COMPRESSED DOMAIN MOTION SEGMENTATION FOR VIDEO OBJECT EXTRACTION

R. Venkatesh Babu and K. R. Ramakrishnan

Department **of** Electrical Engineering, Indian Institute **of** Science, Bangalore 560 012**,** India.
email addresses: {rvbabu,krr}@ee.iisc.ernet.in

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 few frames to enhance the motion information, which are further spatially interpolated to get a dense motion vectors. The final segmentation from the dense motion vectors is obtained by applying 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. Finally, the segmented objects are temporally tracked to obtain the video objects. This work has been carried out in the context of the emerging MPEG-4 standard which aims at interactivity at object level.

1. INTRODUCTION

Though MPEG-I and **2** provide a good representation of digital audio-visual information, the interactivity is only at the frame level. The emerging MPEG-4 standard address this interactivity at object level by defining Audio-visual Objects (AVO). 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 (VO) 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 (VOP), which are the instants of a semantic object in the scene. But extracting objects from digital video is still a challenging task among 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. Though it is not possible to unambiguously define a criterion function for a semantic video object, an object can be characterized based on its homogeneity in motion information. Since, Independently Moving Object can be characterized by coherent motion over the object region, the problem now reduces to clustering pixels that exhibit similar type of motion.

Though, much work is being done in the area of motion based video object segmentation in pixel domain [1], [2], [3], [4], 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 like optical flow estimation, which is computationally very demanding. On the other hand the motion information available from compressed

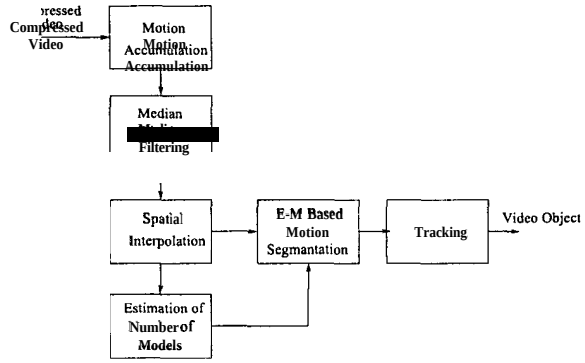

Fig. 1. Overview of the proposed system

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 [5].

To overcome the above difficulty, in this paper we propose a system which incorporates the motion information for a fewer number of 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 inter coded (P and B) frames. The motion vectors are accumulated over 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 in the frame. This temporal accumulation and spatial interpolation 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 EM algorithm [6], 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 sequence of VOPs. Details of EM algorithm and estimation of number of objects are given in Section **3** and the experimental results are given in Section 4.

0-7803-7402-9/02/$17.00 02002 IEEE

2. MOTION ACCUMULATION AND SPATIAL INTERPOLATION

The MPEG compressed video provides one motion vector for each macroblock of size 16x16 pixels. Before starting this motion accumulation process, all the motion vectors considered for this process are scaled appropriately to make the motion vectors 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 the display order). Then, the motion vectors are rounded to nearest integers. In the case of bidirectionally predicted macroblocks the 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.

In the process of accumulating motion vectors, the motion vector obtained from current macroblock of the current frame n is assigned to the center pixel of that macroblock, say (k,l). 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 given as:

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

The motion vector $(m_x^{kl}(n-c), m_y^{kl}(n-c))$ in $(n-c)$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 $c = 1$

The motion accumulation is also done by tracking the frames in 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:

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

Here, the motion vector $(m_x^{kl}(n+c), m_y^{kl}(n+c))$ in $(n+c)$th frame is assigned to the new position $(\hat{k},\hat{l})$ with respect to the current frame. Each frame approximately provides one motion vector per macroblock.

The error introduced by the farther frames in 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 4.

Only the reliable motion vectors are gathered in the above process. The reliability of the motion vector is given by the 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 reliable and each block within this macroblock is assigned the same motion vector. If the macroblock is unreliable or intra coded, then the motion information for each block is interpolated from the neighboring macroblocks.

This motion accumulation is performed over few frames from either side of the current frame. This accumulated data are further processed to remove the noisy motion information. A two dimensional median filter is used to remove the noise from the accumulated sparse motion vectors. This filter operates individually on non zero elements of horizontal and vertical motion data. The set of motion vectors obtained by the above process is sparse and non-uniformly spaced. So a delaunay triangle-based surface interpolation [7] 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.

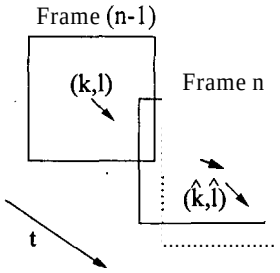

Fig. 2. Relative motion of the macroblock of previous frame $n-1$ with respect to the corresponding macroblock of current frame n. The box drawn with discontinuous lines is the current position of the macro block of previous frame.

3. OBJECT SEGMENTATION BY EM ALGORITHM

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

3.1. The E Step

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

$$L_k(\mathbf{p}) = Pr(\text{Pixel } \mathbf{p} \text{ moving with motion } k). \qquad (3)$$

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

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

where, $\mathbf{p} = [xy]^T$ is the vector representing the position of 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.

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

Let, $$R_k^2(\mathbf{p}) = (\mathbf{u}_k(\mathbf{p},\mathbf{a}) - \mathbf{v}(\mathbf{p}))^2 \qquad (6)$$

be the squared residual between the predicted motion $\mathbf{u_k(p,a)}$ and the interpolated motion $\mathbf{v(p)}$ at pixel location $\mathbf{p}$. Then the likelihood of $\mathbf{p}$ belonging to class k is given by

$$L_k(\mathbf{p}) = \frac{\exp(-R_k^2(\mathbf{p})/2\sigma^2)}{\sum_{j=1}^{K}\exp(-R_j^2(\mathbf{p})/2\sigma^2)} \qquad (7)$$

Where, σ^2 controls the fidelity of the affine model fit to the dense motion vectors.

[1] All the vectors and matrices are represented by bold letters

3.2. The M Step

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

The motion model parameters are relined by minimizing an error function using 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, p represents the position of pixel within the square region **R** with respect to a common origin.

It can be shown [8] that the estimated motion parameters $\mathbf{a}_k$ of class k is a solution to the following equation:

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

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

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

In the affine case the matrix M_k obtained from (10) is of size 6 by **6,** and $\mathbf{B}_k$ obtained from (11) is a column vector of size *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 R. In our simulation we used **R** as non-overlapping 8 by 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 $L_k(\mathrm{p})$. Typically 4 to 6 iterations are sufficient to segment the object layers. Each pixel will be assigned to a distinct class, according to:

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

the final **VOP** mask **for** the k th layer **is** given by $\mathbf{z}_k(\mathbf{p}),\ \forall\ \mathbf{p}$.

After obtaining the segmentation of the initial frame, the same motion parameters **F** are used for tracking the future frames. Thus reduces the computational overhead by avoiding the need to perform the EM iteration **for** subsequent frames. This tracking technique holds good only when no new objects enter or leave the scene.

3.3. Estimating the number of Motion Models

The determination of the number of motion models (layers) **is** an important issue, because the final segmentation heavily depends upon the number of motion models. **If** the number of motion models is less, then the objects are merged, resulting in under segmentation. On the other hand if the number of motion models is more, then it results in splitting the objects which leads to over segmentation. So 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 non-overlapping square regions of the dense motion field whose variance is less than a small threshold τ. 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 x component of the motion field and y affine parameters depends only on y component of motion field. The affine model captures the motion information and is represented by **(4).**

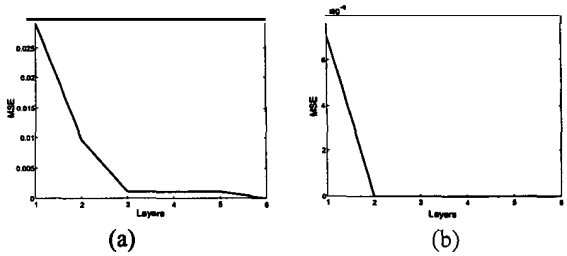

Fig. 3. MSE for various layers of (a) flower garden sequence and (b) table tennis using K-means clustering

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

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

Let $\{\mathbf{a}_1\ \mathbf{a}_2 \ldots, \mathbf{a}_M\}$ be the set of affine vectors corresponding to initial set of motion hypothesis obtained from M non-overlapping blocks having at least one representative for each object in the video. The number of initial hypothesis 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 is proposed to find the number of moving objects and the corresponding motion hypothesis. **All** 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).

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

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 from 1 onwards. 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 K, after which the decrease in E is less than a small threshold ζ, is chosen as the number of motion models. The typical value of ζ is chosen between 5 to 10 percent of the maximum error. Fig. 3 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 indicates that the number of motion models for 'flower garden' sequence is 3 and 2 for the 'table tennis' sequence without the static background.

4. RESULTS AND DISCUSSIONS

Simulation results' have been obtained with the 'flower garden' sequence and 'table tennis' sequence, both having fairly compli-

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

cated motion among the objects. In all simulation we used **8** by **8** square block for affine parameter estimation in the M-step, and the value of σ^2 in E-step is kept at 0.01. The EM algorithm converges within **4** iterations, which was initialized with the cluster centers obtained from the K-means model selection step. For 'flower garden' sequence, apart from the background, the other three independently moving object layers were extracted. The object masks obtained by unidirectional (forward) motion accumulation method and bi-directional method are shown in Fig. **4.** Though the results of both forward and bidirectional methods are almost similar in 'flower garden' sequence, the better performance of the bi-directional method for the 'table tennis' sequence is evident from Fig. **4.** In 'table tennis' sequence out of three VOPs, the black one corresponds to the static background and the other two are independently moving objects. In the 'table tennis' sequence, though the size of the ball is very small, our algorithm is capable of segmenting the ball from the other objects.The segmentation results obtained with the dense motion vectors generated by Black and Anandan's robust optical flow estimation method **[9]** are given in Fig. **4** (b,f) 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 and the smoothening effect of the spatial interpolation from the sparse motion data. Among the proposed method the bi-directional 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.

5. CONCLUSIONS

In this paper a compressed domain automatic video object segmentation scheme has been proposed. The performance of the system has been demonstrated on '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.

This algorithm can be implemented in a massively parallel environment, which gives the hope for online object segmentation. The above segmentation is performed only by taking the sparse motion vectors as input. The segmentation can be further improved by considering the spatial intensity values of the image frame, which can be obtained by partially decoding the MPEG stream.

6. REFERENCES

[1] Noel Brady and Noel O'Connor, "Object detection and tracking using an EM-based motion estimation and segmentation framework," in ***Proceeding of the IEEE International Conference on Image Processing,*** **1996,** pp. **925–928.**

[2] David P. Elias, ***The motion Based Segmentation of Image Sequences,*** Ph.D. thesis, Trinity College, Department of Engineering, University of Cambridge, Aug. **1998.**

[3] Nuno Vasconcelos and A. Lippman, "Empirical Bayesian EM-based motion segmentation," in ***Proceedings of the IEEE CVPR,*** **1997,** pp. **527–532.**

[4] Philip H. S. Torr, Richard Szeliski, and P. Anandan, "An integrated bayesian approach to layer extraction from image sequences," ***IEEE Trans. Pattern Analysis and Machine Intelligence,*** vol. **23,** no. **3,** pp. **297–303,** Mar. **2001.**

[5] O. Sukmarg and K. Rao, "Fast object detection and segmentation in mpeg compressed domain," in ***Proceedings of the IEEE TENCON 2000,*** Kuala Lumpur, Malaysia, Sept. **2000.**

[6] A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," ***Journal of the Royal Statistical Society*** *B,* vol. **39, 1977.**

[7] Sandwell and T David, "Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data," ***Geophysical Research Letters,*** vol. ***14,*** no. **2,** pp. **139–142, 1987.**

[8] Yair Weiss and E. H. Adelson, **"Slow** and smooth: A Bayesian theory for the combination of local motion signals in human vision," Tech. Rep. **1624,** MIT AI Lab, Feb. **1998,** CBCL Paper No. **158.**

[9] M. J. Black and P. Anandan, "The robust method of multiple motions: Parametric and piecewise-smooth flow fields," ***Computer Vision and Image Understanding,*** vol. ***63,*** no. 1, pp. **75–1**04, Jan. **1996.**

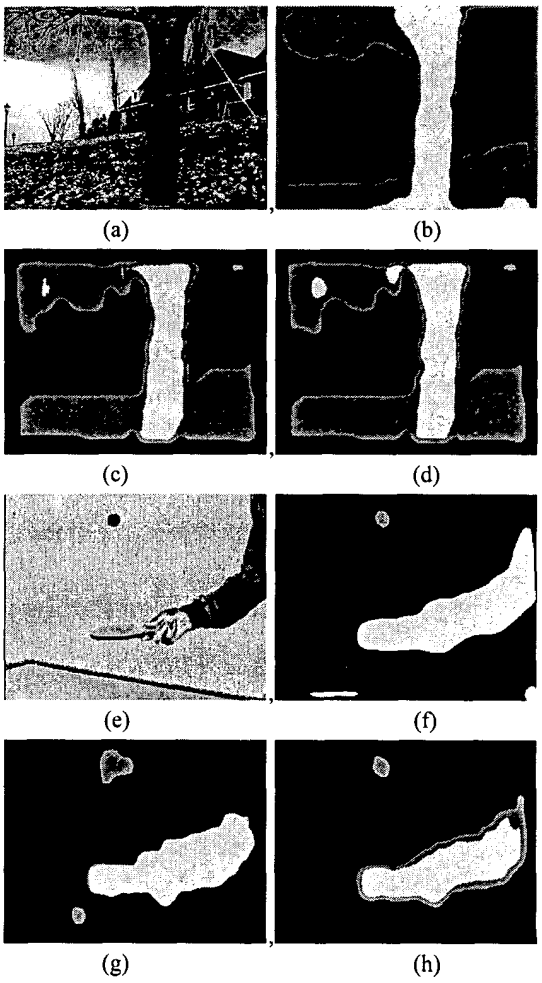

Fig. 4. Results: (a) 4th frame of the original flower garden sequence; (b) Segmentation by robust-optical flow method; (c) forward (d) bi-directional motion accumulation based segmentation; (e) 5th frame of the original table tennis sequence; (f) Segmentation by robust-optical flow method; (g) forward (h) bi-directional motion accumulation based segmentation