CONTENT-BASED VIDEO RETRIEVAL USING MOTION DESCRIPTORS EXTRACTED FROM COMPRESSED DOMAIN

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

Video content description has become an important task with the standardization effort of MPEG-7, which aims at easy and efficient access to visual information. In this paper we propose a system tu extract the features from compressed MPEG video based on the motion vector information. The global features like motion activity and camera motion parameters are extracted from the decoded motion vectors and the object features such as speed, area and trajectory are obtained after the object segmentation stage. The number of objects in a given video shot is determined by the proposed K-means clustering algorithm and the object segmentation is done by applying EM algorithm.

Keywords: Compressed Domain, Content-based Video Retrieval, Motion Descriptor, MPEG-7

1. INTRODUCTION

Multimedia search and retrieval has become a active field after the standardization of MPEG-7. The syntactic information used in MPEG-7 include color, texture, shape and motion. There are plenty of image/video retrieval systems [1] based on the spatial features such as color, texture and shape, where as systems using motion related information which distinguishes the image from a video are rare in literature [2]. There are few research work done in compressed domain video indexing and retrieval [3, 4, 5]. They use features such as motion vectors and DCT coefficients. Since the features are directly extracted from the compressed MPEG video, the costly overhead of decompressing and operating at pixel level is avoided.

So far all the compressed domain video indexing and retrieval techniques available in the literature [3, 4, 5] are operating at the frame level without considering the contents of the video. The underlying semantic content of the video is given by the characteristics of the objects present in the video. So it is essential to describe the video objects for efficient indexing and retrieval of the video. Though there are few papers [6] incorporate the above task in pixel domain, the computational cost involved in the pixel domain process is very high.

In this paper we try to extract various features of the video objects from the compressed video shot. Though defining a semantic video object is a difficult task, in most of the cases a video object can be defined as a coherently moving image region. The readily available motion vector information from the compressed MPEG video is used for segmenting the video objects from background. Presently the system takes the following frame based queries regarding (i) motion activity (ii) camera motion (zoom factor, pan and tilt rate) and the object based queries regarding (iii) approximate object area (iv) velocity and (v) trajectory.

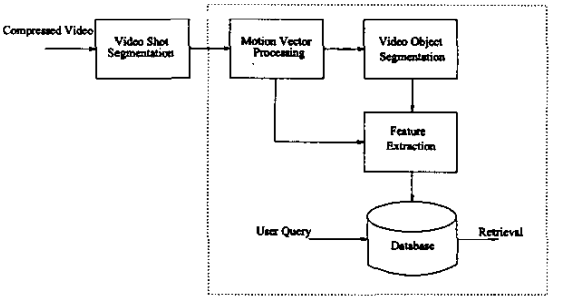

Fig. 1. Overview of the proposed system

2. SYSTEMOVERVIEW

The proposed system initially takes the compressed MPEG video shots and extract motion vectors by partial decoding of the MPEG stream. The proposed system consists of the following three stages (i) Motion vector processing (ii) Object segmentation and tracking (iii) Feature extraction. Fig.1 shows the various parts of the proposed system.

Since the motion vectors obtained from the MPEG video are noisy, they can not be directly used for object segmentation. To increase the reliability of motion information, the motion vectors of few neighboring frames on either side of the current frame are also used, the details of this process is given in the next section. The segmentation stage takes the motion information obtained from the previous stage for segmenting the coherently moving video objects by EM algorithm. The number of objects in the video is determined by a proposed K-means clustering algorithm. The segmented objects are tracked temporally to get the trajectory information of the object. Section 4 describes the aforementioned segmentation and tracking phase. Finally from the segmented object and the consecutive tracking, the features of the corresponding object such as velocity, location of object center and approximate area are computed for indexing the scene. The global features such as motion activity and camera motion are directly obtained from the motion vector processing block.

3. PROCESSING THE MOTION VECTORS

MPEG compressed video provides one motion vector (usually noisy) for each macro block of size 16 x 16 pixels. To remove noise, the motion vectors are subjected to the following steps (i) Motion Accumulation (ii) Determination of representative motion vectors.

0-7803-7448-7/02/$17.00 ©2002 IEEE

3.1. Motion Accumulation

Initially, the motion vectors are scaled appropriately to make them independent of frame type [3]. 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, 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.

The motion vector obtained from 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:

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

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 additional motion vector per macroblock.

3.2. Determination of representative motion vectors

After the motion accumulation process the representative motion vector for each macroblock is obtained by taking the median value of all the motion vectors falling within the corresponding macroblock region. The above process increases the reliability of the motion information by removing the noisy motion vectors present in the current frame. The representative motion vectors are given as input for the segmentation stage. Since we are not interested in extracting the exact shape of the object, this sparse motion information is sufficient to get the motion characteristics of the video object. Working with the sparse motion information reduces the computational burden involved in segmentation process to a greater extent.

4. COARSE VIDEO OBJECT SEGMENTATION

The objective of this segmentation stage is to group together the coherently moving object blocks from the background by applying EM algorithm. Given the number of motion models N_o (number of objects) and the corresponding initial motion hypothesis expressed in terms translational parameter vectors, the EM algorithm alternates between the E-step and M-step until convergence.

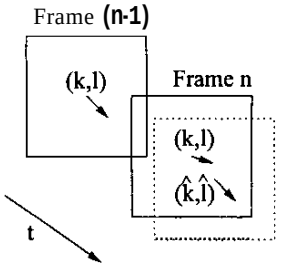

Fig. 2. Relative motion of the macroblock of previous frame $n-1$ with respect to the corresponding macroblock of current frame n. The box drawn with dotted lines is the current position of the macroblock of previous frame.

4.1. Determination of Number of Objects

The number of motion models is to be determined before starting the segmentation process. The determination of the number of motion models 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. All the motion models obtained from the motion accumulation phase are clustered using a K-means clustering algorithm by increasing the number of centers from one onwards and the mean square error (MSE) is observed. Since the clustering converges to a local minima, we use multiple restart for each number of clusters with randomly chosen cluster centers and pick the minimum value of MSE. The number of classes N_o, after which the decrease in MSE 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 15 percent of the maximum error.

4.2. The E Step

The E step computes the probabilities associated with the classification of each representative motion vector as belonging to jth class (*i.e.*, having motion parameter $\mathbf{a}_j$).

$$L_j(\mathrm{B}) = Pr(\text{Motion in macroblock B is represented by aj}). \qquad (3)$$

where, $\mathrm{B} = [kl]^T$ is the vector representing the position of macroblock in the image plane, and $\mathbf{a}_j^T = [v_x^j\ v_y^j]$ is the translation parameter vector which characterizes the motion of the jth object.

Let

$$R_j^2(\mathbf{B}) = \|\mathrm{a}_j - \mathbf{v}(\mathbf{B})\|^2 \qquad (4)$$

be the squared residual between the predicted motion 4 and the motion obtained from the motion accumulation stage $\mathbf{v}(\mathrm{B})$ at location B.

Then the likelihood of B belonging to class j is given by

$$L_j(\mathbf{B}) = \frac{e^{-R_j^2(\mathbf{B})/2\sigma^2}}{\sum_{i=1}^{N_o} e^{-R_i^2(\mathbf{B})/2\sigma^2}} \qquad (5)$$

where, σ controls the fidelity of the motion model fit to the dense motion vectors. Typical value of σ ranges from 0.1-0.3.

4.3. The M Step

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

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

$$J(\mathbf{a}_j) = \sum_{\forall \mathbf{B}} L_j(\mathbf{B}).R_j^2(\mathbf{B}) \tag{6}$$

The estimated motion parameters are given by

$$\mathbf{a}_j = \left[\sum_{\forall \mathbf{B}} L_j(\mathbf{B})\right]^{-1} \sum_{\forall \mathbf{B}} L_j(\mathbf{B})\mathbf{v}(\mathbf{p}) \tag{7}$$

After few iterations involving both the E- and M-steps, the macroblocks belonging to each object is obtained by hard thresholding the posterior probability $L_j(\mathbf{B})$. Typically **4** to 6 iterations are sufficient for segmentation. Each macroblock will be assigned to a distinct class, according to:

$$\mathcal{Z}_j(\mathbf{B}) = \begin{cases} 1 & : \quad L_j(\mathbf{B}) > T_l \\ 0 & : \quad \text{otherwise} \end{cases} \tag{8}$$

The final object mask for the j th motion model is given by $\mathcal{Z}_j(\mathrm{B})$, $\forall$ B. The threshold T_l is fixed in such a way to assign the macroblocks to the object which are very close to the object motion model (the typical value of T_l ranges from **0.7** -0.9).

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. This tracking technique holds good only when no new objects enter or leave the scene. This tracking phase is interrupted whenever **the** clustering error of the current frame with the previously estimated motion parameters exceed a threshold T_{err}, and the **new** motion parameters are estimated by EM algorithm.

5. FEATURE EXTRACTION

The features required for indexing the video shots are extracted from each frame after segmentation. The global shot attributes and the object based features used for indexing the video shots are explained below.

1. Motion Activity: This descriptor gives an idea about the 'pace of action' in the video segment. The value of the descriptor is high for more dynamical shots such as sport sequence and low for video-telephony sequences. The motion activity is measured by the standard deviation of the magnitudes of motion vectors [7].
2. Object Area: This gives the rough estimate of area the object measured from the object mask obtained from the segmentation stage. The object area is represented as the fraction of object macroblocks available in the frame.
3. Velocity of the Object: The object velocity **a** (estimated from the M step of EM algorithm) describes the speed of the object along horizontal and vertical direction. This value is updated whenever the EM iteration is performed.
4. Object Trajectory: The trajectory of the object is represented by two second order polynomials, one each for the horizontal and the vertical directions. Both trajectories are computed from the motion trail of the object center, represented by a sequence $\{x[i], y[i]\}, i \in 1, \ldots, \mathrm{N}$, where i indicates the temporal location of the object. Each shot is divided into blocks containing N number of frames (with an overlap of one frame) and the trajectories of the object for both horizontal and vertical directions are obtained by fining, in least square sense, a second order polynomial to the above trail.
5. **Camera Motion:** The global camera motion such as pan, zoom and tilt can be easily determined from the decoded motion vectors of the video shot. The camera motion parameters are determined for all P frames by the algorithm proposed in [8]. The zoom factors is given by,

$$s = \frac{\sum_{k=1}^{N} (w_k' - \bar{w}')^T (w_k - \bar{w})}{\sum_{k=1}^{N} \|w_k - \bar{w}\|^2} \tag{9}$$

The zoom factor indicates : $s > 1$: zoom in, $s < 1$: zoom out and $s = 1$: no zoom. The pan (q_3) and tilt (q_4) rates are given by,

$$\begin{pmatrix} q_3 \\ q_4 \end{pmatrix} = \frac{\bar{w}'}{s} - \bar{w} \tag{10}$$

where,

$$w_k' = \begin{pmatrix} x_k' \\ y_k' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \left[\begin{pmatrix} i_k \\ j_k \end{pmatrix} - \begin{pmatrix} i_o \\ j_o \end{pmatrix} \right] \tag{11}$$

represents the center of the kth macroblock with respect to the image-centered Cartesian axis, (i_k, j_k) is the center of the kth inter-coded macroblock in the current frame with motion vector (m_x^{kl}, m_y^{kl}) and (i_o, j_o) are the coordinates of the center of the image. Further, $\bar{w}' = \frac{1}{N}\sum_{k=1}^{N} w_k'$, $\bar{w} = \frac{1}{N}\sum_{k=1}^{N} w_k$; $w_k = w_k' + \begin{pmatrix} m_x^{kl} \\ -m_y^{kl} \end{pmatrix}$ and N is the number of inter coded macroblocks with nonzero motion vectors.

6. QUERY RESULTS *AND* DISCUSSIONS

The user is asked to submit a query by giving the values for global features (motion activity and camera motion) and object features (size, speed and trajectory) with their corresponding weights. For giving object trajectory information, the user is asked to sketch the trajectory as a sequence of vertices in the xy plane and the corresponding temporal information of each point is obtained by mapping these points linearly to [1, N], the length of the predefined block. Then the polynomial coefficients (second order) **are** extracted from the user sketch for both horizontal and vertical direction for matching purposes. The trajectory information is ma& independent of the location of the object by excluding the constant offset term from both the polynomials.

The total distance between the query and target is simply the weighted sum: $D_g = \sum_i w_i D_i$, where w_i's, the weights for the individual features, are assigned to values such that $\sum w_i = 1$ and D_i is the Euclidean distance between the query and target for ith feature. Fig. 3 shows the retrieved video shots and the corresponding query trajectory. Table 1 shows the query data[1] for the results shown in Fig. 3.

[1]For queries involving information on trajectories, the **weights** for **the** horizontal **and** vertical trajectories are set at 0.1.

	MA	Camera Motion			Object	
		ZF	PR	TR	Area (%)	Velocity $[m_x, m_y]$
Q1	1.2	1	2.5	0.5	10	[3, -0.51
W1	0.1	0.05	0.2	0.05	0.1	[0.2,0.1]
Q2	1	1	-2	1	15	[-3,1]
W2	0.05	0	0.1	0.1	0.15	[0.2,0.2]
Q3	2.5	1	0	0	25	[-4,-3]
w 3	0.1	0	0.1	0.1	0.1	[0.2,0.2]
Q4	2	1	2	0.5	10	[-4,0]
w 4	0.2	0.1	0.1	0.1	0.1	[0.2,0.2]
Q5	2.5	1	-3	-1	30	[0.5,0]
w 5	0.2	0.2	0.1	0.1	0.1	[0.2,0.1]

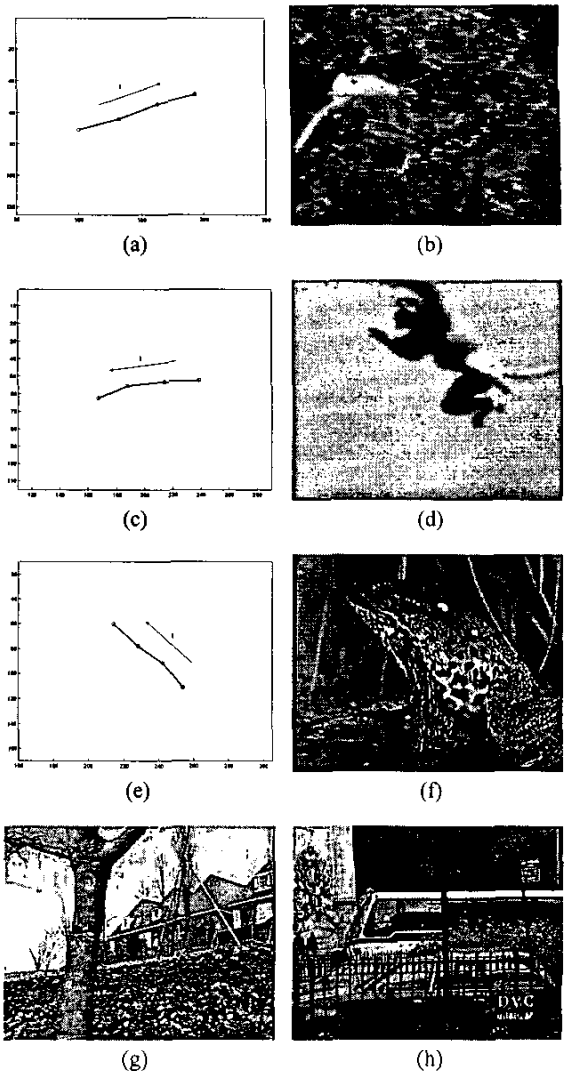

Fig. 3. (a), (c) and (e) are query trajectories; (b), (d) and (f) are corresponding frames from retrieved video; (g), (h) are the retrieved videos corresponding to the query Q4 and Q5 in Table I without query trajectory.

7. CONCLUSIONS

The use of motion information in the compressed domain allows for rapid analysis of the content of the video. A video indexing and retrieval system based on the motion information obtained from the compressed MPEG video has been presented. The number of objects in the video shot is determined by applying the proposed K-means algorithm on the refined motion data and the object features are obtained by segmenting the objects by EM algorithm. The global and object features with the user given weights are used for retrieval. The system can be further improved by considering the spatial features such as DCT dc coefficients that can be easily extracted from the MPEG video.

8. REFERENCES

[1] M. Flickner, H. Sawhney, W. Niblack, J. Aashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system:' ***IEEE Computer Magazine,*** vol. **28,** pp. **23–32,** Sept. **1995.**

[2] Sylvie Jeannin, Radu Jasinschi, Alfred She, T. Naveen, Benoit Mory, and Ali Tabatabai, "Motion descriptors for content-based video representation:' ***Signal Processing: Iimage Communication,*** vol. **16,** pp. **59-85, 2000.**

[3] V. Kobla, D. S. Doermann, and K-I. Lin, "Archiving, indexing and retrieval of video in compressed domain:' in ***SPIE Conference on Multimedia Storage and Archiving Systems,*** **1996,** vol. **2916, pp.** 78–89.

[4] V. Kobla, D. S. Doermann, K-I. Lin, and C. Faloutsos, "Compressed domain video indexing techniques using dct and motion vector information in MPEG video:' in ***SPIE Conference on Multimedia Storage and Archiving Systems,*** **1997,** vol. **3022,** pp. **200–211.**

[5] K. Yoon, D. F. DeMenthon, and D. Doermann, "Event detection from **MPEG** video in the compressed domain," in ***Int. Conf. on Pattern Recognition,*** Barcelona, Spain, **2000.**

[6] D. Zhong and S. F. Chang, **"An** integrated approach for content-based video object segmentation and retrieval:' ***IEEE Trans. on Circuits and Systems for yideo Technology,*** vol. **9,** no. **8,** pp. **1259-1268,** Dec. **1999.**

[7] Sylvie Jeannin and Ajay Divakaran, "MPEG-7 visual motion descriptors,'' ***IEEE Trans. on Circuits and Systems for yideo Technology,*** vol. **1**1, no. **6,** pp. **720–724,** June **2001.**

[8] Y. P. Tan, Drew D. Saur, S. R. Kulkarni, and P. J. Ramadge, "Rapid estimation of camera motion from compresses video with applications to video annotation:' ***IEEE Trans. on Circuits and Systems for Video Technology,*** vol. 10, no. 1, pp. **133–146,** Feb. **2000.**