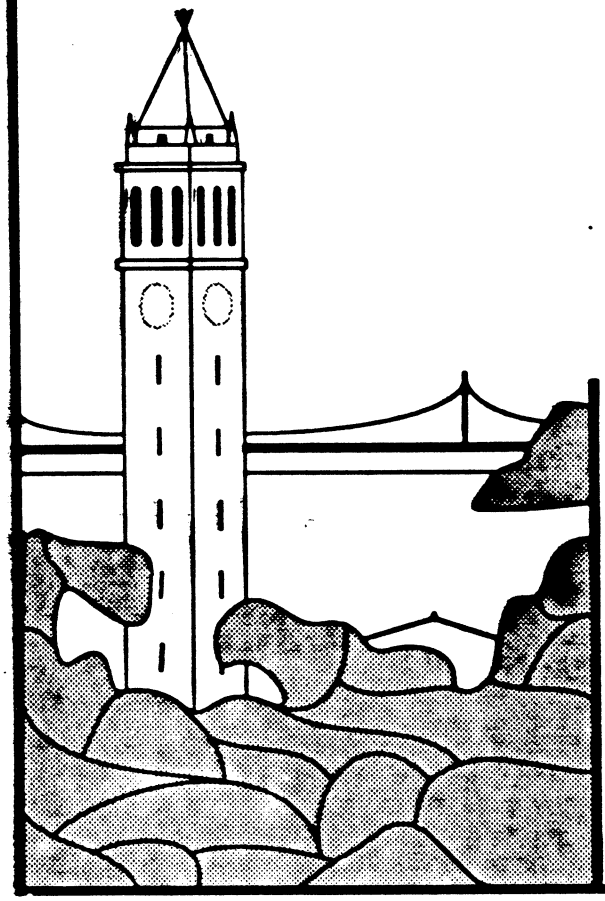
A Bitrate Control Algorithm for the Berkeley MPEG-1 Video Encoder

Darryl C. Brown



Report No. UCB/CSD-97-937

May 1995

Computer Science Division (EECS)
University of California
Berkeley, California 94720

A Bitrate Control Algorithm for the Berkeley MPEG-1 Video Encoder†

Darryl C. Brown
Computer Science Division - EECS
University of California
Berkeley, California 94720

Abstract

A simple bitrate control algorithm was implemented in the Berkeley MPEG-1 Video Encoder. A series of experiments are presented that illustrate the effect of bitrate control.

1. Introduction

MPEG is an international standard for compressed audio and video that will be widely used in multimedia applications and interactive television. There are two standards in use today. MPEG-1 is designed to deliver VHS-quality video at 1.5 megabits/second (Mbs)[3], while MPEG-2 is designed to deliver studio-quality video at 4-8 Mbs[2]. While a video stream can be encoded as a variable bitrate stream (VBR), many applications require a constant bitrate stream (CBR) to satisfy other resource constraints (e.g., CD-ROM transfer rates or static allocation of cable transmission bandwidth).

The Berkeley MPEG encoder is a freely-distributed software package that encodes video using MPEG-1 syntax and semantics [4]. The current version of the encoder produces VBR streams. We implemented the bitrate control algorithm described in the MPEG-2 standard and conducted a series of experiments to illustrate the effect of rate control.

The remainder of this report is organized as follows. Section 2 describes the MPEG-1 standard. Section 3 reviews previous work on rate control algorithms. Section 4 describes the rate control algorithm that we implemented in the Berkeley encoder. Section 5 presents the results of various experiments that show the effect of rate control on bits/picture and picture quality.

2. MPEG Video

The MPEG-1 standard was defined by the Motion Picture Experts Group (MPEG) of the International Standards Organization (ISO) [3]. The standard specifies the compression of an audio and video stream by defining the syntax of a compressed bitstream. Audio and video information is interleaved in the bit stream along with system information required to decode it (e.g., time codes that specify when a picture should be played).

Compression is necessary to allow conventional video material to be used in computer applications. A full-screen image may contain 1 million pixels, each requiring three bytes to represent the color and brightness. The standard film rate of 24 frames/second (fps) results in a data stream of 72 megabits/second (Mb/s) or approximately 260 gigabytes/hour (GB/h). The transmission and storage required is clearly beyond the capabilities of current systems. A *standard*, non-propriety compression algorithm such as MPEG will facilitate interoperability between the products of different companies. In turn, this standard will lead to more widespread use of multimedia data and manufacturing economies of scale which will decrease hardware and software costs.

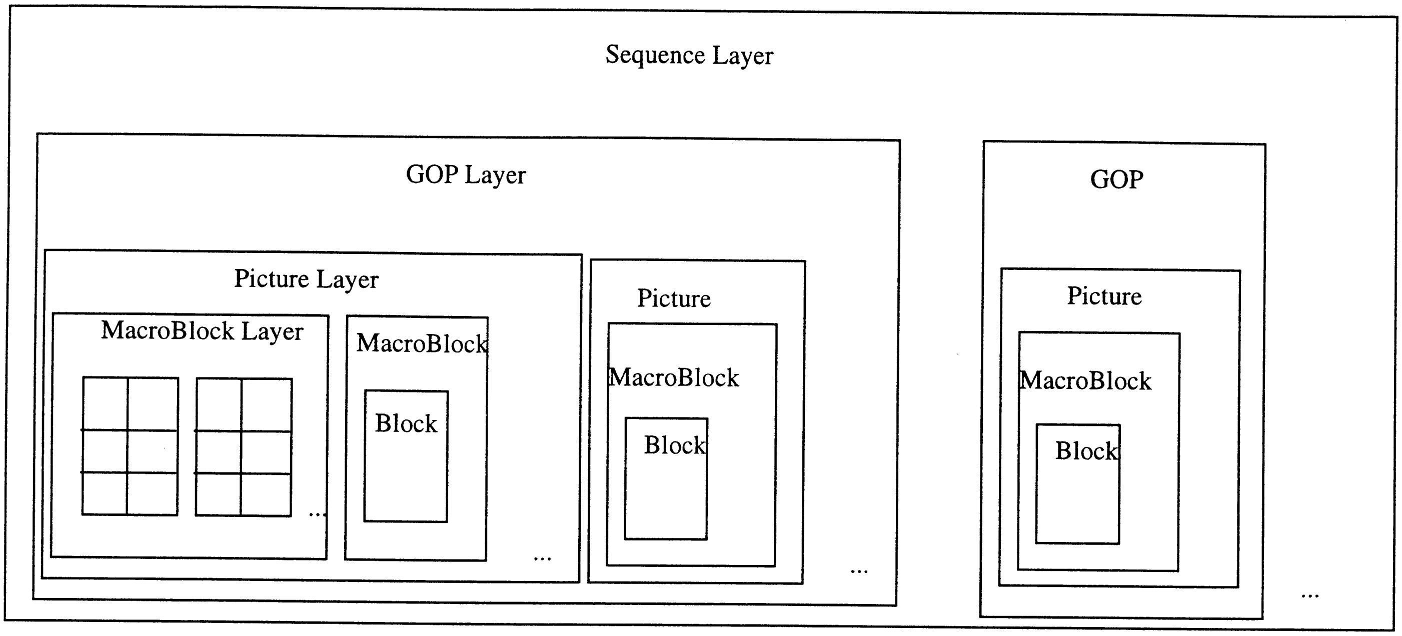
MPEG uses an algorithm similar to the JPEG (Joint Photographic Experts Group) baseline compression algorithm. A video sequence, which is essentially a sequence of still images, is compressed by applying a series of operations to each image: transform coding, quantization, run-length coding, entropy coding, and, for some images, motion compensation. The usual rule of thumb is that JPEG images can be compressed to 1 bit/pixel (bpp) without noticeable effects. MPEG, which adds motion compensation, improves the compression to about 0.5 (bpp).

An MPEG video stream is organized as a hierarchy of layers, as shown in Figure 1. The first layer, called the Sequence Layer, defines the overall video sequence. Context information for the stream is contained here, such as the image size and picture rate. Although the MPEG standard allows variability in how a sequence is coded, we will use only the parameter settings called Constrained Parameters Bitstream (CPB), which is designed for CD-ROM playback. These settings bound a sequence to no less than CCIR size (720x576 pixels), 30 fps and about 1.5 Mbs data rate. This is a compromise within the large range of permitted sequences, which any MPEG encoder should be able to play

Below the Sequence layer is the Group of Pictures (GOP) Layer, which provides a random access entry point. The decoder can always begin decoding at the start of a GOP, without needing to reference the pictures that come before or after.

The next layer is the Picture Layer, which contains actual image data. MPEG uses the YCrCb color space that

Figure 1: MPEG Layers



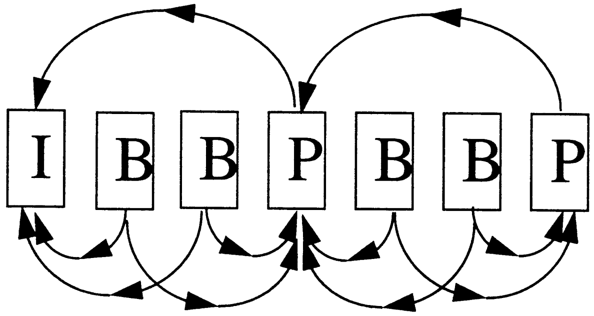
contains a luminance (brightness) component (Y) and two chrominance (color) components (Cr and Cb) for each picture element. This color space is comparable to the more familiar Red-Green-Blue (RGB) color space.

The amount of video data input to the encoder is reduced by subsampling the chrominance values of the image. Research has shown that hue is less perceptually significant than brightness. Accordingly, the chrominance values are specified for every other picture element in both the horizontal and vertical directions, resulting in a 50% reduction of data.

Each picture is encoded as one of three types of frames: intracoded (I), predicted (P), or bidirectional (B). I-frames are encoded using solely the information in that frame. The coding algorithm is nearly identical to the JPEG still image algorithm. P-frames use information from the image being coded or from a previous frame, while B-frames can use information from the frame being coded or from a previous or future frame. A GOP starts with an I-frame, which is why it can be accessed randomly. A sequence of P- and B-frames, typically in a fixed pattern, follows the I-frame.

Figure 2 below shows a typical GOP pattern with arrows indicating the dependencies among the frames. P-frames may use the preceding I- or P-frame as a reference frame. B-frames can use the preceding or following I- and P-frames as reference frames. Therefore, both frames must be present when the B-frames are reconstructed in the decoder. By necessity then, the GOP sequence that is displayed in the order: $I_1B_2B_3P_4B_5B_6P_7$, must be transmitted in the order: $I_1P_4B_2B_3P_7B_5B_6$, to ensure that the proper reference frames are available when decoding.

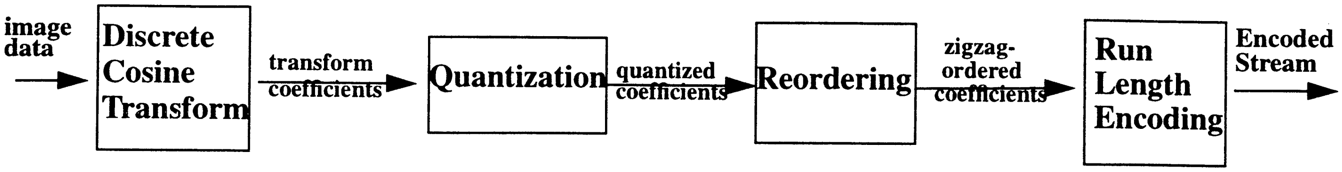
Figure 2: Frame Dependencies



A frame is encoded by breaking the image up into macroblocks and coding each one separately. A macroblock contains six subblocks: four luminance blocks and two chrominance blocks. In other words, a 4:2:0 digital representation is used. Each subblock is an 8x8 block of pixels, so the macroblock represents a 16x16 pixel area of the image. The macroblock serves as the basic unit for motion compensation, while subblocks are the coding units to which the Discrete Cosine Transform (DCT) is applied.

Motion compensation takes advantage of the fact that much of an image typically remains unchanged within a short sequence of video. If the camera is motionless, the

Figure 3: Compression Pipeline



background will be static, while characters interact in the foreground, for example. In this case, it is unnecessary to transmit the portion of the frame that is unchanging over time. More interestingly, it is possible to use compression on those elements of the scene which *do* change. Over time a particular area of the frame in the video scene, will "move." That is, the same object will appear in different locations on the screen in succeeding frames. This effect will occur as the object moves across the screen, or as the camera "pans" across a scene in the opposite direction. The encoding process takes advantage of this motion by allowing for motion estimation prediction. Blocks can be coded with motion vectors that tell where this block is in an adjacent frame.

For example, a camera may pan across a map of the United States from West to East. The state of Indiana will become visible about midway through the scene. Once revealed, the state will not change shape if the camera angle remains the same. Therefore, it is unnecessary to recode and retransmit the information about that state repeatedly. Instead, the next frame can merely refer to the location of Indiana in the previous picture and reposition it. It requires less data to send the location information than to encode the image. The MPEG algorithm exploits this fact. A search algorithm in the encoder examines macroblocks in adjacent frames to see if the current macroblock appears in those frames. If so, the difference in the location of the block will be passed as "motion vectors" between the two frames.

A motion vector is a pair (x-offset, y-offset) that specifies the position of the matching macroblock relative to the macroblock being coded. P-frames can only use motion vectors for succeeding frames. B-frames can use motion vectors to a forward or backward frame or even both.

To give better fidelity, the difference between luminance blocks is passed as well. Thus, if the lighting changed during the example scene and Indiana was obscured by shadow, this change is passed and motion prediction still utilized, even though the area no longer looked quite the same. The luminance error is passed because it yields a perceived improvement at little cost. B-frames can use the pixel values from past or future frames, or a combination of both as the value of the blocks. In this case, the resulting image is an "interpolated" average of the two predicting images. For both frame types, if a motion vector that is a good match is not found in adjacent frames, the macroblock can be encoded directly, as in I-frames. Or, if the motion vector is (0,0) and the luminance change is small, the macroblock is coded as a skipped block, which transmits nothing.

Figure 3 shows the basic operations in the coding pipeline. The DCT is used to compress both image data and the prediction-error from motion estimation. The DCT works on 8x8 blocks of pixels to produce 64 transform coefficients. It concentrates the energy of the block into a few coefficients, weighted toward the low-frequency range which is in the upper left corner of the block. Theoretically, this step is lossless, within the limits of the accuracy of the mathematical implementation and underlying hardware. The inverse DCT function (IDCT) can be applied be to reconstruct the blocks.

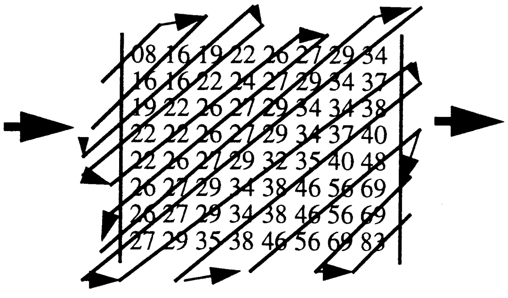
In the "middle" of the MPEG compression pipeline is the quantization step. That is, quantization follows motion compensation and transform coding, but precedes run-length encoding. Quantization serves to reduce the magnitude of the transform coefficients and increases the number of zero coefficients, which leads to more efficient compression. Quantization is the sole means of controlling the bitrate and, along with run-length coding, contributes most to the compression achieved by the MPEG algorithm.[2] This step is a "lossy" operation, as the information discarded by quantization cannot be fully reconstructed.

Image blocks in I-frames are quantized by dividing the transformed values of each 8x8 pixel block by the elements of a Quantization matrix (Qmatrix). The standard specifies a default Qmatrix or the encoder can use a custom matrix specified in the sequence header. The Qmatrix is designed to increase compression with minimal visual effects. The default matrix for intracoded blocks contains integer values steadily increasing from 8 to 83, in the zigzag order of the Huffman encoding, as shown below.

Table 1: Qmatrix for intracoded blocks

Figure 4: Reordering by Zigzag Scanning Pattern

$$\begin{vmatrix} 08 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{vmatrix}$$



08, 16, 16, 19, 16, 19, 22, 22, ... 69, 69, 83

$$\begin{vmatrix} 08 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{vmatrix}$$

This pattern allows coarser quantization of higher frequencies, which contribute little to image quality and serves to emphasize the low-frequency information, which is more significant to quality perception. By contrast, all entries in the default Qmatrix for non-intracoded blocks are set to 16. The data in predicted blocks is error data, rather than image values, so some believe it is a bad idea to make the same assumptions about the relative importance of high-frequency data.

To provide finer control over how much quantization is applied, the Qmatrix is scaled by multiplying each element by a constant, named MQUANT, which ranges between 1 and 31. Using higher MQUANT values makes the elements of the Qmatrix larger, which reduces the magnitude of the transform coefficients and thereby increases the number of zero coefficients after quantization. This scaling is the sole means of dynamically varying the size of the output, at the macroblock level. The price for this control is an overhead of one signal bit and five bits for the value itself. An overhead of six bits per macroblock is not very significant for I- or even P-frames. However, the overhead can be onerous for B-frames, which often require only 5-15 bits/block to code.

Huffman coding is used at the end of the compression process, on the quantized transform coefficients. These values are reorganized by a zigzag scanning pattern to order the high-energy, low-frequency coefficients first, as can be seen in Figure 4, in which the default Qmatrix is reordered.

Putting the higher frequency coefficients last tends to increase the runs of coefficients with zero values which improves run-length coding. Coefficients are represented by a number pair telling the amplitude of the coefficient and the length of the run of consecutive zeroes that follow. Common pairs are assigned short Huffman codes, while less likely combinations are coded by an escape code followed by the sequence.

3. Previous Work on Rate Control

As noted in the introduction, a VBR coder does not attempt to regulate or even keep track of the output bits while encoding a video sequence. The number of bits produced for each picture will tend to vary, as the complexity of the scene being coded can change dramatically over time. Additionally, a major tactic to reduce the bits required to code a picture is to predict the pixels either from other pixels in the same frame or from pixels in other frames. Consequently, unpredictable picture changes will require more bits to encode.

The encoder can control the output bitrate by varying the quantization scale. Varying the coarseness of the quantization step applied will directly affect the size of the output. However, simple rate control is necessary, but not sufficient for optimal results because coarse quantization significantly reduces the bitrate while reducing image quality at the same time. It is preferable to distribute the bits available amongst the frames, and the macroblocks within those frames, in an intelligent manner so as to avoid visible artifacts in the reconstructed image.

The quality of the frames used as references must be the highest, as errors there will propagate to other frames in a GOP. For this reason, it is preferable to allocate proportionally more bits to encode I- and P-frames. Few bits are allocated to B-frames because they do not act as a reference frame and cannot spread errors to other frames. Furthermore, B-frames benefit from more extensive motion-compensation tactics and are therefore naturally the smallest frames. A typical ratio of frame sizes is 12:8:1 for I:P:B frames.

There are similar, though less straightforward, levels of importance to the blocks within a frame. Areas within images differ greatly in the amount of "detail" they contain. A scene of a clear blue sky is monochromatic and almost featureless, showing little distinction between one area of the picture and another. On the other hand, a photo of New York City has a great variety of color and contrast: buildings, cars, people and trees. Noticeable errors from block transform-based compression techniques often arise on the block boundaries. If quantization is overly-coarse, it will be possible to see distinct "edges" on the block boundaries. Experiments show that the human visual system perceives error more readily in areas of lesser detail, compared to those with more activity [11]. In an image area that is "plain," artifacts such as "blockiness" from coarse quantization are more readily apparent. In light of this finding, it is preferable to allocate more bits to those areas within an image that are of relatively low detail. There are models of compression that attempt to implement rate control while controlling quantization so as to maximize the picture quality, using perceptual goals such as those mentioned above.

Viscito and Gonzales at IBM demonstrated an MPEG encoder that performs rate control and adaptive quantization [5]. Macroblocks are assigned to different classes, based on the "coding difficulty" of the image within the block. A higher level of coding difficulty signifies that this area of the picture is relatively more complex and will require more bits to encode.

The coding difficulty for intracoded blocks is estimated by the Mean Average Difference (MAD) from the overall average pixel value for each of the four luminance subblocks in the macroblock. For predictive blocks, the measure is calculated as the MAD between the subblocks of the macroblock and the corresponding region in the predicting picture- effectively the prediction error. Successful prediction will find an area very similar to the current macroblock, resulting in low values in the transmitted error block.

The encoder assigns classes to each macroblock in the frame using information gathered during motion estimation. Importantly, for both intracoded and predicted macroblocks, class assignment is based on the minimum variance value among the subblocks. The minimum is chosen so that the encoder will finely quantize macroblocks with ANY smooth regions.

A model is constructed to predict the number of bits needed to code a macroblock, given its class and the quantization factor applied. This model is based on the quantization scale, the class of a block and a variable parameter. This formula is used to predict the number of bits needed to encode the entire picture.

As the frame is encoded, rate control information is updated at the end of each row of macroblocks to keep track of how well the actual number of bits used matches the predicted output. If adherence to the predictive model is outside certain boundaries, the quantization scale is updated and the new value is applied to the next row. After an entire picture is coded, the parameters for the predicting model are updated, to match how many bits were actually produced.

This algorithm performs motion estimation for the entire frame before rate control information can be determined. This approach cannot be easily implemented in the Berkeley encoder because motion estimation is done on a macroblock-by-macroblock basis.

A second design by the same research team uses transform coefficients rather than pixel values to determine the coding difficulty of the macroblock [6]. A large coefficient indicates an area of high activity. that can be quantized more coarsely without visible errors. As in the previous algorithm, the model is still driven by the block with the least activity.

The energy is measured once the coefficients have been reduced by quantization, which acts as a normalization step, equalizing the contribution of every transform coefficient to the overall image quality. The minimum of the four maximum coefficients (one for each subblock) is used to assign the macroblock to a class, from which point the rate control process proceeds much as in the first design.

The other work referenced in the MPEG-1 standard is authored by Puri and Aravind, at AT&T Bell Laboratories [7]. Their design uses very similar techniques to those of the IBM group, but focuses on enhancing the perceptual quality of the stream through tests on individual macroblocks.

Macroblocks are classified by the amount of variance within the four subblocks. A macroblock with relatively little variance is considered to be "homogenous." Additionally, particularly bright or dark areas are noted. The authors cite research showing that luminance extremes tend to mask visual artifacts and thus require fewer bits to code. The class of a homogenous macroblock is based on these two factors. Non-homogenous blocks are tested to determine whether an "edge" exists within that area. The low-variance classes and macroblocks containing edges are considered more sensitive to visual artifacts and quantized more finely.

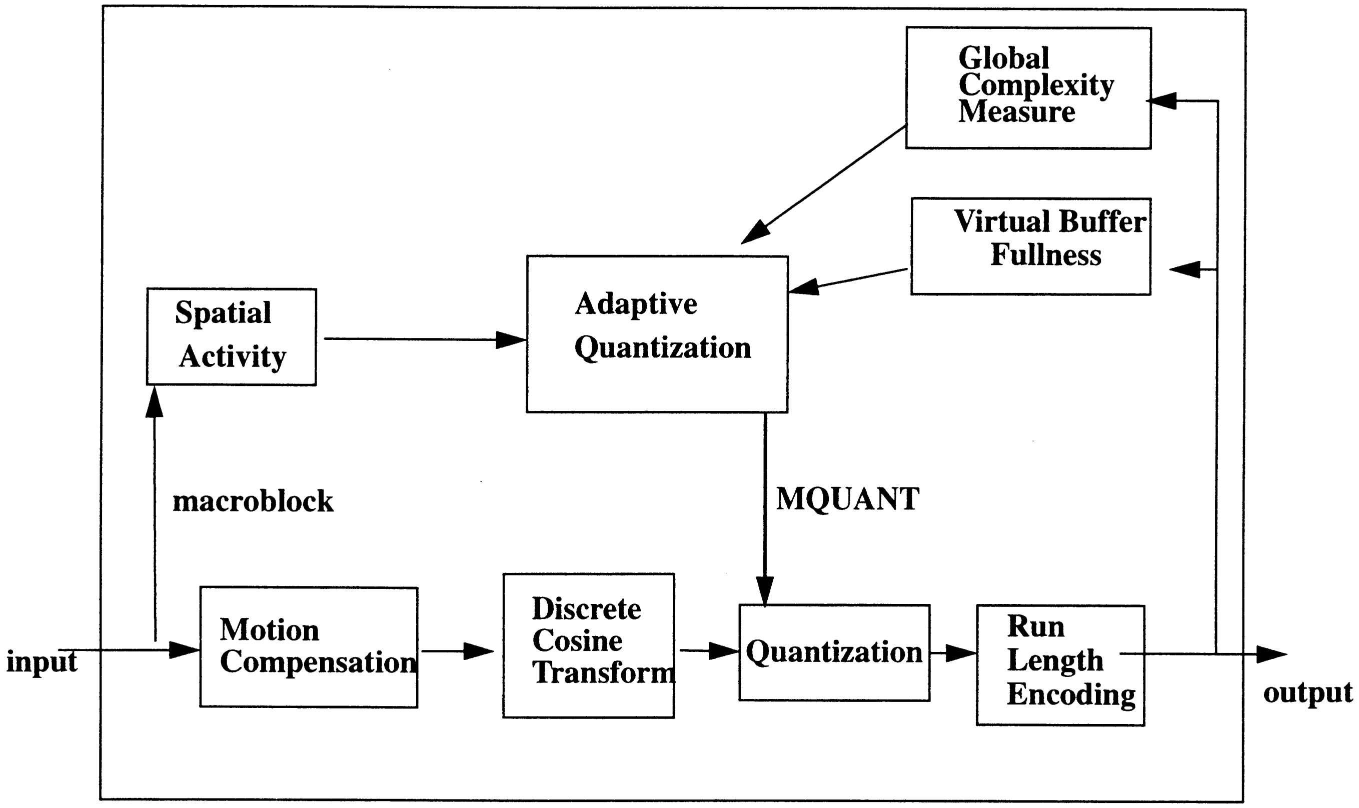
The target perceptual quality for a scene to be encoded is determined by the scene complexity and the output bitrate. Training images for each class of macroblocks are used to estimate how many bits will be required to code the different macroblock classes. The complexity of a frame can be measured by determining the class of all macroblocks in the frame. As the encoder does not have access to future image data, this measure is estimated for the current frame by using the value of the previously coded frame of the same picture type. Bit allocation is based on this global complexity figure as well as the frame type. With that information, the quantization factor can be assigned for each class to produce the desired output bitrate.

As the image is encoded, a histogram is kept of the distribution of macroblocks in each class. That information is used to update the bit-estimation models after each frame is coded.

4. Berkeley Encoder Rate Control Algorithm

The algorithm implemented in the Berkeley encoder is taken from the MPEG-2 draft [1]. A simple rate control and

Figure 5: Encoder Compression and Rate control



adaptive quantization mechanism was used by the standards committee for testing, based on the work detailed above.

Version 1.3 of the Berkeley encoder uses a fixed MQUANT for each frame type. The default values are 8 for I-frames, 10 for P-frames and 25 for B-frames. The user can change these values but all frames in the sequence use the same values.

To implement a CBR encoder, many changes had to be made to the software. Target bit allocations are determined on a frame-by-frame basis. These targets are matched to the desired bitrate by adjusting the quantization scale as each macroblock is encoded. As a prerequisite for rate control, new information must be input to the encoder, most obviously the desired bitrate.

The new input parameters and their range of values are summarized in the following table:

Table 2: Rate Control Parameters

Parameter	Possible Values
Target bitrate	400 - 104,000,000 bits/second
Picture Rate	24 - 60 frames/second
VBV Buffer Size	0 - 16,777,000 bits

The output bitrate must be entered, of course. The possible values allowed in the specification permit rates exceeding 104 Mbs. The Picture Rate parameter specifies the number of frames per second in the sequence. It is used to calculate the target bit allocation for each frame. Allowable values range between 24 and 60 frames per second, corresponding to film and TV picture rates in the U.S. and Europe.

The size of the Video Buffering Verifier parameter, called the VBV buffer, is another new parameter. This theoretical buffer is used by the standard to place limits on the acceptable variation of the bitstream from the target. The model buffer is filled at a constant rate, given by the bitrate and emptied at regular intervals, corresponding to the frame rate. The buffer could overflow, if too many bits are coming from the stream, or underflow in the opposite case. Overflow occurs if a picture or sequence is producing more bits than are being "taken" to display the video. Underflow occurs if only a partial image is available when it is time to decode the next picture for display. The current implementation monitors the VBV buffer. but as in the MPEG-2 draft on which it is based, VBV compliance is not guaranteed. To provide compliance the algorithm will have to be modified to use the VBV fullness as a factor in quantization scaling routines.

The changes needed to provide rate control and adaptive quantization were principally additions to the existing code, rather than alterations as can be seen in Figure 5. This made it easier to implement and easier for others to build upon the new codeline.

When beginning to encode a sequence, a call is made to initialize the rate control module. Each GOP is allocated an even share of the number of bits available. For example, if the sequence defines a 15-picture GOP at 30 fps, half the bitrate will be allocated to each group. Of course, as the encoder operates, fluctuations from the target bitrate affect the allocations to succeeding GOPs. Within each GOP, the bit allocation is divided among the constituent frames, depending on the type of each frame. There is a minimum allocation defined to assure acceptable picture quality. Removing this minimum raises the possibility of effectively "dropping" frames when the compression rate became unexpectedly low.

A GOP must start with an I-frame, but any sequence of frames may then follow. The bits are be divided among these frames, but not evenly as the three picture types differ in their importance to the overall image quality. Because they are always intracoded and thus do not benefit from motion compression, I-frames require the most bits. As well, these frames are used as references for the ones that follow. Although P-frames can be predicted, they also act as reference frames and receive a correspondingly significant ration of bits. B-frames never serve as reference frames and are the most compressed, due to the bidirectional interpolation and prediction. They receive the fewest bits per picture. The above rules are enforced with simple weighting factors when advance allocation is performed.

Along with the target allocations, a "global complexity measure" (GCM) is kept for each frame type. GCM is a rough measure of the difficulty to code the frames of that type. It tracks the number of bits needed to encode the picture and the average quantization parameter value during this encoding. As a "complexity" measure it represents a deviation among the energies of macroblocks in the frame. Greater variations are harder to compress. A picture with a high GCM is more difficult to encode, requiring more bits, coarser quantization or both. Initialized to an arbitrary "average" value, this measure is continually updated during the encoder's operation, steadily becoming more accurately attuned to the video sequence with time. Of course, scene changes will reduce the value of any information concerning past scenes.

The encoder proceeds through the input blocks. The rate control module is called at the beginning of the compression process for each macroblock, before motion prediction, and before transform coding. In contrast to the algorithm of Viscito and Gonzales [6], this algorithm computes measures of spatial activity based on the original blocks of the picture, rather than the transformed blocks.

The number of bits generated as the frame is coded is compared to the target allocation for the frame, which is used as a virtual buffer that is filled with the output bits. Before each macroblock is processed, it is determined if the encoding is proceeding on pace with the rate control target. The fullness of the virtual buffer is used to determine the "reference quantization" parameter for the current macroblock.

The last step in the rate control module is to calculate the adaptive quantization factor. This calculation is done before each macroblock is coded, at the same time as the rate control calculations. A spatial activity measure is calculated for the current macroblock from the luminance subblocks. The variance is calculated by summing the mean squared difference from the average pixel value for each block. Again, this measurement is performed on the original blocks, rather than the motion-predicted blocks. The minimum value among the four blocks is taken as the value for the macroblock. Selecting the block with the smallest measure as the value for the entire macroblock serves to base the adaptive quantization process on the smallest amount of activity (pixel differences) among the blocks making up a unit.

The calculated activity measure is normalized with regards to the value of the last picture to be encoded. If the activity is higher than the past result, the normalized activity increases to reflect this. A drop shows a less active frame. This variable gives a running, relative measure of the spatial activity in the sequence. Historical results are limited to the last frame encoded, as the frame type has no bearing on the spatial activity. It is evident that after a scene change, the historical result will be completely inaccurate. The formula prevents the normalized value from exceeding two or dropping below one-half.

The quantization scale for the macroblock is a combination of the spatial activity and virtual buffer fullness measures. An increase in buffer fullness indicates that more bits are being used to code the picture than expected. This situation tends to increase MQUANT and coarsen the quantization to reduce the bit output. Correspondingly, if the normalized activity value increases, the current picture has more activity in the scene than past frames. This makes the image easier to code and causes the quantization scale to increase, again coarsening the quantization. These two measures are calculated independently and can affect the scaling in opposite ways. However, the spatial activity is normalized such that it can only alter the other determinant by a factor of two in either direction.

The MQUANT computed by the rate control module is scaled to the integer range 1-31 and passed to the functions that perform the quantization steps. The entire process is visible to the remaining code by a flag alerting the main functions that a changed MQUANT value has been calculated.

5. Rate Control Experiments

This section presents the results of a series of experiments we performed to test the rate control algorithm.

Two video sequences were used for the experiments. The "flower garden" sequence shows a house bordered by bright flowers, as seen from a passing car. It is a CIF image

Table 3: MQUANT by Type

		I			P			B			ALL	
RATE	Min	MAX	Avg	MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX	AVG
1.2M	3	25	10.4	5	31	17.4	7	31	24.1	3	31	21.1
VBR	8	8	8	10	10	10	25	25	25	8	25	20.0
2.6M	1	13	4.1	3	20	8.3	6	31	14.0	1	31	11.7

Table 4: Frame Sizes by Type

		I			P			B			ALL	
RATE	Min	MAX	Avg	MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX	AVG
1.2M	151,448	171,992	159,752	61,200	81,368	71,681	7,184	26,504	13,543	7,184	171,992	42,716
VBR	191,000	205,472	199,074	115,608	142,712	130,168	7,648	20,592	12,071	7,648	205,472	58,131
2.6M	279,968	327,408	296,317	121,200	185,680	159,876	16,736	48,192	34,621	16,736	327,408	91,076

(i.e., 352 by 240 pixels). The second video sequence was the "fireworks" test sequence, which is a full-sized image, measuring 720 by 480 pixels. It is a night scene of fireworks exploding.

We choose CBR target output rates of 1.2 and 2.6 Mbs. The former rate corresponds to the video component of the CPB rate of 1.5 Mbs, while the latter corresponds to the 3Mbs rate used in many interactive TV trials. Video itself is only about 4/5ths of the total bit stream, the remainder being audio and system information.

Figure 6 shows the accuracy with which the CBR encoder adheres to the target bitrate. The graph plots the cumulative average bitrate on a frame-by-frame basis (after a one-second start-up). Clearly, the algorithm is quite effective at matching the desired bitrate. The output fluctuates within about 3% of the target rate. However, it clearly *does* fluctuate, even in a sequence without any of the scene changes that we know will cause the accuracy of the historical measures governing rate control to suffer. The algorithm does not *guarantee* a certain rate, nor does it even set a floor or ceiling. To provide such a guarantee, stricter management of the virtual buffer fullness is required. Also, it would be necessary to drop frames, should the rate rise unexpectedly. Such close control generally requires constant, minute alterations to the encoding process. A compromise solution might be to disable adaptive quantization on B-frames and vary MQUANT solely for rate control purposes. This strategy is part of both encoders designed by Viscito and Gonzalez.

Table 3 shows summary statistics on MQUANT for encodings of the "flower garden." As expected, the quantization scale is constant in the VBR encoder. The modifications to the scale are clearly shown in the data for the two CBR runs. It is shown that MQUANT is much higher for the encoder set to a lower rate, causing coarser quantization and a lower bitrate.

Table 4 shows the same summary statistics for the frame sizes. Clearly the frame sizes vary by the target output rate, as would be expected. Examination of the table shows that the CBR encoder allocates many more bits to encode B-frames, while the VBR encoder uses more bits in encoding the I- and P-frames. The comparison shows that the average P-frame is allocated almost as many bits in the VBR encoder than in high-rate CBR coder and many more than in the lower-rate CBR encoder. Also, the VBR encoding uses fewer bits for the B-frames than either CBR encoding.

Allocating more bits to the B-frames is not a optimal pattern for the CBR encoder, as the reference frames are more important and should preferably receive a greater proportion of the total bits. Of course, this heuristic can be easily tuned within the rate control functions by changing the relative weighting of the frame types.

Figure 7 shows the bits used by the encoder on a frame-by-frame basis. A VBR encoding using the default Qscale parameters and two CBR encodings are shown- one at 1 Mbs and one at 3Mbs. The CBR encoding set to a 3Mbs target actually produces an output bitrate of 3.1Mbs. In contrast, when the encoder is set to 1Mbs output, the actual rate was about 1.07 Mbs. For comparison, the output of the VBR-mode encoder was slightly less than 1.75Mbs.

Figure 8 shows the effect rate control has on quality by plotting the Signal/Noise (S/N) ratio for each frame. The S/N ratio measures the difference between the original input frame and the decoded frame reconstructed from the output of the encoder. It is calculated by reconstructing the encoded frame and performing a pixel-by-pixel comparison to the original luminance blocks. Not surprisingly, quality improves as the output rate is increased. Setting a higher bitrate allows a finer granularity of quantization, which improves the picture quality.

However, S/N ratio is not the best measure of quality for these purposes. It is extremely objective, in that it measures the difference between every pair of bits in the test images. It cannot take into account that differences in one picture may be much more noticeable than those from another encoding. In effect, it has no perceptual component. Other researchers in the Berkeley MPEG Research Group are currently implementing a perceptual distortion measure to determine if it will be a better quality metric [10]

As an illustration of this problem, Gonzales and Viscito judged that their encoder provided a significant improvement in image quality. However, they also found that mathematical measurements, similar to the S/N ratio, failed to show that improvement. They concluded that there was a mismatch between objective and subjective measures of image quality.

However, subjective results were also inconclusive in our tests. The expected results would be that the sky above the horizon, a smooth, almost featureless region of the flower garden sequence, would be seen as an area of less spatial activity and thus quantized more finely. Correspondingly, the garden itself, a mixture of many textures, should be an area in which coarse quantization is appropriate. We would expect close examination of this area to reveal more significant artifacts in this area of the CBR encoding.

It is possible that there was a somewhat less smooth quality to the sky in the CBR sequence. It also seemed that the garden was less-detailed than in the VBR encoding. However, those results were not apparent to more objective observers than the researchers themselves. In all, VBR and CBR encodings at similar rates appeared quite similar, though no formal visual comparison was performed.

Summary

An algorithm to perform simple rate control and adaptive quantization was implemented in the Berkeley MPEG encoder. It is included in the March 1995 release of the Berkeley MPEG Tools [12].

References

[1] Test Model Editing Committee, "Draft MPEG-2 Test Model 5," ISO/IEC/JTC1/SC29/WG11, April 1993.

[2] Didier LeGall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, vol. 34, no. 4, April 1991.

[3] MPEG-1 Standard (ISO/IEC International Standard 11172-2)

[4] Kevin Gong and Lawrence Rowe. "Parallel MPEG-1 Video Encoding."

[5] Eric Viscito and Cesar Gonzales. "A Video Compression Algorithm with Adaptive Bit Allocation and Quantization," *SPIE Visual Communications and Image Processing '91: Visual Communications*, Vol. 1605, 1991.

[6] C. A. Gonzales and E. Viscito. "Motion Adaptive Quantization in the Transform Domain," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 1, No. 4, December 1991

[7] A. Puri and R. Aravind. "Motion-Compensated Video Coding with Adaptive Perceptual Quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 1 No. 4, December 1991

[8] A. Puri and R. Aravind. "On Comparing Motion-Interpolation Structures for Video Coding," *SPIE Visual Communications and Image Processing*, Vol. 1360, 1990.

[9] Gregory Wallace, "The JPEG Still Picture Compression Standard," *Communications of the ACM*, vol. 34, no. 4, April 1991

[10] P. Teo and D. Heeger, "Perceptual Image Distortion," *Human Vision, Visual Porcessing, and Digital Display V*, SPIE Proceedings, Vol. 2179, February 1994.

[11] K. Ngan, Kk. Leong and H. Singh, "Adaptive Cosine Transform Coding of Images in Perceptual Domain," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 11, November 1989.

[12] Available through anonymous ftp to "mm-ftp.CS.-Berkeley.edi", in directory "/pub/multimedia/mpeg."