

Subblock Matching-Based Conditional Motion Estimation With Automatic Threshold Selection for Video Compression

Gagan Bihari Rath and Anamitra Makur, *Senior Member, IEEE*

Abstract—In this paper, we propose a block-based conditional motion estimation scheme for video compression. The blocks in a frame are classified as active or inactive based on a threshold and only the active blocks are subjected to motion estimation. The threshold value is decided automatically through an iterative algorithm. The motion vectors of the boundary active blocks are estimated using a subblock matching-based scheme. Unlike existing variable size block-matching algorithms, the proposed motion estimation scheme preserves the basic framework of the conventional block-matching algorithm such as equal number of motion vectors and one-pass processing. A small amount of side information needs to be transmitted for the boundary blocks to indicate the displacements of their subblocks. The proposed scheme provides a better tradeoff between the complexity and the rate-distortion performance of the encoder than other existing schemes and better subjective quality of the motion-compensated frames than the conventional motion compensation.

Index Terms—Block-matching algorithm, motion estimation.

I. INTRODUCTION

THE KEY to effective video compression is the exploitation of its inherent temporal correlation, which spurred motion-compensated prediction as the primary video compression technique. Among various motion-estimation techniques, block-matching algorithms [1], [2] are the most popular. In these algorithms, each frame is partitioned into blocks of pixels and then the motion of each target block is estimated as a vector indicating the displacement of its best match in the previous frame. The best matching blocks are used to form the prediction for the present frame. The prediction error frame, called the displaced frame difference and the set of motion vectors, called the motion field, are transmitted to the receiver.

Though the conventional block-matching algorithm is simple and robust, it has many drawbacks. It does not consider the background and the foreground statistics in a frame. It estimates the motion vectors of all the blocks even if some blocks belong to the background region. As a result, the motion-estimation time is considerably high. Further, background blocks may

have noisy motion vectors which give rise to more side information and an inaccurate motion field. Second, in the conventional block-matching algorithm, all the pixels in a block are assumed to have the same displacement. As a result, the blocks at the boundaries of moving objects, which contain parts of both the stationary background and the moving objects, get inaccurately compensated and produce blocking artifacts when the displaced frame difference is lossy-coded.

Block matching is the most time consuming part in the encoding process. This has resulted in a number of fast block-matching algorithms that search only at a few positions inside the search area [1]–[5], or use a subset of the pixels in a block for computing the matching criterion [6], or use a simpler matching criterion than the minimum mean-squared error (MMSE) [4], [7]–[9] producing suboptimal (in prediction error) matches. Despite fast algorithms, the computation needed is enormous because of the huge data in a video and remains a bottleneck in realizing real-time applications.

In this paper, in Section II we propose a conditional motion-estimation scheme where only selected blocks are subjected to motion estimation. The blocks are selected based on a threshold, whose value is decided automatically through an iterative approach unlike the existing thresholding schemes. In Section III, the boundary blocks (BBs) are divided into subblocks and the motion vectors of those blocks are decided based on the subblock matching. Unlike the variable-size block-matching algorithms [10]–[13], the proposed method preserves the basic framework of the conventional block-matching algorithm such as an equal number of motion vectors and one-pass processing. A small amount of side information needs to be transmitted for the BBs to indicate the displacements of the subblocks. We discuss two approaches: subblock motion estimation (SME) and subblock matching-based motion estimation (SMME). For each approach, we discuss two algorithms. In Section IV, we discuss the prediction error and the side information rate of these algorithms and explain the threshold selection. Section V deals with the automatic selection of the threshold value using an iterative algorithm. We present simulation results in Section VI, and finally, draw conclusions in Section VII.

II. CONDITIONAL MOTION ESTIMATION

In the conventional block-matching motion estimation, for a given matching criterion, the number of search locations determines the required computation. A larger number of search

Manuscript received October 18, 2001; revised May 16, 2003. This paper was recommended by Associate Editor S. Chen.

G. B. Rath was with the Electrical Communication Engineering Department, Indian Institute of Science, Bangalore, India. He is now with the TEMICS Group, IRISA, Rennes, France.

A. Makur was with the Electrical Communication Engineering Department, Indian Institute of Science, Bangalore, India. He is now with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

Digital Object Identifier 10.1109/TCSVT.2003.816512

locations help in finding a better match, but requires more computation. Thus there is a tradeoff between the complexity and the coding efficiency of motion estimation.

The conventional block-matching algorithm estimates the motion of each target block. For a stationary camera, the intensities over the stationary regions of a video frame remain the same (except for small changes due to noise). Consequently, most stationary target blocks produce null motion vectors and stationary target blocks with nonnull motion vectors produce very small prediction error improvement over that with the null vectors. If we restrict the motion estimation only to moving/changing target blocks, one benefit will be the reduction in computation proportional to the fraction of stationary target blocks. In addition, it will eliminate the noisy motion vectors in the stationary regions, thus producing a motion field which is closer to the true motion. Moreover, the fast search techniques [1]–[6] can still be applied to the moving target blocks.

Keeping with the objective of complexity reduction, the procedure of classifying the target blocks as moving or stationary should be computationally simple. Puri *et al.* [13] use a motion detector with two thresholds. If the gray value difference between a target pixel and the pixel at the same location in the previous frame is greater than a threshold T_g , then the pixel is *moving*, else *stationary*. If the number of *moving* pixels in a target block is less than another threshold T_p , then the block is considered *stationary*, else it is considered *moving*. They choose T_g arbitrarily. In this paper, we use the same measure of change, but propose a scheme to automatically select the T_g for each frame in such a way that, besides the computation reduction, better rate-distortion performance is also achieved.

Instead of *moving* and *stationary*, we use the terms *active* and *inactive* for the block classification. This terminology is motivated by a better rate-distortion performance requirement rather than just finding the moving and the nonmoving target blocks. Active target blocks have large intensity changes and are subjected to motion estimation. Inactive target blocks have small intensity changes and do not undergo motion estimation (null motion is assumed).

Let $I^n(\mathbf{r})$ denote the intensity of the pixel at position \mathbf{r} in frame n . Let I_b^n denote the binary image obtained by thresholding the difference of frames n and $n-1$. The intensity value at position \mathbf{r} is determined as follows:

$$I_b^n(\mathbf{r}) = 1, \quad \text{if } |I^n(\mathbf{r}) - I^{n-1}(\mathbf{r})| \geq T_g \quad (1)$$

$$= 0, \quad \text{otherwise} \quad (2)$$

where T_g^n denotes the threshold for frame n . The image I_b^n is subjected to filtering for eliminating the small regions of ones and zeros. In our simulation, we use a simple counting method to eliminate the small regions. Let \bar{I}_b^n denote the filtered binary image. The pixel at position \mathbf{r} is termed *active* if $\bar{I}_b^n(\mathbf{r}) = 1$. Let b_{ij}^n denote the (i, j) th target block in frame n . Let B_{ij} denote the set of pixel coordinates in the block b_{ij}^n . The block b_{ij}^n is classified as *active* if

$$\sum_{\mathbf{r} \in B_{ij}} \bar{I}_b^n(\mathbf{r}) \geq T_p^n \quad (3)$$

where T_p^n denotes the threshold for the number of *active* pixels in a target block in frame n . In this paper, we have assumed that T_p^n is a constant, though in general, it can vary from frame to frame. Though T_g^n can have any real value, assuming an 8-bit grayscale image, we choose only the integral values lying in the range $0 \leq T_g^n \leq 256$. Because we always refer to the present frame, from here onwards, we will drop the superscript n and refer to the thresholds as T_g and T_p .

Note that, in the presence of the camera motion, almost all the target blocks will be active and the conditional motion estimation may not give any significant computation reduction. So the camera motion should be cancelled using an existing global motion compensation technique [14], [15] before applying the conditional motion estimation. To limit the scope of this paper, we have assumed that the camera is stationary.

In the proposed scheme, the block size, search window, search scheme, and the matching criterion (assumed to be the mean-squared error (MSE)) are the same as in the conventional motion estimation. The prediction error of each target block, active or inactive, is transmitted to the decoder. In practice, the prediction error is lossy-coded and the motion field is lossless-coded. For some given coding schemes, the optimal threshold is the one that gives the best rate-distortion performance. But finding the optimal threshold requires the decoder output fed back to the encoder and needs enormous computation, which makes it impractical for both hardware as well as software implementations [16]. Therefore, in the rate-distortion optimized motion compensation [12], [16], the prediction error energy and the motion field bit rate are used as the performance measures. In our approach, since we do not encode the prediction error, we use the same performances measures. Instead of actual rate-distortion optimization, we take the performance of the conventional motion compensation (CMC) as the baseline and choose the threshold to produce comparable or lower prediction error energy and side information rate. If both lower prediction error and lower rate than CMC is achieved, a better rate-distortion performance is claimed.

It is obvious that the motion field rate (with variable-length coding) with the conditional motion estimation is equal to or less than that with CMC, but its prediction error is equal to or more than that with CMC. In order to achieve both lower prediction error and lower rate, we now propose splitting some of the target blocks into subblocks and estimating the subblock motion vectors.

III. MOTION ESTIMATION WITH BLOCK SPLITTING

In CMC, each target block has only one motion vector with the assumption that all its pixels have equal displacements. As a result, blocks containing parts of moving object and stationary background, blocks containing parts of multiple moving objects and blocks containing objects having motion other than translation (for example, rotation, depth change, deformation, etc.), produce large prediction errors. The prediction error may be decreased by reducing the block size, but that will increase the number of motion vectors and the side information. Of late, there has been considerable research in optimally allocating bits to achieve the best tradeoff between the prediction error

and the side information [16], [17]. These algorithms require huge computation which makes them unsuitable for real-time implementations.

A simple approach to obtain a better tradeoff than CMC is to adaptively decide the target block size [10]–[13] so that the assumption of equal displacement for all the pixels in a block becomes more valid. The blocks at the moving object boundaries, or the blocks containing objects having nontranslational motion, are made very small, whereas the stationary blocks or blocks with translational motion are made relatively large. Chan *et al.* [10] show that a variable block size achieves higher SNR and lower side information than the fixed block size CMC. They use a bin-tree decomposition of the frame where a block at a leaf node is split into two subblocks if its prediction error is above a preset threshold. Sullivan and Baker [12] similarly perform a quad-tree decomposition of the frame using a rate-distortion criterion.

The disadvantage of variable block size motion estimation is its high computational load, two to three times as much as fixed block size [10]. The computation can be reduced by combining variable block size motion estimation with conditional motion estimation, where only selected blocks should be split further based upon some statistics. Puri *et al.* [13] classify the moving blocks as compensable or uncompensable after the first round of motion estimation and then split the uncompensable blocks into four subblocks which undergo a second round of motion estimation. The classification is done using the same motion detector which classifies the blocks as moving or stationary. Since the threshold is selected arbitrarily, the performance and the computation depends on the chosen threshold value.

In this paper, we propose two motion-estimation schemes based on block splitting. In scheme 1, the subblock motion vectors are estimated from the motion vectors of their parent blocks in the same neighborhood. In scheme 2, the motion vectors of the parent blocks are estimated using subblock matching. Both schemes are simple and preserve the block-based framework of CMC. These two schemes are modified slightly to give two more schemes: scheme 1a and scheme 2a. Scheme 2a is the proposed scheme. Scheme 1 and scheme 1a are similar to Puri *et al.*'s scheme and are presented for the sake of comparison.

A. SME

In this method, first the motion vectors of the active blocks are estimated using the conditional motion estimation. Let N denote the block size. Each block is then split into four $N/2 \times N/2$ subblocks. Let the four subblocks of a block b_{ij}^n be denoted by $b_{ij;kl}^n$, $k, l = 0, 1$. Let $B_{ij;kl}$ denote the set of pixel coordinates in the subblock $b_{ij;kl}^n$. Let \mathbf{v}_{ij}^n and $\mathbf{v}_{ij;kl}^n$ denote the motion vectors of the block b_{ij}^n and the subblock $b_{ij;kl}^n$, respectively.

Let $V_{ij;kl}^n$, the set of neighborhood motion vectors for the subblock $b_{ij;kl}^n$, be

$$V_{ij;kl}^n \equiv \left\{ \mathbf{v}_{ij}^n, \mathbf{v}_{(i-1+2k)j}^n, \mathbf{v}_{i(j-1+2l)}^n, \mathbf{v}_{(i-1+2k)(j-1+2l)}^n, \mathbf{0} \right\}$$

that is, it consists of the motion vector of its parent block, the motion vectors of the three adjacent blocks, and the null vector.

For example, the set of neighborhood motion vectors of the subblock $b_{ij;00}^n$ is given as

$$V_{ij;00}^n \equiv \left\{ \mathbf{v}_{ij}^n, \mathbf{v}_{(i-1)j}^n, \mathbf{v}_{i(j-1)}^n, \mathbf{v}_{(i-1)(j-1)}^n, \mathbf{0} \right\}.$$

The motion vector of the subblock $b_{ij;kl}^n$ then becomes

$$\mathbf{v}_{ij;kl}^n = \arg \min_{\mathbf{u} \in V_{ij;kl}^n} f_{ij;kl}^n(\mathbf{u}) \quad (4)$$

where $f_{ij;kl}^n(\mathbf{u})$ denotes the mse of subblock $b_{ij;kl}^n$ for displacement \mathbf{u} and is defined as

$$f_{ij;kl}^n(\mathbf{u}) = \frac{4}{N^2} \sum_{\mathbf{r} \in B_{ij;kl}} (I^n(\mathbf{r}) - I^{n-1}(\mathbf{r} + \mathbf{u}))^2. \quad (5)$$

Accordingly, the prediction for the block b_{ij}^n is given as

$$\hat{I}^n(\mathbf{r}) = I^{n-1}(\mathbf{r} + \mathbf{v}_{ij;kl}^n); \forall \mathbf{r} \in B_{ij;kl}; k, l = 0, 1. \quad (6)$$

The transmitted side information consists of the motion vectors \mathbf{v}_{ij}^n and the extra information to identify the motion vectors of the subblocks.

The inclusion of the null vector in $V_{ij;kl}^n$ is to make it nonincreasing (in the sense of set inclusion) with the increase in the threshold value (this will be explained in the next section). For a given subblock, if any of the four block motion vectors is null, the inclusion of the null vector is redundant. On the contrary, if the motion vectors of the parent block and three adjacent blocks are all nonnull, it is very likely that all the four blocks belong to one moving object (assuming that the block size is small). Therefore, it is very unlikely that the subblock will have less distortion with the null vector than with any of those four nonnull vectors. Therefore, the performance will not be severely affected if the null vector is excluded from the neighborhood set in practical implementations.

It is obvious that, if all four subblock motion vectors in a block are same as the parent block motion vector, then the prediction error of SME is equal to that of CMC. Otherwise, the prediction error is less. For subblock $b_{ij;kl}^n$, if $V_{ij;kl}^n$ is a singleton set, i.e., the five candidate vectors are equal (hence, null), then there is no need to estimate subblock motion vector. If a block and its eight neighbor blocks all have null motion vectors (such as, all are inactive), then it does not undergo SME. Therefore, the extra computation in SME depends on the number of subblocks with nonsingleton neighborhood motion vector sets and the cardinality of those sets. The extra side information that needs to be transmitted for each subblock $b_{ij;kl}^n$ depends on the cardinality of $V_{ij;kl}^n$ and ranges from 0 to 3 bits (cardinality 1 to 5) assuming fixed-length coding. Though this method is equivalent to a variable block size motion-estimation scheme, no side information is needed for encoding the tree. The motion field, which is known to the decoder, determines which subblock requires how many bits. We refer to this method as scheme 1.

Note that if a parent block has nonnull motion vector, all its subblocks undergo motion estimation since the cardinalities of $V_{ij;kl}^n$ are at least 2. Also, if the null vector is excluded from $V_{ij;kl}^n$, then at most 2 bits of side information per subblock is needed and $V_{ij;kl}^n$ can be a singleton set even for a nonnull parent motion vector.

Scheme 1, which considers all the blocks for SME, can be improved further. Motion vectors of adjacent blocks located completely inside a moving object are very similar. The prediction error reduction may not be significant for the subblocks inside these blocks. Therefore, we restrict SME to the BBs where the CMC prediction error is large and its reduction due to SME can be significant. If the set of motion vectors of a block and its eight neighbors contain both null and nonnull vector, then it is a BB. Only the subblocks of BBs having nonsingleton $V_{ij;kl}^n$ undergo motion estimation. No extra information is needed for encoding the tree since the BBs may be determined at the decoder from the motion field. We refer to this variation as scheme 1a. The difference between scheme 1a and Puri *et al.*'s scheme is that scheme 1a finds the subblock motion vectors only in the BBs from the neighborhood sets, whereas Puri *et al.*'s scheme finds the subblock motion vectors only in the uncompensable blocks through unrestricted motion estimation.

B. SMME

Both scheme 1 and scheme 1a are two-pass approaches: first the motion vectors of the active blocks are estimated, then the motion vectors of the subblocks are selected from those motion vectors. So some extra computation is needed in the second pass. Also, the motion vectors obtained by CMC do not truly represent the actual motion, which is more true for BBs. Since the subblock motion vectors are selected from a set of motion vectors estimated using CMC (thus, may be erroneous), the reduction in the prediction error may not be comparable to that with the unrestricted SME. These two drawbacks are now improved upon by incorporating the subblock matching in the first pass of the motion estimation itself.

In CMC with MSE as the matching criterion, the motion vector \mathbf{v}_{ij}^n of a block b_{ij}^n is determined as

$$\mathbf{v}_{ij}^n = \arg \min_{\mathbf{u} \in S} \frac{1}{N^2} \sum_{\mathbf{r} \in B_{ij}} (I^n(\mathbf{r}) - I^{n-1}(\mathbf{r} + \mathbf{u}))^2 \quad (7)$$

where S denotes the search window. Accordingly, the motion-compensated prediction for the block b_{ij}^n becomes

$$\hat{I}^n(\mathbf{r}) = I^{n-1}(\mathbf{r} + \mathbf{v}_{ij}^n) \quad \forall \mathbf{r} \in B_{ij}. \quad (8)$$

In the proposed SMME, \mathbf{v}_{ij}^n is estimated as

$$\mathbf{v}_{ij}^n = \arg \min_{\mathbf{u} \in S} \sum_{k=0}^1 \sum_{l=0}^1 (\min \{f_{ij;kl}^n(\mathbf{u}), f_{ij;kl}^n(\mathbf{0})\}) \quad (9)$$

where $f_{ij;kl}^n(\mathbf{u})$ is defined in (5). Accordingly, the motion-compensated prediction for b_{ij}^n is constructed as

$$\hat{I}^n(\mathbf{r}) = I^{n-1}(\mathbf{r}), \text{ if } f_{ij;kl}^n(\mathbf{0}) \leq f_{ij;kl}^n(\mathbf{v}_{ij}^n) \quad (10)$$

$$= I^{n-1}(\mathbf{r} + \mathbf{v}_{ij}^n), \text{ otherwise;} \quad (11)$$

$$\forall \mathbf{r} \in B_{ij;kl}; k, l = 0, 1.$$

This modified motion estimation is applied to all the active blocks.

It is obvious that the prediction error of an active block with SMME is always less than or equal to the same with CMC. In

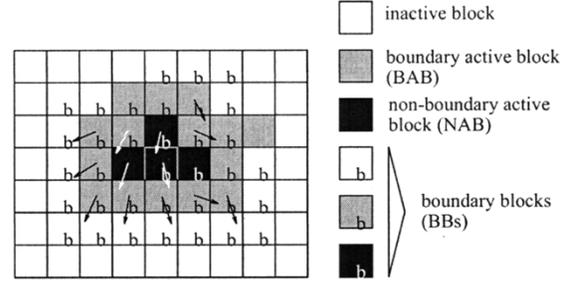


Fig. 1. Example of BBs, BABs, and NABs. Blocks with arrows have nonnull motion vectors and blocks without arrows have null motion vectors.

CMC, all the subblocks in a parent block have the parent block's motion vector. In SMME, the motion vector of a subblock is either the parent block's motion vector or the null vector. If the parent motion vector is null (active block may have null motion vector), then all four subblock motion vectors are null and no extra side information is needed. If the parent motion vector is nonnull, the decoder needs few extra bits of side information to construct the motion-compensated frame. 1 bit is used if all the subblocks have the parent motion vector and 5 bits otherwise (1 bit to denote diverse motion field, followed by 1 bit per subblock indicating if it has the parent motion vector or the null vector). Alternatively, four extra bits in all cases may be used. SMME costs four extra comparison operations per (active) block with the same number of additions, subtractions and multiplications as compared to CMC. We refer to this method as scheme 2.

Since a subblock can have either the parent block's motion vector or the null vector, for better rate-distortion performance, it is preferable to apply SMME to the BBs. But the BBs cannot be determined prior to the motion estimation. Therefore, we propose to apply SMME only to the boundary active blocks (BABs) (which are known prior to the motion estimation). A BAB is defined as an active block with at least one inactive neighbor block. All remaining active blocks [nonboundary active blocks (NABs)] undergo the conventional motion estimation. Since inactive blocks do not undergo conditional motion estimation, SMME is not applied to the boundary inactive blocks (inactive blocks with at least one active neighbor block). We refer to this as scheme 2a.

Note the difference between a BB and a BAB. A BAB with a nonnull motion vector is a BB. A BAB with a null motion vector is a BB or not depending on the neighborhood motion vectors. On the contrary, if a BB has a nonnull motion vector, it is active (BAB or NAB). If it has a null vector, it can be either active (BAB or NAB) or inactive. Refer to the example in Fig. 1.

In this scheme, the prediction errors of the BABs are less than or equal to that of CMC. The NABs have the same prediction error as that of CMC. If a BAB has a null motion vector, then no extra side information is transmitted. If it has a nonnull motion vector, then 1 or 5 extra bits (like scheme 2) are transmitted. Unlike scheme 2, some extra bits may have to be transmitted for the NABs too. This is because the decoder does not know which blocks are active or inactive, so it cannot determine BABs. However, from the decoded motion field, it knows which blocks are BBs. So it assumes that SMME has been applied only to the BBs with nonnull motion vectors. Therefore, if a NAB having

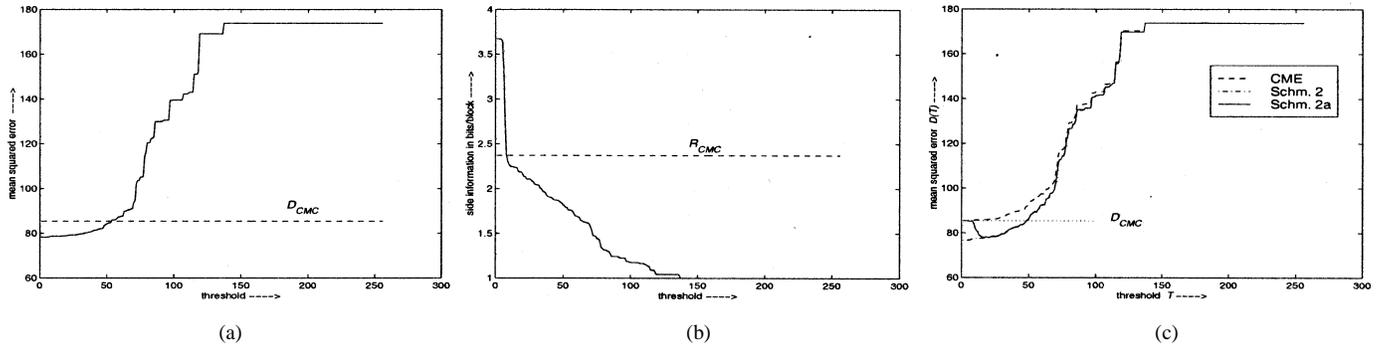


Fig. 2. (a) $D(T)$ with scheme 1; (b) $R(T)$ with scheme 1; (c) $D(T)$ with scheme 2a; for frame #1 of the Tennis sequence.

a nonnull motion vector turns out to be a BB after the motion estimation (because at least one of its neighbor blocks has the null motion vector), then 1 extra bit needs to be transmitted (indicating that all the subblocks have the parent block's motion vector). Therefore, the extra information for an active block is decided only after the motion estimation.

Both scheme 2 and scheme 2a can be generalized to include the motion vectors of the past neighbor blocks in the motion estimation. To keep the side information low and the motion estimation simple, we do not consider such generalizations in this paper.

Both scheme 1 and scheme 2 are also conceptually equivalent to the forward segmentation based BMA [18]. Instead of segmenting an active block into regions representing parts of the moving object, we segment them into subblocks. The advantages are very less computation and very low side information. Though the subblocks can be further divided into sub-subblocks, to keep the side information low, we stop the segmentation process after the second level.

To summarize, scheme 1 comprises CMC applied to all the active blocks in the first pass and SME applied to all the blocks in the second pass. Scheme 1a comprises CMC applied to all the active blocks in the first pass and SME applied to only the BBs in the second pass. Scheme 2 comprises SMME applied to all the active blocks. Scheme 2a comprises SMME applied to only the BABs and CMC applied to the NABs. Scheme 2a is the proposed scheme. In all the four schemes, the side information consists of the motion vectors of the blocks and some extra bits to identify the motion vectors of some of the subblocks.

IV. THRESHOLD SELECTION

The drawback of choosing the threshold T_g arbitrarily, as was done by Puri *et al.* [13], is the following. If T_g is small, the number of active blocks is very high, which results in little computational improvement over CMC. In the worst case, when $T_g = 0$, all the blocks are active and the computation needed is more than CMC because of the added computation for classification. If T_g is large, the number of active blocks is too small to obtain the benefits of motion compensation. In the worst case when $T_g = 256$, all the blocks are inactive (no motion estimation at all) and the prediction error is the difference between the present frame and the previous frame. Therefore, T_g should be chosen to not only reduce computation, but also produce comparable rate-distortion performance to that with CMC.

Let us assume that a null motion vector is coded using 1 bit and a nonnull vector is coded using ≥ 2 bits (since the percentage of the null vectors in a scene without the camera motion is very high). The nonnull vectors are either fixed-length or variable-length coded. For notational convenience, let us use T for T_g . For a given frame, let D_{CMC} and R_{CMC} denote the prediction error energy and the side information rate obtained using CMC and let $D(T)$ and $R(T)$ denote the same quantities obtained using any one of the four schemes mentioned at threshold T .

Consider scheme 1. If we assume that in CMC there is at least one block with a nonnull motion vector, then $D_{CMC} < D(256)$ and $R_{CMC} > R(256)$. This is because when $T = 256$, all the blocks are inactive, so $R(256) = 1$ bit per block and $D(256)$ is equal to the frame difference energy. If even one block has a nonnull motion vector, its prediction error energy is less than the block difference energy.

If we assume that at $T = 0$, there is at least one subblock motion vector not equal to its parent motion vector, $D(0) < D_{CMC}$ and $R(0) > R_{CMC}$.

For the considered frame, let $V_{ij;kl}(T)$ denote the set $V_{ij;kl}$ at threshold T . It is easy to see that $V_{ij;kl}(T_1) \supseteq V_{ij;kl}(T_2) \forall i, j, k, l$ when $T_2 > T_1$. This is because, at threshold T_2 , the nonnull elements of $V_{ij;kl}(T_1)$ remain the same or become null. Therefore, $D(T_1) \leq D(T_2)$ and $R(T_1) \geq R(T_2)$ when $T_2 > T_1$.

Since $D(0) < D_{CMC} < D(256)$ and $D(T)$ is nondecreasing in T , there exists a maximum value T_D , $0 \leq T_D < 256$, such that $D(T) < D_{CMC}$ for $T \leq T_D$. Similarly, since $R(0) > R_{CMC} > R(256)$ and $R(T)$ is nonincreasing in T , there exists a minimum value T_R , $0 < T_R \leq 256$, such that $R(T) < R_{CMC}$ for $T \geq T_R$. Note that both $D(T)$ and $R(T)$ are bounded functions of T . Depending on the relative values of T_D and T_R , we obtain the following two cases:

Case 1 : $T_R \leq T_D$

$$D(T) < D_{CMC}, R(T) \geq R_{CMC}; \text{ for } 0 \leq T < T_R$$

$$D(T) < D_{CMC}, R(T) < R_{CMC}; \text{ for } T_R \leq T \leq T_D$$

$$D(T) \geq D_{CMC}, R(T) < R_{CMC}; \text{ for } T_D < T \leq 256$$

Case 2 : $T_R > T_D$

$$D(T) < D_{CMC}, R(T) \geq R_{CMC}; \text{ for } 0 \leq T \leq T_D$$

$$D(T) \geq D_{CMC}, R(T) \geq R_{CMC}; \text{ for } T_D < T < T_R$$

$$D(T) \geq D_{CMC}, R(T) < R_{CMC}; \text{ for } T_R \leq T \leq 256.$$

Thus, when $T_R \leq T_D$, there exists an interval $[T_R, T_D]$, in which both the prediction error energy and the side information rate for scheme 1 are less than the corresponding quantities for CMC. Therefore, select T such that $T_R \leq T \leq T_D$. A T value closer to T_D saves more side information, whereas a value closer to T_R reduces more prediction error energy. When $T_R > T_D$, choose T to be either less than T_D or greater than T_R . Fig. 2(a) and (b) display the $D(T)$ and $R(T)$ plots for frame #1 of the Tennis sequence. As we can see, $D(T)$ is nondecreasing, $R(T)$ is nonincreasing and $T_R < T_D$ for this frame.

Now consider scheme 1a. Let $D_1(T)$ and $R_1(T)$ denote the prediction error and the side information obtained using scheme 1 and let $D(T)$ and $R(T)$ denote the same quantities obtained using scheme 1a, at threshold T . Consider also the conditional motion estimation (CME) without SME (i.e., no second-pass motion estimation, the conventional motion estimation is applied to only the active blocks in the first pass). Let $D_2(T)$ and $R_2(T)$ denote the above quantities obtained using CME. It is easy to see that

$$D_1(T) \leq D(T) \leq D_2(T), R_1(T) \geq R(T) \geq R_2(T). \quad (12)$$

Because the SME is applied only to the BBs, $D(T)$ and $R(T)$ are not necessarily monotonic. Let $A(T)$ denote the set of active blocks at threshold T . It is easy to see that $A(T)$ is non-increasing (in the sense of set inclusion) with T . Therefore, $D_2(T)$ is nondecreasing and $R_2(T)$ is nonincreasing. We have already seen that $D_1(T)$ is nondecreasing and $R_1(T)$ is non-increasing. Therefore, because of the relationship in (12), the general behaviors of $D(T)$ and $R(T)$ will be nonincreasing and nondecreasing respectively. Since $D(T)$ and $R(T)$ are not strictly monotonic, they can intersect the lines $D = D_{\text{CMC}}$ and $R = R_{\text{CMC}}$, respectively, at multiple intervals. In general, we would like to have at least one interval $[T_R, T_D]$ in which both the prediction error energy and the side information rate are lower than the corresponding quantities with the CMC. Then T should be chosen in that interval. If such an interval does not exist, then like scheme 1, we should choose T to get improvement in at least one of the quantities.

$D(T)$ and $R(T)$ for scheme 2 are nondecreasing and nonincreasing like those of scheme 1. The shape of $D(T)$ for scheme 2a will be explained in the next section. Like other schemes, $R(T)$ for scheme 2a behaves opposite to its $D(T)$.

V. AUTOMATIC THRESHOLD SELECTION

T_R and T_D will vary from frame to frame, since $D(T)$, $R(T)$, D_{CMC} and R_{CMC} depend on the motion and the gray scale statistics in the scene. Therefore, for better performance over the entire scene, we use a variable threshold. Out of all the six possible ranges of T in the two cases mentioned in the previous section, only $T_D < T < T_R$ is unwanted. This choice can be avoided if $T \leq T_D$ or if $T \geq T_R$. Since T_R depends on the motion vector coding scheme, we select $0 < T \leq T_D$. If T_R happens to be less than T , then we save in both the prediction error and the side information; else, we save in the prediction error but pay some extra side information.

The previous section's method of choosing the threshold value requires that D_{CMC} and R_{CMC} should be found and

that T should be varied over the range $[0, 256]$ to determine T_D and T_R . Since we want to reduce computation, D_{CMC} and R_{CMC} can not be found because CMC, or scheme 1 or 1a first pass at $T = 0$ (all blocks are active), should not be performed. Therefore, T_D and T_R cannot be determined. Instead, we propose the following.

A. Optimal Threshold Selection Criterion

We assume that predominant changes occur due to object motion (covered and uncovered background regions and parts of moving objects where there is large gray scale gradient). The noise is assumed zero mean with small variance compared to the predominant changes. The total area of predominant changes is assumed to be small compared to the frame size. The block size is small enough such that objects span multiple blocks and a BB contains parts of only one moving object and the stationary background.

Consider scheme 2a. At $T = 0$, there are no BABs (since there are no inactive blocks). Therefore $D(0) = D_{\text{CMC}}$. Because of the assumption stated in the previous section, $D(256) > D_{\text{CMC}}$. Hence, if there exists a T such that $D(T) < D_{\text{CMC}}$, then $D(T)$ must possess a global minimum in the interval $(0, 256)$ whose value is less than D_{CMC} . To make $D(T) < D_{\text{CMC}}$, we choose T as

$$T_{\text{opt}} = \arg \min_{T \in [T_{\text{min}}, 256]} D(T) \quad (13)$$

where T_{min} is some minimum threshold value.

The functional behavior of $D(T)$ is as follows. Same as before, $D(T)$ may be bounded by two nondecreasing functions, $D_1(T) \leq D(T) \leq D_2(T) \forall T$, where $D_1(T)$ and $D_2(T)$ are prediction error energy with scheme 2 and CME, respectively, with $D_1(256) = D_2(256)$. If at some $T = T_1$ all the active blocks become BABs, then $D(T) = D_1(T) \forall T \geq T_1$. Therefore, for higher values of T , $D(T)$ is nondecreasing. Also, since $D(T)$ and $D_1(T)$ differ in the motion estimation of the NABs and since the set of NABs is nonincreasing (in the sense of set inclusion) with T , $D(T) - D_1(T)$ is nonincreasing. Therefore, for values of T close to zero, the behavior of $D(T)$ depends on the increasing behavior of $D_1(T)$ and the decreasing behavior of $D(T) - D_1(T)$.

$D(T)$ may possess local minima, but exhaustive search to find the global minimum should not be used. Experimental results show that $D(T)$ does possess a global minimum which is less than D_{CMC} and has a few closely located local minima (all less than D_{CMC}) differing from the global minimum by very small magnitudes. So we use a fast search, which reduces the prediction error even if it finds a local minimum. Fig. 2(c) shows the $D(T)$ plot for frame #1 of the Tennis sequence.

During the search for minimum $D(T)$, assume $D(T_1)$ has been computed. For some $T_2 < T_1$, clearly $A(T_1) \subseteq A(T_2)$, where $A(T)$ is the set of active blocks at T . Therefore, threshold T_2 need to be applied only to $A'(T_1)$, where $A'(T)$ denotes the set of inactive blocks at T . To compute $D(T_2)$, motion estimation should be performed only for the blocks in $A(T_2) - A(T_1)$ (i.e., the difference set). Note that a BAB at T_1 (which has undergone SMME) may become a NAB at T_2 (which now needs to undergo CMC). During SMME, the prediction error

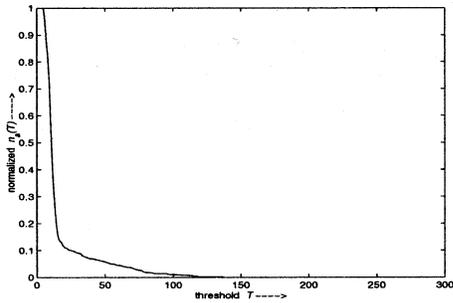


Fig. 3. $n_a(T)$ for frame #1 of the Tennis sequence.

with CMC is also calculated and stored (with an extra three additions and one comparison per motion vector) to avoid this motion estimation.

B. Iterative Algorithm for Threshold Selection

Let $n_a(T)$ denote the number of active blocks at threshold T . We use the following gradient method to find T_{\min} :

$$T_{\min} = \arg \max_{0 < T < 256} \{n_a(T-1) - n_a(T)\}. \quad (14)$$

Since finding $n_a(T)$ does not require any motion estimation, T_{\min} can be found using exhaustive search. Experimental results show that searching the entire gray scale is not necessary and a small range, for example (5,50), is adequate. For efficient computation, T is decreased from a higher to lower value and the number of inactive blocks at T which become active at $T-1$ determines the gradient value at T .

Finding T_{\min} in the above manner is motivated by the experimental plots of $n_a(T)$ and $D(T)$. It is observed that, at a certain value of T , $n_a(T)$ falls sharply (see Fig. 3) and then gradually decreases to zero as T is increased. The fall in $n_a(T)$ corresponds to a fall in $D(T)$. The fall in $n_a(T)$ indicates that change in intensity in a large chunk of blocks is less than that threshold value. Since the scene is assumed to have small gray-level changes in most of the blocks, the fall indicates that the predominant changes take place in the remaining blocks. A fall in $D(T)$ indicates that BABs match with some of the blocks lying at the boundaries of the moving objects. It is these BBs, in which the subblock matching produces much gain in the prediction error. Therefore, minimizing $D(T)$ is equivalent to tracking parts of the boundaries of the moving objects and other predominantly changed areas on the objects themselves.

The global minimum in $D(T)$ is typically very close to T_{\min} . Even if $T_{\text{opt}} < T_{\min}$, due to the slow increase of $D(T)$ near T_{opt} , the suboptimality of choosing $D(T_{\min})$ over $D(T_{\text{opt}})$ will be very small. Since the local minima of $D(t)$ is typically located close to T_{\min} , the search is restricted to $[T_{\min}, T_{\min} + P]$ for some P (say, 25). For all our simulations, the above method was seen to work extremely well in finding a suitable value of T_{\min} . To summarize, the iterative algorithm is as follows.

1. Compute T_{\min} using (14). $T_{\text{opt}} = T_{\min} + P$.
2. If $T_{\text{opt}} = T_{\min}$, stop. Else go to step 3.
3. If $D(T_{\text{opt}}) - D(T_{\text{opt}} - 1) < 0$, stop. Else $T_{\text{opt}} = T_{\text{opt}} - 1$, go to step 2.

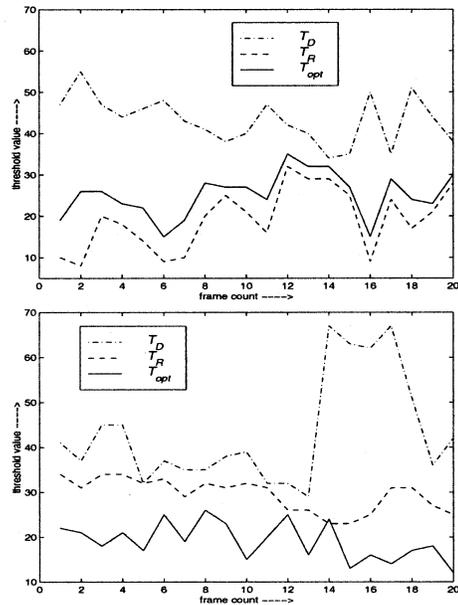


Fig. 4. T_D and T_R using scheme 2 for scene 1 (top) and scene 2 (bottom) of the Tennis sequence.

The advantage of our approach is that the CME, the SMME, and threshold selection are all done at the first pass and there is no second pass. At any threshold T , the cumulative number of blocks for which motion estimation is performed is equal to $A(T)$. So the total blocks for which the motion estimation is performed is equal to the active blocks at $T_{\text{opt}} - 1$, which in the worst case is $A(T_{\min})$. Note that determining the optimal threshold in the above manner is possible only in scheme 2a. Both scheme 1 and scheme 2 have monotonic $D(T)$ and $D(T)$ of scheme 1a most likely possesses minimum at $T = 0$.

VI. EXPERIMENTAL RESULTS

Simulations were performed with 20 352×240 frames each from scene 1 and scene 2 of the Tennis sequence (parts where the camera is stationary, in tune with this paper's assumption) and with 50 frames of the Football sequence. Scene 1, being a closeup, contains larger motion, more moving area, and more background noise than scene 2. Moving area in the Football sequence is larger than scenes 1 or 2. The proposed algorithms were implemented with 8×8 blocks, $\pm 7 \times \pm 7$ search window, integer motion vector, MSE matching criterion, exhaustive search, $T_p = 9$ pixels. Frame 0 in each sequence was not compensated, but each frame $n > 0$ was compensated from frame $n - 1$. Prediction error was calculated assuming all the blocks are motion compensated. To maintain uniformity, motion vectors for all methods were coded using the variable-length coding scheme presented in [15].

In the first part of the simulation, thresholds T_D and T_R for schemes 1 and 2 were found. Fig. 4 shows T_D and T_R for scheme 2. Similar plots were obtained for scheme 1. It is observed that T_R is smaller than T_D for all the frames except for one frame in scene 2 in which they are equal. In scene 1, both the prediction error and the side information can be reduced by a properly chosen constant threshold, since the intersection of the

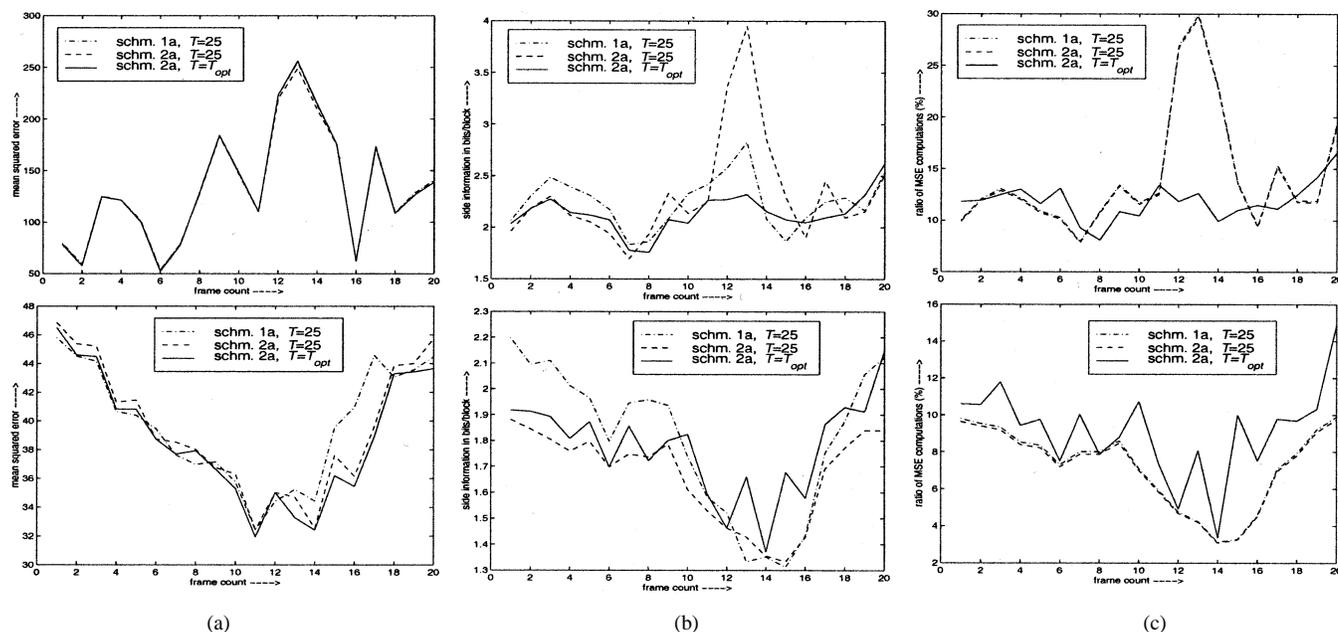


Fig. 5. (a) Mean-squared prediction error, (b) side information rate, and (c) ratio of MSE computation for scene 1 (top) and scene 2 (bottom) of the Tennis sequence using various schemes.

TABLE I
AVERAGE PERFORMANCE OF PROPOSED SCHEMES FOR TENNIS

video	sch→	1	1a	2	2a	2a, T_{opt}
scene1	mse	133.10	133.41	131.50	132.34	133.09
	rate	2.34	2.25	2.46	2.32	2.14
	comp	14.34	14.32	14.16	14.16	11.86
scene2	mse	39.59	39.76	39.39	39.51	38.86
	rate	1.84	1.81	1.69	1.67	1.78
	comp	7.24	7.23	7.12	7.12	9.15

intervals $[T_R, T_D]$ for all frames is nonempty. But scene 2 needs a variable threshold to show such reduction over all frames.

In the second part of the simulation, we compared the performances of schemes 1, 1a, 2 and 2a. All four schemes were implemented with a fixed threshold of 25. Scheme 2a with variable threshold obtained by the proposed threshold selection scheme (the proposed scheme) was also implemented. The selected threshold value T_{opt} for scenes 1 and 2 are shown in Fig. 4. Note that T_D and T_R in this figure are for scheme 2, so T_{opt} lying within T_D and T_R in scene 1 does not really show performance improvement. Rather, the variation of T_{opt} matches with that of T_D and T_R . A much smaller T_{opt} compared to T_R in scene 2 signifies that the scheme likely has more side information than CMC.

The prediction error, the side information, and the number of MSE computations for schemes 1a and 2a are shown in Fig. 5 for scenes 1 and 2. The side information, in bits per block, are the bits for the motion vector plus the subblock side information per block. The number of MSE computations is expressed as a percentage of the number in CMC with a 8×8 block. The prediction error and the computation performances of schemes 1a and 2a($T = 25$) are similar. Comparing schemes 2a(T_{opt}) with 2a($T = 25$), in scene 1 the prediction error is increased which indicates T_{opt} is more than 25 (therefore, computation

is decreased) and side information is decreased. In scene 2 the reverse happens with decrease in the error and increase in the information. So scheme 2a(T_{opt}) performance is comparable to that with $T = 25$. Note that in frame #13 of scene 1, the side information peak for scheme 2a($T = 25$) is due to noise. Many subblock motion vectors differ from CMC vectors. Many of the BABs are due to noise, which get better compensated by the subblock schemes. With T_{opt} this peak disappears.

The average values of the prediction error, the side information rate and the number of computations over each scene are displayed in Table I. As expected, scheme 1 (2) has less prediction error and more side information than scheme 1a (2a) since SME (SMME) is applied to all active blocks, not just BBs (BABs). Scheme 2 (in scene 2) and 2a (in both scenes) performs better than schemes 1/1a in both the prediction error and the side information. The performances of schemes 2a($T = 25$) and 2a(T_{opt}) are comparable. Schemes 2/2a need less computation than schemes 1/1a. Note that the computations actually refer to the percentage of active blocks. Therefore, even if any fast search like three-step search [4], logarithmic search [1], etc., is employed, the computation percentage will be similar.

The third part of the simulation dealt with the comparison of the proposed scheme with CMC and the schemes of Puri *et al.* [13] and Chan *et al.* [10]. To maintain uniformity, Puri's scheme was implemented with threshold 25. Chan's variable block size scheme was implemented with 16×16 (largest) to 4×4 (smallest) block. Chan's mean-squared prediction error threshold for splitting a block was chosen as 200 heuristically in order to have equivalence with $T_g = 25$ and $T_p = 9$. The prediction error, the side information and the number of computations of various schemes for scenes 1 and 2 are shown in Fig. 6. The prediction error of the proposed scheme is less than CMC (more in scene 2), but more than both Puri and Chan. This is because Puri/Chan splits only the blocks having large prediction errors and estimate the motion vectors of the subblocks through

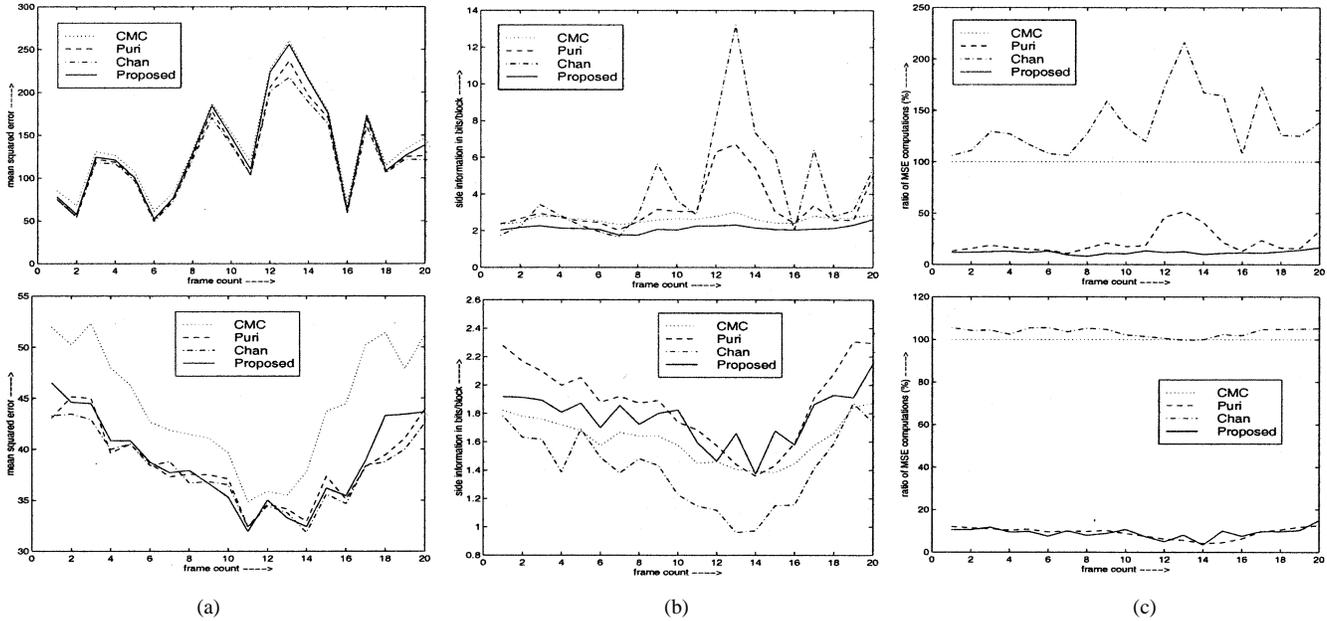


Fig. 6. (a) Mean squared prediction error; (b) side information rate; (c) ratio of MSE computation; for scene 1 (top) and scene 2 (bottom) of the Tennis sequence using existing and proposed schemes.

TABLE II
AVERAGE PERFORMANCE OF EXISTING SCHEMES

video	para- meter	CMC	Puri		Chan	sch2a T_{opt}
			$T_g=25$	$T_g=3$		
scene1	mse	138.92	127.05	116.89	122.92	133.09
	rate	2.63	3.35	16.16	4.29	2.14
	comp	100	21.93	201.52	136.75	11.86
scene2	mse	44.43	38.53	33.69	37.99	38.86
	rate	1.62	1.88	11.16	1.41	1.78
	comp	100	9.12	200.03	103.44	9.15
Foot -ball	mse	188.19	128.69	103.18	121.74	187.66
	rate	5.45	10.15	25.01	10.25	5.17
	comp	100	63.96	201.38	156.08	43.35

the full search. Side information of both Puri and Chan is much more than CMC in scene 1, while that of the proposed scheme is always less. But in scene 2, Chan performs the best. Note that scene 2 has little noise and Chan uses 16×16 blocks in the first level, which explains its better performance. In scene 2, the side information of the proposed scheme is marginally more than CMC. The computation plots clearly indicate the advantage of the conditional motion estimation. Puri and the proposed scheme require much less computation compared to CMC and Chan.

The average values of the prediction error, the side information rate and the number of MSE computations over each sequence are shown in Table II. The tables also include Puri's scheme with $T_g = 3$ and $T_p = 10$, the values used in [13]. Clearly, for all sequences, $T_g = 3$ is too small; almost all the blocks become uncompensable, so prediction error decreases and side information increases significantly. Note that for the Football sequence, there are more active blocks due to its larger moving area and consequently more computations than the Tennis sequence. Since there is some deformatory motion in the

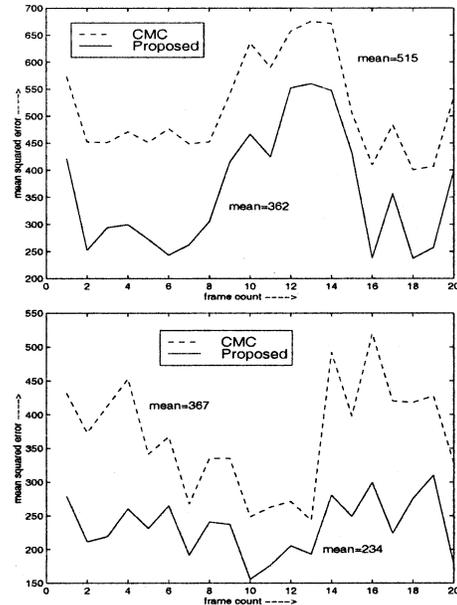


Fig. 7. Mean squared prediction errors over BABs in scene 1 (top) and scene 2 (bottom) of the Tennis sequence.

scene (due to the human body movements), a smaller block size (4×4) improves the matching accuracy. Therefore, Puri and Chan's scheme produces much less prediction error than CMC, but at the cost of large side information and computation. The proposed scheme, with comparable prediction error and side information, needs only 43% MSE computations than CMC. Considering all three performance measures, one concludes that the proposed scheme provides a better tradeoff between computation and rate-distortion performance than both Chan's and Puri's scheme.

From Table II, it is seen that the active blocks undergoing motion estimation is very small (12% in scene 1 and 9% in scene

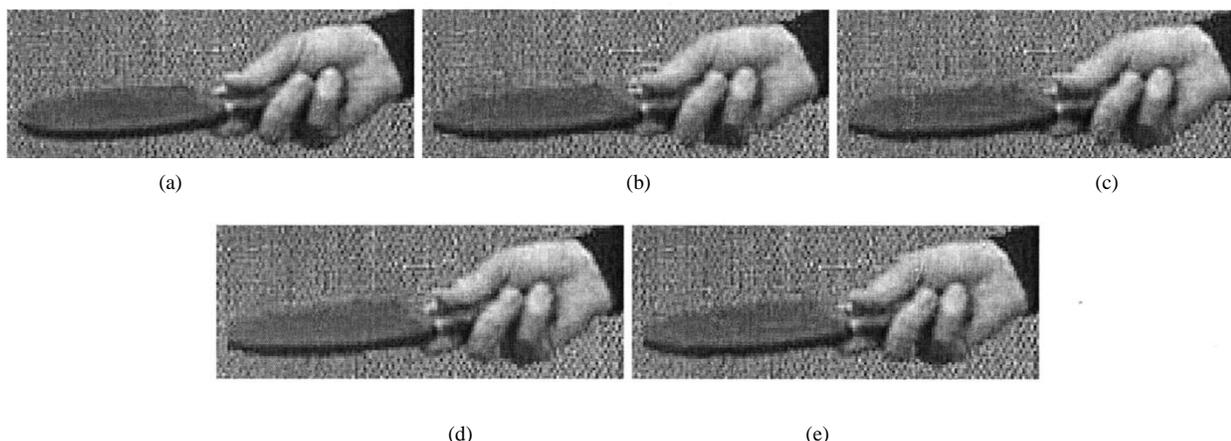


Fig. 8. Motion-compensated prediction due to various schemes for frame #5 from scene 1 of the Tennis sequence. (a) Original frame. (b) Prediction by CMC. (c) Prediction by Puri, $T_g = 25$. (d) Prediction by Chan. (e) Prediction by the proposed scheme.

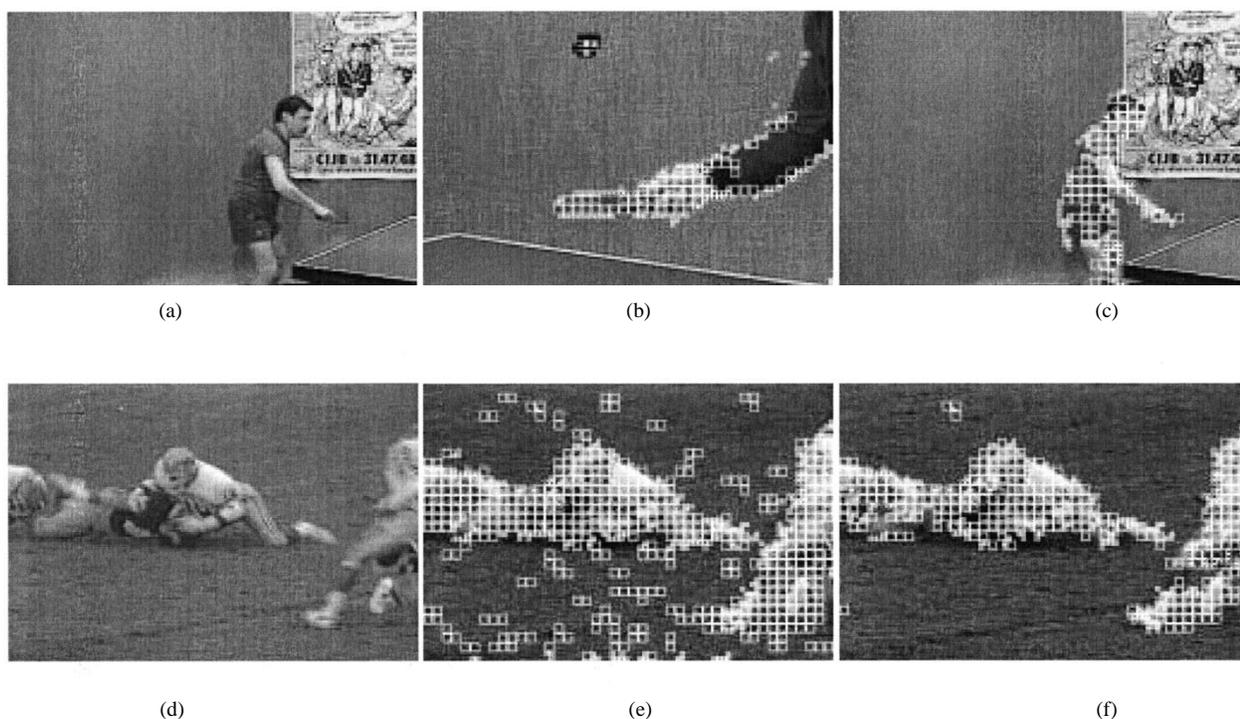


Fig. 9. Blocks and subblocks having nonnull motion vectors in few frames of Tennis and Football sequences using the proposed scheme. (a) Original frame #1 of scene 2. (b) Frame #5 of scene 1. (c) Frame #1 of scene 2. (d) Original frame #1 of Football. (e) Football frame, $T = 25$. (f) Football frame, T_{opt} .

2). Out of these, only the BABs undergo the SMME, offering a reduction in the prediction error. The prediction error in the NABs remain unchanged since these undergo CMC. The prediction error in the inactive blocks are same or more than CMC. The plots of Fig. 6(a) show that the decrease in very few BABs is more than the total increase in all the inactive blocks. Fig. 7 shows the prediction error plots only over BABs. We observe that the percentage of prediction error reduction over CMC is 30% in scene 1 and 36% in scene 2.

Fourth, to compare the subjective quality of the predictions, we display a part of frame #5 of scene 1 and its predictions in various schemes in Fig. 8. Note the blockiness at the lower edge of the bat (darker hue) and the hand (little finger) in CMC when compared to the original frame. This blockiness is partly present in Puri, but Chan and the proposed scheme produce smoother

predictions. Note also the fuzzy upper edge of the bat (lighter hue) in Puri and partly in Chan, while the proposed scheme gives a clearer shape. Chan’s scheme is compute-intensive. Puri’s prediction could be made smoother with a smaller threshold, but that increases its computation. Therefore, the proposed scheme provides the best tradeoff between the subjective quality and the computation.

Finally, original frame #5 of scene 1, frame #1 of scene 2, and frame #1 of the Football sequence are displayed in Figs. 8(a), 9(a), and (d). The corresponding frames showing the blocks and the subblocks having nonnull motion vectors are displayed in Fig. 9(b), (c), and (f). Most subblocks are aligned along the boundaries of the moving areas. This is seen to continue in all frames. This substantiates that the proposed automatic threshold selection scheme adapts to the motion in a frame so that most of

the BABs match with the moving boundaries. In the proposed scheme, the advantage of a variable threshold T_{opt} over a fixed threshold $T = 25$ is clearly seen in Fig. 9(e) and (f). T_{opt} is seen to select the most significant blocks in a frame for motion estimation.

VII. CONCLUSION

In this paper, we proposed a conditional motion-estimation scheme that classifies the blocks as active or inactive based on a threshold. Only the active blocks undergo motion estimation. Further, the BABs undergo SMME. The threshold value is selected automatically using an iterative method. During the iterations, all active blocks undergo motion estimation, so no more motion estimation is required after the threshold is found. It was observed that the proposed scheme provides better tradeoff between the computation and the rate-distortion performance than the CMC and two other existing schemes. Subjectively, the proposed scheme provided a smoother motion-compensated frame than others.

It may be concluded that predominant changes in a scene take place at the boundaries of the moving objects, therefore the blocks lying there can be better compensated by partitioning them into subblocks. The proposed scheme significantly saves computation, but at the same time produces equal or better rate-distortion performance with better subjective quality. Because of its low computation, the proposed algorithm can be a strong candidate for real-time video compression. While simulation results are dependent on the test data, considering the generality of this test data, it may be concluded that the proposed algorithm will perform similarly with other sequences.

REFERENCES

- [1] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, no. 12, pp. 1799–1808, Dec. 1981.
- [2] H. G. Musmann, P. Pirsh, and H.-J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, no. 4, pp. 523–548, 1985.
- [3] R. Srinivasan and K. R. Rao, "Motion-compensated coder for video-conferencing," *IEEE Trans. Commun.*, vol. COM-35, pp. 297–304, Mar. 1987.
- [4] T. Koga, K. Inuma, A. Hirano, Y. Iiyima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. NTC81*, New Orleans, LA, 1981, pp. G5.3.1–G5.3.5.
- [5] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, no. 7, pp. 950–953, July 1990.
- [6] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 148–157, Apr. 1993.
- [7] H. Gharavi and M. Mills, "Block-matching motion estimation algorithm—New results," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 649–651, May 1990.

- [8] M.-J. Chen, L.-G. Chen, T.-D. Chiueh, and Y.-P. Lee, "A new block-matching criterion for motion estimation and its implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 231–236, June 1995.
- [9] G. B. Rath and A. Makur, "A fast matching criterion for VLSI implementation of block based motion estimation," *Signal Processing*, vol. 73, no. 3, pp. 297–301, Mar. 1999.
- [10] M. H. Chan, Y. B. Yu, and A. G. Constantinides, "Variable size block matching motion compensation with applications to video coding," *Proc. Inst. Elect. Eng.*, pt. I, vol. 137, no. 4, pp. 205–212, Aug. 1990.
- [11] J. W. Kim and S. U. Lee, "Hierarchical variable block size motion estimation technique for motion sequence coding," *Opt. Eng.*, vol. 33, no. 8, pp. 2553–2561, Aug. 1994.
- [12] G. J. Sullivan and R. L. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," in *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, 1991, pp. 85–90.
- [13] A. Puri, H.-M. Hang, and D. L. Schilling, "Interframe coding with variable block-size motion compensation," in *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, 1987, pp. 65–69.
- [14] K. Kamikura and H. Watanabe, "Global motion compensation in video coding," *Electron. Commun. Jpn.*, pt. 1, vol. 78, no. 4, pp. 91–102, 1995.
- [15] G. B. Rath and A. Makur, "Iterative least squares and compression based estimations for a 4-Parameter linear global motion model and global motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1075–1099, Oct. 1999.
- [16] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
- [17] G. M. Schuster and A. K. Katsaggelos, "A theory for the optimal bit allocation between displacement vector field and displaced frame difference," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1739–1751, Dec. 1997.
- [18] M. T. Orchard, "Predictive motion-field segmentation for image sequence coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 54–70, Feb. 1993.

Gagan Bihari Rath received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, in 1990 and the M.E. and Ph.D. degrees in electrical communication engineering from the Indian Institute of Science, Bangalore in 1993 and 1999, respectively. He is currently with the TEMICS Group, IRISA, Rennes, France. His research interests include digital signal processing and communications.

Anamitra Makur received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, in 1985 and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1986 and 1990, respectively. He is currently an Associate Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. During 1990–2002, he was with the Electrical Communication Engineering Department, Indian Institute of Science, Bangalore, India, as an Assistant Professor, Associate Professor, and Professor. He also held visiting positions with the University of California, Santa Barbara, and the University of Kaiserslautern, Germany. His research interests are in signal compression, subband coding and filterbank design, and image/video processing.

Dr. Makur currently serves as the Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING.