IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor:	Isaac LEVANON *et al.*
Asignee:	3VDU, Inc.
Serial No.:	10/035,981
Filed:	December 24, 2001
Title:	System And Methods of Network Image Delivery…
Examiner:	Philip B. Tran
Group:	2155
Confirmation No.:	3619
Attorney Docket:	927/2

This is submitted in response to the Office Action of March 17 2006.

Declaration of Inventors

Applicants Isaac Levanon and Yoni Lavi being duly sworn, depose and affirm that:

1. We hereby declare that we believe we are the original, first and co-inventors of the subject matter which is claimed herein for which a utility patent is sought on the invention described and claimed in the above-identified application; that we have reviewed and understand the contents of the application, including the claims; and, that we acknowledge our duty to disclose to the PTO information of which we are aware which is material to patentability of this invention as defined in 37 C.F.R. 1.56.

2. The invention as claimed in independent claims 1, 7, 12 and 16 was first defined on paper in October 1999 and reduced to practice by a working computer program in December 1999.

It is established that a working product was in applicant's position by 24 January 2000 in view of the Website presentation evidenced by the screens shots of Exhibit A as filed in Affidavit of 10 January 2006.

Independent claims 1, 7, 12 and 16 are reproduced hereinbelow with part annotations a, b, c, d, e…added, for ease of referencing to the various components:

Claim 1 relates to a client system for dynamic visualization of image data provided through a network communications channel, said client system comprising:

(a) a parcel request subsystem, including a parcel request queue, operative to request discrete image data parcels in a priority order and to store received image data parcels in a parcel data store, said parcel request subsystem being responsive to an image parcel request of assigned priority to place said image parcel request in said parcel request queue ordered in correspondence with said assigned priority; and

(b) an parcel rendering subsystem coupled to said parcel data store to selectively retrieve and render received image data parcels to a display memory, said parcel rendering system providing said parcel request subsystem with said image parcel request of said assigned priority.

Applicant affirms that the parcel request subsystem and the parcel rendering subsystem were reduced to practice by 24 January 2000.

A copy of archived source code, substantially identical to that of the version of January 2000 is appended herewith. This source code shows how the invention was reduced to practice by being programmed into a working system implementation and is proof of Reduction To Practice in that it demonstrates that the claimed invention was programmed and worked. Applicants affirm that the code is substantially that of the version of December 1999.

Specifically, the "Parcel Request Subsystem" (a) consists of multiple consumers, and a single producer. The function "DLDownloaderThread" is executed from four threads running in parallel to the main program (See "DLStartDownloaderThreads" and "DLStopDownloaderThreads"). Each of those threads acts as a consumer of the "Parcel Request Queue". The priority queue is implemented over a stack of sorted items named "DTL". The consumers extract items from the top of the stack. The main program acts as

a producer of parcel requests. The function to consume requests from the queue is named DLGetNext.

The function "RenderQuads", in addition to rendering the visible images parcels on the display, also creates requests for parcels that are needed, and evaluates the priority of each request as described in the patent, using function DLAdd. The function DLManager places these requests inside an array, sorts it by priority and replaces the previous stack (Parcel request queue) with the contents of the sorted array.

The priority computation in function DLAdd is performed according to the priority evaluation method explained in the patent application, see paragraphs 59-69 thereof.

The function GenerateTileOffset computes the offset of each requested parcel inside the "Parcel data store" described by the patent. There is an explicit assumption of 2KB per compressed parcel in the data store, possible through use of a fixed-rate compression, which makes the implementation very simple.

The producer side of the Parcel Request Subsystem (explained earlier) in effect shows implementation of *parcel rendering subsystem* (b).

This is also supported by the following routines and functions: DLAdd, DLReset, GenerateOffsetTable, GenerateTileOffset
, PolygonSurface, Distort, Http11Request ,TileManagerRender RecalcTilePlane DrawTileMap, TexelDensity, TileGetDrawlevel, Fast_WithinFrustrum, QuadMake, RenderQuads, FrustrumClipper,

All functions in modules Raster.cppand Tex.cpp

Claim 7 relates to a portable display client system supporting dynamic visualization of image data provided through a wireless network communications channel, said client system comprising:

(c) a display of defined resolution suitable for visual presentation of a graphical image, said display including video memory for storing image data representative of said graphical image;

(d) a network interface coupleable to a wireless network through which

to request and receive image data parcels;

(e) an image parcel data store providing for the storage of image data parcels;

(f) navigation controls providing input information defining a point of view location relative to said graphical image; and

(g) a processor coupled to said video memory, network interface, image parcel data store, and navigational controls, said processor operative to selectively request image data parcels of determined resolution through said network interface in a priority order computed relative to the defined resolution of said display.

Applicant asserts that the invention of claim 7 was reduced to practice in December 1999. All components related to in claim 7 were implemented in the software prior to January 2000. Fragments of source code given herewithin show this implementation.

The display of defined resolution suitable for visual presentation of a graphical image (c) is supported by the FlipWindow function.

The Network interface (d) is utilized by the function Http11Request, which is used to retrieve image parcels through the HTTP/1.1 protocol.

The function "Dynamic_Camera" implements various navigation commands which respond to various keystrokes. This supports (f) *the navigation controls providing input information defining a point of view location relative to said graphical image.*

The hardware specified in (c), (e), (g) by virtue of being hardware, cannot be directly shown in any source code of course. Nevertheless, evidence that these components were available in January 2000 is given through inference and via the OS API and runtime library calls. As evidenced by calls to functions such as GetAsyncKeyState, BitBlt and socket, the program interacted with these hardware elements, available in January 2000 and described in the provisional patent specification filed in December of that year.

Claim 12 relates to a client system providing for the three-dimensional viewing of an image relative to a dynamic viewpoint, said client system comprising:

(h) display memory corresponding to a two-dimensional display of defined resolution;

(i) a network interface through which image data parcels of respective image parcel resolution can be requested and received;

(j) navigational controls providing for the definition of an image viewpoint; and

(k) a processor operative to progressively render image data parcels into said display memory and to progressively request image data parcels through said network interface in a priority order determined with respect to the defined resolution of said two dimensional display and the location of said image viewpoint.

Applicant asserts that the invention of claim 12 was similarly reduced to practice by December 1999.

(h) is evidenced by InitDevice and FlipWindow

Further evidence that these components were available in January 2000 is given through inference and via the OS API and runtime library calls. As evidenced by calls to functions such as GetAsyncKeyState, BitBlt and socket, the program interacted with these hardware elements, available in January 2000 and described in the provisional patent specification filed in December of that year.

The Network interface (i) is utilized by the function Http11Request, which is used to retrieve image parcels through the HTTP/1.1 protocol.

The *navigation controls providing input information defining a point of view location relative to said graphical image* (j) are supported by the function "Dynamic_Camera" which implements various navigation commands in response to various keystrokes.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.