ELSEVIER

Signal Processing: *Image Communication* 6 (1995) 507–520

SIGNAL PROCESSING: IMAGE COMMUNICATION

Global-motion estimation in image sequences of 3-D scenes for coding applications

Amichay Amitay*, David Malah

Department of Electrical Engineering, Technion-Israel Institute for Technology, Technion city, Haifa 32000, Israel

Received 7 July 1993; revised 1 June 1994

Abstract

A technique for *global*-motion estimation and compensation in image sequences of 3-D scenes is described in this paper. Each frame is segmented into regions whose motion can be described by a single set of parameters and a set of motion parameters is estimated for each segment. This is done using an iterative block-based image segmentation combined with the estimation of the parameters describing the global motion of each segment. The segmentation is done using a Gibbs–Markov model-based iterative technique for finding a local optimum solution to a maximum a posteriori probability (MAP) segmentation problem. The initial condition for this process is obtained by applying a Hough transform to the motion vectors of each block in the frame obtained by block matching. In each iteration, given a segmentation, the motion parameters are estimated using the least-squares (LS) technique. To obtain the final segmentation and the more appropriate higher-order motion model for each segment, a final stage of splitting/merging of segments is needed. This step is performed on the basis of maximum-likelihood decisions combined with the determination of the higher-order model parameters by LS. The incorporation of the proposed global-motion estimation technique in an image-sequence coder was found to bring about a substantial reduction in bit-rate without degrading the perceived quality or the PSNR.

Keywords: Image sequence coding; Video coding; Global motion; Local motion; Hough transform; Least squares; ICM; Motion estimation; Motion segmentation; Gibbs distribution; Maximum likelihood splitting/merging

1. Introduction

Techniques for *global*-motion estimation in image sequences are of great interest in image sequence coding [1, 7, 17]. The better the motion compensation, the better is the prediction of the picture to be coded, thus bringing about a reduction in bit-rate and/or improved quality. The savings in bit-rate is due to the smaller prediction error as well as to the reduction in motion-vector information that needs to be coded, as global motion can be described by a rather small number of parameters. Current coding standards (e.g. H.261 [5], MPEG [12, 16]) apply only *local*-motion compensation, assuming that blocks in the image have translatory motion only. However, for image sequences of 3-D scenes this is usually an inadequate assumption, even when the camera is just panned,

* Corresponding author. Presently at Tadiran, 26, Hashoftim St., P.O.B. 267, 58102 Holon, Israel.

SSDI 0923-5965(94)00034-4

since the motion magnitude may vary gradually from pixel to pixel in the block depending on the distance of the corresponding object from the camera. If the camera is zoomed, this situation occurs even for 2-D scenes.

Most reported methods attempt to find a single set of parameters describing the global motion using techniques like least squares (LS) [1, 7, 17], iterative search for a parameter set giving a minimal frame difference [9], or using a modification of the block-matching (BM) technique to include matching over pan and zoom parameters [1]. These approaches could give a good description of the global motion when 2-D scenes are involved. For 3-D scenes, a single set of motion parameters is usually unable to describe the global motion if different pixels in the picture are at different distance from the camera. Furthermore, large locally moving objects reduce the accuracy of the estimated parameters. Hough-based [7] and feature-Hough-based [13] methods are more robust than the above methods when there is more than one dominant motion in the picture. Still, in those works only the motion of the largest moving region is found and the full description of the global motion in the picture is not obtained.

For general 3-D scenes, segmentation-based methods are therefore needed, with each segment having its own set of motion parameters. In [11], the peaks of the cosine area transform determine both the motion parameters and the segmentation in different regions, whose motion can be described by the mathematical model assumed. This method requires processing of a large group of future frames, and therefore is not attractive for image coding, because of storage requirements and the delay caused. Another problem with this method is that long-term consistency of the motion is assumed [11]. There are also methods which try to minimize the frame difference, but usually are computationally costly even when an hierarchical approach is used [4]. Also, when used in coders, a large overhead results from transmitting pixel-based segment boundaries. Therefore, we preferred using in our work a block-based segmentation.

The approach described in this paper is found to provide good performance for quite general 3-D scenes. Unlike the methods mentioned above there is no requirement for processing a large group of frames or for motion consistency, and the block-based segmentation used requires only a small amount of side information to be transmitted to the decoder.

The method proposed in this work combines the block-based segmentation with the estimation of the parameters describing the global motion of each segment. The segmentation is based on a Gibbs–Markov model with an iterative optimization process. This process requires initial conditions for the segmentation and the motion parameters. These are obtained by a Hough transform, using the motion vectors obtained by BM. The Hough transform is performed according to a 3-parameter motion model, because of computation considerations, rather than a more appropriate higher-order 8-parameter model of a perspective projection of a planar surface in space which we eventually compute. Finally, splitting/merging of segments is applied in order to turn the 3-parameter motion-segmentation description into an 8-parameter description. The details of these steps are given below.

2. Description of segmentation and parameter estimation algorithms

2.1. Motion-vectors extraction

In current image-sequence coding standards (CCITT/H.261, ISO/MPEG), the motion between two consecutive frames in an image sequence is assumed to consist of translatory motion of blocks. The motion vectors of the image blocks are usually found by BM techniques [19] and hardware chips capable of doing this type of motion estimation are available. The algorithms presented in this paper are using these block motion vectors in order to segment the image and to estimate the global-motion parameters for each one of the segments. In most image-sequence coding applications motion-vector components have integer values (or optionally half-pixel values – as in MPEG), which is only an approximation of the real motion. In later stages of the proposed algorithm, sub-pixel accuracy of the vectors is found to be very helpful and is

obtained directly from the mean absolute difference (MAD) function calculated by the BM algorithm. The MAD function is assumed to be parabolic in the vicinity of its minimum and the location of the minimum is then computed with subpixel accuracy. This method is chosen because it requires only a small amount of computations.

2.2. Initial Hough-based segmentation

As mentioned above, LS provides a good estimation of the motion parameters if it uses motion vectors which describe the motion of a homogeneously moving area. That is, it consists mainly of the motion of a single region moving according to an expected motion model. However, when the picture contains several objects or regions having different motion parameters, LS will find an 'average' parameter set, which does not describe accurately the motion of any of the objects. Therefore, segmentation should be done first, and only after the segmentation defines a region whose motion can be well described by the assumed motion model, LS can be applied to get a good estimation of the motion parameters.

A robust method for segmentation is the Hough transform [7, 13]. In this method, the BM motion vectors are used for voting for parameter sets in the parameter space, representing the motion of the different segments. The Hough transform used is based on a 3-parameter model (zoom, pan and tilt). By this model a pixel whose coordinates are (x_c, y_c) in the current frame is originated from a pixel at the coordinates

$$(x_p, y_p) = (\xi x_c + P_x, \xi y_c + P_y) \tag{1}$$

in the previous frame, where ξ is the zoom-out factor and P_x, P_y are the pan and tilt parameters, respectively. The BM techniques are performed under the assumption that all the pixels in a block have the same translatory motion. This is not the case when the camera is zoomed, and hence the motion vector found by the BM technique is only an approximation to the real motion in the block. The displacement (motion vector) of a block with the coordinates (X, Y) in the current frame, according to the 3-parameters motion model in (1), is assumed therefore to be given by

$$\begin{aligned}(x_p - x_c, y_p - y_c) &\triangleq (\Delta x, \Delta y) \\ &= (X(\xi - 1) + P_x,\ Y(\xi - 1) + P_y).\end{aligned} \tag{2}$$

We use at this point only a 3-parameter model since the amount of computations involved in performing the Hough transform increases exponentially with the number of the model parameters. Hence, applying it to a model of order higher than 3 is usually computationally prohibitive. The voting process is done by making each data measurement (in our case block-displacement measurements) vote for each of the values in the parameter space that are compatible with that measurement. The parameter space is represented by an accumulator array of N_p dimensions, where N_p is the number of parameters in the motion model (three in our case). Each array cell represents a 'cube' in the parameters space, and a displacement measurement will vote for this cell if the parameter set corresponding to that cube results in a motion vector sufficiently close to that of the voting block. Every set of parameters, which carries enough support in the voting process (i.e. represented by a peak in the parameter space), defines a distinct segment. To avoid aliasing of high-valued peaks, an iterative 'back cleaning' process is used. At each iteration the highest peak is found; the blocks which support it define a segment, and their votes are not considered in the following iterations. This process stops when the highest peak is smaller than some threshold. This threshold is set to be the smallest number of blocks needed for defining a distinct segment.

The parameter set accuracy is defined by the Hough cell resolution, which is limited because of computational considerations. In order to obtain more accurate parameters, an LS-based parameters estimation is performed for each segment, as was explained at the beginning of this section.

In a general linear model the vector of measurements $\boldsymbol{d}$ is a linear function of the set of parameters $\boldsymbol{a}$:

$$M\boldsymbol{a} = \boldsymbol{d}, \tag{3}$$

where M is a given matrix. Usually, in an LS problem the number of measurements is greater than the number of variables. Such is also the case in the

problem of estimating the global-motion parameters (3 or 8 in our case) according to the displacement vectors (usually several tens or hundreds). Therefore, this is an overdetermined system of equations. Assuming additive noise in the measurements, the problem can be put in the form

$$\boldsymbol{M}\boldsymbol{a} + \boldsymbol{n} = \boldsymbol{d}, \tag{4}$$

where $\boldsymbol{n}$ is an additive noise vector. The estimated vector is the one which minimizes the squared error, defined as

$$e^2 \triangleq \boldsymbol{n}^{\mathrm{T}}\boldsymbol{n} = (\boldsymbol{d} - \boldsymbol{M}\boldsymbol{a})^{\mathrm{T}}(\boldsymbol{d} - \boldsymbol{M}\boldsymbol{a}). \tag{5}$$

The estimated parameter vector is then obtained from [14]

$$\hat{\boldsymbol{a}} = (M^{\mathrm{T}}M)^{-1}M^{\mathrm{T}}\boldsymbol{d}. \tag{6}$$

In the case of estimating the parameters of the 3-parameter global-motion model, the formulation in (3) takes the form

$$\underbrace{\begin{vmatrix} X^1 & 1 & 0 \\ Y^1 & 0 & 1 \\ -- & - & -- \\ X^2 & 1 & 0 \\ Y^2 & 0 & 1 \\ -- & - & -- \\ \vdots & \vdots & \vdots \\ X^L & 1 & 0 \\ Y^L & 0 & 1 \end{vmatrix}}_{M} \underbrace{\begin{bmatrix} (\xi - 1) \\ P_x \\ P_y \end{bmatrix}}_{\boldsymbol{a}} = \underbrace{\begin{vmatrix} \Delta x^1 \\ \Delta y^1 \\ ----- \\ \Delta x^2 \\ \Delta y^2 \\ ----- \\ \vdots \\ ----- \\ \Delta x^L \\ \Delta y^L \end{vmatrix}}_{\boldsymbol{d}}, \tag{7}$$

where (X^i, Y^i) are the coordinates of the *center* of the ith block, $i = 1, 2, \ldots, L$, in relation to the center of the frame and L is the number of blocks in the group of blocks for which the parameters are estimated. In the case of estimating the parameter set of a segment this group is the group of blocks that belong to that segment.

Following the determination of the parameter sets for segments corresponding to the selected Hough transform peaks, the initial segmentation can now be completed. A block which is not in any of those segments, because it did not support any of the peaks in the Hough transform domain, is now associated with one of the neighboring segments. The segment chosen is the one having an associated parameter set which gives the lowest MAD for that block.

2.3. Gibbs–Markov model-based segmentation

The Hough transform gives a segmentation which is unsatisfactory in some ways. First there could be individual blocks or groups of blocks that are erroneously matched to another segment as can be seen in Fig. 1(b). This can be the result of stray motion vectors, and of the fact that a vector may vote for more than one set of parameters. Another problem is inaccuracy, especially at segment boundaries. Both of these effects are corrected by the algorithm proposed below, using a Gibbs–Markov model-based segmentation. The Gibbs model is used in many applications, including motion smoothing [15] and segmentation [6].

The problem we wish to solve is: Having two successive frames (A, B) of a sequence, we look for the best compensation (in a sense to be defined) of global and local motion of frame B with respect to frame A. The compensation is done by first applying *global*-motion compensation and then *local*-motion compensation of the *residual* local displacement vector field V. Local-motion compensation is needed in order to correct for

(a) inaccuracy of the global-motion parameters,
(b) global motion that cannot be described by the global-motion model used,
(c) small moving objects within the frames.

The global motion of a segment is defined by its global-motion parameters. However, since these parameters can be completely determined once the segmentation is given, we concentrate on the segmentation problem. The local displacement field is estimated between frame B and the *prediction* of frame B, obtained by global-motion compensation from frame A. Therefore, in order to perform the best compensation, we wish to find the best segmentation field S and the best residual local-motion-vector field V, where in 'best' we mean here

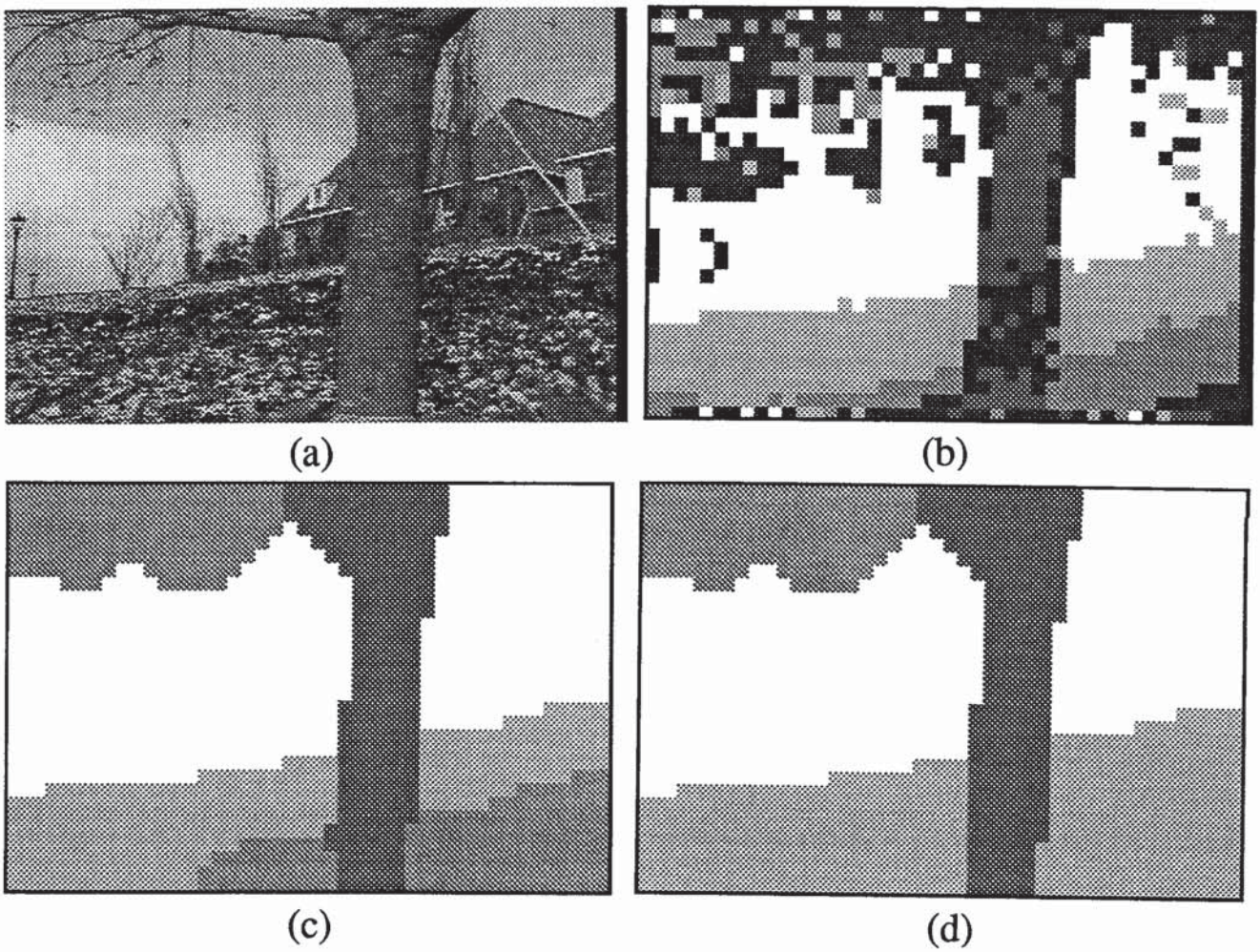

Fig. 1. (a) Original picture from the Flower-Garden sequence. (b) Hough-based segmentation results. (c) Gibbs-model-based segmentation results. (d) Final segmentation results by proposed algorithm.

finding those V and S which will maximize the conditional probability mass function $P(V,S|A,B)$, i.e., finding the maximum a posteriori probability (MAP) solution. Using Bayes' rule,

$$P(V,S|A,B) = \frac{P(V,S,A,B)}{P(A,B)} = \frac{P(A,B|S,V)\,P(V|S)\,P(S)}{P(A,B)}, \tag{8}$$

where $P(V,S|A,B)$ is the probability of the specific segmentation S and motion-vector field V to be occurring, out of all the possible segmentations and motion-vector fields between the two given consecutive frames A and B. $P(S)$ is the probability of the segmentation field S out of all the possible segmentation fields. For example, a checkered segmentation field is expected to have a small probability.

$P(V|S)$ is the probability of the motion-vector field V out of all the possible vector fields to be occurring when the segmentation field is S. $P(A,B|S,V)$ is the probability that the frame B was created from the frame A by global and local motions which can be determined from the segmentation field S, the original block motion vectors, and the residual local motion-vector field V. $P(A,B)$ is the probability of having the two consecutive frames A and B (under the assumption that there was no scene cut). $P(A,B,S,V)$ is the probability of having the combination of two consecutive frames A and B and that the segmentation field and motion-vector fields S and V, respectively, describe the motion between these two frames.

Since $P(A,B)$ does not affect the minimization, we have to find V and S such that $P(A,B|S,V) \times P(V|S)P(S)$ is maximized, or equivalently such that $-\ln\{P(A,B|S,V)P(V|S)P(S)\}$ is minimized.

Using the assumptions of a Laplacian distributed frame difference, with zero mean and quasi-stationarity of the frame difference, so that the standard deviation σ can be assumed constant in the minimization region (which is chosen to be a block, as will be elaborated later), $P(A,B|S,V)$ can be shown to satisfy the following relation:

$$P(A,B|S,V) \propto \prod_{i=1}^{K} \frac{1}{\sigma} \exp\left(\frac{-d(i)}{\sigma}\right), \tag{9}$$

where K is the size, in pixels, of the area on which the minimization is done, and $d(i)$ is the absolute value of the frame difference at the ith pixel. The value of σ in the minimization region is estimated using the maximum-likelihood estimator

$$\hat{\sigma} = \frac{\sqrt{2}}{K} \sum_{j=1}^{K} d(j) \propto \mathrm{MAD}(B,\hat{B}). \tag{10}$$

The proportionality relation in (9) becomes therefore

$$P(A,B|S,V) \propto \mathrm{MAD}(B,\hat{B})^{-K}, \tag{11}$$

where $\hat{B}$ is the prediction of image B and is obtained from A by global- and local-motion compensation. In [15] a similar relation is obtained for the mean squared error (MSE), assuming a Gaussian distribution of the frame-difference data. Since in most image-sequence coding standards, the MAD is calculated, rather than MSE, computations are saved by using the MAD. However, more significant is the fact that the error distribution measured in our simulations appeared to be closer to a Laplacian rather than to a Gaussian.

As expected, the likelihood that frame B is the result of applying to frame A, V and global motion compensation is maximized when the displaced (i.e. compensated) frame difference is minimized. The global-motion compensation is done according to the global-motion parameters which were estimated according to the segmentation S.

$P(V|S)$ (see (8)) is the probability of the residual local motion-vector field V occurring, given the segmentation S (with its associated global-motion parameter set). It is desirable, of course, that the global-motion model will fully describe the motion between the two frames so that the residual local vector field will be zero everywhere, except at blocks belonging to small moving objects. However, because of parameter inaccuracy and the possibility of having a global motion which cannot be fully described by the assumed motion model, we usually obtain nonzero residual local motion vectors also at other blocks. The segmentation process can also be considered as segmenting the initial displacement vector field, obtained by BM (see Section 2.1) such that each segment contains smooth motion, meaning that there could be a large change in the displacement vector field only at segments boundary. This means that we can assume an interaction between the displacement vectors of each segment. This interaction is modeled here by a second-order Gibbs–Markov random field.

A Gibbs–Markov field [2] has the form

$$P(V|S) \propto \exp\left(-\sum_{i=1}^{M} D_i\right), \tag{12}$$

where M denotes the number of cliques in the frame and D_i (which is defined in (13)) stands for the cost associated with the ith clique. A clique can be any combination of blocks but because of computational considerations only a second-order model is used here. Therefore, only cliques consisting of two blocks, where one of the blocks is in the 8-neighborhood of the other, are considered. For example, for vectors v_1 and v_2 of a clique (belonging to segments s_1 and s_2, respectively), a cost term supporting smooth motion inside each segment (allowing a discontinuity at segment boundaries) is defined as

$$D \triangleq \frac{k_1}{l} \| v_1 - v_2 \| \frac{\delta(s_1 - s_2)}{N_{s_{1,2}}}, \tag{13}$$

where k_1 is a weight factor representing the importance of the motion smoothness term relative to the importance of the MAD term in (11). l is the distance between the two blocks, i.e., $l = 1$ for blocks in a 4-neighborhood and $l = \sqrt{2}$ for diagonal neighbors. The Kronecker delta function assures that smoothness is required only for blocks of the

same segment, i.e., only when $s_1 = s_2$. $N_{s_{1,2}}$ is the number of neighboring blocks that belong to the same segment as the blocks of the examined clique, which consists here of blocks 1 and 2. It is used in order to get, in the total cost term, the average distance of the motion vector from those of the neighboring blocks of the same segment. v_1 and v_2 are the vectors found after global compensation, and their difference is the local-motion difference in the same segment, and hence is expected to be rather small. The intuitive explanation to (13) is that D is a positive term which is proportional to the norm of the difference between the two vectors. Therefore, when the difference is large it is seen from (12) that it less likely that the two blocks belong to the same segment.

$P(S)$ (see (8)) is the probability of the segmentation field S. The segmentation field is also assumed to be a second-order Gibbs–Markov field, and its probability function is of the form

$$P(S) \propto \exp\left(- \sum_{i=1}^{M} C_i \right). \tag{14}$$

The cost term C for two blocks of a clique which belong to segments s_1 and s_2 and belonged in the *previous frame* to the segments $s_1^{(\text{old})}$ and $s_2^{(\text{old})}$, respectively, is defined as

$$C \triangleq C_0 - \frac{k_2}{l}\delta(s_1 - s_2) - \frac{k_3}{l}[\delta(s_1 - s_2^{(\text{old})}) + \delta(s_2 - s_1^{(\text{old})})], \tag{15}$$

where C_0 is a constant value ensuring that the cost function C is positive, and k_2, k_3 are weight factors which relate to the importance of the corresponding two terms in comparison to other terms in the total cost defined below (see (16)).

The first term in (15) tends to oppose 'holes' in the segmentation field, so that the probability that a block belongs to a segment is high if many of its neighbors belong to the same segment. The second term is supportive of consistency in the segmentation of the current and previous frames.

The Gibbs cost function should be optimized globally over the entire segmentation field and the motion-vector field. Using the terms defined in (11)–(15), we wish to find the fields S, V which will minimize the total cost J, defined as

$$J \triangleq K \ln[\text{MAD}(B, \hat{B})] + \sum_{(a,b)\in \text{all cliques}} \frac{k_1}{l} \| v_a - v_b \| \frac{\delta(s_a - s_b)}{N_{s_a,b}} - \sum_{(a,b)\in \text{all cliques}} \frac{k_2}{l}\delta(s_a - s_b) - \sum_{(a,b)\in \text{all cliques}} \frac{k_3}{l}[\delta(s_a - s_b^{(\text{old})}) + \delta(s_b - s_a^{(\text{old})})]. \tag{16}$$

The minimization $(\min_{s,v} J)$ is over all the possible combinations of fields S and V in the entire frame, which is an extremely large number of combinations. Because of the extensive computational load, a suboptimal approach is used, by which the optimization is done locally (for each block separately), thus finding a local minimum of the cost function. This means that we seek

$$\min_{s_a, v_a} J_a, \tag{17}$$

where

$$J_a \triangleq N \ln[\text{MAD}(a, \hat{a})] + \sum_{b=1}^{8} \frac{k_1}{l} \| v_a - v_b \| \frac{\delta(s_a - s_b)}{N_{s_a,b}} - \sum_{b=1}^{8} \frac{k_2}{l}\delta(s_a - s_b) - \sum_{b=1}^{8} \frac{k_3}{l}[\delta(s_a - s_b^{(\text{old})}) + \delta(s_b - s_a^{(\text{old})})], \tag{18}$$

where N is the size of a block (we use 8×8 blocks), 'a' denotes the tested block and '$\hat{a}$' denotes its prediction. The summation on b from 1 to 8 is over the 8 neighboring blocks to the tested block 'a'. These blocks form cliques with the tested block.

Because the minimization of J_a is over s_a, v_a only, the term $\delta(s_b - s_a^{(\text{old})})$ does not affect the minimization and can therefore be removed from (18). The minimization of J_a is done iteratively using the so-called iterated conditional modes (ICM) algorithm [6, 2], which requires a reasonable amount of computation while providing good results.

In each iteration, the cost function in (18) is evaluated for all the possible segment labels for the block 'a' but only for few vector candidates of 'v_a'

(to save computation). In our simulations those candidates were the following 14 vectors:

(1) The vector calculated in the previous iterations.
(2) The four vectors differing from candidate (1) by 0.5 pixel.
(3) The 8 neighboring vectors of the vector calculated in the previous iteration.
(4) The weighted average of the eight neighbor vectors applying weights $1/l$ (see (13)).

After obtaining S, the sets of parameters for each one of the segments can be evaluated by applying LS (Section 2.2) to the motion vectors (which were found by BM, as described in Section 2.1) of the blocks belonging to that segment. The residual displacement vectors can also be used in the parameter estimation. Every block which has a residual displacement vector larger than some threshold (for example 2 pixels) will be considered as having local motion and will not be taken into account in the LS estimation of the global-motion parameter sets.

It is usually convenient to separate the Gibbs segmentation process from the Gibbs local vector estimation process. This is done by using at first $k_1 = 0$, and finding V after the segmentation is obtained. The results are not affected much and the computational load is largely reduced. It is only natural to chose $k_2 > k_3$, since the segmentation at the current frame should affect the estimation more than the segmentation at the previous frame. The values $k_2 = 2.5$, $k_3 = 0.5$ were found empirically. These values performed well for the sequences that were tested (for example the ISO test sequence 'Flower-Garden'). Choosing very small k_2, k_3 (for example 0.1) is not useful since they will hardly affect the cost function. On the other hand, very large k_2, k_3 (for example 20) is not beneficial either since in this case the MAD function, which is very important as well, will have no effect on the cost function. Furthermore, using very large values of k_2 may constrain the structure of segments by giving too much weight to the homogeneity of the segment. Very large values of k_3 constrain the change of the segmentation in time (from frame to frame). The results were also found not to be sensitive to different values of k_2, k_3 in quite a wide range. In simulations, values of k_2 in the range 1–5 and (k_2/k_3) in the range 2–10 gave similar results. Optimizing Eq. (18) without the last three terms (i.e. choosing $k_1 = k_2 = k_3 = 0$) corresponds to maximum likelihood (ML) optimization rather than to MAP optimization. As expected, we found in simulations that MAP optimization results in a better segmentation and compensation, as evidenced by the larger reduction in bit-rate (e.g., for the Flower-Garden sequence a reduction in rate of 28.3% was obtained in comparison to 24.2% with ML optimization). Furthermore, since most of the computational load is due to the MAD term, there is certainly no motivation in using here $k_1 = k_2 = k_3 = 0$.

2.4. Joint splitting/merging decision and detailed motion description

As explained above the sets of parameters found by the Hough transform correspond to a 3-parameter motion model. The next step involves estimation of the parameters of an 8-parameter model, for each of the segments. This model can represent the perspective projection of a planar surface moving in space. In the 8-parameter model a pixel whose coordinates are (x_c, y_c) in the current frame is originated from the pixel at coordinates (x_p, y_p) in the previous frame, given by [18],

$$(x_p, y_p) = \left(\frac{(1 + A_1)x_c + A_2 y_c + A_3}{1 + A_7 x_c + A_8 y_c}, \frac{A_4 x_c + (1 + A_5) y_c + A_6}{1 + A_7 x_c + A_8 y_c} \right), \tag{19}$$

where A_i, $i = 1, 2, \ldots, 8$, are the model parameters. Note that if $A_i = 0\ \forall i$, then $(x_p, y_p) = (x_c, y_c)$. The expected motion vector (displacement), according to the 8-parameter motion model, of a block centered at coordinates (X, Y) is therefore

$$\Delta x = -\frac{A_1 X + A_2 Y + A_3 - A_7 X^2 - A_8 XY}{1 + A_7 X + A_8 Y}, \tag{20}$$

$$\Delta y = -\frac{A_4 X + A_5 Y + A_6 - A_7 XY - A_8 Y^2}{1 + A_7 X + A_8 Y}. \tag{21}$$

The LS problem associated with the 8-parameter global-motion model is therefore

$$\underbrace{\begin{bmatrix} X^1 & Y^1 & 1 & 0 & 0 & 0 & X^1(-X^1+\Delta x^1) & Y^1(-X^1+\Delta x^1) \\ 0 & 0 & 0 & X^1 & Y^1 & 1 & X^1(-Y^1+\Delta y^1) & Y^1(-Y^1+\Delta y^1) \\ X^2 & Y^2 & 1 & 0 & 0 & 0 & X^2(-X^2+\Delta x^2) & Y^2(-X^2+\Delta x^2) \\ 0 & 0 & 0 & X^2 & Y^2 & 1 & X^2(-Y^2+\Delta y^2) & Y^2(-Y^2+\Delta y^2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X^L & Y^L & 1 & 0 & 0 & 0 & X^L(-X^L+\Delta x^L) & Y^L(-X^L+\Delta x^L) \\ 0 & 0 & 0 & X^L & Y^L & 1 & X^L(-Y^L+\Delta y^L) & Y^L(-Y^L+\Delta y^L) \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \end{bmatrix}}_{a} = -\underbrace{\begin{bmatrix} \Delta x^1 \\ \Delta y^1 \\ \Delta x^2 \\ \Delta y^2 \\ \vdots \\ \Delta x^L \\ \Delta y^L \end{bmatrix}}_{d}. \tag{22}$$

In order to create a better segmentation into regions whose motion can be described by an 8-parameter motion model, the estimation is combined with the splitting/merging of segments. The splitting of a segment is done in cases like the one seen in Fig. 1(c) for the segment composed of the tree branches and the lower garden region. The two regions have the same parameters according to the 3-parameter model, but have different parameters according to the 8-parameter model. In this case, leaving them joined will result in wrong parameter values when the 8-parameter motion model is computed; therefore, splitting should be done. The merging of segments is done in cases like the one seen at the garden region in Fig. 1(c). When described by a 3-parameter model, for zoom, pan and tilt, the garden was divided into 2 regions of different parameters, while the motion of the two regions can actually be described by the same set of 8 parameters. Merging them produces more accurate parameters and reduces discontinuity effects at segments boundaries.

In the literature, pixel-difference-based splitting/merging decisions are mostly used [8]. In our work the decision is based on the BM local motion vector of each block, which is less costly computationally. It is done by a hypothesis testing technique: Each time a splitting/merging decision is to be made, we have to decide between hypothesis 0 – that both segments have the same motion model – or hypothesis 1 – that they have different models. To make this decision we apply the maximum-likelihood approach. The probability distribution used is that of the error between the expected motion vector, according to the model, and the one found by the sub-pixel BM (see Section 2.1). The error is empirically assumed to be Laplacian distributed. We assume that the error vectors are statistically independent and also that the x and y components of the error are statistically independent. Therefore, the probability for the x component of hypothesis 0 is

$$P(H_0^{(v_x)}) = \left(\prod_{i \in G_1 \cup G_2} \left(\frac{\alpha_{1\cup 2}^{(v_x)}}{2} \right) \times \exp\{ -\alpha_{1\cup 2}^{(v_x)} | x_i - m_{1\cup 2}^{(v_x)} | \} \right), \tag{23}$$

and the probability of hypothesis 1 is

$$P(H_1^{(v_x)}) = \left(\prod_{i \in G_1} \left(\frac{\alpha_1^{(v_x)}}{2} \right) \exp\{ -\alpha_1^{(v_x)} | v_{xi} - m_1^{(v_x)} | \} \right) \times \left(\prod_{i \in G_2} \left(\frac{\alpha_2^{(v_x)}}{2} \right) \exp\{ -\alpha_2^{(v_x)} | v_{xi} - m_2^{(v_x)} | \} \right), \tag{24}$$

where α_i and m_i are the parameters of the Laplacian distribution of segment i, $\alpha_{1\cup 2}$ and $m_{1\cup 2}$ are the parameters of the Laplacian distribution of the merged segment, G_1, G_2 and $G_{1\cup 2}$ are the sets of

elements (blocks) belonging to segment 1, segment 2 and the merged segment, respectively. Expressions similar to (23 and 24) are obtained for the y components of hypotheses 1 and 0 and the decision is made by the following test:

$$\frac{P(H_1^{(v_x)})P(H_1^{(v_y)})}{P(H_0^{(v_x)})P(H_0^{(v_y)})} \underset{\text{merge}}{\overset{\text{split}}{\gtrless}} \text{Th}. \tag{25}$$

Using the maximum-likelihood estimators of α and m for each of the distributions we get, after some mathematical manipulations, that the test for splitting/merging two segments is

$$\frac{N_1}{N_1+N_2}\log\left(\frac{\sum_{i\in G_1}|v_{x_i}-m_1^{(v_x)}|\sum_{i\in G_1}|v_{y_i}-m_1^{(v_y)}|}{N_1^2}\right)$$

$$+\frac{N_2}{N_1+N_2}\log\left(\frac{\sum_{i\in G_2}|v_{x_i}-m_2^{(v_x)}|\sum_{i\in G_2}|v_{y_i}-m_2^{(v_y)}|}{N_2^2}\right)$$

$$-\log\left(\frac{\sum_{i\in G_{1\cup 2}}|v_{x_i}-m_{1\cup 2}^{(v_x)}|\sum_{i\in G_{1\cup 2}}|v_{y_i}-m_{1\cup 2}^{(v_y)}|}{(N_1+N_2)^2}\right)$$

$$\underset{\text{merge}}{\overset{\text{split}}{\gtrless}} \text{Th}, \tag{26}$$

where N_1, N_2 are the number of blocks in segments 1 and 2, respectively.

We can use a larger threshold value for segments which were connected in the previous frame, thus increasing the probability that they will also be merged in the current frame. This will support consistency in the segmentation and improve the results.

In the decision rule (26) it is possible that the variance of the y component is getting considerably worse but the improvement in the variance of the x component compensates it (or vice versa) and the merge is done. This is usually an incorrect merge. The results can be further improved, if the variance of each component is also separately examined to avoid such a wrong merge.

As a result of this last step (of splitting/merging of segments), we get the segmentation and global motion parameters set of each segment according to an 8-parameter motion description.

The splitting/merging process is done as follows: Every two segments that were found in the segmentation process of Section 2.3 are being first considered as candidates for merging according to decision rule (26). Then, we check for every segment if it is composed of regions which are not connected. If it is so, each one of these regions is considered as a candidate to be split from this segment according to the decision rule (26). If splitting or merging of a segment is done, then the new segments that were created by this process are again being considered to be merged to one of the other segments.

In scenes containing a significant distortion of the frame (a large zoom or rotation for example), the BM motion vectors usually contain large errors. In such cases, the accuracy of the estimated parameters can be improved by using an iterative process, as demonstrated in Fig. 2 and elaborated in Appendix A. The idea behind this process is that once the global motion between the two consecutive frames was estimated, most of the distortion can be compensated and more accurate motion vectors can be obtained. Using these more accurate motion vectors, a refinement of the global-motion parameter sets can be evaluated.

3. Simulation results

The global-motion compensation method described above was incorporated in an RM8-based image-sequence coder [3] as shown in Fig. 3. The shaded boxes are the proposed added units to the standard coder. The global-motion estimation and compensation units provide a prediction which is usually better than the standard prediction method. However, since the usual prediction (using local-motion compensation only) is also done and the better prediction between the two is taken, the proposed coder can only improve the RM8 coding results. The local-motion estimation units are the BM units which calculate BM motion vectors between the current frame and the previous reconstructed frame (in estimation unit 1) or the previous reconstructed frame after global-motion compensation (in estimation unit 2). The block matching

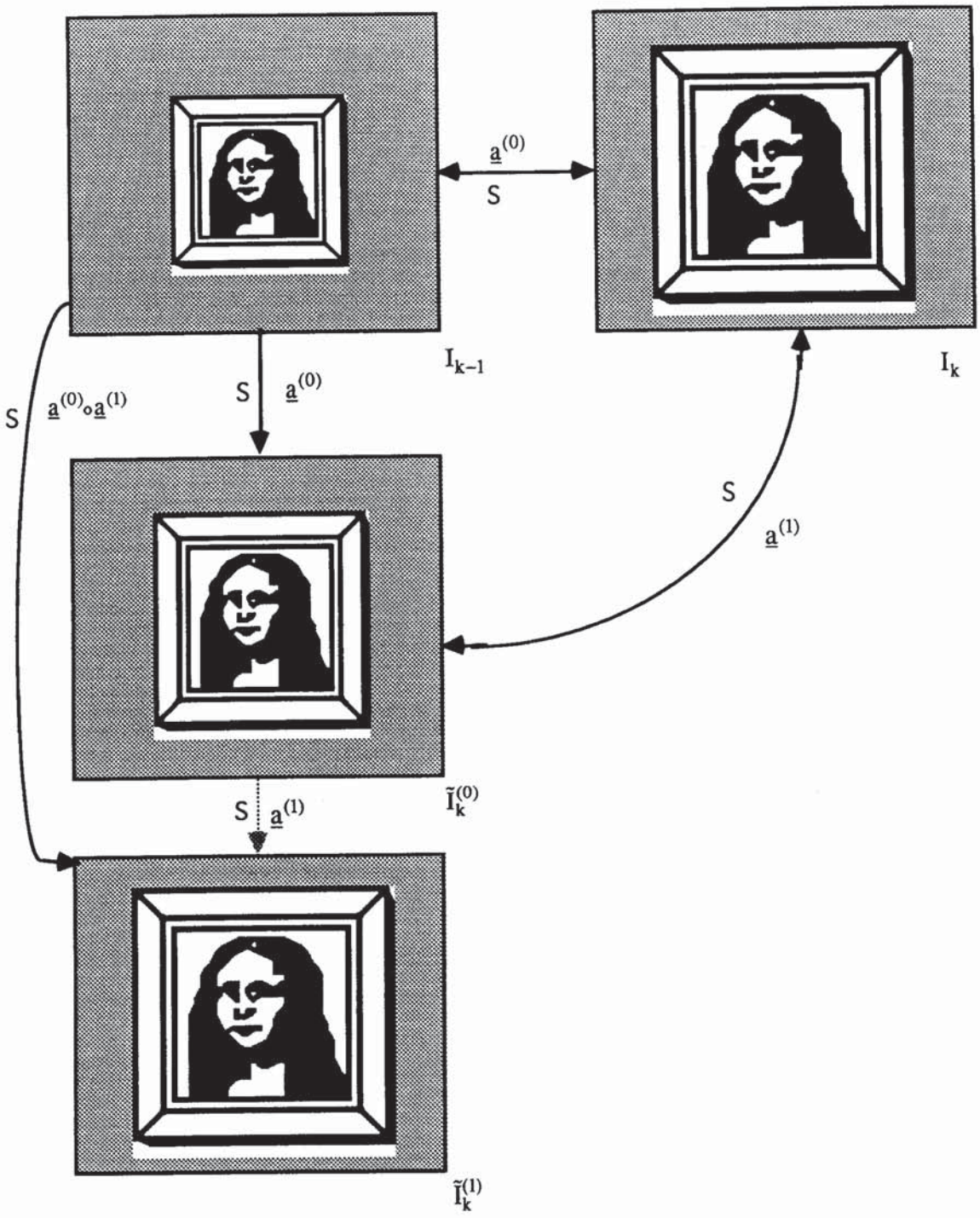

Fig. 2. The iterative process for improving the parameters' estimation accuracy.

(BM) was done using the full-search algorithm. A prediction frame is created in the compensation and prediction units according to the previous reconstructed frame, the local-motion vectors, and the global-motion parameters and segmentation. The discrete cosine transform (DCT) coefficients of each *difference block* are quantized (in unit Q) and are transmitted to the receiver with additional side information such as motion vectors, segmentation, and global-motion parameters. When the

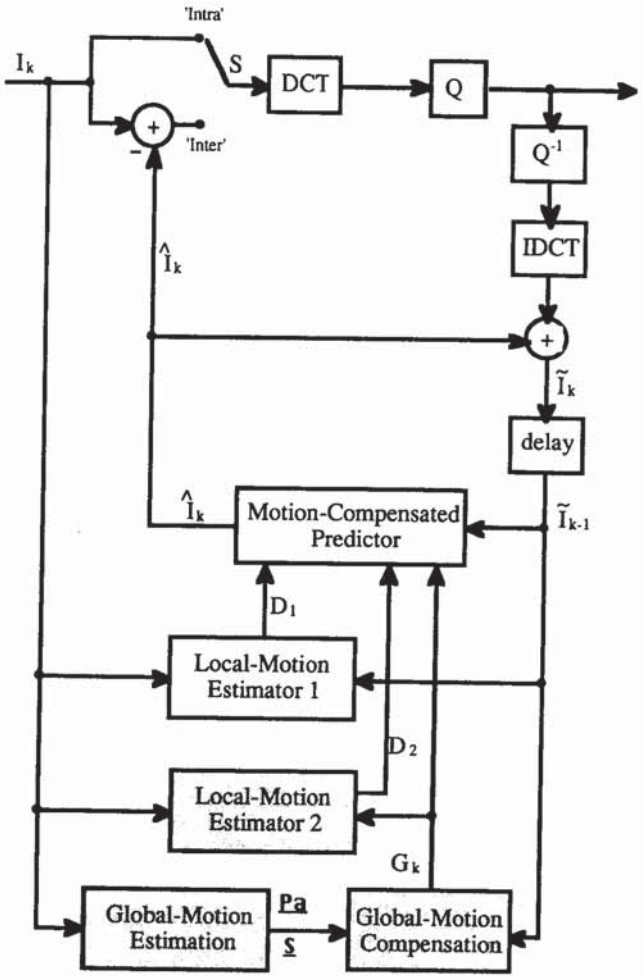

Fig. 3. The proposed coder (the shaded blocks are the blocks which were added to the standard coder.

prediction of the block is not adequate, the quantized DCT coefficients of the current input *block* is transmitted. The reconstructed frame is computed both at the receiver and at the transmitter, using inverse quantization (Q^{-1}) and inverse DCT (IDCT).

The side information of the global-motion parameters and segmentation is also sent to the decoder. The global-motion parameters are sent with 10 bits per parameter. The segmentation is represented by its morphological multistructuring-element skeleton (MSES) representation [10]. The total cost, in bits, of encoding the segmentation information, the global-motion parameters, and the local-motion vectors was found, on average, to be somewhat less than the number of bits needed for encoding the local-motion vectors only in RM8. This is because the global-motion compensation reduces the total number of non-zero local motion vectors.

Table 1 shows simulations results in coding the first 100 frames of the ISO test sequence (luminance only, CIF format) 'Flower-Garden', with and without global-motion compensation (GMC), using a constant quantization step $q_s = 16$.

In comparison to RM8, almost 30% reduction in bit-rate is obtained by using the proposed GMC-based coder, when the sequence is coded in each case with the same quantization step (quantization step = 16). This result is obtained without any noticeable change in quality as judged by an informal subjective comparison. Similar reduction factors were obtained for a wide range of bit-rates. This reduction in bit-rate is due to the reduced prediction error obtained by the improved motion-compensation technique proposed. If zoom or rotation were present an even larger savings in bit-rate could be expected.

Simulations were done also on frames 24–89 of the ISO test sequence (luminance, CIF format) 'Table-Tennis', which is approximately a 2-D scene containing zoom. The coder was activated with and without global-motion compensation (GMC) with a constant quantization step $q_s = 16$. The results in Table 2 show a 32% reduction in bit-rate for this sequence. In this case, the segmentation process results in the whole image being a single segment

Table 1
Simulation results of coding 100 frames (luminance, CIF format) of the 'Flower-Garden' image sequence, with and without global-motion compensation

Coding method	Rate (Mbps)	PSNR (dB)
Standard RM8	2.406	29.54
RM8 with GMC	1.724	30.15

Table 2
Simulation results of coding frames 24–89 (containing zoom) of the 'Table-Tennis' image sequence (luminance, CIF format), with and without global-motion compensation

Coding method	Rate (Mbps)	PSNR (dB)
Standard RM8	0.948	32.19
RM8 with GMC	0.651	33.01

with global motion due to the zoom-out, since the moving parts of the player are not large enough to justify a distinct segment. The local motion of the player is then compensated by the local-motion vectors.

4. Conclusions

The global-motion algorithm described in this paper proves to be promising for coding image sequences of general 3-D (as well as 2-D) scenes which contain global motion. In comparison to a standard coder (RM8-type), about 30% reduction in bit-rate for a wide range of bit-rates was obtained using the proposed GMC-based coder, while maintaining similar subjective image quality.

The main problem is that the computational load is large and probably cannot be implemented at present in real time. For example, the computation time for coding the sequence Flower-Garden with RM8 + GMC is five times more than coding it with an RM8 coder (it should be noted however, that the GMC code was not optimized and a more efficient code could give a somewhat better computation time). However, for off-line applications, the proposed approach appears to be very useful. It should also be noted that the parts of the algorithm which are computationally demanding (like the Hough transform, the Gibbs iterative procedure, the interpolation, and BM) can all be implemented using parallel computing units. This is because the computations required for each 'cell' (Hough accumulator cell, a block in the Gibbs iterative process, a pixel in the interpolation, and a motion vector of a block) can be done independently of the other 'cells' and the computation speed can be boosted in proportion to the number of processors.

Acknowledgements

This work was supported by the Samuel Neaman Institute for Advanced studies in Science and Technology of the Technion.

The authors wish to thank the anonymous reviewers for their useful comments, which helped to improve the paper.

Appendix A. Iterative procedure for improving the accuracy of motion

A.1. Model parameters

The parameters estimated, after applying the algorithm described in Section 2, can be further refined using the iterative process demonstrated in Fig. 2. Following the splitting/merging operation combined with estimation of the 8-parameter model parameter sets, we have the segmentation field S and the parameters sets $\boldsymbol{a}^{(0)}$ which allow the prediction of the current frame I_k from the previous frame I_{k-1}. Because of inaccuracy in the estimation of the parameters (and some additional reasons, which are mentioned in Section 2.3) the prediction frame $\tilde{I}_k^{(0)}$ and I_k are not identical. However, since $\tilde{I}_k^{(0)}$ is closer to I_k than I_{k-1}, because part of the distortion (which can be the result of zoom, rotation, or gradually changing motion), was compensated, further improvement of the accuracy of the global-motion parameters is possible. The next step is, therefore, to find the local-motion vectors between the frames I_k and $\tilde{I}_k^{(0)}$. This is done using the suboptimal 2-D-log search [19] with an initial search step of 0.5 pel and a search step of 0.25 and 0.125 pel in the second and third steps. The block matching with subpel resolution is done by matching blocks which are interpolated by a factor N_s. Motion of a pixel in the enlarged block is equivalent to a motion of $1/N_s$ in the original block. From the resulting motion vectors one can estimate, using the LS method, an additional set of motion parameters $\boldsymbol{a}^{(1)}$ for each one of the segments. Now, using superposition of $\boldsymbol{a}^{(0)}$ and $\boldsymbol{a}^{(1)}$ a more accurate set of global-motion parameters is obtained. The superposition of two parameter sets, $A_1, \ldots, A_8$, and $B_1, \ldots, B_8$, such that

$$
\begin{aligned}
X' &= \frac{(1+A_1)X + A_2Y + A_3}{1 + A_7X + A_8Y}, \\
Y' &= \frac{A_4X + (1+A_5)Y + A_6}{1 + A_7X + A_8Y}, \\
X'' &= \frac{(1+B_1)X' + B_2Y' + B_3}{1 + B_7X' + B_8Y'}, \\
Y'' &= \frac{B_4X' + (1+B_5)Y' + B_6}{1 + B_7X' + B_8Y'},
\end{aligned}
\tag{A.1}
$$

results in a third parameter set, $C_1, \ldots, C_8$ (relating (X'', Y'') directly to (X, Y)), such that

$$X'' = \frac{(1 + C_1)X + C_2Y + C_3}{1 + C_7X + C_8Y},$$
$$Y'' = \frac{C_4X + (1 + C_5)Y' + C_6}{1 + C_7X + C_8Y}. \quad \text{(A.2)}$$

The parameter set $\{C_i\}$ is computed from the parameter sets $\{A_i\}$ and $\{B_i\}$, according to the relations

$$C_1 = \frac{(1 + A_1)(1 + B_1) + A_4B_2 + A_7B_3}{W} - 1,$$
$$C_2 = \frac{A_2(1 + B_1) + (1 + A_5)B_2 + A_8B_3}{W},$$
$$C_3 = \frac{A_3(1 + B_1) + A_6B_2 + B_3}{W},$$
$$C_4 = \frac{(1 + A_1)B_4 + A_4(1 + B_5) + A_7B_6}{W},$$
$$C_5 = \frac{A_2B_4 + (1 + A_5)(1 + B_5) + A_8B_6}{W} - 1, \quad \text{(A.3)}$$
$$C_6 = \frac{A_3B_4 + A_6(1 + B_5) + B_6}{W},$$
$$C_7 = \frac{(1 + A_1)B_7 + A_4B_8 + A_7}{W},$$
$$C_8 = \frac{A_2B_7 + (1 + A_5)B_8 + A_8}{W},$$

where $W = A_3B_7 + A_6B_8 + 1$.

The iterative step can be repeated, but simulations show that most of the improvement was gained in the first iteration.

References

[1] D. Adolph and R. Buschmann, "1.15 Mbit/s coding of video signals including global motion compensation", *Signal Processing: Image Communication* Vol. 3, Nos. 2–3, June 1991, pp. 259–274.

[2] J. Besag, "On the statistical analysis of dirty pictures" *J. Roy. Statist. Soc. B*, 1986, pp. 259–302.

[3] "Description of reference Model 8 (RM8)", Document 525, CCITT SGXV, Working Party XV/4, Specialist group on coding for visual telephony, 1989.

[4] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences", *Signal Processing: Image Communication*, Vol. 3, No. 1, February 1991, pp. 23–56.

[5] "Draft revision of recommendation H.261: Video codec for audiovisual services at $p \times 64$ kbits/s", *Signal Processing: Image Communication*, Vol. 2, No. 2, August 1990, pp. 221–239.

[6] J.N. Driessen and J. Biemond, "Motion field estimation for complex scenes", *SPIE*, Vol. 1605, 1991, pp. 511–521.

[7] Z. Eisips and D. Malah, "Global motion estimation for image sequence coding applications", *Proc. 17th Convention of Electrical and Electronics Engineering*, Israel, May 1991, pp. 186–189.

[8] W. Guse, M. Gilge and C. Stiller, "Region-oriented coding of moving video motion compensation by segment matching", *Signal Processing V*, 1990, pp. 765–768.

[9] M. Hoetter, "Differential estimation of the global motion parameters zoom and pan", *Signal Processing*, Vol. 16, No. 3, March 1989, pp. 249–265.

[10] R. Kresch and D. Malah, "Morphological multi-structuring-element skeleton and its applications", *ISSSE*, Paris, 1992, pp. 166–169.

[11] S.A. Mahmoud, "Motion detection and estimation of multiple moving objects in an image sequence using the cosine area transform (CAT)", *IEE Proc.*, Vol. 138, No. 5, October 1991, pp. 351–356.

[12] Motion Pictures Experts Group, "MPEG video committee draft", MPEG 90/176.

[13] S. Nakajima, M. Zhou, H. Hama and K. Yamashita, "Three-dimensional motion analysis and structure recovering by multistage Hough transform", *SPIE*, Vol. 1605, 1991, pp. 709–719.

[14] L.L. Scharf, "Traitement du signal", in: J.L. Lacoume, T.S. Durrani and R. Stora, eds., *Signal Processing*, North-Holland, Amsterdam 1985, Vol. I.

[15] C. Stiller, "Motion-estimation for coding of moving video at 8 kbit/s with Gibbs modeled vectorfield smoothing", *SPIE*, Vol. 1360, 1990, pp. 468–476.

[16] "Test model 2", document AVC-323 CCITT SGXV, working party xv/1 expert group on ATM video coding, July 1992.

[17] Yi Tong Tse and R.L. Baker, "Global zoom/pan estimation and compensation for video compression", *Internat. Conf. Acoust. Speech Signal Process.*, 1991, Vol. 4, pp. 2725–2728.

[18] R.Y. Tsai and T.S. Huang, "Estimating three dimensional motion parameters of a rigid planar patch", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-29, No. 6, December 1981, pp. 1147–1152.

[19] J.F. Vega-Riveros and K. Jabbour, "Review of motion analysis techniques", *IEEE Proc.*, Vol. 136, Pt. I, No. 6, December 1989.