

Not a TechOnLine User?
Click Here.
It's Free! [Register](#)

TechOnLine Publication Date: Feb. 3, 1998

An Overview of Video Compression Algorithms *Andrew Davis*

JPEG

For single-frame image compression, the industry standard with the greatest acceptance is JPEG (Joint Photographic Experts Group). JPEG consists of a minimum implementation (called a baseline system) which all implementations are required to support, and various extensions for specific applications. JPEG has received wide acceptance, largely driven by the proliferation of image manipulation software which often includes the JPEG compression algorithm in software form as part of a graphics illustration or video editing package. JPEG compressor chips and PC boards are also available to greatly speed up the compression/decompression operation.

JPEG, like all compression algorithms, involves eliminating redundant data. The amount of loss is determined by the compression ratio, typically about 16:1 with no visible degradation. If more compression is needed and noticeable degradation can be tolerated, as in downline loading several images over a communications link that only need to be identified for selection purposes by the recipient, compression of up to 100:1 may be employed.

JPEG compression involves several processing stages, starting with an image from a camera or other video source. The image frame consists of three 2-D patterns of pixels, one for luminance and two for chrominance. Because the human eye is less sensitive to high-frequency color information, JPEG calls for the coding of chrominance (color) information at a reduced resolution compared to the luminance (brightness) information. In the pixel format, there is usually a large amount of low-spatial-frequency information and relatively small amounts of high-frequency information. The image information is then transformed from the pixel (spatial) domain to the frequency domain by a discrete cosine transform (DCT), a DSP algorithm similar to the fast Fourier transform (FFT). This produces two-dimensional spatial-frequency components, many of which will be zero and discarded. Near-zero components are truncated to zero and need not be sent on, either. This quantization step is where most of the actual compression takes place. The remaining components are then entropy coded by the Huffman tree method which assigns short codes to frequent symbols and longer codes to infrequent symbols. This results in additional compression of about 3x.

Decompression reverses this procedure, beginning with the Huffman tree decoding and inverse DCT, transforming the image back to the pixel domain. Since the computational complexity is virtually identical in either direction, JPEG is considered a symmetrical compression method.

JPEG, while designed for still images, is often applied to moving images, or video. Motion JPEG is possible if the compression/decompression algorithm is executed fast enough (on a fast chip or chip set) to keep up with the video data stream, but at typical compression ratios of about 16:1,

there is not a considerable saving (compared to the 100:1 ratios of MPEG-1) and not good enough compression to play from a CD-ROM (which is constrained to transfer data at 150 kbps or 300 kbps, depending on rotation speed). However, full-motion JPEG will be employed in professional video processing, since there are no missing frames in the bit stream, and frame-by-frame editing can be precise. Note that JPEG does not address the question of audio tracks and audio/video synchronization.

MPEG

MPEG is the "Moving Picture Experts Group", working under the joint direction of the International Standards Organization (ISO) and the International Electro-Technical Commission (IEC). This group works on standards for the coding of moving pictures and associated audio.

MPEG involves fully encoding only key frames through the JPEG algorithm (described above) and estimating the motion changes between these key frames. Since minimal information is sent between every four or five frames, a significant reduction in bits required to describe the image results. Consequently, compression ratios above 100:1 are common. The scheme is asymmetric; the MPEG encoder is very complex and places a very heavy computational load for motion estimation. Decoding is much simpler and can be done by today's desktop CPUs or with low cost decoder chips.

The MPEG encoder may choose to make a prediction about an image and transform and encode the difference between the prediction and the image. The prediction accounts for movement within an image by using motion estimation. Because a given image's prediction may be based on future images as well as past ones, the encoder must reorder images to put reference images before the predicted ones. The decoder puts the images back into display sequence. It takes on the order of 1.1-1.5 billion operations per second for real-time MPEG encoding. In the past MPEG-1 compression was applied in a post-production process requiring expensive hardware and operator interaction for best results; now newer silicon is enabling some forms of MPEG compression on the desktop in real time.

MPEG-1, the Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbps, is an International Standard ISO-11172, completed in October, 1992. MPEG-1 is intended primarily for stored interactive video applications (CD-ROM); with MPEG-1, one can store up to 72 minutes of VHS quality (640 x 480 s 30fps) video and audio on a single CD-ROM disk. MPEG-1 can deliver full-motion color video at 30 frames per second from CD-ROM. Because audio is usually associated with full motion video, the MPEG standard also addresses the compression of the audio information at 64, 96, 128, and 192 kbps and identifies the synchronization issues between audio and video by means of time stamps. The first volume application for MPEG-1 decode chips (from C-Cube Microsystems) was a Karaoke entertainment system by JVC.

MPEG-2 is the "Generic Coding of Moving Pictures and Associated Audio." The MPEG-2 standard is targeted at TV transmission and other applications capable of 4 Mbps and higher data rates. MPEG-2 features very high picture quality. MPEG-2 supports interlaced video formats, increased image quality, and other features aimed at HDTV. MPEG-2 is a compatible extension of MPEG-1, meaning that an MPEG-2 decoder can also decode MPEG-1 streams. MPEG-2 audio will supply up to five full bandwidth channels (left, right, center, and two surround channels), plus an additional low-frequency enhancement channel, or up to seven commentary

channels. The MPEG-2 systems standard specifies how to combine multiple audio, video, and private-data streams into a single multiplexed stream and supports a wide range of broadcast, telecommunications, computing, and storage applications.

MPEG-3 was merged into MPEG-2 and no longer exists.

MPEG-4 is for very low bitrate coding and is scheduled to be available in draft form in late 1997 or early 1998. MPEG-4 and H.324 are likely to merge. MPEG-4 began life with a mandate by the Moving Picture Experts Group to create audiovisual coding schemes at very low bit rates for wireless telecommunications. The scope of the work has been expanded to target a broader spectrum of applications. MPEG-4 completion is scheduled for 1998. MPEG-4 is an inclusive superset of MPEG-1 and MPEG-2 and is based on a common communication language to describe tools, algorithms, and profiles necessary for coding of objects, rather than standardizing a coding algorithm. MPEG-4 specs will include methods for combining synthetic scenes or objects with natural scenes, and for coding and manipulating them without first converting the objects into video frames. MPEG-4 would also code audio and video objects at their native resolutions, supporting content-based manipulation and bit-stream editing without the need for transcoding.

MPEG starts with images in YUV color space and samples the U and V data at half the vertical and half the horizontal frequency as the luminance values. This scheme takes advantage of the reduced sensitivity of the human perceptual system to color changes compared to brightness changes.

The basic scheme is to predict motion from frame to frame in the temporal direction, and then to use DCT's (discrete cosine transforms) to organize the redundancy in the spatial directions. The DCT's are done on 8x8 blocks, and the motion prediction is done in the luminance (Y) channel on 16x16 blocks. For a 16x16 block in the current frame being compressed, the encoder looks for a close match to that block in a previous or future frame (there are backward prediction modes where later frames are sent first to allow interpolating between frames). The DCT coefficients (of either the actual data, or the difference between this block and the close match) are quantized. Many of the coefficients end up being zero. The quantization can change for every macroblock, which is 16x16 of Y and the corresponding 8x8's in both U and V. The results of all of this, which include the DCT coefficients, the motion vectors, and the quantization parameters are Huffman coded using fixed tables. The DCT coefficients have a special Huffman table that is two-dimensional that one code specifies a run-length of zeros and the non-zero value that ended the run. Also, the motion vectors and the DCT components are DPCM (subtracted from the last one) coded.

With MPEG there are three types of coded frames. "I" or intra frames are simply frames coded as individual still images; "P" or predicted frames are predicted from the most recently reconstructed I or P frame. Each macroblock in a P frame can either come with a vector and difference DCT coefficients for a close match in the last I or P, or it can just be "intra" coded if there was no good match. "B" or bidirectional frames are predicted from the closest two I or P frames, one in the past and one in the future. The encoder searches for matching blocks in those frames, and tries three different things to see which works best: using the forward vector, using the backward vector, and averaging the two blocks from the future and past frames and subtracting the result from the block being coded.

A typical sequence of decoded frames might be: IBBPBBPBBPBBIBBPBBPB... where there are

12 frames from I to I, based on a random access requirement that there should be a starting point at least once every 0.4 seconds or so. The ratio of Ps to Bs is based on experience.

H.261

H.261 (last modified in 1993) is the video compression standard included under the H.320 umbrella (and others) for videoconferencing standards. H.261 is a motion compression algorithm developed specifically for videoconferencing, though it may be employed for any motion video compression task. H.261 allows for use with communication channels that are multiples of 64 kbps ($P=1,2,3\dots30$), the same data structure as ISDN. H.261 is sometimes called Px64.

An overview of the H.261 CODEC, taken from the ITU reference documentation, shows the major components used to code and decode the bitstreams.

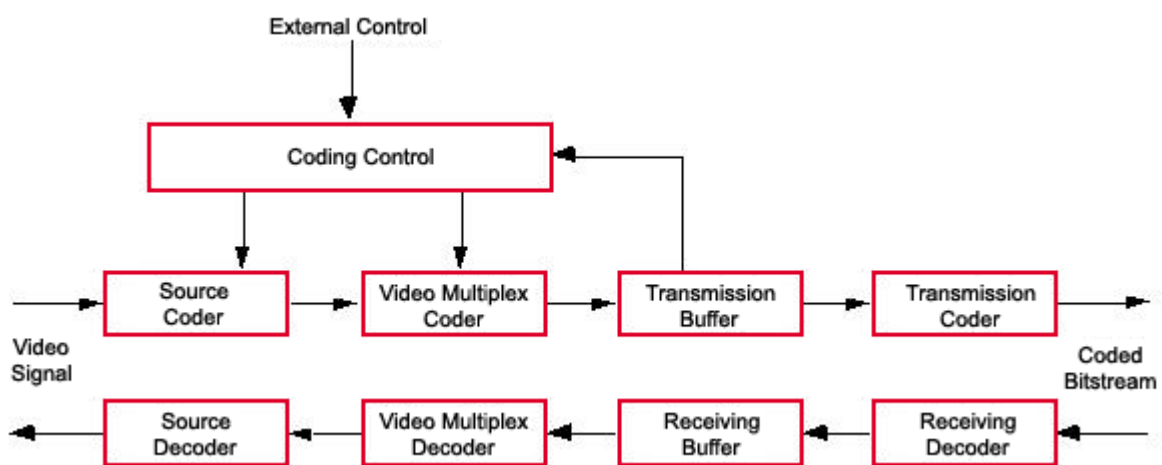


Figure 1: H.261 block diagram from ITU recommendation

Note: In actual fact, the ITU recommendations specify how a decoder must work and what it must support. In essence, this has the effect of standardizing the bitstream in and the bitstream out of the decompression functional block. The encoder design is unspecified, except of course that it must be compatible with the recommended decoder.

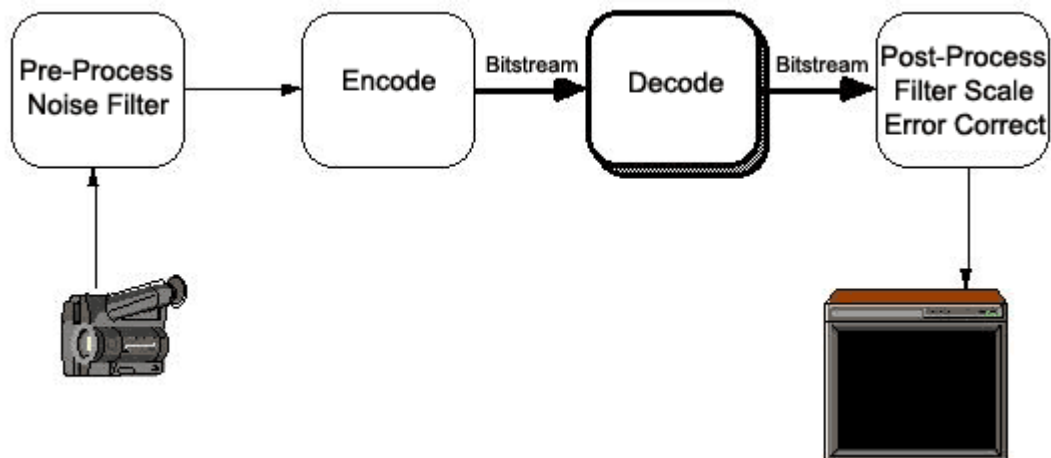


Figure 2: ITU structure for CODEC standards. Only parts in bold are specified.

To permit a single Recommendation to cover use in and between regions using 625- and 525-line television standards, the source coder operates on pictures based on a common intermediate format (CIF). The standards of the input and output television signals, which may, for example, be composite or component, analog or digital and the methods of performing any necessary conversion to and from the source coding format are not subject to Recommendation H.261. The video coder provides a self-contained digital bit stream which may be combined with other signals. The video decoder performs the reverse process. Pictures are sampled at an integer multiple of the video line rate. This sampling clock and the digital network clock are asynchronous.

H.261 encoding is based on the discrete cosine transform (DCT) and allows for fully-encoding only certain frames (INTRA-frame) while encoding the differences between other frames (INTER-frame). The main elements of the H.261 source coder are prediction, block transformation (spatial to frequency domain translation), quantization, and entropy coding. While the decoder requires prediction, motion compensation is an option. Another option inside the recommendation is loop filtering. The loop filter is applied to the prediction data to reduce large errors when using interframe coding. Loop filtering provides a noticeable improvement in video quality but demands extra processing power.

The operation of the decoder allows for many H.261-compliant CODECs to provide very different levels of quality at different cost points. One example is quantization. A large part of the compression and data rate control provided under H.320 occurs in the quantization step. As the quantization level rises, fewer bits are needed to specify all the frequency components, and higher frequencies may be eliminated altogether, which may cause loss of image sharpness. While some implementations employ a fixed quantization level that is based on line rate, the better solution is to dynamically adapt the quantization level based on the image content. The H.261 standard does not specify a particular adaptive quantization method.

The transmitted bit-stream contains a BCH code (Bose, Chaudhuri, and Hocquengham) forward error correction code. Use of this by the decoder is also optional. Features necessary to support switched multipoint operation are included.

The H.261 source coder operates on non-interlaced pictures occurring 30 000/1001 (approximately 29.97) times per second. The tolerance on picture frequency is ± 50 ppm. Pictures are coded as luminance and two color difference components (Y , C_B , and C_R). These components and the codes representing their sampled values are as defined in CCIR Recommendation 601: black = 16, white = 235, zero color difference = 128, and peak color difference = 16 and 240. These values are nominal ones and the coding algorithm functions with input values of 1 through to 254.

Unlike many other CODEC algorithms that are resolution and image size independent, H.261 specifies two picture formats. In the first format (CIF), the luminance sampling structure is 352 pixels per line, 288 lines per picture in an orthogonal arrangement. Sampling of each of the two color difference components is at 176 pixels per line, 144 lines per picture, orthogonal. Color difference samples are sited such that their block boundaries coincide with luminance block boundaries. The picture area covered by these numbers of pixels and lines has an aspect ratio of 4:3 and corresponds to the active portion of the local standard video input. The second format, quarter-CIF (QCIF), has half the number of pixels and half the number of lines stated above. All CODECs must be able to operate using QCIF. CIF support is optional.

H.261 requires the designer to provide a means to restrict the maximum picture rate of encoders by having at least 0, 1, 2, or 3 non-transmitted pictures between transmitted ones. Selection of this minimum number and selection of CIF or QCIF are by external means, typically by the Coding Control block in the functional diagram and via H.221.

An overview of the H.261 source coder, taken from the ITU reference documentation, shows the relationship between the DCT, prediction, and motion estimation logic flow.

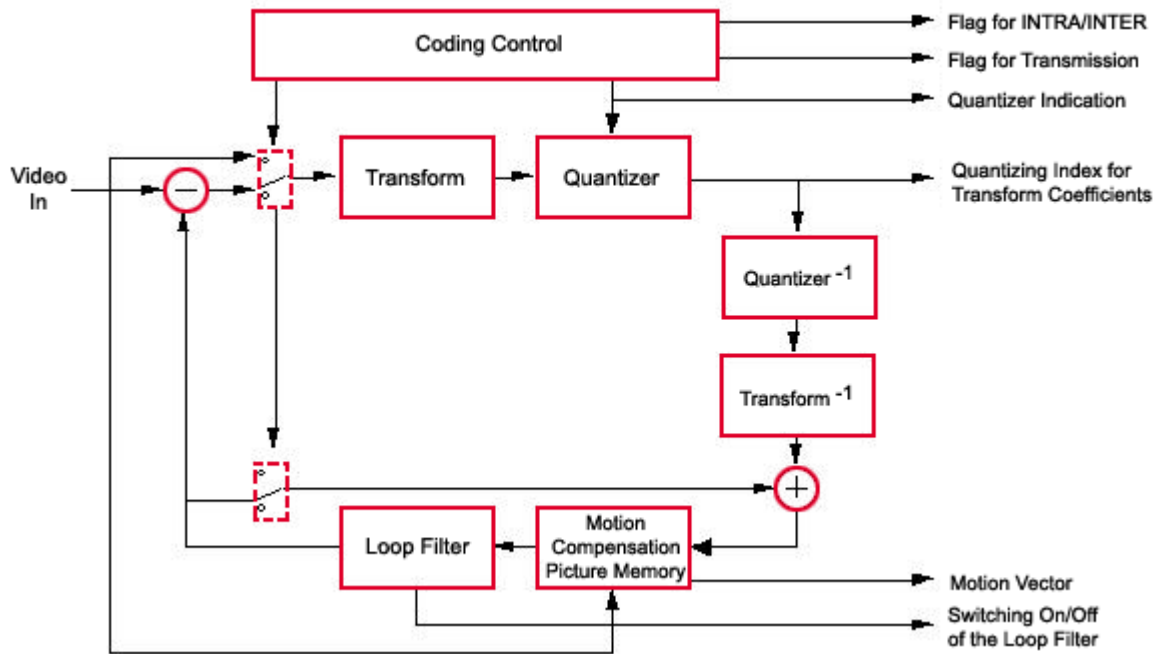


Figure 3: H.261 source coder block diagram

It is important to understand the hierarchical structure of video data used by H.261. At the top layer is the picture. Each picture is divided into groups of blocks (GOBs). Twelve GOBs make up a CIF image; three make up a QCIF picture. A GOB relates to 176 pixels by 48 lines of Y and the spatially corresponding 88 pixels by 24 lines for each chrominance value.

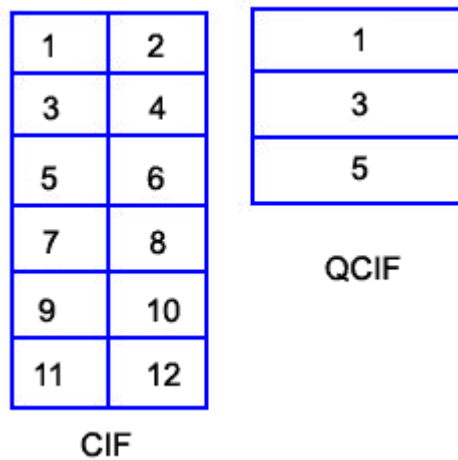


Figure 4: Arrangement of groups of blocks in an H.261 picture

Each GOB is divided into 33 macroblocks. A macroblock relates to 16 pixels by 16 lines of Y

and the spatially corresponding 8 pixels by 8 lines of each chrominance value. Macroblocks are the basic element used for many prediction and motion estimation techniques.

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33

Figure 5: Arrangement of macroblocks within a group of blocks under H.261

At the bottom level of the hierarchy are blocks which consist of 8x8 pixel arrays of luminance values and two 4x4 arrays of chrominance values. Four luminance blocks and the spatially corresponding color difference blocks make up a macroblock.

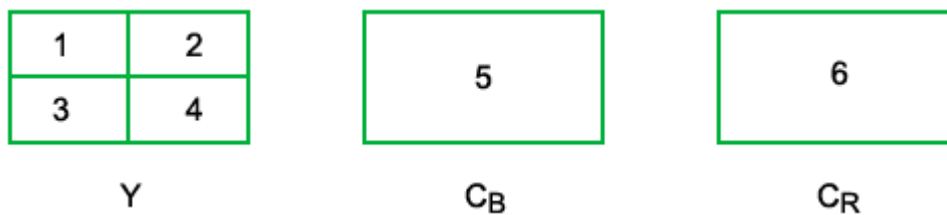


Figure 6: Arrangement of blocks within a macroblock under H.261

When an H.261 controller decides to perform an intraframe compression or an interframe compression, or when it segments data as transmitted or non-transmitted, these decisions are made on a block-by-block basis, not on a picture-by-picture basis.

If the top switch in the block diagram is set by the controller to the up position, then an INTRA mode will occur; the data will be transformed and quantized and transmitted. In the INTER mode, each block is encoded and then decoded, compared to the same block from the next image, and if the differences are small, the block is not transmitted. If the differences are not small, in other words, there has been sufficient change, then the encoder transforms, quantizes, entropy encodes, and transmits the differences. The definition of sufficient change can vary from block to block. (Change is typically due to motion, but can also be falsely inferred from noise. Hence, noise filters on the video signal, which are outside the scope of the standard, can add great value to H.261-compliant products. Simple low pass filtering will reduce noise but also reduce image sharpness; more sophisticated approaches yield better results but cost more to implement. Macroblocks carry a flag to indicate whether they are predicted or intraframe, and a second flag to indicate whether the data should be transmitted or not. The criteria for choice of mode and transmitting a block are not detailed by the recommendation and may be varied dynamically as part of the control strategy. Hence they are vendor- and product-specific.

Forced updating is achieved by forcing the use of the INTRA mode of the coding algorithm. The update pattern is not defined. For control of accumulation of inverse transform mismatch error, a macroblock must be forcibly updated at least once every 132 times it is transmitted.

While prediction is inter-picture and required, motion control is allowed, but optional, as is a loop filter shown in the coder block diagram. With motion compensation (MC) the H.261 decoder will accept one vector per macroblock. Both horizontal and vertical components of these

motion vectors must have integer values not exceeding ± 15 . The same vector is used for all four luminance blocks in the macroblock. The motion vector for both color difference blocks is derived by halving the component values of the macroblock vector and truncating the magnitude parts towards zero to yield integer components.

A positive value of the horizontal or vertical component of the motion vector signifies that the prediction is formed from pixels in the previous picture which are spatially to the right or below the pixels being predicted. Motion vectors are restricted such that all pixels referenced by them are within the coded picture area. Note that the Recommendation specifies only enough information for a decoder to use motion vectors; the Recommendation does not specify how motion vectors are to be calculated. This again, leaves room for product differentiation.

The prediction process may be modified by a two-dimensional spatial filter which operates on pixels within a predicted 8x8 block. The filter is separable into one-dimensional horizontal and vertical functions. Both are non-recursive with coefficients of 1/4, 1/2, 1/4 except at block edges where one of the taps would fall outside the block. In such cases the 1-D filter is changed to have coefficients of 0, 1, 0. Full arithmetic precision is retained with rounding to 8-bit integer values at the 2-D filter output. Values whose fractional part is one half are rounded up. The filter is switched on/off for all six blocks in a macroblock according to the macroblock type.

In summary, like MPEG, H.261 encoding is DCT-based (compression ratios of 80 to 100:1 are typical, but can also go as high as 500:1) and calls for fully-encoding only certain frames. However, instead of required motion estimation and its severe computational load, H.261 calls for coding only the difference between a frame and the previous frame. Motion compensation involves working with groups of pixels (16x16 macroblocks) to identify a group in the previous frame that best matches a group in the current frame, coding the difference along with a vector that describes the offset (movement) of that group. As in JPEG and MPEG, the remaining data is entropy coded by hierarchical Huffman coding for even greater compression.

Unlike JPEG and MPEG, which are resolution- and image-size independent, Px64 specifies two image sizes, either common interchange format (CIF), which is 352x288, or quarter CIF (QCIF), which is 176x144. Both MPEG and H.261 use prediction and motion estimation to reduce temporal redundancy, but differ in their approach. MPEG's design center is to maintain picture quality with maximum compression. H.261, intended for telephony, minimizes encoding and decoding delay while achieving a fixed data rate. H.261 implementations allow a tradeoff between frame rate and picture quality. As the motion content of the images increases (subject moves, for example), the CODEC has to do more computations and usually has to give up on image quality to maintain frame rate, or the reverse. Furthermore, H.320, through H.221, supports dynamic bandwidth allocation every 50 ms. Bandwidth allocation with most other codecs is fixed.

To design an H.261 CODEC covering the full scope of the standard, in other words, 30 fps with full motion estimation and loop filtering, the video codec subsystem must be able to execute approximately 8 billion operations/sec (Gops). Most of this is for motion estimation. However it is possible to reduce the number of operations required at the expense of picture quality. For example, a hierarchical 4:1 decimated motion-estimation algorithm can bring the system requirements down to about 1.5 Gops. Another scheme is to limit the video to 15 fps, often imposed by ISDN bandwidths anyway.

H.263

H.263 is the video codec introduced with H.324, the ITU recommendation "Multimedia Terminal for Low Bitrate Visual Telephone Services Over the GSTN". H.324 is a comprehensive, flexible umbrella recommendation covered in detail elsewhere in this report. H.324 is for videoconferencing over the analog phone network (POTS). While video is an option under H.324, any terminal supporting video must support both H.263 and H.261.

H.263 is a structurally similar refinement (a five year update) to H.261 and is backward compatible with H.261. At bandwidths under 1000 kbps, H.263 picture quality is superior to that of H.261. Images are greatly improved by using a required 1/2 pixel new motion estimation rather than the optional integer estimation used in H.261. Half pixel techniques give better matches, and are noticeably superior with low resolution images (SQCIF). Optional support for H.263 was amended to H.320 in 1996.

Both H.261 and H.263 algorithms are available in host-software formats, and both are already supported by several coprocessor chips out in the market. The basic configuration of the H.263 algorithm is based on ITU-T Recommendation H.261.

Like H.261, the decoder has motion compensation capability, allowing optional incorporation of this technique in the coder. Half-pixel precision is used for the motion compensation, as opposed to Recommendation H.261 where full-pixel precision and a loop filter are used. Variable length coding is used for the symbols to be transmitted. In addition to the basic video source coding algorithm, four negotiable coding options are included for improved performance: Unrestricted Motion Vectors, Syntax-based Arithmetic Coding, Advanced Prediction, and PB-frames. All these options can be used together or separately, except the Advanced Prediction mode which requires the Unrestricted Motion Vector mode to be used at the same time.

The H.263 block diagram is similar to that of H.261.

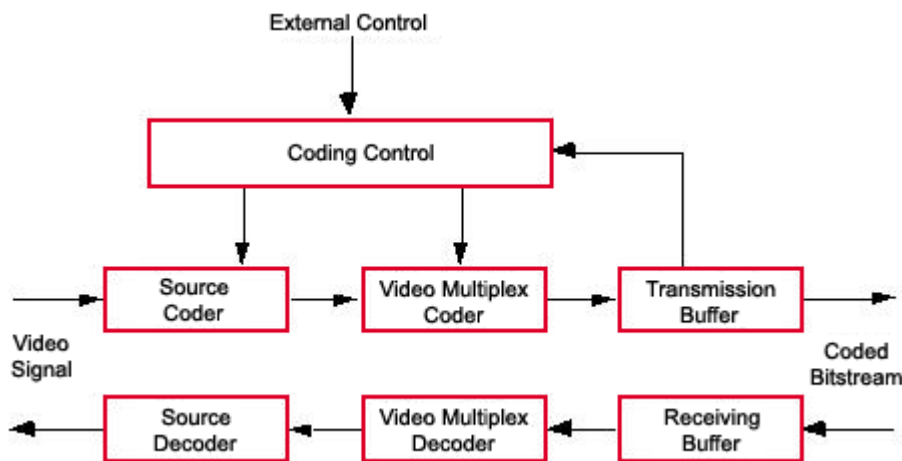


Figure 7: H.263 overall block diagram

H.263 uses a hybrid of inter-picture prediction to utilize temporal redundancy and transform coding of the remaining signal to reduce spatial redundancy is adopted.

With the Unrestricted Motion Vector mode, motion vectors are allowed to point outside the picture. The edge pixels are used as prediction for the "not existing" pixels. With this mode, a

significant gain is achieved if there is movement across the edges of the picture, especially for the smaller picture formats.

The Syntax-based Arithmetic Coding mode means that arithmetic coding is used instead of variable length coding. The SNR and reconstructed frames will be the same, but significantly fewer bits will be produced.

In the Advanced Prediction mode, overlapped block motion compensation (OBMC) is used for the luminance part of P-pictures. Four 8x8 vectors instead of one 16x16 vector are used for some of the macroblocks in the picture. The encoder has to decide which type of vectors to use. Four vectors use more bits, but give better prediction. The use of this mode generally gives a considerable improvement. A subjective gain is achieved because OBMC results in less blocking artifacts.

Also new in H.263 are PB frames. A PB-frame consists of two pictures being coded as one unit. The name PB comes from the name of picture types in MPEG where there are P-pictures and B-pictures. Thus a PB-frame consists of one P-picture which is predicted from the last decoded P-picture and one B-picture which is predicted from both the last decoded P-picture and the P-picture currently being decoded. This last picture is called a B-picture, because parts of it may be bidirectionally predicted from the past and future P-pictures. With this coding option, the picture rate can be increased considerably without increasing the bitrate significantly.

Error handling in H.263 is provided by external means, for example as specified in Recommendation H.223. If it is not provided by external means an optional error correction code and framing can be used.

A decoder can send a command to encode one or more GOBs of its next picture in INTRA mode with coding parameters in order to avoid buffer overflow. A decoder can also send a command to transmit only non-empty GOB headers. The transmission method for these signals is by external means (for example per Recommendation H.245).

Although H.263 specifies five standardized picture formats (sub-QCIF, QCIF, CIF, 4CIF and 16CIF), the last two are unlikely to see use on the desktop and CIF is unlikely over POTS lines. For each of the picture formats, color difference samples are sited such that their block boundaries coincide with luminance block boundaries. The 4:3 pixel aspect ratio is the same for each of these picture formats.

Picture Format	# pixels luminance	# lines luminance	# pixels chrominance	# lines chrominance	H.261	H.263
sub-QCIF	128	96	64	48	optional	required
QCIF	176	144	88	72	required	required
CIF	352	288	176	144	optional	optional
4CIF	704	576	352	288	NA	optional

16CIF	1408	1152	704	576	NA	optional
-------	------	------	-----	-----	----	----------

Table 1: H.263 picture formats

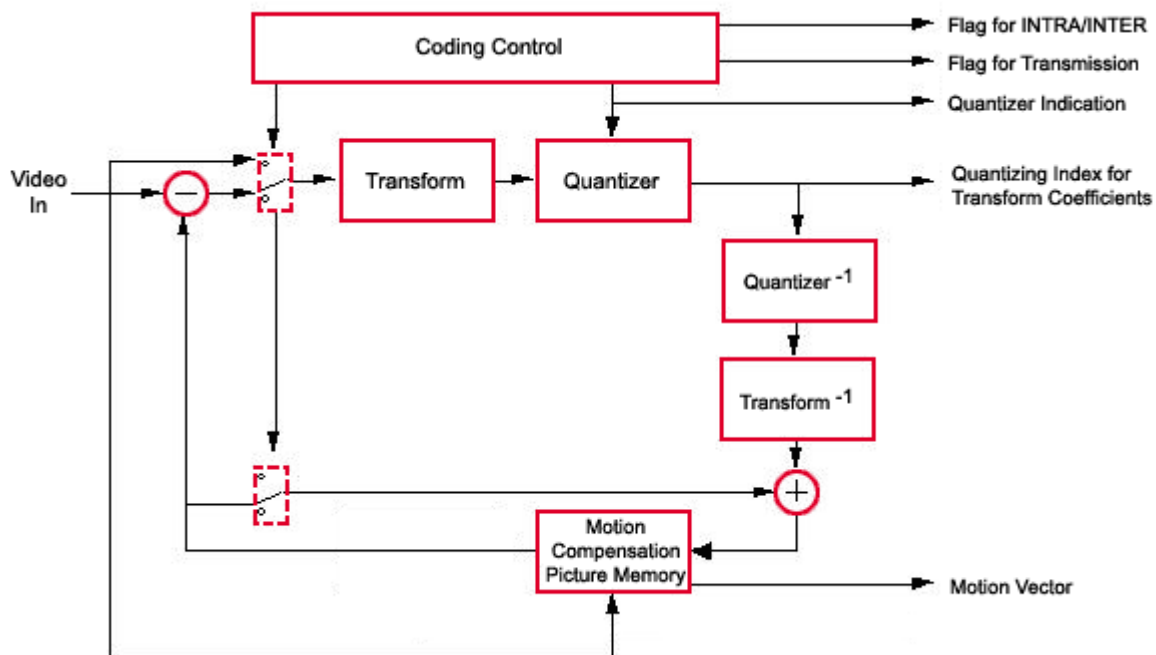


Figure 8: H.263 Source coder

With H.263, as with H.261, each picture is divided into groups of blocks (GOBs). A group of blocks (GOB) comprises of $k \cdot 16$ lines, depending on the picture format ($k = 1$ for sub-QCIF, QCIF and CIF; $k = 2$ for 4CIF; $k = 4$ for 16CIF). The number of GOBs per picture is 6 for sub-QCIF, 9 for QCIF, and 18 for CIF, 4CIF and 16CIF.

Each H.263 GOB is divided into macroblocks. A macroblock relates to 16 pixels by 16 lines of Y and the spatially corresponding 8 pixels by 8 lines of CB and CR. Further, a macroblock consists of four luminance blocks and the two spatially corresponding color difference blocks. Each luminance or chrominance block relates to 8 pixels by 8 lines of Y, CB or CR. A GOB comprises one macroblock row for sub-QCIF, QCIF and CIF, two macroblock rows for 4CIF and four macroblock rows for 16CIF.

The criteria for choice of mode and transmitting a block are not subject to recommendation and may be varied dynamically as part of the coding control strategy. Transmitted blocks are transformed and resulting coefficients are quantized and entropy coded.

H.263 prediction is inter-picture and may be augmented by motion compensation. (This allows vendor differentiation.) The coding mode in which prediction is applied is called INTER; the coding mode is called INTRA if no prediction is applied. The INTRA coding mode can be signalled at the picture level (I-picture for INTRA or P-picture for INTER) or at the macroblock level in P-pictures. In the optional PB-frames mode B-pictures are always coded in INTER mode. The B-pictures are partly predicted bidirectionally.

Motion compensation is optional in the H.263 encoder. The decoder will accept one vector per macroblock or if the Advanced Prediction mode of H.263 is used one or four vectors per

macroblock. If the PB-frames mode is used, an additional delta vector can be transmitted per macroblock for adaptation of the forward motion vector for prediction of the B-macroblock. Both horizontal and vertical components of the motion vectors have integer or half integer values restricted to the range [-16,15.5].

Normally, a positive value of the horizontal or vertical component of the motion vector signifies that the prediction is formed from pixels in the previous picture which are spatially to the right or below the pixels being predicted. The only exception is for the backward motion vectors for B-pictures, where a positive value of the horizontal or vertical component of the motion vector signifies that the prediction is formed from pixels in the next picture which are spatially to the left or above the pixels being predicted. Motion vectors are restricted such that all pixels referenced by them are within the coded picture area, except when the Unrestricted Motion Vector mode is used.

Several parameters may be varied to control the rate of generation of H.263 coded video data. These include processing prior to the source coder, the quantizer, block significance criterion and temporal subsampling. The proportions of such measures in the overall control strategy are not subject to recommendation but rather left up to the decision of the individual designer. When invoked, temporal subsampling is performed by discarding complete pictures. A decoder can signal its preference for a certain tradeoff between spatial and temporal resolution of the video signal. The encoder shall signal its default tradeoff at the beginning of the call and shall indicate whether it is capable to respond to decoder requests to change this tradeoff. The transmission method for these signals is by external means, for example Recommendation H.245. Forced updating is the same as in H.261.

H.263 establishes a framework for many future improvements, likely to be implemented over time with new and more powerful video silicon. H.263, like H.261, is flexible. For example, H.263 allows (but does not require) implementation of PB (predictive) frames as well as the I (DCT coded frames) in the codec. This is similar to the approach used by MPEG. While PB frames add greatly to the computational load and frame delay, they also add quality to the video stream by raising the frame rate. H.263 also incorporates an optional, advanced motion estimation technique which is estimated to be 4x as complex as the technique used in H.261. All these hooks give the standards room to grow over time. Compatibility and interoperability can be maintained, while the quality of audio and video can improve with inevitable improvements in silicon price/performance.