Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 26 (2008) 1384–1396
www.elsevier.com/locate/imavis

Automatic detection of salient objects and spatial relations in videos for a video database system

Tarkan Sevilmiş, Muhammet Baştan, Uğur Güdükbay, Özgür Ulusoy *

Department of Computer Engineering, Bilkent University, Bilkent, 06800 Ankara, Turkey

Received 10 March 2006; received in revised form 25 December 2007; accepted 3 January 2008

Abstract

Multimedia databases have gained popularity due to rapidly growing quantities of multimedia data and the need to perform efficient indexing, retrieval and analysis of this data. One downside of multimedia databases is the necessity to process the data for feature extraction and labeling prior to storage and querying. Huge amount of data makes it impossible to complete this task manually. We propose a tool for the automatic detection and tracking of salient objects, and derivation of spatio-temporal relations between them in video. Our system aims to reduce the work for manual selection and labeling of objects significantly by detecting and tracking the salient objects, and hence, requiring to enter the label for each object only once within each shot instead of specifying the labels for each object in every frame they appear. This is also required as a first step in a fully-automatic video database management system in which the labeling should also be done automatically. The proposed framework covers a scalable architecture for video processing and stages of shot boundary detection, salient object detection and tracking, and knowledge-base construction for effective spatio-temporal object querying.

Keywords: Multimedia databases; Salient object detection and tracking; Camera focus estimation; Object labeling; Knowledge-base construction; Spatio-temporal queries

1. Introduction

The rapid increase in the amount of multimedia data has resulted in the development of various technologies for handling large volumes of data. These technologies concentrate on efficient compression and storage of the multimedia data. One of the particularly interesting storage systems is the "multimedia database", which stores multimedia content that can be queried by various features [1]. Some examples of prototype multimedia database systems include QBIC [2], VisualSeek [3], and VideoQ [4]. If the media is video, the queried features are spatial, spatio-temporal, semantic, motion (e.g., object trajectories) and object features (e.g., color, shape, texture) [5]. Before retrieval, the videos need to be processed to extract the features that can be queried. Since the first appearance of multimedia databases, research on efficient means of feature extraction has become popular. To this end, some manual, semi-automatic and automatic tools have been developed [3,4,6–8].

This paper proposes a framework for automatic salient object extraction. The framework is used to detect and track salient objects in a given video automatically and construct a list of spatio-temporal features for our video database system, *BilVideo* [9]. This process aims to reduce the huge amount of time and effort required for manual selection and labeling of objects for video indexing.

We first use a color histogram-based shot boundary detector to find the shot boundaries. Then, each video is processed on a shot basis following the common practice; salient objects within each shot are detected and tracked. At the end of processing each shot, tracked objects satisfying

* Corresponding author. Tel.: +90 312 290 15 77; fax: +90 312 266 40 47.
E-mail addresses: sevilmis@cs.bilkent.edu.tr (T. Sevilmiş), bastan@cs.bilkent.edu.tr (M. Baştan), gudukbay@cs.bilkent.edu.tr (U. Güdükbay), oulusoy@cs.bilkent.edu.tr (Özgür Ulusoy).

0262-8856/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2008.01.001

a consistency requirement are saved. Once the object extraction is completed, the extracted objects are labeled (annotated) manually. Finally, the spatio-temporal relations between the labeled objects are computed using their minimum bounding rectangles and stored into the database.

The rest of the paper is organized as follows. Section 2 gives an overview of our video database system, BilVideo, and the motivation for this work. We discuss related work on video processing for automatic object extraction and tracking in Section 3. In Section 4, we give an overview of the proposed framework and explain our approach for salient object detection, tracking, and labeling. We present experimental results on news videos from TRECVID 2006 video corpus and evaluate the performance of our system in Section 5. We conclude with possible future improvements to the system in Section 6.

2. BilVideo video database system

2.1. System architecture

BilVideo is our prototype video database system [9]. The architecture of *BilVideo* (Fig. 1) is original in that it provides full support for spatio-temporal queries that contain any combination of spatial, temporal, object-appearance, external-predicate, trajectory-projection, and similarity-based object-trajectory conditions by a rule-based system built on a knowledge-base, while utilizing an object-relational database to respond to semantic (keyword, event/activity, and category-based), color, shape, and texture queries. The knowledge-base of BilVideo contains a fact-base and a comprehensive set of rules implemented in Prolog. The rules in the knowledge-base significantly reduce the number of facts that need to be stored for spatio-temporal querying of video data [10]. Query processor interacts with both the knowledge-base and object-relational database to respond to user queries that contain a combination of spatio-temporal, semantic, color, shape, and texture video queries. Intermediate query results returned from these two system components are integrated seamlessly by the query processor, and final results are sent to the Web clients.

To the best of our knowledge, *BilVideo* is by far the most feature-complete video DBMS (database management system), as it supports spatio-temporal, semantic, color, shape, and texture queries in an integrated manner. Moreover, it is also unique in its support for retrieving any segment of a video clip, where the given query conditions are satisfied, regardless of how video data is semantically partitioned. To our knowledge, none of the video query systems available today can return a subinterval of a scene as part of a query result, simply because video features are associated with shots defined to be the smallest semantic units of video data. In our approach, object trajectories, object-appearance relations, and spatio-temporal relations between video objects are represented as Prolog facts in a knowledge-base, and they are not explicitly related to semantic units of videos. Thus, *BilVideo* can return precise answers for user queries, when requested, in terms of frame intervals. Moreover, our assessment for the directional relations between two video objects is also novel in that two overlapping objects may have directional relations defined for them with respect to one another, provided that center points of the objects' MBRs (minimum bounding rectangles) are different. Furthermore, BilVideo query language provides three aggregate functions, average, sum, and count, which may be very attractive for some applications, such as sports analysis systems and mobile object tracking systems, to collect statistical data on spatio-temporal events.

Fact Extractor (see Fig. 1) is used to populate the fact-base of *BilVideo*, and extract color and shape histograms of the objects in video key frames. Spatio-temporal relations between objects, object-appearance relations, and

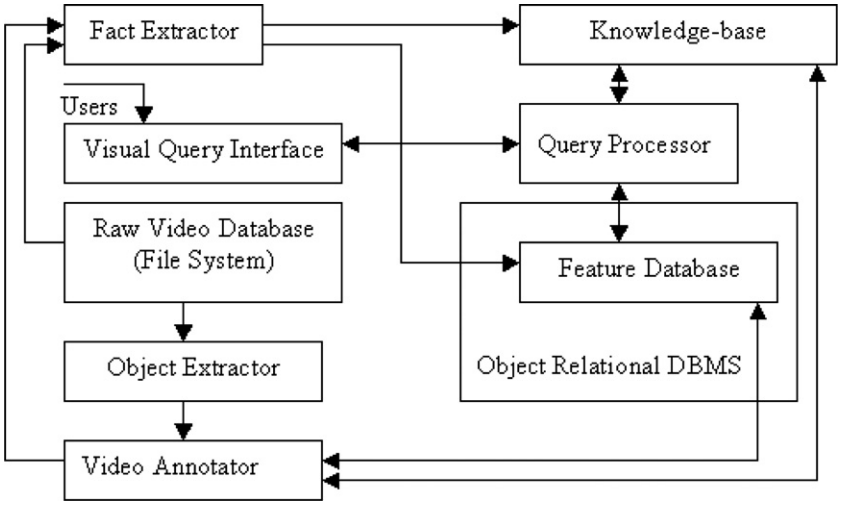

Fig. 1. Overall architecture of *BilVideo* video database system.

object trajectories are extracted semi-automatically. The set of facts representing the relations and trajectories are stored in the fact-base, and it is used for spatio-temporal query processing. Sets of facts are kept in separate fact-files for each video clip processed, along with some other video specific data, such as video length, frame rate, key frames list, etc., extracted automatically by the tool. Extracted color and shape histograms of salient objects are stored in the feature database to be used for color and shape queries. Previously, the fact extraction relied on a semi-automatic object extraction process in which the user needed to specify objects in each video frame by their MBRs, requiring a huge amount of time and effort to index even very short videos.

2.2. Motivation for this work

The motivation for this work is twofold:

1. If the object extraction tool used in a video database system is operated manually, this induces a serious drawback on the usefulness of the system on large data sets. On a fully-manual system, the user should specify the object MBRs and labels for each frame. It takes about 20–30 s to process each frame manually. An average 5-min video contains 7500 frames which requires a total of 52 h of manual work. On the other hand, if the objects can be detected and tracked automatically, the user should label the objects only at the beginning of each shot or when they first appear. The system will label the rest of the frames in the shot automatically. Hence, the required manual work decreases tremendously from tens of hours to a few minutes. Therefore, automatic detection of objects and relationships between them can provide significant contribution to the effectiveness of a video database system.
2. Another very important motivation for an automatic salient object extraction tool is the need for fully-automatic video database systems, in which the object labeling (annotation) is also done automatically. Therefore, we focus on the development of an automatic salient object extraction tool for *BilVideo* in this work. We use classical computer vision techniques to build the system and emphasize the general framework rather than individual algorithms that already exist in the literature.

3. Related work

In this section, we review related work on video processing for automatic video object extraction and tracking, still being challenging issues in computer vision. The literature is huge; therefore, we only provide pointers to some of the recent work that helped us in developing our system. Video object segmentation, extraction, detection are all similar in meaning with slight differences. Therefore, we use them interchangeably.

3.1. Shot boundary detection

Almost every application in video processing exploits shot boundary information obtained from a shot boundary detector. This information is helpful in delimiting the video into frame intervals during which different sets of objects appear. Shots in a video are collections of consecutive frames that are usually taken by a single camera and that share a common scene. Shots are considered as the elementary units of a video. The transition between shots are various, ranging from easy-to-detect hard cuts to more difficult dissolves, wipes, fades, and slides. There are several works on detecting shot boundaries, as reviewed in [11,12]. This is also one of the tasks in TRECVID conference and workshop series [13]. Some of the methods used to detect shot boundaries are as follows:

- *Color and edge histogram methods* are based on comparison of successive frames by their color and edge histograms.
- *Edge/contour based methods* utilize the discontinuity in edges and contours at the shot boundaries.
- *Motion based methods* measure the discontinuity in motion during the transition.
- *Pixel-differencing methods* count the pixels changed according to a certain threshold from one frame to another, and assume a shot boundary if the number exceeds another threshold.
- *Statistical methods* calculate statistical features of pixels, such as mean and variance, and compare them with the preceding frames to detect shot boundaries.

3.2. Video object segmentation and tracking

Video object segmentation is used to identify regions of interest in a scene and is one of the most challenging tasks in video processing. It is a key step in many applications, including content-based indexing and retrieval, compression, recognition, event analysis, understanding, video surveillance, intelligent transportation systems, and so on. The problem of unsupervised video object segmentation is ill-defined because semantic objects do not usually correspond to homogeneous spatio-temporal regions in color, texture, or motion.

Existing approaches to video object segmentation include spatial segmentation and motion tracking, motion-based segmentation, moving object extraction, region growing using spatio-temporal similarity. These approaches can be grouped in two broad categories: spatial segmentation followed by integration of temporal information to merge regions and motion-based segmentation. Both of the approaches involve no user interaction, therefore, the segmented objects are often not consistent with human visual perception. Consequently, practical application of these algorithms is normally limited to region segmentation rather than video object segmentation [14].

There is a large literature on spatial image segmentation ranging from graph-based methods, region merging techniques, graph cuts to spectral methods. In JSEG algorithm [15], images are first quantized to several representative classes. Then, each pixel is replaced by its representative class label. By applying a "good" segmentation criterion to local windows, they produce what they call a "J-image". Finally, a region growing approach is used to segment the image based on multi-scale J-images. It is also applied to video sequences with an additional region tracking scheme and shown to be robust on real images and video.

In Blobworld [16], segmentation is obtained by clustering pixels in a joint color–texture-position feature space using Expectation Maximization (EM). In [17], the authors construct an edge flow graph based on detected edges, and use the graph to find objects in the scene. Normalized Cut [18] algorithm constructs a graph from the image; each node (pixel) has an edge to all other nodes (pixels). The segmentation is obtained by finding the normalized cuts of the graph. It is one of the most successful image segmentation algorithms in literature but it is computationally costly. An efficient graph-based segmentation is proposed in [19]. It runs in time linearly with the number of graph edges and is faster than the Normalized Cut algorithm. It is a greedy algorithm and works by first sorting the edges in increasing order of weight and then processing the edges in this order in the segmentation of the graph. Finally, a disjoint set forest (DSF) is obtained; each set corresponds to one component in the image.

The details of moving object segmentation and spatio-temporal segmentation can be found in [14,20–26] and tracking in [27–31].

3.3. Saliency detection

In the literature, salient objects are defined as the visually distinguishable, conspicuous image components that attract our attention at the first glance. These are usually high contrast regions, or regions with significantly different appearance compared to their surroundings. Detection of salient regions is also referred to as *image attention analysis*.

The literature on saliency analysis is broad. The first remarkable work in the field is [32]. It combines multi-scale image features into a single topographical saliency map. Using this map and a dynamic neural network, the attended image locations are selected in order of decreasing saliency. In [33], a saliency map is generated based on local contrast analysis, and a fuzzy growing method is used to extract attended areas or objects from the saliency map by simulating human perception. In [34], the authors propose a salient object extraction method by a contrast map using three features (luminance, color, and orientation), and salient points for object-based image retrieval. The work in [35] investigates empirically to what extent pure bottom-up attention can extract useful information about the location, size and shape of objects from images and demonstrates how this information can be utilized to enable unsupervised learning of objects from unlabeled images. In [36], image segmentation is formulated as the identification of single perceptually most salient structure in the image. In [37], salient regions in remote-sensed images are detected based on scale and contrast interaction using local contrast features obtained by Gabor filters. The detection of salient structure exploits a probabilistic mixture model taking two series of multi-scale features as input related to contrast and size information. The model parameters are learned by an EM-type algorithm, and each pixel is classified as being salient or not, resulting in a binary segmentation. In [38], salient object detection is formulated as an image segmentation problem, in which salient object is separated from the image background. A set of novel features are proposed: multi-scale contrast, center-surround histogram, and color spatial distribution to describe a salient object locally, regionally, and globally. A Conditional Random Field (CRF) is learned using a human labeled set of training images to effectively combine these features for salient object detection.

4. System overview

In this section, we first describe the structure of our automatic salient object extraction tool, and then give an overview of the framework that is used to process videos. Our tool works by performing four major tasks in series:

1. shot boundary detection,
2. salient object detection,
3. object tracking, and
4. user interaction, object labeling and knowledge-base construction.

Properties of the video corpus for which a video database system is designed affect the choice of methods to be employed in the system. For example, an object extraction tool designed for surveillance videos will not perform well in news videos. We have made the following assumptions while designing our system targeting complex news videos.

1. The scene background is not known and is not static, since news videos have video footage of multiple unknown locations during interviews. Most of the backgrounds are outdoor environments, such as busy streets, resulting in highly dynamic backgrounds.
2. The scene changes often since news videos include interviews and video footage from different places.
3. Objects are not distinct from the background. Many objects have the same color distribution as the background. An immediate example is a person being interviewed in a crowd, or an anchor worn in black in a dark studio.

4. The number of interesting objects in the scene is limited. Since news generally focuses on a specific person, or a meeting of 2 or 3 people, we do not generally expect more than 4–5 salient objects in the scene.
5. Object movement is smooth, which means there will not be any sudden changes in direction or speed of objects.
6. Occlusions may happen but are not common. Since news videos mainly focus on the target person or event, it is quite uncommon for the target to be occluded by another object.

Some of these assumptions, which are observed in news videos, limit the usability of existing methods and algorithms. Assumptions 1 and 2 prevent efficient use of background subtraction and construction methods for object detection. Assumption 3 limits the usability of basic segmentation methods for images.

In news videos, most important objects are usually humans. Therefore, detailed spatio-temporal search for people will be very useful. For example, the user may want to retrieve the video segments in which *Bush* is having a conversation with *Putin*, and *Bush* is to the right of *Putin*. To improve the performance of the system on such queries, we employ a face detector running in parallel with the salient object detector. In this way, we aim to ensure higher accuracy in detecting the presence of people in case we cannot obtain a good segmentation, which frequently occurs in complex and dark scenes.

4.1. Shot boundary detection

We need a good enough shot boundary detector with performance close to the state-of-the-art in this field. Histogram-based methods are simpler and computationally more efficient compared to the other methods, yet the performance is also satisfactory [11,12]. Therefore, we take a simple approach to shot boundary detection using a normalized hue-saturation histogram-based frame difference. The distance between successive frames is measured using the Bhattacharyya metric. Sharp transitions (cuts) are easy to detect by setting a single threshold above which a shot boundary is declared. To detect smooth transitions (fades, dissolves), we use two thresholds: *a low threshold* to detect the starting point of possible smooth transition and *a high threshold* to declare a shot boundary. When the low threshold is exceeded, we save the last frame and compare it to the first frame in which the inter frame distance falls below the low threshold again. If the distance is larger than the high threshold, there is a shot boundary.

4.2. Salient object detection

Salient object detection is the most important part of our system. Decoded video frames are first preprocessed to remove noise and flatten the texture for better segmentation. Then, each frame is segmented into regions and saliencies of these regions are measured according to some heuristics (as described in Section 4.2.5). Finally, salient regions are tracked throughout the shot at the end of which tracked object information is saved for manual labeling in a later stage.

Considering the assumptions related to the content and difficulty level of the videos, we take a spatio-temporal segmentation approach. First, we segment the individual frames using a spatial segmentation method. Then, we try to merge the regions of the over-segmentation by first using color, and then, velocity properties of the regions. In our implementation, temporal information is limited to the velocity obtained by optical flow between two successive frames. At the end of the segmentation, we expect to get more meaningful regions in terms of human visual perception, corresponding to semantic objects (e.g., a human, a car). However, this is still not achievable with the current state-of-the-art in computer vision; final segmentation does not usually correspond to semantic objects. For example, a person may be segmented into two pieces, a face region and a body region.

4.2.1. Preprocessing

We use bilateral filtering [39] to smooth the video frames while preserving the edges as much as possible. To obtain good segmentation, textured areas should be flattened but edges should not be lost. This is exactly what the bilateral filtering is designed for. We use Lab color space (CIELAB) since unlike RGB space, Lab color is designed to approximate human vision. It is perceptually more uniform, and its L component closely matches human perception of lightness.

4.2.2. Initial segmentation

Graph-based approaches are very successful for image segmentation. We obtain the initial segmentation of video frames using the efficient graph-based approach described in [19]. We adapted the freely available implementation of the author to our system. Although the graph-based approach in [18] is more successful, it is computationally demanding.

First, the frames are smoothed with 3×3 bilateral filtering to flatten the textured regions. Then, an undirected graph is constructed. Each node corresponds to a pixel and each edge connects a pair of neighboring pixels. We use 4-connectedness for each pixel and calculate the edge weights as the Euclidean distance in color between the pair of connected pixels. The threshold and minimum area size parameters of the algorithm are set as 300 and 100 as suggested in the paper. For the affect of these parameters on the segmentation, please see the original paper [19]. The graph is segmented into a disjoint set forest (DSF) using a greedy approach by first sorting the edges into increasing order according to edge weights. All edges are scanned in increasing order of weight. If the edge weight between a pair of nodes is less than the maximum edge weight of both of the components, they are merged and maximum edge

weight of the new component is updated. This way a disjoint set forest is obtained. Each disjoint set in the DSF corresponds to a connected component in the image.

4.2.3. Region merging

Initial segmentation results in an over-segmented image in most cases. Therefore, a subsequent region merging operation is required to obtain more meaningful components. To this end, we first construct a Region Adjacency Graph (RAG) to represent the initial over-segmented image. In this graph, each node represents a connected component, and each edge represents neighborhood relations between each pair of components; these are average color and velocity difference between the two components. We compute the dense optical flow in x and y directions between successive frames to obtain the velocity information. Additionally, each region is represented with a set of simple color and shape features:

- mean and variance of color values,
- color histogram,
- area,
- *A2P*: ratio of region area to region perimeter
- *A2MBR*: ratio of region area to area of the minimum bounding rectangle (MBR) of the region, and
- average region velocities in x and y directions.

We use the following heuristics derived from experimental observations on segmented region properties to decide on whether to merge two neighboring regions A and B.

- Do not merge A and B if *A2P* ratio of one of them is very small (e.g., less than 1) or aspect ratio is very large or very small and resulting component has a much lower *A2MBR* ratio. This is to avoid merging of thin, long regions to more compact regions in a perpendicular direction, hence favor compact regions.
- Merge A and B if the average color difference on the boundary between A and B is sufficiently small. This allows merging of an over segmented region with another region if there is a slowly varying gradient on the boundary due to illumination effects.
- Merge A and B if the distance between the color histogram, mean and standard deviation of the regions are small and average color and velocity difference on the boundary is not too large. This allows merging of regions with similar color. The velocity constraint is imposed to be able to differentiate two distinct objects with similar color based on velocity difference on the boundary.
- Merge A and B if they are both moving and they have velocities close to each other. This is to merge regions with different color belonging to the same object if the object is moving. An example would be a walking man wearing a white t-shirt and blue jeans.
- Finally, merge small components. Merge A and B if A is small and completely contained in B. Merge A and B if one of them is small and color and velocity difference on the boundary is minimum among all neighbors.

4.2.4. Face detection

We use the object detector of OpenCV library [40] based on the object detection algorithm initially proposed by Paul Viola [41]. This object detection algorithm works as follows. First, a classifier (namely a cascade of boosted classifiers working with Haar-like features) is trained with a few hundreds of sample views of a particular object (i.e., a face), called *positive examples* (e.g., face regions), and *negative examples* (e.g., non-face regions) that are scaled to the same size. After a classifier is trained, it can be applied to a region of interest (of the same size as used during the training) in an input image. The classifier outputs a 1 if the region is likely to contain the object (i.e., face), and 0 otherwise. To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily resized in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. Therefore, to find an object of an unknown size in the image, the scan procedure should be done several times at different scales. We use the pre-trained face models of OpenCV for face detection and find *frontal* faces with size greater than 20×20 pixels.

4.2.5. Saliency determination

Our saliency concept is somewhat different from how it is defined in literature. Rather than the regions that attract our attention at the first glance, we aim to extract objects that may be important in indexing and searching the video. Hence, our sense of saliency is broader, encompassing the saliency concept discussed in Section 3.3.

Different people may select different objects as salient on the same scene. However, some class of objects are almost always salient to everyone, such as a person in a news video. Therefore, we decide on the saliency of objects using the following simple heuristics.

- Faces are salient since they indicate the presence of people in the scene, and people are the most important actors especially in news videos.
- Moving objects in a scene are mostly salient. Examples are moving car, flying airplane, walking person.
- Objects that are in camera focus are probably salient.
- Visually conspicuous objects are usually salient.

To determine the objects in focus we need to estimate the camera focus. We observe that in-focus objects have higher contrast and sharper edges than out-of-focus objects which are usually blurred (see Fig. 2). Therefore, we can estimate the camera focus by computing the region

Fig. 2. In-focus objects have sharper edges and higher contrast than out-of-focus objects.

variance or entropy, which is also associated with variance in pixel values. We prefer computing the region color variance since it is much easier to compute than entropy. We compute the variance over the original, unsmoothed image so that the computed variance is not adversely affected by the smoothing operation in preprocessing. We use the following features while deciding on the saliency of a region.

- Detected faces are directly taken to be salient.
- Speed of the object is important since moving objects are more salient according to our assumptions.
- Region variance is important to detect in-focus objects.
- In-focus salient objects are located in the middle of the frames most of the time (e.g., a talking head centered in the frame). Therefore, position information may be helpful in such cases.
- Regions with aspect ratios too small or too large, regions with small *A2P*, *A2MBR* ratios (see Section 4.2.3), regions that are too large (probably background) or too small should be eliminated. These are to promote compact regions and eliminate very thin and long regions which are most probably not salient or due to segmentation inaccuracies. Therefore, we also use these shape features.
- To help differentiate visually conspicuous regions we use the color difference of a region from its neighbors and all other regions as features.
- Salient regions should be consistent, i.e., they should be detected in a specified minimum number of frames within the shot (e.g., 10% of the frames).

We train a Support Vector Machine (SVM) with linear kernel by selecting few hundreds of positive and negative salient region examples from segmented frames and representing each region by position, shape, color and motion features as described above. Then, saliency test for all regions is carried out using the trained classifier for each frame. Among the qualified regions, five regions with maximum score are selected provided that the score is above a threshold since we do not expect more than five salient objects in the scene based on our assumptions. Finally, when the salient object information is to be saved at the end of each shot, we perform a consistency check for each detected and tracked object and accept qualified objects as salient.

4.3. Tracking

Unless we track the detected objects, salient object detection is not of much use in our video database system in terms of required manual labeling work since otherwise the gain in manual labeling time will not improve significantly. Moreover, manual selection of objects would produce much more accurate results. Therefore, our primary aim is to detect and track salient objects so that manual labeling will be done only once for each object within the shot when they first appear. The system will give the same label to the objects with the same ID in the following frames, saving the labeler a huge amount of time in the ideal case in which all the objects can be detected and tracked accurately.

We take a feature-based approach to the tracking of salient regions. We try to find a one-to-one correspondence between salient regions of successive frames. We first eliminate some of the regions by using position and shape information, and then we find the similarity between the remaining regions using color histogram as feature. Our tracking algorithm is as follows.

1. Add the first salient regions to the list of tracked regions.
2. For each frame, first check if the tracked objects can be found in the current frame and if so mark them as *matched*.
3. For each detected salient region in the current frame that is not *matched* yet, try to find a matching region in the list of *unmatched* tracked regions. If a *match* cannot be found, add this region to the list of tracked regions as a new region.
4. Two matching regions should be close enough in successive frames within the limits of their velocities. They should also have similar shape (aspect ratio) and their areas should be close to each other.
5. The similarity is measured by Euclidean distance between the feature vectors. If the most similar region has a similarity larger (or distance smaller) than a threshold, the correspondence is established.

4.4. Object labeling and knowledge-base construction

After salient objects are detected, tracked and saved on a shot basis, they are manually labeled. The primary aim of the

object extraction tool is to decrease the manual work required in the construction of the video database. This is achieved by the tracking of the objects throughout the shot so that for each distinct object in the shot only one labeling is required. If all the objects appear in the first frame of the shot and all the objects are accurately tracked, then it is enough to label the first frame and let the system label the remaining frames automatically. This results in tremendous savings in time and effort. False positives are eliminated at the labeling stage. The user just skips the object if it is not salient. This brings a small overhead of pressing a key to pass to the next object. Tracking errors can be corrected in labeling stage to some extent. If a tracked object is lost in some of the frames and detected and tracked as a new object later, and if the user gives the same label to both objects then the error is corrected by the system considering the two objects to be the same. Finally, *Fact Extractor* computes the spatio-temporal relations between the labeled objects using their MBRs and inserts them into the database.

4.5. Implementation issues

We have implemented our system on Windows in C++ by making extensive use of the Open Source Computer Vision Library, OpenCV [40], for image processing. We also used the open source FFmpeg library [42] to decode the MPEG video streams. It takes about 4 s to completely process one frame (detection and tracking) on an Intel Celeron 1.3 GHz notebook computer.

5. Experimental results

In this section, we present results of experiments carried out on TRECVID 2006 news videos and evaluate the performance of the system.

5.1. Data set

We tested our system on videos from TRECVID 2006 video corpus, which consists of news videos from television broadcasts from different countries in different languages (English, Arabic, Chinese). The quality of videos varies. Most of the outdoor scenes are noisy. Dark, cluttered and highly dynamic scenes make them very challenging for an automatic object extraction system.

5.2. Shot boundary detection

We evaluated the performance of shot boundary detector on one of the news videos with 58,031 frames. We counted the number of correct detections, false detections and the number of missed shot boundaries, but did not discriminate the different kinds of transitions. Of 328 shot boundaries detected, recall is 94% and precision is 93%. The performance is satisfactory and are on the same scale as the latest results reported in TRECVID contest. Cuts are easy to detect since the inter frame distance is usually quite high. Gradual transitions are harder to detect, and most of the missed boundaries are composed of smooth gradual

Fig. 3. Face detection examples from TRECVID 2006 data set. There is one false detection in the last image.

transitions. Occasionally, some cuts are missed if the color distribution of successive frames are close, even though the transition is sharp. Using edge information additionally may help improve the performance in such cases.

5.3. Face detection

Fig. 3 shows face detection examples from TRECVID 2006 news videos. We measured the face detection performance as having 87% recall and 83% precision. While measuring the performance, we considered very similar frames only once since the detection accuracy does not change much if the frame does not change. We used 130 such frames from different videos and different shots. For example, in the first image in Fig. 3, the anchors are detected very accurately. If we were to consider all the frames of the anchor shots in the performance measurement, the detection performance would turn out to be much higher since a high proportion of news videos is composed of such frames.

5.4. Salient object detection and tracking

The most crucial part of salient object detection is the segmentation since it directly affects the quality of the detected object, saliency score and tracking. Fig. 4 shows segmentation examples of indoor and outdoor scenes from

Fig. 4. Segmentation examples of both indoor and outdoor scenes. From left to right: the original image, the result after initial segmentation, and the result after merging. Colors for the segmented components are randomly generated for each segmented image.

our test videos. The segmentation quality is higher in indoor (studio) scenes due to higher quality and less noise. Overall, the quality of segmentation is as good as the ones reported in literature, in which usually simpler video sequences (e.g., Children sequence, Foreman sequence, Miss America sequence, etc.) are used to demonstrate the segmentation performance. Segmentation in dark and noisy scenes is not successful. Currently, we do no use global motion compensation; therefore, when camera motion is large regions get merged resulting in low quality segmentation (under-segmentation).

Fig. 5 shows examples for detected salient regions. If the regions are successfully segmented, they are also classified correctly most of the time. In the examples shown, faces are detected as salient objects; we did not employ a face detector to detect them. The face detector is useful when

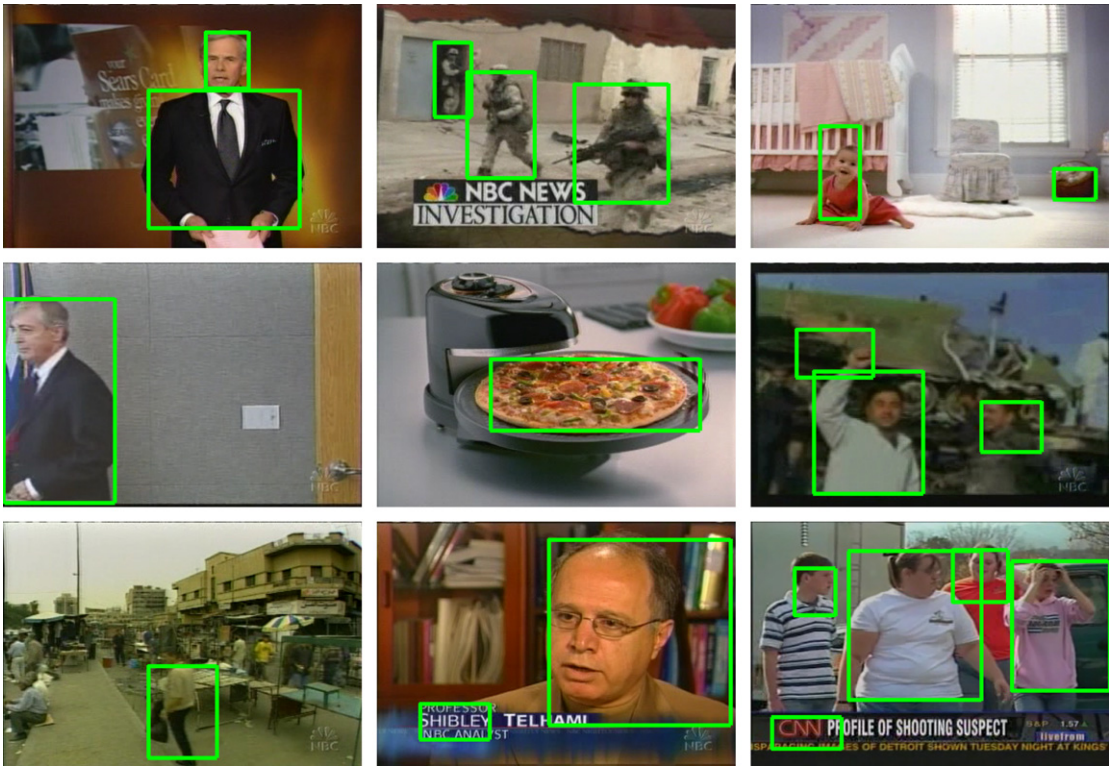

Fig. 5. Examples of salient regions detected in TRECVID 2006 news videos. In these examples, a face detector is not employed, the regions are obtained directly from segmentation.

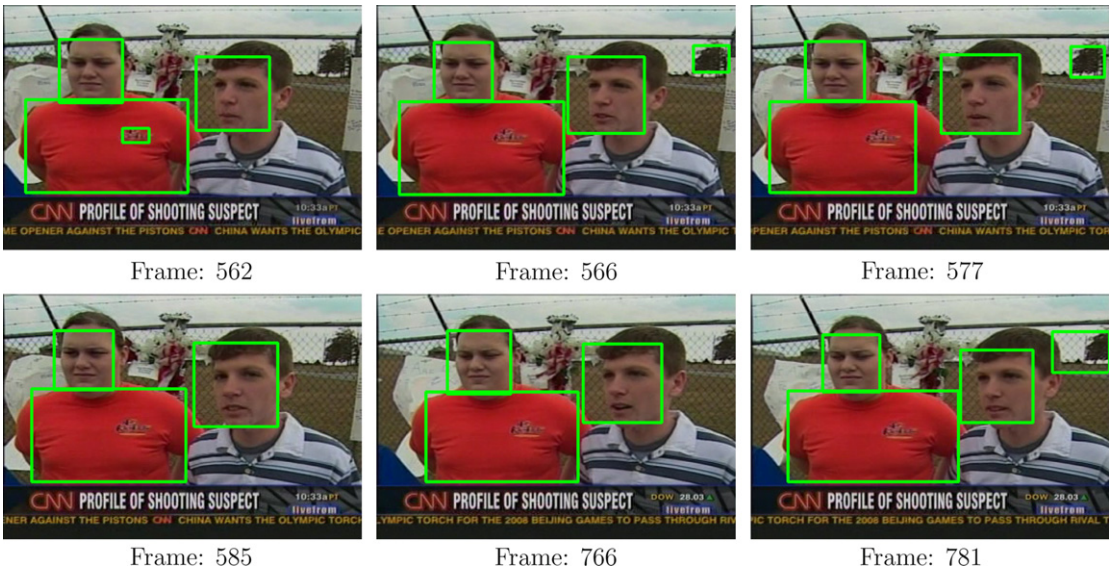

Fig. 6. An example of tracking salient regions within a shot from frame 562 to 781.

the segmentation quality is low, and hence, face regions cannot be segmented accurately. Fig. 6 shows an example of salient regions being detected and tracked successfully within a shot from frames 562 to 781.

We compared the salient object detection performance of our system with one of the leading saliency model (SM) approaches described in [32], using the author's own MATLAB implementation which is available on the web [43]. We used two test videos with 668 frames containing mostly outdoor scenes (Fig. 7). We run the SM tool on the videos frame by frame and retrieved the contours of the first five fixation points for evaluation. We run our system without face detection on the same videos. Fig. 7 shows examples of salient regions detected by both methods.

Fig. 7. Visual comparison of our salient object detection without face detection (right) with SM (left) [32]. On the right column, numbers within rectangles indicate the rank of saliency.

Visually, our approach outputs more meaningful regions in terms of human perception. Fig. 8 shows the precision–recall graph obtained on the same set of videos by visually evaluating the correctness of the detected regions in ranked order. The SM approach is somewhat better in precision at low and high recall rates. If we also employ a face detector in our system, the performance improves depending on the number and quality of face regions contained in the video frames.

The recall and precision values can be adjusted by tuning the thresholds for saliency detection. For instance, if we use a low threshold for consistency, recall will increase while precision will decrease. Low precision will result in larger amount of manual work to skip the false detections without giving a label; low recall will result in low system performance. After labeling, all false detections are eliminated, therefore, the precision in the resulting system becomes 100%.

5.5. Performance evaluation

In order to evaluate the effectiveness of our system in terms of reduction in labeling time, which is our primary goal, we measured time requirements for fully manual labeling and labeling after automatic salient object detection as summarized in Table 1. We manually labeled several frames in different shots and computed an average labeling time per frame as 25 s. Then, we estimated the total time requirement for a video segment by multiplying the total number of frames with the average labeling time per frame. For the automatic case, we run the system, with face detection capability, which detected, tracked and saved salient objects after which we labeled the saved objects as described in Section 4.4 since it does not take much time in this case. Table 1 shows a huge reduction of approximately 99% in labeling time. If we also consider the offline automatic processing time, the reduction is 87%.

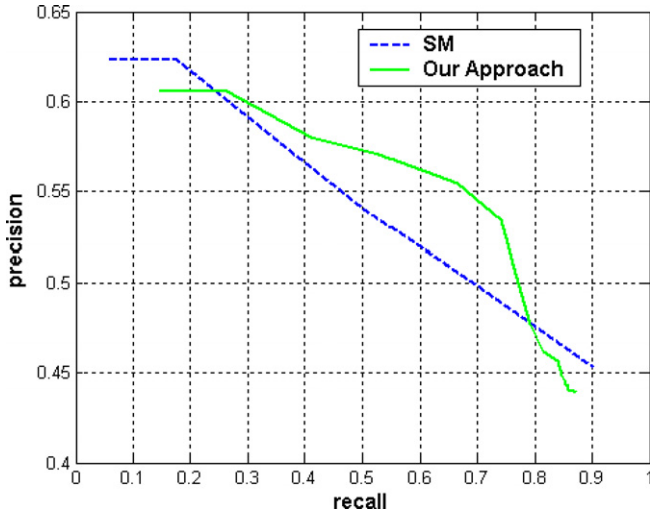

Fig. 8. Precision–recall graph for salient object detection, comparing our approach (no face detection) with SM [32].

Table 1
Effectiveness of the system in terms of reduction in labeling time for a video consisting of 10 shots and 1572 frames

Process	Total	Per frame
Automatic processing time	104 min	4 s
Fully manual labeling time	655 min	25 s
Labeling time after automatic processing	7 min	0.3 s

The systems uses our face detector, and performance of the system on the labeled data is 79% recall, 67% precision.

6. Conclusion and future work

In this paper, we have proposed an automatic salient object extraction tool, as a component of a video database system, *BilVideo*. The tool extracts salient objects and spatio-temporal relations among them from a video automatically in order to speed up the processing, labeling, and indexing of videos for spatio-temporal querying. The proposed tool greatly reduces the time and user effort required for video indexing. To our knowledge, our framework is the first attempt to address this problem.

The performance of the tool can be improved in several ways. Global motion compensation should be supported to account for camera motion. In addition to the currently used easily computable, simple feature set other features proposed in the literature should also be experimented for saliency detection to improve the accuracy. Finally, we are planning to automate the whole process of detection, tracking and labeling to completely eliminate human intervention so that our video database system, *BilVideo*, can accommodate huge video archives.

Acknowledgments

This work is supported by European Union 6th Framework Program under Grant No. FP6-507752 (MUSCLE Network of Excellence Project) and TÜBİTAK under Grant No. EEEAG-105E065 (New Information Society Technologies for Turkey). We are grateful to Kirsten Ward for proofreading the paper.

References

[1] S. Marcus, V.S. Subrahmanian, Foundations of multimedia database systems, Journal of the ACM 43 (3) (1996) 474–523.
[2] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, Query by image and video content: the QBIC system, IEEE Computer 28 (9) (1995) 23–32.
[3] J.R. Smith, S.F. Chang, VisualSEEk: a fully automated content-based image query system, in: ACM Multimedia, 1996, pp. 87–98. Available from: <citeseer.ist.psu.edu/smith96visualseek.html>.
[4] S.F. Chang, W. Chen, H.J. Meng, H. Sundaram, D. Zhong, VideoQ: an automated content based video search system using visual cues, in: ACM Multimedia, 1997, pp. 313–324. Available from: <citeseer.ist.psu.edu/chang97videoq.html>.

[5] E. Saykol, U. Güdükbay, Özgür Ulusoy, A histogram-based approach for object-based query-by-shape-and-color in image and video databases, Image and Vision Computing 23 (13) (2005) 170–1180.
[6] J. Ashley, R. Barber, M. Flickner, J. Hafner, D. Lee, W. Niblack, D. Petkovic, Automatic and semiautomatic methods for image annotation and retrieval in query by image content (QBIC), in: Wayne Niblack, Ramesh C. Jain, (Eds.), Proceedings of the SPIE Storage and Retrieval for Image and Video Databases III, vol. 2420, March 1995, pp. 24–35.
[7] O. Marques, N. Barman, Semi-automatic semantic annotation of images using machine learning techniques, Lecture Notes in Computer Science, vol. 2870, Springer-Verlag, 2003, pp. 550–565.
[8] C. Gu, M.-C. Lee, Semiautomatic segmentation and tracking of semantic video objects, IEEE Transactions on Circuits and Systems for Video Technology 8 (5) (1998) 572–584.
[9] M.E. Dönderler, E. Şaykol, U. Arslan, Özgür Ulusoy, U. Güdükbay, Bilvideo: design and implementation of a video database management system, Multimedia Tools and Applications 27 (2005) 79–104.
[10] M.E. Dönderler, Özgür Ulusoy, U. Güdükbay, A rule-based video database system architecture, Information Sciences 143 (2002) 13–45.
[11] J.S. Boreczky, L.A. Rowe, Comparison of video shot boundary detection techniques, in: Proceedings of the SPIE Storage and Retrieval for Image and Video Databases IV, vol. 2670, San Jose, California, USA, 1996, pp. 170–179.
[12] R. Lienhart, Reliable transition detection in videos: a survey and practitioner's guide, International Journal of Image and Graphics (IJIG) 1 (3) (2001) 469–486.
[13] A.F. Smeaton, P. Over, W. Kraaij, Evaluation campaigns and TRECVid, in: MIR'06: Proceedings of the Eighth ACM International Workshop on Multimedia Information Retrieval, New York, NY, USA, 2006, pp. 321–330.
[14] F. Porikli, Y. Wang, Automatic video object segmentation using volume growing and hierarchical clustering, Journal of Applied Signal Processing, special issue on Object-Based and Semantic Image and Video Analysis, July 2004. Available from: <http://www.porikli.com/pdfs/jasp2004-porikli.pdf>.
[15] Y. Deng, B. Manjunath, Unsupervised segmentation of color–texture regions in images and video, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (8) (2001) 800–810. Available from: <http://vision.ece.ucsb.edu/publications/01PAMIJseg.pdf>.
[16] 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.
[17] W. Ma, B. Manjunath, Edgeflow: a technique for boundary detection and segmentation, IEEE Transactions on Image Processing 9 (8) (2000) 1375–1388. Available from: <http://vision.ece.ucsb.edu/publications/00edgeflow.pdf>.
[18] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905. Available from: <citeseer.ist.psu.edu/article/shi97normalized.html>.
[19] P.F. Felzenszwalb, D. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision 59 (2) (2004) 167–181. Available from: <http://people.cs.uchicago.edu/pff/papers/seg-ijcv.pdf>.
[20] D. Zhang, G. Lu, Segmentation of moving objects in image sequences: a review, Circuits, Systems, and Signal Processing 20 (2) (2001) 143–183.
[21] F. Moscheni, S. Bhattacharjee, M. Kunt, Spatio-temporal segmentation based on region merging, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (9) (1998) 897–915.
[22] L. Li, W. Huang, I.Y. Gu, Q. Tian, Foreground object detection from videos containing complex background, in: Proceedings of the Eleventh ACM International Conference on Multimedia, New York, NY, USA, 2003, pp. 2–10.
[23] S.M. Desa, Q.A. Salih, Image subtraction for real time moving object extraction, in: International Conference on Computer Graphics, Imaging and Visualization, 2004, pp. 41–45.
[24] K. Ryan, A. Amer, L. Gagnon, Video object segmentation based on object enhancement and region merging, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), 2006, pp. 273–276.
[25] J. Hsieh, J. Lee, Video object segmentation using kernel-based models and spatiotemporal similarity, in: Proceedings of IEEE International Conference on Image Processing (ICIP), 2006, pp. 1821–1824.
[26] G. Zhang, W. Zhu, Automatic video object segmentation by integrating object registration and background constructing technology, in: Proceedings of the International Conference on Communications, Circuits and Systems, June 2006, pp. 437–441.
[27] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–577.
[28] A. Cavallaro, O. Steiger, T. Ebrahimi, Multiple video object tracking in complex scenes, in: Proceedings of the Tenth ACM International Conference on Multimedia, New York, NY, USA, 2002, pp. 523–532.
[29] C. Kim, J. Hwang, Fast and automatic video object segmentation and tracking for content-based applications, IEEE Transactions on Circuits and Systems for Video Technology 12 (2) (2002) 122–129.
[30] L. Xu, J. Landabaso, B. Lei, Segmentation and tracking of multiple moving objects for intelligent video analysis, BT Technology Journal 22 (3) (2004) 140–150.
[31] J. Shao, R. Chellappa, F. Porikli, Shape-regulated particle filtering for tracking non-rigid objects, in: Proceedings of the International Conference on Image Processing (ICIP), August 2006, pp. 2813–2816. Available from: <http://www.porikli.com/pdfs/IP2006-shao.pdf>.
[32] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (11) (1998) 1254–1259.
[33] Y. Ma, H. Zhang, Contrast-based image attention analysis by using fuzzy growing, in: Proceedings of the Eleventh ACM International Conference on Multimedia, 2003, pp. 374–381.
[34] S. Kwak, B. Ko, H. Byun, Automatic salient-object extraction using the contrast map and salient points, in: Advances in Multimedia Information Processing, Lecture Notes in Computer Science (LNCS), vol. 3332, 2004, pp. 138–145.
[35] U. Rutishauser, D. Walther, C. Koch, P. Perona, Is bottom-up attention useful for object recognition?, in: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition (CVPR), vol. 2, July 2004, pp. 37–44.
[36] F. Ge, S. Wang, T. Liu, Image-segmentation evaluation from the perspective of salient object extraction, in: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition (CVPR), vol. I, 2006, pp. 1146–1153.
[37] B. Chalmond, B. Francesconi, S. Herbin, Using hidden scale for salient object detection, IEEE Transactions on Image Processing 15 (9) (2006) 2644–2656.
[38] T. Liu, J. Sun, N.N. Zheng, X. Tang, H. Shum, Learning to detect a salient object, in: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition (CVPR), June 2007, pp. 1–8.
[39] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: Proceedings of IEEE International Conference on Computer Vision, Bombay, India, 1998, pp. 839–846.
[40] Open Source Computer Vision Library. Available from:<http://opencvlibrary.sourceforge.net>.
[41] P.A. Viola, M.J. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition (CVPR), vol. 1, 2001, pp. I-511–I-518.
[42] FFmpeg Library. Available from: <http://ffmpeg.mplayerhq.hu>.
[43] Saliency Toolbox. Available from: <http://www.saliencytoolbox.net>.