Comparing Media Codecs for Video Content

Jeremiah Golston (Distinguished Member Technical Staff, Texas Instruments)
j-golston@ti.com
#250 at Embedded Systems Conference – San Francisco 2004

Introduction

Digital video is being adopted in an increasing range of applications including video telephony, security/surveillance, DVD, digital television, Internet video streaming, digital video camcorders, cellular media, and personal video recorders. Video compression is an essential enabler for these applications and an increasing number of video codec (compression/decompression) industry standards and proprietary algorithms are available to make it practical to store and transmit video in digital form. Compression standards are evolving to make use of advances in algorithms and take advantage of continued increases in available processing horsepower in low-cost integrated circuits such as digital media processors. Differences exist in the compression standards and within implementation of standards based on optimizations for the primary requirements of the target application. This paper provides an overview of the different compression standards and highlights where they are best suited. It also provides an overview of compression rules of thumb for different standards and the corresponding performance requirements for real-time implementations.

The Video Compression Challenge

A major challenge for digital video is that raw or uncompressed video requires lots of data to be stored or transmitted. For example, standard definition NTSC video is typically digitized at 720x480 using 4:2:2 YCrCb at 30 frames/second. This requires a data rate of over 165 Mbits/sec. To store one 90-minute video requires over 110 GBytes or approximately 140x the storage capability of a CDROM. Even lower resolution video such as CIF (352x288 4:2:0 at 30 frames/second) which is often used in video streaming applications requires over 36.5 Mbits/s - much more than can be sustained on even broadband networks such as ADSL. So, it is clear that compression is needed to store or transmit digital video.

The goal for image and video compression is to represent (or encode) a digital image or sequence of images in the case of video using as few bits as possible while maintaining its visual appearance. The techniques that have emerged are based on mathematical techniques but require making subtle tradeoffs that approach being an art from.

Compression Tradeoffs

There are many factors to consider in selecting the compression engine to use in a digital video system. The first thing to consider is the image quality requirements for the application and the format of both the source content and target display. Parameters include the desired resolution, color depth, the number of frames per second, and whether the content and/or display are progressive or interlaced.

Compression often involves tradeoffs between the image quality requirements and other needs of the application. For example, what is the maximum bit rate in terms of bits per second? How much storage capacity is available and what is the recording duration? For two-way video communication, what is the latency tolerance or allowable end-to-end system delay? The various compression standards handle these tradeoffs including the image resolution and target bit rate differently depending on the primary target application.

Another tradeoff is the cost of real-time implementation of the encoding and decoding. Typically newer algorithms achieving higher compression require increased processing which can impact the cost for encoding and decoding devices, system power dissipation, and total memory in the system.

Standards Bodies

There have been two primary standards organizations driving the definition of image and video compression standards. The International Telecommunications Union (ITU) is focused on telecommunication applications and has created the H.26x standards for video telephony. The Internal Standards Organization (ISO) is more focused on consumer applications and has defined the JPEG standards for still image compression and MPEG standards for compressing moving pictures.

The two groups often make slightly different tradeoffs based on their primary target applications. On occasions the two groups have worked together such as recent work by the JVT (or Joint Video Team) on a common standard referred to as both H.264 and MPEG-4 AVC. While almost all video standards were targeted for a few specific applications, they are often used to advantage in other kinds of applications when they are well suited

Standards have been critical for the widespread adoption of compression technology. The ITU and ISO have been instrumental in creating compression standards the marketplace can use to achieve interoperability. These groups also continue to evolve compression techniques and define new standards that deliver higher compression and enable new market opportunities.

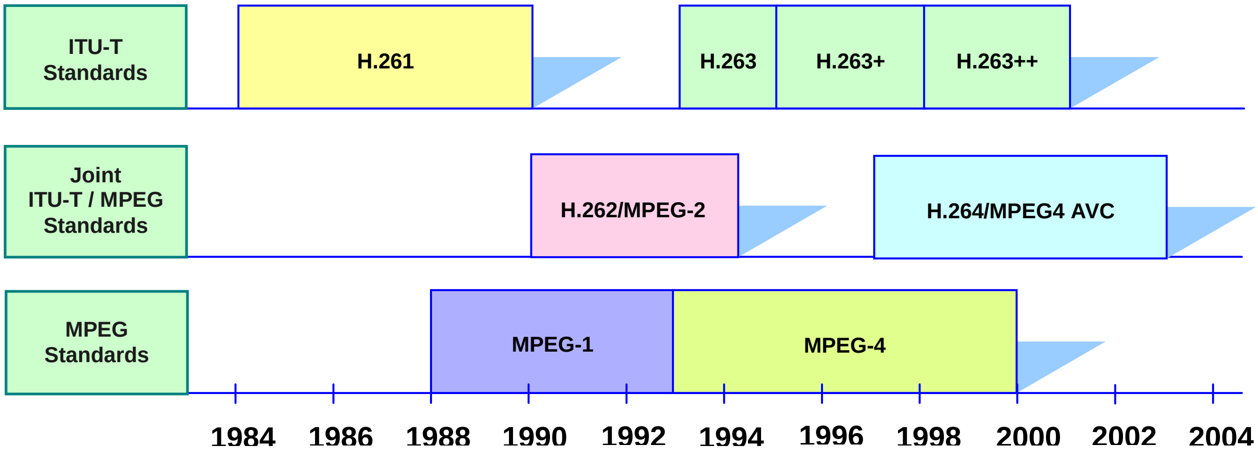


Figure 1. Progression of the ITU-T Recommendations and MPEG standards.

In addition to industry standards from the ITU and ISO, several popular proprietary solutions have emerged particularly for Internet streaming media applications. These include Real Networks Real Video (RV10)

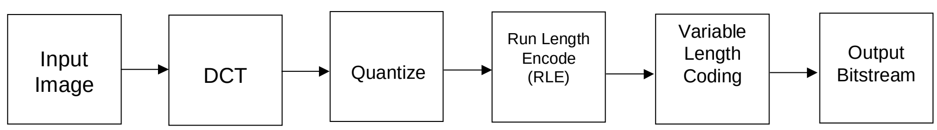
Microsoft Windows Media Video 9 Series, ON2 VP6, and Nancy among others. Because of the installed base of content in these formats, they can become de facto standards.

The number of standards and de facto standards is rapidly increasing creating an increasing need for flexible solutions for encoding and decoding. We'll step through some of the industry standard formats in a little more detail in the next few sections focusing on key features and target applications.

JPEG

JPEG developed by the ISO was the first widespread image compression standard [1]. It was designed to allow compression of digital images and is now widely used for Internet web pages and digital still cameras. The compression quality at a given rate is somewhat dependent on the image content such as the amount of detail or high frequency content in the image. However, using JPEG compression factors of 10:1 can typically be achieved without introducing serious effects from the compression. As you go above this 10 to 1 ratio, you can start to easily notice compression artifacts such as blockiness, contouring, and blurring of the image.

Although JPEG is not optimized for video, it has often been used for coding video with what is sometimes referred to as "Motion JPEG". Motion JPEG is not defined in the standard but typically consists of independently coding individual frames in a digital video sequence using JPEG. A common application for Motion JPEG is network video surveillance. Since each image is coded independently, it is easy to search through the content and also has benefits for interoperability with PC browsers.



Key Components

– DCT (translate spatial data to frequency domain)

– Quantization (scale the bit allocation for different frequencies generating many zero valued coefficients)

– Run-length code (RLE) non-zero coefficients

– Variable Length Coding (VLC)

–Entropy code (e.g., Huffman) run-length codes

Figure 2: JPEG (Intra-frame) Compression Block Diagram

The main functions in the JPEG standard shown in Figure 2 formed the core for all of the major compression algorithms that followed. Key functions include the following:

Block-based Processing: Dividing each frame into blocks of pixels so that processing of the image or video frame can be conducted at the block level.
Intra-frame Coding: Exploiting the spatial redundancies that exist within the image or video frame by coding the original blocks through transform, quantization, and entropy coding. The frame is coded based on spatial redundancy only. There is no dependence on surrounding frames.
8x8 DCT: Each 8x8 block of pixel values is mapped to the frequency domain producing 64 frequency components
Perceptual Quantization: Scale the bit allocation for different frequencies typically generating many zero valued coefficients.
Run-length Coding: Represent the quantized frequency coefficients as a non-zero coefficient level followed by runs of zero coefficients and a final end of block code after the last non-zero value.
Variable Length (Huffman) Coding: Huffman coding converts the run-level pairs into variable length codes (VLCs) with the bit-length optimized for the typical probability distribution.

JPEG has extensions for lossless and progressive coding. Unlike most of the video compression standards, JPEG supports a variety of color spaces including RGB and YCrCb.

JPEG2000

JPEG2000 is a new still image coding standard from the ISO that was adopted in December 2000 [2]. It was targeted at many of the same applications as JPEG including high-quality digital still cameras, hard copy devices and Internet picture applications. The primary goals were to provide improved compression along with more seamless quality and resolution scalability.

JPEG2000 achieves key improvements in scalability of resolution and bitrate through use of several key functions that are not used by the JPEG, MPEG, and H.26x standards.

Discrete Wavelet Transform: The wavelet transform is used replacing the DCT to achieve higher compression and improve support for scalable transmission. Wavelets are new basis functions, unlike the usual cosines (DCT) and sines (FFT). They are called wavelets because they look like small waves. They have an excellent ability to represent both stationary as well as transient phenomena with few coefficients. Wavelets represent signals as a linear summation of shifted and translated versions of a basic wave. JPEG2000 is coded in frequency sub-bands using the wavelet transform to allow resolution scalability. The same bitstream can be decoded at different resolutions. Also, a thumbnail can be sent providing excellent quality at lower resolution and the resolution can be gradually increased as more sub-bands are received. This structure also helps improve error resilience for wireless and Internet applications.
Bit Plane Coding: The quantized sub-bands from the wavelet transform are divided into code blocks. Code blocks are entropy coded along bit planes using a combination of a bit plane coder and binary arithmetic coding. In JPEG2000, embedded block coding with optimized truncation (EBCOT) is used to implement bit plane coding. The algorithm uses symmetries and redundancies within and across the bit planes. The bit plane coding structure can be used to offer bitrate scalability since increasing detail can be added as more bit planes are decoded. Also, different quality bitstreams can be decoded at the same resolution, depending on the client's bandwidth without having to re-encode separately for each client.

Binary Arithmetic Coding: The bit plane coding outputs are entropy coded using binary arithmetic coding to generate the bitstream. Binary coding allows more flexibility than Huffman coding because symbols don't have to be represented by an integer number of bits. The JPEG2000 arithmetic coder uses predetermined probability values and the adaptation state machine is also supplied by the standard.

JPEG2000 can sustain much better quality than JPEG at high compression ratios because the wavelet transform degrades more gracefully. As the compression rate decreases, the gap narrows between JPEG and JPEG2000. For excellent quality, the wavelet transform yields about ~30% more compression (e.g., 13:1) than JPEG (10:1). The wavelet transform is more computationally intensive than the DCT but it is the really the bit-plane coding and binary arithmetic encoding functions that add most of the complexity to JPEG2000.

JPEG2000 includes both lossy and lossless compression modes. Motion JPEG2000 support is also being defined in the standard. One of the potential uses of motion JPEG2000 is for applications such as digital cinema requiring some compression but with the primary focus on highest video quality. JPEG2000 can support pixel depths greater than 8-bit such as 10 or 12-bits/pixel and is flexible in terms of the color space.

The widespread benefit of wavelets for low-bit rate video compression is still not clear. Motion estimation typically works best on small blocks. However, dividing images into small blocks degrades the wavelet performance. This makes it difficult to apply motion compensation using wavelets in the spatial domain. Meanwhile, wavelets are not shift invariant so shifted versions of the image result in a totally new representation. This creates problems for recognizing shifted versions of objects in images in transform domain. Since the transform is no longer block based, research is ongoing to find efficient ways to exploit motion in the hierarchical or sub-band domain. There have been some proprietary video codecs that use wavelets for I pictures, and DCT for P and B pictures. However resulting bit-rates are not comparable with today's advanced media codecs such as H.264.

H.261

H.261 defined by the ITU was the first major video compression standard [3]. It was targeted for video conferencing applications and was originally referred to as Px64 since it was designed for use with ISDN networks that supported multiples of 64 kbps. As seen in Figure 3, H.261 and the video compression standards that followed use similar core functions as JPEG such as block-based DCT but add features to explore the temporal redundancy or commonality from one image to the next in typical video content. For example, background areas often stay the same from one frame to the next and do not need to be retransmitted each frame. Video compression algorithms typically encode the differences between neighboring frames instead actual pixel values. Key features added in H.261 over JPEG include:

Inter-frame Coding: Coding uses both spatial redundancy and temporal redundancy to achieve higher compression.
P Frames: Frames are coded using data the previous decoded frame to predict the content in the new frame and exploiting any remaining spatial redundancies within the video frame by coding the residual blocks, i.e., the difference between the original blocks and the corresponding predicted blocks, using DCT, quantization, and entropy coding.
Motion Estimation: Used in the encoder to account for motion between the reference frame and the frame being coded to allow best possible prediction. This is usually the most performance intensive function in

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.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.