Rate-distortion optimizations for motion estimation in low-bit-rate video coding

(extended abstract)

Dzung T. Hoang

Brown University, Dept. of Computer Science
Box 1910, Providence, RI 02912-1910

Philip M. Long

Duke University, Dept. of Computer Science
Box 90129, Durham, NC 27708-0129

Jeffrey S. Vitter

Duke University, Dept. of Computer Science
Box 90129, Durham, NC 27708-0129

ABSTRACT

We make a case that 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 videos. First, by modifying a "vanilla" implementation of the H.261 standard, we show that choosing motion vectors explicitly to minimize rate (in a greedy manner), subject to implicit constraints on distortion, yields better rate-distortion tradeoffs than minimizing notions of prediction error. Locally minimizing a linear combination of rate and distortion results in further improvements. Using a heuristic function of the prediction error and the motion vector code-length results in compression performance comparable to the more computationally intensive coders while requiring a practically small amount of computation. We also show that making coding control decisions to minimize rate yields further improvements.

keywords: motion estimation, rate-distortion, video compression, H.261

1 INTRODUCTION

Hybrid video coding that combines block-matching motion compensation[1] (BMMC) 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 lossy transform coder, such as the 2D DCT.

In previous work on BMMC, motion vectors are typically chosen to minimize prediction error, and this task is often separated from the other components of the video coding system that control rate and distortion. Much of the emphasis of research in motion estimation for video coding has been on speeding up the motion search. In this paper, 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 rates, the coding of motion vectors takes up a significant portion of the bandwidth. We investigate the use of cost measures that more directly estimate the effect of the choice of motion vector on the total code-length and reconstruction distortion. We first develop and present computationally intensive coders that attempt to optimize explicitly for rate and distortion. Insights from these implementations lead to faster coders that minimize an efficiently computed heuristic function. Our experimental results show that using these measures yields substantially better rate-distortion performance.

We implemented and tested our motion estimation algorithms using an existing implementation of 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 very low bit rates are required, not much motion is present, and frames are to be transmitted essentially as they are generated. Our experimental results are for benchmark videos typical of the type for which the $p \times 64$ standard was intended: they consist of a "head-and-shoulders" view of a single speaker.

In the next section, we briefly describe an existing implementation of the $p \times 64$ standard that we use as a basis for comparison. We then show how to modify the base implementation, but remain within the $p \times 64$ standard, to choose motion vectors that more directly minimize code-length and distortion. Experiments show that when transmitting two benchmark QCIF video sequences, *Miss America* and *Claire*, at 18 kb/s using rate control, choosing motion vectors explicitly to minimize code-length improves average PSNR by 0.87 dB and 0.47 dB respectively. In the $p \times 64$ standard, two binary coding decisions must be made from time to time. In the base implementation, heuristics based on prediction error are used to make these decisions. When bit minimization is also applied to make the coding decisions, the improvement in PSNR becomes a significant 1.93 dB (respectively 1.35 dB). If instead of minimizing the bit rate, we minimize a combination of rate and distortion, we observe improvements of 2.09 dB and 1.45 dB. We then describe coders that minimize a heuristic function of the prediction error and motion vector code-length. These heuristic coders give compression performance comparable to the explicit minimization coders while running much faster. Experimental results are presented in Sections 3.4 and 4.3.

Our bit-minimization philosophy for motion estimation, manifest in Algorithms M1 and M2 described below, was presented earlier.[5] This paper extends the previous work with Algorithm RD, which performs rate-distortion optimization, and Algorithms H1 and H2, which minimize an efficiently computed heuristic cost function. In related work, Chung, Kossentini and Smith[6] consider rate-distortion optimizations for motion estimation in a video coder based on subband coding and vector quantization.

2 PVRG IMPLEMENTATION OF P $\times$ 64

As a basis for comparing the different motion estimation schemes described in this paper, we use the "vanilla" $p \times 64$ coder supplied by the Portable Video Research Group[7] (PVRG). In the base PVRG implementation, a motion vector $\vec{v}$ is determined for each macroblock M using standard full-search block-matching. Only the luminance blocks are compared to the determine the best match, with the mean absolute difference (MAD) being used as the measure of prediction error. Several heuristics are used to make the coding decisions. The variance V_{P} of the prediction error for the luminance blocks in M by using $\vec{v}$ is compared against the variance V_{Y} of the luminance blocks in M to determine whether to perform intraframe or interframe coding. If interframe motion compensation mode is selected, the decision of whether to use motion compensation with a zero motion vector or the estimated motion vector is made by comparing the MAD of motion compensation with zero motion against that with the estimated motion vector. In the former case, no motion vector needs to be sent. The loop filter in interframe mode is enabled if V_{P} is below a certain threshold. The decision of whether to transmit a transform-coded block is made individually for each block in a macroblock by considering the sum of absolute values of the quantized transform coefficients. If the sum falls below a preset threshold, the block is not transmitted. These heuristics implement many of the recommendations put forth in the H.261 Reference Model 8.[8]

3 EXPLICIT MINIMIZATION ALGORITHMS

In the PVRG coder, full-search motion estimation is performed to minimize the MAD of the prediction error. A rationale for this is that minimizing the MSE (approximated with the MAD) of the prediction error is equivalent to minimizing the variance of the 2D DCT coefficients of the prediction error, which tends to result in more coefficients being quantized to zero. However, minimizing the variance of the DCT coefficients does not necessarily lead to a minimum coding of the quantized coefficients, especially since the quantized coefficients are then Huffman and run-length encoded. Furthermore, since coding decisions are made independently, the effect of motion estimation on code-length is further made indirect. In this section, we describe two algorithms that perform motion estimation explicitly to minimize code-length and a third algorithm that minimizes a combination of code-length and distortion. We then present results of experiments that compare these algorithms with the standard motion estimation algorithm used by the PVRG coder.

3.1 Algorithm M1

In Algorithm M1, motion estimation is performed explicitly to minimize (locally) the code-length of each macroblock. To compute the code-length, we make the same coding decisions as the PVRG coder and invoke the appropriate encoding subroutines for each choice of motion vector within the search area, picking the motion vector that results in the minimum code-length for the entire macroblock. The computed code-length includes the coding of the transform coefficients for the luminance blocks,[a] the motion vector, and all other side information. When choosing the motion vector to minimize the coding of the current macroblock, we use the fact that the motion vectors for previous macroblocks (in scan order) have been determined in order to compute the code-length. However, since the choice of a motion vector for the current macroblock affects the code-length of future macroblocks, this is a greedy minimization procedure which may not result in a globally minimal code-length.

3.2 Algorithm M2

In Algorithm M1, the decisions of whether to use a filter and whether to use motion compensation are made the same way as in the PVRG $p \times 64$ implementation. In Algorithm M2, however, 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. Since M2 is able to make decisions on how to code each macroblock, it is able to take into account the coding of side information in minimizing the code-length. For very low bit rates, where the percentage of side information is significant compared to the coding of motion vectors and transform coefficients, one would expect M2 to be effective in reducing the code-length of side information.

3.3 Algorithm RD

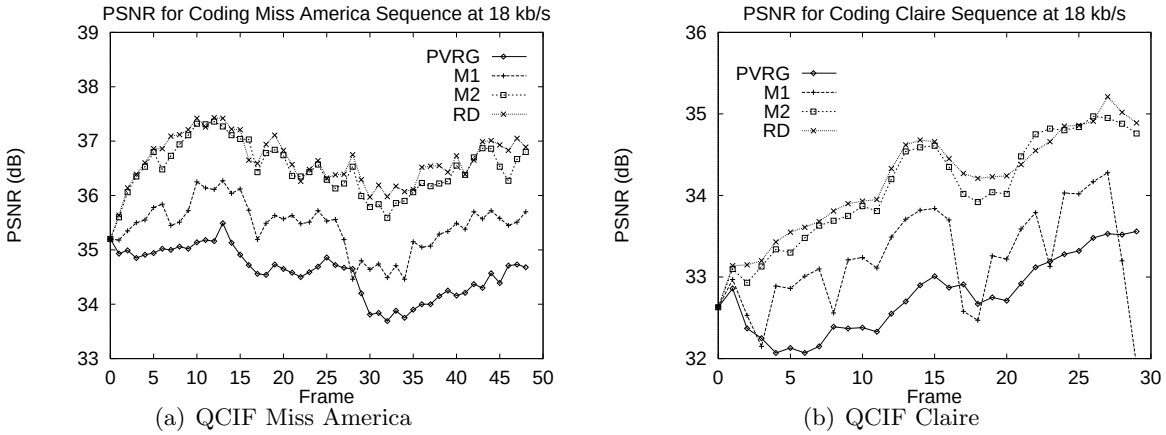
With Algorithms M1 and M2, we minimize code-length without regard to distortion and then choose the quantization step size to achieve the desired distortion level. This is not always the best policy. There may be cases where the choice of motion vector and coding decisions that minimize code-length results in a relatively high distortion, whereas another choice would have a slightly higher code-length but substantially lower distortion. In terms of rate-distortion tradeoff, the second choice may be better. Since the ultimate goal is better rate-distortion performance, we expect further improvements if we minimize a combination of code-length and distortion. M1 and M2 call encoder routines in the minimization steps. By adding calls to decoder routines, we can compute the resulting distortion. We incorporated this idea into Algorithm RD.

Algorithm RD minimizes a linear combination of code-length and distortion.[b] Let $B(\vec{v}, \vec{c})$ denote the number of bits to code the current macroblock using motion vector $\vec{v}$ and coding decisions $\vec{c}$. Similarly, let $D(\vec{v}, \vec{c})$ be the resulting mean squared error. RD minimizes the objective function

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

[a] The transform coding of the chrominance blocks could be included as well. However, we chose not to do so in order to make a fair comparison to the base PVRG coder. This is also the policy for the other coders described in this paper.

[b] This is similar to the standard rate-distortion optimization procedure based on Lagrange multipliers.



(a) QCIF Miss America (b) QCIF Claire

Figure 1: Comparison of explicit-minimization motion estimation algorithms for coding the Miss America and Claire sequences at 18 kb/s.

The choice of the parameter λ depends on the operational rate-distortion curve for the particular input video. A good choice is to set λ to be equal to the negative of the slope of the line tangent to the operational distortion vs. rate curve at the operating point. This can be determined, for example, by preprocessing a portion of the input video to estimate the rate-distortion curve. An online iterative search method could also be used.[9] In our experiments, we code the test sequence several times with different quantizer step sizes to estimate the rate-distortion function, and determine λ from the slope of the function.

3.4 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 rate controller outlined in RM8.[8] The average PSNR for each coded frame is plotted for the Miss America and Claire sequences in Figure 1. The average PSNR for inter-coded frames are tabulated in Table 1. For each sequence, all the coders used the same quantization step size for the initial intra-coded frame.

4 HEURISTIC ALGORITHMS

While Algorithms M1, M2, and RD generally exhibit better rate-distortion performance than the base PVRG coder, they are computationally intensive. The additional computation is in the explicit evaluation of the rate (and distortion in the case of RD). To address the computational complexity, we introduce Algorithms H1 and H2, which minimize a heuristic function of the prediction error and the motion vector code-length, both of which can be computed efficiently. The idea is that the prediction error (MSE, MAD, or similar measure) can be used to estimate the rate and distortion for transform

coding. The motion vector code-length (readily available with a table lookup) is included since it is significant when coding at low bit rates. For H1, coding control is performed using the same decision rules used in the PVRG and M1 coders. With H2, coding control is performed to minimize rate in the same manner as is done in M2.

4.1 Algorithm H1

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. Let $B(\vec{v})$ denote the number of bits to code the motion vector $\vec{v}$. Algorithm H1 chooses $\vec{v}$ to minimize an objective function $C_{\mathrm{H}}(\vec{v}, Q)$ defined as

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

where Q is the quantization step size.[c] Intuitively, the function H can be thought of as providing an estimate of the number of bits needed to code the prediction error with quantizer step size Q. As we will discuss later, it can also be used to estimate a combination of the rate and distortion for coding the prediction error.

The choice of error measure, $\vec{E}$, and heuristic function, H, are parameters to the 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. The results were histogramed and examined and suggested the following forms for H:

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

$$H(\xi) = c_1 \log(\xi + 1) + c_2, \tag{4}$$

$$H(\xi) = c_1 \log(\xi + 1) + c_2\xi + c_3. \tag{5}$$

The parameters c_i could be determined offline by trial-and-error or by standard curve fitting techniques, or they could be determined online by applying adaptive techniques such as the Widrow-Hoff learning rule[10] and recursive curve fitting. The above forms assume a fixed Q. In general, H also

[c] Generally, H depends on the quantization step size Q as well as $\vec{v}$. For simplicity, we sometimes assume that we are coding with a fixed Q. With rate control, Q will necessarily vary. In this case, we appeal to the more general formulation of H.

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 could do a surface fit for $H(\xi, Q)$. However, determining the appropriate functional form for such a surface fit would be a more involved task. Instead, we treat the fit parameters c_i as functions of Q. Since there is a small number (31) of possible values for Q, we can store the parameters in a lookup table, for instance.

We can also consider modeling the reconstruction distortion as a function of prediction error. We used the RD coder to generate experimental data for distortion versus MAD and found a similar relationship as existed for code-length versus MAD. Again, we can consider Equations 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:

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

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

If we use either (3) or (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.

4.2 Algorithm H2

Like M1, the H1 coder uses a fixed coding control. As with M2, we can consider trying out all the coding control choices and choosing the one that results in the fewest coded bits. We apply this modification to H1 and call the resulting coder H2. (We note that the heuristic function is used only to perform motion estimation and not for coding control.) 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.

4.3 Experimental results

For the H1 and H2 coders, we used the same test sequences and followed the procedures described in Section 3.4. We tested the different forms for the heuristic function given in (3)–(5). To determine the coefficients for the heuristic functions, we used linear least squares regression, fitting to the $R + \lambda D$ statistic, as discussed earlier. A set of regression coefficients, for each value of the quantizer step size Q, were stored in a lookup table.

As in Section 3.4, we ran the heuristic coders with rate control targeted to 18 kb/s. Comparative plots of the resulting PSNR are shown in Figures 2 and 3. 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 functions correlate well with both code-length and distortion, whereas M1 and M2 only consider code-length.

5 CONCLUSION

We have demonstrated that, when coding video at low rates, choosing motion vectors and coding control to minimize code-length, a combination of code-length and distortion, and computationally efficient approximations thereof, yields substantial improvements in rate-distortion performance compared to just minimizing prediction error.

In this paper, 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. 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.

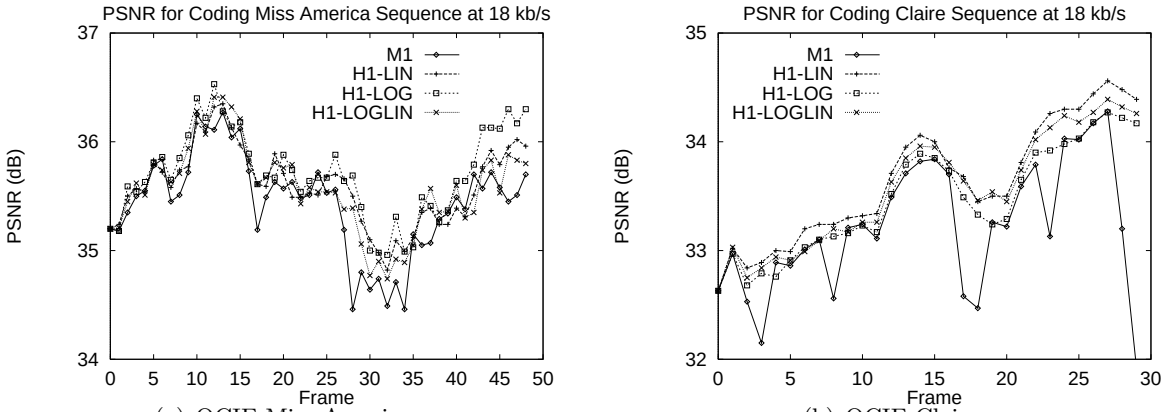
We have thus far only investigated a limited number of functional forms for the heuristic function H. These were suggested by visual examination of the histograms; however, perhaps some sort of theoretical analysis would suggest alternative forms. Another possibility would be to use techniques from nonparametric statistics,[11] where one estimates a functional relationship without choosing a form for the hypothesis a priori, instead implicitly using smoothness assumptions on the relationship to be modeled. We are also 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.

Preliminary indications are that the parameters can be fixed independent of the quantization step size for some of the heuristic functions considered in this paper without affecting performance much, resulting in a cheaper coder. These results point to the robustness of the heuristic coders and are likely due to the observation that the heuristic functions are somewhat correlated to both the rate and distortion.

We expect that the methods of this paper could 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

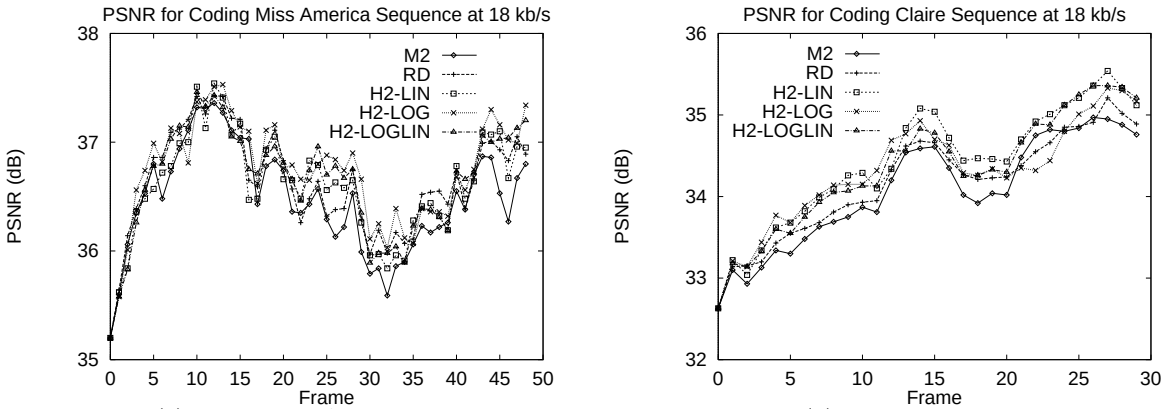
Sequence	PVRG	M1	M2	RD	H1-lin	H1-log	H1-loglin	H2-lin	H2-log	H2-loglin
Miss America	34.58	35.44	36.51	36.67	35.60	35.72	35.58	36.63	36.77	36.68
Claire	32.77	33.24	34.12	34.22	33.68	33.50	33.60	34.47	34.36	34.39

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

Figure 2: Distortion for coding Miss America and Claire sequences at 18 kb/s. Comparison of H1 heuristics with M1 coder.



(a) QCIF Miss America (b) QCIF Claire

Figure 3: Distortion for coding Miss America and Claire sequences at 18 kb/s. Comparison of H2 heuristics with M2 and RD coders.

clear that these methods will work well with H.263. An example of the use of the techniques of this paper in a coder where the motion field is encoded using a quadtree has been described earlier.[5]

6 ACKNOWLEDGEMENTS

Dzung T. Hoang was supported in part by an NSF Graduate Fellowship and by Air Force Office of Strategic Research grant F49620–92–J–0515. Philip M. Long was supported in part by Air Force Office of Strategic Research grant F49620–92–J–0515. Jeffrey Scott Vitter was supported 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.

7 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. 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.

6. 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.

7. A. C. Hung, "PVRG-p64 codec 1.1", 1993, Available by anonymous ftp from `havefun.stanford.edu`.

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

9. 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.

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

11. W. Haerdle, *Smoothing Techniques*, Springer Verlag, 1991.