To be published in the Proceedings of VCIP'98

Multiple Reference Picture Video Coding Using Polynomial Motion Models

Thomas Wiegand, Eckehard Steinbach, Axel Stensrud, and Bernd Girod

Telecommunications Laboratory
University of Erlangen-Nuremberg
Cauerstr. 7/NT, D-91058 Erlangen, Germany

ABSTRACT

We present a new video coding scheme that uses several reference frames for improved motion-compensated prediction. The reference pictures are warped versions of the previously decoded frame applying polynomial motion compensation. In contrast to global motion compensation, where typically one motion model is transmitted, we show that in the general case more than one motion model is of benefit in terms of coding efficiency. In order to determine the multiple motion models we employ a robust clustering method based on the iterative application of the Least Median of Squares estimator. The approach is incorporated into an H.263-based video codec and embedded into a rate-constrained motion estimation and macroblock mode decision frame work. It is demonstrated that adaptive multiple reference picture coding in general improves rate-distortion performance. PSNR gains of 1.2 dB in comparison to the H.263 codec for the high global and local motion sequence *Stefan* and 1 dB for the sequence *Mobile & Calendar*, which contains no global motion, are reported. These PSNR gains correspond to bit-rate savings of 21 % and 30 % comparing to the H.263 codec, respectively. The average number of motion models selected by the encoder for our test sequences is between 1 and 7 depending on the actual bit-rate.

Keywords: Multiple Pictures, Robust Motion Clustering, Rate-Constrained Video Coding, Motion Estimation

1. INTRODUCTION

Utilizing inter-frame prediction for video compression leads to the question of the rate distortion efficiency of motion-compensated prediction (MCP). For a certain bit-rate required to transmit the motion-related information, MCP provides a version of the video signal with a certain distortion. The rate distortion trade-off can be controlled by various means. Our approach is to treat MCP as a special case of entropy-constrained vector quantization (ECVQ).[1] In case of translational motion-compensation, the image blocks to be encoded are quantized using individual code books that consist of image blocks of the same size in the previously decoded frame within the motion search range. A code book entry is addressed by the translational motion parameters which are entropy-encoded. The coding efficiency of MCP can be improved using several techniques including

1. multi-hypothesis MCP,
2. variable-shaped segment-based MCP,
3. complex motion models,
4. multiple reference picture MCP.

The success of item 1 is expressed in the use of overlapped block motion compensation[3] and B-frames.[4] Viewing MCP as a vector quantization (VQ) problem, multi-hypothesis MCP relates to predictive VQ.[5] However, because of the space and time variant statistics in image sequences, the various hypotheses have to be chosen adaptively and transmitted as side information limiting the efficiency of the approach.

Further author information —
E-mail: {wiegand|steinb|stensrud|girod}@nt.e-technik.uni-erlangen.de
URL: http://www-nt.e-technik.uni-erlangen.de/{wiegand|steinb|stensrud|girod}

Item 2 has mainly been emphasized in the context of variable block size (e.g., see[6,7]) and region-based MC (e.g., see[8,9]). Region-based MC can be viewed as an extension of variable block size MC. The usefulness of variable block size MC is well understood which is expressed for example by its incorporation into Annex F of the H.263 video coding standard.[10] In variable block size MC, the motion search in most cases consists of minimizing an affine tree-functional mapped on the block segmentation that can be done optimally by tree pruning.[11] In contrast, the difficulties in demonstrating the efficiency of region-based coding mainly relate to the mutual dependencies that are introduced by the selection of the regions, their contours, and coding parameters. In addition to that, the region-based coding approach very often exhibits a considerable amount of bits to describe the region contours questioning the entire approach. A compromise between variable block size and region-based coding can be found in.[8]

Highly connected to item 2 is item 3, the use of complex motion models such as affine (6 parameter) or bilinear (12 parameter) polynomial motion models. Large regions of images in video sequences are not likely to be motion-compensated by a simple displacement. Hence, more complex motion models should be utilized in the context of region-based MC[8] to reduce the number of regions and, thus, bit-rate. Interpreting polynomial MC in the context of VQ, the code book is generated by all possible motion parameters and an entry is addressed by choosing a particular motion parameter set. The estimation of multiple affine or bilinear motion models appears to be rather difficult. The problem can be viewed as polynomial regression in clustered subspaces which are mutually dependent. The task is complicated by the fact that it is unknown how many motion clusters are in the scene and which parts of the image belong to them. In addition, the ME becomes ill-conditioned in image areas with homogeneous intensities. In this work we will tackle the problem by a robust clustering method based on the iterative application of the Least Median of Squares estimator.

Item 4 refers to MC techniques where several reference pictures are employed which relates to increasing the code book size in VQ. In general, any technique that provides useful image data for MCP may be utilized to generate reference frames. These techniques may include "Global Motion Compensation" (GMC), "Dynamic Sprites",[12] "Background Memory" prediction,[13] "layers" from the layered coding scheme,[14] or video object planes as defined within MPEG-4.[15] The decoder just needs to be informed about parameters that are needed to generate the reference frames and be given a reference coordinate system to conduct the MC. Dynamic Sprites and GMC are used to improve prediction efficiency in case of camera motion by warping a motion-compensated version of the reference frame. However, for Dynamic Sprites, past frames are warped and blended into a Sprite memory. In contrast to GMC, where the global MC is applied using the previously decoded frame, Dynamic Sprite uses the Sprite memory buffer to provide the second reference frame. The motion models used are affine or bilinear. If the Sprite memory is equal to the frame size and the blending factor equals one, Dynamic Sprites and GMC are equivalent, thus Sprites are an extension to GMC. Dynamic Sprites or GMC are restricted to one polynomial model in most cases representing the dominant motion in the scene. In contrast, our approach extends GMC to several motion models.

Other approaches to multiple reference MC with more than one reference frame are "Short Term Frame Memory/Long Term Frame Memory" (STFM/LTFM) prediction.[16] As proposed in[16] the encoder is enabled to use two frame memories to improve prediction efficiency. The STFM stores the most recently decoded frame, while the LTFM stores a frame that has been decoded earlier. In[16] a refresh rule is specified that is based on a detection of scene change. An extension to[16] is presented in[17–19] where the use of several decoded frames that are collected in a long-term memory buffer is permitted for MC. The utilization of the long-term memory buffer provides improved prediction performance. Typically, the gains are bit-rate savings of 20-30 %.[17–19] However, the approach proposed in[17–19] requires the availability of previously decoded frames at encoder and decoder which may not always be the case. The results in[17–19] suggest that increasing the probability of finding a good match in our motion search space at reasonable costs can improve our overall video codec significantly. The costs, i.e., the motion-related bit-rate, can be controlled by imposing a rate constraint on the ME as it is done in our VQ analogy by ECVQ.

This paper is organized as follows. We first describe our approach to improve MCP in section 2. In section 3 we describe the quantization and robust estimation of the various polynomial motion models. The integration of our scheme into an H.263-based hybrid video codec is explained in section 4 where we also state an algorithm to iteratively select motion parameter sets subject to their usefulness in the multiple reference picture coder. Finally, experimental results are given in section 5.

2. MULTIPLE REFERENCE PICTURE WARPING USING POLYNOMIAL MOTION MODELS

Our approach is to generate multiple reference pictures simultaneously at encoder and decoder by warping the last decoded frame using polynomial motion parameters. The various motion parameter sets are transmitted as side information to the decoder. These reference frames are utilized by a video coder that performs translational block-based MC. In other words, for N motion parameter sets transmitted we utilize $M = N + 1$ reference frames that can be selected to independently predict each block by the encoder. Blocks in the reference frames are addressed by a combination of the code words for the spatial displacement and a frame selection parameter that has to be transmitted to the decoder as well. Hence, the transmission of several motion parameter sets and the frame selection parameters potentially increases the bit-rate. But, if the improvements obtained by finding a good match in our extended motion search range make up for the extra bit-rate, we gain coding efficiency of our video codec. The architecture of the motion-compensated predictor using multiple warped reference frames is shown in Fig. 1.

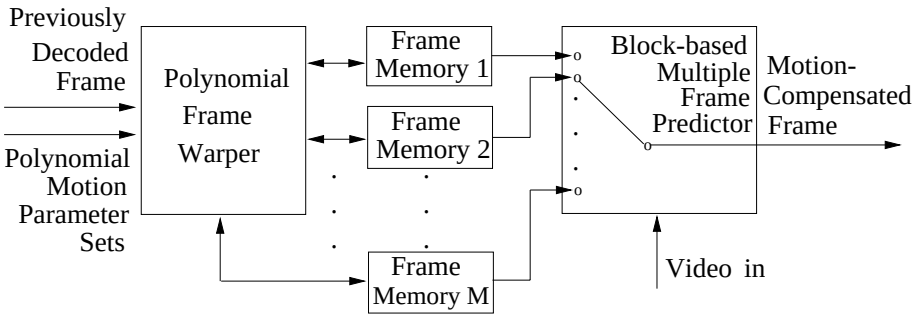

Figure 1. Motion-compensated predictor using M reference frames that are obtained by using the previously decoded frame and $N = M - 1$ warped frames given the corresponding polynomial motion parameter sets. The block-based multiple frame predictor can generate the motion-compensated frame by half-pel accurate motion compensation using one of the M reference frames.

Figures 2 and 3 show an example for multiple reference frame warping. The left hand frame (a) in Fig. 2 is the reference frame that is used to predict the right hand frame (b) in Fig. 2. Figure 3 shows four warped versions of the left hand frame (a) in Fig. 2. Hence, instead of just searching over the previously decoded frame (Fig. 2), the block-based motion estimator also searches positions in the warped frames in Fig. 3 and transmits the corresponding spatial displacement and frame selection parameter.

Relating our approach to region-based coding with polynomial motion models, we note that we also transmit various motion parameter sets and that the various "regions" associated with these motion parameter sets are indicated by the frame selection parameter. But, the "regions" in our scheme do not have to be connected. They are restricted to the granularity of the fixed or variable block-size segmentation of the block-based video codec. Furthermore, each block belonging to a "region" may have an individual spatial displacement vector. This is beneficial if the motion exhibited in the scene cannot be compensated by a few polynomial motion models. In addition, if the video scene does not lend itself to a description by various polynomial motion models, the coder drops into its fall-back mode which is block-based MC using the previously decoded frame only.

Comparing the method of warping reference pictures to using past decoded frames as reference pictures[17–19] we can draw the following conclusions. Both methods increase the probability of finding a good match for the block-based motion search at the cost of transmitting the reference frame index as side information. But, keeping the past decoded frames as reference pictures underlies the assumption that there are repetitions in the scene and that the frames are correlated. This is more a random approach to code book generation relating MCP to ECVQ compared to the idea of warping reference frames, where the motion parameter sets are estimated with respect to the frame or blocks that are to be predicted. However, the motion parameter sets have to be transmitted as side information limiting the number of reference frames. Similarly, the reference frames used when employing past decoded frames as reference pictures can be signaled to the decoder.

Nevertheless, the idea of multiple reference picture warping can be viewed as an extension to GMC, wherein also less dominant motion is captured by additional motion parameter sets to the global one.

(a) (b)

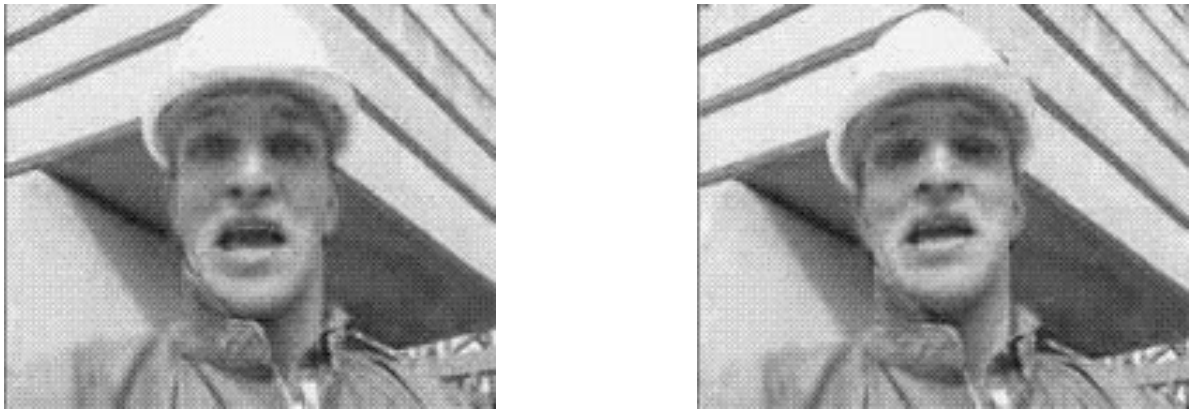

Figure 2. Frame (b) is to be predicted from frame (a).

Figure 3. Four warped reference frames generated by applying four distinct affine motion parameter sets to the decoded reference frame that is depicted in Fig. 2 (a).

3. MULTIPLE POLYNOMIAL MOTION PARAMETER ESTIMATION

The polynomial motion parameter sets have to be transmitted as side information. Hence, we have to quantize them. The polynomial motion model can be viewed as a transform, i.e., quantizing the motion parameter sets is equivalent to quantizing transform coefficients. In order to apply equal bit-allocation to all motion parameters, the motion model needs to be orthonormalized which is explained in section 3.1. The remainder of section 3 is devoted to the problem of polynomial motion clustering.

3.1. THE ORTHONORMALIZED POLYNOMIAL MOTION MODEL

The motion model employed in our investigation is an orthonormalized version of the well known affine (6 parameter) transformation model for approximating the pel-displacement field

$$\begin{aligned} d_x(\boldsymbol{a'}, x, y) &= c'_1 + c'_2 x + c'_3 y \\ d_y(\boldsymbol{a'}, x, y) &= c'_4 + c'_5 x + c'_6 y. \end{aligned} \tag{1}$$

Orthonormalization allows an independent and uniform quantization of the model coefficients. The relationship between the model coefficients and the displacement vector $(d_x(\boldsymbol{a}, x, y), d_y(\boldsymbol{a}, x, y))$ at an image point (x, y) in the current frame I_k is given as[8]

$$\begin{aligned} d_x(\boldsymbol{a}, x, y) &= c_1 f_1(x, y) + c_2 f_2(x, y) + c_3 f_3(x, y) \\ d_y(\boldsymbol{a}, x, y) &= c_4 f_1(x, y) + c_5 f_2(x, y) + c_6 f_3(x, y) \end{aligned} \tag{2}$$

with $\boldsymbol{a} = (c_1, c_2, c_3, c_4, c_5, c_6)^T$ being the model coefficient vector, and $f_i(x, y)$ the basis functions. Following the approach presented in,[8] the orthonormalized basis functions $f_i(x, y)$ can be derived as

$$\begin{aligned} f_1(x, y) &= \alpha_{00}\beta_{00} \\ f_2(x, y) &= \alpha_{00}(\beta_{10} + \beta_{11} y) \\ f_3(x, y) &= \alpha_{00}(\alpha_{10} + \alpha_{11} x) \end{aligned} \tag{3}$$

with

$$\alpha_{00} = \sqrt{\frac{1}{L_X + 1}}, \quad \alpha_{10} = \sqrt{\frac{3L_X}{(L_X + 1)(L_X + 2)}}, \quad \alpha_{11} = -2\sqrt{\frac{3}{L_X(L_X + 1)(L_X + 2)}}. \tag{4}$$

The β_{ij} are computed in the same manner replacing the image width L_X in (4) with the image height L_Y. The estimated coefficients which are typically in the range ± 20 are multiplied by 2 and rounded to the nearest integer, i.e., $Q(c_i) = ROUND(c_i \cdot 2)/2$.

Given a particular motion parameter set $\boldsymbol{a}_n$ or $Q(\boldsymbol{a}_n)$, the corresponding reference picture in the multiple frame buffer $\hat{I}_k^m(x, y)$ with $m = n + 1$ is computed as follows

$$\hat{I}_{k-1}^m(x, y) = I'_{k-1}(x + d_x(\boldsymbol{a}_n, x, y), y + d_y(\boldsymbol{a}_n, x, y)) \tag{5}$$

with I'_{k-1} being the previously decoded frame. Note that the previously decoded frame $I'_{k-1}(x, y)$ can be viewed as compensated by a motion parameter set with zero coefficients, i.e., $I'_{k-1}(x, y) = \hat{I}_{k-1}^1(x, y)$.

Finally, the reference frame warping, i.e., the realization of Eq. (5) is computed using the cubic spline interpolation[8]

$$f(x) = \begin{cases} -0.5|x|^3 + 2.5|x|^2 - 4.0|x| + 2.0 & 0 \le |x| < 1 \\ 1.5|x|^3 - 2.5|x|^2 + 1.0 & 1 \le |x| < 2 \\ 0 & 2 \le |x| \end{cases} \tag{6}$$

which we tested against simpler versions like bi-linear or nearest-neighbor interpolation. It turned out that the cubic spline method yields higher coding gain than the other, simpler methods.

3.2. ROBUST MOTION CLUSTERING

The basic idea for motion clustering employed in this work is to use a Least Median of Squares (LMedS) estimator[20] to determine a dominant motion model from a dense set of displacement vectors, then to remove the corresponding image points from the estimation process and to repeat the dominant motion estimation on the remaining pixels. This procedure continues until the maximum number of motion models have been tested for their coding efficiency.

In order to derive the cost function that has to be minimized we define the residual for an image point (x, y) in the current frame as

$$r(\boldsymbol{a}, x, y) = I_k(x, y) - I'_{k-1}(x + d_x(\boldsymbol{a}, x, y), y + d_y(\boldsymbol{a}, x, y)). \tag{7}$$

The LMedS estimator minimizes the following cost function in order to obtain a dominant motion model $\hat{\boldsymbol{a}}$

$$\hat{\boldsymbol{a}} = \underset{\boldsymbol{a} \in \mathcal{A}}{\operatorname{argmin}} \left(\operatorname{med}_{\forall (x,y)} \quad r^2(\boldsymbol{a}, x, y) \right). \tag{8}$$

The estimator yields the smallest median of the squared residuals for image points (x, y) under consideration while varying over all coefficient sets in $\mathcal{A}$. The value of the median does not change even if half of the displacements are outliers. Outliers are image points where the displacement does not fit to the estimated dominant motion model.

In order to keep the computational complexity reasonable, we solve the minimization problem (8) following a Monte-Carlo sampling technique described in,[21] wherein a probabilistic method is presented to obtain the coefficient sets $\mathcal{A}$ over which we evaluate the LMedS estimator in (8). We choose a selected number O (typically $O = 100$) of randomly picked K-tuples of image points (x, y) in $I_k(x, y)$ and compute the corresponding displacement vectors $(\Delta_x(x, y), \Delta_y(x, y))$ with respect to image $I_{k-1}(x, y)$. In case of our polynomial motion model with 6 degrees of freedom, the K-tuple is required to be of minimum size 3. For each K-tuple we solve the set of linear equations in (2) leading to an estimate for the parameter vector $\boldsymbol{a}$. We quantize the motion parameter vector $\boldsymbol{a}$. We then compute the corresponding residuals evaluating (7) for all image points and determine the median. The K-tuple leading to the smallest median is considered to be the solution $\hat{\boldsymbol{a}}$. Note that an exhaustive search would require the evaluation of $\binom{O}{K}$ combinations, involving high computational complexity.

4. INTEGRATION INTO AN H.263-BASED VIDEO CODEC

Our motivation for integrating multiple reference picture video coding into an H.263-based video system is twofold: (*i*) the algorithm is well defined,[10] (*ii*) the test model of the H.263 standard, TMN-2.0, can be used as reference for comparison.[22] The H.263 inter-prediction modes INTER and UNCODED* are extended to multiple frame MC. Both modes are assigned one code word representing the frame selection parameter for the entire macroblock (MB).

To run our H.263 as well as our multiple reference frame coder, we have modified the encoding strategy as utilized by the TMN-2.0 coder. Our encoding strategy differs for the ME and the mode decision, where our scheme is motivated by rate-distortion theory. The problem of optimum bit allocation to the motion vectors and the residual coding in any hybrid video coder is a non-separable problem requiring a high amount of computation. To circumvent this joint optimization, we split the problem into two parts: motion estimation and mode decision.

Since there are now two Lagrangian cost functions to be minimized, we employ two different Lagrange multipliers: one for the motion search (λ_{motion}), the other one for the mode decision (λ_{mode}). Furthermore, the distortion measures differ in order to keep our comparison to the TMN-2.0 fair. Hence, the selection of the Lagrange parameters remains rather difficult in our coder. In this work, we employ the heuristic $\lambda_{motion} = \sqrt{\lambda_{mode}}$, which appears to be sufficient. The parameter λ_{mode} itself is derived from the rate distortion curve that we computed using the TMN-2.0 H.263 coder. Note that there are various ways to obtain the desired Lagrange parameter and more sophisticated ones than ours especially when applying Lagrangian bit allocation in a practical video coder. However, we have chosen this approach due to its simplicity and its reproducibility.

*We call the INTER mode the UNCODED mode, when the COD bit indicates copying the macroblock from the reference frame without residual coding.[10]

4.1. RATE-CONSTRAINED MOTION ESTIMATION

The motion estimation is conducted as follows. For each frame the "best" motion vector is found by full search on integer-pel positions followed by half-pel refinement. The integer-pel search is conducted over the range $[-15 \ldots 15] \times [-15 \ldots 15]$ pels. Since the motion vectors constituting the spatial displacement vectors and the reference frame selection parameter have to be transmitted as side information requiring additional bit-rate the "best" motion vector is defined as the one which minimizes the Lagrangian cost function

$$J(\boldsymbol{v}) = D(\boldsymbol{v}) + \lambda_{motion} R(\boldsymbol{v} - \boldsymbol{p}), \tag{9}$$

where $D(\boldsymbol{v})$ is a distortion measure for a given motion vector $\boldsymbol{v} = (v_x, v_y, v_f)$, such as the SSD or the sum of the absolute differences (SAD) between the displaced frame from the multiple reference frame buffer and the original, and $R(\boldsymbol{v} - \boldsymbol{p})$ is the bit-rate associated with a particular choice of the spatial displacement and time delay given its prediction $\boldsymbol{p} = (p_x, p_y, p_f)$. In our current implementation, the predictor $\boldsymbol{p}$ is computed using the H.263 median methodology[10] without considering that a motion vector containing the spatial displacement vector possibly points into a different frame than the adjacent spatial displacement vectors. The reference frame selector is transmitted without predicting it, i.e., $p_f = 0$. For the reference frame selection parameter v_f we have generated a Huffman code table.

4.2. RATE-CONSTRAINED MODE DECISION IN MULTIPLE REFERENCE PICTURE CODING

The idea of rate-constrained mode decision has been published in.[23] In contrast to the work presented in[23] we consider all MBs as coded independently, i.e., the current MB is coded given the past MBs. The extension of one to multiple picture MC with M frames can be viewed as multiplying the number of MB modes by M. Adopting the framework developed in[23] we can state the rate-constrained mode decision problem as follows.

Consider an image given by its partition into MBs $\mathcal{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_S)$. For our video coder operating on M reference frames, each MB in $\mathcal{X}$ can be coded using only one of $2M + 1$ possible modes given by the set $\mathcal{M} = \{I, P_1, \ldots, P_M, U_1, \ldots, U_M\}$, where the I relates to the INTRA mode, the subset $\{P_1, \ldots, P_M\}$ corresponds to the M possible INTER modes and the subset $\{U_1, \ldots, U_M\}$ to the UNCODED modes. The notation P_m or U_m states that the respective INTER or UNCODED mode are computed with their motion vector pointing into frame m. Let $H_s \in \mathcal{M}$ be the mode selected to code MB $\mathbf{X}_s$. Then, the modes assigned to the elements in $\mathcal{X}$ are given by the S-tuple, $\mathcal{H} = (H_1, \ldots, H_S) \in \mathcal{M}^S$. The problem of finding the combination of modes that minimizes the distortion satisfying a given rate constraint R_c can be formulated as an unconstrained minimization problem[24,25]

$$\min_{\mathcal{H}} \sum_{s=1}^{S} J(\mathbf{X}_s, \mathcal{H}), \tag{10}$$

where $J(\mathbf{X}_s, \mathcal{H})$ is the Lagrangian cost function for MB $\mathbf{X}_s$ and is given by

$$J(\mathbf{X}_s, \mathcal{H}) = D(\mathbf{X}_s, \mathcal{H}) + \lambda_{mode} \cdot R(\mathbf{X}_s, \mathcal{H}). \tag{11}$$

In our actual video coder, the rate distortion cost of a particular MB $\mathbf{X}_s$ depends only on the set of the current and past MB modes $\mathcal{H}_s$ simplifying the optimization procedure. Nevertheless, we restrict the optimization so that both the rate and distortion for a given MB are impacted only by the content of the current MB and its respective operational mode. As a result, the rate and distortion associated with each MB can be computed given the operational modes of all previously coded MBs. In this case, the optimization problem of (10) reduces to

$$\min_{\mathcal{H}} \sum_{s=1}^{S} J(\mathbf{X}_s, \mathcal{H}) \;=\; \sum_{s=1}^{S} \min_{H_s | \mathcal{H}_{s-1}} J(\mathbf{X}_s, \mathcal{H}_s) = \sum_{s=1}^{S} J^*(\mathbf{X}_s, \mathcal{H}_s), \tag{12}$$

and, as a result, can be easily minimized by independently selecting the best mode for each MB.

4.3. GREEDY SELECTION OF MOTION PARAMETER SETS

Since the motion parameter sets have to be transmitted as side information, we propose a greedy algorithm that iteratively warps references frames and accepts or rejects them based on their usefulness in terms of coding efficiency. For that we are defining the following subsets $\mathcal{M}_m = \{I, P_1, \dots P_m, U_1 \dots U_m\} \subseteq \mathcal{M} \quad \forall m : 1 \leq m \leq M$. Furthermore, the overall Lagrangian cost for encoding the entire frame given m reference frames is defined as

$$L_m = \sum_{s=1}^{S} \min_{H_s \in \mathcal{M}_m | \mathcal{H}_{s-1} \in \mathcal{M}_m^{s-1}} J(\mathbf{X}_s, \mathcal{H}_s) = \sum_{s=1}^{S} J^*(\mathbf{X}_s, \mathcal{H}_s \in \mathcal{M}_m^s). \tag{13}$$

The iterative algorithm for greedy motion parameter set selection reads as follows

Step 1 Set the number of frames counter $m = 1$ and the model estimation trial counter $n = 0$. Encode frame I_k using the H.263 encoding scheme. Prediction of MBs is performed using the previously decoded frame only. Compute overall costs L_1.

Step 2 Compute the array of all minimum MB costs $\mathcal{J}\{\mathcal{X}, \mathcal{H} \in \mathcal{M}_m^S) = \{J^*(X_1, \mathcal{H}_1 \in \mathcal{M}_m) \dots J^*(X_S, \mathcal{H}_S \in \mathcal{M}_m)\}$.

Step 3 Sort the array $\mathcal{J}\{\mathcal{X}, \mathcal{H} \in \mathcal{M}_m^S)$ by increasing values and save the result in $\mathcal{J}^{sorted} = \{J_1^{sorted} \dots J_S^{sorted}\}$. Set $m \to m + 1$.

Step 4 Set $n \to n + 1$. Estimate a dominant polynomial motion model $\hat{\boldsymbol{a}}_n$ as described in section 3 for all MBs $\mathbf{X_s} \in \mathcal{X} : J^*(\mathbf{X_s}, \mathcal{H}_s \in \mathcal{M}_{m-1}^s) > J_t^{sorted}$, with J_t^{sorted} being the t'th element in $\mathcal{J}^{sorted}$. The index t is computed as $t = 0.5 \cdot S \cdot (1 + n/N_{max})$ and N_{max} is the maximum number of motion models.

Step 5 Warp the reference frame $\hat{I}_k^m$ using Eq. (5) and $\hat{\boldsymbol{a}}_n$.

Step 6 Encode frame I_k using the multiple reference frame coder with reference pictures I'_{k-1}, $\hat{I}_{k-1}^2$, ..., $\hat{I}_{k-1}^m$, i.e., determine L_m using Eq. 13.

Step 7 If $L_m < L_{m-1}$, accept the motion parameter set $\hat{\boldsymbol{a}}_n$ and go to **step 2**, else reject it and go to **step 4**.

The algorithm stops, if n exceeds the maximum number of motion models N_{max}. In the first step the video coder runs in H.263-mode which is also the fall-back mode of our scheme. The minimum cost that the H.263 coder assigned to all MBs are stored in an array $\mathcal{J}\{\mathcal{X}, \mathcal{H} \in \mathcal{M}_m^S)$ in step 2. In step 3, we compute a sorted array of the costs that the H.263 coder generates which we call $\mathcal{J}^{sorted}$. Setting $m \to m+1$ means that we now are going to add another frame to our multiple reference picture codec, which is in the first iteration $m = 2$. In step 4, we increment the the model trial counter n which is in the first iteration $n = 1$, i.e., here we are computing the first model to warp reference frame $\hat{I}_{k-1}^m$, with $m = 2$ here. Now, using the pixels in all labeled MBs $\mathbf{X_s} \in \mathcal{X} : J^*(\mathbf{X_s}, \mathcal{H}_s \in \mathcal{M}_{m-1}^s) > J_t^{sorted}$ we compute the dominant motion cluster labeled with $\hat{\boldsymbol{a}}_n$ using the LMedS algorithm in step 4. The idea here is that we try to generate a reference frame for MBs which could not be motion-compensated sufficiently using the previously decoded frame only and therefore show large costs. Note that the decision which MBs are labeled depends on the index t which itself is a function of n, our model trial counter. Having computed the warped reference frame in step 5, the multiple reference picture coder is run generating costs L_m in step 6. The decision whether or not to accept a particular motion parameter set is done in step 7. In case the model is accepted the algorithm moves on to generate the next reference frame by jumping to back to step 2. Otherwise we jump back to step 4, where the model trial counter n is incremented and with that the MB labeling index t, leading to a smaller amount of pixels for the LMedS motion clustering. The reason is that we assume a distinct motion cluster to the previously found ones to be more probable at image content that is related to high cost MBs.

Following our paradigm of independent optimization, a fast method for computing the cost when adding an m'th reference frame can be derived. Defining the subset $\tilde{\mathcal{M}}_m$ by $\tilde{\mathcal{M}}_m = \mathcal{M}_m \setminus \mathcal{M}_{m-1} = \{P_m, U_m\}$, the Lagrangian when adding the m'th reference frame can be approximated as

$$L_m \cong \sum_{s=1}^{S} \min\{J^*(\mathbf{X}_s, \{\mathcal{H}_s \in \mathcal{M}_m^{s-1}, H_s \in \mathcal{M}_{m-1}\}), \min_{\tilde{\mathcal{M}}_m} J(\mathbf{X}_s, H_s)\}, \tag{14}$$

that is given the costs of all MBs for which we can choose among modes $\mathcal{M}_{m-1}$, we compute the cost for that MB when coding with modes $\tilde{\mathcal{M}}_m$ and select the minimum. This procedure avoids motion estimation in frames $1 \cdots m-1$ when adding frame m and could be incorporated into the iterative algorithm instead of evaluating Eq. 13 in step 6.

5. EXPERIMENTAL RESULTS

The multiple reference picture approach is integrated into an H.263-based codec as described in the previous section. For the experimental results presented in this section, the modified H.263 codec is run in baseline mode plus the unrestricted motion vector mode turned on. To make the comparisons more meaningful, we have included the results produced by the TMN-2.0 codec[22] which runs with the same settings as our codecs. We select the QCIF sequences *Mobile & Calendar, News, Foreman* and *Stefan*. While the latter two sequences contain global motion as well as local motion, the first two sequences contain no global motion. We encoded the first 100 frames of the sequences at 7.5 Hz varying the DCT quantizer over values $8, 10, 15, 20$. We generated bit-streams that are decodeable producing the same PSNR values as at the encoder. The data on the first INTRA frame are excluded from the results.

Figure 4 shows results derived from runs for the sequences *Mobile & Calendar, News, Foreman* and *Stefan*. The Figs. depict the average PSNR from reconstructed frames produced by the TMN-2.0 codec ($\times$), our rate distortion optimized H.263 codec (H.263+RDO, $+$) and the multiple reference picture video codec (MRPV $\circ, *$). The two runs for the MRPV ($\circ, *$) differ in that for the curve marked with $\circ$ the maximum number of motion models to be transmitted is $N_{max} = 1$ (MRPV_1) whereas the other curve marked with $*$ relates to runs where the maximum number of motion models is set to $N_{max} = 9$ (MRPV_9). Since the iterative model selection algorithm is employed, the multiple reference picture coder can transmit between 0 and N_{max} motion models. Fig. 4 illustrates the following points:

- The rate-constrained video codecs perform always than the TMN-2.0.
- Comparing the H.263+RDO coder and the MRPV coders, multiple reference picture coding when running it adaptively as proposed here always improves rate-distortion performance as well.
- When comparing our scheme (MRPV_9) to the other codecs, we are achieving gains mainly at higher bit-rates.
- Improvements comparing MRPV_1 to MRPV_9 are mainly observed for sequences with no global motion.
- The bit-rate savings obtained by the MRPV_9 codec are up to 20 % against the MRPV_1 and 30 % against the H.263+RDO codec.

In Fig. 5, we depict the average number of motion models chosen by the MRPV_9 codec. There is certainly a dependency on the overall bit-rate. However, as bit-rate gets large, which is the case for sequences like *Mobile & Calendar* and *Stefan*, the ratio of the rate for the affine motion parameter sets to the overall bit-rate gets smaller and with that the dependency of the number of motion parameter sets from the bit-rate decreases. On average, an affine motion parameter set needs about 30 bits to be transmitted. Based on these results we can underline our claim that more than one warped reference frame and with that several motion models are beneficial in terms of coding efficiency.

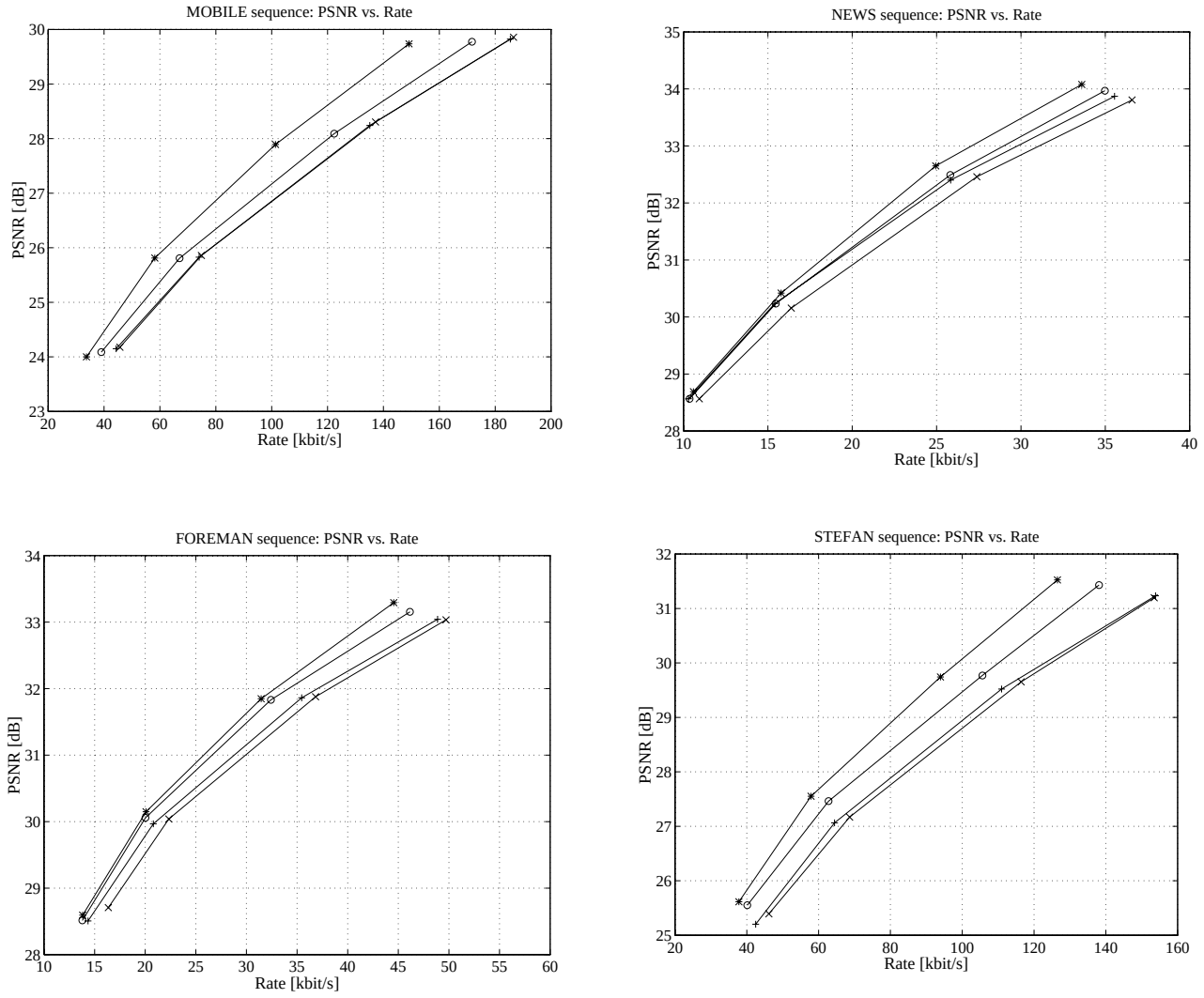

Figure 4. PSNR vs. overall bit-rate for the sequences *Mobile & Calendar, News, Foreman* and *Stefan*. The TMN-2.0 is marked by "×", the H.263+RDO codec by "+" and the multiple reference picture video codecs are indicated by "∘ and "∗" relating to MRPC_1 and MRPV_9, respectively.

6. CONCLUSIONS AND FUTURE WORK

In this paper we demonstrate that the usage of more than one global motion model can be beneficial for the rate-distortion performance of block-based video coding. We use orthonormalized affine polynomial motion models to warp additional reference frames. Since typically more than one motion cluster is present in the frame or the global motion cannot be described by one affine model we iteratively employ the LMedS estimator to successively estimate the different motion parameters in order of their dominance in the scene. The encoder decides in the rate-distortion sense if a particular model is of benefit or not. For the proposed scheme, we observed PSNR improvements up to 1.2 dB PSNR in comparison to the H.263 codec for the high motion sequence *Stefan* and 1 dB PSNR for the sequence *Mobile & Calendar* which contains no global motion. These PSNR gains correspond to bit-rate savings up to 21 % and 30 % comparing against the H.263 codec, respectively. These bit-rate savings can only be partially obtained by permitting only one motion model to be transmitted. In that case the bit-rate savings are 12 % and 7 % when comparing to the H.263 codec. This underlines our claim that transmitting several motion models is beneficial in terms of coding efficiency. Hence, when permitting up to 9 motion models to be sent, the average number of motion models selected by the encoder for our four test sequences is between 1.5 and 6.6 depending on the actual bit-rate.

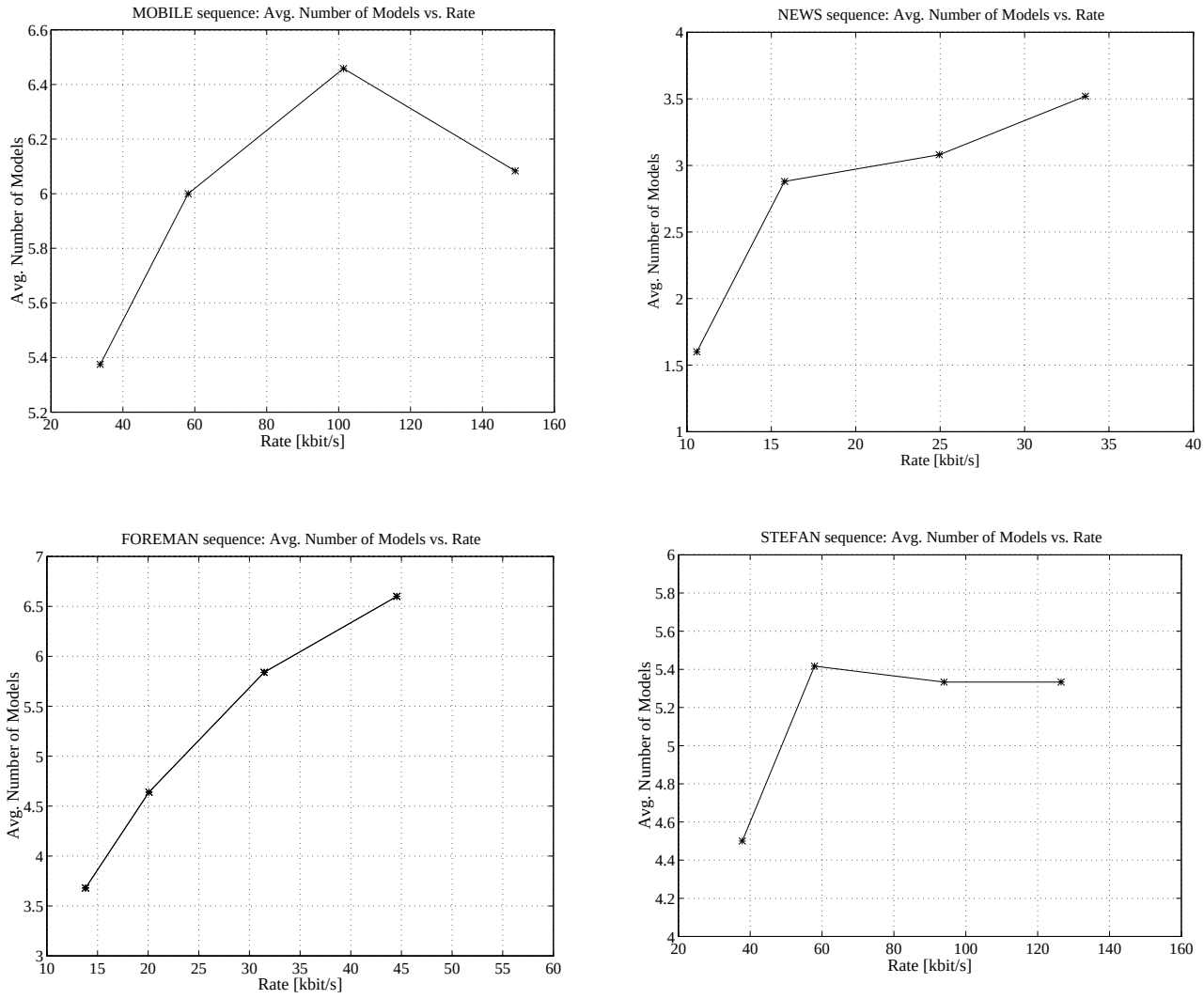

Figure 5. Number of motion models vs. overall bit-rate for the sequences *Mobile & Calendar, News, Foreman* and *Stefan* produced by the MRPV_9 codec.

Several remaining work items are still ahead. Regarding the current implementation, the following items need to be pursued. First of all, the spatial displacement vector prediction needs to be coupled with the reference frame selection. Note that in our current implementation, the predictor for the spatial displacement vector is computed without incorporating that a motion vector containing the spatial displacement vector possibly points into a different reference frame than the adjacent spatial displacement vectors. Second, the motion model estimation needs a speed-up to make it more amenable for practical video coding. A different motion clustering method than the LMedS-based may be considered. Third, another speed-up is possible by incorporating Eq. (14) instead of (13) into the iterative algorithm of section 4.3. Regarding the conception of our idea, we mainly consider an extension of the approach presented here with respect to the approach of long-term memory prediction presented in[17–19] to be promising. The straight forward extension is to use decoded frame in the long-term past for reference picture warping.

REFERENCES

1. P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-Constrained Vector Quantization," *IEEE Transactions on Acoustics, Speech and Signal Processing* **37**, pp. 31–42, Jan. 1989.
2. B. Girod, "Motion-Compensating Prediction with Fractional-Pel Accuracy," *IEEE Transactions on Communications* **41**, pp. 604–612, Apr. 1993.
3. M. T. Orchard and G. J. Sullivan, "Overlapped Block Motion Compensation: An Estimation-Theoretic Approach," *IEEE Transactions on Image Processing* **3**, pp. 693–699, Sept. 1994.
4. B. Girod, "Efficiency Analysis of Multi-Hypothesis Motion-Compensated Prediction for Video Coding." Submitted for publication., 1997.
5. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston,Dordrecht,London, 1992.
6. G. J. Sullivan and R. L. Baker, "Efficient Quadtree Coding of Images and Video," *IEEE Transactions on Image Processing* **3**, pp. 327–331, May 1994.
7. J. Lee, "Optimal Quadtree for Variable Block Size Motion Estimation," in *Proceedings of the IEEE International Conference on Image Processing*, vol. II, pp. 480–483, (Washington, D.C., USA), Oct. 1995.
8. ISO/IEC JTC1/SC29/WG11 MPEG96/M0904, "Nokia research center: Proposal for efficient coding." Submitted to Video Subgroup, July 1996.
9. K. W. Stuhlmüller, A. Salai, and B. Girod, "Rate-Constrained Contour-Representation for Region-Based Motion Compensation," in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, (Orlando, USA), Mar. 1996.
10. ITU-T Recommendation H.263, "Video Coding for Low Bitrate Communication." Draft, Dec. 1995.
11. P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal Pruning with Applications to Tree-Structured Source Coding and Modeling," *IEEE Transactions on Information Theory* **35**, pp. 299–315, Mar. 1989.
12. F. Dufaux and F. Moscheni, "Background Mosaicking for Low Bit Rate Video Coding," in *Proceedings of the IEEE International Conference on Image Processing*, (Lausanne, Switzerland), Sept. 1996.
13. D. Hepper, "Efficiency Analysis and Application of Uncovered Background Prediction in a Low Bit Rate Image Coder," *IEEE Transactions on Communications* **38**, pp. 1578–1584, Sept. 1990.
14. J. Y. A. Wang and E. H. Adelson, "Representing Moving Images with Layers," *IEEE Transactions on Image Processing* **3**, pp. 625–638, Sept. 1994.
15. ISO-IEC/JTC1/SC29/WG11, "MPEG-4 Video Verification Model." Draft, July 1997.
16. ISO/IEC JTC1/SC29/WG11 MPEG96/M0654, "Core Experiment of Video Coding with Block-Partitioning and Adaptive Selection of Two Frame Memories (STFM / LTFM)." Dec. 1996.
17. T. Wiegand, X. Zhang, and B. Girod, "Motion-Compensating Long-Term Memory Prediction," in *Proceedings of the IEEE International Conference on Image Processing*, (Santa Barbara, USA), Oct. 1997.
18. T. Wiegand, X. Zhang, and B. Girod, "Block-Based Hybrid Video Coding Using Motion-Compensated Long-Term Memory Prediction," in *Proceedings of the Picture Coding Symposium*, (Berlin, Germany), Sept. 1997.
19. T. Wiegand, X. Zhang, and B. Girod, "Long-Term Memory Motion-Compensated Prediction." Submitted for publication, `http://www-nt.e-technik.uni-erlangen.de/~wiegand/trcsvt98.ps.gz`, 1997.
20. P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*, John Wiley, New York, 1987.
21. S. Chaudhuri, S. Sharma, and S. Chatterjee, "Recursive Estimation of Motion Parameters," *Computer Vision and Image Understanding* , Nov. 1996.
22. T. Research, "TMN (h.263) Encoder/Decoder, Version 2.0." Download via anonymous ftp to `bonde.nta.no`, June 1997.
23. T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-Distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard," *IEEE Transactions on Circuits and Systems for Video Technology* **6**, pp. 182–190, Apr. 1996.
24. H. Everett III, "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources," *Operations Research* **11**, pp. 399–417, 1963.
25. Y. Shoham and A. Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers," *IEEE Transactions on Acoustics, Speech and Signal Processing* **36**, pp. 1445–1453, Sept. 1988.