Matching Pursuit Video Coding—Part II: Operational Models for Rate and Distortion

Ralph Neff and Avideh Zakhor
Electrical Engineering and Computer Science
University of California, Berkeley

Abstract

We introduce two models for predicting the rate and distortion of the matching pursuit video codec. The first model is based on a pre-coding analysis pass using the full matching pursuit dictionary. The second model is based on a reduced-complexity analysis pass. We evaluate these models for use within existing rate-distortion optimization techniques. Our prediction results suggest that the models have sufficient accuracy to be useful in this context, and that significant complexity reductions could be achieved compared to exact rate-distortion computation.

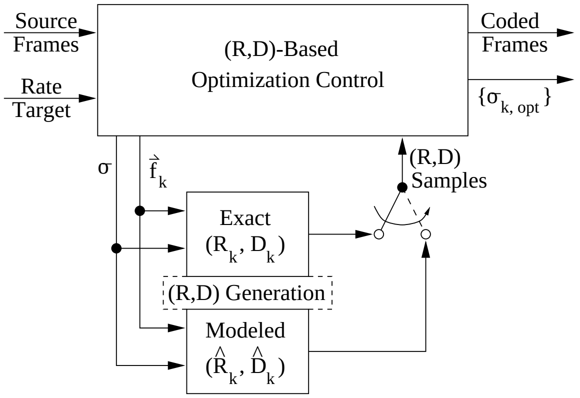
I. Introduction

Operational rate-distortion techniques have recently been used to improve the performance of DCT-based video codecs [1]. Such methods seek to distribute a limited number of bits across multiple coding units in order to minimize a given distortion metric [2]. For the DCT case, the coding units may be individual macroblocks, groups of macroblocks, or entire video frames. A control parameter such as the quantizer stepsize is used to set the rate-distortion tradeoff for each coding unit. An optimization method is then employed to find the set of control parameters which minimize the distortion across all coding units in a window of interest, while simultaneously meeting a given constraint on the rate.

Various examples of this technique have been published for the DCT case (e.g. [2][3][4][5]). A review of these methods may be found in [6]. Each uses a different optimization method to solve a different variation of the optimal bit allocation problem. However, all four methods base their optimization decisions on sampled rate-distortion points (R, D). They thus fit into the "(R, D) farming" framework illustrated in Figure 1. The "Optimization Control" block in the figure represents a particular method for solving the optimal bit allocation problem. This block takes as input the original video frames along with information about the rate constraint. We assume that each frame represents a single coding unit, and that the rate-distortion tradeoff for frame k is a function of control parameter σ_k. The problem is thus to choose the control parameter set

The authors would like to acknowledge the support of NSF grant CCR-9903368.

Fig. 1. Rate-distortion farming framework for solving the optimal bit-allocation problem. The large box represents the optimization method, which relies on sampled rate-distortion points (R_k, D_k) in order to solve for optimal control parameters $\{\sigma_{k,opt}\}$. The samples may be exactly computed or may be generated by a reduced complexity model.



$\{\sigma_k, k = 1...M\}$ which minimizes some distortion metric for the M frames of interest while meeting the given rate constraint. Call this optimal set $\{\sigma_{k,opt}\}$. The optimization block accomplishes this by computing the (R, D) relationship for the various frames over many candidate values of control parameter σ. This function is performed by the "Exact (R_k, D_k)" block of the figure. This block encodes residual signal $\vec{f_k}$ using a given control parameter σ, and returns the resulting (R, D) point. The optimization block thus obtains sampled (R, D) points, and uses them to solve for $\{\sigma_{k,opt}\}$.

In principle, such optimization methods are not DCT-specific, and so could be applied to the matching pursuit video codec described in Part I of this article [7]. The main obstacle is complexity. Matching pursuit encoding is based on an exhaustive local inner product search, which makes it several times more complex than a comparable DCT-based encoder [8]. The (R, D) farming process used in typical optimization methods requires each frame k to be coded with many values of control parameter σ_k. Moreover, the interframe dependency of motion compensated video causes a combinatorial explosion in the number of required frame encodings [2]. For this reason, the optimization complexity for matching pursuit would be much greater than for the DCT case.

One way to reduce this complexity is to replace exact (R, D) computation with reduced complexity models. This is shown in Figure 1 as a second switchable rate-distortion generation unit labeled "Modeled $(\hat{R}_k, \hat{D}_k)$". Such models must satisfy two requirements in order to be useful. First, they must be less complex than exact (R, D) computation. Second, the models must be accurate enough so that obtained parameter set $\{\sigma_{k,opt}\}$ performs nearly as well as the set obtained through exact (R, D) farming.

In this paper, we develop operational (R, D) models for the matching pursuit video codec. This is the first work to show such models for matching pursuit rate and distortion, although various models have been shown for DCT-based systems (e.g. [9][10][5]). This paper does not address theoretical (R, D) functions for overcomplete expansions. Such functions are not known for matching pursuit, but interesting results pertaining to frame expansions are presented in [11].

We develop two classes of models. The first relies on an initial matching pursuit decomposition using the full dictionary, and is thus called the Full Pre-Code Model. Given such an analysis pass, the model predicts changes in (R, D) due to changes in the in-loop quantizer $Q(\cdot)$ and the number of coded matching pursuit atoms N. Many (R, D) points are generated economically from a single analysis pass. The second model class also uses an initial analysis pass, but applies a subset of the full dictionary. For this reason, it is called the Reduced Complexity Pre-Code Model. The reduced analysis pass is used to predict the (R, D) behavior of the full dictionary decomposition. The resulting models are less accurate than the Full Pre-Code Model, but they allow a tradeoff between complexity and prediction accuracy.

We begin in Section II with a review of previous work. We then proceed to develop the two models described above. The Full Pre-Code model is developed in Section III, and the Reduced Complexity Pre-Code models are developed in Section IV. In Section V, we evaluate the two matching pursuit model classes for use within (R, D) based optimization algorithms. Our results show that the models are sufficiently accurate to be used in this context, and that significant complexity reductions may be achieved compared to exact (R, D) farming.

II. Review of Previous Work

Matching pursuit video compression was introduced in [12][8], and reviewed earlier in Part I of this article [7]. The basic algorithm uses a simple rate control method in which the quantizer $Q(\cdot)$ is fixed and the rate varies only with the number of coded atoms N. Essentially, the encoder continues to find atoms until either the rate R is exhausted, or some desired value of distortion D is reached.

More generally, the rate-distortion tradeoff depends on both the quantizer and the number of coded atoms. It is thus possible to improve coding efficiency by jointly adapting $Q(\cdot)$ and N to

TABLE I

Data given by initial decomposition of current frame $\vec{f}$ using operating point $\{Q(\cdot), N\}$.

Pass 1 Moduli:	$\{p_i,$	$i = 1...N\}$
Pass 1 Quant Error:	$\{d_i \overset{\triangle}{=} p_i - Q(p_i),$	$i = 1...N\}$
Overall Rate:	$\{R_i,$	$i = 1...N\}$
Overall Distortion:	$\{D_i \overset{\triangle}{=} \lVert\vec{f}_i\rVert^2,$	$i = 0...N\}$

each coded frame. Two practical algorithms which achieve this were introduced in [13]. The first uses an unquantized matching pursuit decomposition of the current frame to design $Q(\cdot)$, and a second quantized pass to determine N. We refer to this method as "2-pass adaptive quantization." The second algorithm avoids the unquantized analysis pass and instead uses data from a previously coded matching pursuit frame to design $Q(\cdot)$. This method is calld "1-pass adaptive quantization." It achieves nearly the same coding efficiency as the 2-pass algorithm at about half the complexity. Both methods restrict $Q(\cdot)$ to be a uniform quantizer with an expanded deadzone, and so $Q(\cdot)$ is typically specified in terms of the stepsize-to-deadzone ratio $Z = QP/DZ$ where QP is the quantizer step size and DZ is the distance from zero to the edge of the dead zone [13].

The three quantizer design methods (fixed, 2-pass, and 1-pass) are significant to the current work because the complexity of both exact and modeled (R, D) farming depends on which quantizer design method is used. We will thus take quantizer design into consideration in Section V when we compute the complexity reduction achieved by each of the two (R, D) model classes.

III. The Full Pre-Code Model

Assume that the current motion residual frame has been encoded once with some initial quantizer $Q(\cdot)$ and number of atoms N. This initial encoding pass produces a trace of values for modulus p_i and modulus quantization error $d_i \overset{\triangle}{=} p_i - Q(p_i)$, as well as cumulative rate R_i and distortion $D_i \overset{\triangle}{=} ||\vec{f}_i||^2$ for each coded atom i. This information is summarized in Table I. The problem is then to predict the rate-distortion performance for encoding the same motion residual frame with a different quantizer $Q'(\cdot)$ and number of atoms $N' < N$. By solving this problem, we enable any number of rate-distortion points to be quickly estimated from a single analysis encoding pass. We begin in Section III-A with a model for distortion, and continue in Section III-B with a model for rate. Prediction results are presented in Section III-C.

A. Distortion Model

Based on the set of known "Pass 1" values from Table I, we seek to predict the overall distortion $\{D'_i, i = 0...N\}$ for a second "Pass 2" encoding of $\vec{f}$ using a different quantizer $Q'(\cdot) \neq Q(\cdot)$. Additional Pass 2 parameters such as $\{p'_i, d'_i, i = 1...N\}$ will be predicted along the way. We will consistently use prime notation to distinguish Pass 2 values such as D'_i from their Pass 1 counterparts. Predicted Pass 2 values will be distinguished from their true counterparts through the use of hat notation. For example, $\hat{p}'_i$ is a prediction for p'_i.

Define $\overline{ER}_i$ to be the single-stage reduction in distortion due to the encoding of the ith atom. This was shown in [13] to be

$$\begin{aligned} \overline{ER}_i &\stackrel{\triangle}{=} D_{i-1} - D_i \\ &= p_i^2 - d_i^2 \end{aligned} \tag{1}$$

By summing $\overline{ER}_i$ over the first i stages, one may calculate $\overline{CR}_i$, the cumulative reduction in distortion due to the first i atoms:

$$\begin{aligned} \overline{CR}_i &\stackrel{\triangle}{=} D_0 - D_i \\ &= \sum_{j=1}^{i} p_j^2 - \sum_{j=1}^{i} d_j^2 \end{aligned} \tag{2}$$

Equation 2 shows that $\overline{CR}_i$ has separate contributions from unquantized modulus p_i and quantization error d_i. We will consider these separately in developing our model. Rearranging Equation 2 gives a useful expression for distortion:

$$D_i = D_0 - \sum_{j=1}^{i} p_j^2 + \sum_{j=1}^{i} d_j^2 \tag{3}$$

We apply Equation (3) to the Pass 2 values and use the fact that $D'_0 = D_0$ to obtain the following progressive method for computing Pass 2 distortion:

$$D'_i = \begin{cases} D_0 & \{i = 0\} \\ D'_{i-1} - (p'_i)^2 + (d'_i)^2 & \{i = 1...N\} \end{cases} \tag{4}$$

Accurate predictions of p'_i and d'_i will thus allow us to predict distortion D'_i according to Equation 4. In order to predict p'_i and d'_i, we must first consider how the modulus and quantization error change when the quantizer changes from $Q(\cdot)$ to $Q'(\cdot)$.

We first consider how modulus values p_i change with the quantizer. The majority of this change occurs because a change in the in-loop quantizer affects the accuracy of the coded moduli and thus alters the effectiveness of the matching pursuit process. For example, if $Q'(\cdot)$ introduces more quantization error than $Q(\cdot)$, then each Pass 2 atom is less effective at removing residual energy then the corresponding Pass 1 atom. The distortion reduction process is consequently slower for Pass 2 than for Pass 1. This results in an overall increase in the modulus values seen in Pass 2, since at a given stage i there will typically be more energy remaining in the progressive residual image $\vec{f_i}$, and so the computed inner product for that stage will typically be larger. Another way to state this is that if $D'_i > D_i$ on the average, then $|p'_i| > |p_i|$ will also be true on average.

Our model for p'_i is thus based on the change in signal energy, and follows from the empirical observation that Pass 1 and Pass 2 modulus values for stage i are approximately related as:

$$p'_i \approx p_i \sqrt{\frac{D'_{i-1}}{D_{i-1}}} \tag{5}$$

As shown in Table I, p_i and D_{i-1} are known from Pass 1. Pass 2 distortion values D'_{i-1} are the target values to be predicted, and so are not known exactly. However, we may predict these values progressively according to Equation (4). This allows us to predict p'_i as:

$$\hat{p}'_i = p_i \sqrt{\frac{\hat{D}'_{i-1}}{D_{i-1}}} \tag{6}$$

We must also consider the quantization error component. We assume that Equation (6) provides a reasonable estimate of p'_i, and so we estimate d'_i by re-quantizing this estimate using the known Pass 2 quantizer:

$$\hat{d}'_i = \hat{p}'_i - Q'(\hat{p}'_i) \tag{7}$$

Using the approximated values from Equations (6) and (7), we may then estimate distortion D'_i according to Equation (4) as:

$$\hat{D}'_i = \hat{D}'_{i-1} - (\hat{p}'_i)^2 + (\hat{d}'_i)^2 \tag{8}$$

Equations (6), (7), and (8) allow us to progressively estimate Pass 2 distortion over the N stages for which Pass 1 was initially performed. The algorithm is summarized in Figure 2.

It will be useful to evaluate prediction results in terms of the separate contributions to distortion

Fig. 2. Summary of the Full Pre-Code distortion model.

```
Given:      $p_i, d_i, D_i$ from Pass 1 with $Q(\cdot)$
Predict:    $p'_i, d'_i, D'_i$ for Pass 2 with $Q'(\cdot)$
----------------------------------------------------------
for $i = 1{:}N$                          /* For each atom stage */
   if($i == 1$)                          /* Predict modulus */
      $\hat{p}'_i = p_i$
   else
      $\hat{p}'_i = p_i \sqrt{\frac{\hat{D}'_{i-1}}{D_{i-1}}}$
   endif
   $\hat{d}'_i = \hat{p}'_i - Q'(\hat{p}'_i)$                /* Predict Q-error */
   $\hat{D}'_i = \hat{D}'_{i-1} - (\hat{p}'_i)^2 + (\hat{d}'_i)^2$    /* and Distortion */
end
```

reduction defined in Equation (2). Define the p^2 and d^2 contributions to $\overline{CR}_i$ respectively as:

$$\rho_i \stackrel{\triangle}{=} \sum_{j=1}^{i} p_j^2 \tag{9}$$

$$\delta_i \stackrel{\triangle}{=} \sum_{j=1}^{i} d_j^2 \tag{10}$$

Now consider the change in each of these contributions due to the change in quantizers from $Q(\cdot)$ to $Q'(\cdot)$. Define these contribution changes as:

$$\Delta\rho_i \stackrel{\triangle}{=} \rho'_i - \rho_i \tag{11}$$

$$\Delta\delta_i \stackrel{\triangle}{=} \delta'_i - \delta_i \tag{12}$$

where $\{\rho'_i, \delta'_i\}$ are the equivalent contributions to $\overline{CR}'_i$, the cumulative energy reduction for Pass 2. By replacing $\{\rho'_i, \delta'_i\}$ with $\{\hat{\rho}'_i, \hat{\delta}'_i\}$ in Equations 11 and 12, we obtain predicted values for contribution changes $\Delta\rho_i$ and $\Delta\delta_i$. We use these to evaluate how well the model predicts the separate contributions to the quantizer-related change in distortion D_i.

Prediction results are illustrated in Figure 3. A single frame of the Mobile sequence was encoded to approximately 33 kbits. This corresponds to 1 Mbit/s at 30 frame/s which is the standard MPEG-4 test point for Mobile. The frame was encoded twice with two different quantizers. These were generated using the 2-Pass adaptive quantizer algorithm [13] with stepsize-to-deadzone ratios of 0.6 and 1.0 for quantizers $Q(\cdot)$ and $Q'(\cdot)$, respectively. By design, $Q(\cdot)$ is a finer quantizer than $Q'(\cdot)$, and so we expect D_i to be lower than D'_i.

Figures 3a and 3b show the real and predicted values of $\Delta\rho_i$ and $\Delta\delta_i$ plotted against i. We see that $\Delta\delta_i$ is predicted extremely well by the simple requantization model of Equation (7). The change in modulus component ρ_i is predicted with less accuracy, and some error can be seen in the plot. We attribute this error to an effect we call "atom path noise." This essentially means that the in-loop quantization of atom i affects the selection of all later atoms. In particular, changing the quantizer at one stage introduces random changes to the later atom paramters. Although these changes are difficult to model, we have shown the associated change to modulus p_i is small in practice [6]. A comparison of vertical plot scales shows that the overall change in ρ'_i is less than that of δ'_i by about a factor of four. Accurate prediction of δ'_i is thus more critical to the end prediction of D'_i.

Figure 3c shows the end distortion prediction, effectively combining the results shown in Figures 3a and 3b. The Pass 1 distortion curve D_i is shown as the solid line, and the real and predicted Pass 2 distortion curves are shown by the two dotted lines. The predicted distortion is extremely close to the true value. The absolute prediction error is plotted in Figure 3d, and two different relative error measurements are shown in Figures 3e and 3f. Figure 3e shows the prediction error relative to the gap between D' and D and is defined as:

$$e_{|D'-D|} = \frac{|D' - \hat{D}'|}{|D' - D|} \tag{13}$$

Figure 3f shows the prediction error relative to Pass 2 distortion D', which is the end quantity being predicted. This error measure is defined as:

$$e_{|D'|} = \frac{|D' - \hat{D}'|}{|D'|} \tag{14}$$

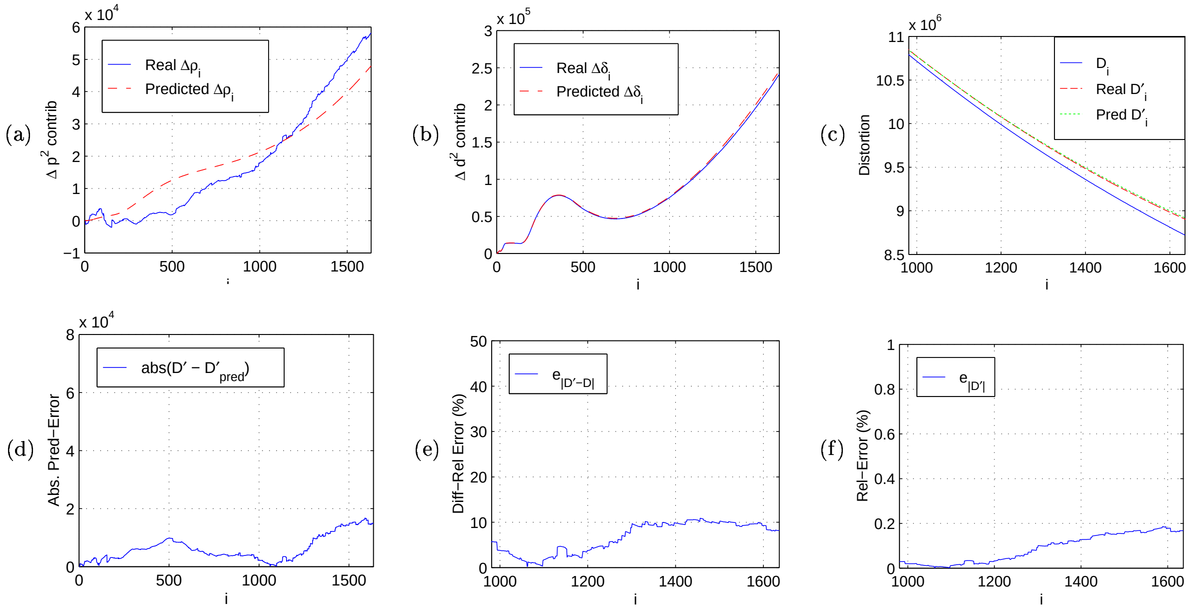
From these two plots, we see that the change in distortion is predicted with at most a ten percent relative error, and that the distortion itelf is predicted with a much lower relative error of around 0.2 percent. These results are typical for other sequences and rates, as shown by additional experiments in [6]. Results for a wider range of sequences will be shown in Section III-C.

B. Rate Model

We now show a corresponding rate model. This model is conceptually simpler than the corresponding distortion model, and typical rate prediction errors of less than 1 percent will be shown.

As shown in Table I, values for overall rate $R_i, i = 1...N$ are known from Pass 1. This rate may

Fig. 3. Illustration of the Full Pre-Code distortion model. (a) Real and predicted $\Delta\rho_i$. (b) Real and predicted $\Delta\delta_i$. (c) Pass 1 distortion D_i compared to real and predicted values of Pass 2 distortion D'_i. (d) Absolute prediction error. (e) Percent differential relative prediction error, $e_{|D'-D|}$. (e) Percent relative prediction error, $e_{|D'|}$.



be expressed as the sum of bits to header, motion, and atoms:

$$R_i = R_{hdr} + R_{mot} + R_{atm(i)} \tag{15}$$

We seek to predict overall Pass 2 rate R'_i. This task falls entirely to predicting the Pass 2 atom budget $R'_{atm(i)}$, since R'_{hdr} and R'_{mot} are independent of the quantizer and are known at the start of atom coding. For any stage i, the Pass 2 atom budget $R'_{atm} = R'_{atm(i)}$ may be written as

$$R'_{atm} = R'_Q + R'_P + R'_I \tag{16}$$

where R'_Q, R'_P, and R'_I are respectively the total number of bits to modulus quantization, atom position, and basis indices. R'_Q may be predicted in a straightforward manner. Recall that within the full distortion model of Figure 2, the Pass 2 modulus values are predicted and requantized using $Q'(\cdot)$. By VLC coding the first i predicted quantizer indices, we obtain $\hat{R}'_Q$, a prediction of the Pass 2 quantizer rate.

As shown in [13], the change in overall bit rate due to a change in the quantizer is seen almost entirely in the rate to quantization. This means the rate variation is almost entirely expressed in R'_Q, and so we predict R'_{atm} as

$$\hat{R}'_{atm} = R_{atm} - R_Q + \hat{R}'_Q \tag{17}$$

Fig. 4. Summary of the Full Pre-Code rate model.

```
Given:      $p_i, d_i, D_i$ for $\{i = 1...N\}$ from Pass 1 with quantizer $Q(\cdot)$
            and all Pass 1 rate information.
Predict:    $R'_i$ for Pass 2 with quantizer $Q'(\cdot)$

For any $i \leq N$
1.  Run full distortion model to predict quantized Pass 2 moduli.
2.  Encode the first $i$ predicted quantizer indices to find $\hat{R}'_Q$.
3.  Predict atom rate using $\hat{R}'_{atm} = R_{atm} - R_Q + \hat{R}'_Q$.
4.  Predict overall rate from $\hat{R}'_i = R_{hdr} + R_{mot} + \hat{R}'_{atm}$.
```

where R_{atm} and R_Q are the Pass 1 equivalents of R'_{atm} and R'_Q, respectively. The predicted Pass 2 bit rate $\hat{R}'_i$ for any stage i may thus be computed from the corresponding value of $\hat{R}'_{atm}$ and the known budgets to header and motion vectors. This follows from Equation (15). A summary of this rate prediction model is shown in Figure 4. Experimental results are shown together with results from the distortion model in the next subsection.

C. Prediction Results

We now show prediction results for the distortion and rate models described in Figures 2 and 4. Sample frames from the MPEG-4 test sequences are encoded using an initial quantizer $Q(\cdot)$, and distortion and rate for one or more additional quantizers are then predicted using the models. We design each sample quantizer using the adaptive 2-pass quantizer algorithm in [13]. Each quantizer is specified by stepsize-to-deadzone ratio Z, with larger values corresponding to coarser quantization.

Table II shows distortion prediction error for a wide range of sequences and sample quantizers. The first three prediction error columns were produced by encoding a sample frame from each of the standard MPEG-4 sequences to the number of atoms shown in the table. Each frame was coded using three different pairs of quantizers $\{Q(\cdot), Q'(\cdot)\}$, with each pair showing a progressively larger change in Z. The Full Pre-Code model was used to predict the distortion using $Q'(\cdot)$ from the known $Q(\cdot)$ result, with the resulting prediction error shown in the appropriate column of the table. For this table, we convert each actual and predicted distortion value to the corresponding Peak Signal-to-Noise Ratio value (PSNR) as:

$$PSNR = 10 \log_{10}(255^2/D)$$

Prediction error is then shown in the table as an error in predicted PSNR (dB).

TABLE II

Table of distortion prediction results for sample frames from each of the standard MPEG-4 test sequences.

			Prediction error in PSNR (dB) for these changes in Z:			
Seq	Label	Atoms	$0.8 \rightarrow 1.2$	$0.6 \rightarrow 1.4$	$0.4 \rightarrow 1.6$	$0.0 \rightarrow 1.0$
1	Cont-10	54	0.0032	0.0043	0.0052	0.0050
2	Hall-10	54	-0.0009	-0.0092	-0.0116	0.0011
3	Mom-10	51	-0.0040	-0.0063	-0.0103	0.0004
4	Cont-24	114	-0.0150	0.0123	0.0068	0.0076
5	Silent-24	106	0.0010	-0.0058	-0.0016	0.0010
6	Mom-24	105	-0.0014	-0.0010	0.0091	0.0074
7	Coast-48	206	0.0086	0.0032	-0.0026	0.0071
8	Fore-48	138	0.0079	-0.0063	0.0228	0.0254
9	News-48	180	0.0065	-0.0136	-0.0062	0.0018
10	Coast-112	217	-0.0013	0.0003	0.0000	-0.0002
11	Fore-112	64	0.0033	0.0010	0.0020	-0.0058
12	News-112	238	0.0089	0.0189	0.0195	0.0147
13	Mob-1024	1606	0.0080	0.0087	0.0170	0.0052
14	Stef-1024	1520	0.0054	0.0225	0.0304	0.0045
Average absolute error (dB):			0.0054	0.0081	0.0104	0.0062

The smallest experimental change in Z is $0.8 \rightarrow 1.2$ as shown by the first of the PSNR error columns. The worst case error for this case is .015 dB for Container at 24 kbit/s. All other sequences have prediction errors of less than .01 dB. For the largest quantizer change of $0.4 \rightarrow 1.6$, the maximum prediction error is 0.0304 dB, and eight of the fourteen sequences have errors below .01 dB. Average absolute errors taken across all sequences are shown at the bottom of the table. These range from .005 to .01 dB, depending on the size of the quantizer change.

The final column of Table II shows that accurate prediction of distortion can be made from an unquantized first pass, that is $Q(\cdot)$ with $Z = 0.0$. This is a valid case, since the model assumes no restrictions on $Q(\cdot)$. This special case is useful for some of the model applications which will be discussed in Section V-A.

Numerical results for the Full Pre-Code rate model are shown in Table III. Test conditions are identical to those of Table II, with prediction error expressed as a percentage of the actual Pass 2 rate. The table shows extremely small rate prediction errors. Of the 42 reported trials, only three cases show errors above 1%. The largest such error is 1.62% for Container at 10 kbit/s with the largest change in quantizer ratio Z, $0.4 \rightarrow 1.6$. Even for this quantizer pair, only two of 14 sequences show greater than a 1% error in predicted rate. The average error across all sequences ranges from .28 to .70 percent, depending on the quantizer change. The prediction error for the Full Pre-Code rate model is thus typically below 1%, even for large changes in quantizer.

TABLE III

Table of rate prediction results for sample frames from each of the standard MPEG-4 test sequences. Percent error compares predicted rate $\hat{R}'_{atm}$ to actual rate R'_{atm}.

Seq	Label	Atoms	Percent error in rate to atoms for these changes in Z:		
			$0.8 \rightarrow 1.2$	$0.6 \rightarrow 1.4$	$0.4 \rightarrow 1.6$
1	Cont-10	54	0.8921	0.7162	1.6245
2	Hall-10	54	0.0855	0.1708	0.8547
3	Mom-10	51	0.0850	0.0853	0.9442
4	Cont-24	114	0.2779	1.1210	0.7907
5	Silent-24	106	0.4470	0.4466	0.9870
6	Mom-24	105	0.6961	0.8784	1.1537
7	Coast-48	206	0.6286	0.0724	0.7029
8	Fore-48	138	0.0357	0.0718	0.4697
9	News-48	180	0.1015	0.1524	0.3313
10	Coast-112	217	0.5606	0.6222	0.5017
11	Fore-112	64	0.0570	0.3432	0.5166
12	News-112	238	0.0000	0.1604	0.7025
13	Mob-1024	1606	0.0070	0.1022	0.2188
14	Stef-1024	1520	0.0652	0.1545	0.0654
Average percent error:			0.2814	0.3641	0.7046

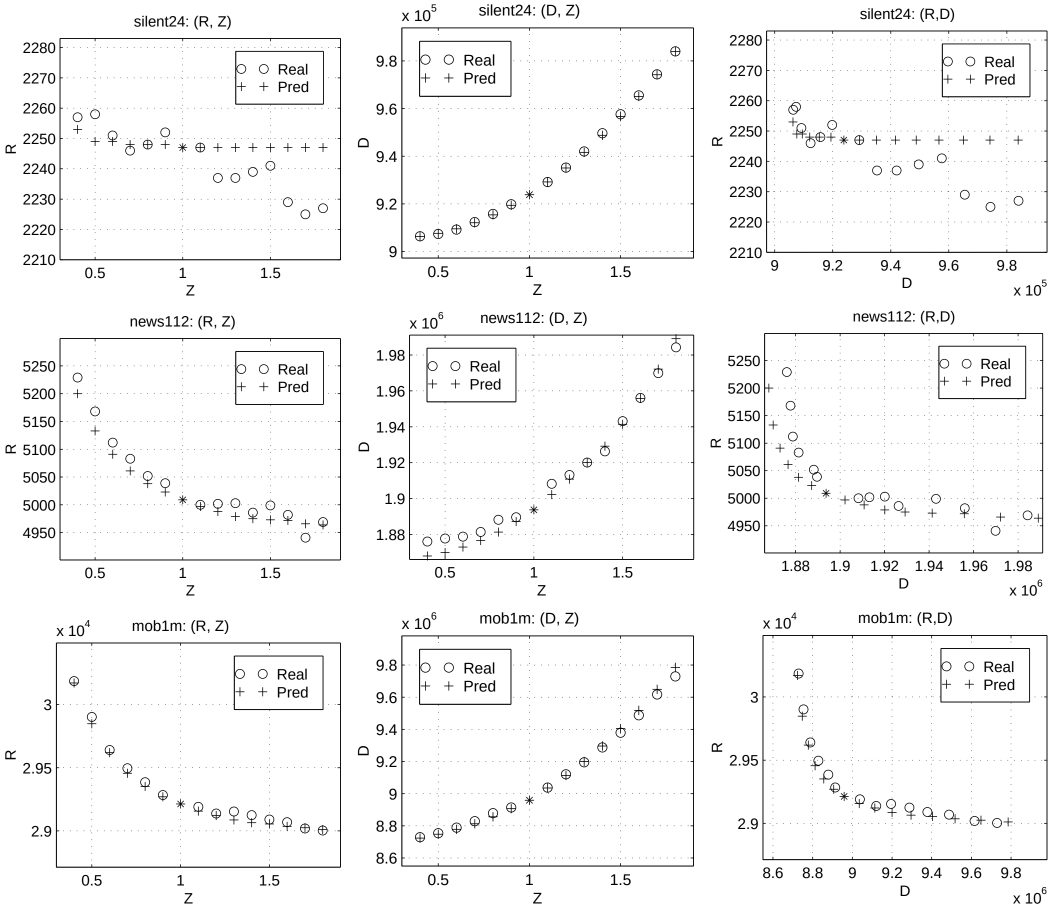
Combined results for overall rate R and distortion D are shown in Figure 5. A frame from each of three sample MPEG-4 sequences is coded using a reference quantizer $Q(\cdot)$, and the resulting Pass 1 data is used to predict (R, D) for fourteen other quantizers over a wide range of Z. Reference quantizer $Q(\cdot)$ is defined using $Z = 1.0$, and is shown as an asterisk symbol (*) on each plot. All results for a single sequence are coded to a common number of atoms N chosen to be typical for the given sequence and bit rate.

The first column of plots shows R plotted against Z. The prediction appears to be poor at the lowest bit rate, but steadily improves as the rate increases. Closer inspection reveals that the poor performance for Silent at 24 kbit/s is really an artifact of scale. At low rates, R is relatively insensitive to the quantizer change. This is reflected by the nearly flat R vs. Z curve predicted for Silent at 24 kbit/s. Atom path effects produce small deviations from the predicted rate on the order of ten to twenty bits, or about the cost of a single atom. The absolute error is thus small, but appears significant due to the small vertical scale of the graph. The corresponding error as a percentage of total bit rate is at most 1%.

The second column of plots in Figure 5 shows distortion D plotted against Z. True distortion is accurately predicted by the model. Some prediction error seen at the low end of News at 112 kbit/s, but a good match is seen at nearly every point of the remaining three sample sequences.

Finally, R is plotted directly against D in the rightmost column of plots in Figure 5. The rate

Fig. 5. Overall rate and distortion (R, D) performance for four sample frames predicted for quantizers defined for a wide range of stepsize to deadzone ratio Z. In each case, the reference quantizer $Q(\cdot)$ is defined using $Z = 1.0$. The (R, D) point for the reference run is shown as (*) in each plot.



prediction effects noted earlier can be seen to cause some mismatch in the two lower rate sequences, especially for Silent at 24 kbit/s. At higher rates, however, the accuracy improves, as illustrated by the result shown for Mobile at 1 Mbit/s. These results illustrate how the Full Pre-Code model may be used to predict the (R, D) performance of matching pursuit for many variations of $Q'(\cdot)$ based on the data from a single analysis pass. We now proceed to develop our second model class, which will reduce the analysis complexity below that of a single matching pursuit pass.

IV. Reduced Complexity Pre-Code Models

The complexity of the Full Pre-Code model is concentrated in an initial pre-analysis pass. This pass relies on the matching pursuit local inner product search, and so has a complexity much greater than the few arithmetic operations per atom used to generate each predicted (R, D) point from the resulting model. To reduce the model complexity, we must decrease the complexity of the pre-analysis step. We hypothesize that useful prediction may be done without the full encoding pass used in the Full Pre-Code model. However, our results shown in [6] suggest that accurate prediction of D_i requires some non-trivial analysis pass to be performed.

A. Reduced Dictionary Analysis

We propose a new (R, D) model based on an analysis pass using a reduced set of dictionary elements. There are two criteria for the design of such a reduced set. First, the basis functions in the reduced set should be representative of those in the full dictionary, so that the two dictionary sets will have similar distortion characteristics. A second design criterion is the complexity of the reduced dictionary. Complexity is determined by the number of basis functions in the reduced set and the spatial extent of those functions. Reducing these factors will in turn reduce the inner-product search complexity.

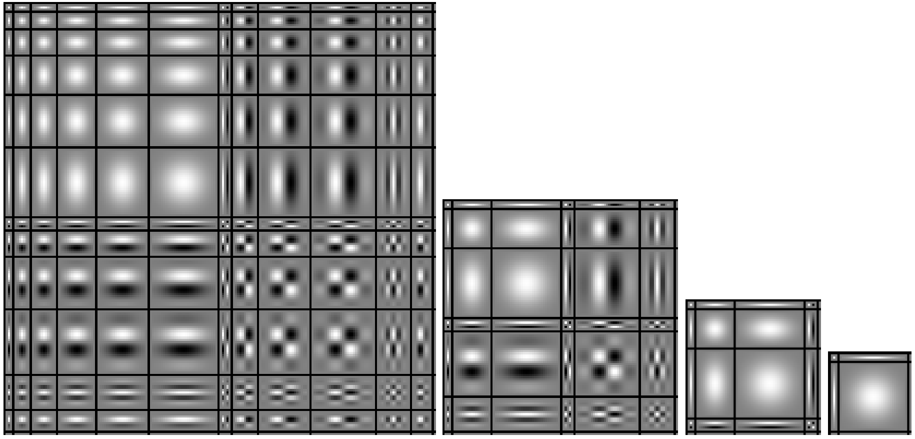
We consider four reduced-complexity dictionary sets, in order of decreasing complexity:

$$\{Red0, Red1, Red2, Red3\}$$

A visualization of each reduced set is provided in Figure 6. Each is a subset of the full 400-element Gabor dictionary shown in Part I of this article [7]. No optimization method was employed to choose each subset. However, during the manual selection process, an attempt was made to maintain the variety of the full Gabor set to the best degree possible in the reduced set. This can be seen by comparing the images in Figure 6 to Figure 2c of [7]. Note that the larger sets $Red0$ and $Red1$ seem to retain the overall four-quadrant structure of the original dictionary. These quadrants might be labeled clockwise from upper left as smooth Gaussian, vertical features, high frequency features, and horizontal features. However, the $Red2$ and $Red3$ sets consist mainly of the smooth Gaussian functions. This lack of variety is required in order to reduce the computational complexity. We calculate the per-atom inner product search complexity associated with each reduced dictionary using a separable inner product search [8]. The resulting complexity numbers are presented in Table IV. To summarize the table, the four reduced dictionary sets respectively have per-atom complexities of 35.7, 11.6, 4.9 and 2.2 percent of the full dictionary.

The reduced complexity distortion model first performs matching pursuit decomposition on the current frame using one of the reduced dictionaries. Data from this analysis pass is used to predict

Fig. 6. Reduced complexity dictionary sets. These are subsets of the full 2-D Gabor dictionary.



*Red*0 *Red*1 *Red*2 *Red*3

TABLE IV
RELATIVE COMPLEXITY OF THE FULL AND REDUCED GABOR DICTIONARIES.

Dictionary	Number of 2-D Bases	Per-Atom Complexity (*)	Relative Complexity, β
Full Gabor	400	1.733	1.0
*Red*0	144	0.607	0.357
*Red*1	36	0.199	0.116
*Red*2	16	0.084	0.049
*Red*3	4	0.037	0.022

(*) Measured in millions of multiply-accumulate operations.

the performance of the full dictionary. Define D_i^F and D_i^R as the distortion for the current frame coded to i atoms using the full and reduced dictionaries, respectively. The equivalent curves from the previous frame are $\dot{D}_i^F$ and $\dot{D}_i^R$. The problem is thus to predict D_i^F from known previous frame curves $\dot{D}_i^F$ and $\dot{D}_i^R$, and current frame pre-analysis curve D_i^R. Define a simple ratio mapping function as:

$$\dot{M}(i) \triangleq \frac{\dot{D}_i^F}{\dot{D}_i^R} \tag{18}$$

This is termed a mapping function because it can be used to map the reduced analysis distortion curve onto prediction space for the full dictionary curve. For the previous frame, we have:

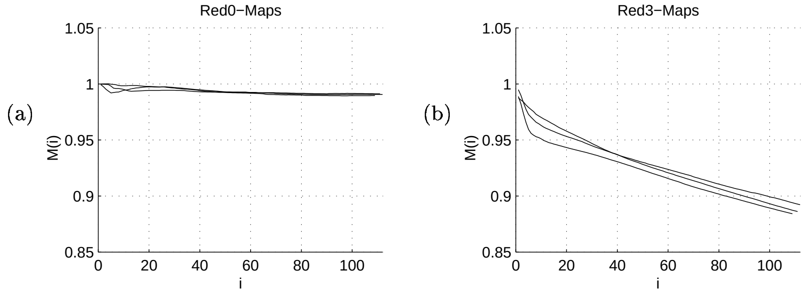
$$\dot{D}_i^F = \dot{M}(i)\dot{D}_i^R \tag{19}$$

By using the known previous frame mapping function, we propose to predict the current frame distortion result as:

$$D_i^F \approx \dot{M}(i)D_i^R \tag{20}$$

The accuracy of the above approximation depends both on how “different” the full and reduced dictionaries are, and on how quickly $\dot{M}(i)$ changes from frame to frame. If the reduced dictionary

Fig. 7. Simple ratio mappings overlaid for successive frames of Mother coded at 24 kbit/s. (a) Mapping Red0 → FullMP for frames $\{5, 6, 7\}$. (b) Mapping Red3 → FullMP for same three frames.

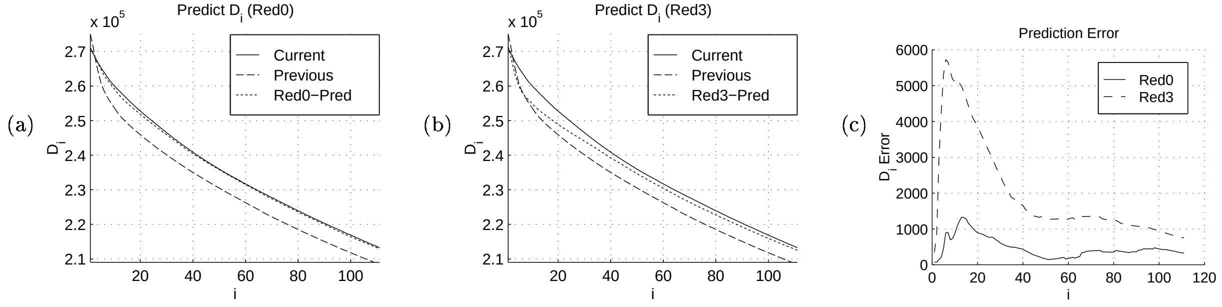


is very close in performance to the full dictionary, then Equation (18) will set $\dot{M}(i) \approx 1$, and Equation (20) effectively reverts to $D_i^F \approx D_i^R$. We expect prediction to work very well for such cases, as will be verified shortly for the *Red*0 dictionary. Suppose instead that the reduced dictionary performance is significantly worse than that of the full dictionary. This is true for the smaller reduced dictionary sets such as *Red*2 and *Red*3. Consistency of the mapping function from frame to frame is more critical in this case. If the mapping function varies slowly from frame to frame, then $M(i) \approx \dot{M}(i)$ will be generally true, and prediction will still work well. However, the model will break down whenever $M(i)$ and $\dot{M}(i)$ are mismatched.

We now illustrate the method using sample distortion curves from a few frames of the Mother 24 kbit/s sequence. We first show the mapping function of Equation (18) as a function of i. Figure 7a overlays the mapping functions which relate *Red*0 to the full dictionary for frames 5, 6 and 7. Note that the mapping function is consistent from frame to frame, and that $M(i) \approx 1$. The *Red*0 dictionary comes close in performance to the full dictionary, even though it has only about a third of the complexity. Figure 7b shows the mapping functions which relate *Red*3 to the full dictionary. The mappings are no longer close to 1, indicating that the less complex *Red*3 dictionary lags behind the full dictionary in coding performance. However, the mapping functions are consistent from frame to frame. This suggests that pre-analysis using *Red*3 dictionary might still be used to predict full dictionary performance.

A sample prediction result is shown in Figure 8. Here the D_i curve of the full dictionary for Frame-7 is predicted from the Reduced Complexity Pre-Code model using the mapping function from Frame-6. Figure 8a uses the *Red*0 dictionary for pre-analysis, while Figure 8b uses the cheaper *Red*3 dictionary. In these two plots, the solid line represents the true D_i for the current frame,

Fig. 8. Illustration of Reduced Complexity Pre-Code distortion model on Mother at 24 kbit/s. D_i of Frame-7 is predicted from a reduced dictionary pass using the Frame-6 mapping. (a) Prediction using Red0 dictionary. (b) Prediction using cheaper Red3 dictionary. (c) Prediction error comparison.



while the fine dotted line shows the predicted curve. The plots show that prediction using $Red0$ is more accurate than that of $Red3$. This is confirmed in Figure 8c, which compares the absolute prediction error. This illustrates an inherent tradeoff between analysis complexity and prediction accuracy.

B. Hybrid Model Extension

In the reduced complexity model of the previous section, the tradeoff between complexity and accuracy is set by the design of the reduced analysis dictionary. The $Red0$ dictionary, for example, uses about one third of the full dictionary complexity to achieve a particular level of accuracy. Some applications may require a continuum of complexity vs. accuracy tradeoffs. Suppose for example that the available computational resources are time varying. It is thus desirable to increase prediction accuracy when more processor cycles are available to do so, and reduce the accuracy when cycles are scarce. The model of the previous section is inconvenient for this case, since each possible complexity-accuracy tradeoff point requires a different reduced-complexity dictionary, and design methods for such dictionaries are still ad-hoc [6].

We introduce a hybrid combination between the Full Pre-Code and Reduced Complexity Pre-Code models which addresses this problem and produces a continuous set of complexity-accuracy tradeoffs. An analysis pass using a reduced dictionary is first performed. This allows the previous frame mapping technique of Equation (20) to be used. One or more full dictionary atoms are then computed and used to improve the prediction by effectively correcting the mapping. Each additional full-dictionary atom increases the analysis complexity and improves the expected prediction accuracy, allowing a more continuous set of tradeoffs to be achieved. Since these tradeoffs are visited progressively, it is possible to stop at any time, thus the desired complexity point need not be known in advance.

We now define the specifics of the hybrid model. Define the anchor point a as the number of full dictionary atoms which have been computed for the current frame and are available for the prediction. Since the end model may be used to predict many (R, D) points with varying quantizers, we do not commit to a particular quantizer for computing the anchor atoms. Instead we compute them without in-loop quantization, and we employ the Full Pre-Code model of the previous section to estimate quantization effects. An example of using Full Pre-Code with an unquantized first pass was previously shown in the final column of Table II. The same technique is used here to produce a limited number of Full Pre-Code distortion points to be used within the Hybrid model. Denote these as $\{D_i^{PC}, i = 1...a\}$. Next define a mapping correction factor μ_a as:

$$\mu_a \triangleq \left. \frac{D_i^{PC}}{D_i^R \dot{M}(i)} \right|_{i=a}$$

This factor is defined in such a way that it corrects the previous frame mapping at the anchor point. Consider modifying the prediction of Equation (20) by multiplying the right side by correction factor μ_a. The corrected prediction is:

$$D_i^F \approx \mu_a \dot{M}(i) D_i^R$$

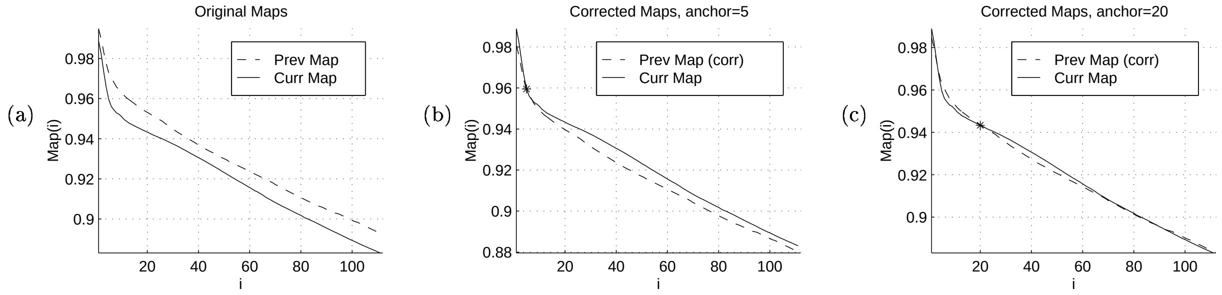
Evaluating the resulting approximation at the anchor point yields:

$$D_a^F \approx \mu_a \left(\dot{M}(i) D_i^R \right) \Big|_{i=a} = D_a^{PC}$$

The Full Pre-Code result is thus recovered. We further claim that the corrected prediction for the remaining points $\{D_i^F, a < i \leq N\}$ is expected to improve because frame-to-frame mapping inconsistencies due to earlier coded atoms have been corrected. In particular, the first few atoms are typically of high energy and the effect on distortion due to coding these atoms is difficult to predict. Correction factor μ is used to adjust the previous frame mapping in a way that overcomes these early inconsistencies. Effectively, the correction factor allows us to "attach" the previous frame mapping to the current frame mapping at the anchor point.

An illustration of this is shown in Figure 9. All three plots use our earlier Mother 24 kbit/s example, and the "current" frame refers to Frame-7. Figure 9a shows the original uncorrected

Fig. 9. An illustration of the hybrid model extension. Frame 7 of Mother at 24 kbit/s is predicted using the *Red*3 dictionary set. (a) Original mapping functions. (b) Previous mapping corrected with $a = 5$ full dictionary atoms. (c) Previous mapping corrected with $a = 20$. The (*) symbol denotes the anchor point in each case.



mappings. Figure 9b shows the effect of correcting the previous frame mapping using an anchor point of $a = 5$. Note that the corrected map matches the current frame map perfectly at the anchor point, and the maps are much closer together for the points beyond the anchor point $(i > 5)$. The consistency improvement is even greater for an anchor point of 20, as shown in Figure 9c. Since prediction accuracy relies on frame-to-frame mapping consistency, this implies that prediction of the distortion points for $(i > a)$ are improved.

C. Complexity vs. Accuracy

We now consider the complexity and accuracy tradeoffs which may be achieved by the Reduced Complexity Pre-Code and Hybrid Extension models. To this end, we code and compile prediction results using ten matching-pursuit coded frames from each of the fourteen standard MPEG-4 sequences. The frames are coded using the 2-pass quantizer design algorithm from [13]. For each coded frame, the distortion achieved at the actual stopping point N is predicted using both the Full Pre-Code model and the Reduced Complexity Pre-Code models with all four reduced dictionaries. Prediction accuracy is then evaluated in terms of relative error in predicted distortion:

$$e_{|D_N|} = \frac{|\hat{D}_N - D_N|}{|D_N|}$$

The achieved relative prediction errors are shown in Table V. The “FullMP” column shows the Full Pre-Code result, and the remaining columns show the Reduced Pre-Code results. As expected, Full Pre-Code achieves the smallest error, with an average of 0.14 percent. The *Red*0 model is not far behind, with an 0.47 percent average error. The remaining models show progressively poorer prediction with average relative errors of 1.4, 2.0 and 3.0 percent.

TABLE V

Average % relative prediction error.

Seq	Label	FullMP	Red0	Red1	Red2	Red3
1	Cont-10	0.0403	0.4859	0.7092	0.9184	1.0441
2	Hall-10	0.2218	1.1986	2.7779	3.7255	6.2328
3	Mom-10	0.0477	0.3492	1.3693	1.8743	2.4651
4	Cont-24	0.1513	0.2655	0.4345	0.5472	0.6689
5	Silent-24	0.1246	0.6333	2.0069	4.2683	7.3063
6	Mom-24	0.1390	0.3516	1.4572	1.8786	2.5570
7	Coast-48	0.2344	0.2237	0.9817	1.4397	1.3739
8	Fore-48	0.3651	0.5281	2.5813	4.2459	7.3988
9	News-48	0.0963	0.5363	1.7749	2.1981	2.3014
10	Coast-112	0.0941	0.3430	0.3107	0.4686	0.3719
11	Fore-112	0.0324	0.2855	0.4847	1.0391	2.0126
12	News-112	0.1016	0.4862	2.8977	3.7956	5.9767
13	Mob-1024	0.1052	0.3889	0.5989	0.6145	0.9835
14	Stef-1024	0.1913	0.5615	0.6435	1.0561	1.3849
	Average	0.1389	0.4741	1.3592	2.0050	3.0056

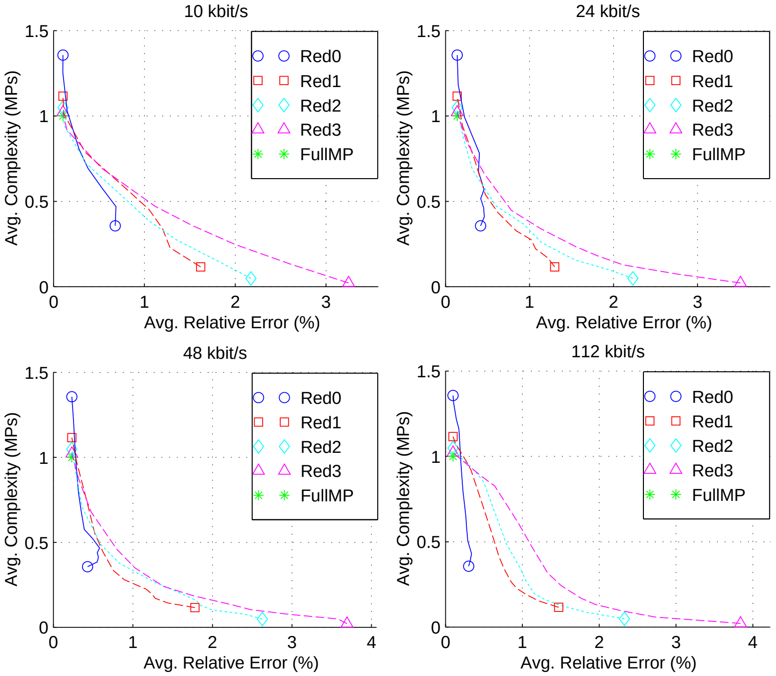
The complexity vs. accuracy tradeoffs are compared for all three model classes in Figure 10. In each plot, relative prediction error is shown on the horizontal axis and relative complexity is shown on the vertical axis. Complexity is measured relative to that of Full Pre-Code. Each plot shows results averaged across ten coded frames from each of the sequences in the given standard MPEG-4 bit rate category. For example, the upper left plot averages the results from the Container, Hall and Mother sequences at 10 kbit/s. The tradeoff provided by the Full Pre-Code model is represented by an asterisk. This point achieves high accuracy at a relative complexity cost of 1. The basic Reduced Complexity Pre-Code models are represented by four symbols spread along the bottom axis of each plot. In general these are both less complex and less accurate than the Full Pre-Code model.

The connecting curves shown in each plot represent the Hybrid Model tradeoffs as anchor point a varies from 0 to N. With $a = 0$, no full dictionary atoms have been added, and the Reduced Complexity models result. With $a = N$, the accuracy of the Full Pre-Code model is achieved, but with the added cost of the Reduced Complexity Pre-coding step. For intermediate anchor points, we see a smooth curve that traces out a continuum of complexity vs. accuracy points.

D. Reduced Complexity Rate Model

We now introduce a rate model compatible with the Reduced Pre-Code distortion model. Because the distortion model is based on known previous frame mapping $\dot{M}(i)$, we propose to base rate prediction on previous frame data as well. If the cumulative rate curve changes slowly from

Fig. 10. Complexity vs. Accuracy tradeoffs, averaged across all sequences in a bit rate category. Full Pre-Code method is (*). Other symbols show Reduced Pre-Code models. Connecting lines show the Hybrid Model tradeoffs achieved as additional full dictionary atoms are added.



frame to frame, then the current frame curve R_i may be predicted as

$$R_i \approx \dot{R}_i \tag{21}$$

where $\dot{R}_i$ is the cumulative rate due to the first i atoms of the previous frame.

Using Equation (21), we predict the rate for the same ten encoded frames per sequence as were used in the experiments of the previous subsection. Results are summarized in Table VI. Absolute errors are typically less than 2 or 3 atoms of bit budget, as can be seen by comparing the "Avg. Error" column to the "Bits/Atom" column. The relative error percentages are at most 2 percent, and are below 1 percent for half the tested sequences. We conclude that the use of previous frame data allows R_i for the current frame to be predicted using no additional computation. We now consider the complexity and accuracy of both models in an (R, D) optimization context.

V. Evaluation of models

A. *Model Complexity*

We illustrate the achievable complexity reduction using an optimization method due to Chen and Lin [4], which approximately solves the optimal bit allocation problem for an ATM network-based

TABLE VI

Results for the simple rate prediction model.

Seq	Label	Atoms/ Frame	Bits/ Atom	Avg. Error (bits)	Avg. % Rel-Error
1	Cont-10	53.89	20.63	10.23	0.92
2	Hall-10	40.78	22.64	16.56	1.67
3	Mom-10	41.22	22.96	15.09	1.60
4	Cont-24	117.89	18.63	29.58	1.35
5	Silent-24	75.00	21.80	31.28	2.05
6	Mom-24	99.89	20.38	28.64	1.32
7	Coast-48	201.33	19.94	39.42	0.98
8	Fore-48	150.67	20.35	58.82	1.98
9	News-48	207.11	21.64	39.17	0.89
10	Coast-112	158.44	23.74	26.82	0.71
11	Fore-112	83.33	25.56	30.74	1.69
12	News-112	293.44	20.68	58.76	0.96
13	Mob-1024	1612.78	18.36	65.08	0.22
14	Stefan-1024	1554.78	19.18	133.91	0.45
Average:			21.17	41.72	1.19

rate constraint. The solution fully expands all possible (R, D) points for a sliding window of three frames, which are numbered $\{k, k+1, k+2\}$. Frame k is coded with each of C candidate control parameters. This produces C candidate motion predictors for frame $k+1$, which are also coded using C candidate control parameters. This in turn produces C^2 candidate motion predictors for frame $k+2$, each of which is coded using all C possible control parameters. With all C^3 possibilities expanded, the algorithm selects the three control parameters which minimize D without violating the constraint on R.

Suppose this method were applied to matching pursuit using a fixed or 1-Pass quantizer [13]. In either case, quantizer $Q(\cdot)$ is known at the start of coding, and so rate control depends solely on stopping point N. For frame k, a single matching pursuit pass to some N_{max} atoms is required to generate the C possible motion predictors for frame $k+1$. This same pass is sufficient to trace out the C desired exact (R, D) points for frame k, and so model-based complexity reduction is not necessary for frame k. In a similar fashion, the C possible predictors for frame $k+1$ each require a single matching pursuit pass in order to generate the C^2 possible motion predictors for frame $k+1$. These passes trace out all desired exact (R, D) points for frame $k+1$, and so the models are again not needed. The situation differs for the coding of frame $k+2$. We call this a *terminal frame* with respect to the current window, since only the (R, D) points are needed, and no future motion predictors are required. For this reason, models may be used to compute (R, D) for frame $k+2$. Exact computation of the C^3 desired rate-distortion points requires C^2 complete matching pursuit passes. Suppose we instead use a Reduced Pre-Code model with a relative complexity of β. The modeled (R, D) complexity for frame $k+2$ is then βC^2 complete matching pursuit passes, with an additional complexity of βC passes required in order to compute $\dot{M}(i)$ for "previous frame" $k+1$. Summing the above expressions, exact (R, D) computation requires $1+C+C^2$ passes, while model-based computation requires $1+C+\beta(C+C^2)$.

Suppose instead that the 2-Pass adaptive quantizer is used [13]. The coding of frame k requires 1 unquantized analysis pass to some N_{max} atoms in order to design C adaptive quantizers. C quantized completion passes are then required to generate the C motion predictors for frame $k+1$. This total of $1+C$ required passes yields the exact (R, D) points for frame k, and so the models would yield no complexity advantage. Similarly, frame $k+1$ requires $C+C^2$ passes to compute the C^2 predictors for frame $k+2$, with the exact rate-distortion points for frame $k+1$ revealed in the process. Models are more useful for the terminal frame since no future motion predictors need be generated. One unquantized analysis pass is needed for each of the C^2 candidate motion residuals in order to design all C^3 possible quantizers. Exact (R, D) computation would then require C^3 completion passes. Instead, we could use the Full Pre-Code model to predict these C^3 rate-distortion points. The associated cost is negligible, since the C^2 analysis passes required for Full Pre-Code were completed during the quantizer design step. The total cost is thus $1+2C+2C^2+C^3$ passes for exact (R, D) farming, and $1+2C+2C^2$ passes for the model-based approach.

Table VII summarizes the exact and modeled (R, D) complexity costs for the optimization method of Chen and Lin [4] and for two additional optimization methods from literature [2][5]. Analysis for the two latter cases is given in [6], and is similar to the Chen and Lin derivation shown above. If we assume $C = 12$ control parameter choices per frame and use the β values from Table IV, we obtain the complexity reduction factors shown in Table VIII. The models are thus shown to provide complexity reduction for each combination of optimization method and quantizer design, with typical complexity reduction factors between two and twenty.

B. Model Accuracy

We now evaluate the accuracy of our distortion models. For a benchmark, we use the cubic spline interpolation method discussed in Lin and Ortega [5]. This model exactly computes the distortion at N_c control points σ_j, each of which represents a candidate rate. Interpolation is then used to predict distortion between these points. The method requires a minimum of four control

TABLE VII

Comparison of exact and modeled (R, D) generation complexity for three optimization methods. Each entry is the number of encoding passes per frame grouping required for (R, D) farming. Where applicable, C is the number of control parameters considered per frame, and β is the complexity reduction factor for the reduced dictionary set.

Optimization Method	Computation Method	Matching Pursuit Quantizer Design	
		Fixed or 1-Pass Adaptive	2-Pass Adaptive
Chen and Lin [4]	Exact	$1 + C + C^2$	$(1 + C)(1 + C + C^2)$
	Modeled	$1 + C + \beta(C + C^2)$	$1 + 2C + 2C^2$
Ramchandran, et.al. [2]	Exact	$2 + C + 3C^2 + 2C^3 + 2C^4$	$(1 + C)(2 + C + 3C^2 + 2C^3 + 2C^4)$
	Modeled	$2 + C + C^2 +$ $\beta(C + 3C^2 + 2C^3 + 2C^4)$	$2 + 3C +$ $4C^2 + 3C^3 + 2C^4$
Lin and Ortega [5]	Exact	16	108
	Modeled	$4 + 14\beta$	36

TABLE VIII

Model-based complexity reduction factors for the various combinations of optimization method and quantizer design. We assume $C = 12$ control parameters per frame.

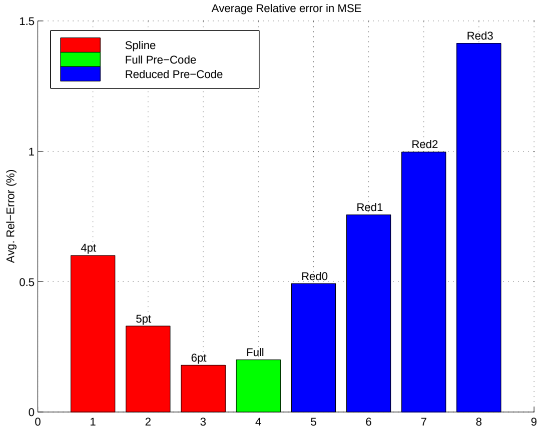
	Fixed or 1-Pass Adaptive				2-Pass Adaptive
Dependency Tree	(Red0)	(Red1)	(Red2)	(Red3)	(Full-PC)
Chen and Lin [4]	2.29	5.05	7.61	9.55	6.52
Ramchandran, et.al. [2]	2.77	8.37	19.05	39.24	12.48
Lin and Ortega [5]	1.78	2.84	3.41	3.71	3.00

points, but five or six are typically used [5].

There are several reasons we consider this distortion model to provide a suitable comparison. First the spline model was shown to be more accurate than three other models when used to predict distortion for the DCT case [5]. Second, the spline model has been used successfully within a gradient-based (R, D) optimization method [5][14], and so has accuracy sufficient for model-based optimization. Finally, the model is generic and easily extends to the matching pursuit case.

We now compare the accuracy of our distortion models to that of the cubic spline. Assume motion compensation has been performed on some frame k to produce residual $\vec{f_k}$. Distortion $D(\sigma)$ corresponding to C candidate values of rate σ are to be predicted using the models. We choose candidate rate points uniformly distributed between $\frac{1}{2}R_{tgt}$ and $3R_{tgt}$ for some target number of bits R_{tgt}. The combination $\{N, Q(\cdot)\}_\sigma$ which achieves target rate σ is derived using the 2-Pass quantizer design [13]. Each of the three models is then used to predict the distortion which would result from encoding $\vec{f_k}$ with each combination $\{N, Q(\cdot)\}_\sigma$. For the spline model, the accuracy depends on the number of control points, so we generate three separate results corresponding to $N_c \in \{4, 5, 6\}$. Control points σ_j are selected to be a subset of the C candidate rate points on which the models are tested. This gives the spline model an advantage, since N_c of the C distortion

Fig. 11. Distortion model accuracy. The plot shows relative prediction error for each model averaged across all tested frames.



points are known *exactly* to the model and thus are predicted without error.

Experimental results are generated using a single frame from each of the 14 MPEG-4 test sequences (see Part I [7], Table III-A). Each frame is coded to $C = 16$ candidate rate points distributed as above with R_{tgt} equal to the average bit rate per frame for the given test sequence. Results are shown in Figure 11. The Full Pre-Code model shows about the same relative error as the 6-point cubic spline. The most accurate reduced Pre-Code model, $Red0$, has accuracy somewhere between that of the 4 and 5-point spline models. The remaining reduced Pre-Code models are progressively less accurate. However, even the worst of the reduced complexity models has an average relative error below 1.5 percent.

We conclude Full Pre-Code and the $Red0$ and $Red1$ reduced Pre-Code models have comparable accuracy to the cubic spline, and thus are accurate enough to be considered for use in (R, D) based optimization methods. Results are not conclusive for the less expensive $Red2$ and $Red3$ models, which are slightly less accurate. However, one might wonder why we don't simply use the spline in place of our models, given the similar accuracy. The answer stems from complexity considerations, which we consider separately for each matching pursuit quantizer design. For Fixed or 1-Pass quantizer design, quantizer $Q(\cdot)$ is known at the start of residual coding. The spline model thus requires a single encoding pass in order to exactly compute the N_c control points on which the model is based. However, this same encoding pass traces out the the exact (R, D) tradeoffs for encoding the given frame, thus removing the need for a model [6]. For 2-Pass quantizer design, a single unquantized matching pursuit pass is required to define the quantizers for each rate to be considered. Each spline control point requires an additional completion pass with a rate-specific quantizer in order to compute the exact distortion points on which the model is based. The spline model thus requires a total of $(1 + N_c)$ full matching pursuit passes. However, the Full Pre-Code model obtains similar accuracy and is completely specified by the single analysis pass which was used to define the quantizers. Full Pre-Code is thus about $(1 + N_c)$ times less complex than the spline, and so is a better modeling choice.

VI. Conclusion

In this paper, we developed operational rate-distortion models for matching pursuit. Two model classes were shown. The first, Full Pre-Code, is an analytical model based on a full matching pursuit encoding pass. It models the effects of a changing quantizer. Our complexity analysis shows this model is especially useful in combination with 2-Pass adaptive quantization [13], in which the quantizer varies with the target rate. The second class, Reduced Complexity Pre-Code, is an empirical model based on a matching pursuit pass with a reduced complexity dictionary. Our analysis showed this model to be useful for fixed or 1-Pass quantization where the quantizer does not depend on the rate. A hybrid cross between the two model classes was shown to allow a continuous set of complexity vs. accuracy tradeoffs.

We evaluated the complexity and accuracy of our models in the context of model based (R, D) farming applications. Complexity analysis showed that significant model based complexity reductions could be achieved. Accuracy analysis showed both model classes to be comparable in accuracy to the cubic spline, which has been successfully used in a model-based (R, D) application. These results suggest that our models provide sufficient complexity reduction and accuracy to be used within similar optimization methods. Additional work is required to adapt one or more of the (R, D) optimization techniques to the matching pursuit case before the models may be usefully applied.

References

[1] Antonio Ortega and Kannan Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, November 1998.

[2] Kannan Ramchandran, Antonio Ortega, and Martin Vetterli, “Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders,” *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 533–545, September 1994.

[3] Chi-Yuan Hsu, Antonio Ortega, and Amy Reibman, “Joint selection of source and channel rate for VBR video transmission under ATM policing constraints,” *IEEE Journal on Selected Areas in Communication*, vol. 15, no. 6, pp. 1016–1027, August 1997.

[4] Jiann-Jone Chen and David Lin, "Optimal bit allocation for coding of video signals over ATM networks," *IEEE Journal on Selected Areas in Communication*, vol. 15, no. 6, pp. 1002–1015, August 1997.

[5] Liang-Jin Lin and Antonio Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 446–459, August 1998.

[6] Ralph Neff, *New Methods for Matching Pursuit Video Compression*, Ph.D. thesis, U. C. Berkeley, December 2000.

[7] Ralph Neff and Avideh Zakhor, "Matching pursuit video coding, Part I: Dictionary approximation," *IEEE Transactions on Circuits and Systems for Video Technology*, this issue.

[8] Ralph Neff and Avideh Zakhor, "Matching pursuit video coding at very low bit rates," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 158–171, February 1997.

[9] Wei Ding and Bede Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 1, pp. 12–20, February 1996.

[10] Hsueh-Ming Hang and Jiann-Jone Chen, "Source model for transform video coder and its application, Part I: Fundamental theory," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 287–298, April 1997.

[11] Vivek K. Goyal, Martin Vetterli, and Nguyen T. Thao, "Quantized overcomplete expansions in R^N: Analysis, synthesis, and algorithms," *IEEE Transactions on Information Theory*, pp. 16–31, January 1998.

[12] Ralph Neff, Avideh Zakhor, and Martin Vetterli, "Very low bit rate video coding using matching pursuits," in *Proceedings of SPIE Conference on Visual Communication and Image Processing*, September 1994, vol. 2308, pp. 47–60.

[13] R. Neff and A. Zakhor, "Modulus quantization for matching pursuit video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 895–912, September 2000.

[14] Liang-Jin Lin, Antonio Ortega, and C.-C. Jay Kuo, "Gradient-based buffer control technique for MPEG," in *Proceedings of SPIE Conference on Visual Communication and Image Processing*, May 1995, vol. 2501, pp. 1502–1513.