig. 6, 9	1-25
X, E	US 5,768,510 A (GISH) 16 June 1998 Abstract; Fig. 6-7, 11-13, 26; C5:24-50; C18:60-C20:18; C22:49-C23:38; C25:66-C26:20	1-25

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date		
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
03 JULY 1998	13 AUG 1998

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	MARK H. RINEHART

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04Q

A2

(11) International Publication Number: WO 97/49252

(43) International Publication Date: 24 December 1997 (24.12.97)

(21) International Application Number: PCT/US97/10758

(22) International Filing Date: 20 June 1997 (20.06.97)

(30) Priority Data:
60/020,094 21 June 1996 (21.06.96) US

(71) Applicant *(for all designated States except US)*: INTEGRATED COMPUTING ENGINES, INC. [US/US]; 460 Totten Pond Road, Waltham, MA 02154 (US).

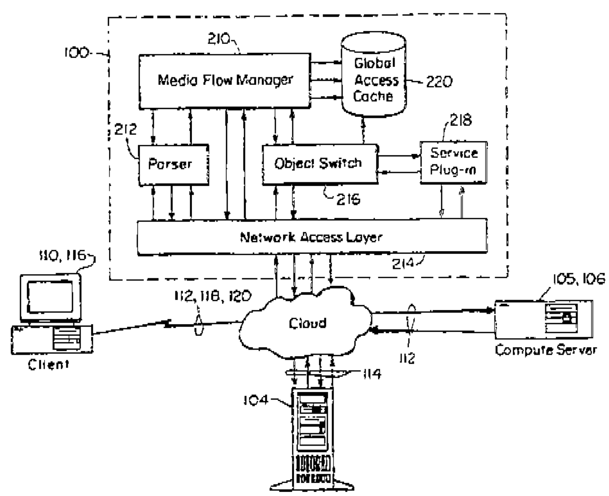
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: SHAH, Ashesh, C. [US/US]; 567 Tremont Avenue, No. 31, Boston, MA 02118 (US). PEDERSEN, Palle [DK/US]; 82 Commonwealth Avenue, No. 10, Boston, MA 02116 (US). RADOVIC, Niksa [HR/US]; 19 Mountain Avenue, Somerville, MA 02143 (US). MANICKAVASAGAM, Senthilkumar [IN/US]; 11 Highland Glen Drive, No. 17, Randolph, MA 02368 (US).

(74) Agents: SMITH, James, M. et al.; Hamilton, Brook, Smith & Reynolds, P.C., Two Militia Drive, Lexington, MA 02173 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: NETWORK BASED PROGRAMMABLE MEDIA MANIPULATOR



(57) Abstract

The media manipulator is a middle layer between the clients, and the remote data servers is the common client-server organization. It transforms the network into a more flexible three-tiered configuration. Requests generated by the clients for media objects from media resources are routed to the media manipulator. It processes the requests and determines if the media objects may be found locally, either cached in the media manipulator itself or in the local data servers. When the media objects are obtained, the media manipulator can be used to perform operations on those objects such as format translations, to apply protective mechanisms for the clients, to speed communications between the remote servers and the clients, or perform compute operations for the clients. In one example, a parser of the manipulator searches for images in the media objects so that service devices can be called to perform data compression or pornography detection on the images. The parser can also search for executable or data files in the media objects and to perform virus scanning or format conversion, respectively.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece		Republic of Macedonia	**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's	**NZ**	New Zealand		
CM	Cameroon		Republic of Korea	**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

NETWORK BASED PROGRAMMABLE MEDIA MANIPULATOR

RELATED APPLICATIONS

This application claims priority to U.S. Provisional Application No. 60/020,094, filed June 21, 1996, the contents of which is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

In a client-server network, on one hand there are clients, typically personal computers, IBM-compatible computers and/or UNIX workstations, for example, equipped with information browsers. On the other hand, there are data servers and compute servers. Data servers are computers with a large storage capacity containing information in different media formats: data records, plain text documents, word processing documents, still pictures, compressed audio and video, and executable files, for example. Compute servers are computers that carry out intensive computational tasks that would typically require too much time for the client to complete. Each compute server might use a single or many processors to complete the given task.

Users interact with their clients in a natural way with a mouse, keyboard, screen, printer, or by some other input/output device. The users need not be concerned about what happens after they make their selection within their clients. Clients then make service requests to geographically dispersed servers. Upon receiving requests from the clients, the servers perform the desired operations and return the retrieved or computed media stream back to the client for display.

SUMMARY OF THE INVENTION

The present invention is connected into the ubiquitous two-tiered client-server network of computers. It is designed as a middle layer, middleware, between the clients and the remote data servers. It transforms the network into a more flexible three-tiered configuration. Requests generated by the clients for media objects from media resources are routed to the media manipulator. It processes the requests and determines if the media objects may be found locally, either cached in the media manipulator itself or in local/remote data servers. When the media objects are obtained, the media manipulator can be used to perform operations on those objects such as format translations, to apply protective mechanisms for the clients such as virus scanning, to speed communications between the remote servers and the clients using compression operations, or perform compute operations for the clients.

In general, according to one aspect, the invention features a middle-ware computing system. It includes a network access system that supports communications with media resources and client computers and a media manipulation system that operates on media objects received from the media resources via the network access system prior to forwarding the media objects to the client computers.

In specific embodiments, a parser is used to identify different media types within the media objects so that service devices may be called to operate on the media types. In one example, the parser searches for images in the media objects and service devices include an image compressor for performing data compression or pornography detection on the images. The parser can also search for executable or data files in the media objects and the service devices then called to perform virus scanning or format conversion, respectively.

In further specifics, a cache is used to store media objects. A media flow manager receives requests for media objects and checks for the presence of the media objects in the cache to preclude the necessity of obtaining the objects from the remote media resources.

The above and other features of the invention including various novel details of construction and combinations of parts, and other advantages, will now be more particularly described with reference to the accompanying drawings and pointed out in the claims. It will be understood that the particular method and device embodying the invention are shown by way of illustration and not as a limitation of the invention. The principles and features of this invention may be employed in various and numerous embodiments without departing from the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings, reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale; emphasis has instead been placed upon illustrating the principles of the invention. Of the drawings:

Fig. 1 is a schematic block diagram illustrating the context in which the inventive media manipulator operates;

Fig. 2 is a block diagram illustrating the interaction between components of the media manipulator according to the invention;

Fig. 3 is an object interaction diagram illustrating the operation of the components of the media manipulator;

Figs. 4A, 4B, and 4C show the message formats for transmitting tasks to compute servers;

Fig. 5 is a block diagram showing the programming of the media manipulator using m-script;

Fig. 6 is another object interaction diagram showing the order of creation of the components of the manipulator; and

Fig. 7 is a block diagram showing another embodiment of the media manipulator.

DETAILED DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates the context in which the media manipulator 100 operates. In many applications, it is important for users to access remote media resources 108 such as the data servers 104 of content providers on the Internet. These users may be at client computers 110 that are inter-connected by a local area network (LAN) 112. The clients 110 access the media resources 108 through a gateway 114 linking the LAN 112 to the Internet. The user's also require access to the media resources 108 remotely at remote clients 116 through, for example, telephone dial-up connections 118 or through cellular/wireless links 120.

The media manipulator 100 is connected into this two-tiered client-server network of computers as a middle layer between the clients 110, 116 and the remote data servers 104 of the media resources 108.

Fig. 2 is a block diagram illustrating the internal organization of the media manipulator 100. It comprises six basic components: media flow manager 210, media parser 212, network access layer 214, object switch 216, multiple service plugins 218, and global access cache 220. In one embodiment, these components are implemented as separate software objects that run on a common microprocessor or multiprocessor system.

The media flow manager 210 serves as the principle controller for the media manipulator 100. It has access to the various components and can alter their behavior. It specifies the operations to be formed on the received media objects. It is also the storehouse for the information on the media objects as they are received from the media parser. The media flow manager 210 also tracks the physical resources that are functional and available in the media manipulator and on the surrounding LAN in order to determine to which of the resources the media objects flow.

The network access layer 214 makes the media manipulator 100 accessible through many different types of network devices and the protocols running on top of them. In one implementation, the network access layer communicates through the Internet gateway 114 using the TCP/IP protocol, connects to the compute or data server using the protocol of the local area network 112, and communicates with the clients using either the LAN or the protocols necessary to communicate with the remote clients 116 over low-bandwidth connections 118, 120.

When communicating with the remote data servers 104 of the content providers, the network accepts the incoming data streams and assembles them into media objects. These media objects are then made available to the media parser 212, object switch 216, and service plugins 218. The media parser 212 analyses all incoming media objects to extract the relevant media types. These media types include executable files, data files, and images, for example. Information concerning the detected media types is forwarded to the media flow manager 210, which decides what operations should be performed on the media.

The object switch 216 supports a number of incoming and outgoing object gates. Media objects enter into the object switch from the network access layer 214 and from the service plugins' output links. The media objects leaving the object switch 216 go into the network access layer 214 and the service plugins' input links. The object switch routes the objects based on the media manager instructions, either directly or indirectly.

The global access cache is an intelligent mechanism that speeds the operation from the perspective of the user at the clients 110, 116. It determines which media objects are most likely to be used in the future and stores them in the fastest available memory. Media objects that are somewhat less likely to be required again are stored in slower memory or a secondary cache. There can be as many levels of the cache as the physical infrastructure allows, and the caching may take place on data servers that are remote from the main computational resources of the media manipulator. This caching minimizes the time that different users need to wait for requests to be processed.

The media manipulator 100 is a programmable device. A system administrator can change its behavior by giving it m-script commands. It is also an extendable device. By adding new service plugins, new capabilities can be added to the device. The construction of the components of the media manipulator allows for redundancy and fault tolerance. A hardware failure does not bring the entire system to a halt. The system will keep working and simply notify the administrator that one of its components needs to be replaced.

Fig. 3 is an object flow diagram illustrating the communication between the client 110, 116, content provider's data server 104, and the components of the multimedia manipulator 100.

The first step is the initial connection 1, Connect, between the client 110, 116 and the media manipulator 100 via the network access layer 214. The network access layer 214 accepts this request 2. In one implementation, it accepts by calling a new incidence of itself such that each incidence of the network access layer object supports a single connection outside the media manipulator 100.

After establishing the connection, the client makes a request 4 for a media object. In the typical example, this will be a universal resource locator (URL) to a data server 104 of a content provider on the Internet. The network access layer then calls the media parser 212 and passes the client request 6.

The media parser 212 looks to two sources for the media object simultaneously. The ProcessURL request 8 is passed to the media flow manager 212, which has knowledge of the contents of the global access cache 220. The parser also issues a request 10, GetPage, to the network access layer.

The media flow manager 212 searches for the object in the cache 220. If the cache returns a cache-miss, the request to the provider has not been delayed waiting for the miss status, whereas in the case of a cache-hit, the request to the provider is simply terminated after verifying the validity of the cached page. Using this scheme, there is little increased latency associated with the use of the manipulator 100 in the worst-case cache-miss scenario.

In the illustrated example, a cache-miss occurred. Thus, rather than supplying the object, the cache 220 is prepared 12 to receive the media object, PutInCache. Also, the network access layer 214 connects 14 to the content provider and retrieves 16 the media object or page.

As the media object is being received by the network access layer from the content provider 104, the parser begins to parse 17 the object. As parsing proceeds, the parser also begins to update 18 the global access cache 220 with the parsed portions of the object. Simultaneously, the parser begins the reply 20, 22 to the client via the network access layer.

In one implementation, the parser searches for images in the media objects to perform compression or pornography detection, for example. On encountering an image, the parser 214 passes a call to the media flow manager to process the image 24 while continuing to parse 26 the media object.

The media flow manager 210 gets the image 28 via the network access layer 214. The fact images are not stored with the page but must be separately requested is an artifact of the HTTP protocol. The network access layer 214 then connects 30 to the content provider 104 and retrieves 32 the image.

When the image is retrieved, the media flow manager 210 places it in the cache with the other portions of the media object and makes a function call to the object switch to process the image 34. The object switch knows the various service plugins that are available and the actions that must be performed on the media types that are discovered by the media parser, which in this example is an image. When called by the object switch 216 to process 36 the media type, the particular service plugin, or multiple plugins when serial operations are required, retrieves 38 the media type, *i.e.*, image, and performs the desired operation on or processes 40 the image. For example, in one instance, this can be compression or thinning to expedite communication to the client. In another case, it can detect the probability of pornography by detecting the percentage of flesh-tone colors in the picture. Once the processing is complete, the new image or revised media object may be placed 42 in the cache 220 or used in a reply to the client 110, 116.

In many instances, the service plugin functionality will be performed by a separate compute server 105. This computer may be directly accessible by the media manipulator 100 or accessible through the local area network 112. Generally out-sourcing this functionality is desirable, rather than running on the same device with the other components of the media manipulator 100, to avoid depriving those other components of processing bandwidth.

When the plugin does utilize the external compute server, it issues a request message. Fig. 4A illustrates the formatting of the message to the compute server. The message has a number of different fields. It has a version field and a length field defining the length of the content. The type field indicates the type of the message, and the message ID is assigned by the network access layer. The source type indicates the media type. In the context of image files, the type indicates whether the image is in a GIF or JPEG type compression format, for example. The source path is the path to where the image is stored in the global access cache 220, to which the compute server has access. The destination type, path length, path, and parameters define the transformed media type and where it is to be sent.

Fig. 4B illustrates the reply message from the compute server. It again has version, length, type, and message ID fields. The reply code indicates whether or not the service was successful. The destination type, path length, and path indicate the type of the final image after the transform of the compute server has been implemented and where that final image is stored in the global access cache or otherwise.

Fig. 4C shows the error message issued by the compute server when service was unsuccessful or error occurred. To contain this information, the message has a computer server error code identifying the server and a field holding the reason for the error.

As illustrated in Fig. 5, the administrator or Internet application developer specifies the actions of the media manipulator by supplying an m-script language to the media flow manager 210. This is a quasi-configuration, script file which forms a high level programming language of the media manipulator. The following illustrates the general structure of the language with examples showing its use in the media manipulator 100.

name := definition

The name of a rule is simply the name itself (without any enclosing "<" and ">") and is separated from its definition by the colon-equal (":=") character.

"literal"

Quotation marks surround literal text. Unless stated otherwise, the text is case-sensitive.

rule1 | rule2

Elements separated by a bar ("|") are alternatives, e.g.,
"yes | no" will accept "yes" or "no".

{rule1 rule2}

Elements enclosed in parentheses are treated as a single element. Thus, "{elem {foo | bar} elem}" allows the token sequences "elem foo elem" and "elem bar elem".

rule*

The character "*" following an element indicates repetition. For example, "foo bar*", implies, "foo" followed by zero or more of "bars".

[rule]

Square brackets enclose optional elements. For example, "foo [bar]" implies, "foo" followed by zero or one of "bar".

The BNF grammar of the m-script is grouped under three logical groups.

Basic

<table>
<tr><td>Alphabets</td><td>:=</td><td>{abc…zABC..Z}</td></tr>
<tr><td>Variable-chars</td><td>:=</td><td>{abc…zABC…Z-_}</td></tr>
<tr><td>Numbers</td><td>:=</td><td>{0…9}</td></tr>
<tr><td>Variable-name</td><td>:=</td><td>Variable-chars {Variable-chars | Numbers}*</td></tr>
<tr><td>Host-name</td><td>:=</td><td>Variable-chars {Variable-chars | Numbers | "."}*</td></tr>
<tr><td>Path-name</td><td>:=</td><td>{Variable-chars | Numbers | "." | "/" | "\" | "~" | ":"}*</td></tr>
<tr><td>Others</td><td>:=</td><td>{!@#$%^&*()"{}-=|\<>,}</td></tr>
<tr><td>EOLN</td><td>:=</td><td>"\n"</td></tr>
<tr><td>Comment</td><td>:=</td><td>"#" {Variable-chars | Others | "." | "/" | "~" | ":" }* EOLN</td></tr>
</table>

This section describes the basic rules used: Alphabets are composed of letters "a" through "z", "A" through "Z"; Numbers are composed of digits zero through nine. Variables-chars are alphabets, dash ("-") and underscore ("_"). A variable name must start with a Variable-char and followed by zero or many variable-chars or numbers. Host-name is similar to variable-name and in-addition can have periods ("."). Path-name is a generic path used for locating files. EOLN is ASCII 13. A comment must start with "#" character and ends with an EOLN.

Generic

m-script	:=	{ comment | section }*
Section	:=	section-key "{" section-Desc "}"
Section-Desc	:=	section-line*
Section-line	:=	section-desc-key "=" section-desc-value [EOLN]

Section	:=	server-section | cache-section | service-section | filter-section | action-section

This section describes a generic m-script file. An m-script is a comment or a section. A section must start with a Section-key, followed by a Section-description enclosed in parentheses. The section description is made up of zero or many section lines. A section line starts with a section description key followed by an equal sign ("=") and the section description key's value. There are five types of sections, *viz.*, server, cache, service, filter and action.

Detail

Server-section	:=	"server" "{" server-sec-desc "}"
Server-sec-desc	:=	server-name-line | server-port-line
Server-name-line	:=	"name" "=" host-name [EOLN]
Server-port-line	:=	"port" "=" numbers [EOLN]

A server section starts with the key "server". This section consists of two lines: Name and port lines. The name line specifies the name of the host on which the MM 100 is run. The port line specifies the main port number on which the MM awaits requests from clients.

Cache-section	:=	"cache" "{" cache-sec-desc "}"

Cache-sec-desc := cache-clean-line | cache-direc-line

Cache-clean-line := "cleanup" "=" { number | "no" }

Cache-direc-line := "directory" "=" Path-name

A cache section starts with the key "Cache". This section consists of two lines as well: Cache-clean and directory lines. The cache-clean line specifies the time interval after which the cache cleaning is performed. It takes two values: a positive number (time interval in seconds) or the string "no"(implying never to be cleaned). The directory line specifies the directory in which the cached files need to be stored.

Service-section := "service" "{" Service-sec-desc "}"

Service-sec-desc := Service -id-line | Service-host-line | Service-port-line

Service-id-line := "id" "=" variable-name

Service-host-line := "host" "=" host-name

Service-port-line := "port" "=" numbers

A service section is for service plugins. There must be a service section for each service that has to be used by the MM 100. This section starts with the key "service". The section consists of three lines: Id, Host and port lines. The id line specifies a user defined identifier that can be used in other sections. The host and port lines respectively specify the name of the host and port number on which the service is available.

Filter-sec := "filter" "{" filter-desc "}"

Filter-desc := filter-object-line | filter-action-line

Filter-Object-line := "object" "=" Filter-Object-Name

Filter-Object-Name := “image” | “video” | “java”

Filter-Action-line := “action” “=” variable-name

A filter section starts with the key "filter". This section consists of two lines: object and action line. The object line specifies the name of the object to be identified and filtered. The action line identifies the rule to be applied on the object. Currently, the objects identified are images. In future, objects like video and Java applets can be identified.

Action-section := “action” “{“ action-desc “}”

Action-desc := action-id-line | action-cond-line | action-proc-line

Action-id-line := “id” “=” variable-name

Action-cond-line := “cond” “=” Action-cond-exp

Action-cond-exp := [Action-exp-bin-op] Action-exp-var [Action-cond-exp-op Action-cond-exp]

Action-exp-bin-op := “!”

Action-cond-exp-op := “&&” | “||” | “==” | “!=” | “>” | “<” | “>=” | “<=”

Action-exp-var := { Filter-Object-Name “.” Parameter } | { variable-name “.” “result” }

Parameter := “any” | “transparent” | “animated”

Action-proc-line := “process” “=” Action-proc-exp

Action-proc-exp := { variable-name| Method-exp } [Action-connect Action-proc-exp]

Method-exp := Filter-Object-Name “.” Method-name “(“ Method-Param* “)”

Method-name := “replace”

Method-Param := “"” Path-name “"”

Action-connect := “&” | “|”

The action section is the most complicated section. The action sections can be linked to other action sections forming a list of actions to be applied in tandem. The section starts with the key "action". This section consists of three lines: id, condition and process lines. The id, as before, is a user assigned identifier. The condition line specifies a condition when the process has to be performed. The condition is like a standard "C" expression. It uses object's properties (e.g., image.transparent - image that has a transparent bit), or result of other rules (e.g. rule1.result). The process can be a service identifier or another rule identifier. Several identifiers can be connected using action connectors: "&"(and) or "|"(or). The "&"(and) connector implies both the rules have to be applied in succession (e.g.: rule1 & rule2 - implies apply rule1 and then rule2). The "|" (or) connector implies that apply either of the process (e.g.: compress1 | compress2 - implies, apply compress1 or compress2).

An example is as shown below:

```
#m-script for manipulating HTML files

#listening host name and port
server {
name = center
port = 8001
}

#cache parameters
cache{
cleanup = no
directory = "/opt/mm/cache/images/"
```

```
13   }
14
15   #compress service server 1
16   service{
17   id = compress1
18   host = center
19   port = 7002
20   }
21
22   #compress service server 2
23   service{
24   id = compress2
25   host = center
26   port = 7003
27   }
28
29   filter{
30   object = image
31   action = rule1
32   }
33
34   action{
35   id = rule1
36   cond = image.any && ! image.transparent
37   process = compress1 | compress2
38   }
```

Line #	Explanation
1.	A comment line starts with a "#" character. Everything to the end of that line is ignored.
5 & 6	The media manipulator listens on the host "center" and on port "8001"
11 & 12	The files are cached (global cache) on the server. Keep them longer. Store them in the directory specified.
17-19	Compute server id is "Compress1". The host address is "center" and is listening on port "7002"
24-26	Compute server id is "Compress2". The host address is "center" and is listening on port "7003"
30 & 31	Filter the images and apply rule1

```
This section is rule1
Do the process for any image that is not transparent.
Process  images by sending to compress1 or to compress2
```

Example #2

Apart from compressing the images, the images can be tested for pornography. For this a service section has to be added and the action section has to be modified. The following m-script accomplishes this.

```
#m-script for manipulating HTML files
#This compresses the images and detects them for pornography

#listening host name and port
server {
name = center
port = 8001
}

#cache parameters
cache{
cleanup = no
directory = "/opt/mm/cache/images/"
}

#compress service server 1
service{
id = compress1
host = center
port = 7002
}

#compress service server 2
service{
id = compress2
host = center
port = 7003
}

#pornography detect service server 1
service{
```

```
id = porno1
host = center
port = 7010
}

#pornography detect service server 2
service{
id = porno2
host = center
port = 7011
}

filter{
object = image
action = all_image_rule
}

action{
id = all_image_rule
cond = image.any
process = compress_rule & porno_rule & destroy_rule
}

action{
id = compress_rule
cond = ! image.transparent
process = compress1 | compress2
}

action{
id = porno_rule
cond = compress_rule.result == 1
process = porno1 | porno2
}

action{
id = destroy_rule
cond = porno_rule.result >= 75
process = image.replace("/opt/mm/lib/images/forbidden.gif")
}
```

Line #	Explanation

5 - 8 Server section
11 - 14 Cache section
16 - 28 Compute servers "Compress1" and "Compress2"
31 - 35 Pornography detection service "porno1" is running in "center" and listening on port 7010
38 - 42 Pornography detection service "porno1" is running in "center" and listening on port 7010
44 - 47 Filter the images and apply all_image_rule
49 - 52 Apply action "all_image_rule" to all images. First apply compress_rule, followed by porno_rule and then by destroy_rule.
55 - 59 Apply action "compress_rule" to non-transparent images. Pass the images to either compress1 or compress2.
61 - 65 Apply action "porno_rule" to images, if compress_rule returned 1. Pass the compressed images to porno1 or porno2.
67 - 71 Apply action "destroy_rule" to images, if porno_rule returned a value greater than or equal to 75(probability of a pornographic image). Replace the image with "forbidden.gif".

Media Flow Manager (MFM)

MFM reads m-script and configures itself and other components based on the m-script. The MFM can be implemented as a multi-threaded object as shown below:

```
class MFM{
private:
        int iPort;
        char *strHostName;
        char *strMFileName;
        MediaParser *pMP;
        GAC *pGAC;
        ObjSw *pOS;
        NAC *pNAC;
        ...
```

```
public:
        MFM();
        ~MFM();
        int Configure(char *strMFileName);
        int ProcessURL(char *strURL, ...);
        int ProcessImage(char *strSrcURL, int iHeight, int iWidth, ...);
        int CheckCacheUpdate(...);
        int CreateInstance();
        ...
};
```

Configure(...)	Parses the m-Script file specified and configures the rest of the components.
ProcessURL(...)	This is called by the parser, when it encounters a new image. This initiates the cache insertion on GAC.
ProcessImage(...)	Mainly invoked by the MediaParser, when it encounters an image tag. This passes the command to the appropriate object switch.
CreateInstance(...)	This creates a new instance of the MFM by first copying the internal data structures and then creating a new thread.

Media Parser - HTML Parser

The media parser can be implemented using generic tools like lex and yacc. The core of the parser can then be packaged to make parser objects.

```
class MediaParser {
private:
        MFM *pMFM;
        GAC *pGAC;
        ...
public:
```



```
        MediaParser(MFM *pMFM, GAC *pGAC, ...);
        ~MediaParser();
        int AddFilter(int iObjectType, ...);
        int Parse(...);
        ...
};
```

AddFilter(...)	Called by the MFM.Configure, adds to the list of objects that the MediaParser has to look for.
Parse(...)	This is called by the NAL, when it successfully establishes a connection with the client. This parses the media. When it encounters the object to be filtered, the parser notifies the MFM by invoking the appropriate function.

Global Access Cache

The global access cache is a specialized cache system, specifically tuned to keep HTML pages and the images. The images can have multiple versions. These have to be cached separately. The cache is also cleaned regularly as described in the cache section of the m-script.

```
class GAC{
private:
        char *apMainBuckets[MAX_HASH_KEY];
        int Hash(char *strURL);
        ...
public:
        GAC(MFM *pMFM, char *strPath, ...);
        ~GAC();
        int SearchCache(char *strURL, ...);
```

```
        int PutInCache(char *strURL, char *strLocalFilename, FILE *fp, ...);
        int UpdateCache(char *strURL, ...);
        int GetFromCache(char *strURL, int iKey, ...);
        ...
};
```

Hash(...)	This is used to create the Hash key based on an URL.
SearchCache(...)	This searches the cache for the given URL.
PutInCache(...)	First searches the cache(SearchCache()) and if not found, inserts the URL int the cache.
UpdateCache(...)	Updates the cache entry with related entries. For example, the URL entry can be updated with image entries that are related to the URL.
GetFromCache(...)	Retrieves an URL or an Image.

Network Access Layer

The network access layer for handling HTML pages, primarily deals with HTTP(Hyper Text Transmission Protocol). It accepts connection from the clients; makes connection to the content provider; requests and receives pages and images from the content provider. In addition to these the layer also provides connection to compute servers.

```
class NAL{
private:
        int iPort;
        char *strHostName;
        int iNumCharsRead;
        int iNumCharsWritten;
        char *strURL;
        ...
public:
        NAL(MFM *pMFM, MediaParser *pMP, ...);
```

```
        ~NAL();
        int Listen(char *strHostName, int iPort, ...);
        int Accept(...);
        int Connect(char *strHostName, int iPort, ...);
        int AcceptClients(char *strHostName, int iPort, ...);
        int GetImage(char *strHostName, int iPort, char *strURL, ...);
        int GetURL(char *strHostName, int iPort, char *strURL, ...);
        int SendRequest(...);
        int ReceiveReply(...);
        ...
};
```

Listen(...)	Creates a listening port.
Accept(...)	Accepts any client requesting a connect.
Connect(...)	Connects to the specified host and port number. Usually called by the GetImage() or GetURL()
GetImage(...)	Connects to the ContentProvider and requests the image specified by the URL. This is responsible for building the appropriate request header etc
GetURL(...)	Connects to the ContentProvider and requests the page specified by the URL. This is responsible for building the appropriate request header etc.
SendRequest(...)	Sends a formatted message to the compute server. The format of the message is shown in the following section.
ReceiveReply(...)	Receives a formatted message that is a reply to the message sent earlier.

Service Plugin

The service sections describe the various servers available for the MM. Each service server is an instance of this object.

```
class ServPlugin{
private:
        char *strId;
        int  iPort;
        char *strHostName;
        ...
public:
        ServPlugin(NAL *pNAL, char *strId, char *strHostName, int iPort, ...);
        ~ServPlugin();
        int Request(char *strSrcPath, char *strDestPath, ...);
        ...
};
```

Request(...)	This initiates the request through the NAL. NAL sends the formatted message to the appropriate Compute Server.

Object Switch

The object switch interfaces the MFM and the service plugins. The object switch mostly implements the rules specified in the action section of the m-script, as instructed by the MFM.

```
class ObjSw{
private:
        MFM *pMFM;
        GAC *pGAC;
        ServPlugin *aSP; //array of service plugins
        ActionList *alAction; //linked list of actions
        _
public:
```

```
        ObjSw(MFM *pMFM, GAC *pGAC, _);
        ~ObjSw();
        int AddServicePlugin(ServPlugin *pSP, _);
        int AddAction(char *strId, char *strCond, char *strProcess, _);
        int ProcessImage(_);
        _
};
```

AddServicePlugin(_)	This is invoked by the MFM during configuration phase. This adds the service plugin to its internal list.
AddAction(_)	This is also invoked by the MFM during the configuration phase. This adds the actions specified in the m-Script
ProcessImage(_)	Invoked by the MFM, this executes the actions in the specified order.

Compute Server

The compute server executes as a separate processor or on a different machine itself. It can be implemented as an object as well.

```
class CompServ{
private:
        int iPort;
        char *strHostName;
        ...
public:
        CompServ();
        ~CompServ();
        int ReceiveRequest(char *strSrcPath, char *strDestPath, ...);
        int ProcessRequest(...);
        int Reply(...);
```

```
    ...
};
```

ReceiveRequest(...)	This receives the formatted message.
ProcessRequest(...)	This processes the request. The user can extend the compute server by adding capabilities to this method.
Reply(...)	Sends the reply.

The compute server can also use the NAL to send and receive messages.

Fig. 6 is an object interaction diagram showing the order of creation of the objects/components of the manipulator 100 and the order in which the m-script is processed or read. Of note is the fact that the object switch 216 is called after service plugins 218. This order ensures that the services are all declared. AddAction takes the pointer to those service plugins, and AddServicePlugin identifies the compute server executing the plugin, its host name, and its port. ObjSw ensures the GAC 220 may be updated by the object switch with the results of the service, once executed.

Fig. 7 illustrates another embodiment of the inventive media manipulator 100. The media manipulator described in the previous sections was used as an intermediate processor between the client 106, 116 and the content provider server 104. In this alternative embodiment, an additional, stripped down tunneler version of the manipulator 100' can be used to interact between the client 106, 116 and the media manipulator 100 as described previously. These two instances of the manipulator 100, 100' can now perform in unison to further enhance the user experience.

The tunneler media manipulator 100' and the media manipulator 100 exchange a compressed format suitable for the transmission over a low-bandwidth connection, while the tunneler 100' and the browser(client) exchange information in the client's native format. Apart from these, the client 106, 116 can be inside a firewall f and still use the services of a main media manipulator 100, which may be outside the firewall f. The tunneler 100' can also be used to set various options such as compression quality, specific to the client's need. These options are forwarded to the main media manipulator 100 along with the client's request. The main media manipulator 100 can categorically act on both the tunneler's and client's request.

Apart from compressing images, the tunneler 100' and main media manipulator 100 combination can be used to compress the HTML page itself. The HTML page is a media, and if the service is available to compress it, the m-script can be modified appropriately to send the page to the text-compress-plugin before sending towards the client. The tunneler can intercept this and decompress the page.

The tunneler 100' has following components of the media manipulator: 1) media flow manager 210, 2) media parser 212, 3) object switch 216, 4) network access layer 214, and 5) service plugin 218. It does not the global access cache 220. The service plugin in the tunneler 100' is the compliment of what is used in the media manipulator to decompress the images.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

CLAIMS

What is claimed is:

1. A middle-ware computing system comprising:
   a network access system that supports communications with media resources and client computers; and
   a media manipulation system that operates on media objects received from the media resources via the network access system prior to forwarding the media objects to the client computers.

2. The computing system described in Claim 1, wherein the media manipulation system comprises:
   a parser that identifies different media types within the media objects; and
   service devices that operate on the media types.

3. The computing system described in Claim 2, wherein the parser searches for images in the media objects and service devices include an image compressor for performing data compression on the images.

4. The computing system described in any of Claims 2-3, wherein the parser searches for executable files in the media objects and service devices include a virus scanner that searches for computer viruses in the files.

5. The computing system described in any of Claims 2-4, wherein the parser searches for images in the media objects and service devices include a pornography detector for assessing a probability that the images are pornographic.

6. The computing system described in any of Claims 2-5, wherein the parser searches for data files in the media objects and service devices include an format converter for changing a format of the data files.

7. The computing system described in any of Claims 2-6, wherein the media manipulation system further comprises an object switch that passes the media types to the service devices to determine operations performed on the different media types.

8. The computing system described in any of Claims 2-7, wherein the media manipulation system further comprises a media flow manager that reassembles the media objects for forwarding to the clients after the manipulation of the media types.

9. The computing system described in Claim 8, further comprising a cache that stores media objects, the media flow manager receiving requests for media objects and checking for the presence of the media objects in the cache to preclude obtaining the objects from the media resources.

10. A middle-ware computing system comprising:

a network access system that supports communications with media resources to obtain media objects from client computers;

a parser that identifies different media types within the media objects;

service devices that manipulate the media types;

an object switch that passes the media types to the service devices to determine operations performed on the different media types; and

a media flow manager that reassembles the media objects for forwarding to the clients after the manipulation of the media types.

11. The computing system described in Claim 10, further comprising a cache that stores media objects, the media flow manager receiving requests for media objects and checking for the presence of the media objects in the cache to preclude obtaining the objects from the media resources.

12. A method for facilitating transmission of media objects between media resources and client computers, the method comprising:

receiving requests for media objects from the client computers to the media resources;

obtaining the media objects;

manipulating the media objects;

forwarding the manipulated media objects to the client computers.

13. The method described in Claim 12, wherein manipulating the media objects comprises:

identifying different media types within the media objects; and

performing separate operations on the different media types.

14. The method described in Claim 13, wherein the step of identifying different media types comprises searching for images in the media objects and the step of performing operations comprises data compressing the images.

15. The method described in any of Claims 13-14, wherein the step of identifying different media types comprises searching for executable files in the media objects and the step of performing operations comprises scanning the files for computer viruses.

16. The method described in any of Claims 13-15, wherein the step of identifying different media types comprises searching for images in the media objects and the step of performing operations comprises assessing a probability that the images are pornographic.

17. The method described in any of Claims 13-16, wherein the step of identifying different media types comprises searching for data files in the media objects and the step of performing operations changing a format of the data files.

18. The method described in any of Claims 13-17, further comprising reassembling the media objects for forwarding to the clients after the manipulation of the media types.

19. The method described in any of Claims 13-18, further comprising routing the media types to form successive operations on the media types.

20. The method described in any of Claims 13-19, further comprising caching media objects that have been received from the media resources and later obtaining the media objects from the cache.

21. The method described in Claim 20, wherein the step of obtaining the media objects comprises requesting the media objects from the media resources while checking for the objects in a cache; and obtaining the media objects from the cache if present.

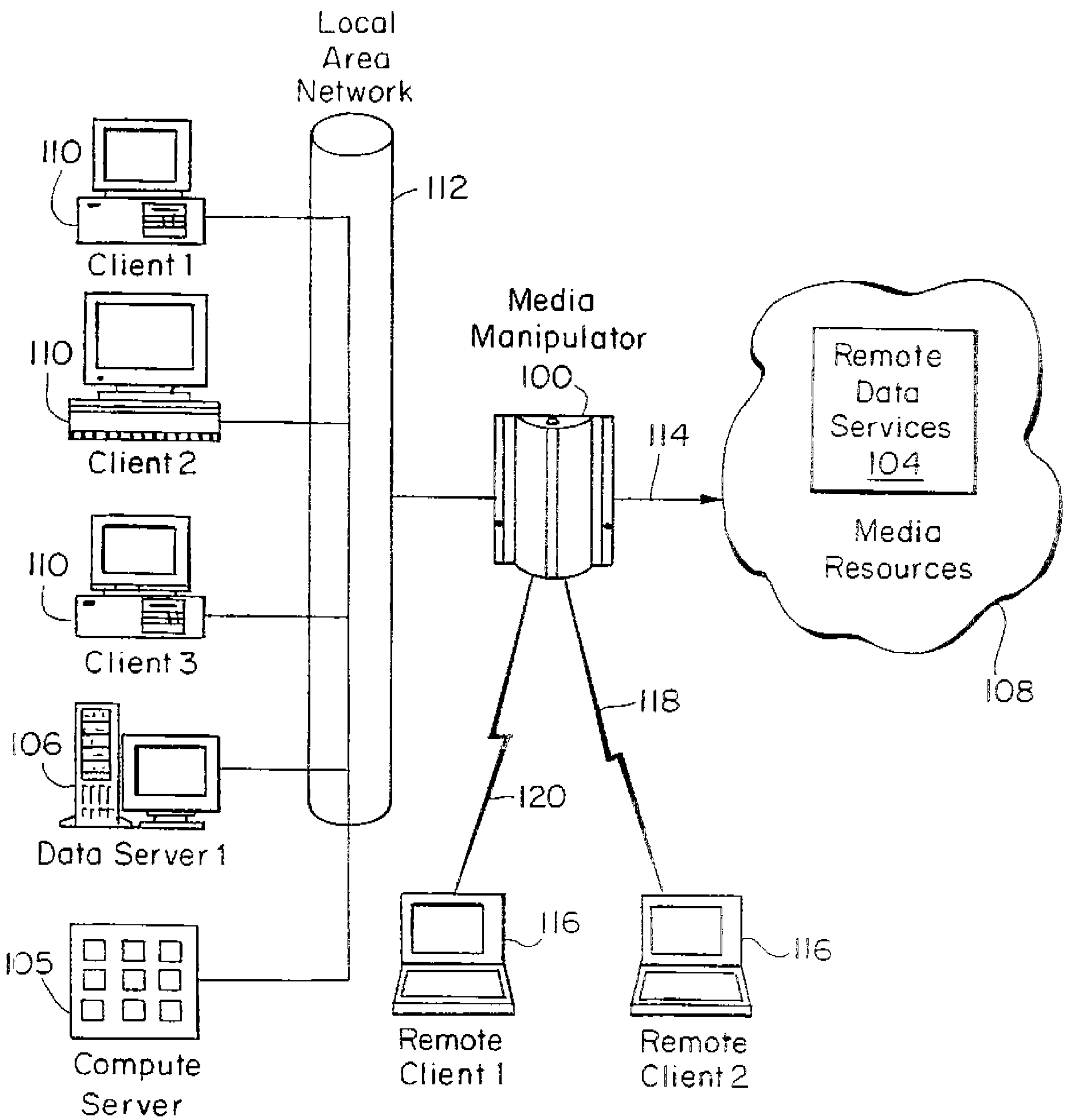

Local
Area
Network
110
112
Client 1
110
Client 2
110
Client 3
106
Data Server 1
105
Compute
Server
Media
Manipulator
100
114
118
120
116
Remote
Client 1
116
Remote
Client 2
Remote
Data
Services
104
Media
Resources
108

FIG. 1

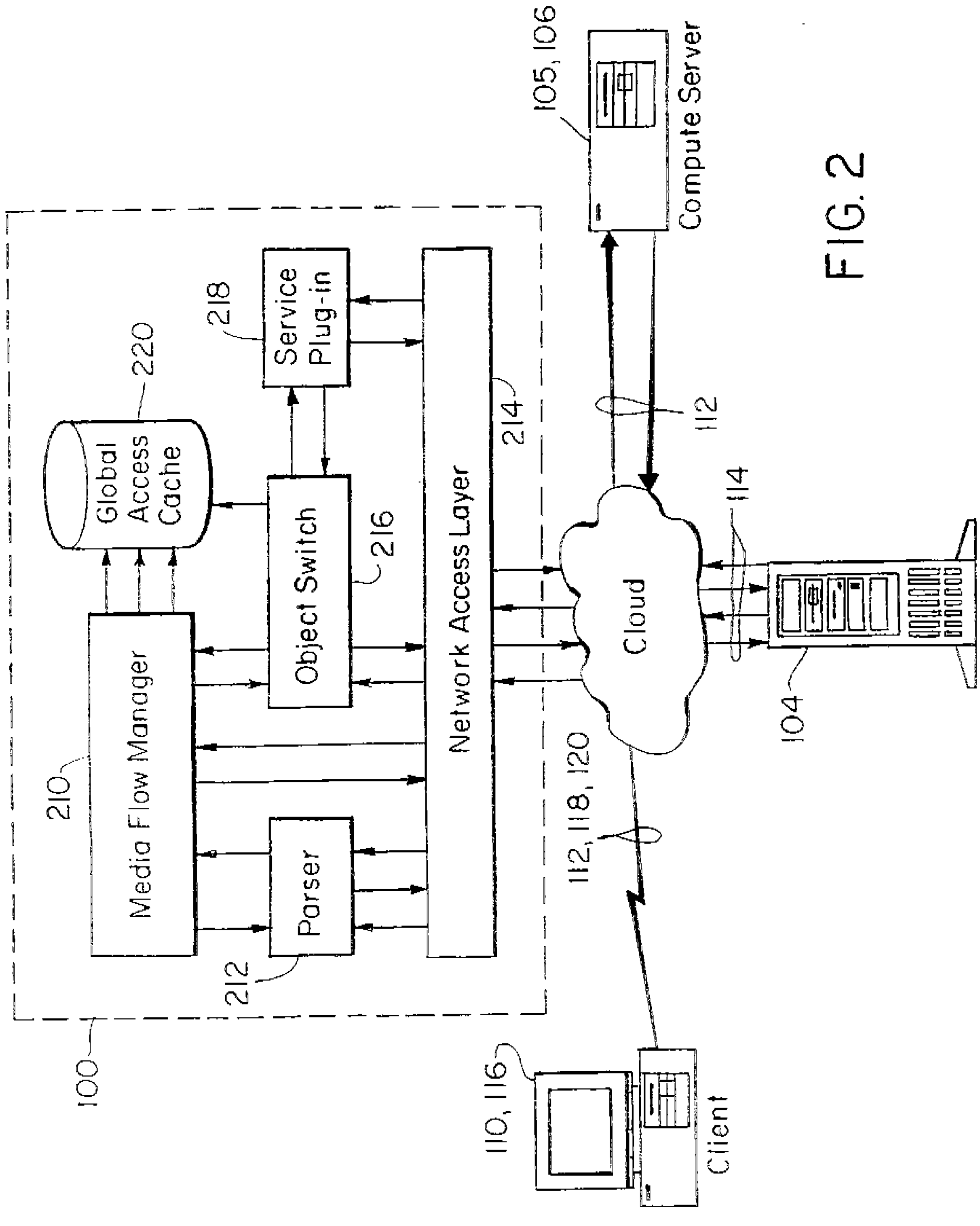

100
210
Media Flow Manager
212
Parser
Global Access Cache
220
Object Switch
216
218
Service Plug-in
Network Access Layer
214
Cloud
112
114
104
112, 118, 120
105, 106
Compute Server
110, 116
Client

FIG. 2

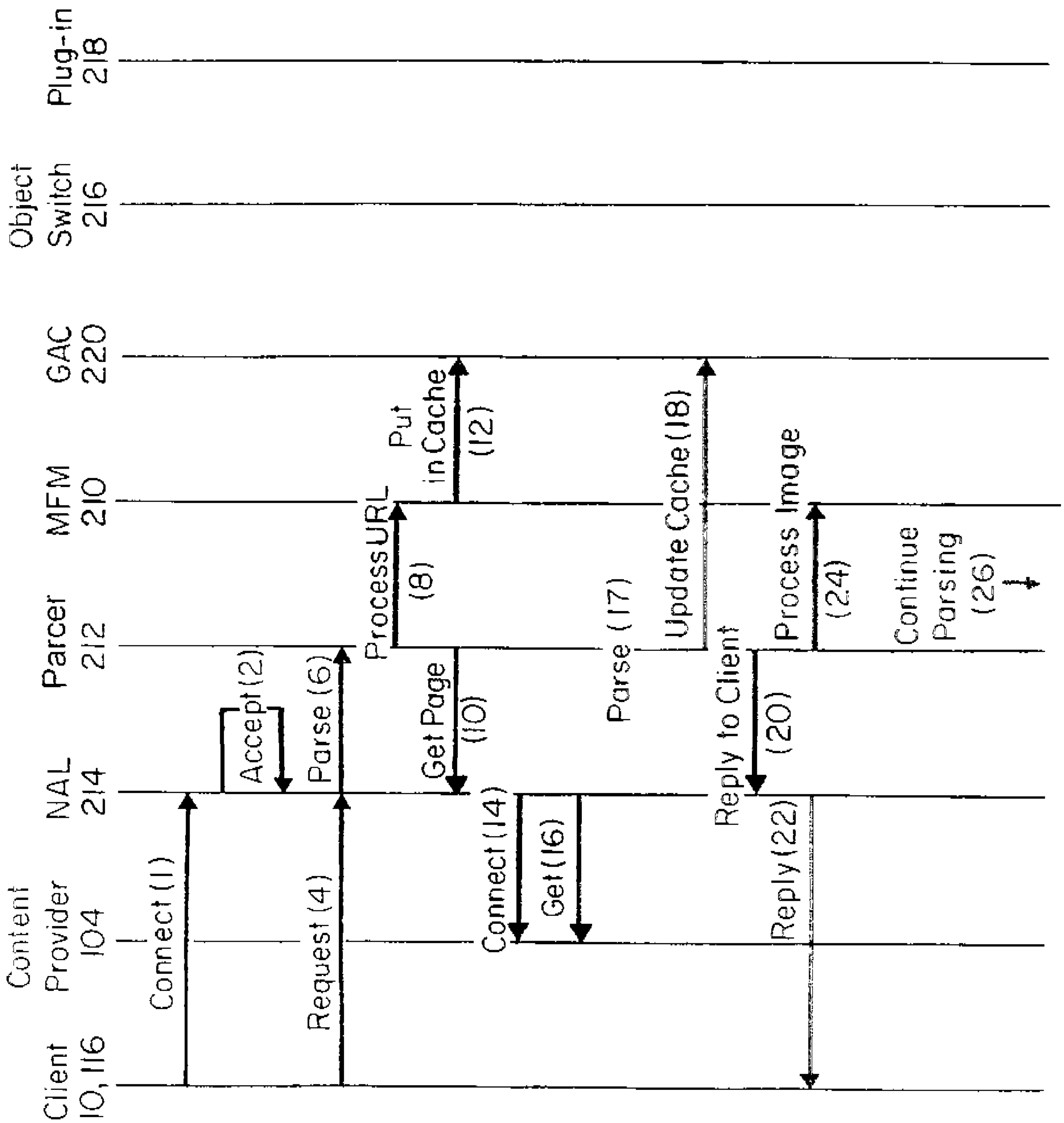
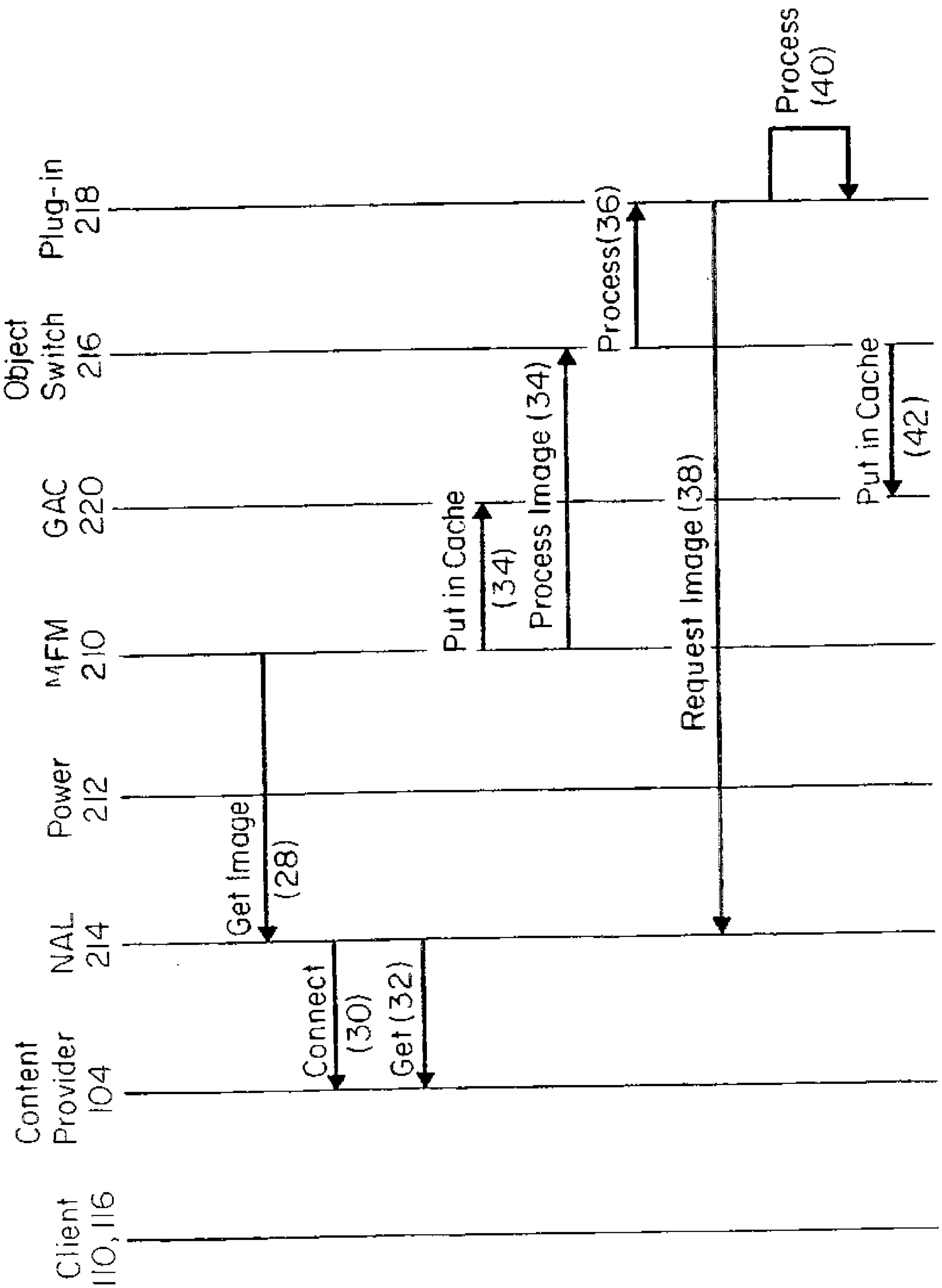

Client 110,116
Content Provider 104
NAL 214
Parcer 212
MFM 210
GAC 220
Object Switch 216
Plug-in 218
Connect (1)
Accept (2)
Request (4)
Parse (6)
Process URL (8)
Get Page (10)
Put in Cache (12)
Connect (14)
Get (16)
Parse (17)
Update Cache (18)
Reply to Client (20)
Reply (22)
Process Image (24)
Continue Parsing (26)

FIG. 3A



Client 110, 116
Content Provider 104
NAL 214
Power 212
MFM 210
GAC 220
Object Switch 216
Plug-in 218
Get Image (28)
Connect (30)
Get (32)
Put in Cache (34)
Process Image (34)
Process(36)
Request Image (38)
Process (40)
Put in Cache (42)

FIG. 3B

Header				Content						
Version	Length	Type	Message Id	Src Type	Src Path Len	Src Path	Dest Type	Dest Path Len	Dest Path	Dest Param

Field	Field Length	Description
Version	4	Message Version Number. E.g. 0100, implies 1.0
Length	4	Length of the content
Type	4	Type of the Message: 1-for request, 2-for reply, 3-for error
Message Id	4	Numeric ID of the message assigned by the NAL
Src Type	4	Numeric type of the source image: 1-GIF, 2-JPEG, 3-MM Compress Format 1
Src Path Len	4	Length of the Src Path
Src Path	-	Path where the image is stored. Can be a network path as well.
Dest Type	4	Numeric type of the final image: 1-GIF, 2-JPEG, 3-MM Compress Format 1
Dest Path Len	4	Length of the Dest Path
Dest Path	-	Path where the final image has to be stored
Dest Param	4	Can be used to set an optional parameter

FIG. 4A

Header				Content			
Version	Length	Type	Message Id	Reply Code	Dest Type	Dest Path Len	Dest Path

Field	Field Length	Description
Version	4	Message Version Number. E.g. 0100, implies 1.0
Length	4	Length of the content
Type	4	Type of the Message: 1-for request, 2-for reply, 3-for error
Message Id	4	Numeric ID of the message assigned by the NAL
Reply Code	4	The success or failure of the service: 1- success, 0- error
Dest Type	4	Numeric type of the final image: 1 - GIF, 2 - JPEG, 3 - MM Compress Format 1
Dest Path Len	4	Length of the Dest Path
Dest Path	-	Path where the final image has to be stored

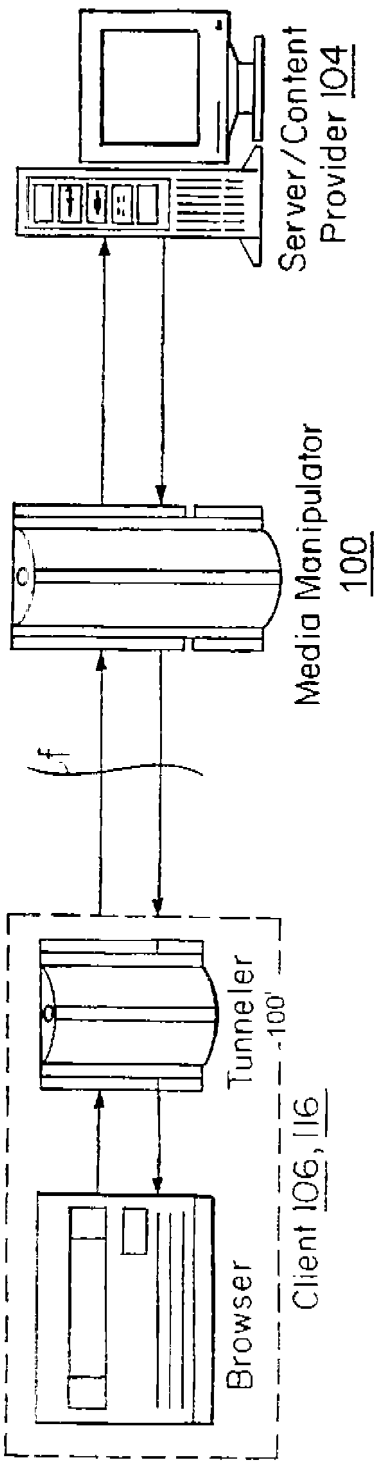
FIG. 4B

Header				Content			
Version	Length	Type	Message Id	Reply Code	Error Code	Error Reason Len	Error Reason

Field	Field Length	Description
Version	4	Message Version Number. E.g. 0100, implies 1.0
Length	4	Length of the content
Type	4	Type of the Message: 1-for request, 2 for reply, 3 for error
Message Id	4	Numeric ID of the message assigned by the NAL
Reply Code	4	The success or failure of the service: 0-error
Error Code	4	Numeric Error Code assigned by the compute server
Error Reason Len	4	Length of the reason, the next field
Error Reason	-	String describing the error

FIG. 4C

FIG. 7


Browser
Tunneler
100'
Client 106, 116
Media Manipulator
100
Server/Content
Provider 104

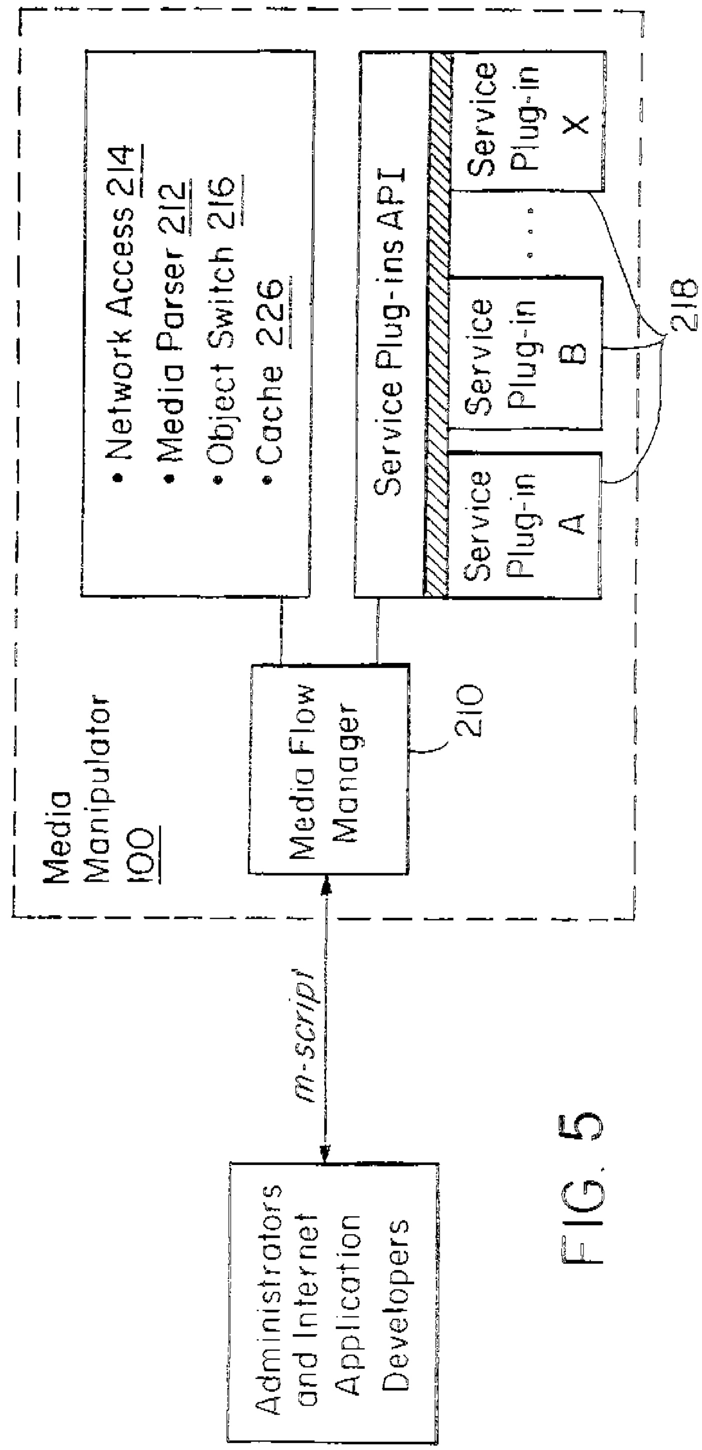

Media
Manipulator
100
Media Flow
Manager
210
• Network Access 214
• Media Parser 212
• Object Switch 216
• Cache 226
Service Plug-ins API
Service
Plug-in
A
Service
Plug-in
B
. . .
Service
Plug-in
X
218
m-script
Administrators
and Internet
Application
Developers

FIG. 5

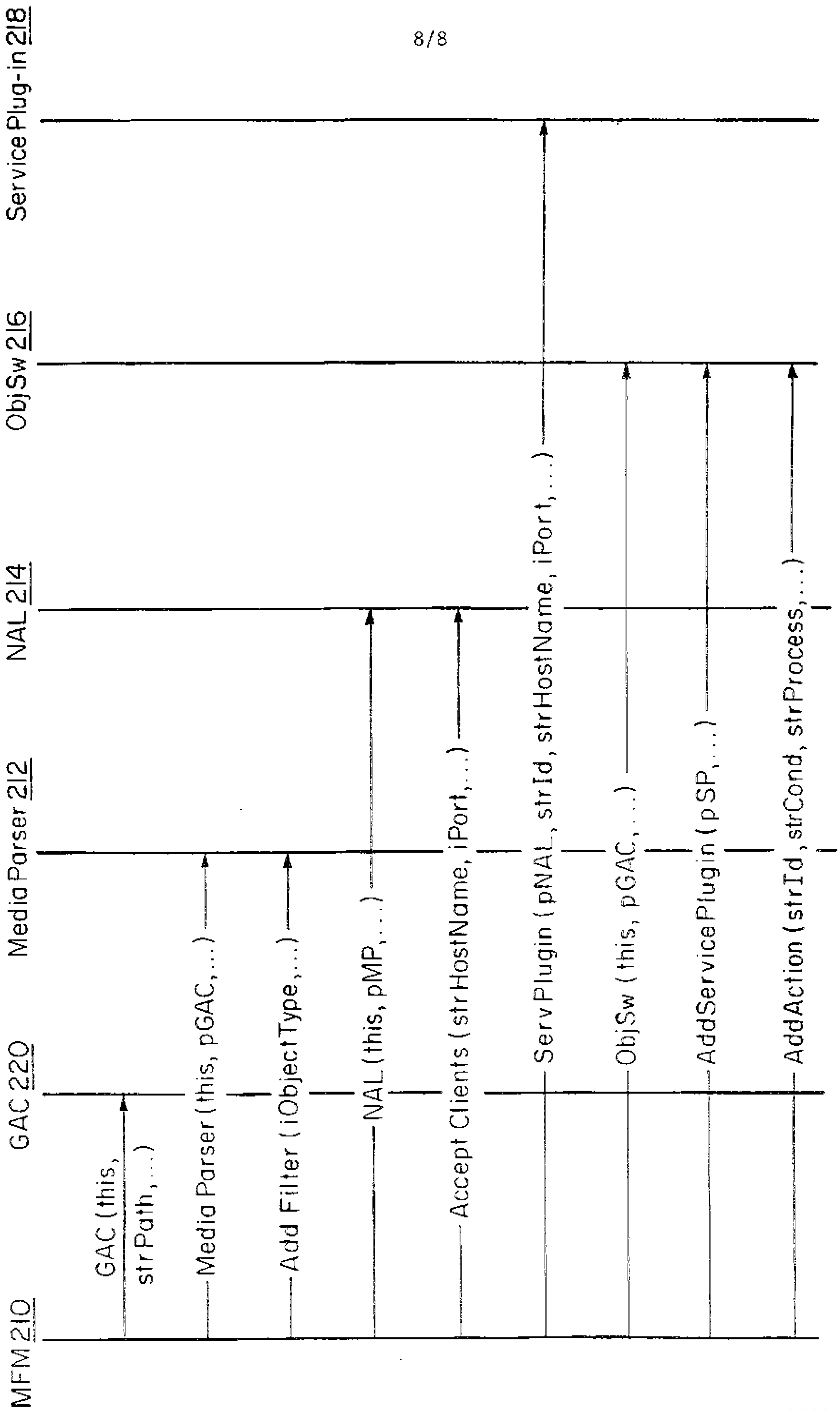

MFM 210
GAC 220
Media Parser 212
NAL 214
ObjSw 216
Service Plug-in 218
GAC (this, strPath, ...)
Media Parser (this, pGAC, ...)
Add Filter (iObjectType, ...)
NAL (this, pMP, ...)
Accept Clients (strHostName, iPort, ...)
ServPlugin (pNAL, strId, strHostName, iPort, ...)
ObjSw (this, pGAC, ...)
AddServicePlugin (pSP, ...)
AddAction (strId, strCond, strProcess, ...)

FIG. 6

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 29/06 A3

(11) International Publication Number: WO 97/49252

(43) International Publication Date: 24 December 1997 (24.12.97)

(21) International Application Number: PCT/US97/10758

(22) International Filing Date: 20 June 1997 (20.06.97)

(30) Priority Data:
60/020,094 21 June 1996 (21.06.96) US

(71) Applicant *(for all designated States except US)*: INTEGRATED COMPUTING ENGINES, INC. [US/US]; 460 Totten Pond Road, Waltham, MA 02154 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: SHAH, Ashesh, C. [US/US]; 567 Tremont Avenue, No. 31, Boston, MA 02118 (US). PEDERSEN, Palle [DK/US]; 82 Commonwealth Avenue, No. 10, Boston, MA 02116 (US). RADOVIC, Niksa [HR/US]; 19 Mountain Avenue, Somerville, MA 02143 (US). MANICKAVASAGAM, Senthilkumar [IN/US]; 11 Highland Glen Drive, No. 17, Randolph, MA 02368 (US).

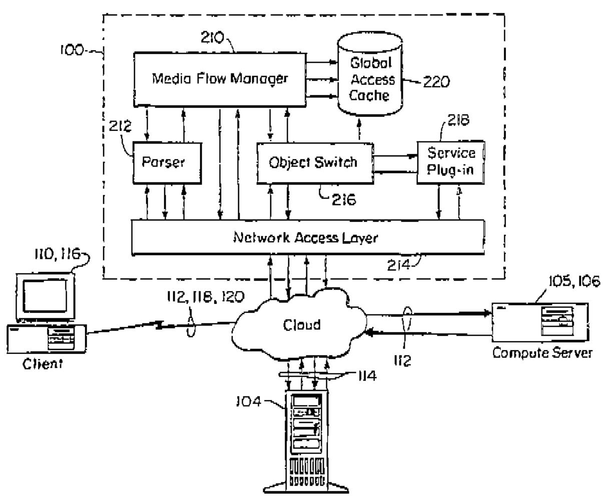
(74) Agents: SMITH, James, M. et al.; Hamilton, Brook, Smith & Reynolds, P.C., Two Militia Drive, Lexington, MA 02173 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(88) Date of publication of the international search report: 30 April 1998 (30.04.98)

(54) Title: NETWORK BASED PROGRAMMABLE MEDIA MANIPULATOR



(57) Abstract

The media manipulator is a middle layer between the clients (110, 116) and the remote data servers (104) is the common client-server organization. It transforms the network into a more flexible three-tiered configuration. Requests generated by the clients (110) for media objects from media resources are routed to the media manipulator (100). It processes the requests and determines if the media objects may be found locally, either cached (220) in the media manipulator (100) itself or in the local data servers (106). When the media objects are obtained, the media manipulator (100) can be used to perform operations on those objects such as format translations, to apply protective mechanisms for the clients (110), to speed communications between the remote servers (104) and the clients (110), or perform compute operations for the clients (110). In one example, a parser (112) of the manipulator (100) searches for images in the media objects so that service devices (218) can be called to perform data compression or pornography detection on the images. The parser can also search for executable or data files in the media objects and to perform virus scanning or format conversion, respectively.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 97/10758

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 669 587 A (AT & T CORP) 30 August 1995	1,12
A		2,3,7,8,10,13,14,18
	see column 4, line 20 - column 5, line 14; figure 1 see column 7, line 13-34; figure 2 see column 9, line 17-29 see column 16, line 39-51 ---	
A	THAU R: "Design considerations for the Apache Server API" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 28, no. 11, May 1996, pages 1113-1122, XP002046988 see paragraph 4 --- -/--	2,10,13

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

20 November 1997

Date of mailing of the international search report

18.12.97

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Dupuis, H

7

INTERNATIONAL SEARCH REPORT

Inte[rnati]onal Application No
PCT/US 97/10758

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	TREVOR J ET AL: "Exorcising daemons: a modular and lightweight approach to deploying applications on the Web" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 28, no. 11, May 1996, pages 1053-1062, XP002046968 see paragraph 3 - paragraph 3.1 ---	2,3,7,8, 10,13, 14,18
A	WO 96 17306 A (ORACLE CORP) 6 June 1996 see page 8, line 16-21 see page 11, line 29 - page 13, line 22; figure 1 see page 15, line 6-16 see page 17, line 27 - page 18, line 15 see page 34, line 2-28 ---	1,6, 9-12,17, 20,21
A	HOWLETT D: "Protection on the Web" COMPUTERS AND SECURITY, vol. 15, no. 4, 1996, page 319 XP002046969 see the whole document -----	4,15

7

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 97/10758

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0669587 A	30-08-95	CA 2140850 A	25-08-95
WO 9617306 A	06-06-96	NONE	

ORIGINAL

AUSTRALIA

Patents Act 1990

COMPLETE SPECIFICATION

APPLICANT: Dudley John MILLS

ADDRESS: 30 Hutchison Crescent,
Kambah, ACT 2902

ACTUAL INVENTOR: Dudley John MILLS

ADDRESS FOR SERVICE: Dudley J. Mills,
30 Hutchison Crescent,
Kambah, ACT2902

ASSOCIATED PROVISIONAL: PO5254 Filed 21 February 1997

INVENTION TITLE: NETWORK-BASED CLASSIFIED INFORMATION SYSTEMS

The following is a full description of the invention including the best method of performing it known to me:

AU9853031

(12) PATENT ABSTRACT (11) Document No. AU-A-53031/98
(19) AUSTRALIAN PATENT OFFICE

(54) Title
NETWORK-BASED CLASSIFIED INFORMATION SYSTEMS

International Patent Classification(s)
(51)[6] G06F 017/30

(21) Application No. : 53031/98 (22) Application Date : 10/02/98

(30) Priority Data

(31) Number (32) Date (33) Country
PO5254 21/02/97 AU AUSTRALIA

(43) Publication Date : 27/08/98

(71) Applicant(s)
DUDLEY JOHN MILLS

(72) Inventor(s)
DUDLEY JOHN MILLS

(57)

A system for automatically creating databases containing industry, service, product and subject classification data, contact data, geographic location data (CCG-data) and links to web pages from HTML, XML or SGML encoded web pages posted on computer networks such as the Internet or Intranets. The web pages containing HTML, XML or SGML encoded CCG-data, database update controls and web browser display controls are created and modified by using simple text editors, HTML, XML or SGML editors or purpose built editors. The CCG databases may be searched for references (URLs) to web pages by use of enquiries which reference one or more of the items of the CCG-data. Alternatively, enquiries referencing the CCG-data in the databases may supply contact data without web page references. Data duplication and coordination is reduced by including in the web page CCG-data display controls which are used by web browsers to format for display the same data that is used to automatically update the databases.

TITLE: NETWORK BASED CLASSIFIED INFORMATION SYSTEMS

FIELD OF INVENTION

This invention relates to network based classified information systems, to methods of automatically building searchable databases of classified information derived from web pages posted on a network, and, to web pages for use in such systems and methods.

The information systems and databases of most relevance to this invention are those which include classified product and service catalogues similar to the Yellow Pages telephone books, contact indexes similar to the White Pages telephone books, and/or subject indexes similar to Library catalogues. Such information systems and databases typically include sets of associated classification, contact and/or geographic items of information. For convenience, classification, contact and/or geographic information will be hereinafter called CCG-data.

The networks with which this invention is concerned are the worldwide public computer/communications network commonly known as the Internet and private networks – sometimes called intranets – which allow common access to markup documents on computers connected to the network. Markup documents are text files prepared using various markup languages such as HyperText Markup Language (HTML) and Extensible Markup Language (XML) which are implementations (or dialects) of the Standard Generalised Markup Language (SGML). The system of accessible files on the Internet is called the World Wide Web (WWW) and the markup documents themselves are commonly called 'web pages'. A web page is said to be 'posted' on a network when it is stored on computer-readable media of a host network computer as a file which is generally accessible to network users. A web page is transported from the host computer to a requesting computer through intermediate network computers as a computer-readable signal embodied in a carrier wave. Though this invention is not limited to Internet based information systems, these terms are used for convenience.

BACKGROUND TO THE INVENTION

It has been estimated that there are about 100 million web pages on the Internet and that the number is doubling every two years. Many of these pages include information concerning commercially offered goods and services and often include contact details. But the difficulty of locating such information is increasing faster than the growth in the number of web pages.

To assist network users locate web pages of interest, certain network service providers create indexes (or databases) of the contents of web pages posted (stored on computer readable media so as to be generally accessible) on the network and provide 'search engines' to use the indexes. These indexes are often created automatically by the use of 'web crawlers' which (i) interrogate computer after computer on the network to locate successive web pages and (ii) index the words in each web page encountered against the network address (eg Internet Protocol Address or IPA) and filing system path or universal resource locator (URL) at which the web page is accessible. Hereinafter the terms URL and URI (Uniform Resource Identifier) are taken to be identical in meaning and to signify network addresses and filing system paths. Usually, the indexes consist of a list of unique words with each word having an associated list of URLs of the web pages wherein the word was found to occur during interrogation. The URL serves as a 'hyperlink' which, if selected by a user/searcher, results in the associated web page being automatically transmitted from the computer where it is posted on the network to the user/searcher's computer where it may be displayed or otherwise processed. The sending and receiving of files in this way is greatly assisted by user interface programs called 'web browsers' (or more simply, 'browsers') such as Netscape and Microsoft Internet Explorer.

The search for web pages of interest using search engines leaves much to be desired:

- simple searches (those using a few keywords in simple combinations) often yield far too many web page references (URLs) to permit them to be interrogated one-by-one,
- complex searches (those using many keywords and/or complex Boolean expressions) require considerable expertise to undertake,
- even using optimum search criteria, many irrelevant web pages are referenced because of inconsistent use of terminology by those who author the original web pages,
- even using optimum search criteria, many relevant pages are missed, again because of inconsistent use of terminology by web page authors, and
- because items of information included in the body of web pages cannot be 'understood' or associated in useful ways by web crawlers; that is recognised as, say, a surname, a street name, a geographic locality, or type of goods or services and, say, a surname strongly associated with a street name, a geographic locality, or a type of goods or service.

The result is that information provided by search engines from databases which are automatically compiled using web crawlers is a very poor equivalent of the common Yellow Pages and White Pages directories which serve the telephone industry (though these directories are not, of course, automatically compiled from web pages).

In an attempt to improve the usefulness of automatically compiled network databases, some search engine providers make use of information contained in URLs, such as the country code and top level domain name codes such as 'com', 'edu', 'net' and 'org' which is sometimes used to signify the subject matter of web pages. It has been proposed to add more content classifying codes to URLs (eg, "chem" to signify chemical subject matter) to allow specialised databases – national, commercial, chemical, etc – to be generated. However, this proposal has serious drawbacks:

- URLs are Internet addresses and it is in principle undesirable to confuse the address function of a URL with that of representing a list of web page classifications or contact details.
- A URL is an inappropriate container of multiple web page classification codes and contact details because the length of the URL would cause it to become unwieldy as an Internet address.
- Including in a URL classification codes drawn from a list of thousands of codes would compromise the mnemonic quality of Internet addresses such as "www.yellowpages.com".
- There is substantial overlap in the subject matter contained in web pages having the various top level domain name codes.
- There is no consensus on, or standard for, content classification codes in URLs.

Another proposal to add content classification data to web pages has arisen from the wish to identify pages containing material that may be offensive to some viewers, or should not be accessed by minors. The Platform for Internet Content Selection (PICS) (see http://www.w3.org/pub/WWW/PICS and other documents at www.w3.org) is a web page ratings standard similar in principle to the ratings systems for motion pictures. This system allows page authors to "internally" self classify their pages through use of the "<meta...>" HTML element. Alternatively, "external" PICS ratings of web pages may be obtained from ratings service providers accessed each time a URL is selected. In practice, the ratings service providers have adopted very limited range of web page classifications. For example, Ararat Software's Commercial Rating System (see http://www.ararat.com.ratings/ararat10.html) provides just 5 categories of web page content: commercial content, technical/customer support, ordering information, downloading information and contact information. In other examples, CyberPatrol (http://www.microsys.com/pics/pics_msi.htm) provides 16 categories, the Recreational Software Advisory Council (http://www.rsac.org/faq.html) provides 4 categories, SafeSurf (http://www.safesurf.com/ssplan.htm) provides 11 categories and Vancouver Webpages Rating Service (http://vancouver-webpages.com/VWP1.0/ provides 11 categories. None of the categories provide classification of web pages by industry, service, product or subject with sufficient specificity to be useful when searching for web pages. Rather, the categories are intended to prevent web browsers from displaying web pages unsuitable for particular types of web browser users. Such rating systems are not intended to be used for the automated creation of Yellow or White pages like databases from web pages and are unsuitable for that purpose because they can not represent contact details. Further, the ratings data may only be encoded in the <meta...> element in the <head> of an HTML document drastically limiting the type and usefulness of the data that can be encoded.

Another proposal for classifying the content of web pages, the "Meta Content Framework" (MCF - see http://mcf.research.apple.com/mcf.html"), requires the content of web pages to be classified and the classification data to be held in a separate non-HTML data file with a MIME type of text/mcf. Storing data in non-HTML encoded documents which describes the content of HTML encoded documents is a technical and economic barrier to the adoption by search engine providers of the proposal. The MCF proposal is thus entirely unsuited to the automated creation of Yellow or White pages like databases from HTML encoded web pages (MIME type text/html) because data stored according to the MCF proposal is not stored in HTML encoded web pages.

The "Electronic Business Card", vCard, (see "vCard The Electronic Business Card" Version 2.1, versit Consortium Specification, Sept 18, 1996 or ftp://ds.internic.net/internet-drafts/draft-ietf-asid-mime-vcard-01.txt) uses non-HTML data file (MIME Content Types of "text/plain" or the non-standard "text/X-vCard") containing contact information equivalent to an extended White Pages entry which can be exchanged on a network using Simple Mail Transfer Protocol (SMTP) or using HTTP. It can be associated with a web page by use of a URL in the web page which refers to the vCard information (eg My vCard). Version 2.1 vCard standard data file format (published 18 September 1996) provides for the inclusion of many items of contact information. The vCard specification recommends that, where possible, there should be consistent mapping of vCard property names to HTML "<input>" element attribute names (eg vCard property name "TITLE" maps to HTML "<input name= 'title'>"). The intention is to facilitate the transfer of vCard data into web page input forms by pasting from a clipboard or by dragging from other computer applications. The VCard proposal is unsuited to the automated creation of Yellow or White pages like databases from HTML encoded web pages because data stored according to the VCard proposal is not stored in HTML encoded web pages.

The inclusion of classified information in separate documents (such as Meta Content files or vCards) has the disadvantage that there is necessarily much duplication of data and coordination of modifications between the separate documents and the web pages. This must be done to allow a person who has accessed a web page using an HTML compliant browser to determine whether it is worth calling up the associated file or vice versa. Also, to allow portions of web pages to be classified, web page contextual information would have to be duplicated in the separate document. vCards in particular do not provide this functionality. Another disadvantage is that non-HTML documents such as vCards contain no details as to how the data they contain is to be displayed. In the display of HTML documents the position, font, size, colour of the text and other elements of the document are of great importance. The restriction of address data in a vCard to untagged ordinally organised fields is inflexible. For example, multiple instances of extended parts of the address are not possible. Also components of names, addresses and telephone numbers and so forth are insufficiently identified.

The Online Computer Library Center Inc (OCLC, Dublin, Ohio, USA) proposal, known as the "Dublin Core", proposes to classifying scholarly web pages by subject (topic of the work, or keywords that describe the content of the work), title, author, publisher, other agent, date, object type (genre of the object such as home page, novel, poem etc), form, identifier, source, language, relationship and coverage (spatial and temporal) (see http://www.oclc.org:5046/~weibel/html-meta.html and other documents at www.oclc.org). This proposal does not include industry, service, product or subject classifications. It also does not include contact details. Names such as that of the author are not specified in sufficient detail to avoid ambiguities such as which is the author's first and last names. The proposal specifies that the details are encoded using the <meta...> element in the <head> of web pages. The proposal is unsuited to the automated creation of Yellow or White pages like databases from web pages because the proposal does not provide for classification of web pages and does not provide adequate contact details. Further, the use of keywords for describing the content of the work adds very little to the effectiveness of indexing of web pages since the web pages are usually indexed on every word of their content and most often the key words would simply be a duplication of words already contained in the document.

It has also been proposed to use the Dewey Decimal System (see http://orc.rsch.oclc.org:6109/eval_dc.html and http://orc.rsch.oclc.org:6109/bintro.html) to rank electronic documents against a Dewey Decimal subject classification. The proposal suggests automatically assigning Dewey Decimal subject classification codes to documents during automated indexing and cataloguing but does not specify the exact nature of the assignment although it is implied that the codes are stored separately from the documents. The proposal admits that such automated classification is less satisfactory than human classification. The proposal is unsuited to the automated creation of Yellow or White pages like databases from web pages because the accuracy of classification is inadequate, does not provide for inclusion of industry, service or product classifications and does not provide for inclusion of contact details. Deriving a subject classification code from an analysis of every word and phrase in a web page is computationally expensive.

The HTML 3.0 standard (see page 23 of the www.w3.org document "draft-ietf-html-specv3-00.txt") provides "class" as an attribute of almost all HTML "<body>" elements. The "class" attribute is intended to be used with style sheets. Style sheets provide a means by which the display of HTML documents may be altered to suit the needs of different classes of browser users. For example, <div class="appendix"> could be used to define a division that acts as an appendix, <h2 class="section"> could be used to define a level 2 header that acts as a section header, although, of course, any string of characters could be defined for those purposes. The "class" attribute, although never having been suggested for holding goods and services classifications, is not suited for such a use as it is, in any case, undesirable to confuse the style sheet function of the "class" attribute.

The HTML 3.0 and earlier standards provided the HTML elements "<person>" and "<address>" but do not specify the form of the content or method of validating the content of those elements. A person's name may be written as first name followed by last name or last name followed by first name. Similarly, different conventions exist for writing addresses. Similar ambiguities arise in the ill defined format of the HTML elements "<person>" and "<address>". As such they are of little use in the automatic compilation of searchable databases.

The XML language (see: http://textuality.com/sgml-erb/WD-xml.html) was developed to extend HTML so that software vendors can add new elements and new element attributes to HTML which are not specifically defined in any HTML standard. The intention is to ensure that all new elements and attributes could be parsed by all XML parsers even if the new elements held no significance for any particular XML parser. However, like HTML, XML does not provide a standard for the representation of industry, service, product or subject classification, contact or geographic location details within an web page.

Of course, many useful databases of the Yellow Pages or White Pages type are made available by service providers on networks, but they are not compiled automatically by using web crawlers to scan HTML web pages posted on a network. For example, http://www.yellowpages.com.au and http://www.mcp.com provide classified advertisements of the Yellow Pages type with links to the web pages of paying advertisers or subscribers. There are also directories of email addresses which approximate the White Pages directories, listing the names of individuals and organisations and contact details, (eg http://www.bigbook.com and http://query1.whowhere.com). However, these email directories require listers to manually add their directory entries and enquirers to be aware of and to find the directory enquiry web page. They cannot be automatically generated by scanning web pages using web crawlers since there is no adequate mechanism to relate email addresses to the names of people and organisations and their other contact details which may also exist in the same web page.

OBJECTIVES OF THE INVENTION

The general object of the invention is to provide improved methods for automatically building searchable databases of classification, contact, and/or geographical information by using web crawlers to interrogate web pages posted on a network. [For convenience, this information is collectively referred to as CCG-data].

Other non-essential objectives are to provide methods for including and/or displaying CCG-data within web pages accessed by browsers, for automatically extracting CCG-data from web pages posted on a network and for using the same, and/or to provide methods for searching automatically compiled databases using such data.

Another subsidiary objective of the invention is to provide a new form of web page which is better suited to the automatic compilation (using web crawlers) of databases constructed by the automatic scanning of many such pages posted on a network.

OUTLINE OF THE INVENTION

The invention is based upon the realisation that highly useful databases can be automatically built by successively interrogating web pages posted on a network if one or more HTML encoded CCG phrases are included in the web pages. A CCG phrase is one containing CCG-data in a form which is directly accessible and identifiable. CCG phrases may also include one or more items which provide the web page author with control over how the CCG-data is applied to the database.

Data duplication can be reduced if some of the CCG-data in the coded CCG phrases can be displayed by browsers as well as being used to update databases. Errors due to inexactly duplicated data are also eliminated. Accordingly, it is envisaged that CCG phrases may include one or more items which provide the web page author with control over how the CCG-data is displayed by a browser.

HTML (including version 2 and version 3) and XML are evolving applications (sub-sets or dialects) of ISO Standard 8879 1986 known as Standard Generalised Markup Language (SGML). HTML, in large part, is a language used to describe how text (unstructured data) and graphics is to be formatted for display. The HTML language consists of a finite number of "elements" (for example: "
" where "BR" is the element name, also called the tag name) which may contain "attributes" (for example; "<DL COMPACT>" where "COMPACT" is an attribute named "COMPACT") and may contain values associated with attributes (for example; "<FONT SIZE=+1>" where +1 is the attribute value of the attribute named "SIZE"). XML is a language used to describe structured data. The XML language is similarly composed of elements, attributes and values with a similar syntax to HTML but unlike HTML the element names which may be used are not restricted and the meaning of the XML data may be interpreted in any convenient manner. While the XML language is mute about how data described by XML is to be formatted for display, the data may be used by computer programs for any purpose including description of how XML coded data is displayed. However, due to its historic importance in connection with web pages, the term "HTML" is herein used to refer to all markup languages which are subsets or complete sets of the SGML language. In particular, the term "HTML encoded CCG phrase" and the synonymous term "CCG phrase" are herein used to refer to CCG-data encoded in a subset or complete set of the SGML language. Herein, a "web page" is a document adapted to be or actually accessible through a network and encoded in a subset or complete set of the SGML language.

For convenience, CCG items in HTML encoded CCG phrases, whether they are syntactically represented as elements or as attributes, will be referred to hereinafter as CCG attributes.

A CCG phrase includes at least one of the following identifiable types of CCG-data attributes:

- industry, product, service, and/or subject classifications,
- contact categories, contact person(s) and/or organisation(s) names, titles or associations, contact details including physical and postal addresses, telephone and fax numbers, email and Internet or network addresses or locations, public keys, and
- geographic location details.

A CCG phrase may also include any of the following identifiable types of CCG control attributes:

- database control attributes to indicate which parts of the data are to be used to update databases, and
- display control attributes to indicate how browsers are to display the data.

By virtue of occurring in the same CCG phrase, a plurality of CCG-data attributes are associated with each other.

By virtue of their occurrence in the same CCG phrase, CCG-data attributes are idententified as a set of associated attributes. However the degree of association between attributes can be controlled by the inclusion in the phrase of database control attributes.

The start and end of CCG phrases should be identifiable to clearly distinguish these phrases from other data. To identify the beginning and end of a CCG phrase, at least one HTML element should have a CCG specific HTML element name or CCG specific attribute name or

CCG specific value. Each CCG attribute may consist, with or without other incidental characters, of a CCG attribute name and/or a CCG value or values. Preferably, each CCG phrase is contained in the "<body>" of the web page.

Two examples of a CCG specific HTML element are: "<CCG ...>" or "<CCG ... />" or "<CCG>...</CCG>". (Where a CCG phrase is coded in XML, the elements "<XML>" and "</XML>" may also be needed at the start and end of the CCG phrase.) A less satisfactory example is: "<!–CCG ...–> where the characters "CCG" after HTML comment element name "!–" are used to signify that the comment contains CCG-data. An example of the use of a CCG specific attribute name is: "<START CCG>"..."<END CCG>". An example of the use of a CCG specific value is: "<START TYPE='CCG'>"..."<END TYPE='CCG'>". Obviously, other character strings could be substituted for the element name, element attribute name or element attribute value "CCG" string of the examples.

The codes "<CCG ...>" and "<CCG ... />" are compatible with most HTML specifications, but being non-standard HTML, most web browsers do not display any text or attributes (eg PQ="AQD") within the angle brackets "<" and ">". These codes are preferred where display of the CCG data is not required and compatibility with older browsers is required (eg CCG phrases containing only classification values).

From one aspect, therefore, the invention comprises a web page for posting on a network, the web page being characterised by the inclusion of at least one CCG phrase in the "<body>" of the page, the CCG phrase being such that the CCG attributes contained therein are accessible and identifiable by (i) HTML compliant editors and/or (ii) HTML compliant web crawlers for the automatic construction of databases of classified information, and/or (iii) HTML compliant browsers for display on the computer screens of network users.

From another aspect, the invention comprises a method of constructing web pages of the above described type. The web pages may be constructed on digital computers using simple text editors such as Microsoft Windows Notepad, or preferably, purpose built human controlled editors or automated composing programs which embody knowledge of HTML and CCG syntax and grammar. Which ever process is used, CCG attributes are selected and inserted, modified, deleted and/or organised to form a valid CCG phrases in HTML encoded documents and the documents are posted on computer readable storage devices of computers connected to a computer network so that the documents are generally available to computers on the network.

From another aspect, the invention comprises a method of populating a database with CCG-data extracted from web pages. Web pages posted on a network are successively retrieved by a digital computer program (eg: a web crawler) and CCG phrases contained therein are identified and at least some of the CCG attributes found within the CCG phrases are extracted. The CCG attribute names are used to determine the type of data in the associated values. Generally the CCG attributes of interest are those relating to classification, contact and geographic data and database update controls while the attributes of little or no of interest in relation to database updating are those relating to display controls. Of course, the CCG-data extracted need only be that relevant to the particular database being updated. For example, one database may have been designed to index only web page classifications and URLs while another database may have been designed to index only contact details. Databases also differ in their internal representation of data and means of associating data. For example, some use "flat file" tables, others use pointers to data to create network associations while others use hashing and buckets.

The conventional nomenclature differs considerably between different types of database. Depending on the particular database nomenclature, data of the same type is said to be stored in table columns, fields, attributes and properties. The terms column and field are somewhat related to the physical representation of the data in files while attribute and property is more related to the logical representation of data. To avoid confusion, with the terms "HTML attribute", "CCG attribute" or just "attribute", hereinafter a database property means both a type of data stored in the database and a place in the database where data of the same type is stored. Database properties are referred to by a name ("property name") or similar reference and contain values. For example, a database property with the name "City name" and which contains values which are all the names of cities may be defined as a "City name" type database property.

Whichever style of database is used, it is preferred that the database update program relate the CCG attributes to corresponding database properties used by the database update process so that the database property values are updated with CCG values in a manner which preserves the distinctness, content and meaning of the CCG values and, preferably, preserves the CCG value associations expressed in the CCG phrase as sets of associated database property values of different types.

In some cases, it is desired to know the address of the web page from which the CCG values were extracted. For example, the purpose of building a database might be to allow searching of the database by web page classification to provide a list URLs of web pages or URLs of portions of web pages which contain matching CCG classifications. The URLs could then be inserted in an HTML document and transmitted to a web browser as a list of references to web pages matching a search expression. In that example, associating the URL of a web page or the URL of a portion of a web page with the CCG values extracted from the same web page or web page portion is important and the URL or means of reconstructing it must be available and supplied to the database update process. In one style of database, the values of the same type are held separate rows in a column (property) of a database table, and pointers held in another column (property) are associated with the values by sharing the same table row. The table row constitutes a set of associated property values. Each pointer points to a bucket (block of data) containing a list of URLs or pointers to URLs held in a separate bucket or table. In another style of database, values of different types are held in different tables together with a set number, pointer or similar code which is used to indicate which values are associated as members of the same set. In one variation, the values of set members are prefixed with a code indicating the type of value and all values are held in the same column of a table. If the purpose of the database is to hold contact data, recording the web page URL in the database might not be required although if the URL is not present in the database, updating changes in the CCG contact details contained within a web page is more difficult. Of course, one database may be used to record all types of CCG values contained in web pages and associate with each other any and all values extracted from the same web page or even from other web pages.

From another aspect, the invention comprises a method of searching the databases constructed as outlined above. These databases may be used for a variety of searching purposes. For example, to find web page URLs by using the association of web page URLs with industry, service, product or subject classification or a person's or organisation's name or address or geographic location values or any combination thereof. In another example, the databases may be used to find the contact details for people or organisations by name or location of industry, service, product or web page subject type and so forth by using the association between items of the contact details in the database without having to retrieve web pages associated with the contact details.

More particularly, the searching method involves finding URL references, or finding sets of associated database property values, from databases containing CCG-data. The method including steps of parsing a query phrase received from a computer network to extract query relational expressions and, from each expression, deriving a query field name, query relational operator and query value, determining the type of the query field by reference to its name, relating the query field to a corresponding database property according to type and locating CCG-data database property values in the database property which return a true value when tested against the query value using the query relational operator. Finally, the URL references or the sets of property values associated with the so located CCG-data database property values are extracted.

Database queries are usually expressed in a query language in the form of a phrase or sentence. In query by example style enquiry systems, the user types values into input fields on a form and a program extracts the input values and uses the values to automatically compose a query phrase or sentence. There are many existing examples of query languages used in connection with databases. Generally, they consist of relational expressions (eg Field=Value), logical expressions and grouping of relational and logical expressions by means such as parentheses. They may also contain sorting and output formatting expressions. Often abbreviated notation is used in the expressions such as leaving out field names or relational operators which are then inferred from the value in the expression or implied by default. In an enquiry the nature and format of the output may also be implied, such as a list of URLs of web pages or a list of contact details. Whatever is the mechanism of any particular database, the query expression needs to be parsed and fields in the query expression, explicit, default, implied or inferred, need be related to database properties of similar type. In some styles of database enquiry the query expression is evaluated against each row of a table or record of a file to find rows or records (ie a set of associated property values) which match the query expression. In other styles, sub-sets of the values of the properties are selected according to the interpretation of relational expressions in the query expression and the sub-sets are combined according to logical and grouping expressions in the query to find the sets of associated property values which match the query expression. Often, to make logical operations which combine the selected sub-sets more efficient, it is not the values which are selected but pointers to the values (eg Table name and table row) or unique keys (eg URLs or pointers to URLs) associated with the values. For example, the AND logical operator is often used to combine two lists so that only values or pointers or keys common to both lists are found in the combined list. Usually, the query produces a result list which is then provided to other processes. For example, a list of URLs of web pages is processed to produce an attractively formatted HTML encoded document containing the URLs and is sent to a web browser to allow an enquirer to retrieve interesting web pages. In another example, the contact details associated in the database with each value or pointer in the result list are retrieved from the database and presented as a report in the form of an HTML encoded document and is sent to a web browser for viewing.

From another aspect, the invention comprises a method of displaying CCG-data contained in CCG phrases within web pages which are displayed by a web browser executing on a digital computer. While a web page is loading or has loaded in a web browser, the web browser parses the web page and displays the text (or data) of the web page on a display device connected to the computer. When the web browser parser encounters CCG phrases, the web browser may display the CCG-data (element and/or attribute names (or translations of element and/or attribute names) and/or values) in a number of browser specific ways. For example, the web browser may by default not display any CCG-data, display all CCG-data, not display any CCG-data until a CCG display control attribute explicitly states that subsequent data should be displayed or display all CCG-data until a CCG display control attribute explicitly states that subsequent data should not be displayed. The web browser may also use CGA display controls specifying the size, font, position and so forth to alter the display of the CCG-data.

DESCRIPTION OF EXAMPLES

Having indicated the nature of the present invention, examples or embodiments thereof will now be described by way of illustration only.

Example 1: HTML Syntax Suitable for Representing a CCG Phrase

The following is an example of HTML element syntax suitable for representing CCG phrases in which a control (e.g. "SHOW") may be "good until countermanded" and thus apply to more than one field:

```
<CCG HREF="url"
{{NAME="label"     |     ID="identifier_code"}     &|     {LANG="language_code"     &
CLASS="Class_name"}
{
      {SET_SEPARATOR} &|
      {INDEX | NOINDEX} &|
      {SHOW | HIDE} &|
      {XPOS="horizontal_position_number"} &|
      {YPOS="vertical_position_number"} &|
      {NEWLINE} &|
      {ALIGN=centre | left | right | justify} &|
      {SIZE=[+/-]1 | 2 | 3 | 4 | 5 | 6 | 7} &|
      {COLOR="#rrggbb" | "colour_name"} &|
      {FACE="type_face_name"} &|
      {BLINK &| BOLD &| UNDERLINE &| ITALIC &| STRIKE} &|
      {SUBSCRIPT | SUPERSCRIPT} &|
      {CLEAR{=left | right | all}}
      {NORMAL} &|
      {{{CONTACT &| COPYRIGHT &| DEVELOPER} &|
      {PERSONAL &| BUSINESS &| ASSOCIATION} &|
      {attribute_name="attribute_value(s)"}
}
...
>
```

where: the ellipsis "..." implies optional repetition of the braced ("{" "}") items; the braces are used to group items and are not CCG syntactic elements; "&" (and) implies items must occur together; "|" (or) implies only one item must occur; and "&|" (and/or) implies any including none of the items may appear together.

Using the syntax of this example, each CCG phrase is represented as an HTML element, the element name being "CCG" and the CCG-data (eg attribute_name="attribute_value") and CCG controls (eg SIZE=+1) are represented as attributes of the HTML element. Some of the attributes (eg SIZE) having explicit values (eg +1) and some attributes have implied values depending on the presence or absence in a CCG phrase (eg when the attribute BUSINESS is present it has the implied value of True and the implied value of False when absent).

Representation in XML syntax requires, at most, only a simple translation. All the items, such as "NORMAL" and "attribute_name" may remain unchanged as attributes of the element named "CCG" (eg <CCG size=+1/>). However, when a CCG phrase is encoded in XML, it is preferred that the items are represented as XML elements. For example attribute "SIZE=+1" can be represented as element "<size>+1</size>" or "<size value=+1/>" and "NORMAL" can be represented as "<normal/>.

In this example, the attributes, ID, LANG and CLASS take their meanings from HTML 3.0. The "url" in HREF="url" or may be a link with or without destination anchor labels. For example the URL http://www.w3.org/docs.html does not contain a destination anchor label (or identifier) while http://www.w3.org/docs.html#searching does contain the destination anchor label "#searching" which is intended refer to an anchor in docs.html such as <A NAME="searching">...</A>. There is some confusion in various HTML standards documentation about the distinction between the expression NAME="label" and the expression ID="identifier_code". For most practical purposes the two expressions have the same function or meaning: to uniquely identify within a document a position in or portion of that document.

Database control attributes:
"Set_separator" indicates the end of association between preceding and following data other than through the weaker mutual association with the same CCG phrase or web page; the data are divided into sets. "Index | Noindex" indicates that the following data are / are not to be indexed by a web crawler. These attributes have an implied attribute value of 'True' if present in and 'False' when absent from a CCG phrase.

Display control attributes:
"Show | Hide" indicates that a browser should show / not show the following data. Xpos and Ypos indicate the position (for example in pixel or physical units) on the browser screen where the data is to be displayed. "Newline" may be used in addition or as an alternative method of placing text on a browser screen. "Align" indicates the positioning of data on a browser screen relative to the cursor position set by "Xpos", "Ypos" or "Newline". "Size", "Colour" and "Face" indicates the size, colour and type face or font of the following data when displayed on an browser screen. "Blink", "Bold", "Underline", "Italic", "Strike", "Superscript" and "Subscript" indicates that the following data should be displayed blinking, bold, underlined, italicised, struck through, superscripted or subscripted. "Clear" indicates that the browser screen in the region where data will be displayed should be cleared to background before displaying the following data. "Normal" indicates the data is to be displayed without the "Blink", ..., "Clear" characteristics. The display controls which consist of an attribute name without an explicit value have an implied value of 'True' when present and 'False' when absent.

CCG-data attributes:
"Contact &| Copyright &| Developer" indicates that the following CCG-data refers to details for a person or organisation and/or to the copyright owner and/or to the HTML or web page developer. "Personal &| Business &| Association" indicates that the following data refers to details for a person and/or business and/or association. The previous CCG-data attributes have an implied attribute value of 'True' if present in a CCG phrase or set and 'False' when

absent from a CCG phrase or set. The attribute_name could be standard CCG attribute names or synonyms of standard CCG attribute names or abbreviations of CCG attribute names which refer to the following types of CCG attribute values where square brackets "[" and "]" surround suggested attribute names:

- industry or service or product or subject classifications and sub-classifications:
    - classification name [CN],
    - classification codes [CC].
- display only text [TEXT].
- contact:
  - person:
    - courtesy title [PNC],
    - first given name [PNG],
    - other given names [PNO],
    - family name [PNF],
    - name suffix [PNS],
    - qualifications [PQ],
    - associations [PA],
    - contact person title [PT],
    - contact person role [PR].
  - organisation:
    - name [ON],
    - unit [OU],
    - identifier [OID].
  - physical or post or delivery address:
    - type [AT] (= "PHYSICAL" &| "POST-OFFICE" &| "POSTAL" &| "DELIVERY")
    - post office box number [AP#]
    - post office name [APN]
    - room or suite or office or unit or flat or apartment name &| number [AB#],
    - floor name &| number [ABF],
    - building name [ABN],
    - lane or street or road or highway number [AS#],
    - lane or street or road or highway name [ASN],
    - suburb or town or city name [ACN],
    - region or state or territory or province name [ARN],
    - post code [APC],
    - country or nation name [ANN],
  - telephone:
    - type [TT] (= "PREFERRED" &| "VOICE" &| "MOBILE" &| "CAR" &| "MESSAGE" &|"PAGER" &| "FACSIMILE" &| "MODEM" &| "ISDN" &| "VIDEO")
    - nation or country code number [TC#],
    - trunk access number [TT#],
    - area code number [TA#],
    - local number [TL#],
  - email:
    - type [ET] (= "INTERNET" | {other}),
    - mailer [EM],
    - address [EA],
  - Internet address:
    - url [IURL].
  - date & time:

    - date & time from [DTF],
    - date & time to [DTT],
    - weekday from [DTWF],
    - weekday to [DTWT],
    - weekday time from [DTWFT],
    - weekday time to [DTWTT],
    - time zone [DTZ].
  - brand name [BN].
  - public key:
    - key type[KT].
    - key [K],
- geographical:
  - location units [GLU],
  - location [GL],
  - serviced region units [GLRU].
  - serviced region [GLR],

Suggested attribute name [CN] is the name of an attribute associated with the attribute value containing "classification name" type data. For example, the [CN] attribute value could be the name of a proprietary or national or international or other industry classification standard such as the Australian and New Zealand Standard Industry Classification or "ANZSIC" for short or the U.S. Bureau of the Census Industrial Classifications (USBCIC). The associated classification codes [CC] attribute value could contain the codes and/or descriptions of the codes of the named standard with or without modifications, deletions or extensions. For example: CN="ANZSIC" CC="61;Road transport" or CN="USBCIC" CC="581;Hardware store". Service classifications such as the International Standard Classification of Occupations could be used. For example: CN="ISCOO" CC="4430;Auctioneer" Product classifications such as the Harmonised Commodity Description And Coding System could be used. For example: CN="HSC" CC="8411;Turbojets, turbo-propellers & other gas turbines; parts thereof" For subject classifications, Dewey Decimal, and/or Universal Decimal and/or Library of Congress and/or Bliss and/or Colon Classification could be used. For example: CN="DDC" CC="577.699;Sea shore ecology" The inclusion of subject classifications provides a very simple, straightforward method of classifying the subject matter of an HTML document which could be attractive to commercially oriented copyright owners.

The text ([TEXT]), person ([PNC] - [PR]), organisation ([ON] - [OID]), physical or post or delivery address ([AT] - [ANN]), telephone ([TT] - [TL#]), email address ([ET] - [EA]) and Internet address [IURL] are intended to be associated with each other in the obvious manner. Date & time(s) ([DTF] - [DTZ]) are intended to indicate the times at which the address and/or telephone and/or email will be serviced by the associated person(s) and/or organisation(s). The brand name ([BN]) attribute is intended to hold commercial brand names. Public key ([KT] - [K]) is intended to hold public encryption keys for secure communication with the contact person or organisation.

The geographical location [GL] could be a latitude and longitude (eg E148D31'12.5",S36D40',09.6" or E148.5201,S36.6693 or -148.5201,-36.6693), or a Universal Grid Reference (eg 55FV364402) or other global, national, regional or local location reference with units as specified [GLU], which is typed in or obtained by pointing to a digitally encoded map or other methods. In more populated regions of some countries such as the U.S., street addresses and post codes are associated with a moderately accurate geographic location and can be used to interpolate geographic location data where geographic location data is not explicitly stated in the CCG-data. Using a universally recognised code such as latitude and longitude has advantages when used with international mediums like the Internet. Geographical location is intended to be associated with a post, delivery address or physical address such as place of business or residence. A CCG compliant browser could use this reference to display a map centred on that geographic location. The purpose of the geographical location data is to allow browser users to specify search engine search criteria which will result in the search engine selecting only those Internet accessible documents which provide details about providers which are within a specified region. The serviced region [GLR] is intended to indicate the preferred area of operation of providers expressed in terms of serviced region units [GLRU]. A radial distance (eg in kilometres) or alternate means of expressing an area of interest around a geographic point, such as polygons, are envisaged.

It is envisaged that the CCG attribute_value could be composed of more than one value (actually sub-value) wherein specific characters or character strings separate individual values.

While specific instances of element names and types have been given in this example, of more importance is the type of data and type controls over the display and indexing of the data. As an alternative to the preferred immediately following example where the CCG-data is lumped together under the HTML element named "CCG", certain elements of the data, for example the classification data, could be lumped under separate HTML elements with distinctly different names thereby separating CCG classification data from CCG contact data. However, this is not preferred because the strength of association between the two types of data is weakened.

<u>Example 2: Classification of Portion of a Web Page.</u>

Where it is desired to classify a portion of a web page, such as a paragraph about a product, simple CCG-data may be used in conjunction with the syntax of Example1. For example:

```
<A NAME="Radios">AM-FM radio receivers: </A>
    <CCG HREF="#Radios">
        CN="ANZSIC"
        CC="E23.34.78;Electrical equipment - radio receivers AM"
        CC="E23.34.79;Electrical equipment - radio receivers FM"
    </CCG>
We won't be beaten on the price of these high quality receivers ....
```

In this example, the CCG prase appears after the related anchor (<A NAME=...</A>). However, while such proximity visually provides an obvious association between the anchor and related CCG phrase, it is intended that CCG phrase containing the attribute HREF related to a specific anchor could appear anywhere within the body of a web page and remain related to the named anchor. The CCG phrase containing the attribute HREF could appear in a separate document and thereby relate the CCG-data to the entire document or to a named anchor although, as previously noted, coordinating separate documents can be problematic. In the absence of the HREF and NAME attributes, it is also intended that the CCG-data apply to the whole web page.

<u>Example 3 Classification of Portion of a Web Page using XML Syntax</u>

Using XML syntax and similar attribute names to those of Example 2 the HTML fragment of Example 2 may be rewritten as:

```
<A NAME="Radios">AM-FM radio receivers: </A>
    <XML>
```

<CCG>

<HREF>"#Radios"</HREF>

<CN>"ANZSIC"</CN>

<CC>"E23.34.78;Electrical equipment - radio receivers AM"</CC>

<CC>"E23.34.79;Electrical equipment - radio receivers FM"</CC>

</CCG>

</XML>

We won't be beaten on the price of these high quality receivers ....

This example demonstrates that the translation of CCG-data from HTML to XML (and the reverse) involves simple syntactical and grammatical translations. Of course, the resulting HTML and XML, while "well formed" might not be recognised or, if recognised, might not be understood by some parsers.

<u>Example 4: Constructing a Web Page Containing CCG-data</u>

As an example, a web page developer, Alice Jamieson, is preparing an advertisement for a local electrician John Williams, trading as Kelso Electrical, who wants to advertise on the web for business within 30 kilometres from his office located at 18 Raglan Street, Kelso, New South Wales. Alice uses a graphical user interface web page authoring tool capable of creating and modifying web pages containing HTML (and XML) CCG phrases by accepting inputs from a user. The tool executes on a digital computer having input devices such as a keyboard, mouse, light pen and touch pad, display devices such as a CRT, LED arrays, liquid crystal arrays and computer-readable media such as magnetic and optical disks, memory arrays, magnetic tape and the like.

The authoring tool also embodies knowledge of the content and structure of CCG phrases such as the attribute names, valid ranges and sets of associated attribute values, the normal order of the attributes in the CCG phrase and interdependencies between attribute values. The tool provides a window where web pages may be viewed in layout (browser) mode and another window where the HTML code may be viewed in editing mode. The tool also provides means of inserting, deleting, modifying and organising HTML elements, changing font size, face and colour and so forth. The tool provides means for the user to build CCG phrases by using input devices to select an edit control representing various types of CCG attributes from a list which the tool then inserts in the body of a web page together with, when not already present, HTML code indicative of the start and end of a CCG phrase. The user then types in the value in the attribute. Similarly, the tool provides means of converting web page text to CCG attributes. Using input devices, the user selects the text to be converted to a CCG attribute then selects an edit control from a list; the tool then inserts the HTML code necessary to encode the text as a CCG attribute. However, these semi-manual methods of creating and modifying CCG phrases are inefficient and error prone. The tool also provides a button, which can be activated by using input devices, for access to CCG phrase editing functions. The CCG editing functions consist of a means of extracting the CCG values from existing CCG phrases in the web page being edited, forms for entering and modifying the extracted CCG values, a layout view browser window for altering how the CCG-data displays (position, font size, face, colour, bold, normal, hiding or showing and so forth), a data view browser window to alter which CCG-data values are to be indexed or not indexed in search engine databases, and a means of deleting existing CCG phrases from web pages and inserting new or changed CCG phrases in web pages. Editing cursors marking the current location at which text and/or data may be inserted, deleted or modified are provided in each window and form.

In the current example, the web page initially contains no CCG phrase. Clicking the CCG editing function button of the authoring tool causes a form to appear. The form contains prompts related to CCG attribute names and associated data input fields related to the CCG attribute values associated with the CCG attribute names, that is CCG-data. The fields are blank because, in the web page layout view, the edit cursor is not over a CCG phrase (and can not be since the web page initially contains no CCG phrase). The service classifications relevant to the web age, John Williams physical business contact address, phone and fax numbers, email address and geographic location and his post office business contact addresses are entered into the forms using a keyboard and mouse. The developer, Alice Jamieson, also includes her basic contact details where provided for on the form. The forms use drop down lists to select address blocks (eg physical and post office) for editing. Logic associated with the forms validates the CCG attribute values and interdependencies. Input devices are then used to control the CCG-data layout view browser to modify the appearance of the CCG-data such as font size and colour and positioning. In the layout browser, input devices communicating with the edit cursor are used to highlight individual items and blocks of items to be changed. The post office address is highlighted as a block and moved into position in line with the physical address. The CCG-data view window is then used to check which data items are to be indexed by search engines. In this example all CCG-data (ie all CCG attribute values except display control values and database control values) are to be indexed. Input devices are used to control the edit cursor to highlight the entire data and a mouse is used to click (activate) a button to mark all the data for indexing. Then another button is clicked which builds an HML encoded CCG phrase of CCG attributes derived from the CCG-data values, display control values and database control values and inserts the CCG phrase in the web page at the location pointed to in the web page layout browser window.

The HTML code editing mode window was called up which revealed the following HTML encoded CCG phrase in the web page:

```
<XML>
<CCG>
    <INDEX/>
    <HIDE/>
    <CN>ANZSIC</CN>
    <CC>D36.11.45;Electrical contractors - residential</CC>
    <CC>D36.11.46;Electrical contractors - industrial</CC>
    <SHOW/>
    <CONTACT/> <COPYRIGHT/>
    <BUSINESS/>
    <XPOS>50</XPOS>
    <YPOS>320</YPOS>
    <ALIGN>centre</ALIGN>
    <SIZE>3</SIZE>
    <COLOR>black</COLOR>
    <FACE>Times New Roman</FACE>
    <BOLD/>
    <CLEAR>all</CLEAR>
    <TEXT>Contact :</TEXT>
    <PNC>Mr</PNC>
    <PNG>John</PNG>
    <PNF>Williams</PNF>
    <PQ>AIE</PQ>
```

```
<PA>ARUC</PA>
<NEWLINE/>
<PT>Managing Director</PT>
<NEWLINE/>
<ON>Kelso Electrical Pty. Ltd.</ON>
<NEWLINE/>
<NORMAL/> <ITALIC/>
<SIZE>-2</SIZE>
<TEXT>NSW License 45678C</TEXT>
<NEWLINE/>
<NORMAL/> <BOLD/>
<SIZE>+2</SIZE>
<AT>PHYSICAL</AT>
<AS#>18<AS#>
<ASN>Raglan Street<ASN>
<NEWLINE/>
<ACN>Kelso</CAN>
<NEWLINE/>
<ARN>NSW<ARN>
<NEWLINE/>
<HIDE/>
<ANN>Australia</ANN>
<NEWLINE/>
<SHOW/>
<TEXT>Phone:</TEXT>
<TT>PREFERRED ; VOICE ; MESSAGE</TT>
<HIDE/>
<TC#>61</TC>
<SHOW/>
<TT#>0</TT#>
<TA#>63</TA#>
<TL#>456-7828</TL#>
<TEXT>   Fax:</TEXT>
<TT>FACSIMILE</TT>
<HIDE/>
<TC#>61</TC#>
<SHOW/>
<TT#>0</TT#>
<TA#>63</TA#>
<TL#>456-7829</TL#>
<NEWLINE/>
<ET>INTERNET</ET>
<EA>johnw@firefly.com.au<EA>
<TEXT>   </TEXT>
<GLU>LatLong</GLU>
<GL>="33.3978S;148.5679E</GL>
<GLRU>Km</GLRU>
<GLR>30 </GLR>
<SET_SEPARATOR/>
<XPOS>250</XPOS>
```

```
<YPOS>320</YPOS>
<NEWLINE/>
<NEWLINE/>
<TEXT>Or write to us at :</TEXT>
<NEWLINE/>
<ON>Kelso Electrical Pty. Ltd.</ON>
<NEWLINE/>
<AT>POST-OFFICE</AT>
<AP#>P.O. Box 187</AP#>
<NEWLINE/>
<APN>Sunny Corner</APN>
<TEXT> </TEXT>
<APC>2795</APC>
<NEWLINE/>
<HIDE/>
<ANN>Australia</ANN>
<SET_SEPARATOR/>
<HIDE/>
<DEVELOPER/>
<BUSINESS/>
<PNG>Alice</PNG>
<PNF>Jamieson</PNF>
<ET>INTERNET</ET>
<EA>alijam@firefly.com.au</EA>
<IURL>http://www.firefly.com.au/~aljam/<IURL>
</CCG>
</XML>
```


In the web page layout browser window the CCG-data displayed as follows:

Contact : | Or write to us at:
Mr John Williams, AIE, ARUC,
Managing Director
Kelso Electrical Pty. Ltd. | Kelso Electrical Pty Ltd
NSW License 45678C | P.O. Box 187
18 Raglan Street | Sunny Corner 2795
Kelso
NSW
Phone: 063-456-7828 **Fax:** 063-456-7829
Email: johnw@firefly.com.au Map

Having encoded the web page in this way, Alice then posts it on the storage device of a digital computer connected to the Internet from where it can be retrieved through the Internet using the URL "http://www.firefly.com.au/~johnw/index.html"

Example 4: Constructing a Database from Web Pages Containing CCG-data

During a routine sweep of Internet connected web page servers, a web crawler (or robot) operating on a server named "ccg.search.com" executing on an Internet connected digital computer discovers the URL "http://www.firefly.com.au/~johnw/index.html" in a document it had previously retrieved through the Internet. The web crawler decides that the URL matches it's selection criteria because the URL contains the suffix ".html". The web crawler then successfully retrieves the document by extracting from the URL the address of the computer hosting the document, addressing and sending a message (including the address of the web crawler) requesting the web page through the network to the web page host computer using TCP/IP protocol, the host computer then reads the document, addresses and sends the document to the web crawler using TCP/IP protocol, the web crawler then waiting until it has received all parts of the web page from the host computer before proceeding. It inspects the contents of the document and finds that it matches the additional selection criteria that it is an HTML encoded document. The web crawler program, depending on its state and logic, then parses the document, strips out and saves some or all of the URLs in the document for future examination. The web crawler program then passes the document, together with the URL of the document through a network communications channel to an indexing program executing on a different computer. The indexing computer has database updating software which manipulates a database stored on computer-readable media.

The indexing program parses the document, from first to last character, indexing some of the meta data in the <head> of the document and the words in the text of the document with respect to the document URL. In the database of this example, unique words extracted from the documents already indexed are held in separate rows of a column of a database table and in another column of the same table on each row is an associated pointer to the first bucket or block of URLs of documents containing the word associated with the pointer. As new words are found, the new word is added as a new row in the word column of the table, a new bucket is created, the URL of the document containing the new word is inserted into the bucket and a pointer to the new bucket is written in the new row pointer column. When the same word is found in another document, the row in the table of the word is found, the pointer is retrieved from the table, the bucket pointed to by the pointer is retrieved and the URL of the other document is inserted in the bucket. Where a bucket becomes full of URLs, a new bucket is created and a pointer to the new bucket for holding additional URLs is placed in the full bucket. Deletion of words and URLs of changed or no longer existing documents is also provided for.

In addition to indexing words extracted from the text of the document, the indexing program also indexes the CCG-data in the document as well as indexing words found in the CCG-data. When the parser finds HTML element "<XML>" in the document it switches into XML parsing mode and switches out of that mode when "</XML> is found. When the element "<CCG>" is found, the parser switches into the CCG parsing mode and switches out of that mode when "</CCG>" is found.

The example database has a CCG-data attribute name to database property name correspondence table to show the relationship between the CCG-data attribute names and the database tables and columns (properties) where the CCG-data attribute values are to be stored in the database as database property values. The database property values and associated URLs are stored in much the same way as for words extracted from text as outlined above. However, CCG contact data, for example, which consists of several distinct CCG-data attributes which are related (eg street name, city), is stored in a database table having a column (property) related to each distinct CCG contact attribute name and each separate CCG contact data set (eg person's name, address, telephone number) as separated by "<CCG>", "<SET_SEPARATOR>" and "</CCG>" is held in a separate row in the table. The values stored in each row are considered to be a set of associated property values of different types.

The indexing program, during parsing the document of Example 2 above, encounters the "<CCG>" element and enters the CCG parsing mode. The parser knows to ignore display control attributes and to consider database control elements in the CCG phrase. The example indexing program opts to index all other CCG-data contained in the attribute values until explicitly instructed not to index the attribute values by encountering the "<NOINDEX/>" database control element and then to recommence indexing when the "<INDEX/>" database control element is encountered.

Taking each CCG-data attribute name and associated attribute value(s) in succession, the example indexing program uses the correspondence table to translate the CCG-data attribute name to the database table and column (property) names where the CCG-data attribute value(s) are to be stored as database property value(s). The indexing program may opt to translate the CCG-data attribute values to database property values by, for example, converting character strings of digits to binary encoded decimal representation, the string "True" to a single bit representation and the like. The indexing program then adds or updates the database property value(s), using the database table and column (property) names (or similar references) obtained by translation, in much the same manner as outlined above for the update of the database using words extracted from the document text, including associating the data to the document URL where desired. Where the CCG-data contains a "HREF" attribute (or similar), the URL associated with the other CCG-data is a URL taken from the "HREF" attribute value or composed of the document URL and the "HREF" attribute value if the attribute value is a partial or relative URL. Some CCG attributes, such as "<BUSINESS/> have only an implied value of true if the attribute is present and false if the attribute is absent, the "<SET_SEPARATOR/>", "<CCG>" and "</CCG>" resetting such values to false. However, where attribute value(s) associated with different attribute names are still related, such as a person's name and a street name, the related values of different types are stored on the same row of the same database table but in a different column (database property) to preserve the relationship. "<SET_SEPARATOR/>" limits the degree of relatedness between, for example, a person's name occurring before the separator and a street name occurring after the separator. Using the example document and using the same database column (property) names as used for the CCG-data attribute names a portion of the table constructed database table would look like:

	PNC	PNG	PNF	PQ	PA	PT		URL
...	...	...	...	...	...	...	...	...
...	Mr	John	Williams	AIE	ARUC	Managing Director	...	(pointer)
...	...	...	...	...	...	...	...	...

Difficulties not highlighted by this example are the need to handle properties having multiple values of the same type, "sparse rows" where only a few values are not null (blank) and tables with extremely large numbers of rows. For example, the CCG-data of this example could have contained multiple values of personal qualifications ("PQ"). To represent this type of data using a 2 dimensional table database system, the database would be "normalised" so that the multiple values were stored in a separate table and keys or pointers were used to relate the relate the items in the two tables. Numerous alternate database systems, for example those based on key hashing and data buckets, or tagging data values with prefixes or suffixes related to the type of data value may be used. Preferably, however, whatever database system is used, it should preserve the associations of CCG-data items present in the CCG phrases.

Because the geographic location data was missing from the postal address of the CCG-data in the example document, but a post code was present, the indexing program inferred the geographic location from the post code.

Example 6: Finding Web Page References Using a CCG Database

As an example, Kevin Robson lives in Sydney but owns and has rented out a house in Bathurst. He wants to use the web to find some electricians based in the general Bathurst region (not only in Bathurst City) to contact for estimating the cost of modifying the wiring in the house. He uses his web browser to open the web page "http://www.ausline.com.au/web_search.html" containing AusLine's search engine web page search criteria input form encoded using the HTML "<form>" element.

The search criteria input form contains several input fields including those labelled "Service classification", "Key words", "City./Suburb/Town", "Country", "Lat/Long" and "Radius". The form also displays a button labelled "Map" to allow latitude and longitude to be selected by pointing to map images. The word "electrician" is typed into the "Service classification" field, "house wiring" into the "Keywords" field, "Bathurst" into the "City/Suburb/Town" field and "10" into the field "Radius". The country "Australia" was already showing in the country field because the web page server had received cookie data from the browser indicating that that was the country used when the browser last used the web page. The "submit search" button on the web page was clicked. The browser transmitted a message using TCP/IP protocol to the AusLine server containing the input field values encoded in the header of the message.

After a short delay, the search result HTML encoded web page was returned. Clicking on the "Service classification" input field drop down list box to check the classifications used in the search revealed three items:

- Electrical contractors - residential
- Electrical contractors - industrial
- Electrical engineers

The search engine attached to the server obtained those classifications by using word stemming and searching the text of the service classifications held in it's database. The Lat/Long field contained the value "33.3856S;148.5743E" which the search engine obtained by looking up the latitude and longitude of the town "Bathurst" in the country "Australia" in it's database. Clicking on the "Map" button retrieved a web page having the image of a map centred on the town of Bathurst and showing the area 20 Km around it. The search engine obtained the map by making a request to another Internet connected server and supplying the latitude, longitude and radius. Clicking on the browser "Back" button returned to the search results page.

The search results contained 8 titles, brief descriptions and URLs including a reference containing the URL "http://www.firefly.com.au/~johnw/index.html". Retrieving each in turn revealed that all were well focused according to the search criteria being related to electricians, electrical contractors and engineers in the Bathurst area. The search engine obtained these references to web pages by:

- searching it's database of service classification titles with words stemming from "electrician" which resulted in three service classification codes,
- searching it's database using the three service classification codes to obtain an intermediate list of URLs of web pages containing those CCG codes

- searching it's database for the two keywords to obtain an intermediate list of URLs of web pages containing those words in the web page text,
- Searching it's database to find the latitude and longitude of Bathurst, Australia,
- searching it's database to obtain an intermediate list of web pages which contain latitude and longitude data lying within 10 Km of the latitude and longitude of Bathurst, Australia,
- producing as a result list, a list of URLs which are common to all the intermediate lists,
- obtaining from it's database the title and brief description of the web pages,
- formatting the titles, descriptions and URLs into an HTML encoded report,
- transmitting the report to the enquiring web browser.

Example 7: Finding Contact Details Using a CCG Database

As an example, Jim Jones of Jones and Sons wants to send a recall notice about a faulty batch of UV stabilised electrical power cable to all Electrical contractors and Electrical wholesalers in Australia who have email addresses. He uses his web browser to open the web page "http://www.ausline.com.au/contact_search.html" containing AusLine's search engine contact search criteria input form encoded using the HTML "<form>" element.

The search criteria input form contains several input fields including those labelled "Service classification", "Country" and "Output format". The word "electric*" is typed into the "Service classification" field, the word "Australia" is typed into the "Country" field and the "Tabular - Name & Email" option in the "Output format" drop down list box is selected. The "Submit search" button on the web page is clicked. The browser transmits a message using TCP/IP protocol to the AusLine server containing the input field values encoded in the header of the message.

After a short delay, the search result HTML encoded web page is returned. Clicking on the "Service classification" input field drop down list box to check the classifications used in the search revealed too many classifications for the result to be sufficiently focused. The following four classifications were selected from the list:

- Electric cable - ducting systems
- Electrical contractors - residential
- Electrical contractors - industrial
- Electrical wholesalers

and the "Submit search" button is pressed again to refine the search.

The search results contained 3,473 names and associated email addresses and URLs to full contact details. Jim saved the search result page on his computer so that he could use his email program to send the recall notice to each email address in the list. The email address "johnw@firefly.com.au" was included in the list.

The search engine obtained these references to web pages by:

- searching it's database using the four service classification titles which resulted in four service classification codes,
- searching it's database using the four service classification codes to obtain an intermediate list of database primary keys of database table rows containing those service classification codes in the database Service classification attribute,
- searching it's database using the country name "Australia" to obtain an intermediate list of database primary keys of database table rows containing that word in the database Country attribute,

- producing as a result list, a list of database primary keys which are common to both the intermediate lists,
- obtaining from it's database using the result list the values of the name and email attributes,
- using the HTML <table> element to format the name values, email values and full detail URLs into an HTML encoded report,
- transmitting the report to the enquiring web browser.

This example relates to finding sets of associated database contact values without requiring references to web pages. However, finding other sets of associated database values such as sets of associated industry classification values and geographic location values might also be useful for some purposes.

Thus it is appreciated that the afore stated goals, advantages and objectives are achieved by the teachings herein. In particular it is seen that, unlike the prior art, efficiently searchable Yellow pages and White pages databases and the like may be automatically constructed from HTML encoded web pages. Additionally the database entries may be automatically linked to specific web pages and portions of web pages allowing convenient methods of indexing of product and service catalogues and the like. It is also appreciated that simpler methods of constructing databases suited to a variety of other uses such as industry and subject directories are also provided.

From the foregoing teachings and with the knowledge of those skilled in the art, it is apparent that other modifications and adaptations of the invention will become apparent. For example, the method steps disclosed and claimed herein may be practiced in a variety of different orders. CCG-data may take on a variety of different forms within the meaning of the claims. Thus, it is our intention to include within the scope of the claims not only the invention literally embraced by the language of the claims but to include all such modifications and adaptations which may come to those skilled in the art.

What I claim is:

1. An HTML encoded web page embodied on a computer-readable medium, said web page comprising at least one HTML encoded CCG phrase, each CCG phrase comprising:
   a) HTML code indicative of the start of a CCG phrase,
   b) at least one CCG-data attribute, and
   c) HTML code indicative of the end of a CCG phrase.

2. An HTML encoded web page embodied on a computer-readable medium, said web page comprising at least one HTML encoded CCG phrase, each CCG phrase comprising:
   a) HTML code indicative of the start of a CCG phrase,
   b) at least two CCG-data attributes,
   c) at least one database control attribute separating said CCG-data attributes into at least two sets of CCG attributes, and
   d) HTML code indicative of the end of a CCG phrase.

3. An HTML encoded web page embodied on a computer-readable medium, said web page comprising at least one HTML encoded CCG phrase, each CCG phrase comprising:
   a) HTML code indicative of the start of a CCG phrase,
   b) at least one CCG-data attributes,
   c) at least one attribute of: database control attributes, display control attributes; and
   d) HTML code indicative of the end of a CCG phrase.

4. A computer implemented method of building a web page comprising at least one HTML encoded CCG phrase, the method comprising the steps of:
   a) displaying a web page on a computer display device,
   b) displaying an edit cursor indicating a character position on said display device and a corresponding character position in said web page, said edit cursor being positionable within the display of said web page by use of computer input devices,
   c) separately displaying on said computer display device a set of edit controls representing CCG-data attribute types,
   d) positioning said edit cursor within said display of said web page using said input devices,
   e) selecting an edit control from said set of edit controls using said input devices,
   f) relating said selected edit control to a corresponding CCG-data attribute name,
   g) constructing a CCG-data attribute character string comprising a character string representing said attribute name and another character string representing an empty CCG-data value,
   h) if the said edit cursor is positioned outside a CCG phrase,
      i) inserting into said web page, at the character position indicated by said edit cursor, a start character string comprising HTML code indicative of the start of a CCG phrase,
      ii) inserting into said web page, immediately after the end of said start character string, an end character string comprising HTML code indicative of the end of a CCG phrase, and
      iii) positioning said edit cursor between said start and end character strings,

i) inserting said CCG-data attribute character string into said web page at the character position indicated by said edit cursor,

j) positioning said edit cursor at the character position in said web page of the CCG-data value of said inserted CCG-data attribute character string,

k) inputting characters using a keyboard,

l) inserting said input characters into said web page at the character position indicated by said edit cursor, thereby converting said empty CCG-data value to a non-empty CCG-data value, and

m) writing said web page on computer-readable media.

5. A computer implemented method of building a web page comprising at least one HTML encoded CCG phrase, the method comprising the steps of:

a) displaying a web page on a computer display device,

b) displaying a start edit cursor and an end edit cursor on said display device, each said edit cursors indicating a character position on said display device and a corresponding character position in said web page, said edit cursors being positionable within the display of said web page by use of computer input devices,

c) separately displaying on said computer display device a set of edit controls representing CCG-data attribute types,

d) selecting a string of web page characters on said display device using said input devices to position said start edit cursor to indicate the start said string of web page characters and said end edit cursor to indicate the end of said string of web page characters,

e) selecting an edit control from said set of edit controls using said input devices,

f) relating said selected CCG-data control to a corresponding CCG-data attribute name,

g) constructing a CCG-data attribute character string comprising a character string representing said attribute name and another character string representing a CCG-data value containing said string of web page characters,

h) deleting said string of web page characters from said wen page,

i) if the said start edit cursor is positioned outside a CCG phrase,

    i) inserting into said web page, at the character position indicated by said start edit cursor, a start character string comprising HTML code indicative of the start of a CCG phrase,

    ii) inserting into said web page, immediately after the end of said start character string, an end character string comprising HTML code indicative of the end of a CCG phrase, and

    iii) positioning said start edit cursor between said start and end character strings,

j) inserting said CCG-data attribute character string into said web page at the character position indicated by said start edit cursor, thereby converting said string of web page characters to a CCG-data attribute value contained within a CCG-data attribute contained within CCG-phrase, and

k) writing said web page on computer-readable media.

6. A computer implemented method of building a web page comprising at least one HTML encoded CCG phrase, the method comprising the steps of:

a) displaying a CCG-data input form on a computer display device,

b) inputting CCG-data values into fields of said data input form using computer input devices,

c) inserting into the body of a web page a start character string comprising HTML code indicative of the start of a CCG phrase,
d) inserting into said web page body immediately after the end of said start character string an end character string comprising HTML code indicative of the end of a CCG phrase,
e) extracting successive field values from said data entry form together with related field value type information,
f) relating the type of each extracted field value to a corresponding CCG-data attribute name,
g) constructing a CCG-data attribute character string comprising a character string representing said attribute name and another character string representing said field value,
h) inserting said CCG-data attribute character string into said web page between said start and end character strings.
i) writing said web page on computer-readable media.

7. A computer implemented method of building a database which comprises sets of associated property values wherein each set includes at least two property values of different types, the property values being any of classification values, contact values, geographic location values, hereinafter collectively referred to as CCG-data, the method comprising the steps of:
a) retrieving successive web pages from a computer network, each web page being identified by a URL,
b) searching each web page for a CCG phrase that includes a plurality of different types of CCG-data attributes,
c) extracting a plurality of said attributes from said phrase,
d) from each extracted attribute, deriving an attribute name and a related attribute value,
e) determining the type of said extracted attribute and said attribute value by reference to said attribute name,
f) relating said type of attribute value so determined to a corresponding type of database property value,
g) relating the URL of said web page to an other type of database property value,
h) writing said derived attribute value to the database property value of said determined corresponding type in a set of associated property values, and
i) writing the URL of said web page to a database property value of said other type in said set of associated property values.

8. A computer implemented method of building a database which comprises sets of associated property values wherein each set includes at least two property values of different types, the property values being any of classification values, contact values, geographic location values, hereinafter collectively referred to as CCG-data, the method comprising the steps of:
a) retrieving successive web pages from a computer network, each web page being identified by a URL,
b) searching each web page for a CCG phrase that includes at least one type of CCG-data attribute,
c) extracting at least one said attribute from said phrase,
d) from each extracted attribute, deriving an attribute name and a related attribute value,

e) determining the type of said extracted attribute and said attribute value by reference to said attribute name,

f) relating said type of attribute value so determined to a corresponding type of database property value,

g) relating the URL of said web page to an other type of database property value,

h) writing said derived attribute value to the database property value of said determined corresponding type in a set of associated property values, and

i) writing the URL of said web page to a database property value of said other type in said set of associated property values.

9. A computer implemented method of building a database which comprises sets of associated property values wherein each set includes at least two property values of different types, the property values being any of classification values, contact values, geographic location values, hereinafter collectively referred to as CCG-data, the method comprising the steps of:

a) retrieving successive web pages from a computer network,

b) searching each web page for a CCG phrase that includes a plurality of different types of CCG-data attributes.

c) extracting a plurality of said attributes from said phrase,

d) from each extracted attribute, deriving an attribute name and a related attribute value,

e) determining the type of said extracted attribute and said attribute value by reference to said attribute name,

f) relating said type of attribute value so determined to a corresponding type of database property value, and

g) writing said derived attribute value to the database property value of said determined corresponding type in a set of associated property values.

10. A computer implemented method of finding references to web pages posted on computer network the method using a database comprising sets of associated property values, the property values being any of classification values, contact values, geographic location values, hereinafter collectively referred to as CCG-data, and URL references, the method comprising the steps of:

a) receiving a query phrase including query relational expressions from a computer network,

b) parsing said query phrase and extracting each of said query relational expressions included therein,

c) from each extracted query relational expression, deriving a query field name,

d) determining the type of said query relational expression by reference to its derived query field name,

e) relating said type of query relational expression so determined to one of the following query relational expression types: CCG-data type, other type,

f) provided said query relational expression is a CCG-data type, deriving a query relational operator and query value related to its query field name from said query relational expression,

g) determining the type of said query value by reference to said query field name,

h) relating said type of query value so determined to a corresponding type of database property value,

i) locating database property values of said determined corresponding type which return a true value when tested against said query value using said query relational operator,

j) extracting from said database a list of the URL references associated with the so located database property values.

11. A computer implemented method of finding sets of associated database property values the method using a database comprising sets of associated property values wherein each set includes at least two property values of different types, the property values being any of classification values, contact values, geographic values, hereinafter collectively referred to as CCG-data, the method comprising the steps of:

a) receiving a query phrase including query relational expressions from a computer network,

b) parsing said query phrase and extracting each of said query relational expressions included therein,

c) from each extracted query relational expression, deriving a query field name,

d) determining the type of said query relational expression by reference to its derived query field name,

e) relating said type of query relational expression so determined to one of the following query relational expression types: CCG-data type, other type,

f) provided said query relational expression is a CCG-data type, deriving a query relational operator and query value related to its query field name from said query relational expression,

g) determining the type of said query value by reference to said query field name,

h) relating said type of query value so determined to a corresponding type of database property value,

i) locating database property values of said determined corresponding type which return a true value when tested against said query value using said query relational operator,

j) extracting from said database sets of associated database property values associated with the so located database property values.

12. A method of displaying a web page comprising at least one HTML encoded CCG phrase, the method comprising the steps of:

a) retrieving a web page from a computer network,

b) parsing said retrieved web page to locate an HTML code indicative of the start of a CCG phrase,

c) parsing said located CCG phrase and extracting successive CCG attributes contained therein until an HTML code indicative of the end of said CCG phrase is found,

d) from each extracted attribute, deriving an attribute name,

e) determining the type of said extracted attribute by reference to its derived attribute name,

f) relating said type of attribute so determined to one of the following attribute types: database control, display control, CCG-data,

g) provided said extracted attribute is not a database control type, deriving an attribute value related to its attribute name from said extracted attribute,

h) determining the type of said attribute value by reference to said attribute name.

i) *relating* said type of attribute value so determined to a corresponding type of parameter of a display-device-control-program,

j) writing said attribute value to said parameter, and

k) where said type of attribute is a CCG-data type, causing said display-device-control-program to effect display of said attribute value on a display device, formatted and positioned according said display-device-control-program parameters whereby successive values of CCG-data of the CCG phrase are displayed.

ABSTRACT

A system for automatically creating databases containing industry, service, product and subject classification data, contact data, geographic location data (CCG-data) and links to web pages from HTML, XML or SGML encoded web pages posted on computer networks such as the Internet or Intranets. The web pages containing HTML, XML or SGML encoded CCG-data, database update controls and web browser display controls are created and modified by using simple text editors, HTML, XML or SGML editors or purpose built editors. The CCG databases may be searched for references (URLs) to web pages by use of enquiries which reference one or more of the items of the CCG-data. Alternatively, enquiries referencing the CCG-data in the databases may supply contact data without web page references. Data duplication and coordination is reduced by including in the web page CCG-data display controls which are used by web browsers to format for display the same data that is used to automatically update the databases.

Electronic Acknowledgement Receipt

EFS ID:	4764965
Application Number:	12238842
International Application Number:	
Confirmation Number:	2317
Title of Invention:	Automated Media Delivery System
First Named Inventor/Applicant Name:	Sean Barger
Customer Number:	22862
Filer:	Michael Glenn/Christine Ortt
Filer Authorized By:	Michael Glenn
Attorney Docket Number:	EQUI0016D
Receipt Date:	10-FEB-2009
Filing Date:	26-SEP-2008
Time Stamp:	13:04:43
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		IDSEQUI0016D.pdf	262480 96b61efc5714fda84fdef3824d5840b88da75e9b	yes	6

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Miscellaneous Incoming Letter	1	1
	Information Disclosure Statement Letter	2	3
	Information Disclosure Statement (IDS) Filed (SB/08)	4	6

Warnings:

Information:

2	NPL Documents	A-Sakaguchi-ABrowsingTool.PDF	494426 1d96d1dc1240a3508d5539ad0bc687b614dae8fd	no	9

Warnings:

Information:

3	NPL Documents	B-Zaiane-MiningMultimediaData.pdf	1370264 564270bf9ee4122477794da53486be356ccdad6d	no	19

Warnings:

Information:

4	NPL Documents	C-Bulterman-ModelsMediaAndMotion.PDF	690629 0345df97b463fa006f0df427272df739d9fad73d	no	12

Warnings:

Information:

5	NPL Documents	D-Mohler-MigratingCourseMaterials.PDF	77628 7f940080689bd2c8d05a89945320fcabae8e32e6	no	2

Warnings:

Information:

6	NPL Documents	E-Dobson-AnimatingYourWebPages.PDF	293322 759a70d5547c34da189436bcd3acf15a66c914a7	no	5

Warnings:

Information:

7	NPL Documents	F-Berinstein-TheBigPicture.PDf	526864 055c120228c026c7ce046cec7039c80a2b8b6968	no	11

Warnings:

Information:

8	NPL Documents	G-McNeil-ResearchInterests.PDF	100387 a9668ca00802930bbdd9f94836d40b2e61c13419	no	3

Warnings:

Information:					
9	NPL Documents	H-GeoSciences-TableOfContents.PDF	45795 c41562a3bd6be90a4d9142499ca7dbeb799677e2	no	2
Warnings:					
Information:					
10	Foreign Reference	35-EP0747842.pdf	1664575 2af9850cc271a2199bf6d59e70568e93d507f6c1	no	27
Warnings:					
Information:					
11	Foreign Reference	36-EP0782085.pdf	791335 2b7c787340dafd993b77b5b55fa1e4343e211d36	no	18
Warnings:					
Information:					
12	Foreign Reference	37-EP0818907.pdf	542895 b690be55020c3094e256e0988aa06af4a7fe681d	no	8
Warnings:					
Information:					
13	Foreign Reference	38-EP0843276.pdf	1854757 eb56a7b43f7c18e6a7d7c9a1e2b128bac6789fe3	no	42
Warnings:					
Information:					
14	Foreign Reference	39-EP0876034.pdf	1026073 26f5619a632d73360d25feb2506f03da590b7130	no	16
Warnings:					
Information:					
15	Foreign Reference	40-EP0883068.pdf	1172032 fa47e50bf1595b8f41ebc3335f55a7d26cb432ef	no	22
Warnings:					
Information:					
16	Foreign Reference	41-EP0886409.pdf	1365504 dbb211b2c33c03334e6171f7d45569c580bcc3b4	no	28
Warnings:					
Information:					
17	Foreign Reference	42-EP0895171.pdf	2463238 d4e5c48419de6f5f3edab9a42a57b6686ef1603e	no	39
Warnings:					

Information:					
18	Foreign Reference	43-EPO926607.pdf	3783168 82f99a0149416291f9775b3e6a903a8202ac47b0	no	74
Warnings:					
Information:					
19	Foreign Reference	44-EPO949571.pdf	2246210 e090ac4df027a18cb51a141b53a1f824a9b3788b	no	40
Warnings:					
Information:					
20	Foreign Reference	45-WO98-40842.pdf	3996847 f631e807a428f3eb336305478b1ce88ffbff0020	no	118
Warnings:					
Information:					
21	Foreign Reference	46-WO98-43177.PDF	1839850 054919e44072c80d458b1a87557159880acbaef1	no	46
Warnings:					
Information:					
22	Foreign Reference	47-WO97-49252.pdf	1511607 7bfa5f299ba517e99c8f27e90c68e0efe07676b3	no	50
Warnings:					
Information:					
23	Foreign Reference	48-AUA53031-98.pdf	1705632 410c5e3f5665f1ac14f8282911acfdba41cf54a1	no	32
Warnings:					
Information:					
		Total Files Size (in bytes):	29825518		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

ELECTRONIC TRANSMITTAL COVER SHEET

Application Serial No. 12/238,842 Attorney Docket No. EQUI0016D

I hereby certify that this correspondence is being ELECTRONICALLY TRANSMITTED to the United States Patent and Trademark Office

From: GLENN PATENT GROUP
Customer No.: 22,862
Tel: (650) 474-8400
Fax: (650) 474-8401

on February 10, 2009.
Date

Signature

Christine Ortt

Typed or printed name of person signing Certificate

Note: Each paper must have its own certificate of transmission, or this certificate must identify each submitted paper.

Attached to this cover sheet please find the following documents:

- Information Disclosure Statement (2 pages);
- 1449 (3 pages); and
- Cited References

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
12/238,842	09/26/2008	Sean Barger	EQUI0016D

CONFIRMATION NO. 2317

PUBLICATION NOTICE

OC000000035301664

22862
GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

Title: Automated Media Delivery System

Publication No.US-2009-0089422-A1
Publication Date:04/02/2009

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently http://pair.uspto.gov/. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

Form 1449 (Modified)	Serial No.: 12/238,842
	Atty. Docket No.: EQUI0016D
Information Disclosure	Applicant: Sean Barger
Statement By Applicant	Art Unit: 2454
	Confirmation No.: 2317
(Use Several Sheets if Necessary)	Filing Date: September 26, 2008

U.S. Patent Documents

Examiner Initials	No.	Patent No.	Issue Date	Patentee	Filing Date
	1	5,442,771	8/1/1995	Filepp, et al.	
	2	6,484,149	11/1/2002	Jemmes, et al.	

Published U.S, Patent Application

Examiner Initials	No.	Publication No.	Publication Date	Applicant

Foreign Patent or Published Foreign Patent Application

Examiner Initials	No.	Document No.	Publication Date	Applicant

Non-Patent Literature Documents

Examiner Initials	No.	Author, Title, Date, Place (e.g. Journal) of Publication

Examiner's Signature:______________________ Date:____________

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Electronic Acknowledgement Receipt

EFS ID:	5156729
Application Number:	12238842
International Application Number:	
Confirmation Number:	2317
Title of Invention:	Automated Media Delivery System
First Named Inventor/Applicant Name:	Sean Barger
Customer Number:	22862
Filer:	Michael Glenn/Christine Ortt
Filer Authorized By:	Michael Glenn
Attorney Docket Number:	EQUI0016D
Receipt Date:	15-APR-2009
Filing Date:	26-SEP-2008
Time Stamp:	12:28:57
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		IDSEQUI0016D.pdf	145818 1534753f2601a91d10b65525b231fe0514c72362	yes	4

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Miscellaneous Incoming Letter	1	1
	Transmittal Letter	2	3
	Information Disclosure Statement (IDS) Filed (SB/08)	4	4

Warnings:

Information:

Total Files Size (in bytes):	145818

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

ELECTRONIC TRANSMITTAL COVER SHEET

Application Serial No. 12/238,842 Attorney Docket No. EQUI0016D

I hereby certify that this correspondence is being ELECTRONICALLY TRANSMITTED to the United States Patent and Trademark Office

From: GLENN PATENT GROUP
Customer No.: 22,862
Tel: (650) 474-8400
Fax: (650) 474-8401

on April 15, 2009.
Date

Signature

Christine Ortt
Typed or printed name of person signing Certificate

Note: Each paper must have its own certificate of transmission, or this certificate must identify each submitted paper.

Attached to this cover sheet please find the following documents:

- Information Disclosure Statement (2 pages);
- 1449 (1 page)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor	:	Sean BARGER
Serial No.	:	12/238,842
Filed	:	September 26, 2008
Art Unit	:	2454
Confirmation Number	:	2317
Examiner	:	Nathan J. Flynn
Title	:	Automated Media Delivery System
Attorney Docket No.	:	EQUI0016D

April 15, 2009

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

Examiner:

This Information Disclosure Statement is submitted:

(X) under 37 CFR 1.97(b), or
(Within three months of filing national application; or date of entry of international application; or before mailing date of first office action on the merits; whichever occurs last)

( ) under 37 CFR 1.97(c) together with either a:
( ) Certification under 37 CFR 1.97(e), or
( ) a $180.00 fee under 37 CFR 1.17(p), or
(After the CFR 1.97(b) time period, but before final action or notice of allowance, whichever occurs first)

( ) under 37 CFR 1.97(d) together with a:
( ) Certification under 37 CFR 1.97(e), and
( ) a $180.00 fee under 37 CFR 1.17(p).
(Filed after final action or notice of allowance, whichever occurs first, but before payment of the issue fee)

(X) The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 07-1445 (Order No. EQUI0016D).

(X) Applicant(s) submit herewith PTO Form 1449 (Modified) -- Information Disclosure Citation together with copies of patents, publications or other information of which applicant(s) are aware, which applicant(s) believe(s) may be material to the examination of this application and for which there may be a duty to disclose in accordance with 37 CFR 1.25.

( ) A concise explanation of the relevance of foreign language patents, foreign language publications and other foreign language information listed on PTO Form 1449 (Modified), as presently understood by the individual(s) designated in 37 CFR 156(c) most knowledgeable about the content is given on the attached sheet, or where a foreign language patent is cited in a search report or other action by a foreign patent office in a counterpart foreign application, an English language version of the search report or action which indicates the degree of relevance found by the foreign office is listed on form PTO Form 1449 (Modified) and is enclosed herewith.

It is requested that the information disclosed herein be made of record in this application.

Respectfully Submitted,

Michael A. Glenn
Reg. No. 30,176

Customer No. 22862

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/238,842	09/26/2008	Sean Barger	EQUI0016D	2317

22862 7590 04/30/2010
GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

EXAMINER	
BRYANT, CHRISTOPHER D	

ART UNIT	PAPER NUMBER
2178	

NOTIFICATION DATE	DELIVERY MODE
04/30/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eptomatters@glenn-law.com

Office Action Summary	Application No.	Applicant(s)
	12/238,842	BARGER ET AL.
	Examiner	**Art Unit**
	CHRISTOPHER BRYANT	2178

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 September 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 September 2008* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *2/10/2009, 4/15/2009*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed on 9/26/2008. ***This action is made Non-Final.*** Claims 1-21 are pending in the case. Claims 1, 10, and 19 are independent claims.

Information Disclosure Statement

2. The information disclosure statement's (IDS) submitted on 2/10/2009 and 4/15/2009 has been entered, and considered by the Examiner.

Priority

3. Applicant has made for claim of priority from application 11/269,916 filed 11/7/2005.

Drawings

4. The drawings filed on 5/12/2006 have been accepted by the Examiner.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-5, and 7-9 are rejected under 35 U.S.C. 102(b) as being anticipated by Mendhekar et al. (United States Patent 6,938,073), hereinafter, referenced as Mendhekar.

Regarding claim 1, Mendhekar discloses receiving requests from the multiple devices for concurrent playback of media content at a first quality level, “that automatically modify an application in view of one or more of the following: the application, and the capability of the client requesting the application and the transmission medium delivering the application” (pg.2, lines 10-14); determining a set of independent transformations of the media content that fulfill the requests at the first quality level, “each of these clients has different capabilities connected to different transmission media” (column 2, lines 18-21); if transformations are required, determining whether processing resources available on the host computer are sufficient to perform the independent transformations, “the server computer automatically modifies the specific application in view of the resources available so as to maximize the benefits of the resources” (column 2, lines 22-24); and if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer, “a term with an application appended by a client designates an appliance-specific transducer 150 for that application and that client. For example, a DocTV designates an appliance specific transducer for modifying a document for television viewing. This transducer can include a number of sub-transducers, such as a format sub-transducer for TV to change the format of the image depending on the television. This can be changing the size of the text so that a user six feet away can see the text. So there can be one or more of such sub-transducers to adapt the application for television viewing. Note that television is typically watched six or more feet away, while a computer monitor is less than one foot away from a viewer. Other sub-transducers, depending on needs, can be a text sub-transducer to change the text characteristics based upon the client, an image sub-transducer to change the image characteristics based upon the client's display capabilities, and a user-interface sub-transducer to enhance the usability of the application on the client. The appliance-specific transducer 150 can add one or more sub-transducers dynamically to modify the application accordingly" (column 5, lines 34-49, corresponding to Fig.4).

Regarding claim 2, Mendhekar discloses performing independent transformations, "the application can be of an interactive type, such as a web-page, which a user can interact with. This embodiment includes a server apparatus in a server. The server apparatus includes an appliance-specific transducer and an adaptive transmission transducer. The appliance-specific transducer, in view of a client's request for an interactive application in the server, is configured to modify the application based on the client and the application to generate an appliance-specific output. This output is modified automatically depending on the client and the application itself. The adaptive transmission transducer, which is coupled to the appliance-specific transducer, is configured to modify the appliance-specific output. The modification is again based on the application and the corresponding medium of transmission" (column 2, lines 25-35).

Regarding claim 3, Mendhekar discloses performing the dependent transformations, "the modification, an adapted output is generated, and is delivered through the medium to the client. Then, the client decodes the adapted output to produce a modified version of the interactive application that is adapted for the client" (column 2, lines 40-43).

Regarding claim 4, Mendhekar discloses monitoring available bandwidth of the network, "applications are automatically mapped onto various clients based on the resources available. In one embodiment, the resources include the bandwidth, acceptable error rates, and the latency of the transmission medium, and the processing power and memory capacity of a client" (column 2, lines 63-67); determining whether a requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network, "the compression algorithm and the network protocol to be operated on the appliance-specific output 179 depend on at least one characteristic of the application, such as the reliability or error rate of the application required, and the speed or bandwidth of the transmission medium 154" (column 6, lines 18-23); and if the requested set of media streams is not transmissible within the available bandwidth of the network, determining the set of dependent transformations such that a modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network, "the parameters used to choose the compression algorithm include: (a) Input data type, whether it is text graphics, natural images, audio or video. Note that the application can have a number of different input data type, such as there can be text and images on the same page. (b) Rate-distortion requirement, or the compromise between bandwidth versus quality. (c) The transmission medium 154, which affects the bandwidth, error rate and latency. (d) Capabilities of the server and the client, such as the processing power and memory available" (column 7, lines 0-12).

Claim 5 contains limitations similar to claim 4; therefore claim 5 is rejected under the same limitations that claim4 was rejected under.

Regarding claim 7, Mendhekar discloses where the second quality level is lesser than the first quality level, "Rate-distortion requirement, or the compromise between bandwidth versus quality. (c) The transmission medium 154, which affects the bandwidth, error rate and latency. (d) Capabilities of the server and the client, such as the processing power and memory available" (column 7, lines 0-12).

Regarding claim 8, Mendhekar discloses where the quality level is measured in terms of one of the following: a selected bit rate of the media content, "Rate-distortion requirement, or the compromise between bandwidth versus quality" (column 7, lines 0-12).

Regarding claim 9, Mendhekar discloses having instructions for performing a computer process implementing the method of 1, "apparatus that automatically modify an application in view of one or more of the following: the application, and the capability of the client requesting the application and the transmission medium delivering the application" (column 2, lines 10-12).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mendhekar in view of Gabriel et al. (United States Patent Application Publication US 2006/0015580), hereinafter, referenced as Gabriel.

Regarding claim 6, Mendhekar teaches everything claimed above; however the art does not disclose determining capabilities of the playback devices for processing a media stream; determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations. However, the examiner maintains that it was well known in the art to check a users playback devices before streaming, as taught by Gabriel.

In a similar field of endeavor Gabriel discloses determining capabilities of the playback devices for processing a media stream, "multimedia encoding formats typically also include support for scalability. For example, video scalability offers a set of tools by which video can be coded at different resolutions (e.g., different scales) in a single bitstream. On the decoder side, video can be decoded at a suitable resolution by extracting the portion of the bitstream that represents a particular resolution. Among other advantages, scalability adds compatibility with devices of different capability but usually at the cost of some bitstream overhead" (pg.1, paragraph 11); determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations, “multimedia content distribution facility (MCDF) that supports device-specific delivery of multimedia content to a user's playback device as a function of the playback device's capabilities” (pg.2, paragraph 26); and if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations, “Consider further the same subscriber with two HD digital set-tops, one in the living room and one in the bedroom, but where the one in the living room is connected to an HD display and the one in the bedroom is connected to a display with only SD capability. In this case, assuming the HD digital set-top in the bedroom can down-convert the HD version of the movie to SD before outputting the decoded output to the SD display, the subscriber might be able to purchase the movie and watch the first half in the living room in full HD resolution and still watch the second half of the movie in the bedroom, albeit at a reduced resolution” (pg.2, paragraph 16).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar, by checking a users playback devices before streaming, as taught by Gabriel, for the purpose of providing media to customers based on their device.

Claims 10-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mendhekar in view of Zhang et al. (United States Patent 7,477,688), hereinafter, referenced as Zhang.

Regarding claim 10, Mendhekar teaches requests from the multiple devices for concurrent playback of media content at a first quality level, “that automatically modify an application in view of one or more of the following: the application, and the capability of the client requesting the application and the transmission medium delivering the application" (pg.2, lines 10-14); determining a set of independent transformations of the media content that fulfill the requests at the first quality level, "each of these clients has different capabilities connected to different transmission media" (column 2, lines 18-21); determining whether processing resources available on the host computer are sufficient to perform the independent transformations, "the server computer automatically modifies the specific application in view of the resources available so as to maximize the benefits of the resources" (column 2, lines 22-24); and if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer, "a term with an application appended by a client designates an appliance-specific transducer 150 for that application and that client. For example, a DocTV designates an appliance specific transducer for modifying a document for television viewing. This transducer can include a number of sub-transducers, such as a format sub-transducer for TV to change the format of the image depending on the television. This can be changing the size of the text so that a user six feet away can see the text. So there can be one or more of such sub-transducers to adapt the application for television viewing. Note that television is typically watched six or more feet away, while a computer monitor is less than one foot away from a viewer. Other sub-transducers, depending on needs, can be a text sub-transducer to change the text characteristics based upon the client, an image sub-transducer to change the image characteristics based upon the client's display capabilities, and a user-interface sub-transducer to enhance the usability of the application on the client. The appliance-specific transducer 150 can add one or more sub-transducers dynamically to modify the application accordingly" (column 5, lines 34-49, corresponding to Fig.4); however the art does not disclose if the processing resources available on the host computer are sufficient to perform the independent transformations, performing the independent transformations to create a requested set of media streams; and performing the dependent transformations to create a modified set of media streams; and transmitting the requested set of media streams or the modified set of media streams across the network. However, the examiner maintains that it was well known in the art to modify and transmit media based on the resources available, as taught by Zhang.

In a similar field of endeavor Zhang discloses if the processing resources available on the host computer are sufficient to perform the independent transformations, performing the independent transformations to create a requested set of media streams, "Each of the local networks 23, 27, and 29 has a different available bandwidth. The available bandwidth determines the maximum bit rate that may be used between the network device 22 and each respective decoders 24, 30 and 36" (column 7, lines 10-16); performing the dependent transformations to create a modified set of media streams, "Each of the local networks 23, 27, and 29 has a different available bandwidth. The available bandwidth determines the maximum bit rate that may be used between the network device 22 and each respective decoders 24, 30 and 36. By way of example, the local network 23 has an available bandwidth that allows a bit rate of 2.5

million bits per second (Mbps), the local network 27 has an available bandwidth that allows a bit rate of 1.5 Mbps, and the local network 29 has an available bandwidth that allows a bit rate of 2.0 Mbps" (column 7, lines 16-20); and transmitting the requested set of media streams or the modified set of media streams across the network, "The present invention allows the compressed video stream produced by the encoder 14 to be sent across any of the channels 17, 18 and 19 to the decoders 24, 30 and 36" (column 7, lines 22-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar, by modifying and transmitting media based on the resources available, as taught by Zhang, for the purpose of modifying compressed bitstream provided by encoder to fit available bandwidth provided by local networks.

Claim 11 contains limitations similar to claim 6; therefore claim 11 is rejected under the same limitations that claim 6 was rejected under.

Claim 12 contains limitations similar to claim 4; therefore claim 12 is rejected under the same limitations that claim 4 was rejected under.

Claim 13 contains limitations similar to claim 5; therefore claim 13 is rejected under the same limitations that claim 5 was rejected under.

Regarding claim 14, Mendhekar teaches everything claimed above; however the art does not disclose transmitting across the network a modified set of media streams resulting from the dependent transformations across the network. However, the examiner maintains that it was well known in the art to modify and transmit media based on the resources available, as taught by Zhang.

In a similar field of endeavor Zhang discloses, “network device 22 may alter the modified compressed bitstream provided by the encoder 14 to fit the available bandwidth provided by the local networks 23, 27 and 29. The network device 22 may alter the modified compressed bitstream by selectively re-quantizing video data in different portions of the incoming compressed bitstream. It does so using multiple re-quantization schemes and in a timely manner that does not sacrifice real-time transmission of the video data” (column 7, lines 22-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar, by modifying and transmitting media based on the resources available, as taught by Zhang, for the purpose of modifying compressed bitstream provided by encoder to fit available bandwidth provided by local networks.

Regarding claim 15, Mendhekar teaches everything claimed above; however the art does not disclose storing a modified set of media streams resulting from the dependent transformations as media files on a storage device. However, the examiner maintains that it was well known in the art to store the streams, as taught by Zhang.

In a similar field of endeavor Zhang discloses, “it may employ one or more memories or memory modules (e.g., memory 861) configured to store program instructions for the network operations and other functions of the present invention described herein. The program instructions may specify an operating system and one or more applications, for example. Such memory or memories may also be configured to store data bitstreams" (column 19, lines 60-66).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar, by storing streams, as taught by Zhang, for the purpose of saving streams.

Claim 16 contains limitations similar to claim 7; therefore claim 16 is rejected under the same limitations that claim 7 was rejected under.

Claim 17 contains limitations similar to claim 8; therefore claim 17 is rejected under the same limitations that claim 8 was rejected under.

Claim 18 contains limitations similar to claim 9; therefore claim 18 is rejected under the same limitations that claim 9 was rejected under.

Claim 19 is the "system" claim which is a combination of claims 1 and 10; claim 19 contains limitations similar to claims 9 and 10; therefore claim 19 is rejected under the same grounds those claims were rejected.

Claim 20 is the "system" claim which is a combination of claims 4 and 5; claim 20 contains limitations similar to claims 4 and 5; therefore claim 20 is rejected under the same grounds those claims were rejected.

Claim 21 is the "system" claim of claim 15; claim 21 contains limitations similar to claims 15; therefore claim 21 is rejected under the same grounds claim 15 was rejected.

Conclusion

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure, “User Attention-Based Adaptation of Quality Level to Improve the Management of Real-Time Multi-Media Content Delivery and Distribution” (United States Patent 7,284,201) and "Methods for Streaming Media Data” (United States Patent 7,673,063).

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRISTOPHER BRYANT whose telephone number is (571)270-7260. The examiner can normally be reached on Monday-Friday 9:00 A.M. to 6:00 P.M. EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Stephen Hong can be reached on (571)272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-270-8260.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/CHRISTOPHER BRYANT/
Examiner, Art Unit 2178

/Joshua D Campbell/
Primary Examiner, Art Unit 2178

Notice of References Cited	Application/Control No. 12/238,842	Applicant(s)/Patent Under Reexamination BARGER ET AL.	
	Examiner CHRISTOPHER BRYANT	Art Unit 2178	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-7,477,688	01-2009	Zhang et al.	375/240
*	B	US-2006/0015580	01-2006	Gabriel et al.	709/219
*	C	US-6,938,073	08-2005	Mendhekar et al.	709/217
*	D	US-7,284,201	10-2007	Cohen-Solal, Eric	715/738
*	E	US-7,673,063	03-2010	Xie et al.	709/231
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No. 12238842	Applicant(s)/Patent Under Reexamination BARGER ET AL.
	Examiner CHRISTOPHER BRYANT	Art Unit 2178

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	04/21/2010								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								
	21	✓								

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	12238842	BARGER ET AL.
	Examiner	**Art Unit**
	CHRISTOPHER BRYANT	2178

SEARCHED

Class	Subclass	Date	Examiner
709	219, 236	4/20/2010	cb
715	733, 738	4/20/2010	cb

SEARCH NOTES

Search Notes	Date	Examiner
see file EastSearchHistory	4/20/2010	cb

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

/CHRISTOPHER BRYANT/ Examiner.Art Unit 2178	

tanceForm 1449 (Modified) **Information Disclosure Statement By Applicant** (Use Several Sheets if Necessary)	**Atty. Docket No.** EQUI0016D **Applicant:** Sean Barger, et al. **Filing Date:** September 26, 2008	**Serial No.:** 12/238,842 **Group:** 2454 **Confirmation No:** 2317

U.S. Patent Documents

Examiner Initial	No.	Patent No.	Issue Date	Patentee	Class	Sub-class	Filing Date
/C.B./	1	5,088,052	Feb-92	Spielman, et al.			
/C.B./	2	5,355,472	Oct-94	Lewis			
/C.B./	3	5,530,852	Jun-96	Meske, Jr., et al.			
/C.B./	4	5,701,451	Dec-97	Rogers, et al.			
/C.B./	5	5,708,845	Jan-98	Wistendahl, et al.			
/C.B./	6	5,710,918	Jan-98	Lagarde, et al.			
/C.B./	7	5,737,619	Apr-98	Judson			
/C.B./	8	5,745,908	Apr-98	Anderson, et al.			
/C.B./	9	5,758,110	May-98	Boss, et al.			
/C.B./	10	5,761,655	Jun-98	Hoffman, Michael T.			
/C.B./	11	5,793,964	Aug-98	Rogers, et al.			
/C.B./	12	5,819,261	Oct-98	Takahashi, et al.			
/C.B./	13	5,822,436	Oct-98	Rhoads			
/C.B./	14	5,845,084	Dec-98	Cordell, et al.			
/C.B./	15	5,845,299	Dec-98	Arora, et al.			
/C.B./	16	5,860,068	Jan-99	Cook			
/C.B./	17	5,860,073	Jan-99	Ferrel, et al.			
/C.B./	18	5,861,881	Jan-99	Freeman, et al.			
/C.B./	19	5,862,325	Jan-99	Reed, et al.			
/C.B./	20	5,864,337	Jan-99	Marvin, John			
/C.B./	21	5,870,552	Feb-99	Dozier, et al.			
/C.B./	22	5,880,740	Mar-99	Halliday, et al.			
/C.B./	23	5,890,170	Mar-99	Sidana			
/C.B./	24	5,895,476	Apr-99	Orr, et al.			
/C.B./	25	5,895,477	Apr-99	Orr, et al.			
/C.B./	26	5,903,892	May-99	Hoffert, et al.			
/C.B./	27	5,937,160	Aug-99	Davis, et al.			
/C.B./	28	5,943,680	Aug-99	Ohga, et al.			
/C.B./	29	5,956,737	Sep-99	King, et al.			
/C.B./	30	6,009,436	Dec-99	Motoyama, et al.			
/C.B./	31	6,456,305	Sep-02	Qureshi, et al.			
/C.B./	32	6,563,517	May-03	Bhagwat, et al.			
/C.B./	33	6,591,280	Jul-03	Orr			
/C.B./	34	6,623,529	Sep-03	Lakritz			

Published U.S. Patent Application

Examiner Initial	No.	Document No.	Publication Date	Assignee	Class	Sub-class	Translation Yes	Translation No

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.B./

Foreign Patent or Published Foreign Patent Application

Examiner Initial	No.	Document No.	Publication Date	Assignee	Class	Sub-class	Translation Yes	Translation No
/C.B./	35	EP 0747842	12/11/1996	International Business Machines Corp.				
/C.B./	36	EP 0782085	7/2/1997	International Business Machines Corp.				
/C.B./	37	EP 0818907	1/14/1998	AT&T Corp				
/C.B./	38	EP 0843276	5/20/1998	Canon Information Systems Inc				
/C.B./	39	EP 0876034	11/4/1998	International Business Machines Corp.				
/C.B./	40	EP 0883068	12/9/1998	Home Informaiton Services, Inc.				
/C.B./	41	EP 0886409	12/23/1998	Digital Vision Laboratories Corporation				
/C.B./	42	EP 0895171	2/3/1999	Neoforma, Inc.				
/C.B./	43	EP 0926607	6/30/1999	Ricoh KK				
/C.B./	44	EP 0949571	10/13/1999	Xerox Corp				
/C.B./	45	WO 98/40842	9/17/1998	Computer Information and Sciences, Inc.				
/C.B./	46	WO 98/43177	10/1/1998	Intel Corp				
/C.B./	47	WO 97/49252	12/24/1997	Senthilkumar, et al.				
/C.B./	48	AU-A-53031/98	8/27/1996	Dudley, John Mills				

Other Documents

Examiner Initial	No.	Author, Title, Date, Place (e.g. Journal) of Publication
/C.B./	A	Sakaguchi, et al.; " A browsing tool for multi-lingual documents for users without multi-lingual fonts"; 1996; ACM International Conference On Digital Libraries, pp. 63-71
/C.B./	B	Zaiane, et al.; "Mining multimedia data"; Nov. 1998; ACM Conference of the Center for Advanced Studies on Collaborative research, pp. 1-18
/C.B./	C	BULTERMAN, DICK.C.A.; Models, Media and Motion: Using the Web to Support Multimedia Documents; Proceedings of 1997 Intnl Conf on Multimedia Modeling; p. 17-20; November 1997; SINGAPORE
/C.B./	D	MOHLER, J.L.; Migrating Course Materials to the World Wide Web: A Case Study of the Department of Techinal Graphics at Purdue University; Computer Networks and ISDN Systems; Vol. 30, Issues 20-21, p,1981-1990; November 12, 1988

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.B./

/C.B./	E	DOBSON, R.; Animating Your Web Pages with Direct Animation; Web Techniques; vol.3, no. 6, p. 49-52; June 1998
/C.B./	F	BERINSTEIN, Paula; "The Big Picture; Text and Graphics on UMI's ProQuest Direct: The Best (Yet) of Both Words"; March 1997; retrieved on 3/23/04 from website: http://www.infotoday.com/online/MarOL97/picture3.html
/C.B./	G	McNeil, Sara; Research Interests; retrieved on March 18, 2004 from website: http//www.coe.uh.edu/`smcneil/research.htm
/C.B./	H	Tables of Contents service for Computers & Geosciences; Copyright 1997; Computers and GeoSciences, Volume 23, Issue 5, retrieved on 3/18/04 from website: http://library.iem.ac.ru/comp&geo/00983004/sz977014.html

Examiner's Signature /Christopher Bryant/ Date 04/19/2010

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.B./

Form 1449 (Modified)	Serial No.: 12/238,842
	Atty. Docket No.: EQUI0016D
Information Disclosure	Applicant: Sean Barger
Statement By Applicant	Art Unit: 2454
	Confirmation No.: 2317
(Use Several Sheets if Necessary)	Filing Date: September 26, 2008

U.S. Patent Documents

Examiner Initials	No.	Patent No.	Issue Date	Patentee	Filing Date
/C.B./	1	5,442,771	8/1/1995	Filepp, et al.	
/C.B./	2	6,484,149	11/1/2002	Jemmes, et al.	

Published U.S, Patent Application

Examiner Initials	No.	Publication No.	Publication Date	Applicant

Foreign Patent or Published Foreign Patent Application

Examiner Initials	No.	Document No.	Publication Date	Applicant

Non-Patent Literature Documents

Examiner Initials	No.	Author, Title, Date, Place (e.g. Journal) of Publication

Examiner's Signature: /Christopher Bryant/ Date: 04/19/2010

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.B./

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIB DATA SHEET

CONFIRMATION NO. 2317

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
12/238,842	09/26/2008 **RULE**	715	2178	EQUI0016D

APPLICANTS

Sean Barger, Mill Valley, CA;
Steve Johnson, Mill Valley, CA;
Matt Butler, Beaverton, OR;
Jerry Destremps, Sausalito, CA;
David Pochron, Cambridge, WI;
Trent Brown, San Anselmo, CA;

** **CONTINUING DATA** *************************

This application is a DIV of 12/173,747 07/15/2008
which is a DIV of 11/269,916 11/07/2005 ABN
which is a CIP of 09/929,904 08/14/2001 PAT 6,964,009
which is a CON of 09/425,326 10/21/1999 PAT 6,792,575

** **FOREIGN APPLICATIONS** *************************

** **IF REQUIRED, FOREIGN FILING LICENSE GRANTED** ** ** SMALL ENTITY **
10/06/2008

Foreign Priority claimed / 35 USC 119(a-d) conditions met / Verified and Acknowledged	Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ❑ Yes ☑ No 35 USC 119(a-d) conditions met ❑ Yes ☑ No Verified and Acknowledged /CHRISTOPHER D BRYANT/ Examiner's Signature	❑ Met after Allowance cb Initials	CA	23	21	3

ADDRESS

GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025
UNITED STATES

TITLE

Automated Media Delivery System

FILING FEE RECEIVED	FEES: Authority has been given in Paper	
525	No.____________ to charge/credit DEPOSIT ACCOUNT No.____________ for following:	❑ All Fees ❑ 1.16 Fees (Filing) ❑ 1.17 Fees (Processing Ext. of time) ❑ 1.18 Fees (Issue) ❑ Other ________________ ❑ Credit

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L6	31	709/236.ccls. and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2010/04/21 17:05
L7	140	709/219.ccls. and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2010/04/21 17:05
L8	3	715/733.ccls. and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2010/04/21 17:06
L9	6	715/738.ccls. and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2010/04/21 17:07
S1	1	"6964009".pn.	US-PGPUB; USPAT	OR	ON	2010/04/19 11:57
S2	441	barger.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:49
S3	4	S2 with sean	US-PGPUB; USPAT	OR	ON	2010/04/19 12:49
S4	49550	johnson.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:50
S5	97	S4 with steve	US-PGPUB; USPAT	OR	ON	2010/04/19 12:50
S6	24	S5 and media	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51
S7	5894	butler.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51

S8	4	S7 with matt	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51
S9	9	destremps.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51
S10	2	S9 with jerry	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51
S11	4	pochron.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:52
S12	6	brown.in. with trent	US-PGPUB; USPAT	OR	ON	2010/04/19 12:52
S13	378	bandwidth same media same modify	US-PGPUB; USPAT	OR	ON	2010/04/19 13:08
S14	214	S13 and ((downgrade or quality or lower) same media)	US-PGPUB; USPAT	OR	ON	2010/04/19 13:16
S15	12	S14 and (transform$ same media)	US-PGPUB; USPAT	OR	ON	2010/04/19 13:18
S16	122	S14 and (format$ same media)	US-PGPUB; USPAT	OR	ON	2010/04/19 13:19
S17	214	S13 and ((downgrade or quality or lower or lesser) same media)	US-PGPUB; USPAT	OR	ON	2010/04/19 13:48
S18	4446	media same lower same resolution	US-PGPUB; USPAT	OR	ON	2010/04/19 14:25
S19	0	S18 and bandwith and request and playback	US-PGPUB; USPAT	OR	ON	2010/04/19 14:25
S20	12	S18 and bandwith and request	US-PGPUB; USPAT	OR	ON	2010/04/19 14:25
S21	283	S18 and (request same media same content)	US-PGPUB; USPAT	OR	ON	2010/04/19 14:28
S22	381	request same media same lower same (resolution or bandwidth)	US-PGPUB; USPAT	OR	ON	2010/04/19 15:09

S23	268	S22 and quality	US-PGPUB; USPAT	OR	ON	2010/04/19 15:09
S24	182	S23 and (format or reformat)	US-PGPUB; USPAT	OR	ON	2010/04/19 15:10
S25	73	S24 and transform	US-PGPUB; USPAT	OR	ON	2010/04/19 15:10
S26	237	optimize same media same playback	US-PGPUB; USPAT	OR	ON	2010/04/19 16:02
S27	84	S26 and (lower same resolution)	US-PGPUB; USPAT	OR	ON	2010/04/19 16:02
S28	150	media same transfer same device same lower same quality	US-PGPUB; USPAT	OR	ON	2010/04/19 16:24
S29	64	media same transfer same device same lower same resolution	US-PGPUB; USPAT	OR	ON	2010/04/19 16:26
S30	124	transmitting same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2010/04/19 16:39
S31	55	S30 and bandwidth	US-PGPUB; USPAT	OR	ON	2010/04/19 16:39
S32	19	formatting same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2010/04/19 17:02
S33	5034	format$ same media same multiple same device	US-PGPUB; USPAT	OR	ON	2010/04/19 17:08
S34	1455	S33 and (media same stream)	US-PGPUB; USPAT	OR	ON	2010/04/19 17:09
S35	539	S34 and resolution and bandwidth	US-PGPUB; USPAT	OR	ON	2010/04/19 17:16
S36	496	S35 and network	US-PGPUB; USPAT	OR	ON	2010/04/19 17:16
S37	448	S36 and quality	US-PGPUB; USPAT	OR	ON	2010/04/19 17:16

S38	354	S37 and (bit same rate)	US-PGPUB; USPAT	OR	ON	2010/04/19 17:17
S39	183	S38 and transformation	US-PGPUB; USPAT	OR	ON	2010/04/19 17:19
S40	442	available same bandwidth same device same media same stream	US-PGPUB; USPAT	OR	ON	2010/04/21 13:08
S41	442	S40 same device	US-PGPUB; USPAT	OR	ON	2010/04/21 13:10
S42	61	S41 and (resolution same different)	US-PGPUB; USPAT	OR	ON	2010/04/21 13:17
S43	1	"7477688".pn.	US-PGPUB; USPAT	OR	ON	2010/04/21 14:32

EAST Search History (Interference)

<This search history is empty>

4/21/2010 5:25:02 PM
C:\Documents and Settings\cbryant2\My Documents\EAST\Workspaces\09262008automatedmediadeliv.wsp

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 12/238,842	Filing Date 09/26/2008	☐ To be Mailed

APPLICATION AS FILED – PART I

FOR	NUMBER FILED (Column 1)	NUMBER EXTRA (Column 2)	SMALL ENTITY ☒ RATE ($)	FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	15 minus 20 =	* 0	X $25 =	0	OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	2 minus 3 =	* 0	X $105 =	0		X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
			TOTAL	0		TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

AMENDMENT 07/16/2010	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 15	Minus	** 20	= 0	X $26 =	0	OR	X $ =	
Independent (37 CFR 1.16(h))	* 2	Minus	***3	= 0	X $110 =	0	OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE	0	OR	TOTAL ADD'L FEE	

AMENDMENT	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	RATE ($)	ADDITIONAL FEE ($)	OR	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/GERALDINE STANLEY/

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor:	Sean Barger
Serial No.:	12/238,842
Filed:	September 26, 2008
Art Unit:	2178
Examiner:	Christopher Bryant
Title:	AUTOMATED MEDIA DELIVERY SYSTEM
Attorney Docket No.:	EQUI0016D

July 28, 2010

Mail Stop: Amendments
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO OFFICE ACTION

Applicant submits this Response to the Office Action dated April 30, 2010 in connection with the above-identified patent application.

A **Listing of Claims** begins on Page 2 of this paper, and
Remarks begin on Page 10 of this paper.

Applicant does not believe that the filing of this amendment will incur additional fees. However, the Commissioner is authorized to charge any fees that may be due and to credit any overpayments to Deposit Account 07-1445 (Order No. EQUI0016D). Applicant considers this document to be filed in a timely manner.

LISTING OF CLAIMS

1. (Currently Amended) A method in a host computer for developing transformation processing operations to optimize media content playback across multiple playback devices connected with the host computer in a network, the method comprising:

determining capabilities of the playback devices for processing a media stream;

receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following:

a selected compression format of the media content;

a selected bit rate of the media content; and

an image resolution of the media content;

determining a set of independent transformations of the media content that fulfill the requests at the first quality level;

if transformations are required, determining whether processing resources available on the host computer are sufficient to perform the independent transformations; ~~and~~

if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer;

determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and

if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations.

2. (Original) The method of claim 1, wherein if the processing resources available on the host computer are sufficient to perform the independent transformations, the method further comprises performing the independent transformations.

3. (Original) The method of claim 1, further comprising performing the dependent transformations.

4. (Original) The method of claim 2 further comprising monitoring available bandwidth of the network; determining whether a requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network; and if the requested set of media streams is not transmissible within the available bandwidth of the network, determining the set of dependent transformations such that a modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network.

5. (Original) The method of claim 3 further comprising monitoring available bandwidth of the network; determining whether a modified set of media streams resulting from the dependent transformations is transmissible within the available bandwidth of the network; and if the modified set of media streams is not transmissible within the available bandwidth of the network, determining a revised set of dependent transformations such that a revised set of media streams resulting from the revised set of dependent transformations is transmissible within the available bandwidth of the network.

6. (Cancelled).

7. (Original) The method of claim 1, wherein the second quality level is lesser than the first quality level.

8. (Cancelled).

9. (Original) A computer-readable medium having computer-executable instructions for performing a computer process implementing the method of claim 1.

10. (Currently Amended) A method in a host computer for developing transformation processing operations to optimize media content playback across multiple playback devices connected with the host computer in a network, the method comprising:

determining capabilities of the playback devices for processing a media stream;

receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following:

a selected compression format of the media content;

a selected bit rate of the media content; and

an image resolution of the media content;

determining a set of independent transformations of the media content that fulfill the requests at the first quality level;

determining whether processing resources available on the host computer are sufficient to perform the independent transformations; and if the processing resources available on the host computer are sufficient to perform the independent transformations, performing the independent transformations to create a requested set of media streams;

if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer; ~~and~~

performing the dependent transformations to create a modified set of media streams; and transmitting the requested set of media streams or the modified set of media streams across the network;

determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and

if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations.

11. (Cancelled).

12. (Original) The method of claim 10 further comprising monitoring available bandwidth of the network; determining whether the requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network; and if the requested set of media streams is not transmissible within the available bandwidth of the network, determining the set of dependent transformations such that the modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network.

13. (Original) The method of claim 10 further comprising monitoring available bandwidth of the network; determining whether the modified set of media streams resulting from the dependent transformations is transmissible within the available bandwidth of the network; and if the modified set of media streams is not transmissible within the available bandwidth of the network, determining a revised set of dependent transformations such that a revised set of media streams resulting from the revised set of dependent transformations is transmissible within the available bandwidth of the network.

14. (Original) The method of claim 10 further comprising transmitting across the network a modified set of media streams resulting from the dependent transformations across the network.

15. (Original) The method of claim 10 further comprising storing a modified set of media streams resulting from the dependent transformations as media files on a storage device.

16. (Original) The method of claim 10, wherein the second quality level is lesser than the first quality level.

17. (Currently Amended) The method of claim 10, wherein the quality level is measured in terms of one or[[e]] more of the following: a selected format of the media content, a selected bit rate of the media content, and an image resolution of the media content.

18. (Original) A computer-readable medium having computer-executable

instructions for performing a computer process implementing the method of claim 10.

19. (Currently Amended) A host computer system comprising: ~~for~~

at least one processor operatively coupled with memory, the processor configured for determining capabilities of the playback devices for processing a media stream;

the processor further configured for performing transformative processing operations to optimize media content playback across multiple devices, which are connected with the host computer in a network,

the processor further configured for requesting concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following:

a selected compression format of the media content;

a selected bit rate of the media content; and

an image resolution of the media content; ~~, the system comprising~~

one or more media processing units;

a policy engine module that determines a set of independent transformations of the media content that fulfill requests at the first quality level;

wherein the policy engine module determines whether processing resources of the media processing units are sufficient to perform the independent transformations; ~~and~~

wherein if the processing resources are sufficient to perform the independent transformations, the policy engine module directs the media processing units to perform the independent transformations to create a requested set of media streams;

wherein and if the processing resources are insufficient to perform the independent transformations, the policy engine module determines a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources; ~~and~~

wherein the policy engine module directs the media processing units to perform the dependent transformations to create a modified set of media streams; ~~and~~

a network link that communicates with the network and transmits either the requested set of media streams or the modified set of media streams to the multiple devices;

a network monitor module that monitors available bandwidth of the network and passes bandwidth information to the policy engine module; and wherein the policy engine module further determines:

whether the requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network;

if the requested set of media streams is not transmissible within the available bandwidth of the network, determines the set of dependent transformations such that the modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network; and determines whether the modified set of media streams resulting from the dependent transformations is transmissible within the available bandwidth of the network; and

if the modified set of media streams is not transmissible within the available bandwidth of the network, determining a revised set of dependent transformations such that a revised set of media streams resulting from the revised set of dependent transformations is transmissable within the available bandwidth of the network.

20. (Cancelled).

21. (Original) The system of claim 20 further comprising a storage device that stores an output stream of transformed media content processed by the media processing units.

REMARKS

Applicants respectfully request further examination and reconsideration in view of the above amendments and arguments set forth fully below. Claims 1-21 were previously pending in the present application. Within the Office Action, Claims 1-21 have been rejected.

Claim Rejections under 35 U.S.C. § 102(e)

Within the Office Action, Claims 1-5 and 7-9 were rejected under 35 U.S.C. § 102(e) as being anticipated by United States Patent No. 6,938,073 to Mendhekar et al. (hereinafter referred to as "Mendhekar").

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (Fed. Cir. 1987).

The Applicants respectfully traverse this rejection because each and every element set forth in Claims 1-5 and 7-9 are not found in Mendhekar, either expressly nor inherently described. Specifically, Mendhekar does not describe systems and methods for "developing transformation processing operations to optimize media content playback ... comprising: determining capabilities of the playback devices for processing a media stream; receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following: a selected compression format of the media content; a selected bit rate of the media content; and an image resolution of the media content; ... determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations."

Mendhekar involves a system of generating pre-determined renditions of an image in a plurality of formats.

On the other hand, the Applicants claim a system for converting input content on-the-fly (i.e. transcoding simultaneously said multi-media data into at least one alternate format) and delivering it. Furthermore, the only transcode disclosed in Mendhekar is a transcode between bit rates (as in the rejection of Claim 8). Indeed, Mendhekar does not involve on-the-fly transcoding between selected compression formats of the media content; a selected bit rate of the media content; and an image resolution of the media content, nor does the Examiner suggest that he does.

On the contrary, Claims 1-5, 7, and 9 include the above-mentioned limitation. For at least this reason, Claims 1-5, 7, and 9 are not anticipated by Mendhekar.

Claim Rejections under 35 U.S.C. § 103 - Mendhekar and Gabriel

Also within the Office Action, Claim 6 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Mendhekar in view of United States Patent application publication no. 2006/0015580 to Gabriel et al. (hereinafter referred to as "Gabriel"). Applicants cancel claim 6 herein, thereby rendering the rejection moot. However, the Applicants incorporate the limitations of Claim 6 into Claims 1, 10, and 19. Below is an explanation of why Claims 1, 10, and 19 are not rendered obvious by a hypothetical combination of Mendhekar and Gabriel.

To establish a *prima facie* case of obviousness of a claimed invention, all the claimed features must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). However, neither Mendhekar nor Gabriel discloses all of the limitations of Claims 1, 10, and 19.

Specifically neither Mendhekar nor Gabriel teach or suggest systems and methods for "developing transformation processing operations to optimize media content playback ... comprising: determining capabilities of the playback devices for processing a media stream; receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following: a selected compression format of the media content; a selected bit rate of the media content; and an image resolution of the media content; ... determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations," as explained above.

On the contrary, Claims 1, 10, and 19, recite these limitations and, for at least this reason, Claims 1, 10, and 19 are not rendered obvious by a hypothetical combination of Mendhekar and Gabriel.

Claim Rejections under 35 U.S.C. § 103 - Mendhekar and Zhang

Also within the Office Action, Claims 10-21 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Mendhekar in view of United States Patent Number 7,477,688 to Zhang et al. (hereinafter referred to as "Zhang"). The Applicants respectfully traverse this rejection because neither Mendhekar nor Zhang teach or suggest systems and methods for "developing transformation processing operations to optimize media content playback ... comprising: determining capabilities of the playback devices for processing a media stream; receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following: a selected compression format of the media content; a selected bit rate of the media content; and an image resolution of the media content; ... determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations," as explained above.

On the contrary, Claims 10-21 recite these limitations and, for at least this reason, Claims 1-21 are not rendered obvious by a hypothetical combination of Mendhekar and Zheng.

CONCLUSION

Applicant respectfully posits that the pending claims have been distinguished from the art of record, and that all objections to and rejections of the claims have been overcome. Accordingly, Applicant respectfully requests allowance. Should the Examiner deem it helpful he is encouraged to contact Applicant's attorney at (650) 474-8400.

Respectfully submitted,

Michael A. Glenn
Reg. No. 30,176

Customer No. 22862

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor:	Sean Barger
Serial No.:	12/238,842
Filed:	September 26, 2008
Art Unit:	2178
Examiner:	Christopher Bryant
Title:	AUTOMATED MEDIA DELIVERY SYSTEM
Attorney Docket No.:	EQUI0016D

July 28, 2010

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

Examiner:

This Information Disclosure Statement is submitted under 37 CFR 1.97(b).

The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 07-1445 (Order No. EQUI0016D).

Applicant(s) submit herewith PTO Form 1449 (Modified) -- Information Disclosure Citation together with copies of patents, publications or other information of which applicant(s) are aware, which applicant(s) believe(s) may be material to the examination of this application and for which there may be a duty to disclose in accordance with 37 CFR 1.56.

Each of the Claims in the originally-filed Application is copied from United States patent publication no. 2008/0205389 to Fang et al. (hereinafter referred to as "Fang").

37 C.F.R. § 1.607 and MPEP § 2001.06(d) require an Applicant to identify the patent and the numbers of the patent claims of the copied patent application. The Applicants identify how the Claims are mapped to Fang as follows:

Claim 1 corresponded to Claim 1 of Fang;

Claim 2 corresponded to Claim 2 of Fang;

Claim 3 corresponded to Claim 3 of Fang;

Claim 4 corresponded to Claim 4 of Fang;

Claim 5 corresponded to Claim 5 of Fang;

Claim 6 corresponded to Claim 6 of Fang;

Claim 7 corresponded to Claim 7 of Fang;

Claim 8 corresponded to Claim 8 of Fang;

Claim 9 corresponded to Claim 9 of Fang;

Claim 10 corresponded to Claim 10 of Fang;

Claim 11 corresponded to Claim 11 of Fang;

Claim 12 corresponded to Claim 12 of Fang;

Claim 13 corresponded to Claim 13 of Fang;

Claim 14 corresponded to Claim 14 of Fang;

Claim 15 corresponded to Claim 15 of Fang;

Claim 16 corresponded to Claim 16 of Fang;

Claim 17 corresponded to Claim 17 of Fang;

Claim 18 corresponded to Claim 18 of Fang;

Claim 19 corresponded to Claim 19 of Fang;

Claim 20 corresponded to Claim 20 of Fang; and

Claim 21 corresponded to Claim 21 of Fang.

It is requested that the information disclosed herein be made of record in this application.

Respectfully Submitted,

Michael A. Glenn
Reg. No. 30,176

Customer No. 22862

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor:	Sean Barger
Serial No.:	12/238,842
Filed:	September 26, 2008
Art Unit:	2178
Examiner:	Christopher Bryant
Title:	AUTOMATED MEDIA DELIVERY SYSTEM
Attorney Docket No.:	EQUI0016D

July 28, 2010

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATION FOR
INFORMATION DISCLOSURE STATEMENT
UNDER 37 CFR §1.97(e)

Dear Examiner:

A certification is being made for the Information Disclosure Statement accompanying this certification.

CERTIFICATION

I, the person signing below, certify

( ) that each item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the statement. (37 CFR 1.97(e)(1))

(**X**) that no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application or to the knowledge of the person signing the certification after making reasonable inquiry, was known to any individual designated in §1.56(c) more than three months prior to the filing of the statement. 37 CFR 1.97(e)(2).

The person making this certification is the attorney who signs below on the basis of the information:

(X) supplied by the inventor(s)

( ) supplied by an individual designated in §1.56(c)

( ) in the attorney's file

Respectfully submitted,

Michael A. Glenn
Reg. No. 30,176

Customer Number 22862

Form 1449 (Modified)	Serial No.: 12/238,842
	Atty. Docket No.: EQUI0016D
Information Disclosure	Applicant: Sean Barger
Statement By Applicant	Art Unit: 2178
	Confirmation No.: 2317
(Use Several Sheets if Necessary)	Filing Date: September 26, 2008

U.S. Patent Documents

Examiner Initials	No.	Patent No.	Issue Date	Patentee	Filing Date

Published U.S, Patent Application

Examiner Initials	No.	Publication No.	Publication Date	Applicant
	1	2008/0205389		Fang

Foreign Patent or Published Foreign Patent Application

Examiner Initials	No.	Document No.	Publication Date	Applicant

Non-Patent Literature Documents

Examiner Initials	No.	Author, Title, Date, Place (e.g. Journal) of Publication

Examiner's Signature:________________________ Date:______________

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Electronic Acknowledgement Receipt

EFS ID:	8111154
Application Number:	12238842
International Application Number:	
Confirmation Number:	2317
Title of Invention:	Automated Media Delivery System
First Named Inventor/Applicant Name:	Sean Barger
Customer Number:	22862
Filer:	Michael Glenn/Jessica Pallach
Filer Authorized By:	Michael Glenn
Attorney Docket Number:	EQUI0016D
Receipt Date:	28-JUL-2010
Filing Date:	26-SEP-2008
Time Stamp:	18:39:18
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		ResponseIDS-EQUI0016D--efiled072810.pdf	681731 ca16b8178bdcab544282baaa3c15f5db0680d640	yes	20

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Miscellaneous Incoming Letter	1	1
	Amendment/Req. Reconsideration-After Non-Final Reject	2	2
	Claims	3	10
	Applicant Arguments/Remarks Made in an Amendment	11	14
	Information Disclosure Statement (IDS) Filed (SB/08)	15	20
Warnings:			
Information:			
	Total Files Size (in bytes):	681731	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/21 (09-08)
Approved for use through 10/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	12/238,842
Filing Date	September 26, 2008
First Named Inventor	Sean Barger
Art Unit	2178
Examiner Name	Christopher Bryant
Total number of pages including transmittal form	
Attorney Docket Number	EQUI0016D

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☑ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☑ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) ____________
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) (please Identify below):

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Glenn Patent Group		
Signature			
Printed name	Michael A. Glenn		
Date	July 28, 2010	Reg. No.	30,176

Certificate of Electronic Filing

I hereby certify that this correspondence is being electronically transmitted to the USPTO via EFS-Web on the date shown below:

Signature			
Typed or printed name	Jessica Pallach	Date	July 28, 2010

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

Application or Docket Number	Filing Date	
12/238,842	09/26/2008	☐ To be Mailed

APPLICATION AS FILED – PART I

SMALL ENTITY ☒ OR OTHER THAN SMALL ENTITY

FOR	NUMBER FILED (Column 1)	NUMBER EXTRA (Column 2)	RATE ($)	FEE ($)		RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
			TOTAL			TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

SMALL ENTITY OR OTHER THAN SMALL ENTITY

AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
07/28/2010	Total (37 CFR 1.16(i))	* 17	Minus	** 21	= 0	X $26 =	0	OR	X $ =	
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $110 =	0	OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE	**0**	OR	TOTAL ADD'L FEE	

AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/VENESSA JONES/

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/238,842	09/26/2008	Sean Barger	EQUI0016D	2317

22862 7590 10/29/2010
GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

EXAMINER	
BRYANT, CHRISTOPHER D	
ART UNIT	**PAPER NUMBER**
2178	
NOTIFICATION DATE	**DELIVERY MODE**
10/29/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eptomatters@glenn-law.com

PTOL-90A (Rev. 04/07)

Office Action Summary	Application No.	Applicant(s)	
	12/238,842	BARGER ET AL.	
	Examiner	Art Unit	
	CHRISTOPHER BRYANT	2178	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 7/28/2010.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5, 7, 9-10, 12-19, and 21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *19* is/are allowed.

6)☒ Claim(s) *1-3, 7, 9-10, and 14-18* is/are rejected.

7)☒ Claim(s) *4,5,12,13 and 21* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 September 2008* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *7/28/2010*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed on 9/26/2008, and amended on 7/28/2010. ***This action is made Final.*** Claims 1-5, 7, 9-10, 12-19, and 21 are pending in the case. Claims 1, 10, and 19 are independent claims.

Information Disclosure Statement

2. The information disclosure statement's (IDS) submitted on 2/10/2009, 4/15/2009, and 7/28/2010 has been entered, and considered by the Examiner.

Priority

3. Applicant has made for claim of priority from application 11/269,916 filed 11/7/2005.

Drawings

4. The drawings filed on 5/12/2006 have been accepted by the Examiner.

Allowable Subject Matter

5. Claims 19 and 21 are allowed.

6. Claims 4-5, and 12-13 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

7. Claim 21 is objected to because of its dependence on canceled claim 20, the examiner recommends changing the dependency to claim 19.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3, 7, and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mendhekar et al. (United States Patent 6,938,073), hereinafter, referenced as Mendhekar; in view of Leinonen et al. (United States Patent Application Publication 2005/0278794), hereinafter, referenced as Leinonen; in further view of Fanning (United States Patent Application Publication 2006/0127059), hereinafter, referenced as Fanning; in further view of Gabriel et al. (United States Patent Application Publication US 2006/0015580), hereinafter, referenced as Gabriel.

Regarding claim 1, Mendhekar discloses:

receiving requests from the multiple devices for concurrent playback of media content at a first quality level, "that automatically modify an application in view of one or more of the following: the application, and the capability of the client requesting the application and the transmission medium delivering the application" (pg.2, lines 10-14);

where the quality level is measured in terms of:

a selected bit rate of the media content, "Rate-distortion requirement, or the compromise between bandwidth versus quality" (column 7, lines 0-12).

However the art does not disclose where the quality level is measured in a selected compression format of the media content. However the examiner maintains that it was well known in the art to have a compression format as taught by Leinonen.

In a similar field of endeavor Leinonen discloses, **"media files may be compressed according to any of various current and developing standards, such as MP-3, MP-4, and AAC for music files, JPEG and Bitmap for image files, MPEG-4 and H.263 for video files and the like. For the purposes used herein, a change in the compression format of an underlying media file is not a change to the substantive media file unless otherwise stipulated, despite differences in playback quality that may be inherent in the changed compression format" (pg.2, paragraph 22).**

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar, by allowing for a media to be presented in a compressed format, as taught by Leinonen, for the purpose of providing media to a user at a different quality.

Mendhekar and Leinonen teaches everything claimed above; however the art does not disclose where the quality level is measured in terms of an image resolution of the media content. However, the examiner maintains that it was well known in the art to have an image resolution regarding media content, as taught by Fanning.

In a similar field of endeavor Fanning discloses determining capabilities of the playback devices for processing a media stream, **"the media server 410 further includes a secondary low -quality encoder 414 that uses the same information in the content storage unit 412 to generate an encoded, low -quality representation of the content (e.g., a low -resolution image frame). The transmitter 418 also transmits these low-quality frames to a media player 420 through a network 430. Note that a low-quality frame may be transmitted to the media player 420 before the associated high-quality frame (e.g., the corresponding frame that represent the same image from the content storage unit 412). For example, the transmitter 418 might multiplex the two streams by including a high-quality frame several seconds after an associated low-quality frame has been inserted. Thus, a redundant, time-shifted, low-quality version of the content may be provided to the media player 420"** (pg.2, paragraph 20).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar and Leinonen, by presenting media content with a different resolution, as taught by Fanning, for the purpose of providing media to customers based on their device.

Mendhekar also discloses:

determining a set of independent transformations of the media content that fulfill the requests at the first quality level, "each of these clients has different capabilities connected to different transmission media" (column 2, lines 18-21);

if transformations are required, determining whether processing resources available on the host computer are sufficient to perform the independent transformations, "the server computer automatically modifies the specific application in view of the resources available so as to maximize the benefits of the resources" (column 2, lines 22-24);

if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer, "a term with an application appended by a client designates an appliance-specific transducer 150 for that application and that client. For example, a DocTV designates an appliance specific transducer for modifying a document for television viewing. This transducer can include a number of sub-transducers, such as a format sub-transducer for TV to change the format of the image depending on the television. This can be changing the size of the text so that a user six feet away can see the text. So there can be one or more of such sub-transducers to adapt the application for television viewing. Note that television is typically watched six or more feet away, while a computer monitor is less than one foot away from a viewer. Other sub-transducers, depending on needs, can be a text sub-transducer to change the text characteristics based upon the client, an image sub-transducer to change the image characteristics based upon the client's display capabilities, and a user-interface sub-transducer to enhance the usability of the application on the client. The appliance-specific

transducer 150 can add one or more sub-transducers dynamically to modify the application accordingly" (column 5, lines 34-49, corresponding to Fig.4);

however the art does not disclose determining capabilities of the playback devices for processing a media stream; determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations. However, the examiner maintains that it was well known in the art to check a users playback devices before streaming, as taught by Gabriel.

In a similar field of endeavor Gabriel,

discloses determining capabilities of the playback devices for processing a media stream, "multimedia encoding formats typically also include support for scalability. For example, video scalability offers a set of tools by which video can be coded at different resolutions (e.g., different scales) in a single bitstream. On the decoder side, video can be decoded at a suitable resolution by extracting the portion of the bitstream that represents a particular resolution. Among other advantages, scalability adds compatibility with devices of different capability but usually at the cost of some bitstream overhead" (pg.1, paragraph 11);

determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations, "multimedia content distribution facility (MCDF) that supports

device-specific delivery of multimedia content to a user's playback device as a function of the playback device's capabilities" (pg.2, paragraph 26);

and if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations, "Consider further the same subscriber with two HD digital set-tops, one in the living room and one in the bedroom, but where the one in the living room is connected to an HD display and the one in the bedroom is connected to a display with only SD capability. In this case, assuming the HD digital set-top in the bedroom can down-convert the HD version of the movie to SD before outputting the decoded output to the SD display, the subscriber might be able to purchase the movie and watch the first half in the living room in full HD resolution and still watch the second half of the movie in the bedroom, albeit at a reduced resolution" (pg.2, paragraph 16).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar, Leinonen, and Fanning by checking a users playback devices before streaming, as taught by Gabriel, for the purpose of providing media to customers based on their device.

Regarding claim 2, Mendhekar discloses performing independent transformations, "the application can be of an interactive type, such as a web-page, which a user can interact with. This embodiment includes a server apparatus in a server. The server apparatus includes an appliance-specific transducer and an adaptive transmission transducer. The appliance-specific transducer, in view of a client's request for an interactive application in the server, is configured to modify the application based on the client and the application to generate an appliance-specific output. This output is modified automatically depending on the client and the application itself. The adaptive transmission transducer, which is coupled to the appliance-specific transducer, is configured to modify the appliance-specific output. The modification is again based on the application and the corresponding medium of transmission" (column 2, lines 25-35).

Regarding claim 3, Mendhekar discloses performing the dependent transformations, "the modification, an adapted output is generated, and is delivered through the medium to the client. Then, the client decodes the adapted output to produce a modified version of the interactive application that is adapted for the client" (column 2, lines 40-43).

Regarding claim 7, Mendhekar discloses where the second quality level is lesser than the first quality level, "Rate-distortion requirement, or the compromise between bandwidth versus quality. (c) The transmission medium 154, which affects the bandwidth, error rate and latency. (d) Capabilities of the server and the client, such as the processing power and memory available" (column 7, lines 0-12).

Regarding claim 9, Mendhekar discloses having instructions for performing a computer process implementing the method of 1, "apparatus that automatically modify an application in view of one or more of the following: the application, and the capability of the client requesting the application and the transmission medium delivering the application" (column 2, lines 10-12).

Claims 10, and 14-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mendhekar in view of Leinonen; in further view of Fanning; in further view of Gabriel; in further view of Zhang et al. (United States Patent 7,477,688), hereinafter, referenced as Zhang.

Regarding claim 10, Mendhekar, Leinonen, Fanning, and Gabriel teaches everything claimed above, in addition Mendhekar discloses **determining a set of independent transformations of the media content that fulfill the requests at the first quality level, "each of these clients has different capabilities connected to different transmission media"** (column 2, lines 18-21);

determining whether processing resources available on the host computer are sufficient to perform the independent transformations, "the server computer automatically modifies the specific application in view of the resources available so as to maximize the benefits of the resources" (column 2, lines 22-24); and

if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer, "a term with an application appended by a client designates an appliance-specific transducer 150 for that application and that client. For example, a DocTV designates an appliance specific transducer for modifying a document for television viewing. This transducer can include a number of sub-transducers, such as a format sub-transducer for TV to change the format of the image depending on the television. This can be changing the size of the text so

that a user six feet away can see the text. So there can be one or more of such sub-transducers to adapt the application for television viewing. Note that television is typically watched six or more feet away, while a computer monitor is less than one foot away from a viewer. Other sub-transducers, depending on needs, can be a text sub-transducer to change the text characteristics based upon the client, an image sub-transducer to change the image characteristics based upon the client's display capabilities, and a user-interface sub-transducer to enhance the usability of the application on the client. The appliance-specific transducer 150 can add one or more sub-transducers dynamically to modify the application accordingly" (column 5, lines 34-49, corresponding to Fig.4);

however the art does not disclose if the processing resources available on the host computer are sufficient to perform the independent transformations, performing the independent transformations to create a requested set of media streams; and performing the dependent transformations to create a modified set of media streams; and transmitting the requested set of media streams or the modified set of media streams across the network. However, the examiner maintains that it was well known in the art to modify and transmit media based on the resources available, as taught by Zhang.

In a similar field of endeavor Zhang discloses **if the processing resources available on the host computer are sufficient to perform the independent transformations, performing the independent transformations to create a requested set of media streams, "Each of the local networks 23, 27, and 29 has a**

different available bandwidth. The available bandwidth determines the maximum bit rate that may be used between the network device 22 and each respective decoders 24, 30 and 36" (column 7, lines 10-16);

performing the dependent transformations to create a modified set of media streams, "Each of the local networks 23, 27, and 29 has a different available bandwidth. The available bandwidth determines the maximum bit rate that may be used between the network device 22 and each respective decoders 24, 30 and 36. By way of example, the local network 23 has an available bandwidth that allows a bit rate of 2.5 million bits per second (Mbps), the local network 27 has an available bandwidth that allows a bit rate of 1.5 Mbps, and the local network 29 has an available bandwidth that allows a bit rate of 2.0 Mbps" (column 7, lines 16-20); and

transmitting the requested set of media streams or the modified set of media streams across the network, "The present invention allows the compressed video stream produced by the encoder 14 to be sent across any of the channels 17, 18 and 19 to the decoders 24, 30 and 36" (column 7, lines 22-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar, Leinonen, Fanning, and Gabriel, by modifying and transmitting media based on the resources available, as taught by Zhang, for the purpose of modifying compressed bitstream provided by encoder to fit available bandwidth provided by local networks.

Regarding claim 14, Mendhekar teaches everything claimed above; however the art does not disclose transmitting across the network a modified set of media streams resulting from the dependent transformations across the network. However, the examiner maintains that it was well known in the art to modify and transmit media based on the resources available, as taught by Zhang.

In a similar field of endeavor Zhang discloses, "network device 22 may alter the modified compressed bitstream provided by the encoder 14 to fit the available bandwidth provided by the local networks 23, 27 and 29. The network device 22 may alter the modified compressed bitstream by selectively re-quantizing video data in different portions of the incoming compressed bitstream. It does so using multiple re-quantization schemes and in a timely manner that does not sacrifice real-time transmission of the video data" (column 7, lines 22-24).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar, by modifying and transmitting media based on the resources available, as taught by Zhang, for the purpose of modifying compressed bitstream provided by encoder to fit available bandwidth provided by local networks.

Regarding claim 15, Mendhekar teaches everything claimed above; however the art does not disclose storing a modified set of media streams resulting from the dependent transformations as media files on a storage device. However, the examiner maintains that it was well known in the art to store the streams, as taught by Zhang.

In a similar field of endeavor Zhang discloses, “it may employ one or more memories or memory modules (e.g., memory 861) configured to store program instructions for the network operations and other functions of the present invention described herein. The program instructions may specify an operating system and one or more applications, for example. Such memory or memories may also be configured to store data bitstreams” (column 19, lines 60-66).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Mendhekar, by storing streams, as taught by Zhang, for the purpose of saving streams.

Claim 16 contains limitations similar to claim 7; therefore claim 16 is rejected under the same limitations that claim 7 was rejected under.

Claim 17 contains limitations similar to claim 1; therefore claim 17 is rejected under the same limitations that claim 1 was rejected under.

Claim 18 contains limitations similar to claim 9; therefore claim 18 is rejected under the same limitations that claim 9 was rejected under.

Response to Arguments

10. Applicant's arguments with respect to claims 1-5, 7, 9-10, 12-19, and 21 have been considered but are moot in view of the new ground(s) of rejection. The examiner has found new art to address the limitations in the amended claims within this application.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure, “User Attention-Based Adaptation of Quality Level to Improve the Management of Real-Time Multi-Media Content Delivery and Distribution” (United States Patent 7,284,201) and "Methods for Streaming Media Data” (United States Patent 7,673,063).

12. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRISTOPHER BRYANT whose telephone number is (571)270-7260. The examiner can normally be reached on Monday-Friday 9:00 A.M. to 6:00 P.M. EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong can be reached on (571)272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-270-8260.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/CHRISTOPHER BRYANT/
Examiner, Art Unit 2178

/Joshua D Campbell/
Primary Examiner, Art Unit 2178

Notice of References Cited	Application/Control No. 12/238,842	Applicant(s)/Patent Under Reexamination BARGER ET AL.	
	Examiner CHRISTOPHER BRYANT	Art Unit 2178	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,938,073	08-2005	Mendhekar et al.	709/217
*	B	US-2005/0278794	12-2005	Leinonen et al.	726/032
*	C	US-2006/0127059	06-2006	Fanning, Blaise	386/125
*	D	US-2006/0015580	01-2006	Gabriel et al.	709/219
*	E	US-7,477,688	01-2009	Zhang et al.	375/240
*	F	US-7,284,201	10-2007	Cohen-Solal, Eric	715/738
*	G	US-7,673,063	03-2010	Xie et al.	709/231
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	12238842	BARGER ET AL.
	Examiner	**Art Unit**
	CHRISTOPHER BRYANT	2178

SEARCHED

Class	Subclass	Date	Examiner
709	219, 236	4/20/2010	cb
715	733, 738	4/20/2010	cb
709	219, 236	10/16/2010	cb
715	733, 738	10/16/2010	cb

SEARCH NOTES

Search Notes	Date	Examiner
see file EastSearchHistory	4/20/2010	cb
see file EastSearchHistory	10/16/2010	cb

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

/CHRISTOPHER BRYANT/ Examiner.Art Unit 2178	

Index of Claims

Application/Control No. 12238842	**Applicant(s)/Patent Under Reexamination** BARGER ET AL.	
Examiner CHRISTOPHER BRYANT	**Art Unit** 2178	

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	04/21/2010	10/14/2010							
	1	✓	✓							
	2	✓	✓							
	3	✓	✓							
	4	✓	O							
	5	✓	O							
	6	✓	-							
	7	✓	✓							
	8	✓	-							
	9	✓	✓							
	10	✓	✓							
	11	✓	-							
	12	✓	O							
	13	✓	O							
	14	✓	✓							
	15	✓	✓							
	16	✓	✓							
	17	✓	✓							
	18	✓	✓							
	19	✓	=							
	20	✓	-							
	21	✓	=							

Attorney Docket No.: EQUI0016D U.S. Serial No.: 12/238,842

Form 1449 (Modified)	Serial No.: 12/238,842
	Atty. Docket No.: EQUI0016D
Information Disclosure	Applicant: Sean Barger
Statement By Applicant	Art Unit: 2178
	Confirmation No.: 2317
(Use Several Sheets if Necessary)	Filing Date: September 26, 2008

U.S. Patent Documents

Examiner Initials	No.	Patent No.	Issue Date	Patentee	Filing Date

Published U.S, Patent Application

Examiner Initials	No.	Publication No.	Publication Date	Applicant
/C.B./	1	2008/0205389		Fang

Foreign Patent or Published Foreign Patent Application

Examiner Initials	No.	Document No.	Publication Date	Applicant

Non-Patent Literature Documents

Examiner Initials	No.	Author, Title, Date, Place (e.g. Journal) of Publication

Examiner's Signature: /Christopher Bryant/ Date: 10/14/2010

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.B./

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	1	"6964009".pn.	US-PGPUB; USPAT	OR	ON	2010/04/19 11:57
S2	441	barger.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:49
S3	4	S2 with sean	US-PGPUB; USPAT	OR	ON	2010/04/19 12:49
S4	49550	johnson.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:50
S5	97	S4 with steve	US-PGPUB; USPAT	OR	ON	2010/04/19 12:50
S6	24	S5 and media	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51
S7	5894	butler.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51
S8	4	S7 with matt	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51
S9	9	destremps.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51
S10	2	S9 with jerry	US-PGPUB; USPAT	OR	ON	2010/04/19 12:51
S11	4	pochron.in.	US-PGPUB; USPAT	OR	ON	2010/04/19 12:52
S12	6	brown.in. with trent	US-PGPUB; USPAT	OR	ON	2010/04/19 12:52
S13	378	bandwidth same media same modify	US-PGPUB; USPAT	OR	ON	2010/04/19 13:08

S14	214	S13 and ((downgrade or quality or lower) same media)	US-PGPUB; USPAT	OR	ON	2010/04/19 13:16
S15	12	S14 and (transform$ same media)	US-PGPUB; USPAT	OR	ON	2010/04/19 13:18
S16	122	S14 and (format$ same media)	US-PGPUB; USPAT	OR	ON	2010/04/19 13:19
S17	214	S13 and ((downgrade or quality or lower or lesser) same media)	US-PGPUB; USPAT	OR	ON	2010/04/19 13:48
S18	4446	media same lower same resolution	US-PGPUB; USPAT	OR	ON	2010/04/19 14:25
S19	0	S18 and bandwith and request and playback	US-PGPUB; USPAT	OR	ON	2010/04/19 14:25
S20	12	S18 and bandwith and request	US-PGPUB; USPAT	OR	ON	2010/04/19 14:25
S21	283	S18 and (request same media same content)	US-PGPUB; USPAT	OR	ON	2010/04/19 14:28
S22	381	request same media same lower same (resolution or bandwidth)	US-PGPUB; USPAT	OR	ON	2010/04/19 15:09
S23	268	S22 and quality	US-PGPUB; USPAT	OR	ON	2010/04/19 15:09
S24	182	S23 and (format or reformat)	US-PGPUB; USPAT	OR	ON	2010/04/19 15:10
S25	73	S24 and transform	US-PGPUB; USPAT	OR	ON	2010/04/19 15:10
S26	237	optimize same media same playback	US-PGPUB; USPAT	OR	ON	2010/04/19 16:02
S27	84	S26 and (lower same resolution)	US-PGPUB; USPAT	OR	ON	2010/04/19 16:02
S28	150	media same transfer same device same lower same quality	US-PGPUB; USPAT	OR	ON	2010/04/19 16:24

S29	64	media same transfer same device same lower same resolution	US-PGPUB; USPAT	OR	ON	2010/04/19 16:26
S30	124	transmitting same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2010/04/19 16:39
S31	55	S30 and bandwidth	US-PGPUB; USPAT	OR	ON	2010/04/19 16:39
S32	19	formatting same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2010/04/19 17:02
S33	5034	format$ same media same multiple same device	US-PGPUB; USPAT	OR	ON	2010/04/19 17:08
S34	1455	S33 and (media same stream)	US-PGPUB; USPAT	OR	ON	2010/04/19 17:09
S35	539	S34 and resolution and bandwidth	US-PGPUB; USPAT	OR	ON	2010/04/19 17:16
S36	496	S35 and network	US-PGPUB; USPAT	OR	ON	2010/04/19 17:16
S37	448	S36 and quality	US-PGPUB; USPAT	OR	ON	2010/04/19 17:16
S38	354	S37 and (bit same rate)	US-PGPUB; USPAT	OR	ON	2010/04/19 17:17
S39	183	S38 and transformation	US-PGPUB; USPAT	OR	ON	2010/04/19 17:19
S40	442	available same bandwidth same device same media same stream	US-PGPUB; USPAT	OR	ON	2010/04/21 13:08
S41	442	S40 same device	US-PGPUB; USPAT	OR	ON	2010/04/21 13:10
S42	61	S41 and (resolution same different)	US-PGPUB; USPAT	OR	ON	2010/04/21 13:17
S43	1	"7477688".pn.	US-PGPUB; USPAT	OR	ON	2010/04/21 14:32

S44	31	709/236.ccls. and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2010/04/21 17:05
S45	140	709/219.ccls. and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2010/04/21 17:05
S46	3	715/733.ccls. and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2010/04/21 17:06
S47	24	scaling with media with capabilities and bandwidth	US-PGPUB; USPAT	OR	ON	2010/10/12 17:05
S48	1	(bit same rate) and compresssion and (image same resolution) and quality and media	US-PGPUB; USPAT	OR	ON	2010/10/13 19:27
S49	36115	compression same format	US-PGPUB; USPAT	OR	ON	2010/10/13 19:32
S50	7010	S49 and (media same stream)	US-PGPUB; USPAT	OR	ON	2010/10/13 19:33
S51	1092	S49 same (media same stream)	US-PGPUB; USPAT	OR	ON	2010/10/13 19:33
S52	624	S51 and (quality and media and content)	US-PGPUB; USPAT	OR	ON	2010/10/13 19:33
S53	27	S52 and "715".clas.	US-PGPUB; USPAT	OR	ON	2010/10/13 19:34
S54	95	compression with format with media with quality	US-PGPUB; USPAT	OR	ON	2010/10/14 11:10
S55	990	image with resolution with media with quality	US-PGPUB; USPAT	OR	ON	2010/10/14 11:16
S56	23	S55 and "715".clas.	US-PGPUB; USPAT	OR	ON	2010/10/14 11:18

S57	91	(image with resolution with media with quality) and (stream with media)	US-PGPUB; USPAT	OR	ON	2010/10/14 11:19
S58	1	"20080205389"	US-PGPUB; USPAT	OR	ON	2010/10/14 11:35
S59	44706	transmission and media and content and quality and device and transform$	US-PGPUB; USPAT	OR	ON	2010/10/14 11:38
S60	7890	transmission and media and content and quality and device and transform$ and bit and resolution and play$	US-PGPUB; USPAT	OR	ON	2010/10/14 11:42
S61	4900	transmission and media and content and quality and device and transform$ and bit and resolution and play$ and compression and format	US-PGPUB; USPAT	OR	ON	2010/10/14 11:43
S62	4120	transmission and media and content and quality and device and transform$ and bit and resolution and play$ and compression and format and stream	US-PGPUB; USPAT	OR	ON	2010/10/14 11:44
S63	3290	transmission and media and content and quality and device and transform$ and bit and resolution and play$ and compression and format and stream and bandwidth	US-PGPUB; USPAT	OR	ON	2010/10/14 11:45
S64	131	transmission and media and content and quality and device and transform$ and bit and resolution and play$ and compression and format and stream and bandwidth and "715". clas.	US-PGPUB; USPAT	OR	ON	2010/10/14 11:48

EAST Search History (Interference)

<This search history is empty>

10/14/2010 4:28:06 PM
C:\ Documents and Settings\ cbryant2\ My Documents\ EAST\ Workspaces \ 09262008automatedmediadeliv.wsp

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL
(Submitted Only via EFS-Web)

Application Number	12/238,842	Filing Date	2008-09-26	Docket Number (if applicable)	EQUI0016D	Art Unit	2178
First Named Inventor	Sean Barger			Examiner Name	Christopher Bryant		

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV

SUBMISSION REQUIRED UNDER 37 CFR 1.114

Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

☐ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.

- ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on __________
- ☐ Other __________

☒ Enclosed

- ☒ Amendment/Reply
- ☐ Information Disclosure Statement (IDS)
- ☐ Affidavit(s)/ Declaration(s)
- ☐ Other __________

MISCELLANEOUS

☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months __________
(Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)

☐ Other __________

FEES

☒ **The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.**
The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No 071445

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

☒ Patent Practitioner Signature
☐ Applicant Signature

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Signature of Registered U.S. Patent Practitioner			
Signature	/JRW/	Date (YYYY-MM-DD)	2011-01-26
Name	Joseph R. Weatherbee	Registration Number	64810

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor	:	Sean Barger
Serial No.	:	12/238,842
Filed	:	September 26, 2008
Art Unit	:	2178
Confirmation Number	:	2317
Examiner	:	Christopher D. Bryant
Title	:	AUTOMATED MEDIA DELIVERY SYSTEM
Attorney Docket No.	:	EQUI0016D

January 26, 2011

Commissioner for Patents
Mail Stop Amendment
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO OFFICE ACTION

Applicant submits this Response to the Office Action dated October 29, 2010 in connection with the above-identified patent application.

A **Listing of Claims** begins on Page 2 of this paper, and

Remarks begin on Page 10 of this paper.

The Commissioner is authorized to charge any fees that may be due and to credit any overpayments to Deposit Account 07-1445 (Order No. EQUI0016D). Applicant considers this document to be filed in a timely manner.

LISTING OF CLAIMS

1. (Currently Amended) A method in a host computer for developing transformation processing operations to optimize media content playback across multiple playback devices connected with the host computer in a network, the method comprising:

determining capabilities of the playback devices for processing a media stream;

receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following:

a selected compression format of the media content;

a selected bit rate of the media content; and

an image resolution of the media content;

determining a set of independent transformations of the media content that fulfill the requests at the first quality level;

if transformations are required, determining whether processing resources available on the host computer are sufficient to perform the independent transformations;

if the processing resources available on the host computer are sufficient to perform the independent transformations, the method further comprises performing the independent transformations;

monitoring available bandwidth of the network; determining whether a requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network; and if the requested set of media streams is not transmissible within the available bandwidth of the network, determining the set of dependent transformations such that a modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network;

if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer;

determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and

if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations.

2. (Cancelled).

3. (Original) The method of claim 1, further comprising performing the dependent transformations.

4. (Cancelled).

5. (Currently Amended) The method of claim [[3]] 1 further comprising monitoring available bandwidth of the network; determining whether a modified set of media streams resulting from the dependent transformations is transmissible within the available bandwidth of the network; and if the modified set of media streams is not

transmissible within the available bandwidth of the network, determining a revised set of dependent transformations such that a revised set of media streams resulting from the revised set of dependent transformations is transmissible within the available bandwidth of the network.

6. (Cancelled).

7. (Original) The method of claim 1, wherein the second quality level is lesser than the first quality level.

8. (Cancelled).

9. (Original) A computer-readable medium having computer-executable instructions for performing a computer process implementing the method of claim 1.

10. (Currently Amended) A method in a host computer for developing transformation processing operations to optimize media content playback across multiple playback devices connected with the host computer in a network, the method comprising:

determining capabilities of the playback devices for processing a media stream;

receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following:

a selected compression format of the media content;

a selected bit rate of the media content; and

an image resolution of the media content;

determining a set of independent transformations of the media content that fulfill the requests at the first quality level;

determining whether processing resources available on the host computer are sufficient to perform the independent transformations; and

if the processing resources available on the host computer are sufficient to perform the independent transformations, performing the independent transformations to create a requested set of media streams;

monitoring available bandwidth of the network; determining whether the requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network; and if the requested set of media streams is not transmissible within the available bandwidth of the network, determining the set of dependent transformations such that the modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network;

if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer;

performing the dependent transformations to create a modified set of media streams; and transmitting the requested set of media streams or the modified set of media streams across the network;

determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and

if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations.

11. (Cancelled).

12. (Cancelled).

13. (Original) The method of claim 10 further comprising monitoring available bandwidth of the network; determining whether the modified set of media streams resulting from the dependent transformations is transmissible within the available bandwidth of the network; and if the modified set of media streams is not transmissible within the available bandwidth of the network, determining a revised set of dependent transformations such that a revised set of media streams resulting from the revised set of dependent transformations is transmissible within the available bandwidth of the network.

14. (Original) The method of claim 10 further comprising transmitting across the network a modified set of media streams resulting from the dependent transformations across the network.

15. (Original) The method of claim 10 further comprising storing a modified set of media streams resulting from the dependent transformations as media files on a storage device.

16. (Original) The method of claim 10, wherein the second quality level is lesser than the first quality level.

17. (Previously Presented) The method of claim 10, wherein the quality level is measured in terms of one or more of the following: a selected format of the media content, a selected bit rate of the media content, and an image resolution of the media content.

18. (Original) A computer-readable medium having computer-executable instructions for performing a computer process implementing the method of claim 10.

19. (Previously Presented) A host computer system comprising:

at least one processor operatively coupled with memory, the processor configured for determining capabilities of the playback devices for processing a media stream;

the processor further configured for performing transformative processing operations to optimize media content playback across multiple devices, which are connected with the host computer in a network,

the processor further configured for requesting concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following:

a selected compression format of the media content;

a selected bit rate of the media content; and

an image resolution of the media content;

one or more media processing units;

a policy engine module that determines a set of independent transformations of the media content that fulfill requests at the first quality level;

wherein the policy engine module determines whether processing resources of the media processing units are sufficient to perform the independent transformations;

wherein if the processing resources are sufficient to perform the independent transformations, the policy engine module directs the media processing units to perform the independent transformations to create a requested set of media streams;

wherein and if the processing resources are insufficient to perform the independent transformations, the policy engine module determines a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources;

wherein the policy engine module directs the media processing units to perform the dependent transformations to create a modified set of media streams;

a network link that communicates with the network and transmits either the requested set of media streams or the modified set of media streams to the multiple devices;

a network monitor module that monitors available bandwidth of the network and passes bandwidth information to the policy engine module; and wherein the policy engine module further determines:

whether the requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network;

if the requested set of media streams is not transmissible within the available bandwidth of the network, determines the set of dependent transformations such that the modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network; and determines whether the modified set of media streams resulting from the dependent transformations is transmissible within the available bandwidth of the network; and

if the modified set of media streams is not transmissible within the available bandwidth of the network, determining a revised set of dependent transformations such that a revised set of media streams resulting from the revised set of dependent transformations is transmissable within the available bandwidth of the network.

20. (Cancelled).

21. (Currently Amended) The system of claim [[20]] 19 further comprising a storage device that stores an output stream of transformed media content processed by the media processing units.

REMARKS

Applicants respectfully request further examination and reconsideration in view of the above amendments and arguments set forth fully below. In this Amendment and Response, Claims 1, 10, and 21 have been amended.

Allowable Subject Matter

Within the Office Action, the Examiner indicated that Claims 19 and 21 are allowed. Additionally, the Examiner indicated that Claims 4-5 and 12-13 would be allowable if re-written in independent form including all of the limitations of the base claims and any intervening claims. Accordingly, the Applicants amend the Claims 1 and 10 herein to include the subject matter identified as allowable. Accordingly, Claims 1, 3, 5, 7, 9-10, 13-19, and 21 are allowable.

Claim Rejections under 35 U.S.C. § 103 - Cutler in view of Sernec

Also within the Office Action, Claims 1-3, 7, and 9 were rejected under 35 U.S.C. § 103(a) as being unpatentable over United States Patent No. 6,938,073 to Mendhekar et al. (hereinafter referred to as "Mendhekar") in view of United States Patent publication no. 2005/0278794 to Leinonen (hereinafter referred to as "Leinonen") in view of United States Patent publication no. 2006/0127059 to Fanning (hereinafter referred to as "Fanning") and in view of United States Patent publication no. 2006/0015580 to Gabriel (hereinafter referred to as "Gabriel").

As explained above, the Applicants amend the Claims 1 and 10 herein to include the subject matter identified as allowable. Accordingly, Claims 1-3, 7, and 9 are allowable over a hypothetical of Mendhekar, Leinonen, Fanning, and Gabriel.

Also within the Office Action, Claims 10 and 14-18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over United States Patent No. 6,938,073 to Mendhekar et al. (hereinafter referred to as "Mendhekar") in view of United States Patent publication no. 2005/0278794 to Leinonen (hereinafter referred to as "Leinonen") in view of United States Patent publication no. 2006/0127059 to Fanning (hereinafter referred to as "Fanning") in view of United States Patent publication no. 2006/0015580 to Gabriel (hereinafter referred to as "Gabriel") and in view of United States Patent No. 7,477,688 to Zhang (hereinafter referred to as "Zhang").

As explained above, the Applicants amend the Claims 1 and 10 herein to include the subject matter identified as allowable. Accordingly, Claims 10 and 14-18 are allowable over a hypothetical of Mendhekar, Leinonen, Fanning, Gabriel, and Zhang.

CONCLUSION

Applicant respectfully posits that the pending claims have been distinguished from the art of record, and that all objections to and rejections of the claims have been overcome. Accordingly, Applicant respectfully requests allowance. Should the Examiner deem it helpful he is encouraged to contact Applicant's attorney at (650) 474-8400.

Respectfully submitted,

Joseph Weatherbee
Reg. No. 64,810

Customer No. 22862

Electronic Patent Application Fee Transmittal

Application Number:	12238842
Filing Date:	26-Sep-2008
Title of Invention:	Automated Media Delivery System
First Named Inventor/Applicant Name:	Sean Barger
Filer:	Michael Glenn/Jessica Pallach
Attorney Docket Number:	EQUI0016D

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Request for continued examination	1801	1	810	810
		Total in USD ($)		**810**

Electronic Acknowledgement Receipt

EFS ID:	9313816
Application Number:	12238842
International Application Number:	
Confirmation Number:	2317
Title of Invention:	Automated Media Delivery System
First Named Inventor/Applicant Name:	Sean Barger
Customer Number:	22862
Filer:	Michael Glenn/Jessica Pallach
Filer Authorized By:	Michael Glenn
Attorney Docket Number:	EQUI0016D
Receipt Date:	26-JAN-2011
Filing Date:	26-SEP-2008
Time Stamp:	16:50:12
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$810
RAM confirmation Number	4267
Deposit Account	071445
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

1		RCEresponse-EQUI0016D--efiled012611.pdf	546200 e3c7ebfc46b9f8987f193b93b5d0497809adc26d	yes	15

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Miscellaneous Incoming Letter	1	1
	Request for Continued Examination (RCE)	2	3
	Amendment Submitted/Entered with Filing of CPA/RCE	4	4
	Claims	5	12
	Applicant Arguments/Remarks Made in an Amendment	13	15

Warnings:

Information:

2	Fee Worksheet (PTO-875)	fee-info.pdf	30187 83587c92a491aa5d99baae14c3bc07afa5ecf944	no	2

Warnings:

Information:

Total Files Size (in bytes):	576387

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/21 (09-08)
Approved for use through 10/31/2008. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	12/238,842
Filing Date	September 26, 2008
First Named Inventor	Sean Barger
Art Unit	2178
Examiner Name	Christopher Bryant
Attorney Docket Number	EQUI0016D
Total number of pages including transmittal form	

ENCLOSURES *(Check all that apply)*

- [ ] Fee Transmittal Form
  - [ ] Fee Attached
- [x] Amendment/Reply
  - [x] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) ____________
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [x] Other Enclosure(s) (please Identify below):
  - RCE transmittal

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Glenn Patent Group		
Signature			
Printed name	Joseph R. Weatherbee		
Date	January 26, 2011	Reg. No.	64,810

Certificate of Electronic Filing

I hereby certify that this correspondence is being electronically transmitted to the USPTO via EFS-Web on the date shown below:

Signature			
Typed or printed name	Jessica Pallach	Date	January 26, 2011

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

Application or Docket Number	Filing Date	
12/238,842	09/26/2008	☐ To be Mailed

APPLICATION AS FILED – PART I

FOR	NUMBER FILED (Column 1)	NUMBER EXTRA (Column 2)	SMALL ENTITY ☒ RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
			TOTAL			TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

AMENDMENT

01/26/2011	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 14	Minus	** 21	= 0	X $26 =	0	OR	X $ =	
Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $110 =	0	OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE	**0**	OR	TOTAL ADD'L FEE	

AMENDMENT

	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	RATE ($)	ADDITIONAL FEE ($)	OR	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/DALE HALL/

Form 1449 (Modified)	Serial No.: 12/238,842
	Atty. Docket No.: EQUI0016D
Information Disclosure	First Named Inventor: Sean BARGER
Statement By Applicant	Art Unit: 2177
	Confirmation No.: 2317
(Use Several Sheets if Necessary)	Filing Date: September 26, 2008

U.S. Patent Documents

Examiner Initials	No.	Patent No.	Issue Date	Patentee	Filing Date
	1	5,845,279	12/1998	Garofalakis, et al.	
	2	6,483,851	11/2002	Neogi	
	3	6,909,708	06/2005	Krishnaswamy, et al.	

Published U.S, Patent Application

Examiner Initials	No.	Publication No.	Publication Date	Applicant
	4	2008/0155230	06/2008	Robbins et al
	5	2009/0070485	03/2009	Barger, et al.
	6	2009/0254672	10/2009	Zhang
	7	2010/0153495	06/2010	Barger, et al.

Foreign Patent or Published Foreign Patent Application

Examiner Initials	No.	Document No.	Publication Date	Applicant

Non-Patent Literature Documents

Examiner Initials	No.	Author, Title, Date, Place (e.g. Journal) of Publication

Examiner's Signature:______________________ Date:______________

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Electronic Acknowledgement Receipt

EFS ID:	10418713
Application Number:	12238842
International Application Number:	
Confirmation Number:	2317
Title of Invention:	Automated Media Delivery System
First Named Inventor/Applicant Name:	Sean Barger
Customer Number:	22862
Filer:	Michael Glenn/Christine Ortt
Filer Authorized By:	Michael Glenn
Attorney Docket Number:	EQUI0016D
Receipt Date:	29-JUN-2011
Filing Date:	26-SEP-2008
Time Stamp:	17:01:55
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		EQUI0016D-IDS-2011-06-29.pdf	325397 46a648a1668d7635f89461c9603e50ff02f7c2f6	yes	3

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Transmittal Letter	1	2
	Information Disclosure Statement (IDS) Form (SB08)	3	3
Warnings:			
Information:			
	Total Files Size (in bytes):	325397	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor	:	Sean BARGER
Serial No.	:	12/238,842
Filed	:	September 26, 2008
Art Unit	:	2177
Confirmation Number	:	2317
Examiner	:	David Faber
Title	:	Automated Media Delivery System
Attorney Docket No.	:	EQUI0016D

June 29, 2011

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

Examiner:

This Information Disclosure Statement is submitted:

(X) under 37 CFR 1.97(b), or
(Within three months of filing national application; or date of entry of international application; or before mailing date of first office action on the merits; whichever occurs last)

( ) under 37 CFR 1.97(c) together with either a:
( ) Certification under 37 CFR 1.97(e), or
( ) a $180.00 fee under 37 CFR 1.17(p), or
(After the CFR 1.97(b) time period, but before final action or notice of allowance, whichever occurs first)

( ) under 37 CFR 1.97(d) together with a:
( ) Certification under 37 CFR 1.97(e), and
( ) a $180.00 fee under 37 CFR 1.17(p).
(Filed after final action or notice of allowance, whichever occurs first, but before payment of the issue fee)

(X) The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 07-1445 (Order No. EQUI0016D).

(X) Applicant(s) submit herewith PTO Form 1449 (Modified) -- Information Disclosure Citation together with copies of patents, publications or other information of which applicant(s) are aware, which applicant(s) believe(s) may be material to the examination of this application and for which there may be a duty to disclose in accordance with 37 CFR 1.56.

It is requested that the information disclosed herein be made of record in this application.

Respectfully Submitted,

Michael A. Glenn
Reg. No. 30,176

Customer No. 22862

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/238,842	09/26/2008	Sean Barger	EQUI0016D	2317

22862 7590 06/08/2012
GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

EXAMINER	
FABER, DAVID	

ART UNIT	PAPER NUMBER
2177	

NOTIFICATION DATE	DELIVERY MODE
06/08/2012	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptomatters@glenn-law.com

Office Action Summary	Application No.	Applicant(s)
	12/238,842	BARGER ET AL.
	Examiner	Art Unit
	DAVID FABER	2177

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 January 2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *1,3,5,7,9,10,13-19 and 21* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☒ Claim(s) *1,3,5,7,10 and 13-17* is/are allowed.

7)☒ Claim(s) *9 and 18-21* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

10)☒ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

12)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *6/29/11*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the Request for Continued Examination filed on 26 January 2011 and the Information Disclosure Statement filed on 29 June 2011.

This office action is made Non Final.

2. Claims 1, 10, 21 have been amended.

3. Claims 2, 4, 12, 20 were cancelled.

4. All rejections of Claims 1-5, 7, 9-10, and 12-18 under 103(a) have been withdrawn as necessitated by the amendment. The allowance of Claims 19 and 21 has been withdrawn as necessitated by the new grounds of rejection applied. The objection to Claim 21 has been withdrawn as necessitated by the amendment.

5. Claims 1, 3, 5, 7, 9-10, 13-19, and 21 are pending. Claims 1, 10, and 19 are independent claims.

Information Disclosure Statement

6. The information disclosure statement (IDS) submitted on 6/29/11 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Specification

7. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: The phrase "computer readable medium" is not found to

have proper antecedent basis in the specification; however it is necessary to use this terminology in order to properly define the claim within the boundaries of statutory subject matter. In order to overcome the object, an amendment to the specification is necessary constituting a non-exhaustive statement of what the phrase "computer readable medium" would be as it would have been known to one of ordinary skill in the art at the time of the invention, in order to verify that the term "computer readable medium" could not be taken in the context of non-statutory subject matter.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

9. Claims 9 and 18 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

10. Claim 9 discloses the limitation "A computer readable medium …implementing the method of claim 1." It is unclear to the Examiner what is actually being claimed when it states “implementing the method of claim 1.” Therefore, the claim is vague and indefinite. Furthermore, since this feature is not clearly defined in the claims for the instant application, the examiner is forced to make a broad interpretation for this feature. The Examiner recommends incorporating all of the limitations from Claim 1 into Claim 9 to overcome the rejection.

11. Claim 18 discloses the limitation "A computer readable medium …implementing the method of claim 10." It is unclear to the Examiner what is actually being claimed when it states “implementing the method of claim 10.” Therefore, the claim is vague and indefinite. Furthermore, since this feature is not clearly defined in the claims for the instant application, the examiner is forced to make a broad interpretation for this feature. The Examiner recommends incorporating all of the limitations from Claim 10 into Claim 18 to overcome the rejection.

Claim Rejections - 35 USC § 101

12. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

13. Claims 9, 18-19, 21 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

14. Claims 9 and 18 disclose a computer readable storage medium; however, a computer readable storage medium is not explicitly defined in the disclosure, thus the broadest reasonable interpretation of the claims is applied

*“The broadest reasonable interpretation of a claim drawn to a computer readable medium (also called machine readable medium and other such variations) typically covers forms of non-transitory tangible media and transitory propagating signals per se in view of the ordinary and customary meaning of computer readable media, particularly when the specification is silent. See MPEP 2111.01. When the broadest reasonable interpretation of a claim covers a **signal** per se, the claim must be rejected under 35 U.S.C. § 101 as covering non-statutory subject matter. See In re Nuijten, 500 F.3d 1346, 1356-57 (Fed. Cir. 2007) (transitory embodiments are not directed to statutory subject matter) and Interim Examination Instructions for Evaluating Subject Matter Eligibility Under 35*

U.S.C. § 101, Aug. 24, 2009; p. 2."

15. Claims 19, 21 disclose a host computer system comprising a processor and memory. However, the disclosure fails to explicitly disclose or mention a computer. Nonetheless, the specification discloses a web server, wherein at times a web server is viewed as computer. However, Page 1, lines 19-21 discloses the invention relates to software systems and explicitly states using an Internet server-based software system. Therefore, a server is merely a software server. Furthermore, most figures (i.e. FIG 1) disclose a computer icon; however, the figures merely state the icon is a Client browser wherein a browser is merely software. In addition, Claim 18 discloses the host computer system comprising a processor and a memory; however the claims and the specification fail to specifically disclose if the processor and/or memory is hardware. Therefore, the processor is a software processor and the memory is merely a database. In addition, the use of the word "system" does not inherently mean that claim is directed to a physical machine. Therefore, a host computer system is not fully supported to be only hardware and is viewed of being merely a software system after reviewing Applicant's disclosure. Thus, the claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 101. The claims appear to be claiming "software systems" i.e. systems without hardware indication, which is a computer program per se; therefore, appearing non-statutory.

Any claim not specifically addressed, above, is being rejected as its failure to overcome the incorporated deficiencies of a claim upon which is depends on.

Allowable Subject Matter

16. Claims 1, 3, 5, 7, 10, 13-15 are allowed.

17. Claims 9, 18, 19, 21 would be allowable if rewritten or amended to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph and/or 35 U.S.C. 101, set forth in this Office action.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David Faber whose telephone number is 571-272-2751. The examiner can normally be reached Monday-Thursday, and every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Cesar Paula, can be reached on 571-272-4128. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/David Faber/
Examiner, Art Unit 2177

/CESAR B PAULA/
Supervisory Patent Examiner, Art Unit 2177

Form 1449 (Modified)	Serial No.: 12/238,842
	Atty. Docket No.: EQUI0016D
Information Disclosure	First Named Inventor: Sean BARGER
Statement By Applicant	Art Unit: 2177
	Confirmation No.: 2317
(Use Several Sheets if Necessary)	Filing Date: September 26, 2008

U.S. Patent Documents

Examiner Initials	No.	Patent No.	Issue Date	Patentee	Filing Date
	1	5,845,279	12/1998	Garofalakis, et al.	
	2	6,483,851	11/2002	Neogi	
	3	6,909,708	06/2005	Krishnaswamy, et al.	

Published U.S, Patent Application

Examiner Initials	No.	Publication No.	Publication Date	Applicant
	4	2008/0155230	06/2008	Robbins et al
	5	2009/0070485	03/2009	Barger, et al.
	6	2009/0254672	10/2009	Zhang
	7	2010/0153495	06/2010	Barger, et al.

Foreign Patent or Published Foreign Patent Application

Examiner Initials	No.	Document No.	Publication Date	Applicant

Non-Patent Literature Documents

Examiner Initials	No.	Author, Title, Date, Place (e.g. Journal) of Publication

Examiner's Signature: /David Faber/ Date: 05/23/2012

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /DF/

Search Notes

	Application/Control No.	Applicant(s)/Patent Under Reexamination
	12238842	BARGER ET AL.
	Examiner	**Art Unit**
	DAVID FABER	2177

SEARCHED

Class	Subclass	Date	Examiner
709	219,224,236	5/23/2012	/df/
715	201,224,238,249,273,704,733,736,738,748	5/23/2012	/df

SEARCH NOTES

Search Notes	Date	Examiner
see file EastSearchHistory	5/23/2012	/df/

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner

Index of Claims	Application/Control No. 12238842	Applicant(s)/Patent Under Reexamination BARGER ET AL.
	Examiner David Faber	Art Unit 2177

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE									
Final	Original	04/21/2010	10/14/2010	05/23/2012							
	1	✓	✓	=							
	2	✓	✓	-							
	3	✓	✓	=							
	4	✓	O	-							
	5	✓	O	=							
	6	✓	-	-							
	7	✓	✓	=							
	8	✓	-	-							
	9	✓	✓	✓							
	10	✓	✓	=							
	11	✓	-	-							
	12	✓	O	-							
	13	✓	O	=							
	14	✓	✓	=							
	15	✓	✓	=							
	16	✓	✓	=							
	17	✓	✓	=							
	18	✓	✓	✓							
	19	✓	=	✓							
	20	✓	-	-							
	21	✓	=	✓							

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	6	"238842".ap. and medium	US-PGPUB; USPAT	OR	ON	2012/05/23 13:39
L2	4	"238842".ap. and memory	US-PGPUB; USPAT	OR	ON	2012/05/23 13:40
L3	6	US-5845279-$.DID. OR US-6483851-$.DID. OR US-6909708-$.DID. OR US-20080155230-$.DID. OR US-20090070485-$.DID. OR US-20090254672-$.DID.	US-PGPUB; USPAT	OR	ON	2012/05/23 13:50
L4	1	"20100153495"	US-PGPUB; USPAT	OR	ON	2012/05/23 13:51
L5	1	"238842".ap. and repository	US-PGPUB; USPAT	OR	ON	2012/05/23 13:52
L6	3	"238842".ap. and software	US-PGPUB; USPAT	OR	ON	2012/05/23 13:52
L7	1	"6964009".pn.	US-PGPUB; USPAT	OR	ON	2012/05/23 14:18
L8	496	barger.in.	US-PGPUB; USPAT	OR	ON	2012/05/23 14:18
L9	6	8 and sean	US-PGPUB; USPAT	OR	ON	2012/05/23 14:18
L10	31	johnson.in. with steve and media	US-PGPUB; USPAT	OR	ON	2012/05/23 14:19
L11	5	butler.in. with matt	US-PGPUB; USPAT	OR	ON	2012/05/23 14:19
L12	1746	bandwidth same media same modif$5	US-PGPUB; USPAT	OR	ON	2012/05/23 14:19
L13	817	12 and (downgrade quality lower) same media	US-PGPUB; USPAT	OR	ON	2012/05/23 14:22
L14	817	12 and ((downgrade quality lower) same media)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:22
L15	128	14 and (transform$4 same media)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:23
L16	937	12 and (downgrade quality lower less$3) same	US-	OR	ON	2012/05/23

		media	PGPUB; USPAT			14:23
L17	15343	media same low$4 same resolution	US-PGPUB; USPAT	OR	ON	2012/05/23 14:24
L18	843	17 and bandwidth and request$4 and playback	US-PGPUB; USPAT	OR	ON	2012/05/23 14:24
L19	2028	17 and bandwidth and request$4	US-PGPUB; USPAT	OR	ON	2012/05/23 14:26
L21	949	17 and (request$4 same media same content)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:27
L22	788	request$4 same media same lower same (resolution or bandwidth)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:28
L23	593	22 and quality	US-PGPUB; USPAT	OR	ON	2012/05/23 14:28
L24	436	23 and (format$4 reformat$4)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:28
L25	176	24 and transform$6	US-PGPUB; USPAT	OR	ON	2012/05/23 14:28
L26	797	optimiz$4 same media same playback	US-PGPUB; USPAT	OR	ON	2012/05/23 14:29
L27	141	26 and (lower same resolution)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:29
L28	246	media same transfer$5 same device same lower same quality	US-PGPUB; USPAT	OR	ON	2012/05/23 14:29
L29	103	media same transfer$5 same device same lower same resolution	US-PGPUB; USPAT	OR	ON	2012/05/23 14:30
L30	500	transmit$5 same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2012/05/23 14:30
L31	265	30 and bandwidth	US-PGPUB; USPAT	OR	ON	2012/05/23 14:30
L32	738	format$4 same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2012/05/23 14:31
L33	7102	format$4 same media same multiple same device	US-PGPUB; USPAT	OR	ON	2012/05/23 14:31
L34	2120	33 and (media same stream)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:31
L36	778	34 and resolution and bandwidth	US-PGPUB; USPAT	OR	ON	2012/05/23 14:32
L37	731	36 and network	US-	OR	ON	2012/05/23

			PGPUB; USPAT			14:32
L38	656	37 and quality	US-PGPUB; USPAT	OR	ON	2012/05/23 14:32
L39	480	37 and (bitrate or bit adj rate)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:32
L40	385	39 and transform$6	US-PGPUB; USPAT	OR	ON	2012/05/23 14:32
L41	671	available same bandwidth same device same media same stream	US-PGPUB; USPAT	OR	ON	2012/05/23 14:33
L42	671	41 same device	US-PGPUB; USPAT	OR	ON	2012/05/23 14:33
L43	105	42 and (resolution same different)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:33
L44	2599	715/733,738.ccls.	US-PGPUB; USPAT	OR	ON	2012/05/23 14:33
L45	11516	709/219,236.ccls.	US-PGPUB; USPAT	OR	ON	2012/05/23 14:34
L46	26942	709/219,224,236.ccls.	US-PGPUB; USPAT	OR	ON	2012/05/23 14:42
L47	7959	715/201,224,238,249,273,704,733,736,738,748.ccls.	US-PGPUB; USPAT	OR	ON	2012/05/23 14:44
L48	59	47 and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2012/05/23 14:46
L49	3	47 and optimiz$4 playback media	US-PGPUB; USPAT	WITH	ON	2012/05/23 14:50
L50	3	(US-20120011014-$ or US-20090089422-$).did. or (US-7155676-$).did.	US-PGPUB; USPAT	OR	OFF	2012/05/23 14:50
L51	0	47 and optimiz$4 playback video	US-PGPUB; USPAT	WITH	ON	2012/05/23 14:50
L52	11	47 and optimiz$4 video device	US-PGPUB; USPAT	WITH	ON	2012/05/23 14:51
L53	11	(US-20120110317-$ or US-20120066608-$ or US-20120011014-$ or US-20100245107-$ or US-20100153853-$ or US-20090225164-$ or US-20090070682-$ or US-20090070681-$ or US-20080270890-$ or US-20040177323-$).did. or (US-7287220-$).did.	US-PGPUB; USPAT	OR	OFF	2012/05/23 14:51
L54	14	47 and optimiz$4 media device	US-PGPUB; USPAT	WITH	ON	2012/05/23 14:52
L55	14	(US-20120066601-$ or US-20120060096-$ or US-	US-	OR	OFF	2012/05/23

		20120011014-$ or US-20110179356-$ or US-20110060997-$ or US-20090089422-$ or US-20050108775-$ or US-20050060640-$ or US-20040177323-$ or US-20040049733-$ or US-20020169797-$).did. or (US-7944456-$ or US-7934155-$ or US-7287220-$).did.	PGPUB; USPAT			14:52
L56	5	47 and optimiz$4 media and (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/05/23 14:52
L57	5	(US-20120066601-$ or US-20110060998-$ or US-20090265617-$ or US-20090063983-$).did. or (US-7555715-$).did.	US-PGPUB; USPAT	OR	OFF	2012/05/23 14:52
L58	369	optimiz$4 media and (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/05/23 14:52
L59	104	optimiz$4 media device and (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/05/23 14:53
L60	38	optimiz$4 media device SAMe (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/05/23 14:54

EAST Search History (Interference)

<This search history is empty>

5/23/2012 2:55:19 PM
C:\Users\dfaber\Documents\EAST\Workspaces\12238842.wsp

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor	:	Sean Barger
Serial No.	:	12/238,842
Filed	:	September 26, 2008
Art Unit	:	2177
Confirmation Number	:	2317
Examiner	:	David Faber
Title	:	AUTOMATED MEDIA DELIVERY SYSTEM
Attorney Docket No.	:	EQUI0016D

August 30, 2012

Mail Stop AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO OFFICE ACTION

Applicant submits this Response to the Office Action dated June 8, 2012 in connection with the above-identified patent application.

A **Listing of Claims** begins on Page 2 of this paper, and

Remarks begin on Page 8 of this paper.

The Commissioner is authorized to charge any fees that may be due and to credit any overpayments to Deposit Account 07-1445 (Order No. EQUI0016D). Applicant considers this document to be filed in a timely manner.

LISTING OF CLAIMS

1. (Previously Presented) A method in a host computer for developing transformation processing operations to optimize media content playback across multiple playback devices connected with the host computer in a network, the method comprising:

determining capabilities of the playback devices for processing a media stream;

receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following:

a selected compression format of the media content;

a selected bit rate of the media content; and

an image resolution of the media content;

determining a set of independent transformations of the media content that fulfill the requests at the first quality level;

if transformations are required, determining whether processing resources available on the host computer are sufficient to perform the independent transformations;

if the processing resources available on the host computer are sufficient to perform the independent transformations, the method further comprises performing the independent transformations;

monitoring available bandwidth of the network; determining whether a requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network; and if the requested set of media streams is not transmissible within the available bandwidth of the network, determining the set of dependent transformations such that a modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network;

if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer;

determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and

if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations.

2. (Cancelled).

3. (Previously Presented) The method of claim 1, further comprising performing the dependent transformations.

4. (Cancelled).

5. (Previously Presented) The method of claim 1 further comprising monitoring available bandwidth of the network; determining whether a modified set of media streams resulting from the dependent transformations is transmissible within the

available bandwidth of the network; and if the modified set of media streams is not transmissible within the available bandwidth of the network, determining a revised set of dependent transformations such that a revised set of media streams resulting from the revised set of dependent transformations is transmissible within the available bandwidth of the network.

6. (Cancelled).

7. (Original) The method of claim 1, wherein the second quality level is lesser than the first quality level.

8. – 9. (Cancelled).

10. (Previously Presented) A method in a host computer for developing transformation processing operations to optimize media content playback across multiple playback devices connected with the host computer in a network, the method comprising:

determining capabilities of the playback devices for processing a media stream;

receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following:

a selected compression format of the media content;

a selected bit rate of the media content; and

an image resolution of the media content;

determining a set of independent transformations of the media content that fulfill the requests at the first quality level;

determining whether processing resources available on the host computer are sufficient to perform the independent transformations; and

if the processing resources available on the host computer are sufficient to perform the independent transformations, performing the independent transformations to create a requested set of media streams;

monitoring available bandwidth of the network; determining whether the requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network; and if the requested set of media streams is not transmissible within the available bandwidth of the network, determining the set of dependent transformations such that the modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network;

if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer;

performing the dependent transformations to create a modified set of media streams; and transmitting the requested set of media streams or the modified set of media streams across the network;

determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and

if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations.

11. – 12. (Cancelled).

13. (Original) The method of claim 10 further comprising monitoring available bandwidth of the network; determining whether the modified set of media streams resulting from the dependent transformations is transmissible within the available bandwidth of the network; and if the modified set of media streams is not transmissible within the available bandwidth of the network, determining a revised set of dependent transformations such that a revised set of media streams resulting from the revised set of dependent transformations is transmissible within the available bandwidth of the network.

14. (Original) The method of claim 10 further comprising transmitting across the network a modified set of media streams resulting from the dependent transformations across the network.

15. (Original) The method of claim 10 further comprising storing a modified set of media streams resulting from the dependent transformations as media files on a storage device.

16. (Original) The method of claim 10, wherein the second quality level is lesser than

the first quality level.

17. (Previously Presented) The method of claim 10, wherein the quality level is measured in terms of one or more of the following: a selected format of the media content, a selected bit rate of the media content, and an image resolution of the media content.

18. – 21. (Cancelled)

REMARKS

Allowable Subject Matter

Applicant thanks the Examiner for indicating that Claims 1, 3, 5, 7, 10, and 13-17 are allowed. Applicant cancels herein all non-allowed claims, thus placing the application in allowable condition.

In particular, Claims 9 and 18 are cancelled, thereby obviating the objection to the Specification and the Claim Rejections under 35 USC § 112 ¶ 2 and, with regard to Claims 9 and 18, 35 USC § 101. Further, Claims 19 and 21 are cancelled, thereby obviating the rejection thereof under 35 USC § 101.

The cancellation of Claim 9, 18, 19, and 21 is made for purposes of expedience only. Applicant expressly reserves the right to pursue claim coverage of a similar scope in a further submission to the Patent Office at a future date.

Should the Examiner deem it helpful he is encouraged to contact Applicant's attorney, Michael A. Glenn, at (650) 474-8400.

Respectfully submitted,

Michael A. Glenn
Reg. No. 30,176

Customer No. 22862

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	12/238,842
	Filing Date	Sep 26, 2008
	First Named Inventor	Sean Barger
	Art Unit	2177
	Examiner Name	Faber, David
	Attorney Docket Number	EQUI0016D

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	1	US-6311185		Oct 2001	Markowitz et al.	
	2	US-6463445		Oct 2002	Suzuki et al.	
	3	US-7313361		Dec 2007	Steelberg et al.	
	4	US-7406434		Jul 2008	Chang et al.	

U.S.PATENT APPLICATION PUBLICATIONS

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	5	US-20030225568		Dec 2003	Salmonsen	
	6	US-20040025176		Feb 2004	Franklin et al.	
	7	US-20050255852		Nov 2005	Steelberg et al.	
	8	US-20070061198		Mar 2007	Ramer et al.	
	9	US-20080195938		Aug 2008	Tischer et al.	
	10	US-20080207182		Aug 2008	Maharajh et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	12/238,842
	Filing Date	Sep 26, 2008
	First Named Inventor	Sean Barger
	Art Unit	2177
	Examiner Name	Faber, David
	Attorney Docket Number	EQUI0016D

	11	US-20090013347		Jan 2009	Ahanger et al.	
	12	US-20090240569		Sep 2009	Ramer et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]

NON-PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[5]

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document.
[4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	12/238,842
	Filing Date	Sep 26, 2008
	First Named Inventor	Sean Barger
	Art Unit	2177
	Examiner Name	Faber, David
	Attorney Docket Number	EQUI0016D

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e) (1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e) (2).

☐ See attached certification statement.

☒ Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ No certification statement submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	[signature]	Date (YYYY-MM-DD)	2012-08-30
Name/Print	Michael A Glenn	Registration Number	30176

Electronic Patent Application Fee Transmittal

Application Number:	12238842
Filing Date:	26-Sep-2008
Title of Invention:	Automated Media Delivery System
First Named Inventor/Applicant Name:	Sean Barger
Filer:	Michael Glenn/Christine Ortt
Attorney Docket Number:	EQUI0016D

Filed as Small Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Submission- Information Disclosure Stmt	1806	1	180	180
		Total in USD ($)		**180**

Electronic Acknowledgement Receipt

EFS ID:	13632448
Application Number:	12238842
International Application Number:	
Confirmation Number:	2317
Title of Invention:	Automated Media Delivery System
First Named Inventor/Applicant Name:	Sean Barger
Customer Number:	22862
Filer:	Michael Glenn/Christine Ortt
Filer Authorized By:	Michael Glenn
Attorney Docket Number:	EQUI0016D
Receipt Date:	30-AUG-2012
Filing Date:	26-SEP-2008
Time Stamp:	19:20:45
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$180
RAM confirmation Number	8931
Deposit Account	071445
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		EQUI0016D-Response_IDS-2012-08-30.pdf	719055 454ee18d9c6e518af09127bc85562b959bfb2d6a	yes	11

Multipart Description/PDF files in .zip description

Document Description	Start	End
Amendment/Req. Reconsideration-After Non-Final Reject	1	1
Claims	2	7
Applicant Arguments/Remarks Made in an Amendment	8	8
Information Disclosure Statement (IDS) Form (SB08)	9	10
Transmittal Letter	11	11

Warnings:

Information:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
2	Fee Worksheet (SB06)	fee-info.pdf	29999 efc5ba9d3c20539270eb2efca7ff37e6e2e1eb07	no	2

Warnings:

Information:

Total Files Size (in bytes):	749054

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

Application or Docket Number: 12/238,842
Filing Date: 09/26/2008
☐ To be Mailed

APPLICATION AS FILED – PART I

FOR	NUMBER FILED (Column 1)	NUMBER EXTRA (Column 2)	SMALL ENTITY ☒ RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
			TOTAL			TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

AMENDMENT 08/30/2012	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 10	Minus	** 21	= 0	X $30 =	0	OR	X $ =	
Independent (37 CFR 1.16(h))	* 2	Minus	***3	= 0	X $125 =	0	OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE	**0**	OR	TOTAL ADD'L FEE	

AMENDMENT	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	RATE ($)	ADDITIONAL FEE ($)	OR	RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/SUSAN FORD/

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

22862 7590 10/11/2012
GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

EXAMINER	
FABER, DAVID	
ART UNIT	PAPER NUMBER
2177	

DATE MAILED: 10/11/2012

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/238,842	09/26/2008	Sean Barger	EQUI0016D	2317

TITLE OF INVENTION: AUTOMATED MEDIA DELIVERY SYSTEM

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$885	$300	$0	$1185	01/11/2013

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

22862 7590 10/11/2012

GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/238,842	09/26/2008	Sean Barger	EQUI0016D	2317

TITLE OF INVENTION: AUTOMATED MEDIA DELIVERY SYSTEM

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$885	$300	$0	$1185	01/11/2013

EXAMINER	ART UNIT	CLASS-SUBCLASS
FABER, DAVID	2177	715-200000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
   - ❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
   - ❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
   (1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
   (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
   1 ____________
   2 ____________
   3 ____________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:
- ❑ Issue Fee
- ❑ Publication Fee (No small entity discount permitted)
- ❑ Advance Order - # of Copies ____________

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ❑ A check is enclosed.
- ❑ Payment by credit card. Form PTO-2038 is attached.
- ❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number ____________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ❑ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ❑ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ____________ Date ____________

Typed or printed name ____________ Registration No. ____________

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/238,842	09/26/2008	Sean Barger	EQUI0016D	2317

22862 7590 10/11/2012

GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

EXAMINER	
FABER, DAVID	
ART UNIT	**PAPER NUMBER**
2177	

DATE MAILED: 10/11/2012

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 534 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 534 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No. 12/238,842	Applicant(s) BARGER ET AL.	
	Examiner DAVID FABER	Art Unit 2177	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *8/30/12*.
2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
3. ☒ The allowed claim(s) is/are *1,3,5,7,10 and 13-17*.
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 8/30/12
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/CESAR B PAULA/
Supervisory Patent Examiner, Art Unit 2177

EXAMINER'S STATEMENT

1. This notice of allowance is in response to the amendment and Information Disclosure Statement filed on 30 August 2012.

2. Claims 9, 18, 19, and 21 have been cancelled.

3. All rejections from the previous office action have been withdrawn as necessitated by the amendment.

4. Claims 1, 3, 5, 7, 10, 13-17 are pending. Claims 1 and 10 are independent claims.

Information Disclosure Statement

5. The information disclosure statement (IDS) submitted on 8/30/12 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Allowable Subject Matter

6. Claims 1, 3, 5, 7, 10, 13-17 are allowed.

7. The following is an examiner's statement of reasons for allowance:

- Regarding independent claim 1, none of the references, either singularly or in combination, teach or suggest to a person of ordinary skill in the art at the time of the invention the combination of limitations of claim 1 including: "A method in a host computer for developing transformation processing operations to optimize media content playback across multiple playback devices connected with the host computer in a network, the method comprising: determining capabilities of the playback devices for processing a media stream; receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following: a selected compression format of the media content; a selected bit rate of the media content; and an image resolution of the media content; determining a set of independent transformations of the media content that fulfill the requests at the first quality level; if transformations are required, determining whether processing resources available on the host computer are sufficient to perform the independent transformations; if the processing resources available on the host computer are sufficient to perform the independent transformations, the method further comprises performing the independent transformations; monitoring available bandwidth of the network; determining whether a requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network; and if the requested set of media streams is not transmissible within the available bandwidth of the network, determining the set of dependent transformations such that a modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network; if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer; determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations."

- Regarding independent claim 10, none of the references, either singularly or in combination, teach or suggest to a person of ordinary skill in the art at the time of the invention the combination of limitations of claim 10 including: "A method in a host computer for developing transformation processing operations to optimize media content playback across multiple playback devices connected with the host computer in a network, the method comprising: determining capabilities of the playback devices for processing a media stream; receiving requests from the multiple devices for concurrent playback of media content at a first quality level, wherein the quality level is measured in terms of each of the following: a selected compression format of the media content; a selected bit rate of the media content; and an image resolution of the media content; determining a set of independent transformations of the media content that fulfill the requests at the first quality level; determining whether processing resources available on the host computer are sufficient to perform the independent transformations; and if the processing resources available on the host computer are sufficient to perform the independent transformations, performing the independent transformations to create a requested set of media streams; monitoring available bandwidth of the network; determining whether the requested set of media streams resulting from the independent transformations is transmissible within the available bandwidth of the network; and if the requested set of media streams is not transmissible within the available bandwidth of the network, determining the set of dependent transformations such that the modified set of media streams resulting from the set of dependent transformations is transmissible within the available bandwidth of the network; if the processing resources are insufficient to perform the independent transformations, determining a set of dependent transformations that fulfill the requests at a second quality level within limits of the processing resources of the host computer; performing the dependent transformations to create a modified set of media streams; and transmitting the requested set of media streams or the modified set of media streams across the network; determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the independent transformations; and if not, determining whether the capability of each of the playback devices is sufficient to process a requested media stream resulting from the dependent transformations."

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David Faber whose telephone number is 571-272-2751. The examiner can normally be reached Monday-Thursday, and every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Cesar Paula, can be reached on 571-272-4128. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/David Faber/
Examiner, Art Unit 2177

/CESAR B PAULA/

Supervisory Patent Examiner, Art Unit 2177

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	12238842	BARGER ET AL.
	Examiner	Art Unit
	DAVID FABER	2177

SEARCHED

Class	Subclass	Date	Examiner
709	219,224,236	10/3/2012	/df/
715	201,224,238,249,273,704,733,736,738,748	10/3/2012	/df

SEARCH NOTES

Search Notes	Date	Examiner
see file EastSearchHistory	10/3/2012	/df/

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner
715	201,224,238,249,273,704,733,736,738,748	10/3/2012	/DF/
709	219,224,236	10/3/2012	/DF/

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Field
PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	12/238,842
	Filing Date	Sep 26, 2008
	First Named Inventor	Sean Barger
	Art Unit	2177
	Examiner Name	Faber, David
	Attorney Docket Number	EQUI0016D

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	1	US-6311185		Oct 2001	Markowitz et al.	
	2	US-6463445		Oct 2002	Suzuki et al.	
	3	US-7313361		Dec 2007	Steelberg et al.	
	4	US-7406434		Jul 2008	Chang et al.	

U.S.PATENT APPLICATION PUBLICATIONS

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	5	US-20030225568		Dec 2003	Salmonsen	
	6	US-20040025176		Feb 2004	Franklin et al.	
	7	US-20050255852		Nov 2005	Steelberg et al.	
	8	US-20070061198		Mar 2007	Ramer et al.	
	9	US-20080195938		Aug 2008	Tischer et al.	
	10	US-20080207182		Aug 2008	Maharajh et al.	

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /DF/

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Field

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	12/238,842
	Filing Date	Sep 26, 2008
	First Named Inventor	Sean Barger
	Art Unit	2177
	Examiner Name	Faber, David
	Attorney Docket Number	EQUI0016D

	11	US-20090013347		Jan 2009	Ahanger et al.	
	12	US-20090240569		Sep 2009	Ramer et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]

NON-PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[5]

EXAMINER SIGNATURE

Examiner Signature	/David Faber/	Date Considered	10/03/2012

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document.
[4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /DF/

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	6	"238842".ap. and medium	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L2	4	"238842".ap. and memory	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L3	6	US-5845279-$.DID. OR US-6483851-$.DID. OR US-6909708-$.DID. OR US-20080155230-$.DID. OR US-20090070485-$.DID. OR US-20090254672-$.DID.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L4	1	"20100153495"	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L5	1	"238842".ap. and repository	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L6	3	"238842".ap. and software	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L7	1	"6964009".pn.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L8	508	barger.in.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L9	6	L8 and sean	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L10	33	johnson.in. with steve and media	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L11	5	butler.in. with matt	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L12	1837	bandwidth same media same modif$5	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L13	862	L12 and (downgrade quality lower) same media	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L14	862	L12 and ((downgrade quality lower) same media)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L15	135	L14 and (transform$4 same media)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L16	985	L12 and (downgrade quality lower less$3) same	US-	OR	ON	2012/10/03

		media	PGPUB; USPAT			11:01
L17	15931	media same low$4 same resolution	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L18	899	L17 and bandwidth and request$4 and playback	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L19	2134	L17 and bandwidth and request$4	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L20	1026	L17 and (request$4 same media same content)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L21	850	request$4 same media same lower same (resolution or bandwidth)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L22	647	L21 and quality	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L23	481	L22 and (format$4 reformat$4)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L24	199	L23 and transform$6	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L25	834	optimiz$4 same media same playback	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L26	147	L25 and (lower same resolution)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L27	258	media same transfer$5 same device same lower same quality	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L28	109	media same transfer$5 same device same lower same resolution	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L29	515	transmit$5 same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L30	271	L29 and bandwidth	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L31	778	format$4 same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L32	7480	format$4 same media same multiple same device	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L33	2262	L32 and (media same stream)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L34	831	L33 and resolution and bandwidth	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L35	783	L34 and network	US-	OR	ON	2012/10/03

			PGPUB; USPAT			11:01
L36	699	L35 and quality	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L37	517	L35 and (bitrate or bit adj rate)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L38	413	L37 and transform$6	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L39	721	available same bandwidth same device same media same stream	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L40	721	L39 same device	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L41	115	L40 and (resolution same different)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L42	2740	715/733,738.ccls.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L43	12129	709/219,236.ccls.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L44	28561	709/219,224,236.ccls.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L45	8331	715/201,224,238,249,273,704,733,736,738,748.ccls.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L46	59	L45 and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L47	4	L45 and optimiz$4 playback media	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L49	1	L45 and optimiz$4 playback video	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L50	11	L45 and optimiz$4 video device	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L51	11	(US-20120110317-$ or US-20120066608-$ or US-20120011014-$ or US-20100245107-$ or US-20100153853-$ or US-20090225164-$ or US-20090070682-$ or US-20090070681-$ or US-20080270890-$ or US-20040177323-$).did. or (US-7287220-$).did.	US-PGPUB; USPAT	OR	OFF	2012/10/03 11:01
L52	14	L45 and optimiz$4 media device	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L53	14	(US-20120066601-$ or US-20120060096-$ or US-20120011014-$ or US-20110179356-$ or US-20110060997-$ or US-20090089422-$ or US-20050108775-$ or US-20050060640-$ or US-20040177323-$ or US-20040049733-$ or US-	US-PGPUB; USPAT	OR	OFF	2012/10/03 11:01

		20020169797-$).did. or (US-7944456-$ or US-7934155-$ or US-7287220-$).did.				
L54	5	L45 and optimiz$4 media and (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L55	5	(US-20120066601-$ or US-20110060998-$ or US-20090265617-$ or US-20090063983-$).did. or (US-7555715-$).did.	US-PGPUB; USPAT	OR	OFF	2012/10/03 11:01
L56	392	optimiz$4 media and (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L57	111	optimiz$4 media device and (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L58	40	optimiz$4 media device SAMe (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L59	13	US-2238842-$.DID. OR US-6311185-$.DID. OR US-6463445-$.DID. OR US-7313361-$.DID. OR US-7406434-$.DID. OR US-20030225568-$.DID. OR US-20040025176-$.DID. OR US-20050255852-$.DID. OR US-20070061198-$.DID. OR US-20080195938-$.DID. OR US-20080207182-$.DID. OR US-20090013347-$.DID. OR US-20090240569-$.DID.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L60	6	"238842".ap. and medium	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L61	4	"238842".ap. and memory	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L62	6	US-5845279-$.DID. OR US-6483851-$.DID. OR US-6909708-$.DID. OR US-20080155230-$.DID. OR US-20090070485-$.DID. OR US-20090254672-$.DID.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L63	1	"20100153495"	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L64	1	"238842".ap. and repository	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L65	3	"238842".ap. and software	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L66	1	"6964009".pn.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L67	508	barger.in.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L68	6	L67 and sean	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L69	33	johnson.in. with steve and media	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L70	5	butler.in. with matt	US-PGPUB;	OR	ON	2012/10/03 11:01

			USPAT			
L71	1837	bandwidth same media same modif$5	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L72	862	L71 and (downgrade quality lower) same media	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L73	862	L71 and ((downgrade quality lower) same media)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L74	135	L73 and (transform$4 same media)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L75	985	L71 and (downgrade quality lower less$3) same media	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L76	15931	media same low$4 same resolution	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L77	899	L76 and bandwidth and request$4 and playback	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L78	2134	L76 and bandwidth and request$4	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L79	1026	L76 and (request$4 same media same content)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L80	850	request$4 same media same lower same (resolution or bandwidth)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L81	647	L80 and quality	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L82	481	L81 and (format$4 reformat$4)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L83	199	L82 and transform$6	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L84	834	optimiz$4 same media same playback	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L85	147	L84 and (lower same resolution)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L86	258	media same transfer$5 same device same lower same quality	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L87	109	media same transfer$5 same device same lower same resolution	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L88	515	transmit$5 same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L89	271	L88 and bandwidth	US-PGPUB;	OR	ON	2012/10/03 11:01

			USPAT			
L90	778	format$4 same media same lower same resolution	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L91	7480	format$4 same media same multiple same device	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L92	2262	L91 and (media same stream)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L93	831	L92 and resolution and bandwidth	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L94	783	L93 and network	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L95	699	L94 and quality	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L96	517	L94 and (bitrate or bit adj rate)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L97	413	L96 and transform$6	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L98	721	available same bandwidth same device same media same stream	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L99	721	L98 same device	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L100	115	L99 and (resolution same different)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L101	2740	715/733,738.ccls.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L102	12129	709/219,236.ccls.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L103	28561	709/219,224,236.ccls.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L104	8331	715/201,224,238,249,273,704,733,736,738,748.ccls.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L105	59	L104 and ((resolution or bandwidth) same device same stream)	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L106	4	L104 and optimiz$4 playback media	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L108	1	L104 and optimiz$4 playback video	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L109	11	L104 and optimiz$4 video device	US-PGPUB;	WITH	ON	2012/10/03 11:01

			USPAT			
L110	11	(US-20120110317-$ or US-20120066608-$ or US-20120011014-$ or US-20100245107-$ or US-20100153853-$ or US-20090225164-$ or US-20090070682-$ or US-20090070681-$ or US-20080270890-$ or US-20040177323-$).did. or (US-7287220-$).did.	US-PGPUB; USPAT	OR	OFF	2012/10/03 11:01
L111	14	L104 and optimiz$4 media device	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L112	14	(US-20120066601-$ or US-20120060096-$ or US-20120011014-$ or US-20110179356-$ or US-20110060997-$ or US-20090089422-$ or US-20050108775-$ or US-20050060640-$ or US-20040177323-$ or US-20040049733-$ or US-20020169797-$).did. or (US-7944456-$ or US-7934155-$ or US-7287220-$).did.	US-PGPUB; USPAT	OR	OFF	2012/10/03 11:01
L113	5	L104 and optimiz$4 media and (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L114	5	(US-20120066601-$ or US-20110060998-$ or US-20090265617-$ or US-20090063983-$).did. or (US-7555715-$).did.	US-PGPUB; USPAT	OR	OFF	2012/10/03 11:01
L115	392	optimiz$4 media and (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L116	111	optimiz$4 media device and (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L117	40	optimiz$4 media device SAMe (bitrate or bit adj rate) SAME compress$5	US-PGPUB; USPAT	WITH	ON	2012/10/03 11:01
L118	13	US-2238842-$.DID. OR US-6311185-$.DID. OR US-6463445-$.DID. OR US-7313361-$.DID. OR US-7406434-$.DID. OR US-20030225568-$.DID. OR US-20040025176-$.DID. OR US-20050255852-$.DID. OR US-20070061198-$.DID. OR US-20080195938-$.DID. OR US-20080207182-$.DID. OR US-20090013347-$.DID. OR US-20090240569-$.DID.	US-PGPUB; USPAT	OR	ON	2012/10/03 11:01
L134	4	(US-20090089422-$ or US-20120011014-$ or US-20120210377-$).did. or (US-7155676-$).did.	US-PGPUB; USPAT	OR	OFF	2012/10/03 11:05
L135	4	(US-20090089422-$ or US-20120011014-$ or US-20120210377-$).did. or (US-7155676-$).did.	US-PGPUB; USPAT	OR	OFF	2012/10/03 11:05

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L119	8406	715/201,224,238,249,273,704,733,736,738,748.ccls.	US-PGPUB; USPAT; UPAD	OR	ON	2012/10/03 11:01
L120	28712	709/219,224,236.ccls.	US-PGPUB; USPAT; UPAD	OR	ON	2012/10/03 11:01

L121	36539	L119 L120	US-PGPUB; USPAT; UPAD	OR	ON	2012/10/03 11:01
L122	624	L121 and stream$4 device play$5	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L123	295	L121 and media stream$4 device play$5	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L124	10	L121 and media device play$5 quality level	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L125	2	L121 and media play$5 quality level bit rate	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L126	6	L121 and media play$5 quality level SAME bit rate	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L127	0	L121 and media play$5 quality level SAME compression same resolution	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L128	1	L121 and transformation media quality level	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L129	34	L121 and transformation process$4 resource	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L130	1	L121 and transformation process$4 resource independent	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L131	0	L121 and monitor$4 bandwidth media transformation	US-PGPUB; USPAT; UPAD	WITH	ON	2012/10/03 11:01
L132	1	L121 and monitor$4 bandwidth media transformation	US-PGPUB; USPAT; UPAD	SAME	ON	2012/10/03 11:01
L133	3	L121 and second quality level transformation	US-PGPUB; USPAT; UPAD	SAME	ON	2012/10/03 11:01

10/3/2012 11:07:38 AM
C:\Users\dfaber\Documents\EAST\Workspaces\12238842.wsp

Issue Classification

Application/Control No.	Applicant(s)/Patent Under Reexamination
12238842	BARGER ET AL.
Examiner	**Art Unit**
DAVID FABER	2177

ORIGINAL

CLASS	SUBCLASS
715	736

INTERNATIONAL CLASSIFICATION

CLAIMED					NON-CLAIMED
G	0	6	F	17 / 00 (2006.0)	

CROSS REFERENCE(S)

CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
715	201	224	238	249	273
715	704	733	736	738	748

☐ **Claims renumbered in the same order as presented by applicant** ☐ **CPA** ☐ **T.D.** ☐ **R.1.47**

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	10	17												
	2		18												
2	3		19												
	4		20												
3	5		21												
	6														
4	7														
	8														
	9														
5	10														
	11														
	12														
6	13														
7	14														
8	15														
9	16														

/DAVID FABER/ Examiner.Art Unit 2177 (Assistant Examiner)	10/3/12 (Date)	**Total Claims Allowed:** 10	
/CESAR PAULA/ Supervisory Patent Examiner.Art Unit 2177 (Primary Examiner)	10/05/2012 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 1

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

22862 7590 10/11/2012

GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

Della Revecho	(Depositor's name)
[signature]	(Signature)
January 11, 2013	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/238,842	09/26/2008	Sean Barger	EQUI0016D	2317

TITLE OF INVENTION: AUTOMATED MEDIA DELIVERY SYSTEM

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$885	$300	$0	$1185	01/11/2013

EXAMINER	ART UNIT	CLASS-SUBCLASS
FABER, DAVID	2177	715-200000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
   - ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
   - ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
   (1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
   (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
   1 Michael A. Glenn
   2 Glenn Patent Group
   3 ______

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: EQUILIBRIUM

(B) RESIDENCE: (CITY and STATE OR COUNTRY): SAUSALITO, CALIFORNIA

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
- ☒ Issue Fee
- ☒ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies ______

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 071445 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature: Deborah L Caswell   Date: January 11, 2013

Typed or printed name: Deborah L. Caswell   Registration No. 61766

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor	:	Sean Barger
Serial No.	:	12/238,842
Filed	:	September 26, 2008
Art Unit	:	2177
Confirmation Number	:	2317
Examiner	:	David Faber
Title	:	AUTOMATED MEDIA DELIVERY SYSTEM
Attorney Docket No.	:	EQUI0016D

January 11, 2013

Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

COMMENTS ON STATEMENT OF REASONS FOR ALLOWANCE

Dear Sir/Madam:

In the Notice of Allowance, mailed October 11, 2012, the Examiner provided Reasons for Allowance (hereinafter referred to as "Reasons"). Applicant believes the Reasons, to the extent understood, may be misconstrued and, as such, are incomplete. Applicant submits that indeed, the claims of the instant application, individually or in combination with other claims (via dependency), describe the patentable subject matter of Applicant's invention(s). Accordingly, the Reasons in no way bind or affect the interpretation, infringement, validity and/or enforceability of any claims(s) or patent(s) resulting from, or relating to this application.

Applicant does not believe any fees are due with this submission. However, the Commissioner is hereby authorized to charge any fees due or credit any overpayments to the Glenn Patent Group Deposit Account 071445 (Order No. EQUI0016D).

Respectfully Submitted,

Deborah L. Caswell

Deborah L. Caswell
Reg. No. 61,766

Customer No. 22862

Electronic Patent Application Fee Transmittal

Application Number:	12238842
Filing Date:	26-Sep-2008
Title of Invention:	AUTOMATED MEDIA DELIVERY SYSTEM
First Named Inventor/Applicant Name:	Sean Barger
Filer:	Michael Glenn/Della Revecho
Attorney Docket Number:	EQUI0016D

Filed as Small Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	2501	1	885	885
Publ. Fee- early, voluntary, or normal	1504	1	300	300

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**1185**

Electronic Acknowledgement Receipt

EFS ID:	14683860
Application Number:	12238842
International Application Number:	
Confirmation Number:	2317
Title of Invention:	AUTOMATED MEDIA DELIVERY SYSTEM
First Named Inventor/Applicant Name:	Sean Barger
Customer Number:	22862
Filer:	Michael Glenn/Della Revecho
Filer Authorized By:	Michael Glenn
Attorney Docket Number:	EQUI0016D
Receipt Date:	11-JAN-2013
Filing Date:	26-SEP-2008
Time Stamp:	19:48:01
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$1185
RAM confirmation Number	6935
Deposit Account	071445
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)					

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		2013_01_11_IssueFee-Statement_EQUI0016D.pdf	1190243 fb0ff7139cb6b694576319eece6491adb1e29d5f	yes	3
	Multipart Description/PDF files in .zip description				
	Document Description		**Start**	**End**	
	Issue Fee Payment (PTO-85B)		1	1	
	Post Allowance Communication - Incoming		2	3	
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	31806 65e986d9fd82293fe6bfcc8bb2b302e9dfb73255	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	1222049	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No.: EQUI0016D U.S. Serial No.: 12/238,842

Form 1449 (Modified)	Serial No.: 12/238,842
	Atty. Docket No.: EQUI0016D
Information Disclosure	Applicant: Sean Barger
Statement By Applicant	Art Unit: 2178
	Confirmation No.: 2317
(Use Several Sheets if Necessary)	Filing Date: September 26, 2008

U.S. Patent Documents

Examiner Initials	No.	Patent No.	Issue Date	Patentee	Filing Date

Change(s) applied to document /K.D.D/ 11/13/2012

Published U.S, Patent Application

Examiner Initials	No.	Publication No.	Publication Date	Applicant
/C.B./	1	2008/0205389	8/2008	Fang

Foreign Patent or Published Foreign Patent Application

Examiner Initials	No.	Document No.	Publication Date	Applicant

Non-Patent Literature Documents

Examiner Initials	No.	Author, Title, Date, Place (e.g. Journal) of Publication

Examiner's Signature: /Christopher Bryant/ Date: 10/14/2010

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.B./

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
12/238,842	02/19/2013	8381110	EQUI0016D	2317

22862 7590 01/30/2013

GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 534 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Sean Barger, Mill Valley, CA;
Steve Johnson, Mill Valley, CA;
Matt Butler, Beaverton, OR;
Jerry Destremps, Sausalito, CA;
David Pochron, Cambridge, WI;
Trent Brown, San Anselmo, CA;

IR103 (Rev. 10/09)