

# Iterative Least Squares and Compression Based Estimations for a Four-Parameter Linear Global Motion Model and Global Motion Compensation

Gagan B. Rath and Anamitra Makur

**Abstract**—In this paper, a four-parameter model for global motion in image sequences is proposed. The model is generalized and can accommodate global object motions besides the motions due to the camera movements. Only the PAN and the ZOOM global motions are considered because of their relatively more frequent occurrences in real video sequences. Besides the traditional least-squares estimation scheme, two more estimation schemes based on the minimization of the motion field bit rate and the global prediction error energy are proposed. Among the three estimation schemes, the iterative least-squares estimation is observed to be the best because of the least computational complexity, accuracy of the estimated parameters, and similar performance as with the other schemes. Four global motion compensation schemes including the existing pixel-based forward compensation are proposed. It is observed that backward compensation schemes perform similarly to the corresponding forward schemes except for having one frame delay degradation. The pixel-based forward compensation is observed to have the best performance. A new motion vector coding scheme is proposed which has similar performance as the two-dimensional entropy coding but needs much less computation. Using the proposed coding scheme with the pixel-based forward compensation, we obtain 61.85% savings in motion field bit rate over the conventional motion compensation for the Tennis sequence.

**Index Terms**—Global motion compensation, global motion estimation, motion compensation, motion estimation, motion field, motion vector, video coding.

## I. INTRODUCTION

ADVANCES in digital electronics, miniaturization technology, fiber optics, and information technology have heralded a new era of communications. They have not only changed the lifestyles of people but made things like information and computers basic necessities. Among all forms of communications, digital video is the most superior, but the most demanding in terms of channel bandwidth and storage requirements. Applications such as multimedia, videophone, teleconferencing, high-definition television (HDTV), digital television (DTV), and CD-ROM storage have recently emerged because of the progress in compression technology. Although the capacities of the communication channels and the storage media are increasing day by day, the tremendous user demand and the high per-unit service cost have made the compression of digital video very important.

Manuscript received April 16, 1998; revised April 22, 1999. This paper was recommended by Associate Editor B. Zeng.

The authors are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore-560012, India.

Publisher Item Identifier S 1051-8215(99)08175-6.

The basic idea to compress a digital video sequence is to remove temporal as well as spatial redundancies. For high compression of video, effective temporal redundancy removal is very essential. One of the efficient methods commonly used for removing temporal redundancy is motion compensated predictive coding. In this method, a frame is predicted based on a previous reference frame and the motion between the two frames. The crucial thing in motion compensated predictive coding is the estimation of motion between the current frame and the reference frame. The motion in a video sequence can occur due to moving objects or the motion of the camera. Moving objects cause local luminance changes whereas the changes due to the motion of the camera are global. The receiver or the decoder needs only this change information to reconstruct the current frame from the previously decoded frame. Hence the amount of compression of the sequence depends on how this change information is coded. A lesser required bit rate for the change information results in a higher compression. Since the luminance changes are primarily because of the motion of objects or camera, they can be encoded in terms of motion information. Because the new information in the current frame, besides the changes due to motion, is relatively small, the amount of compression to a large extent depends on how the motion between successive frames is estimated and encoded. This realization has brought forth a host of motion estimation and compensation techniques for compressing a video sequence.

The block matching algorithm (BMA) [1], [2] has been one of the popular motion estimation techniques because of its simplicity, robustness, and implementational advantages. In this technique, a frame to be coded is segmented into square blocks of pixels. Motion of each block is estimated as a displacement vector by finding its best match in a search area in the previously decoded frame. The best matching blocks are used to form a so-called motion compensated frame which is used as a prediction for the current frame. The prediction error frame, called the displaced frame difference, and the set of displacement vectors, called the displacement vector field or the motion field, are transmitted to the receiver. Clearly, in this approach, the compression performance depends on the coding schemes used for the prediction error frame and the motion field.

The conventional block matching algorithm cannot differentiate between the global and the local motion. As a result, in the presence of camera motion, the motion field

represents the combination of both the global as well as the local motion. Transmitting the combined motion field as such requires a large number of bits since the global motion affects all the blocks in a frame. But the global motion itself can be encoded in terms of a few parameters. Therefore, considerable bit rate can be saved by encoding the local motion and the global motion separately. The conventional block matching algorithm also assumes that all the pixels in a block have equal displacements. This assumption is no longer valid when the scene contains nontranslational motions, for example, zoom, rotation, etc. Because of the improper model, the motion is inefficiently or incorrectly compensated. As a result, the prediction error energy increases, which results in lower compression efficiency. This drawback can be removed by performing global motion compensation along with local motion compensation.

Various global motion modeling, estimation, and compensation schemes have been proposed in the literature [3]–[10]. These schemes try to estimate the parameters corresponding to different camera motions. Further, the compression performance of the encoder is not taken into account in the estimation of the parameters. In this paper, we first propose a generalized four-parameter model for the global motion. This model includes not only the motion due to camera but also that due to global objects, i.e., objects occupying the whole frame. In addition to the least-squares estimation of parameters, we propose two more estimation methods based on the minimization of the bit rate for the motion field and the minimization of the global motion compensated prediction error. Since these two methods are very compute-intensive, we suggest alternative fast search methods. Following similar approach as in existing schemes, motion compensation is decomposed into two stages where the first stage compensates the global motion and the second stage compensates the local motion. To reduce computation, we propose three more global motion compensation schemes in addition to the existing pixel-based forward compensation scheme. Finally, for efficient coding of the local motion vectors, we also propose a variable-length coding scheme.

The paper is organized as follows: In Section II, we propose the four-parameter model for the global motion field. First, we model the isolated motions of the camera and then generalize them to include global object motions. Section III deals with the estimation of the global parameters. We consider both motion field based and frame-based estimation schemes. We propose the global motion compensation schemes in Section IV and the motion vector coding scheme in Section V. Experimental results are presented in Section VI, and we draw conclusions in the last section.

## II. GLOBAL MOTION MODELING

It has been common in the video compression literature to refer to the motion due to the camera as global motion and that due to the objects as local motion. This is because the motion due to the camera takes place all over the frame, whereas the motion due to the objects is localized. Thus, there are global

motion models such as pan, tilt, zoom, etc., which correspond to different movements of the camera.

These models are appropriate if the goal is to extract the individual camera motion parameters from the video sequence. In video compression, the purpose of global motion modeling is to compress the global motion field using as few parameters as possible. In some scenes, the object motion can produce global changes all over the frame. For example, in close-up scenes when the moving objects occupy the entire frame, the motion is global. We will refer to such objects as global objects. The global object motion has more degrees of freedom than the camera motion. Therefore, the above models should be generalized. Another reason for generalization is that different camera motions produce similar motion fields; for example, zoom and the translation along the camera axis produce similar motion fields. The same is true for slow pan and translation parallel to the image plane. Since the objective is not to find these individual motion parameters, these motions can be included in a single model. Moreover, generalization will make the encoder free from the interpretation of the motion field as done by the human visual system.

Let us assume that the luminance changes between successive frames are only because of the camera motion or the global object motion. Corresponding pixels in successive frames are assumed to have equal intensity values. Let there be  $I$  rows and  $J$  columns of pixels in a frame of the sequence. Let the coordinates of the pixel  $(i, j)$ ,  $i = 0, 1, \dots, I - 1$ ,  $j = 0, 1, \dots, J - 1$  be  $\mathbf{s}_{ij} \equiv (s_i, s_j)$  with respect to the center of the frame, where  $s_i$  and  $s_j$  denote the  $x$  and  $y$  coordinates, respectively. The distance between two adjacent pixels, vertically or horizontally located, is assumed to be unity. Let us denote the displacement of the pixel  $(i, j)$  by  $\mathbf{u}_{ij}$ . Let us assume that the camera has the central projection model [3] in which the camera coordinate system lies at the lens of the camera, and the image coordinate system sits at the focal plane. With these assumptions, we present the following two models. It is to be noted here that, though the names of the models are the same as those for the movements of the camera, they refer to generalized global motions.

### A. PAN or Constant Motion

The rotation of the camera about the  $x$ -axis (vertical) or the  $y$ -axis (horizontal) of the camera coordinate system is commonly known as pan. Some authors refer to the vertical pan as tilt. The pan parameter is normally represented as a two-dimensional vector in which the scalar components refer to the rotation angles about the two axes. Let  $\theta_x$  and  $\theta_y$  be the rotation angles about  $x$ -axis and  $y$ -axis, respectively. If  $\theta_x$  and  $\theta_y$  are sufficiently small, the displacement of the pixel  $(i, j)$  is given as [3]

$$\mathbf{u}_{ij} = F \begin{bmatrix} \theta_y \\ \theta_x \end{bmatrix} \quad (1)$$

where  $F$  is the focal length of the camera. We generalize this concept as follows. We call the global motion PAN when the motion field is constant, i.e., it does not depend on the

position of the pixels, i.e.,

$$\mathbf{u}_{ij} = \mathbf{p}, \quad i = 0, 1, \dots, I-1, \quad j = 0, 1, \dots, J-1 \quad (2)$$

where  $\mathbf{p} \equiv [p_x \ p_y]^t$  is a constant vector. We call  $p_x$  and  $p_y$  the PAN parameters. It is easy to see that this model includes not only the slow pan of the camera, but also the camera translation and the global object translation along a plane parallel to the image plane. If the motion is because of the slow pan of the camera, then

$$p_x = F\theta_y \quad (3)$$

$$p_y = F\theta_x. \quad (4)$$

Note that if  $\theta_x$  and  $\theta_y$  are not sufficiently small, then the resulting motion is not constant [3]; therefore it cannot be modeled using the PAN model.

### B. ZOOM or Linear Motion

The camera is said to be zoomed when the focal length of its lens system is changed. The zoom parameter is normally expressed as a scalar which is the ratio of the focal lengths. Zoom causes linear motion along both  $x$ -axis and  $y$ -axis of the image plane, i.e., the scalar components of the motion vector of a pixel are directly proportional to the corresponding scalar components of its displacement from the center of the frame. The proportionality constants along the  $x$ -axis and the  $y$ -axis, which are functions of the zoom parameter, are equal. It has been shown [4] that, under zoom, the motion vector of the pixel  $(i, j)$  is

$$\mathbf{u}_{ij} = \left( \frac{F_2}{F_1} - 1 \right) \begin{bmatrix} s_i \\ s_j \end{bmatrix} \quad (5)$$

where  $F_1$  and  $F_2$  are the focal lengths before and after the zoom.  $F_2/F_1$  is usually called the zoom parameter.

A similar motion field is created when the camera is translated along the direction of view. Let  $z_1$  and  $z_2$  be the  $z$ -coordinates (i.e., along the direction of view) of the object point, which corresponds to the pixel  $(i, j)$  on the image plane, before and after the camera translation, respectively. The displacement of the pixel  $(i, j)$  is given as

$$\mathbf{u}_{ij} = \left( \frac{z_1}{z_2} - 1 \right) \begin{bmatrix} s_i \\ s_j \end{bmatrix}. \quad (6)$$

Similar equations are obtained when a global object moves toward or away from the camera along the direction of view. We generalize the above concept as follows. We call the global motion ZOOM when the scalar components of the motion vector of a pixel are directly proportional to the corresponding scalar components of its displacement from the center of the frame, i.e.,

$$\mathbf{u}_{ij} = \begin{bmatrix} z_x s_i \\ z_y s_j \end{bmatrix} \quad (7)$$

where  $z_x$  and  $z_y$  are called ZOOM parameters. The purpose of bringing in two ZOOM parameters is to include linear motion due to global objects in the model, in which case  $z_x$  and  $z_y$

need not be equal. When the linear motion is because of the zooming of the camera

$$z_x = z_y = \left( \frac{F_2}{F_1} - 1 \right) \quad (8)$$

and when it is due to the translation of camera along the direction of view

$$z_x = z_y = \left( \frac{z_1}{z_2} - 1 \right). \quad (9)$$

Another reason for using two ZOOM parameters is that in majority of situations, global motion is usually accompanied by local motion. The values of the estimated parameters are affected differently along the  $x$ -axis and the  $y$ -axis depending on the nature of the local motion. One of the estimated parameters may be a better estimate than the other. Representing both the parameters by one parameter will not produce a better estimate.

Simultaneous occurrence of global motions, although relatively less frequent, is not rare. For example, simultaneous zoom and pan of camera is a frequent occurrence in many video sequences. In such cases, the global motion field will be the combination of both PAN and ZOOM. For mathematical formulation, the effective global motion vector has to be broken into two global motion vectors, each corresponding to one of the above mentioned models. The order of these two motions is important, since different orders give rise to different models for the resultant motion vector. ZOOM followed by PAN gives

$$\mathbf{u}_{ij} = \begin{bmatrix} z_x s_i \\ z_y s_j \end{bmatrix} + \mathbf{p} \quad (10)$$

whereas PAN followed by ZOOM gives

$$\mathbf{u}_{ij} = \begin{bmatrix} z_x(s_i + p_x) \\ z_y(s_j + p_y) \end{bmatrix} + \mathbf{p} \quad (11)$$

$$= \begin{bmatrix} z_x s_i \\ z_y s_j \end{bmatrix} + \begin{bmatrix} (1 + z_x)p_x \\ (1 + z_y)p_y \end{bmatrix}. \quad (12)$$

We can generalize these two models by a single model as follows:

$$\mathbf{u}_{ij} = \begin{bmatrix} a_1 s_i \\ a_3 s_j \end{bmatrix} + \begin{bmatrix} a_2 \\ a_4 \end{bmatrix} \quad (13)$$

where

$$a_1 = z_x \quad (14)$$

$$a_2 = f_1(p_x, z_x) \quad (15)$$

$$a_3 = z_y \quad (16)$$

$$a_4 = f_2(p_y, z_y). \quad (17)$$

The functions  $f_1$  and  $f_2$  depend on the order of the global motions. It is not necessary to solve the above equations for  $p_x$ ,  $p_y$  and  $z_x$ ,  $z_y$  since we can now consider  $a_1$ ,  $a_2$ ,  $a_3$ , and  $a_4$  to be the model parameters.

### III. ESTIMATION OF GLOBAL PARAMETERS

Global motion parameters can be estimated either from the motion field [4]–[8] or directly from the image frames [3], [9], [10]. The estimation procedure depends on the proposed global motion model. Tse and Baker [4] and Kamikura and Watanabe [5] use the  $l_2$  norm of the displacement error as the estimation criterion. Reference [6] uses a weighted least-squares criterion where the weights are determined based on the displacement errors of a selected set of blocks. A recursive least-squares estimation method is used in [7]. The method of [8] finds the parameters from the Hough transform of the motion vectors. Hoetter's differential technique [3] uses a least-squares approach based on a second-order luminance signal model. In [9], camera motion is estimated using binary matching of the edges in successive frames. Reference [10] finds the global parameters by applying a matching technique to the background pixels. The estimates obtained from the motion field are affected by the displacement measurement errors. In the conventional block matching algorithm, these errors can be due to noise, finite resolution of the motion vector, the assumption that all the pixels in one block have equal displacements, limited search space, etc. The estimates obtained directly from the image frames are free from these errors. Nevertheless, both the motion field based and frame based estimates are affected by local motion. Therefore the parameters are estimated either from the histograms or through iterative methods.

In the following, we propose three estimation methods. The first two methods compute the parameters from the motion field obtained by the block matching algorithm. The first method applies the least-squares criterion to the individual components of the motion field. The second method is based on the minimization of the bit rate for the motion field. The third scheme estimates the parameters directly from the image frames. It is based on the minimization of the energy of the globally compensated difference frame, i.e., the difference between the current frame and the globally compensated frame. Although the criterion is similar to [10], the estimation procedure is quite different. In the context of compression, the second and the third schemes are more appropriate. But as we will see, these two require very large amounts of computation. Based on experimental results, we suggest alternative fast estimation schemes for these two.

#### A. Iterative Least-Squares Estimation (ILSE)

The conventional block matching algorithm assumes that all the pixels in a block have equal displacements, and thus estimates one motion vector for each block. Let there be  $K$  rows and  $L$  columns of blocks in a frame of the sequence. Let us assume that the motion vector of a block is the motion vector of the center pixel of that block. Let  $\mathbf{v}_{kl} \equiv (v_{klx}, v_{kly})$  be the measured motion vector of the block  $(k, l)$ ,  $k = 0, 1, \dots, K-1$ ,  $l = 0, 1, \dots, L-1$ , whose center pixel's coordinates are  $(s_k, s_l)$  with respect to the center of the frame.  $v_{klx}$  and  $v_{kly}$  denote the  $x$  and  $y$  components of the motion vector  $\mathbf{v}_{kl}$ , respectively.  $s_k$  and  $s_l$  denote the  $x$  and  $y$  coordinates of the center pixel of the block  $(k, l)$ , respectively.

We can estimate the global parameters using the following criteria:

$$\min_{a_1, a_2} \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} (v_{klx} - a_1 s_k - a_2)^2 \quad (18)$$

$$\min_{a_3, a_4} \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} (v_{kly} - a_3 s_l - a_4)^2. \quad (19)$$

By differentiating with respect to the parameters, and setting the derivatives to zero, we obtain the following solution as shown in (20)–(23) found at the bottom of the next page.

Since the blocks are symmetrically located with respect to the center of the frame

$$\sum_{k=0}^{K-1} s_k = 0 \quad (24)$$

$$\sum_{l=0}^{L-1} s_l = 0 \quad (25)$$

which simplifies the solution to

$$a_1 = \frac{\sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{klx} s_k}{L \sum_{k=0}^{K-1} s_k^2} \quad (26)$$

$$a_2 = \frac{\sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{klx}}{KL} \quad (27)$$

$$a_3 = \frac{\sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{kly} s_l}{K \sum_{l=0}^{L-1} s_l^2} \quad (28)$$

$$a_4 = \frac{\sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{kly}}{KL}. \quad (29)$$

Since all the blocks are taken into account, the estimates will be affected by the local motion if it is present. To eliminate the influence of local motion, we can follow an iterative procedure as follows: using the above estimated values and the model equation (13), we compute the motion vectors of the center pixels of all the blocks. We call them global motion vectors and call the motion field consisting of these global motion vectors the global motion field. Now we reestimate the parameters using only the blocks whose motion vectors match with the global motion field. By matching, we mean that a motion vector lies within a threshold distance from the corresponding global motion vector. We call the threshold the motion vector matching threshold. It is to be noted that, for the computation of the parameters after the first iteration, the simplified equations (26)–(29) are no longer valid. We can repeat this procedure till convergence is achieved. Experimental results show that convergence occurs in a very few iterations.

### B. Motion Field Bit Rate Minimization (MFBRM)

One of the purposes of the global motion estimation is to remove the contribution of the global motion from the original motion field and transmit only the local motion information. It is expected that the bit rate for transmitting the local motion information will be less than that for transmitting the original motion field. For maximum compression, we would like to minimize the bit rate for the local motion field. The least-squares estimation method may not achieve this optimality since the estimation criteria do not include the bit rates for the motion vectors.

Given the parameter values, we can generate the global motion field using the model equation (13). Let us denote the global motion field by  $\mathbf{V}^G$ . Thus

$$\mathbf{V}^G \equiv \{\mathbf{v}_{kl}^G; \quad k = 0, 1, \dots, K-1; l = 0, 1, \dots, L-1\} \quad (30)$$

where

$$\mathbf{v}_{kl}^G = \begin{bmatrix} a_1 s_k + a_2 \\ a_3 s_l + a_4 \end{bmatrix} \quad (31)$$

and  $(s_k, s_l)$  are coordinates of the center pixel of the block  $(k, l)$ .  $\mathbf{v}_{kl}^G$  denotes the global motion vector of the block  $(k, l)$ . The motion field obtained by subtracting the generated global motion field from the original motion field can be called the local motion field. Let us denote this motion field by  $\mathbf{V}^L$ . Thus

$$\mathbf{V}^L \equiv \{\mathbf{v}_{kl}^L; \quad k = 0, 1, \dots, K-1; l = 0, 1, \dots, L-1\} \quad (32)$$

$$= \{\mathbf{v}_{kl} - \mathbf{v}_{kl}^G; \quad k = 0, 1, \dots, K-1; \\ l = 0, 1, \dots, L-1\} \quad (33)$$

where  $\mathbf{v}_{kl}^L$  denotes the local motion vector of block  $(k, l)$ . Let  $B_{kl}$  and  $B_{kl}^L(a_1, a_2, a_3, a_4)$  denote the number of bits required to encode  $\mathbf{v}_{kl}$  and  $\mathbf{v}_{kl}^L$ , respectively. Clearly,  $B_{kl} = B_{kl}^L(0, 0, 0, 0)$ . The estimation criterion can be stated as follows:

$$\min_{a_1, a_2, a_3, a_4} \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} B_{kl}^L(a_1, a_2, a_3, a_4). \quad (34)$$

Note that this criterion holds if the parameters are transmitted using a preassigned number of bits. If the parameters are transmitted using a variable number of bits, then the criterion can be modified to add the bits for the parameters. In the backward global motion compensation scheme (explained in the next section), the parameters are not transmitted, and therefore the above criterion holds good.

This criterion requires, first of all, a predetermined coding scheme for the motion vectors. Second, the coding scheme has to be variable length so that  $B_{kl}^L$  becomes a function of  $a_1, a_2, a_3$ , and  $a_4$ . In Section V, we present a coding scheme which is appropriate for the purpose.

The solution of the above minimization problem requires that  $B_{kl}^L$  be expressed as a differentiable function of  $a_1, a_2, a_3$ , and  $a_4$ . This function, on the other hand, depends on the coding scheme for the motion vectors. Since such a function is not available, we would have to take recourse to exhaustive search for finding the global minimum. But exhaustive search over a four-dimensional space is extremely compute-intensive, and is not practical. Experimental results in Fig. 1 show that the function is convex over a wide region near the global minimum. If the initialized values of the parameters are

$$a_1 = \frac{\left( K \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{klx} s_k \right) - \left( \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{klx} \right) \left( \sum_{k=0}^{K-1} s_k \right)}{\left( KL \sum_{k=0}^{K-1} s_k^2 \right) - \left( L \left( \sum_{k=0}^{K-1} s_k \right)^2 \right)} \quad (20)$$

$$a_2 = \frac{\left( \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{klx} \right) \left( \sum_{k=0}^{K-1} s_k^2 \right) - \left( \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{klx} s_k \right) \left( \sum_{k=0}^{K-1} s_k \right)}{\left( KL \sum_{k=0}^{K-1} s_k^2 \right) - \left( L \left( \sum_{k=0}^{K-1} s_k \right)^2 \right)} \quad (21)$$

$$a_3 = \frac{\left( L \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{kly} s_l \right) - \left( \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{kly} \right) \left( \sum_{l=0}^{L-1} s_l \right)}{\left( KL \sum_{l=0}^{L-1} s_l^2 \right) - \left( K \left( \sum_{l=0}^{L-1} s_l \right)^2 \right)} \quad (22)$$

$$a_4 = \frac{\left( \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{kly} \right) \left( \sum_{l=0}^{L-1} s_l^2 \right) - \left( \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} v_{kly} s_l \right) \left( \sum_{l=0}^{L-1} s_l \right)}{\left( KL \sum_{l=0}^{L-1} s_l^2 \right) - \left( K \left( \sum_{l=0}^{L-1} s_l \right)^2 \right)} \quad (23)$$

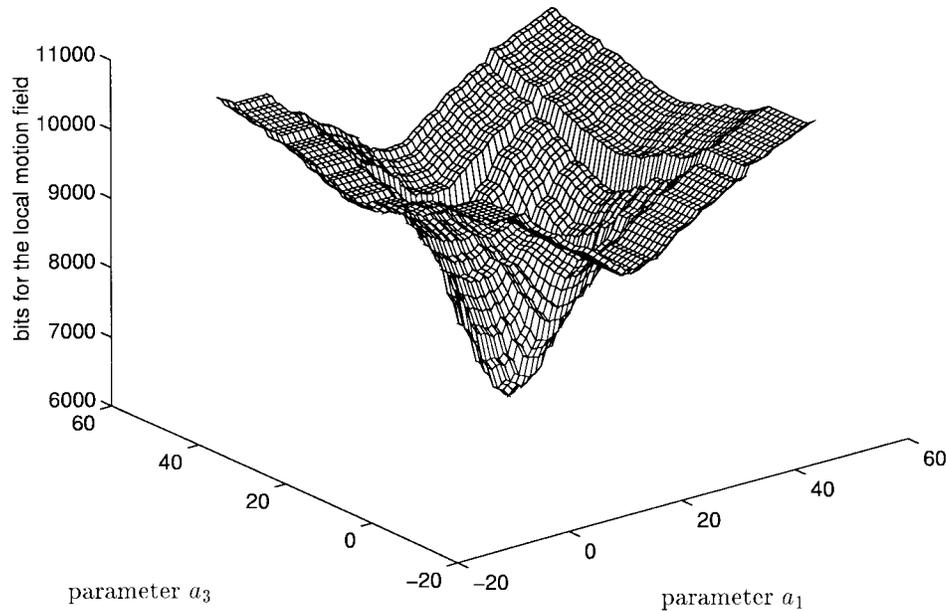


Fig. 1. Parameters  $a_1$  and  $a_3$  versus bit rate plot for motion field #40 of the Tennis sequence.

selected to be in this region, then a fast search can yield the global minimum instead of being caught in any local minimum. We can initialize the parameters using the values obtained by the least-squares estimation. We can also use the parameters of the previous frame as initialization since the parameters of two consecutive frames in a scene are observed to be highly correlated. It will avoid the computation required for the least-squares estimation, but it cannot be applied at scene changes.

### C. Global Motion Prediction Error Minimization (GMPEM)

In the motion compensated predictive coding framework, the compression performance depends on both the prediction error coding and the motion field coding. In a practical encoder, the prediction error is lossy-coded and the motion field is lossless-coded. Therefore the rate-distortion performance of the encoder depends on the quantization error in the lossy coder and the bit rates for the prediction error and the motion field. Ideally, the motion vectors should be estimated in such a manner that, for a given lossy coding scheme for the prediction error and a given lossless coding scheme for the motion field, the encoder has the best rate-distortion performance. This process requires a closed-loop approach with huge amounts of computation which makes it unsuitable for practical implementations. In the existing open-loop approach, only the prediction error energy is minimized with the implicit assumption that the bit rate for encoding the prediction error will be minimized.

Global motion affects both the motion field and the prediction error. The purpose of global motion estimation and compensation (explained in the next section) is to remove the effect of global motion from the present frame so that the transmitted prediction error and the motion field represent only the local motion information. Ideally, we should choose the global parameters such that, for a given lossy coding

scheme for the prediction error and a given lossless coding scheme for the motion field, the encoder has the best rate-distortion performance. Since this is not a feasible scheme, we can choose the global motion parameters such that the global motion prediction error energy is minimized. Here global motion prediction error refers to the error between the present frame and its prediction based on the previous frame and the global parameters. The prediction is obtained by transforming the previous frame using the global parameters. It is to be noted here that global motion prediction error minimization and motion field bit rate minimization are related to each other in the sense that the estimated parameter values are very close. Minimizing both the prediction error and the motion field bit rate is not required since the improvement in rate-distortion performance of the encoder may not be worth the extra computation.

Let  $Y_n$  and  $Y_{n-1}^r$  denote the current frame and the previous decoded frame, respectively. Let  $Y_n^G$  denote the predicted frame which is obtained from  $Y_{n-1}^r$  using the parameters  $a_1, a_2, a_3$ , and  $a_4$ , and bilinear interpolation. We call  $Y_n^G$  the globally compensated frame. The optimal values of  $a_1, a_2, a_3$ , and  $a_4$  can be found out by solving the following minimization problem:

$$\min_{a_1, a_2, a_3, a_4} \sum_i \sum_j (E_n(i, j, a_1, a_2, a_3, a_4))^2 \quad (35)$$

where

$$E_n(i, j, a_1, a_2, a_3, a_4) = Y_n(i, j) - Y_n^G(i, j, a_1, a_2, a_3, a_4). \quad (36)$$

$Y_n(i, j)$  and  $Y_n^G(i, j, a_1, a_2, a_3, a_4)$  denote the intensity values at the pixel  $(i, j)$  in the frames  $Y_n$  and  $Y_n^G$ , respectively, and  $E_n$  denotes the global prediction error frame. The ranges of  $i$  and  $j$  in the summation depend on the dimensions of the interpolated frame. The solution of this problem requires that

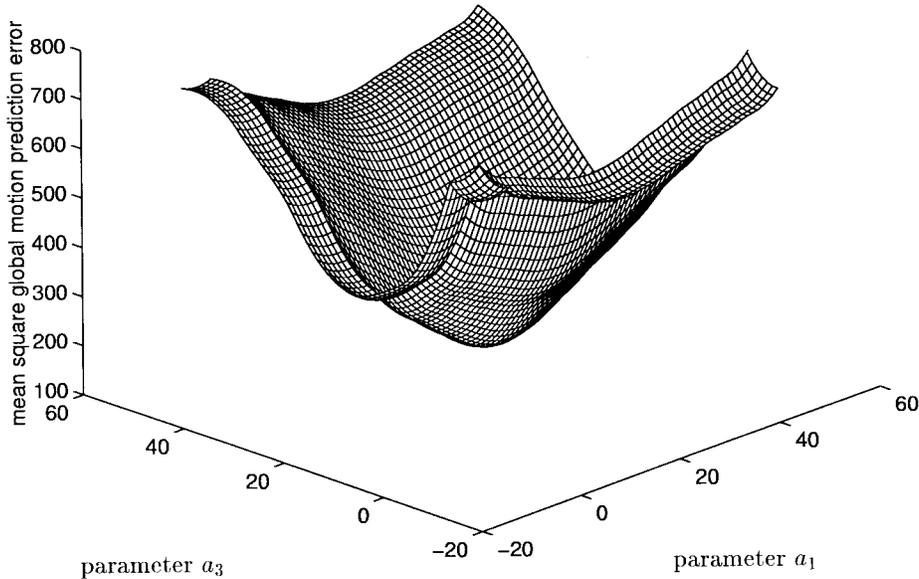


Fig. 2. Parameters  $a_1$  and  $a_3$  versus global motion prediction error plot for frame #40 of the Tennis sequence.

$Y_n^G$  be expressed as a differentiable function of  $a_1, a_2, a_3,$  and  $a_4$ . Since such a function is not available, the only way of obtaining the global minimum is to employ exhaustive search. For a four-dimensional parameter space, exhaustive search is extremely compute-intensive, and hence, practically infeasible. Experimental results in Fig. 2 show that the function  $\sum_i \sum_j E_n(i, j, a_1, a_2, a_3, a_4)^2$  is convex over a wide region near the global minimum. Therefore, like the previous method, we can use a fast search technique with initialization in the convex region. The initial values can be either the values obtained from the least-squares estimation, or the parameter values corresponding to the previous frame. To reduce the computation further, we can use a selected set of blocks (for example, the blocks with motion vectors matching with the global parameters), or we can even use a subsampled version of the frame to compute the prediction error.

#### IV. GLOBAL MOTION COMPENSATION

Once the global motion parameters are known, the encoder can remove the global motion field from the current motion field and transmit the remaining local motion field. This procedure can be followed for every motion field without affecting the prediction error energy. Thus, the only improvement over the conventional block matching algorithm based compression scheme will be in terms of the bit rate for the motion field.

Ideally, the local motion field should correspond only to the local moving objects. It should not only show their positions correctly but represent their motions accurately. The local motion field obtained as above often does not satisfy these conditions. As a result, the effect of global motion is not completely eliminated from the prediction error frame. This is because of several reasons: first, the conventional block matching algorithm assumes that all the pixels in a block have equal displacements. This is not true when the scene contains ZOOM. Depending on the size of the block, the pixels on the opposite edges of a block can have different displacements.

Second, the block matching algorithm uses finite resolution for the motion vectors. Third, because of the global motion, the apparent local motion may exceed the search range. These drawbacks can be eliminated by incorporating the global motion in the compensation. This will produce not only a more accurate local motion field but a prediction error frame corresponding only to the local motion, resulting in maximum compression.

The existing global motion compensation methods [4]–[6] all use the same procedure: motion compensation is performed in two stages. In the first stage, the global motion in the present frame is compensated. The global motion parameters are estimated from the motion field which is obtained after running the block matching algorithm with the present frame and the previous decoded frame. These parameters are used to construct a global motion compensated frame from the previous decoded frame. This frame is used as reference frame for local motion compensation in the second stage. Local motion compensation in the second stage is similar to the conventional motion compensation except that it uses a different reference frame. It produces a prediction error frame and a motion field called local motion field. As a result of this two stage method, the present frame is compressed by encoding the prediction error frame, the local motion field, and the global motion parameters. The method of Kamikura and Watanabe [5] compares the performances of conventional motion compensation and the two-stage compensation for each block. It uses one bit extra information per block to select the better scheme.

Our approach is also similar to the existing ones in the sense that we also perform motion compensation in two stages. The first stage compensates the global motion in the present frame using one of the four compensation schemes presented below. It produces an intermediate frame (IF) and an intermediate motion field (IMF). If the estimated global parameters are all null, then the intermediate frame is the same as the previous

decoded frame, and the intermediate motion field consists of null motion vectors. The second stage compensates the local motion in the present frame. It uses the block matching algorithm to find the local motion vectors. For that, it uses the intermediate frame as the reference frame and the intermediate motion field as the reference motion field. The reference motion field is the initialization motion field for the block matching algorithm. Searches for best matching blocks are performed about the initialized motion vectors. The second stage produces the prediction error frame and the local motion field which are transmitted to the receiver.

In the receiver, decoding of the present frame is done by first computing the intermediate frame and the intermediate motion field. The intermediate motion field is added to the decoded local motion field to obtain the total motion field. The motion compensated prediction frame is generated using the intermediate frame and the total motion field. The present frame is reconstructed by adding the decoded prediction error frame to this motion compensated prediction frame.

One of the drawbacks of the existing global motion compensation schemes is that they use the present frame to compute the global motion parameters. As a result, they execute the block matching algorithm two times for each frame: once for computing the parameters and once for local motion compensation. It is well known that the block matching algorithm is the most time-consuming part in motion compensation. Therefore the existing schemes are not practical for real-time applications. The alternative is to estimate the parameters from the previous motion field or the previous decoded frame. In real-life sequences with global motion, the parameters for two consecutive frames are highly correlated. This is because global motion changes very slowly from one frame to the next frame; in addition, the parameters are estimated from fixed-resolution motion vectors and then quantized. Therefore the performance degradation due to the delay of one frame is very low. In this case, the estimation of the parameters can be carried out by the receiver, and therefore the parameters need not be transmitted.

Based on the above concepts, we obtain the following two global motion compensation schemes.

#### A. Forward Global Motion Compensation

We call it forward global motion compensation when the present frame is involved in the computation of the global motion parameters. In this case, the parameters are transmitted to the receiver. The parameters can be fixed-length coded or variable-length coded.

#### B. Backward Global Motion Compensation

We call it backward global motion compensation when the parameters are computed from the previous motion field or the previous decoded frame. In this case, the parameters are not transmitted. The receiver also performs global motion estimation.

Depending on the output of the first stage, each of the above compensation schemes can be categorized into the following two.

#### C. Block-Based Compensation

In this method, the intermediate frame is the same as the previous decoded frame. The intermediate motion field is computed as follows.

- 1) The frame is segmented into square blocks. The size of the blocks is the same as used for the block matching algorithm in the second stage.
- 2) The motion vector of the center pixel of each block is computed by substituting the estimated parameter values and its coordinates in the model equation (13). These motion vectors are then quantized to required resolution.
- 3) These motion vectors represent the global motion vectors of the corresponding blocks. If the motion vector of a block is such that it points to a matching block which does not lie completely inside the previous decoded frame, its value is set to null.

In this case, the intermediate motion field is called the global motion field. This compensation scheme initializes the local motion estimation with the global motion vector. Therefore, the local motion, provided it is within the preset search range, is better compensated. But this scheme has the disadvantage that all the pixels in a target block are forced to have equal fixed-resolution displacements with respect to the previous frame.

#### D. Pixel-Based Compensation

In this method, the intermediate motion field is set to null motion vectors. The intermediate frame is computed as follows for each pixel.

- 1) The global motion vector is computed by substituting the estimated parameter values and its coordinates in the model equation (13).
- 2) The position of the pixel is displaced by this motion vector.
- 3) If the displaced position lies outside the frame, the pixel intensity value is copied from the same pixel position in the previous decoded frame. Else, using the four neighbor pixels in the previous decoded frame, bilinear interpolation is performed to find the pixel intensity.

In this case, the intermediate frame is called the global motion compensated frame. Like block-based compensation, this compensation scheme better compensates the local motion provided it is within the preset search range. In addition, during ZOOM, it lets the pixels in a target block have linear motion with respect to the previous frame. Note that the decoder computes the global motion vector of each pixel using the global motion parameters. Therefore there are no additional bits required for transmitting these vectors.

As a result, we obtain four compensation schemes. We call them pixel-based forward global motion compensation (PFGMC), block-based forward global motion compensation (BFGMC), pixel-based backward global motion compensation (PBGMC), and block-based backward global motion compensation (BBGMC), respectively. All the existing global motion compensation schemes are PFGMC. The modifications of Kamikura and Watanabe [5] can also be applied to BFGMC.

## V. MOTION VECTOR CODING

The amount of compression in a two-stage scheme as described above clearly depends on the coding efficiency of the prediction error frame, the local motion field, and additionally, the global motion parameters if the global motion compensation is forward. In this section, we will present a scheme for motion vector coding [11] which is not only very efficient but most suitable for the estimation based on motion field bit rate minimization.

Motion vectors have to be lossless-encoded. Depending on the amount of local motion in the scene, the bandwidth required for them can be a significant portion of the total required bandwidth. It has been experimentally found [12] that the amount of bits transmitted for the motion field can sometimes reach up to 40% of the total transmitted bits. The percentage can be even more in very low bit rate applications. There have been mainly two kinds of motion vector coding schemes in the literature: fixed-length coding (FLC) and entropy coding. The FLC is inefficient because it does not consider the statistics of the motion vectors. The entropy coding improves upon this by finding the probability of the motion vectors and thus optimizing the code length. Koga and Ohta [13] have considered the following three schemes: two-dimensional entropy coding, one-dimensional entropy coding with two codebooks, and one-dimensional entropy coding with one codebook. In the first scheme each motion vector is assigned one codeword from a set of codewords. The second scheme entropy codes the two scalar components of the motion vector separately, representing each motion vector with two codewords effectively. The third scheme uses one common codebook for the two components of the motion vector. They have shown that the two-dimensional entropy coding gives the best performance as far as the bit rate is concerned. Choi and Park [12], besides the above three schemes, have considered three more similar schemes applied to the difference motion vectors. Schiller and Chaudhuri [14] have applied the last scheme of Koga and Ohta [13] to difference motion vectors obtained by spatial prediction. The disadvantages of the two-dimensional differential entropy coding scheme are: 1) the codebook size is very large which increases the complexity and 2) since this is a differential coding scheme, it cannot be used for finding the parameters in the second estimation method. For this purpose, we need a variable-length coding scheme in which the length of the code depends on the magnitude of the motion vector. We present such a coding scheme in the following.

The set of possible motion vectors for a block depends on the dimension of the search area used for the matching process. If the search area is given by  $\pm w \times \pm w$  (we assume a square search area), then the set of possible motion vectors  $\mathbf{S}$  is given as

$$\mathbf{S} = \{(w, w), (w, w-1), \dots, (w, -w), \dots, (-w, w), (-w, w-1), \dots, (-w, -w)\}. \quad (37)$$

Let  $\mathbf{v} \equiv (v_x, v_y) \in \mathbf{S}$ . We can denote the chessboard distance of  $\mathbf{v}$  as

$$d_{cb}(\mathbf{v}) = \max(|v_x|, |v_y|). \quad (38)$$

TABLE I  
BIT ALLOCATION ACCORDING TO THE CHESSBOARD DISTANCE FOR  $w = 15$

$i$	$ S_i $	# of bits	$i$	$ S_i $	# of bits
0	1	1	8	64	11
1	8	8	9	72	12
2	16	9	10	80	12
3	24	10	11	88	12
4	32	10	12	96	12
5	40	11	13	104	12
6	48	11	14	112	12
7	56	11	15	120	12

Let  $S_i = \{\mathbf{v} \in \mathbf{S}: d_{cb}(\mathbf{v}) = i\}$ ,  $i = 0, 1, 2, \dots, w$ . It is easily seen that  $S_i$ 's constitute a partition of  $\mathbf{S}$  and  $\|S_0\| = 1$ ,  $\|S_i\| = 8i$  for  $i = 1, 2, \dots, w$ , where  $\|S_i\|$  denotes the cardinality of set  $S_i$ . Therefore given a vector  $\mathbf{v} \in S_i$ ,  $i \neq 0$ , we need  $\lceil \log(8i) \rceil$  bits to specify it within  $S_i$ .  $S_i$ 's can be either fixed-length coded or entropy coded. For the fixed-length case, it needs  $\lceil \log(w+1) \rceil$  bits to code an  $S_i$ . Considering the case that most of the motion vectors are null (typically with probability  $>0.5$ ), we can have a simple variable-length coding scheme (SVLC) for  $S_i$ 's: we code  $S_0$  with one bit and  $S_i$ ,  $i \neq 0$ , with  $1 + \lceil \log w \rceil$  bits. Therefore to code a motion vector  $\mathbf{v}$ , effectively, it requires one bit if  $\mathbf{v} \in S_0$  and  $1 + \lceil \log(8i) \rceil + \lceil \log w \rceil$  bits if  $\mathbf{v} \in S_i$ ,  $i \neq 0$ . If  $S_i$ 's are entropy coded then  $\mathbf{v}$  is coded by  $\mathcal{L}(S_i) + \lceil \log(8i) \rceil$  bits where  $\mathcal{L}(S_i)$  is the entropy code length of  $S_i$ .

Table I shows bit allocation with  $S_i$ 's SVLC coded for  $w = 15$ . In this case, the FLC scheme requires 10 bits per motion vector. It is clear from the table that higher compression can be achieved only if the  $S_i$ 's with lower  $i$  are more probable than those with higher  $i$ . On the contrary, the performance will be worse if  $S_i$ 's with larger  $i$  are more probable. Experimental results in Fig. 3 show that in real video sequences without global motion, short distance motion vectors are more probable. Therefore the above coding scheme can be applied to the motion vectors giving higher compression. The other advantages of the above scheme are that it is very simple; it can be implemented without a table lookup since the motion vectors can be mapped to fixed codewords using a preset rule. The codebook need not be transmitted when  $S_i$ 's are SVLC coded. When  $S_i$ 's are entropy-coded, the codebook size is very small. Therefore the complexity is very low. It is also suitable for use in the estimation criterion in (34).

## VI. EXPERIMENTAL RESULTS

To verify the efficiency and to compare the performances of the proposed global motion estimation and compensation schemes, simulations were performed over 201 frames of the Tennis sequence. The sequence is interlaced, 30 frames per second with "Y" frame size  $704 \times 480$ . We considered only the even fields with the width down-sampled to 352 pixels. The sequence consists of three scenes with scene 1 spanning from frame 0 (starting frame) to frame 88, scene 2 spanning from frame 89 to frame 147, and the last scene occupying the frames 148 through 200. The first scene contains zoom (frames

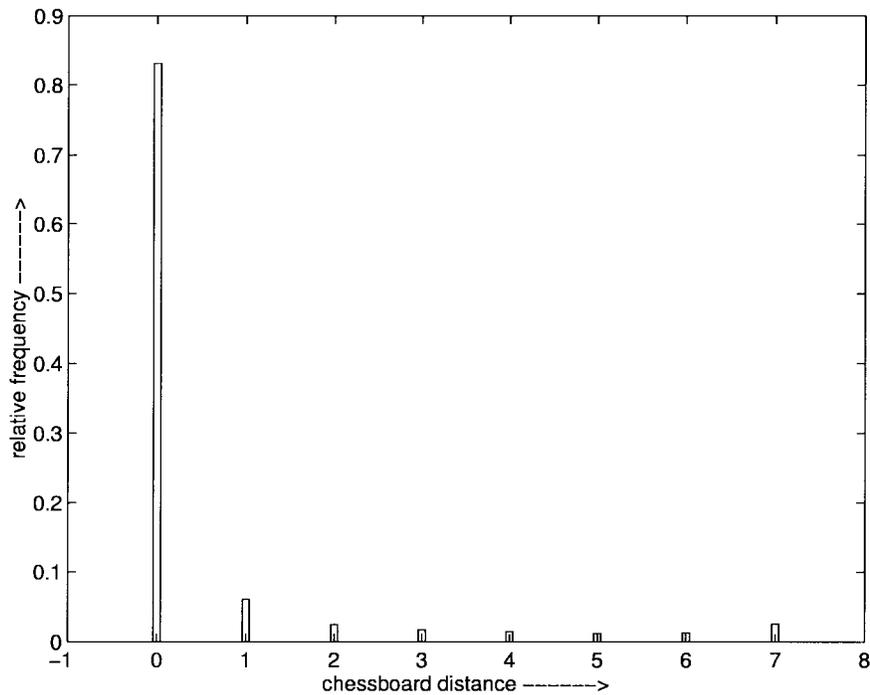


Fig. 3. Relative frequencies of motion vector chessboard distances during the normal motion in Tennis sequence. Search area is  $\pm 7 \times \pm 7$ .

24–88) together with normal motion. Similarly the last scene includes pan (frames 177–200) along with normal motion. The second scene has only normal motion. By normal motion, we mean local object motion without the camera motion. For the block matching algorithm, the block size and the search region were chosen to be  $8 \times 8$  and  $\pm 7 \times \pm 7$ , respectively. The motion vectors were computed using exhaustive search and MSE matching criterion. For easy readability, the following abbreviations are used in this presentation: CMC for conventional motion compensation, ILSE for iterative least-squares estimation, MFBRM for motion field bit rate minimization, GMPEM for global motion prediction error minimization, PFGMC for pixel-based forward global motion compensation, BFGMC for block-based forward global motion compensation, PBGMC for pixel-based backward global motion compensation, and BBGMC for block-based backward global motion compensation.

In the first part of the simulation, we compared the proposed three global motion estimation schemes with Tse and Baker's [4] least-squares estimation scheme and Kamikura and Watanabe's [5] histogram-based estimation scheme. For comparison, we performed PFGMC with each estimation scheme and considered the global motion prediction error as the performance measure. In the ILSE estimation scheme, iteration was performed till convergence. Here, by convergence, we mean that the estimated parameter values in two successive iterations are identical. It was observed that the convergence is achieved in a very few, typically less than five, iterations in all the frames. The motion vector matching threshold was chosen to be one for all the iterations. The MFBRM and GMPEM estimation schemes were simulated using three-step searches with step sizes 4, 2, and 1, respectively. These searches were initialized with the parameter values obtained after the first

iteration in the ILSE scheme. In all the three estimation schemes, the parameters  $a_1$  and  $a_3$  were quantized with step size  $1/1024$ , and  $a_2$  and  $a_4$  were quantized with step size 1. Tse and Baker's [4] least-squares estimation was iterated twice with motion vector matching threshold 2. In Kamikura and Watanabe's [5] estimation scheme, first, histograms of the parameters were generated by computing the parameter values for each pair of blocks symmetrically located with respect to the center of the frame. Then the values corresponding to the peaks in the histograms were taken to be the estimated values of the parameters. For both these schemes, the parameter  $a_1$  was quantized with a step size  $1/1024$ , and  $a_2$  and  $a_4$  were quantized with step size 1. Note that both Tse and Baker [4], and Kamikura and Watanabe [5] have three-parameter models for the global motion with  $a_1 = a_3$ .

The mean squared global prediction error plots for Tse and Baker's [4] (T & B) estimation scheme, Kamikura and Watanabe's [5] (K & W) estimation scheme, and the proposed three schemes are shown in Figs. 4–6. We have displayed the plots for the three scenes separately for the sake of clarity. For the same reason, we also have displayed the plots for each scene in two separate figures wherever needed. The global prediction error for the conventional motion compensation is nothing but the difference frame energy. We notice that all the schemes estimate null parameter values in the normal motion regions (frames 1–23 in scene 1, all frames in scene 2, and frames 149–176 in scene 3). Therefore, they are equivalent to the conventional motion compensation during normal motion. In the zoom region, Kamikura and Watanabe's [5] scheme, being extremely sensitive to the noise, has the worst performance. The performance of Tse and Baker's [4] scheme is similar to that of ILSE. The performances of the proposed three schemes are also very similar; GMPEM

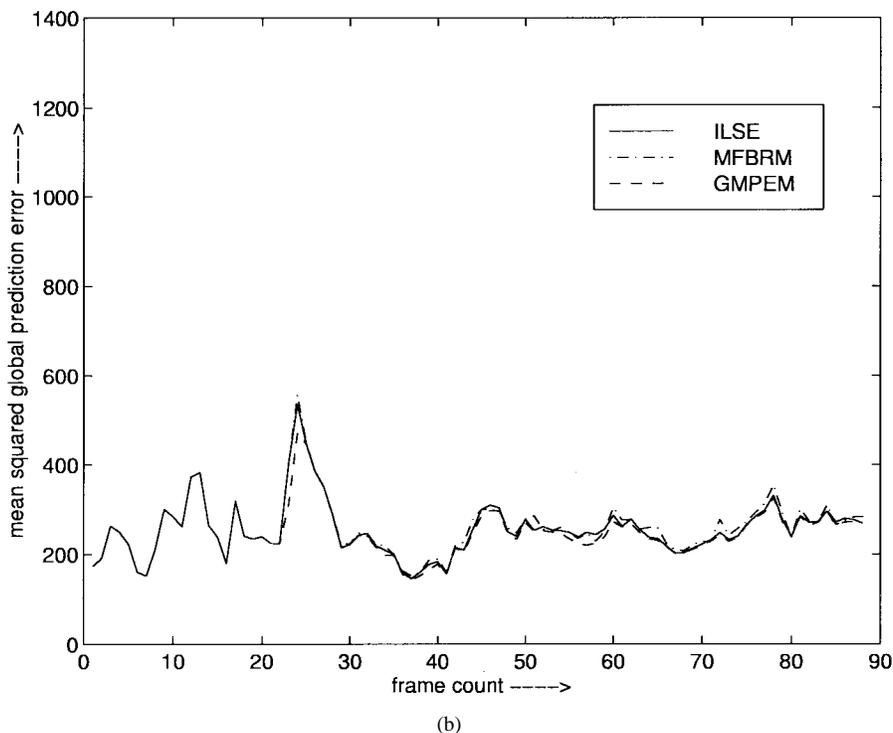
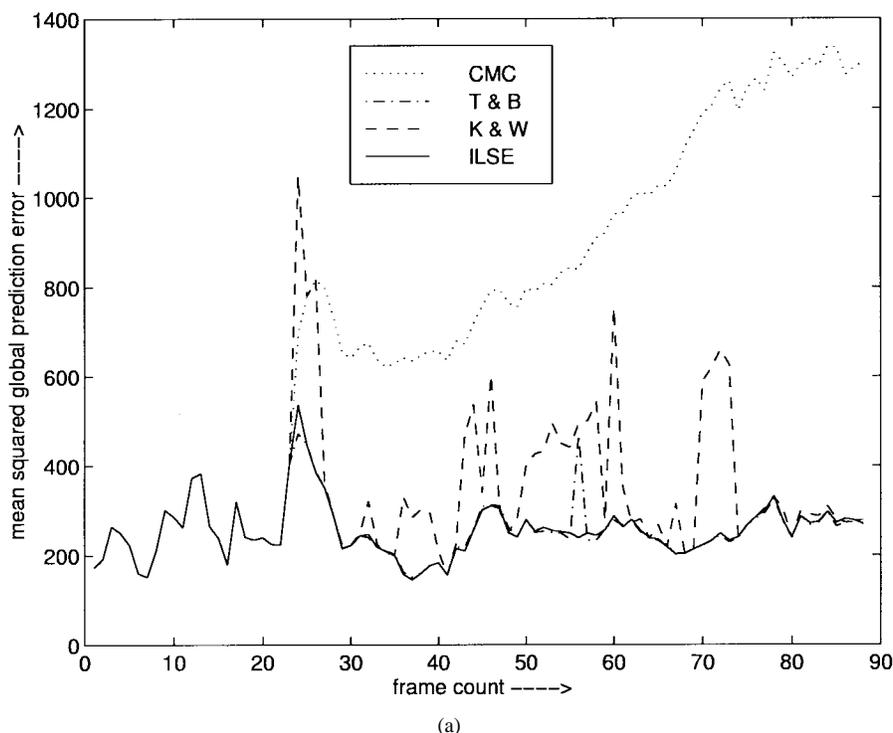


Fig. 4. Mean squared global prediction error for scene 1 with PFGMC compensation.

performs the best because of its estimation criterion. In the pan region, except Kamikura and Watanabe’s [5] scheme, all other global motion estimation schemes have identical results. Table II displays the average mean squared global prediction errors for the entire sequence (including the scene changes) as well as the zoom and the pan durations. We observe that PFGMC with all the estimation schemes reduces the difference frame energy significantly in the zoom and the pan regions.

We also observe that ILSE, though it does not use GMPEM’s estimation criterion, has negligible performance degradation compared to the latter in the zoom region.

Figs. 7–10 show the estimated values of the parameters  $a_1$ ,  $a_2$ ,  $a_3$ , and  $a_4$ , respectively. For Tse and Baker’s [4] and Kamikura and Watanabe’s [5] estimation schemes  $a_1 = a_3$ . The peaks at the eighty-ninth and one hundred forty-eighth frames in all the plots are because of the scene change at

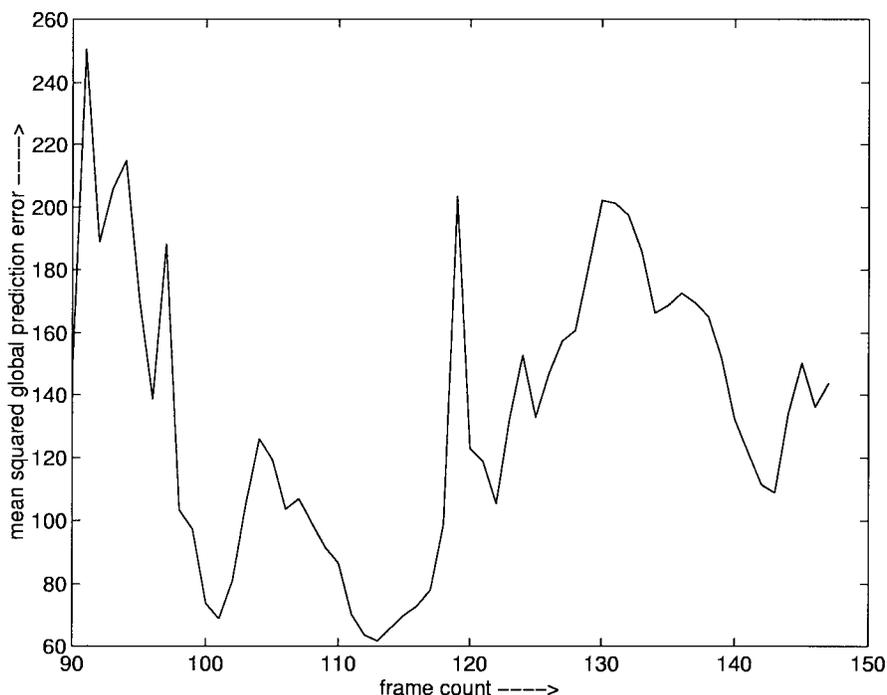


Fig. 5. Mean squared global prediction error for scene 2 with PFGMC compensation. All the global motion estimation schemes produce identical results due to null parameters.

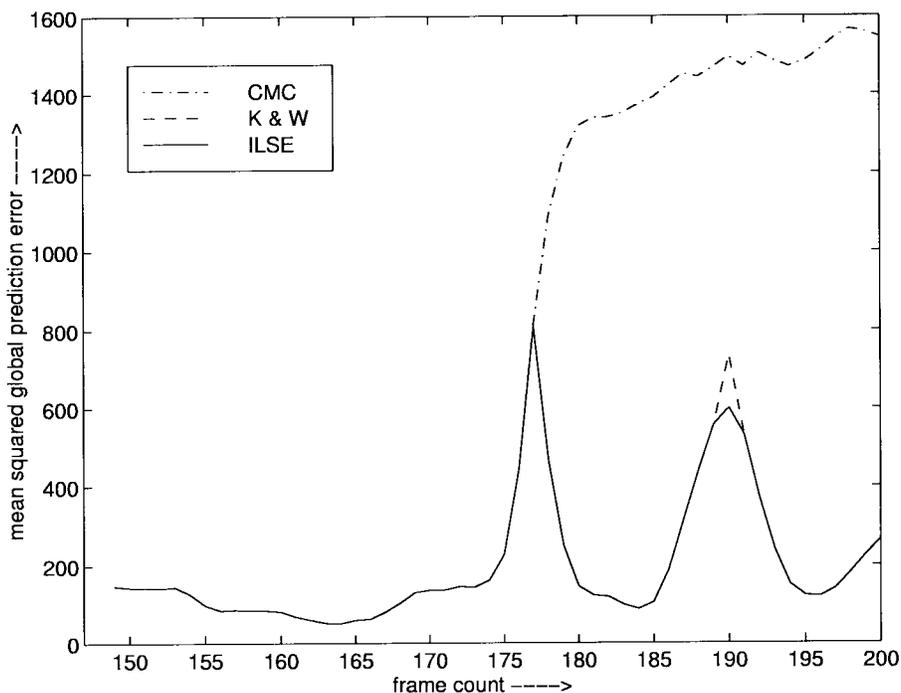


Fig. 6. Mean squared global prediction error for scene 3 with PFGMC compensation. Tse and Baker's scheme, MFBRM and GMPEM produce the same results as with ILSE.

those frames. The parameter plots for some schemes ( $a_2$  and  $a_4$  for Tse and Baker's scheme,  $a_4$  for MFBRM) are not displayed because of the difficulty in distinguishing them. We observe that Kamikura and Watanabe's [5] scheme produces wrong estimates for the parameters  $a_2$  and  $a_4$  for many frames in camera zoom. With the same quantization step size 1, other schemes, except GMPEM, and Tse and Baker's [4]

scheme for a few frames, produce null estimates for both  $a_2$  and  $a_4$ . Since the three-step search process leads to a local minimum, GMPEM produces erroneous estimates for  $a_1$  and  $a_2$  at frames 57–59. Tse and Baker's scheme gives a wrong estimate of  $a_4$  at frame 57 because of only two iterations. With converging iterations, ILSE produces the most stable and reliable estimates for all the parameters. Since the ZOOM is

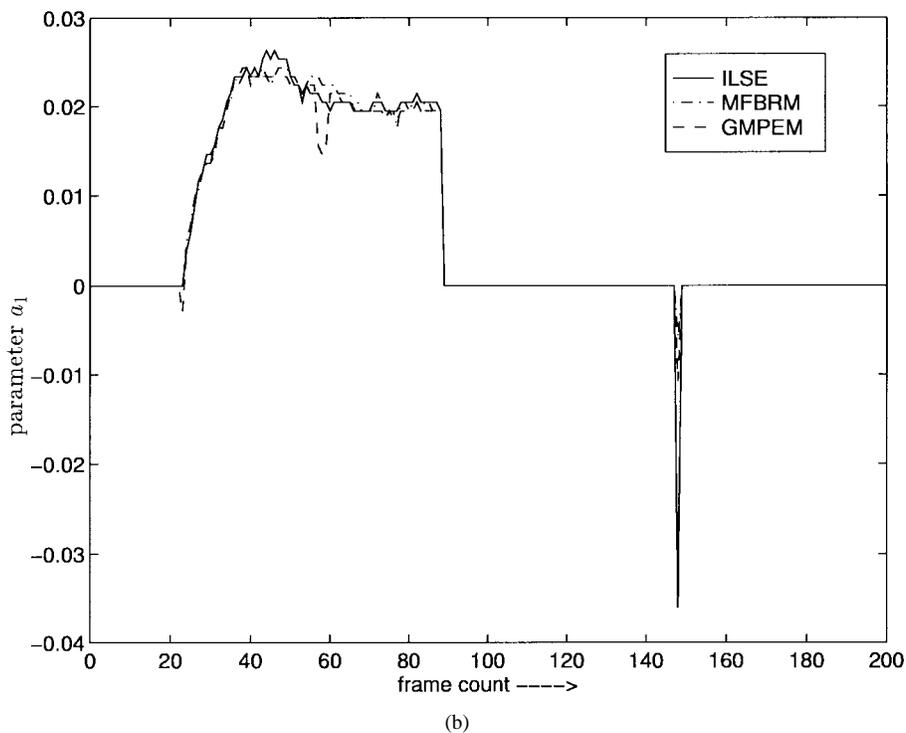
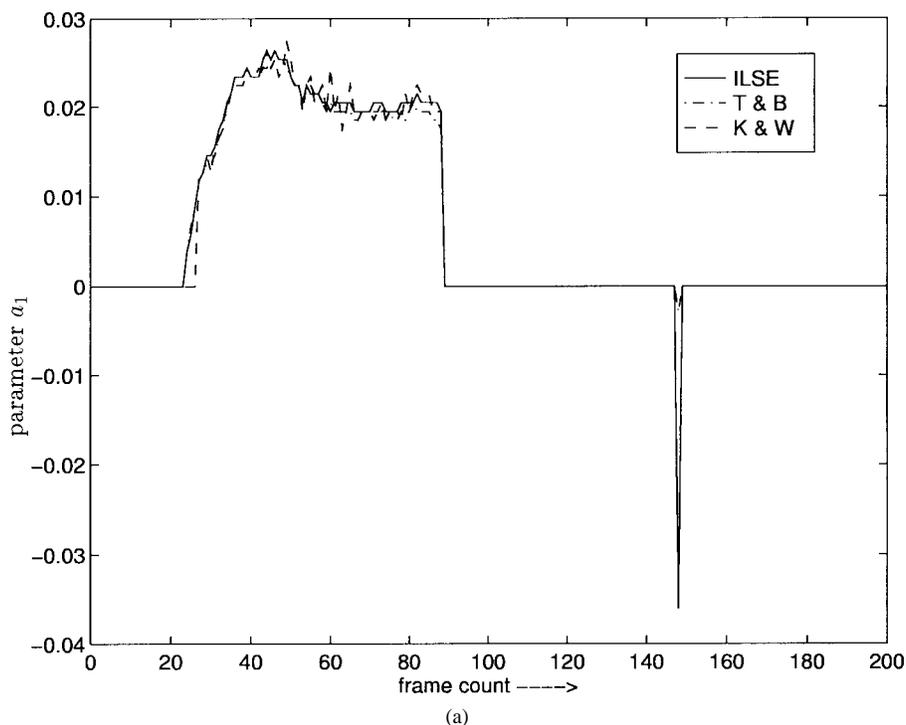


Fig. 7. Estimated value of parameter  $a_1$ .

TABLE II  
AVERAGE MEAN SQUARED GLOBAL PREDICTION  
ERROR FOR VARIOUS ESTIMATION SCHEMES

scheme	entire sequence	zoom	pan
CMC	594.497	939.789	1403.30
Tse & Baker	236.633	258.139	277.057
Kamikura & Watanabe	274.011	366.590	282.693
ILSE	234.801	256.513	277.057
MFBRM	237.628	262.496	277.057
GMPEM	231.979	249.970	277.057

because of camera motion, the estimated values of  $a_1$  and  $a_3$  are very close. The discrepancy is because of the local object motion and the rectangular shape of the frame. Positive values of parameters  $a_1$  and  $a_3$  indicate zoom-out of the camera. It is also observed that all the estimation schemes, except Kamikura and Watanabe's [5] scheme for one frame, produce correct parameters during the entire pan region, i.e.,  $a_1 = 0$ ,  $a_2 = 1$ ,  $a_3 = 0$ , and  $a_4 = 0$ . Moreover, the plots justify our assumption that the parameters of two successive frames are highly correlated.

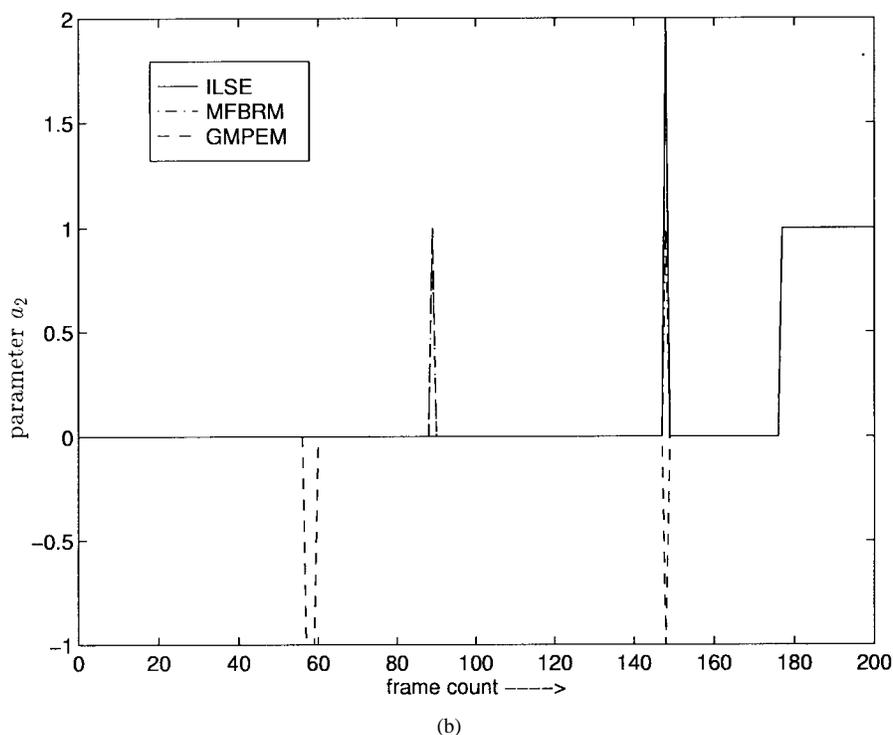
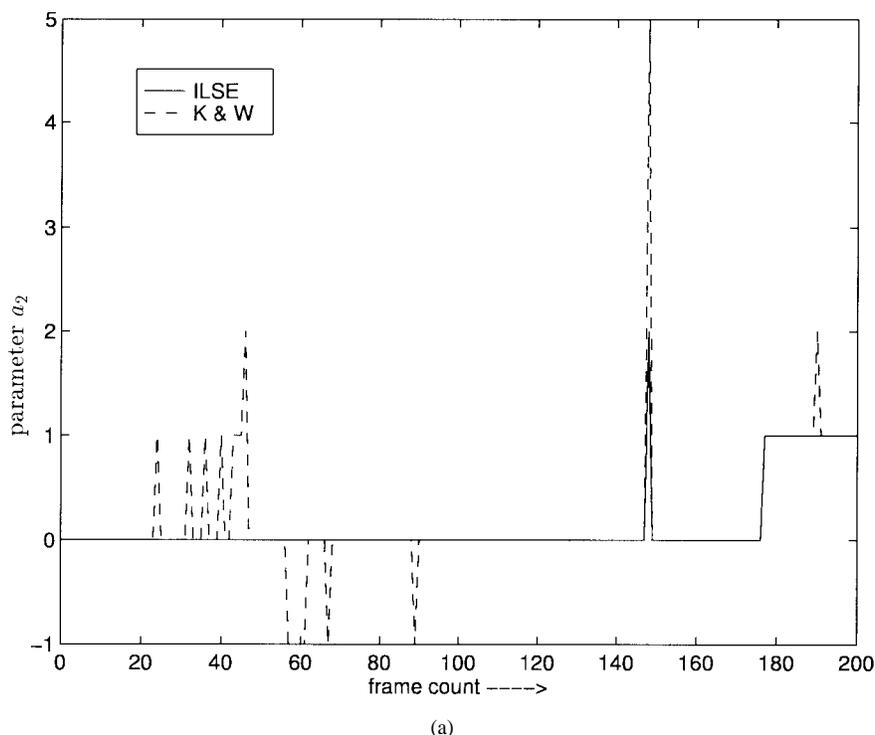


Fig. 8. Estimated value of parameter  $a_2$ . Tse and Baker's scheme produces the same parameter values as with ILSE except at frame 148.

In the second part of the simulation, we compared the performances of the four global motion compensation schemes explained in Section IV. Because of its similar performance to GMPEM, accuracy of the estimates, and lower computational complexity, we chose ILSE as the estimation criterion. As we have mentioned earlier, for the backward compensation schemes, the parameters of the previous motion field are used to compensate the global motion in the present

frame. For the next frame, the parameters are updated based on the received motion field. In the pixel-based backward compensation, the parameters estimated from the received motion field are added to the current parameters. In the block-based backward compensation, a new set of parameter values are estimated from the effective motion field, i.e., the motion field obtained by adding the received motion field to the global motion field generated using the current

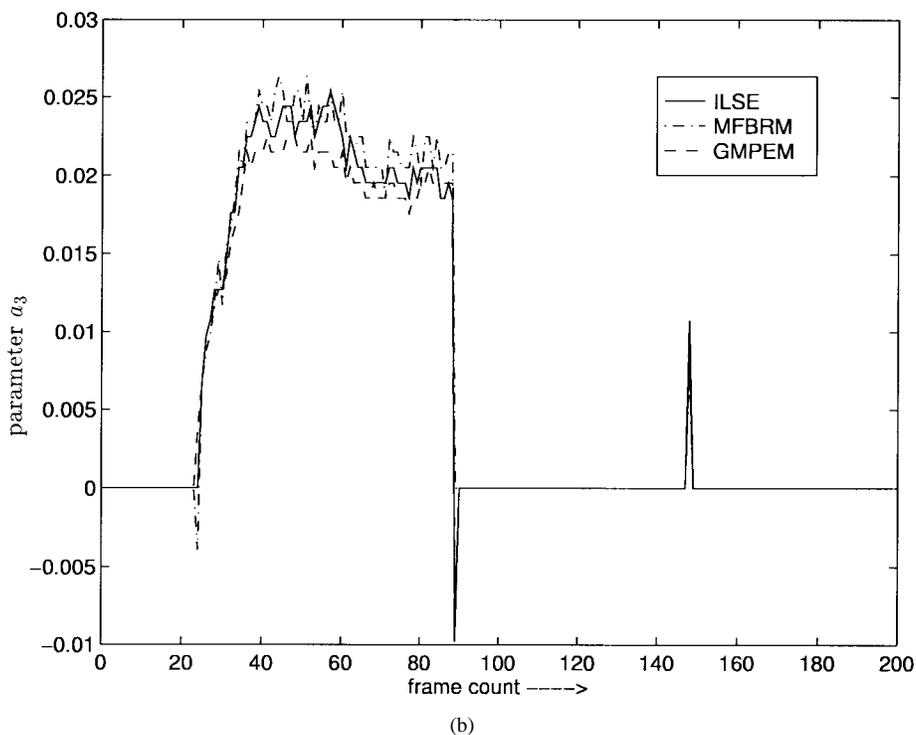
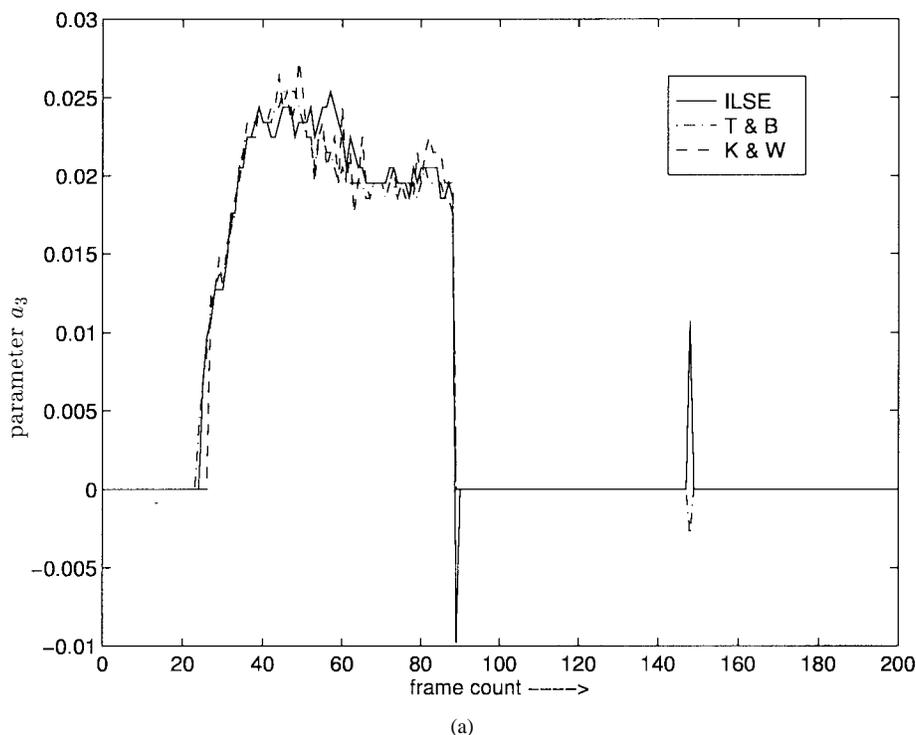


Fig. 9. Estimated value of parameter  $a_3$ .

parameters. This is done to increase the accuracy of the estimates.

The mean squared global prediction error for scenes 1–3 are plotted in Figs. 11–13, respectively. We observe that the pixel-based compensation schemes are better than the corresponding block-based schemes in both zoom and pan regions. The improvement is more pronounced in the zoom region than in the pan region. This is because the performance improvement

in the pan region is due to better compensation for pixels only in the border blocks, i.e., the blocks lying at the frame border, whereas in the zoom region, it is because of better compensation for pixels in all the blocks. We also observe that the backward compensation schemes perform almost similarly the corresponding forward compensation schemes except for having one frame delay degradation. In the pan region, the performance degradation occurs only at the transition frame

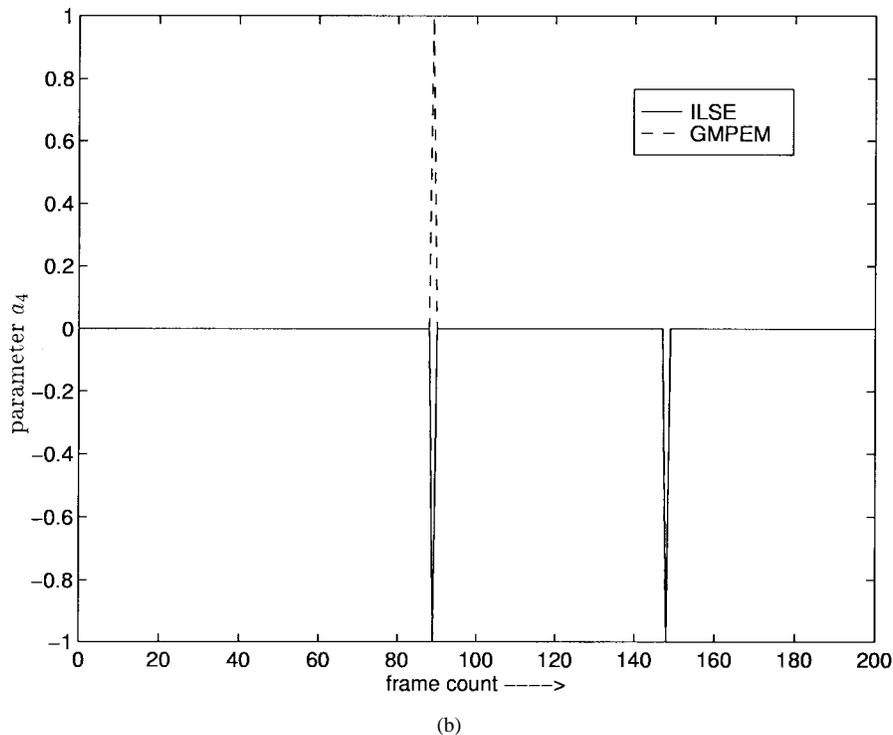
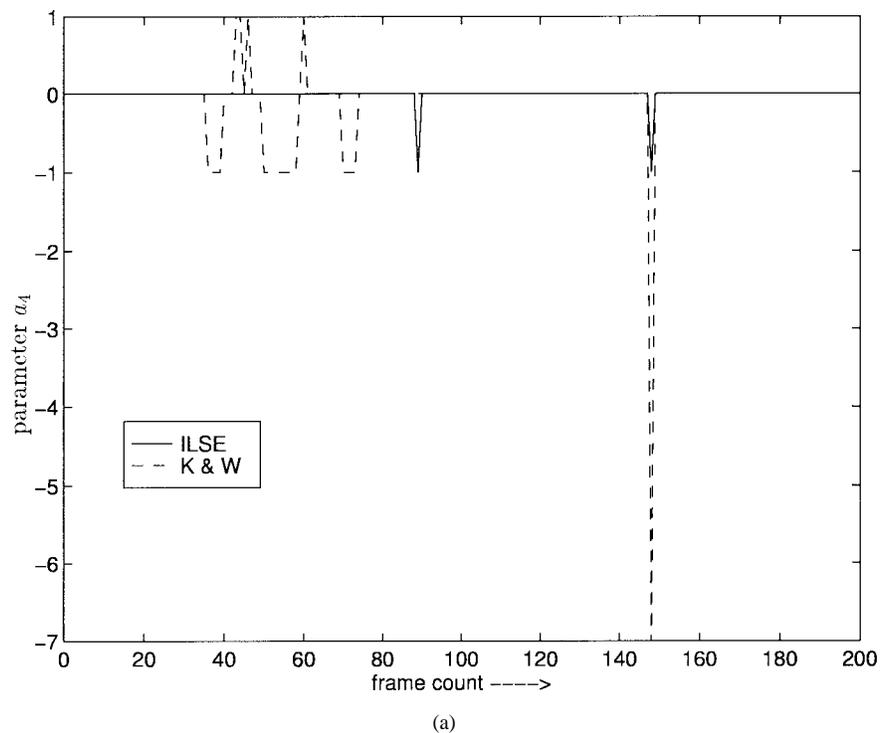


Fig. 10. Estimated value of parameter  $a_4$ . Tse and Baker's scheme produces the same parameter values as with ILSE except at frames 57, 89, and 148. MFBRM produces the same parameter values as with ILSE except at frame 148.

(i.e., when the normal motion changes to pan) since the PAN parameters remain constant throughout the pan. In the zoom region, the degradation is more observable at the transition frame than during the zoom. In fact, at some of the frames during zoom motion, the backward scheme performs better than the forward scheme. The average mean squared errors for the entire sequence (including the scene changes) as well

as the zoom and the pan durations are shown in Table III. As expected, the backward compensation methods show poorer results than the corresponding forward compensation methods; but the performance degradations are very small.

Fig. 14 shows the estimated values of the parameters  $a_1$ ,  $a_2$ ,  $a_3$ , and  $a_4$ . The plots corresponding to the forward compensation schemes (i.e., PFGMC and BFGMC) are

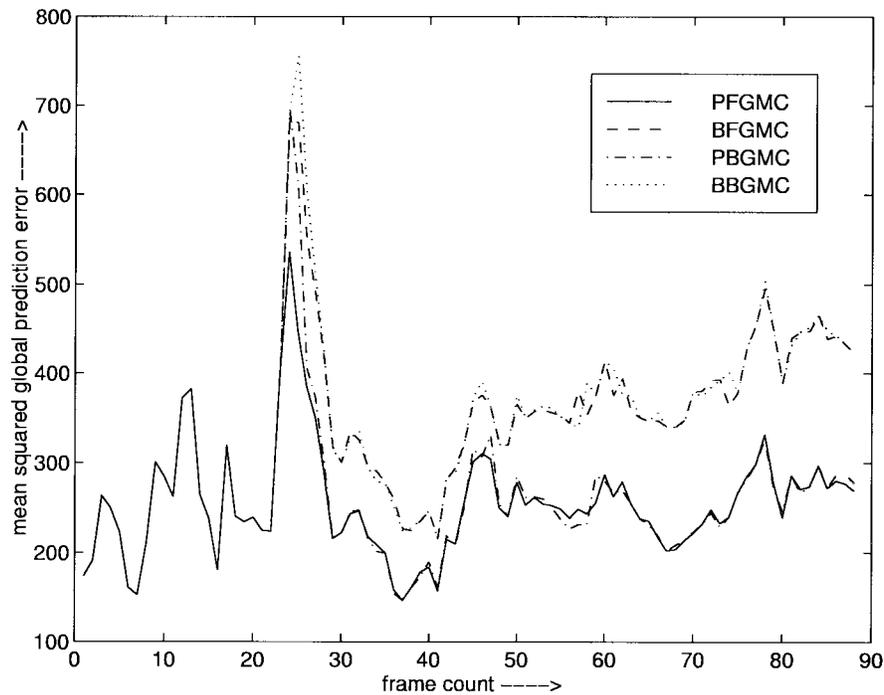


Fig. 11. Mean squared global prediction error for scene 1 with ILSE estimation.

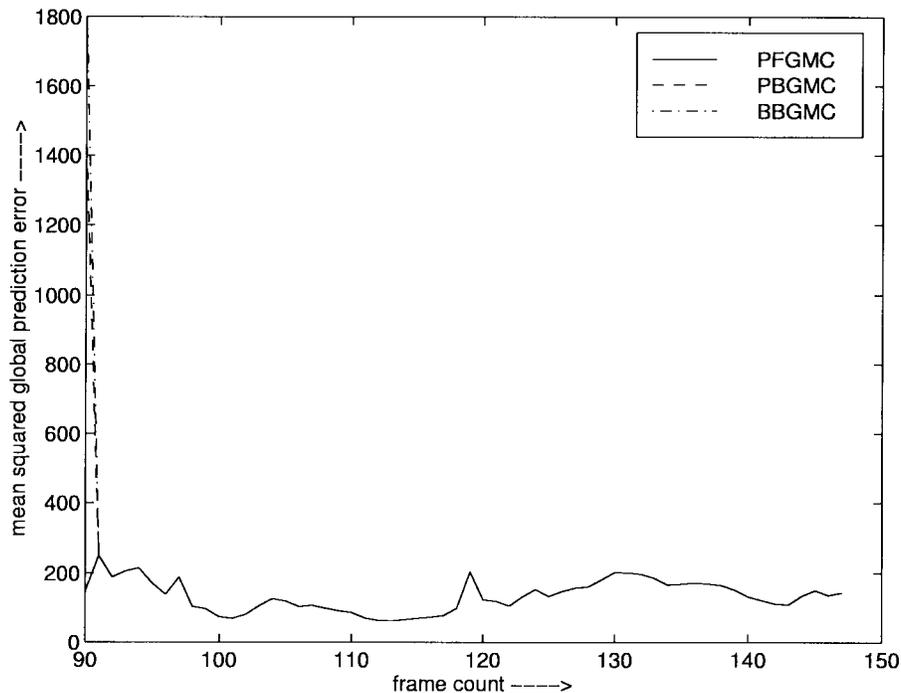


Fig. 12. Mean squared global prediction error for scene 2 with ILSE estimation. The plots are identical except at the starting frame. BFGMC produces the same results as with PFGMC.

identical and they are the same as the plots corresponding to ILSE in Figs. 7–10. The plots corresponding to the backward compensation schemes (i.e., PBGMC and BBGMC) are different from these since the parameters are estimated from different motion fields as mentioned earlier. Since the plots due to BBGMC and PBGMC are very similar, for clarity, we do not show the results for BBGMC. We observe that

the parameter values in normal motion and pan regions are identical for all the four schemes except one frame delay for the backward compensation schemes. In the zoom region, they are comparable with one frame delay for the backward compensation schemes.

To compare the effective performances of the two-stage compensation schemes and the conventional one-stage motion

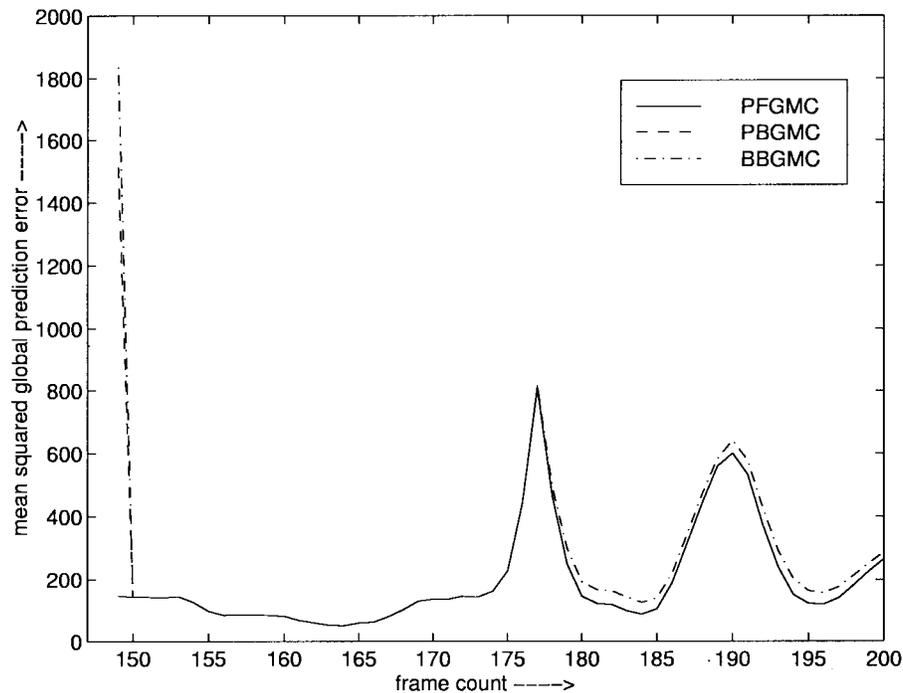


Fig. 13. Mean squared global prediction error for scene 3 with ILSE estimation. PFGMC and PBGMC are identical except at the starting frame. BBGMC has the same performance as with BBGMC except at the starting frame.

TABLE III  
AVERAGE MEAN SQUARED GLOBAL PREDICTION  
ERROR FOR VARIOUS COMPENSATION SCHEMES

scheme	entire sequence	zoom	pan
CMC	594.497	939.789	1403.30
ILSE-PFGMC	234.801	256.513	277.057
ILSE-BFGMC	279.574	373.306	313.539
ILSE-PBGMC	251.417	262.781	277.381
ILSE-BBGMC	297.452	377.065	313.983

compensation, we performed local motion estimation and compensation in addition to global motion compensation. Among the proposed compensation schemes, we selected ILSE-PFGMC because of its superior performance over others. But it is to be noted that, if computation is a constraint, ILSE-PBGMC can be selected instead. The mean squared local prediction error and the average bits per motion vector for scenes 1–3 are plotted in Figs. 15–17, respectively. The motion vectors with two-stage compensation schemes were encoded using the coding scheme presented in Section V. Since this coding scheme is not suitable for motion fields with high relative frequencies for large chessboard distance motion vectors, which is the case during camera zoom, the motion fields for CMC, for simplicity, were FLC coded with 8 bits per motion vector. Considering the mean squared error plots, we observe that two-stage compensation schemes perform better than the conventional motion compensation in the zoom and the pan regions on the average. In some frames, during zoom, the conventional motion compensation performs better than the other schemes. This is because, besides the change due to motion, zoom causes additional intensity changes which we

have not taken into account. The performance improvement in the pan region is due to better compensation for pixels in the border blocks. Considering the motion field bit rate plots, we observe that Tse and Baker [4] and the proposed ILSE-PFGMC compensation schemes perform significantly better than the CMC in the zoom and the pan regions. Kamikura and Watanabe's [5] estimation scheme produces wrong parameter estimates at those frames in which the bit rate for its motion field is relatively very large compared to Tse and Baker [4] and ILSE-PFGMC. The performance improvement in the normal motion region is because of the proposed motion vector coding scheme, not because of the global motion compensation. Note that we have not included the ON/OFF control mechanism suggested by Kamikura and Watanabe [5] to improve the compensation.

Tables IV and V display the average mean squared local prediction errors and the average bits for motion vector, respectively, for the entire sequence (including the scene changes) as well as the pan and the zoom durations. We observe that all the two-stage compensation schemes perform better than the CMC in the zoom and the pan regions both in terms of the mean squared error and the motion field bit rate. Although their performances are comparable, ILSE-PFGMC performs the best. Another observation is that the improvement over the conventional motion compensation is much more effective for the motion field bit rate than for the prediction error. Considering the ILSE-based compensation, we obtain 37.09 and 71.9% savings in bit rate over the conventional motion compensation for the zoom and the pan regions, respectively.

To improve the coding of the motion field, we Huffman-coded the chessboard distance of the motion vectors. The

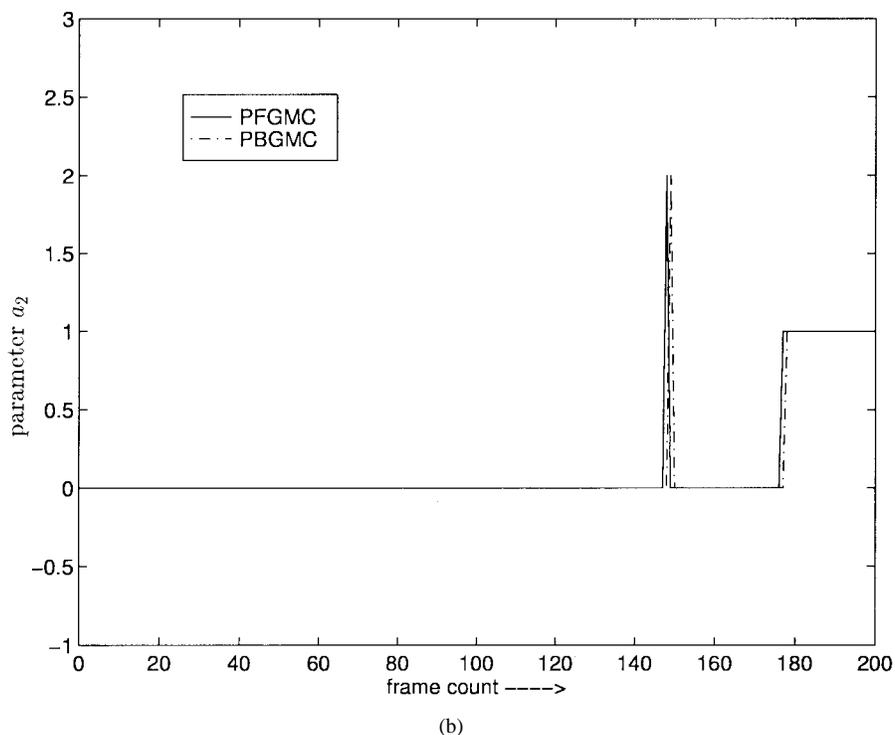
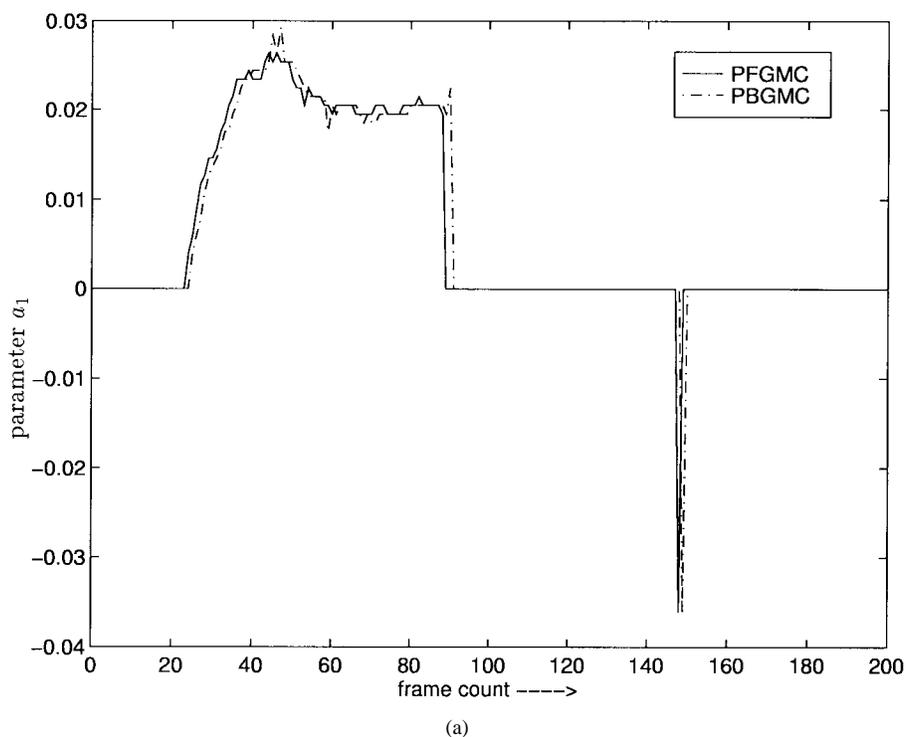
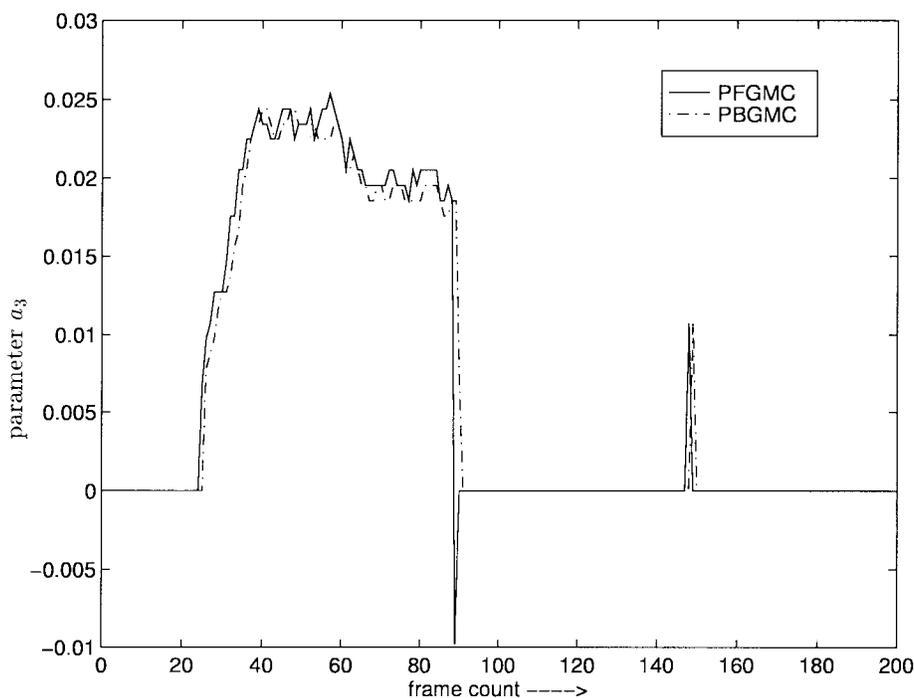


Fig. 14. Estimated value of global parameters for various compensation schemes. PFGMC and BFGMC have identical results. BBGMC produces similar results as with PBGM.

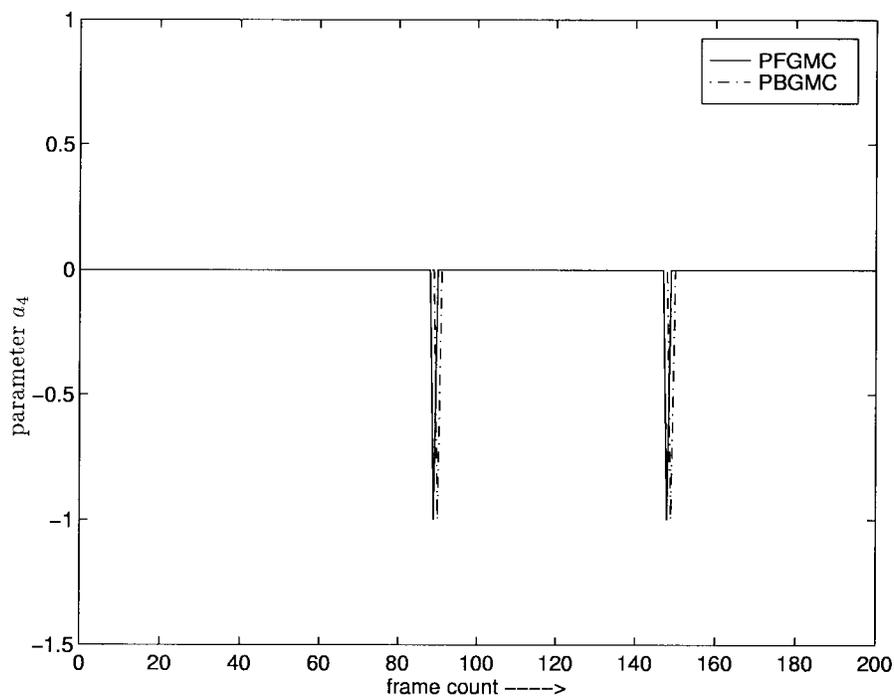
Huffman code was obtained by finding the relative frequencies of the chessboard distances over the entire sequence. Table VI shows the relative frequencies and the Huffman codes for the three global motion compensation schemes. Table VII shows the average bits per motion vector when the motion vectors are coded using this table. Using ILSE-PFGMC, we get 39.25 and 73.38% savings in bit rate over the conventional motion

compensation in the zoom and the pan regions, respectively. The savings in bit rate over the entire sequence is 61.85%.

Finally, to compare the proposed motion vector coding scheme with the existing two-dimensional (2-D) entropy coding, we computed the 2-D entropy and the differential 2-D entropy (i.e., entropy of the differential motion field) for all the motion fields considered. The differential motion fields



(c)



(d)

Fig. 14. (Continued.) Estimated value of global parameters for various compensation schemes. PFGMC and BFGMC have identical results. BBGMC produces similar results as with PBGM.

were constructed by subtracting each motion vector from the previous motion vector on the same line in a given motion field. The results shown in Table VIII have been obtained by taking the statistics over the entire sequence. It is clearly seen that 2-D entropy coding applied to the motion fields of the conventional motion compensation is always expensive compared to any of the three global motion compensation

schemes with the proposed motion vector coding scheme (Table VII). Differential 2-D entropy coding applied to the same motion fields achieves a closer bit rate, but needs a huge table look-up for implementation. It is also observed that for all the three global compensation schemes, the proposed coding scheme produces a very close bit rate to that due to 2-D entropy coding.

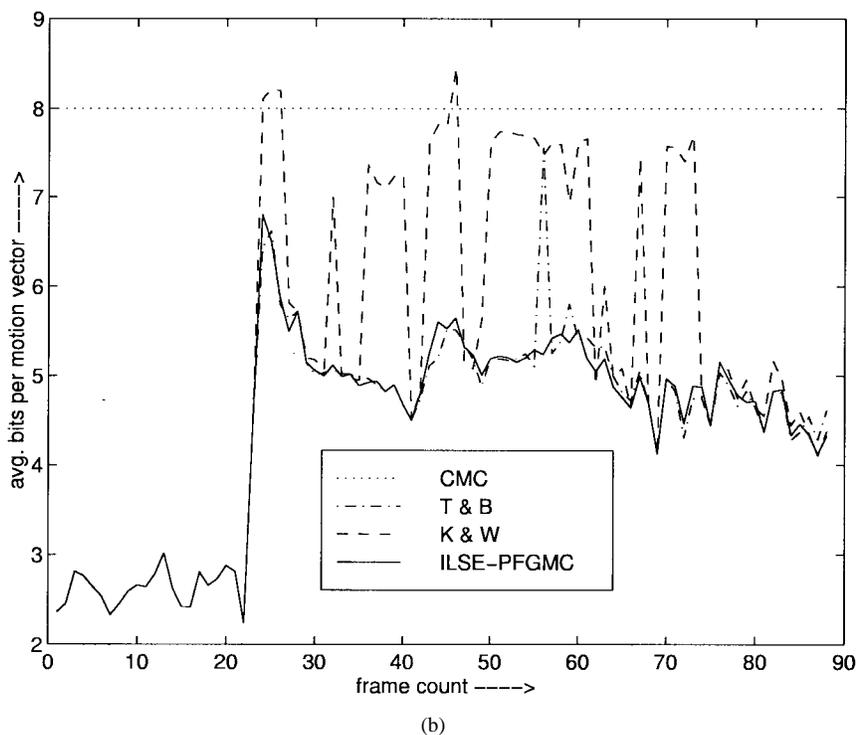
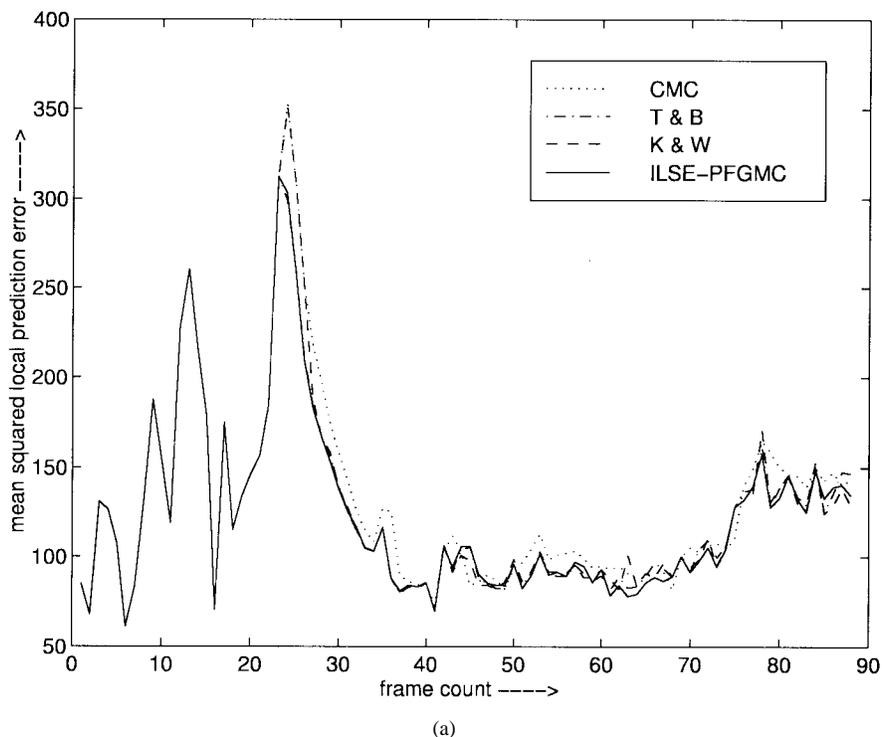


Fig. 15. Mean squared local prediction error and average bits per motion vector plots for scene 1.

## VII. CONCLUSION

In this paper, we presented a four-parameter global motion model. This model is sufficient to account for most of the global motions in real video sequences. Since the rotation of the camera is comparatively much less frequent than the zoom and the pan, we have not included it in our model. In sequences having rotational motion, the model can be

modified to a six-parameter one. The important feature of the model is that it is generalized and can account for global object motions besides the motions due to the camera. The generalized model also gives rise to a simpler least-squares estimation formula for the estimation of global parameters, thus saving computation.

Besides the traditional least-squares estimation scheme, we have presented two other estimation schemes which are more

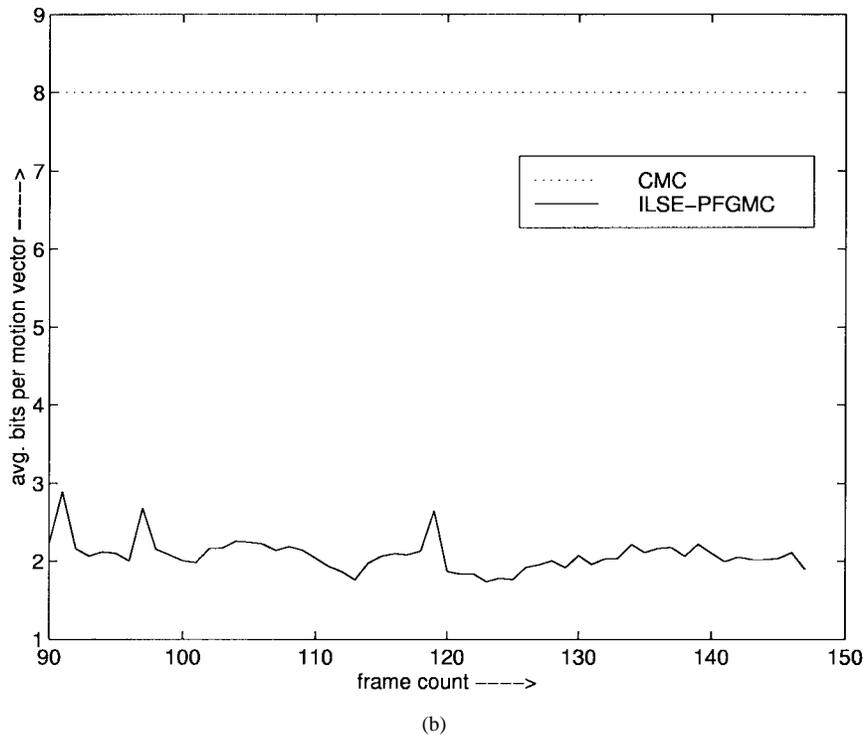
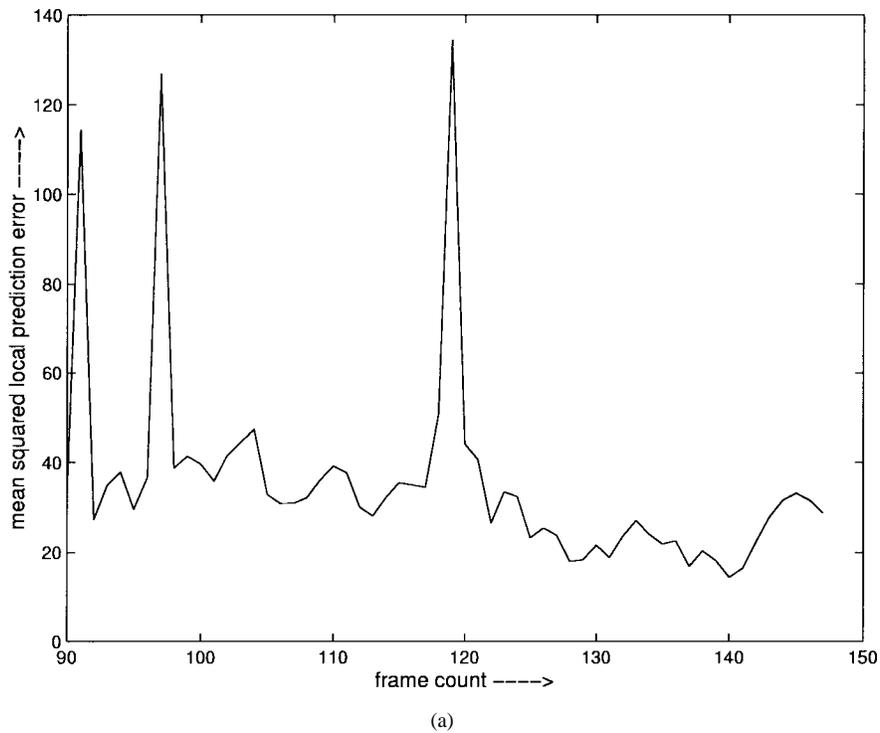
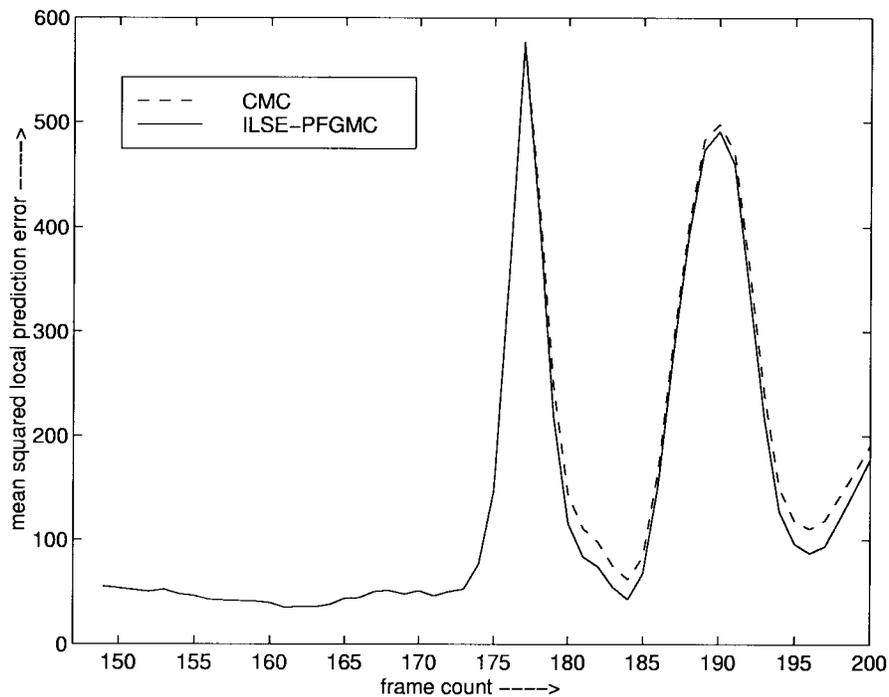


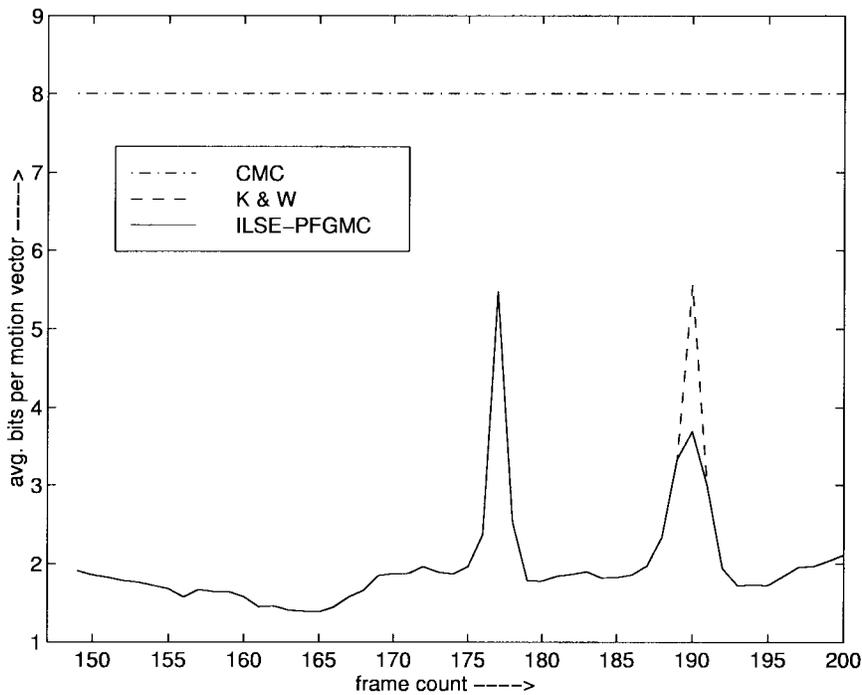
Fig. 16. Mean squared local prediction error and average bits per motion vector plots for scene 2. The conventional scheme and all the global motion compensation schemes have identical prediction error plots since the scene does not have global motion. Motion field bit rates for Tse and Baker's scheme and Kamikura and Watanabe's scheme are the same as with ILSE-PFGMC.

appropriate in the context of compression than the former. But these two schemes are extremely compute intensive; therefore we have used one of the conventional fast search techniques to compute the optimal parameter values. To eliminate the effect of the local object motion from the estimates, we performed least-squares estimation iteratively till convergence.

We have compared these three estimation schemes with Tse and Baker's [4] iterative estimation scheme and Kamikura and Watanabe's [5] histogram-based estimation scheme. Although the iterative least-squares estimation scheme does not consider either the motion field bit rate or the global prediction error, its performance was seen to be comparable



(a)



(b)

Fig. 17. Mean squared local prediction error and average bits per motion vector plots for scene 3. Tse and Baker have the same prediction error and motion field bit rate plots as with ILSE-PFGMC. Kamikura and Watanabe have same prediction error as with ILSE-PFGMC except at frame 190.

to the other two proposed schemes. Among the two existing estimation schemes considered, Tse and Baker's [4] scheme was found to perform in a similar manner. But this scheme is for estimating the parameters of a three-parameter global motion model. The proposed iterative least-squares estimation is an equivalent method for the proposed four parameter model which accommodates further global object motions.

Considering the computational advantages and the accuracy of the estimated parameters, we have chosen the iterative least-squares estimation to be the best. Since, in the case of a six-parameter model, the computations become prohibitively large even for fast search techniques, the least-squares estimation is the sole choice among the proposed schemes if the rotational motion is to be included in the model.

TABLE IV  
AVERAGE MEAN SQUARED LOCAL PREDICTION  
ERROR FOR VARIOUS COMPENSATION SCHEMES

scheme	entire sequence	zoom	pan
CMC	126.952	123.772	238.778
Tse & Baker	121.557	115.299	220.833
Kamikura & Watanabe	122.449	116.943	220.815
ILSE-PFGMC	120.756	114.187	220.833

TABLE V  
AVERAGE BITS PER MOTION VECTOR FOR VARIOUS COMPENSATION SCHEMES

scheme	entire sequence	zoom	pan
CMC	8	8	8
Tse & Baker	3.165	5.061	2.248
Kamikura & Watanabe	3.529	6.153	2.326
ILSE-PFGMC	3.156	5.033	2.248

TABLE VI  
RELATIVE FREQUENCIES AND THE HUFFMAN  
CODES FOR DIFFERENT CHESSBOARD DISTANCES

chessboard distance	ILSE-PFGMC		Tse & Baker		Kamikura & Watanabe	
	R.F.	code	R.F.	code	R.F.	code
0	0.7094	1	0.7069	1	0.6483	1
1	0.1040	011	0.1092	00	0.1608	00
2	0.0445	0101	0.0440	0101	0.0474	0110
3	0.0330	0011	0.0319	01111	0.0322	01111
4	0.0258	0010	0.0253	01110	0.0270	01110
5	0.0234	0001	0.0233	01101	0.0232	01010
6	0.0230	0000	0.0232	01100	0.0238	01011
7	0.0369	0100	0.0362	0100	0.0373	0100

TABLE VII  
AVERAGE BITS PER MOTION VECTOR FOR VARIOUS  
COMPENSATION SCHEMES. CHESSBOARD DISTANCES OF THE  
MOTION VECTORS ARE HUFFMAN-CODED USING TABLE VI

scheme	entire sequence	zoom	pan
CMC	8	8	8
Tse & Baker	3.051	4.8914	2.0465
Kamikura & Watanabe	3.314	5.6823	2.0984
ILSE-PFGMC	3.052	4.8602	2.1296

TABLE VIII  
ENTROPY AND DIFFERENTIAL TWO-DIMENSIONAL ENTROPY  
OF MOTION FIELDS FOR VARIOUS COMPENSATION SCHEMES

scheme	2-D entropy (bits/motion vector)	differential 2-D entropy (bits/motion vector)
CMC	4.206	3.334
ILSE-PFGMC	2.767	3.032
Tse & Baker	2.776	3.038
Kamikura & Watanabe	3.107	3.064

We also presented four global motion compensation schemes and compared their performances. The mean squared global prediction error was the performance measure. Among the four global motion compensation schemes, the pixel-

based compensation schemes were found to have better performance than the block-based ones. But, since the block-based compensation schemes require less computations than the corresponding pixel-based ones, they can be preferred in computation-constrained situations. The backward compensation schemes were seen to perform similarly as the corresponding forward compensation schemes except having one frame delay. Since they require less computation than the corresponding forward compensation schemes, they can be preferred for real-time applications. To compare the overall performance with the conventional motion compensation and the existing two-stage compensation schemes [4], [5], we considered the pixel-based forward compensation because of its best performance and performed local motion estimation and compensation in addition. The performance measures were the mean squared local prediction error and the motion field bit rate. The motion fields for the conventional motion compensation were encoded using a fixed-length coding scheme whereas those for the two stage compensation schemes were encoded using the proposed motion vector coding scheme. The proposed pixel-based forward compensation was found to perform better than the existing two-stage compensation schemes [4], [5], and much better than the conventional motion compensation in terms of the motion field bit rate. All the compensation schemes were comparable in terms of the mean squared local prediction error.

Finally to compare the performance of the proposed coding scheme with those of the existing lossless coding schemes for motion vectors, we computed the 2-D entropy and the differential 2-D entropy of the motion fields. The computed values for the conventional motion compensation were found to be larger than the bit rate for the pixel-based forward compensation computed using the proposed scheme. We also observed that the bit rate for the pixel-based forward compensation was comparable to the two-dimensional entropy of its own motion field.

The proposed global motion compensation can be easily incorporated into the existing motion compensation as an add-on block with a little modification in the latter's structure. The global motion compensation block can be disabled or enabled depending on the nature of the video sequence.

## REFERENCES

- [1] H. G. Musmann, P. Pirsh, and H.-J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, no. 4, pp. 523-548, 1985.
- [2] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799-1808, Dec. 1981.
- [3] M. Hoetter, "Differential estimation of the global motion parameters zoom and pan," *Signal Processing*, vol. 16, pp. 249-265, 1989.
- [4] Y. T. Tse and R. L. Baker, "Global zoom/pan estimation and compensation for video compression," in *Proc. ICASSP*, 1991, pp. 2725-2728.
- [5] K. Kamikura and H. Watanabe, "Global motion compensation in video coding," *Electronics Commun. Japan*, vol. 78, no. 4, pt. 1, pp. 91-102, 1995.
- [6] M. M. de Sequeira and F. Pereira, "Global motion compensation and motion vector smoothing in an extended H.261 recommendation," in *Video Communications and PACS for Medical Applications, Proc. SPIE*, Apr. 1993, pp. 226-237.
- [7] Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge, "A new method for camera motion parameter estimation," in *Proc. Int. Conf. Image Processing-ICIP*, 1995, vol. 1, pp. 405-408.

- [8] K. Illgner, C. Stiller, and F. Müller, "A robust zoom and pan estimation technique," in *Proc. Int. Picture Coding Symp., PCS'93*, Mar. 1993, pp. 10.5–10.6.
- [9] A. Zakhor and F. Lari, "Edge-based 3-D camera motion estimation with application to video coding," *IEEE Trans. Image Processing*, vol. 2, pp. 481–498, Oct. 1993.
- [10] F. Moscheni, F. Dufaux, and M. Kunt, "A new two-stage global/local motion estimation based on a background/foreground segmentation," in *Proc. ICASSP*, 1995, pp. 2261–2264.
- [11] G. B. Rath and A. Makur, "Efficient motion vector coding for video compression," in *Pattern Recognition, Image Processing and Computer Vision Recent Advances*, P. P. Das and B. N. Chatterji, Eds. Narosa Publishing House, 1995, pp. 49–54.
- [12] W. Y. Choi and R.-H. Park, "Motion vector coding with conditional transmission," *Signal Processing*, vol. 18, no. 3, pp. 259–267, 1989.
- [13] T. Koga and M. Ohta, "Entropy coding for a hybrid scheme with motion compensation in subprimary rate video transmission," *IEEE J. Select. Areas. Commun.*, vol. SAC-5, pp. 1166–1174, 1987.
- [14] H. Schiller and B. B. Chaudhuri, "Efficient coding of side information in a low bit rate hybrid image coder," *Signal Processing*, vol. 19, no. 1, pp. 61–73, 1990.



**Gagan B. 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. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1993. He is currently a doctoral student in the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore.

His current research interests are video compression, video communication systems, and digital signal processing.



**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 electronic engineering from the California Institute of Technology, Pasadena, in 1986 and 1990.

He is currently an Associate Professor in ECE, Indian Institute of Science, Bangalore. He had a visiting assignment in ECE, University of California, Santa Barbara, during 1997–1998. His research interests in signal compression includes vector quantization, subband coding and filterbank design, motion field coding, other image/video compression schemes and standards, and multimedia applications. His interests in image/video processing includes halftoning, image restoration, and two-dimensional filter design.

Dr. Makur is the recipient of the 1998 Young Engineer Award from the Indian National Academy of Engineering.