Compressed Domain Video Retrieval using Object and Global Motion Descriptors ⋆

R. Venkatesh Babu, [1]

Center for Quantifiable Quality of Service in Communication Systems,
O.S. Bragstads plass 2E, N-7491 Trondheim, Norway,
e-mail: venkat@q2s.ntnu.no

K. R. Ramakrishnan

Department of Electrical Engineering, Indian Institute of Science,
Bangalore, Karnataka, India - 560 012.

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 to extract the object-based and global features from compressed MPEG video using the motion vector information for video retrieval. The reliability of the motion information is enhanced by a motion accumulation process. The global features like motion activity and camera motion parameters are extracted from the above enhanced motion information. The object features such as speed, area and trajectory are then obtained after the proposed object segmentation. The number of objects in a given video shot is determined by the proposed K-means clustering procedure. The object segmentation is done by applying EM algorithm.

Key words: Compressed Domain, Content-based Video Retrieval, Motion Descriptors, Object trajectory, MPEG-7

1 Introduction

With the advent of broadband networks, high-powered workstations and various compression standards, the multimedia information systems have became a very

⋆ An earlier, brief version of this paper has appeared in the Proceedings of the IEEE International Symposium on Circuits and Systems 2002, (ISCAS 2002) Vol. 4, pp. 141-144
[1] Corresponding Author Ph: +47 73 59 27 46, Fax: +47 73 59 69 73

important aspect of today's life. Due to the requirement of large storage space and processing power to handle visual media, there is a strong need for indexing and retrieval of visual information from a huge multimedia database. A straight forward approach to index the visual database is to represent the contents in textual form (e.g. Keywords and attributes). Here the visual database can be easily accessed by standard query language like SQL, however, this requires extra storage and a lot of manual processing. As a result there has been new focus on developing efficient tools for content- based indexing and retrieval.

'Multimedia search and retrieval' has become an 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 [9] based on the spatial features such as color, texture and shape, where as systems using motion-related information which distinguish images from a video are rare in literature [11]. Though there are many research works reported in 'compressed domain video indexing and retrieval' [12–14,19,21], they use features such as, motion vectors and DCT coefficients, at frame level. The global features are harnessed, while the contents of the frame remain unused. Since the features are directly extracted from the compressed MPEG video, the costly overhead of decompressing and operating at pixel level is avoided. The advantages of such an approach include the following:

(1) Size of uncompressed video is large whereas MPEG which combines several compression techniques, reduces the memory occupied by the same video. Processing time and memory requirements reduce to a greater extent if done in the compressed domain.
(2) The computational time required for decoding all the pixels is high. With an approach that requires only *partial* decoding of readily available motion vectors reduces the computational time involved in the classification process.
(3) With the availability of hardware encoders and decoders for MPEG, such an approach involving processing in the compressed-domain is amenable to hardware implementation, which can further reduce the computational time.

Most of the compressed domain video indexing and retrieval techniques available in the literature [12,13,21] operate 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 [22,6] that incorporate the above task in pixel domain, the computational cost involved in the pixel domain process is very high.

The motion features, which are not available for image indexing, in spite of being an important attribute for representing animated visual data, have been relatively rarely used to describe object activity [11]. Majority of the past work in 'video indexing and retrieval' include camera motion estimation techniques [1,13,18], tra-

jectory matching techniques [16,7] and action recognition techniques [2,5]. The performance of the similarity-based video retrieval systems show significant improvement by integrating motion based features along with the image features such as color, texture, etc [3]. Further, motion based queries are very useful in contexts where motion is significant.

In this paper we address the problem of video retrieval using the motion features extracted from the independently moving objects (IMOs) in compressed domain framework. 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. Though the motion vectors do not necessarily correspond to the real motion, it provides valuable information about the content of the video in the compressed domain context. The objective of our paper is to make use of this information to characterize the content of the video for retrieval purpose. 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.

The main contributions of the paper are:

(1) Obtaining the reliable motion information through motion accumulation which eliminates the noisy motion vectors.
(2) Estimating the appropriate number of motion models through the proposed K-means clustering technique.
(3) Adapting the EM algorithm for spatial segmentation of the sparse motion information.
(4) Extracting the object based features (approximate area, speed and trajectory) in compressed domain using the sparse motion information.
(5) Utilization of motion descriptors at global and object level for retrieval of generic videos.

The rest of the paper is organized as follows: Sec. 2 briefly reviews the retrieval systems using motion information, Sec. 3 gives the overview of the proposed system, Sec. 4 deals with the pre-processing of the motion vectors, Sec. 5 explains the coarse video object segmentation using EM algorithm, Sec. 6 explains the global as well as the object based feature extraction, Sec. 7 discusses about the experimental results and Sec. 8 concludes the paper.

2 Related Work

In this section we review few important works related to video indexing/retrieval. The related work are reported under the following two classifications i) Pixel-domain approach and ii) Compressed-domain approach. The pixel-domain approach assumes no compression and works on raw video data. The works under the second category make use of the information extracted from the compressed domain such as motion vectors and DCT coefficients.

2.1 Pixel-domain Approach

A statistical framework for motion-base video classification and retrieval is is described in [8]. Contrary to other proposed methods, here the main idea is to interpret the dynamic contents without any prior motion segmentation. The probabilistic model extracted from the given video sequence is expressed as a temporal Gibbs random field. The retrieval process is a Bayesian framework whose objective is to retrieve in a video database examples similar to a query video in terms of motion content. Courtney [4] proposes a segmentation, tracking and characterization of moving objects in order to determine a spatio temporal representation of the video shot constrained to static camera and known background. The motion analysis of the objects are then classified to identify events such as motion/rest, entrance/exit etc.

The system proposed in [3] allows query on object motion trails. Here the trajectory of an object is described by the ordered sequence of the motion vectors pertaining to the centroid of the object (amplitude and positions), for each time instant in the sequence. This system works with the uncompressed video. Nabil *et al,*. [15] propose a retrieval model which integrates the representations of individual moving objects in a scene with the spatio-temporal relations between them. Scenes and information pertaining to the scenes are retrieved by specifying both spatial and temporal properties of the object.

2.2 Compressed-domain Approach

Ardizzone *et al,*. [1] mainly use global motion field for similarity query. Here a frame is divided into 4 or 16 quadrants with each quadrant a motion features in terms of magnitude (average value) and direction (either average or dominant direction as well angle histogram with 90 directions of 4 degree each) are constructed. They also use a sequential labeling method to segment frames based on similarity of motion vectors (both magnitude and angle) and a clustering method to extract dominant regions with dominant motion using magnitude histogram of motion vec-

tors. These dominant regions are then defined by their size and average motion. Dimitrova and Golshani [7] used the approximate movement of macroblocks (macroblock tracing) given by the MPEG motion vectors are used to retrieve low-level motion information. The middle-level object motion information is obtained by averaging or clustering method for rigid and non-rigid objects respectively. At high-level analysis, the object trajectories are represented by chain code, supporting both exact-match and similarity match queries. Camera operation usually causes a global and dominant motion, which is an important feature in video indexing. However, camera operation estimation becomes less reliable when dominant motion is caused by a large object.

Sahouria and Zakhor [16] describe a model based video indexing system for street surveillance. Here the compressed video of a scene is analyzed by a segmentation and tracking program which extracts the trajectory (two dimensional curves parameterized by time) of the moving objects such as cars, people and bicycle etc. The coarse scale wavelet transform components are stored as keys in an index which are matched against the user drawn trajectory. Tan *et al,*. [20] modeled the camera operation as a combination of rotations about the three axes and a translation of the coordinates. The 6 parameters of the above mentioned projective transformation model are estimated by an iterative algorithm with outliers rejection. Also a closed form solution exists for a simplified faster version which considers only pan,tilt and zoom factor. Here the scene is assumed to be planar, which is approximately true when the object is far away from the camera. Kobla *et al,*. [13] estimate camera pan and tilt by detecting dominant motion which is determined by the largest bin in the directional histogram (in 8 directions). The zoom is detected by checking the Focus of Contraction (FOC) or Focus of Expansion (FOE). The $ViBE$ [19] system uses the the DC image sequence extracted from the DC components of the DCT transformations for the following functions: scene change detection, shot representation using key-frame and active browsing. Here the videos are handled only at frame level without considering the content of the video.

3 System Overview

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 (details given in the next section). The segmentation stage

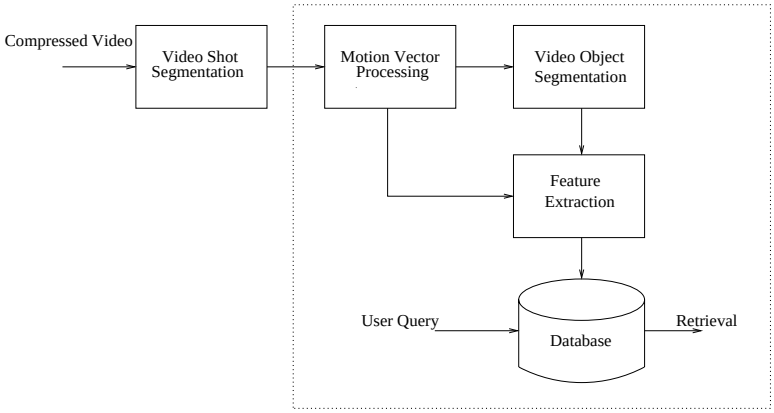


Fig. 1. Overview of the proposed system

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 variation of K-means clustering algorithm. The segmented objects are tracked temporally to get the trajectory information of the object. Section 5 describes the aforementioned segmentation and tracking phase. Finally from the segmented object and the subsequent 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.

4 Processing the Motion Vectors

First, the purpose of motion compensation in MPEG video compression standard [17] is to remove the temporal redundancy by finding a good match in the anchor frame for each inter coded macroblock.. Hence, the resultant motion vector does not necessarily represent the real motion of the pixels in the inter coded macroblock. So these MPEG motion vectors which give a noisy version of the real motion should be pre-processed before using. MPEG compressed video provides one motion vector for each macro block of size 16×16 pixels. To increase the reliability of the motion information, the motion vectors are subjected to the following steps (i) Motion Accumulation (ii) Determination of representative motion vectors.

4.1 Motion Accumulation

Initially, the motion vectors are to be scaled appropriately to make them independent of frame type [12]. This is accomplished by dividing the motion vectors by the difference between the corresponding frame number and the reference frame num-

ber (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:

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

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 $c = 2$.

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:

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

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.

4.2 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. Sometimes it is possible to get the motion vectors that belong to different objects in the macroblocks that lie in the object boundary. In this situation the median value corresponds to the object which has majority of motion vectors in that macroblock. Hence the representative motion vectors always indicates the reliable object motion. The representative motion vectors are given as input for the segmentation stage. Since we are not particular about 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

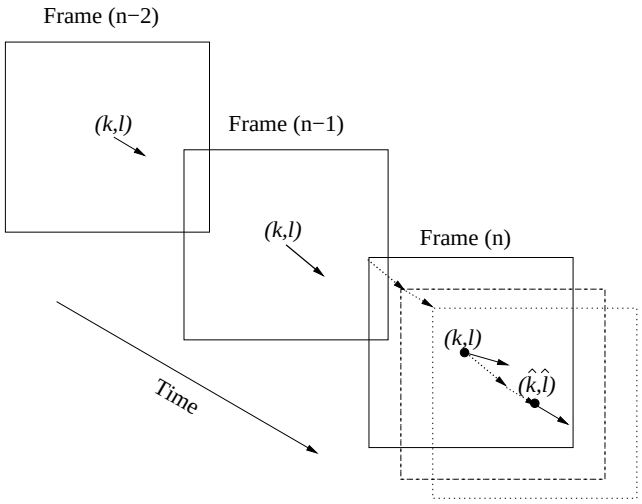


Fig. 2. Relative motion of the macroblock of frame $(n-2)$ with respect to the corresponding macroblock of current frame n. The box drawn with dotted lines is the current position of the macroblock of frame $(n-2)$. Projection of frame $(n-1)$ with respect to current frame is shown as box drawn with dashed lines.

computational burden involved in segmentation process to a greater extent.

5 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.

5.1 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 critically 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 monitored. 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. Fig 3 illustrates the effectiveness of the above procedure for flower garden sequence, indicating the number of classes $N_o = 3$. Here the three distinct motions correspond to the tree, flower-bed and the background.

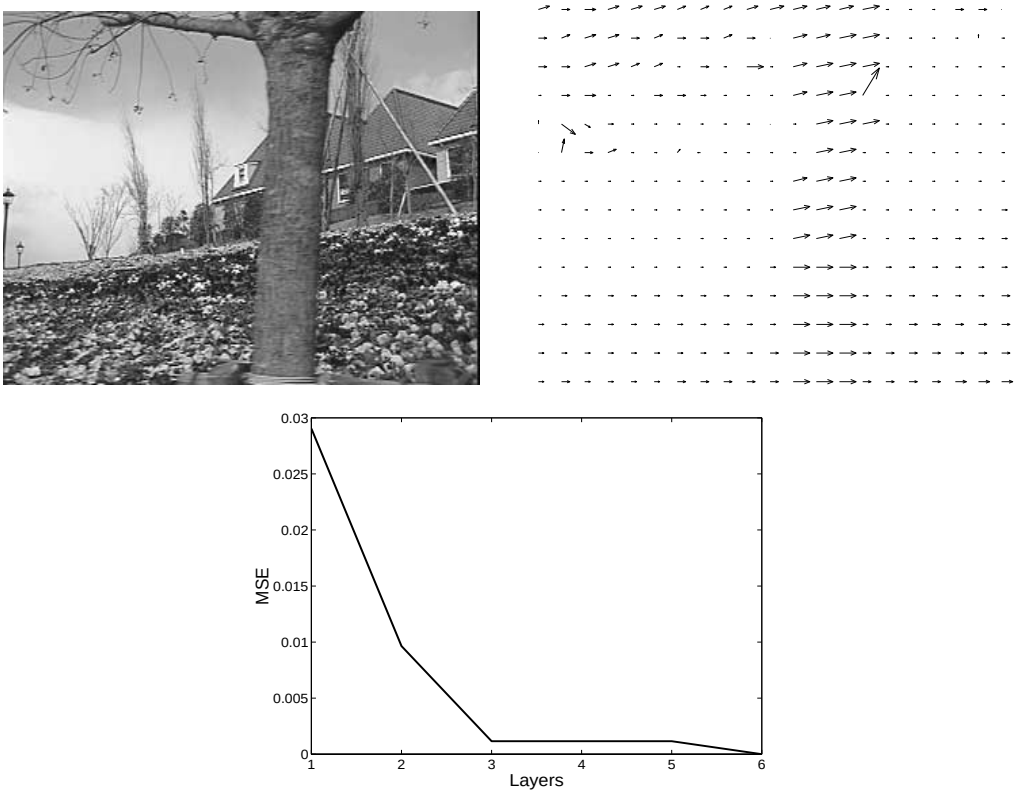


Fig. 3. MSE for various layers of flower garden sequence. The motion vectors corresponding to the fourth frame of flower garden sequence is shown to illustrate the layers of different motion.

5.2 The E Step

The E step computes the probabilities associated with the classification of each representative motion vector as belonging to *j*th class (*i.e.*, having motion parameter $\mathbf{a_j}$). The motion parameter $\mathbf{a_j}$ is initialized with the K-means cluster centers that were obtained in the previous step with number classes equal to N_o.

$$L_j(\mathbf{B}) = Pr(\text{Motion in macroblock } \mathbf{B} \text{ is represented by } \mathbf{a}_j). \tag{3}$$

where, $\mathbf{B} = [k\ l]^T$ 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 *j*th object.

Let

$$R_j^2(\mathbf{B}) = \|\mathbf{a_j} - \mathbf{v}(\mathbf{B})\|^{\mathbf{2}} \tag{4}$$

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

Then the likelihood of $\mathbf{B}$ belonging to class j is given by

$$L_j(\mathbf{B}) = \frac{\mathbf{e}^{-\mathbf{R_j^2(B)/2}\sigma^2}}{\sum_{\mathbf{i=1}}^{\mathbf{N_O}} \mathbf{e}^{-\mathbf{R_1^2(B)/2}\sigma^2}} \tag{5}$$

where, σ^2 controls the fidelity of the motion model fit to the dense motion vectors. Typical value of σ ranges from 0.1-0.3.

5.3 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

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

The estimated motion parameters are given by

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

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:

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

The final object mask for the j th motion model is given by $\mathcal{Z}_j(\mathbf{B})$, $\forall\ \mathbf{B}$. The threshold T_l is fixed in a way so as 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.

6 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 [10] of each frame. More reliable measure of motion activity is obtained by using the representative motion vectors.

(2) **Object Area:** This gives the rough estimate of the area of 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. The jth object area at nth frame is given by

$$A_j(n) = \frac{\#\{B : B \in \mathcal{Z}_j\}}{\bigcup_{k=1}^{No} \#\{B : B \in \mathcal{Z}_k\}} \tag{9}$$

(3) **Velocity of the Object:** The object velocity $\mathbf{a}$ (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 $\{x_i, y_i, n_i\}, i \in 1, \ldots, N$, where x_i and y_i represents the horizontal and vertical positions of the object at frame number n_i. Several normalizations are now used to make the trajectory sensitive only to the shape and direction of the trajectory and independent of spatial and temporal scale. Each shot is divided into blocks containing N number of frames (with an overlap of one frame) and the horizontal and vertical trajectory informations are smoothed by a median filter which removes the noisy motion trail information. Trajectory normalization is done by the following operations.

$$\begin{aligned}
\hat{x}_i &= (x_i - \min\{x_i\}) \,/\, \max\{x_i, y_i\} \\
\hat{y}_i &= (y_i - \min\{y_i\}) \,/\, \max\{x_i, y_i\} \\
\hat{n}_i &= (n_i - \min\{n_i\}) \,/\, \max\{n_i\}
\end{aligned} \tag{10}$$

Now second order polynomials are fitted for both horizontal ($\hat{x}_i$) and vertical ($\hat{y}_i$) positions of the object centers with respect to the normalized temporal index ($\hat{n}_i$) in least square sense, to represent the trajectory. The polynomial coefficients $\mathbf{\Theta}$ are obtained by the following least squares fit

$$\mathbf{\Theta} = \left(\mathbf{T^T T}\right)^{-1} \mathbf{T^T P} \tag{11}$$

where,

$$\mathbf{\Theta}^T = \begin{bmatrix} a_{x0} & a_{x1} & a_{x2} \\ a_{y0} & a_{y1} & a_{y2} \end{bmatrix},$$

$$\mathbf{T} = \begin{pmatrix} 1 & \hat{n}_1 & \hat{n}_1^2 \\ 1 & \hat{n}_2 & \hat{n}_2^2 \\ \vdots & \vdots & \vdots \\ 1 & \hat{n}_N & \hat{n}_N^2 \end{pmatrix} \text{ and } \mathbf{P} = \begin{pmatrix} \hat{x}_1 & \hat{y}_1 \\ \hat{x}_2 & \hat{y}_2 \\ \vdots & \vdots \\ \hat{x}_N & \hat{y}_N \end{pmatrix}$$

Only these polynomial coefficients (Θ) with their starting and ending frame numbers and the video shot identification details are retained for retrieval purpose.

Query Trajectory: The user is asked to draw the query trajectory by marking sequence of points on a 2-D plane. The input points are subjected to the normalization using (10) and the polynomial coefficients for the query trajectories ($\mathbf{\Theta}_q$) are obtained by the least squares fitting as explained in (11).

Trajectory Matching : Let,

$$\begin{aligned} f_x(t; \mathbf{\Theta}_t) &= a_{x0} + a_{x1} t + a_{x2} t^2 \\ f_y(t; \mathbf{\Theta}_t) &= a_{y0} + a_{y1} t + a_{y2} t^2 \end{aligned} \tag{12}$$

be the polynomial representation of the horizontal and vertical target trajectory, and

$$\begin{aligned} f_x(t; \mathbf{\Theta}_q) &= b_{x0} + b_{x1} t + b_{x2} t^2 \\ f_y(t; \mathbf{\Theta}_q) &= b_{y0} + b_{y1} t + b_{y2} t^2 \end{aligned} \tag{13}$$

represents the query trajectory. Now the following metric $D_{Tr}(Q, T)$ is defined to measure the distance between the query and target trajectory.

$$D_{Tr}(Q,T) = \int_0^1 \left(|a_{x0} - b_{x0}| + |a_{y0} - b_{y0}|\right) dt + \int_0^1 \left(|a_{x1} - b_{x1}| + |a_{y1} - b_{y1}|\right) t dt + \int_0^1 \left(|a_{x2} - b_{x2}| + |a_{y2} - b_{y2}|\right) t^2 dt \quad (14)$$

$$= \left(|a_{x0} - b_{x0}| + |a_{y0} - b_{y0}|\right) + \frac{1}{2}\left(|a_{x1} - b_{x1}| + |a_{y1} - b_{y1}|\right) + \frac{1}{3}\left(|a_{x2} - b_{x2}| + |a_{y2} - b_{y2}|\right) \quad (15)$$

(5) **Camera Motion Estimation:** Under the assumption that the camera is undergoing rotation and zoom but no translation, the change of image intensity between frames can be modeled by the following 6-parameter projective transformation [20].

$$x' = \frac{p_1 x + p_2 y + p_3}{p_5 x + p_6 y + 1} \quad (16)$$

$$y' = \frac{-p_2 x + p_1 y + p_4}{p_5 x + p_6 y + 1} \quad (17)$$

here, $p_1, \ldots p_6$ are the camera motion parameters and (x, y) and (x', y') are the image coordinates of the corresponding points in two neighboring frames with respect to the standard orthogonal set of axes with origin $(0, 0)$ at the image center. Suppose the camera undergoes small rotations (α, β, γ) about the X, Y and Z camera axes and the focal length changes from f to sf between two consecutive frames, then the parameters $p_1, \ldots p_6$ satisfy [20].

$$\begin{pmatrix} p_1 & p_1 & p_3 \\ -p_2 & p_1 & p_4 \\ p_5 & p_6 & 1 \end{pmatrix} = \begin{pmatrix} s & s\gamma & -sf\alpha \\ -s\gamma & s & sf\beta \\ \alpha/f & -\beta/f & 1 \end{pmatrix} \quad (18)$$

Here we have examined two methods for estimating the camera parameters.
Method 1:
If we assume that the perspective distortion effects are negligible, then the parameters p_5 and p_6 can be set to zero. Now the rest of the parameters $p_1 \ldots p_4$ are estimated by the following weighted least squares technique.

$$\widehat{\theta} = \mathbf{A}^{-1}\mathbf{B} \quad (19)$$

where,

$$\mathbf{A} = \begin{pmatrix} \sum_i w_i(x_i^2 + y_i^2) & 0 & \sum_i w_i x_i & \sum_i w_i y_i \\ 0 & \sum_i w_i(x_i^2 + y_i^2) & \sum_i w_i y_i & -\sum_i w_i x_i \\ \sum_i w_i x_i & \sum_i w_i y_i & \sum w_i & 0 \\ \sum_i w_i y_i & -\sum_i w_i x_i & 0 & \sum w_i \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} \sum_i w_i(x_i' x_i + y_i' y_i) \\ \sum_i w_i(x_i' y_i - x_i y_i') \\ \sum_i w_i x_i' \\ \sum_i w_i y_i' \end{pmatrix}$$

and $\widehat{\theta}^T = [p_1\ p_2\ p_3\ p_4]$. Here, p_1 indicates the inter-frame zoom factor ($p_1 > 1$, $p_1 = 1$ and $p_1 < 1$ correspond to zoom in, no zoom, zoom out respectively). The change in camera angle about the lens axis is given by the ratio p_2/p_1, camera pan and tilt rates are given by the ratios $-p_3/p_1$ and p_4/p_1 respectively. The weights w_i are determined by the instantaneous estimate of the variance of the ith motion vector.

$$w_i = \hat{w}_i / \sum \hat{w}_i \tag{20}$$

where, $\hat{w}_i = \left[\left(m_x^i - \bar{m}_x\right)^2 + \left(m_y^i - \bar{m}_y\right)^2 \right]^{-1}$

Here, $\bar{m}_x$ and $\bar{m}_y$ are the mean horizontal and vertical components of the background motion vectors.

In most of the cases the camera motion is captured by the background object which usually occupies more area than any other object. Hence, only the motion vectors corresponding only to the biggest object in the frame is considered for camera motion estimation while the motion vectors corresponding to the other objects are ignored.

$$\mathbf{w}_k' = \begin{pmatrix} x_k' \\ y_k' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \left[\begin{pmatrix} i_k \\ j_k \end{pmatrix} - \begin{pmatrix} i_o \\ j_o \end{pmatrix} \right] \tag{21}$$

Here, $\mathbf{w}_k'$ represents the new location of the kth accumulated motion vector (m_x^k, m_y^k) with respect to the image-centered Cartesian axis, whose original location in the current frame is $(i_k,\ j_k)$ and (i_o, j_o) are the coordinates of the center of the image. The corresponding locations of the points (x_k, y_k) in the previous frame for the current frame point (x_k', y_k') is given by

$$\mathbf{w}_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} x_k' \\ y_k' \end{pmatrix} + \begin{pmatrix} m_x^k \\ -m_y^k \end{pmatrix} \tag{22}$$

Method 2:
This method is based on the closed form expressions given in [20] for estimating the camera zoom factor (p_1) and pan $(-p_3/p_1)$ and tilt (p_4/p_1) rates. Here

it is assumed i) the perspective distortion effects are minimal $(p_5 = p_6 = 0)$ and ii) the camera does not rotate about the lens axis Z $(p_2 = 0)$.

$$p_1 = \frac{\sum_{k=1}^{N} \left(\mathbf{w_k'} - \bar{\mathbf{w}}'\right)^T \left(\mathbf{w_k} - \bar{\mathbf{w}}\right)}{\sum_{k=1}^{N} \|\mathbf{w_k} - \bar{\mathbf{w}}\|^2} \tag{23}$$

$$\begin{pmatrix} -p_3/p_1 \\ p_4/p_1 \end{pmatrix} = \frac{\bar{\mathbf{w}}'}{p_1} - \bar{\mathbf{w}} \tag{24}$$

here, $\bar{\mathbf{w}} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{w_k}$; $\bar{\mathbf{w}}' = \frac{1}{N}\sum_{k=1}^{N} \mathbf{w_k}'$.

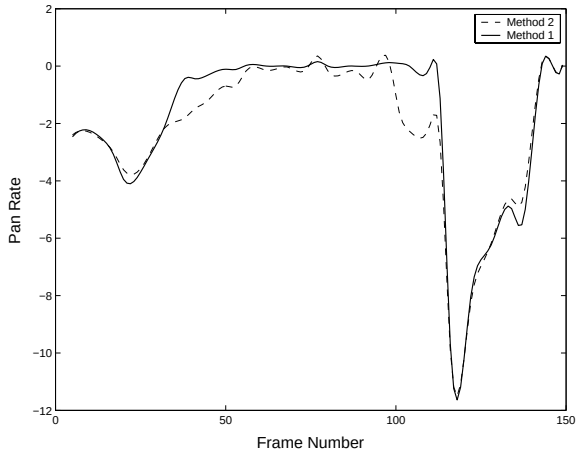
Though the above formulation is computationally efficient, in [20], the motion vectors of all the inter coded macroblocks with non-zero motion vectors are considered for estimating the camera parameters. This will not hold true if there are foreground objects, occupying considerable area of the frame. The above problem is overcome by considering the motion vectors corresponding to the background object alone. The results obtained by both methods for a particular sequence are shown in fig 4. The results of both methods are consistent when the variance of the background motion is very less (*i.e.* when the motion is evenly distributed in the background region). Since the camera motion dynamics does not change rapidly within a small time interval, in both the above mentioned techniques the camera parameters are extracted only for the P frames and interpolated for other frames. Since the first method gives the better estimate of the camera motion, in this paper we used the camera motion parameters estimated by the first method for video retrieval.

Finally the representative features for each block of size N frames are obtained by taking the median value of the features within the corresponding block. Now each block is represented by the triplet $(\mathcal{O}, \mathcal{G}, \mathcal{V})$. where,

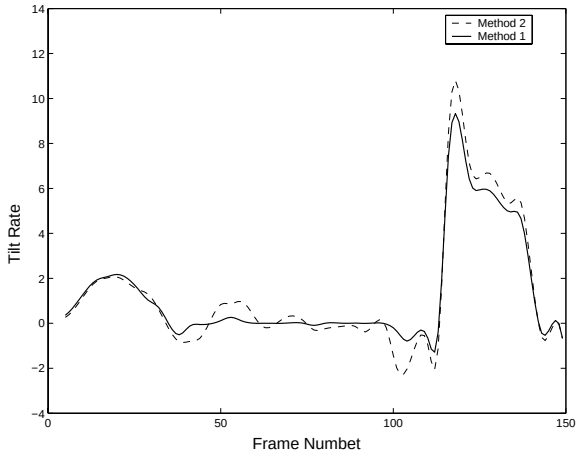
- $\mathcal{O}$: An object representation, consists of objects represented by its extracted features (object area,velocity, trajectory informations).
- $\mathcal{G}$: Global Frame representation, consists of global motion features (zoom, pan, tilt etc.).
- $\mathcal{V}$: Identity of the video subsequence to which the object belongs to. It consists of a unique video sequence identity and starting/ending frame number of the corresponding video block.

7 Results and Discussion

The user is asked to submit a query by giving the values for global features $(\mathcal{G})$ (motion activity and camera motion) and object features $(\mathcal{O})$ (size, speed and trajectory) with their corresponding weights $(\mathbf{w}_o, \mathbf{w}_g)$. In order to input object trajectory information, the user is asked to sketch the trajectory as a sequence of vertices in the



(a)



(b)

Fig. 4. (a)Camera pan rate and (b)tilt rate estimated using both methods for a particular sequence.

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 after the normalization procedure as explained in (10). This normalized trajectory information is independent of the spatio-temporal location of the object and represents the curvature and direction of the object motion.

The following distortion metric is used to compute the distance between the query and target.

$$D(T, Q) = \mathbf{w}_o.\|\mathcal{O}_T - \mathcal{O}_Q\|^2 + \mathbf{w}_g.\|\mathcal{G}_T - \mathcal{G}_Q\|^2 \tag{25}$$

where, $\mathbf{w}_o, \mathbf{w}_g$ represents the user-defined weight vectors for the object features and global features respectively. The distance between the query and target trajectory is computed using the metric given by (14), which is further scaled by the corresponding user weight.

Table 1
Details of queries and corresponding weights for Fig. 5

	MA	Camera Motion			Object	
		ZF	PR	TR	Area (%)	Velocity $[m_x,\ m_y]$
Q1	1.2	1	2.5	0.5	10	[3, -0.5]
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]
W3	0.1	0	0.1	0.1	0.1	[0.2,0.2]
Q4	2	1	2	0.5	10	[-4,0]
W4	0.2	0.1	0.1	0.1	0.1	[0.2,0.2]
Q5	2.5	1	-3	-1	30	[0.5,0]
W5	0.2	0.2	0.1	0.1	0.1	[0.2,0.1]

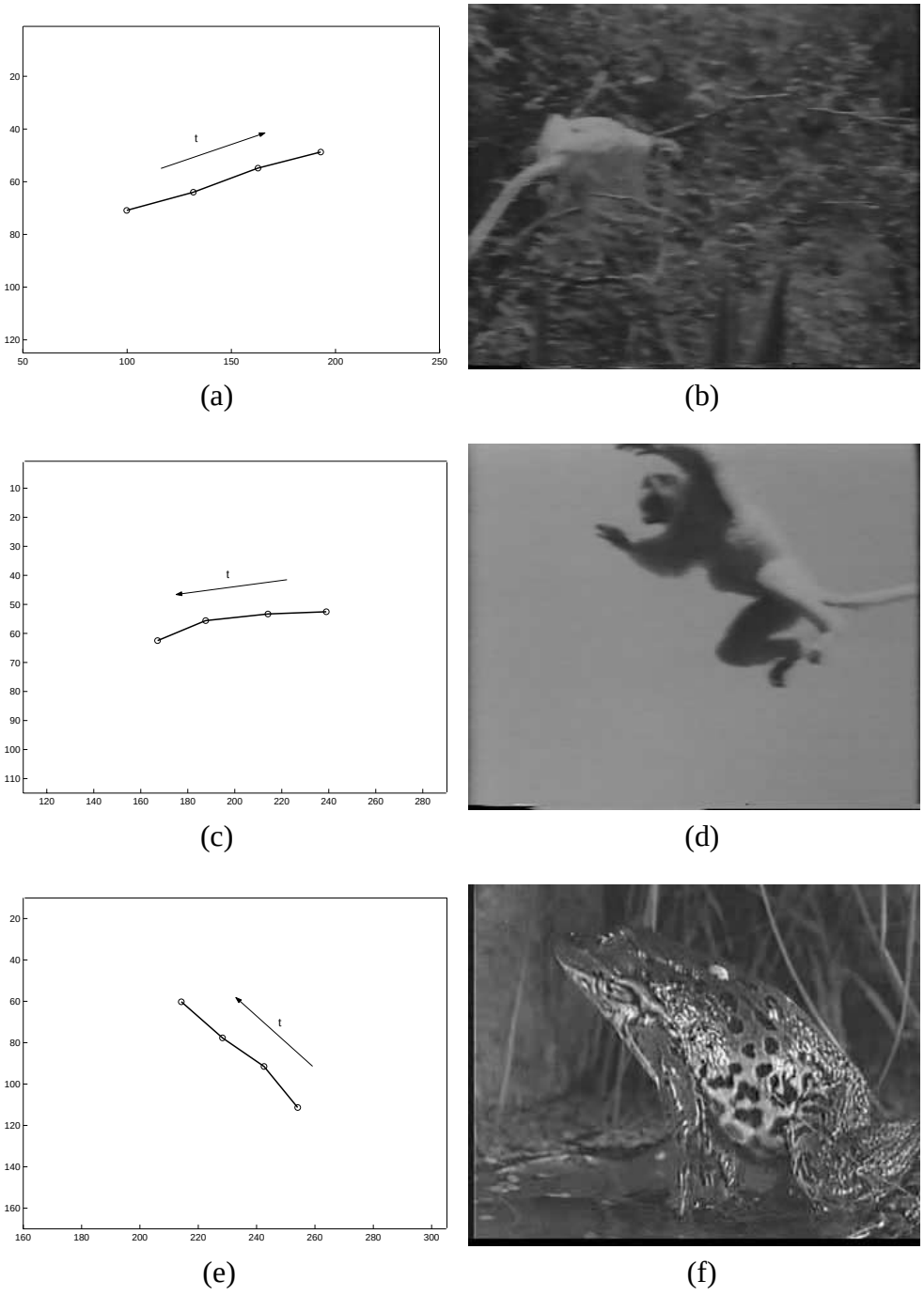
MA: Motion Activity; ZF: Zoom Factor
PR: Pan Rate; TR: Tilt Rate

Table 1 shows the query data [2] for the results shown in Fig. 5. Further results based on object trajectory and camera motion are given in Figs. 6 and 7. In the first query Q1 of table 1, more weight is given to pan rate, velocity and object trajectory. The closest target for Q1 is the video containing an object (monkey) that moves from left to right direction (as indicated by the trajectory) at a rate of approximately 3 pixels/frame (as the query velocity m_x) with respect to global camera pan (here the camera too pans from left to right along the object at approximately 2-3 pixels/frame). For queries Q2 and Q3 more weightage is given to the object velocity than other features and we can see the corresponding retrieved closest videos containing the objects that move at the velocity given by the user (for Q2 the object moves left around 3 pixels/frame; for Q3 the object moves left around 4 pixels/frame and 3 pixels/frame upwards). For queries Q4 and Q5 the weights are distributed to all the features except object trajectory. Since the object velocity is measured as the relative velocity with respect to the global motion, in Q4 though the tree is not moving, it appears to be moving with a velocity of approximately 4 pixels/frame leftwards with respect to the camera pan rate of 2 pixels/frame towards right. The query Q5 expects a video with pan rate of 3 pixels/frame leftwards

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

with approximately no object velocity. The closest retrieved video for Q5 is the bus moving approximately at the camera pan rate of 3 pixels/frame towards left. Fig. 6 shows the first three hits for the query only with global camera motion (pan rate=2, tilt rate=0). Fig. 7 shows the effectiveness of the trajectory-based query.

Unlike pixel domain approaches, it is difficult to segment and track objects of



(a) (b) (c) (d) (e) (f)

Fig. 5. (a), (c) and (e) are query trajectories; (b), (d) and (f) are corresponding frames from retrieved video.

(g) (h)

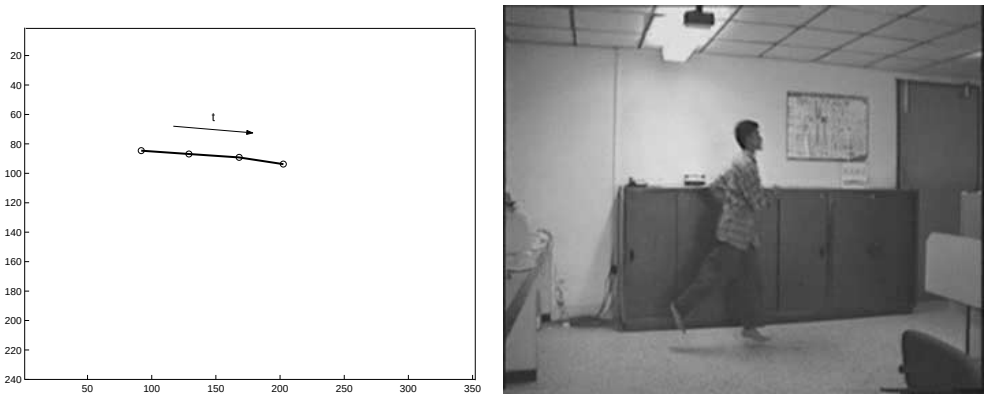
Fig. 5. (contd.) (g), (h) are the retrieved videos corresponding to the query Q4 and Q5 in Table 1 without query trajectory.

(a) (b)

(c)

Fig. 6. Camera motion based query: (a)-(c) are the first three retrieved video shots for the query: pan rate=2, tilt rate=0 and zoom=1 with equal weightage.

smaller size in the compressed domain approach. Since the objects occupying area smaller than a macroblock may not produce a reliable motion vector, this compressed domain technique may fail to track such objects. But often the object of interest in a general video occupies considerable image space and suitable for using the proposed compressed domain indexing scheme. The proposed method may



(a) (b)

(c) (d)

(e) (f)

Fig. 7. Trajectory only query (a) Input trajectory, (b)-(f) are the snap shot of the first five retrieved video shots.

not be suitable for applications like games where smaller objects are to be tracked. In other words, this method is useful for indexing video objects that can produce motion vectors indicating the object motion.

The authors have not come across any work that uses both global and object features extracted from compressed domain for video retrieval purpose. The limitations of

other approaches as compared to our approach, are explicitly stated subsequently. For instance, the video content classification work done by Dimitrova *et al.*, [7] uses only the object trajectory information extracted from motion vectors for retrieval of video. Hence it may not be suitable for application where size, speed of the object and global motion/camera motion are important. Further the work by Sahouria and Zakhor [16] also makes use of the trajectory information with velocity of object for indexing videos of street surveillance system. However, the above compressed domain system is designed for surveillance purpose with a fixed camera angle and is not suitable for general videos. Other compressed domain approaches [1,20,13,19] use the global features such as camera motion for indexing and browsing the video database. Since these methods do not use the object features, they are not suitable for indexing videos with interesting objects. In contrast, our technique overcomes the above limitations by using both object as well as global features for generic video indexing.

8 Conclusions

Video retrieval system is useful for video editing, video archiving,interactive video and multimedia information systems. Since the method of user annotation of video database requires a great amount of human effort, it is essential to have effective systems for automatic video shot indexing. Many video retrieval techniques critically rely on the spatial (still image) information and ignoring the motion features of the objects. The use of motion information in the compressed domain allows for rapid analysis of the content of the video.

In this paper an object-based video indexing and retrieval system using the motion information obtained from the compressed MPEG video has been presented. The main contribution of the proposed system is the utility of the readily available motion information of MPEG video for global and object-based retrieval of generic video. The retrieval results are close to the user expectation. Since the sparse motion vectors are used for extracting features, the computational burden is minimized to a great extent. 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.

References

[1] E. Ardizzone, M. L. Casciai, A. Avanzato, and A. Bruna. Video indexing using MPEG motion compensation vectors. In *IEEE International Conference on Multimedia Computing and Systems*, volume 2, pages 725–729, June 1999.

[2] R. Venkatesh Babu, B. Anantharaman, K. R. Ramakrishnan, and S. H. Srinivasan. Compressed domain action classification using HMM. *Pattern Recognition Letters*, 23(10):1203–1213, August 2002.

[3] S. Chang, W. Chen, H. Horace, H. Sundaram, and D. Zhong. A fully automated content based video search engine supporting spatio-temporal queries. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5):602–615, September 1998.

[4] J.D. Courtney. Automatic video indexing via object motion analysis. *Pattern Recognition*, 30(4):607–625, April 1997.

[5] J. W. Davis. Recognizing movement using motion histograms. Technical Report 487, MIT Media Lab, April 1998.

[6] Yining Deng, Debargha Mukherjee, and B. S. Manjunath. NeTra-V: Toward an Object-Based Video Representation. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 202–215, 1998.

[7] N. Dimitrova and F. Golshani. Motion recovery for video content classification. *ACM Transactions on Information Systems*, 13(4):408–439, October 1995.

[8] Ronan Fablet and Patrick Bouthemy. Statistical motion-based retrieval with partial query. In *Visual Information and Information Systems*, pages 96–107, 2000.

[9] 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*, 28:23–32, September 1995.

[10] Sylvie Jeannin and Ajay Divakaran. MPEG-7 visual motion descriptors. *IEEE Trans. on Circuits and Systems for Video Technology*, 11(6):720–724, June 2001.

[11] Sylvie Jeannin, Radu Jasinschi, Alfred She, T. Naveen, Benoit Mory, and Ali Tabatabai. Motion descriptors for content-based video representation. *Signal Processing: Iimage Communication*, 16:59–85, 2000.

[12] 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*, volume 2916, pages 78–89, 1996.

[13] 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*, volume 3022, pages 200–211, 1997.

[14] M. Mandal, F. Idris, and S. Panchanathan. A critical evaluation of image and video indexing techniques in the compressed domain. *Journal of Image and Vision Computing- special issue on Content-based Image Indexing*, 17(7):513–529, May 1999.

[15] Mohammad Nabil, Anne H. H. Ngu, and John Shepherd. Modeling and retrieval of moving objects. *Multimedia Tools and Applications*, 13(1):35–71, 2001.

[16] Emile Sahouria and Avideh Zakhor. A trajectory based video indexing system for street surveillance. In *IEEE International Conference on Image Processing*, 1999.

[17] Standard MPEG1: ISO/IEC 11172. Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s.

[18] G. Sudhir and J. Lee. Video annotation by motion interpretation using optical flow streams. *Journal of Visual Communication and Image Representation*, 7(4):354–368, December 1996.

[19] C. Taşkiran, Jau-Yuen Chen, Charles A. Bouman, and Edward J. Delp. A compressed video database structured for active browsing and search. In *IEEE International Conference on Image Processing*, volume 3, pages 133–137, October 1998.

[20] 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*, 10(1):133–146, February 2000.

[21] K. Yoon, D. F. DeMenthon, and D. Doermann. Event detection from MPEG video in the compressed domain. In *International Conference on Pattern Recognition*, pages 819–822, Barcelona, Spain, 2000.

[22] D. Zhong and S. F. Chang. An integrated approach for content-based video object segmentation and retrieval. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(8):1259–1268, December 1999.