Video Object Segmentation: A Compressed Domain Approach

R. Venkatesh Babu, K. R. Ramakrishnan, *Member, IEEE*, and S. H. Srinivasan

***Abstract*—This paper addresses the problem of extracting video objects from MPEG compressed video. The only cues used for object segmentation are the motion vectors which are sparse in MPEG. A method for automatically estimating the number of objects and extracting independently moving video objects using motion vectors is presented here. First, the motion vectors are accumulated over a few frames to enhance the motion information, which are further spatially interpolated to get dense motion vectors. The final segmentation, using the dense motion vectors, is obtained by applying the expectation maximization (EM) algorithm. A block-based affine clustering method is proposed for determining the number of appropriate motion models to be used for the EM step and the segmented objects are temporally tracked to obtain the video objects. Finally, a strategy for edge refinement is proposed to extract the precise object boundaries. Illustrative examples are provided to demonstrate the efficacy of the approach. A prominent application of the proposed method is that of object-based coding, which is part of the MPEG-4 standard.**

***Index Terms*—Compressed domain, expectation maximization (EM) algorithm, motion segmentation, MPEG-4, object segmentation, tracking, video object planes.**

I. Introduction

THOUGH MPEG-1 and -2 provide a good representation of digital audio-visual information, the interactivity is only at the frame level. The MPEG-4 [1] standard addresses this interactivity at the object level by defining audio-visual objects (AVOs). The approach taken by the MPEG-4 group in coding of video for multimedia applications relies on the content-based visual data representation of scenes. In contrast to the conventional video coding techniques, in content-based coding, a scene is viewed as a composition of video objects (VOs) with intrinsic properties such as shape, motion, and texture. This content-based representation is the key for facilitating interactivity with objects for variety of multimedia applications. Each frame of MPEG-4 scene is defined in terms of video object planes (VOPs), which are the instants of a semantic object in the scene. However, extracting objects from digital video is still a challenging task among the video processing community. Because of the ill-posed nature of the object segmentation problem, still there is no single robust and reliable technique for VOP generation.

The state-of-the-art approaches to segmentation related to MPEG-4 application can be broadly classified as: 1) intraframe segmentation and 2) motion segmentation. In the first approach, each frame of the video sequence is segmented independently into regions of homogeneous intensity or texture, using the traditional image segmentation techniques [2], [3]. This method suffers from oversegmentation, and segmentation using intensity or texture remains a difficult problem. In the second approach, a dense motion field is used for segmentation and the pixels with homogeneous motion field are grouped together [4]–[7]. In this case, the difficulty lies in estimating a reliable dense motion vector which is computationally very demanding. Since both approaches have their own difficulties, most of the video object segmentation tools integrates both spatial and temporal segmentation techniques [8]–[14].

Though it is not possible to unambiguously define a criterion function for a semantic video object, an object can be characterized very well based on its homogeneity in motion information. In this paper, we consider video sequences containing multiple objects with a smooth motion field [15]. Since an independently moving object can be characterized by coherent motion over the object region, the problem now reduces to clustering pixels that exhibit a similar type of motion. Though much work has been done in the area of motion-based video object segmentation in the pixel domain [16]–[19], very little work has been carried out in the area of compressed domain VOP extraction. Pixel domain motion segmentation is performed based on the motion information at each pixel location (e.g., optical flow estimation [20]), which is computationally very demanding. On the other hand, the motion information available from compressed MPEG video is just one motion vector per macroblock, which is too sparse to perform motion segmentation. So, most of the compressed domain methods are based on the spatial information such as color, spatio-temporal similarities, and edge information [21].

To overcome the above difficulty, in this paper we propose a system which incorporates the motion information corresponding to few frames on either side of the current frame to enrich the motion information. The block diagram of the system for VOP generation is given in Fig. 1. The motion accumulation step takes the compressed video sequence as input and the motion vectors are decoded from the intercoded (P and B) frames. The motion vectors are accumulated over a few frames from the reliable macroblocks. Then the temporally accumulated motion vectors are subjected to median filtering and further spatially interpolated to get the dense motion field. The interpolation step assigns a motion vector to each pixel

Manuscript received August 16, 2001; revised September 25, 2003. This paper was recommended by Associate Editor H. Watanabe.

R. V. Babu is with Center for Quantifiable Quality of Service in Communication Systems, NTNU, Trondheim, Norway (e-mail: venkat@Q2S.ntnu.no).

K. R. Ramakrishnan is with Department of Electrical Engineering, Indian Institute of Science, Bangalore 560 012, India (e-mail: krr@ee.iisc.ernet.in).

S. H. Srinivasan is with Satyam Computer Services Ltd., Bangalore 560 025, India (e-mail: SH_Srinivasan@satyam.com).

Digital Object Identifier 10.1109/TCSVT.2004.825536

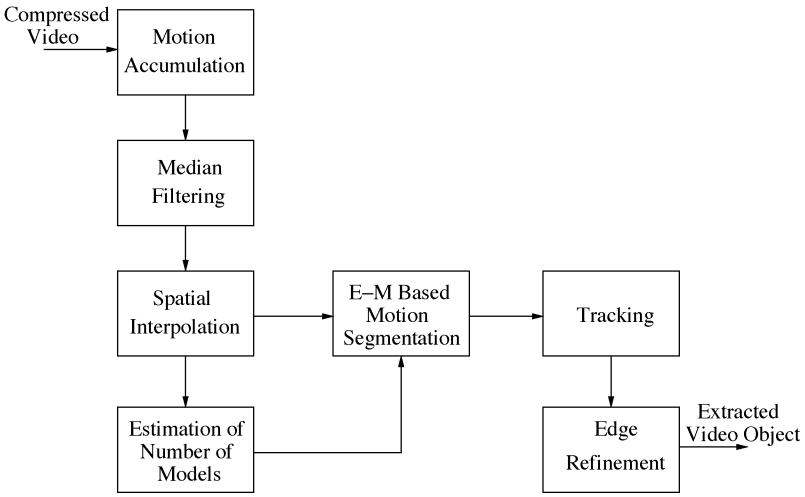


Fig. 1. Overview of the proposed system.

in the frame. Temporal accumulation and spatial interpolation techniques are explained in detail in the next section.

The dense motion vectors and the number of motion models are given as input to the segmentation module. Since each video object can be characterized by the motion information, an affine parametric motion model is used to describe the corresponding object region. Given the number of motion models, estimating the corresponding motion parameters from the dense motion vectors is difficult due to insufficiency of data. This problem is overcome by the expectation maximization (EM) algorithm [22], which is an iterative technique that alternately estimates and refines the segmentation and motion estimation. Determining the number of motion models is a crucial step in object segmentation. The algorithm based on K-means clustering is proposed for estimating the suitable number of motion models to avoid splitting or merging of objects. Once the initial segmentation of the objects are obtained, the following frames are tracked further temporally to generate a sequence of VOPs. The VOs are further subjected to the edge refinement phase, where pixels belonging to the edge regions are assigned to the correct VO. Details of the EM algorithm and estimation of the number of objects are given in Section III followed by the edge refinement algorithm in Section IV. The experimental results are discussed in Section V, and concluding remarks in Section VI.

II. Motion Accumulation and Spatial Interpolation

MPEG compressed video provides one motion vector for each macro block of size 16 × 16 pixels. Initially, the motion vectors are scaled appropriately to make them independent of frame type. This is accomplished by dividing the motion vectors by the difference between the corresponding frame number and the reference frame number (in display order). Then, the motion vectors are rounded to the respective nearest integers. In the case of bidirectionally predicted macroblocks, reliable motion information is obtained either from the forward or backward motion vector depending upon which of the reference frames (I/P) is closer. If backward prediction is chosen, the sign of the motion vector is reversed after normalization.

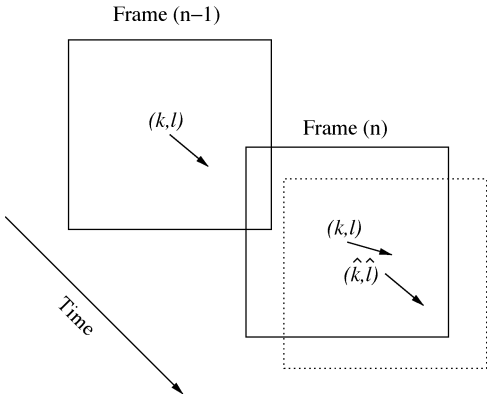


Fig. 2. Relative motion of the macroblock of the previous frame $n-1$ with respect to the corresponding macroblock of current frame n. The box drawn with dotted lines is the current position of the macroblock of the previous frame.

In the process of accumulating motion vectors, the motion vector obtained from the current macroblock of the current frame n is assigned to the center pixel (k,l) of that macroblock. Let $m_x^{kl}(n-c)$ and $m_y^{kl}(n-c)$ represent the motion vectors along the horizontal and vertical directions for the macroblock centered at (k,l) in frame $(n-c)$. Then, the new position for this macroblock in the current frame can be estimated as

$$(\hat{k},\hat{l}) = (k,l) + \sum_{f=n-1}^{n-c} \left(m_x^{kl}(f), m_y^{kl}(f)\right). \tag{1}$$

The motion vector $(m_x^{kl}(n-c), m_y^{kl}(n-c))$ in the $(n-c)$th frame is assigned to the new position $(\hat{k},\hat{l})$ with respect to the current frame. Fig. 2 explains the above process for the case of $c=1$.

Motion accumulation is also done by tracking frames in the forward direction from the current frame. This is achieved by keeping few future frames in the buffer. In forward tracking, the motion vectors are accumulated according to the following equation:

$$(\hat{k},\hat{l}) = (k,l) - \sum_{f=n+1}^{n+c} \left(m_x^{kl}(f), m_y^{kl}(f)\right) \tag{2}$$

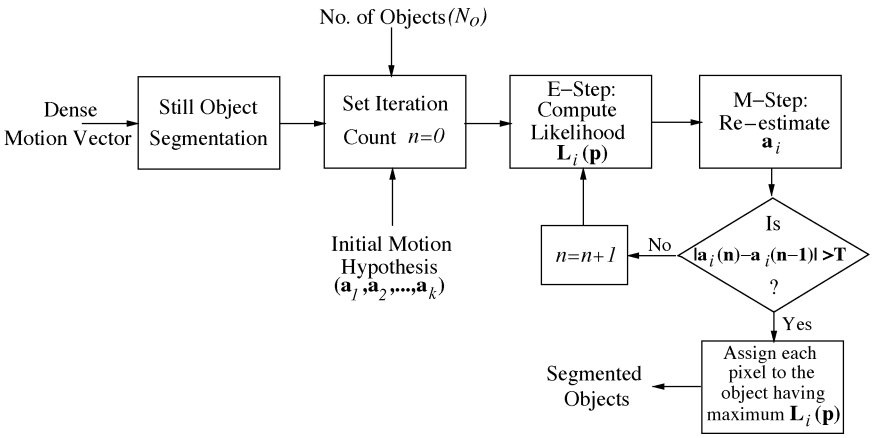


Fig. 3. Various modules of EM-based segmentation.

where the motion vector $(m_x^{kl}(n+c), m_y^{kl}(n+c))$ in the $(n+c)$th frame is assigned to the new position $(\hat{k}, \hat{l})$ with respect to the current frame. Each frame approximately provides one additional motion vector per macroblock. The error introduced by the farther frames in the unidirectional motion accumulation method is reduced very much by this bidirectional method, which reduces the distance between the current frame and the end frames. The results obtained by both methods are discussed in Section V.

Only the reliable motion vectors are gathered in the above process. The reliability of the motion vector is given by the discrete cosine transform (DCT) error energy of the corresponding macroblock. If the total error of the macroblock is less than a threshold T_{err} and the error variation of each block in the corresponding macroblock is within another threshold T_{var}, then the motion vector of the macroblock is considered to be reliable and each block within this macroblock is assigned the same motion vector. If the macroblock is unreliable or intracoded, then the motion information for each block is interpolated from the neighboring macroblocks.

This motion accumulation is performed over few frames on either side of the current frame. The accumulated data is further processed to remove the noisy motion information. A two-dimensional (2-D) median filter is used to remove the noise from the accumulated sparse motion vectors. This filter operates individually on nonzero elements of horizontal and vertical motion data. The set of motion vectors obtained by the above process is sparse and nonuniformly spaced. Thus, a Delaunay triangle-based surface interpolation [23] scheme is used to get the dense motion field for the current frame. This interpolation technique always fits the surface that passes through the given data points. The dense motion field obtained is further processed by a Gaussian filter to get a smoother dense motion field.

III. Object Segmentation by the EM Algorithm

First, the static object (usually the static background) is segmented by assigning the pixels with zero motion to a single layer. This is done just to reduce the computational burden involved in the EM algorithm. The remaining pixels with motion are segmented into different layers by applying the EM algorithm. The details of this EM-based segmentation is illustrated in Fig. 3. Given the number of motion models[1] N_o and the corresponding initial motion hypothesis expressed in terms of affine parameter vectors $\{\mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_{N_o}\}$ (each $\mathbf{a}_i$ is a six-dimensional vector), the EM algorithm alternates between the E-step and M-step until convergence.

A. E-Step

The E-step computes the probabilities associated with the classification of each pixel as belonging to the kth class (i.e., having motion parameter $\mathbf{a}_k$)

$$L_k(\mathbf{p}) = \Pr(\text{Motion at pixel } \mathbf{p} \text{ is represented by } \mathbf{a}_k). \quad (3)$$

For the kth motion model, the computed motion vector for pixel $\mathbf{p}$ is given by

$$\mathbf{u}_k(\mathbf{p}, \mathbf{a}) = \mathbf{\Pi}(\mathbf{p})\mathbf{a}_k \quad (4)$$

where $\mathbf{p} = [x\ y]^T$ is the vector representing the position of the pixel in the image plane, and $\mathbf{a}_k^T = [a_1\ a_2\ a_3\ a_4\ a_5\ a_6]$ is the affine parameter vector which characterizes the motion of the kth object as follows:

$$\mathbf{\Pi}(\mathbf{p}) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix}. \quad (5)$$

Let

$$R_k^2(\mathbf{p}) = \left(\mathbf{u}_k(\mathbf{p}, \mathbf{a}) - \mathbf{v}(\mathbf{p})\right)^2 \quad (6)$$

be the squared residual between the predicted motion $\mathbf{u}_k(\mathbf{p}, \mathbf{a})$ and the interpolated motion $\mathbf{v}(\mathbf{p})$ at pixel location $\mathbf{p}$.

Then the likelihood of $\mathbf{p}$ belonging to class k is given by

$$L_k(\mathbf{p}) = \frac{e^{-\frac{R_k^2(\mathbf{p})}{2\sigma^2}}}{\sum_{j=1}^{N_o} e^{-\frac{R_j^2(\mathbf{p})}{2\sigma^2}}} \quad (7)$$

where σ^2 controls the fidelity of the affine model fit to the dense motion vectors. A typical value of σ ranges from 0.1 to 0.3.

[1]Here the number of motion models N_o represents the appropriate number of objects in the video sequence. See Section III-C for the estimation of N_o.

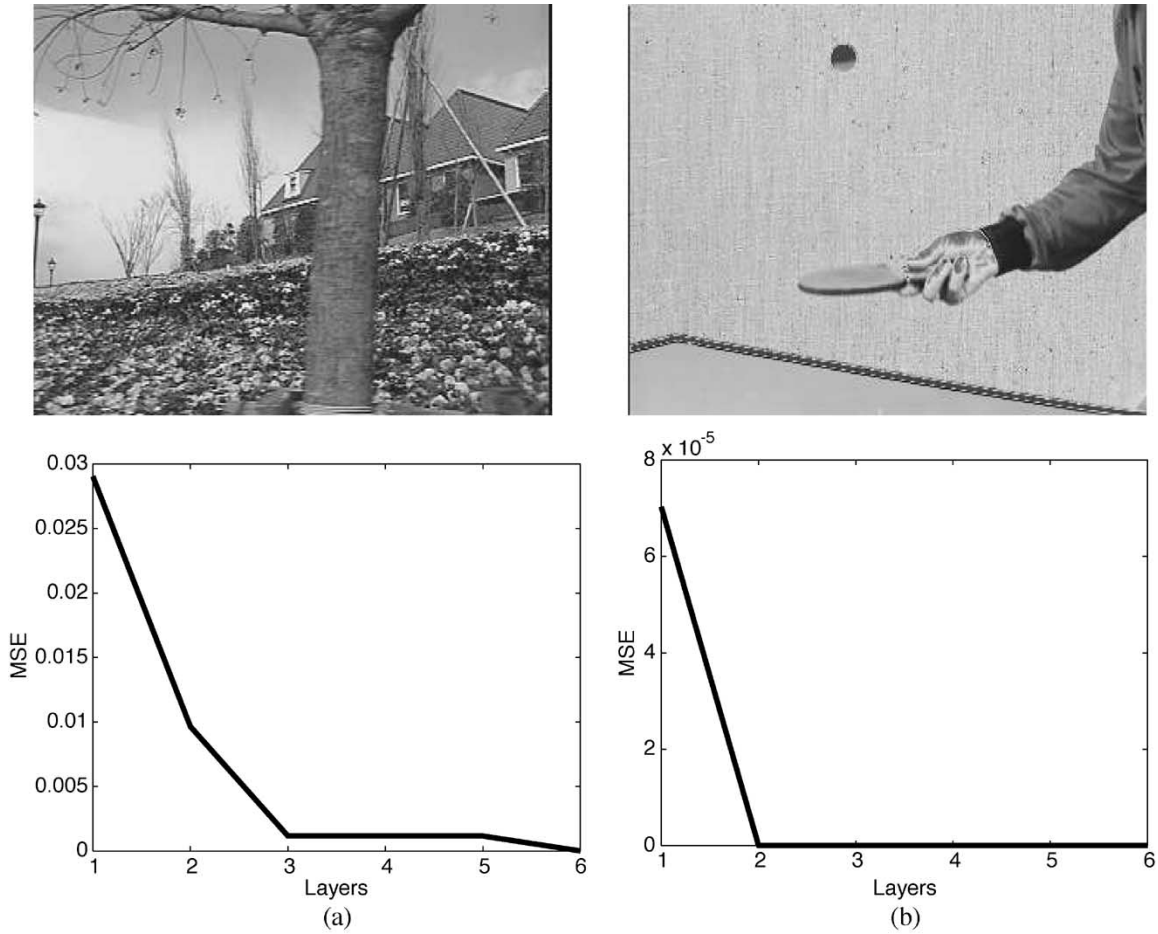


Fig. 4. MSE for various layers of (a) the flower garden sequence and (b) the table tennis sequence using K-means clustering.

B. M-Step

The M-step refines the motion model estimates given the new classification arrived at the E-step.

The motion model parameters are refined by minimizing an error function using a weighted least-squares estimation. The function to be minimized is

$$J(\mathbf{a}_k) = \sum_{\mathbf{p}\epsilon\mathbf{R}} L_k(\mathbf{p}).R_k^2(\mathbf{p}) \tag{8}$$

where $\mathbf{p}$ represents the position of the pixel within the square region $\mathbf{R}$ with respect to a common origin.

It can be shown [24] that the estimated motion parameters $\mathbf{a}_k$ of class k is a solution to

$$\mathbf{M}_k.\mathbf{a}_k = \mathbf{B}_k \tag{9}$$

with

$$\mathbf{M}_k = \sum_{\mathbf{p}\epsilon\mathbf{R}} L_k(\mathbf{p})\bar{\mathbf{\Pi}}^T(\mathbf{p})\bar{\mathbf{\Pi}}(\mathbf{p}) \tag{10}$$

and

$$\mathbf{B}_k = \sum_{\mathbf{p}\epsilon\mathbf{R}} L_k(\mathbf{p})\mathbf{\Pi}^T(\mathbf{p})\mathbf{v}(\mathbf{p}). \tag{11}$$

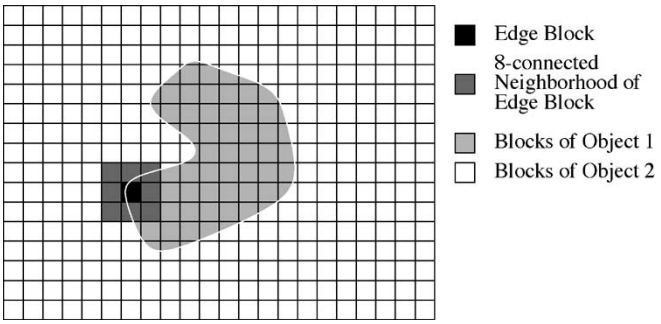


Fig. 5. Illustration of a typical edge-block and its eight-connected neighborhood.

In the affine case the matrix $\mathbf{M}_k$ obtained from (10) is of size 6×6, and $\mathbf{B}_k$ obtained from (11) is a column vector of dimension 6. The estimated motion parameters are given by

$$\mathbf{a}_k = \mathbf{M}_k^{-1}\mathbf{B}_k. \tag{12}$$

In (10) and (11), the summation is applied within each square region $\mathbf{R}$. In our simulation, we used $\mathbf{R}$ as a nonoverlapping 8×8 block of image plane.

After few iterations between the E-step and M-step, the final video object plane is obtained by hard thresholding the posterior probability $L_k(\mathbf{p})$. Typically four to six iterations are sufficient

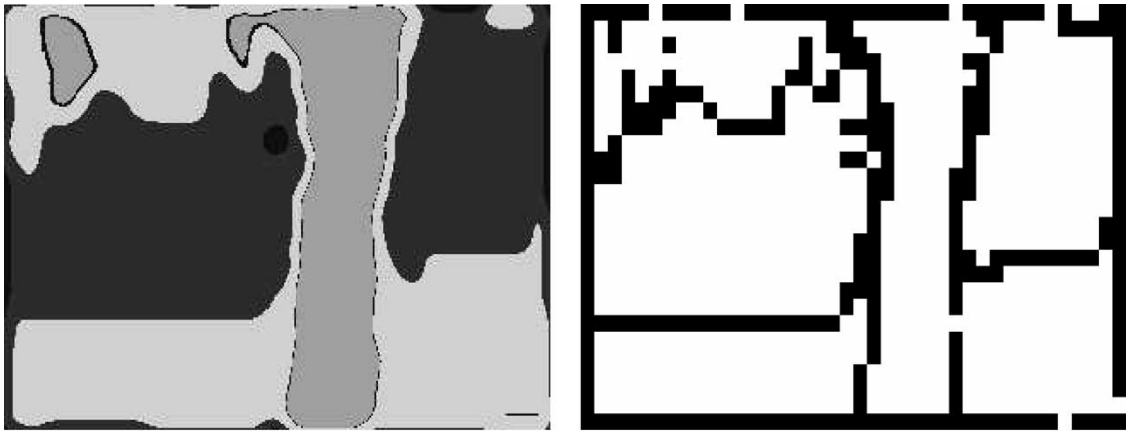


Fig. 6. (a) Coarse VOP mask and (b) the corresponding edge-blocks of the fourth frame of flower garden sequence.

to segment the object layers. Each pixel will be assigned to a distinct class, according to

$$\mathcal{Z}_k(\mathbf{p}) = \begin{cases} 1, & L_k(\mathbf{p}) > L_j(\mathbf{p}), \forall j \neq k \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The final VOP mask for the kth layer is given by $\mathcal{Z}_k(\mathbf{p}), \forall \mathbf{p}$.

After obtaining the segmentation of the initial frame, the same motion parameters are used for tracking the future frames. This reduces the computational overhead by avoiding the need to perform the EM iteration for subsequent frames. The estimated parameters should be updated by the EM algorithm whenever the clustering error of the current frame exceeds a predefined threshold. However, this tracking technique holds good only when no objects enter or leave the scene. In such a case, the number of objects in the present video frame has to be determined again, prior to segmentation.

C. Estimating the Number of Motion Models

The determination of the number of motion models (layers) is an important issue, because the final segmentation is very sensitive to the number of motion models. If the number of motion models is less, then the objects are merged, resulting in undersegmentation. On the other hand, if the number of motion models is large, then it results in splitting the objects which leads to oversegmentation. Thus, it is essential to determine the appropriate number of motion models before starting the segmentation process.

To determine the number of motion models, we extract the affine parameters from each nonoverlapping square region of the dense motion field whose variance is less than a predefined threshold τ. This is done to avoid considering those blocks that lie at the object boundaries and thereby consist of more than one motion model. Ignoring such blocks makes the motion models more reliable. In our simulations, the motion vectors are normalized to lie in the $[-1, 1]$ interval and the value of τ is set at 0.1. The affine parameters are estimated by standard linear regression techniques. The regression is applied separately on each motion component because x affine parameters depend only on the x component of the motion field and the y affine parameters depend only on the y component of motion field. The affine model captures the motion information and is represented by

$$\mathbf{a}_i^T = \left[\sum_{\mathbf{p}\epsilon\mathbf{R}} \mathbf{\Pi}(\boldsymbol{p})^T \mathbf{\Pi}(\boldsymbol{p})\right]^{-1} . \sum_{\mathbf{p}\epsilon\mathbf{R}} \mathbf{\Pi}^T(\mathbf{p})\mathbf{v}(\mathbf{p}). \tag{14}$$

With the regressor $\mathbf{\Pi}(\mathbf{p})$, the linear least-squares estimate of the affine parameter $\mathbf{a}_i$ for the ith block is given by (14).

Let $\{\mathbf{a}_1\ \mathbf{a}_2\ \ldots,\ \mathbf{a}_M\}$ be the set of affine vectors corresponding to the initial set of motion hypothesis obtained from M nonoverlapping blocks having at least one representative for each object in the video. The number of initial hypotheses M is very high compared to the number of moving objects in the video due to the redundancy in the initial hypothesis. Now the problem is to find the number of objects in the video and the corresponding representative motion hypothesis for each video object from the initial hypothesis. A method based on K-means clustering [25] is proposed to find the number of moving objects and the corresponding motion hypothesis.

All of the affine motion models obtained from the initial hypothesis are clustered using a K-means clustering algorithm and the cluster centers are taken as representative for each object. K-means clustering is an iterative technique which takes the number of motion models and the randomly chosen initial cluster means as input and assigns each motion model to the class of nearest cluster mean by minimizing the performance index [mean square error (MSE)] as follows:

$$E = \sum_{l=1}^{N} \sum_{i\epsilon\Lambda_l^{(j)}} \left\| \mathbf{a}(i) - \boldsymbol{\mu}_l^{(j+1)} \right\|^2 \tag{15}$$

where N denotes the number of clusters and $\Lambda_l^{(j)}$ denotes the set of motion models assigned to cluster l after the jth iteration, and μ_l denotes the mean of the lth cluster. This iteration is terminated if $\boldsymbol{\mu}_l^{(j+1)} = \boldsymbol{\mu}_l^{(j)}$. The index E gives the sum of the squared distances of each sample from their respective cluster means.

To find the appropriate number of motion models for the given dense motion field, the performance index E is computed by increasing the number of clusters N step-wise, with the ini-

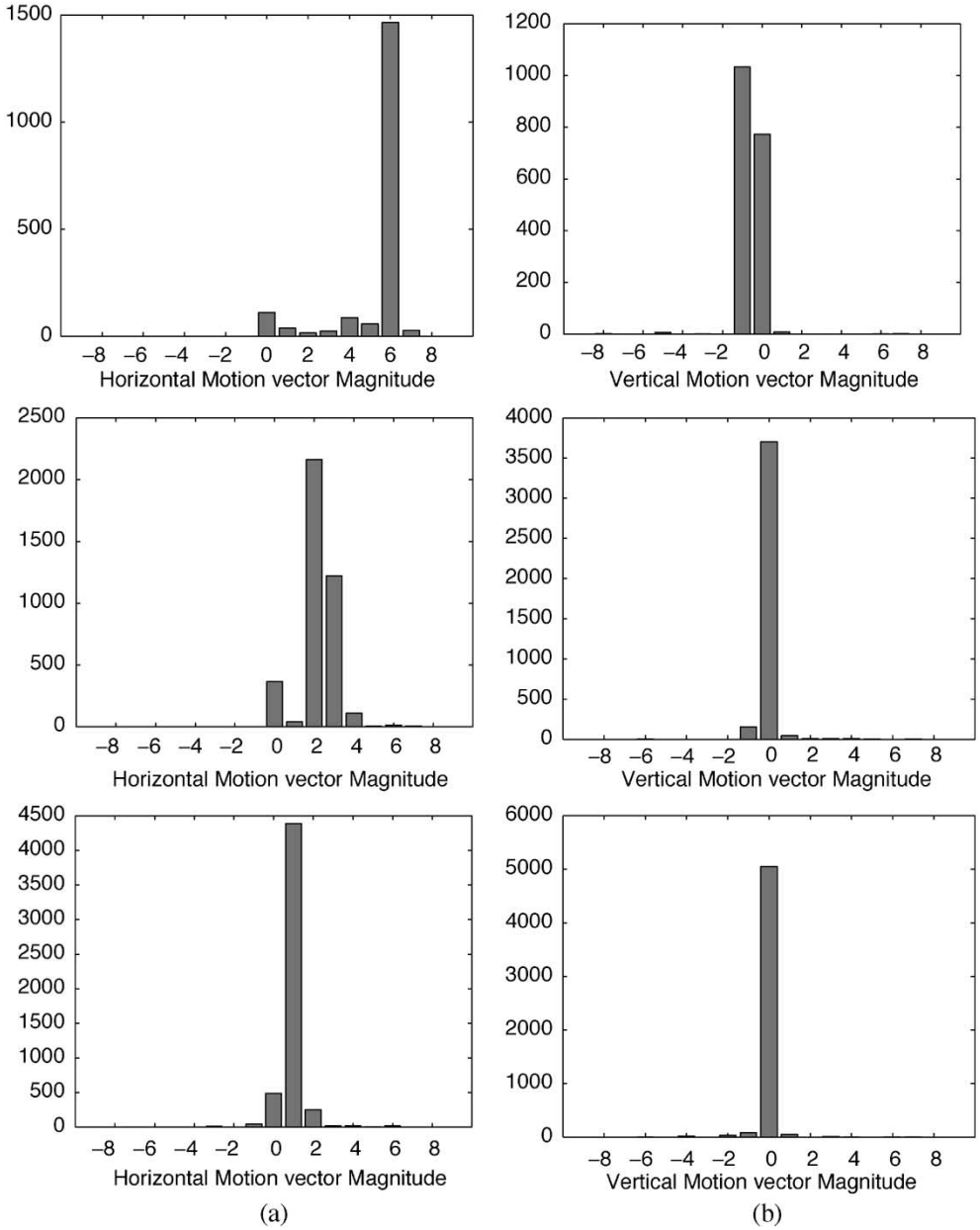


Fig. 7. Histograms of (a) horizontal and (b) vertical components of motion vectors corresponding to each object.

tial value set to 1. Since the performance index E converges to a local minima, we use multiple restarts for each number of clusters with randomly chosen cluster centers and pick the minimum value of E. The number of classes N_o, beyond which the decrease in E is less than a small threshold ζ, is chosen as the appropriate number of motion models. The typical value of ζ is chosen to be between 5%–10% of the maximum error. The plot in Fig. 4 shows the variation of E with respect to the number of motion models for the two test sequences "flower garden" and "table tennis." The graphs clearly indicate that the number of motion models for the "flower garden" sequence is 3 and 2 for the "table tennis" sequence without the static background.

IV. Edge Refinement

The video objects obtained from the EM algorithm suffer from poor edge localization. This is due to the insufficient motion information obtained from the macroblocks and the interpolation performed to get a dense motion vector which smooths out the edges between the objects. Therefore, it is essential to get good edge location of each object for further use in applications such as MPEG-4.

To get the better edges of the VOs, the VOPs obtained from the previous stage is subjected to further processing. First, the blocks through which the edges of the objects (which are called edge-blocks) pass through and the eight connected neighboring blocks are decoded. This is to take into account the possibility of the edges extending beyond the given edge-block. The notion of an edge-block and its eight-connected neighborhood is illustrated in Fig. 5, where two objects are shown along with a *single* edge-block and its neighborhood. The pixels within the edge-blocks are to be classified to the correct object. Here the classification is done based on the direction of motion of each pixel within the edge-block.

Fig. 8. Fourth frame of the original flower garden sequence.

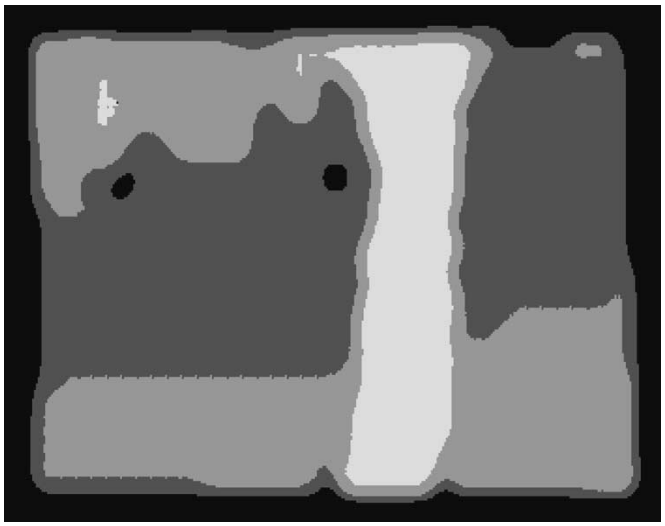


Fig. 9. Extracted VOP mask for the fourth frame of the flower garden sequence using unidirectional (forward) motion accumulation with two frames.

Fig. 10. Extracted VOP mask for the fourth frame of the flower garden sequence using bidirectional motion accumulation with one frame on either side.

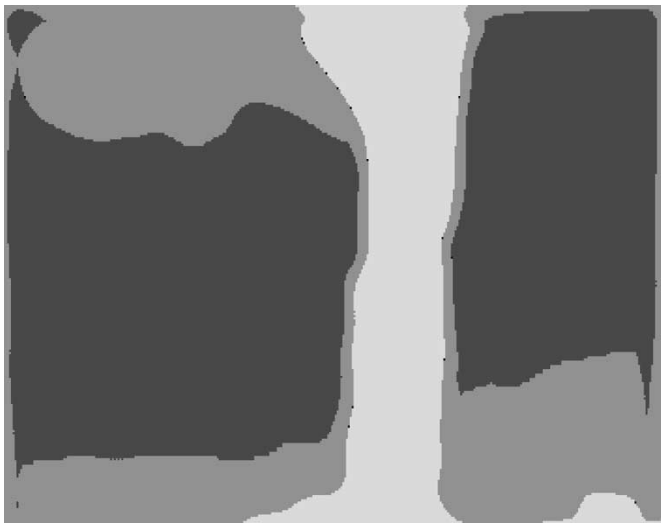


Fig. 11. Extracted VOP mask for the fourth frame of the flower garden sequence using robust optical flow method.

Based on the representative motion vectors, the search region in the previous frame is determined and the blocks falling within the search region are decoded. To classify each pixel to an object, a small block $N \times N$ (usually 3×3 or 5×5) centered around each pixel (n_1, n_2) is matched against the previous frame with the search range localized by the representative motion vectors. To reduce the computational burden involved in getting the best matching block, an adaptive search window for each object is determined based on its motion vector distribution. The maximum size of the search window is limited to $(N+2) \times (N+2)$. The minimum mean absolute difference (MAD) matching criteria is used to find the best matching block. The MAD value is computed for all of the objects by localizing the search window with the corresponding representative motion vectors. Finally the pixel will be assigned to the object whose MAD value is minimum and is within a predefined threshold ζ_{MAD}.

The notations and definitions followed in the sequel are as follows:

- m_x^{ij} and m_y^{ij} represent the motion vectors at position (i,j) along the horizontal and vertical directions. m_x^{kl}, $m_y^{kl} \in [m_{\min}, \ldots, m_{\max}]$.
- Let $I(i,j;k)$ represent the intensity value at pixel location (i,j) of frame k.

Definition 1—Block: A set of pixels within an $N \times N$ box is defined as a block. Mathematically,

$$B^{xy} = \{(i,j) : 0 \le i - x \le N-1;\ 0 \le j - y \le N-1\} \tag{16}$$

defines a block starting at (x,y). Typically, $N = 8$. B will be used to represent any general block.

Definition 2—Subblock: Let M^{kl} define a subblock, centered at (k,l) and of size $(2\beta+1) < N$. These subblocks are used for matching purposes, and typically $\beta = 1, 2$. Mathematically,

$$M^{kl} = \{(i,j) : -\beta \le i - k \le \beta \text{ and } -\beta \le j - l \le \beta\}. \tag{17}$$

M will be used to represent any subblock.

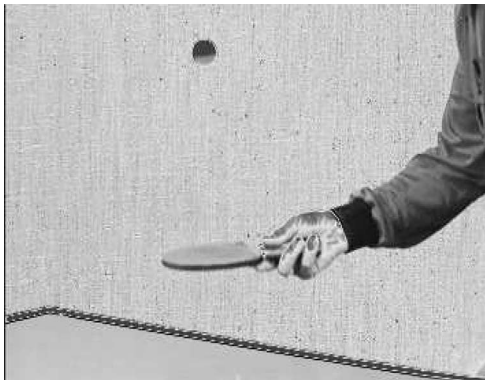


Fig. 12. Fifth frame of the original table tennis sequence.

Definition 3—Edge Block: A block B is said to be an edge-block if it contains pixels corresponding to two or more objects. Mathematically, let $O_i, 0 \leq i \leq N_o$, be the objects in the given scene (which are required to be segmented). That is,

$$O_i = \{(x, y) : \mathcal{Z}_i(x, y) = 1\}. \tag{18}$$

Then, a block B is said to be an edge-block if

$$B \subset \bigcup_{i \in \mathcal{K}} O_i \tag{19}$$

where $\mathcal{K} \subset [1, \ldots, N_o]$ and $\#\mathcal{K} \geq 2$ where # denotes the cardinality of a set.

Against this background, the steps involved in the edge refinement processes are listed as follows.

1) Edge-Block Determination:
   a) Determine the set of edge blocks $\mathcal{E}$, defined as
   $$\mathcal{E} = \{B : B \text{ is an edge block}\}. \tag{20}$$
   b) Let $\mathcal{T} = \{N_B : N_B$ be the eight-connected neighborhood of block $B \; \forall B \in \mathcal{E}\}$. Decode all of the blocks, in current frame n, which are elements of the set
   $$\mathcal{U} = \mathcal{T} \cup \mathcal{E}. \tag{21}$$
   The edge blocks for the fourth frame of the flower garden sequence is shown in Fig. 6(b).
2) Computation of Representative Motion Vectors:
   a) Collect the accumulated motion vectors corresponding to each object. Let $\mathcal{F}_i$ be the set of motion vectors corresponding to all of the blocks that are *completely* within i^{th} object, i.e.,
   $$\mathcal{F}_i = \left\{\left(m_x^{kl}, m_y^{kl}\right) \; \forall (k, l) \in \{B : B \subset O_i\}\right\}. \tag{22}$$
   b) Remove the noisy motion vectors obtained from the noisy segmentation. Let $\mathcal{C}_{ij}$ be the set of motion vectors common to objects i and j, i.e.,
   $$\mathcal{C}_{ij} = \mathcal{F}_i \cap \mathcal{F}_j. \tag{23}$$
   For each $(m_x, m_y) \in \mathcal{C}_{ij}$, let
   $$\mathcal{K}_i^{m_x, m_y} = \left\{(k, l) : \left(m_x^{kl}, m_y^{kl}\right) \in \mathcal{F}_i \text{ and } \left(m_x^{kl}, m_y^{kl}\right) = (m_x, m_y)\right\}. \tag{24}$$
   Then, the removal of noisy motion vectors is accomplished as follows:

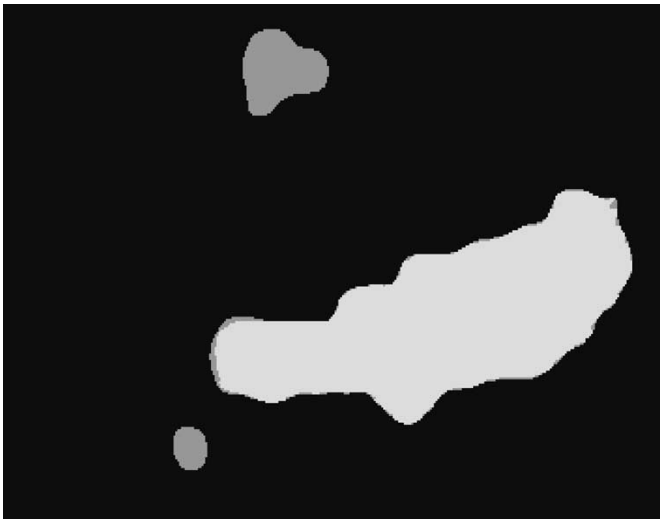


Fig. 13. Extracted VOP mask for the fifth frame of table tennis sequence using unidirectional (forward) motion accumulation with two frames.

Fig. 14. Extracted VOP mask for the fifth frame of the table tennis sequence using bidirectional motion accumulation with two frames on either side.

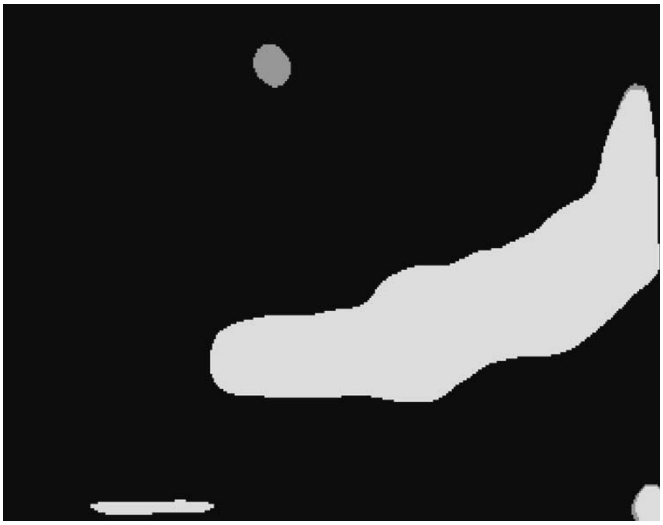


Fig. 15. Extracted VOP mask for the fifth frame of the table tennis sequence using the robust optical flow method.

Fig. 16. Extracted hand object from frame 5 of the table tennis sequence.

Fig. 17. Extracted ball object from frame 5 of the table tennis sequence.

if $(\#\mathcal{K}_i^{m_x,m_y} > \#\mathcal{K}_j^{m_x,m_y})$ **then**

$$\mathcal{F}_j' = \mathcal{F}_j \setminus (m_x, m_y)$$

else

$$\mathcal{F}_i' = \mathcal{F}_i \setminus (m_x, m_y)$$

where $\mathcal{F}_i'$ indicates the set of motion vectors of object i after removing noisy motion vectors and $(\setminus)$ represents the set subtraction operation.

c) Determine the representative motion vector values, along the horizontal and vertical directions, by taking the median value. That is, Let

$$\mathcal{Q}_i = \left\{(k,l) : \left(m_x^{kl}, m_y^{kl}\right) \in \mathcal{F}_i' \text{ and } (k,l) \in O_i\right\}. \quad (25)$$

Form arrays A_i^x and A_i^y containing the motion vectors $(m_x^{kl}, m_y^{kl}) \forall\, (k,l) \in \mathcal{Q}_i$. Then rm_i^x and rm_i^y, indicating the representative motion vectors for object i along the horizontal and vertical directions, are given as

$$rm_i^x = \text{Med } A_i^x \quad (26)$$

$$rm_i^y = \text{Med } A_i^y \quad (27)$$

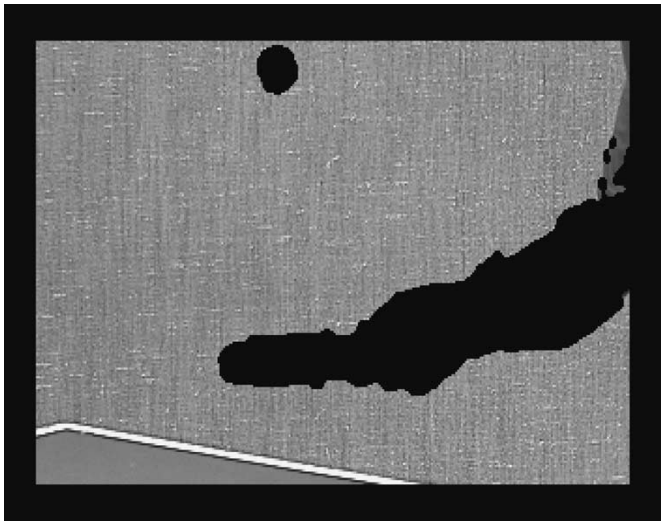


Fig. 18. Extracted background object from frame 5 of the table tennis sequence.

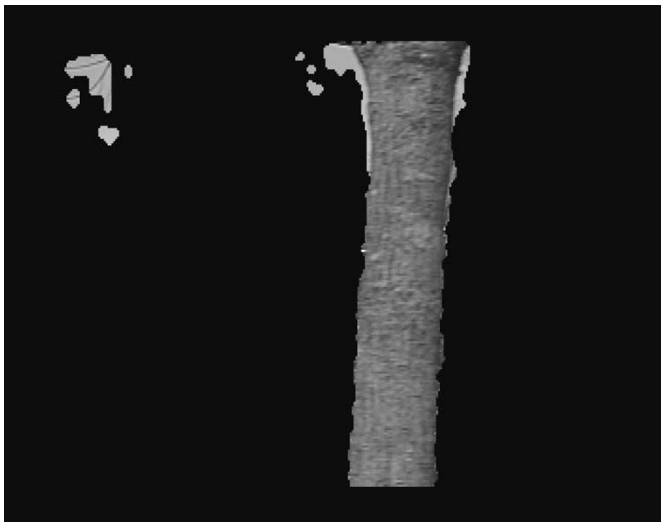


Fig. 19. Extracted tree object from frame 4 of the flower garden sequence.

where Med represents the median filtering operation.

3) Determination of the Search Range:

a) Find the histograms of the horizontal and vertical motion vectors for each object. Let $P_i^x(m_x)$ and $P_i^y(m_y)$ represent the histograms along the horizontal and vertical directions for object i, spanning the motion vectors for all $(k,l) \in \mathcal{Q}_i$.

b) Determine the horizontal and vertical search range, in the previous frame, for each object as follows:

$$S_i^x = \{m_x : P_i^x(m_x) > T \text{ and } |rm_i^x - m_x| = 1\} \quad (28)$$

$$S_i^y = \{m_y : P_i^y(m_y) > T \text{ and } |rm_i^y - m_y| = 1\}. \quad (29)$$

T is the threshold on the value of the bins.

c) For each pixel $(k,l) \in B^{xy} \in \mathcal{U}$, find the search window in the previous frame, corresponding to object i, as follows: $SW_i(M^{kl}) = \{M^{mn} : (m-k) = m_x \,\forall\, m_x \in S_i^x \text{ and } (n-l) = m_y \,\forall\, m_y \in S_i^y\}$. Decode blocks which contain the subblocks

defined by $SW_i(M^{kl}) \forall (k,l) \in B^{xy} \in \mathcal{U}$, $i \in [1,\ldots,N_o]$.

For example, Fig. 7 shows the histograms of horizontal and vertical components of motion vectors corresponding to the "tree," "flower bed," and "background" objects of flower garden sequence. The representative motion vectors corresponding to the objects are (6,−1), (2,0), and (1,0). The search range determined for the corresponding objects are (6,{−1, 0}), ({2,3}, 0), and (1,0) with T set at 15% of $\#\mathcal{Q}_i$.

4) Final Segmentation:
   a) Match a given subblock M^{kl} in the current frame n against subblocks (in the previous frame, $n-1$) defined by $SW_i(M^{kl})$ using the MAD criteria, given as

$$\mathrm{MAD}_i(M^{kl}) = \min d(M^{kl}, M) \quad \forall\, M \in SW_i(M^{kl}), \quad i \in [1,\ldots,N_o] \tag{30}$$

where d denotes the MAD distance between two subblocks and is given as

$$d(M^{kl}, M^{mn}) = \sum_{i,j=-\beta}^{+\beta} |I(k+i,l+j;n) - I(m+i,n+j;n-1)| \tag{31}$$

with the equation indicating explicitly that the search takes place in the previous frame (frame $n-1$).

   b) For each pixel $(k,l) \in \mathcal{U}$, assign (k,l) to the object whose MAD value is minimum and less than a threshold ζ_{MAD}. More precisely,

```
Let κ = arg min_i MAD_i(M^kl). Then,
if (MAD_κ(M^kl) ≤ ζ_MAD) then
    (k,l) ∈ O_κ
else
    (k,l) ∈ O_o
```

where O_o represents the object corresponding to the outliers.

   c) The binary masks of the objects O_κ are finally subjected to morphological operations for noise removal. First the binary masks are subjected to *clean* (removes isolated pixels, i.e., ones surrounded by zeros) and *fill* (fills isolated interior pixels, i.e., zeros surrounded by ones) operations, and finally the spurious edges are removed by *closing* operation.

This edge refinement process is done only for the pixels in the edge-blocks and their neighborhood. These pixels typically occupy a small fraction of the space of the entire image plane. Additionally, the search is localized by the prior motion knowledge of each object. These factors considerably reduce the computational burden compared to the traditional optical-flow algorithms where the prior knowledge of object motion is not available.

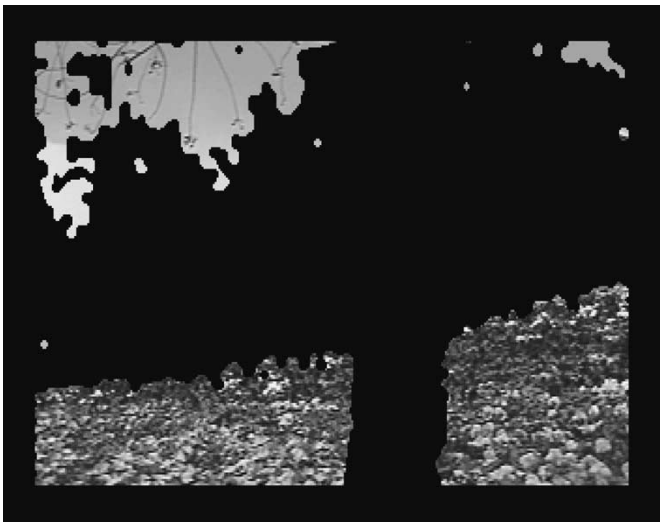


Fig. 20. Extracted flower bed object from frame 4 of the flower garden sequence.

Fig. 21. Extracted background object from frame 4 of the flower garden sequence.

V. Results and Discussion

Simulation results[2] have been obtained with the "flower garden" and "table tennis" sequences, both having fairly complicated motion among the objects. In every simulation, we have used an 8×8 square block for affine parameter estimation in the M-step, and the value of σ^2 in the E-step is kept at 0.01. The EM algorithm converges within four iterations, which was initialized with the cluster centers obtained from the K-means model selection step. For the flower garden sequence, apart from the background, the other three independently moving object layers were extracted. Though the flower garden sequence has a single camera motion without any independently moving objects, the depth information of the region is reflected in the motion vectors. For example, the tree which is closer to the camera moves faster than the flower bed behind it. Thus, the motion clustering algorithm segments the objects

[2]In all of the simulations, the picture border macroblocks are not considered for segmentation.

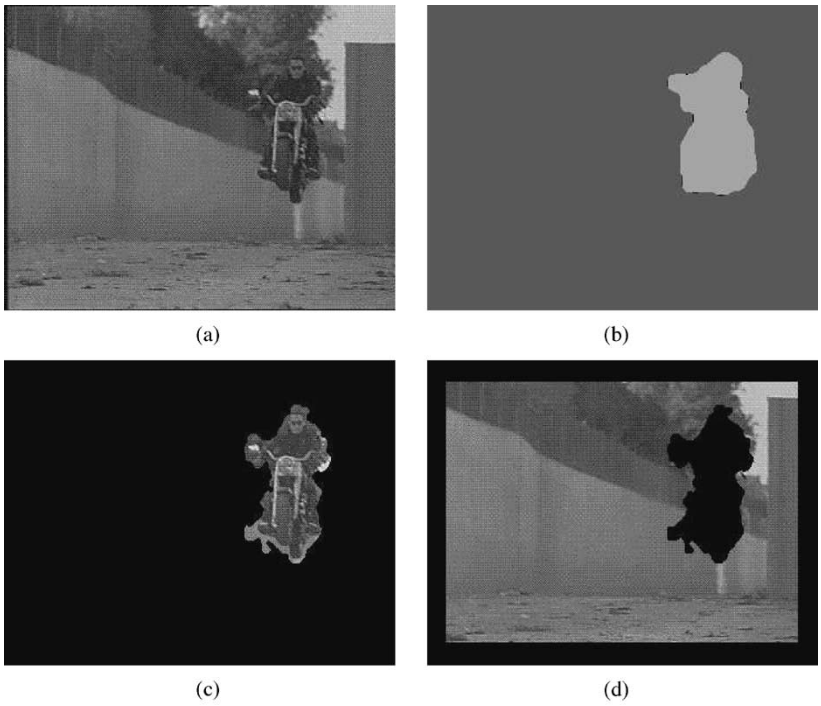


Fig. 22. (a) Original 66th frame of the bike sequence. (b) Coarse segmentation result. (c)–(d) Extracted objects after edge refinement.

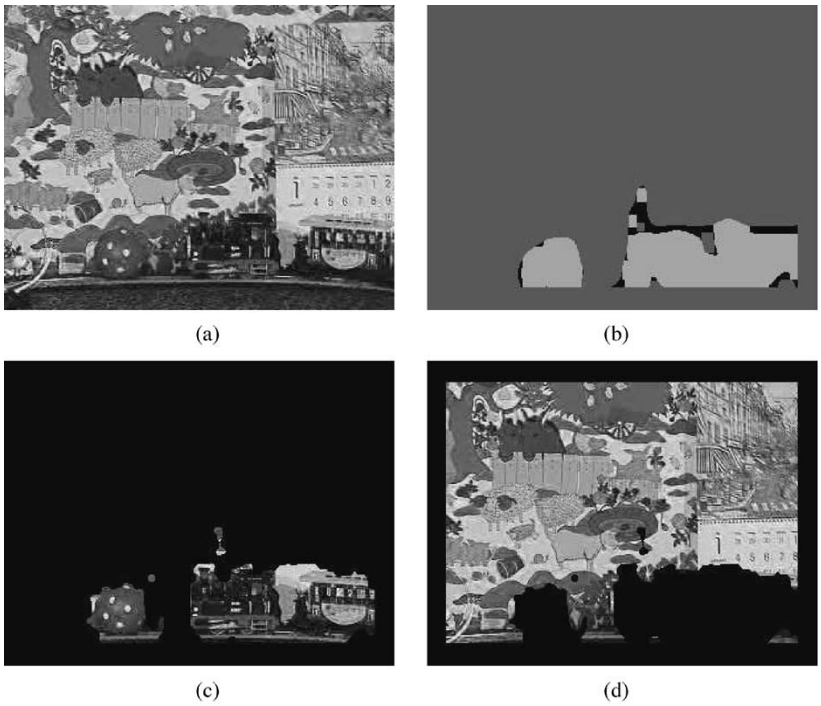


Fig. 23. (a) Original 25th frame of the mobile and calendar sequences. (b) Coarse segmentation result. (c)–(d) Extracted objects after edge refinement.

based on its relative depth level. The object masks for the fourth frame of the flower garden sequence (Fig. 8) obtained by the unidirectional (forward) motion accumulation method and the bidirectional method are shown in Figs. 9 and 10. The coresonding segmentation result obtained by Black and Anandan's robust optical-flow estimation method [26] is shown in Fig. 11. Though the results of both forward and bidirectional methods are almost similar in flower garden sequence, the better performance of the bidirectional method for the table tennis sequence (frame no. 5 shown in Fig. 12) is evident from Figs. 13 and 14. In the table tennis sequence out of three VOPs, the black one corresponds to the static background and the other two are independently moving objects. The corresponding segmentation result obtained with the dense motion vectors generated by Black and Anandan's method is given in Fig. 15 for comparison. The blurring effect at the object boundary in the proposed method is due to the insufficient information provided by the motion vectors of compressed video stream

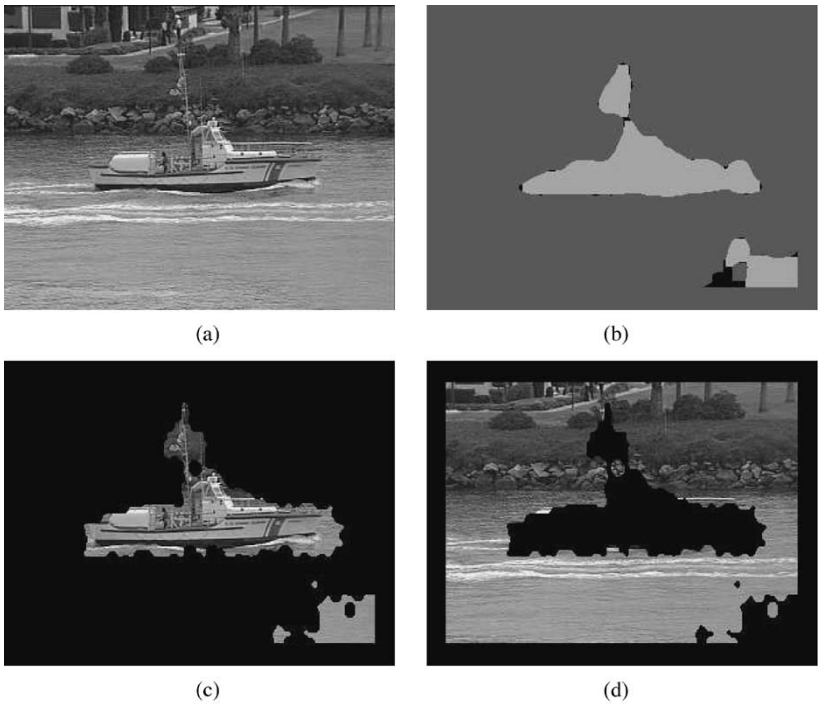


Fig. 24. (a) Original 150th frame of the coast guard sequence. (b) Coarse segmentation result. (c)–(d) Extracted objects after edge refinement.

and the smoothening effect of the spatial interpolation from the sparse motion data. Among the proposed methods, the bidirectional method provides a better object boundary than the unidirectional method. This is due to the fact that the reliability of the motion information decreases as the temporal distance increases from the current frame. The VOs extracted after edge refinement are shown in Figs. 16–21 for the table tennis and flower garden sequences. Segmentation results for the "bike," "mobile-calendar," and "coast guard" sequences are shown in Figs. 22–24. In all of the above experiments, a matching block size of 5×5 is used for edge refinement.

VI. Conclusion

In this paper, an automatic video object segmentation scheme has been proposed in the compressed domain. The performance of the system has been demonstrated on the flower garden and table tennis sequences. The system takes the sparse motion vectors from the compressed video stream as the only input. The sparse motion information is enriched by a motion accumulation procedure. The number of motion models for the interpolated dense motion field is automatically determined without the user's assistance using K-means clustering procedure. The EM algorithm is initialized with the cluster centers to avoid the local minima and for faster convergence. The tracking stage uses the converged motion models in the initial segmentation for subsequent frames, thereby avoiding iteration. The proposed edge refinement algorithm brings out exact contours of the object by decoding minimal number of blocks and makes the segmentation results suitable for MPEG-4 applications such as video editing and manipulation. This algorithm can be implemented in a massively parallel environment, which gives the hope for online object segmentation.

References

[1] Overview of the MPEG-4 Standard, V.18—Singapore Version, ISO/IEC JTC1/SC29/WG11 N4030, Mar. 2001.
[2] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997, pp. 1190–1203.
[3] P. Salembier and F. Marques, "Region-based representations of image and video: segmentation tools for multimedia services," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1147–1169, Dec. 1999.
[4] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Processing*, vol. 3, pp. 625–638, Sept. 1994.
[5] S. Ayer and H. Sawhney, "Layered representation of motion video using robust maximum-likelihood estimation of mixture models and and mdl encoding," in *Proc. 5th ICCV*, Cambridge, MA, June 1995, pp. 777–784.
[6] T. Darrell and A. P. Pentland, "Cooperative robust estimation using layers of support," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 474–487, May 1995.
[7] G. D. Borshukov, G. Bozdagi, Y. Altunbasak, and A. M. Tekalp, "Motion segmentation by multistage affine classification," *IEEE Trans. Image Processing*, vol. 6, pp. 1591–1594, Nov. 1997.
[8] T. Meier and K. N. Ngan, "Video segmentation for content-based coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1190–1203, Dec. 1999.
[9] D. Zhong and S. F. Chang, "An integrated approach for content-based video object segmentation and retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1259–1268, Dec. 1999.
[10] C. Gu and M. C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 572–584, Sept. 1998.
[11] M. Kim, J. G. Choi, D. Kim, H. Lee, M. H. Lee, C. Ahn, and Y. S. Ho, "A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1216–1226, Dec. 1999.
[12] F. Dufaux, F. Moscheni, and A. Lippman, "Spatiotemporal segmentation based on motion and static segmentation," in *Proc. IEEE Conf. Image Processing*, vol. 1, Oct. 1995, pp. 306–309.
[13] M. Gelgon and P. Bouthemy, "A region level graph labeling approach to motion based segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997, pp. 514–519.
[14] I. Patras, E. A. Hendriks, and R. L. Lagendijk, "Video Segmentation by Map Labeling of Watershed Segments," Delft University of Technology, Tech Rep. ICT -00-01, Nov. 2000.

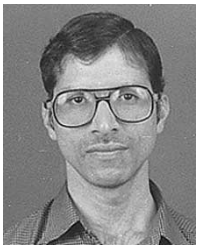
[15] Y. Weiss and E. H. Adelson, "Slow and Smooth: A Bayesian Theory for the Combination of Local Motion Signals in Human Vision," MIT AI Lab, Tech. Rep. 1624, Feb. 1998.
[16] N. Brady and N. O'Connor, "Object detection and tracking using an EM-based motion estimation and segmentation framework," in *Proc. IEEE Int. Conf. Image Processing*, 1996, pp. 925–928.
[17] D. P. Elias, "The Motion Based Segmentation of Image Sequences," Ph.D. dissertation, Trinity College, Dept. of Engineering, Univ. of Cambridge, Aug. 1998.
[18] N. Vasconcelos and A. Lippman, "Empirical Bayesian EM-based motion segmentation," in *Proc. IEEE CVPR*, 1997, pp. 527–532.
[19] P. H. S. Torr, R. Szeliski, and P. Anandan, "An integrated Bayesian approach to layer extraction from image sequences," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, pp. 297–303, Mar. 2001.
[20] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.
[21] O. Sukmarg and K. R. Rao, "Fast object detection and segmentation in MPEG compressed domain," in *Proc. IEEE TENCON*, Kuala Lumpur, Malaysia, Sept. 2000.
[22] A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B*, vol. 39, pp. 1–38, 1977.
[23] D. Sandwell and T. David, "Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data," *Geophys. Res. Lett.*, vol. 14, no. 2, pp. 139–142, 1987.
[24] Y. Weiss, "The Motion Bayesian Motion Estimation and Segmentation," Ph.D. dissertation, Massachusetts Inst. of Technol., Cambridge, May 1998.
[25] A. M. Tekalp, *Digital Video Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[26] M. J. Black and P. Anandan, "The robust method of multiple motions: parametric and piecewise-smooth flow fields," *Comput. Vis. Image Understanding*, vol. 63, pp. 75–104, Jan. 1996.

R. Venkatesh Babu received the B.E. degree in electrical engineering from Bharathiar University, Coimbatore, India, in 1993, the M.E. degree from Madurai Kamaraj University, Madurai, India, in 1995, and the Ph.D. degree in electrical engineering from the Indian Institute of Science, Bangalore, India, in 2003.

He is currently pursuing his post-doctoral research at the Center for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, through the ERCIM fellowship. His research interests include image and video processing, computer vision, and video streaming.

K. R. Ramakrishnan (M'00) received the B.E., M.E., and Ph.D. degrees in electrical engineering from the Indian Institute of Science, Bangalore, India in 1974, 1976, and 1983, respectively.

He is currently a Professor with the Department of Electrical Engineering, Indian Institute of Science. His research interests include image processing, computer vision, medical imaging, watermarking, and multimedia communication.



S. H. Srinivasan received the Ph.D. degree from the Indian Institute of Science, Bangalore, in 1993.

He was a freelance researcher, teacher, and consultant until 2001. He is currently with the Applied Research Group, Satyam Computer Services Ltd., Bangalore, India. His fields of interest include multimedia, audio processing, computer vision, and networking.