Available online at www.sciencedirect.com

ScienceDirect

Computer Vision and Image Understanding 110 (2008) 360–377

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Content based video matching using spatiotemporal volumes

Arslan Basharat *, Yun Zhai, Mubarak Shah

School of Electrical Engineering and Computer Science, University of Central Florida, 4000 Central Florida Boulevard, Orlando, FL 32816, USA

Received 2 November 2006; accepted 5 September 2007
Available online 6 February 2008

Abstract

This paper presents a novel framework for matching video sequences using the spatiotemporal segmentation of videos. Instead of using appearance features for region correspondence across frames, we use interest point trajectories to generate video volumes. Point trajectories, which are generated using the SIFT operator, are clustered to form motion segments by analyzing their motion and spatial properties. The temporal correspondence between the estimated motion segments is then established based on most common SIFT correspondences. A two pass correspondence algorithm is used to handle splitting and merging regions. Spatiotemporal volumes are extracted using the consistently tracked motion segments. Next, a set of features including color, texture, motion, and SIFT descriptors are extracted to represent a volume. We employ an Earth Mover's Distance (EMD) based approach for the comparison of volume features. Given two videos, a bipartite graph is constructed by modeling the volumes as vertices and their similarities as edge weights. Maximum matching of this graph produces volume correspondences between the videos, and these volume matching scores are used to compute the final video matching score. Experiments for video retrieval were performed on a variety of videos obtained from different sources including BBC Motion Gallery and promising results were achieved. We present qualitative and quantitative analysis of retrieval along with a comparison with two baseline methods.

Keywords: Video retrieval; Video matching; Spatiotemporal volumes; Motion segmentation

1. Introduction

The amount of digital content generated in the form of video has seen tremendous growth over the last decade. Key elements providing impetus for this growth are: proliferation of inexpensive digital cameras, hand held devices, popularity of web based video streaming, and adoption of digital video by broadcast industry as a part of their distribution services. As a record number of video clips are generated and added into digital libraries every day all over the world, the need for management of this content by means of efficient storage, indexing, and retrieval has never been more pressing than today. Recent major search initiatives in video domain by companies such as Google, Yahoo, MSN etc., show realization on part of the industry for the proper management of this video content. Apparently they want to build upon their experience of text based search to develop video search engines. Pivotal to achieving this goal will be a viable search methodology capable of computing video content similarities.

Content based video matching is considered to be a complex task. One main reason for this is the amount of intra-class variation where the same semantic concept can occur under different illumination, appearance, and scene settings, just to name a few. For example, videos containing a person riding a bicycle can have variations such as different viewpoints, sizes, appearances, bicycle types, and camera motions. Most of the research in the area of content based video matching is therefore aimed at addressing these challenges.

In this paper we present a content based video matching framework that aims to address certain limitations of the existing methods. The crux of the proposed approach is to use features computed from spatiotemporal volumes as

* Corresponding author.
E-mail addresses: arslan@cs.ucf.edu (A. Basharat), yzhai@cs.ucf.edu (Y. Zhai), shah@cs.ucf.edu (M. Shah).

1077-3142/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2007.09.016

the basic building blocks. The intuition behind this representation stems from the observation that there are several factors that should be considered for deciding whether two videos are similar or not. These factors include similarity of the foreground objects, object motion, background appearance, camera motion, etc. The method presented in this paper addresses these issues by detecting important regions in the (foreground and background) scene, extracting features that are less sensitive to the aforementioned variations, and finally employing a volume correspondence technique that handles partial video matches.

1.1. Related work

Image and video retrieval have been an active area of research in the multimedia community and provides the foundation for tasks like video similarity matching. Over the year several methods and systems have been proposed for the content based image retrieval (CBIR). Most of these earlier systems like MIT's Photobook [1], IBM's QBIC [2] etc., were based on global image features. However, in most cases a user of a CBIR system is interested in searching for images of a particular object (e.g. car, boat, airplane etc.,) or a semantic concept which are functions of local image features. Therefore, CBIR systems relying only on global image features are expected to have limited performance in such scenarios. To overcome this problem, researchers proposed region based image features. Such content representation and modelling approach has been used in a variety of ways. See [3–7] for some of the region based image retrieval (RBIR) systems. The RBIR systems have been shown to perform better than the CBIR systems that are based on only global image features.

A comprehensive video matching system should fuse information from all available media types that can be extracted from a video. This can include audio, video, caption, and text transcript. Some of the earlier video retrieval system like [8–10] focused on the integration of these different types of media. An important issue here is to ensure that the content extraction and matching of the any individual medium is accurate and robust. This challenging aspect of Content Based Video Retrieval (CBVR) has been addressed by several researchers [11–16]. Similar to the paradigm of RBIR, many CBVR approaches also rely on region based features. Often these are spatial regions belonging to keyframe the video [16]. However, since video is a spatiotemporal entity, spatial region based approaches can be extended to represent spatiotemporal regions of the video volume. The approach described in this paper belongs to this category of methods which rely on motion based spatiotemporal segmentation.

Region based video retrieval starts by computation of spatial regions for every frame which are then extended to spatiotemporal regions. For instance, methods proposed in [11,12,15,17] compute spatial color segmentation of every frame in the video which is followed by the temporal correspondence of these regions. However, in highly textured scenes these approaches are not able to perform adequately due to over-segmentation which leads to incorrect region matches. In addition, a complex video can have significant variations in the appearance of the same object throughout the video. Therefore, a simple color segmentation, which is known to give inconsistent results under varying noise and illumination conditions, will not be a viable option. This in turn, limits the effectiveness of several CBVR methods that rely on color based spatial segmentation.

In this line of research, few approaches also used global and local motion information to recover coherent image regions. For example, the motion segmentation and object tracking method presented in [11] relies on color segmentation and optical flow computation. The accuracy and reliability of optical flow is known to be limited in case of large motion or textureless regions (aperture problem). Region tracking in [17] also relies on appearance features computed from regions. Again the performance of these approaches is also limited due to the adverse quality of color segmentation.

Recently, vocabulary based text retrieval techniques have been applied in [18] for object matching in videos. However, their method did not perform explicit object extraction before the matching step. In [19], spatiotemporal volumes were extracted which were specific to faces in the video sequence. This approach relies on the facial structure and the appearance features related to it. In contrast, the framework presented in this paper is more general and applicable to a wide variety of objects and scenarios. Furthermore, [20] presented a framework where specific objects were recognized using the tracked salient regions. However, they require to manually select the particular object that is to be searched in the query video. The main difference of the proposed approach from their technique is that they focus on specific object recognition, whereas our emphasis is more on object/scene category matching. Moreover, in our method, we consider the entire content of the query video and automatically compute the matching between different foreground and the background volumes. In short, we propose a more general framework that can be used to match video shots with similar kinds of objects and scenes. In another recent work, [21] addresses the matching of similar shots and presents a solution based on three-dimensional models of scene content, which are built using affine covariant patches. Another interesting work for matching background scenes in movie shots was presented in [22]. Their matching technique relies on the local similarity of features, an epipolar constraint, and a temporal constraint. Unlike their approach, we consider static as well as moving objects in the foreground to match the video shots.

We feel that there is a need of a better content based video matching approach that could handle partial matches based on similar types of the foreground objects, and the background scene. We consider motion information as a strong que in a video and feel that it should be utilized to extract more reliable video contents. For the video retrieval task, it is desirable to build a system that does not require extensive training for each semantic concept. The following section presents the proposed approach that addresses these issues.

1.2. Proposed framework

The proposed framework comprises of two major components: video volume extraction and video matching using volume features. Unlike conventional approaches, we utilize the interest point trajectories in the video sequence to extract spatiotemporal video volumes. Interest points and their correspondences are established using the Scale Invariant Feature Transform (SIFT [23]) operator. The point correspondences are used to generate trajectories, which are further refined by performing velocity prediction to merge the broken trajectories. These trajectories are then grouped into clusters based on their motion similarity and the spatial proximity. The temporal correspondence between the estimated motion segments is then established based on the highest number of SIFT correspondences. A two pass algorithm is used to handle region noise, splitting, and merging. The tracked regions are then stacked together to produce spatiotemporal volumes. Each volume encompasses independently moving region, which could either belong to the scene background or the foreground object. This provides a more structured information about the scene for the task of video matching. A set of features including color, texture, motion, and SIFT descriptors are extracted from each volume. The weighted combination of feature similarities between two volumes provides a measure of their similarity. The degree of similarity between the features is computed through Earth Mover's Distance. Two videos to be matched are modeled as a bipartite graph, where volumes are represented by vertices and similarities between them as edge weights. The maximum matching of this graph is then used to establish the correspondences between the volumes. The score between each pair of matched volumes is then combined towards the final video matching score. The proposed video matching framework is tested on several videos for the task of content based video retrieval.

It should be noted that our framework is not designed to search for exact matches of an object observed in a video shot as suggested by [20]. On the other hand, our approach is more suitable for establishing similarity among videos based on similar types of the foreground objects and the background scene. The novelty of our approach for video matching lies in (a) the extraction of spatiotemporal volumes that correspond to meaningful foreground and background objects (b) a partial video matching framework based on several strong features from the volumes.

The details of the proposed framework are discussed in the following sections. Steps involved in the extraction of volumes are described in Section 2. Section 3 discusses the volume features used and their role in the matching task. The graph based video matching technique is described in Section 4. The experimental results and performance analysis are presented in Section 5. Finally, the conclusions and future directions are discussed in Section 6.

2. Spatiotemporal volume extraction

In this paper we propose a framework that relies on spatiotemporal regions (volumes) for solving the video matching problem. For a given video, we first extract interest point trajectories using SIFT correspondences (see Section 2.1). These trajectories are then used to recover different motion segments in each frame (see Section 2.2). The correspondence between the motion segments is then resolved using a two pass algorithm (see Section 2.3). In this paper, the term *foreground* refers to the moving objects in the video and the term *background* refers to the physical environment where the objects reside. They are used to assist the reader of this paper to better understand the problem that we address. Our framework does not distinguish between the foreground and background volumes, and are used in the same way for matching.

2.1. Trajectory generation

For a given video shot we generate a set of motion trajectories based on SIFT interest points. Main steps involved in this task are shown in Fig. 1 along with the intermediate results after each step. The first step is to detect SIFT interest points in every frame of the given video shot. This produces a 128 dimensional feature vector as a descriptor of each interest point. The second step is to recover interest point correspondences in every pair of neighboring frames. [23] presents a descriptor matching method that produces robust interest point correspondence. Output of this step produces the connection between the corresponding interest points in any two neighboring frames as shown by link (a) in Fig. 1. The third step is to connect these corresponding links in the neighboring frames and construct longer trajectories. Link (b) shows the effect of trajectory generation after correspondence merging. There can be some erroneous correspondences with abnormally large displacement in image coordinates. This is handled by the fourth step where these large displacements are pruned out assuming locally uniform velocity. Link (c) represents the step where trajectory segments with abnormally large acceleration have been removed. At this stage there might be many broken trajectories that can be merged across time to generate longer and more meaningful trajectories. For this, a uniform local velocity model is assumed. Each trajectory is projected forward in time and a similar SIFT descriptor is searched within a small spatiotemporal window. If a match is found and if it is the start of another trajectory, then both of the trajectories are merged. Link (d) shows that the length of the trajectories can be increased after merging. Note that for illustrative purposes we are only showing a subset of trajec-

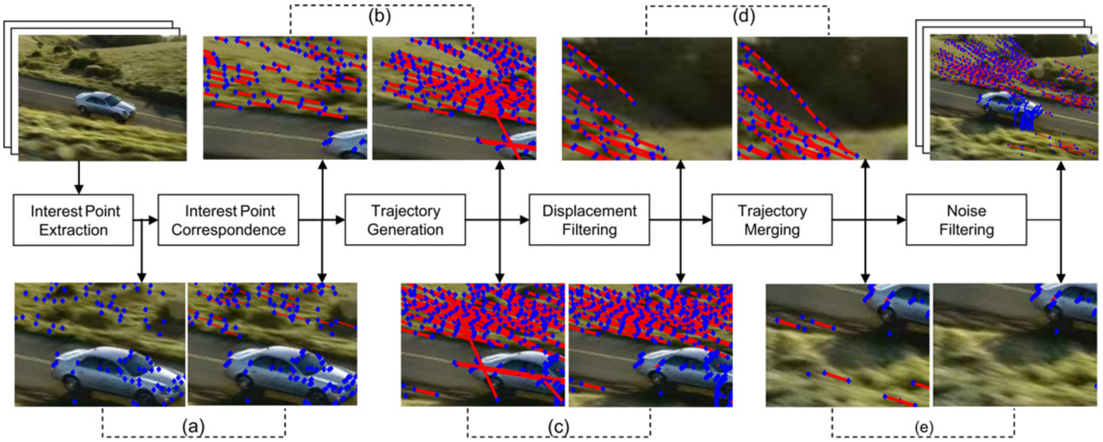


Fig. 1. The steps for generating interest point trajectories are shown here. Dotted links in the figure connect the same image areas before and after a particular step. The images contain blue diamonds representing SIFT interest points and the red lines showing the trajectory connection between them. Following steps are highlighted by the mentioned links: (a) interest point correspondence, (b) initial trajectory generation by merging point correspondences, (c) removal of irregular trajectory segments, (d) merging broken trajectories, (e) removal of small trajectories. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)

tories that have points in the current frame. The last step is to remove any trajectories that are not meaningful and span less than three frames. Link (e) shows the removal of these noisy trajectories. This generates a set of trajectories for the complete video and it is used to determine different motion segments in the video shot.

2.2. Region detection

After generating a set of interest point trajectories for the complete video, the next step is to perform initial motion segmentation. Note that depending on the texture on the foreground and background regions in a given video, we will get the trajectories from both parts. Our assumption is that a region's motion can be approximated by the motion of a plane. Hence, we can use a homography to capture an object's motion from one frame to the next. We use the RANSAC based homography computation to recover significant motion segments. Four interest point correspondences are randomly chosen to generate a homography transformation. The most dominant motion segment is first chosen by selecting the homography with the largest number of inliers. The rest of the correspondences go through the same process to recover the next most dominant motion segment and so on. This produces a set of motion segments in one frame with respect to the other. Note that our approach can only handle rigid body motion. In case of highly deformable objects, we will obtain separate homographies corresponding to consistent local motions of the object parts.

Fig. 2c shows the output of motion segmentation, where trajectories belonging to the foreground and the background are clustered into two separate segments. Note that some small trajectories from Fig. 2b have been removed in Fig. 2c and d. Since very small trajectories do not provide much reliable information as compared to the longer ones for volume extraction, they are hence removed during the region detection stage. In case of a more complicated scene containing multiple objects with the same motion as shown in Fig. 3, four different motion segments are detected as shown in Fig. 3a. Red and green colored trajectories represent two most dominant segments including five cars. In frame 10 (first column of Fig. 3a) two cars (green segment) are moving together in one direction while the other three cars (red segment) are moving together in a different direction. This process is applied to all the frames and motion segments are recovered in every frame.

Next, we use the initial motion segments to detect separate objects moving similarly. These objects are separated into detected regions by using the spatial proximity of the interest points in each frame. Similar to the connected components algorithm, we isolate different groups of neighboring interest points using a threshold on the distance from the nearest neighbor. Fig. 3a shows the initial motion segments which are further refined to extract detected regions as shown in Fig. 3b. These regions are shown by different bounding boxes. Note that there are two regions detected from the green initial motion segment and three regions detected in the red segment shown in Fig. 3a. We apply this process to all of the frames in the video and a set of motion regions is generated for every frame in the video. These detected regions are then tracked through the method explained in the next section.

One possible drawback of using the spatial proximity constraint, for region detection, could be over segmentation. However, we have observed that during tracking these over segmented regions are assigned the same label because the proposed algorithm addresses *splitting-and-merging*. This will be further explained in the following section. The detected regions are represented by the bounding boxes that usually encompass main parts of the objects. Although this representation does not provide a tight boundary capturing the object shape, but we found during experiments that this does not hurt the performance signif-

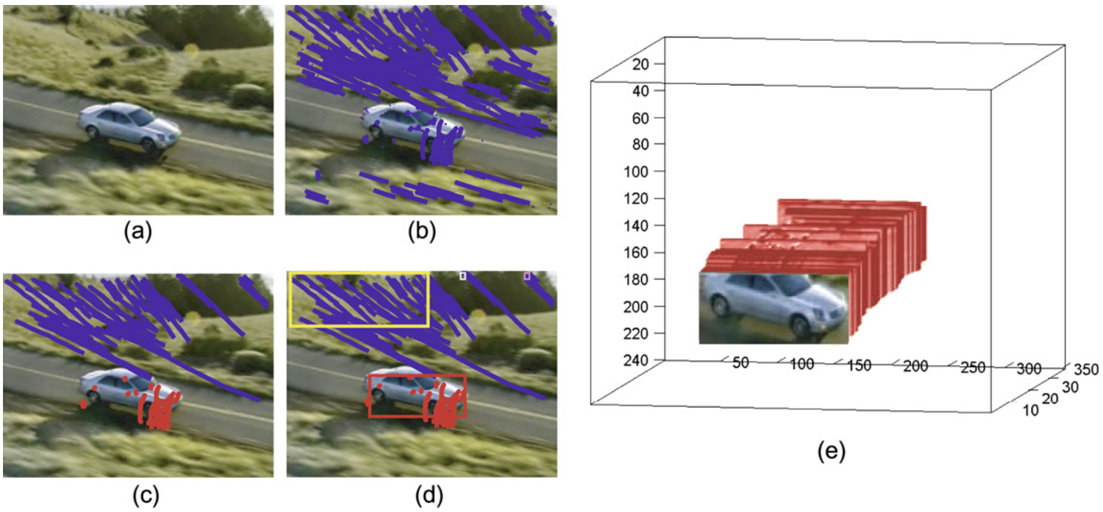
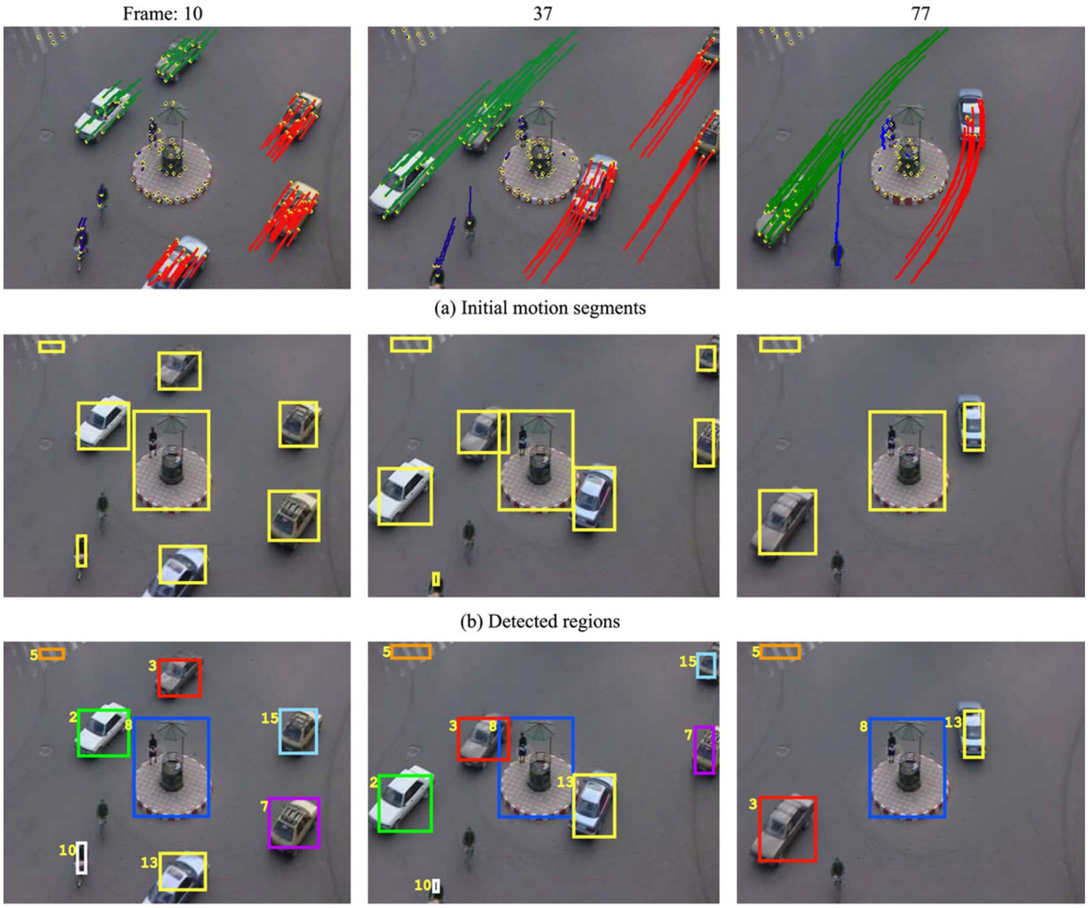


Fig. 2. Stages from trajectory generation to volume extraction. (a) Input video, (b) trajectories generated using method explained in Section 2.1, (c) two initial motion segments in blue and red colors (see Section 2.2), and (d) red trajectory cluster produces a single region shown by the red bounding box. Blue trajectory cluster is divided into one large (yellow), and two very small (white and pink) spatially coherent regions, (e) 3D volume for the region in red bounding box. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)



(a) Initial motion segments

(b) Detected regions

(c) Tracked regions

Fig. 3. Each column here presents the intermediate results for a specific frame. (a) Different motion segments (color coded) are determined using homography based motion segmentation (see Section 2.2). (b) Within each of these segments, different objects are detected using a spatial proximity constraint (see Section 2.2). (c) Common trajectory membership is used to solve the region correspondence that generates consistent labels. (see Section 2.3). (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)

icantly. This is possible due to the type of the volume features used. We compute features using both sparse (SIFT descriptors and motion) and dense (color and texture) video information. The former type does not depend on the kind of the object boundary. On the other hand, color and texture features are more sensitive to the type of boundary. However, it has been observed that the actual object is dominant in the region, so features extracted do not change substantially. If extraction of tighter boundary is desirable, then techniques like snakes [24] and level sets can be used.

2.3. Region tracking

At this stage every frame has a set of detected regions with local labels. We solve region correspondence across frames by using maximum common trajectories between the regions. The motivation for our approach is based on the fact that multiple trajectories from a given region provide several constraints for tracking this region in the following frames.

We start with an initial set of region labels from the first frame. These labels are propagated through the following frames by using a trailing temporal window. For a particular region in the current frame, the member trajectories vote for the labels in the previous frames. Each trajectory votes for the most common label it belongs to. The label with the maximum votes is chosen as the region label in the current frame. Intuitively, one region is matched to another based on the highest frequency of common trajectories. A new label is generated if the maximum vote is by a set of new trajectories starting from the current frame. This is only the forward labeling pass of the correspondence algorithm. A second pass of backward labeling is proposed to handle split and merge scenarios explained in the following. During the second pass, labels from the last frame are propagates to the first frame similar to the forward labeling.

This two pass algorithm is able to assign the same label to both boxes as they split from one region and later merge into a common region. This kind of region *split-and-merge* typically occurs for only a short number of frames at a time. *Split only* and *merge only* are other two possible scenarios, which occur for instance when two objects moving together move away from each other or two objects moving separately come together. Fig. 4 shows an illustration of these three types of scenarios. Recall that we used the spatial proximity constraint to detect regions and that can cause oversegmentation (see Section 2.2). An example of this can be observed with multiple regions in the second frame of the airplane video in Fig. 8. This can happen in cases of objects with large size and sparse feature points. The backward labeling is useful in handling these cases. Labels from both labeling directions are merged into one final set of labels. This is done at every frame by considering a pair of regions at a time. The rectangles are assigned the same final label only if they have common local labels in both the sets. This particular case signifies the *split-and-merge* scenario. On the other hand, in case of *split only* the two regions will not get the same final label because they get a common label only in forward labeling. Similarly, for *split only* the two regions will not get the same final label because they get a common label only in backward labeling. These scenarios, and the final labeling is shown in Fig. 4. Note in this figure there is a *merge only* event right before the *split-and-merge* event which is correctly detected by the proposed two pass algorithm.

Fig. 3c presents consistent region correspondence across frames. Fig. 2e shows a volume extracted using the tracked region. Fig. 8 also shows the result of consistent object tracking through various stages of the video sequence and finally the spatiotemporal volume is extracted from every video sequence.

3. Volume features extraction

Once the volumes are available for a complete video shot, we extract features that are used for the video matching step. We use features that capture interest point descriptors, color, texture, and motion of video volumes. The features are local for the video volume as opposed to using the global video features. A common representation of these features has been used in form of a group set of clusters in the corresponding feature space. Each cluster

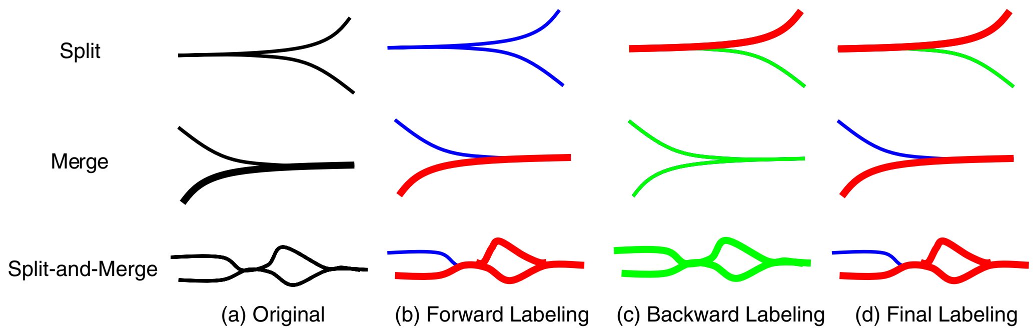


Fig. 4. Effect of the two pass tracking algorithm on three scenarios of objects splitting and merging. (a) Types of relative motion of two segments in a video. (b) Forward labeling generates first set of labels by progressing the labels from the first to the last frame of the video. (c) Backward labeling is applied in the reverse direction to produce another set of labels. (d) Two sets of labels are merged to produce the final labels. (see Section 2.3 for details).

in this set is represented by the mean feature vector and the number of feature points in the cluster. The latter value is normalized to make these features invariant to volume size. This form of feature representation has been found to be more discriminative than the conventional forms like histogram [25]. Details about the features and the method used to compute similarities between them are presented in the following.

3.1. Interest point descriptors

Motivated by the robustness of the interest point descriptors we use SIFT descriptors (128 dimensions [23]) to generate a representation for this feature. Every volume contains a set of interest point trajectories. The SIFT descriptors from all the trajectories are represented as a set of clusters in the 128-dimensional feature space. Clustering in this higher dimensional space is performed using isodata clustering [26]. Unlike K-nearest neighbors, the isodata clustering does not require the number of clusters to be specified. However, the bandwidth of the clusters has to be specified in form of the distance from the cluster center. If the distance to the nearest cluster center is larger than a threshold, then a new cluster is formed. The number of clusters depends on the type and the amount of trajectories. This feature is used to capture a set of prominent interest points observed within a volume.

3.2. Color

We use 3D HSV color values to compute the color feature for the volume. Every pixel belonging to the volume contributes to this feature. The 3D color-space is then clustered using isodata clustering [26] which generates a set of clusters. Typically, representation like color histogram is used along with histogram intersection for feature matching. However, the color representation used here has proven to perform better in case of image retrieval application [25]. This feature is also computed for the residual background region of the volume that does not belong to any volume.

3.3. Texture

To capture the texture inside the volume we employ edge orientation information. Canny edge detector is applied to recover an edge map. The gradient directions are computed for every pixel along these edges. This process is repeated for every image region inside the volume and individual orientations are accumulated. The edge orientations are quantized into eight directions in the $[0, 2\pi]$ range. The final representation of this feature is a set of eight clusters formed using K-Means clustering. Fig. 5 presents an illustration of this feature in case of a volume corresponding to a car. Similar to a histogram, this illustration presents normalized number of samples within each one of the eight clusters. The approach for matching of these feature representations is presented later.

3.4. Motion

Along with the appearance based features we also use the object motion features. Interest point trajectories encapsulated within the volume are used for computing this feature. Different trajectories belonging to a volume start and end independently, but depict the characteristic volume motion. The motion feature of a volume is captured by the direction of motion which is quantized into eight directions in the $[0, 2\pi]$ range. Each point along a trajectory is treated as a separate feature point when the quantization is performed by K-Means clustering into eight motion directions. Unlike a single representative velocity for the complete volume, this feature is more robust to noise in the velocity values and captures the dominant direction of object motion. These directions are influenced by both camera and object motion, hence, this feature alone is not sufficient for volume matching. Fig. 6 shows an illustration of this feature computed for a boat video.

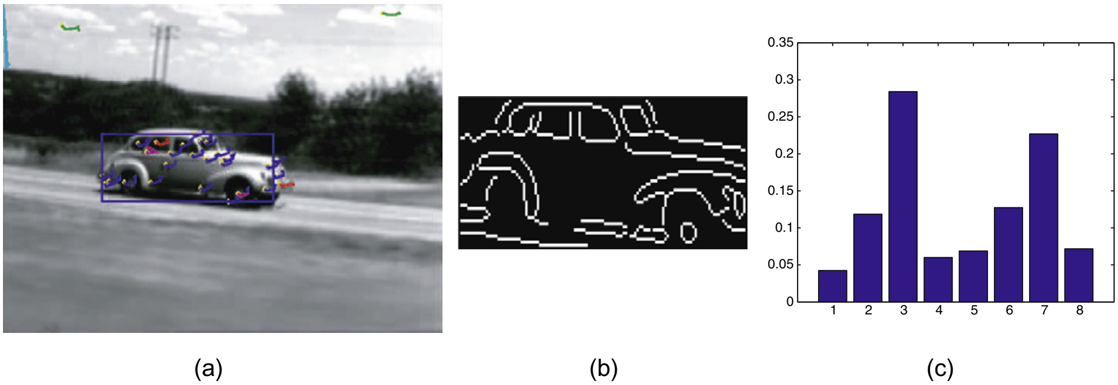


Fig. 5. For each volume we use gradient orientations from the Canny edges to captures texture information. (a) Sample frame with car's segment, (b) Canny edges on the current frame from the car volume, and (c) representation of the eight quantized gradient orientations of complete volume. The two peaks correspond to the mostly horizontal edges in the car volume.

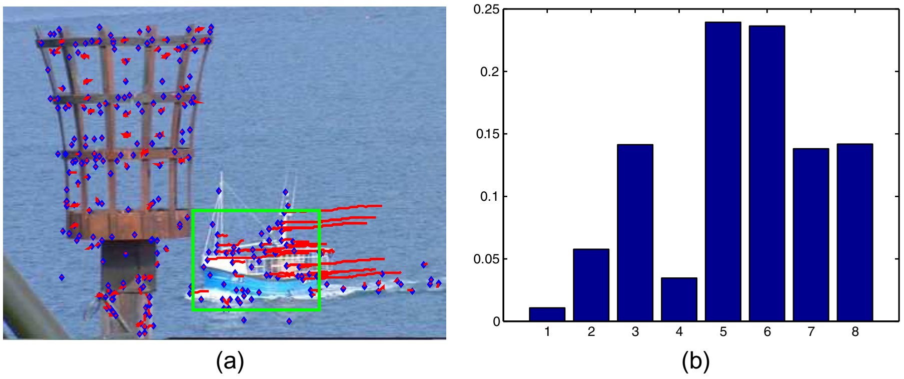


Fig. 6. Motion of a volume is captured by eight quantized directions of motion of the member trajectories. (a) Rectangle captures a boat moving towards bottom left in the video. (b) The peaks correspond to left and bottom directions (π to $\frac{3}{2}\pi$ out of $[0, 2\pi]$ range). Interest point trajectories are shown in red with blue end points. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)

The illustration for this feature is showing the population in the eight clusters, similar to the texture feature.

3.5. Feature matching

To compute feature similarity, we propose a common approach for all the above-mentioned features. The feature representation of set of clusters used here is similar to the *signature* representation presented in [25]. A *signature* S is defined as a set of C clusters $\{(\mathbf{m_i}, w_i) | 1 \leqslant i \leqslant C\}$, where each cluster is represented by the mean feature vector $\mathbf{m_i}$, and the population of feature points w_i. To determine the similarity between these *signatures*, Earth Mover's Distance (EMD) [25] has proven to be quite useful in finding the dissimilarity between *signatures*. Previously, EMD has been successfully used for region based image retrieval [25,5]. More recently, EMD has been used for matching the texture patterns [27] and for classifying texture and object categories [28]. Optimal EMD is computed based on a solution to the transportation problem [29] and provides the cost required to map one *signature* to the other. For two *signatures* P and Q, this distance is given by,

$$\mathrm{EMD}(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}, \tag{1}$$

where, d_{ij} is the distance between two cluster representatives $\mathbf{m_i}$ and $\mathbf{m_j}$. This distance is computed as an L^2-norm of the difference between mean feature vectors. f_{ij} is the flow which depends on the population w_i and w_j of the clusters. These terms are governed by following constraints:

$$f_{ij} \geqslant 0, \quad 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n,$$

$$\sum_{j=1}^{n} f_{ij} \leqslant w_{p_i}, \quad 1 \leqslant i \leqslant m,$$

$$\sum_{i=1}^{m} f_{ij} \leqslant w_{q_j}, \quad 1 \leqslant j \leqslant n,$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j}\right).$$

The cost computed by EMD is transformed to the similarity by,

$$\mathrm{sim}(P, Q) = \exp\left(-\frac{EMD(P, Q)^2}{2\sigma^2}\right). \tag{2}$$

This feature matching approach provides the basis for determining the degree of similarity between two volumes from two different videos. The proposed feature matching and the following video matching modules are not dependent on the specific features we have used. Other types of features like color correlograms, wavelet responses etc., can also be incorporated easily.

4. Volume based video matching

This section explains the method used to determine similarity between two given videos. In this framework, it is desirable that the matching technique should be able to handle partial matches between videos. For instance, in case of two very similar foreground objects observed in two dissimilar backgrounds, the system should be able to generate a high similarity score. Different parts of the scene are captured by volumes and corresponding set of features computed from each of these volumes represent its contents. To generate the partial match between videos, we can use the features similarities for volume correspondence. These individual volume correspondences can then be combined to recover the final video matching score. A model based on maximum matching in bipartite graph is considered suitable for this problem because it directly maps to the above-mentioned scenario of partial matching.

A video can be presented as a set of volumes given by,

$$\{v_{11}, v_{12}, \ldots, v_{1m}\} \subseteq V_1,$$
$$\{v_{21}, v_{22}, \ldots, v_{2n}\} \subseteq V_2,$$

where, v_{1i} represents volume i in video V_1, and m & n represent the number of volumes in video V_1 and V_2, respectively. Note that the set of volumes can be a proper subset of the complete video shot. A part of the video can be excluded from this set when there is very low texture in that section. We incorporate only the color feature of that region. We have observed that this does not affect the results severely in most cases. For every volume in a video we compute a set of features as explained in Section 3. Let there be K different types of features computed for each volume. Then for any feature k $(1 \leqslant k \leqslant K)$, we have the following two sets of descriptors, one for each video,

$$F_1^k = \{f_{11}^k, f_{12}^k, \ldots, f_{1m}^k\},$$
$$F_2^k = \{f_{21}^k, f_{22}^k, \ldots, f_{2n}^k\}.$$

The size and dimensionality of each feature descriptor f_{ij}^k depends on the type of feature k. For every feature k, we also have a function that provides a degree of feature similarity

$$0 \leqslant \text{sim}_k(f_{1i}^k, f_{2j}^k) \leqslant 1. \tag{3}$$

In case of EMD, Eq. (2) provides this similarity function. For the current feature k and videos V_1 and V_2, this metric is used to recover a complete similarity matrix S_{12}^k of size m by n. Note that every element of this matrix represents the similarity between features of two particular volumes, one from each video, such that

$$S_{12}^k = \text{sim}_k(f_{1i}^k, f_{2j}^k), \qquad i\epsilon\{1, \ldots, m\}, \quad j\epsilon\{1, \ldots, n\}. \tag{4}$$

For K different features we have K corresponding similarity matrices. We also improve the credibility of each entry in the matrix S_{12}^k by suppressing noisy volume matches. If the two volumes have significantly different temporal length, then we mark that entry as dissimilar. For volume i in video V_1 and volume j in V_2, the condition is checked

$$\frac{|L(v_{1i}) - L(v_{2j})|}{\max(L(v_{1i}), L(v_{2j}))} \leqslant \varepsilon, \tag{5}$$

where, $L(v_{1i})$ provides the temporal length of volume i in video V_1 and ε is the maximum percentage of size difference allowed.

The similarities from multiple features are combined into a single similarity matrix S_{12} which captures the complete similarity. This is done by computing a linear combination of of K different similarity matrices.

$$S_{12} = \sum_{k=1}^{K} S_{12}^k w_k, \tag{6}$$

where, S_{12} is the final volume similarity matrix between video V_1 and V_2, and w_k represents the normalized weight assigned to feature k, such that $\sum_{k=1}^{K} w_k = 1$. These weights are computed through empirical evaluation and represent the confidence in each feature. In our current experiments, we manually assigned the weights as 0.3 for color, 0.25 for texture, 0.3 for interest point descriptor, and 0.15 for motion feature. For a more extensive set of features along with an annotated dataset one could use boosting techniques [30] to learn the feature weights automatically.

The last step is to use the similarity matrix S_{12} to compute the volumes correspondence between the two videos. This is done by employing a graph theoretic solution where we construct a weighted bi-partite graph as shown in Fig. 7 to model the two videos. The volumes form the vertices and the feature similarities between them are used as edge weights. These edge weights are obtained from the corresponding entries of the similarity matrix S_{12}. The volume correspondence is obtained from the maximum matching in this bi-partite graph. This is achieved by using the Kuhn Munkres [31] algorithm. The mean of edge weights between the corresponding volumes is used to compute the final video matching score between videos V_1 and V_2.

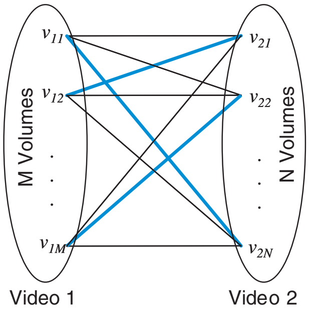


Fig. 7. A bipartite graph is constructed with volumes as the vertices and their feature similarities as edge weights. The maximum matches in this bipartite graph provide the volume correspondence between two videos. The blue edges represent possible correspondence.

5. Experimental results

Several experiments were performed to verify the effectiveness of the proposed framework. Section 5.1 presents some implementation details along with the results of volume extraction. An application of the proposed framework for the task of content based video retrieval is presented in Section 5.2. We also compare our approach with two baseline methods, and present qualitative and quantitative analysis of the retrieval.

We have performed experiments on a dataset of 337 videos obtained from TRECVID 2005 Explore BBC Rushes [32] and online video archives including Google Video [33] and BBC Motion Gallery [34]. There are four main categories of objects present in these videos including boats, cars, airplanes, and tanks. There are 74 boat, 80 car, 148 airplane, and 35 tank videos in the dataset. Keyframes from these videos are presented in Fig. 19. There is a significant amount of variation in viewpoint, motion, size, and appearance of objects in these videos. In addition, there is no restriction on type of camera motion, therefore, stationary, moving, and zooming videos are used. All these variations in the dataset makes it very challenging for the task of content based video matching. Many of the original videos contained multiple shots and there was no shot boundary information available. As the proposed framework is applicable to the individual shots only, therefore, the shot boundary detection was performed manually for the experiments. Automatic shot boundary detection is outside the scope of this work and available specialized techniques [35,36] can be used for this purpose. Typically, the length of video shots used in our experiments lies between 150 and 250 frames. Also note that since our method relies on motion segmentation, the contents of the video should depict significant motion; otherwise, the matching of videos reduces to matching of keyframes.

5.1. Volume extraction

The accuracy and reliability of video matching depends on the quality of the volumes extracted by the method explained in Section 2. The quality of a volume can be gauged by the tracking accuracy of spatial regions detected in each frame. Since the regions are tracked using reliable interest point correspondences, the quality of region tracking was noticed to be good. Fig. 8 presents the tracking results of the foreground object from three different videos along with the corresponding volumes. The regions shown in this figure are correctly tracked and they retain the same label (color) until the last frame. Similar observation was made for many other videos in our dataset. Fig. 3c shows tracking of multiple regions in another video. The proposed approach also handles the cases of region splitting and merging. One such example is shown in case of airplane video in Fig. 8. In the second frame, two different bounding boxes are shown on the airplane. This is because two spatially isolated regions were detected. We are able to assign the same (yellow) label to both regions because they split from one region in the earlier frames and then merge again into a common region in the following frames. The two pass correspondence algorithm presented in Section 2.3 discusses the details.

5.2. Video matching for retrieval

We have applied the video matching framework for content based video retrieval. The matching score is used to rank the videos retrieved from the dataset. The method presented in this paper is fully automatic and does not require any training. This proves to be a great advantage towards automatic structuring in a large database of unorganized videos. These databases with no content annotation or labeling can benefit from such kind of content based matching for better organization and indexing.

In our experiments we initiate the retrieval using a query by example. Other approaches using query by name, query by expansion, etc. are out of the scope of this work. Given a query video, volumes and corresponding features are extracted using the proposed approach. In addition to the features from the volumes a color feature from the residual background is also used. It covers the low textured background region of the video that does not belong to any volume. The color feature from this region is handled like that of any other volume during the matching stage. These

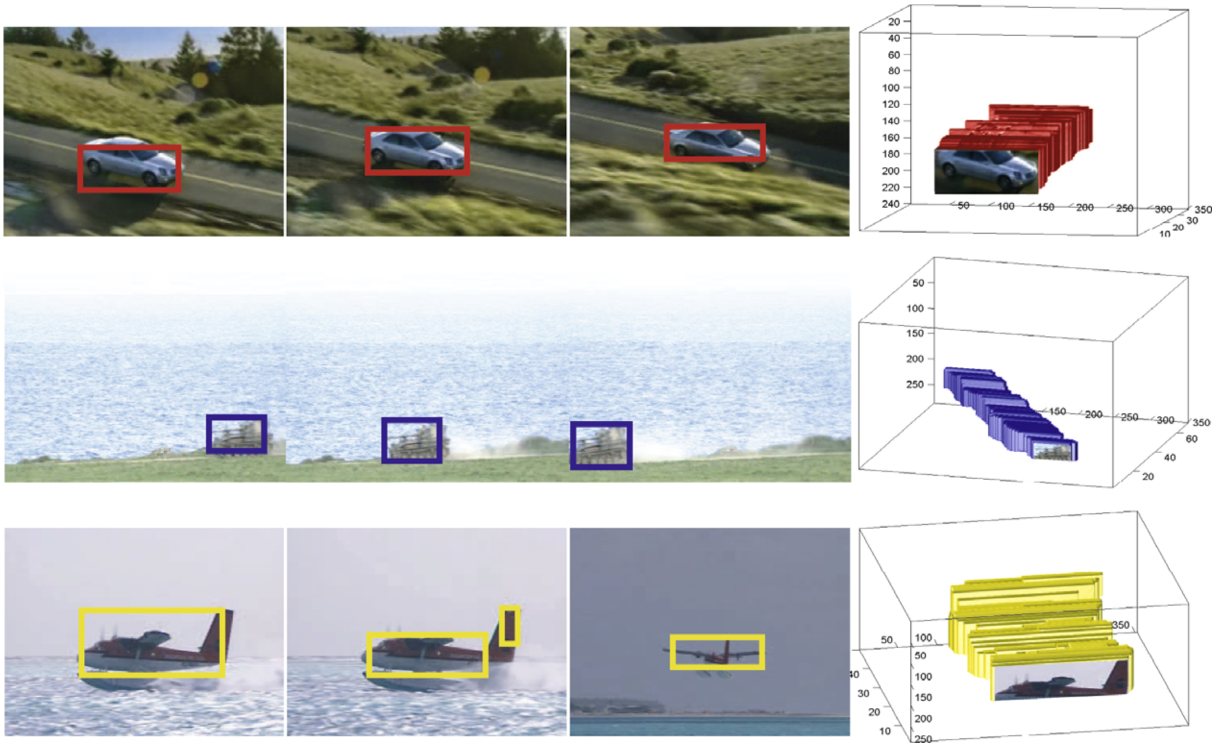


Fig. 8. Volume extraction from three input video shots. Each row contains three sample frames from the video along with the extracted volume. Each bounding box represents one of the several regions being tracked in each video. The consistent color of the bounding box represents the accuracy of region tracking. We are also able to handle split and merge in case of two regions belonging to the airplane (second frame). Both of these regions (with same yellow label) contribute to the airplane volume shown on the extreme right. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)

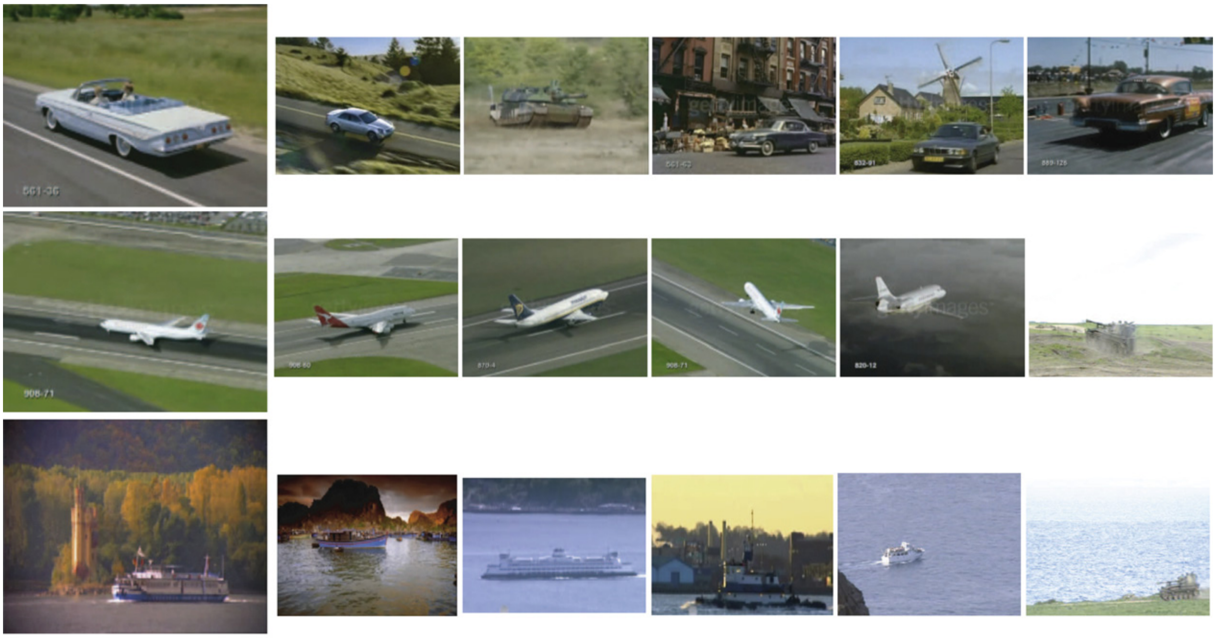
features are then compared against the features of other videos in the dataset. The matching score generated from each of these comparisons is used to rank the retrieved videos. Most similar video with the highest score is ranked first followed by the next most similar one and so on. The values of matching scores are normalized within the $[0, 1]$ interval. The highest similarity is represented by 1 and the lowest by 0. In our experiments we have used the top 100 returned videos for quantitative analysis of our framework. We argue that an optimized indexing technique is outside the scope of this paper. The purpose of our experiments here is to evaluate and prove the strengths of the proposed content based matching approach. We perform volume and feature extraction for the dataset offline, and at the retrieval time the feature matching and volume correspondence is performed against the query video.

Fig. 9 presents a sample output of the video retrieval for three query videos with the corresponding top 5 retrieved videos. Most of the retrieved videos represent similar type of foreground objects and background scenes. However, a very few of the retrieved videos appear to have objects of different types and that is because of high similarities in object motion and the background appearance. For instance, in case of the boat query video in Fig. 9, a tank video is retrieved at the end due to this reason.

The dataset used for our experiments did not have any ground truth annotation available for the video similarities. The idea of video similarity is dependent on many factors like appearance, type, and motion of foreground and background regions. Different people might have different perception of video similarity between two given videos. Hence, we asked five users from different backgrounds to help us with the annotation of similarity between videos in our dataset. They were asked to mark two videos similar if they thought that they contained similar semantic contents. For a given query video, the user annotates every retrieved video shots as *similar* or *not similar*. After looking at the query video, the user watches the retrieved videos one by one and provides the respective annotation. A snapshot of the annotation tool used for this task is shown in Fig. 10.

5.2.1. Qualitative analysis

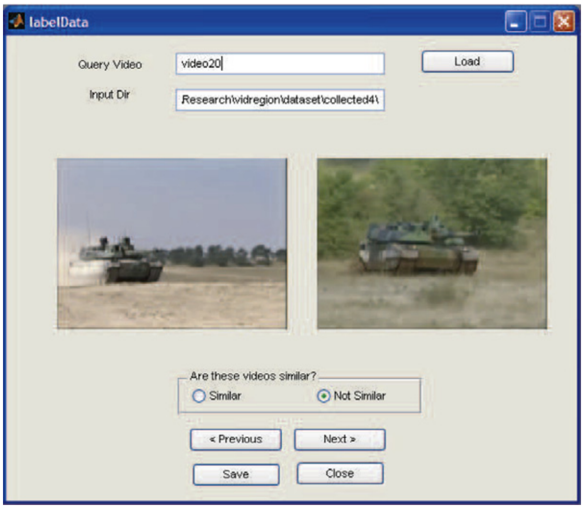
For the qualitative analysis of the results of our experiments we present the top 10 most relevant videos retrieved for different queries. Fig. 11 shows one example where the camera follows the car. Fig. 11c shows keyframes from the top 10 ranked videos after the retrieval is performed. All of these video shots were marked similar to the query video by the users except for the airplane and boat videos. The video containing a tank was marked similar by some but not by others. Fig. 12a presents another query containing an airplane that has just taken off from the ground. Fig. 12c lists the top 10 ranked results for this query. All of the retrieved videos contain moving airplanes except for one. It should be noted that in some of these videos the viewpoint of the airplane is very different. For instance, in the third ranked video here, also shown in Fig. 13, there is significant variation in the airplane's size and view. The SIFT descriptors from different frames (views) contribute to the respective volume features and are very useful for matching objects even when the views are similar only for a short interval. This signifies the usefulness of our approach where we extract complete spatiotemporal volumes instead



(a) Query Video

(b) Matched videos

Fig. 9. A snapshot of video retrieval. (a) Keyframes for three different queries, and (b) the top 5 ranked retrieved videos for each query.



Query Video video20

Input Dir Research\vidregion\dataset\collected4\

Are these videos similar?
○ Similar ● Not Similar

Fig. 10. Software tool used for annotation of videos. The user is first shown a query video and then the retrieved videos are shown in a sequence. The user labels each video as *Similar* or *Not Similar* to the query.

of relying on a single or multiple keyframes that could miss some of the views. Fig. 16a shows the input query video where a tank moves across the field in a stationary camera. Fig. 16c presents keyframes of the top 10 ranked videos. Note that our approach successfully matches the videos with quite different types and views of the tanks. The use of a wide variety of features including color and interest point features helps make this possible. Interest point descriptors capture the features on the object that help identify the same type of objects. There are two incorrect matches in this case and they occur because of background similarities. Similarly, results for two more queries are shown in Figs. 17 and 18.

5.2.2. Quantitative analysis

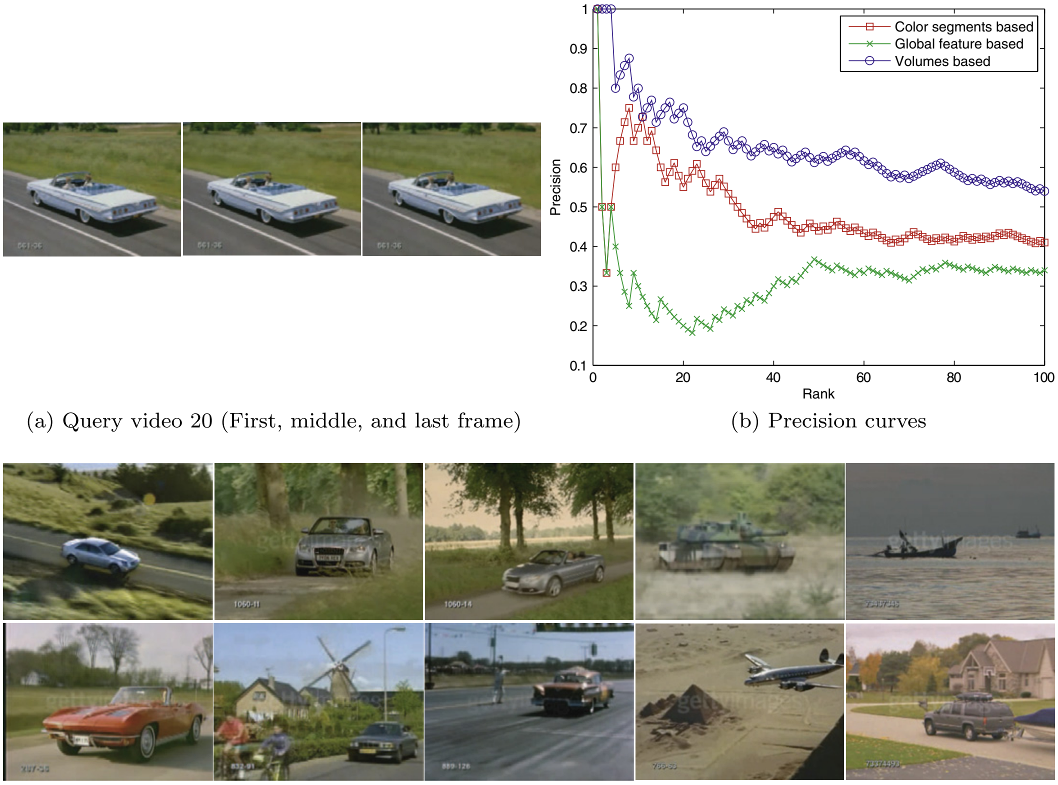
We also performed quantitative analysis of the video retrieval experiments. For a retrieval system the most useful performance measures include precision, recall, and average precision. These measures are defined as:

$$\text{Precision} = \frac{\{\text{Similar Videos}\}\bigcap\{\text{Retrieved Videos}\}}{\{\text{Retrieved Videos}\}},$$

$$\text{Recall} = \frac{\{\text{Similar Videos}\}\bigcap\{\text{Retrieved Videos}\}}{\{\text{Similar Videos}\}},$$

$$\text{Average Precision} = \frac{\sum_{i=1}^{m}\text{Precision}(r)\delta(r)}{\{\text{Similar Videos}\}},$$

where δ is the binary function on the relevance of the given rank r. The quality of retrieval ranking can be gauged by the average precision value. This metric favors the relevant videos to be ranked higher. It is an average of the precision values obtained after every video is retrieved. For every

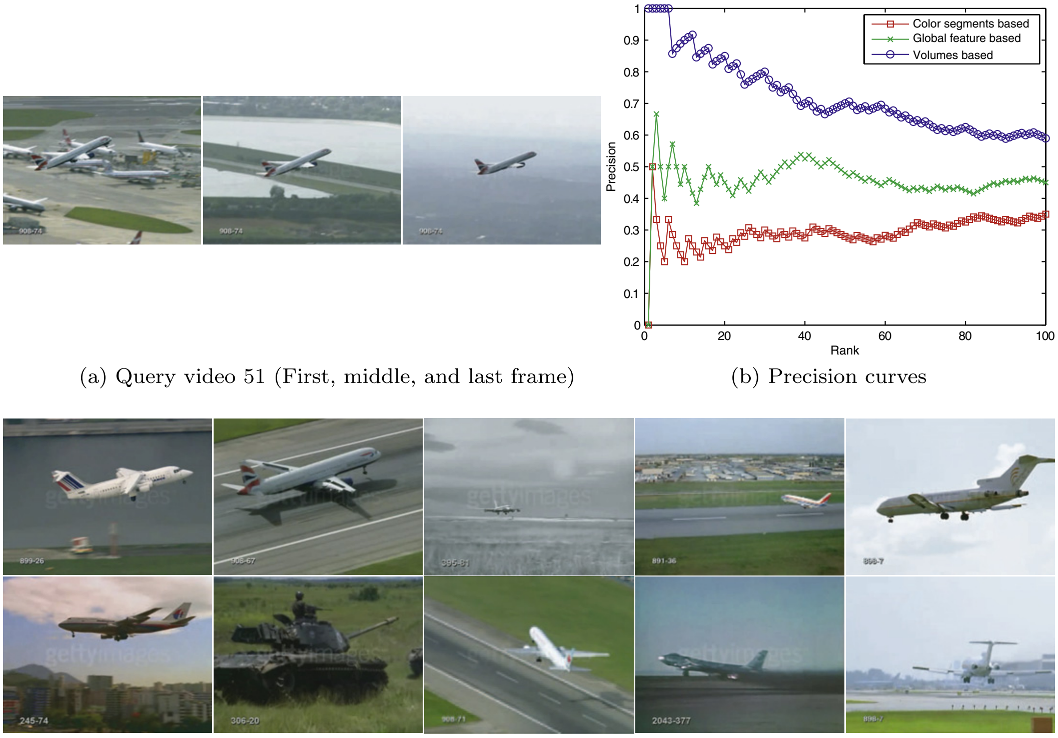


(a) Query video 20 (First, middle, and last frame)

(b) Precision curves

(c) Top ranked videos out of 100 returned videos

Fig. 11. Video retrieval result. (a) Frames of the input query video. (b) A comparison of performance against two other approaches which are based on keyframes. The proposed volume based (blue) method produces higher precision values. (c) Keyframes of the top 10 similar videos out of the 100 retrieved videos. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)



(a) Query video 51 (First, middle, and last frame)

(b) Precision curves

(c) Top ranked videos out of 100 returned videos

Fig. 12. Video retrieval result. (a) Frames of the input query video. (b) A comparison of performance against two other approaches which are based on keyframes. The proposed volume based (blue) method produces higher precision values. (c) Keyframes of the top 10 similar videos out of the 100 retrieved videos. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)

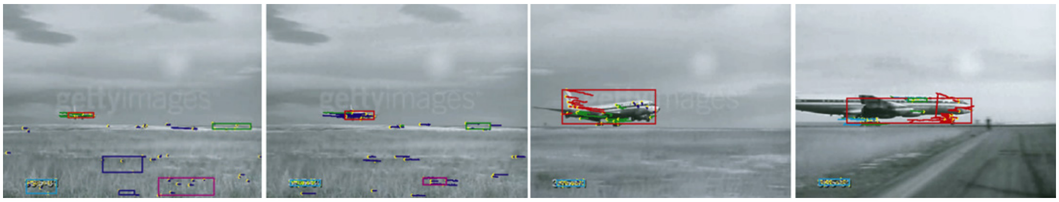


Fig. 13. Note the change in size and viewpoint of the tracked (red label) airplane through the video shot. The extracted volume corresponding to the airplane captures all these different views and can be very useful in matching an airplane with only one of the views. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)

retrieval query we compute the average precision value using the similarity annotation marked by the user.

We randomly selected five query videos. As explained earlier, five users provide similarity annotations between the query video and the retrieved videos. Table 1 presents average precision values computed for different users and query videos. For each user we also compute the mean average precision over the five queries that he/she has annotated. This provides us a measure of the quality of retrieval over a variety of query videos analyzed by different users. Only average precision is not sufficient to perform a meaningful evaluation because it provides information about only one query. For example, Table 1 shows that the average precision values for User4 varies between 0.49 and 0.77. The mean average precision values shown in the last column of the table are consistent for several users. We get the final values between 0.68 and 0.70 which seem to be more reasonable for a retrieval system than 0.49 average precision noted for video18. Fig. 17 shows query video 18 with 0.59 (by User1) average precision as shown in Fig. 14. The average precision falls to such a low value mainly because of low quality of features from the boat. Due to the strong similarity of the backgrounds, some videos with airplanes are matched to this boat video. Note that this is a more challenging query video and the other two approaches perform even worse with 0.34 and 0.28 average precisions as shown in Fig. 14.

Another interesting observation can be made from Table 1 regarding the variation in the interpretation of the video similarity by different users. For these five queries, although there were some minor differences in the interpretations, but we did not observe any significant discrepancy in any particular user's annotation. For instance, the maximum variation in the average precision value for a

Table 1
Mean average precision computed for our method using several query videos shown in Figs. 11, 12, 16, 17, and 18. The ground truth is annotated by five users for our experiments

Annotator ID	Average precision					Mean average precision
	Video20	Video18	Video51	Video117	Video5	
User1	0.69	0.59	0.75	0.71	0.74	0.70
User2	0.70	0.56	0.75	0.71	0.75	0.69
User3	0.68	0.61	0.72	0.69	0.76	0.69
User4	0.71	0.49	0.71	0.70	0.77	0.68
User5	0.71	0.55	0.77	0.72	0.76	0.70

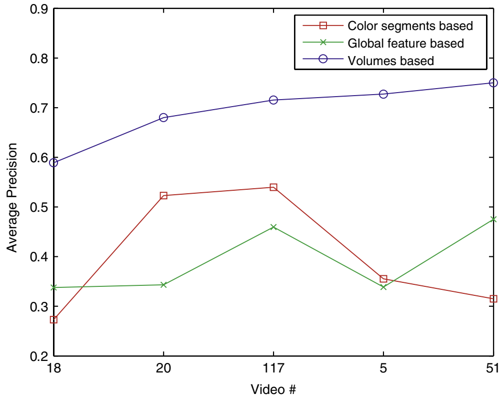


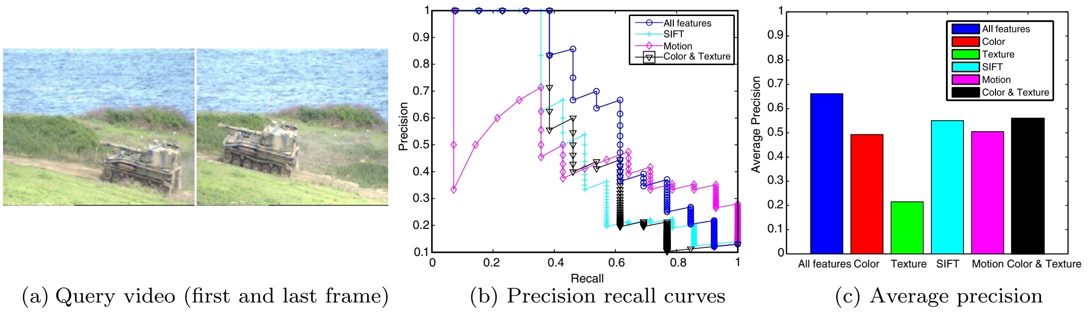
Fig. 14. The performance of the proposed and the two baseline approaches is compared using the average precision values for five queries shown in Figs. 11, 12, 16, 17, and 18. The proposed volume based approach performs better for a variety of different queries as shown here with higher average precision value.

particular query video (video18) is 12%. This is a reasonable amount of variation and tells us that the users perceive the matching videos in our dataset similarly. We can also use this degree of variation to detect any abnormal annotation by a particular user and video combination. The outlier annotation will be detected if there is large (greater than 50%) variation from other user's annotation of the same video.

5.2.3. Comparison

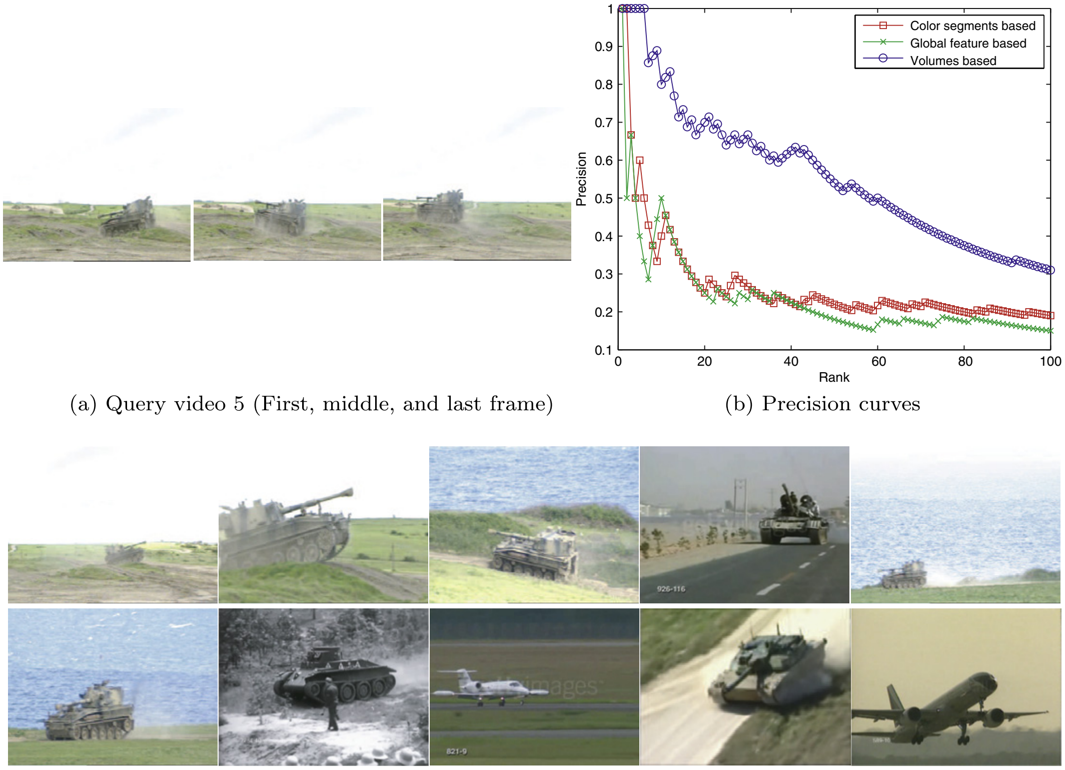
We have compared our approach with two baseline methods to analyze the significance of using the spatiotemporal volumes and their features. Both of these approaches use a keyframe to represent the entire video shot and we choose the middle frame for this purpose. The first method is based on the global visual features computed for the complete frame. The second method relies on color segments based visual features. Mean shift color segmentation algorithm [37] is used for this purpose. The features computed in both of these approaches include a 3D HSV color histogram with (18,3,3) bins and an edge orientation histogram in eight directions. For the second method we also used the region size to prune out noisy region matches. The technique for region correspondence is exactly the same as the one used for volume correspondence as described in Section 4.

Similar to our approach, we also compute the precision, and average precision values corresponding to the retrieval from these approaches. Fig. 11b shows the precision vs rank curves for the three approaches in case of the car query. Our approach (blue circles) clearly outperforms the other two approaches by showing higher precision and recall values. Similarly, the precision curves in Figs. 12b, 16b, 17b, and 18b clearly show the superior performance of the volume based approach. Since our criteria for better performance is the quality of ranking, it should therefore be noted that the curves for the volume based approach show more relevant videos in the top ranked results. Fig. 14 clearly shows that the volume based approach presents higher average precision values and outperforms the other two approaches. This supports the claim that the proposed approach is able to handle variations in object and background appearance, views, motion,



(a) Query video (first and last frame) (b) Precision recall curves (c) Average precision

Fig. 15. This figure presents the effects of different combinations of features on the quality of retrieval for the query video. Several combinations can be chosen based on different weights of these features. It was found that the best performance is achieved using a combination of all the volume based features.

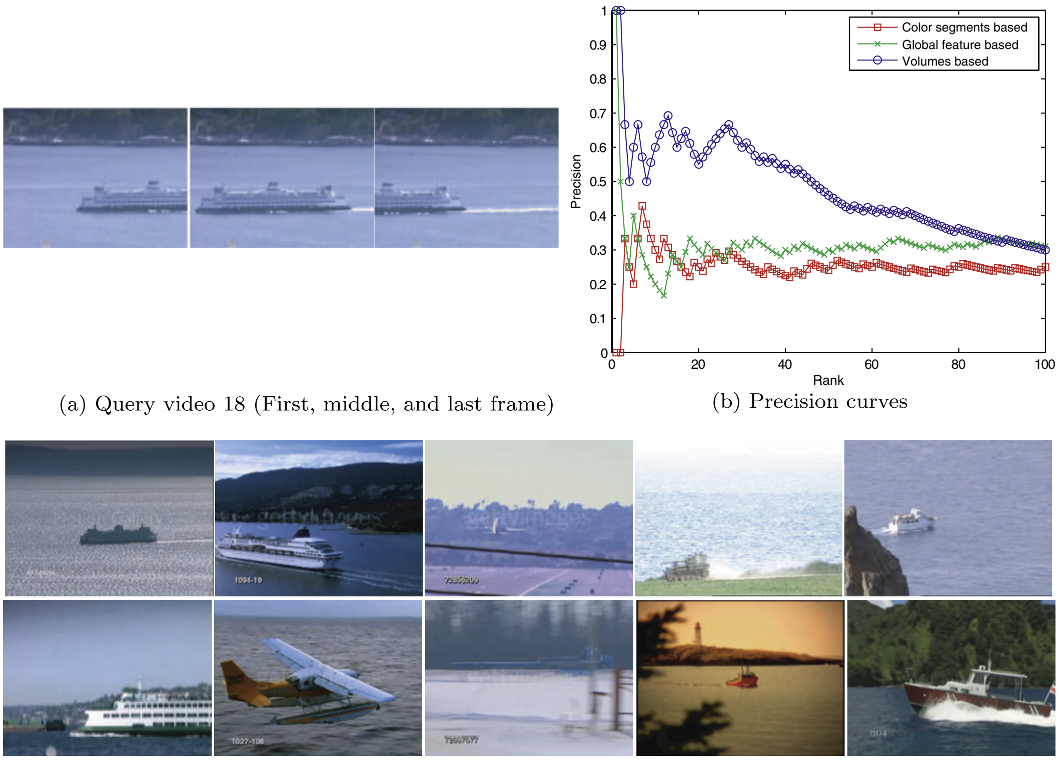


(a) Query video 5 (First, middle, and last frame)

(b) Precision curves

(c) Top ranked videos out of 100 returned videos

Fig. 16. Video retrieval result. (a) Frames of the input query video. (b) A comparison of performance against two other approaches which are based on keyframes. The proposed volume based (blue) method produces higher precision values. (c) Keyframes of the top 10 similar videos out of the 100 retrieved videos. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)

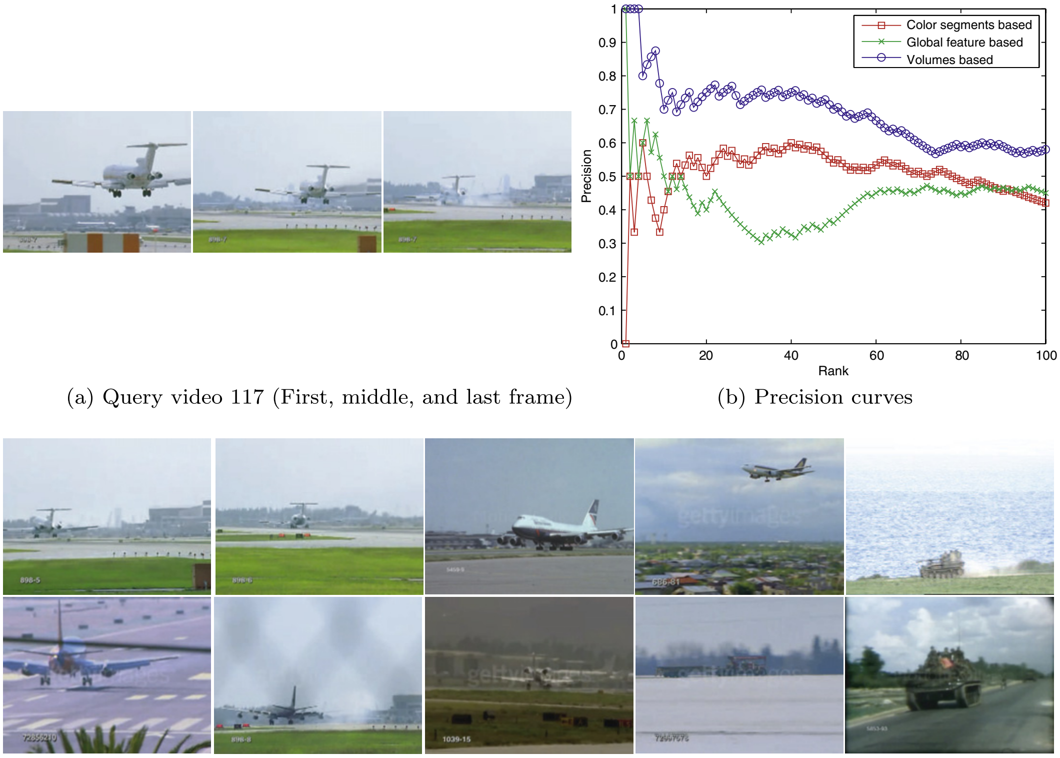


(a) Query video 18 (First, middle, and last frame)

(b) Precision curves

(c) Top ranked videos out of 100 returned videos

Fig. 17. Video retrieval result. (a) Frames of the input query video. (b) A comparison of performance against two other approaches which are based on keyframes. The proposed volume based (blue) method produces higher precision values. (c) Keyframes of the top 10 similar videos out of the 100 retrieved videos. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)



(a) Query video 117 (First, middle, and last frame)

(b) Precision curves

(c) Top ranked videos out of 100 returned videos

Fig. 18. Video retrieval result. (a) Frames of the input query video. (b) A comparison of performance against two other approaches which are based on keyframes. The proposed volume based (blue) method produces higher precision values. (c) Keyframes of the top 10 similar videos out of the 100 retrieved videos. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)

etc., in a better way and produces superior video matching results.

5.2.4. Feature combinations

We also performed experiments to analyze the significance of different types of features used for matching. As described in Section 3, there are four different types of volume features we have used in this framework including color, texture, interest point descriptor (SIFT), and motion features. We observed that SIFT feature was very useful for matching objects with variations in size and pose. Feature performance was initially analyzed individually and then in combinations. Fig. 15b shows four precision recall curves with the best rankings achieved by *All features*. The features were combined using the method explained in Section 4. Similarly, Fig. 15c shows that *All features* give the best performance with the highest value of average precision. Individual precision recall curves for color and texture features have been omitted to make the figure clear, however, average precision values are shown. It is clear from the figure that the combination of conventional *Color & Texture* features show limited performance. However, after incorporating SIFT and motion features extracted from the video volumes, we achieve a higher level of performance in video matching. The usefulness of these features have been observed and explained in the retrieval results presented here.

6. Conclusions

In this paper, we have proposed a novel and robust video matching framework by analyzing properties of spatiotemporal volumes in videos. Volumes are constructed based on the clustering of the interest point trajectories. Multiple features are extracted to model the appearance of the volumes, including color, texture, motion, and interest point descriptors. Similarity between two videos is computed by solving the maximum matching problem of the graph formed by the volumes. Utilizing the proposed video matching framework, we have achieved very promising and competitive performance in video matching for retrieval.

In the current form, our framework focuses on the importance of spatiotemporal volumes and the related features to match the videos. We feel that the particular combination of the color, texture, motion, and interest point descriptors based features can be crucial for different queries. One possible approach for improving the feature combination is relevance feedback by the user. In future, we plan to explore relevance feedback for this task. In addition to this, it can also be applied towards reducing the semantic gap between low level feature matching and higher level video matching. Another interesting future direction is to study the interaction between different volumes within a video. This can provide useful information regarding differ-

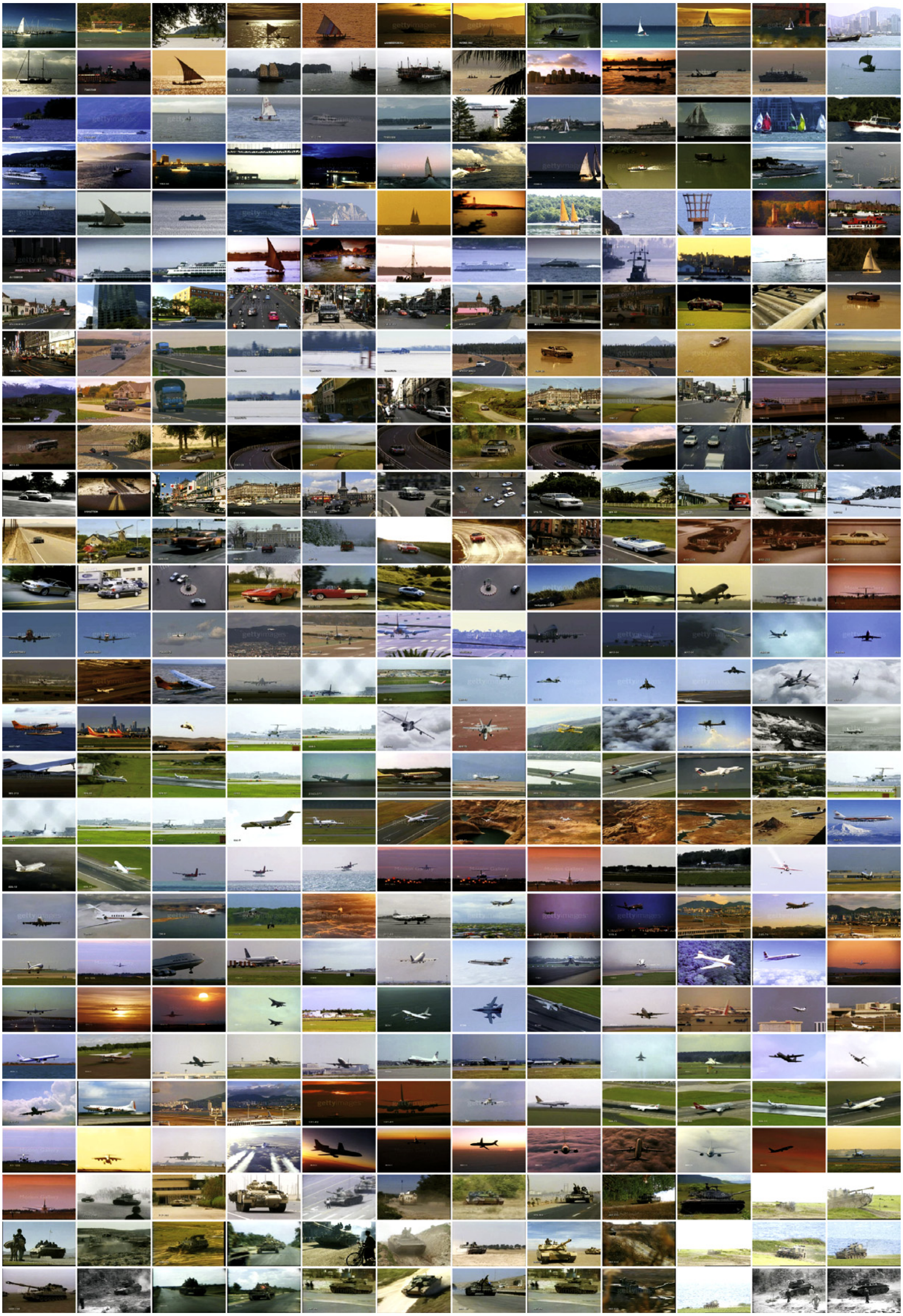


Fig. 19. A snapshot of our dataset that includes 337 videos comprising of 74 boat, 80 car, 148 airplanes, and 35 tank videos. The significant variation in the object appearance within each category makes this dataset very challenging.

ent types of events and activities being performed in a given video.

References

[1] A. Pentland, R. Picard, S. Sclaroff, Photobook: content-based manipulation of image databases, International Journal of Computer Vision 18 (3) (1996) 233–254.

[2] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, W. Equitz, Efficient and effective querying by image content, Journal of Intelligent Information Systems 3 (3) (1994) 231–262.

[3] J. Smith, S. Chang, Visualseek: a fully automated content-based image query system, in: Proceedings of the 4th ACM International Conference on Multimedia, 1997, pp. 87–98.

[4] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld: image segmentation using expectation-maximization and its application to image querying, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002) 1026–1038.

[5] H. Greenspan, G. Dvir, Y. Rubner, Region correspondence for image matching via emd flow, in: Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries, 2000, pp. 27–31.

[6] F. Jing, M. Li, H. Zhang, B. Zhang, Region-based relevance feedback in image retrieval, in: IEEE International Symposium on Circuits and Systems, 2002, vol. 4, ISCAS 2002.

[7] J. Wang, J. Li, G. Wiederhold, Simplicity: semantics-sensitive integrated matching for picturelibraries, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (9) (2001) 947–963.

[8] R. Mohan, Text-based search of tv news stories, Proceedings of SPIE 2916 (1996) 2.

[9] A. Hampapur, A. Gupta, B. Horowitz, C. Shu, C. Fuller, J. Bach, M. Gorkani, R. Jain, Virage video engine, in: Proceedings of SPIE 3022, 1997, p. 188.

[10] A. Hauptmann, M. Witbrock, Informedia: News-on-demand multimedia information acquisition and retrieval, Intelligent Multimedia Information Retrieval (1997) 215–239.

[11] S. Chang, W. Chen, H. Meng, H. Sundaram, Videoq: an automated content based video search system using visual cues, in: Proceedings of the 5th ACM International Conference on Multimedia, 1997, pp. 313–324.

[12] J. Lee, J. Oh, S. Hwang, Strg-index: spatio-temporal region graph indexing for large video databases, in: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, 2005, pp. 718–729.

[13] S. Dagtas, W. Al-Khatib, A. Ghafoor, R. Kashyap, Models for motion-based video indexing and retrieval, IEEE Transactions on Image Processing 9 (1) (2000) 88–101.

[14] S. Sav, N. OConnor, A. Smeaton, N. Murphy, Associating low-level features with semantic concepts using video objects and relevance feedback, in: 6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005), Montreux, Switzerland, 2005, pp. 13–15.

[15] A. Smeaton, H. Le Borgne, N. OConnor, T. Adamek, O. Smyth, S. De Burca, Coherent segmentation of video into syntactic regions, in: 9th Irish Machine Vision and Image Processing Conference, 2005.

[16] E. Ardizzone, M. La Cascia, D. Molinelli, Motion and color based video indexing and retrieval, in: Proceedings of the International Conference on Pattern Recognition 1996, pp. 135–139.

[17] Y. Deng, B. Manjunath, Netra-v: toward an object-based video representation, IEEE Transactions on Circuits and Systems for Video Technology 8 (5) (1998) 616–627.

[18] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: Proceedings of 9th IEEE International Conference on Computer Vision, 2003, pp. 1470–1477.

[19] J. Sivic, M. Everingham, A. Zisserman, Person spotting: video shot retrieval for face sets, in: Proceedings of CIVR, July.

[20] J. Sivic, F. Schaffalitzky, A. Zisserman, Object level grouping for video shots, International Journal of Computer Vision 67 (2) (2006) 189–210.

[21] F. Rothganger, S. Lazebnik, C. Schmid, J. Ponce, Segmenting, modeling, and matching video clips containing multiple moving objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 477–491.

[22] F. Schaffalitzky, A. Zisserman, Automated location matching in movies, Computer Vision and Image Understanding 92 (2003) 236–264.

[23] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[24] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, International Journal of Computer Vision 1 (4) (1988) 321–331.

[25] Y. Rubner, C. Tomasi, L. Guibas, The earth mover's distance as a metric for image retrieval, International Journal of Computer Vision 40 (2) (2000) 99–121.

[26] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., 1988.

[27] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1265–1278.

[28] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, International Journal of Computer Vision 73 (2) (2007) 213–238.

[29] F. Hitchcock, The distribution of a product from several sources to numerous localities, Journal of Mathematical Physics 20 (1941) 224–230.

[30] R. Meir, G. Ratsch, An introduction to boosting and leveraging: advanced Lectures on Machine Learning, LNCS, 2003, pp. 119–184.

[31] L. Lov'asz, M.D. Plummer, Matching Theory, North-Holland, 1986.

[32] Trec video retrieval track (2005). URL http://www-nlpir.nist.gov/projects/trecvid/.

[33] Google video. URL http://video.google.com/.

[34] BBC motion gallery. URL http://www.bbcmotiongallery.com/Customer/RoyaltyFree.aspx.

[35] G. Ahanger, T. Little, A survey of technologies for parsing and indexing digital video, Journal of Visual Communication and Image Representation 7 (1) (1996) 28–43.

[36] A. Yilmaz, M. Shah, Shot detection using principle coordinate system, in: International Conference, Internet and Multimedia Systems and Applications, 2000, pp. 223–225.

[37] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.