WHITE PAPER

The right Video Compression for Digital CCTV

Author:

Yves Joskin
Chief Architect & Chief Technology Officer

December 2008

EURESYS™
Excellence in vision

If you are responsible for designing a new video surveillance system, you have to make a technology choice regarding image storage and transmission. For sure, it will be digital. But which video compression scheme is the most suitable for your application? It is so easy to get puzzled by the diversity of options. As a system engineer, you will find pragmatic technical answers in this paper to support your decision.

Why digital?

Video sequences are stored and networked as digital bitstreams.

In the old days, the pictures of the monitored scenes were delivered by video surveillance cameras as analog signals through coaxial cables. The video signals were recorded on magnetic tapes, in an analog way, too. Nowadays, pictures are digitized. They are made of an array of pixels, and each pixel has a color represented by a group of bits. In our digital world, a video sequence has become a *bitstream*, a succession of zero-or-one elements encoding the video sequence. The nice thing with bitstreams is that they naturally fit the way computers are working. It strictly makes no difference for a computer to store or transmit a bitstream representing a video sequence or the PDF file representing this white paper. They both are simply *bits*…

This is probably the major advantage of getting digital. We can send images to any recipient, close to here or far away, through a local network or the Internet. And we can temporarily store images, for a short or long period, on a hard disk or a DVD. General-purpose computers and networks are excellent at doing that, and we will never again need those CCTV-dedicated devices such as VCR, switches, amplifiers, transmitters… Even video monitors are replaced by computer screens, including those of PDAs or cell phones.

What is a compression standard?

A compression standard ensures the interoperability of encoders and decoders.

We understand now that all video surveillance images of the 21st century are and will be transportable and storable bitstreams. A bit of curiosity make us wonder how a bitstream is representing a sequence of images. Are there several kinds of bitstreams? What are the technical rules governing the creation of bitstreams? Well, the answer is simple: those rules are collected in a *compression standard*.

Transforming a video sequence into a bitstream is called coding or encoding. This is usually performed by an encoder inside or close to the camera. Transforming a bitstream back to an image is called decoding. A decoder is needed each time the bitstream has to be displayed. A device able to perform both functions is called a codec (coder/decoder)

A compression standard is above all an official set of rules that allows encoders and decoders to work together correctly. Any decoder obeying the rules of a given standard is able to recover the video sequence created by an encoder complying with the same standard. Encoders and decoders may be separated from each other in time or in space, the standard guarantees the *interoperability* of bit streams transported from the former to the latter.

This represents an essential requirement for compression in an open digital video surveillance system. It should be governed by a popular and reliable standard, in order to allow various systems to use your images, and/or various cameras to feed your system.

Why compression?

Compressing images is not a recent idea. Excerpting a 1929 patent[1]: "*It has been customary in the past to transmit successive complete images of the transmitted picture. In accordance with this invention, this difficulty is avoided by transmitting only the difference between successive images of the object*".

Video compression eliminates space and time redundancies.

This sets the tone of video compression. In the images of a video sequence, there is so much repeated information that it makes absolutely no sense to transmit everything. A lot of mathematical and computational techniques have been developed to detect every occurrence of redundancy, both in the space or time domain. For instance, where several neighboring pixels in an image share a common color, it is not needed to encode every pixel individually in the bitstream. A kind of signature that characterizes the extent of the region and its contents is cheaper in term of bit count. Similarly, corresponding regions of successive images are usually very similar, especially when there are few movements. In that case, very efficient techniques avoid encoding several times the same content.

It is not a goal of this paper to give details of modern compression schemes. We just want to stress the point that the redundant nature of video information, and therefore its ability to be compressed, is an incredibly precious property. We need compression to transport bitstreams over existing networks, and

[1] US patent 341,811, April 25, 1929

we need compression to store bitstreams on existing hard disks. Uncompressed video would immediately saturate common public networks and overcrowd low-cost storage media.

This leads us to the second essential requirement in relation to your technology choice. It should be efficient in term of compression. Higher compression ratios means more stored images per Gigabyte of hard disk capacity, and/or more networked images per Megabit/s of transmission bandwidth.

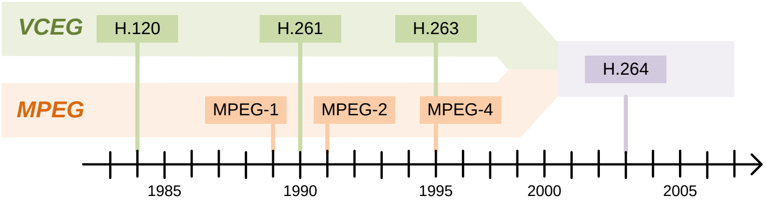
Which standard?

Today, H.264 is the highest performance video compression standard.

As you can guess, successive editions of compression standards are increasingly more powerful. So the following question arises: will the latest top-notch compression scheme comply with the essential requirement of prevalence and stability? Or will it be superseded in six months by a better one, invalidating the technology decision I make today?

Fortunately, the creation of a robust international standard is not that fast. We are now five years behind the introduction of a compression standard known as *H.264*[2]. The growing acceptance of this standard, in conjunction with its remarkable capabilities, allows to state that H.264 *is* the right video compression for CCTV. Arguments to support this assertion follow.

Genesis of H.264



H.264 is the result of joint efforts of two major standardization organizations, the ITU-T[3] Video Coding Expert Group (VCEG) and the ISO/IEC[4] Moving Picture Experts Group (MPEG). On the one side, VCEG issued a generation of compression standards mainly targeting video conferencing applications, the "H-series*".* On the other side, MPEG originated the famous "MPEG-series". MPEG-1 supported the first popular digital videos on CD. MPEG-2 became the mainstream compression technology underlying broadcast television. MPEG-4 is a huge multipart construction (24 parts as of 2008) focusing on a consistent representation of multimedia objects (audio and video).

H.264 is the result of joint efforts of major standardization organizations.

In this world, it seems that, after a period of existence and success, a standard originated by one organization is adopted by the other one. For instance, the very successful MPEG-2 is also standardized by ITU-T as H.262. Similarly, the part 2 of MPEG-4, called "Visual", defines a compression scheme close to H.263.

The point here is that those two powerful organizations partnered for the first time under the JVT (Joint Video Team) *before* developing a new standard. In 2003, JVT introduced the first version of what is simultaneously referred to as H.264 (VCEG parlance) or MPEG-4 Part 10 (MPEG parlance). In the beginning, the standard was often called AVC (Advanced Video Coding). A correct denomination would be "H.264/MPEG-4 AVC", but usage seems to favor the simpler "H.264".

The goal of H.264 was to design a culminating version of video compression gathering the latest algorithmic refinements, and, above all, serving *all* applications. H.264 is incredibly flexible!

The next paragraph shows that JVT has reached its goal: H.264 is already everywhere…

Widespread adoption of H.264

Quoting Wikipedia[5]: "*H.264/AVC experienced widespread adoption within a few years of the completion of the standard. It is employed widely in applications ranging from television broadcast to video for mobile devices. In order to ensure compatibility and problem-free adoption of H.264/AVC, many standards bodies have amended or added to their video-related standards so that users of these standards can employ H.264/AVC.*"

[2] *A concise technical description of H.264 is available in www.vcodex.com/files/H.264_overview.pdf*

[3] *International Telecommunication Union www.itu.int*

[4] *International Organization for Standardization www.iso.org*

[5] *en.wikipedia.org/wiki/H.264*

You will find in the Wikipedia article an impressive list of applications in which H.264 is currently setting the rules. Just to name a few:

- Format of the Blu-ray Disc, and its predecessor the HD DVD.
- Terrestrial television broadcast systems in many countries: Brazil, China, Estonia, France, Japan, Korea, Lithuania, New Zealand, Norway, Poland, Slovenia, Spain, United Kingdom, United States…
- Direct broadcast satellite services in the following countries or regions: Europe, Germany, India, Ireland, Italy, Singapore, Sweden, United Kingdom, United States…
- Television over the Internet (IPTV), in particular in Australia and Canada.
- Multimedia services in mobile telephony.
- Military application (NATO, DoD).
- Enrichment of network protocols (IETF, ISMA).
- Video conferencing.
- Electronic commerce of films and TV series (Apple).
- Consumer video recording (Sony, Panasonic).

It is true that video surveillance applications are not often reported as strongly concerned by H.264. But there are sure signs that this is happening: in May 2008 a consortium grouping Axis, Bosch and Sony announced ONVIF[6], Open Network Video Interface Forum, for the creation of a global standard for the interface of network video products. ONVIF includes H.264 as an underlying layer for video streaming.

What makes a good video compression?

It is time now to get an understanding of how much better H.264 is compared to the competing compression schemes. Following Ian E.G. Richardson[7], there are three axes along which you can rate a compression scheme:

- Quality,
- Bit rate,
- Computational cost.

The performance of a video codec is always a trade-off among the three variables. Let us learn a bit more about the three axes.

Quality

This notion is inseparable from the fact that all video compression schemes are *lossy*. This means that the cycle consisting in coding a video sequence into a compressed bitstream, then decoding the bitstream, yields a recovered uncompressed video sequence which is not strictly the same as the original one. There are information losses. Incidentally, there are lossless compression schemes, but their performance along the bit rate axis is poor, and they cannot be used for the vast majority of applications. An exception might be medical imaging for diagnosis purposes.

Compression quality is measured with PSNR, but this measurement alone is insufficient

So each image of the recovered sequence is different from the corresponding image in the original sequence, leading to image degradation. How annoying is that degradation? This is what the image quality axis attempts to quantify. The most common way to assess the image quality is the PSNR (Peak-to-peak Signal to Noise Ratio). This is a well defined formula that you apply to two corresponding images before and after the codec cycle. The color is usually neglected in this assessment. Any luminance difference between two pixels is considered as unwanted noise, and a quadratic mean of that noise for all pixels is established by the computation. The PSNR is expressed in decibels on a logarithmic scale. The higher the PSNR, the better the image quality. To fix ideas, 45 dB denotes an excellent image, 35 dB is medium, 25 dB is poor.

The PSNR is a simple and objective quality assessment method, and as such is largely used. However, it does not exactly report the subjectively perceived quality. Other figures have been introduced to reflect more correctly the subjective notion of quality. Some of them address visually annoying artifacts such as *blurring* (losses in details) and *blocking* (appearance of artificial rectangular structures in the reconstructed image). Today, unfortunately, there is no universally accepted assessment method for the subjective quality of video codecs.

[6] www.onvif.org
[7] *H.264 and MPEG-4 Video Compression, Wiley 2003*

Bit rate

Subjecting a video sequence to a particular encoder yields a compressed bitstream which is progressively generated. The average production rate of bits is the bit rate. Bit rates for compressed video may range somewhere between 50 kbits/s and 250 Mbits/s. The bit rate is strongly related to the compression performance of the coder, but it also depends on the size of the original images (SDTV, HDTV) and their frame rate. You can use it to compare encoders compressing images of identical size and rate.

Knowing the bit rate is nice to match the compression with a network channel planned for transmitting bitstreams. The bit rate and the channel bandwidth are expressed in the same unit.

For instance, if you can count on a network bandwidth of 40 Mbits/s, you know that you can safely consider the simultaneous transmission of 16 cameras compressed at a controlled bit rate of 2 Mbits/s each.

Bit rate is related to compression performance, but this characteristic alone is insufficient

If you are concerned with storage, the bit rate will help you as well. A quick computation tells you that your 1 TB disk (8 Tbits) is able to continuously and simultaneously record the bitstreams of the 16 cameras compressed at 2 Mbits/s for 250,000 seconds, i.e. approximately 70 hours.

You can also have an idea of the mean size of an individual encoded image. Suffice it to divide the bit rate by the frame rate of the camera. Assume a PAL-like camera working at 25 fps (frames per second), and originating a 2 Mbits/s bitstream. The average size of the compressed frame is 80,000 bits, or 10 kBytes.

This leads to the point where the bit rate relates to the compression performance. Remaining focused on our example, assume that the size of the PAL-like image is 720H x 576V (an SDTV resolution). This corresponds to 414,720 pixels. We assume further an original representation in a YUV format at 16 bits per pixel. Thus the uncompressed image weights 829,440 bytes. Being encoded as 10 kBytes, the compression ratio is 83:1.

By the way, there is another way to quantify the compression performance: it is the number of *bits per pixel* (bpp). In our example, the 414,720 pixels in the image are encoded as 80,000 bits, providing 0.19 bpp. Each pixel, originally represented by 16 bits, is effectively represented by 0.19 bit in the bit stream. The ratio is consistently 83:1. This method has the advantage of qualifying the compression performance independently of the size and the rate of the mages. It is often used for still images.

Computational cost

Encoding and decoding images involve a large amount of real-time computation. Generally speaking, increased compression performance involves higher computational cost. What kind of computing devices can we use to perform the computation? In the early days, general-purpose computers were inadequate, even for the relatively simple compression standards available then. With the ever increasing processing power of computing hardware (the famous Moore's law), it became possible to run sophisticated compression or decompression algorithms on low-cost PCs. However, it does not come as a free lunch. If you entrust your computer with such a high-priority task as real-time video coding or decoding, you will experience a performance reduction for all other software applications.

Let us be a bit more explicit in the computational requirements of H.264. The requirement is *asymmetric*. There is a need for more processing power to encode the video sequence than to decode the bitstream. This is a rather general characteristic of compression algorithms: compressing is more complex and more expensive than decompressing. Roughly speaking, the main reason for this is that compressing is a two-step process: (1) detecting and locating occurrences of redundancy (in space and in time), (2) running an algorithm to condense each occurrence into a special bit pattern. The decoder has the simpler task of turning back the bit patterns into image elements, without having to search where or when they occur.

In many cases, the computational asymmetry is affordable because the bitstreams are *broadcasted*. There is a single place where the video sequence is generated (e.g. a video-on-demand server), and many places where it is consumed (e.g. set-top boxes). The former calls for a common encoder, the latter call for many decoders. The high cost of compression is absorbed by the plurality of customers.

Using H.264 in video surveillance imposes dedicated hardware.

Unfortunately, video surveillance applications do not follow the broadcast pattern. There are many video originators (the CCTV cameras) and few consumers (a monitoring room). Cost-wise, the CCTV application is not in the best situation: many high-cost encoders, and few low-cost decoders.

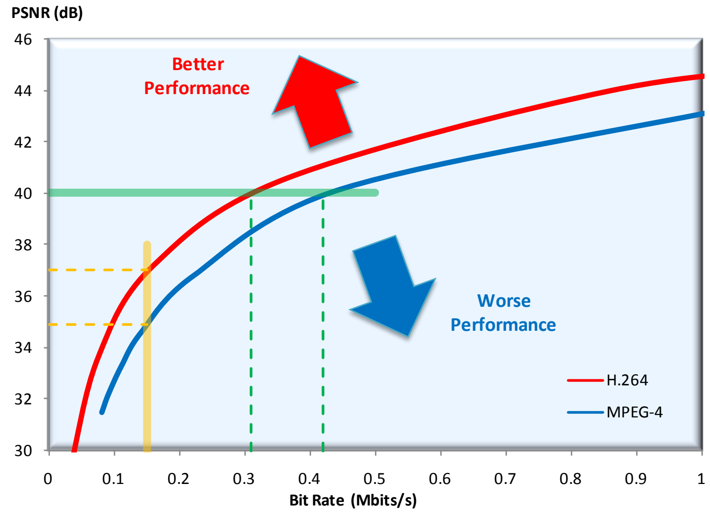
This is probably why the acceptance of H.264 as a standard in video surveillance systems has been delayed. Despite its evident technical advantages, we have had to wait for low-cost hardware implementations of the H.264 compression algorithms to become commercially available.

PC implementations are not conceivable. The H.264 compression achieves its performance with high complexity algorithms, one order of magnitude above the previous standards. Upon the introduction of

H.264 in 2003, no off-the-shelf PC could sustain the processing power requested for decoding low-resolution CIF video. Today, PC decoding is clearly manageable, but encoding remains an issue. If you add up the plurality of cameras in CCTV systems, you reach the conclusion that dedicated low-cost compression hardware is unavoidable to implement H.264 in video surveillance systems. There are now several silicon manufacturers offering ASICs (Application Specific Integrated Circuit) or DSPs (Digital Signal Processor) targeting H.264 encoding[8].

Rate-distortion analysis

The universally admitted method to qualify a video codec is to combine the quality and bit rate axes into a graph representing what is called the *rate-distortion* performance[9]. Here is a rate-distortion analysis comparing two ways of compressing a CIF video sequence at 50 fps. This is extracted from the above-mentioned book of Ian E.G. Richardson.



The red curve shows how the quality (PSNR) is related to the bit rate (Mbits/s) for an H.264 encoder. The blue curve has the same meaning for an MPEG-4 Part 2 encoder, considered as the predecessor of H.264. As you can see, the H.264 curve strictly dominates the MPEG-4 curve, meaning lower bit rates and higher PSNRs for all settings.

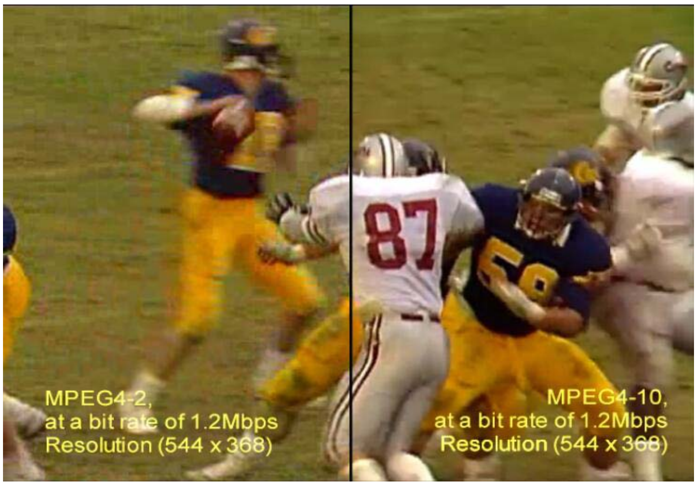
Rate-distortion analysis is the correct way to qualify compression performance.

The rate-distortion graphs will help you to make objective appreciations. For instance, you can compare the two encoders generating high quality images at 40 dB PSNR. The curves show you that the MPEG-4 encoder performs at 420 kbits/s while the H.264 encoder contains the bit stream to 310 kbit/s. This is about 25 % increase in compression ratio.

Conversely, suppose that you can afford 150 kbits/s to store or transmit your video sequence. You see that, in this case, the MPEG-4 encoder achieves a 35 dB PSNR, while the H.264 encoder boosts the quality to 37 dB.

This provides an approach to compare competing codecs and compression technologies. If you could take the time to scan all available encoders, you would discover that H.264 is clearly the winner in rate-distortion efficiency. In practically all cases, it provides a lower bit rate for a given quality, or a better quality for a given bit rate.

As a rule of thumb, it is recognized that, at identical quality, H.264 provides a factor of two in bit rate savings compared to MPEG-2, the prevalent technology for broadcast television. The coding gain of H.264 over H.263 is in the range of 25 % to 50 % depending on the type of applications.



In order to give some practical meaning to the above graphs and numbers, here is a split image from an encoded video sequence[10]. The left side shows the effect of an MPEG-4 Part 2 codec, and the right side the effect of the reference implementation of H.264, at a given bit rate. The improvement is clear enough. Note the blurring and blocking artifacts on the left side.

[8] A good example is the Californian company Stretch Inc (www.stretchinc.com).

[9] Rate-Distortion is not a recent theory. Seminal work was done by T. Berger (Rate-Distortion Theory, Prentice-Hall, 1971), on the footsteps of Shannon.

[10] Published at www.balooga.com/mpeg4.php3.

Network friendliness

Contrary to earlier compression standards, H.264 has been designed from the ground up with the concern of networking. Practically, the coding scheme is explicitly divided in a Video Coding Layer (VCL) and a Network Abstraction Layer (NAL). The VCL supports the coding efficiency, and the NAL supports video specific transport features for a variety of networks.

More precisely, an H.264 bitstream is transported over a network as a succession of independent *NAL units*. These units make sense both from the coding perspective and the network perspective.

So a NAL unit is at the same time a bitstream section that encodes an individual frame, and a regular payload for the standardized RTP protocol[11]. This principle constitutes a significant advantage of H.264 that facilitates the integration of the technology into many network-oriented areas.

Stream-based versus frame-based compression

All compression standards we have considered so far are *stream-based*. They eliminate space redundancy (within a frame) and time redundancy (between frames). We pay here some attention to the compression of still images, or *frame-based* compression. There are two renowned compression standards issued by the Joint Photographic Expert Group committee, namely JPEG and JPEG 2000. Since 1991, JPEG is the reference method to compress individual images with space redundancy elimination. In assembling the compressed successive images of a video sequence, you get a storable and networkable bitstream that has been informally named M-JPEG (Motion JPEG).

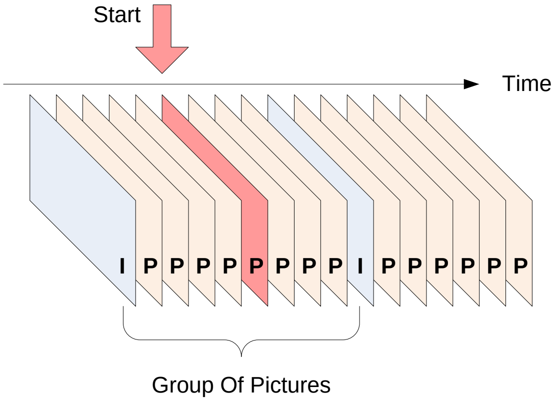
M-JPEG has been extensively used in video surveillance applications. A reason for this is the maturity of the technology. With the evolution of hardware performance, implementing JPEG codecs is easily achieved in regular PCs and low-cost DSPs. The drawback is the poor compression performance. Lacking the ability to take advantage of similarities between successive images in a sequence (time redundancy), frame-based compression is considerably less efficient than stream-based compression. You can easily count on a compression depreciation of one order of magnitude when switching from stream- to frame-based compression. This is especially true in video surveillance applications where the scene may remain motionless for hours!

This is largely enough to justify stream-based compression in digital CCTV applications, but there is a consideration which is often claimed to favor frame-based compression. This consideration finds its roots in the fact that stream-based compression uses *groups of pictures* (often called GOPs). Refer to the following figure.

Top-notch frame-based compression cannot beat stream-based compression.

The GOP is headed by a so-called I-frame (intra), and contains several P-frames (predictive). The I-frame is coded as an isolated image, eliminating space redundancy. The P-frames of the GOP encode differences between the heading I-frame and each actual image. This is how time redundancy is eliminated.

What is the problem? Suppose you want to play a recorded bit stream starting at the pink frame. Before being able to build the decoded image, you have to "rewind" the bitstream until you get the heading I-frame. It is even worse if you receive the bitstream in real-time from a network: you cannot accept the bitstream at any frame but an I-frame.

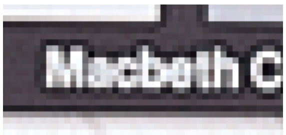


This lack of flexibility of GOP-based bitstreams has encouraged in the past some CCTV system designers to adopt M-JPEG compression. It is clear that randomly accessing the bitstream is facilitated with frame-based compression. Supporters of this idea had a lot of expectations regarding JPEG 2000, the latest advance in frame-based compression, introduced as an international standard (ISO/IEC) in the year 2000. It is true that JPEG 2000 represent a 20 % improvement in compression ratio compared to JPEG at identical quality. It is also true that concatenating JPEG 2000 frames to support video sequences has been introduced as a supplement to the ISO/IEC standard in 2001 (Motion JPEG 2000).

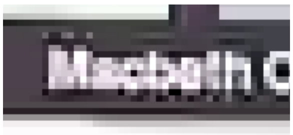
[11] *RTP means Real-time Transfer Protocol. See RFC 3984, RTP payload for H.264 Video, tools.ietf.org/html/rfc3984*

However, there are arguments that impair the acceptance of JPEG 2000 as a dominant standard in video surveillance.

A recent paper[12] shows that using H.264 in intra mode exclusively (no predictive frames) always works better than JPEG 2000. Rate-distortion performance is just higher. A detail of a test image is shown below. Encoding is performed with a bit rate of 0.24 bpp, a compression ratio of 100 for an RGB24 image. H.264 clearly offers better objective and subjective quality.



Uncompressed detail



H.264 PSNR 29.47 dB

JPEG 2000 PSNR 27.14 dB

Scalability

Video compression appears as a one-shot process. You set the encoder for a given quality and a given bit rate, you feed it with a video sequence at a given frame rate with images at a given resolution, and you let the magic begin. What you get is an opaque bitstream that you have to feed to a compatible decoder. You recover a video sequence with the predetermined quality, resolution and frame rate. It is irreversible in the sense that you cannot get back to better quality, higher resolution or faster frame rate.

Latest improvements of H.264 promise a bright future.

However, you may want to use the bitstream for a range of applications. As a simple example, suppose you expect users to watch the bitstream-encoded video on large monitors as well as on small portable devices. You have only one choice at encoding: setting the compression for the most demanding usage, i.e. highest resolution. The high-end display decodes the bitstream and consumes the video as is. The low-end display needs an additional step after decoding consisting in scaling down the video images before consuming them.

Scalability consists in supporting several versions of the video sequence inside the bitstream to serve several applications. The point is that you can get one version without decoding the bit stream in its entirety. The decoder is able to locate in the supplied bitstream only those sections that participate to the reconstruction of the video with the desired parameters.

The JVT group supplemented H.264 with scalability through an amendment to the standard approved in July 2007. This is called SVC (Scalable Video Coding). SVC has provision for scalability in several dimensions:

- Temporal scalability: obtaining sequences at various frame rates.
- Spatial scalability: obtaining images at various resolutions.
- Quality scalability: obtaining images at various PSNR.

It is amazing to note that scalability is achieved in H.264 with only a 10 % bit rate increase[13], compared to a non-scalable bitstream with the same performance level. If you are interested in the mathematical theory underlying this unexpected property, see reference[14]. You will discover how the theory of *superposition coding*, invented in the 70s, opened the way to scalability.

In video surveillance, scalability is potentially very useful. Suppose you are designing a flexible PC-based video server. An SVC bitstream is generated for each video source in the capture card that collects the video signals from a set of CCTV cameras monitoring a particular site. You set the encoding parameters to guarantee top quality in the bitstreams, which are recorded in the local hard-disk at the highest level of frame rate, resolution and quality.

At the time your application plays back the data for local display, you may spare processing power of your PC in decoding at the right resolution, e.g. full screen display or split-screen option. If you have to network the bitstream to a low-end client, you can obtain, at low computational cost, a tailored lower resolution and/or lower frame rate bitstream, sparing network bandwidth. At some time, your storage disk may get full. You can recover space in stripping the bitstreams encoding the oldest sequences and keeping lower quality images, but still usable. The nice thing is that all this is done without decoding and recoding the bitstream, and this is *extremely* important.

SVC is the emerging breakthrough in compression technology. If you want to understand how it works, you will find plenty of references on the web[15]. Basically, SVC relies on a layered structure with a low-scale base layer directly compliant with native H.264, and enhancement layers made available as additional NAL units. In picking up the right combination of enhancement elements in addition to the base

[12] *H.264/AVC intra coding and JPEG 2000 comparison, Camperi & Picco, Politecnico di Torino, April 11, 2008*
[13] *SVC Verification test Report, available at www.chiariglione.org/mpeg/quality_tests.htm*
[14] *Advances in Network Information Theory, P. Gupta, G. Kramer & A. van Wijngaarden, American mathematical Society, 2003*
[15] *For instance: Overview of Scalable Video Coding, www.chiariglione.org/mpeg/technologies/mp04-svc/*

elements, you obtain what is needed to decode the bitstream at any level in the desired scalability dimension.

Incidentally, Motion JPEG 2000 is inherently scalable. For temporal scalability, this is a consequence of the frame-based nature of the compression. For spatial scalability, this results from the multi-resolution representation of images in the JPEG 2000 algorithm. This is another reason that made some people believe JPEG 2000 could win the digital CTTV contest. Today, the question is solved. SVC brought what H.264 was missing in preserving the bit rate improvement of stream-based compression. JPEG 2000 is out of the running.

H.264 components

Right now, H.264 is confirmed as the technology of choice for digital CCTV. We have seen that the cost of implementing the technology mainly resides at the encoding side of video surveillance systems. This is related to the need of specialized components supporting massive H.264 compression, as dictated by the high camera count.

In designing your H.264-based system, you will be faced with an architectural alternative:

- Cameras with built-in individual H.264 compression capability.
- Centralized H.264 compression devices gathering images from a set of non-compressing camera.

This is an open question with no definite answer, involving trade-offs regarding the distance between cameras, application constraints, cost of wiring and cost of components. H.264-capable components are already available on the market. As they are at the beginning of their existence, you may expect cost reductions sooner or later.

Cost effective high-end video surveillance applications use off-the-shelf H.264-capable components.

What we can safely say today is that high-class CCTV applications are more cost-effective with centralized solutions. By high-class, we mean (1) video quality comparable with SDTV broadcast video or better, (2) clusters of 10 monitoring cameras or more. For that class of applications, in 2008, a video surveillance server will be advantageously based on a PC platform implementing H.264 compression in add-in dedicated hardware[16]. The PC will offer all storage, networking, user interaction and display facilities on a standard basis.

In any case, we expect an abundance of H.264 components in the future. We hope this paper has provided you with useful hints to understand what you can expect from them, and how to qualify their technical performance.

[16] *A modern H.264 capable capture board is described here: www.euresys.com/Products/picolo/PicoloV16H264.asp*