CS–1995–16

Rate-Distortion Optimizations for Motion Estimation in Low-Bitrate Video Coding

Dzung T. Hoang[1] Philip M. Long[2]
Jeffrey Scott Vitter[3]

Department of Computer Science

Duke University

Durham, North Carolina 27708–0129

June 19, 1995

[1]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.

[2]Support was provided in part by Air Force Office of Strategic Research grant F49620–92–J–0515. Current address: Research Triangle Institute, 3040 Cornwallis Road, P.O. Box 12194, Research Triangle Park, NC 27709.

[3]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.

Abstract

We present and compare methods for choosing motion vectors for motion-compensated video coding. Our primary focus is on videophone and videoconferencing applications, where very low bit rates are necessary, where motion is usually limited, and where frames must be coded in the order they are generated. We provide evidence, using established benchmark videos typical of these applications, that choosing motion vectors explicitly to minimize rate, subject to implicit constraints on distortion, yields better rate-distortion tradeoffs than minimizing notions of prediction error. Minimizing a linear combination of rate and distortion results in further rate-distortion improvements. Using a heuristic function of the prediction error and the motion vector codelength results in compression performance comparable to the more computationally intensive coders while running much faster. We incorporate these ideas into coders that operate within the $p \times 64$ standard.

1 Introduction

The typically strong correlation between successive frames of a video sequence makes video highly compressible, since the pixels of the previous frame can be used to predict the intensities of the current frame. The difference between the predicted and the true frame often is small and can be encoded efficiently, for example, by a lossy transform coder using the two-dimensional discrete cosine transform (2D DCT). Improved compression is readily obtained by first estimating what portions of the current frame correspond to moving objects and then transmitting *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. The most popular method for estimating these motion vectors originated with Jain and Jain [6] and is called *block-matching*. In their approach, the current frame is divided into blocks (usually 8×8 or 16×16) whose pixels are assigned the same motion vector. The motion vector for a given block B is usually obtained by (approximately) minimizing, from among candidates $\vec{v}$ within a limited search area, some norm of the difference between B and the prediction obtained from $\vec{v}$. The mean squared error (MSE) is a commonly used difference measure, although for example the mean absolute difference (MAD) is often substituted because it can be implemented efficiently. Block-matching motion compensation is combined with DCT transform coding of residuals in hybrid video coding systems such as H.261 (also known informally as the $p \times 64$ standard) [1, 8] and MPEG [2].

In previous work on motion compensation for video coding, motion vectors are chosen to minimize prediction error, and this task is often separated from other components of the video coding system that control the rate and distortion. Most of the emphasis of research in motion estimation has been on speeding up the motion search without sacrificing too much rate/distortion performance. In this paper, we are concerned primarily with video coding at low bitrates for applications such as videophone and videoconferencing, where the bitrate is typically limited to 64 kbps or less. At such low bitrates, 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 codelength and reconstruction distortion. Our experimental results show that using these measures yields substantially better rate-distortion tradeoffs.

Initially, our emphasis is on codelength and quality, not on computation time, in order to determine the limits on the compressibility of video. We first develop and present computationally intensive coders that attempt to explicitly optimize for rate and distortion. Insights from these implementations lead to faster coders that minimize an efficiently computed heuristic function.

We implemented and tested our motion estimation algorithms within an experimental implementation of the $p \times 64$ standard. The $p \times 64$ standard is intended for applications like videophone and videoconferencing, where very low bitrates are required, not much motion is present, and frames

are to be transmitted essentially as they are generated. Unlike the case of the MPEG standard, we cannot first compress a subsampling of frames, and then use frames both before and after a given frame to predict it. 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.

Using block-matching, we create only a crude, but concise, model of the motion. For video coding, we do not necessarily want to find the "correct" motion vectors, in contrast to a goal of research in optic flow, for example [9]. If a motion vector field that does not correspond to the actual motion in the scene yields the shortest description, that is sufficient for purposes of compression. However, an accurate motion field is desirable for motion interpolation, where a non-coded frame is interpolated from two successively coded frames by performing motion compensation using an interpolated motion field.

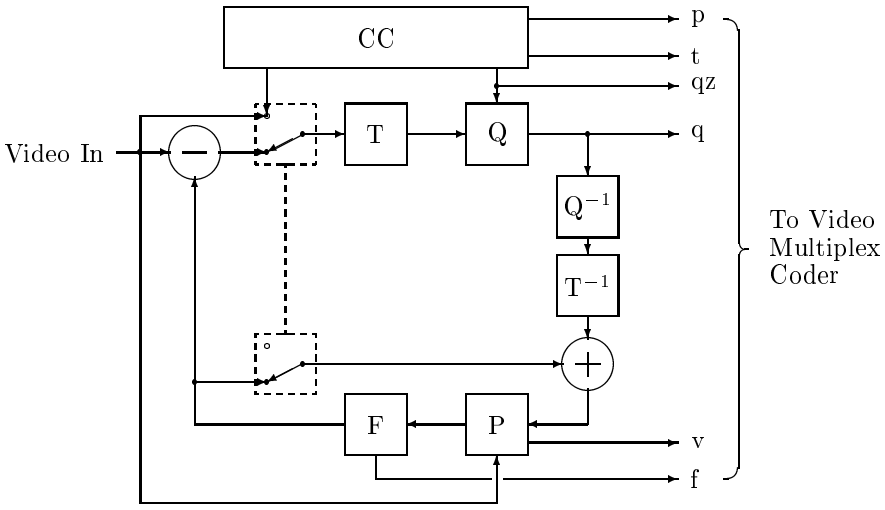
In the next section, we describe the PVRG implementation of the $p \times 64$ standard, and then show how to modify the PVRG implementation, but remain within the $p \times 64$ standard, to choose motion vectors that more directly minimize codelength and distortion. For comparable quality, at the level roughly required for transmitting a benchmark QCIF video sequence at 2,000 bits per frame, explicit bit minimization reduces the bitrate by about 11% on average. In the $p \times 64$ standard, two binary decisions must be made from time to time (for details, see Section 2). In the base PVRG implementation, heuristics based on prediction error are used to make these decisions. When the explicit bit minimization philosophy is also applied to make the coding decisions, the improvement becomes a significant 27%. If instead of minimizing the bitrate, we minimize a combination of rate and distortion, we observe a bitrate reduction of 30% for the same level of quality. We then describe a heuristic function that gives compression performance comparable to explicitly minimizing bitrate, while running much faster. Experimental results are presented in Sections 4.4 and 5.3.

To the best of our knowledge, ours is the first work investigating the effect of minimizing codelength subject to quality constraints (implicit and explicit) to choose motion vectors, although the importance of coding motion vectors at very low bitrates is acknowledged in [7].

2 Overview of the P × 64 Standard

We first provide a brief overview of key components of the $p \times 64$ standard. The $p \times 64$ standard specifies a three-component color system as the format for the video data. The three components are a luminance band Y and two chrominance bands C_{B} and C_{R}. Since the human visual system is more sensitive to the luminance component and less sensitive to the chrominance components, C_{B} and C_{R} are subsampled by a factor of 4 compared to Y. The image is composed of *Groups of Blocks* (GOB) which are further decomposed into *macroblocks* (MB) each consisting of four 8×8 Y blocks, one 8×8 C_{B} block and one 8×8 C_{R} block. Motion prediction and compensation are performed by treating each macroblock as an atomic entity; that is, there is one motion vector per macroblock.

Figure 1 shows a block diagram of the $p \times 64$ coder. At a high level, the basic process is as follows. The macroblocks are scanned in a lexicographic order. 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.



T: Transform
Q: Quantizer
P: Picture Memory with motion-compensated variable delay
F: Loop Filter
CC: Coding Control

p: Flag for INTRA/INTER
t: Flag for transmitted or not
qz: Quantizer indication
q: Quantizing index for transform coefficients
v: Motion vector
f: Switching on/off of the loop filter

Figure 1: Block diagram of the $p \times 64$ source coder [1]

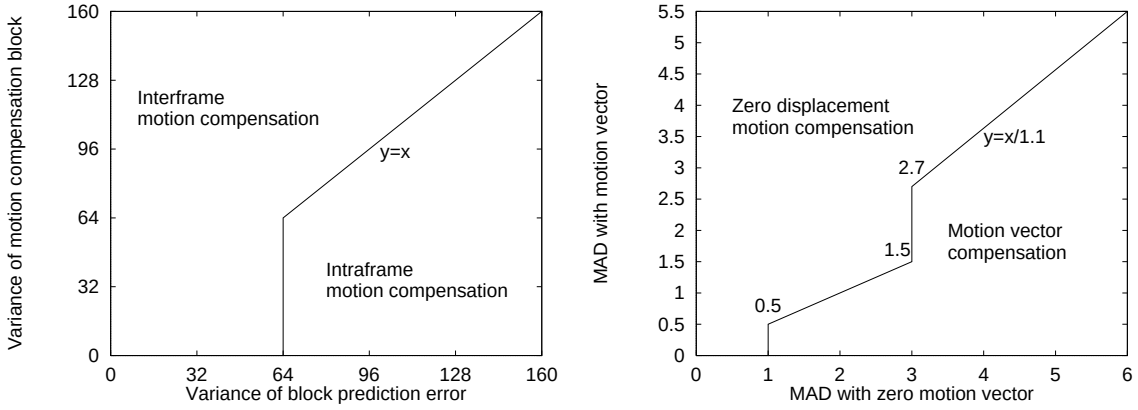
As indicated in Figure 1, the encoder has the option of changing certain aspects of the above process. First, the encoder might simply not transmit the current macroblock; the decoder is then assumed to use the corresponding macroblock in the previous frame in its place. If transmitted, the macroblock 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 these decisions.

3 PVRG Implementation of P × 64

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)[1].

In the base PVRG implementation, a motion vector $\vec{v}$ is determined for each macroblock M by means of standard full-search block-matching. Only the luminance blocks are compared to the determine the best match, with the 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 errors 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

[1] As of the publication date, the source code for this implementation can be obtained via anonymous ftp from `havefun.stanford.edu`.



(a) Intraframe/interframe decision diagram

(b) Motion vector decision diagram

Figure 2: Default decision diagrams for coding control [5]

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 default decision diagrams are shown in Figure 2. 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.

Video coders for videophone and videoconferencing often have to operate with fixed bandwidth limitations. Since the $p \times 64$ standard specifies entropy coding, resulting in a variable bitrate, some form of *rate control* is required for operation on bandwidth-limited channels. For example, if the coder's output exceeds the channel capacity, then frames could be dropped or the quality decreased in order to meet the bandwidth limitations. On the other hand, if the coder's output is well below the channel's capacity, the quality and/or frame-rate can be increased. A simple technique for rate control uses a virtual buffer model in a feedback loop in which the buffer's "fullness" controls the level of quantization performed by the lossy transform coder. The PVRG coder uses this approach. The quantization step size Q is determined from the buffer fullness B_{F} using the equation:

$$Q = \min(\lfloor 80 B_{\mathrm{F}} + 1 \rfloor, 31),$$

where Q has a integral range of $[1, 31]$, B_{F} has a real-valued range of $[0, 1]$. This feedback function is plotted in Figure 3.

4 Explicit Minimization Algorithms

In this section, we describe and compare the performance of three coders which conform to the $p \times 64$ standard. The first coder is the same as the PVRG coder, except that motion vectors are chosen in order to minimize (locally) the number of bits used to code the motion vectors as well as the residuals. In the second coder, certain binary decisions made in the first algorithm using heuristics based on error are instead made again to minimize total codelength. In the third,

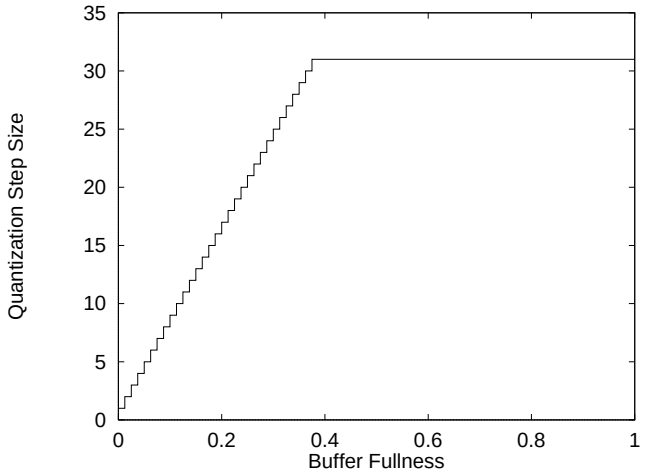


Figure 3: Feedback function controlling quantization step size based on buffer fullness

motion vectors and coding decisions are chosen to minimize a linear combination of codelength and distortion.

4.1 Algorithm M1

In the PVRG coder, motion estimation is performed (independent of the coding decisions) to minimize the MAD of the prediction errors. The rationale for this is that minimizing the MSE (approximated with the MAD) of the prediction errors is equivalent to minimizing the variance of the 2D DCT coefficients of the prediction errors, 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 effects of motion estimation on codelength is further made indirect.

In Algorithm M1, motion estimation is performed explicitly to minimize (locally) the codelength of each macroblock. To compute the codelength, 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 codelength for the entire macroblock. The computed codelength includes the coding of the transform coefficients for the luminance blocks[2], 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 codelength. However, since the choice of a motion vector for the current macroblock affects the codelength of future macroblocks, this is a greedy minimization procedure, and we may not obtain a globally minimal codelength.

Since we are explicitly attempting to minimize the codelength, we are almost assured to have higher prediction error than if we just minimize the prediction error. The higher prediction error is likely to result in higher reconstruction distortion. Instead of attempting to deal with quality directly, we rely on the transform coder and quantizer to deliver a desired level of quality; that is, the M1 coder may require a finer quantization step size to deliver the same quality as the PVRG

[2]The transform coding of the chrominance blocks could be included as well. However, we chose not to 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.

coder. As we will see in the results section, for low bitrates bit-minimization does indeed result in consistently better rate-distortion curves.

4.2 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 codelength: All three combinations of the decisions are tried, and the one resulting in the smallest codelength is used. Here, even more than with M1, we rely on the transform coder and the quantizer to code the prediction errors with adequate quality. Our hope is that the gain in compression efficiency will offset the decrease in reconstruction quality for a given quantization step size; that is, to achieve a certain quality level, we may be able to use finer quantization and still get improvements in compression.

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 codelength. For low bitrates, where the percentage of side information is significant compared to the coding of motion vectors and transform coefficients, one would expect that M2 will be able to reduce the codelength of side information.

As with M1, M2 does not explicitly consider the effect of the motion vector on the resulting distortion. However, experimental results show better rate-distortion performance compared to the base PVRG coder.

4.3 Algorithm RD

With Algorithms M1 and M2, we minimize the codelength without regards 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 codelength result in a relatively high distortion, whereas another choice would have a slightly higher codelength 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 codelength 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 codelength and distortion[3]. 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

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

The choice of the Lagrange parameter λ depends on the 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 rate-distortion 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.

4.4 Experimental Results

We performed experiments using 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 videophone and videoconferencing applications.

[3]This is a standard rate-distortion optimization procedure based on Lagrange multipliers.

4.4.1 Rate-Distortion Characteristics

To evaluate the rate-distortion performance of the various motion estimation algorithms, we ran them with various fixed quantization step sizes Q, without rate-control. For RD, we determine the values for λ from the slope of the rate-distortion curve for the M2 coder. The search region used for block-matching is ± 7 in both directions. The averaged rate-distortion curves achieved by varying Q are plotted for low bitrates in Figures 4 and 5.

As indicated in the plots, M1 performs moderately better than the PVRG implementation and M2 and RD significantly better. For instance, transmitting the QCIF Miss America sequence at 2,000 bits per frame would result in an average PSNR of 35dB. For the same distortion, the M1, M2, and RD coders would require approximately 11%, 27%, and 30% less bandwidth, respectively.

Tables 1–8 show the bitrate consumption of the coders in more detail. The tabulated figures are averages for interframe coding and do not include the initial intra-coded frame. At low bitrates, the PVRG coder spends more bits for motion vectors than transform coefficients. M1, M2, and RD achieve significant reductions in codelength with the coding of the motion vectors. Also, since M2 and RD are able to make coding control decisions to reduce the codelength, further gains are achieved in the coding of side information. At higher bitrates, both M2 and RD achieve significant rate reduction with the transform coefficients. Furthermore, M2 and RD are able to trade off additional motion-vector coding for less transform-coefficient coding.

The improvement of RD over M2 is minimal in these experiments. However, experiments on CIF (352×288) format sequences at higher bitrates show noticably better results (see Figure 6). The diminishing returns at very low bitrates is due to the relatively high overhead for coding side information at very low bitrates in the $p \times 64$ standard. For example, with the Miss America sequence, RD is able to reduce the DCT coding by about 10% over M2, but due to the relatively high cost of coding motion vectors and side information, the total improvement is less.

4.4.2 Rate Control

We performed additional experiments using rate control to achieve a target bitrate of 16 kbps. The virtual buffer feedback rate control mechanism described in Section 3 is used, with the buffer size set to 8 kbits. The average PSNR for each coded frame is plotted for the Miss America and Claire sequences in Figures 7 and 8, respectively. All the coders start with the same quantization step size for the initial intra-coded frame.

5 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 codelength, 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 tranform coding. The motion vector codelength (readily available with a table lookup) is included since it is significant when coding at low bitrates. 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.

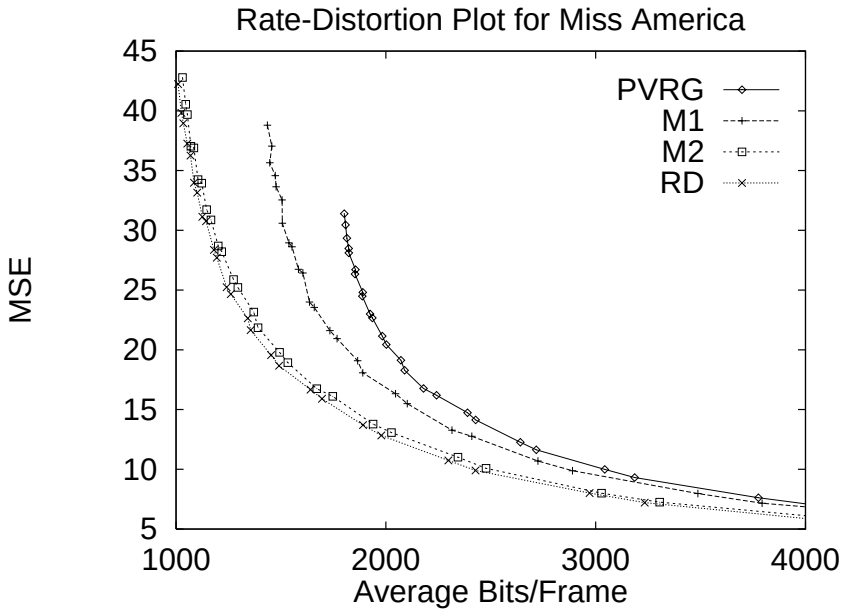


Figure 4: Averaged rate-distortion curve for QCIF Miss America sequence

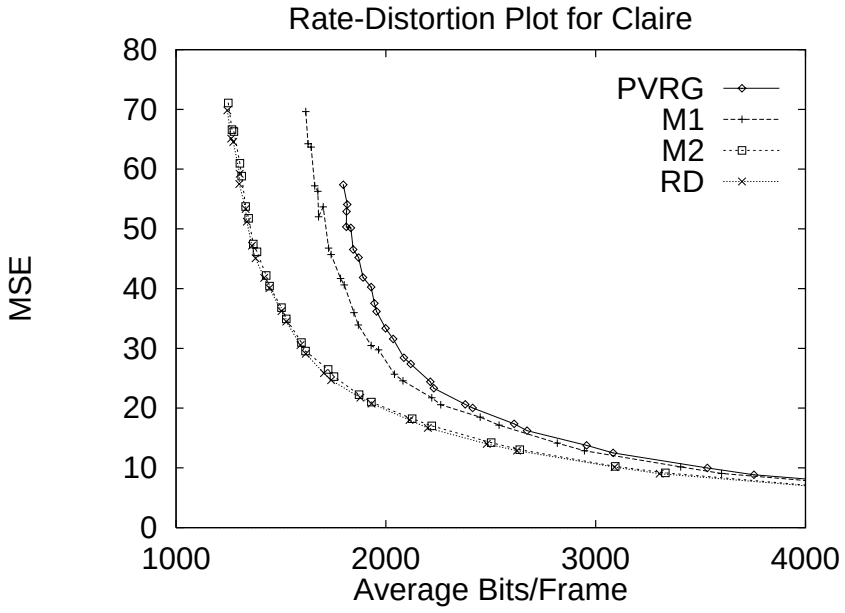


Figure 5: Averaged rate-distortion curve for QCIF Claire sequence

Table 1: Coding Miss America sequence with PVRG coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	1687.83	87.2917	576	1024.54	33.1775
28	1705.83	117.521	559.854	1028.46	33.5985
24	1767.77	176.5	555.396	1035.88	34.1942
20	1853.75	269.854	540.438	1043.46	34.8846
16	2040.25	459.771	515.062	1065.42	35.8919
12	2480.77	884.333	494.271	1102.17	37.2402
8	3585.4	1977.73	435.812	1171.85	39.3037

Table 2: Coding Miss America sequence using M1 coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	1313.6	86.6042	229.75	997.25	32.245
28	1348.65	113.083	228.729	1006.83	32.7477
24	1408.17	170.458	222.062	1015.65	33.516
20	1499.58	254.375	221.333	1023.88	34.3269
16	1719.94	457.104	211.333	1051.5	35.3213
12	2147.85	859.542	199.542	1088.77	36.8958
8	3290.83	1949.85	179.104	1161.88	39.1048

Table 3: Coding Miss America sequence using M2 coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	900.896	73.8542	248.125	578.917	31.8192
28	937.667	98.0417	247.521	592.104	32.4433
24	1008.21	141.812	253.083	613.312	33.111
20	1130	226.271	256.083	647.646	33.9965
16	1340.21	391.729	251.125	697.354	35.1646
12	1764.17	734.458	250.375	779.333	36.7285
8	2822.38	1670.94	247.25	904.188	39.0923

Table 4: Coding Miss America sequence using RD coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	879.333	63.3542	249.854	566.125	31.8756
28	920.354	85.8542	253.146	581.354	32.4188
24	986.875	129.292	254.729	602.854	33.1948
20	1096.21	208.396	253.562	634.25	34.105
16	1298.44	366.438	249.375	682.625	35.2135
12	1715.1	708.396	246.479	760.229	36.7519
8	2763.9	1636.88	242.938	884.083	39.0858

Table 5: Coding Claire sequence with PVRG coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	1547.28	83.8621	383.862	1079.55	30.5452
28	1544.72	101.241	362.931	1080.55	31.1097
24	1597.21	150.241	366.931	1080.03	31.9059
20	1672.31	245.655	344.103	1082.55	32.8897
16	1842.34	412.069	340.724	1089.55	34.2483
12	2168.48	761.379	308.103	1099	35.7203
8	2941.93	1541.9	280.448	1119.59	38.1193

Table 6: Coding Claire sequence with M1 coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	1361.93	99.7586	191.862	1070.31	29.6931
28	1389.38	131.069	186.759	1071.55	30.5431
24	1427.41	169.759	183.862	1073.79	31.4224
20	1515.97	259.103	176.759	1080.1	32.5579
16	1665.97	410.655	169.552	1085.76	34.0238
12	2000.93	745.276	160.207	1095.45	35.4817
8	2810.45	1555.86	139.483	1115.1	38.0403

Table 7: Coding Claire sequence with M2 coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	980.31	78.4483	204.724	697.138	29.5866
28	1020.38	104.586	204.724	711.069	30.2531
24	1056.07	124.793	195.586	735.69	31.3438
20	1158.52	206.862	193.931	757.724	32.4472
16	1338.69	364.138	189.345	785.207	33.8834
12	1665.83	669.966	179.448	816.414	35.5045
8	2488.55	1444.66	165.586	878.31	38.0028

Table 8: Coding Claire sequence with RD coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	975.483	74.0345	204.483	696.966	29.6652
28	1019.21	105.759	201.586	711.862	30.381
24	1050.1	124.31	193.448	732.345	31.3707
20	1158.72	210.586	192.621	755.517	32.519
16	1317.69	350.241	186.586	780.862	33.9886
12	1652.45	660.897	179	812.552	35.5586
8	2484.83	1440.76	163.138	880.931	38.0597

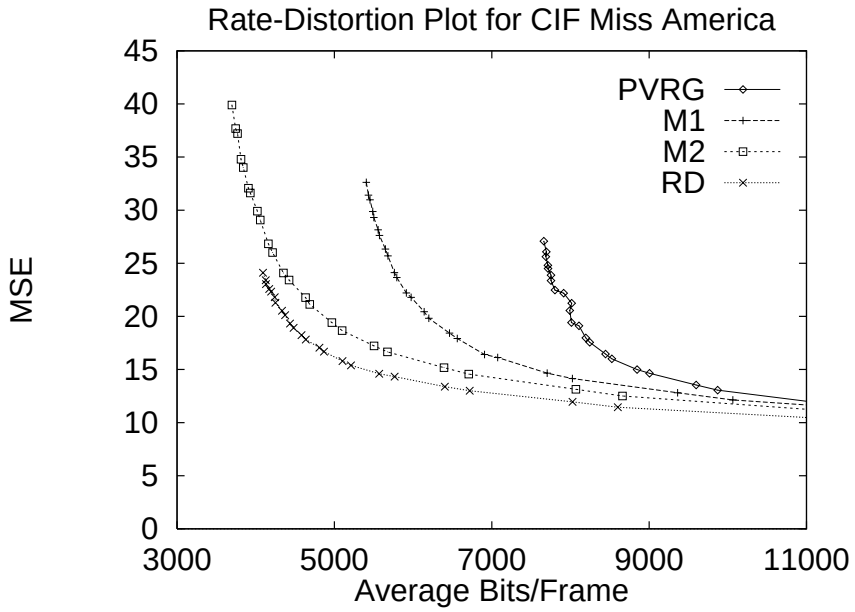


Figure 6: Averaged rate-distortion curve for CIF Miss America sequence

5.1 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 the 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[4].

Intuitively, the function H can be thought of as providing an estimate of the number of bits to code the prediction error. 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 examined 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

[4]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.

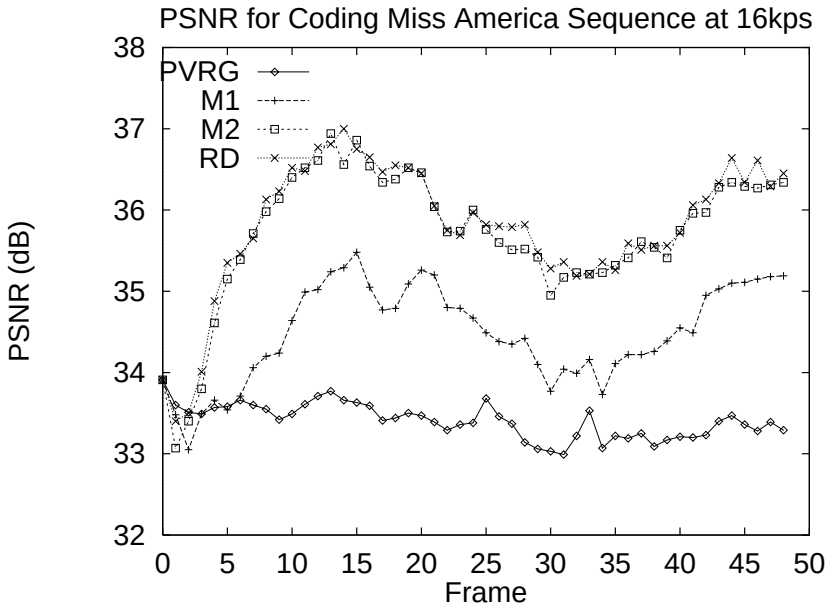


Figure 7: Distortion for coding Miss America sequence at 16kbps with rate control.

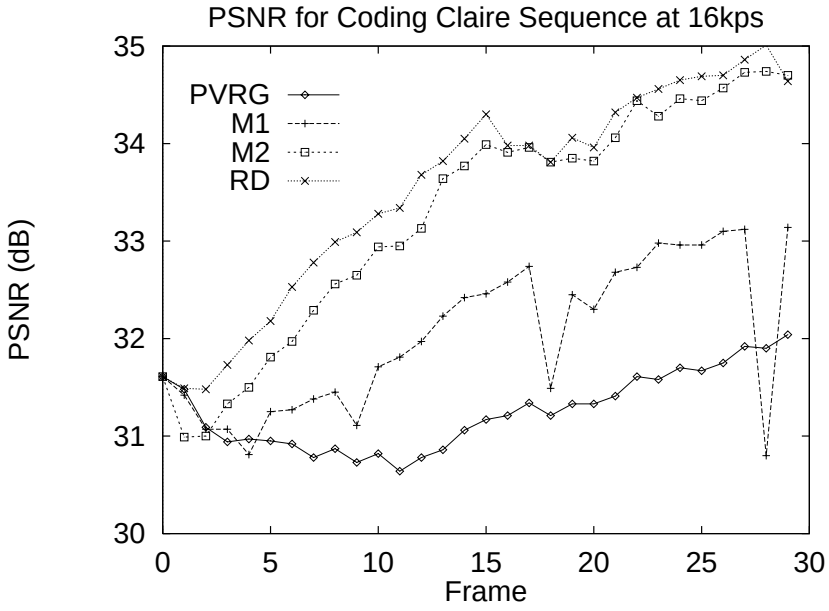


Figure 8: Distortion for coding Claire sequence at 16kbps with rate control.

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 are histogrammed and shown as density plots in Figure 9.

These plots suggest 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 online using adaptive techniques such as Widrow-Hoff and recursive curve fitting. 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 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 are a small number (31) of possible values for Q, we can store the parameters in a lookup table, for instance.

We can also consider modelling the reconstruction distortion as a function of prediction error. Again, we use the RD coder to generate experimental data for distortion versus MAD. The resulting density plots are shown in Figure 10. The plots are somewhat similar to the ones relating DCT bits to 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, and λ is the Lagrange parameter.

If we use the same functional form to model both rate and distortion, the combined heuristic function, $H = H_{\mathrm{R}} + \lambda H_{\mathrm{D}}$, would have the same form in the case of Equations 3–5. In this case, we can perform curve fitting once for the combined heuristic 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 in RD, the parameter λ can be determined from the rate-distortion curve, for example.

5.2 Algorithm H2

As with M1, the H1 coder uses a fixed coding control. As with M2, we can consider trying out all the coding control choices and chosing the one that results in the fewest coded bits. We apply this modification to H1 and call the resulting coder H2. Since H2 has to try out three coding control choices, it will be about three times slower than H1. However, H2 can easily take advantage of parallel hardware.

5.3 Experimental Results

For the H1 and H2 coders, we used the same test sequences and followed the same procedures described in Section 4.4. We tested the different forms for the heuristic function given in Equations 3–5.

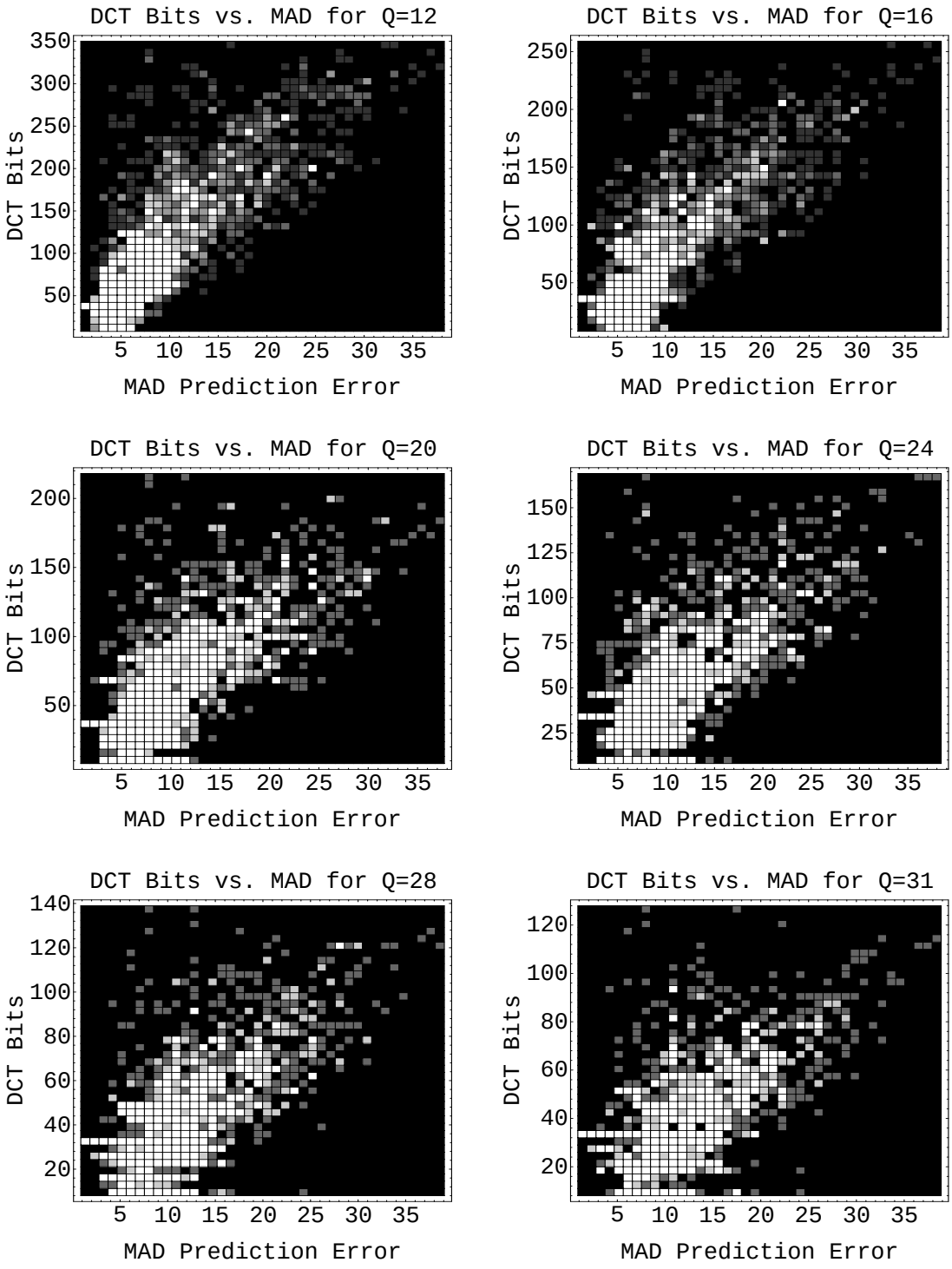


Figure 9: Density plots of DCT coding bits vs. MAD prediction error for first inter-coded frame of Miss America sequence at various levels of quantization.

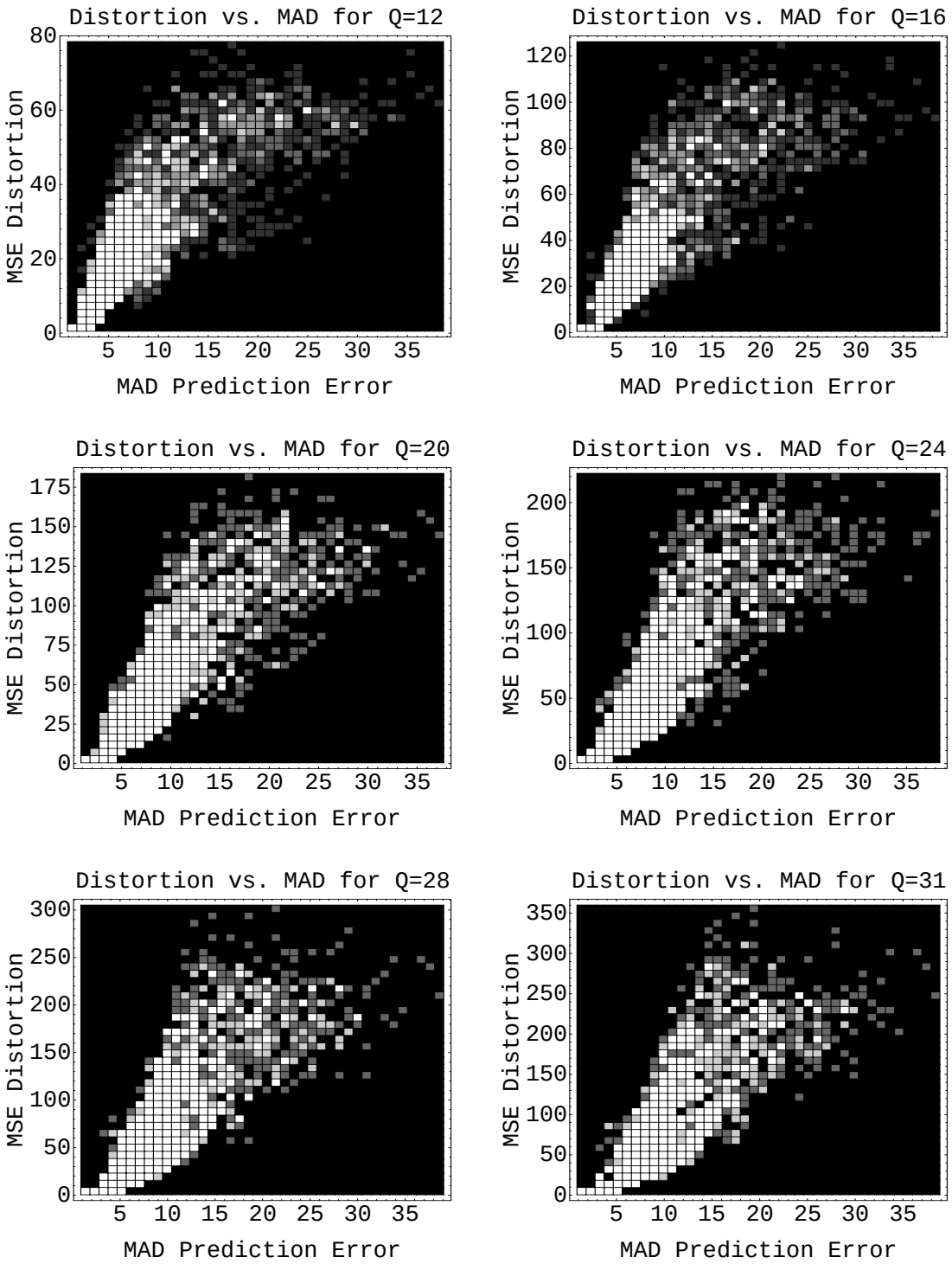


Figure 10: Density plots of MSE reconstruction distortion vs. MAD prediction error for first inter-coded frame of Miss America sequence at various levels of quantization.

5.3.1 Curve Fitting

To determine the coefficients for the heuristic functions, we used linear least squares regression. For each value for the quantizer step size Q, we trained a set of coefficients and stored them in a lookup table. For the heuristic functions given in Equations 3–5, we performed curve fitting to the $R + \lambda D$ statistic, as discussed earlier.

5.3.2 Rate-Distortion Characteristics

We ran H1 and H2 using the various heuristic functions on the Miss America and Claire sequences, varying values of the quantization step size Q. The resulting averaged rate-distortion curves are plotted in Figures 11 and 12. In particular, the bitrate consumption of the H1 and H2 coders using the linear heuristic function is tabulated for both test sequences in Tables 9–12. Compared with the results in Tables 1–8, the H1 and H2 coders use fewer bits to code motion vectors, while spending more bits to code DCT coefficients. As expected, the heuristic coders generate more bits than the explicit minimization coders for the same quantization step size. On the other hand, since the heuristic coders incorporate the prediction error in the cost function, they result in less reconstruction distortion. This tradeoff between rate and distortion is favorable for the Claire sequence but not for the Miss America sequence. However, for both test sequences, the heuristic coders perform better than the base PVRG coder.

For the Claire sequence, H1 performs slightly better than M1, and H2 better than M2. However, for the Miss America sequence, the performance of H1 lies between PVRG and M1. The performance of H2 is competitive with M2 when using the linear fit.

5.3.3 Rate Control

As in section 4.4.2, we ran the heuristic coders with rate control targetted to 16 kbps. Comparative plots of the resulting average PSNR are shown in Figures 15–18. These results show that the heuristic coders perform as well as the explicit minimization coders with rate control.

6 Conclusion and Future Work

We have shown that choosing motion vectors and coding control to (greedily) minimize codelength, a combination of codelength and distortion, and computationally efficient approximations thereof, yields substantial improvements in rate-distortion performance when coding video at low rates.

In general, the Lagrangian parameter λ in Equation 1 is dependent on characteristics of the block being coded and could vary as the statistics of the input sequence vary. In this paper, we considered only the simple case of using a fixed λ. An online adaptation of λ to track variations in the input sequence is certainly possible and would result in more robust coders. Since λ controls rate to some extent, it could be used in conjunction with the quantization step size in performing rate control. Preliminary investigations along these lines show promising improvements.

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 (see [3]), 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 modelled. At a glance, it does not seem as if this would work, since the most popular methods for nonparametric regression require

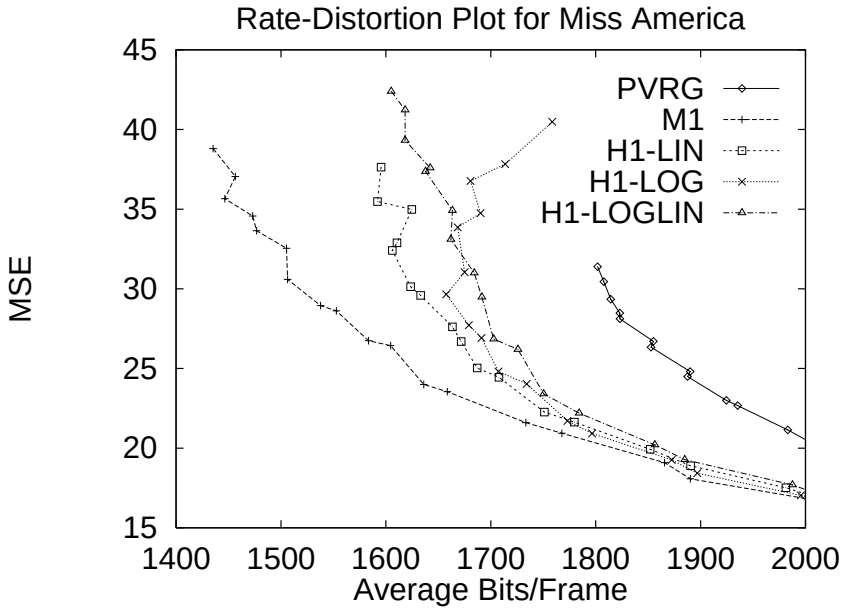


Figure 11: Comparison of averaged rate-distortion for coding Miss America sequence with H1 coder, with parameters determined by least squares curve fitting.

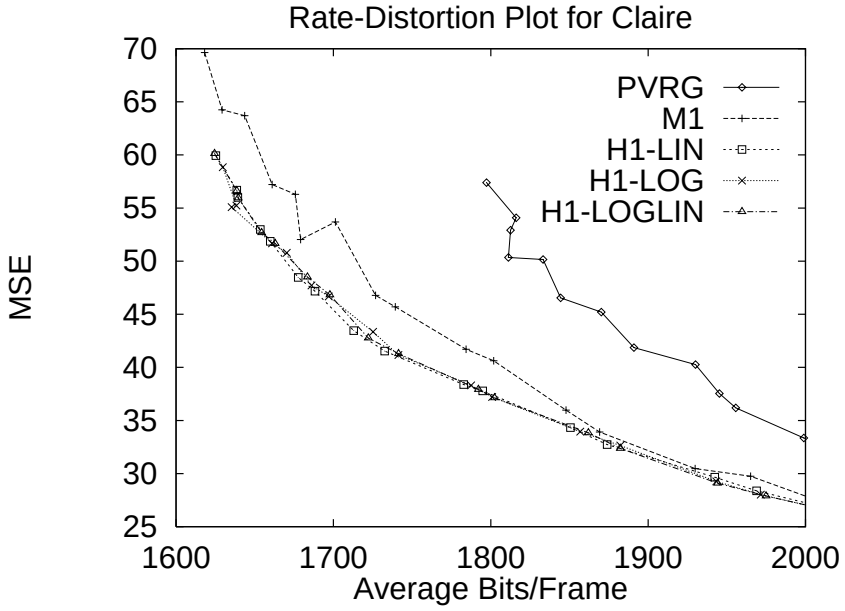


Figure 12: Comparison of averaged rate-distortion for coding Claire sequence with H1 coder, with parameters determined by least squares curve fitting.

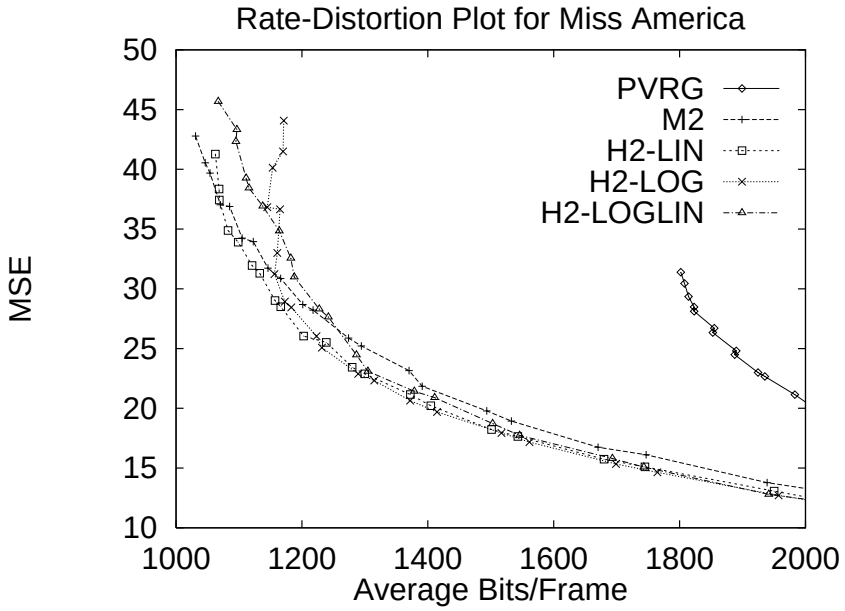


Figure 13: Comparison of averaged rate-distortion for coding Miss America sequence with H2 coder, with parameters determined by least squares curve fitting.

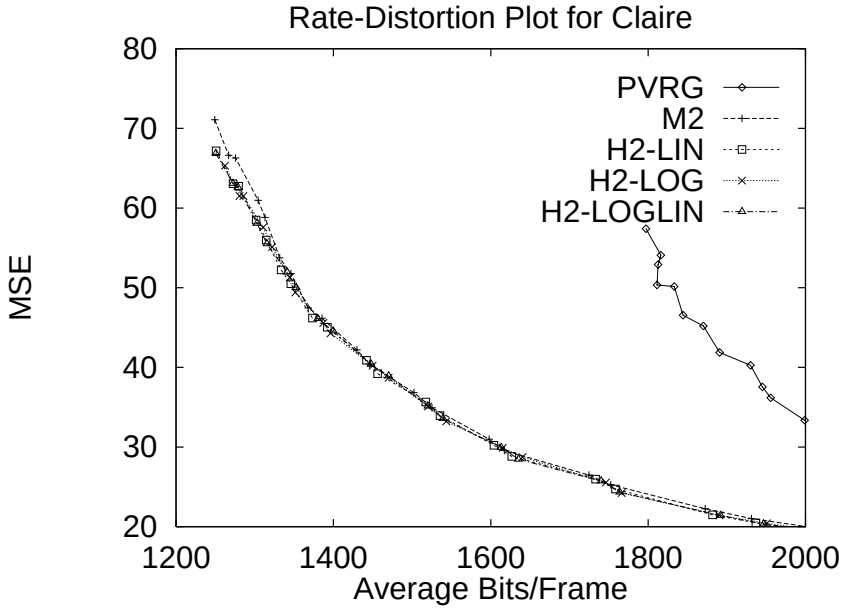


Figure 14: Comparison of averaged rate-distortion for coding Claire sequence with H2 coder, with parameters determined by least squares curve fitting.

Table 9: Coding Miss America sequence using H1-LIN coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	1399.52	160.621	188.069	1050.83	32.3676
28	1409.41	168.379	195.448	1045.59	32.9559
24	1453.24	202.483	203.793	1046.97	33.7169
20	1528.97	278.621	206.793	1043.55	34.6541
16	1744.07	470.552	207.552	1065.97	35.6986
12	2177.86	877.931	202.793	1097.14	37.1852
8	3296.07	1943.52	183.069	1169.48	39.3141

Table 10: Coding Miss America sequence using H2-LIN coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	933	130.104	192.833	610.062	31.9721
28	950.062	137.625	200.021	612.417	32.7019
24	1019.69	180.604	208.229	630.854	33.5
20	1135.73	260.021	215.729	659.979	34.429
16	1347.85	425.729	215.396	706.729	35.5227
12	1775.65	774	215.958	785.688	36.9613
8	2860.17	1734.21	212.208	913.75	39.2417

Table 11: Coding Claire sequence with H1-LIN coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	1369.34	101.552	187.379	1080.41	30.3479
28	1381.59	116.379	186.552	1078.66	30.8797
24	1413	147.483	186.379	1079.14	31.7386
20	1518.83	255.69	182.034	1081.1	32.759
16	1683.07	419	175.517	1088.55	34.1155
12	2027.17	761.552	167	1098.62	35.661
8	2810.03	1544.24	148.103	1117.69	38.0693

Table 12: Coding Claire sequence with H2-LIN coder.

Q	Bits/Frame	DCT Bits	MV Bits	Side Bits	PSNR
31	981.966	88.8276	186.759	706.379	29.8383
28	1017.69	114.655	184.172	718.862	30.4376
24	1061.79	139.103	184.069	738.621	31.4659
20	1174.1	226.586	185.103	762.414	32.5917
16	1347.34	377.345	178.414	791.586	33.9697
12	1673.41	682.31	172.241	818.862	35.5955
8	2500.79	1463.69	154.207	882.897	38.0831

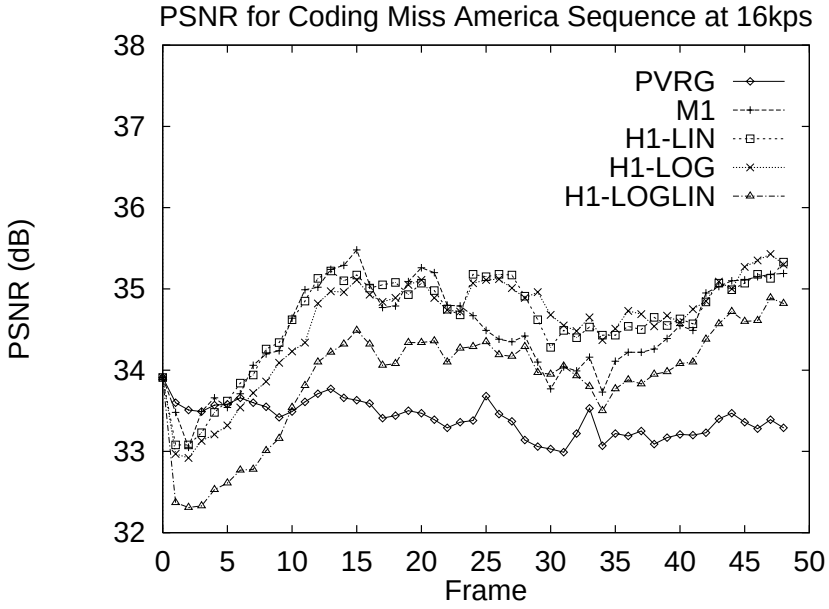


Figure 15: Distortion for coding Miss America sequence at 16kbps with rate control. Comparison of H1 heuristic with M1 coder.

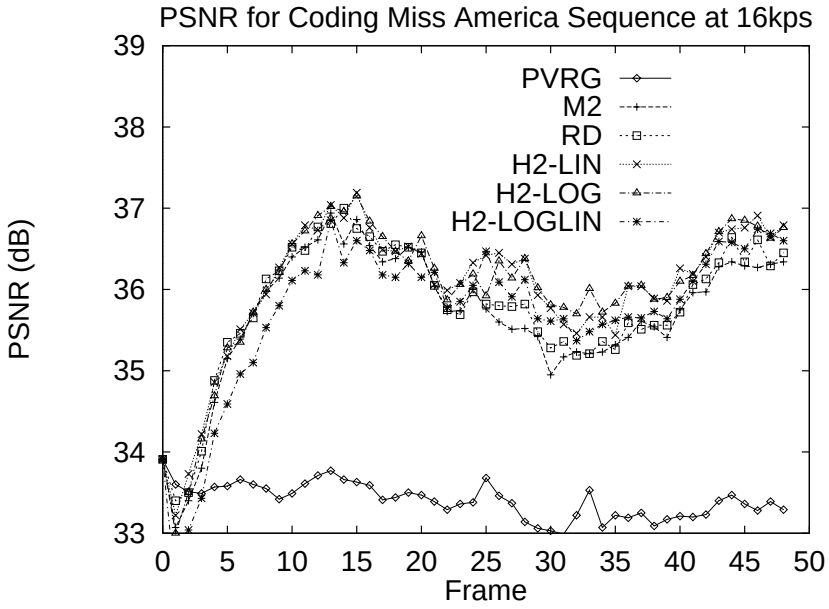


Figure 16: Distortion for coding Miss America sequence at 16kbps with rate control. Comparison of H2 heuristic with M2 and RD coders.

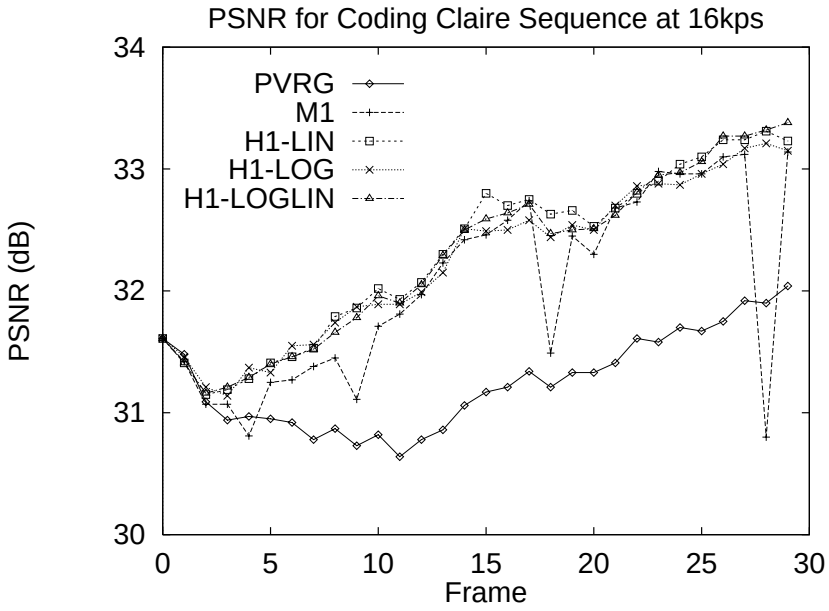


Figure 17: Distortion for coding Claire sequence at 16kbps with rate control. Comparison of H1 heuristic with M1 coder.

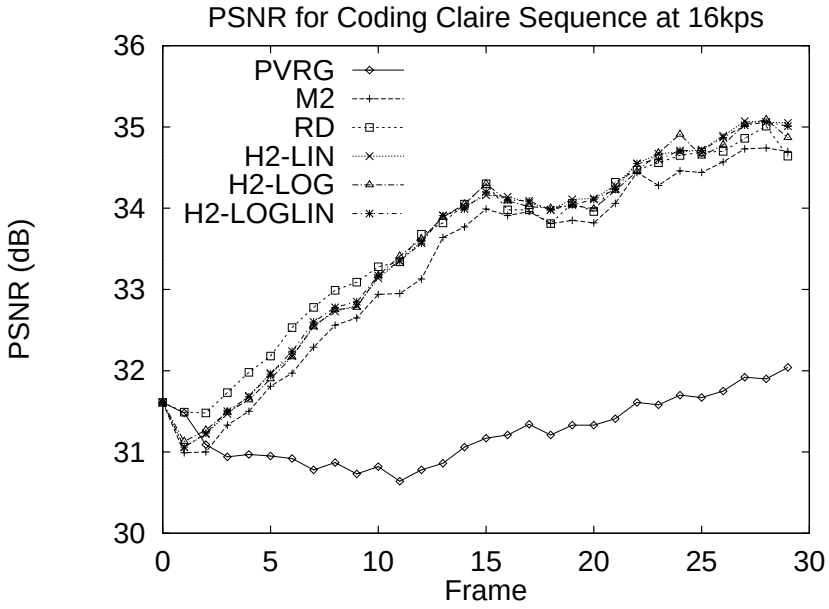


Figure 18: Distortion for coding Claire sequence at 16kbps with rate control. Comparison of H2 heuristic with M2 and RD coders.

a lot of space to represent their estimated functions, and a fair amount of time to evaluate them. However, since the number of values, say of MAD, that are likely to be observed in practice is fairly small, the function H obtained could be stored cheaply and evaluated quickly by simply using a ROM.

We are also interested in examining whether other quickly computed statistics are better indicators of the value of a particular motion vector, or whether they could fruitfully be 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. As discussed earlier, other reasonable candidates include the DC transform coefficient and possibly a handful of other transform coefficients.

For the purposes of computational efficiency, it would be nice if the parameters used in any parametric method could be fixed independently of the quantization step size, with good results. Preliminary indications are that this is the case for some of the heuristic functions considered in this paper.

Finally, it would be interesting to determine how well the methods of this paper work in conjunction with 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 the H.263 standard. An example of the use of the techniques of this paper in a coder where the motion field is encoded using a quadtree is described in [4].

References

[1] CCITT. Video codec for audiovisual services at $p \times 64$ kbit/s, 1990. Study group XV – Report R 37.

[2] D. Le Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, Apr. 1991.

[3] W. Haerdle. *Smoothing Techniques.* Springer Verlag, 1991.

[4] D. T. Hoang, P. M. Long, and J. S. Vitter. Explicit bit-minimization for motion-compensated video coding. In *Proceedings of the 1994 IEEE Data Compression Conference*, pages 175–184, Snowbird, UT, 1994.

[5] A. C. Hung. Pvrg-p64 codec 1.1, 1993. Available from Stanford University by anonymous ftp.

[6] J.R. Jain and A.K. Jain. Displacement measurement and its application in interframe coding. *IEEE Transactions on Communications*, COM-29(12):1799–1808, 1981.

[7] H. Li, A. Lundmark, and R. Forchheimer. Image sequence coding at very low bitrates: A review. *IEEE Transactions on Image Processing*, 3(5):589–609, Sep. 1994.

[8] M. Liou. Overview of the $p \times 64$ kbit/s video coding standard. *Communications of the ACM*, 34(4):60–63, Apr. 1991.

[9] A. Singh. *Optic Flow Computation.* IEEE Computer Science Press, 1991.