2013-12-18

Spatio-temporal Visual Information Analysis for Moving Object Detection and Retrieval in Video Sequences

Dianting Liu
diantingliu@gmail.com

UNIVERSITY OF MIAMI

SPATIO-TEMPORAL VISUAL INFORMATION ANALYSIS FOR MOVING OBJECT DETECTION AND RETRIEVAL IN VIDEO SEQUENCES

By

Dianting Liu

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

December 2013

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

SPATIO-TEMPORAL VISUAL INFORMATION ANALYSIS FOR MOVING OBJECT DETECTION AND RETRIEVAL IN VIDEO SEQUENCES

Dianting Liu

Approved:

____
Mei-Ling Shyu, Ph.D.
Professor of Electrical and Computer Engineering

____
Xiaodong Cai, Ph.D.
Associate Professor of Electrical and Computer Engineering

____
Saman Aliari Zonouz, Ph.D.
Assistant Professor of Electrical and Computer Engineering

____
Nigel John, Ph.D.
Lecturer of Electrical and Computer Engineering

____
Shu-Ching Chen, Ph.D.
Professor of School of Computing and Information Sciences
Florida International University

____
M. Brian Blake, Ph.D.
Dean of the Graduate School

LIU, DIANTING (Ph.D., Electrical and Computer Engineering)
Spatio-temporal Visual Information Analysis for Moving Object Detection and Retrieval in Video Sequences (December 2013)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Mei-Ling Shyu.
No. of pages in text. (159)

The development of the Internet makes the number of online videos increase dramatically, which brings new demands to the video search engines for automatic retrieval and classification. We propose an unsupervised moving object detection and retrieval framework by exploiting and analyzing spatio-temporal visual information in the video sequences. The motivation is to use visual content information to estimate the locations of the moving objects in the spatio-temporal domain. Compared with the existing approaches, our proposed detection algorithm is unsupervised. It does not need to train models for specific objects. Furthermore, it is suitable for the detection of unknown objects. Therefore, after object detection, the object-level features can be extracted for video retrieval.

The proposed moving object detection algorithm consists of two layers: global motion estimation layer and local motion estimation layer. The two layers explore and estimate motion information from different scopes in the spatio-temporal domain. The global motion estimation layer uses a temporal-centered estimation method to obtain a preliminary region of motion. Specially, it analyzes the motion in the temporal domain by using our proposed novel motion representation method called the weighted histogram of

Harris3D volume which combines the optical flow field and Harris3D corner detector to obtain a good spatio-temporal estimation in the video sequences. The idea is motivated by taking advantages of the two sources of motion knowledge identified by different methods to get a complementary motion data to be kept in the new motion representation. The method, considering integrated motion information, works well with the dynamic background and camera motion, and demonstrates the advantages of integrating multiple spatio-temporal cues in the proposed framework. In addition, a center-surround coherency evaluation model is proposed to compute the local motion saliency and weight the spatio-temporal motion to find the region of a moving object by the integral density algorithm. The global motion estimation layer passes the preliminary region of motion to the local motion estimation layer. The latter uses a spatial-centered estimation method to integrate visual information spatially in adjacent frames to obtain the region of the moving object. The visual information in the frame is analyzed to find visual key locations which are defined as the maxima and minima of the result of the difference-of-Gaussian function. A motion map of adjacent frames is obtained to represent the temporal information from the differences of the outcomes from the simultaneous partition and class parameter estimation (SPCPE) framework. The motion map filters visual key locations into key motion locations (KMLs) where the existence of the moving object is implied. The integral density method is employed to find the region with the highest density of KMLs as the moving object. The features extracted from the motion region are used to train the global Gaussian mixture models for the video representation. The representation significantly reduces the classification model training time in comparison to the time needed when the whole feature sets are used. It also achieves better classification performance. When combined with the information of scenes, the performance is further enhanced.

Besides the proposed spatio-temporal object detection work, two other related methods are also proposed since they play subsidiary roles in the detection model. One is the innovative key frame detection method which selects representative frames as the key frames to provide key locations in the spatial-centered estimation method. By analyzing the visual differences between frames and utilizing the clustering technique, a set of key frame candidates is first selected at the shot level, and then the information within a video shot and between video shots is used to adaptively filter the candidate set to generate the final set of key frames for spatial motion analysis. Another new method is to segment and track two objects under occlusion situations, which is useful in multiple object detection scenarios.

To my family

Acknowledgments

My sincere appreciation and gratitude first go to Dr. Mei-Ling Shyu, my dissertation supervisor and chairman of the committee, for her continual suggestions, guidance, and support on my Ph.D. career. I would like to extend my best appreciation to Dr. Shu-Ching Chen, who provides me precious research experience on Florida Public Hurricane Loss Model (FPHLM) at Florida International University (FIU).

In addition, I appreciate my committee members Dr. Xiaodong Cai, Dr. Saman Aliari Zonouz, and Dr. Nigel John of the Department of Electrical and Computer Engineering (ECE) at University of Miami (UM) for their helpful suggestions on my dissertation proposal and defense. Moreover, I would like to thank all group members in Data Mining, Database and Multimedia Research Group at UM and Distributed Multimedia Information Systems Laboratory at FIU, for their encouragement, considerations, and assistance in past several years. Furthermore, my thanks go to National Oceanic and Atmospheric Administration and Department of ECE at UM for the financial support throughout my Ph.D. work. I thank Graduate Activity Fee Allocation Committee (GAFAC) for providing travel funds and Center for Computational Science (CCS) for using the resources of scientific computing.

I would like to thank all my beloved friends who support and love me throughout the lifetime. I want to send the mostly sincere gratitude to my whole family and Anhui Huang for their love all the way, without reserve, without recall.

DIANTING LIU

University of Miami

December 2013

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the fast development of the Internet, more and more people search information on the Internet. Consequently, many text-based search engines appear on the Internet for topic and event search tasks [6]. Although texts share certain correlation information with content data, it is not easy to search a multimedia content because multimedia data, as opposed to texts, needs more pre-processing steps to yield indices relevant for the query [7, 8]. With the amount of online multimedia data increasing at an explosive speed, more challenges on data searching, retrieval, browsing and categorization arise. These challenges motivate many researchers to devote their efforts into the multimedia semantic retrieval area [9, 10, 11]. Especially, with the rapid advances of the Internet and Web 2.0, the traditional ways of manually assigning a set of labels to a record, storing it, and matching the stored label with a query obviously are not feasible and effective for large multimedia databases. The successor methods, called content-based video retrieval approaches, are developed to quickly and automatically identify the semantic concepts and annotate the video sequences [12, 13, 14, 15, 16].

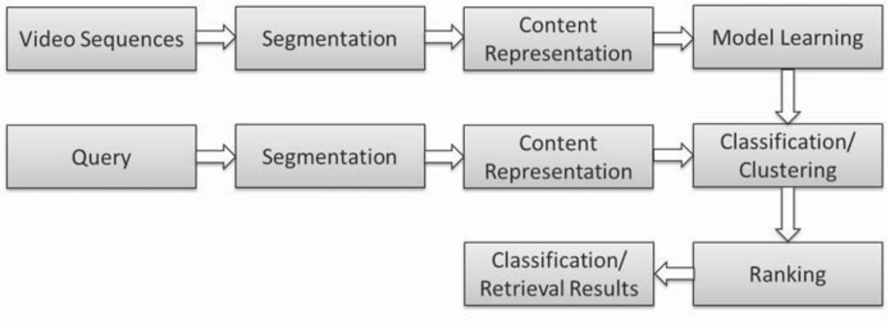
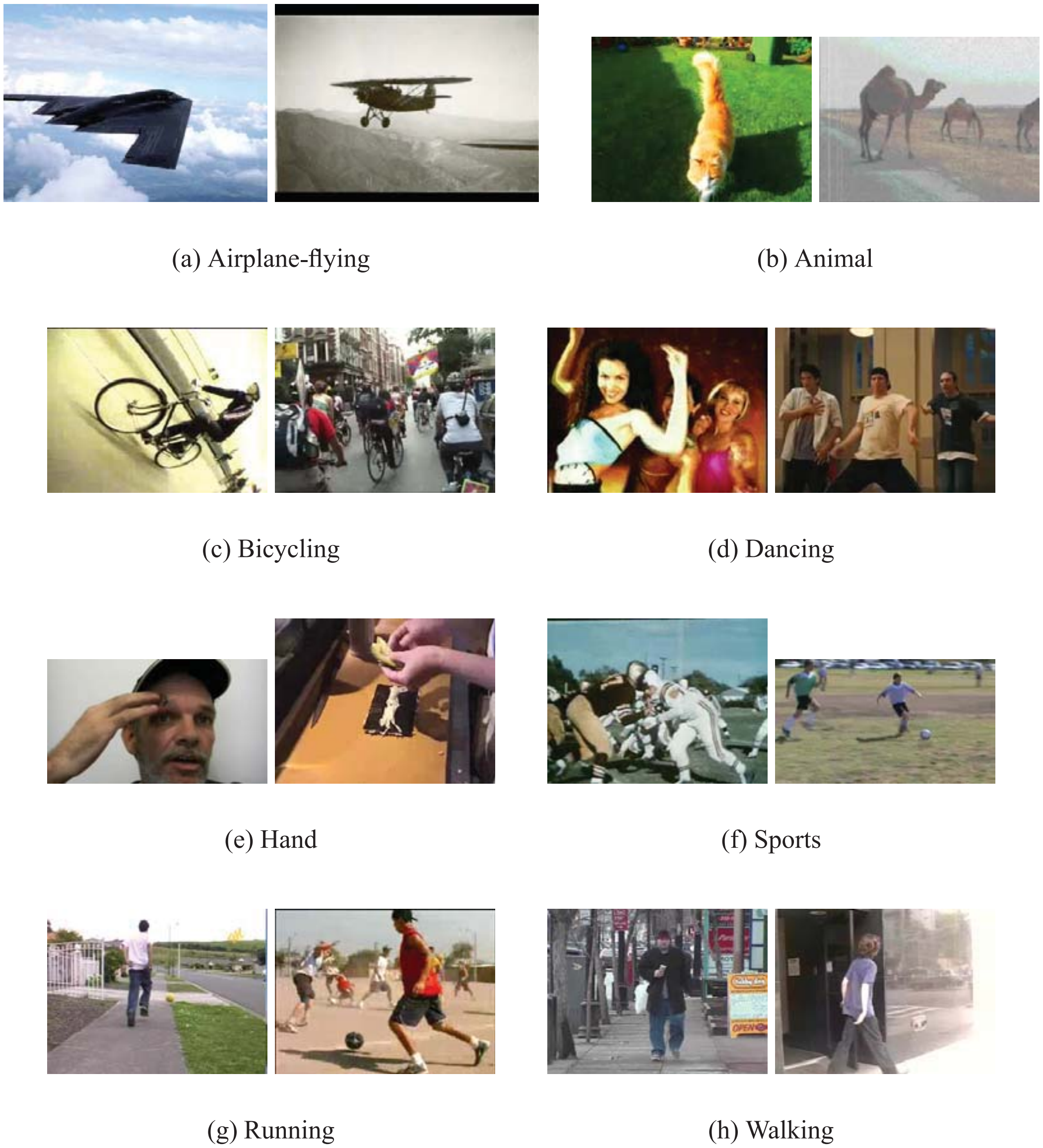


Figure 1.1: An example content-based video classification/retrieval system

1.1 Challenges and Motivations

An example content-based video classification/retrieval system consists of several processing steps [17]. As shown in Figure 1.1, the content representation is a key step in a content-based video classification/retrieval system. A discriminate content representation would benefit model learning and result in good classification/retrieval performance. To obtain more class related information, automatic object detection algorithms are utilized to segment video frames into a set of semantic regions and each region corresponds to an object that is meaningful to the human vision system, such as a dog and a tree. Then those features extracted from the objects are used for video content representation. Compared with a content-based image retrieval (CBIR) system, the available information and the challenges of object detection in video data are different. For example, the temporal information in video sequences enables us to utilize the moving object-level information for moving object detection.

After years of development, many object detection models have been proposed with reasonably good performance in videos captured under controlled backgrounds. Nevertheless, little progress is achieved toward model robustness when dealing with videos with uncontrolled backgrounds, such as videos recorded by an amateur using a hand-



(a) Airplane-flying

(b) Animal

(c) Bicycling

(d) Dancing

(e) Hand

(f) Sports

(g) Running

(h) Walking

Figure 1.2: Snap-shots extracted from video clips recorded in an uncontrolled condition of eight concepts from the TRECVID 2010 video collection

held camera containing significant camera motion, background clutter, and changes in object appearance, scale, and illumination conditions. Figure 1.2 shows sample snapshots extracted from video clips recorded in an uncontrolled condition. These videos with uncontrolled backgrounds pose lots of challenges for multimedia retrieval over the Internet. This calls for the development of more advanced techniques for rapid processing and summarization. The drawbacks of most of the existing techniques include the following three requirements:

- static cameras or approximate compensation of camera motion;
- foreground objects that move in a consistent direction or have faster variations in appearance than the background;
- explicit background models [18].

These requirements are mostly unrealistic and particularly questionable when an ego-motion happens, e.g., a camera that tracks a moving object in a manner such that the latter has a very small optical flow, or the background is dynamic. In addition, background learning requires either a training set of the “background-only” images [19] or batch processing (e.g., median filtering [20]) of a large number of video frames. The latter must be repeated for each scene and is difficult for dynamic scenes where the background changes continuously.

Psychological studies find that a human vision system perceives external features separately [21] and is sensitive to the difference between the target region and its neighborhood. Such kind of high contrast is more likely to attract human’s first sight than their surrounding neighbors [22]. Extensive psychophysics experiments have shown that these mechanisms can be driven by a variety of features, including intensity, color, orientation, or motion, and local feature contrast plays a predominant role in the perception of saliency. Neurophysiological experiments on primates have also shown that neurons in the middle temporal (MT) visual area compute local motion contrast with center-surround mechanisms. In fact, it has been hypothesized that such neurons underlie the perception of motion pop-out and figure-ground segmentation [23]. The center-surround saliency mechanisms of biological systems support the idea of motion region estimation on measurements of local motion contrast. There is no need for training samples or pre-build a "global background model" for the testing instances, which is one of the advantages of the proposed framework. Instead, a motion region can be efficiently calculated using solely local motion information and could immediately adapt to different kinds of unknown scenes. Also, using local motion contrast could make the model robust to camera motion and dynamic background.

Besides extracting motion features from the temporal domain, key frames provide important spatial information for object detection and recognition. How to select informative key frames to represent the video sequences is another hot research topic, since key frame extraction has broad applications in multimedia area. we can simply use the sampling technique to select key frames from video sequences (e.g., uniform sampling). However, the sampling approaches suffer from seriously problems due to the characteristics of non-homogeneous content distribution of the video visual information. For example, many frames within a shot have very visually similar content, and thus content redundancies widely exist in the sampling results. Many key frame extraction algorithms have been proposed and developed to handle the drawback of sampling approaches. One category of the algorithms generates key frames when the content change exceeds a certain threshold [24, 25, 26]. Another category of methods is called coverage-based approaches in [27, 28, 29, 30], which aim to get a small number of key frames by maximizing each key frame's coverage towards adjacent frames. Furthermore, key frames could be extracted based on clips [31, 32] or shots [33, 34, 35, 36]. Compared with motion information, visual content in shot/video is described more reliably by color features. Mendi and Bayrak [37] created saliency maps based on color and luminance features in video frames, and a similarity between frames was calculated by using a new signal fidelity measurement called S-SSIM. Frames with the highest S-SSIM in each shot were extracted as key frames. In [38], coarse regions were first detected, and then interesting points in these detected regions served as a basis to compare the similarity between frames at the shot level. Later, one key frame is extracted from each shot. Although [37] and [38] reported good experimental results, they still faced the same problem that many frames containing important visual content might not be extracted as key frames, since one key frame per shot usually is insufficient to represent the shot content. In order to completely summarize videos, more key frames need to be extracted from each shot; otherwise, the summarization quality represented by key frames would be compromised. On the other hand, the incensement of the number of frames extracted from each shot will increase the redundancy to the final set of key frames. Therefore, the issue to balance quantity and quality in key frame extraction is the major concern in our proposed work.

1.2 Contributions and Limitations

In this section, the contributions of the proposed methods are reviewed and outlined in order to give the readers a brief understanding of the highlights of the proposed framework. On the other hand, the proposed framework has a few limitations from various perspectives, providing a lot of spaces for the improvement of the current framework in the future. We summarize the contributions and limitations below.

Contributions of the proposed framework are:

1. A two-layer detection framework is proposed for moving objects in an unsupervised manner. The framework achieves good performance in realistic videos having camera motion and unconstrained environment. The new framework estimates the motion of a video in a two-layer coarse-to-fine approach in spatio-temporal domain. The first layer (global motion estimation layer) employs the idea of temporal-centered estimation to preliminarily analyze motion and gives the results to the second layer (local motion estimation layer). In the second layer, the spatial-centered estimation method is utilized to capture local motion information and find the regions of moving objects in the video sequence.

2. The temporal-centered motion estimation method first extracts Harris3D corner information from video streams directly. The spatio-temporal information is then combined with the optical flow field to represent the motion content by a weighted histogram of Harris3D volume. The Harris3D corner information and optical flow are two different sources of motion information obtained from different methods. The combination achieves complementary motion cues and keeps them in the new motion representation which weakens the influence from the camera motion and background clutter. To the best of our knowledge, not much work has been reported using similar methods on the moving object detection under uncontrolled background in an unsupervised way.

3. The spatial-centered estimation method defines a motion map to capture temporal information between adjacent frames. This method can simply and quickly get the pixels having significant visual changes which imply the moving foreground. On the other hand, key motion locations (KMLs) are selected via the motion map

from the set of key locations which are obtained as maxima and minima of the result of the difference of Gaussians function. The region of a moving object is defined as the area with the highest density of KMLs, which is obtained by the proposed integral density method since it allows the fast implementation of the box type convolution filters.

4. An innovative key frame extraction approach is proposed with an attempt to achieve a balance between the quantity of key frames for summarizing the shots in a video sequence and the quality of key frames to represent the whole video. The new method identifies transitive regions and informative regions by analyzing the differences between consecutive frames at the shot level and proposes a modified clustering technique that is utilized as the key frame extractor to select a set of key frame candidates (KFCs) in informative regions, while transitive regions are not used for key frame extraction. In addition, the method integrates the frame information within a video shot and between video shots to filter redundant KFCs to generate a better set of key frames.

5. A general solution of splitting overlapped objects in video sequences is proposed. Compared with the previous methods, the new method can not only effectively partition the objects with similar sizes, but also be able to process the objects with a large variety on contour. This work can be viewed as an object tracking method under an occlusion situation, which can later be integrated into the spatio-temporal moving object detection model to process the case of multiple objects.

The limitations of the proposed framework are listed as below.

1. The spatio-temporal detection work is triggered by the motion contrast, so it is not suitable for processing videos with only static objects and scenes. The

static object and scene segmentation issue is usually solved by converting two-dimensional (2D) videos or multiple images into three-dimensional (3D) models, which consists of camera calibration and depth map determination. If the visual change between frames happens only due to the depth dimension, the results of the proposed methods are not ideal.

2. The interactive information between objects in videos has not been fully exploited. This kind of information will be useful for object detection or human activity recognition. The proposed framework can be later improved by adding an object trajectory analysis component to extract the interactive information of objects for semantic retrieval. The information of previous frames could also be involved for helping object detection and recognition in subsequent frames.

3. The proposed detection model only returns one motion region with the biggest motion contrast to the scene at one time. For scenarios having more than one moving object in the scene, the less active ones are ignored in the model. Though the method could find the location of a moving object, but the size and shape of the bounding box of the detected object has to be improved to fit various types of objects in the future work.

4. The framework of key frame detection currently is motivated by the visual change between frames. It is not a motion-driven model. If the model can involve more foreground or moving object information, it would better summarize the videos. On the other hand, the object-driven key frame detection method may provide high quality key frames to the spatio-temporal moving object detection model to improve the latter's performance.

1.3 Outline of the Work

The whole proposal is organized as follows. Chapter 2 reviews the important and related studies in the areas of moving object detection and key frame extraction. The advantages and limitations of the peer approaches are analyzed and discussed. Some related techniques employed as prior knowledge in our framework are briefly presented. Chapter 3 overviews the proposed solutions of moving object detection and retrieval. In addition, two methods used in the proposed framework are presented. One is an unsupervised key frame detection approach using within and between shot visual information. Another one addresses the issue of segmenting objects under occlusion situations in the video stream. In Chapter 4, the temporal-centered and spatial-centered estimation methods are presented for detecting moving objects in a non-static background. Next, a center-surround motion coherency evaluation model is discussed to enhance the detection work by using the proposed integral density approach. Furthermore, a two-layer moving object detection model is proposed to integrate the temporal-centered and spatial-centered methods. The framework is verified in terms of detection and recognition accuracy. Chapter 5 concludes the proposed framework and shows the future work.

Chapter 2

Literature Review and Related Techniques

In this chapter, we intensively review the literature with the related work. Section 2.1 focuses on discussing representative techniques of moving object detection in video sequences. Section 2.2 discusses the recent important studies on key frame detections. Several related key techniques employed in the proposed framework are presented in section 2.3 for a better understanding of our framework.

2.1 Moving Object Detection Algorithms

In video sequences, the action of objects will dominate the frame and human perceptual reactions will mainly focus on motion contrast regardless of visual texture in the scene. Several researchers have extended the study from the spatial attention to the temporal domain where prominent motion plays an important role. Chen *et al.* [39] proposed a backtrack-chain-updation split algorithm that can distinguish two separate objects that were overlapped previously. It found the split objects in the current frame and used the information to update the previous frames in a backtrack-chain manner. Thus, the algorithm could provide more accurate temporal and spatial information of the semantic objects for video indexing. In [5], the authors proposed a spatio-temporal video attention detection technique for detecting the attended regions that correspond to both interesting objects and actions in video sequences. The presented temporal attention model utilized the interest point correspondences (instead of the traditional dense optical fields) and the geometric transformations between images. Motion contrast was estimated by applying RANSAC (RANdom SAmple Consensus) on point correspondences in the scene. Obviously, the performance of the temporal attention model is greatly influenced by the results of point correspondences.

A batch of action detection and recognition models have been proposed and achieved good performance in videos captured under controlled backgrounds [1, 40, 41]. Nevertheless, more progresses toward model robustness are expected in order to handle the complexities of unconstrained backgrounds, such as videos recoded by an amateur using a hand-held camera containing significant camera motion, background clutter, and changes in object appearance, scale, and illumination conditions (as shown in Figure 2.1). These uncontrolled videos are the major challenges to the multimedia retrieval engines on the Internet, which calls for rapid summarization and processing algorithms.

One related work is by Liu *et al.* on recognizing actions from videos "in the wild" [1]. They estimated the centroid of the region of action by using the mean of the coordinates of the interest points. Dimensions of the region are calculated by the second central moments of the corresponding centroid. This strategy can obtain good results when the interest points are mainly located on the action, but it would fail when the background is non-static since it contributes to a lot of interest points. Ikizler-Cinbis *et al*. [40] estimated the location(s) of the person(s) by using the human detector proposed by Felzenswalb *et al*. [4]. To fill the gap in which the person detector did not fire due to the motion blur and pose variations, the mean-shift tracking method was used to locate the person in every frame [42]. The work considers any moving region as a "candidate object", and then finds the associated tracks and the corresponding features

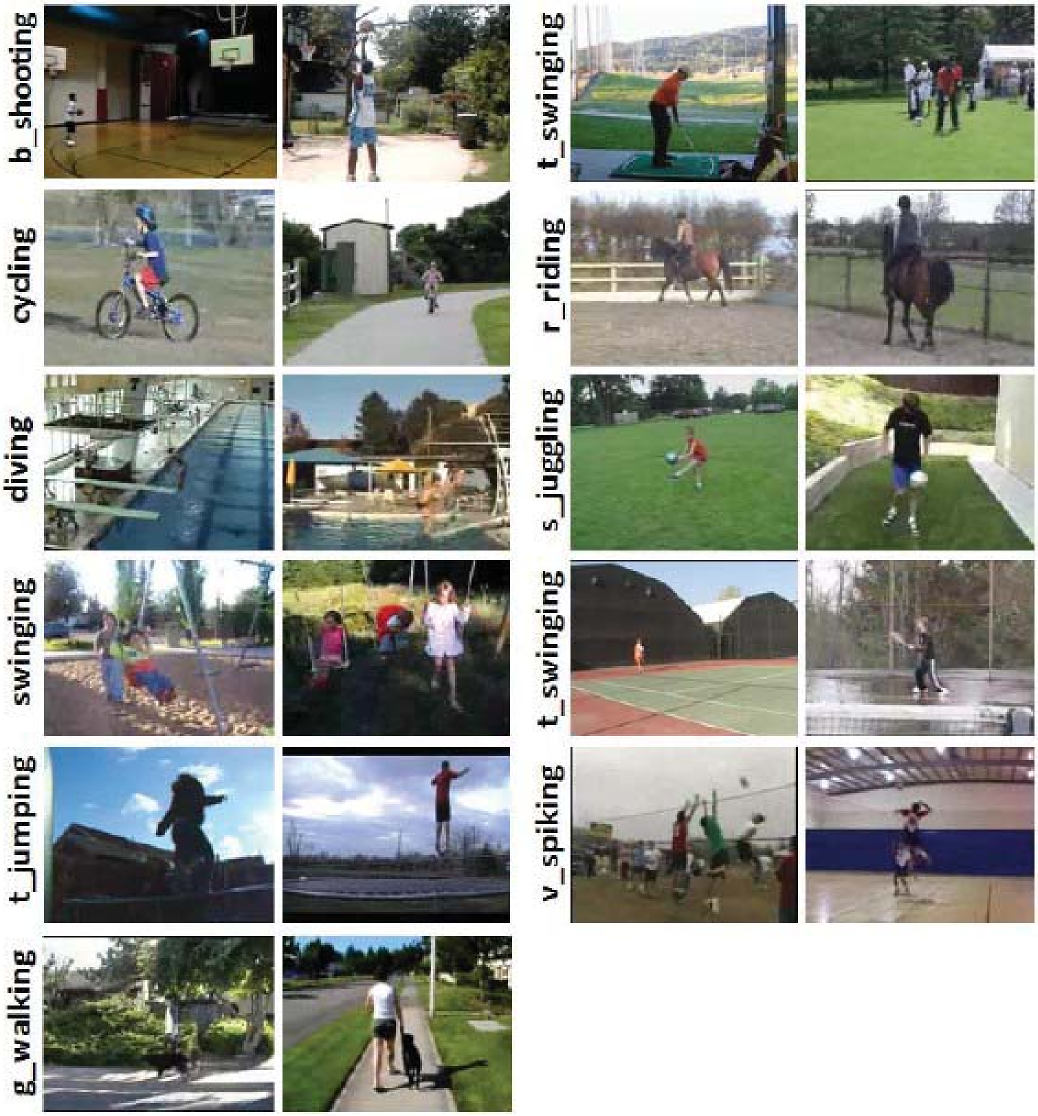


Figure 2.1: Examples of UCF Youtube action (UCF11) data set with approximately 1,168 videos in 11 categories [1]

from each track. The approach, to some degree, was able to capture the human and object features in the video. However, from the illustrated examples shown in the paper, the detected regions of objects inevitable include noise from the region of persons and background. The paper did not provide an effective solution to solve the issue. Optical flow is utilized by Reddy *et al*. [41] to give a rough estimate of the velocity at each pixel given two consecutive frames. A threshold on the magnitude of the optical flow was then applied to decide if the pixel is moving or stationary. The stationary pixels are regarded as background, while the moving pixels are viewed as the region of motion. This method performs well in videos with static scenes, but the strategy fails in the realistic videos with the unconstrained background.

In many natural frames, objects seldom laid out in well-separated poses as they often, more or less, overlapped on top of each other. Wittenberg *et al.* [43] first clustered the neighboring pixels into several regions, yielding a full segmentation of an image, and then combined these regions to objects that carried a semantic meaning. A pixel in an image may be affiliated to one region only, but a region can be part of more than one object. In this way, ambiguities occurred due to overlaps can be resolved on a semantic level. Such an approach was applied to medical images containing overlapping cervical cells, which achieved good results. In [44], the authors presented a new snake algorithm extending conventional snake algorithms by utilizing a pair of stereo images. The authors defined a unique energy function in the disparity space enabling successful boundary detection of the objects even when those objects were overlapped one another and the background was cluttered. An example was presented to demonstrate a successful result of this stereo-snake algorithm for detecting an object out of a complex image, though a set of interested points (including those objects to be segmented) needs to be manually pre-selected. Another novel marker extraction method was proposed to extract those markers labeling the target fruit and the background [45]. Based on this marker detection method, a new marker-controlled watershed transform algorithm was developed for accurate contour extraction of the target fruit. The face validity of the segmentation algorithm was tested with a set of grape images, and the segmentation results were overlaid onto the original images for visual inspection. Quantitative comparison was conducted and it showed that the segmentation algorithm can obtain good spatial segmentation results.

With the increasing amounts of digital video data becoming available in the Web, more and more attentions have been paid to content-based video processing approaches that can automatically identify the semantic concepts in a video [12, 13, 14, 15, 16]. To achieve this, object detection is a crucial step and thus special attentions are devoted to segmenting a video frame into a set of semantic regions, each of which corresponds to an object that is meaningful to human viewers, such as a car, a person, and a tree. The extra temporal dimension of the video allows the motion of the camera or the scene to be used in processing. In [46], a region-based spatio-temporal Markov random field (STMRF) model was proposed to segment moving objects semantically and the motion validation was used to detect occluded objects. The STMRF model combined the segmentation results of four successive frames and integrated the temporal continuity in the uniform energy function. First, moving objects were extracted by a region-based MRF model between two frames in a frame group of four successive frames. Then, the ultimate semantic object was labeled by minimizing the energy function of the STMRF model. Experimental results of the STMRF model showed that the proposed algorithm could accurately extract moving objects.

Some other approaches handled occlusion during object tracking [47, 48, 49, 50]. Senior *et al.* [48] used the appearance models to localize objects during partial occlusions, detect complete occlusions, and resolve depth ordering of the objects. The authors reported a good result on the PETS 2001 data set, though the performance was influenced by the pre-selected parameters used to update the probability mask values in the appearance models. [50] maintained a shape prior method to recover the missing object regions during occlusion, while the algorithm was initialized with the boundaries of the objects in the first frame. Stein *et al.* [49] proposed a mid-level model for reasoning more globally about object boundaries and propagating such local information to extract improved, extended boundaries with the utilization of subtle motion cues such as parallax induced by a moving camera. The method is mainly a boundary-based algorithm which needs to combine with other techniques to build up a region-based approach for object detection purpose.

From a brief overview of the existing approaches, it shows that many efforts have been made to solve the problem of detecting occluded objects in a video or the sequences of images. However, various kinds of restrictions were imposed before or during the detection processing. For example, domain knowledge was needed in [43, 45], interest points [44] or probability parameters [48] needed to be manually pre-selected, the boundaries of the objects in the first frame should be known in [50], etc. Aiming at designing a more generalized detection system, an unsupervised approach is proposed in our work to identify moving objects under occlusion situations.

2.2 Key Frame Detection Algorithms

The simplest way to get key frames is to use the sampling technique. For example, uniform sampling generates key frames at a fixed sampling rate. However, since the sampling methods do not consider the characteristics of non-homogeneous distribution of the video visual information, they suffer seriously from two major issues. First, the sampling results may miss a lot of important frames which contain the significant content of the videos. Second, since many frames within a shot are very visually similar to each other, content redundancies widely exist in sampling results. To overcome these problems, many key frame extraction algorithms have been proposed and developed.

One category of the algorithms generates key frames when the content change exceeds a certain threshold [24, 25, 26, 51, 52]. The content change could be measured by a function based on histograms, accumulated energy, etc. The algorithms belonging to this category do not require the existence of the frames coming afterwards. Moreover, no shot segmentation is required before applying key frame extraction methods. Therefore, they are suitable for real time applications. However, one problem for this kind of algorithms is that the key frames are generated without considering the content of frames in the remaining video sequence. Therefore, the selected key frames may still contain lots of redundancies and become suboptimal, since they cannot represent the content temporally after them. In other words, the content coverage of these key frames is only limited to the preceding frames.

To overcome the above problem, the coverage-based approaches are proposed in [27, 28, 29, 30], which aim to get a small number of key frames by maximizing each key frame's coverage towards adjacent frames. One method presented by Chang *et al*. [30] applied the greedy search to find key frames with the maximum coverage iteratively until all frames were represented by key frames. The major drawback of the coverage-based approaches is the heavy computation. In order to search key frames according to coverage, dissimilarity scores need to be calculated on all pairs of frames. Therefore, the performance of the coverage-based approaches is limited by the computation power of the underlying hardware. Another category of key frame extraction methods that gains much attention is the cluster-based algorithms [53, 54, 55, 56].

Cluster-based algorithms [56] require a preprocessing step that transforms the frames into the points of a feature space, where the clustering methods are applied and all points are grouped into a bunch of clusters. A cluster selection step is usually followed by picking up the significant clusters and extracting frames that are close to the cluster centers as the key frames. Cluster-based algorithms rely on a suitable feature space to represent the content of frames. However, good and clean clusters are not easy to be formed and therefore the patterns of data points in the feature space are not straightforward. In addition, the cluster-based methods are more complicated than the aforementioned key frame extraction methods. Theoretically, inter-cluster visual variance is large; while intra-cluster variance is small. Therefore, the redundancy within the extracted key frames can be kept below a certain level.

There are also some other algorithms that focus on addressing the redundancy problem in the extracted key frames. One method used the integration of local and global information to remove redundancy in the set of key frame candidates and achieved good results [57]. Minimum-Correlation based algorithms [58, 59, 60] assumed that the key frames had little correlation with each other. By pruning some significantly correlated frames, the algorithms could ensure that the extracted key frames hold a low level of redundancy. However, minimum-correlation-based algorithms were vulnerable to the outliers.

Furthermore, key frames could be extracted based on clips [31, 32] or shots [33, 34, 35, 36]. Shot-based approaches are quite intuitive since shots are regarded as the basic semantic unit in videos. Furthermore, shot segmentation techniques are quite mature recently, and thus it is applicable to detect shots within a video before applying the key frame extraction algorithms. The simplest approach of shot-based key frame extraction is to choose the first frame of each shot as a key frame, which works well for shots with low motion. The work in [61, 62] adopted motion changes within a shot as a criterion to select key frames. The idea was that more key frames should be extracted in shots that consist of frequent motion activity changes. However, motion features obtained in [61, 62] were from MPGE-7 motion activity descriptors which were not easy to be applied to the uncompressed videos. Therefore, the application of their work is limited to the compressed domain. To compensate such a blank area, [63] built a motion energy model to perceive motion patterns in the uncompressed domain. However, motion features do not always represent major content within shots. Therefore, their approaches work well for a particular application that is highly related with motion, such as sports [61, 62, 63].

2.3 Related Techniques

In this section, we briefly introduce the related techniques employed in this dissertation, which either provide the essential information of visual features or evaluate the performance of the new algorithms. In section 2.3.1, the detector of scale-space extrema of differences-of-Gaussians (DoG) is introduced, which is employed to locate the key locations in section 4.2. Section 2.3.2 discusses how to detect space time interest points in spatio-temporal domain using the Harris3D corner detector. Section 2.1 briefly presents the general Gaussian mixture models (GMM) and GMM supervector for feature representation prepared for the classification.

2.3.1 Detector of Key Location by Differences-of-Gaussians

The key locations detected on the video frame in our work are obtained from scale-space extrema of differences-of-Gaussians (DoG) within a difference-of-Gaussians pyramid [64, 65]. A Gaussian pyramid is constructed from the input image by repeated smoothing and subsampling, and a difference-of-Gaussians pyramid is computed from the differences between the adjacent levels in the Gaussian pyramid. Then, the key locations are obtained from the locations at which the difference-of-Gaussians values assume extrema with respect to both the spatial coordinates in the image domain and the scale level in the pyramid [66].

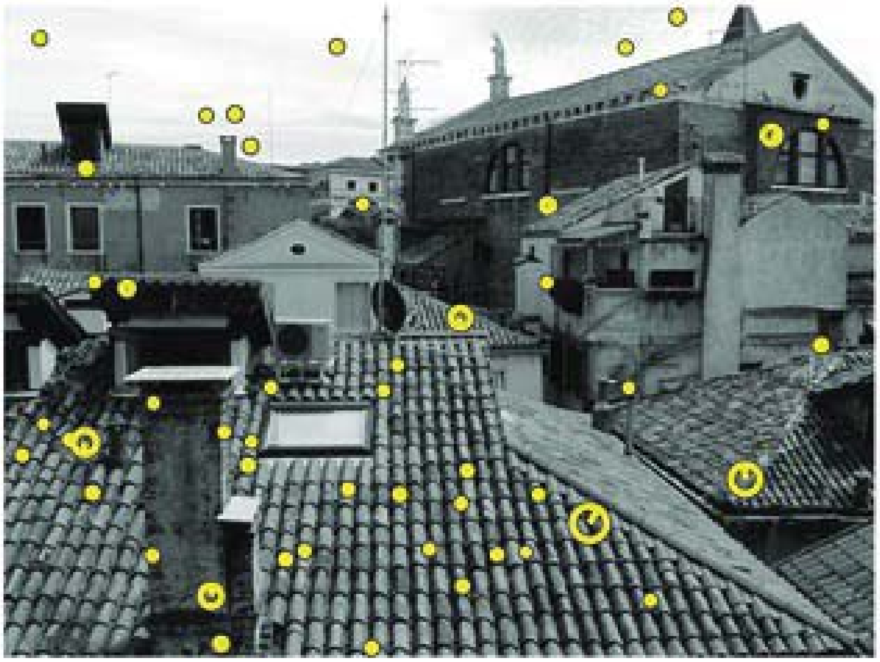


Figure 2.2: Key locations detected from a grey-level frame using scale-space extrema of the Laplacian. The radii of the circles illustrate the selected detection scales of the key locations [2].

2.3.2 Harris3D Corner Detector of Space-Time Interest Points

As a space-time extension of the Harris detector [67], Laptev and Lindeberg proposed the Harris3D detector in [68]. The spatio-temporal second-moment matrix at each video point is computed by using independent spatial and temporal scale values σ, τ, a separable Gaussian smoothing function g, and space-time gradients ∇L as shown below.

$$\mu(\cdot;\sigma;\tau) = g(\cdot;s\sigma;s\tau) * (\nabla L(\cdot;\sigma;\tau))(\nabla L(\cdot;\sigma;\tau))^T$$

The final locations of the space-time interest points are given by local maxima of H which is computed as below.

$$H = \det(\mu) - k\text{trace}^3(\mu), H > 0.$$

Laptev *et al.* [3] used those points extracted at multiple scales based on a regular sampling of the scale parameters σ and τ and achieved promising results. Therefore, we use the implementation codes on-line and standard parameter settings $k = 0.0005$, $\sigma^2 = 4, 8, 16, 32, 64, 128$,and $\tau^2 = 2, 4$ in this dissertation. Figure 2.3 is an illustration of interest points obtained by Harris3D corner detector. The size of the circle indicates the scale and the center of the circle indicates the location of the interest point.

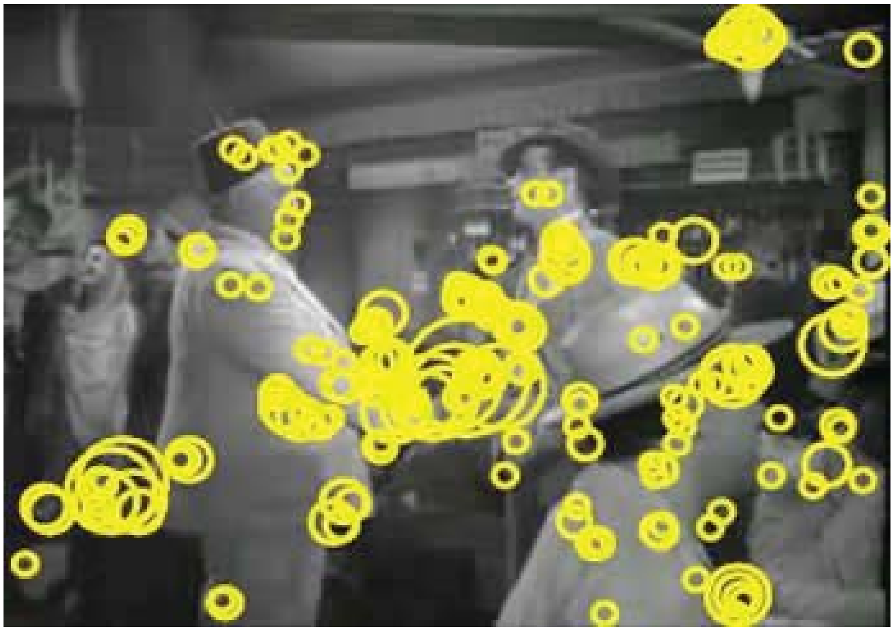


Figure 2.3: Space-time interest points detected for a video clip with human action - hand shake [3].

2.3.3 General Gaussian Mixture Models and GMM Supervector

Gaussian mixture models (GMM) have been proven extremely successful for multimedia semantic indexing [69]. Most methods in multimedia classification utilize the idea of stacking the means of the GMM model to form a GMM mean supervector which is viewed as a content representation of the training or testing instance. A Gaussian mixture model is a weighted sum of K component Gaussian densities as given by Equation (2.1),

$$p(x|\theta) = \sum_{k=1}^{K} \omega_k N(x|\mu_k, \Sigma_k), \tag{2.1}$$

where x is a D-dimensional continuous-valued data vector (i.e., features), ω_k ($k = 1, \ldots, K$) are the mixture weights, and $N(x|\mu_k, \Sigma_k)$ $k = 1, \ldots, K$, are the component Gaussian densities. Each component density is a D-variate Gaussian function of the form with mean vector μ_k and covariance matrix Σ_k as given in Equation (2.2).

$$N(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp\{-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)\}, \tag{2.2}$$

The mixture weights satisfy the constraint that $\Sigma_{k=1}^{K}\omega_k = 1$. The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices, and mixture weights from all component densities. These parameters are collectively represented by the following notation:

$$\lambda = \{\omega_k, \mu_k, \Sigma_k\} \quad k = 1, \ldots, K. \tag{2.3}$$

Given the training vectors, several techniques can be used to estimate the parameters of a GMM [70]. Maximum likelihood (ML) estimation is a common one which aims to find the model parameters that maximize the likelihood of a GMM, given the training data. For a sequence of T training vectors $X = \{x_1, \ldots, x_T\}$, the GMM likelihood can be written as follows [71].

$$p(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda). \tag{2.4}$$

However, the expression is a non-linear function of the parameter λ and it is impossible to get a direct maximization. Therefore, a special case of the expectation-maximization

(EM) algorithm proposed in [72] is employed to iteratively estimate the ML parameter. On each EM iteration, the mixture weights $\overline{\omega}_k$, means $\overline{\mu}_k$, and variances $\overline{\sigma}_k^2$ are re-estimated using Equations (2.5) to (2.7) to guarantee a monotonic increase in the model's likelihood value.

$$\begin{aligned}\overline{\omega}_k &= \frac{1}{T}\sum_{t=1}^{T}\Pr(k|x_t,\lambda), && (2.5)\\ \overline{\mu}_k &= \frac{\sum_{t=1}^{T}\Pr(k|x_t,\lambda)x_t}{\sum_{t=1}^{T}\Pr(k|x_t,\lambda)}, && (2.6)\\ \overline{\sigma}_k^2 &= \frac{\sum_{t=1}^{T}\Pr(k|x_t,\lambda)x_t^2}{\sum_{t=1}^{T}\Pr(k|x_t,\lambda)} - \overline{\mu}_k^2. && (2.7)\end{aligned}$$

The *a posteriori* probability for component k is given by Equation (2.8)

$$\Pr(k|x_t,\lambda) = \frac{\omega_k N(x_t|\mu_k,\Sigma_k)}{\sum_{k=1}^{K}\omega_k N(x_t|\mu_k,\Sigma_k)}. \qquad (2.8)$$

In most cases, the number of feature vectors extracted from a single video is not enough to estimate the GMM parameters precisely. Thus, in our work, the global Gaussian mixture models (called Universal Background Model (UBM)) is learnt by using the features from all the training videos. Then the UBM parameters are adapted in order to fit each particular data distribution (a training or testing video sequence). This adaptation is made by using the Maximum A Posteriori (MAP) approach [73].

The first step is to determine the probabilistic alignment of the training vectors with the UBM Gaussian components. For a Gaussian component k in the UBM, we compute

$$\begin{aligned}\mathrm{P}r(k,x_t) &= \frac{\omega_k p_k(x_t)}{\sum_{k=1}^{K}\omega_k p_k(x_t)};\\ n_k &= \sum_{t=1}^{T}\mathrm{P}r(k,x_t);\\ E_k(x) &= \frac{1}{n_k}\sum_{t=1}^{T}\mathrm{P}r(k,x_t)x_t.\end{aligned}$$

Here, x_t represents the tth feature vector of the video to be modeled. These statistical values are then used for adapting the mean vector $\hat{\mu}$ of each Gaussian component.

$$\begin{aligned} \hat{\mu}_k &= \alpha_k E_k(x) + (1 - \alpha_k)\mu_k; \\ \alpha_k &= \frac{n_k}{n_k + r}, \end{aligned}$$

where r is a fixed “relevance factor”. The concatenation of all the mean vectors of the K Gaussian components is called the GMM supervector which is first proposed as a speaker recognition method [74] and then has been applied to semantic indexing [69] and music similarity [75].

Chapter 3

Overview of Proposed Framework

3.1 Proposed Solutions

With the requirements of processing videos with uncontrolled backgrounds in an unsupervised way, we propose to estimate the locations of the moving objects in video sequences by analyzing and integrating spatio-temporally visual contrast information. Since the proposed framework mimics a human vision system and spontaneously focuses on the big motion contrast in the spatio-temporal domain, it has a good capability to automatically capture the main object-level motion and ignore the interference from the scenes. In addition, a new key frame extraction method is proposed to provide representative frames as the summarization of the video sequences, which can also provide a good foundation for the proposed moving object detection methods. Moreover, under object occlusion situations, an extended moving object detection method is presented to split and track the occluded objects, which later can be utilized in the detection step to enhance the performance. The final goal of our proposed methods is to supply discriminant object-level features on the moving objects to improve the performance of classification and retrieval. Figure 3.1 shows the flowchart of the proposed framework.

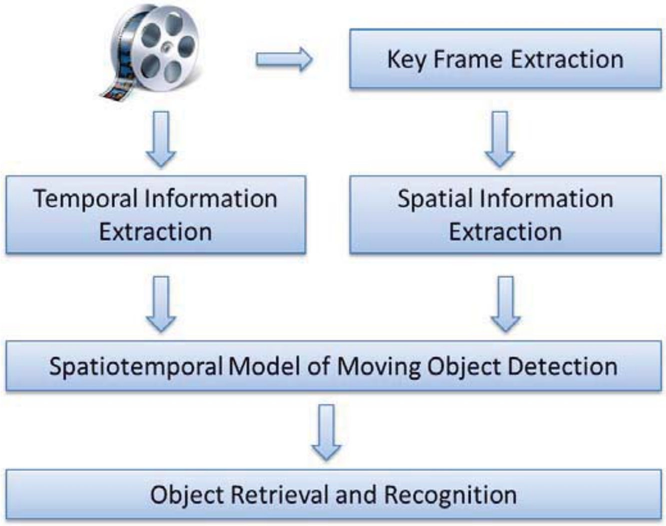

Key Frame Extraction
Temporal Information Extraction
Spatial Information Extraction
Spatiotemporal Model of Moving Object Detection
Object Retrieval and Recognition

Figure 3.1: The flowchart of the proposed framework

3.1.1 Spatio-temporal Framework of Moving Object Detection

The framework has two layers: global motion estimation layer and local motion estimation layer. The global motion estimation layer aims to quickly exclude the non-motion regions and passes the information to the next layer for further evaluation. At the global motion estimation layer, a new motion representation, called weighted histogram of Harris3D volume, is presented to integrate two sources of motion messages into a complementary one. The Harris3D corner detector is utilized to compute the space-time interest points by using the spatial and temporal scale values independently. The optical flow fields of the videos are calculated to weight the representation of Harris3D corners for a complete expression of the motion information. A fast grouping and searching method, named the integral density method, is proposed to find the region of the highest density as the moving object. The optical flow and Harris3D corner detector describe the motion signals from different perspectives, we expect the integrated representation enables to capture the moving object information accurately while resist the noise from the uncontrolled environment.

The local motion estimation layer receives the preliminary result from the global motion estimation layer, and analyzes the locally spatio-temporal information to identify the final region of moving object. First, key locations on the key frames are detected from the locations at which the difference of Gaussians values assume extrema with respect to both the spatial coordinates in the frame domain and the scale level in the difference of Gaussians pyramid. On the other hand, the Simultaneous Partition and Class Parameter Estimation (SPCPE) framework [76] is employed to preliminarily segment the key frames into foreground and background. The difference of the adjacent segmented key frames is defined as motion map to filter the key locations. The outcomes are called key motion locations (KMLs) which indicate the temporal changes in the video. We used the integral density algorithm to find the region that has the highest density of KMLs as the moving object.

3.1.2 Center-Surround Coherency Evaluation

The proposed model is inspired by biological mechanisms of human vision which make motion salience (defined as attention due to motion) more “attractive” than some other low-level visual features to people while watching the videos. Under this biological observation, motion vectors are calculated using the optical flow algorithm to estimate the movement of a block from one frame to another. A center-surround coherency evaluation model is proposed to compute the local motion saliency in a completely unsupervised manner. In the integral density algorithm, the local motion saliency can either work alone or be used to weight the spatio-temporal motion to find the region of moving object. Our proposed model evaluates video sequences captured in the non-static background. The promising experimental results verify the effectiveness of the proposed model integrating the local saliency and spatio-temporal motion information.

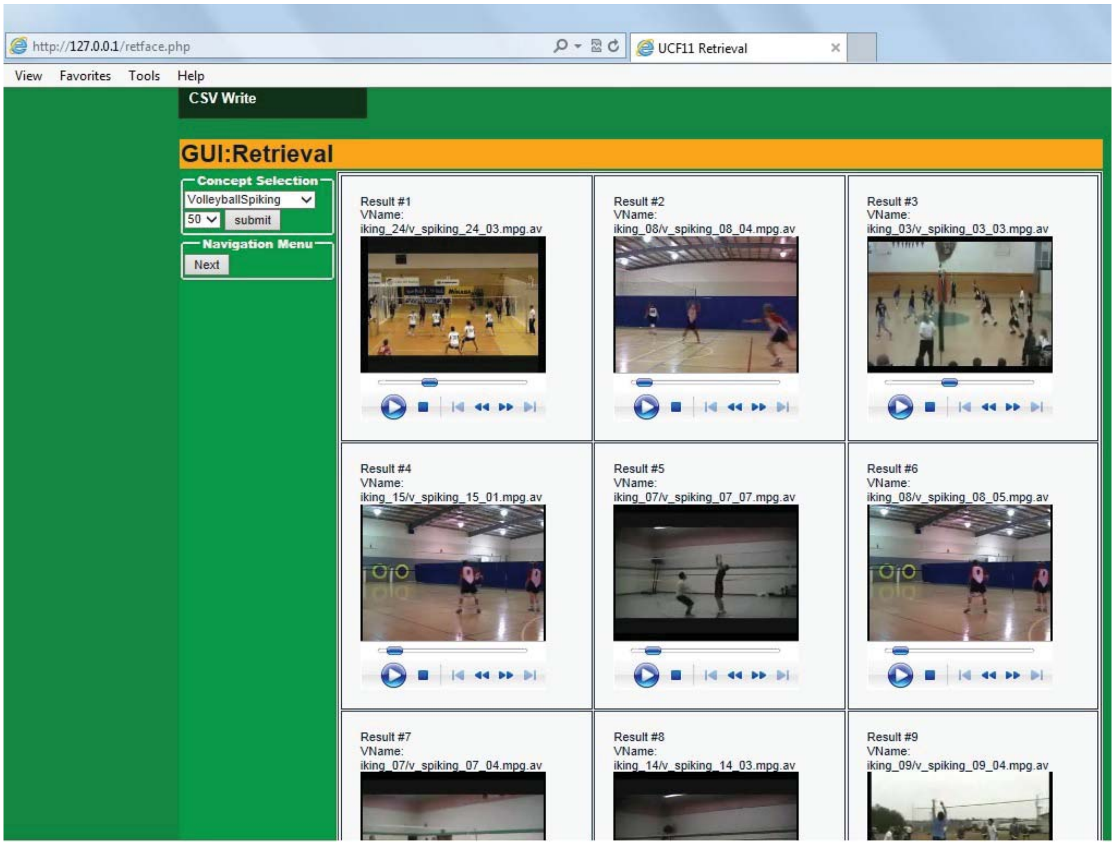
3.1.3 Key Frame Detection Using Local Motion Information

The local motion estimation method of moving object detection starts from analyzing the frames to obtain visual frame information. A set of good representative frames enables to abstract the video effectively. We propose a new key frame extraction method to extract a set of frames as the basis for the local motion analysis. The visual variance between frames is checked first to get the description of the frames, followed by a clustering technique to extract a set of key frame candidates for further filtering. The information within a video shot and between video shots is employed to adaptively select the final set of key frames.

3.1.4 Moving Object Detection under Object Occlusion Situations

For moving objects in a video sequence, their movements can bring extra spatio-temporal information of successive frames, which helps object detection, especially for occluded objects. A moving object detection approach is proposed for occluded objects in a video sequence with the assistance of the Simultaneous Partition and Class Parameter Estimation (SPCPE) unsupervised video segmentation method [76]. Based on the preliminary foreground estimation result from SPCPE and object detection information from the previous frame, an n-steps search (NSS) method is proposed and utilized to identify the locations of the moving objects, followed by a size-adjustment method that adjusts the bounding boxes of the objects. Several experimental results show that the proposed approach achieves good detection performance under object occlusion situations in serial frames of a video sequence.

3.1.5 Motion-Based Object Retrieval and Recognition



Result #1
VName:
iking_24/v_spiking_24_03.mpg.av

Result #2
VName:
iking_08/v_spiking_08_04.mpg.av

Result #3
VName:
iking_03/v_spiking_03_03.mpg.av

Result #4
VName:
iking_15/v_spiking_15_01.mpg.av

Result #5
VName:
iking_07/v_spiking_07_07.mpg.av

Result #6
VName:
iking_08/v_spiking_08_05.mpg.av

Result #7
VName:
iking_07/v_spiking_07_04.mpg.av

Result #8
VName:
iking_14/v_spiking_14_03.mpg.av

Result #9
VName:
iking_09/v_spiking_09_04.mpg.av

Figure 3.2: Demo interface of the retrieval system

The spatio-temporal motion detection framework identifies the location of a moving object where the object-level features can be extracted. Global Gaussian mixture models are trained from the object-level features and maximum a posteriori method is employed to finish the video representation in the classification step. The advantage of using the object-level only features, instead of using the full feature set, to learn the global Gaussian mixture models is the decreases of the off-line training time and better classification performance. To further improve the performance, scene features are considered to be integrated into the retrieval framework. In addition, a web-based video retrieval demo was implemented to visualize the retrieval performance. As shown in

Figure 3.2, the users could choose a concept (e.g., VolleyballSpiking) and the number of videos they want to retrieve (e.g., 50), then click “submit”. The relevant videos are shown and ranked based on the the similarity scores. Users can play the ranked videos in the demo to check whether the video contains the selected concept.

3.2 Using Within and Between Shot Information for Key Frame Extraction

As the cost of creating, acquiring, and transmitting video sharply decreases in the recent decade, huge amounts of video data have been created and delivered every day. Millions of YouTube videos are clicked each day and meanwhile, hundreds of thousands of new videos are uploaded to the expanding YouTube website. All of these create new demands on efficient video browsing, searching, categorization, and indexing. Videos can be regarded as a sequence or combination of video frames which are basic units of a video. Usually, the amount of frames within a video is quite large. For example, a video that lasts for 10 minutes at a frame rate of 25 frames per second has a total of $15,000$ frames. The analysis of a video based on its frames could be computationally unaffordable if the video is very long. Therefore, representative frames that commonly called key frames are selected and extracted from a video. These extracted key frames are supposed to be able to describe the content of the video and summarize the contained information [77, 78].

Usually, a video sequence is first divided into meaningful segments (shots), and then each shot is represented by key frames. Today, some websites like MEGAVIDEO provide key frame-based browsing functionality to each video, so that a user who wants to briefly browse a video’s content only needs to put the cursor on the interested video and glance at a sequence of key frames rather than to operate it in the traditional way

– clicking and watching the whole video clips. The key frame-based browsing functionality not only decreases the time that users spend in searching their favorite videos, but also reduces the network traffic by delivering a few images rather than the whole video streaming. Key frames are also widely used in video searching and indexing tasks. Almost all the teams utilized key frame-based features from video sources in the TRECVID high-level feature extraction and semantic indexing task [79]. Key frames make it practical for each team to analyze video contents and construct learning/ranking models, providing a list of ranked shots by their relevance to the concerned high-level features or concepts.

In this section, to achieve a balance between the quantity of key frames for summarizing the shots in a video sequence and the quality of key frames to represent the whole video, an innovative key frame extraction approach is proposed. Our proposed approach has the following contributions: (1) It identifies transitive regions and informative regions by analyzing the differences between consecutive frames at the shot level; (2) A modified clustering technique is utilized as the key frame extractor to select a set of key frame candidates (KFCs) in informative regions, while transitive regions are not used for key frame extraction; and (3) It integrates the frame information within a video shot and between video shots to filter redundant KFCs to generate the final set of key frames.

The rest of section 3.2 is organized as follows. The proposed key frame extraction approach is presented in Section 3.2.1. In Section 3.2.2, the experimental results and analysis are provided followed by time complexity analysis. Finally, Section 3.2.3 concludes the proposed method and discusses the contributions and limitations.

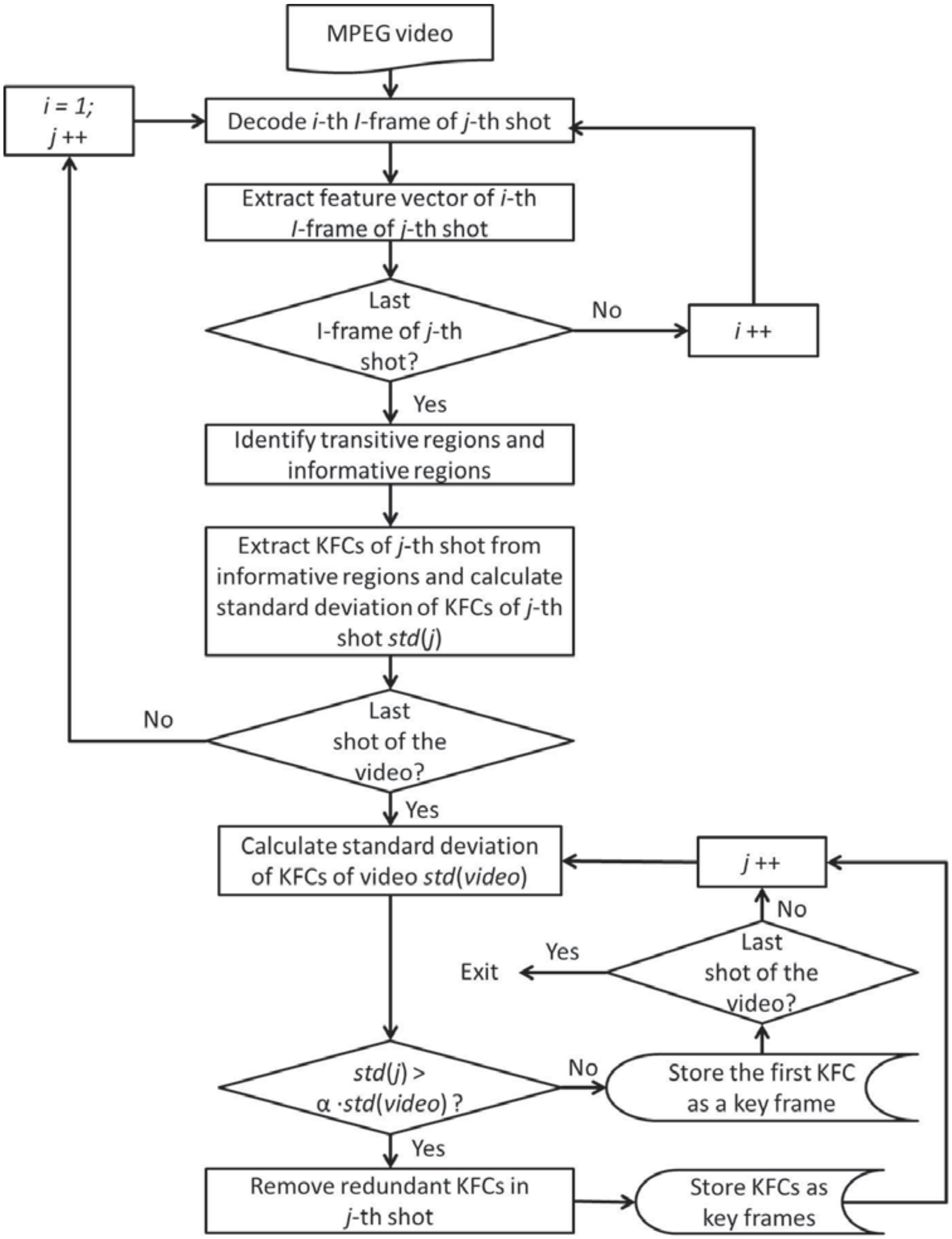

MPEG video
i = 1;
j ++
Decode i-th I-frame of j-th shot
Extract feature vector of i-th I-frame of j-th shot
Last I-frame of j-th shot?
No
i ++
Yes
Identify transitive regions and informative regions
Extract KFCs of j-th shot from informative regions and calculate standard deviation of KFCs of j-th shot std(j)
No
Last shot of the video?
Yes
Calculate standard deviation of KFCs of video std(video)
j ++
No
Yes
Exit
Last shot of the video?
std(j) > α ·std(video) ?
No
Store the first KFC as a key frame
Yes
Remove redundant KFCs in j-th shot
Store KFCs as key frames

Figure 3.3: System architecture of the proposed key frame extraction approach

3.2.1 The Proposed Video Key Frame Extraction Approach

Two main phases are included in the proposed approach. The first one is the cluster-based key frame candidate extraction phase, which extracts a group of key frame candidates (KFCs) on informative regions for each shot from the video sequences. The second one is the filtering phase, in which the information within a video shot and between video shots extracted from KFCs is used to remove those redundant KFCs. Figure 3.3 presents the system architecture of our proposed key frame extraction approach.

A Difference Measure Between Consecutive Frames

One of the frequently used methods that measure the differences of images is color histogram because of its computation simplicity. A color histogram is a representation of the distribution of colors in an image. For digital images, a color histogram is represented by counting the number of pixels belonging to each color. It provides a compact summarization of the color information in an image. However, the color histogram has one drawback that is loses the spatial distribution information of the color data. For example, by analyzing the color histogram, we can infer that the image is red and green, but it cannot tell which part of the image is red or green. Considering such a drawback of the color histogram, we use a color feature vector to represent the video frames and employ the Euclidean distance of the feature vectors to measure the difference between two frames.

For the sake of fast computation, each frame has been partitioned into several squares, and each square is a 16×16 block of pixels (as proposed by Kim *et al.* [80]). Then the average pixel value of each square is calculated as a feature of that frame. In other words, the frame has been resized to $1/256$ of the original size (each dimension being re-size to $1/16$ of its original size), and each pixel is represented by the average pixel value of a 16×16 block. For instance, a new image with a lower resolution (18×22) will be generated from an original frame with 288×352 pixels. If the columns are sequentially concatenated, the 18×22 image can be converted into a feature vector with 396 features. Therefore, a frame with 288×352 pixels is projected into the feature space as a feature vector with 396 features. Before re-sizing, the original frame was first transferred from RGB color space to YCbCr color space, and was calculated to represent each pixel in the frame by Equation (3.1) [80].

$$YCbCr_{avg} = \frac{2}{3} \cdot Y + \frac{1}{6} \cdot C_b + \frac{1}{6} \cdot C_r. \tag{3.1}$$

The problem of estimating the difference in the visual content of two frames is converted into a similarity measure of their feature vectors. There are several distance formulas for measuring the similarity of the feature vectors. Euclidean distance $d(\cdot)$ is employed to measure the similarity of the two feature vectors p and q using Equation (3.2).

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}, \tag{3.2}$$

where $p = [p_1 p_2 \ldots p_n]$, $q = [q_1 q_2 \ldots q_n]$, and n is the dimension of the feature vector (e.g., n = 396 in the above example).

A Cluster-Based Frame Extraction Method on Informative Regions

A video stream is made up of a group of shots and each shot is a series of consecutive frames from the start to the end of recording in a camera. For some shots, there is little visual content difference between successive frames. In such cases, a single frame may be sufficient to represent the content of all the frames in a shot. On the other hand, for those shots whose contents are more complex, more key frames are needed to represent the shots. One of the most common methods to select key frames is the temporal sampling method. Though the method is fast and easy to use, it usually does not provide a successful representation since it ignores the fact that the variation of the content within a shot is usually not proportional to the shot length.

In order to effectively select a proper set of key frames to represent the corresponding shot, the frames in each shot are first separated into two types: transitive frames (TFs) or informative frames (IFs). TFs are those frames that have large pictorial content differences compared with their adjacent frames, implying the transition of visual content due to the relatively fast movement of the camera. Compared with TFs, IFs contain the content and objects that are more stable and those are the real visual information that the recorder wants to take. Based on the above assumption, key frames are selected among IFs, but TFs are ignored.

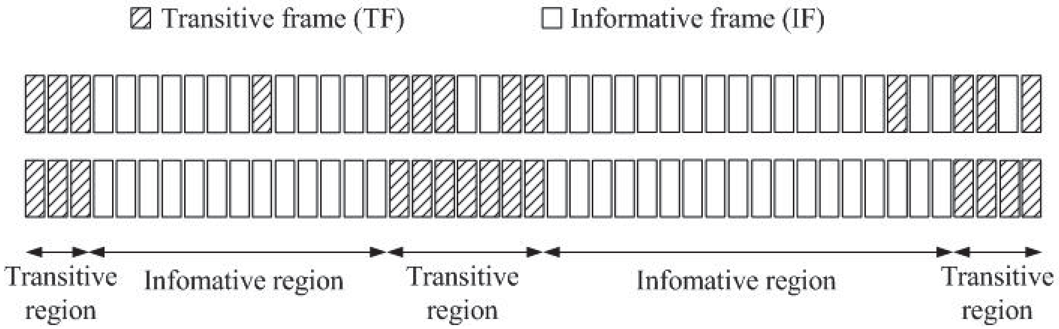


Figure 3.4: Identification of transitive regions and informative regions

Followed by the difference calculation between adjacent frames, the idea of constructing a binary classifier to identify TFs and IFs from [81] is adopted. It comes from the field of computer vision and image processing, and computes a global threshold that can be used to convert an intensity image to a binary image by choosing the threshold to minimize the intra class variance of the black and white pixels. The final goal of classifying TFs and IFs is to divide a shot stream into informative regions and transitive regions. A transitive region whose members are mostly TFs contains blurry objects or uniformly colored images that are meaningless in terms of the information supplied. Therefore, key frames should be selected from the informative regions. As can be seen from Figure 3.4, the first row is an initial classification of IFs and TFs, and the second row shows more continuous regions after the smoothing process.

For key frame selection, a clustering technique is employed in our proposed approach. Cluster analysis is the formal study of methods and algorithms for grouping data [82]. The general idea of cluster-based key frame extraction methods is to consider all the frames of a shot together and cluster them based on the similarity of their feature vectors. The frame that is closest to the cluster's center is usually selected as the key frame. One of the problems in clustering is that in order to identify a key frame, the center of a cluster needs to be first calculated. In our proposed approach, this step is omitted in order to reduce the computation time. In other words, instead of calculating the cluster's center and its distance with nearby frames, our proposed approach utilizes the middle frame of each shot as the first KFC f_1. Based on f_1, the second KFC f_2 is chosen using the following criterion:

$$\arg\max_{f_2} d(f_1, f_2) \tag{3.3}$$

The above expression is the set of values of f_2 for which $d(f_1, f_2)$ has the largest value, where *argmax* stands for the argument of the maximum and $d(f_1, f_2)$ indicates difference in the visual content between f_1 and f_2 obtained by Equation (3.2). To generalize the selection rule, the *n*th KFC f_n is selected using the following criterion:

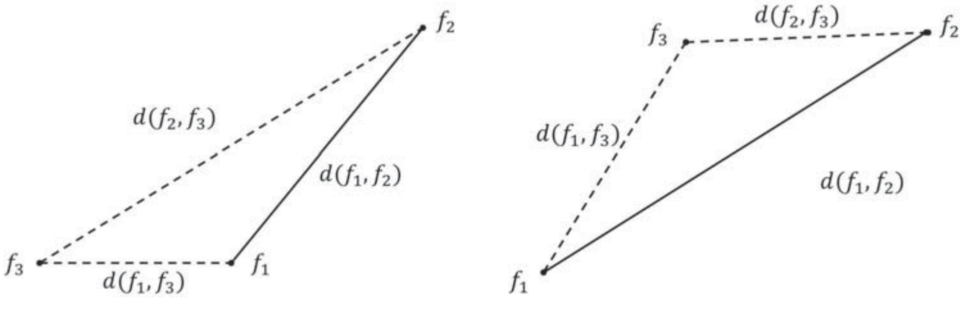
$$\arg\max_{f_n} \sum_{k=1}^{n-1} d(f_k, f_n). \tag{3.4}$$

This selection criterion chooses f_n for which the sum of differences between f_n and previous $n-1$ KFCs ($k = 1, 2, 3, \ldots, n-1$) has the largest value among all non-KFCs in the shot.

KFC Filtering by Integrating Information within a Video Shot and Between Video Shots

One of the major issues in the topic of key frame extraction is to decide the amount of key frames that should be selected per shot. A large set of key frames would give a more comprehensive representation of the video stream, but at the same time it would result in content redundancy. On the other hand, a small set of key frames could restrict pictorial redundancy, but it hardly represents the video content completely. Another issue is that the numbers of extracted key frames in different shots should be varied due to the unequal quantities of information conveyed in different shots. A commonly used method is pre-setting a threshold T in the key frame extractor, but the determination of the threshold T is another decisive factor to affect the final performance of the key frame extractor. Chatzigiorgaki *et al*. [83] used two videos from TRECVID 2007 test data set [79] as a training set to conduct the threshold selection process, which achieved good results in their experiments. It is an acceptable approach that employs the training videos to calculate the threshold for key frame extractor, but it would be more inspiring if a self-adapted method can be developed to decide the number of key frames to be extracted in each shot based on the video stream information itself. Such an extraction method would be more compact and accurate than those adopt the threshold calculated by other training videos.

In this section, we utilize the information within a shot and between shots (e.g., standard deviation of KFCs) to filter the KFCs. Assume that the shot content changes are relatively small, so the value of the standard deviation of KFC's feature vectors in the shot should be small, and vice versa. On the basis of such an assumption, the standard deviation of the whole KFCs set is used as a threshold to measure the content variation of each shot. The strategy is that if the standard deviation of the jth shot is less than the standard deviation of the whole KFCs set of the video, only KFC f_1 is reserved to represent the jth shot. Otherwise, Euclidean distances between KFCs are used to decide how many KFCs are kept as key frames to represent the shot. For instance, using the first two KFCs f_1 and f_2 that have been kept as key frames of the jth shot to decide whether f_3 should be kept as a key frame by evaluating the relationships among $d(f_1, f_2)$, $d(f_1, f_3)$ and $d(f_2, f_3)$. If in the case of Figure 3.5(a) that $d(f_1, f_3) + d(f_2, f_3) > 2 \cdot d(f_1, f_2)$, which means f_3 may contain extra information other than f_1 and f_2, then f_3 would be reserved as a key frame. However, if f_1, f_2 and f_3 construct a relation as shown in Figure 3.5(b) that $d(f_1, f_3) + d(f_2, f_3) \leq 2 \cdot d(f_1, f_2)$, it implies f_3 may have similarly visual content with f_1 or f_2.



(a) (b)

Figure 3.5: Two kinds of space layouts. (a) $d(f_1, f_3) + d(f_2, f_3) > 2 \cdot d(f_1, f_2)$; (b) $d(f_1, f_3) + d(f_2, f_3) \leq 2 \cdot d(f_1, f_2)$

Furthermore, in the case of Figure 3.6, assuming that f_3 has been selected as a key frame, KFC f_4 would be evaluated to obey the similar strategy. If the average length of the three dash lines in Figure 3.6 is larger than the average length of the other solid lines, f_4 is kept as a key frame. The same process applies to evaluate f_5 and so on.

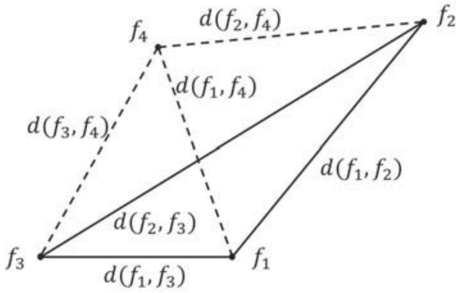


Figure 3.6: The space layouts of f_1, f_2, f_3 and f_4

If f_3 is removed, f_4 will not be considered and the process of filtering KFCs in the current shot terminates. The general rules in the filtering phase are as follows.

```
Keep $f_1$ as a key frame and set $n = 2$
IF    $std(j) >= \alpha \cdot std(video)$    THEN
    WHILE    $(\frac{1}{n-1}\sum_{k=1}^{n-1} d(f_k, f_n) > \frac{(n-1)(n-2)}{2}\sum_{p=1,q=2}^{n-1} d(f_p, f_q))$
        Keep $f_n$ as a key frame and $n = n+1$
    END-WHILE;
END-IF;
```

where $std(j)$ denotes the standard deviation of the KFCs in the jth shot, $std(video)$ denotes the standard deviation of all KFCs in the video, and α is the coefficient whose value is between 0 and 1. The expression in the “while” statement indicates that if the average Euclidean distance between KFC f_n and the other $n-1$ key frames in the same shot is larger than the average Euclidean distance between $n-1$ key frames, KFC f_n should be kept as a key frame and the process to evaluate KFC f_{n+1} is continued.

3.2.2 Experimental Results and Analyses

Shot boundary detection algorithm aims to break up the video into meaningful sub-segments by using pixel comparison between consecutive frames, edge changes, greyscale and RGB color histogram features, and similarity analysis. In this section, shot boundary information was given by the data provider, so we do not discuss shot boundary detection here, but focus on testing the robustness of the proposed key frame extraction method at the shot level.

Evaluation Metrics

We carried out the evaluation of the proposed approach in terms of the percentage of the extracted key frames and the retrieval precision. In the premise of no significant scenario missing during the key frame extraction process, the smaller number of key frames we use to represent the video, the better performance the key frame extractor does. Therefore, low key frame percentage is preferred to avoid unnecessary content redundancy in the set of key frames. The percentage of the extracted key frames (%KF) is defined as follows.

$$\%KF = \frac{number\ of\ extracted\ key\ frames}{total\ number\ of\ frames} \times 100\% \tag{3.5}$$

In statistics, the precision is the number of correct results divided by the number of all returned results. To define correctly extracted key frames, we introduced a concept called *hit deviation* to evaluate the quality of the extracted key frames. Hit deviation is defined as the difference between true frame (ground truth) index number and extracted key frame index number. The distance of the frame index number between a true and an extracted key frame that is less than the preset hit deviation means a correct extraction. For instance, if the index number of a true key frame is 39, and the hit deviation threshold is set to 5, an extracted key frame's index number between 34 and 44 is viewed as a correct hit. If two or more extracted key frames hit one true key frame, only the nearest one was recorded and the others are ignored. As shown in Figure 3.7, KF2 fails to hit the adjacent ground truths GT1 and GT2, since there has other key frames that are closer to the true key frames GT1 and GT2 than KF2. In the case of MPEG-1 video with 25fps, if the difference between two frames' index numbers is less than 25, it means the time interval between the two frames is less than one second. Generally speaking, the visual content changes would be relatively small within one second and the frames in one second have similarly pictorial content.

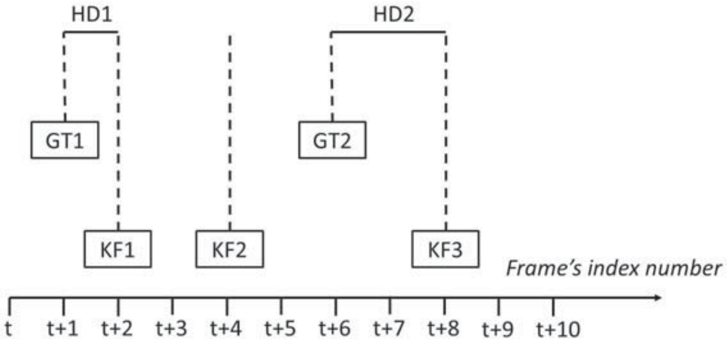


Figure 3.7: Hit deviation measure of extracted key frames. *t* denotes frame index number at time *t*; KF1, KF2, KF3 denote extracted key frames; GT1 and GT2 denote true key frames (ground truth), and HD1 and HD2 denote hit deviations.

Videos Data Sets

Fourteen MPEG-1 video sequences with 25fps from TRECVID 2007 test video collection [79] were used to evaluate the performance of the key frame extraction approach. Table 3.1 summarizes the characteristics of the fourteen video test sequences including the name, length, number of shots, number of frames, average number of frames per shot, number of ground truth, and average number of ground truths per shot.

In addition to the proposed approach, for the purpose of comparison, we also tested the commonly used key frame extraction method: temporal sampling. In particular, temporal sampling was implemented into two versions. One is the average temporal sampling that samples a pre-fixed number of frames per shot with an equal interval. Another sampling method is called adaptive temporal sampling [84] whose initial pur-

Table 3.1: The fourteen videos used to test the key frame extraction approaches. # of shot denotes the number of shots the video contains, # of Fr denotes the number of frames the video contains, and FrPerShot indicates the average number of frames for each shot; while # of true KF and tKFPerShot indicate the number of ground truth and the average number of ground truth per shot respectively.

Video name	Length(hh:mm:ss)	# of shots	# of Fr	FrPerShot	# of ture KF	tKFPerShot
BG_2196	00:26:13	124	39339	317.234	147	1.185
BG_10241	00:15:40	131	23517	179.504	146	1.115
BG_11369	00:06:33	59	9828	166.542	90	1.525
BG_34837	00:15:05	156	22624	145.026	203	1.301
BG_35447	00:14:59	127	22489	177.079	174	1.370
BG_35751	00:15:11	108	22786	210.981	137	1.269
BG_35757	00:14:53	85	22327	262.671	112	1.318
BG_35767	00:14:27	74	71679	968.635	94	1.270
BG_36304	00:18:06	163	27169	166.681	290	1.779
BG_36366	00:14:28	92	21706	235.935	113	1.228
BG_36511	00:10:03	72	15075	209.347	86	1.194
BG_37613	00:15:53	117	23830	203.675	139	1.188
BG_37796	00:15:16	74	22912	309.622	92	1.243
BG_38002	01:08:53	700	103347	147.639	1003	1.433

pose was to select more frames in a rapidly changing shot region. The method selects sampling rate on the basis of accumulated value of the color histogram differences in the video. We use a modified version of adaptive temporal sampling by using our feature vectors instead of color histogram. *I*-frames of each shot were first selected as basic frames for the KFC extraction.

Results

The experimental results of the key frame percentages are presented in Table 3.2. One of the drawbacks in temporal sampling method is that the sampling rate should be preset manually. Since the average number of true key frames per shot is greater than one, here for adaptive temporal sampling, we extracted twice of the number of ground truth key frames to represent the video. For the average temporal sampling method, we set the sampling rate to two and three.

Table 3.2: Key frame percentage (%). AvgTS (3) and AvgTS (2) denote the average temporal sampling with sampling rate at 3 and 2 frames per shot, respectively; AdaTS (2) denotes the adaptive temporal sampling with an average sampling rate on two frames per shot.

%KF	AvgTS(3)	AvgTS(2)	AdaTS(2)	Proposed method	Ground truth
BG_2196	0.946	0.630	0.630	0.493	0.374
BG_10241	1.671	1.114	1.114	0.850	0.621
BG_11369	1.801	1.201	1.201	0.926	0.916
BG_34837	2.069	1.379	1.379	1.224	0.897
BG_35447	1.694	1.129	1.129	1.009	0.774
BG_35751	1.422	0.948	0.948	0.812	0.601
BG_35757	1.142	0.761	0.761	0.703	0.502
BG_35767	0.310	0.206	0.206	0.187	0.131
BG_36304	1.800	1.200	1.200	1.097	1.067
BG_36366	1.272	0.848	0.848	0.691	0.521
BG_36511	1.433	0.955	0.955	0.736	0.570
BG_37613	1.473	0.982	0.982	0.864	0.583
BG_37796	0.969	0.646	0.646	0.554	0.402
BG_38002	2.032	1.355	1.355	1.161	0.971
Average	1.431	0.954	0.954	0.808	0.638

As can be seen from the results, the key frame percentages of our proposed approach were all limited to a maximum of 1.224%; while on average it reached 0.808%. In other methods, the maximum is 1.379% or 2.069% and the average key frame percentage is 0.954% or 1.431%. Compared with the ground truths that reach 1.067% on maximum and 0.638% on average, it shows that our proposed approach effectively eliminates the redundancy and controls the key frame percentage in an acceptable level (i.e., about 1.2 times of the ground truths on average). It indicates that our proposed key frame extraction approach is able to find a smaller subset of frames, which is in fact a better representation in summarizing the video. This is preferable in any key frame extraction method. Figures 3.8, 3.9, 3.10 and 3.11 show the precision values of four methods with hit deviation setting at 5, 15, 30 and 60S, respectively. When we set the hit deviation to 5, 15, 30 and 60, the best average results of temporal samplings are 15.4%, 35%, 50.6%

and 63.4%, in contrast with the average precision values of our proposed approach, namely 22.8%, 45.2%, 63.9% and 72.1%, respectively. It indicates that our proposed approach outperforms the temporal samplings approaches for both the average one and the adaptive one, which verifies the effectiveness of our proposed approach in video summarization.

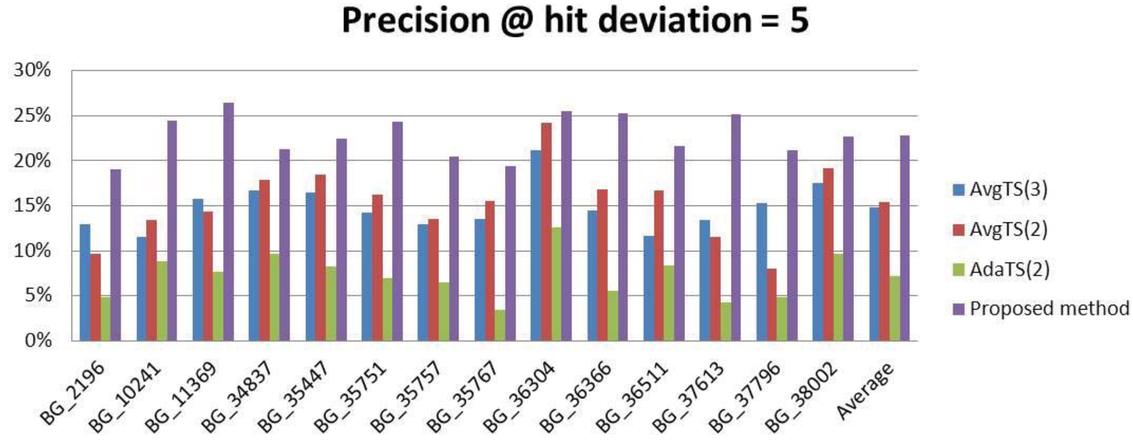


Figure 3.8: Precision results when hit deviation (HD) was set to 5, AvgTS (3) and AvgTS (2) denote the average temporal sampling with sampling rate at 3 and 2 frames per shot, respectively; AdaTS (2) denotes the adaptive temporal sampling with an average sampling rate on two frames per shot.

In order to make a pictorial comparison on visual content of the extracted key frames, we selected a video clip to evaluate the proposed approach and the temporal sampling approach. The extraction results are shown in Figures 3.12, 3.13, 3.14 and 3.15. Compared with the ground truth in Figure 3.16, lots of redundant frames existed in Figure 3.12. The same problem also happens in Figures 3.13 and 3.14. Furthermore, content redundancy in Figures 3.13 and 3.14 also suffered from the issue of missing key frames. The results of our proposed approach are shown in Figure 3.15 which has successfully reserved effective key frames and reduced the overlapped information.

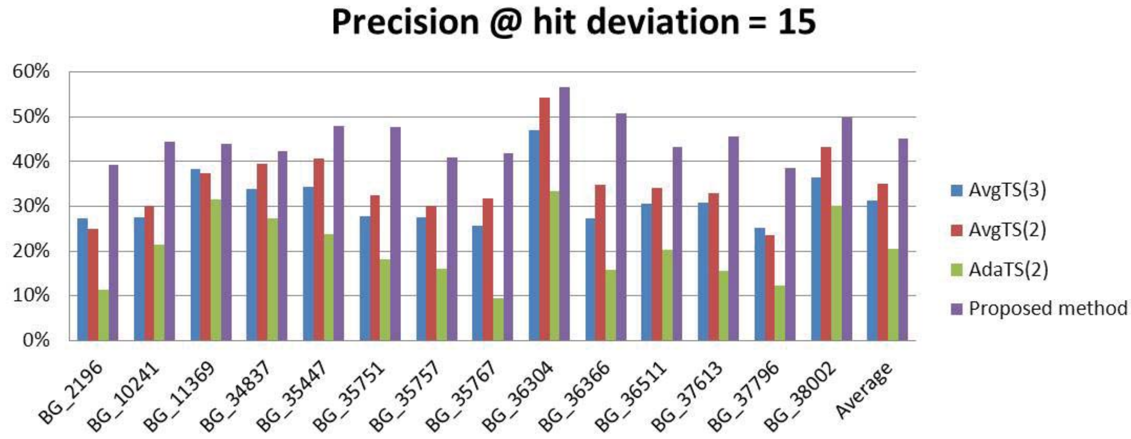


Figure 3.9: Precision results when hit deviation (HD) was set to 15, AvgTS (3) and AvgTS (2) denote the average temporal sampling with sampling rate at 3 and 2 frames per shot, respectively; AdaTS (2) denotes the adaptive temporal sampling with an average sampling rate on two frames per shot.

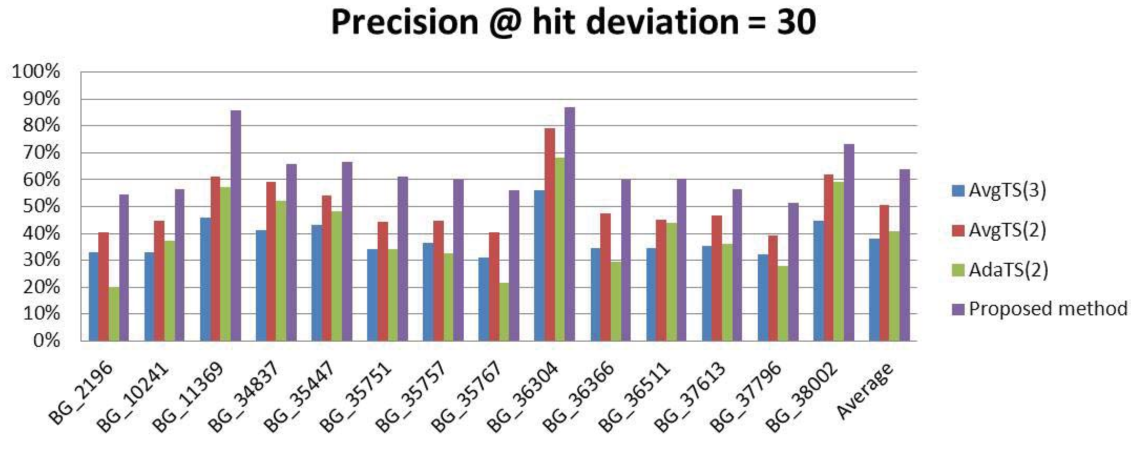


Figure 3.10: Precision results when hit deviation (HD) was set to 30, AvgTS (3) and AvgTS (2) denote the average temporal sampling with sampling rate at 3 and 2 frames per shot, respectively; AdaTS (2) denotes the adaptive temporal sampling with an average sampling rate on two frames per shot.

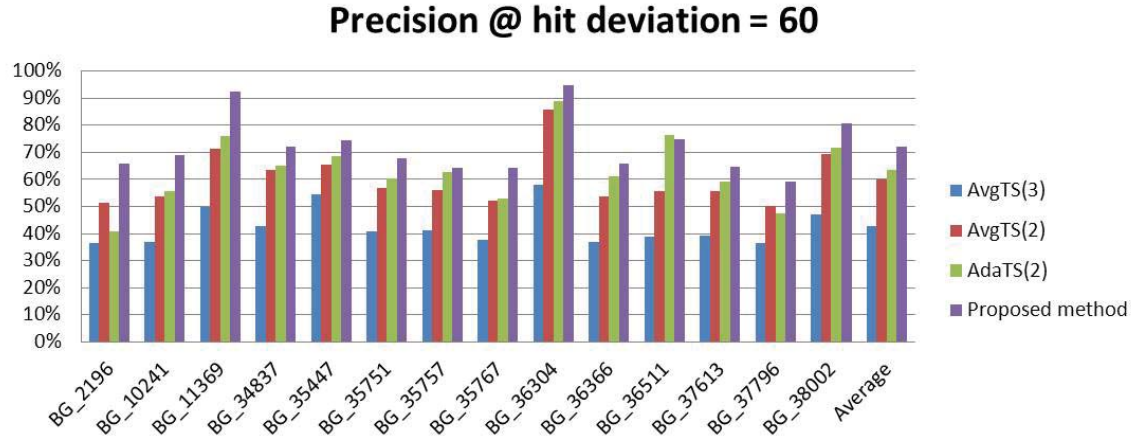


Figure 3.11: Precision results when hit deviation (HD) was set to 60, AvgTS (3) and AvgTS (2) denote the average temporal sampling with sampling rate at 3 and 2 frames per shot, respectively; AdaTS (2) denotes the adaptive temporal sampling with an average sampling rate on two frames per shot.

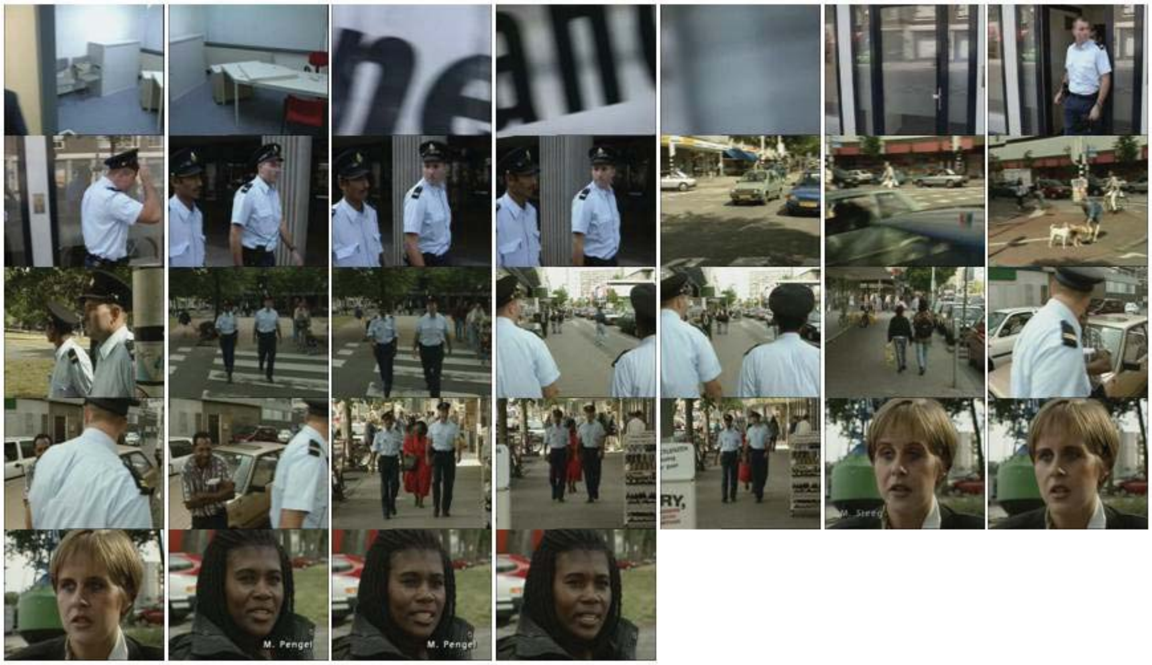


Figure 3.12: Key frames extracted by average temporal sampling (3 frames per shot) on 11 consecutive shots from shot 11 to shot 21 in video BG_2196

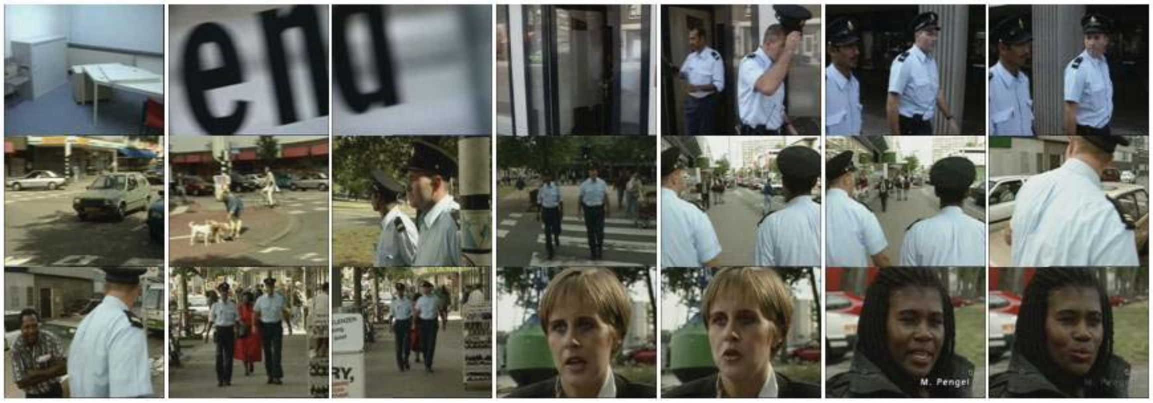


Figure 3.13: Key frames extracted by average temporal sampling (2 frames per shot) on 11 consecutive shots from shot 11 to shot 21 in video BG_2196

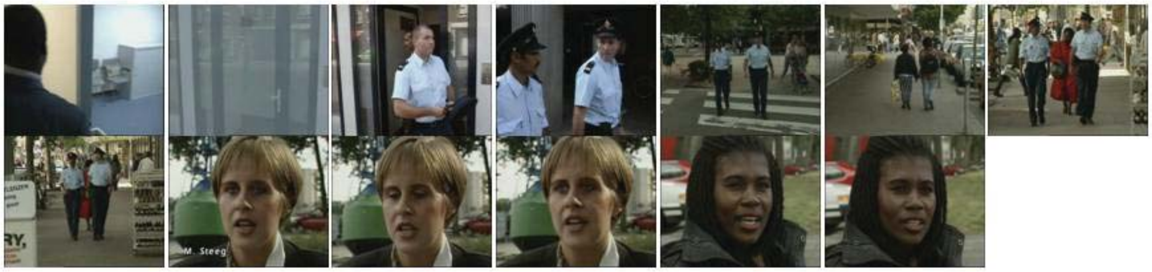


Figure 3.14: Key frames extracted by adaptive temporal sampling (averagely 2 frames per shot) on 11 consecutive shots from shot 11 to shot 21 in video BG_2196

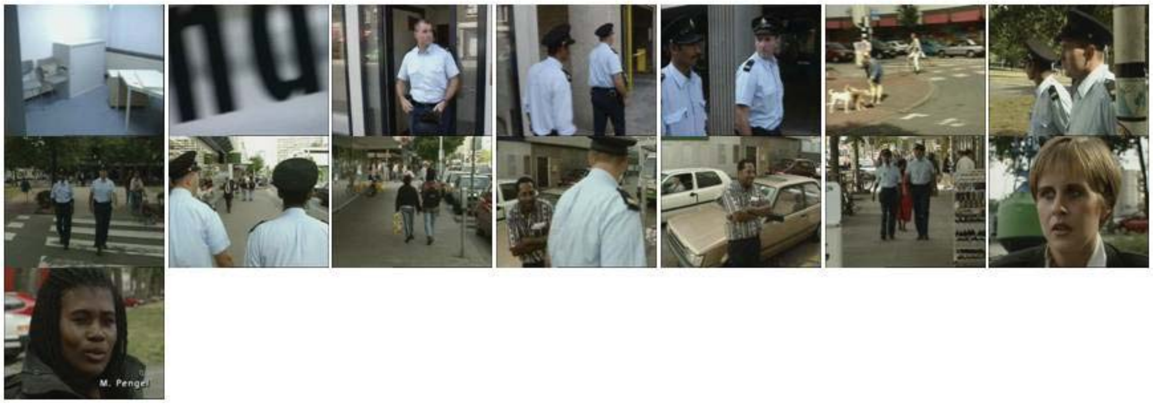


Figure 3.15: Key frames extracted by proposed method on 11 consecutive shots from shot 11 to shot 21 in video BG_2196

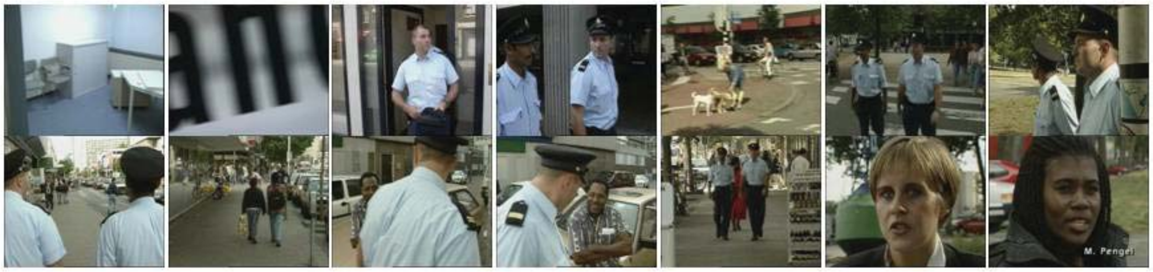


Figure 3.16: Ground Truth on 11 consecutive shots from shot 11 to shot 21 in video BG_2196

Time Complexity Analysis

In terms of the theoretical complexity, the proposed approach is more expensive than the adaptive temporal sampling and average temporal sampling approaches. Specifically, the adaptive temporal sampling and average temporal sampling approaches have the time complexities of $O(N)$ and $O(1)$, respectively; while the proposed approach takes $O(N^2)$, where N is the number of frames in a video sequence. However, the extraction time complexity might not be an issue since in most of cases, key frames can be extracted off-line. In this case, due to the lower percentage and better precision of key frames of the proposed approach, the key frame based applications such as searching and browsing become more efficient in terms of time and space complexities. Table 3.3 shows the computation time (in minutes). We have implemented the key frame extraction algorithm in Matlab the R2008a development environment. The computer used for the comparison was an Intel Core2 Duo CPU T6400 (2.00GHz) with 4GB of RAM, and running a Windows 7 Home Premium operating system.

3.2.3 Conclusions

This section proposes an effective key frame extraction approach by utilizing the information within a video shot and between video shots. Our proposed approach first selects a set of key frame candidates in the informative regions, and then removes a few

Table 3.3: Computation time of four methods on the 14 videos, AvgTS (3) and AvgTS (2) denote the average temporal sampling with sampling rate at 3 and 2 frames per shot, respectively; AdaTS (2) denotes the adaptive temporal sampling with an average sampling rate on two frames per shot.

Extraction time (minute)	AvgTS(3)	AvgTS(2)	AdaTS(2)	Proposed method
BG_2196	0.017082	0.016620	1.362779	243.447
BG_10241	0.017331	0.060535	4.431702	166.835
BG_11369	0.007284	0.005452	4.095344	68.483
BG_34837	0.017959	0.010439	1.383119	150.277
BG_35447	0.016773	0.016574	0.563536	142.304
BG_35751	0.017110	0.016050	1.224688	164.526
BG_35757	0.009854	0.015565	0.750621	151.175
BG_35767	0.015513	0.009274	0.733494	142.495
BG_36304	0.020402	0.012377	1.636738	182.165
BG_36366	0.015871	0.015543	1.405718	159.646
BG_36511	0.010955	0.009050	4.134540	99.356
BG_37613	0.017339	0.010614	2.167435	159.450
BG_37796	0.016350	0.015164	0.961948	150.424
BG_38002	0.060810	0.064711	1.760058	733.047
Average	0.018617	0.019855	1.900837	193.831

of redundant key frame candidates to finalize the set of key frames based on the evaluation of within and between shot information. Through the filtering process, most of the redundant key frame candidates are successfully deleted to obtain a reduced set of key frames. According to the performance in term of extraction percentage and retrieval precision, the proposed approach effectively demonstrates its good capability of reserving effective key frames while reducing overlapped visual content. One of our further improvement directions is to self-adaptively choose the position of the initial KFC f_1, while currently we use the middle frame of a shot as the initial KFC f_1. Another enhancement direction is to extract KFCs by using the object and motion information in both temporal and spatial dimensions from the video sequences. We believe it would deliver compensatory information which is not available in the current image-based key frame extraction methods.

3.3 Moving Object Detection under Object Occlusion Situations

It is a great challenge to detect an object that is overlapped or occluded by other objects in images. For moving objects in a video sequence, their movements can bring extra spatio-temporal information of successive frames, which helps object detection, especially for occluded objects. A moving object detection approach is proposed for occluded objects in a video sequence with the assistance of the SPCPE (Simultaneous Partition and Class Parameter Estimation) unsupervised video segmentation method. Based on the preliminary foreground estimation result from SPCPE and object detection information from the previous frame, an n-steps search (NSS) method is utilized to identify the location of the moving objects, followed by a size-adjustment method that adjusts the bounding boxes of the objects. Several experimental results show that our proposed approach achieves good detection performance under object occlusion situations in a series of frames of a video sequence.

3.3.1 The Proposed Approach

Figure 3.17 presents the system architecture of the proposed approach for each frame i ($i > 1$) in a video sequence. It includes four steps to handle the occlusion situation between moving objects.

In the first step, background and foreground of frame i are estimated with the help of the unsupervised SPCPE video segmentation method using the background and foreground of frame $i-1$ as the initial class partition. After removing the background, bounding boxes are used to describe the foreground objects. Of note, the first frame of the video needs to be processed through SPCPE using an arbitrary initial class partition to get the bounding boxes of its foreground objects. In Step 2, an idea from [39] is adopted to detect object occlusion situations by using the size and location information

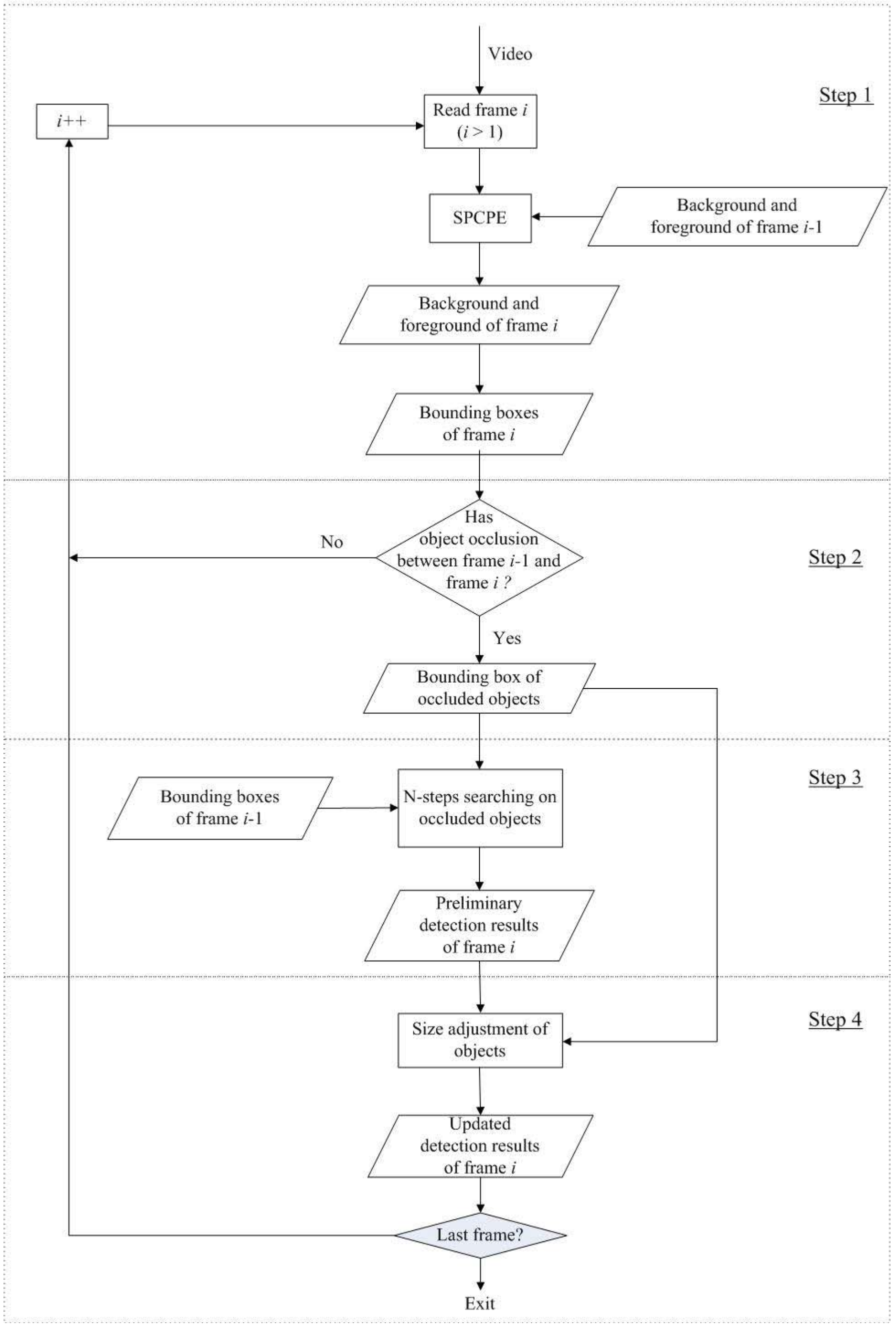

Video
Step 1
i++
Read frame i (i > 1)
SPCPE
Background and foreground of frame i-1
Background and foreground of frame i
Bounding boxes of frame i
Has object occlusion between frame i-1 and frame i ?
No
Step 2
Yes
Bounding box of occluded objects
Step 3
Bounding boxes of frame i-1
N-steps searching on occluded objects
Preliminary detection results of frame i
Step 4
Size adjustment of objects
Updated detection results of frame i
Last frame?
Exit

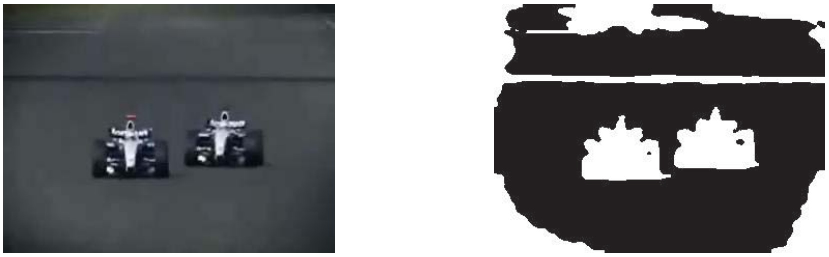
Figure 3.17: The system architecture of the propose approach

of bounding boxes in two consecutive frames (i.e., frames $i-1$ and i). If object occlusion occurs, the bounding box of the occluded objects is passed to Step 3; otherwise, the loop goes back to Step 1 to process the next frame. In order to identify the location of the occluded objects more generally, an n-steps search (NSS) method is employed by using the spatial information of the objects in frame $i-1$, and the preliminary detection results of frame i are generated in Step 3. Finally, in Step 4, a size-adjustment method is developed to adjust the bounding boxes of the occluded objects for the purpose of obtaining more accurate sizes and positions of the objects. The same steps are iterated for all the frames i ($i > 1$) of the video.

Background and Foreground Estimation

The background and foreground estimation method presented here is based on the SPCPE algorithm [76] that is able to partition objects from the background. The segmentation starts with an arbitrary class partition (for the first frame) and then an iterative process is employed to jointly estimate the class partition and its corresponding class parameters (for the rest of the frames in the video sequence).

The SPCPE algorithm is applied to segment each pixel in frames into two classes, namely background and foreground. Let the segmentation variable be $c = \{c_b, c_f\}$ and the class parameter be $\theta = \{\theta_b, \theta_f\}$. Let all the pixel values y_{ij} (where i and j are the row number and column number of the pixel, respectively) in the frame belonging to class k be put into a vector Y_k, where $k = b$ means background and $k = f$ means foreground. Each row of the matrix Φ is given by $(1, i, j, ij)$, and α_k is the vector of



(a) A video frame

(b) Estimated background and foreground for (a)

Figure 3.18: An example of the SPCPE estimation result

parameters $(\alpha_{k0}, \ldots, \alpha_{k3})^T$.

$$y_{ij} = \alpha_{k0} + \alpha_{k1} i + \alpha_{k2} j + \alpha_{k3} ij, \ \forall (i, j) \ y_{ij} \in c_k \tag{3.6}$$

$$\mathbf{y}_k = \Phi \alpha_k \tag{3.7}$$

$$\widehat{\alpha}_k = \{\Phi^T \Phi\}^{-1} \Phi^T \mathbf{y}_k \tag{3.8}$$

Here, it is assumed that the adjacent frames in a video do not differ much, and thus the estimation results of background and foreground of successive frames do not change significantly. Under this assumption, the segmentation of the previous frame is used as an initial class partition, so the number of iterations for processing is greatly decreased. Since the first frame does not have a previous frame, an arbitrary class partition is used to start the estimation process. Figure 3.18(a) is a color frame extracted from a video of race cars, and Figure 3.18(b) is its background (shown in black) and foreground (shown in white) estimation result by the SPCPE algorithm.

Previous Occlusion Detection Strategy

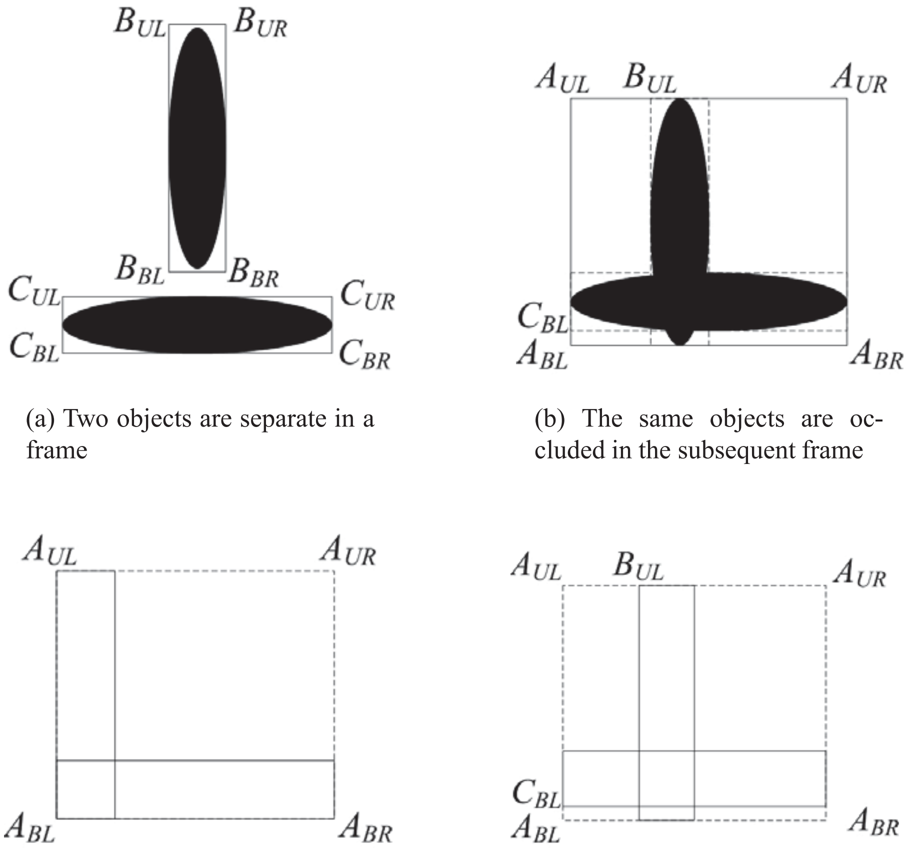
An effective method was proposed in [39] to detect the occlusion of objects by utilizing the concept of minimal bounding rectangle (MBR) in R-trees [85] to bound each semantic object by a rectangle. The main idea is to measure the distances and sizes of the bounding boxes between frames to check if two segments in adjacent frames represent the same object. If a segment cannot find its successor in the subsequent frame, then a merge or split of objects may happen between the two frames.

In [39], the authors proposed a backtrack-chain-updation split algorithm and a vertex recovery method to identify the occluded objects, which work well under the situation that two objects with similar sizes and shapes merge or split from the diagonal direction. However, the vertex recovery method may fail in other situations. For example, in Figure 3.19(a) and Figure 3.19(b), the vertex recovery method would "paste" vertex B_{UL} onto vertex A_{UL} and vertex C_{BL} onto vertex A_{BL}, leading to the detection result as shown in Figure 3.19(c), while the correct bounding boxes should be located as shown in Figure 3.19(d).

New Occluded Objects Detection Approach

Assume that the appearance of the same object in adjacent frames does not change a lot, the idea of a quick block motion estimation method [86], called three-step search (TSS), is extended to identify the location of occluded objects from the spatial information in the previous frame. The TSS algorithm is based on a coarse-to-fine approach with logarithmic decreases in steps as shown in Figure 3.20. In TSS, the initial step size is half of the maximum motion displacement p. For each step, nine checking points are matched and the minimum Mean Absolute Difference (MAD) [87] point of that step is chosen as the starting center of the next step whose size is reduced by half. When the step size is reduced to 1, the searching process terminates. The three-step is obviously designed for a small search window (i.e., $p = 7$).

In this section, TSS is extended to an n-steps search by using the same searching strategy. For the sake of quick computation of MAD, the search process is conducted on



(a) Two objects are separate in a frame

(b) The same objects are occluded in the subsequent frame

(c) Split result by vertex recovery method

(d) Correct detection result

Figure 3.19: An example when the vertex recovery method would fail

the SPCPE segmentation result instead of the color frame, and we use the bounding box in the previous frame as the reference block. Figure 3.21(a) is the SPCPE segmentation result of the current frame where two objects are identified as one. Figure 3.21(b) is the segmentation result utilizing the positions of the bounding boxes of the previous frame, which is not precise; while on the basis of Figure 3.21(b), n-steps search returns an acceptable object detection result (as shown in Figure 3.21(c)).

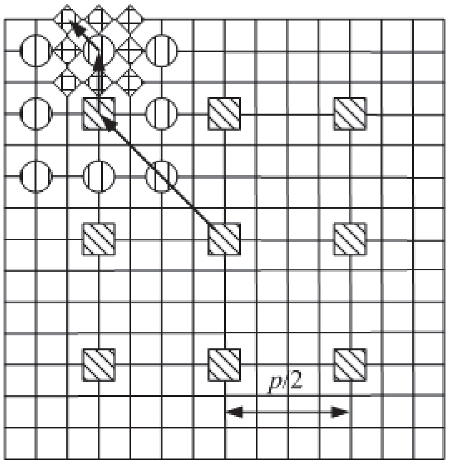
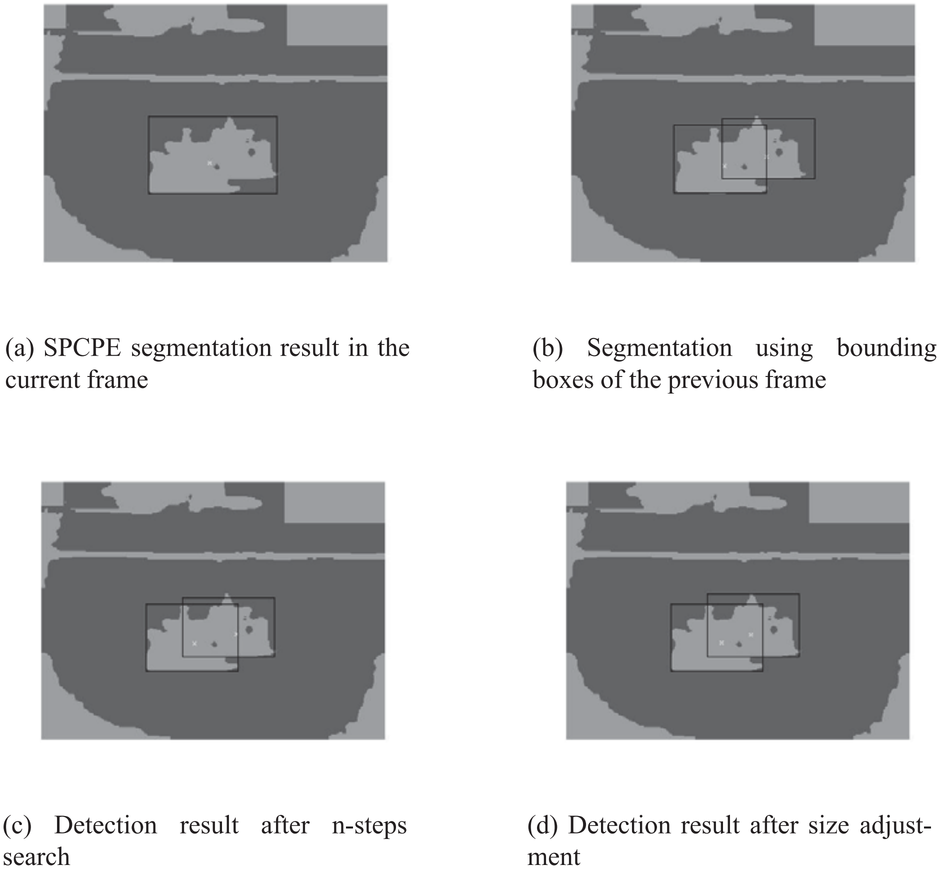


Figure 3.20: Illustration of three-step search

Size Adjustment of Occluded Objects

The positions of the occluded objects are roughly located by the n-steps search as shown in Figure 3.21(c). Since the shapes of the moving objects in a video sequence may change, size adjustment is needed to re-size the bounding boxes of the objects. Unlike the size adjustment method in [39] which used the ratio information of size changes on length and width of the split objects in successive frames to update the bounding box of each object, our proposed size adjustment method uses the contour of the occluded objects in the current frame to re-size the object's bounding box.

Let B_{all} denote the bounding box of the occluded objects (shown in Figure 3.21(a)), and B_{O1} and B_{O2} denote the bounding boxes of the individual objects $O1$ and $O2$ (shown in Figure 3.21(c)). The final bounding boxes of $O1$ and $O2$ are defined as follows. With the size restriction of the bounding box of the occluded objects, our proposed method has the ability of size adjustment as shown in Figure 3.21(d).



(a) SPCPE segmentation result in the current frame

(b) Segmentation using bounding boxes of the previous frame

(c) Detection result after n-steps search

(d) Detection result after size adjustment

Figure 3.21: Detection of occluded objects

$$B'_{O1} = B_{O1} \bigcap B_{all}; \tag{3.9}$$

$$B'_{O2} = B_{O2} \bigcap B_{all}. \tag{3.10}$$

3.3.2 Experimental Results and Analyses

Two video sequences containing object occlusion situations are employed to evaluate the performance of the proposed moving object detection approach. Table 3.4 lists the information of two video sequences used in our experiments. One is from YouTube [88], and the other is from TRECVID 2007 test video collection [79]. Sev-

Table 3.4: Three examples of video sequences

Category	# of frames	Solution	Source
“Ravens” Video	58	360 * 480	YouTube
“Girl on the street” Video	81	288 * 352	TRECVID

eral sample frames in these videos are shown to demonstrate the effectiveness of our proposed approach.

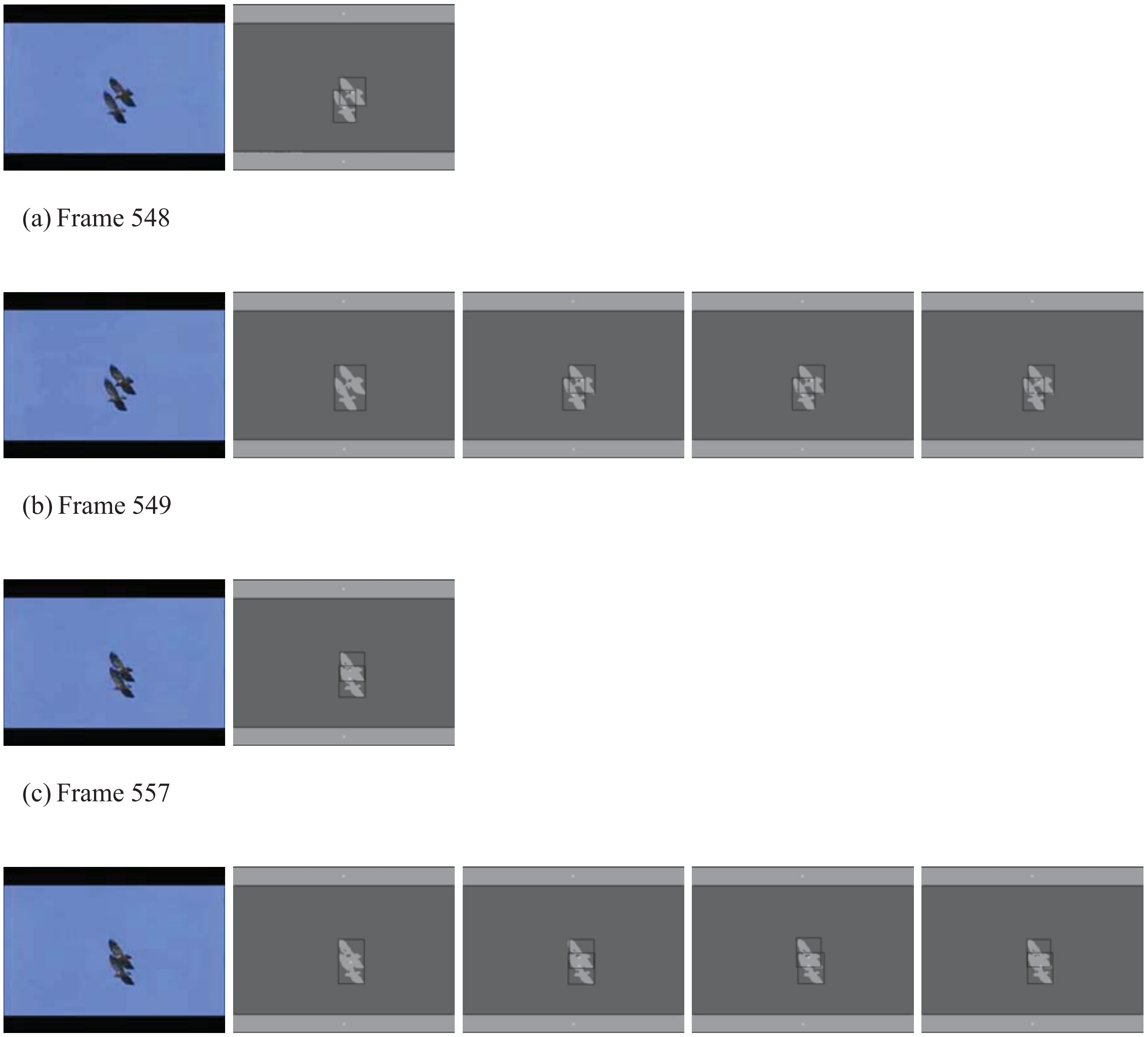
The first column in Figures 3.22 and 3.23 shows the original frames from the video sequences. The second column is the segmentation results of the objects from the background by SPCPE. If there are more than two columns, the third column indicates the positions of the bounding boxes of the previous frame, which are used as the initial searching positions of the NSS method on the current frame. The displacement is set to 10 in the experiment. The searching results are shown in the fourth column, and the fifth column displays the final detection results tuned by the size adjustment method.

Two scenarios are shown in Figure 3.22. One happens at the beginning of the overlapping of two ravens, and one is the severe occlusion. For the first scenario, the occlusion is not significant, and thus it is easier to get good detection result than in the latter scenario. For the second scenario, one object is heavily occluded by another one, resulting in lots of loss of shape and size information. Therefore, the detection result greatly depends on the previous detection result. It can be seen from the split results in Figure 3.22(b) and Figure 3.22(d) that our proposed approach gives a satisfactory performance on both scenarios.

Figure 3.23 gives an example that has complicated spatial relationships between objects in a video. In this video, the girl and curb are overlapped and segmented as one partition initially. Figure 3.23(b)- Figure 3.23(e) give the detection results. It shows that based on the information from Figure 3.23(a), our proposed approach successfully splits the girl from the road curb in the following four consecutive frames.

3.3.3 Conclusions

This section proposes a moving object detection approach utilizing the spatio-temporal information of successive frames in video sequences. It first employs the SPCPE algorithm to estimate the background and foreground of the frames, followed by detecting the object occlusion situations with the help of the size and location information of the bounding boxes in two consecutive frames. Next, the n-steps search and size-adjustment methods are utilized to obtain the preliminary location of each object and tune the size of each object to address the shape changes in the video sequence, respectively. Experimental results on three video sequences with severe object occlusion situations demonstrate that our proposed approach is able to cope with the more generalized object occlusion situations and achieve satisfactory detection results for the moving objects under object occlusion situations.



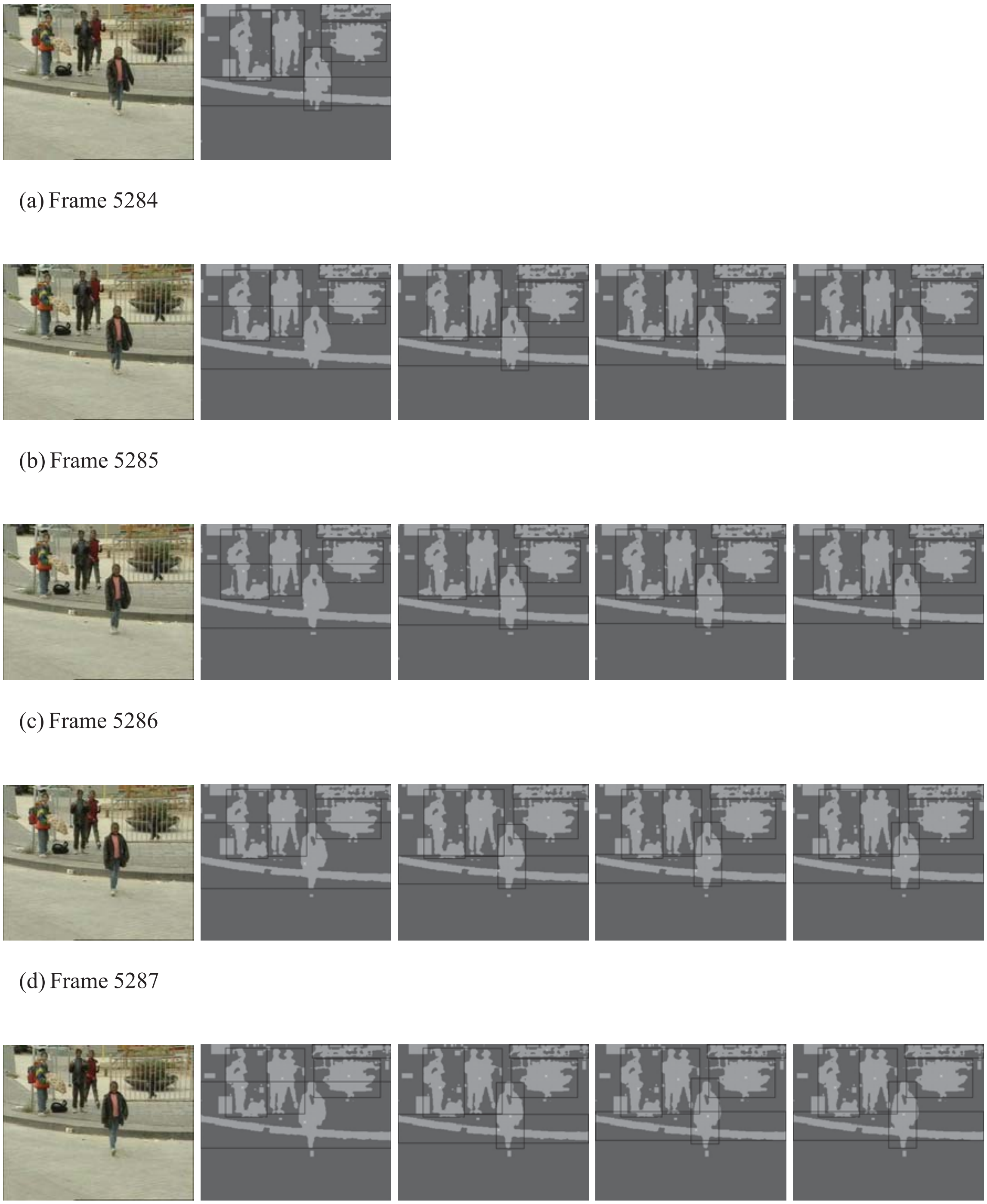
(a) Frame 548

(b) Frame 549

(c) Frame 557

(d) Frame 558

Figure 3.22: Results for the “Ravens” video sequence



(a) Frame 5284

(b) Frame 5285

(c) Frame 5286

(d) Frame 5287

(e) Frame 5288

Figure 3.23: Results for the “Girl on the street” video sequence

Chapter 4

Proposed Detection and Retrieval Framework

4.1 Temporal-Centered Motion Estimation and Recognition in Non-Static Background

In the recent years, video content analysis has been used in a broad range of applications, such as real-time surveillance, activity monitoring, video indexing and retrieval, human-computer interaction, etc. [47, 89, 15]. Various motion detection methods have been proposed in the past decade, but there are seldom attempts to investigate the advantages and disadvantages of different detection mechanisms so that they can complement each other to achieve a better performance. Toward such a demand, this section proposes a human action detection and recognition framework to bridge the semantic gap between low-level pixel intensity change and the high-level understanding of the meaning of an action. To achieve a robust estimation of the region of action with the complexities of an uncontrolled background, we propose the combination of the optical flow field and Harris3D corner detector to obtain a new spatial-temporal estimation in the video sequences. The action detection method, considering the integrated motion information, works well with the dynamic background and camera motion, and demonstrates the advantage of the proposed method of integrating multiple spatial-temporal cues. Then the local features (SIFT and STIP) extracted from the estimated region of action are used to learn the Universal Background Model (UBM) for the action recognition task. The experimental results on KTH and UCF YouTube Action (UCF11) data sets show that the proposed action detection and recognition framework can not only better estimate the region of action but also achieve better recognition accuracy comparing with the peer work.

In this section, we propose a robust action detection and recognition framework that integrates multiple motion detectors and takes the complementary advantages of the motion cues to estimate the region of action [90]. Features extracted from the region are minimally disturbed by scene noise and represent the characteristics of the action. To the best of our knowledge, not much work has been reported on the region detection of action from unconstrained videos in an unsupervised way. In this section, we investigate the ideas of motion detectors and propose a framework that detects region(s) of action by integrating multiple spatial-temporal cues and recognizes actions by using static and motion features on the region of action. The main contributions of this section are summarized as follows.

1. A weighted integration approach is proposed to fuse spatial-temporal information from the optical flow field and the Harris3D detector into a new robust motion representation in the videos.

2. The idea of integral density is utilized to estimate the region of action by using the new motion field. The region of action is defined as the area with a high density of motion.

3. SIFT and STIP features extracted from the region of action are employed to train the universal background model (UBM) for the purpose of action recognition, instead of using the whole feature set. This method is verified to be an effective and efficient way of training recognition model.

To the best of our knowledge, no one has trained UBM by using only action-related features (less than 20% of the whole feature set) and is able to receive a better performance than using the full feature set.

The rest of the section is organized as follows. Section 4.1.1 describes the details of the region of action estimation by integrating multiple spatial-temporal motion fields and quickly locating the high density area of motion. In Section 4.1.2, we present the method of action recognition that uses multiple features from the region of action to train UBM and classifies the actions. The experiments and results of the KTH and UCF11 data sets with discussions are provided in Section 4.1.3. Finally, a conclusion is given in Section 4.1.4.

4.1.1 Moving Object Detection Using Spatio-temporal Information

State-of-the-art motion recognition approaches mainly use the features extracted from the whole frame, no matter the background or the region of motion, to generate the code book which inevitably involves unrelated scene information that may affect the recognition performance. In order to decrease the influence of the background on the motion recognition task, a new motion region estimation method is presented in this section. The proposed algorithm comprehensively analyzes and integrates the motion information from space and time domain in an unsupervised manner, and is robust to non-static scene and camera motion. The motion features extracted from the estimated region of motion are employed to learn the Universal Background Model (UBM) for the motion recognition purposes, which is able to achieve a good performance. The proposed framework is shown in Figure 4.24.

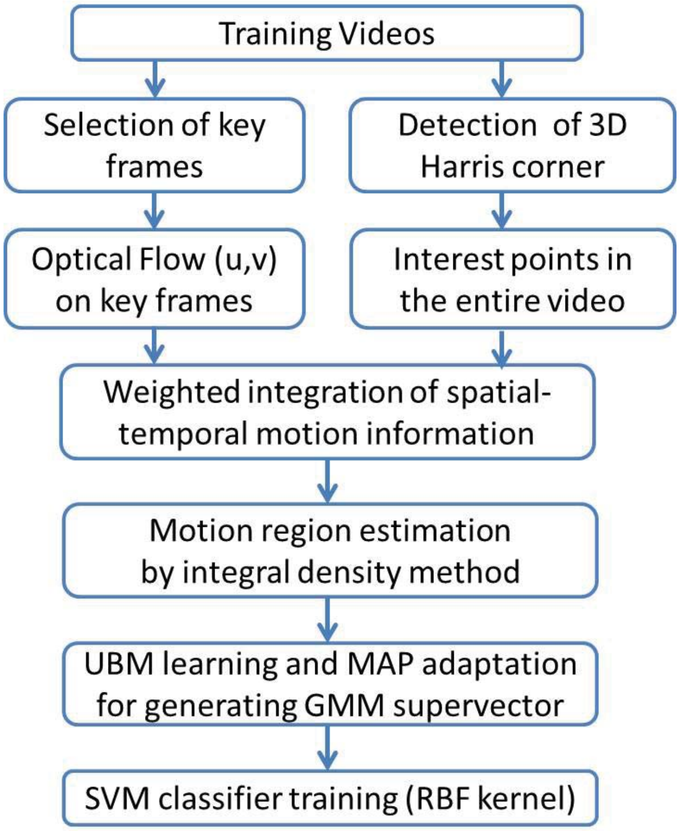

Training Videos
Selection of key frames
Detection of 3D Harris corner
Optical Flow (u,v) on key frames
Interest points in the entire video
Weighted integration of spatial-temporal motion information
Motion region estimation by integral density method
UBM learning and MAP adaptation for generating GMM supervector
SVM classifier training (RBF kernel)

Figure 4.1: Proposed Framework

Biological Motivation

Psychological studies find that a human vision system perceives external features separately [21] and is sensitive to the difference between the target region and its neighborhood. Such kind of high contrast is more likely to attract human's first sight than their surrounding neighbors [22]. Extensive psychophysics experiments have shown that these mechanisms can be driven by a variety of features, including intensity, color, orientation, or motion, and local feature contrast plays a predominant role in the perception of saliency. Neurophysiological experiments on primates have also shown that neurons in the middle temporal (MT) visual area compute local motion contrast with center-surround mechanisms. In fact, it has been hypothesized that such neurons underlie the perception of motion pop-out and figure-ground segmentation [23]. The center-surround saliency mechanisms of biological systems support the idea of motion region estimation on measurements of local motion contrast. There is no need for training samples or pre-build a "global background model" for the testing instances, which is one of the advantages of the proposed method. Instead, a motion region can be efficiently calculated using merely local motion information and could immediately adapt to different kinds of unknown scenes. Also, using local motion contrast could make the model robust to the camera motion and dynamic background.

Apparent Motion Descriptor - Optical Flow

Optical flow is the pattern of motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. In 1981, Horn and Schunck [91, 92] deduced a basic equation of optical flow estimation when the interval of consecutive frames was short, and the gray change in the image was also small. If at time t, the coordinates of a pixel on the image with its gray value is $I(x,y,t)$, and at time $(t+\triangle t)$, the pixel has moved to new position, its location on the image becomes $(x+\triangle x, y+\triangle y)$, and the gray value becomes $I(x+\triangle x, y+\triangle y, t+\triangle t)$. $dI(x,y,t)/dt = 0$ is obtained based on the assumption that intensity is conserved. Then the equation can be re-written as $I(x,y,t) = I(x+\triangle x, y+\triangle y, t+\triangle t)$, whose Tayor expansion can be used to derive the gradient constraint equation as below.

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0.$$

Suppose u and v are two components of the optical flow along the x coordinate and y coordinate, and they are defined as $u = dx/dt$, $v = dy/dt$. Then the basic optical flow equation is obtained as $I_x u + I_y v + I_t = 0$, where I_x denotes the partial x coordinate derivative of $I(x,y,t)$, I_y denotes the partial y coordinate derivative of $I(x,y,t)$, and I_t denotes the partial time derivative of $I(x,y,t)$.

The advantage of using the optical flow is that it does not require any priori knowledge on the object appearance which satisfies the requirement of an unsupervised method in this section. The disadvantage is that the computation is usually too complex to be used in real-time applications if there is no special hardware support. With the attempt to reduce the computation complexity of the optical flow technique, the motion vector idea using the optical flow technique to work on the block-level instead of pixel-level motion is adopted.

Motion vector is an integral part of many video compression algorithms which are used for motion compensation. The idea behind block matching is to divide the current frame into a matrix of blocks that are then compared with the corresponding block and its neighbors in the previous frame to determine a motion vector that estimates the movement of a block from one frame to another. For fast motion estimation purposes, we employ the optical flow method to describe the spatial motion of blocks in the frame.

Harris3D Corner Detector

If the video sequences are captured by a moving camera or in a non-static background, no satisfactory results can be obtained by simply relying on the motion described by optical flow to estimate the motion region. Thus, in our proposed framework, the space time interest point detector, Harris3D corner detector [93], is employed to integrate the motion presented by optical flow. The Harris3D corner detector is used to detect the spatial-temporal corners with velocity changes over a sequence of frames.

We consider a 3D window at a space-time point $I(x,y,t)$ and analyze the average intensity change (gradient) as the window is shifted by a small amount (σ, τ) in spatial as well as temporal dimensions (σ is the spatial scale and τ is the temporal scale). The space-time gradient is obtained as $\nabla L = (L_x, L_y, L_t)^T$. The interest point is identified by evaluating the distribution of ∇L within a local neighborhood. The matrix μ of the second moments measures the variation of the gradients. μ is a 3-by-3 matrix composed of the first order spatial and temporal derivatives being averaged using a Gaussian weighting function $g(\cdot; \sigma_i^2, \tau_i^2)$. A high variation of ∇L implies large eigenvalues of μ, and the spatial-temporal corners are obtained from the local maxima of H over $I(x,y,t)$.

$$H = det(\mu) - k \cdot trace^3(\mu) = \lambda_1\lambda_2\lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3,$$

where λs are the eigenvalues of H and k is a constant with a value close to 0.15.

Integrated Spatial-Temporal Motion

The above discussion shows that the optical flow field and Harris3D corner detector have their individual characteristics in the spatial-temporal motion calculation. The integration of these two sources of motion information may provide the complementary motion information to improve the region of motion estimation.

Suppose N key frames are sampled from a motion video sequence, and $N-1$ optical flow fields are generated. Spatial-temporal volumes created around the Harris3D corners are illustrated in gray boxes in Figure 4.2. All volumes are clustered into $N-1$ groups based on the time stamp of the key frames. As shown in Figure 4.2, if the center of the volume is between $[n-0.5, n+1.5]$, the volume belongs to group (n, $n+1$). The histogram of Harris3D volumes of group (n, $n+1$) is then generated based on the distribution of the volumes along the time line. The new motion $M(x,y)$ at pixel (x,y) between key frames n and $n+1$ is calculated as given in Equation (4.1).

$$M(x,y) = O(x,y) * H(x,y) \tag{4.1}$$

where $O(x,y)$ is the the motion vector of optical flow and $H(x,y)$ is the histogram of Harris3D volumes at pixel (x,y). This method is also viewed as a weighted optical flow approach which uses the histogram of Harris3D volumes to weigh the optical flow field. In this way, two sources of motion information are integrated in terms of the key frames.

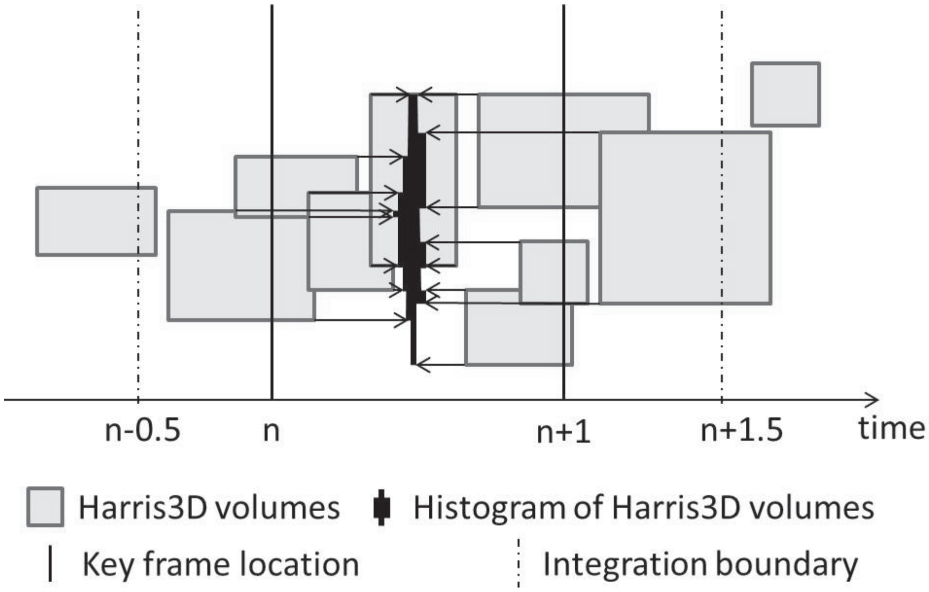


Figure 4.2: Illustration of the histogram of Harris3D volumes

Region of Motion Estimation

An unsupervised motion region estimation method is proposed in this section by analyzing the new motion field generated from the integration of optical flow and the histogram of Harris3D volumes. The idea of the integral density, as defined in [94], is adopted in the method since it allows fast implementation of the box type convolution filters. The entry of a summed area table $I_{\Sigma(\mathbf{x})}$ at a location $\mathbf{x}$=(x,y) represents the sum of all values in the input $2D$ matrix I of a rectangular region formed by the point $\mathbf{x}$ and the origin, i.e.,

$$I_{\Sigma(\mathbf{x})} = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j).$$

With $I_{\Sigma(\mathbf{x})}$ calculated, it takes only four additions to calculate the sum of the values over any upright rectangular areas, independent of their sizes. In the same way, the maximal motion region is identified as the region of motion. We define the maximal motion region as an area having the highest motion density as shown below, where $v(x,y)$ indicates the integrated motion at pixel (x,y).

$$\arg\max_{R} \int\int_{R} v(x,y)dxdy.$$

4.1.2 Moving Object Recognition Framework

Gaussian Mixture Models (GMM) are employed in our proposed framework, whose probability density function (pdf) is given by $p(x|\theta) = \sum_{k=1}^{K} \omega_k N(x|\mu_k, \Sigma_k)$, where K is the number of Gaussian mixtures, and $\theta = \{\omega_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ is a set of parameters including a mixing coefficient ω_k and a pdf of Gaussian distribution $N(x|\mu_k, \Sigma_k)$ with the mean vector μ_k and the variance matrix $\sum_k$. The GMM parameters are estimated by an expectation maximization (EM) algorithm. The EM algorithm is known as a method for finding the maximum likelihood of a model with latent variables.

SIFT [95] and STIP [93] features are used to describe the action video sequences in the action recognition. However, the number of features (SIFT or STIP) from a single video is not enough to estimate the GMM parameters precisely. Thus, we first learn a global GMM (called Universal Background Model (UBM)) by using the features from all training videos, then adapt the UBM parameters in order to fit each particular data distribution. This adaptation is made by using the Maximum A Posteriori (MAP) approach [73]. The first step consists of determining the probabilistic alignment of the training vectors with the UBM Gaussian components. For a Gaussian component k in the UBM, we compute:

$$\begin{aligned} \mathrm{P}r(k,x_t) &= \frac{\omega_k p_k(x_t)}{\sum_{k=1}^{K} \omega_k p_k(x_t)}, \\ n_k &= \sum_{t=1}^{T} \mathrm{P}r(k,x_t), \\ E_k(x) &= \frac{1}{n_k} \sum_{t=1}^{T} \mathrm{P}r(k,x_t) x_t, \end{aligned}$$

where, X_t represents the tth feature vector of the video to be modeled. These statistical values are then used for adapting the mean vector $\hat{\mu}$ of each Gaussian.

$$\begin{aligned} \hat{\mu}_k &= \alpha_k E_k(x) + (1-\alpha_k)\mu_k; \\ \alpha_k &= \frac{n_k}{n_k + r}, \end{aligned}$$

where r is a fixed “relevance factor”, usually set between 8 and 20 [75]. The concatenation of all the mean vectors of the N Gaussian components is called the GMM supervector which was first proposed as a speaker recognition method [74] and then has been applied to semantic indexing [69] and music similarity [75]. Knowing the parameter of the UBM, a particular video model can be resumed by the mean vectors of its Gaussian mixture components. The testing videos are classified by using the support vector machines (SVMs) with the RBF kernel [96]. In this section, to save UBM training time, only features extracted from each region of action are used to model the overall data distribution.

4.1.3 Experimental Results on KTH and UCF11 Data Sets

The detection and recognition experiments were conducted on the KTH and UCF Youtube Action (UCF11) data sets. The KTH data set has 25 actors performing six actions four times in four different environments, resulting in 599 video sequences in total. The video sequences were recorded in a controlled setting with slight camera motion and a simple background. The six categories of actions are boxing, hand clapping, hand waving, jogging, walking, and running [97]. UCF Youtube Action (UCF11) data set is more challenging than the KTH data set since it includes $1,168$ videos and has 11 categories of actions collected from YouTube with non-static background, low quality, camera motions, poor illumination conditions, etc. [1].

Experimental Setup

For the interest point detection, the difference of Gaussians edge detection method proposed by Lowe [95] and the Harris3D corner detector proposed by Laptev [93] are used to locate the interest points of SIFT and STIP, respectively. Three key frames were equally sampled along the video for SIFT feature extraction and optical flow computation. The dimensions of the SIFT and STIP features are reduced to 32 by applying the Principle Component Analysis [98] from 128 and 162, respectively. The number of Gaussian components in GMM (i.e., K) is set to 256 for the KTH and UCF11 data sets. For classifier selection, we have many choices, such as [99, 100, 101, 102, 103]. In the experiment, support vector machine (SVM) is used to cope with the multi-class classification task due to its robustness to different data sets. We adopt the empirical setting in libSVM [104], and for comparison purposes, the leave one out cross validation (LOOCV) scheme is employed to compare with some existing approaches.

Experimental Results on the KTH Data Set

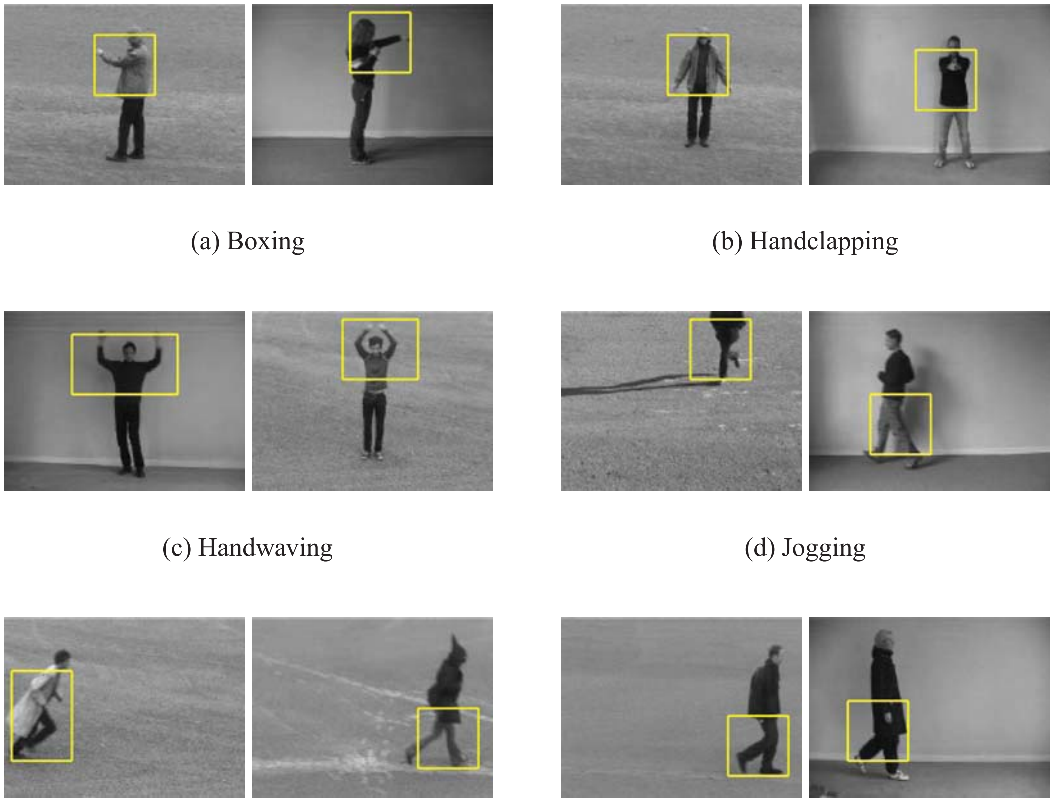
Though the proposed framework is mainly designed to deal with videos captured in unconstrained environments, it is also proved to achieve pretty good performance in videos recorded in a “clean” background, such as the KTH data set. First, the accurate

Table 4.1: Accuracy comparison on the KTH data set (%)

Algorithm	Accuracy (%)
Proposed framework	**93.67**
Reddy *et al.* [41]	89.79
Dollar *et al.* [105]	81.2
Liu *et al.* [106]	91.3
Wong *et al.* [107]	83.9
Laptev *et al.* [3]	91.8

localization of motion is verified. Sample regions of motion estimation results are illustrated in Figure 4.3. The features extracted from the regions of motion were used to learn UBM and a classification accuracy of 93.67% was obtained if combining the SIFT and STIP similarity scores, whereas the accuracy was 84.65% if using the SIFT features alone and was 90.65% if using the STIP features alone. The combination of two kinds of features achieved 3% improvement in the performance. Table 4.1 lists several state-of-the-art performance results on the KTH data set, and indicates that our proposed framework outperforms the peer work. Of note, the amount of features used to train UBM is less than 15% of the total features over all video sequences, which clearly shows to reduce lots of offline training time.

Figure 4.4 shows the confusion table containing the detailed confusion values between action categories. Based on the moving part of a person, the six action categories can be grouped into limb action (boxing, hand clapping, and hand waving) and leg action (jogging, running, and walking). Of note, the confusion happens either within limb action or leg action videos. From the figure, it can be seen that no limb action is misclassified as leg action, and vice versa. This indicates our proposed framework is reasonably good.



(a) Boxing

(b) Handclapping

(c) Handwaving

(d) Jogging

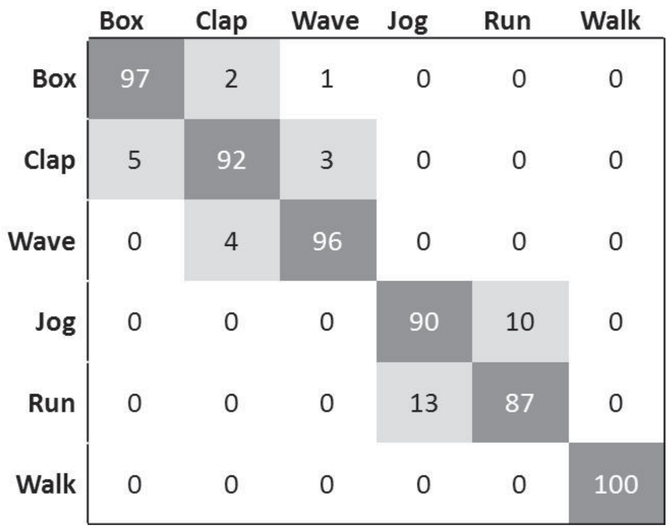
(e) Running

(f) Walking

Figure 4.3: Region of motion detection results on sample frames in the KTH data set.

Experimental Results on the UCF11 Data Set

The UCF11 data set is more challenging than the KTH data set, since it contains realistic actions, camera motions, and complicated backgrounds. Figures 4.5 - 4.7 illustrates some sample results of motion region estimation of the proposed framework (on the left of each sub-figure) and felzenszwalb's part-based models (on the right of each sub-figure) [4]. The codes we used to conduct felzenszwalb's algorithm were downloaded from [108]. Note that felzenszwalb's method works well with human vertical positions in simple backgrounds, such as in Figures 4.5(d), 4.6(a) and 4.7(a). Since the method does not consider temporal information, it may fail in cluttered scenes such as in the first

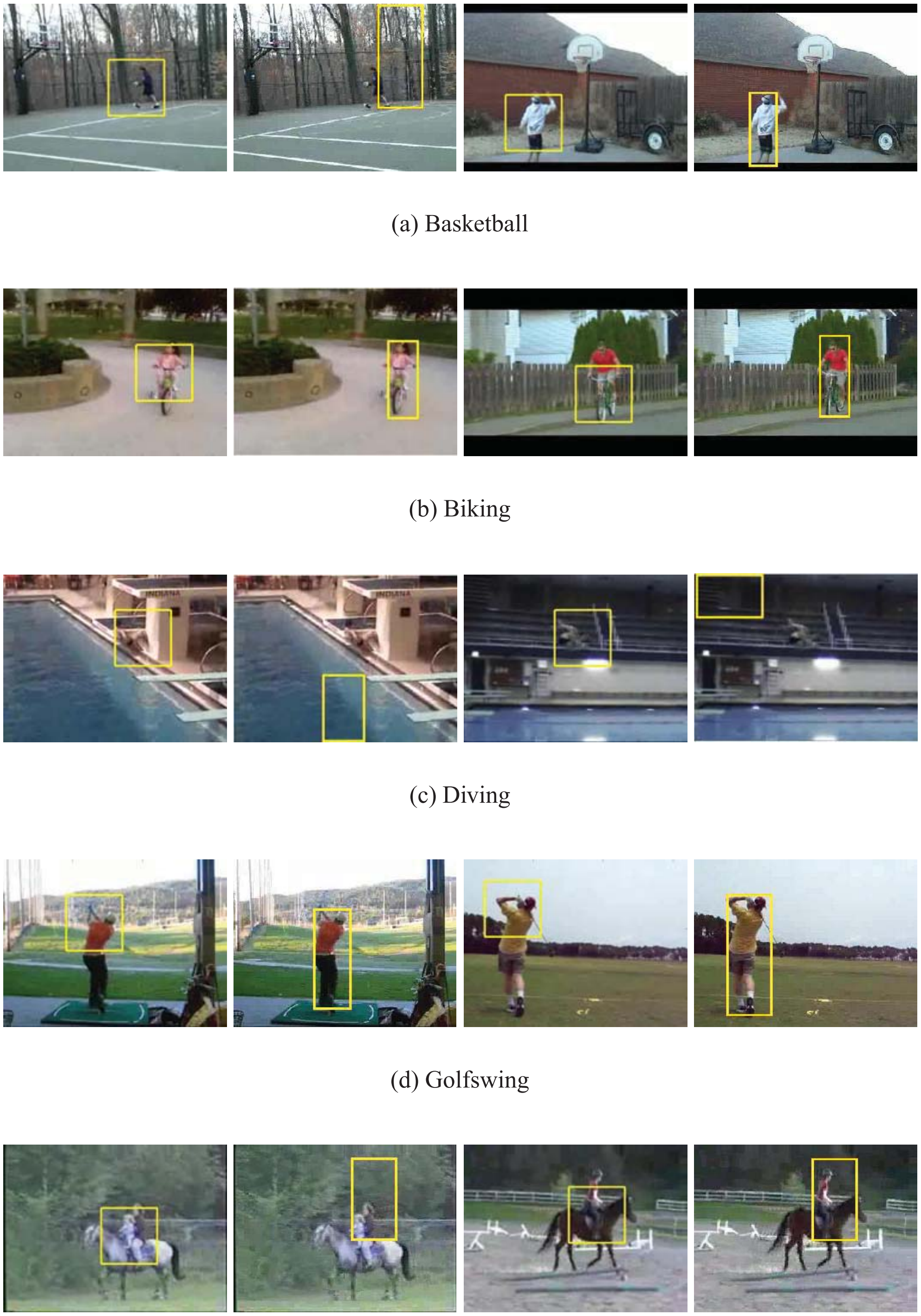


	Box	Clap	Wave	Jog	Run	Walk
Box	97	2	1	0	0	0
Clap	5	92	3	0	0	0
Wave	0	4	96	0	0	0
Jog	0	0	0	90	10	0
Run	0	0	0	13	87	0
Walk	0	0	0	0	0	100

Figure 4.4: Confusion matrix of 6 action categories on the KTH data set with an average performance of 93.67%.

example of Figure 4.5(a) which has a lot of trees having similar appearances with the person. In contrast, since our proposed framework is unsupervised, it could effectively locate the region of motion without many appearance constraints obtained from the training data. Furthermore, our proposed framework is motion-driven so it is more suitable for motion detection which includes the interaction of humans and objects like biking (Figure 4.5(b)), horse riding (Figure 4.5(e)), etc.

In addition, unlike previous approaches that use all features in the videos (extracted from the scene and object), we use only those features extracted from the region of motion to train UBM, which significantly reduces the training time. In this experiment, the motion-related features are about 20% of the full feature set (scene and object), but our proposed framework achieves better performance than the previous approaches which use a full set of features. Of note, our proposed framework achieves the performance of 76.06% (as presented in Figure 4.8) after fusing the similarity scores of SIFT and STIP; while the performance obtained from the SIFT descriptor alone is 55.85% and that from



(a) Basketball

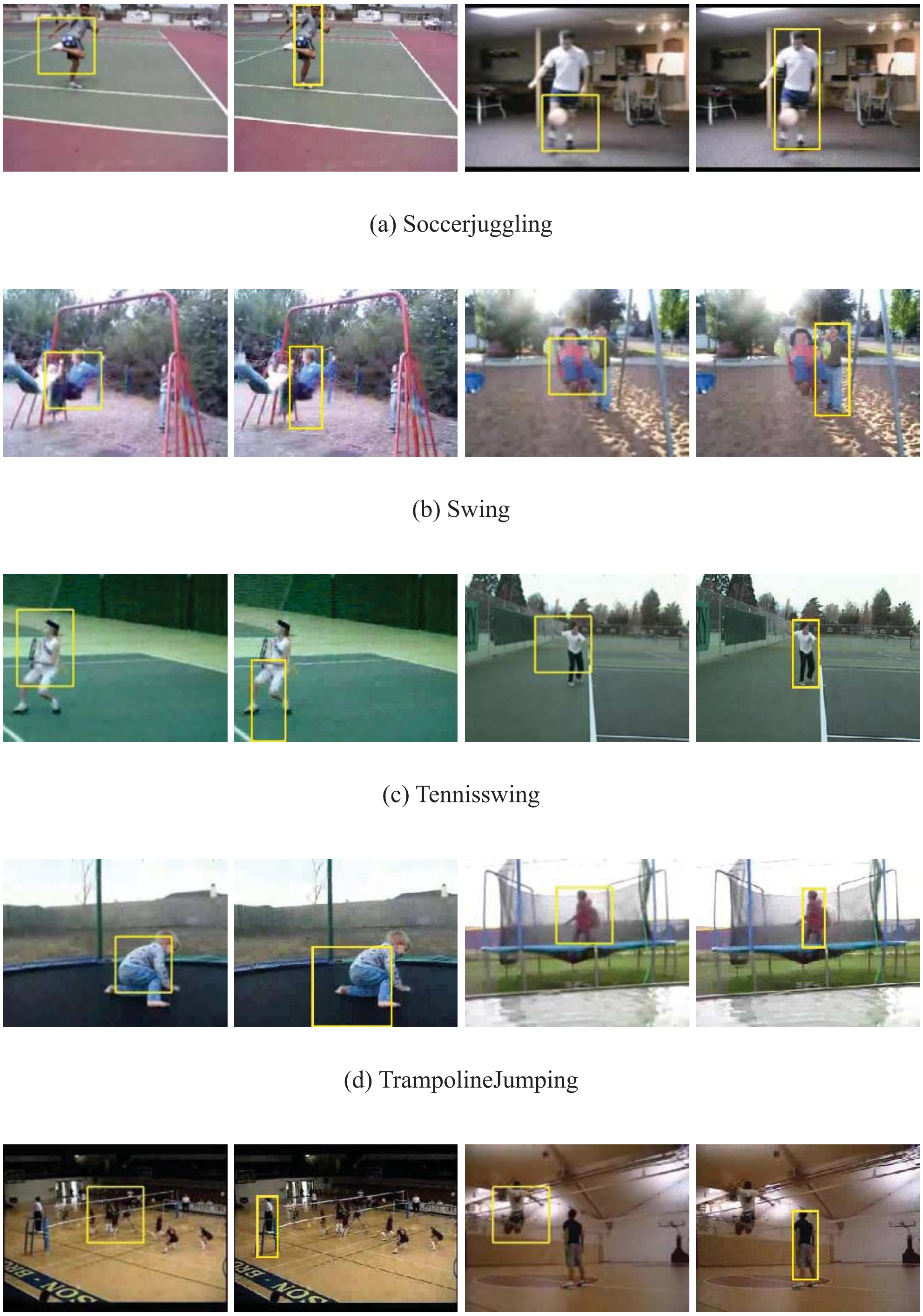
(b) Biking

(c) Diving

(d) Golfswing

(e) Horseriding

Figure 4.5: Sample results of the proposed region of motion detection method (left) and felzenszwalb's part-based models [4] (right) in UCF11 data set.



(a) Soccerjuggling

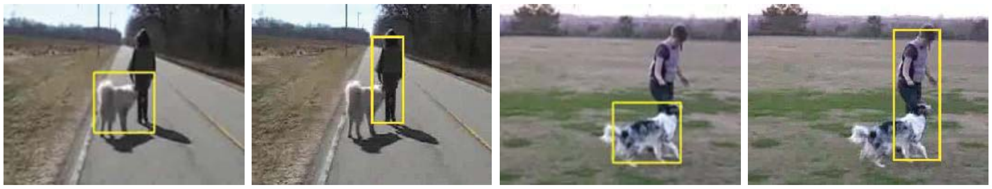
(b) Swing

(c) Tennisswing

(d) TrampolineJumping

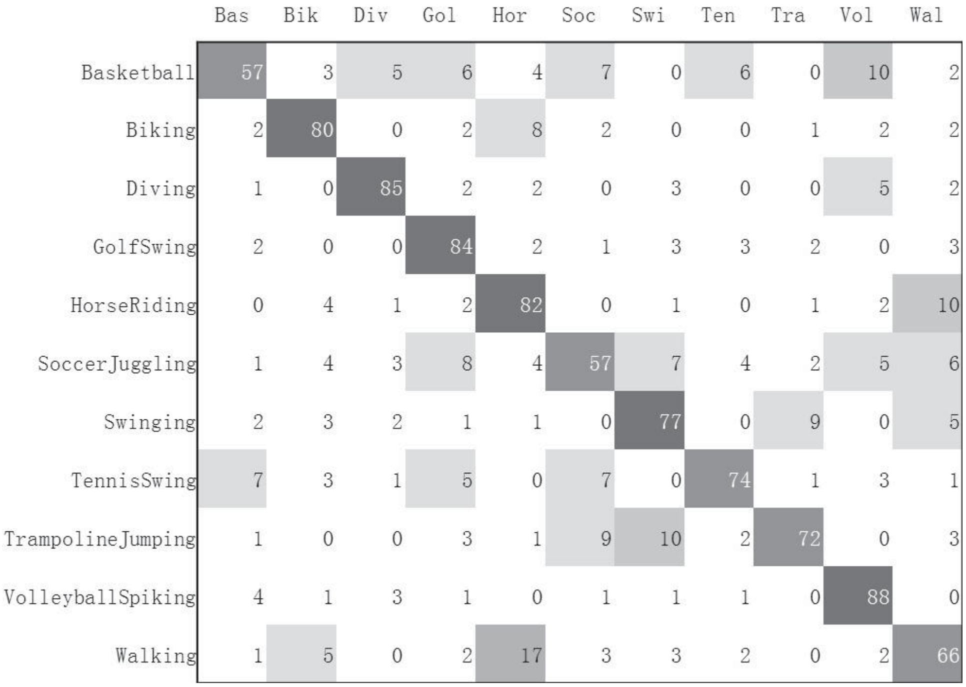
(e) VolleyballSpiking

Figure 4.6: Sample results of the proposed region of motion detection method (left) and felzenszwalb's part-based models [4] (right) in UCF11 data set.



(a) Walkingwithdog

Figure 4.7: Sample results of the proposed region of motion detection method (left) and felzenszwalb's part-based models [4] (right) in UCF11 data set.



	Bas	Bik	Div	Gol	Hor	Soc	Swi	Ten	Tra	Vol	Wal
Basketball	57	3	5	6	4	7	0	6	0	10	2
Biking	2	80	0	2	8	2	0	0	1	2	2
Diving	1	0	85	2	2	0	3	0	0	5	2
GolfSwing	2	0	0	84	2	1	3	3	2	0	3
HorseRiding	0	4	1	2	82	0	1	0	1	2	10
SoccerJuggling	1	4	3	8	4	57	7	4	2	5	6
Swinging	2	3	2	1	1	0	77	0	9	0	5
TennisSwing	7	3	1	5	0	7	0	74	1	3	1
TrampolineJumping	1	0	0	3	1	9	10	2	72	0	3
VolleyballSpiking	4	1	3	1	0	1	1	1	0	88	0
Walking	1	5	0	2	17	3	3	2	0	2	66

Figure 4.8: Confusion matrix of 11 action categories on the UCF data set with an average performance of 76.06%.

the STIP descriptor alone is 72.82%. In our proposed framework, the size of the code book used in the experiments is only 256, which is relatively smaller than those in the state-of-the-art work, and it also achieves good performances. This demonstrates that the features are extracted from the correct regions of motions and can describe the class-related information. The recognition performance reported by Liu *et al*. is 71.2% using hybrid features obtained by pruning the motion and static features [1]. Another similar work that split the moving foreground from the static background and then combined the motion and the scene context features obtained 73.2% [41].

4.1.4 Conclusions

In this section, a new motion detection and recognition framework that integrates the spatial-temporal motion obtained from the optical flow field and the Harris3D corner detector is proposed. It is motivated by taking the advantages of the two sources of motion information identified by different methods to obtain the complementary motion information which is kept in the new motion representation. A fast region of motion estimation method is also proposed by using the integral density algorithm. The SIFT and STIP features extracted from the regions are used to learn UBM for the motion recognition proposes. The experimental results verify that the proposed framework achieves good performance on both motion detection and recognition tasks.

4.2 Spatial-Centered Motion Estimation and Retrieval in Unconstrained Environment

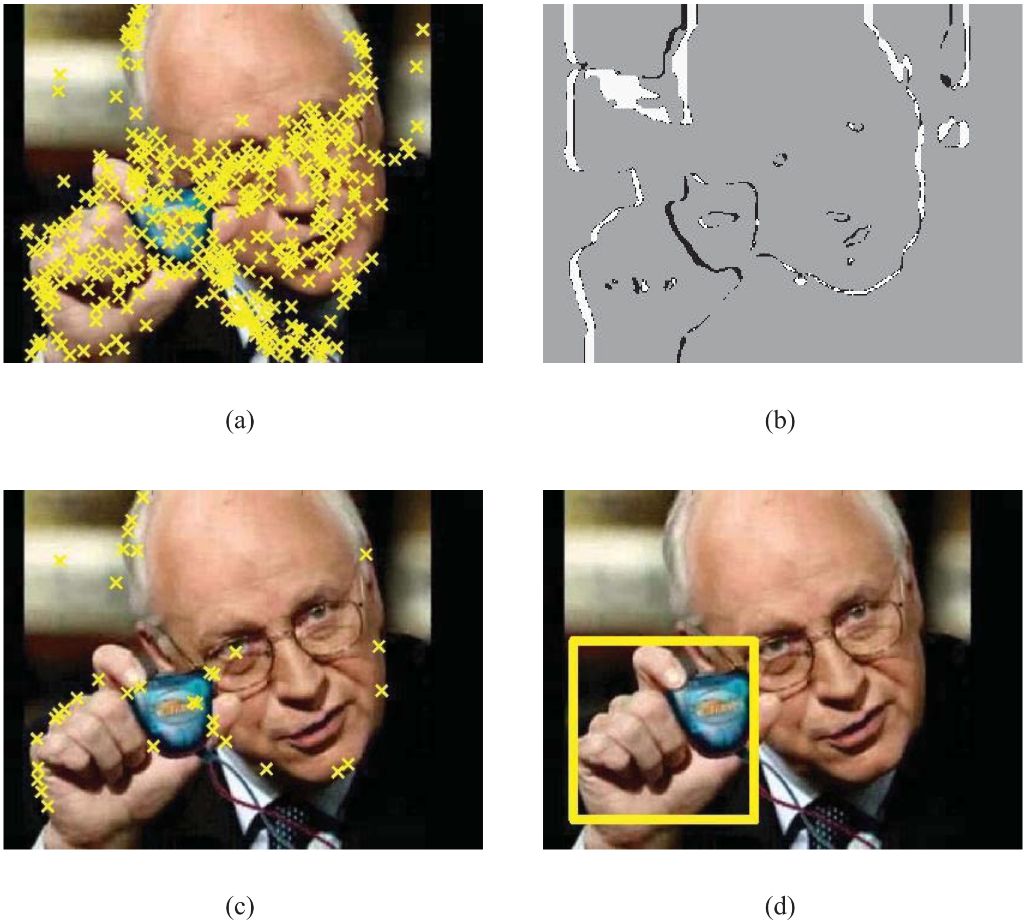
In the area of multimedia semantic analysis and video retrieval, automatic object detection techniques play an important role. Without the analysis of the object-level features, it is hard to achieve high performance on semantic retrieval. As a branch of object detection study, moving object detection also becomes a hot research field and gets a great amount of progress recently. For example, the analysis of vehicle make and model needs the vehicle-level information from the video sequence [109, 110]. This section proposes a moving object detection and retrieval model that integrates the spatial and temporal information in video sequences and uses the proposed integral density method to quickly identify the motion regions in an unsupervised way. First, key information locations on video frames are achieved as maxima and minima of the result of Difference of Gaussians algorithm. On the other hand, a motion map of adjacent frames is obtained from the diversity of the outcomes from Simultaneous Partition and Class Parameter Estimation (SPCPE) framework. The motion map filters key information locations into key motion locations (KMLs) where the existence of moving objects is implied. Besides showing the motion zones, the motion map also indicates the motion direction which guides the proposed "integral density" approach to quickly and accurately locate the motion regions. The detection results are not only illustrated visually, but also verified by the promising experimental results which show that the concept retrieval performance can be improved by integrating the global and local visual information [94].

With the rapid advances of Internet and Web 2.0, the amount of online multimedia data increases in an explosive speed, which brings many challenges to data retrieval, browsing, searching and categorization [9, 10, 11]. Manual annotation obviously cannot catch up the speed of increasing multimedia data, so content-based video processing approaches are developed to quickly and automatically identify the semantic concepts and annotate the video sequences [12, 16, 13, 14, 15].

Automatic object detection techniques, as a key step in content-based multimedia data analysis framework, has also attracted lots of attention these years. It aims to segment a visual frame into a set of semantic regions, each of which corresponds to an object that is meaningful to the human vision system, such as a car, a person, a tree, etc. When the object detection issues move from image area to video domain, temporal information in video sequences brings moving object-level information which can be utilized for moving object detection. From this perspective, this section integrates spatial information locations (the yellow crosses shown in Figure 4.9(a)) and temporal motion cues (the white and black zones shown in Figure 4.9(b)) to find the locations that are rich in spatial-temporal information (the yellow crosses shown in Figure 4.9(c)) and uses integral density method to identify the motion region (the yellow bounding box shown in Figure 4.9(d)). The motion region is also verified to be helpful to improve the content-based multimedia retrieval performance.

Psychological studies find that a human vision system perceives external features separately [21] and is sensitive to the difference between the target region and its neighborhood. Such kind of high contrast is more likely to attract human's first sight than their surrounding neighbors [22]. Following this finding, many approaches have focused on the detection of feature contrasts to trigger human vision nerves. This research field is usually called visual attention detection or salient object detection. Liu, et al. [111] employed a conditional random field method which was learned to effectively combine multiple features (including multi-scale contrast, center-surround histogram, and color spatial distribution) for salient object detection.

The main contributions of this section include: (1) Define a motion map based on the segmentation results of Simultaneous Partition and Class Parameter Estimation (SPCPE) [76] and identify key motion locations (KMLs) by filtering key information locations via the motion map. The motion map not only shows the motion areas, but also indicates the moving direction of the objects which help the identification of the moving objects later. (2) Propose an integral density method inspired by the idea of integral



(a)

(b)

(c)

(d)

Figure 4.9: Illustration of moving object detection. (a) Key information locations (yellow crosses) on a sample image. (b) Motion map of the sample image (white and black zones). (c) Key motion locations (KMLs) on the motion map. (d) Motion region detected by the proposed integral density method from (c).

image in order to quickly and accurately detect the moving object regions from KMLs. (3) Present a multimedia retrieval framework to integrate global and local features in order to enhance the existing retrieval framework that uses only global features [112].

The remainder of this section is organized as follows. The moving object detection framework is presented in Section 4.2.1. Section 4.2.2 describes the proposed moving object detection and retrieval model that fuses the global and local features to enhance the retrieval performance. The new content-based multimedia retrieval framework is also introduced in this section. Section 4.2.3 presents the experimental results and analyzes the performance from the detection and retrieval angles, respectively. Section 4.2.4 concludes the proposed moving object detection and retrieval model.

4.2.1 Moving Object Detection Framework

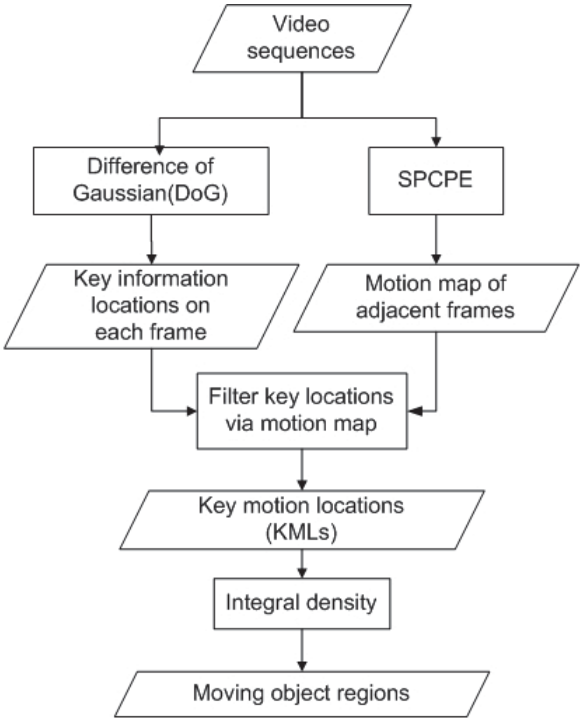

Video sequences
Difference of Gaussian(DoG)
SPCPE
Key information locations on each frame
Motion map of adjacent frames
Filter key locations via motion map
Key motion locations (KMLs)
Integral density
Moving object regions

Figure 4.10: The proposed moving object detection framework

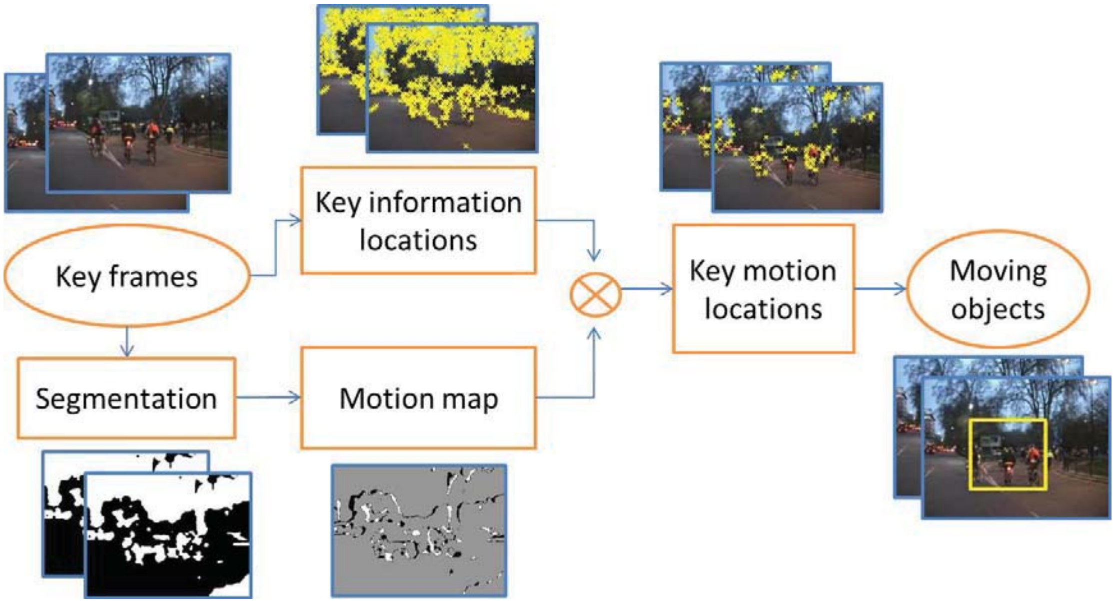


Figure 4.11: An example of the detection flowchart

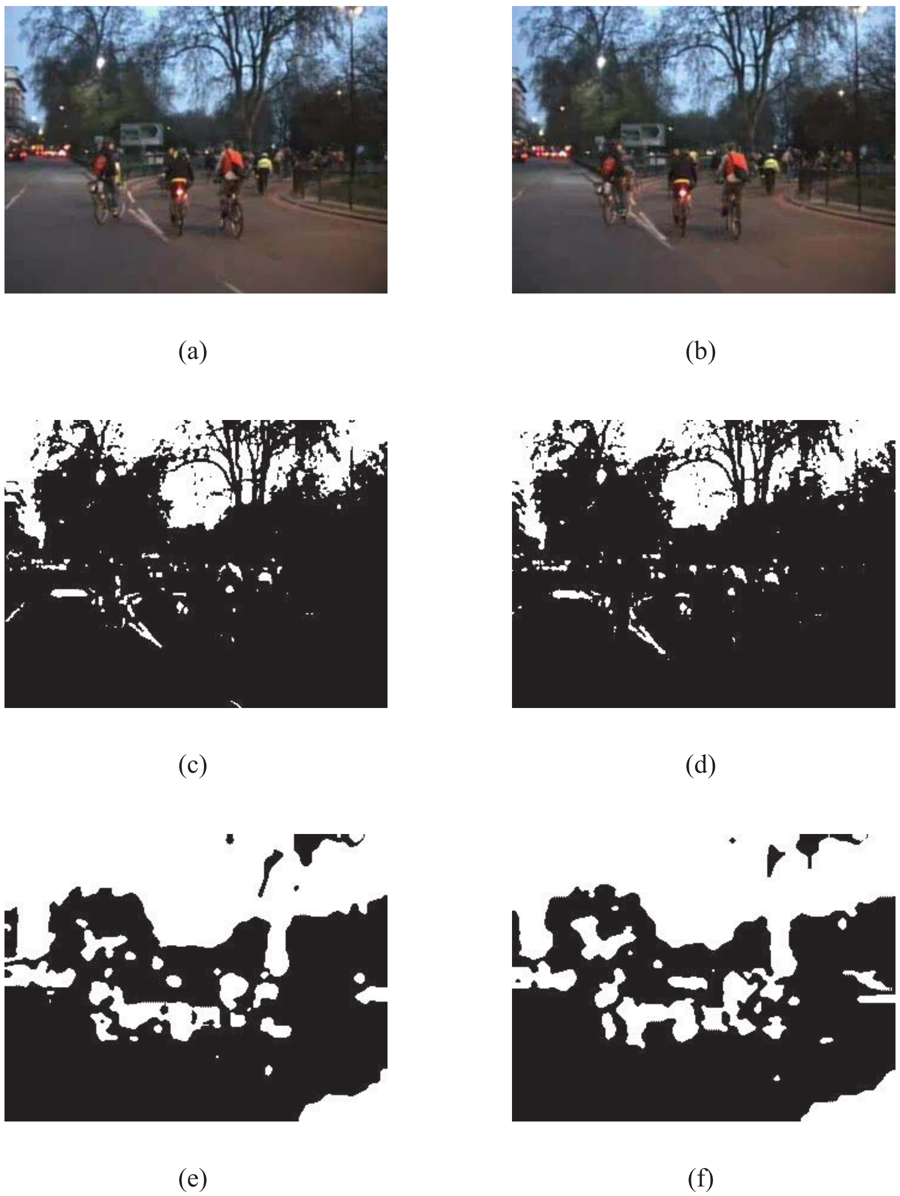
In the motion detection field, the optical flow method is commonly used to compute motion contrast between visual pixels. However, it has obvious drawbacks. For instance, when multiple motion layers exist in the scene, optical flows at the edge pixels are noisy. Also, in texture-less regions, optical flows may return error values. To address these drawbacks, instead of using the above pixel-wise computations, we employ an unsupervised object segmentation method called SPCPE (Simultaneous Partition and Class Parameter Estimation) to segment the frame approximately, and then compute the difference between the two frame segments whose results are called the "motion map" in this section. This motion information is used to filter the key information locations obtained from the result of difference of Gaussians algorithm applied in scale space to a series of smoothed and re-sampled videos frames [95]. Finally, the integral density method is utilized to identify those regions as the moving objects where the KMLs is high. Figure 4.10 shows the flowchart of our proposed moving object detection framework. Figure 4.11 illustrates a detailed example of the detection flowchart.

Motion Map Generation

We aim to separate the moving objects from the relatively static background in an unsupervised manner or a bottom-up approach. Unlike those top-down approaches which are task-driven and need to know the prior knowledge of the target, bottom-up approaches are referred to as the stimuli-driven mechanism which is based on the human reaction to the external stimuli (for example, the prominent motion from the surroundings).

As shown in the third row of Figure 4.12, the pixels in the video frames are segmented into two classes by using the SPCPE algorithm. It starts with an arbitrary class partition and then an iterative process is employed to jointly estimate the class partition and its corresponding class parameters. The iteration is terminated when the areas of the two classes are stable. Assume that the content of the adjacent frames in a video sequence does not change much (as shown in Figures 4.12(a) and 4.12(b)), and thus the estimation result of the two classes of successive frames does not differ a lot as shown in Figures 4.12(e) and 4.12(f). Under this assumption, the segmentation of the previous frame is used as an initial class partition for the next frame, so the number of iterations for processing is significantly decreased.

Though the contours of the objects are not very precise as shown in Figures 4.12(e) and 4.12(f), the segmentation is considered to reflect the object information in the frame. Even though using binary images as shown in Figures 4.12(c) and 4.12(d) can in some degrees represent object information, the difference of binary images shown in Figure 4.13(a) contains too much noise so that it fails to give the motion cues of moving objects as the difference of the SPCPE results does in Figure 4.13(b). Assume the white regions and black regions in Figures 4.12(e) and 4.12(f) stand for class 1 and class 2, respectively. The gray area in Figure 4.13(b) shows the pixels which do

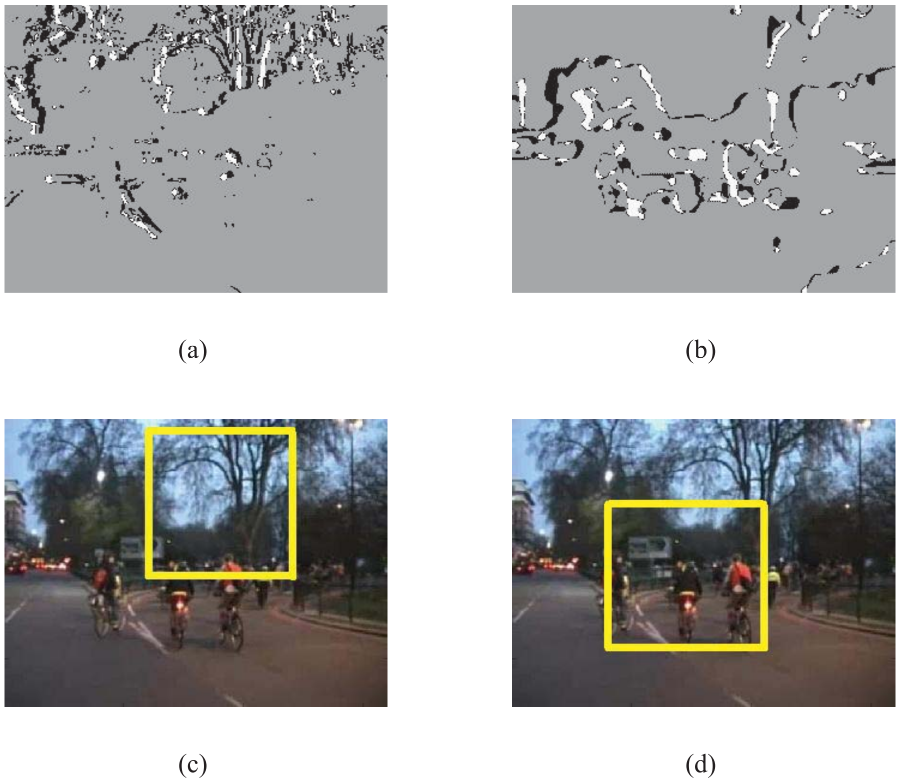


(a) (b)

(c) (d)

(e) (f)

Figure 4.12: Comparison of the binary images and SPCPE segmentation results of sample frames. (a) and (b) are adjacent frames containing trees, roads, and bicycling persons. (c) and (d) are binary images converted from (a) and (b). (e) and (f) are the two-class SPCPE segmentation results of (a) and (b).



(a)

(b)

(c)

(d)

Figure 4.13: Comparison of the motion maps and corresponding moving object detection results. (a) is the motion map generated from Figures 4.12(c) and 4.12(d). (b) is the motion map created from Figures 4.12(e) and 4.12(f). (c) and (d) are detection results by using (a) and (b).

not change the class labels from Figures 4.12(e) to 4.12(f). The white zones in Figure 4.13(b) show those pixels which change from class 1 to class 2, and the black zones show those pixels which change from class 2 to class 1. Obviously, these white and black zones contain the contour information of the moving objects and background, as well as the moving direction information of the objects. Thus we define the white and black zones in Figure 4.13(b) as the motion map of Figures 4.12(e) and 4.12(f). Figures 4.13(c) and 4.13(d) are the motion detection results by using different motion maps (Figures 4.13(a) and 4.13(b)), respectively. It shows that SPCPE aims to keep the general object information while ignoring the detailed texture information, so it is good for getting a robust motion map. In contrast, the binary images contain more detailed object contour information which may influence the quality of the motion map if the background in the frames contains many detailed texture information.

Key Motion Locations Identification via Motion Map

Our proposed moving object detection and retrieval model identifies the key information locations by searching the maxima and minima of the results of the DoG function when it is applied in scale space to a series of smoothed and re-sampled frames [95]. Some of the key information locations describe the moving objects and the others describe the background. Based on this observation, we use the motion map generated in the previous step to filter those key information locations which are not located on the contour of the moving object. Actually, only the key information locations on the motion map are kept as the so-called “key motion locations” (KMLs) to help find the moving object regions as shown in Figure 4.9(c), since we consider KMLs are motion related.

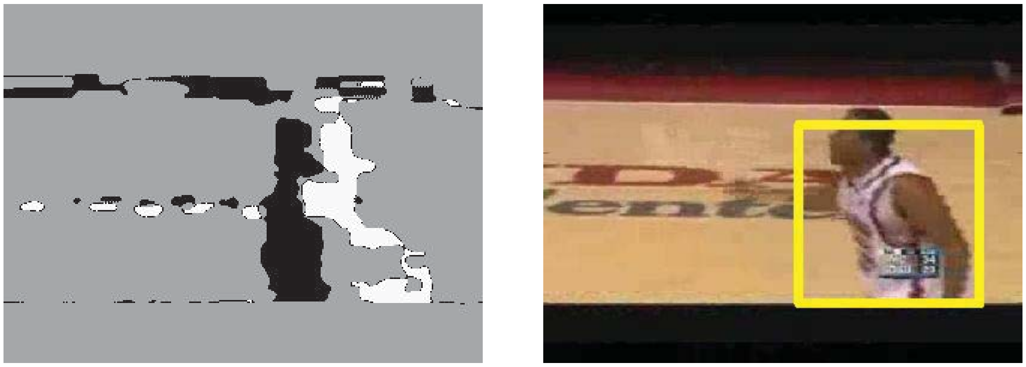
Moving Object Region Detection

After identifying KMLs, how to group them into meaningful moving objects becomes a critical issue. This is a global searching problem that is very time-consuming. To solve this problem, we propose a method to quickly find the moving object regions that have a high density of KMLs and satisfy the direction constraint in the motion map. In the proposed model, the idea of integral images, as defined in [113], is adopted since it allows the fast implementation of the box type convolution filters. The entry of an integral image $I_{\Sigma(\mathbf{x})}$ at a location $\mathbf{x}$=(x,y) represents the sum of all pixels in the input image I of a rectangular region formed by the point $\mathbf{x}$ and the origin, i.e.,

$$I_{\Sigma(x)} = \sum_{i=0}^{i\leq\, x} \sum_{j=0}^{j\leq y} I(i,j).$$

With $I_{\Sigma(\mathbf{x})}$ calculated, it only takes four additions to calculate the sum of the intensities over any upright rectangular areas, independent of their sizes.

Inspired by the integral images, we calculate the density of KMLs in the input image instead of the sum of all pixels. This new approach is defined as the “integral density” in this section. This provides us a fast way to find the region where the density of KMLs is high, and we consider this region is greatly related to the moving objects. In order to bound the whole moving object instead of part of it, the satisfied region is subject to one condition. That is, the moving object region needs to satisfy the constraint that the ratio of two motion zones (white zone and black zone) in the motion map is not high. Ideally, the two zones should have the same area, which indicates the moving direction of the object. Of note, in the section the ratio is set to 2. However, the determination of this ratio threshold can be adjusted depending on the applications. As shown in Figure 4.14(a), the white zone and black zone are separate. Without the above constraint, only a half of the person in Figure 4.14(b) may be bounded where the density of the interest points is high.



(a) (b)

Figure 4.14: Demonstrate the necessity of the proposed constraint for integral density. (a) is the motion map of frame (b), which shows a correct moving object detection result under the constraint.

4.2.2 Effective Retrieval Using Global and Local Features

Our proposed moving object detection and retrieval model consists of a new content-based multimedia retrieval framework that integrates the global and local features to enhance the retrieval performance. The motivation of this framework is to utilize the information obtained from the moving object detection part of the model so that the local or object-level features can be integrated with the commonly used global features for the retrieval. As shown in Figure 4.15, the training phase of the retrieval framework includes two main modules: feature extraction and subspace training, which work on the moving object regions and original frames, respectively. The Subspace modeling and ranking (SMR) algorithm [114] is adopted to train the subspace in this proposed content-based multimedia retrieval framework. That is, a subspace called local subspace will be trained for the local features extracted from the moving object regions, and a subspace called global subspace will be trained for the global features extracted from the original video frames.

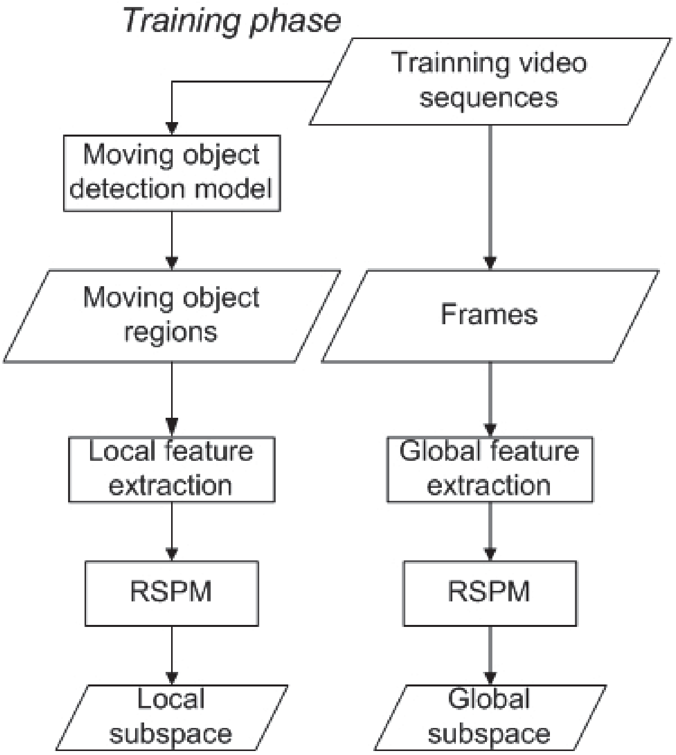


Figure 4.15: Training phase of the proposed moving object detection and retrieval model (with a new content-based multimedia retrieval framework)

For each query (concept), the training data set (local or global) is first split into a positive subset and a negative subset. The positive subset is made up of positive instances; whereas the negative subset consists of negative instances. Two subspace models are built from the two subsets separately. First, the z-scores normalization is applied to the positive instances and the negative instances, respectively. Then, Singular Value Decomposition (SVD) is used to derive the Principal Components (PCs) and eigenvalues of the normalized positive instances and those of the normalized negative instances from their covariance matrix. Those PCs attached to zero eigenvalues are discarded since they contain no extra information. A subspace is built for the positive training instances and likewise a subspace is built for the negative training instances. The two subspaces as well as those related eigenvalues are used in the testing phase for each testing instance.

For the testing phase that is shown in Figure 4.16, the feature extraction process is the same as that in the training phase. The visual features are projected onto the subspace obtained in the training phase. That is, the local features extracted from the moving object regions in the testing data set will be projected onto the local subspace obtained in the training phase (from the moving object regions in the training data set), and the global features extracted from the video frames in the testing data set will be projected onto the global subspace obtained in the training phase (from the video frames in the training data set). Each testing feature vector will be converted into a similarity score after the subspace projection. A fusion process is necessary to combine the similarity scores from local and global subspaces to give a final similarity score to represent each video shot. In this section, the logistic regression method is employed to fuse the global and local similarity scores from the different features. In the future, other fusion methods will be explored in our proposed model.

4.2.3 Experimental Results and Analyses

The effectiveness of our proposed moving object detection and retrieval model is evaluated on a subset of the TRECVID 2010 video collection [79]. The subset contains ten queries (concepts) which involve in motion (consisting of motion information) as shown in Table 4.2. Some shots are multi-labeled, so the total number of shots is less than the sum of the numbers of shots in all ten queries. For example, a shot which is annotated as “running” is possibly also labeled as “sports”.

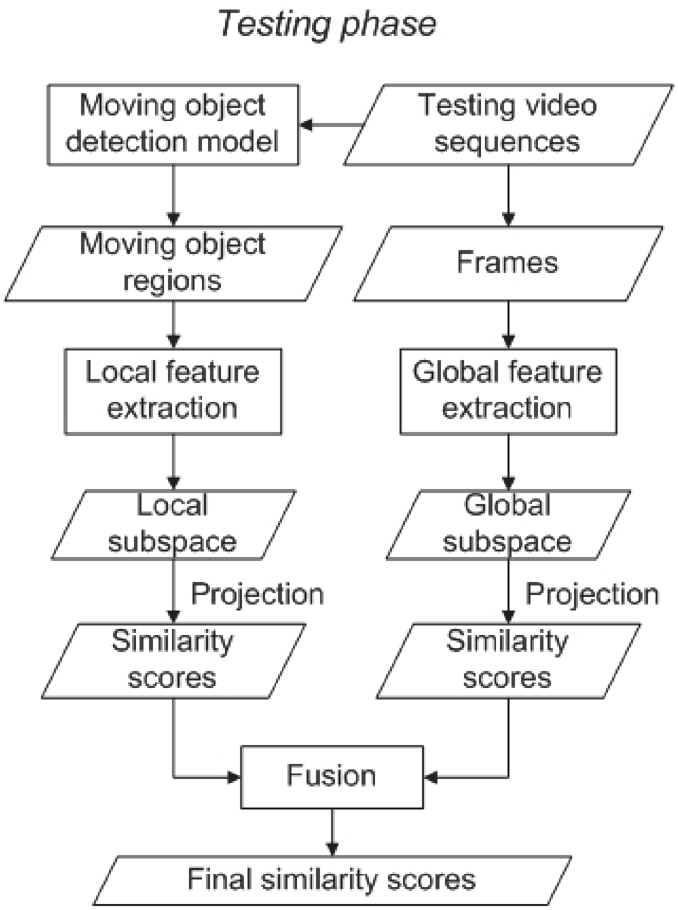
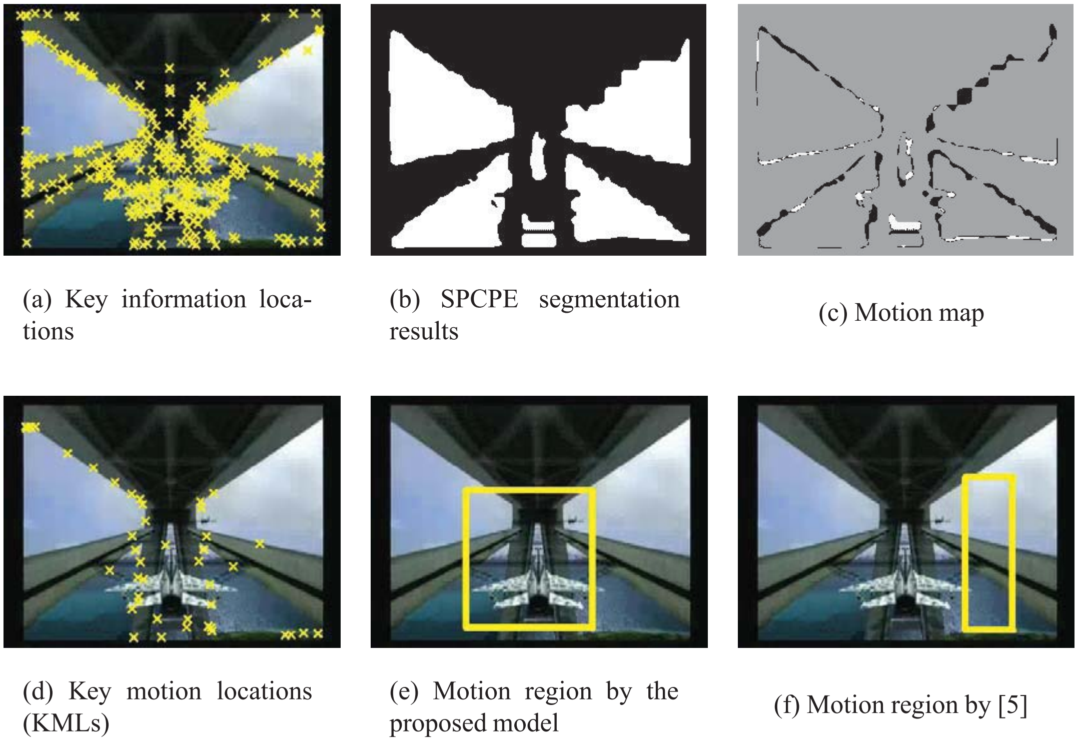


Figure 4.16: Testing phase of the proposed moving object detection and retrieval model (with a new content-based multimedia retrieval framework)

Table 4.2: Data set parameters of the experiments

Concept ID	Concept	# of shots in training	# of shots in testing
4	Airplane-flying	83	113
6	Animal	687	1069
13	Bicycling	79	55
38	Dancing	390	250
59	Hand	759	287
100	Running	245	116
107	Sitting down	1555	536
111	Sports	607	839
127	Walking	1067	412
128	Walking or running	2145	766
	Total	7617	3604



(a) Key information locations

(b) SPCPE segmentation results

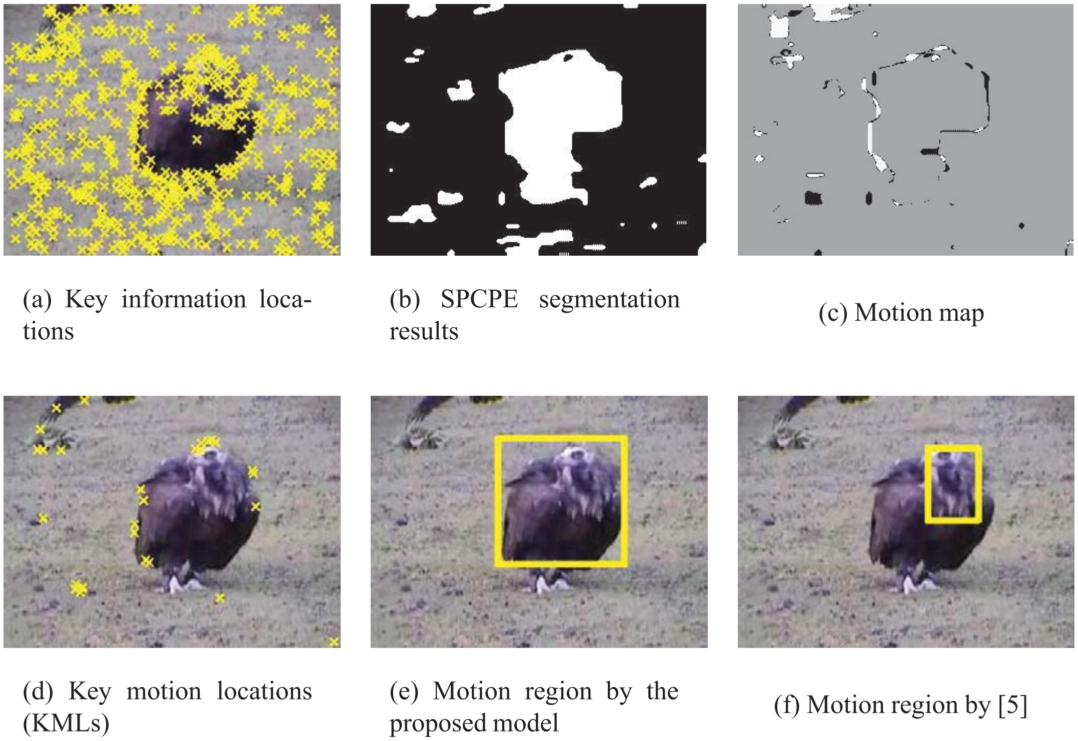
(c) Motion map

(d) Key motion locations (KMLs)

(e) Motion region by the proposed model

(f) Motion region by [5]

Figure 4.17: Sample detection results on a video clip for airplane flying compared with method in [5]. (a) key information locations, (b) SPCPE segmentation results, (c) the motion map, (d) key motion locations (KMLs), (e) moving object regions obtained by the proposed model, and (f) object regions obtained by temporal model in [5].



(a) Key information locations

(b) SPCPE segmentation results

(c) Motion map

(d) Key motion locations (KMLs)

(e) Motion region by the proposed model

(f) Motion region by [5]

Figure 4.18: Sample detection results on a video clip for animal compared with method in [5]. (a) key information locations, (b) SPCPE segmentation results, (c) the motion map, (d) key motion locations (KMLs), (e) moving object regions obtained by the proposed model, and (f) object regions obtained by temporal model in [5].

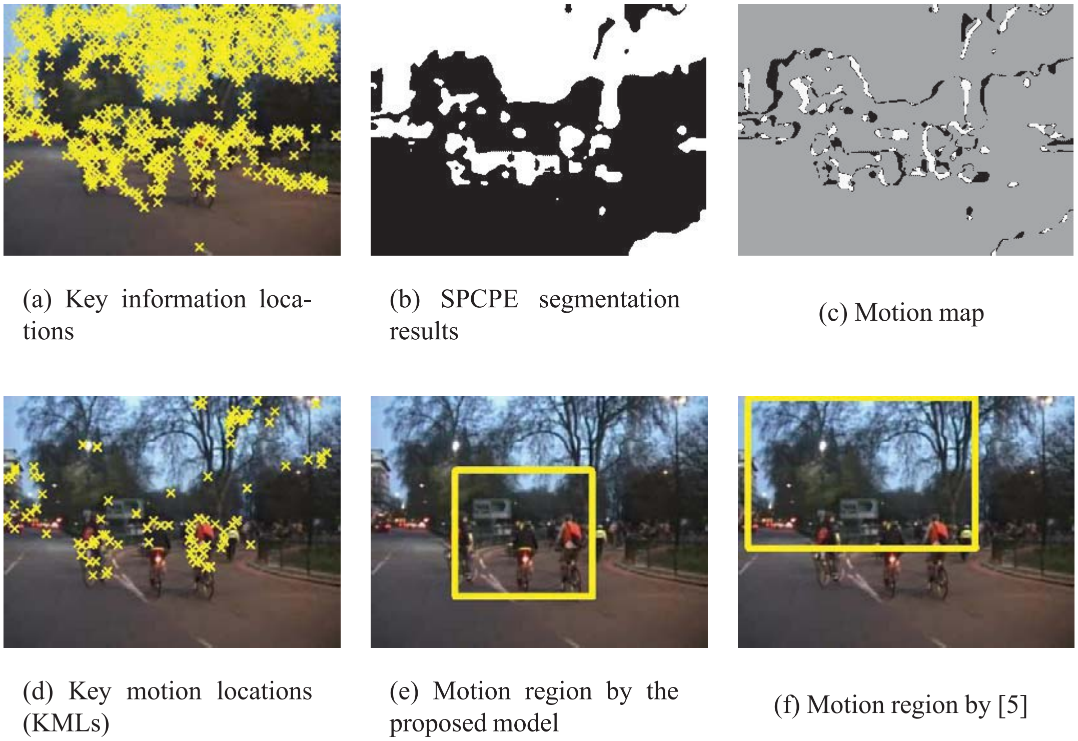
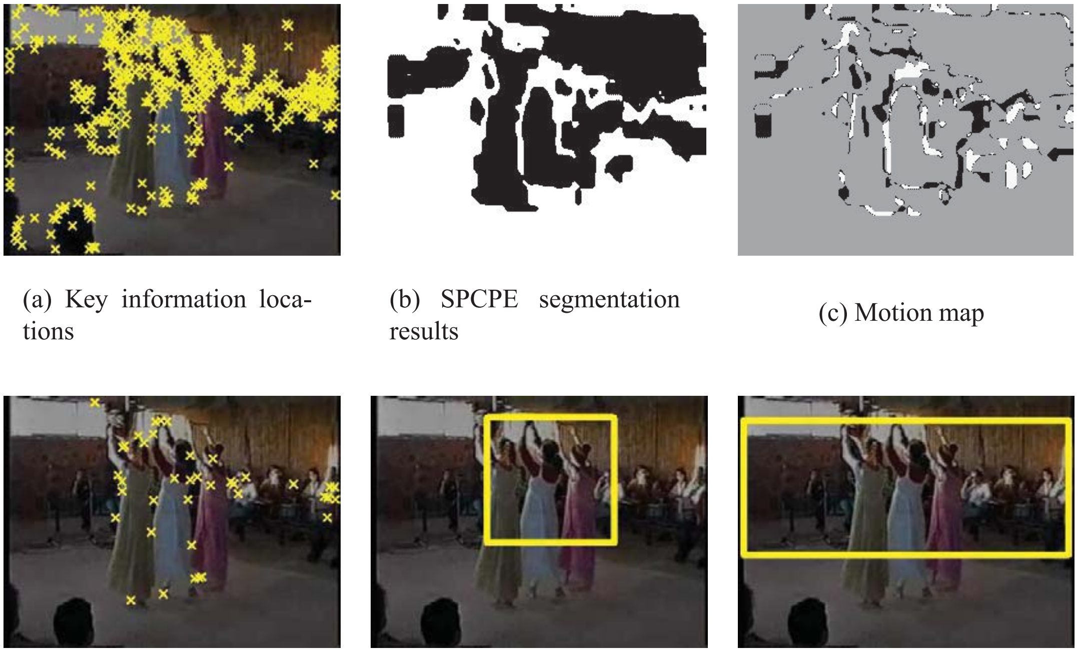

(a) Key information locations
(b) SPCPE segmentation results
(c) Motion map
(d) Key motion locations (KMLs)
(e) Motion region by the proposed model
(f) Motion region by [5]

Figure 4.19: Sample detection results on a video clip for bicycling compared with method in [5]. (a) key information locations, (b) SPCPE segmentation results, (c) the motion map, (d) key motion locations (KMLs), (e) moving object regions obtained by the proposed model, and (f) object regions obtained by temporal model in [5].



(a) Key information locations

(b) SPCPE segmentation results

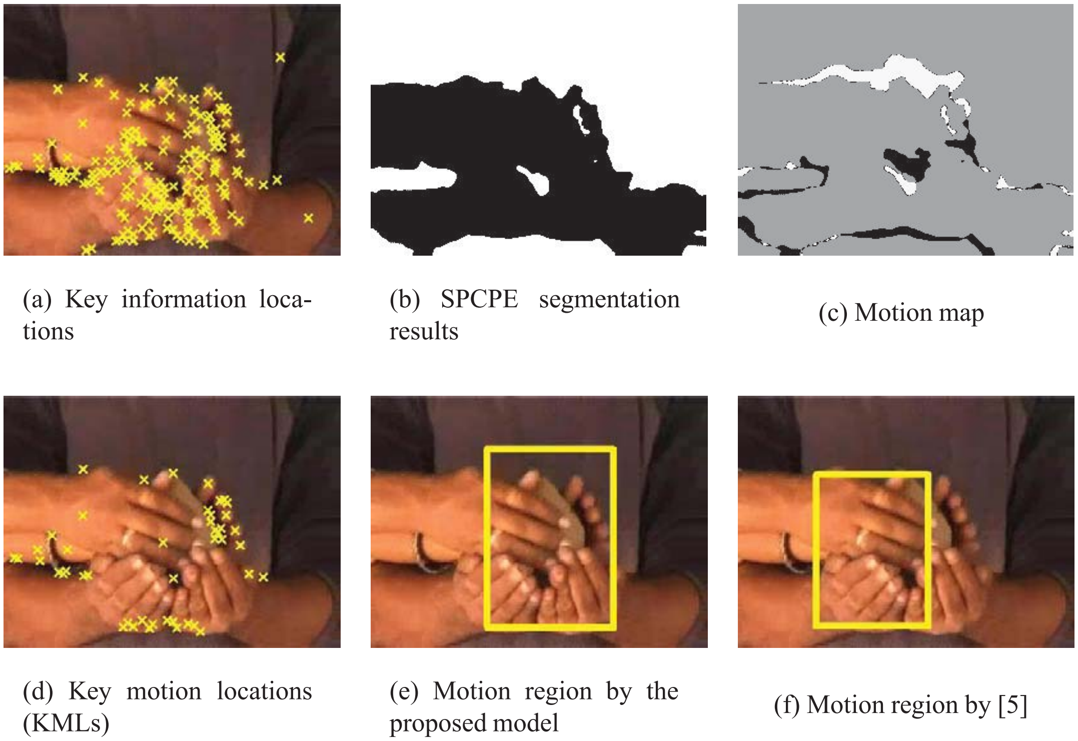
(c) Motion map

(d) Key motion locations (KMLs)

(e) Motion region by the proposed model

(f) Motion region by [5]

Figure 4.20: Sample detection results on a video clip for dancing compared with method in [5]. (a) key information locations, (b) SPCPE segmentation results, (c) the motion map, (d) key motion locations (KMLs), (e) moving object regions obtained by the proposed model, and (f) object regions obtained by temporal model in [5].



(a) Key information locations

(b) SPCPE segmentation results

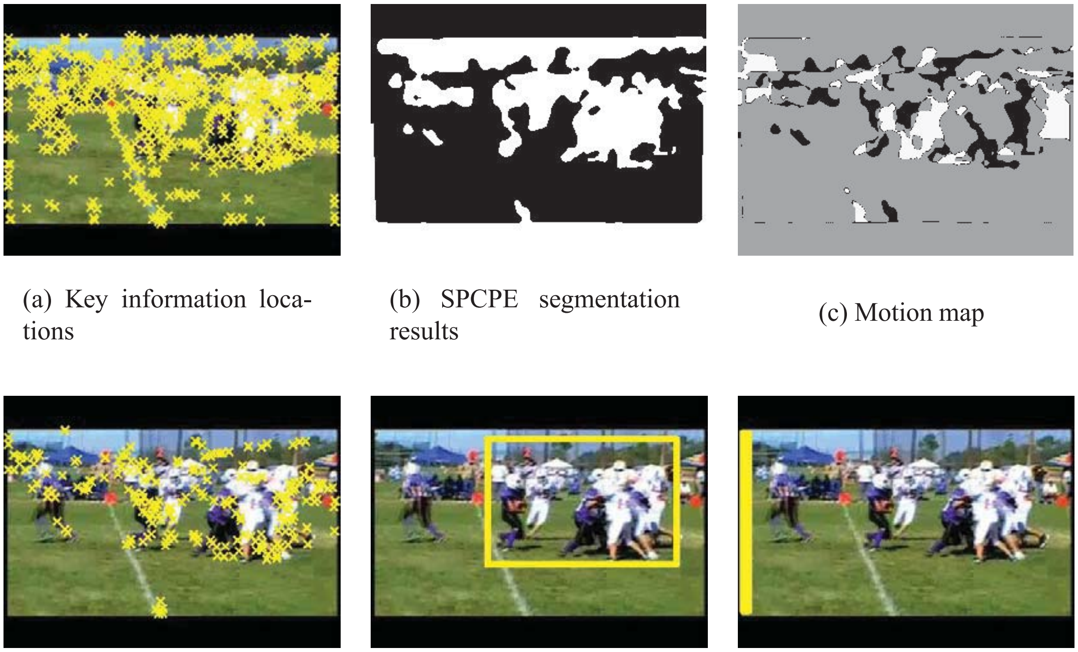
(c) Motion map

(d) Key motion locations (KMLs)

(e) Motion region by the proposed model

(f) Motion region by [5]

Figure 4.21: Sample detection results on a video clip for hand compared with method in [5]. (a) key information locations, (b) SPCPE segmentation results, (c) the motion map, (d) key motion locations (KMLs), (e) moving object regions obtained by the proposed model, and (f) object regions obtained by temporal model in [5].



(a) Key information locations

(b) SPCPE segmentation results

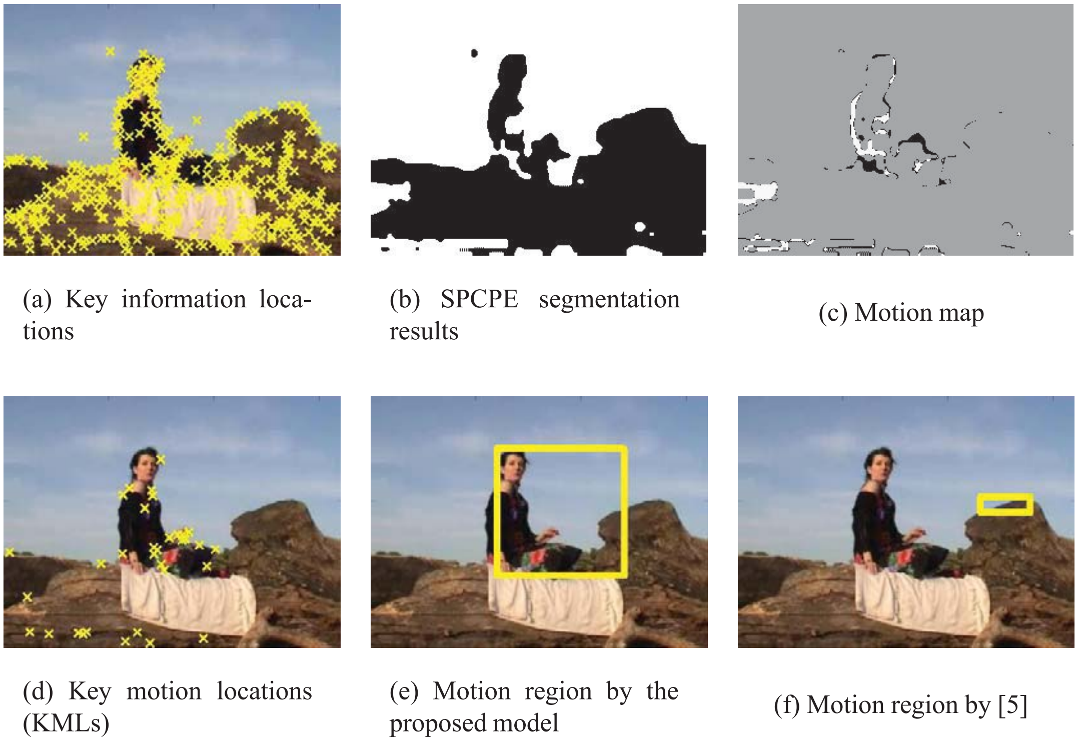
(c) Motion map

(d) Key motion locations (KMLs)

(e) Motion region by the proposed model

(f) Motion region by [5]

Figure 4.22: Sample detection results on a video clip for running compared with method in [5]. (a) key information locations, (b) SPCPE segmentation results, (c) the motion map, (d) key motion locations (KMLs), (e) moving object regions obtained by the proposed model, and (f) object regions obtained by temporal model in [5].



(a) Key information locations

(b) SPCPE segmentation results

(c) Motion map

(d) Key motion locations (KMLs)

(e) Motion region by the proposed model

(f) Motion region by [5]

Figure 4.23: Sample detection results on a video clip for sitting down compared with method in [5]. (a) key information locations, (b) SPCPE segmentation results, (c) the motion map, (d) key motion locations (KMLs), (e) moving object regions obtained by the proposed model, and (f) object regions obtained by temporal model in [5].

Performance of the Moving Object Detection Framework

To form the experimental data set, a reference key frame for each shot is kept, assuming that the reference key frame represents the content of the shot. In the proposed motion region detection model and the semantic concept retrieval model, four extra frames around the reference key frame in each shot are extracted for the purpose of calculating the motion of the shots. In this study, the time interval between two frames is set to 0.2 seconds. Such a value can be adjusted or adaptively computed based on the motion speed in the shot in our future work. Furthermore, to achieve fast computation, the minimum motion region size is set to 0.4 times of the shorter dimension size of the frame. This assumes that a small region only includes a part of a moving object. Some examples of motion region detection results in the data set are provided in Figures 4.17-4.23.

Without considering the temporal motion information in the video sequences, there can be a large number of key information locations which represent a rich texture information area from the spatial angle, but some of them are considered noise (as shown in (a) of Figures 4.17-4.23). As the result of applying our proposed key information location filtering via the motion map (in (c) of Figures 4.17-4.23 obtained from the SPCPE algorithm), it can be clearly seen that the resulting key motion locations (KMLs) are able to keep the spatio-temporal information which is suitable for the motion region detection purpose (as shown in (d) of Figures 4.17-4.23). The detection results from the temporal model of [5] are given in (f) of Figures 4.17-4.23. The reason why the model proposed in [5] fails in some cases is that the model is greatly influenced by the results of point correspondences. Though the new model proposed in this section uses a similar strategy as in [5] to locate the key information locations, our proposed motion map removes those motion-unrelated information and the integral density method successfully gets the motion region by precisely analyzing the distribution of the KMLs. Finally, the effectiveness of our proposed motion region detection model is demonstrated via (e) of Figures 4.17-4.23.

Performance of the Proposed Moving Object Detection and Retrieval Model

The motion regions detected in the previous subsection can be viewed as a kind of local features that describe the object-level texture of the shot. This may be complementary to the global information for multimedia retrieval. To verify this assumption, a set of comparable experiments is conducted in three data sets (reference key frame (RKF), multiple frames (MF), and multiple frames plus motion regions (MF+MR)). The data set of the reference key frames is the same set used in the moving object detection model. Four frames are extracted per shot around the reference key frame. That is, including the reference key frame, each shot is represented by five frames which consist of the MF data set. On each frame, one or more motion regions (MR) may be detected. Motion regions detected in the MF data set plus the MF data set itself form the MF+MR data set. The experimental design aims to check whether the motion region features can complement the global features to enhance multimedia retrieval.

In the feature extraction step, three kinds of texture features (YCbCr, Gabor, and LBP) are extracted from each data set. For YCbCr features, the frame or region is first converted to the YCbCr color space from the RGB color space. Then the frame or region is divided into nine blocks. Mean, variance, skewness, and kurtosis are calculated on Y, Cb, and Cr components, respectively. Considering the mean, variance, skewness, and kurtosis calculated on Y, Cb, and Cr components of the global frame, there are totally 120 features that are obtained from each frame or region. For Gabor features,

Table 4.3: MAP Comparison when different numbers of results are requested (%) - YCbCr features

Top	10	100	1000	2000	All
RKF	0.4528	0.4657	0.3745	0.3400	0.3063
MF	0.5484	0.5503	0.4147	0.3752	0.3460
MF+MR	**0.6746**	**0.6103**	**0.4355**	**0.3968**	**0.3691**
Impr. % to RKF	48.98	31.05	16.29	16.71	20.50
Impr. % to MF	23.01	10.90	5.02	5.76	6.68

Table 4.4: MAP Comparison when different numbers of results are requested (%) - Gabor features

Top	10	100	1000	2000	All
RKF	0.7034	0.5773	0.4045	0.3630	0.3349
MF	0.6511	0.6051	0.4563	0.4051	0.3798
MF+MR	**0.7286**	**0.6550**	**0.4799**	**0.4271**	**0.3997**
Impr. % to RKF	3.58	13.46	18.64	17.66	19.35
Impr. % to MF	11.90	8.25	5.17	5.43	5.24

a set of Gabor filters with different frequencies and orientations is convolved with the frame or region to generate 108 features to describe the frame or region. LBP (Local Binary Pattern) is a simple yet very efficient texture operator which labels the pixels of a frame or region by thresholding the neighborhood of each pixel and considers the result as a binary number. After summarization of the binary numbers, 59 LBP features are returned to represent the frame or region.

In this section, we transform the multi-class issue into the binary class problem. This means that in the training phase, the one-again-all strategy is utilized. Logistic regression method is used to fuse the similarity scores of multiple frames (MF) as well as multiple frames and motion region (MF+MR).

The mean average precision (MAP) is defined as the mean of the average precision (AP) of all queries, and is used as the criterion to evaluate and compare the performance of different approaches. Average precision (AP) is a popular measure that takes into

Table 4.5: MAP Comparison when different numbers of results are requested (%) - LBP features

Top	10	100	1000	2000	All
RKF	0.4929	0.5287	0.4370	0.4079	0.3915
MF	0.5316	0.5541	0.4729	0.4437	0.4281
MF+MR	**0.6209**	**0.6034**	**0.4983**	**0.4659**	**0.4501**
Impr. % to RKF	25.97	14.13	14.03	14.22	14.97
Impr. % to MF	16.80	8.90	5.37	5.00	5.14

Table 4.6: MAP Comparison when different numbers of results are requested (%) - LBP + Gabor + YCbCr

Top	10	100	1000	2000	All
RKF	0.7159	0.6787	0.5460	0.5113	0.4952
MF	0.7801	0.7221	0.5759	0.5423	0.5261
MF+MR	**0.8563**	**0.7741**	**0.6134**	**0.5748**	**0.5594**
Impr. % to RKF	19.61	14.06	12.34	14.42	12.96
Impr. % to MF	9.77	7.20	6.51	5.99	6.33

account of both recall and precision in the information retrieval field. Strictly speaking, the average precision is the precision averaged across all recall values between 0 and 1. In practice, the integral is closely approximated by a sum over the precisions at every possible threshold value, multiplied by the change in recall. Let k be the rank in the sequence of retrieved shots, n be the number of retrieved shots, $P(k)$ be the precision at cut-off k in the list, and $\Delta r(k)$ be the change in recall from items $k-1$ to k [115]. AP is defined as shown in Equation (4.2).

$$AP = \sum_{k=1}^{n} P(k)\Delta r(k). \tag{4.2}$$

Tables 4.3-4.6 show the MAP values when retrieving 10, 100, 1000, 2000, and all shots in the three data sets. RKF means the reference key frame data set; MF means the multiple-frame data set including the reference key frame and four extra frames; and MF+MR is the union of MF and motion-region data set, including multiple frames with the moving object region obtained from the multiple frames. Though using different features, the retrieval results are consistent among three kinds of data sets. The results of MF generally outperform those of RKF at different numbers of the retrieval shots, which indicates that using multiple frames could provide more useful information to improve the concept retrieval performance than using a single reference key frame. On the other hand, MR+MF outperforms both MF and RKF on all ten queries. This verifies that the moving object region has the concept-related information that can be utilized in the semantic retrieval domain. When comparing the MAP values in the same data set among Tables 4.3- 4.5, the YCbCr, Gabor, and LBP return similar MAP values. If using multiple features, the retrieval performance is improved in a considerable degree (20% more in Table 4.6). Also, we observed that the proposed detection model indeed effectively identifies the moving object in the frame as shown in Figures 4.17-4.23.

4.2.4 Conclusions

This section proposes a new moving object detection and retrieval model to analyze and retrieve the spatial-temporal video sequence information. A motion map is generated from the SPCPE segmentation results to keep the motion related key information locations, called key motion locations (KMLs). Next, an integral density method is proposed to quickly and precisely identify the motion region by analyzing the density of the KMLs under the motion direction restraint generated by the motion map. A new multimedia retrieval framework using the global and local features is presented to effectively combine and fuse the texture information from the global features via the original frames and the local features from the motion regions. Experimental results show that our proposed moving object detection and retrieval model achieves good performance in terms of the moving object detection and multimedia concept retrieval.

4.3 Motion Saliency Detection Using Center-Surround Coherency Model

In this section, a video semantic retrieval framework is proposed based on a novel unsupervised motion region detection algorithm which works reasonably well with dynamic background and camera motion. The proposed framework is inspired by biological mechanisms of human vision that motion saliency (defined as attention due to motion) is more "attractive" than some other low-level visual features to people while watching videos. Under this biological observation, motion vectors in frame sequences are calculated using the optical flow algorithm to estimate the movement of a block from one frame to another. Next, a center-surround coherency evaluation model is proposed to compute the local motion saliency in a completely unsupervised manner. The integral density algorithm is employed to search for the globally optimal solution of the minimum coherency region as the motion region which is then integrated into the video semantic retrieval framework to enhance the performance of video semantic analysis and understanding. Our proposed framework is evaluated using video sequences in non-static background, and the promising experimental results reveal that the semantic retrieval performance can be improved by integrating the global texture and local motion information.

The main contributions of this section include: (1) Define a center-surround coherency model to describe motion contrast computed by motion vectors obtained from the optical flow algorithm. (2) Employ the integral density algorithm to calculate the globally optical minimum coherency as the motion region in the frame. (3) Present a multimedia retrieval framework to integrate global texture and local motion in order to enhance the existing retrieval framework that uses only global features [116].

The remainder of this section is organized as follows. The motion saliency region detection framework is presented in Section 4.3.1. Section 4.3.2 describes the new semantic retrieval model that fuses the global texture and local motion features to enhance the retrieval performance. The new content-based multimedia retrieval framework is also introduced in this section. Section 4.3.3 presents the experimental results and analyzes the performance on KTH and TRECVID 2010 data sets from the detection and retrieval perspectives, respectively. Section 4.3.4 concludes the proposed motion saliency detection and semantic retrieval model.

4.3.1 Motion Saliency Region Detection

The studies on the human vision system reveal that it perceives external features separately and is sensitive to the diversity of the target region and its neighborhood [22][21]. The center-surround mechanisms of biological systems support the idea of motion saliency detection on the measurements of local motion contrast. In order to build an unsupervised detection framework on motion saliency while avoiding the “global background model” or any type of training processing, a center-surround coherency model is proposed in our proposed framework (as shown in Figure 4.24) to measure the motion contrast of a local region and its neighborhoods. After that, the integral density algorithm is utilized to achieve global minimum coherency as the expected motion region.

It is not necessary to train samples or pre-build a “global background model” for the testing instances in the proposed model. Local motion information can be utilized to compute the motion saliency, so that the model could immediately adapt to different kinds of unknown backgrounds. Moreover, the model is robust to the camera motion and dynamic background because of the exploration of its global minimum coherency.

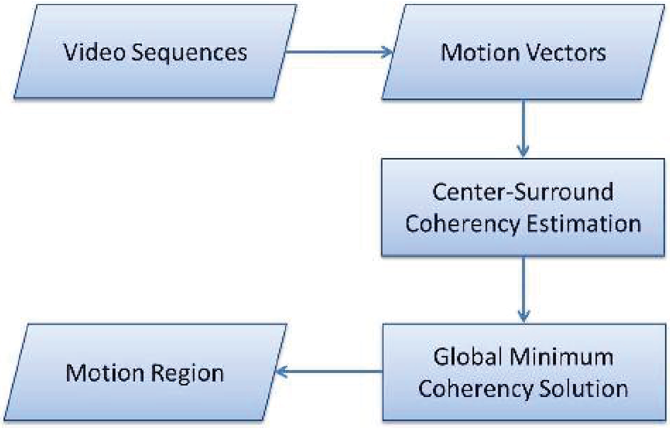
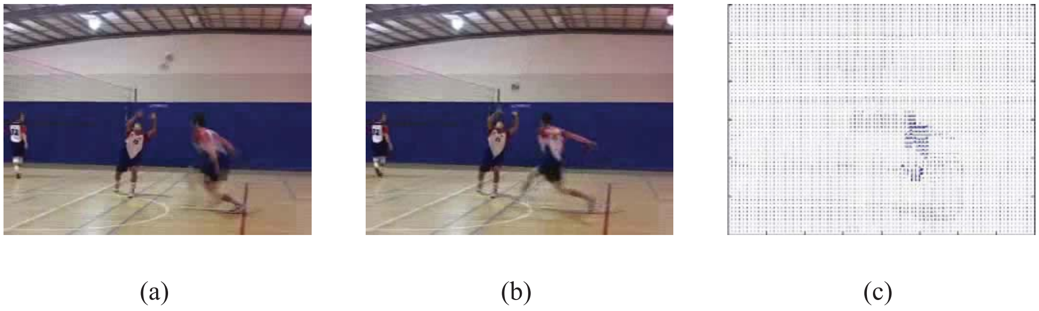

Video Sequences
Motion Vectors
Center-Surround Coherency Estimation
Global Minimum Coherency Solution
Motion Region

Figure 4.24: Motion region detection model

Motion Vector by Optical Flow

The concept of optical flows was introduced by James J. Gibson in the 1940s to describe the visual stimulus provided to animals moving through the world. In 1981, Horn and Schunck [91][92] conducted a performance analysis of a number of optical flow techniques. Recently the term optical flow has been co-opted to incorporate related techniques from image processing and control of navigation, such as motion detection, object segmentation, etc. The optical flow methods try to calculate the motion between two image frames which are taken at times t and $t+\Delta t$, as shown in Figures 4.25(a) and 4.25(b).


(a)

(b)

(c)

Figure 4.25: Illustration of two frames at t and $t+\Delta t$, following by the optical flow calculated from the two frames.

If the coordinates of a pixel on the image with its gray value at time t is $I(x,y,t)$ and the pixel moves to new position at time $(t+\Delta t)$, its location on the image becomes $(x+\Delta x, y+\Delta y)$, and the gray value becomes $I(x+\Delta x, y+\Delta y, t+\Delta t)$. Assuming that the intensity is conserved, we can have Equation (4.3) which can be re-written as Equation (4.4). The gradient constraint equation is easily derived from a Tayor expansion of Equation (4.4) as shown in Equation (4.5).

$$dI(x,y,t)/dt = 0; \tag{4.3}$$

$$I(x,y,t) = I(x+\Delta x, y+\Delta y, t+\Delta t); \tag{4.4}$$

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0. \tag{4.5}$$

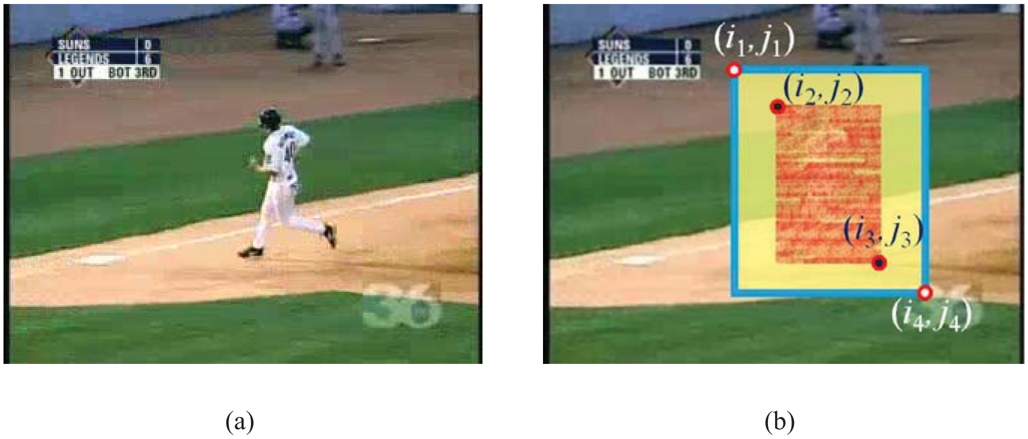
Let the two components of the optical flow along the x and y coordinates be $u = dx/dt$ and $v = dy/dt$, and let I_x, I_y, and I_t denote the partial x coordinate, partial y coordinate, and partial time derivatives of $I(x,y,t)$. Equation (4.6) presents the basic optical flow equation. Figure 4.25(c) illustrates the optical flow calculated from Figures 4.25(a) and 4.25(b).

$$I_x u + I_y v + I_t = 0. \tag{4.6}$$

The optical flow does not require any a priori knowledge on the object appearance, which is an important merit. However, its complex computation time makes it unsuitable for real-time applications (if without special hardware). To address such an issue, in this section, motion vectors are used to decrease its processing time, i.e., to use the optical flow technique on the block-level motion instead of the pixel-level one. An integral part of many video compression algorithms is the motion vectors since they are used for motion compensation. The idea of the so-called block matching is to divide the current frame into a matrix of blocks that are then compared with the corresponding block and its neighbors in the previous frame to determine the motion vector. In other words, the motion vector is calculated using the optical flow method, but the motion information of the frame is presented in the block-level.

Center-Surround Coherency Model

To deal with the issues raised by the camera motion or the dynamic background, a center-surround coherency model is presented, which enables the automatic adaption of the background variations. There is no need to build or train a global model of the background. Because coherency compares the center and surrounded regions, it depends only on the relative disparity between the motion values, and therefore the new model is invariant to the camera motion.



(a)

(b)

Figure 4.26: Illustration of a center-surround sample window. (a) the original frame; (b) an illustrated center-surround region. The red area denotes the center region, and the yellow area denoted the surround region.

Suppose an image is divided into $m \times n$ blocks, $1 \leq i \leq m$, $1 \leq j \leq n$, so $u_{i,j}$ and $v_{i,j}$ denote two components of the motion vectors at block (i, j). Given a center-surround region R which includes a center region R_c and a surrounded region R_s as shown in the red and yellow areas in Figure 4.26(b). If the block $(i, j) \subset R$, then block (i, j) belongs to either R_c or R_s, where $1 < i_1 \leq i \leq i_4 < m$, $1 < j_1 \leq j \leq j_4 < n$. Here, i_1, j_1, i_4, and j_4 are the block boundary coordinates of R; while i_2, j_2, i_3, and j_3 are the block boundary coordinates of R_c. The motion vectors U_c and V_c of the center region R_c are computed by summing up the motion vectors of the blocks located in the center region as shown in Equation (4.7) and Equation (4.8), where $\forall$ block $(i, j) \subset R_c$. If $\forall$ block $(i, j) \subset R_s$, then the motion vectors U_s and V_s of the surrounded region are calculated in the same way as given in Equation (4.9) and Equation (4.10).

$$
\begin{aligned}
U_c &= \sum u_{i,j}; && (4.7) \\
V_c &= \sum v_{i,j}; && (4.8) \\
U_s &= \sum u_{i,j}; && (4.9) \\
V_s &= \sum v_{i,j}. && (4.10)
\end{aligned}
$$

The coherency C of the center region and the surrounded region can be obtained by computing the cosine similarity over the center and surrounded areas (as shown in Equation (4.11)).

$$C = \cos\theta = \frac{M_c \cdot M_s}{\|M_c\| \|M_s\|},$$

where $M_c = [U_c \ V_c]$ and $M_s = [U_s \ V_s]$ denote the motion energy values in the center region R_c and the surrounded region R_s, respectively. The smaller the value C is, the lower the probability that the center region and the surrounded region have similar motion activities.

The motion vector gives a measure of the block movement direction in the image. The greater cosine similarity of the two motion vectors, the more likely the two motion vectors come from the same object. Considering the temporal consistence of the object motion in continuous frame sequences, the sum of the coherency C_t in $\triangle t$ time window is calculated as the estimation criterion of the motion activity in region R in Equation (4.11).

$$C_t = \sum\nolimits_t^{t+\triangle t} C.$$

Global Minimum Coherency

After the discussion of the temporal coherency C_t, how to quickly find the global minimum coherency region in the image turns into an urgent problem in the unsupervised motion detection topic. Such a problem is a global search issue, which is usually very time-consuming. To solve this issue, a quick search method is presented to find the possible motion regions that have a low center-surround coherency. The integral density concept in [94], which was developed based on the integral images in [113], is adopted. The rationale is because it allows a fast implementation of the box type convolution filters. Each entry in the summed area table $I_{\Sigma(\mathbf{x})}$ at a location $\mathbf{x}$=(x,y) represents the sum of all the values in the input 2D matrix I of a rectangular region formed by the point $\mathbf{x}$ and the origin (please see Eq. (4.11)). After $I_{\Sigma(\mathbf{x})}$ is calculated, the calculation of the sum of the values over any upright rectangular areas, independent of their sizes, will take only four additions.

$$I_{\Sigma(\mathbf{x})} = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j).$$

Inspired by the summed area table algorithm, the motion vectors of each block in an image are written as a matrix. The summed area table of this motion matrix is then generated for the fast computation of the center-surround coherency of every location.

After traversing the center-surrounded region in the image, the global minimum coherency can be quickly obtained.

4.3.2 Semantic Retrieval Model

Based on the proposed motion saliency region detection algorithm, a semantic retrieval model is presented. It consists of a new multimedia semantic retrieval framework that integrates the global texture and local motion features to enhance the retrieval performance. The motivation of this framework is to utilize the information obtained from the motion saliency region detection part of the model so that the local or object-level features can be integrated with the commonly used global features for the retrieval. As shown in Figure 4.27, the training phase of the retrieval framework includes two main modules: feature extraction and subspace training, which work on the motion regions and original frames, respectively. The representative subspace projection modeling (*RSPM*) algorithm [15] is adopted to train the subspace in this proposed multimedia semantic retrieval framework. That is, a subspace called the local subspace will be trained for the local features extracted from the motion regions, and a subspace called global subspace will be trained for the global features extracted from the original video frames.

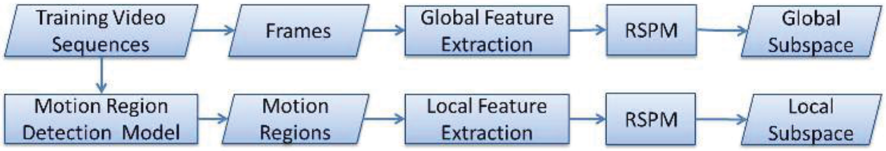


Figure 4.27: Training phase in the proposed semantic retrieval framework

In the testing phase given in Figure 4.28, the feature extraction process is the same as that in the training phase. The visual features are projected onto the subspace obtained in the training phase. That is, the local features extracted from the motion regions in

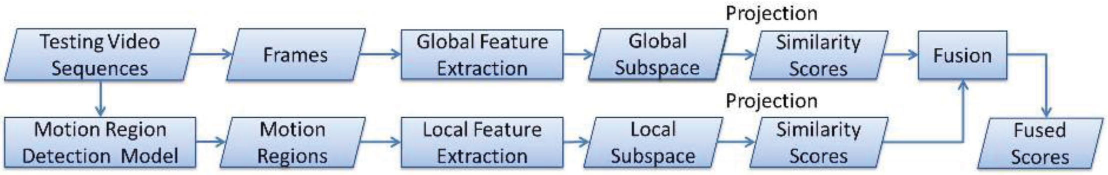


Figure 4.28: Testing phase in the proposed semantic retrieval framework

the testing data set will be projected onto the local subspace obtained in the training phase (from the motion regions in the training data set), and the global features extracted from the video frames in the testing data set will be projected onto the global subspace obtained in the training phase (from the video frames in the training data set). Each testing feature vector will be converted into a similarity score after the subspace projection. A fusion process is necessary to combine the similarity scores from the local and global subspaces to give a final similarity score to represent each video shot. A good fusion strategy can further improve the final performance of the semantic retrieval framework. In this section, the logistic regression algorithm is employed to combine the global and local similarity scores. In the future, more fusion methods will be explored in our proposed model.

4.3.3 Experimental Results and Analyses

We use two data sets, KTH [97] and TRECVID 2010 (in semantic indexing task) [79], to evaluate the performance of the proposed framework. In the KTH data set, there are 25 actors performing six actions four times in four different environments with a total number of 599 video sequences. There are six action categories, namely boxing, hand clapping, hand waving, jogging, walking, and running. One characteristic of these video sequences is that they were recorded in a controlled setting with slight camera motion and a simple background.

Table 4.7: TRECVID 2010 Data Set Used in the Experiments

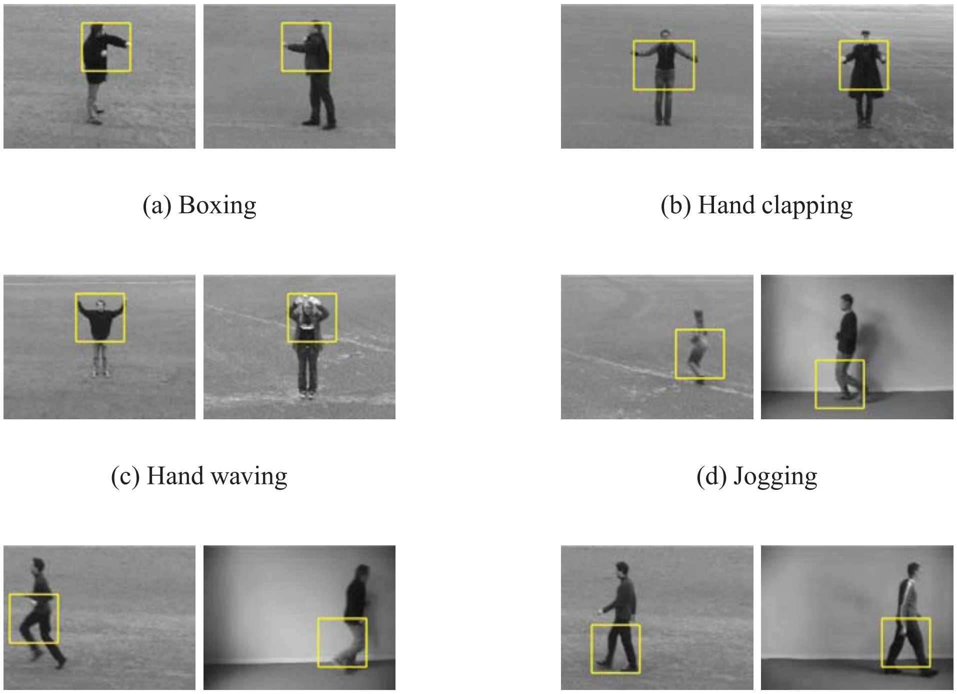
Concept ID	Concept name	# of shots in training	# of shots in testing
4	Airplane-flying	83	113
6	Animal	729	1087
13	Bicycling	111	64
38	Dancing	411	255
59	Hand	764	289
100	Running	638	252
111	Sports	1024	275
127	Walking	2237	850
	Total	5997	3185

On the other hand, the data set in semantic indexing task of TRECVID 2010 contains 130 queries, while the majority belongs to static concepts. Eight queries describing moving objects were chosen to build a subset for testing our framework, namely airplane flying, animal, bicycling, dancing, hand, running, sports, and walking. These all involve salient motion. More detailed information is shown in Table 4.7. Some shots are multi-labeled. For example, one shot annotated as “sports” can also be labeled as “running”. In this section, the multi-class issue is transformed into the binary class problem, meaning that the one-again-all strategy is utilized in the training phase.

Experiments on the KTH Data Set

This KTH data set is used to demonstrate that our proposed framework is able to achieve pretty good performance in videos recorded in a “clean” background, even though our proposed framework is designed to deal with videos captured in uncontrolled environments. In the frame extraction step, we did not use a key frame extraction algorithm to select the representative key frames as did in [57]. Instead, three frames per second on average are used to compute the motion saliency in the KTH data set.

First, the accurate localization of motion is verified. Samples of motion saliency regions are illustrated in yellow boxes in Figure 4.29. We notice that the motion saliency of the human body is accurately identified from the videos, while the static parts of the body are excluded from the boxes. This property of the motion saliency detection model will later be transferred to the advantage of moving object-level feature extraction, and proved helpful for semantic retrieval.



(a) Boxing

(b) Hand clapping

(c) Hand waving

(d) Jogging

(e) Running

(f) Walking

Figure 4.29: Samples of motion saliency detection on KTH data set

In the experiments, we test the precision of the concept retrieval using those features extracted in frame-wide and region-wide, respectively. To avoid the feature bias, three kinds of texture features (Gabor, LBP, and HOG) are employed to represent each frame and motion region. For Gabor features, a set of Gabor filters with different frequencies and orientations is convolved with the frame or region to generate 108 features to describe the frame or region. LBP (Local Binary Pattern) is a simple yet very efficient texture operator which labels the pixels of a frame or region by thresholding the neighborhood of each pixel and considers the result as a binary number. After the summarization of the binary numbers, 59 LBP features are returned to represent the frame or region. Histogram of Oriented Gradients (HOG) are feature descriptors that are used in computer vision and image processing. HOG features count the occurrences of the gradient orientation in the localized portions of an image. It is computed on a dense grid of uniformly spaced cells and uses the overlapping local contrast normalization for improved accuracy. The dimension of the HOG features used in the experiment is 135.

The Mean Average Precision (MAP) value is used to evaluate the performance of different approaches in the section. MAP is the mean of the average precision (AP) of all queries. For approaches that return a ranked sequence of video shots, the Average Precision (AP) value is a criterion that considers the order in which the returned shots are presented. In the other word, AP is the precision value averaged across all recall values between 0 and 1. Let k be the rank in the sequence of retrieved shots, n be the number of retrieved shots, $P(k)$ be the precision at cut-off k in the list, and $rel(k)$ be an indicator function with 1 if the item at rank k is a relevant shot, and 0 otherwise [115]. AP is defined as

$$AP = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{number\ of\ relevant\ shots} \tag{4.11}$$

In the KTH data set, we select the frames of ’person’ 01 - 15 as the training set and the frames of ’person’ 16 - 25 as the testing set. In each frame, Gabor, HOG, and LBP features are extracted in frame-wide and region-wide, respectively. The purpose of extracting frame-wide features is to estimate the performance of only using global features and ignoring the object-level features. Then, the features of the object-level are estimated as region-wide features. We expect the features from a small area, but the motion saliency should be more discriminative than those from frame-wide.

Meanwhile, considering the possibility of the complementary information among different methods, we also test the performance of the fused similarity scores of frame-wide and region-wide. The scores are fused by Logistic Regression (LR) method. Tables 4.9, 4.10, and 4.8 present the retrieval results in terms of mean average precision (MAP). The columns show the number of videos requested in each method. Note that the region-wide method outperforms the frame-wide one using the LBP features, while using Gabor features, the frame-wide method exceeds the region-wide one. For HOG features, if retrieving the top 5, 10, or 20 related videos, the region-wide method performs better than frame-wide one; while if retrieving more than 50 related videos, the frame-wide approach obtains a higher MAP. This result indicates that a single method does not achieve good precision on all kinds of features. Thus, a fusion technique is utilized to integrates the advantages of frame-wide and region-wide methods.

The experimental results of the fused method (labeled as "Fused") are shown in Tables 4.10, 4.9, and 4.8. The last two rows of Tables 4.10, 4.9, and 4.8 list the improvement of the fused method compared to the frame-wide and region-wide methods, respectively. It can be observed that the fused results, finding that the performance improvement is prominent.

The average improvements of the fused method by using the Gabor and HOG features are 39.75% and 14.52%, respectively. For the LBP features, the poor performances of the frame-wide method affect the fusion results, resulting in the decrease in MAP comparing to the region-wide method in the top 50 and 100 retrieved videos. However, in the top 5, 10, and 20, the fused method achieves an increase in MAP though the performances of the frame-wide and region-wide methods are not commensurable.

Table 4.8: MAP Comparison when different numbers of results are requested (%) - LBP Features

LBP	5	10	20	50	100
Frame-wide	3.33	5.42	9.42	15.09	18.14
Region-wide	32.36	39.48	40.82	43.29	44.24
Fused	49.17	49.51	47.29	42.24	41.08
Impr. % to frame-wide	1376.58	813.47	402.02	179.92	126.46
Impr. % to region-wide	51.95	25.41	15.85	-2.43	-7.14

Table 4.9: MAP Comparison when different numbers of results are requested (%) - Gabor Features

Gabor	5	10	20	50	100
Frame-wide	52.99	53.88	48.10	43.02	39.02
Region-wide	42.92	36.60	33.88	33.13	33.56
Fused	67.78	65.07	54.34	52.01	47.25
Impr. % to frame-wide	27.91	20.77	12.97	20.90	21.09
Impr. % to region-wide	57.92	77.79	60.39	56.99	40.79

The overall performance of the fused method verifies that the global (frame-wide) and local (region-wide) information has the complementary discriminative potential for information retrieval.

Experiments on the TRECVID Data Set

One reference key frame in each shot is provided in the TRECVID 2010 video collection. In addition to the reference key frame that stands for the content of the shot, we also extract four extra frames around the reference key frame in each shot for the

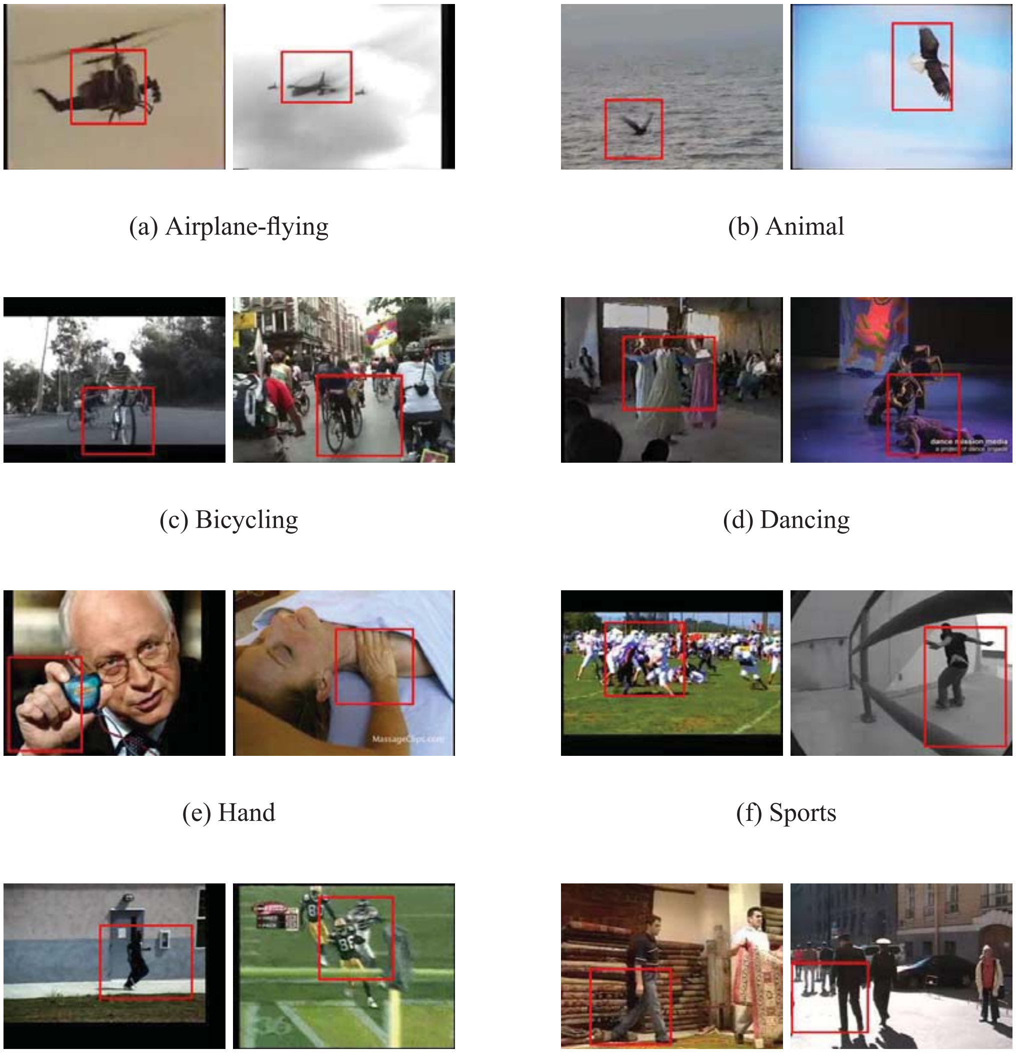
Table 4.10: MAP Comparison when different numbers of results are requested (%) - HOG Features

HOG	5	10	20	50	100
Frame-wide	57.57	62.58	67.07	68.06	65.25
Region-wide	70.09	67.35	67.33	63.91	62.10
Fused	78.68	78.14	74.33	71.54	69.09
Impr. % to frame-wide	36.67	24.86	10.82	5.11	5.89
Impr. % to region-wide	12.26	16.02	10.40	11.94	11.26

purpose of calculating the motion of the shots. In the experiments, the time interval between two frames is set to 0.2 seconds. This value can be adaptively computed based on the motion speed in the shot in our future work. Also, for the fast computation purpose, the minimum motion region size is set to 0.4 times of the shorter dimension size of the frame based on the assumption that a small region only includes a part of a moving target.

Figure 4.30 shows several detection results of motion saliency. Images in the first and third columns come from TRECVID 2010 training data set; while the ones in the second and fourth columns are from TRECVID 2010 testing data set. These images are extracted from videos containing non-static background, and some of them have complex scenes. Of note, as an unsupervised motion region detection framework, the proposed motion saliency region detection algorithm successfully identifies the main motion region in various backgrounds. This provides a good foundation for the further semantic retrieval task which views the motion regions as a kind of local information that describes the object-level texture of the shot. This may be complementary to the global information for multimedia semantic retrieval task.

To verify this assumption, a set of comparable experiments is conducted in three subsets (reference key frame, multiple-frame, and multiple-frames plus motion-region). The data set of the reference key frames is provided by TRECVID 2010 regarded as the representative frames of the video shots. The multiple-frame data set is made up of the reference key frame and four extra frames extracted in the region detection step. On each of the five frames, one motion-region is located. Therefore, motion-region detected in the multiple-frame data set plus the multiple-frame data set itself form the multiple-frames plus motion-region data set. The experimental design aims to check whether the local motion region features can complement the global texture features to



(a) Airplane-flying

(b) Animal

(c) Bicycling

(d) Dancing

(e) Hand

(f) Sports

(g) Running

(h) Walking

Figure 4.30: Some results of motion saliency region detection

Table 4.11: MAP Comparison when different numbers of results are requested (%) - YCbCr Features

YCbCr	5	10	20	50	100
RKF	54.48	59.95	53.64	51.03	49.09
MF	54.06	58.02	58.56	56.74	53.40
FR	77.01	79.60	78.42	75.93	72.38
Impr. % to RKF	41.35	32.78	46.20	48.79	47.44
Impr. % to MF	42.45	37.19	33.91	33.82	35.54

Table 4.12: MAP Comparison when different numbers of results are requested (%) - Gabor Features

Gabor	5	10	20	50	100
RKF	60.10	64.31	63.45	59.63	55.54
MF	59.48	60.95	62.03	59.16	56.62
FR	64.38	64.96	68.42	72.49	71.53
Impr. % to RKF	7.12	1.01	7.83	21.57	28.79
Impr. % to MF	8.24	6.58	10.30	22.53	26.33

enhance multimedia semantic retrieval.

Since the video shots in the TRECVID 2010 data set are color ones, we use YCbCr features, together with Gabor, HOG, and LBP features, to evaluate the retrieval performance. For the YCbCr features, the frame or region is first converted to the YCbCr color space from the RGB color space. The frame or region is then divided into nine blocks. Mean, variance, skewness, and kurtosis are calculated on Y, Cb, and Cr components, respectively. Considering the mean, variance, skewness, and kurtosis calculated on Y, Cb, and Cr components of the global frame, there are totally 120 features that

Table 4.13: MAP Comparison when different numbers of results are requested (%) - LBP Features

LBP	5	10	20	50	100
RKF	49.01	50.73	51.18	51.50	51.33
MF	44.69	48.28	50.37	52.14	52.01
FR	67.50	68.33	67.36	60.34	55.92
Impr. % to RKF	37.73	34.69	31.61	17.17	8.94
Impr. % to MF	51.04	41.53	33.73	15.73	7.52

Table 4.14: MAP Comparison when different numbers of results are requested (%) - HOG Features

HOG	5	10	20	50	100
RKF	74.44	70.69	66.06	59.95	55.34
MF	65.76	63.45	59.91	55.55	52.52
FR	80.28	79.32	70.20	68.42	65.72
Impr. % to RKF	7.85	12.21	6.27	14.13	18.76
Impr. % to MF	22.08	25.01	17.18	23.17	25.13

are obtained from each frame or region. Logistic regression method is used to fuse the similarity scores of multiple-frame as well as multiple-frames plus motion-region.

Tables 4.11, 4.12, 4.13, and 4.14 show the MAP values when retrieving 5, 10, 20, 50, and 100 shots in the RKF, MF and FR data sets. RKF means the reference key frame data set; MF means the multiple-frame data set including the reference key frame and four neighbor frames; and FR is the union of multiple-frame and motion-region data set, including multiple-frame with the motion-region obtained from the multiple frames. Though using different features, the retrieval results are consistent among the three kinds of data sets. One interesting observation is that using multiple frames does not necessarily get better retrieval results than using single frames. The results of MF do not always outperform those of RKF at different numbers of the retrieval shots, which indicates that sometime a representative frame could provide better texture information to improve the concept retrieval performance than those from multiple frames. On the other hand, FR outperforms both MF and RKF on all four features. This verifies that the motion region has the concept-related information that can be utilized in the semantic retrieval area. When comparing the MAP values in the same data set among Tables 4.11, 4.12, 4.13, and 4.14, the YCbCr and Gabor features keep a stable retrieval performance with different numbers of retrieval instances, i.e., MAP does not change much, even a little increasing using the Gabor features. However, using the LBP and HOG features, the retrieval performance may experience a significant drop when retrieving more instances. Considering the different performances of the features, using multiple features in semantic retrieval system to improve the performance seems to be a good choice.

4.3.4 Conclusions

Inspired by the biological mechanisms of human visions that motion saliency attracts more attention than other low-level visual features in videos, a new semantic retrieval framework for videos in non-static background is proposed, based on a novel motion saliency region detection algorithm. This framework defines a center-surround coherency model to describe the motion contrast computed by the motion vectors obtained via the optical flow algorithm, and it utilizes the integral density algorithm to calculate the global optical minimum coherency as the motion region in the frame. Further, our semantic retrieval framework integrates the global texture and local motion information obtained from the proposed motion region detection method in order to enhance the existing retrieval framework that uses only the global features.

4.4 Combining Object and Scene Information for Moving Object Retrieval

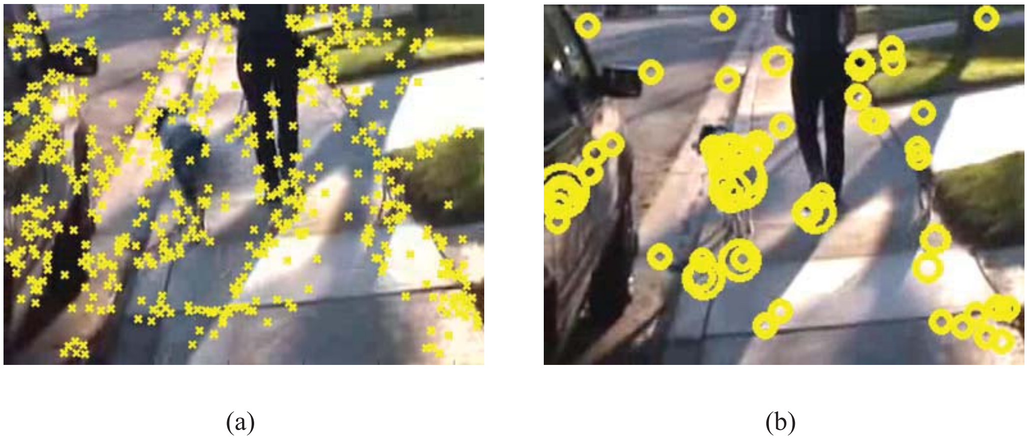
In many video clips, moving objects not only can be identified by features extracted from region of object, but also can be inferred from the surrounding environment. For example, if the background is a sky in the video, “bird” becomes a possible moving object; otherwise, if the environment is soccer field, “soccer player” is more possible appeared than “bird” within the scene. Based on this observation, the features from the object and the surrounding scene may be complementary to each other on information representation and consequently can be combined together to help the video retrieval.

An integrated detection framework of moving object is proposed in this section to analyze motion information in space and time dimension. Specially, a two-layer moving object detection method is presented to decrease the influence of dynamic background. The first layer, temporal-centered estimation presented in section 4.1, analyzes the motion information in temporal dimension and preliminarily estimates the region of motion (ROM) in the video sequence. The second layer, spatial-centered estimation presented in section 4.2, further analyzes motion information based on the preliminary ROM and finalizes the location of ROM. The proposed algorithm comprehensively integrates the motion information in spatio-temporal space in an unsupervised manner, and is robust to non-static scene and camera motion.

4.4.1 Two-Layer Detection Model of Region of Motion (ROM)

The proposed two-layer detection framework aims to maximize the complementary features and obtain accurate ROMs in uncontrolled video sequences. The proposed detection framework of ROM is shown in Figure 4.32. Temporal-centered and spatial-centered estimation methods proposed in section 4.1 and section 4.2 enable to detect moving object in video sequences. Due to the different avenues of obtaining motion information, the two methods work subject to different constraints. Spatial-centered estimation method gets motion information by computing difference between two adjacent key frames. If the moving object moves at a low speed, the difference between key frames is not distinct. This would consequently affect the detection performance.

On the other hand, the spatio-temporal interest points returned by Harris3D detector are sparse, providing limited locally motion hints of moving object. Figure 4.31 shows the interest point distribution on a sample frame. Difference of Gaussians edge detector is sensitive to edge on the frame, so that it is more easy to be affected by background



(a)

(b)

Figure 4.31: Comparison of interest point distribution. (a) interest points detected by difference of Gaussians edge detector are illustrated in yellow crosses; (b) interest points detected by Harris3D detector are illustrated in yellow circles.

clutter. Harris3D detector finds interest points in the spatio-temporal domain, therefore it keeps the space-time sudden changes and provides less interest points than difference of Gaussians edge detector. Based on the observations of the characteristics of the two detectors, the proposed two-layer detection framework therefore first narrows down the motion searching space by utilizing the information from temporal-centered estimation method. Then, using locally spatial information to estimate the ROM. The two-layer strategy employs the complementary advantages of spatial and temporal approaches, and achieves improved performance comparing with the single estimation approach presented in section 4.1 and section 4.2.

The first layer is called global motion estimation layer, in which interest points in spatio-temporal space are detected by Harris3D detector. The mechanism of Harris3D detector has been discussed in section 4.1.1. Comparing with the optional mechanism for spatio-temporal scale selection proposed in [68], [3] shows promising results by using interest points extracted at multiple scales based on a regular sampling of the scale parameters σ and τ. Integral density algorithm combines interest points and optical

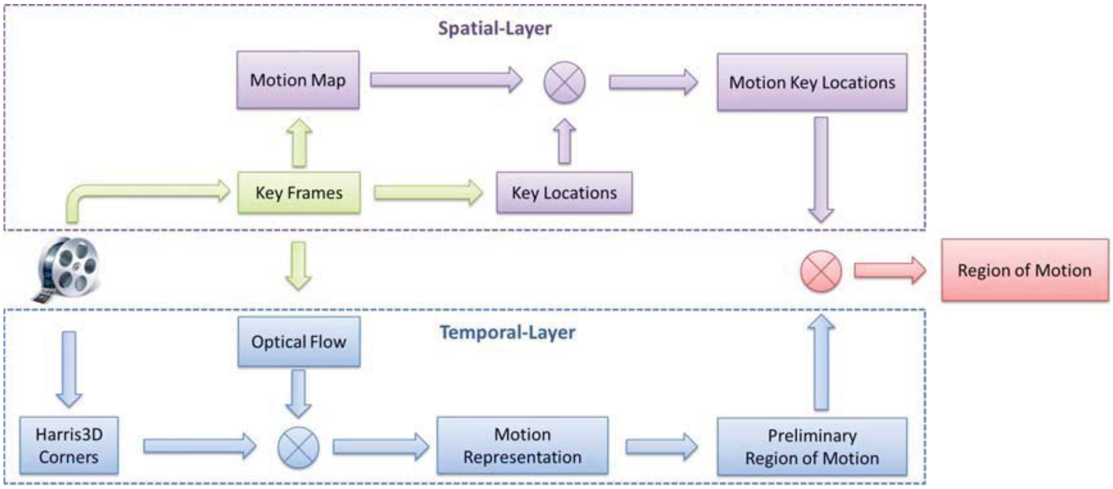


Figure 4.32: The proposed two-layer framework

flow computed from key frames to give a general motion estimation. The goal of the first layer is to comprehensively analyze the general motion cues in video sequences to identify a broad region of motion. The layer aims to filter non-static background and provide high-quality base to the second layer. Figure 4.33(a) shows three example results given by global motion estimation layer which filters the background and keeps the generated region of motion (illustrated in yellow box).

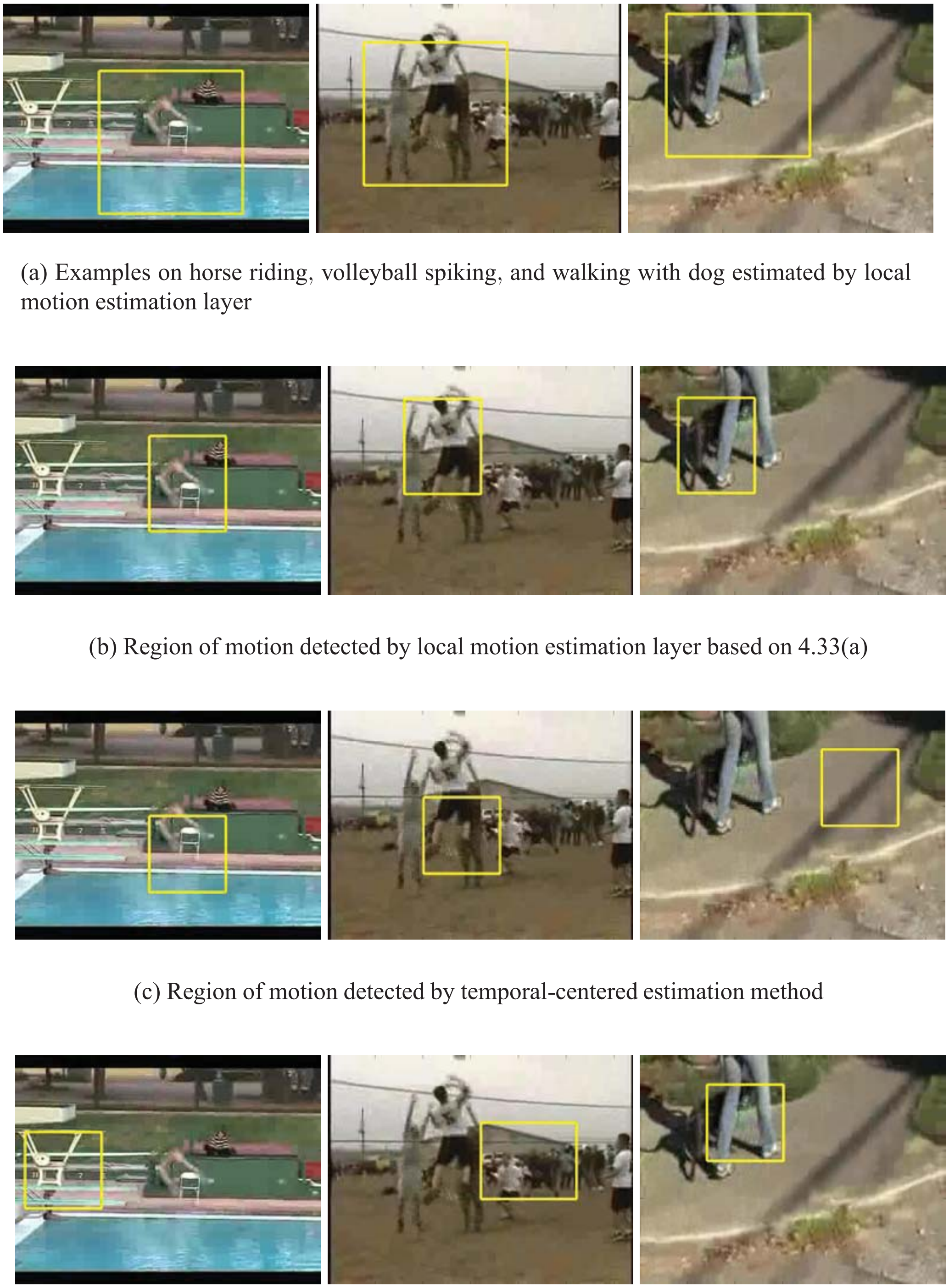
The second layer is named as local motion estimation layer, which analyzes motion key locations on the generated region of motion obtained by the global motion estimation layer and identifies the final location of moving object. Figure 4.33(b) illustrates the final region of given by global motion estimation layer to filter the background and keep the generated region of motion. Temporal-centered method presented in section 4.1 returned unsatisfied results of certain videos as shown in Figure 4.33(c). On the other hand, as shown in Figure 4.33(d) spatial-centered method proposed in section 4.2 also fails in most cases. Of note, the detection results in Figure 4.33(c) and Figure 4.33(d) are complementary. Temporal-centered method fails on the third example (“walking with a dog”) while achieves relatively better results on the first two videos (“diving” and “volleyball spiking”) than the spatial-centered method. The latter obtains a better result on the third video than the former. The proposed two-layer framework effectively utilizes the complementary characteristics of the two methods to improve the detection performance of moving object.

The new framework also works well on videos on which temporal-centered estimation method achieved good results. We employ two-layer detection framework to re-test the samples on which temporal-centered estimation method achieved good detection results that illustrated in Figure 4.34 - 4.36. The comparison results of two-layer framework and temporal-centered estimation method in Figure 4.34 - 4.36 show similar detection results. It verifies that the two-layer framework could not only successfully find ROM in some cases that temporal-centered estimation method fails, but also have similar convincing performance as temporal-centered estimation method in most videos.

4.4.2 Moving Object Retrieval Using Object and Scene Features

In many cases, people can recognize objects from not only the appearance, but also the properties of the surrounding scenes. For example, if the environment of the video sequences is a pool, it is more possible to have a person diving there than bicycling. The video sequence will not be ranked to top if the query is “bicycling”. In this section, the features from scenes are exploited to the retrieval model to help object retrieval.

We aims to apply the proposed retrieval framework to the real-world videos, i.e. videos recorded under controlled environments. The videos are usually weakly labeled (class-level). One class label is for the whole video sequence. The exactly time interval of the appearance is not available. More than one moving objects may be present in video sequences, and only a subset of detected regions is related to the retrieval tar-



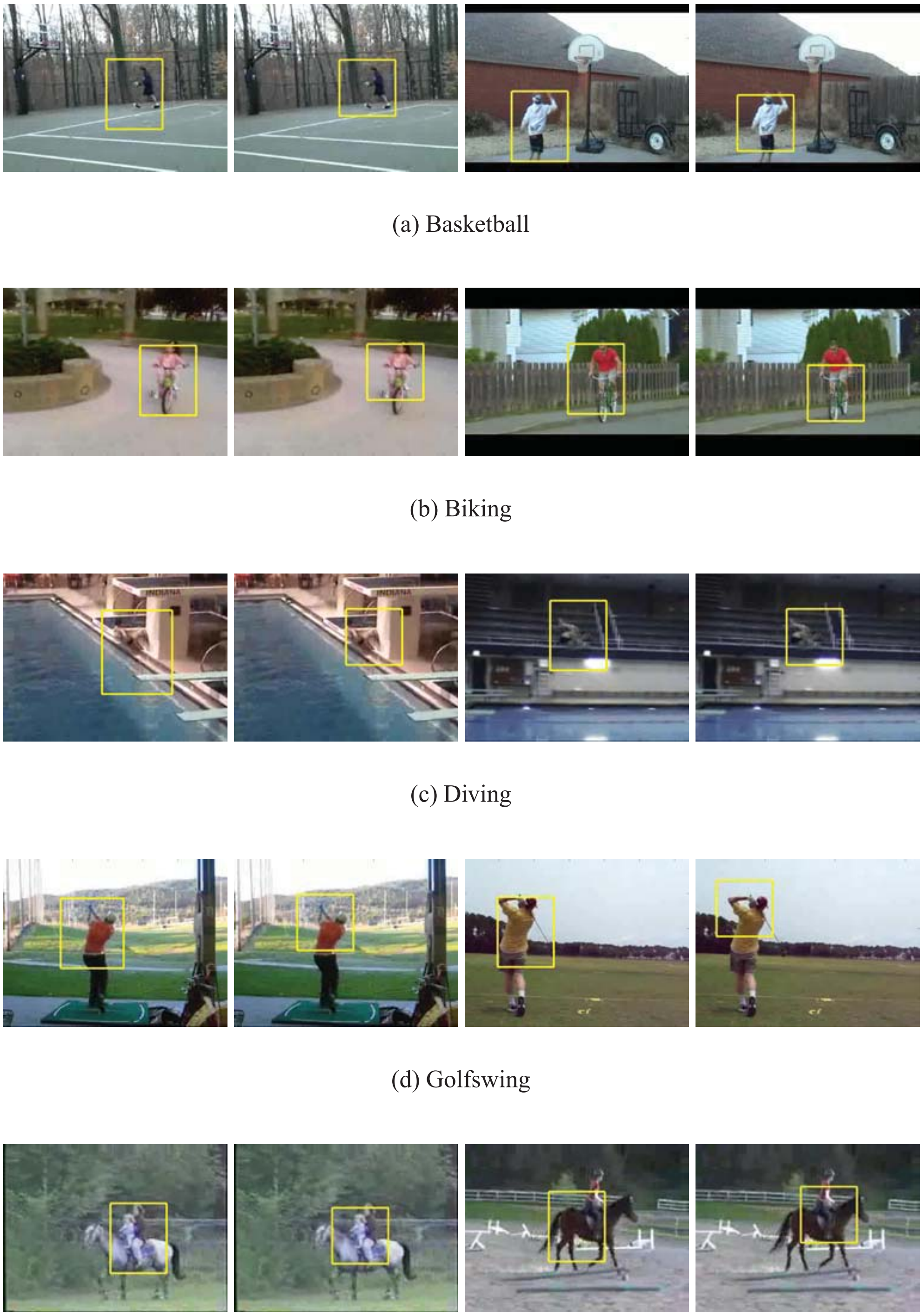
(a) Examples on horse riding, volleyball spiking, and walking with dog estimated by local motion estimation layer

(b) Region of motion detected by local motion estimation layer based on 4.33(a)

(c) Region of motion detected by temporal-centered estimation method

(d) Region of motion detected by spatial-centered estimation method

Figure 4.33: Illustration of comparison of temporal-centered method and two-layer framework



(a) Basketball

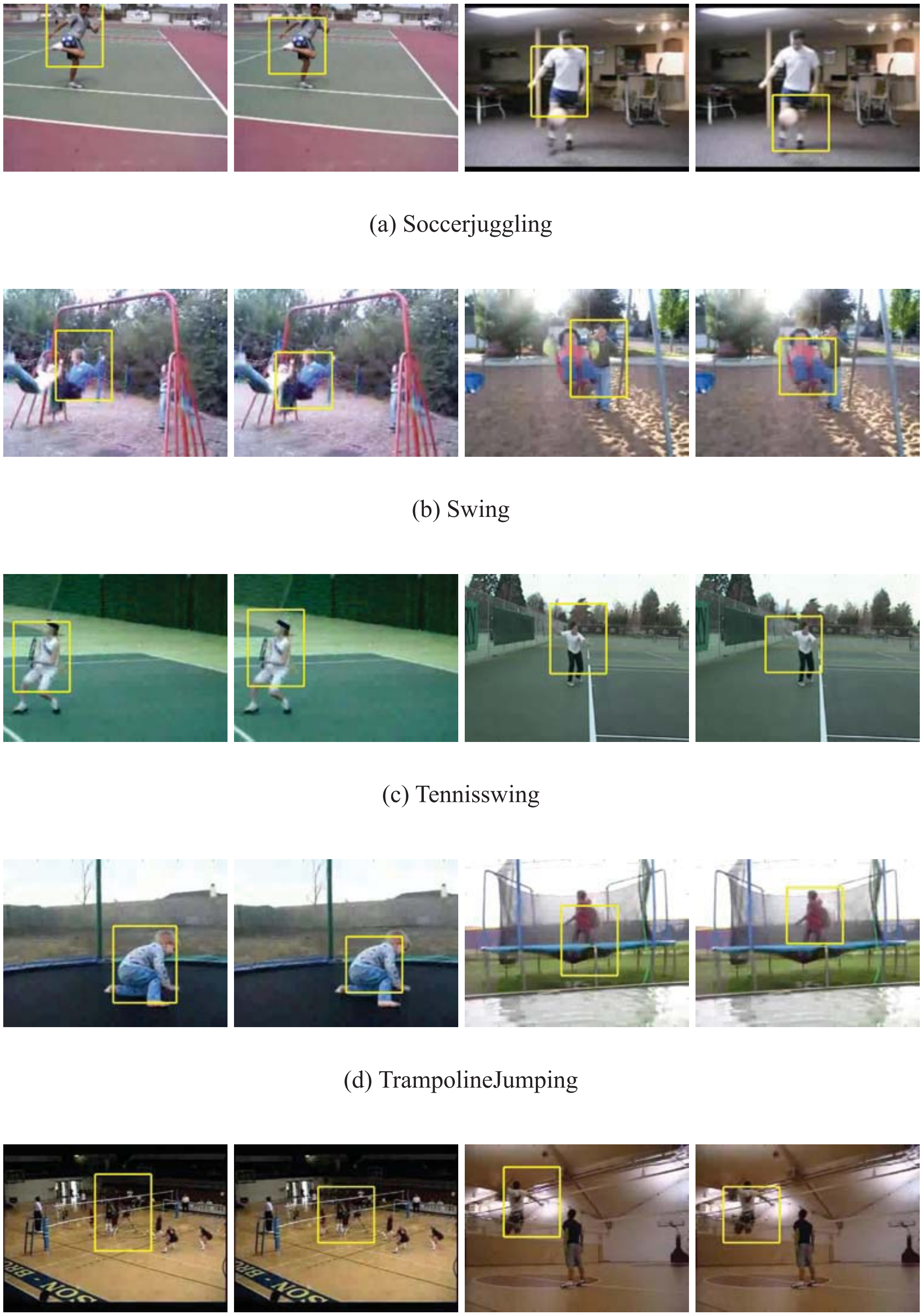
(b) Biking

(c) Diving

(d) Golfswing

(e) Horseriding

Figure 4.34: Comparison of detection results of the proposed two-layer moving object detection method (left) and temporal-centered estimation method (right) in UCF11 data set.



(a) Soccerjuggling

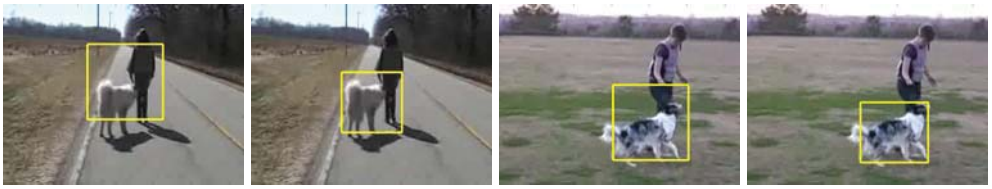
(b) Swing

(c) Tennisswing

(d) TrampolineJumping

(e) VolleyballSpiking

Figure 4.35: Comparison of detection results of the proposed two-layer moving object detection method (left) and temporal-centered estimation method (right) in UCF11 data set.



(a) Walkingwithdog

Figure 4.36: Comparison of detection results of the proposed two-layer moving object detection method (left) and temporal-centered estimation method (right) in UCF11 data set.

get. Meanwhile, the quality of videos is not adequate and guaranteed. The camera is non-stationary in many videos captured by an amateur. Moreover, the background usually is complex and cluttered with poor illumination condition. If the foreground is moving, it is very difficult to identify the semantic meaning delivered by the moving object. Under these challenges, we propose to extract features of moving object and scene, respectively. The complementary information from moving objects and scene is integrated for capturing relationships to enhance the retrieval performance.

Some of the recent works have explored the possibility of integrating features of moving object and surrounding scene to improve object recognition performance. Reddy *et al*. [41] used optical flow to give a rough estimate of the velocity at each pixel given two consecutive frames. They, then, applied a threshold on the magnitude of the optical flow to decide if the pixel was moving or stationary. The stationary pixels were regarded as background, while the moving pixels were viewed as the ROM. This method performs well in videos with static scenes, but the strategy was not suitable for the realistic videos with the unconstrained background. Ikizler-Cinbis *et al*. [40] estimated the location(s) of the person(s) by using the human detector proposed by Felzenswalb *et al*. [4]. To fill the gap in which the person detector did not fire due to the motion blur and pose variations, the mean-shift tracking method was used to locate the person in every frame [42]. The work considered any moving region as a “candidate object”, and then found the associated tracks and the corresponding features from each track. The approach, to some degree, was able to capture the human and object features in the video. However, from the illustrated examples shown in the paper, the detected regions of object inevitable included noisy from region of person and background. The paper did not provide an effective solution to solve the issue.

Our proposed retrieval framework is set up on the basis of the presented two-layer moving object detection framework. We use the bounding box as the lines to partition foreground and background. The interest points on foreground and background are detected by Harris3D detector. HOG/HOF descriptors [117] are employed to compute the histograms of spatial gradient and optical flow accumulated in spatio-temporal neighborhoods of interest points detected by Harris3D detector. For the combination of HOG/HOF descriptors with interest point detectors, the descriptor size is defined by $\Delta_x(\sigma) = \Delta_y(\sigma) = 18\sigma$, $\Delta_t(\tau) = 8\tau$. Each volume is subdivided into a $n_x \times n_y \times n_t$ grid of cells. For each cell, 4-bin histograms of gradient orientations (HOG) and 5-bin histograms of optic flow (HOF) are computed. Normalized histograms are concatenated into HOG/HOF descriptor vectors. Peer work names the feature as STIP features. We use the implementation codes on-line and standard parameter settings $\sigma^2 = 4, 8, 16, 32, 64, 128$, $\tau^2 = 2, 4$,n_x, n_y = 3, n_t = 2. This choice is motivated by the reduced computational complexity, the independence from scale selection artifacts and the recent evidence of good recognition performance using dense scale sampling [3].

4.4.3 Experimental Results and Analyses

We test the proposed retrieval framework using UCF11 data set collected by Liu *et al* [1]. UCF11 is a large data set containing 1168 videos of human action. Videos in UCF11 are divided into 11 categories; they are basketball shooting, bicycling, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. The rich categories make it a particularly suitable data set to learn the integration of moving object and scene. Meanwhile, quite challenges are existed in the data set, such as lots of camera movement, cluttered backgrounds, different viewing directions and varying illumination conditions. Videos for each category of motion are divided into 25 related subsets, and leave-one-out cross validation (LOOCV) is applied over these subsets, following the same evaluation methodology of work in [1].

For comparison purposes, besides STIP feature descriptor, SIFT (Scale-Invariant Feature Transform) is employed to provide video representations [95]. The descriptor represents the spatial structure and the local orientation distribution of a patch surrounding keypoints into a 128-dimensional feature vector. SIFT is regards as one of the best descriptors for keypoints [118].

The work in [41] exploited the influence of fusion strategy on the performance. The experimental results show late fusion (probabilistic fusion) achieves best performance among all fusion strategies. We apply probabilistic fusion to integrate SIFT and STIP features since the two descriptors are considered to be conditionally independent. In probabilistic fusion the individual probabilities are multiplied and normalized. For object and scene fusion, we use sum-rule to combine the results from classifier (e.g. LibSVM in section 4.1.3).

Table 4.15 summarizes the overall quantitative performance on UCF11 data set. The classification accuracy shown in the Table 4.15 is normalized with respect to the number of videos for each motion category.

Table 4.15: Overall performance evaluation of single and integrated feature channels

	% Correct classification using single feature channels												
	Bas	Bic	Div	Gol	Hor	Soc	Swi	Ten	Tra	Vol	Wal	Avg.	Std.
STIP-object	50.4	77.9	79.4	84.5	82.2	61.5	76.6	64.1	79.8	76.7	58.5	72.2	10.8
SIFT-object	55.5	70.34	85.6	73.9	76.7	21.2	52.6	61.1	57.1	83.6	67.5	64.1	17.2
CSIFT [41]	56.0	49.0	81.0	80.0	55.0	56.0	55.0	59.0	61.0	71.0	36.0	59.9	12.7
Gradient [41]	38.0	60.0	94.0	72.0	72.0	12.0	52.0	47.0	75.0	87.0	51.0	60.1	22.4
STIP-scene	51.8	79.3	81.1	85.2	83.3	64.7	79.6	65.9	79.0	75.9	61.0	73.3	10.3
SIFT-scene	59.9	74.5	86.3	76.8	80.2	20.5	58.4	59.9	58.8	83.6	66.7	66.0	17.5
GIST [40]	38.4	60.7	69.0	61.0	66.0	9.0	42.0	61.0	54.0	81.0	43.1	53.2	18.5
Color [40]	33.3	44.8	86.0	65.0	43.0	22.0	27.0	47.0	57.0	73.0	43.9	49.3	18.6
	% Correct classification using integrated feature channels												
Object+Scene	59.9	86.9	87.6	90.1	85.8	60.3	78.1	64.7	69.8	91.4	74.8	77.2	11.5
Ikizler [40]	48.5	75.2	95.0	95.0	73.0	53.0	66.0	77.0	93.0	85.0	66.7	75.2	15.3
Reddy [41]	55.0	67.0	98.0	89.0	83.0	49.0	67.0	68.0	76.0	92.0	59.0	73.2	15.2

The first eight rows of Table 4.15 show the performance of the individual feature channels. Of note, STIP-object and STIP-scene are STIP features extracted from the region of object and scene of video sequences, respectively. SIFT-object and SIFT-scene are SIFT features extracted from the region of object and scene of frames, respectively. CSIFT and Gradient denote results of color SIFT and motion features shown in [41]. GIST and Color denote results of scene features in [40]. The results show that, even using the single feature channel SIFT-object gives 65.95% average accuracy in the data set, whereas the STIP-scene feature is able to perform with 73.33% average accuracy. These numbers are obviously high. This observation shows that using scene features can provide a great deal of useful information of the possible motion, especially in this data set. For instance, for the motion of soccer juggling and swinging, STIP-scene features achieve 64.74% and 79.56% accuracy, whereas for basketball shooting, the SIFT-scene features give 59.85%. These results suggest that when the person is less visible, the scene features can be used for helping deciding the category of possible motions.

The second part of Table 4.15 is the overall combination results compared with peer work. We notice that the proposed integration method brings an improvement over the peer work (results in [40] is the best reported performance in the data set). The average improvement is higher than the peer work in [41] and [40] (5.5% and 2.6%, respectively). Also, the standard deviations of our method are much smaller than the peer work. This means the proposed integration gets better performance on difficult categories (e.g., Basketball shooting, soccer juggling, walking with a dog); while keeps good accuracy on categories.

Table 4.16 compares the overall accuracies between the proposed integration method and the state-of-the-art methods on UCF11 data set. The experimental results demonstrate that object and scene properties are informative and complementary to each other. Different feature channels bring diversely useful information for the identification of moving objects.

Table 4.16: Accuracy comparison of the proposed method and state-of-the-art approaches

Method	Ikizler [40]	Reddy [41]	Liu [1]	Proposed method
Accuracy(%)	75.21	73.2	71.2	77.2

4.4.4 Conclusions

In this section, we first present a two-layer framework for moving object detection in video sequences. The framework contains global motion estimation layer and local motion estimation layer. The detection framework is unsupervised which means we bound the moving object in the videos without previously knowing what object it is. The global motion estimation layer analyzes the motion information and returns a preliminarily region to the second detection layer. The final bounding box of the moving object is obtained from the local motion estimation layer. The design aims to complement the advantages of two kind of motion estimation methods presented in section 4.1 and section 4.2. Temporal method works with videos having low resolution and under unstable camera conditions (e.g. YouTube videos), since the method has good noise-tolerant characteristics. At local motion estimation layer, spatial method could provide more detailed motion map for local motion estimation.

In addition, we explore object and scene properties to use complementary features of each other for the better motion identification. The scene helps when the moving object is captured from a far distance and the distinct features of the object can not be fully visible. In that case, the surrounding scene can provide a hint for the identification of the moving object. Our results verify the effectiveness of integration of the moving object and scene. The proposed integration framework can be extended to handle more feature channels. The framework could also be extended to the field of human activity identification. The relationship of object and human body can be exploited as the work in [40]. On the other hand, the object and scene information could be integrated on feature level. It means more features have to be considered and combined. In that case, feature selection approaches need to be involved in our future works for choosing discriminant features to improve retrieval performance [119, 120, 121, 122].

Chapter 5

Conclusions and Future Work

In this chapter, we summarize the proposed solutions of moving object detection and retrieval in video sequences in section 5.1. The contributions are concluded and highlighted. In section 5.2, we discuss the current limitations in the proposed framework, and future work is proposed.

5.1 Conclusions

We have presented a two-layer detection framework to automatically detect moving objects in the spatio-temporal domain. The global motion estimation layer employed the temporal-centered method for a preliminary motion estimation. It used Harris3D corner key points and the optical flow field to represent the motion message in the video stream [90]. A center-surround coherency model was proposed to estimate the local motion and included in the integral density algorithm to find the preliminary regions having a high motion energy. The global motion estimation layer found the preliminary motion region for the local motion estimation layer. The spatial-centered estimation algorithm used in the local layer extracts the texture information from the visual key frames. The temporal information on adjacent frames was presented by the proposed motion map which was calculated from the visual change of the result generated from the SPCPE algorithm from the key frames [94]. The motion map then filtered the key locations to generate the motion-related key points which were used as the input to the proposed integral density algorithm to find the motion region with the highest density of motion points.

To enhance the proposed spatio-temporal object detection algorithm, a new key frame detection method was proposed to extract the representative frames as the base frame for the motion information computing [123]. Informative regions were defined and utilized by a modified clustering technique to select a set of key frame candidates (KFCs), while transitive regions were not used for key frame extraction. The final set of key frames was adaptively determined from the speed of the visual change within a video shot and between video shots. Another algorithm was proposed to solve the object detection under occlusion situations. The approach extended an existing algorithm that uses the preliminary foreground estimation result and object detection information from the adjacent frames to identify the locations of the moving object by an n-steps search (NSS) method, followed by a size-adjustment method that adjusts the bounding boxes of the objects [89].

The proposed framework of moving object detection has been verified to enable the enhancement on the video recognition and retrieval by providing the object-level features. The main contribution of the recognition work is to prove that using a subset of features (extracted from the region of the object) to train the global Gaussian mixture models can achieve a high accuracy. The method saved a lot of off-line training time compared to when using the whole set of feature from the video streams. On the other hand, the integration of scene and object-level features can further improve object retrieval and recognition performance.

5.2 Future Work

Although we proposed several solutions on the topic of moving object detection in video sequences, the framework can be further improved from various perspectives. For example, the information of the relationships and interactions of multiple objects in the video sequence can be mined to help object recognition. Another potential work is to use the information of moving objects and background to estimate the static objects in the environment. For instance, we can build a general system consisting of moving object detection, static object detection, object relationship estimation, and background estimation. We give a blueprint for the future work on the basis of the proposed solutions in this section.

5.2.1 Integration of Tracking and Detection into Framework

Current detection model applies a simple tracking strategy [124] to track the detected moving object. The strategy is used to deal with the situation that a more active object comes in view. Without tracking function, the detection model will lose the position of the previous object as shown in Figure 5.1. At the first frame, only one player is playing the volleyball. From the second frame, another player enters the screen at a higher speed. The detection model returns the most action region as the location of the moving object, so that the position of the first player is lost.

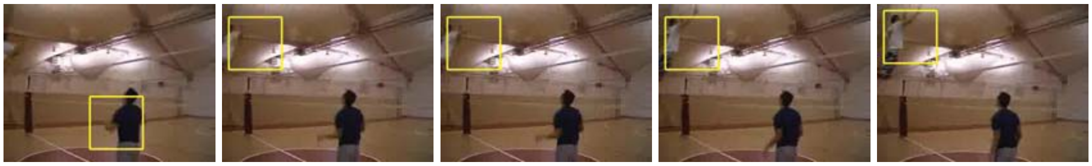


Figure 5.1: Moving object detection results without the tracking in a sampled video sequence (frame sampling rate is 3 fps)

After involving tracking component into the detection model, the position of every detected object can be well kept in the following sequences as shown in Figure 5.2. We define the following criterion to checking whether a new object comes in view.

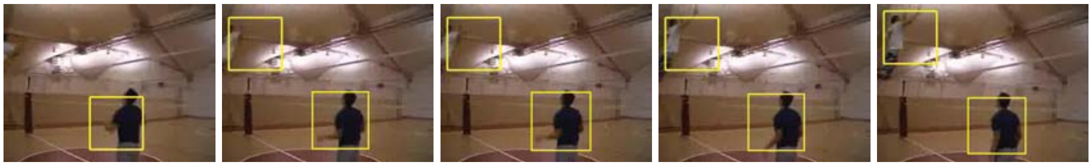


Figure 5.2: Modified object detection results with the tracking

$$dist(ctr_t - ctr_d) = \|ctr_t - ctr_d\|_2 \geq \delta,$$

where ctr_t is the centroid of the bounding box from the tracking component, ctr_d is the centroid of the bounding box from the detection model, and δ is an empirical distance (using 20% of the width of video in the experiments). Figure 5.3 shows an example of the tracking result (in yellow box) and detection result (in red box). The blue line illustrates the Euclidian distance between centroids of the two boxes. If the distance is greater than δ, the boxes are regarded as two objects; otherwise, the detection result is used as the region of moving object and the tracking result is omitted.

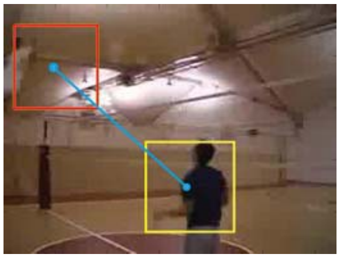


Figure 5.3: Checking whether new object appears

Actually, both of the tracking results and the detection results could provide information of the moving objects. The integration of the two information channels should be able to provide complementary information and enhance the detection performance.

In the future, the previous detection and tracking results in the same sequence can be involved in giving an initial location of the moving object in the current frame. Assume the motion status does not change too much in the continuous video sequences, the information from the previous frames may give a good start up to the current detection in the same video sequence. The idea can not only save the processing time, but also decrease influence caused by the local space-time visual anomalies as shown in Figure 5.4. Many state-of-the-art works have achieved successes in object tracking area, which can help quickly integrate the latest developments into our framework [125, 126].

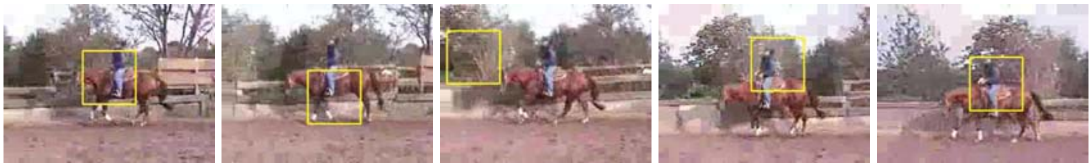


Figure 5.4: Moving object detection in a sampled video sequence (frame sampling rate is 3 fps)

5.2.2 Relationship and Interaction Mining Among Multiple Objects

The realistic videos usually contain multiple objects at the same time which have different moving directions and many occlude each other. The proposed detection model currently only returns one motion region with the highest motion contrast than other foregrounds. As shown in Figure 5.5, the model is confused by the two moving objects because it is designed for detecting only one moving object. To generalize the proposed detection and retrieval model, a detection method of multiple objects has to be included. For scenarios having more than one moving object in the scene, the less active ones should also been considered in the model. One of the solutions is to include a tracking component to track detected object, by doing that the framework enables to handle the new object coming in view while keeps tracking the previous objects. Another possible approach is to set up a criterion to select the proper motion regions. For example, the distance between the two centroids of the motion regions can be used as a threshold to decide whether the two regions describe the same object. Once the information of the multiple objects is available, the relationship and interaction can be mined and bring informative messages to facilitate many research topics, such as human activity recognition. The work in [127, 128] used the distance between two persons as an important feature to evaluate the actions. Besides the distance between objects, we hope to mine more information among multiple objects to enhance performance in video retrieval. The multiple objects in the same video may belong to the different concepts, whose properties can help the recognition of each other. For example, two objects are detected in a video. If one object is recognized as a basketball, the chance of the other object being a player increases. Recently, many research progresses have been achieved in the area of the association information integration for semantic concept detection [129, 130, 131, 132, 133]. These works analyze and utilize the information of the co-occurred concepts to improve the concept retrieval performance.

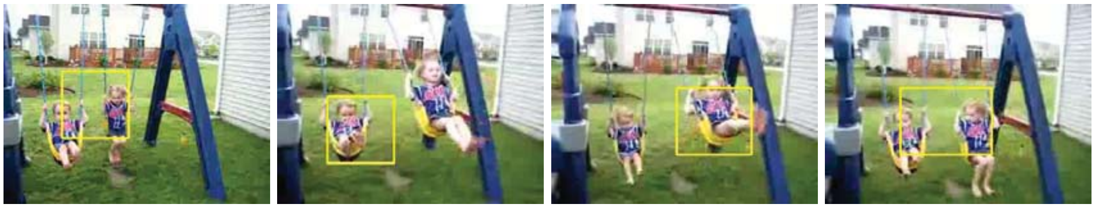


Figure 5.5: Moving object detection in a sampled video sequence with two objects

Sometimes an occlusion happens between objects. The proposed object tracking method under the occlusion situation can be employed to solve the issue [89]. When the objects split, a simple tracking strategy is sufficient to locate the objects. On the other hand, the current detection model only considers motion contrast in the frame sequences, which is not applicable for processing videos with only static objects and scenes. The static object and scene segmentation issue is usually solved by converting a 2D video or multiple images into a 3D model, requiring the techniques of camera calibration and depth map estimation. If the visual change between frames happens only due to the depth dimension, the results of the proposed methods are not ideal. The information of foreground (moving objects) should be able to help detect the non-moving objects in the non-static background if combined with other computer vision and image processing technologies. For example, the appearance of basketball players could raise the probability of basketball field and backboards as shown in Figure 5.6.

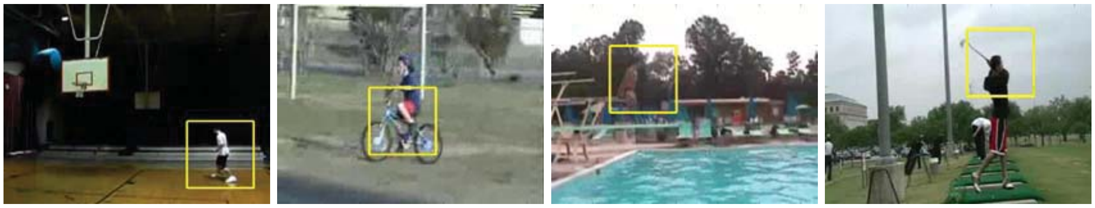


Figure 5.6: Examples of frames containing non-moving objects

5.2.3 Design of Motion-Driven Key Frame Detection Algorithm

The proposed key frame detection model is driven by the visual change between the frames, which achieved pretty good performance compared to the state-of-the-art approaches. The current model provides key frames as the input of the moving object detection framework. Therefore, the quality of key frames affects the performance of the detection framework. If the model can be driven by the motion message in the video sequence, it would serve the object detection model better. That is, the model could intensively analyze the frames with a moving object in the center of the frame. Figure 5.7 is an example to illustrate the good and bad key frames from the angle of the detection model. The five frames are selected from the same video sequence as key frames to represent the content of the video sequence. From the object detection perspective, obviously the third one is a proper one for the input of the object detection model since the appearance and size of the object in Figure 5.7 is suitable for detection

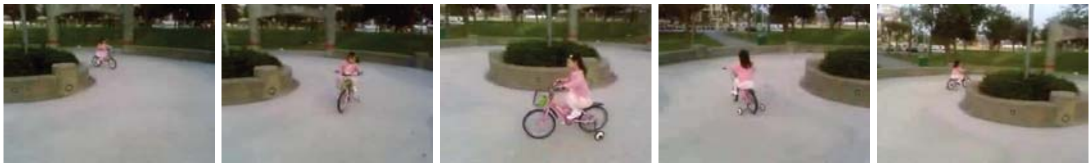


Figure 5.7: Examples of key frames extracted from a video sequence

and recognition. The objects in the second and fourth frames can be easily detected since it is not occluded by other foreground. But they are not proper for the recognition due to the blur appearance and the small size of the object. The first and the last frames in Figure 5.7 should be avoid cases in the key frame extraction. The objects are too small to be detected and recognized by the framework. The significance of the key frame extraction model in the proposed framework is clearly shown in Figure 5.7. If the key frame extraction model could consider the location of the object to select the key frames, it would definitely improve the performance of the detection and recognition framework. On the other hand, a motion-driven key frame detection model may provide a better summarization of the video, which is another important application of online multimedia.

5.2.4 Integrate multiple channels of multimedia information for video retrieval

Currently work only analyzes the visual information for moving object detection. However, most video sequences provide audio channel as well. Audio data may deliver sound-related message to help the video retrieval in many concepts (e.g., baby crying, explosion, applause). In addition, users could also provide text information (e.g., title, tags) when uploading videos to the social networks and the Internet. The text data could deliver information which is difficult to obtain from visual and acoustic channels such as date, time, location, etc [134, 135]. Multiple channels of multimedia information may complement each other to achieve better retrieval performance [136, 137, 138, 139]. We can refer existed research works to analyze the complementary information in various channels to enhance our proposed video retrieval framework [140, 141, 142, 143]. Furthermore, we can consider to integrate the multiple information channels into the demo video retrieval system shown in Figure 3.2. That would give a more comprehensive evaluation of the video retrieval system comparing with the state-of-the-art web-based multimedia retrieval system [144, 145, 146].

Bibliography

[1] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1996–2003.

[2] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[3] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[5] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *Proceedings of the 14th annual ACM international conference on Multimedia*, 2006, pp. 815–824.

[6] M.-L. Shyu, C. Haruechaiyasak, and S.-C. Chen, "Category cluster discovery from distributed www directories," *Information Sciences*, vol. 155, no. 3, pp. 181–197, 2003.

[7] M.-L. Shyu, S.-C. Chen, M. Chen, and C. Zhang, "A unified framework for image database clustering and content-based retrieval," in *Proceedings of the 2nd ACM international Workshop on Multimedia databases*, no. 13, 2004, pp. 19–27.

[8] M.-L. Shyu, S.-C. Chen, M. Chen, C. Zhang, and K. Sarinnapakorn, "Image database retrieval utilizing affinity relationships," in *Proceedings of the 1st ACM international workshop on Multimedia databases*, 2003, pp. 78–85.

[9] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys (CSUR)*, vol. 40, no. 2, pp. 1–60, 2008.

[10] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of art and challenges," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 2, no. 1, pp. 1–19, 2006.

[11] L. Lin, C. Chen, M.-L. Shyu, and S.-C. Chen, "Weighted subspace filtering and ranking algorithms for video concept retrieval," *IEEE Multimedia*, vol. 18, no. 3, pp. 32–43, 2011.

[12] M. Chen, S.-C. Chen, M.-L. Shyu, and K. Wickramaratna, "Semantic event detection via temporal analysis and multimodal data mining," *IEEE Signal Processing Magazine, Special Issue on Semantic Retrieval of Multimedia*, vol. 23, no. 2, pp. 38–46, October 2006.

[13] L. Lin and M.-L. Shyu, "Effective and efficient video high-level semantic retrieval using associations and correlations," *International Journal of Semantic Computing*, vol. 3, no. 4, pp. 421–444, December 2009.

[14] Z. Peng, Y. Yang and et al., "PKU-ICST at TRECVID 2009: High level feature extraction and search," in *Proceedings of TRECVID 2009 Workshop*, November 2009.

[15] M.-L. Shyu, Z. Xie, M. Chen, and S.-C. Chen, "Video semantic event/concept detection using a subspace-based multimedia data mining framework," *IEEE Transactions on Multimedia, Special Issue on Multimedia Data Mining*, vol. 10, no. 2, pp. 252–259, February 2008.

[16] A. Hauptmann, M. Christel, and R. Yan, "Video retrieval based on semantic concepts," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 602–622, April 2008.

[17] L. Lin, "Multimedia data mining and retrieval for multimedia databases using associations and correlations," *ACM SIGMultimedia Records*, vol. 3, no. 1, pp. 21–22, 2011.

[18] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 171–177, 2010.

[19] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, 2004, pp. 28–31.

[20] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337–1342, 2003.

[21] A. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, January 1980.

[22] J. Duncan and G. Humphreys, "Visual search and stimulus similarity," *Psychological Review*, vol. 96, no. 3, pp. 433–58, July 1989.

[23] R. Born, J. M. Groh, R. Zhao, and S. J. Lukasewycz, "Segregation of object and background motion in visual area mt: Effects of microstimulation on eye movements," *Neuron*, vol. 26, pp. 725–734, 2000.

[24] Z. Rasheed and M. Shah, "Scene detection in hollywood movies and tv shows," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. 343–348.

[25] H.-W. Kang and X.-S. Hua, "To learn representativeness of video frames," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, 2005, pp. 423–426.

[26] C. Kim and J.-N. Hwang, "Object-based video abstraction for video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1128–1138, 2002.

[27] I. Yahiaoui, B. Merialdo, and B. Huet, "Automatic video summarization," in *Proceedings of CBMIR Conf*, 2001.

[28] J. Rong, W. Jin, and L. Wu, "Key frame extraction using inter-shot information," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, 2004, pp. 571–574.

[29] M. Cooper and J. Foote, "Discriminative techniques for keyframe selection," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2005.

[30] H. S. Chang, S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1269–1279, 1999.

[31] X.-D. Yu, L. Wang, Q. Tian, and P. Xue, "Multilevel video representation with application to keyframe extraction," in *Proceedings of the 10th International Conference on Multimedia Modelling*, 2004, pp. 117–123.

[32] A. Girgensohn and J. Boreczky, "Time-constrained keyframe selection technique," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, vol. 1, 1999, pp. 756–761.

[33] L. Liu and G. Fan, “Combined key-frame extraction and object-based video segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 869–884, 2005.

[34] P. Kelm, S. Schmiedeke, and T. Sikora, “Feature-based video key frame extraction for low quality video sequences,” in *Proceedings of the 10th Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS’09)*, 2009, pp. 25–28.

[35] W. Abd-Almageed, “Online, simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing,” in *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP)*, 2008, pp. 3200–3203.

[36] G. Liu, X. Wen, W. Zheng, and P. He, “Shot boundary detection and keyframe extraction based on scale invariant feature transform,” in *Proceedings of the Eighth IEEE/ACIS International Conference on Computer and Information Science (ICIS)*, 2009, pp. 1126–1130.

[37] E. Mendi and C. Bayrak, “Shot boundary detection and key frame extraction using salient region detection and structural similarity,” in *Proceedings of the 48th Annual Southeast Regional Conference*, 2010, p. 66.

[38] L. Honghua, Y. Xuan, and P. Jihong, “Key frame extraction based on multi-scale phase-based local features,” in *Proceedings of the 9th International Conference on Signal Processing (ICSP)*, 2008, pp. 1031–1034.

[39] S.-C. Chen, M.-L. Shyu, C. Zhang, and R. L. Kashyap, “Identifying overlapped objects for video indexing and modeling in multimedia database systems,” *International Journal on Artificial Intelligence Tools*, vol. 10, no. 4, pp. 715–734, 2001.

[40] N. Ikizler-Cinbis and S. Sclaroff, “Object, scene and actions: combining multiple features for human action recognition,” in *Proceedings of the 11th European Conference on Computer Vision: Part I*, ser. ECCV’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 494–507. [Online]. Available: http://dl.acm.org/citation.cfm?id=1886063.1886101

[41] K. K. Reddy and M. Shah, “Recognizing 50 human action categories of web videos,” *Machine Vision and Applications*, pp. 1–11, 2012.

[42] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[43] T. Wittenberg, M. Grobe, C. M*ü*nzenmayer, H. Kuziela, and K. Spinnler, "A semantic approach to segmentation of overlapping objects," *Methods Inf Med*, vol. 43, pp. 343–353, 2004.

[44] S.-H. Kim, J.-H. Choi, H.-B. Kim, and J.-W. Jang, "A new snake algorithm for object segmentation in stereo images," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2004, pp. 13–16.

[45] Q. Zeng, Y. Miao, C. Liu, and S. Wang, "Algorithm based on marker-controlled watershed transform for overlapping plant unit segmentation," *Optical Engineering*, vol. 48, no. 2, pp. 1–10, 2009.

[46] W. Zeng and W. Gao, "Semantic object segmentation by a spatio-temporal MRF model," in *Proceedings of the 17th International Conference on Pattern Recognition*, 2004, pp. 775–778.

[47] S.-C. Chen, M.-L. Shyu, S. Peeta, and C. Zhang, "Spatiotemporal vehicle tracking: The use of unsupervised learning-based segmentation and object tracking," *IEEE Robotics and Automation Magazine, Special Issue on Robotic Technologies Applied to Intelligent Transportation Systems*, vol. 12, no. 1, pp. 50–58, March 2005.

[48] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle, "Appearance models for occlusion handling," *Image and Vision Computing*, vol. 24, no. 11, pp. 1233–1243, November 2006.

[49] A. N. Stein and M. Hebert, "Occlusion boundaries from motion: Low-level detection and mid-level reasoning," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 325–357, May 2009.

[50] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1531–1536, November 2004.

[51] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern recognition*, vol. 30, no. 4, pp. 643–658, 1997.

[52] X.-D. Zhang, T.-Y. Liu, K.-T. Lo, and J. Feng, "Dynamic selection and effective compression of key frames for video abstraction," *Pattern recognition letters*, vol. 24, no. 9, pp. 1523–1532, 2003.

[53] X. Zeng, W. Li, X. Zhang, B. Xu *et al.*, "Key-frame extraction using dominant-set clustering," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2008, pp. 1285–1288.

[54] F. Shi and X. Guo, “Keyframe extraction based on kmeas results to adjacent dc images similarity,” in *Proceedings of the 2nd International Conference on Signal Processing Systems (ICSPS)*, vol. 1, 2010, pp. V1–611.

[55] S.-S. Cheung and A. Zakhor, “Fast similarity search and clustering of video sequences on the world-wide-web,” *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 524–537, 2005.

[56] Y. Peng, J. Pei, and Y. Xuan, “Behavior key frame extraction using invariant moment and unsupervised clustering,” in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 5, 2008, pp. 2503–2508.

[57] D. Liu, M.-L. Shyu, C. Chen, and S.-C. Chen, “Integration of global and local information in videos for key frame extraction,” in *Proceedings of the IEEE Information Reuse and Integration (IRI)*, 2010, pp. 171–176.

[58] N. D. Doulamis, A. D. Doulamis, Y. S. Avrithis, and S. D. Kollias, “Video content representation using optimal extraction of frames and scenes,” in *Proceedings of the International Conference on Image Processing (ICIP)*, vol. 1, 1998, pp. 875–879.

[59] T. Liu and J. R. Kender, “Optimization algorithms for the selection of key frame sequences of variable length,” in *Proceedings of the European Conference on Computer Vision*, 2006, pp. 403–417.

[60] S. V. Porter, M. Mirmehdi, and B. T. Thomas, “A shortest path representation for video summarisation,” in *Proceedings of the 12th International Conference on Image Analysis and Processing*, 2003, pp. 460–465.

[61] A. Divakaran, R. Radhakrishnan, and K. A. Peker, “Motion activity-based extraction of key-frames from video shots,” in *Proceedings of the International Conference on Image Processing*, vol. 1, 2002, pp. I–932.

[62] R. Narasimha, A. Savakis, R. Rao, and R. De Queiroz, “Key frame extraction using mpeg-7 motion descriptors,” in *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers*, vol. 2, 2003, pp. 1575–1579.

[63] T. Liu, H.-J. Zhang, and F. Qi, “A novel video key-frame-extraction algorithm based on perceived motion energy model,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 10, pp. 1006–1013, 2003.

[64] P. Burt and E. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[65] J. L. Crowley and R. M. Stern, “Fast computation of the difference of low-pass transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 212–222, 1984.

[66] T. Lindeberg, “Scale invariant feature transform,” *Scholarpedia*, vol. 7, no. 5, p. 10491, 2012.

[67] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*, vol. 15. Manchester, UK, 1988, pp. 147–151.

[68] I. Laptev and T. Lindeberg, “Space-time interest points,” in *Proceedings of International Conference on Computer Vision*, 2003, pp. 432–439.

[69] N. Inoue and K. Shinoda, “A fast map adaptation technique for gmm-supervector-based video semantic indexing systems,” in *Proceedings of the 19th ACM International Conference on Multimedia*, 2011, pp. 1357–1360.

[70] G. MacLachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering*, ser. Statistics Series. Marcel Dekker Incorporated, 1988. [Online]. Available: http://books.google.com.au/books?id=1Id9QgAACAAJ

[71] D. A. Reynolds, “Gaussian mixture models,” in *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain, Eds. Springer Publishing Company, Incorporated, 2009, pp. 659–663.

[72] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[73] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[74] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, “Support vector machines using gmm supervectors for speaker verification,” *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[75] C. Charbuillet, D. Tardieu, and G. Peeters, “Gmm-supervector for content based music similarity,” in *Proceedings of the International Conference on Digital Audio Effects*, 2011, pp. 425–428.

[76] S. Sista and R. L. Kashyap, “Unsupervised video segmentation and object tracking,” *Computers in Industry Journal*, vol. 42, no. 2-3, pp. 127–146, July 2000.

[77] B. T. Truong and S. Venkatesh, “Video abstraction: A systematic review and classification,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 3, no. 1, p. 3, 2007.

[78] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z.-H. Zhou, “Multi-view video summarization,” *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 717–729, 2010.

[79] A. F. Smeaton, P. Over, and W. Kraaij, “Evaluation campaigns and TRECVid,” in *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, 2006, pp. 321–330.

[80] H.-s. Kim, J. Lee, H. Liu, and D. Lee, “Video linkage: group based copied video detection,” in *Proceedings of the International Conference on Content-Based Image and Video Retrieval*, 2008, pp. 397–406.

[81] N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[82] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[83] M. Chatzigiorgaki and A. N. Skodras, “Real-time keyframe extraction towards video content identification,” in *Proceedings of the 16th International Conference on Digital Signal Processing*, 2009, pp. 1–6.

[84] S. H. Han, K. J. Yoon, and I. S. Kweono, “A new technique for shot detection and key frames selection in histogram space,” in *Proceedings of the 12th Workshop on Image Processing and Image Understanding*, 2000.

[85] A. Guttman, “R-trees: a dynamic index structure for spatial searching,” in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1984, pp. 47–57.

[86] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion-compensated interframe coding for video conferencing,” in *Proceedings of the National Telesystems Conference*, New Orleans, LA., November 1981, pp. C9.6.1–9.6.5.

[87] Z.-N. Li and M. S. Drew, *Fundamentals of Multimedia*. Prentice-Hall, 2004.

[88] PBS. (2007, Dec 7) Nature, Ravens, Raven Courting Ritual, PBS. [Online]. Available: http://www.youtube.com/watch?v=os5jcMjiXKI&feature=related

[89] D. Liu, M.-L. Shyu, Q. Zhu, and S.-C. Chen, “Moving object detection under object occlusion situations in video sequences,” in *Proceedings of the IEEE International Conference on Semantic Computing (ICSC)*, 2011, pp. 271–278.

[90] D. Liu, M.-L. Shyu, and G. Zhao, “Spatial-temporal motion information integration for action detection and recognition in non-static background,” in *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2013)*, 2013, pp. 626–633.

[91] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[92] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, 1981.

[93] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[94] D. Liu and M.-L. Shyu, "Effective moving object detection and retrieval via integrating spatial-temporal multimedia information," in *Proceedings of the IEEE International Conference on Semantic Computing (ICSC)*, 2012, pp. 364–371.

[95] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal on Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[96] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[97] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the 17th IEEE International Conference on Pattern Recognition*, vol. 3, 2004, pp. 32–36.

[98] I. T. Jolliffe, *Principal component analysis*. Springer-Verlag New York, 1986, vol. 487.

[99] C. Chen and M.-L. Shyu, "Integration of semantics information and clustering in binary-class classification for handling imbalanced multimedia data," in *Information Reuse and Integration in Academia and Industry*. Springer, 2013, pp. 281–298.

[100] Q. Zhu, L. Lin, and M.-L. Shyu, "Correlation maximisation-based discretisation for supervised classification," *International Journal of Business Intelligence and Data Mining*, vol. 7, no. 1/2, pp. 40–59, August 2012.

[101] Q. Zhu, L. Lin, M.-L. Shyu, and S.-C. Chen, "Effective supervised discretization for classification based on correlation maximization," in *Proceedings of the IEEE Information Reuse and Integration (IRI)*, 2011, pp. 390–395.

[102] C. Chen and M.-L. Shyu, "Clustering-based binary-class classification for imbalanced data sets," in *Proceedings of the IEEE Information Reuse and Integration (IRI)*, 2011, pp. 384–389.

[103] C. Chen, L. Lin, and M.-L. Shyu, “Utilization of co-occurrence relationships between semantic concepts in re-ranking for information retrieval,” in *Proceedings of the IEEE International Conference on Semantic Computing (ICSC)*, 2011, pp. 53–60.

[104] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, p. 27, 2011.

[105] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65–72.

[106] J. Liu and M. Shah, “Learning human actions via information maximization,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[107] S.-F. Wong, T.-K. Kim, and R. Cipolla, “Learning motion categories using both semantic and structural information,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6.

[108] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, “Discriminatively trained deformable part models, release 5,” http://people.cs.uchicago.edu/ rbg/latent-release5/.

[109] L.-C. Chen, J.-W. Hsieh, D.-Y. Chen, and Y. Yan, “Vehicle make and model recognition using sparse representation and symmetrical surfs,” in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems Society Conference Management System (ITSC)*, October 2013, pp. 1143–1148.

[110] L.-C. Chen, J.-W. Hsieh, Y. Yan, and B.-Y. Wong, “Real-time vehicle make and model recognition from roads,” in *Proceedings of the Conference on Information Technology and Applications in Outlying Islands (ITAOI)*, May 2013, pp. 1033–1040.

[111] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, “Learning to detect a salient object,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353–367, February 2011.

[112] D. Liu and M.-L. Shyu, “Semantic motion concept retrieval in non-static background utilizing spatial-temporal visual information,” *International Journal of Semantic Computing*, vol. 7, no. 1, pp. 43–67, 2013.

[113] P. A. Viola and M. J. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2001, pp. 511–518.

[114] M.-L. Shyu, C. Chen, and S.-C. Chen, “Multi-class classification via subspace modeling,” *International Journal of Semantic Computing*, vol. 5, no. 01, pp. 55–78, 2011.

[115] M. Zhu, “Recall, precision and average precision,” *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2004.

[116] D. Liu and M.-L. Shyu, “Semantic retrieval for videos in non-static background using motion saliency and global features,” 2013, pp. 294–301.

[117] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid *et al.*, “Evaluation of local spatio-temporal features for action recognition,” in *Proceedings of British Machine Vision Conference (BMVC 2009)*, 2009.

[118] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[119] X. Cai, A. Huang, and S. Xu, “Fast empirical bayesian lasso for multiple quantitative trait locus mapping,” *BMC bioinformatics*, vol. 12, no. 1, p. 211, 2011.

[120] A. Huang, S. Xu, and X. Cai, “Empirical bayesian lasso-logistic regression for multiple binary trait locus mapping,” *BMC genetics*, vol. 14, no. 1, p. 5, 2013.

[121] Q. Zhu, L. Lin, M.-L. Shyu, and S.-C. Chen, “Feature selection using correlation and reliability based scoring metric for video semantic detection,” 2010, pp. 462–469.

[122] Q. Zhu, M.-L. Shyu, and S.-C. Chen, *Discriminative Learning Assisted Video Semantic Concept Classification*. CRC Press, 2012.

[123] D. Liu, M.-L. Shyu, C. Chen, and S.-C. Chen, “Within and between shot information utilisation in video key frame extraction,” *Journal of Information & Knowledge Management*, vol. 10, no. 3, pp. 247–259, 2011.

[124] C. Stauffer and W. E. L. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000. [Online]. Available: http://dx.doi.org/10.1109/34.868677

[125] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[126] C. Zhang, J. C. Platt, and P. A. Viola, “Multiple instance boosting for object detection,” in *Proceedings of Advances in Neural Information Processing Systems*, 2005, pp. 1417–1424.

[127] J.-W. Hsieh, K.-T. Chuang, Y. Yan, and L.-C. Chen, “Sparse representation for recognizing object-to-object actions under occlusions,” in *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*. New York, NY, USA: ACM, 2013, pp. 117–120.

[128] K.-T. Chuang, J.-W. Hsieh, and Y. Yan, “Modeling and recognizing action contexts in persons using sparse representation,” in *Advances in Intelligent Systems and Applications - Volume 2*, ser. Smart Innovation, Systems and Technologies, J.-S. Pan, C.-N. Yang, and C.-C. Lin, Eds. Springer Berlin Heidelberg, 2013, vol. 21, pp. 531–541.

[129] T. Meng and M.-L. Shyu, “Leveraging concept association network for multimedia rare concept mining and retrieval,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, July 2012, pp. 860–865.

[130] C. Chen, L. Lin, and M.-L. Shyu, “Re-ranking algorithm for multimedia retrieval via utilization of inclusive and exclusive relationships between semantic concepts,” *International Journal of Semantic Computing*, vol. 6, no. 02, pp. 135–154, 2012.

[131] T. Meng and M.-L. Shyu, “Model-driven collaboration and information integration for enhancing video semantic concept detection,” in *Proceedings of the IEEE Information Reuse and Integration (IRI)*, August 2012, pp. 144–151.

[132] C. Chen, M.-L. Shyu, and S.-C. Chen, “Data management support via spectrum perturbation-based subspace classification in collaborative environments,” in *Proceedings of the IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2011, pp. 67–76.

[133] T. Meng and M.-L. Shyu, “Concept-concept association information integration and multi-model collaboration for multimedia semantic concept detection,” *Information System Frontiers*, vol. 15, no. 2, pp. 1–13, April 2013.

[134] Q. Zhu, L. Lin, M.-L. Shyu, and D. Liu, “Utilizing context information to enhance content-based image classification,” *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 2, no. 3, pp. 34–51, 2011.

[135] C. Chen, Q. Zhu, L. Lin, and M.-L. Shyu, “Web media semantic concept retrieval via tag removal and model fusion,” *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, pp. 61:1–61:22, October 2013.

[136] D. Liu, S. Hua, Z. Ou, and J. Zhang, “Ir and visible-light face recognition using canonical correlation analysis,” *Journal of Computational Information Systems*, vol. 5, no. 1, pp. 291–297, 2009.

[137] J. Zhang, Z. Ou, and D. Liu, “Palmprint verification system based on mobile device,” *Computer Engineering*, vol. 36, no. 4, pp. 164–168, 2010.

[138] D. Liu, X. Zhou, and C. Wang, “Wavelet-based multispectral face recognition,” *Optoelectronics Letters*, vol. 4, no. 5, pp. 384–386, 2008.

[139] D. Liu, S. Hua, T. Su, Z. Ou, and J. Zhang, “Multi-spectral face fusion recognition based on fisher projection,” *Computer Engineering*, vol. 36, no. 8, pp. 180–182, 2010.

[140] Q. Zhu, Z. Li, H. Wang, Y. Yang, and M.-L. Shyu, “Multimodal sparse linear integration for content-based item recommendation,” to appear in Proceedings of the IEEE International Conference on Semantic Computing (ICSC), 2013.

[141] H.-Y. Ha, Y. Yang, F. C. Fleites, and S.-C. Chen, “Correlation-based feature analysis and multi-modality fusion framework for multimedia semantic retrieval,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2013, pp. 1–6.

[142] Q. Zhu, M.-L. Shyu, and H. Wang, “Videotopic: Content-based video recommendation using a topic model,” to appear in Proceedings of the IEEE International Conference on Semantic Computing (ICSC), 2013.

[143] H.-Y. Ha, F. C. Fleites, and S.-C. Chen, “Building multi-model collaboration in detecting multimedia semantic concepts,” to appear in Proceedings of the IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2013.

[144] Y. Yang, F. C. Fleites, H. Wang, and S.-C. Chen, “An automatic object retrieval framework for complex background,” to appear in Proceedings of the IEEE International Conference on Semantic Computing (ICSC), 2013.

[145] C. Chen, T. Meng, and L. Lin, “A web-based multimedia retrieval system with mca-based filtering and subspace-based learning algorithms,” *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 4, no. 2, pp. 13–45, 2013.

[146] Y. Yang, H.-Y. Ha, F. C. Fleites, and S.-C. Chen, “A multimedia semantic retrieval mobile system based on hidden coherent feature groups,” *IEEE MultiMedia*, 2013, in press.