Efficient Cost Measures for Motion Compensation at Low Bit Rates

(extended abstract)

Dzung T. Hoang[*] Philip M. Long[†] Jeffrey Scott Vitter[‡]

Department of Computer Science
Duke University
Box 90129
Durham, NC 27708–0129

Abstract

We make a case that, even with severe efficiency constraints, taking the number of bits to code each motion vector into account when estimating motion for video compression results in significantly better performance at low bit rates, using simulation studies on established benchmark image sequences. In particular, we examine an algorithm that differs from a "vanilla" implementation of the H.261 standard by choosing motion vectors to minimize a cost function of prediction error and the number of bits to code a particular motion vector, where the coefficients of the cost function are adapted on-line using the Widrow-Hoff rule. We show that this algorithm performs comparably to a variety of more idealized, computationally intensive methods we examined in earlier papers and substantially better than the original "vanilla" method, which ignores the number of bits to code the motion vector when choosing it.

1 Introduction

Hybrid video coding that combines block-matching motion compensation (BMMC) [1] with transform coding of the residual is a popular scheme for video compression adopted by international standards such as H.261 [2, 3] and MPEG [4]. Motion compensation is a technique that exploits the typically strong correlation between successive frames of a video sequence by coding *motion vectors* that tell the decoder where to look on the previous frame for predictions of the intensity of each pixel in the current frame. With BMMC, the current frame is divided into blocks (usually 8×8 or 16×16) whose pixels are assigned the same motion vector. The residual from motion compensation is then coded with a block-transform coder, such as the 2D DCT.

[*]Affiliated with Brown University. Supported in part by an NSF Graduate Fellowship and by Air Force Office of Strategic Research grant F49620–92–J–0515.

[†]Support was provided in part by Air Force Office of Strategic Research grant F49620–92–J–0515.

[‡]Support was provided in part by Air Force Office of Strategic Research grant F49620–92–J–0515, Army Research Office grant DAAH04–93–G–0076, and by an associate membership in CESDIS.

We are concerned primarily with video coding at low bit rates for applications such as video-phone and video-conferencing, where the bit rate is typically limited to 64 kb/s or less. At such low bit rates, the coding of motion vectors and side information takes up a significant portion of the bandwidth. This observation had been previously made in [5], but much of the work on block-matching motion compensation remained focused on speeding up the motion search until [6], where we described techniques that explicitly minimize rate (which includes coding of residuals, motion vector, and side information) when choosing motion vectors. In [7] we extended this work by performing motion estimation to minimize a combination of rate and distortion. In related work, Chung, Kossentini and Smith [8] considered rate-distortion optimizations for motion estimation in a video coder based on subband coding and vector quantization.

The explicit minimization algorithms described in [6, 7] are computationally intensive and therefore not practical for use in real-time, cost-effective solutions for desktop video-conferencing and video-phone. However, they serve to demonstrate that by taking the coding of motion vectors into consideration when performing motion estimation, we can get appreciably better results. In [7], we reported on preliminary experiments regarding the use of efficiently computed cost measures for motion estimation that take into account the coding of motion vectors as well as the prediction error energy. A way to think of that work was that it evaluated "two-pass" algorithms, where a first pass was used to estimate the parameters of the cost function used. Such algorithms are clearly not useful for videoconferencing and videophone applications. In this paper, we examine the use of algorithms which adaptively set the parameters of the cost function "on-the-fly". Our results show that this practice improves performance in addition to its obvious advantages of practicality and flexibility.

We implemented and tested our motion estimation algorithms within the H.261 standard (also known informally as the $p \times 64$ standard). The $p \times 64$ standard is intended for applications like video-phone and video-conferencing, where low bit rates are required, not much motion is present, and frames are to be transmitted with low encoding delay. We provide experimental results for coding benchmark videos typically encountered in video-phone and video-conferencing applications.

In the next section, we provide a brief overview of the $p \times 64$ standard. In Section 3, we review the explicit minimization coders described in [6, 7]. We then describe heuristic cost functions based on the prediction error and motion vector code-length in Section 4. These heuristic functions are implemented in two $p \times 64$ coders and are found to give compression performance comparable to the explicit minimization coders while running much faster.

2 Overview of the P $\times$ 64 Standard

In 1990, the International Telegraph and Telephone Consultative Committee (CCITT) approved an international standard for video coding at bit rates of $p \times 64$ kb/s, where p is a small integer. Officially known as Recommendation H.261, it is also informally called the $p \times 64$ standard. In this section, we provide a brief summary of key aspects of the standard.

The $p \times 64$ standard uses the Common Intermediate Format (CIF) and Quarter-CIF (QCIF) for video frames. These formats employ a three-component color system consisting of a luminance band, Y, and two chrominance bands, C_{B} and C_{R}. The chrominance bands are subsampled by a factor of 4 compared to the luminance band. A video frame is divided into Groups of Blocks (GOB) made up of macroblocks (MB). Each macroblock in turn is composed of four 8×8 Y blocks, one 8×8 C_{B} block, and one 8×8 C_{R} block. Integer-pel motion compensation performed at the macroblock level; that is, there is one motion vector per macroblock.

At a high level, the basic encoding process is as follows. For each macroblock M, the encoder chooses a motion vector $\vec{v}$ (how this is done is left unspecified in the standard), and the difference between $\vec{v}$ and the motion vector for the previous macroblock is transmitted, using a static Huffman code. For each 8×8 block B contained in M, a lossy version of the block of prediction errors obtained by using $\vec{v}$ to predict B is then transmitted. This is done by applying a 2D DCT to the block of prediction errors, quantizing the transform coefficients, and sending the result using a run-length/Huffman coder, where the coefficients are scanned in a zig-zag order. The encoder has the option of changing certain aspects of the above process. First, the encoder may simply not transmit the current macroblock; the decoder is then assumed to use the corresponding macroblock in the previous frame in its place. If transmitted, the macroblock can be transform-coded with motion compensation (interframe coding) or without (intraframe coding). If motion compensation is used, there is an option to apply a linear filter to the previous decoded frame before using it for prediction.

The $p \times 64$ standard does not specify how to make coding decisions. However, to aid in the evaluation of different coding techniques, the CCITT provides an encoder simulation model called Reference Model 8 (RM8) [9]. A fast three-step search is used for motion estimation. RM8 specifies several heuristics used to make the coding decisions. The variance of the prediction error for the luminance blocks in M by using $\vec{v}$ is compared against the variance of the luminance blocks in M to determine whether to perform intraframe or interframe coding. If interframe mode is selected, the decision of whether to use motion compensation with a zero motion vector or with the estimated motion vector is made by comparing the mean absolute difference (MAD) of motion compensation with zero motion against that with the estimated motion vector. The loop filter is enabled if a non-zero motion vector is used. The decision of whether to transmit the block-transform coefficients is made individually for each block in a macroblock by considering the values of the quantized transform coefficients. If all the coefficients are zero for a block, they are not transmitted for that block.

As a basis for comparison of the different motion estimation schemes presented in this paper, we use the “vanilla” $p \times 64$ coder supplied by the Portable Video Research Group (PVRG). The PVRG coder implements many, but not all, of the RM8 heuristics. Most notably, full-search motion estimation is used instead of the three-step process. Details about the PVRG implementation can be found in [10].

3 Explicit Minimization Algorithms

In this section, we briefly describe the explicit minimization algorithms of [6, 7]. The first two algorithms, M1 and M2, minimize code-length and the third, RD, minimizes a combination of code-length and distortion.

The RM8 simulation model and the PVRG coder both perform motion estimation to minimize prediction error. Algorithm M1 applies bit-minimization to estimate motion. For each macroblock, a motion vector is chosen to minimize the number of bits used to encode the macroblock.[1] This includes coding of quantized transform coefficients for the luminance blocks, the motion vector, and other side information. Coding decisions are made in the same way as with the PVRG coder.

Algorithm M2 takes bit-minimization further by including coding control in the minimization. In Algorithm M1, the decisions of whether to perform motion compensation and whether to use a loop filter with motion compensation are made using several heuristics. In Algorithm M2, these decisions are also made to minimize code-length: All three combinations of the decisions are tried, and the one resulting in the minimum code-length is used.

Choosing motion vectors and making coding decision based solely on the code-length may not be the best policy if the resulting distortion is ignored. There may be cases where the choice of motion vector and coding decisions that minimize code-length results in a relatively high distortion, whereas another choice would have a slightly higher code-length but substantially lower distortion. From a rate-distortion perspective, the latter choice may be better. Algorithm RD takes both rate and distortion into account when performing motion estimation. Specifically, algorithm RD minimizes a cost function that is a linear combination of rate and distortion:

$$C_{\mathrm{RD}}(\vec{v}, \vec{c}) = B(\vec{v}, \vec{c}) + \lambda D(\vec{v}, \vec{c}). \tag{1}$$

Here $B(\vec{v}, \vec{c})$ is the number of bits to code the current macroblock using motion vector $\vec{v}$ and coding decisions $\vec{c}$, and $D(\vec{v}, \vec{c})$ is the resulting mean squared reconstruction error. The choice of the parameter λ depends on the operational rate-distortion curve for the particular input video. The parameter λ could be determined either by preprocessing a portion of the input video to estimate the rate-distortion curve or through an online iterative search method [11].

4 Heuristic Algorithms

While Algorithms M1, M2, and RD generally exhibit better rate-distortion performance than the base PVRG coder, they are computationally expensive. The additional computation is in the explicit evaluation of the rate (and distortion in the

[1]Since this minimization is done on a macroblock basis, it is inherently a greedy process that does not result in a global optimization.

case of RD). To reduce the computational complexity, we propose to minimize an efficiently computed model of rate and distortion. The idea is that the prediction error (MSE, MAD, or similar measure) can be used to estimate the rate and distortion for transform coding. This estimate is then combined with the motion vector code-length, which is readily available with a table lookup. We develop such a cost function below and use it in two heuristic coders, H1 and H2, that are analogous to the explicit minimization coders, M1 and M2. Both H1 and H2 choose motion vectors to minimize the cost function. However, H1 makes coding decisions using the same heuristic function that the PVRG and M1 coders use, and H2 chooses the coding control that minimizes the code-length given the estimated motion vectors. Since H2 has to try out three coding control choices, it will be about three times slower than H1. However, H2 gives us an indication of the performance that is achievable under H1 by improving the coding control. Algorithms H1 and H2 were presented in [7].

4.1 Heuristic Cost Function

Let $\vec{E}(\vec{v})$ denote a measure of the prediction error that results from using motion vector $\vec{v}$ to code the current macroblock. For example, the error measure could be defined as $\vec{E}(\vec{v}) = \langle \mathrm{MAD}(\vec{v}), \mathrm{DC}(\vec{v}) \rangle$, where $\mathrm{MAD}(\vec{v})$ is the mean absolute prediction error and $\mathrm{DC}(\vec{v})$ is the average prediction error. Suppose we have a model $H(\vec{E}(\vec{v}), Q)$ that gives us an estimate of the number of bits needed to code the motion compensation residual, where $\vec{E}(\vec{v})$ is defined above and Q is the quantization step size. We could then combine this estimate with $B(\vec{v})$, the number of bits to code the motion vector $\vec{v}$. The result is a cost function that we can use for motion estimation:

$$C_{\mathrm{H}}(\vec{v}, Q) = H(\vec{E}(\vec{v}), Q) + B(\vec{v}). \tag{2}$$

As defined above, the function H provides an estimate of the number of bits needed to code the motion compensation residual with quantizer step size Q. As we will discuss later, it can also be used to estimate a combination of the rate and distortion.

The choice of error measure, $\vec{E}$, and heuristic function, H, are parameters to the motion estimation algorithm. In our investigations, we used MAD as the error measure, for computational reasons. We also looked at using the MSE, but this did not give any clear advantages over the MAD. It is also possible to define $\vec{E}$ to be a function of several variables. For the rest of this paper, we report only on the use of MAD for $\vec{E}$ and denote $\vec{E}(\vec{v})$ by ξ for convenience, where the dependence on $\vec{v}$ is implicit. We examine several choices for H and describe them below.

As mentioned above, we can use H to estimate the number of bits used to transform-code the prediction error. To get an idea of what function to use, we gathered experimental data on the relationship between the MAD and DCT-coded bits per macroblock for a range of motion vectors. Fixing the quantization step size Q at various values, the data was generated by running the RD coder on two frames of the QCIF Miss America sequence and outputting the MAD and DCT coded bits per macroblock for each choice of motion vector. Visual inspection of the histogramed data suggests the following forms for H:

$$H(\xi) \quad = \quad c_1\xi + c_2, \tag{3}$$

$$H(\xi) = c_1 \log(\xi + 1) + c_2, \quad (4)$$
$$H(\xi) = c_1 \log(\xi + 1) + c_2\xi + c_3. \quad (5)$$

The above forms assume a fixed Q. In general, H also depends on Q; however, when using H to estimate the motion motion for a particular macroblock, Q is held constant to either a preset value or to a value determined by the rate control mechanism. We can treat the parameters c_i as functions of Q. Since there is a small number (31) of possible values for Q, we can perform curve fitting for each value of Q and store the parameters in a lookup table. We can also determine the parameters adaptively.

We can also model the reconstruction distortion as a function of prediction error. We use the RD coder to generate experimental data for distortion versus MAD and find a similar relationship as existed for code-length versus MAD. Again, we can consider using (3)–(5) to model the distortion. As with the RD coder, we can consider jointly optimizing the heuristic estimates of rate and distortion with the following cost function:

$$C_{\mathrm{H}}(\vec{v}, Q) = H_{\mathrm{R}}(\xi, Q) + \lambda H_{\mathrm{D}}(\xi, Q) + B(\vec{v}), \quad (6)$$

where H_{R} is the model for rate, H_{D} is the model for distortion.

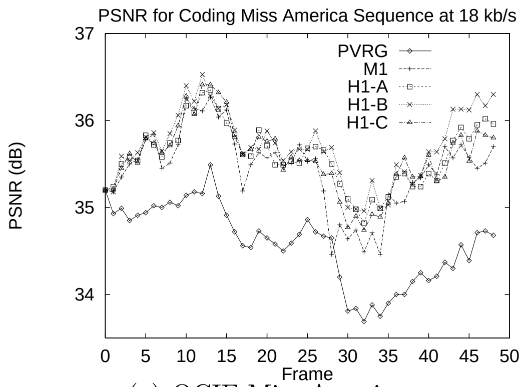
If we use one of (3)–(5) for both H_{R} and H_{D}, the combined heuristic function, $H = H_{\mathrm{R}} + \lambda H_{\mathrm{D}}$, would have the same form as H_{R} and H_{D}. Therefore we can interpret the heuristic as modeling a rate-distortion function. In this case, we can perform curve fitting once for the combined heuristic function by training on the statistic $R + \lambda D$, where R is the DCT bits for a macroblock and D is the reconstruction distortion for the macroblock. As with RD, the parameter λ can be determined from the rate-distortion curve, for example.

5 Experimental Results

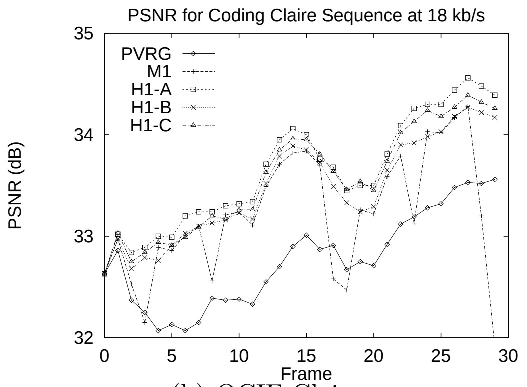
For our experiments, we coded 49 frames of the "Miss America" sequence and 30 frames of the "Claire" sequence, both in QCIF (176×144) format sampled at 10 frames per second. These are "head and shoulders" sequences typical of the type found in video-phone and video-conferencing applications. We present results here for coding at 18 kb/s using the buffer-feedback rate controller specified in RM8.

5.1 Static Cost Function

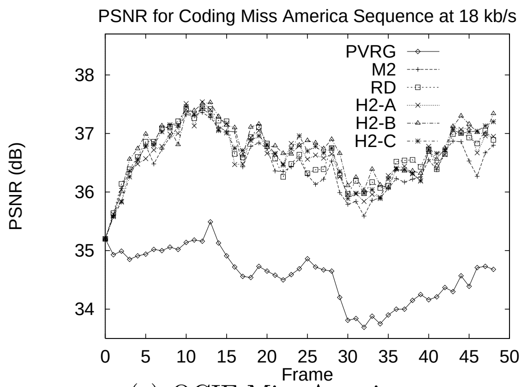
Here, we present results using a static set of coefficients. To determine the coefficients for the heuristic functions, we performed linear least squares regression, fitting data generated by the RD coder to the $R + \lambda D$ statistic, as discussed earlier. A set of regression coefficients are stored in a lookup table, indexed by the quantizer step size Q. We tested the different forms for the heuristic function given in (3)–(5). Comparative plots of the resulting PSNR are shown in Figure 1. The average PSNR for coding at 18 kb/s is tabulated in Table 1. These results show that the heuristic coders perform comparably to the explicit minimization coders. In particular, the heuristic coders seem more robust than M1 and M2, most likely because the heuristic



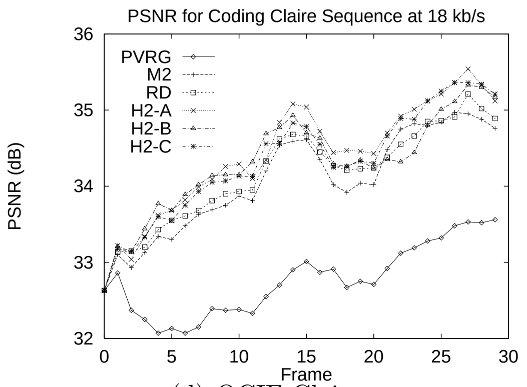
(a) QCIF Miss America



(b) QCIF Claire



(c) QCIF Miss America



(d) QCIF Claire

Figure 1: Distortion for coding Miss America and Claire sequences at 18 kb/s with rate control. In (a) and (b), we compare the H1 coder with the PVRG and M1 coders. In (c) and (d), we compare the H2 coder with PVRG, M2 and RD coders. H1-A (H2-A), H1-B (H2-B), and H1-C (H2-C) use the heuristic functions (3), (4), and (5), respectively.

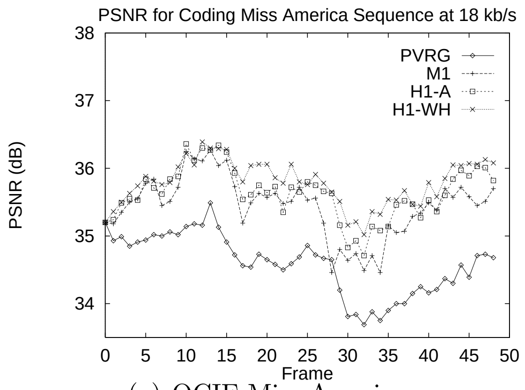
functions correlate well with both code length and distortion, whereas M1 and M2 only consider code length.

5.2 Adaptive Cost Function

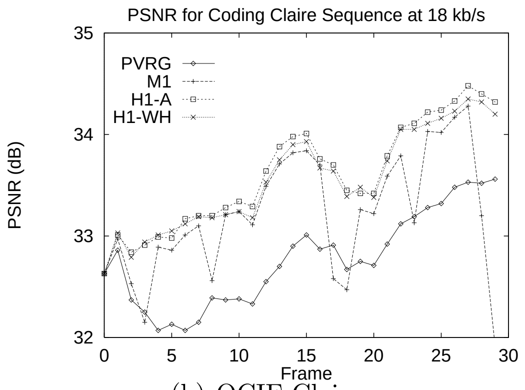
The above results rely on pre-training the model parameters c_i for each value of Q for each video sequence. This is a tedious and time-consuming operation. Instead, we can use an adaptive on-line technique, such as the Widrow-Hoff learning rule [12, 13], to train the model parameters. The training examples could be generated each time we encode a macroblock using motion compensation mode. However, we cannot possibly hope to train one model for each value of Q simply because there would not be enough training examples. We need a single model whose parameters are independent of Q. The curve fitting results from pre-training show a strong correlation between the model parameters and Q^{-1}. This agrees well with previous work on rate-quantization

Sequence	PVRG	M1	M2	RD	H1-A	H1-B	H1-C	H2-A	H2-B	H2-C
Ms Amer.	34.58	35.44	36.51	36.67	35.60	35.72	35.58	36.63	36.77	36.68
Claire	32.77	33.24	34.12	34.22	33.68	33.50	33.60	34.47	34.36	34.39

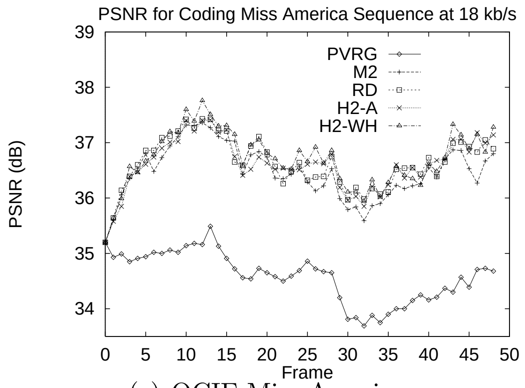
Table 1: Average PSNR (in dB) of inter-coded frames for coding test sequences at 18 kb/s.



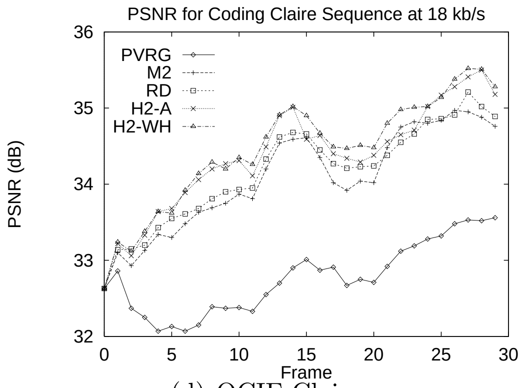
(a) QCIF Miss America



(b) QCIF Claire



(c) QCIF Miss America



(d) QCIF Claire

Figure 2: Distortion for coding Miss America and Claire sequences at 18 kb/s with rate control. In (a) and (b), we compare the adaptive H1 coder with the static H1, PVRG and M1 coders. In (c) and (d), we compare the adaptive H2 coder with the static H2, PVRG, M2 and RD coders. H1-A and H2-A use the heuristic function (3).

Video	PVRG	M1	M2	RD	H1-A	H1-WH	H2-A	H2-WH
Miss America	34.58	35.44	36.51	36.67	35.60	35.80	36.63	36.75
Claire	32.77	33.24	34.12	34.22	33.68	33.58	34.47	34.51

Table 2: Average PSNR (in dB) of inter-coded frames for coding test sequences at 18 kb/s.

modeling [14]. Therefore we propose the following form for the cost function:

$$H(\xi, Q) = c_1 \frac{\xi}{Q} + c_2. \tag{7}$$

Since the previous sections showed that the simple linear model performs relatively well, we do not consider more complex models here.

We ran experiments using the Widrow-Hoff training rule on the Miss America and Claire sequences. The training rate is determined in a trial-and-error phase and fixed for both sequences. Training is performed on $R + \lambda D$ statistic, as described earlier. The parameter λ is also determined by trial-and-error and held constant for both test sequences. Comparative plots of the resulting PSNR are shown in Figures 2. The average PSNR for coding at 18 kb/s is tabulated in Table 2. These results show that the adaptive heuristic coders perform comparably to and sometimes better than the static heuristic coders and the explicit minimization coders. Furthermore, the adaptive heuristic coders perform well on both sequences with the same initial parameter values.

6 Conclusions

In this paper, we have demonstrated that, at low bit rates, choosing motion vectors to minimize an efficiently computed heuristic cost function gives substantially better rate-distortion performance than the conventional approach of minimizing prediction error. Furthermore, by adapting the heuristic function to the input sequence, we are able to achieve coding performance comparable to more computationally expensive coders that explicitly minimize rate or a combination of rate and distortion.

We have considered only the simple case of using a fixed parameter λ to trade rate and distortion. An online adaptation of λ to track variations in the input sequence is certainly possible and would result in more robust coders. On the other hand, we observed that the behavior of these algorithms is quite robust with respect to moderate variations in λ, and that, for example, the best setting of λ for one test sequence worked well when used for the other. Thus, it seems as fixing λ is safe in practice. Still, since λ influences rate to some extent, it can be used in conjunction with the quantization step size in performing rate control. Preliminary investigation along these lines show promising results.

We are interested in examining whether other quickly computed statistics are better indicators of the value of using a particular motion vector, or whether they could be fruitfully combined with the MAD. For example, the maximum prediction error for a block can be computed almost for free if the MAD is already being computed. Other reasonable candidates include the DC transform coefficient and possibly a handful of other transform coefficients.

We expect that the methods of this paper can be gainfully applied to video compression methods other than the H.261 standard. The upcoming H.263 standard is similar enough to H.261 that it seems clear that these methods will work well with H.263.

References

[1] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe coding", *IEEE Transactions on Communications*, vol. COM-29, no. 12, pp. 1799–1808, 1981.

[2] CCITT, "Video codec for audiovisual services at $p \times 64$ kbit/s", Aug. 1990, Study Group XV—Report R 37.

[3] M. Liou, "Overview of the $p \times 64$ kbit/s video coding standard", *Communications of the ACM*, vol. 34, no. 4, pp. 60–63, Apr. 1991.

[4] D. Le Gall, "MPEG: A video compression standard for multimedia applications", *Communications of the ACM*, vol. 34, no. 4, pp. 46–58, Apr. 1991.

[5] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: A review", *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 589–609, Sept. 1994.

[6] D. T. Hoang, P. M. Long, and J. S. Vitter, "Explicit bit-minimization for motion-compensated video coding", in *Proceedings of the 1994 Data Compression Conference*, Snowbird, UT, Mar. 1994, pp. 175–184, IEEE Computer Society Press.

[7] D. T. Hoang, P. M. Long, and J. S. Vitter, "Rate-distortion optimizations for motion estimation in low-bit-rate video coding", in *Proceedings SPIE 1996 Digital Video Compression Conference*, San Jose, CA, Jan. 1996.

[8] W. C. Chung, F. Kossentini, and M. J. T. Smith, "A new approach to scalable video coding", in *Proceedings of the 1995 Data Compression Conference*, Snowbird, UT, Mar. 1995, pp. 381–390, IEEE Computer Society Press.

[9] CCITT, "Description of reference model 8 (RM8)", June 1989, Study Group XV—Document 525.

[10] A. C. Hung, "PVRG-p64 codec 1.1", 1993, Available from Stanford University by anonymous ftp.

[11] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, Sept. 1988.

[12] B. Widrow and M. E. Hoff, "Adaptive switching circuits", in *1960 IRE WESCON Convention Record*, 1960, vol. 4, pp. 96–104.

[13] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA, 1991.

[14] W. Ding and B. Liu, "Rate-quantization modeling for rate control of MPEG video coding and recording", in *Proceedings SPIE Electronic Imaging – Digital Video Compression*, San Jose, CA, Feb. 1995, vol. 2419.