12-2014

Identification, indexing, and retrieval of cardio-pulmonary resuscitation (CPR) video scenes of simulated medical crisis.

Surangkana Rawungyot
University of Louisville

IDENTIFICATION, INDEXING, AND RETRIEVAL OF CARDIO-PULMONARY RESUSCITATION (CPR) VIDEO SCENES OF SIMULATED MEDICAL CRISIS

By

Surangkana Rawungyot
B.S., Computer Science, Chiang Mai University, Thailand, 1998
M.S., Information Technology, King Mongkut′*s* Institute of Technology Ladkrabang, Thailand, 2002

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

December 2014

IDENTIFICATION, INDEXING, AND RETRIEVAL OF CARDIO-PULMONARY RESUSCITATION (CPR) VIDEO SCENES OF SIMULATED MEDICAL CRISIS

By

Surangkana Rawungyot
B.S., Computer Science, Chiang Mai University, Thailand, 1998
M.S., Information Technology, King Mongkut$'s$ Institute of Technology Ladkrabang, Thailand, 2002

A Dissertation Approved On

December 1, 2014

by the following Dissertation Committee:

Hichem Frigui, Ph.D., Dissertation Director

Aaron W. Calhoun, MD

Adrian P. Lauf, Ph.D.

Olfa Nasraoui, Ph.D.

Roman V. Yampolskiy, Ph.D.

C. Tim Hardin, Ph.D.

ACKNOWLEDGEMENTS

First of all, I would like to extend my gratitude to my advisor, Professor Hichem Frigui, for his guidance, support and motivation throughout the duration of this dissertation. He provided me with an excellent working environment for doing research. He helped me to learn and expand my knowledge from basic concepts in computer vision to advanced statistical modeling. Our numerous discussions have provided me with great motivations on my research, especially in the writing task. His encouragement made it possible to accomplish this work. All of his efforts will continue to be useful throughout my career. I am really honored to have studied under his guidance.

I would also like to express my gratitude to the members of my committee, Dr.Aaron Calhoun, MD , Dr.Adrian Lauf, Dr.Olfa Nasraoui, Dr.Roman Yampolskiy and Dr.C. Tim Hardin for all of their guidance, discussion, ideas and feedback that have been absolutely invaluable.

I would like to thank Dr.Mehmed Kantardzic who had guided me through each of the great steps in the program study during my first year at the Computer Engineering and Computer Science department. I also thank all the faculty, staff members and lab technicians of the Computer Engineering and Computer Science Department, whose services have turn helped to my research to success.

I would like to thank my friends in the Multimedia Research Laboratory, friends in the Computer Engineering and Computer Science Department, and friends in the University of Louisville for their support and friendship.

I would like to thank the Royal Thai scholarship, which has provided me the financial support that is necessary for studying in the United States. Without their care and funding,

it was impossible to accomplish this work.

Finally, my greatest gratitude goes to my family, my mom, and my dad for their endless support and encouragement over the years. In particular, I would like to thank my husband and my lovely son for being always by my side. Without their love and support, I would not be successful.

ABSTRACT

IDENTIFICATION, INDEXING, AND RETRIEVAL OF CARDIO-PULMONARY RESUSCITATION (CPR) VIDEO SCENES OF SIMULATED MEDICAL CRISIS

Surangkana Rawungyot

December 1, 2014

Medical simulations, where uncommon clinical situations can be replicated, have proved to provide a more comprehensive training. Simulations involve the use of patient simulators, which are lifelike mannequins. After each session, the physician must manually review and annotate the recordings and then debrief the trainees. This process can be tedious and retrieval of specific video segments should be automated.

In this dissertation, we propose a machine learning based approach to detect and classify scenes that involve rhythmic activities such as Cardio-Pulmonary Resuscitation (CPR) from training video sessions simulating medical crises. This applications requires different preprocessing techniques from other video applications. In particular, most processing steps require the integration of multiple features such as motion, color and spatial and temporal constrains. The first step of our approach consists of segmenting the video into shots. This is achieved by extracting color and motion information from each frame and identifying locations where consecutive frames have different features. We propose two different methods to identify shot boundaries. The first one is based on simple thresholding while the second one uses unsupervised learning techniques.

The second step of our approach consists of selecting one key frame from each shot and segmenting it into homogeneous regions. Then few regions of interest are identified for

further processing. These regions are selected based on the type of motion of their pixels and their likelihood to be skin-like regions. The regions of interest are tracked and a sequence of observations that encode their motion throughout the shot is extracted. The next step of our approach uses an HMM classifier to discriminate between regions that involve CPR actions and other regions. We experiment with both continuous and discrete HMM.

Finally, to improve the accuracy of our system, we also detect faces in each key frame, track them throughout the shot, and fuse their HMM confidence with the region's confidence.

To allow the user to view and analyze the video training session much more efficiently, we have also developed a graphical user interface (GUI) for CPR video scene retrieval and analysis with several desirable features.

To validate our proposed approach to detect CPR scenes, we use one video simulation session recorded by the SPARC group to train the HMM classifiers and learn the system's parameters. Then, we analyze the proposed system on other video recordings. We show that our approach can identify most CPR scenes with few false alarms.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Digital video generation and distribution is growing exponentially in business, education, sports, entertainment, advertising, social media, and security. Manual annotation of video for the purpose of review and retrieval is very time-consuming. The reason is that tremendous manpower is needed to label the video. Therefore, a content-based approach is needed. Consequently, there have been various studies and applications in video indexing [1][2], analysis [3], retrieval [4, 5, 6], and classification [7][8]. Although several content-based video retrieval prototypes have been proposed [9, 10, 11], the problem remains largely unsolved especially for generic video. As a result, researchers have focused on indexing and retrieval of video for specific applications. For example, Chen et al. [12] developed a SoccerQ for retrieving soccer events from sports videos by using temporal relationships. The content provider needs to extract and present the highlights for the viewer. SoccerQ is developed to support both basic queries and relative temporal queries that can be automatically analyzed. Video shots relevant to the query are then retrieved and displayed via a user-friendly interface. Several other methods have been proposed to detect events in sports videos such as football games [13], basketball games [14], and baseball games [15]. These methods take into account the different rules of the game, the players' actions, and the strategy of the game.

Another application that can benefit from content-based video retrieval is security. In this application, videos from surveillance cameras are used to detect events requiring attention as they happen and to take actions in real time when an unusual human behavior is detected. Detecting human behavior may require only a simple description of each person in the scene, or a count of the number of people present. It may also involve

more complex description of an individual such as overall body motion, limb movement, or hand gesture. For instance, Raiyn [16] introduced a cognitive video surveillance system to retrieve video shots where a suspected person is detected using motion features. Another example in security application is to detect dangerous situations in underground railway stations. Velastin et al. [17] exploited motion-based methods used in a surveillance system with multiple cameras. The system kept recorded video as visual evidence to help with a *posteriori* investigation of unusual events or criminal actions.

Video usage in traffic systems is another application of content-based video analysis. Typically, motion analysis is used to detect anomalous behaviors such as a car driving in the wrong lane or turning left from the right-hand lane that can be detected from a surveillance camera mounted above an intersection. For instance, in [18], Brand and Kettnaker learned patterns of behavior by training a HMM. They applied their system in an office environment to detect unusual behavior such as falling asleep or driving in the wrong lane. Using a similar approach, the Bobick system[18] was proposed to interpret movements in a parking lot. The system can detect events such as a car entering or leaving, a person entering or leaving, a person being picked up or dropped off, or losing or finding a track.

Another application in content-based video analysis involves classification. The objective in this application is to classify video segments into categories, typically with a meaningful label such as sport, movie, or crowd. For instance, Huang et al. [19] proposed a method to categorize sport video into basketball and football games based on color, motions, and audio features from other video sources. Similarly, in [20], Li et al. presented an automatic sports genre categorization based on the bag of visual-words model.

In this dissertation, we focus on analyzing medical simulation training video data. We use data recorded by the Simulation for Pediatric Assessment, Resuscitation and Communication (SPARC) group within the Department of Pediatrics at the University of Louisville. SPARC was developed to teach pediatrics faculty, fellows, and residents how to respond to medical crises. Its objective is to enhance the care of infants and children by using simulation-based educational methodologies to improve patient safety and strengthen

interdisciplinary and clinician-patient interactions. The simulation sessions involve the use of patient simulators which are lifelike mannequins that have respiration and heartbeat, and respond to treatment with virtual drugs. Simulation sessions involve 4 to 9 people and last approximately 30 minutes to one hour. They are scheduled approximately twice per week and are recorded as video data. After each session, the physician/instructor must manually review and annotate the recording and then debrief the trainees on the session. Video-assisted debriefing allows participants to deconstruct and reflect on their experiences, teaching them how to approach such tasks more effectively in the future.

The physician responsible for the simulation sessions has recorded over 100 sessions, and is now realizing that: (1) the manual process of review and annotation is labor intensive; (2) retrieval of specific video segments is not trivial; and (3) there is a wealth of information waiting to be mined form these recordings. Providing the physician with automated tools to segment, semantically index and retrieve specific scenes from a large database of training sessions will enable him/her to: (1) immediately review important sections of the training with the team; (2) allow more efficient debriefing with the team of trainees; and (3) identify similar circumstances in previously recorded sessions. The longer term payback is the potential discovery of similar critical elements in a training session that results in either positive or negative outcomes and thus enhances the effectiveness of the training.

We focus on detecting and classifying scenes that involve rhythmic activities such as Cardio-Pulmonary Resuscitation (CPR) from training video sessions simulating medical crises. Our approach consists of two main steps: The first one segments the video into shots, selects one keyframe for each shot, and identifies regions with skin-like colors in each keyframe. Each skin region is then represented by a sequence of observations that encode its motion in the different frames within the shot boundaries. The second step consists of using motion-based features in a discrete and a continuous HMM classifier to identify the skin-like regions that involve CPR activities. Our framework is illustrated in figure 1.1.

The main contribution of this dissertation could be outlined as follows:

1. We developed, implemented, and tested an algorithm to segment video into shots.

Our approach is based on unsupervised learning and uses features based on color, motion, and temporal information.

2. We developed, implemented, and tested an image segmentation algorithm to partition the shot key frame into homogenous regions. Our approach uses color, optical flow, and spatial information as features and the NCut algorithm to perform clustering.

3. We developed, trained, and a tested classifier to identify regions with skin-like colors.

4. We developed, trained, and a tested discrete and continuous HMM classifier to identify CPR scenes. Our approach is based on tracking regions of interest throughout the shot and encoding their motion information.

5. We implemented and tested a face selection and tracking algorithm. We showed that this information could improve the accuracy of the CPR classifier by analyzing the motion of the face of the subject performing CPR in addition to the motion of his/her hands.

6. We integrated all of the above algorithms into a system that analyzes a video simulating medical crises and identifies CPR scenes in a completely unsupervised way.

7. We developed a GUI prototype for CPR scene retrieval and analysis. The GUI has revealed desirable features that allow the user to view and analyze the video training sessions in an efficient way.

The remainder of this thesis is organized as follows. In chapter II, we provide an overview of related work on image segmentation, skin detection, motion features, and scene detection techniques. In chapter III, we propose the Cardio-Pulmonary Resuscitation (CPR) scene identification system. In chapter IV, we present the experimental results and analysis and describe our graphical user interface (GUI). Finally, in chapter V, we conclude and discuss potential future work.

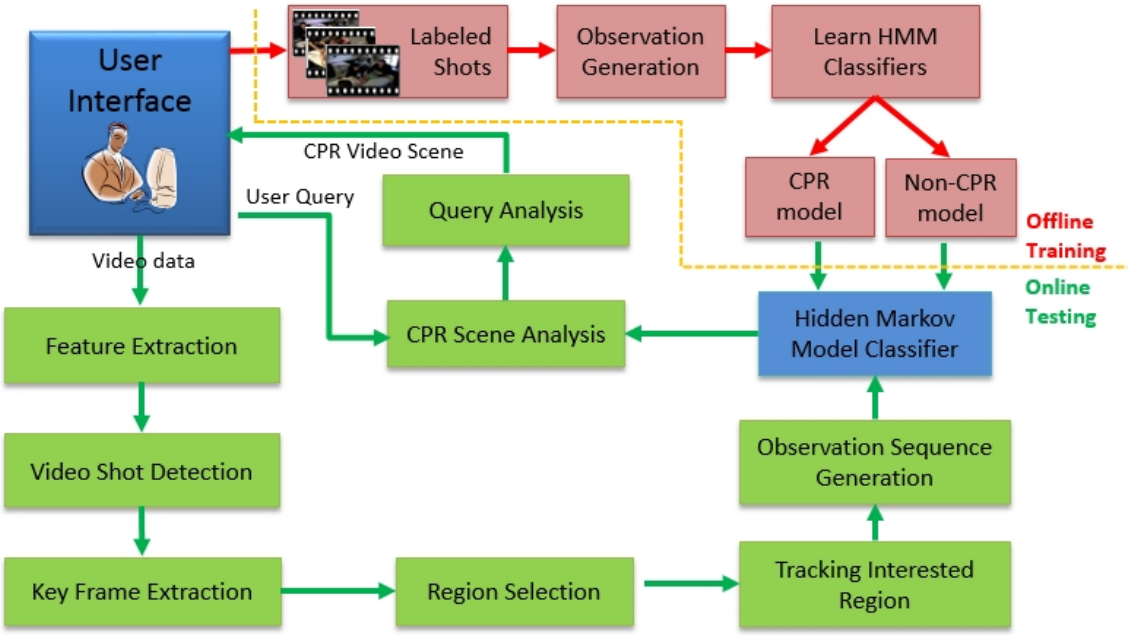


Figure 1.1: Overview of the proposed system to identify and retrieve Cardio-Pulmonary Resuscitation (CPR) scenes from training video sessions simulating medical crisis

CHAPTER 2

RELATED WORK

Understanding, indexing, and retrieving medical video data is an important and necessary tool to take advantage of the large image and video databases. A good introduction the current state of the art in medical image and video analysis can be found in [21][22]. Research on automatic medical video processing has focused on recordings of specific medical procedures such as colonoscopies or echocardiograms, and on the analysis of the training video data. For instance, Gokturk et al. [23] proposed a computer-aided diagnosis system for colonoscopy videos based on a 3D shape pattern processing method and a support vector machine classifier. Cao et al. [24][25] developed a spatial-temporal analysis technique to automatically identify video events in a colonoscopy video that correspond to a diagnostic or therapeutic operation. In [25][26], image segmentation-based spatial analysis techniques are utilized to locate the image with regions of interest in a colonoscopy video. A scene is recognized as a specific configuration of the detectable objects. Li et al. [27] suggested algorithms and tools useful to summarize the echocardiogram videos using temporal video segmentation techniques.

Medical video, used to train the next generation of physicians, is another category of video that needs to be analyzed, indexed, and retrieved by content. In [28], Han et al. proposed an approach for analyzing and providing objective statistics of the trainees' behavior. The main idea is based on tracking the motion and location of the subjects. These parameters are used to judge and compare the behavior of the trainees. To simplify identification and tracking, the trainees wore medical caps with distinctive colors.

In [29], Zhu et al. introduced a framework to mine medical video content and events. First, the video is segmented and key-frames are selected. Second, the scenes are clustered

and grouped to organize the video shot into a hierarchical structure. Then, audio and video processing techniques are integrated to mine event information, such as dialog, presentation, and clinical operation, among the detected scenes. Finally, visualization techniques are used to construct a scalable video skimming tool. Other methods used to segment and analyze medical training video can be found in [30][31].

In general, most of the above systems and other methods used for video indexing and retrieval include the following main steps:

2.1 Hierarchical Representation of Video

The first step in video analysis consists of organizing the video into an ascending hierarchy of frames, shots, and scenes [3, 8, 32] as illustrated in figure 2.1. A frame is a still image in the video stream. Whereas, a shot is a set of consecutive frames from a single camera, and can involve camera motion like panning (horizontal rotation), tilting (vertical rotation), zooming (focal length change), tracking (horizontal transverse movement), booming (vertical transverse movement) and dolling (horizontal lateral movement) as well as movement of objects within or out of the frame of view [2]. Thus, multiple motions can appear within a single shot. Shots usually transition from one to the next using what is called a hard cut (discontinuous), where the last frame of one shot is followed immediately by the first frame of another shot. Shot transitions can also be gradual (continuous) such as a fade in/out, dissolve, wipes, or the more elaborate digitally-based shot transitions [6]. Shots are considered to be the fundamental units to organize the content of video sequences and the basic elements for semantic annotation and retrieval tasks. A scene is the next level in video hierarchy. It is defined as a part of a story that represents a sequence of an action or an observation of an object [7]. Scene segmentation is also known as story unit segmentation. Scenes have normally higher level semantics than shots. They are identified or segmented out by grouping successive shots with similar content into a meaningful semantic unit. Most scene segmentation approaches are based on high-level information that can be extracted from text, images, or the audio track in the video.

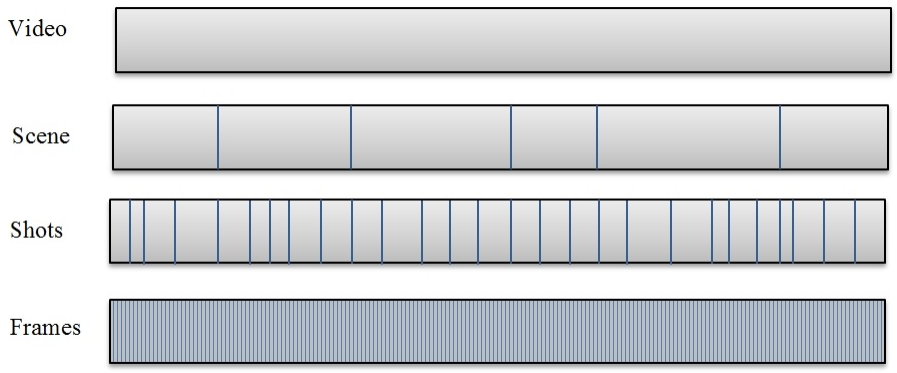


Figure 2.1: Hierarchical video representation

The organization of a video in a hierarchical structure involves 3 main steps: shot boundary detection, scene segmentation, and key frame extraction. These steps are highlighted in the following sections.

2.1.1 Shot Boundary Detection

Shot boundary detection, also known as temporal video segmentation, is usually performed by identifying the transition between adjacent shots. The general approach to detect shot boundaries involves 3 main steps. First, visual features are extracted from each frame. Then, the similarity between adjacent frames, using the extracted features is computed. Finally, shot boundaries are identified between frames with low similarity [1].

The main feature used to measure the similarity between frames involves color histograms [33, 34, 35]. Another feature that proved to be useful in detecting shot boundaries is edge change [36][37]. Other features, that are computationally more expensive, are based on motion [38][39] and on correlation in the frequency domain [40][41]. Combinations of the above features have also been investigated [35, 40, 42]. Many researchers have also exploited information stored in the compressed video files and achieved good accuracy at much lower computation cost [43][44].

Shot boundary detection methods can be based on either thresholding or statistical learning. In the threshold-based techniques, the pair-wise similarities between adjacent

frames are usually compared to a predefined threshold to detect shot boundaries[45]. In [40], Gargi et al. presented an automatic thresholding framework. In statistical learning-based approach, Yuan et al.[46] combine the threshold-based method with an SVM-based classifier. First, candidate boundaries are selected using the threshold-based method. Then, a Support Vector Machine(SVM) classifier is used to verity the boundaries. Other supervised learning algorithms, such as Adaptive Boosting(AdaBoost) algorithm [42], K-nearest-neighbor classifier[47], and Hidden Markov Model(HMM)[48] with visual features, have also been used for shot boundary detection.

2.1.2 Key Frame Extraction

After identifying the shot boundaries, one or more representative frames, called key frames, needs to be extracted for each group of frames or video shot. The visual contents of these key frames are then used to represent salient information of the video shots for the purpose of indexing and retrieval, video summarization, and browsing [49]. Some simple methods pick one or more key frames from every shot at a predefined temporal location such as the first, middle, or last frame. For instance, Divakaran et al.[38] extract the key frame by dividing the shot into segments with equal cumulative motion activity using the MPEG-7 motion activity descriptor and a fidelity measure. The frame located at the halfway point of each segment is then selected as a key frame.

Other key frame extraction methods are based on low level features such as color, shape, edge, optical flow, motion vector, and MPEG discrete cosine coefficients [34]. In [36], Zabih et al. proposed a key frame extraction approach based on edge change ratio. Sze et al. [50] proposed the temporally maximum occurrence frame (TMOF). TMOF is constructed by considering the probability of occurrence of pixel values at each pixel position for all frames within a video shot.

2.1.3 Scene Segmentation

Some short videos can consist of a large number of shots while other longer video can include few shots. Thus, shots may not be useful in video indexing and retrieval. Moreover, it is less likely that users are looking for a single shot. Instead, they may be searching for semantically meaningful scenes consisting of a number of shots. Video segmentation into scenes (or story units) is a harder task that requires the definition of a semantically meaningful scene.

Video segmentation can be achieved using a variety of features and techniques. For instance, in [51], Goela et al. presented a genre-independent method to detect scene boundaries in broadcast video. In their method, scene segmentation is achieved by defining a class of scene change and another class of non-scene change. An SVM is used to classify the scene boundaries. Hand-labeled video scene boundaries, extracted from a variety of broadcast genres, are used to generate positive and negative training samples for the SVM.

Another commonly used approach for scene segmentation is based on graph partitioning. Graph-based algorithms cluster shots based on similarity and arrange them in a graph representation. Nodes represent shots or clusters of shots and edges indicate similarity or temporal closeness between the connected nodes. By using graph segmentation algorithms, the constructed graphs are divided into subgraphs, each representing a shot. For instance, Rasheed and Shah [52] construct a shot similarity graph for a video and partition the graph using normalized cuts. The subgraphs represent individual shots in the video. Another approach, based on scene transition graph, is introduced by Ngo et al. [53]. Benini et al. [54] also use the scene transition graph approach for identifying scenes, where the shot similarity is calculated based on codebooks of visual codewords generated by a vector quantization process.

Another technique for scene segmentation is based on rules. For example, Lin et al. [55] proposed a scene segmentation approach based on splitting and merging forces. For each shot, a dominant color histogram (DCH) and a spatial structure histogram(SSH) are calculated. The splitting force indicates the difference in the previous shots by comparing

each shot with its three ancestors and successors. The merge force indicates the coherence of one shot with its three successor shots. An ideal scene boundary is detected when the splitting force reaches its maximum and the merging force reaches its minimum. In practice, this is not always the case, thus two additional rules have been proposed for scene boundary detection: (1) if the splitting force reaches its maximum and the merging force is under a pre-defined threshold; or (2) if the merging force reaches its minimum the splitting force is above a pre-defined threshold.

Some researchers have used hybrid solutions that combine the individual strength of visual-based, audio-based and text-based methods. The challenge is not only how to combine the different features or different methods but also the choice of an effective similarity measure for each modality. For example, Gatica-Perez et al. [56] presented a scene segmentation for home videos which uses video shots. They exploited the characteristics of home video and combined color and edge density, edge direction, and color ratios into 4-D histograms. Using these features, a hierarchical clustering algorithm was used for shot clustering. Other methods used for scene detection can be found in [32][53].

2.2 Feature Extraction

Video indexing and retrieval rely on features to describe their content. Here, we focus on visual features suitable for video indexing and retrieval. These are mainly color-based, shape-based, motion-based, and object-based features extracted from one key-frame or from multiple frames within a short sequence.

Color-based features include color histograms [33][35], color correlograms [57], mixture of Gaussian models [58], and others. The extraction of color based features requires the specification of a color space such as RGB, HSV, YC_bC_r, YUV and HVC. The choice of the color space depends on the objective of each application. Color features can be extracted from the entire image or from image blocks or regions. Color-based features are the most efficient descriptors for video indexing and retrieval. In particular, color histogram and color moments are simple but efficient. The color histogram approach is based on identifying the

color of every pixel in each frame and gathering these into a histogram with a fixed number of bins. Many video retrieval and indexing applications have used color histograms [33][35]. For instance, Geetha and Palanivel [8] proposed a Block Intensity Comparison Code(BICC) as a feature to detect the shot change in a video stream.

Shape-based features can be extracted from object contours or regions in a video frame. A common approach is to detect edges in the image and then describe the distribution of the edges using histograms. Hauptmann et al. [37] adopted the edge histogram descriptor *(EHD)* to capture the spatial distribution of edges for the video search task in *TRECVid-2005*. The *EHD* is computed by counting the number of pixels that contribute to edges according to their quantized directions. Similarly, Lienhart [35] extracted edge features using the Canny edge detector. The detected edges are then used to detect shot boundaries by computing the edge change ratio in adjacent frames.

Object-based features include dominant colors, texture, and trajectory [7]. These features are derived from detected objects. They can be used to retrieve videos likely to contain similar objects. When used, they tend to focus on identifying specific types of objects, such as faces, pedestrians, and cars. Faces are useful objects in many video retrieval systems such as in human behavior analysis [59]. For instance, Sivic et al.[59] presented a person retrieval system that is able to retrieve a ranked list of shots containing a particular person, given a query face in a shot.

Motion within a video is primarily of two types: movement on the part of the objects in a video sequence or movement due to camera motion. Motion-based features consist mainly of optical flow [60] or MPEG motion vectors [61]. Optical flow is an estimate of motion in a sequence of images calculated from the velocities of pixel brightness patterns. This could be due to object motion or camera motion. There are many ways to measure the optical flow [60]. The most commonly used one is the Horn Schunck algorithm [62].

Another approach detects the total motion in a shot by comparing the histograms of blocks of consecutive frames [63]. In order to detect object motion, they first calculate optical flow using Horn Schunck algorithm [62]. Motion due to camera movement would

result in all blocks having motion. Thus, camera motion can be subtracted, leaving only the motion of objects. These objects are identified by segmenting pixels with parallel motion.

2.3 Video Classification

The goal in video classification is to learn rules or knowledge from the extracted features and then classify video shots into predefined categories[8]. In general, categories are associated with semantically meaningful events. Some classification methods assign meaningful labels to the entire video such as sports video or scary movies [7]. Other methods have focused on classifying segmented video for the purpose of identifying violent scenes in a movie or distinguishing between different news segments within the entire video [64].

Several learning algorithms have been used for video classification. The most common ones are based on Gaussian Mixture Model *(GMM)* [58] and Hidden Markov Model *(HMM)* [65]. In *GMM*, an unknown probability distribution function $p(x)$ can be represented by K Gaussian distribution functions such that

$$p\left(x\right) = \sum_{i=1}^{K} \pi_i N\left(x \mid \mu_i, C_i\right). \tag{2.1}$$

In (2.1), $N(x \mid \mu_i, C_i)$ is the i^{th} Gaussian distribution with mean μ_i and covariance C_i and π_i is the prior probability of component i. *GMMs* have been used for constructing complex probability distributions as well as clustering. For example, Roach et al.[66] performed video classification with a *GMM* into sports, cartoons, and news by using object and camera motion. Camera motion is detected using an optical flow feature, and object motion is detected by comparing pixels between consecutive frames. Both types of motion are represented by a second order signal. A Discrete Cosine Transform *(DCT)* is applied to these signals to reduce the dimensionality. Similarly, in [67], *GMM* was used to classify video as either cartoon or non-cartoon. In this application, the features used for classification are based on motion of foreground objects, which are detected using pixel based frame differentiation.

Hidden Markov Model[65] is a common algorithm for classsifying sequential data. Since video is a collection of features extracted from a sequence of ordered frames, many

researchers have used *HMM* to classify video. In[68], Lu et al. used *HMM* to classify television programs into four categories : news reports, commercials, live basketball games, and live football games. First, for each frame, features are extracted from illumination-invariant color histograms. Then, a hierarchical clustering algorithm is used to segment the video into scenes. Finally, *HMM* is used to classify each scene. Similarly, in [69], the authors used *HMM* to segment soccer video. In this application, dominant color ratio and motion intensity are extracted from the compressed domain of MPEG video. A manually labeled training set is used to train the models. As a result, soccer videos can be classified into play scenes and break scenes.

CHAPTER 3

EXTRACTION, IDENTIFICATION, AND ANALYSIS OF CPR SCENES FROM VIDEO SIMULATING MEDICAL CRISIS

In this chapter, we describe our proposed approach to detect and classify video scenes that involve rhythmic activities. The objective is to provide answers to queries that are of interest to the physician supervising the training sessions such as: *"Show me all the scenes that have a CPR action from a given video simulation training."* First, we describe the feature extraction process. These features are needed to quantify the similarity between adjacent frames. Then, we present our approach to segment the video into a set of shots. After shot detection, a key frame that reflects the main content of each shot is extracted. Finally, we illustrate how to select the regions of interest for detecting and tracking the motion throughout the shot. The motion observations will be used to build the binary classification model. An overview of the proposed system is illustrated in figure 3.1. The steps of our proposed system are described in the following sections.

3.1 Feature Extraction for Identifying Shot Boundaries

The general idea of video shot boundary detection is to look for discontinuity. This can be achieved in different ways. In this dissertation, we use a feature-based approach and investigate three features: global color histogram, average motion magnitude, and motion orientation histogram. These features are described in the following subsection.

3.1.1 Global Color Histogram

Color histogram is an efficient and robust way to describe and encode the color information of a video frame. This is because histograms are insensitive to image rotation

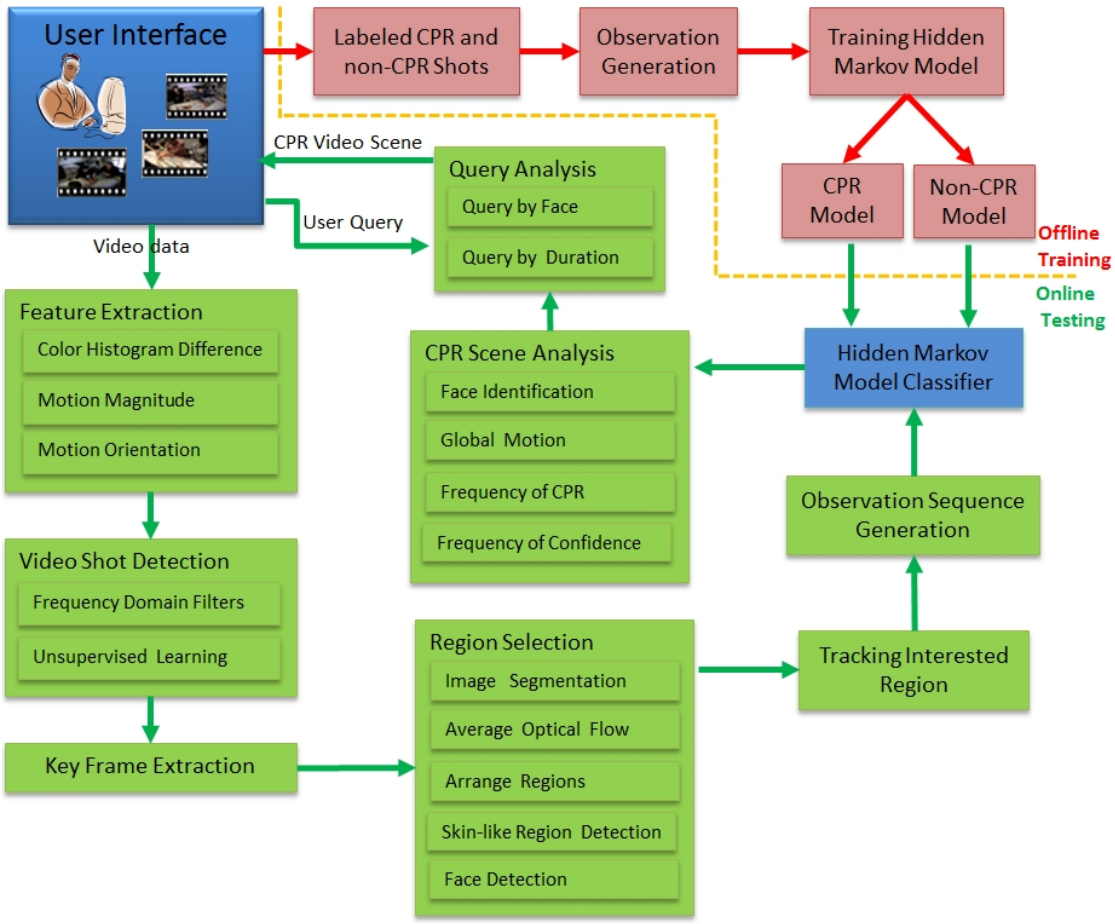


Figure 3.1: Overview of the proposed system to extract and identify CPR scenes

and change slowly with a change in angle and scaling effect. Moreover, two images with similar background and similar object content will have similar histograms. In our proposed method, we quantize the RGB color space into M bins. Then, for each frame, we compute its M-bin histogram. Next, we compute the dissimilarity between histograms of consecutive frames. The dissimilarity between the histogram of frame t (hc_t) and the histogram of frame $t+1$ (hc_{t+1}) is defined as :

$$CHDiff(t) = \sum_{i=1}^{M} |hc(t+1,i) - hc(t,i)|. \tag{3.1}$$

In (3.1), t refers to the frame number and i refers to the bin number. Typically, a large $CHistDiff$ at frame t indicates the possibility of a shot boundary at that location. Figure

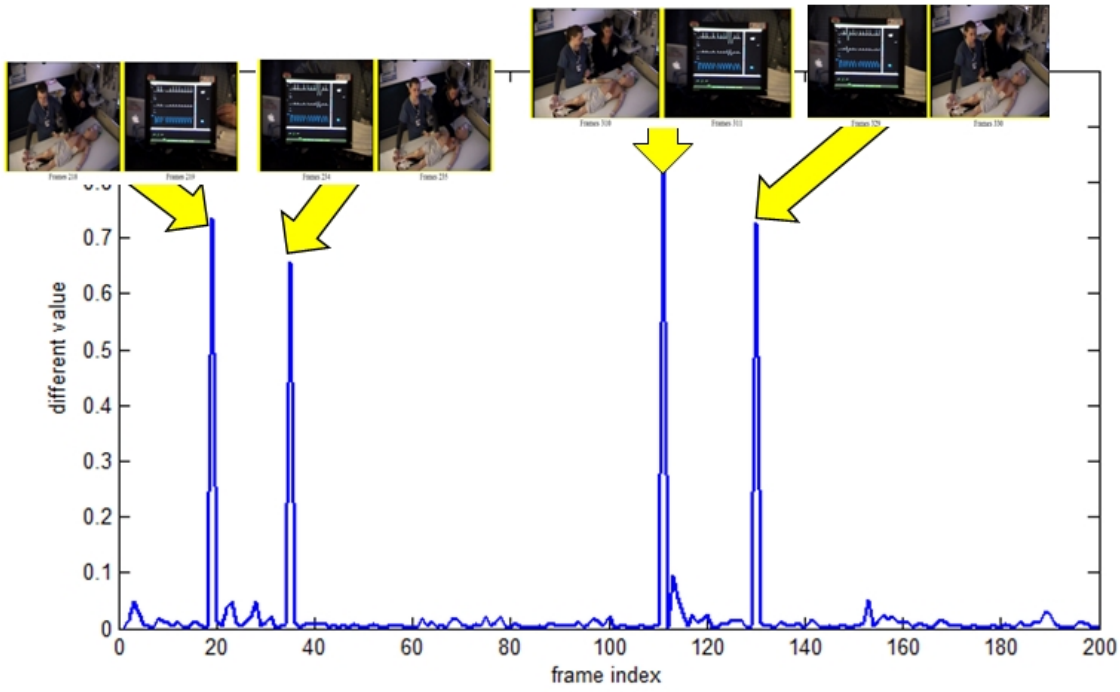


Figure 3.2: Sample color histogram difference of a video sequence with 4 major shot changes

3.2 shows an example of a color histogram difference with 4 strong candidates for shot boundaries.

3.1.2 Global Magnitude of Motion Vector

Color histograms detect shot boundaries that involve discontinuity in the color of adjacent frames. However, they do not consider movement information including camera motion and object movement within the frame. In addition, a color histogram encodes only the image's overall color composition. Thus, images with different color appearances can have similar global color histograms. To overcome this limitation, we propose using motion information as an additional feature to identify shot boundaries. This information can be extracted from the compressed domain and is directly accessible from the MPEG stream as the motion vector for each block [70]. Thus, this feature is intuitive and does not require significant additional computation.

Typically, in an MPEG video, frames are divided into blocks of size 16x16 pixels,

and motion vectors are estimated as the movement from the center of one macroblock in the current frame to the closest matching macroblock in the reference frame. The global motion in frame t (V_t), is defined as the global average of the motion vectors of all macroblocks in the frame.

Shot boundaries can be identified at frames with significantly different motion. We quantify the difference in motion between consecutive frames by defining the absolute difference of average motion magnitude as

$$MDiff(t) = |(\overline{V}_{t+1}) - (\overline{V}_t)| \tag{3.2}$$

In (3.2), $\overline{V}_t$ is the average magnitude of the motion vectors for all macroblocks in frame t and is computed using

$$\overline{V}_t = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=i}^{H} R_{xy}(i,j) \tag{3.3}$$

where W and H denote the number of macroblocks per row and column in each frame respectively. In (3.3), $R_{xy}(i,j) = \sqrt{dx^2_{(i,j)} + dy^2_{(i,j)}}$ is the magnitude of the motion vector of the macroblock at location (i,j) and $dx_{(i,j)}$ and $dy_{(i,j)}$ denote the motion vectors in the x and y directions. A sample of the motion magnitude difference for a video with 200 frames is illustrated in figure 3.3. The yellow arrow illustrates the frame location where two objects in the red circle and the green circle are moving at different speeds.

3.1.3 Motion Orientation Histogram

Another feature that could be used to identify shot boundaries is based on the difference in orientation of the motion of the different macroblocks. The motion orientation histogram is an efficient and effective way to represent the motion orientation information. The distribution of the orientation movement in each frame is estimated from the motion vector of its macroblocks. First, the orientation of the motion vector of each macroblock (i,j) is computed using

$$\theta(i,j) = arctan(dy(i,j)/dx(i,j)) \tag{3.4}$$

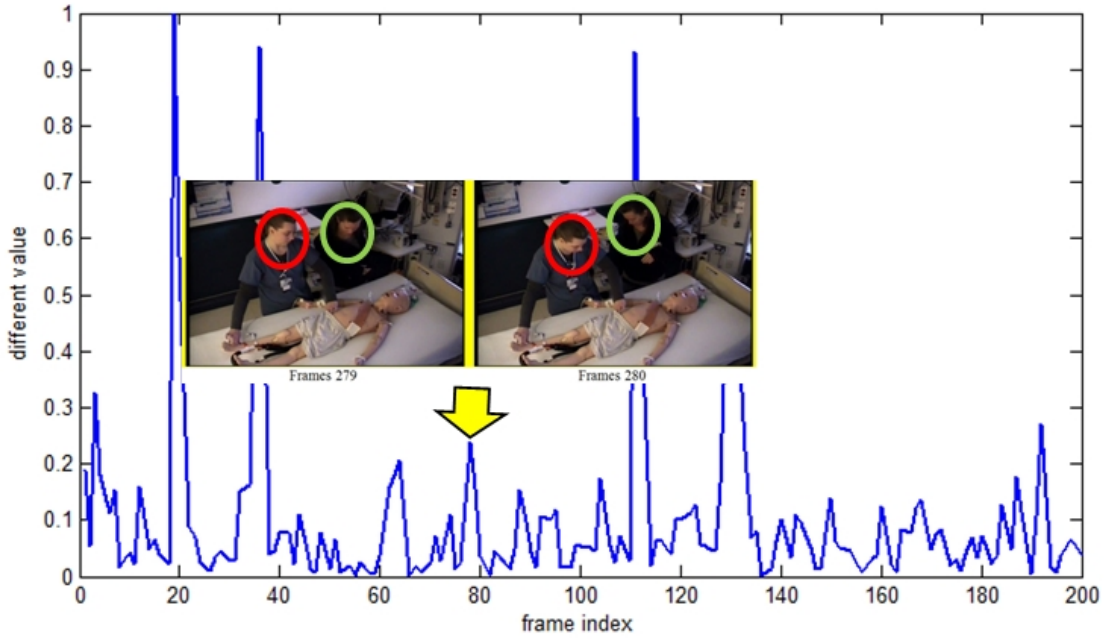


Figure 3.3: Sample motion magnitude histogram difference of a sequence of 200 frames

where $(dx(i,j), dy(i,j))$ is the motion vector associated with macroblock in (i,j). Next, all $\theta(i,j)$ are quantized into M discrete orientations. Let $h\theta(t)$ denote the motion orientation histogram of frame t and let $h\theta(t,i)$ denote the i^{th} bin of $h\theta(t)$. We define the dissimilarity between the motion orientation histogram of frames t $(h\theta_t)$ and the motion orientation histogram of frame $t+1$ $(h\theta_{t+1})$ as

$$OHDiff(t) = \sum_{i=1}^{M} |h\theta(t+1,i) - h\theta(t,i)|. \tag{3.5}$$

Figure 3.4 illustrates the quantization of the motion orientation into 13 bins. Each bin represents the macroblocks that move in a particular direction. For example, bin 1 is used to represent macroblocks with a motion vector angle between 75 and 105 degrees. Bin 0 represents macroblocks with no significant motion. Finally, we compute a motion orientation histogram with M bins where bin i represents the number of macroblocks that have the discrete orientation are associated with these bins.

The dissimilarity between the motion orientation histograms of two consecutive frames at time t and time $t+1$ is illustrated in figure 3.5. Figure 3.5(c) shows the dif-

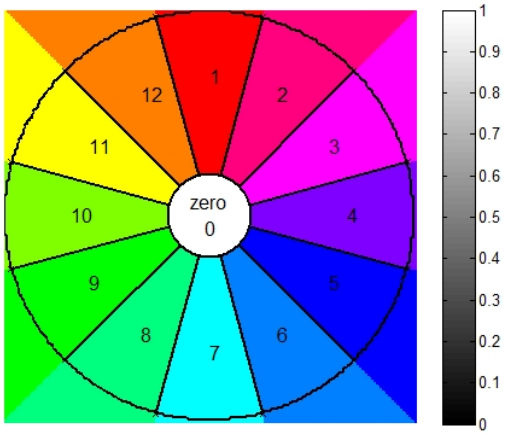
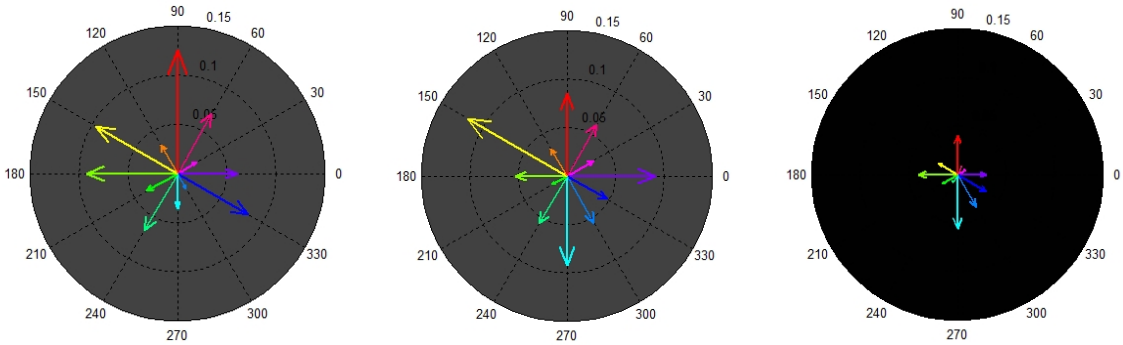


Figure 3.4: Quantization of the motion orientation histogram into $M = 13$ bins.

(a) Motion vectors of macro blocks in frame t

(b) Motion vectors of macro blocks in frame $t+1$

(c) Difference between the motion vectors in frame t and $t+1$



(d) Histogram of motion orientations in (a)

(e) Histogram of motion orientations in (b)

(f) Difference between the 2 motion orientation histograms

Figure 3.5: Illustration of the orientation histogram difference between 2 consecutive frames.

ference of the motion vector between figure 3.5(a) and figure 3.5(b). In figure 3.5(a) and figure 3.5(b) we show the motion of the macroblocks in frames t and $t+1$. The motion orientation histograms of these frames are shown in figure figure 3.5(d) and (e). Figure 3.5(f) shows the dissimilarity orientation histogram between the 2 frames.

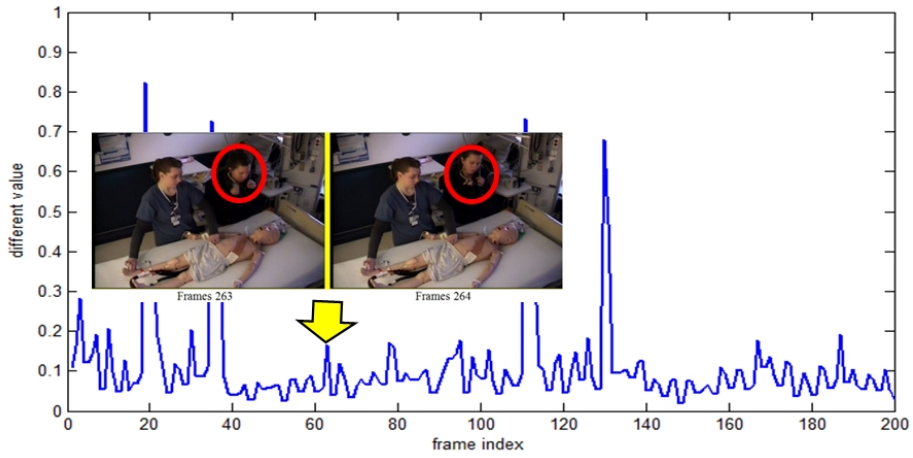


Figure 3.6: Orientation histogram difference of motion vectors for a sequence of 200 frames.

Figure 3.6 plots the orientation histogram difference of the motion vector in a sequence of 200 frames. The yellow arrow indicates one local peak point that is caused by the different motion direction of the objects in the red circle.

3.2 Shot Boundary Detection

The standard approach to shot detection consists of computing the difference between two successive frames and identifying locations with large differences. Identifying the shots boundaries and partitioning the video into shots is the first step of our proposed system. The goal is to have shots that do not combine CPR and non-CPR actions. Thus, we use strict thresholds and partition the video into a large number of small shots. We propose and compare two different methods.

3.2.1 Shot Detection based on Frequency Domain Filtering

Video segmentation could be achieved by detecting peaks in the sequence of the difference between histograms of consecutive frames. In this approach, we propose to use discrete signal processing techniques to identify peaks. This approach, called Shot Detection based on frequency domain(SD_FD), consists of three main modules: feature representation; signal processing and analysis; and estimation of the shot boundaries.

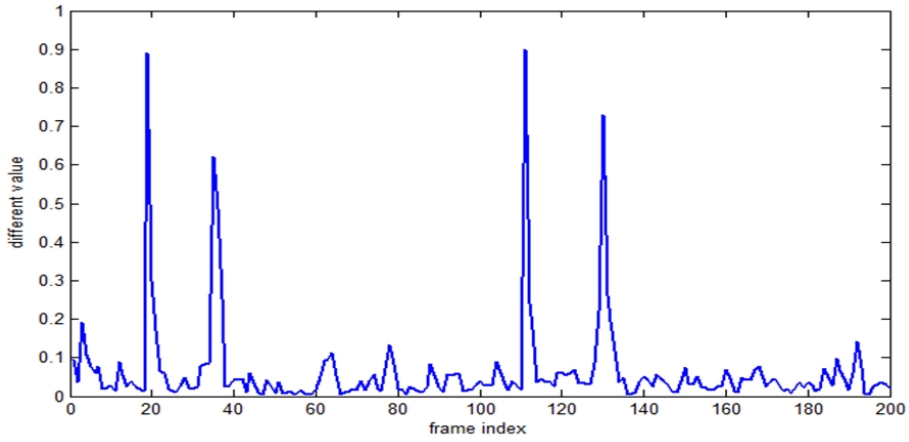


Figure 3.7: Combined color histogram difference, magnitude histogram difference, and orientation histogram difference for a sequence of 200 frames.

For feature representation, the SD_FD approach extracts and combines the color histogram difference in (3.1), the average magnitude histogram difference in (3.2), and the orientation histogram difference in (3.5). Formally, for each frame t, we compute

$$CMOHDiff(t) = w_c CHDiff(t) + w_m MDiff(t) + w_d OHDiff(t) \tag{3.6}$$

where w_c, w_m, and w_d are weights associated with the 3 features such that $w_c + w_m + w_d = 1$. Figure 3.7 shows the combined 3 features for a sequence of 200 frames when $w_c = w_m = w_d = 1/3$. As it can be seen, some peaks are very strong and their detection is trivial. However, other peaks are weak and are embedded within noise. The detection of these peaks is a challenging task.

In our approach, we first smooth the histogram difference by median filtering it. To simplify notation, let $f(f)$ denote the histogram difference, that is $f(t) = CMOHDiff(t)$, and let $\hat{f}(t)$ denote the median filtered $f(t)$. Next, the sequence (f) is transformed into the frequency domain using the Fast Fourier Transform (FFT). We use the FFT based on the overlap-add method [71] for two main reasons. First, our data is normally a long sequence that must be filtered in segments. The FFT based on the overlap-add method is used to break a long sequence into smaller segments for efficient processing. Second, combining the overlap-add organization with use of the FFT yields a very efficient algorithm for calculating

convolution that is faster than other methods, while producing exactly the same result. Let $\hat{F}(n)$ denote the FFT of $\hat{f}(t)$. Next, differentiation is applied to $\hat{F}(n)$ to find the zero crossing and find the coarse transition points. The zero crossing points are identified as the points that have a *sign* change from positive to negative and indicate the location of the peaks in the input histogram. The previous steps can result in many peaks that are very close to each other. Thus, we use an additional post processing step that uses a small sliding window and keeps at most one zero crossing per window that corresponds to the maximum peak.

To illustrate the SD_FD approach, in figure 3.8 we show an example of the integrated histogram differences for a sequence of 200 frames from a training video simulation. The blue curve plots the histogram difference, $CMOHDiff(t)$, or simply $f(t)$. The green curve shows the smoothed curve $\hat{f}(t)$ by median filtering and the red curve shows the Fourier transformed signal $F(n)$. The location of the zero crossing points is indicated by the magenta circle signs and the local maxima (using a sliding window) of all magenta circles are displayed using black edge circles. The location of the latter points is taken as the shot boundaries. As it can be seen, in addition to the various peaks in $f(t)$, our approach can detect smaller peaks that do correspond to actual shot boundaries. The local max of the zero crossing can eliminate peaks that are close to each other such as those around frame 95 and frame 115.

Figure 3.9 provides an overview of the proposed SD_FD approach.

3.2.2 Shot Detection based on Unsupervised Learning

In this section, we describe our second approach to shot boundary detection. This approach, called Shot Detection based on unsupervised learning (SD_UL), identifies the change points in a video stream by applying an unsupervised learning technique to a dissimilarity matrix. This matrix is measured by computing the distance between collections of frames. It has been widely used in several audio applications, such as speech recognition [72], music segmentation [73], and cover song identification [74].

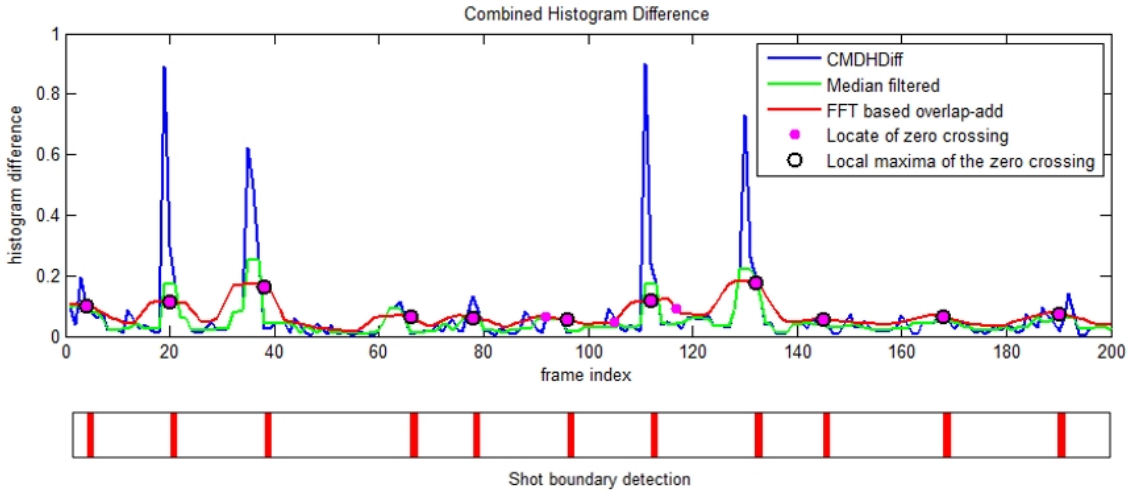


Figure 3.8: Sample output from our proposed video shot detection using Frequency Domain Filters in a sequence of 200 frames.

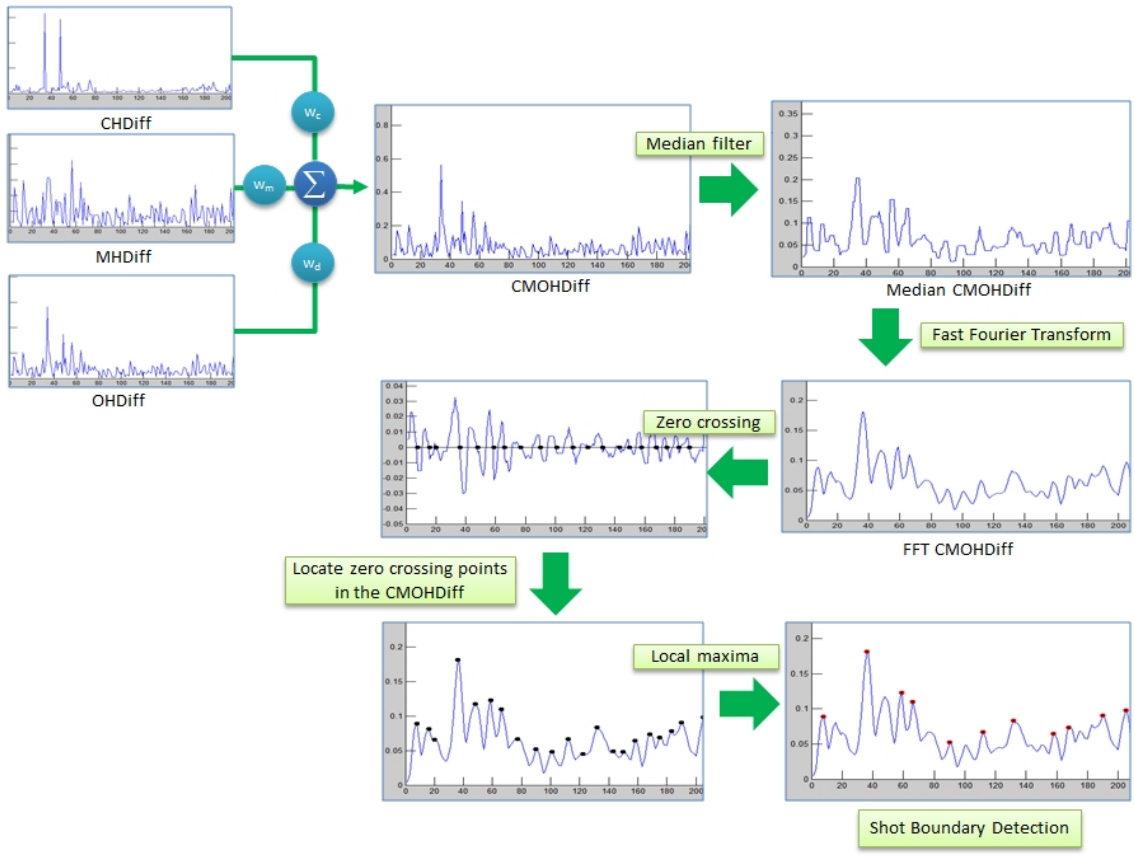


Figure 3.9: Different steps of the proposed SD_FD.

First for a given video stream, we extract the color histogram difference (3.1), the global motion difference (3.2), and the motion orientation histogram (3.5) features. Instead of combining these features as in (3.6), we represent each frame t by a 3-Dimensional feature vector,

$$f(t) = [CHDiff(t), MDiff(t), DHDiff(t)]. \tag{3.7}$$

Next, we define the dissimilarity between frame t and frame $t+1$ as the Euclidean distance between $f(t)$ and $f(t+1)$, i.e.

$$d(t) = \|f(t) - f(t+1)\|^2. \tag{3.8}$$

Local maxima of $d(t)$ can be used to localize the shot boundaries. Instead of using a threshold based approach (as in the SD_FD) to select peaks, we use a statistical learning approach. In particular, we apply an unsupervised learning algorithm to $d(t)$ to identify clusters. Frames assigned to the same cluster will have low distance and will correspond to a shot. Thus, the clusters' boundaries will also correspond to the shot boundaries. To obtain an estimate of the number of clusters and to initialize the cluster centers, we use a sliding window approach to identify the local peaks in $d(t)$. Also, to enforce the constraint that shots should include only consecutive frames, we apply the clustering algorithm to the 2-Dimensional features $(d(t), t)$. Since $t \gg d(t)$, clusters will tend to include only consecutive frames.

Any clustering algorithm could be used for this task. In the current system, we use the Fuzzy C-Mean (FCM) algorithm [75]. After clustering, the local maxima of $d(t)$ within each cluster is identified and selected to be a shot boundary. The proposed SD_UL approach is summarized in figure 3.10.

3.3 Key Frame Extraction

After identifying shot boundaries and partitioning the video into shots, the next step is to identify and select one frame from each shot that is typical of its content. We call these key frames. Most of the early work selects key frames by randomly or uniformly sampling

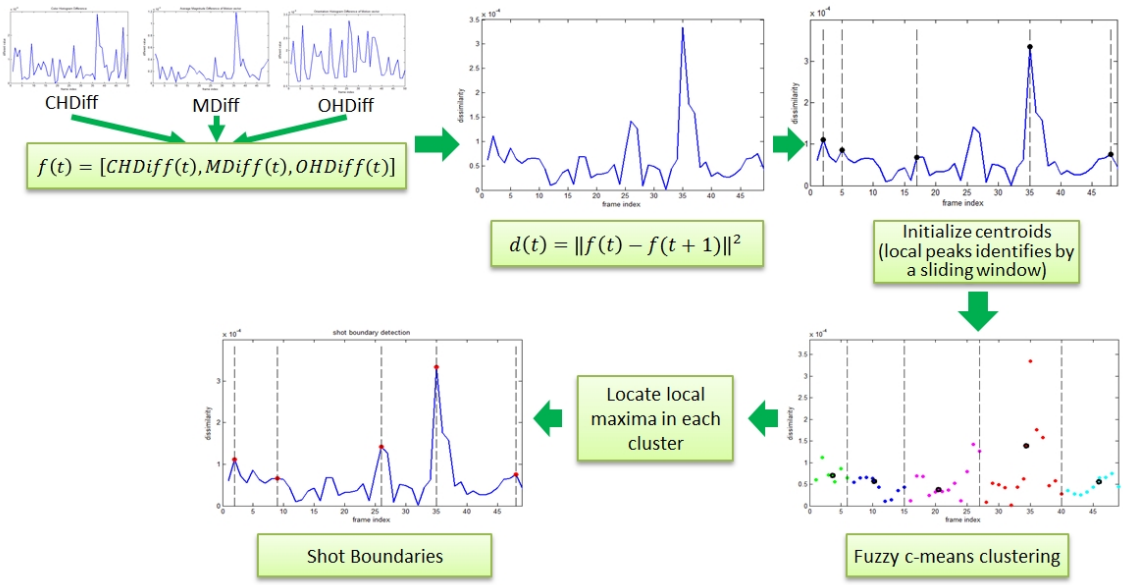


Figure 3.10: Different steps of the proposed SD_UL.

video frames from the original sequence at certain time intervals [76]. Since a shot is defined as a video segment within a continuous capture period, a natural and straightforward way to key frame extraction is to use the first frame of each shot as its key frame [77, 78].

In our system, we use different approaches to select key frames in training and testing modes. In training, the shots are short and uniform (extracted manually). Therefore any frame within the shot could be used as key frame. We simply use the first frame of each shot as its key frame. For testing, every step of the proposed system needs to be automated and unsupervised. Thus, shots could be long and may not be homogeneous. To address this issue, we select the frame that has the least deviation from its neighboring frames.

Let

$$\bar{d}(t) = \sum_{x \in \mathcal{N}(t)} d(x) \tag{3.9}$$

where $d(x)$ is as defined in (3.8) and $\mathcal{N}(t)$ is a fixed neighborhood around frame t. We select frame t^* as key frames of shot k where

$$t^* = argmin_{t \in \mathcal{S}_k} \{\bar{d}(t)\} \tag{3.10}$$

where $\mathcal{S}_k$ is the set of frames within shot k.

3.4 Region Selection

In this section we describe our proposed approach to flag the regions of interest in the key frame in order to track their motion within the video shot. Our approach, inspired by the spatial domain content, is based on clustering local features and identifying the motion magnitude and their skin identification.

In the training phase, the computation efficiency of this operation may not be a major issue as this step is typically performed off-line. The main issue is the accuracy of the extracted information as this will be used to train the models. Therefore, some tasks of this part should be performed manually. The process of region selection in the training phase is outlined in figure 3.11. After identifying shot key frames, each key frame is segmented into a small number of homogeneous regions (detailed in section 3.4.1). In addition, each pixel in the key frame is assigned a probability of being a skin-like pixel (detailed in section 3.4.2). Then, each segmented region is identified as a skin or non-skin region based on the proportion of pixels with high skin-like probability. Finally, we manually select the one skin-like region that can characterize the CPR activity to be the region of interest.

In the testing phase, we developed a CPR video scene query/retrieval operation that is invoked and performed in an interactive way. Hence, this task is performed online, and must be efficient and automated. The block diagram of the region selection in the testing phase is outlined in figure 3.12. The process begins with the feature extraction that consists of optical flow computation [62] (detailed in section 3.4.1) and the edge gradient. To segment the key frame image into regions, we use the normalized cut algorithm (NCut) [79]. Several clustering algorithms could be used to achieve this task. Our choice is motivated by the computational efficiency of this algorithm and its ability to cluster the image into a reasonable number of regions with no supervision information. Next, the average of the magnitude of the optical flow in the key frame is computed for each region. Finally, the region of interest is selected using the skin-like identifications (detailed in section 3.4.2) that has the longest optical flow average magnitude.

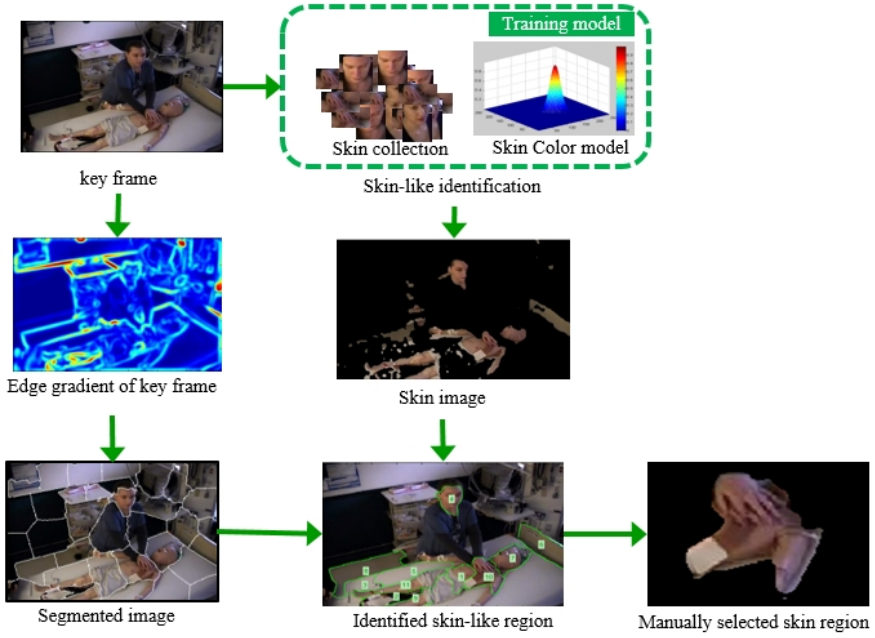


Figure 3.11: Overview of the region selection step in the training phase. Only the last step involves manual selection.

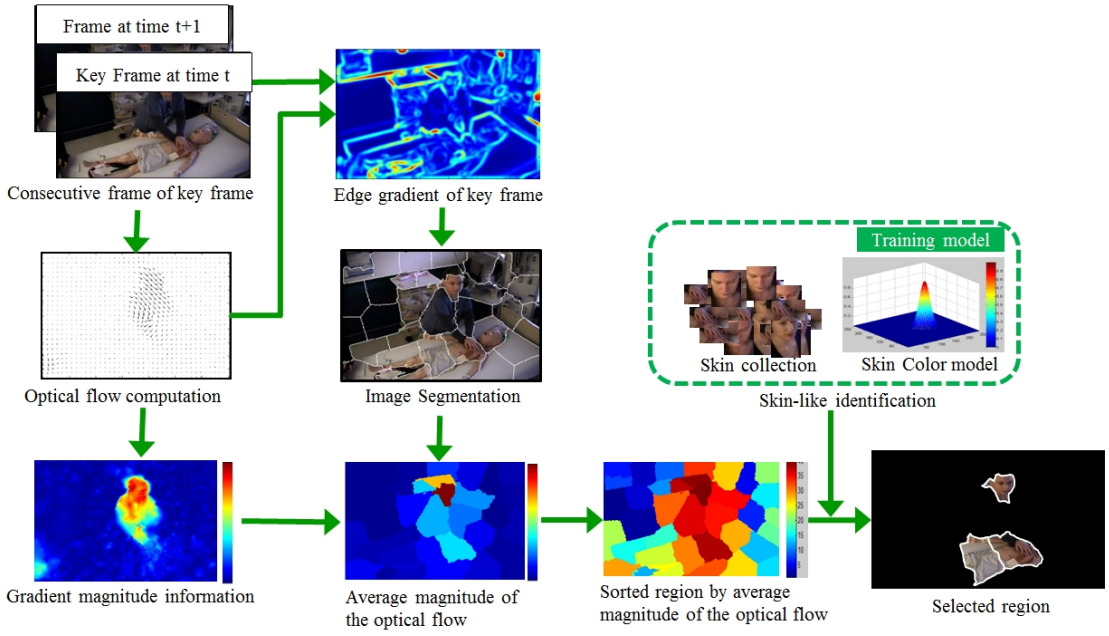


Figure 3.12: Overview of the region selection step in the testing phase.

3.4.1 Key Frame Segmentation

The process of the key frame segmentation starts by subsampling the key frame in order to reduce the time complexity. Then, we represent each pixel (x, y) in the key frame by a 5-dimensional feature vector, $f(x, y)$. The first three components of $f(x, y)$ represent the RGB color information, i.e., $R(x, y)$, $G(x, y)$, and $B(x, y)$. The other two components consist of the motion vector $\bar{u}(x, y)$ and $\bar{v}(x, y)$ extracted using the Horn-Schunck algorithm[62]. The color features and the motion features of all pixels in a frame are normalized to sum to one and averaged to form a single feature at each pixel (x, y), i.e.

$$\bar{f}(x, y) = [R(x, y) + G(x, y) + B(x, y) + \bar{u}(x, y) + \bar{v}(x, y)]/5. \tag{3.11}$$

In the next step, the feature $\bar{f}(x, y)$ is filtered to extract the gradient $E(x, y)$. $E(x, y)$ represents the edge strength at location (x, y). The information of edge gradients is used to segment the key frame into regions of similar pixels. First, we construct the dissimilarity matrix W between all pairs of pixels (x_i, y_i) and (x_j, y_j) using

$$w((x_i, y_i), (x_j, y_j)) = exp^{\frac{-\|E(x_i,y_i)-E(x_j,y_j)\|_2^2}{\sigma_E}} * \begin{cases} \exp^{\frac{-\|(x_i,y_i)-(x_j,y_j)\|_2^2}{\sigma_X}} & \text{if} \|(x_i, y_i) - (x_j, y_j)\|_2 < R \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

In (3.12), $E(x_i, y_j)$ and $E(x_j, y_j)$ are the gradient features at pixel locations (x_i, y_i) and (x_j, y_j), σ_F is a feature tuning parameter, and σ_X is a spatial tuning parameter. We note that in (3.12), there is similarity $w((x_i, y_i), (x_j, y_j)) = 0$ for any pair of pixels that are more than R pixels apart. Finally, the affinity matrix, w, is segmented using the Ncut algorithm [80]. Figure 3.13 illustrates our proposed key frame segmentation approach.

3.4.2 Skin Detection

Instead of processing all detected regions, we identify only those that are of interest. Since our objective consists of identifying CPR scenes, and since this action typically involves the trainee hands and the mannequin chest, we identify and keep only those regions with skin-like colors. To achieve this task, we use a simple but efficient skin pixel classifier

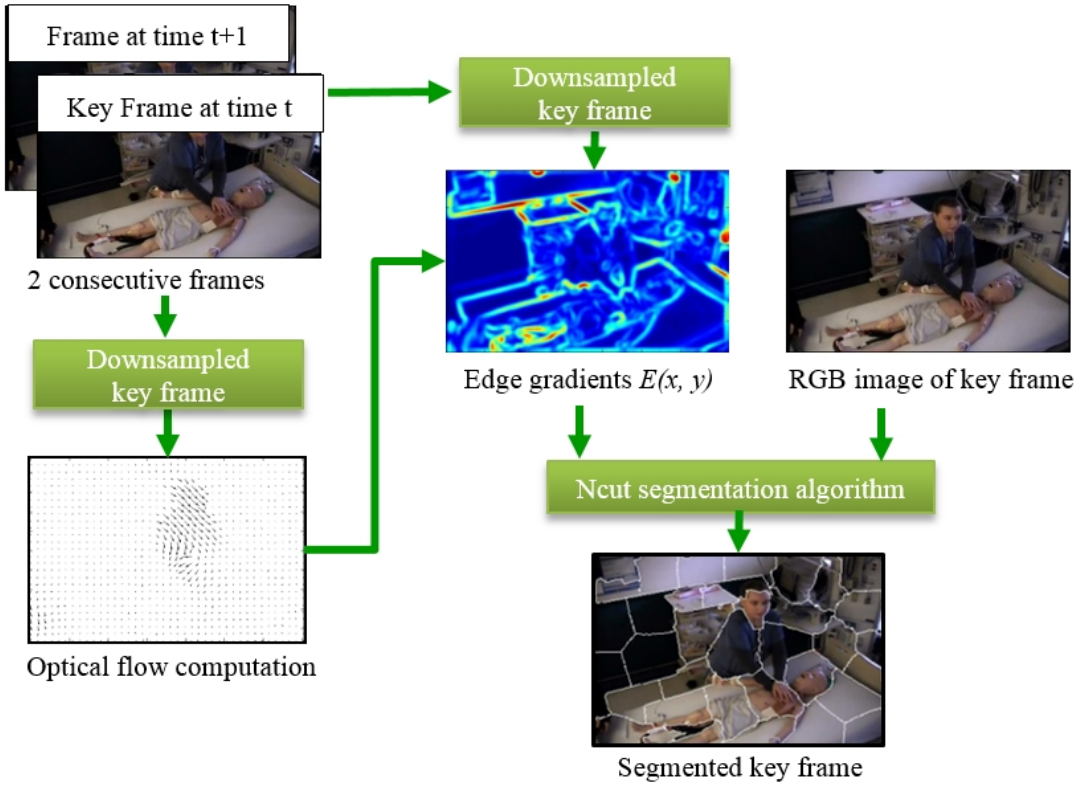


Figure 3.13: Overview of the proposed key frame segmentation algorithm

[81] to discriminate between skin and non-skin regions. This classifier needs to be trained to learn the characteristics of skin regions. We use a collection of skin images to train the classifier. This collection includes 500 images of skin regions at different orientations and under different illumination. These images were selected from regions of segmented key frames. A sample of images from this collection is shown in figure 3.14.

First, each image is mapped into the YC_bC_r color space. The Y, C_b, and C_r components refer to luminance, chrominance blue, and chrominance red, respectively. This YC_bC_r color space has proved to be suitable for representing skin color under different lighting conditions[82]. We only use the chrominance blue and chrominance red component to detect skin. Each image is then filtered using a low-pass filter to reduce the effect of noise.

Second, the color distribution of all pixels in all training images is fitted by a Gaussian

Figure 3.14: Sample images used to train the skin model

model with mean μ and a covariance matrix $\mathcal{C}$. Let x_{ij} be the i^{th} pixel in region R_j of segmented test keyframe with color $(C_{b_{ij}}, C_{r_{ij}})$. The likelihood of x_{ij} in the Gaussian skin model can be estimated using

$$\mathcal{L}(x_{ij}) = \exp[-0.5(x_{ij} - \mu)\mathcal{C}^{-1}(x_{ij} - \mu)]. \tag{3.13}$$

Each image segment is mapped to a likelihood image where the likelihood of each pixel is computed using (3.13). Then, the likelihood images are smoothed again by a low-pass filter. Regions that have a large number of pixels with likelihood larger than a given threshold are retained for further processing. Since different image regions have different illumination and brightness, they require different thresholds. Thus, we use an adaptive threshold to find the optimal threshold for each image region. In particular, we use a sequence of thresholds from 0.75 to 0.05 with a step of 0.1. For each threshold, the number of skin pixels are counted. The optimal threshold is selected to be the one that causes a dramatic change of the total number of skin pixels. Using the optimal threshold, a region is labeled as a skin region if the ratio of its skin pixels to the total number pixel in the region is greater than 50%. Figure 3.15 displays an overview of the proposed skin detection

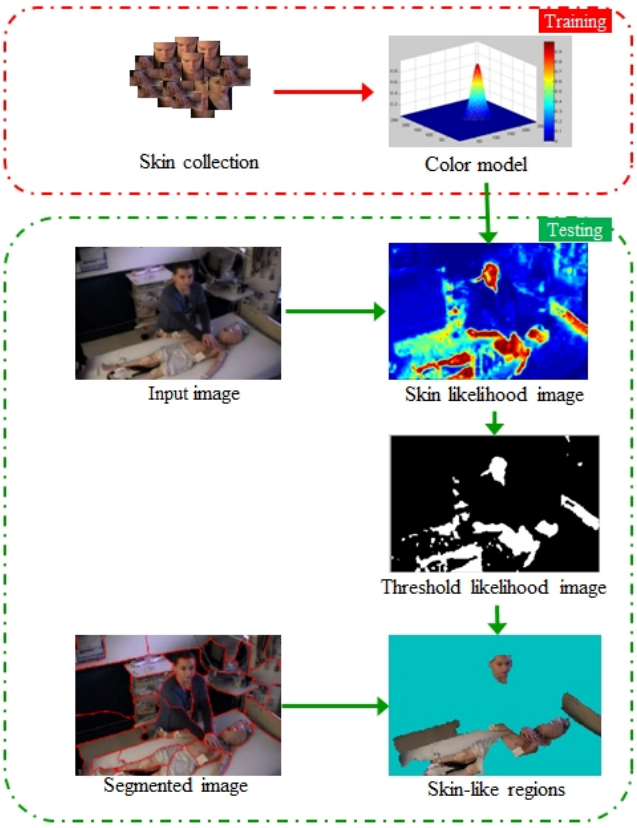


Figure 3.15: Overview of the skin detection process

process.

3.4.3 Face Detection

Most CPR activities in our video data collection are correlated with a human action. In fact, when CPR is being administered, the head of the person performing the action tends to parallel motion of his hands. Thus, to improve the accuracy of our CPR video scene retrieval system, we propose detecting and analyzing the motion of the face of the person performing CPR in addition to analyzing the motion of the skin regions.

Figure 3.16 illustrates our approach to detect faces in the video shot. We start by looking for faces in the key frame of a given video shot. We use the Viola and Jones [83] face detection method. This algorithm can detect faces in real time with very low false alarm rate using Haar-like features trained by the AdaBoost algorithm [84]. If no face is detected

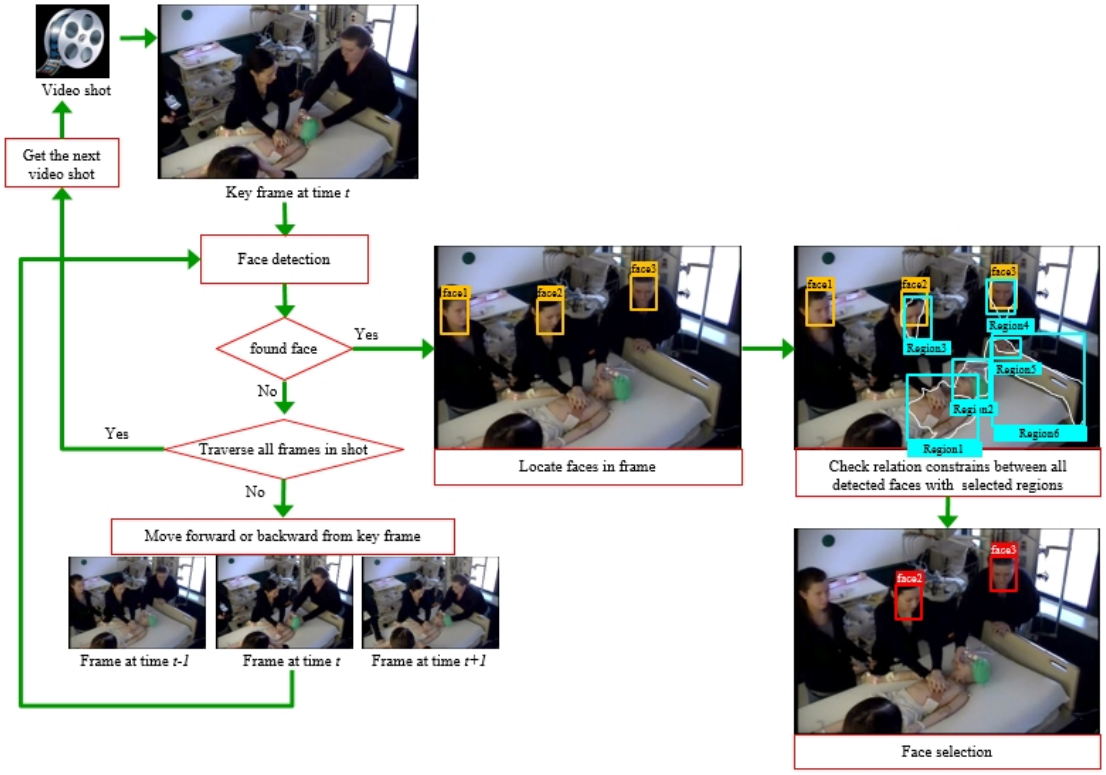


Figure 3.16: Overview of the face detection and selection process

in the shot's key frame, we keep processing adjacent frames until a face is found or all faces have been processed. If no face is detected, the shot will be flagged as a "no-face" shot. Otherwise, if multiple faces are found, we keep only those that are located near a detected region of interest (i.e., skin-like region).

3.5 Observation Generation

To discriminate between skin regions that are involved in CPR activities and other skin regions, we developed and trained a Hidden Markov Model (HMM) classifier that uses motion features. Motion is an efficient feature to analyze moving objects that change location with time [85]. The training phase and the testing phase rely on the same process to generate the observation for CPR identification. The only difference is that for training, only one region of interested is selected manually and used to train the models. For testing,

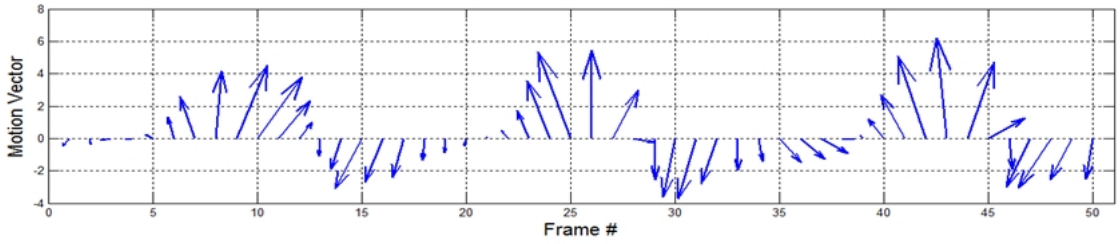
Figure 3.17: Illustration of the motion feature extraction for one key frame

all identified regions of interested will be tested by the HMM.

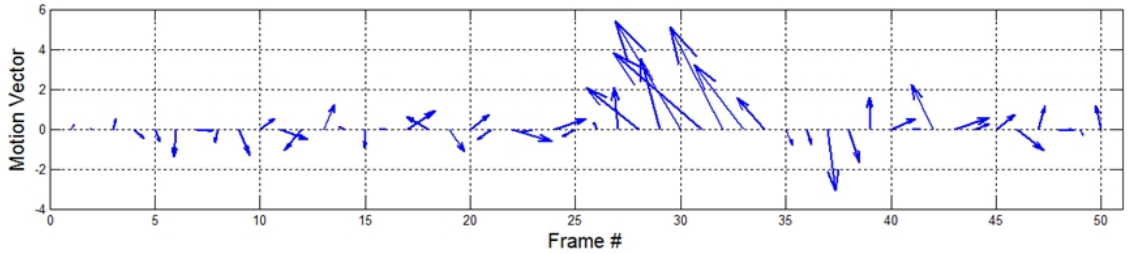
Each video shot will be represented by one or more sequence of observations depending on the number of identified regions of interest and faces. For each face or region of interest, we extract a motion feature vector. First, we compute the optical flow for all pixels within the region using Horn-Schunck algorithm(HS) [62]. Optical flow measures the change in the velocity in terms of speed and direction at each pixel location. Then, the optical flow, $\bar{u}_j = (\bar{u}_j^x, \bar{v}_j^x)$, of a region of interest R_j, is estimated as the average of the optical flow of all of its pixels.

Figure 3.17(a) displays one of the keyframes and Figure 3.17(b) displays the identified skin regions. Figure 3.17(c) displays one selected skin region. In figure 3.17(d), we superimpose the optical flow of each pixel in the region as well as the average optical flow $\bar{u}_j$(in red). In this case, the pixels are moving downward, indicating the push-down phase of the CPR action.

The average optical flow $\bar{u}_j$ is computed for each region within each frame of a given shot. Thus, skin region (or face) R_j within the k^{th} video shot will be represented by the



(a) Average Opticalflow of a typical CPR sequence



(b) Average Opticalflow of a non-CPR sequence

Figure 3.18: Optical flow sequences of 2 sample regions.

sequence of observations:

$$O_{ij} = \{\bar{u}_{jk}^1, \bar{u}_{jk}^t, \cdots, \bar{u}_{jk}^T\}. \tag{3.14}$$

In (3.14), T is the number of consecutive frames within video shot k that includes skin region R_j. Figure 3.18 displays the average optical flow of one skin region that involves CPR activities and another non-CPR skin region over a sequence of 50 frames. As it can be seen, the CPR sequence can be characterized by an upward motion followed by a downward motion. On the other hand, the non-CPR sequence has no specific pattern and the motion vector tends to be random.

3.6 CPR Scene Identification using Hidden Markov Models

A Hidden Markov Model (HMM) is a model of a doubly stochastic process that produces a sequence of random observation vectors at discrete time according to an underlying Markov chain[86]. At each observation time, the Markov chain may be in one of N_s states $s_1, \ldots, s_N$ and, given that the chain is in a certain state, there are probabilities of moving to

other states. These probabilities are called the transition probabilities. An HMM is characterized by three sets of probability density functions, the transition probabilities(A), the state probability density functions(B), and the initial probabilities(π). Let $Q = q_1, \ldots, q_T$ be the state sequence of the T observations of O_{jk}(defined in equation (3.14)). The compact notation

$$\lambda = (A, B, \pi) \tag{3.15}$$

is generally used to indicate the complete parameter set of the HMM model. In [65], $A = [a_{ij}]$ is the state transition probability matrix, where $a(i,j) = Pr(q_t = j | q_{t-1} = i)$ for $i, j = 1, \ldots, N_s$; $\Pi = [\pi_i]$ where $i = Pr(q_1 = s_i)$ is the initial state probabilities; and $B = b_i(O_t)$, $i = 1, \ldots, N$, where $b_i(O_t) = Pr(O_t | q_t = i)$ is the set of observation probability distribution in state i. Hence, A and B are row-stochastic matrices and π is also a stochastic vector:

$$\Sigma_{j=1}^{N_s} a_{ij} = 1;\ \Sigma_{j=1}^{N_s} b_{ij} = 1;\ \Sigma_{j=1}^{N_s} \pi_j = 1; \quad for\ i = 1, \ldots, N_s \tag{3.16}$$

3.6.1 Discrete Hidden Markov Model

In a discrete hidden Markov model (DHMM), the observation probability density functions are characterized by discrete symbols chosen from a finite set of symbols, $v_1, v_2, \ldots, v_M$, called the codebook V. In other words, B becomes a simple set of fixed probabilities for each class, which is an NxM matrix with

$$b_i(o_t) = Pr(o_t = v_j | q_t = s_i); \quad for\ i = 1, \ldots, N_s \tag{3.17}$$

where v_j is the symbol of the nearest code to observation o_t, according to a distance measure. Usually, clustering algorithms such as K-mean clustering [87], Fuzzy C-Mean clustering algorithm [75], or a vector quantization (VQ) algorithm [88] can be used to convert the measurements into discrete observations.

In our system, we use a discrete HMM classifier with 2 models, one for CPR sequences and another one for non CPR sequences. Each model produces a probability value by backtracking through the model states using the Viterbi algorithm [86]. The CPR model,

λ^{CPR}, is designed to capture the smooth transition among the states that characterizes CPR sequences. The non CPR model, $\lambda^{\sim CPR}$, is needed to capture the characteristics of non CPR sequences.

3.6.2 Continuous Hidden Markov Model

For the continuous HMM (CHMM), the observation probability density functions are continuous. In general, a mixture of Gaussian is used to model the observation probability density, i.e.,

$$b_i(o_t) = Pr(o_t|q_t = s_i) = \sum_{j=1}^{M} u_{ij}\mathcal{N}(o_t, \mu_{ij}, C_{ij}) \tag{3.18}$$

such that

$$\sum_{j=1}^{M} u_{ij} = 1, u_{ij} \geq 0. \tag{3.19}$$

In (3.18), u_{ij} is the mixture coefficient for the j^{th} mixture component in state i, and M denotes the number of mixture components per Gaussian mixture model. $\mathcal{N}$ is a kernel of multivariate Gaussian distribution with mean vector μ_{ij} and covariance matrix C_{ij}.

As in the discrete HMM, in our system we use the continuous HMM with 2 models, one for CPR sequences and one for non CPR sequences. For both DHMM and CHMM, the probability value produced by the CPR (non CPR) model can be thought of as an estimate of the probability of the observation sequence given that there is a CPR (non CPR) model present. These probabilities are produced by back-tracking through the model states using the Viterbi algorithm [86]. The confidence value assigned to each observation sequence, $Conf(O)$, depends on : (1) the probability assigned by the CPR model, $Pr(O|\lambda^{CPR})$, and (2) the probability assigned by the non CPR model, $Pr(O|\lambda^{\sim CPR})$. We use:

$$Conf(O) = Pr(O|\lambda^{CPR}) - Pr(O|\lambda^{\sim CPR}) \tag{3.20}$$

CHAPTER 4

EXPERIMENTAL RESULT

4.1 Data Collection

To validate our proposed approach to detect CPR shots, we use four video simulation sessions recorded by the SPARC group. The frame rate of each video is 29.97 frames per second and the resolution in each frame is 720x480 pixels. The size of each video is listed in table 4.1. To provide an idea of the content of the video, in table 4.2, we show key frames of 4 sample shots in each video.

4.2 Analysis of the Proposed System Using CPR1 video

In the first experiment, we use CPR1 video to analyze our proposed system with different parameter settings. After shot boundary detection, key-frame segmentation, and skin detection, we obtain a total of 360 sequences that track various skin regions. We track each region, R_j, over all frames within its shot boundaries, and extract the sequence of observations O_{jk} as defined in (3.14). For validation purposes, we examine each sequence and assign ground truth labels (CPR or non-CPR). We obtain a total of 200 sequences

TABLE 4.1

Statistics of the 4 video simulation sessions used in our experiments.

Video	Duration	Number of frames	Number of CPR scene manually label	Avg number $\bar{x}$ of frame per CPR scene	Standard deviation σ of frame per CPR scene
CPR1	19 m 28 s	35000	36	258.1	132.2
CPR2	16 m 1 s	28831	36	259.7	173.3
CPR3	14 m 57 s	26905	3	2180	2108.3
CPR4	21 m 49 s	39253	24	493.75	513

TABLE 4.2

Key frames of 4 sample shots extracted from the 4 video simulation sessions.

Video name	Key frame in a shot n1	Key frame in a shot n2	Key frame in a shot n3	Key frame in a shot n4
CPR1				
CPR2				
CPR3				
CPR4				

labeled positively as representing CPR activity, and 160 non-CPR sequences. We also fix the sequence length T to 20 even though HMM can handle sequences of variable length. This is a reasonable assumption as 20 sequences tend to cover the upward and downward motion of the hands in most CPR sequences. Moreover, HMM classifiers are simpler and more efficient when all sequences have a constant length. We use a 4-fold cross validation. For each fold, a subset of the data is used for training (D_{T_r}), and the remaining data is used for testing(D_{T_s}). In the testing phase, all tasks including shot boundary detection, key frame extraction, region selection, and observation generation are performed in a completely unsupervised manner. As described in the previous chapter, for region selection, we track multiple regions for each shot, R_{ij}, under the condition that each region has a high motion magnitude and most of its pixels have high likelihood in the skin-like models. In our experiment, each video shot can have 1 to 6 regions of interest to track and test. For each region of interest R_{jk}, we compute the average optical flow of its pixels to generate the

sequence of observations O_{jk} as defined in (3.14).

4.2.1 Discrete HMM Model Training

For training, we build a two-model *CPR* classifier: one Discrete HMM (DHMM) model, λ^{CPR}, based on *CPR* training sequences ($D_{T_r}^{CPR}$); and another model, $\lambda^{\sim CPR}$, based on *non-CPR* training sequence ($D_{T_r}^{\sim CPR}$). We experiment with two classifiers with different complexity. The first one uses two states s_1 and s_2 that are estimated using domain knowledge. One state represents the upward motion of the hands and the second one represents the downward motion. The *non-CPR* DHMM model, $\lambda^{\sim CPR}$, also has two states. However, no prior knowledge could be used to guide the estimation of these states. For both models, we heuristically estimate the states using

$$s_1 = average\{\bar{u}_j \in D_{Tr}^{CPR} | \bar{u}_j^y \geq 0\} \tag{4.1}$$

$$s_2 = average\{\bar{u}_j \in D_{Tr}^{CPR} | \bar{u}_j^y < 0\}. \tag{4.2}$$

For the second classifier, we use DHMM with 4 states s_1, s_2, s_3 and s_4 that are fixed by considering the movement not only in the vertical direction but also in the horizontal direction. Similarly, the *non-CPR* DHMM, $\lambda^{\sim CPR}$, has 4 states as well. For both models, the 4 states are estimated heuristically using

$$s_1 = average\{\bar{u}_j \in D_{Tr}^{CPR} | \bar{u}_j^x \geq 0 \text{ and } \bar{u}_j^y \geq 0\} \tag{4.3}$$

$$s_2 = average\{\bar{u}_j \in D_{Tr}^{CPR} | \bar{u}_j^x < 0 \text{ and } \bar{u}_j^y \geq 0\} \tag{4.4}$$

$$s_3 = average\{\bar{u}_j \in D_{Tr}^{CPR} | \bar{u}_j^x < 0 \text{ and } \bar{u}_j^y < 0\} \tag{4.5}$$

$$s_4 = average\{\bar{u}_j \in D_{Tr}^{CPR} | \bar{u}_j^x \geq 0 \text{ and } \bar{u}_j^y < 0\}. \tag{4.6}$$

For both classifiers, the remaining parameters (A, B, π) of each λ^{CPR} and $\lambda^{\sim CPR}$ are initialized as follows. The priors, π_1 and π_2 (and π_3 and π_4 for models with 4 states) are estimated as the percentage of training sequences that start with state s_1 and s_2 (and

s_3 and s_4) respectively. The transition matrix A is initialized using the state transition probability distribution for each model using:

$$a_{ij} = \frac{t_{ij}}{\sum_{i=1,j=1}^{N_s} t_{ij}}, \tag{4.7}$$

where t_{ij} is the number of observations that change from state i to state j within each training sequence. For the 2 states models, the transition matrix is initialized using the identity matrix, i.e.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.8}$$

and for the 4 states models, the transition matrix is defined as

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.9}$$

We use a codebook with $M = 20$ codewords $\{v_k, k = 1, \ldots, M\}$. These codewords are estimated as the centers obtained by clustering the training data using the K-mean algorithm [87] with $k = 20$. The emission probabilities are first initialized using

$$b_j(k) = \left[\sum_{i=1}^{N_s} \frac{\|v_k - s_j\|}{\|v_k - s_i\|} \right]^{-1} \tag{4.10}$$

for $j \in \{1, \ldots, N_s\}$ and $k \in \{1, \ldots, M\}$, where N_s is the number of states ($N_s = 2$ or 4). These probabilities are then normalized such that the probabilities in each state sum to one. We use

$$b_j(k) \leftarrow \frac{b_j(k)}{\sum_{i=1}^{M} b_i(m)}. \tag{4.11}$$

After initialization, the transition probabilities A and the observation probabilities B of each model are fined-tuned using Baum-Welch learning algorithm [89] and the respective training data.

Figure 4.1 and 4.2 display the observation vectors used to train the CPR model (first cross validation set). We also display the means of the states and the codewords obtained

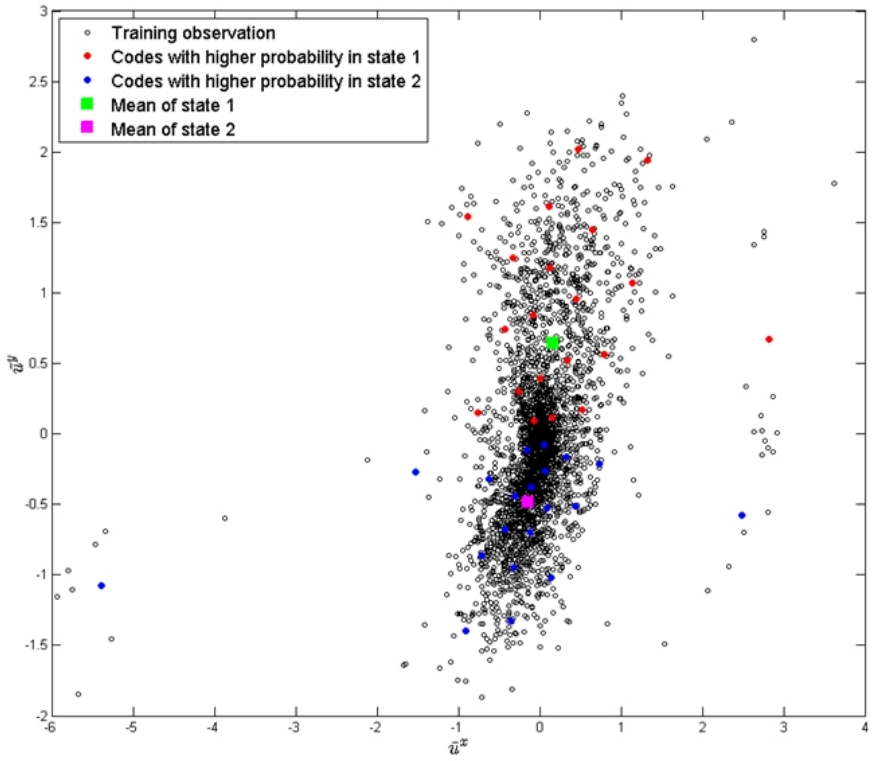


Figure 4.1: Training observations for CPR class, states representatives, and codewords for the DHMM classifier with 2 states.

by clustering the observations. Figure 4.1 shows the results when 2 states per model are used and figure 4.2 show the results for 4 states. As it can be seen, for both cases, the codes summarize the training samples.

In figure 4.3, we display a sample CPR testing sequence. We display the motion vector of 20 observations of a typical CPR sequence. As it can be seen, this sequence corresponds to a region that starts with a downward motion (first 9 observations), then followed by upward motion (8 observations), and finally back to downward motion (3 observations). The optimal state sequence, assigned by the Viterbi algorithm [86], to this test sample is "2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2".

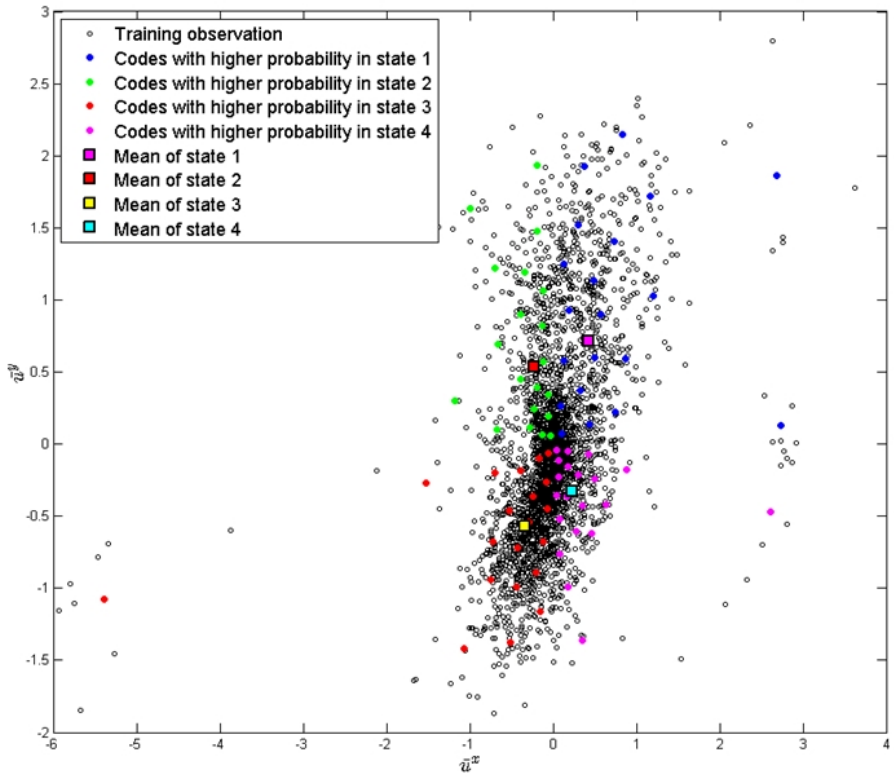


Figure 4.2: Training observations for CPR class, states representatives, and codewords for the DHMM classifier with 4 states.

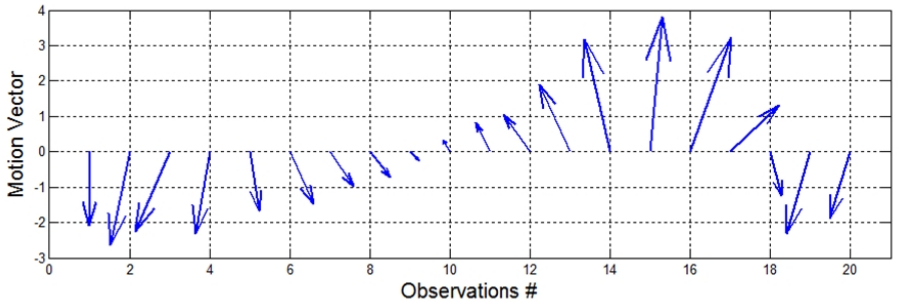


Figure 4.3: 20 Motion vectors of sample testing a CPR sequence.

4.2.2 Continuous HMM Model Training

We have also experimented with the continuous HMM (CHMM). As in the discrete case, we build a two-model CPR classifier: one CHMM model, λ^{CPR}, that represents the CPR training sequences ($C_{T_r}^{CPR}$); and another model, $\lambda^{\sim CPR}$, that represents the non-CPR training sequence ($C_{T_r}^{\sim CPR}$). We assume that the CPR CHMM model, λ^{CPR}, has two states s_1 and s_2 as in the DHMM models. The non-CPR CHMM model, $\lambda^{\sim CPR}$, also has two states. The probability transition matrix A of each model is estimated using (4.7). The priors, π_1 and π_2 are estimated as in the discrete case as the percentage of training sequences that start with state s_1 and s_2 respectively.

For the parameter B, we use a mixture of $M = 3$ Gaussian components. Let $b_i(k)$ be the probability density function of observation vector k in state i. $b_i(k)$ can be written as :

$$b_i(k) = \sum_{j=1}^{M} u_{ij} b_{ij}(k) \tag{4.12}$$

such that

$$\sum_{j=1}^{M} u_{ij} = 1, u_{ij} \geqslant 0. \tag{4.13}$$

In (4.12), u_{ij} represents the mixture coefficient for the j_{th} component in state i, and $b_{ij}(k)$ is a normal distribution $\mathcal{N}(k, \mu_{ij}, \sum_{ij})$ with mean vector μ_{ij} and diagonal covariance matrix $\sum_{ij}$ of observation vector k, such that

$$\mathcal{N}(k, \mu_{ij}, \sum_{ij}) = \frac{1}{(2\pi)^{p/2} \left|\sum_{ij}\right|^{1/2}} exp^{-\frac{1}{2}(k-\mu_{ij})^T} \sum_{ij}^{-1} (k - \mu_{ij}). \tag{4.14}$$

In (4.14), $|\sum_{ij}|$ is the determinant of the covariance matrix.

After initialization, the parameters are fine tuned using Baum-Welch algorithm [89]. Figure 4.4 displays the observation vectors used in the first cross validation set to train the CPR model. In this figure, we also display the learned 3 components of each state.

4.2.3 Performance Evaluation

To evaluate the performance of the proposed approach to identify CPR shots, we use the receiver operating characteristic (ROC). The ROC is a graphical plot commonly

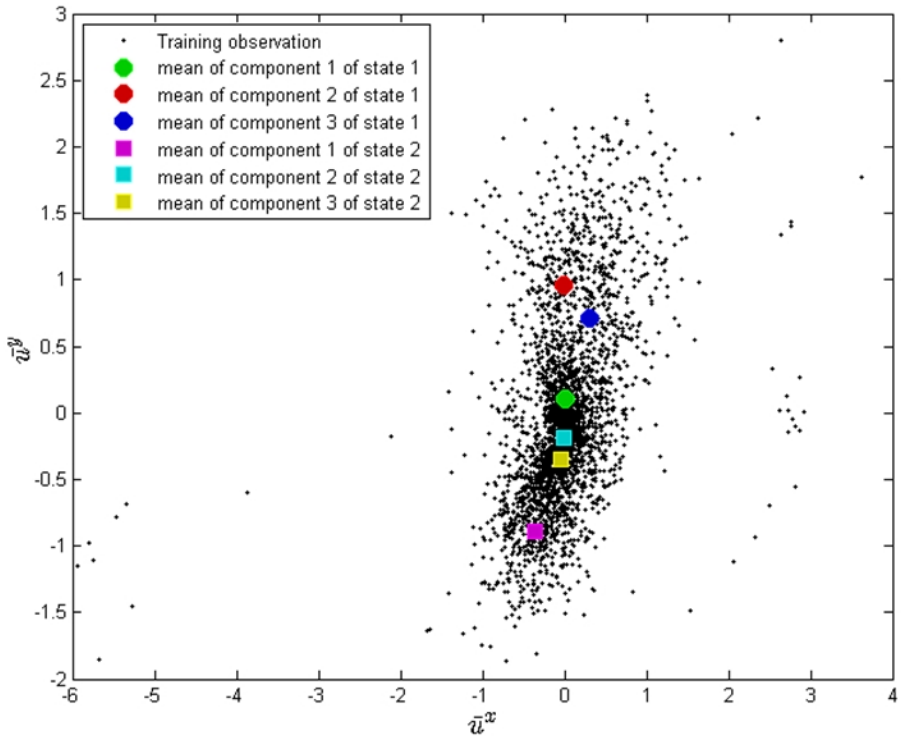


Figure 4.4: 4 Training observations and the means of the 3 Gaussian components of each state for the CPR model training with CHMM.

used to evaluate the quality of a binary classifier as the discrimination threshold is varied. The ROC is created by plotting the probability of detection (PD) versus probability of false alarms (PFA).

4.2.4 Results

First, we evaluate the performance of our approach using the manually segmented shots. Recall that for the manual segmentation, we also manually select the best region of interest. The purpose of this experiment is to illustrate that the HMM classifiers, with motion-based observation features, can discriminate between CPR and non CPR sequences. Figure 4.5 compares the performance of the DHMMs (with 2 and 4 states) and the CHMMs (with 2 states). For all cases, the confidence value is computed using (3.20) which is the difference in the confidence assigned by the CPR model and the one assigned by the non-

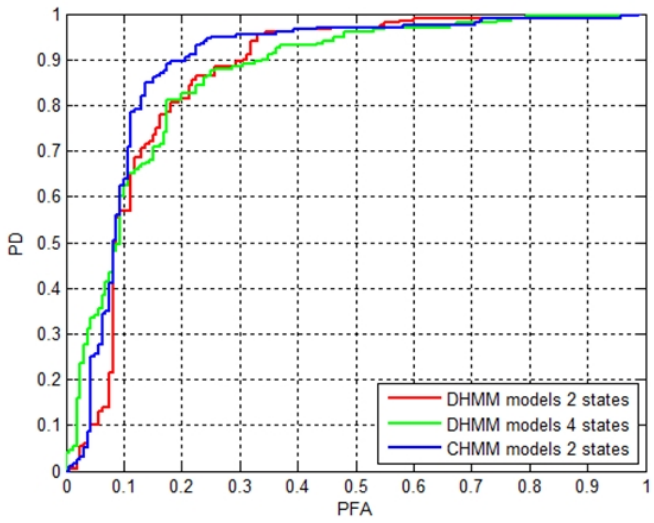


Figure 4.5: The ROCs generated using DHMM with 2 and 4 states and CHMM with 2 states.

CPR model. As it can be seen, our approach can achieve high probability of detection with low probability of false alarms. For instance, using the CHMM with 2 states, we can detect 80% of the CPR shots with only 10% of false alarms. Careful investigation of the missed CPR shots revealed that most of them were due to a non-perfect shot segmentation. For instance, there are shots that include frames with no CPR actions. As a result, the HMM confidence value of the entire sequence is reduced. From figure 4.5, we also note that the CHMM has the best overall performance.

Manual video segmentation and region selection is not practical and need to be by automated. In the next experiment, we evaluate the proposed approaches to automate the shot segmentation and region of interest selection. In particular, we compare the performance of our shot detection based on frequency domain filtering (SD_FD) and shot detection based on unsupervised learning (SD_UL) algorithms. For the region of interested selection, we compute the confidence assigned to the candidates of all regions candidates and selected the one with the maximum confidence in the λ^{CPR} HMM model.

Figure 4.6 compares the ROC curves generated using the SD_FD and SD_UL shot boundary detection methods. For this experiment, we use HMMs with 2 states for both

TABLE 4.3

Evaluation of the proposed CPR detection with the SD_FD and SD_UL shot detection methods.

Shot detection method	Number of detected shots	Number of CPR shots	Number of non-CPR shots	Area under the ROC
SD_FD	165	66	99	0.806
SD_UL	299	82	217	0.863

the continuous and discrete case. As it can be seen, the SD_UL method outperforms the SD_FD one. In table 4.3, we report other statistics that compare the two methods. As it can be seen, the SD_UL approach segments the videos into a larger number of shots (299 vs 165). In other words, the SD_UL segments the video into smaller shots. Most of these are non-CPR shots (217 vs 99). However, the HMM classifier can reject most of these non-CPR shots. The performance could also be measured by the area under the ROC. The larger the area, the more accurate the classifier is. For the remaining experiments in this video, we will use the SD_UL approach to segment the videos into shots. In the next experiment, we illustrate the advantage of integrating the face detection and tracking component into our system. Recall that for this component, we detect faces in the key frame. If multiple faces are detected, we keep the one that is most likely associated with the selected region of interest. In figure 4.7, we display a scatter plot of the DHMM confidence values of the selected regions and the selected faces associated with them. If a given shot no face was detected, the confidence value is set to a low confidence constant (-0.15). As it can be seen, most confidence values are correlated. However, there are some CPR scenes, e.g. those highlighted in region R_1, where the DHMM models tested with the observations generated by face regions is more reliable than the observations generated by the selected skin region. Similarly, other CPR scenes, e.g., those highlighted in region R_2, can be detected more reliably using the skin region of interest. For other scenes, such as those highlighted in region R_3, both confidence values need to be combined to improve the accuracy.

In figure 4.8, we compare the ROCs obtained when DHMM classifiers were used to track skin regions, track faces, and track the two combinations (using their sum). The AUC

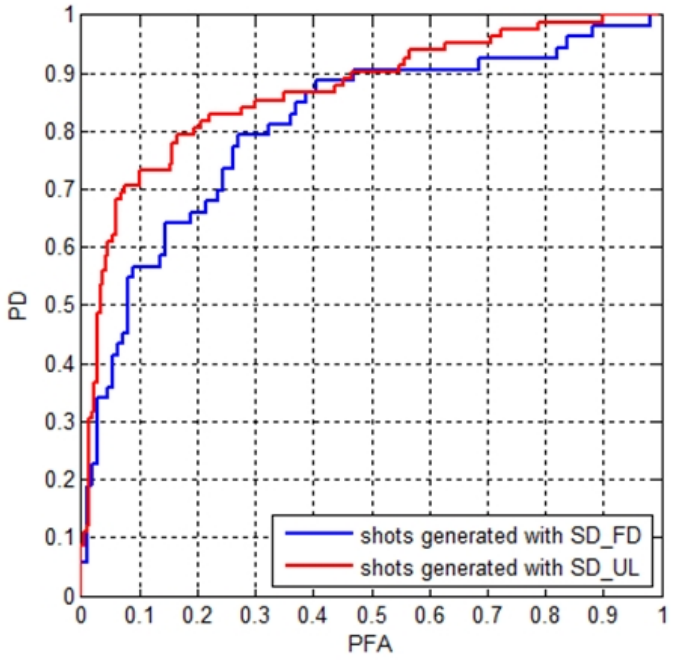


Figure 4.6: ROC generated by the DHMM when the video is segmented into shots using SD_FD and SD_UL methods

of the DHHM classifiers using observation sequences generated by face regions, skin regions, and a combination of the two are 0.685, 0.863, and 0.889 respectively. As it can be seen, the observation sequence generated by the face regions can not identify CPR scene reliably. However, when combined with skin regions, they can improve the performance significantly.

Another way to visualize the results of the proposed system is illustrated in figure 4.9. In this figure, for each shot, we display its ground truth (CPR or non CPR), detection results using skin regions only, faces only, and a combination of the two. This figure shows whether the shots are detected/missed by one or both features.

4.3 Analysis of the Proposed System on Multiple Video Recordings

In this experiment, we use the learned models in CPR1 to test the remaining 3 videos CPR2, CPR3, and CPR4. In the training phase, our proposed system was validated using 4-fold cross validation on CPR1. Since 4 models were learned, in testing the new videos,

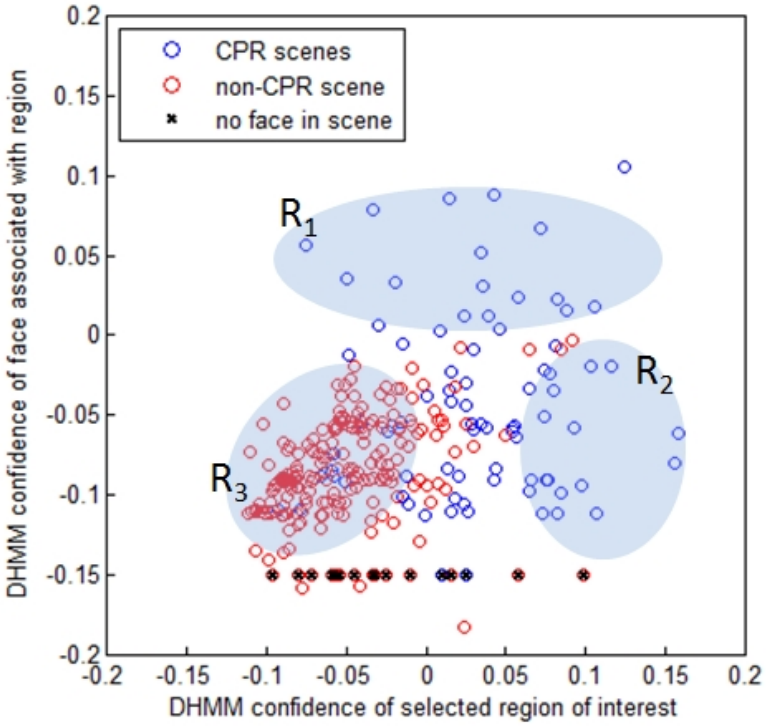


Figure 4.7: Scatter plot of the confidence of the observation sequence generated by the selected region of interest and the observation sequence generated by the face region.

TABLE 4.4

Comparison of the performance of the different DHMM models on all 3 video sessions.

Video Name	1^{st} DHMM models	2^{nd} DHMM models	3^{rd} DHMM models	4^{th} DHMM models	Average of 4 models
CPR2	0.653	0.658	0.637	0.638	0.647
CPR3	0.654	0.607	0.566	0.626	0.613
CPR4	0.756	0.772	0.728	0.754	0.753

each sequence will be tested with the 4 models, and the average of the 4 confidences will be used. For all videos, we use the SD_UL shot detection method to segment video into shots. All other parameters are as learned using CPR1. In table 4.4, we report the AUC with ROC of the 3 videos. The ROCs of video CPR2, CPR3, and CPR4 are shown in figures 4.10, 4.11, and 4.12, respectively.

As it can be seen, the performance is not as good as in CPR1. For instance, our

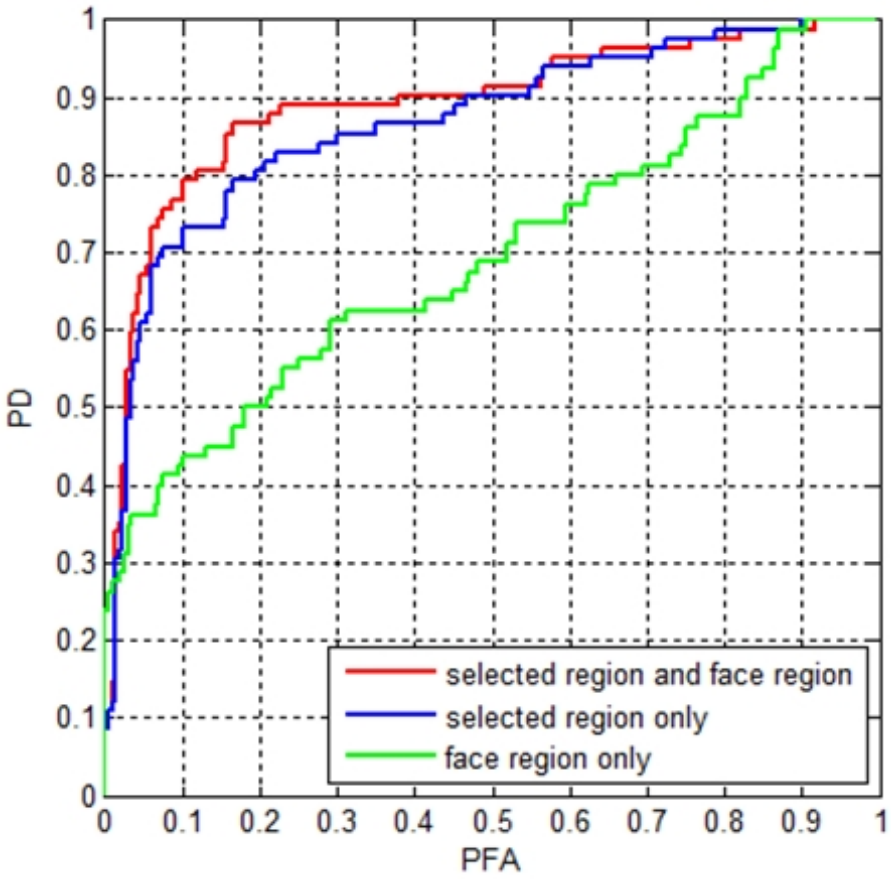


Figure 4.8: Performance of the proposed system when both skin regions and faces are tracked.

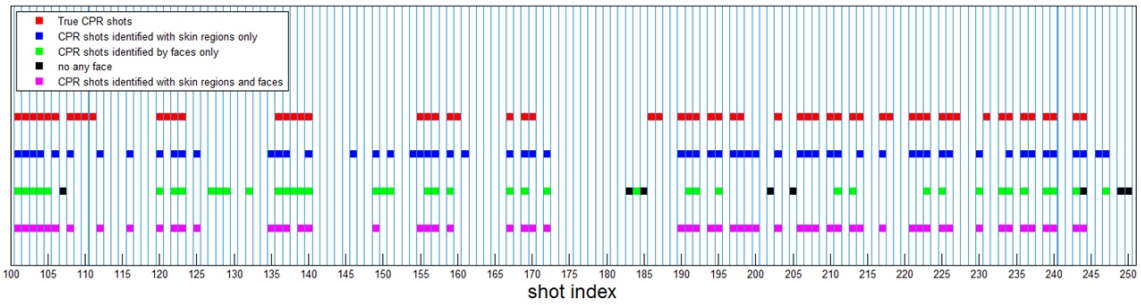


Figure 4.9: Comparison of the CPR identification when skin regions and faces are tracked for 150 shots extracted from video CPR1.

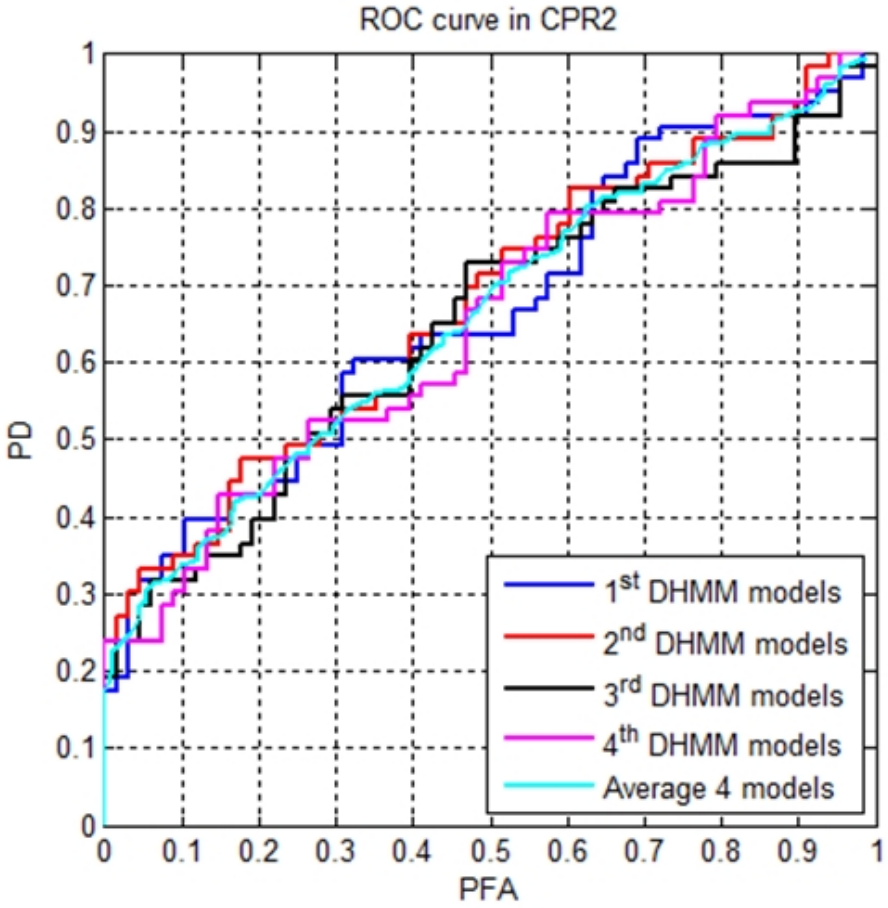


Figure 4.10: Comparison of the 4 DHMM models for CPR2 video.

classifier has incorrectly identified some shots CPR sequences. Examples of these misclassifications are shown in figure 4.13 and figure 4.14. In figure 4.13, the region with the highest confidence is identified as the hand which is pumping the respiratory pump. As it can be seen, the movement of that region has the same characteristics as the motion involved in a CPR activity. Similarly, in figure 4.14, our application has selected the head area of the subject checking the dummy to prepare for CPR. As this figure shows, the head is moving up and down similar to the motion in a CPR activity.

In addition to the larger number of false alarms, our classifier has also missed the detection of a few CPR scenes. For instance, in figure 4.15 the CPR scene was not classified correctly due to bad segmentation of the key frame. Figure 4.16 shows another CPR scene

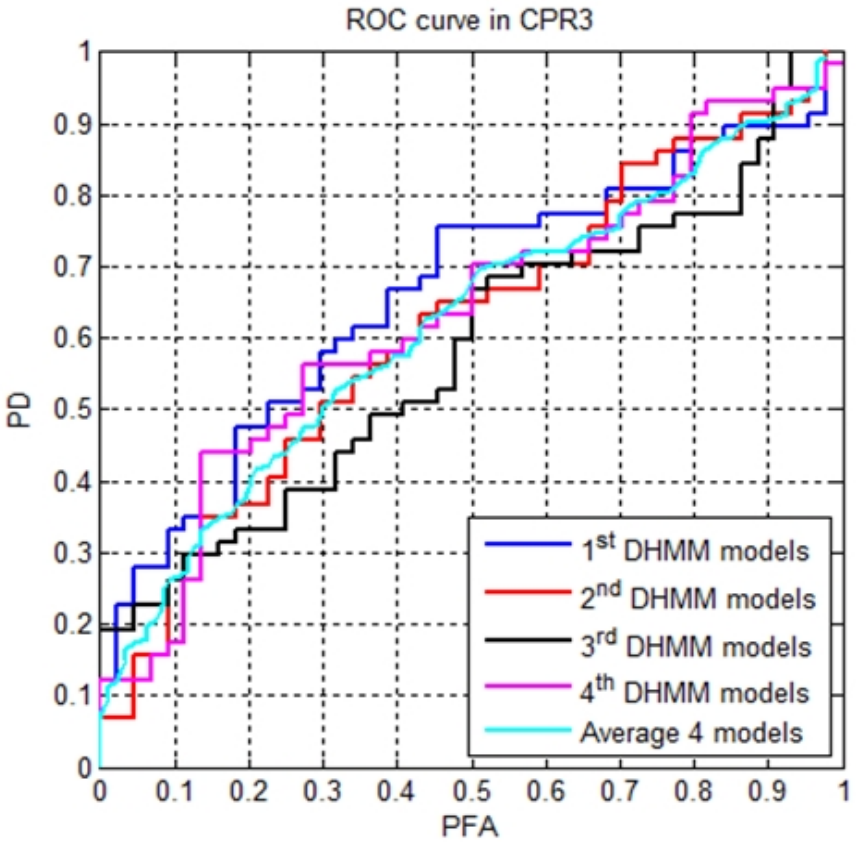


Figure 4.11: Comparison of the 4 DHMM models for CPR3 video.

that was not detected. In this case, the mannequin is similarity an infant and the magnitude of the motion is very small.

In figure 4.17, 4.18, and 4.19, we display the ground truth (CPR and non CPR) for each detected shot. As it can be seen, in figure 4.17 and 4.18 the poor performance on CPR2 and CPR3 video can be explained by the high number of missed CPR scenes. One reason for this poor performance is that the settings for these videos are different from the setting in CPR1 video (camera location, new angle, frame resolution). Another reason is that the mannequin in CPR4 video is simulating an infant and the CPR actions there involve motion with much smaller magnitude. Also, we note that in CPR4 video there is a large number of false detections at the beginning of the video. Investigation of these false detections revealed that they are caused by activities of the subject while preparing for the

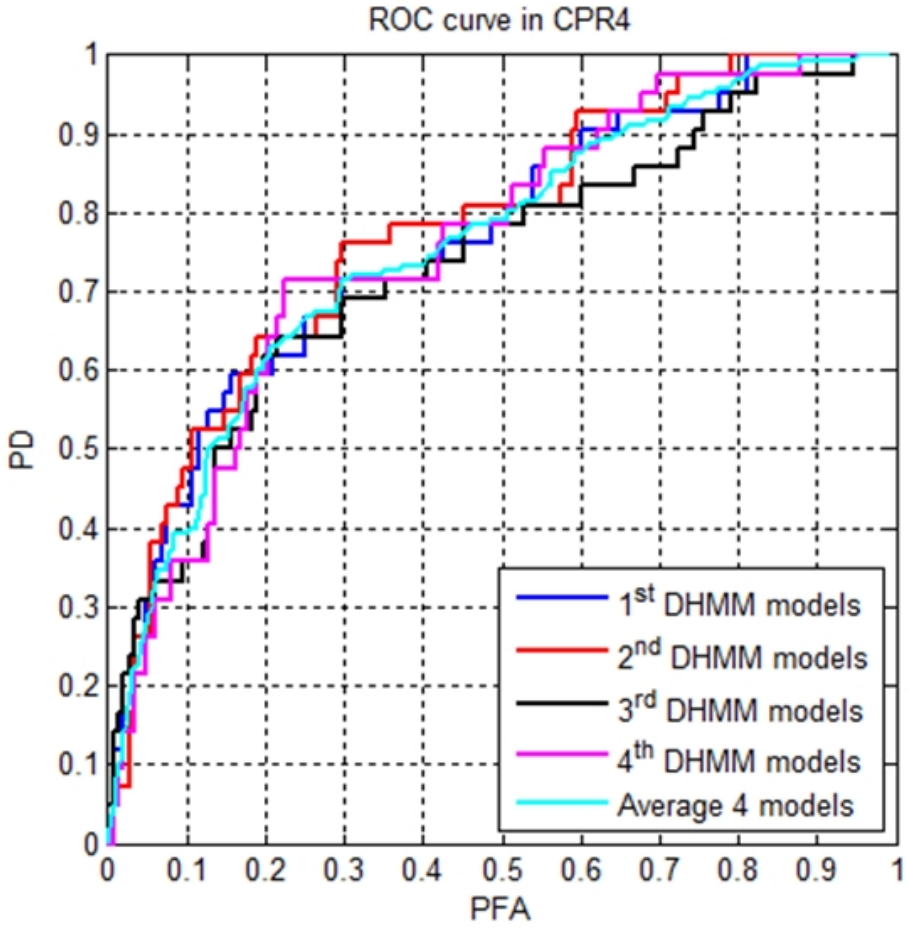


Figure 4.12: Comparison of the 4 DHMM models for CPR4 video.

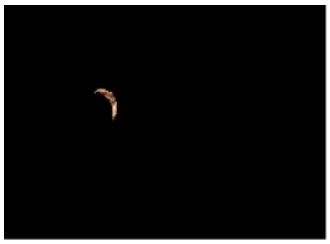
simulation.

4.4 A Graphical User Interface for CPR Video Scene Retrieval and Analysis

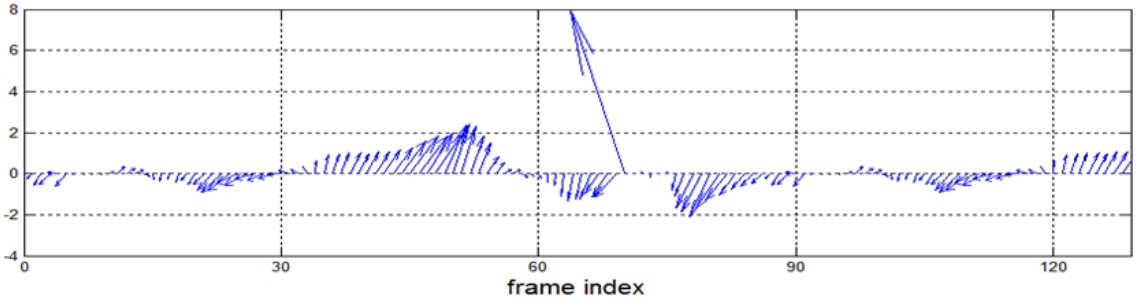
We have developed a CPR scene video retrieval and analysis prototype system with graphical user interfaces (GUI). This GUI has several desirable features that allow the user to view and analyze the video training session much more efficiently. A block diagram of the developed GUI is shown in figure 4.20. Figure 4.21 illustrates the main functionalities of the GUI. First, the user selects the video sequence in the database using the "Open video" command button (refer to step 1 and 2 in figure 4.21). After a video has been selected, it is played in the window media player (refer to step 3 in figure 4.21). Generic statistics of

(a) Key frame of shot 153.

(b) Region of interest of shot (a).

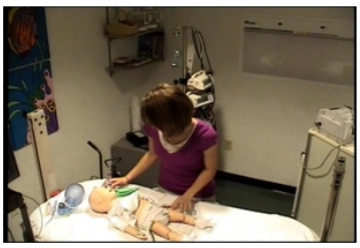


(c) Region with max confidence.

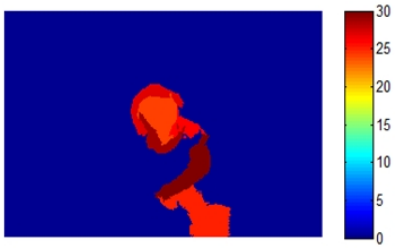


(d) Optical flow sequence.

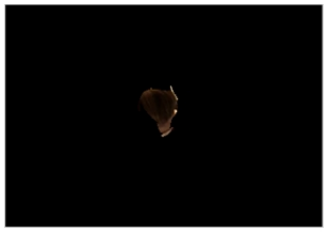
Figure 4.13: A sample false alarm from CPR2 video.



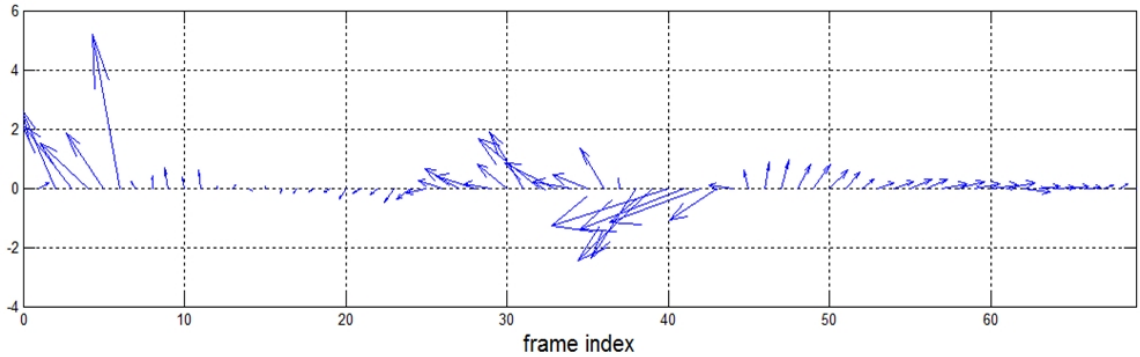
(a) Key frame of shot 39.



(b) Region of interest of shot (a).

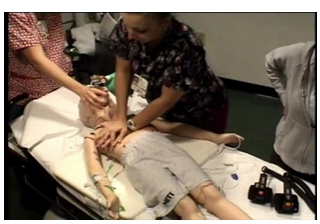


(c) Region with max confidence.



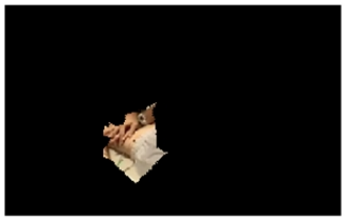
(d) Optical flow sequence.

Figure 4.14: A sample false alarm from CPR4 video.

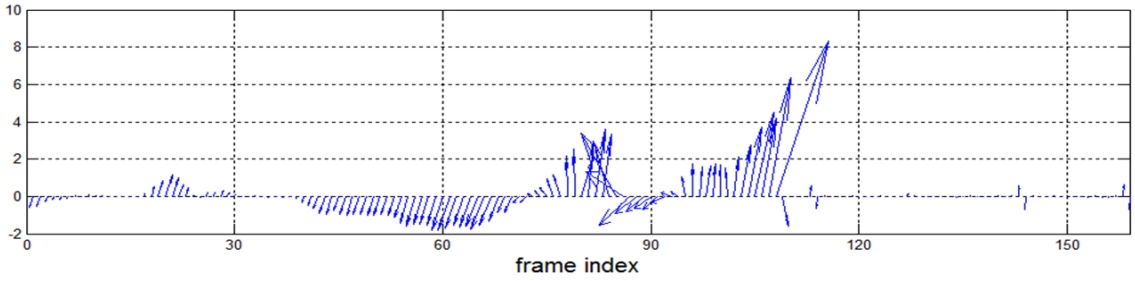


(a) Key frame of shot 155.

(b) Region of interest of shot (a).



(c) Region with max confidence.



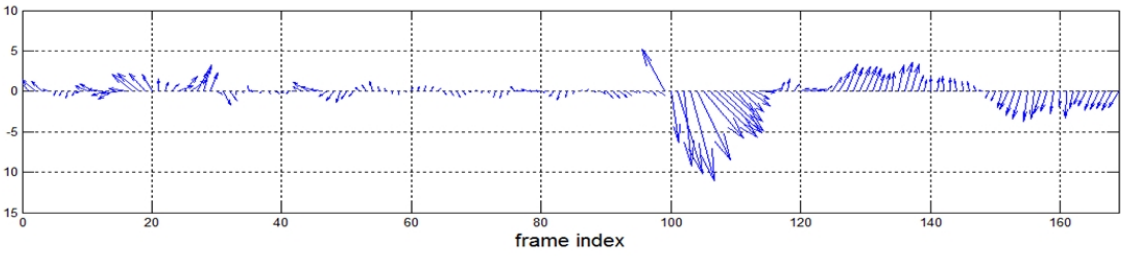
(d) Optical flow sequence.

Figure 4.15: A sample CPR scene from CPR2 video that was not detected by the HMM classifier.

(a) Key frame of shot 183.

(b) Region of interest of shot (a).

(c) Region with max confidence.



(d) Optical flow sequence.

Figure 4.16: A sample CPR scene from CPR4 video that was not detected by the HMM classifier.

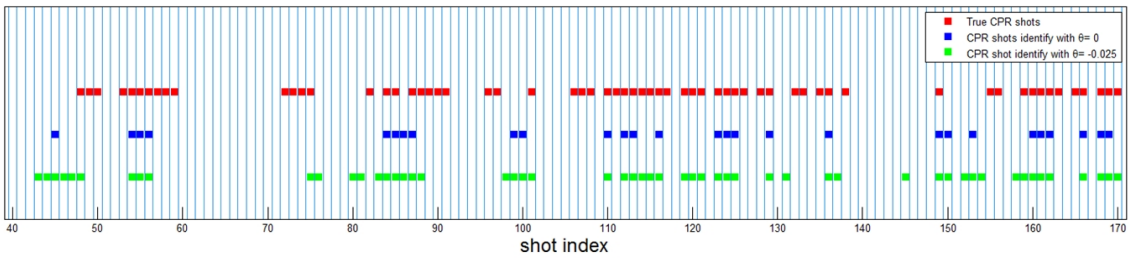


Figure 4.17: Comparison of the CPR identification when skin regions are tracked for 130 shots extracted from video CPR2.

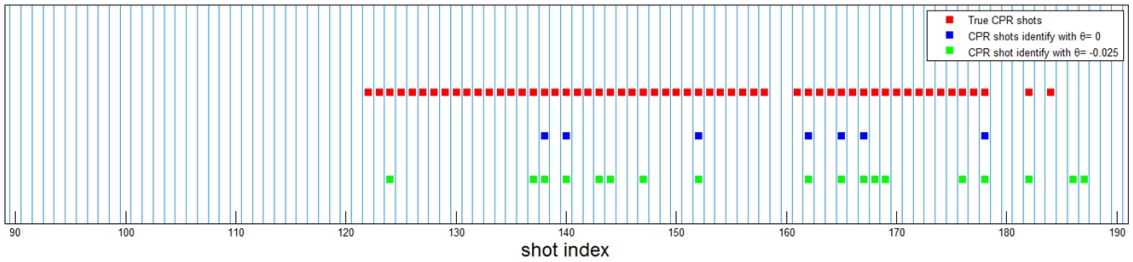


Figure 4.18: Comparison of the CPR identification when skin regions are tracked for 100 shots extracted from video CPR3.

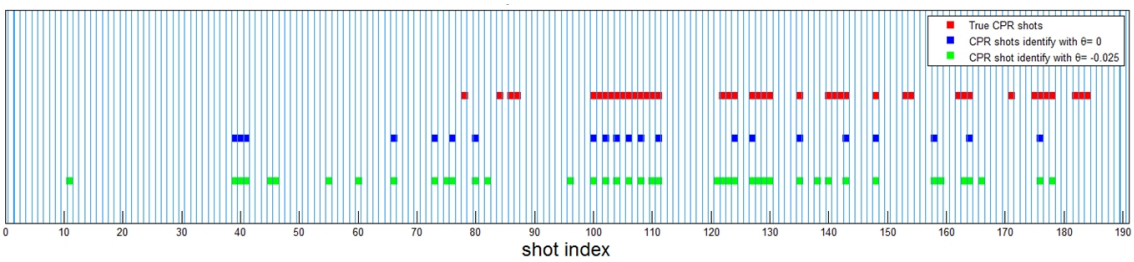


Figure 4.19: Comparison of the CPR identification when skin regions are tracked for 190 shots extracted from video CPR4.

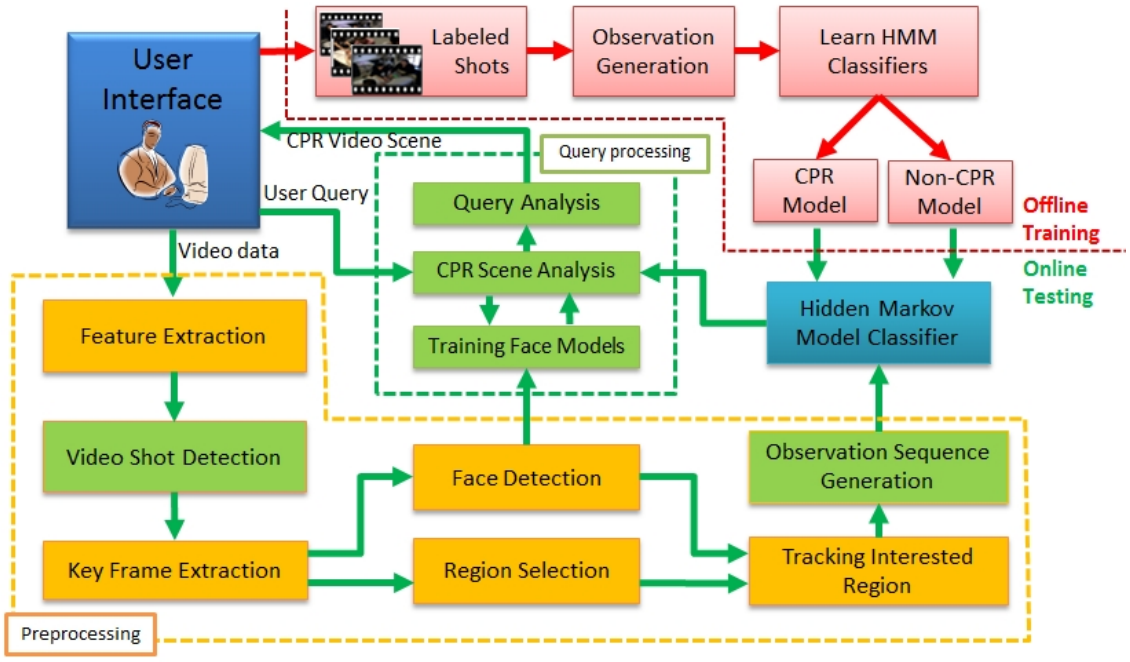


Figure 4.20: Block diagram of the proposed CPR video scene identification and retrieval prototype.

the selected video are displayed (refer to step 4 in figure 4.21) and additional options are presented (refer to step 5, 6, and 7 in figure 4.21).

The action "Feature extraction" in step 5 in figure 4.21 allows the user to extract features for each frame for the purpose of shot boundary detection. Currently, the GUI includes the features described in section 3.1.

The next option, "Show shot boundaries" is shown in step 6 in figure 4.21. This function extracts the shot boundaries and partitions the video into shots using the method described in section 3.2.2.

After a given video has been preprocessed and segmented into shots, the user can use the HMM classifier to identify all video shots that correspond to CPR action (refer to the step 7 in figure 4.21). The system will then combine all consecutive CPR shots to create a CPR scene. Once all CPR scenes are identified, the user will be presented with a new interface, as shown in figure 4.22, to analyze individual CPR scenes. In this view, the full

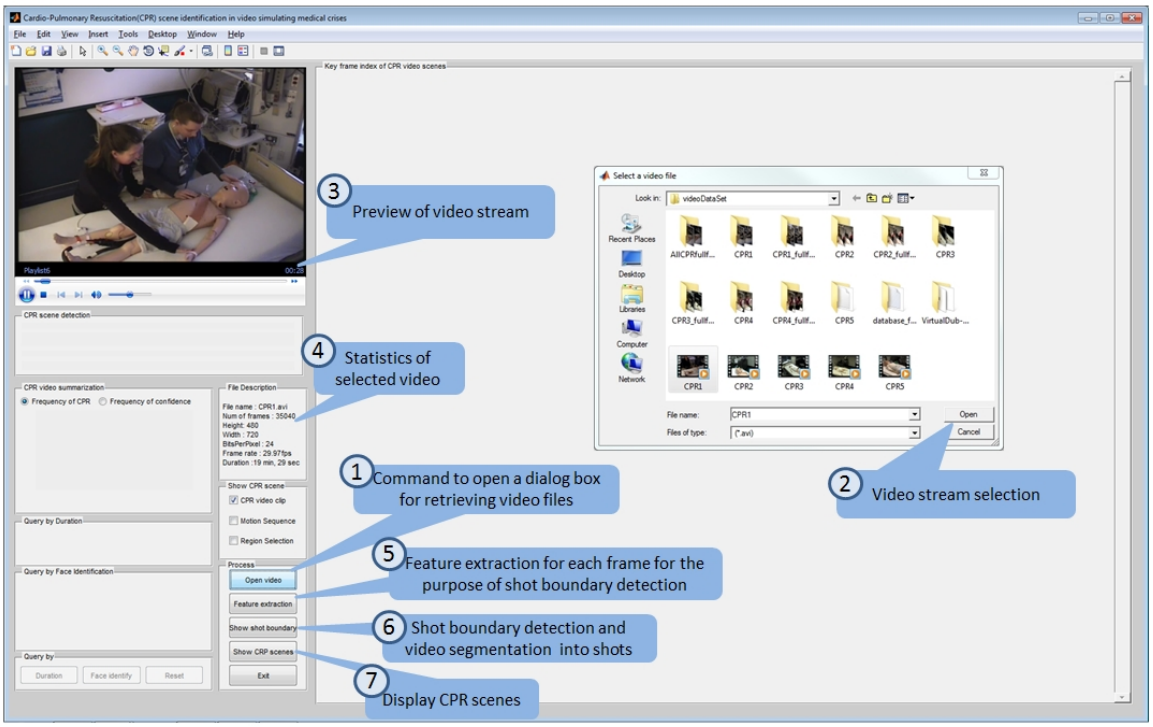


Figure 4.21: GUI of the proposed CPR video scene identification and retrieval prototype.

video can be played in the movie player (top left corner in figure 4.22). The user can select one of the CPR scenes for further analysis. This will highlight the selected CPR scene and display its statistics as show in figure 4.24. If the users wants a more detailed analysis, he can double click to open the GUI shown in figure 4.23. In this GUI, the user can examine the key frame segmentation and region selection processes. The user can also visualize the sequence of observation used by the HMM classifier.

In addition to the individual CPR scenes (shown in the left side of the GUI in figure 4.22), we also display summary statistics of the video. For instance, Part 1 of the GUI in figure 4.22 consists of 2 panels. The top panels in part 1 in figure 4.22 show the timeline of the video where the green bars indicate the temporal location of all CPR scenes. The bottom panel in part 1 in figure 4.22 highlights the average motion vector of each frame. Red bars indicate frames with the highest average motion vector and the cyan bars indicate the frames with the slowest average motion vectors.

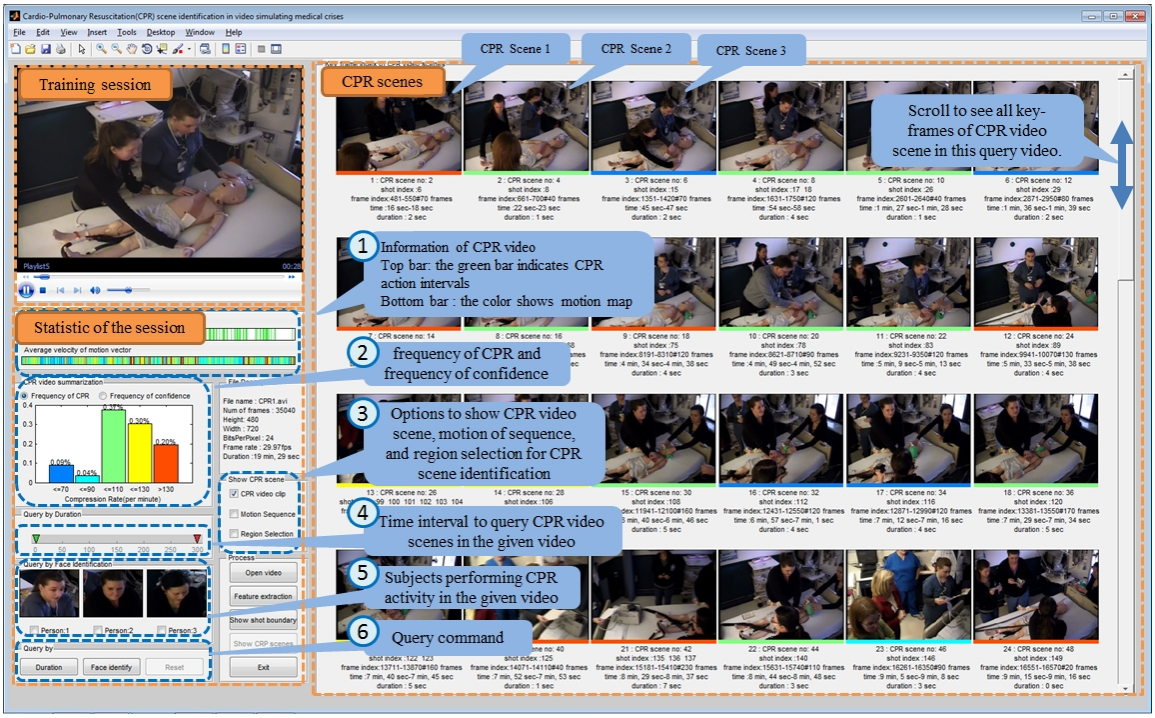


Figure 4.22: GUI for the analysis of CPR scenes.

In panel 2 of the GUI in the figure 4.22, the user can visualize a histogram of the frequency of all CPR scenes or a histogram of confidence value assigned to all CPR scenes. For the frequency option, we use the motion vectors of all observations within the sequence, identify their zero-crossings, and compute their frequency. The frequency of all scenes are quantized into 5 discrete intervals as shown in figure 4.25. Using this option the physician can identify CPR scenes that are too fast or too slow. In fact, the user can choose to visualize only CPR scenes for a given frequency range, then select individual scenes to analyze. For instance, in figure 4.26 and figure 4.27 we display 2 sample CPR scenes. The first one has high frequency while the second one has low frequency.

For the confidence histogram option in panel 2 of figure 4.22, the confidence values (assigned by the HMM classifier) to all CPR scenes is quantized into 5 levels as shown in figure 4.28. This will provide the option for the user to visualize the CPR scenes by confidence values. For instances, the CPR scene with low confidence value may not be reliable and its classification may require confirmation by the user. In figure 4.29 and figure

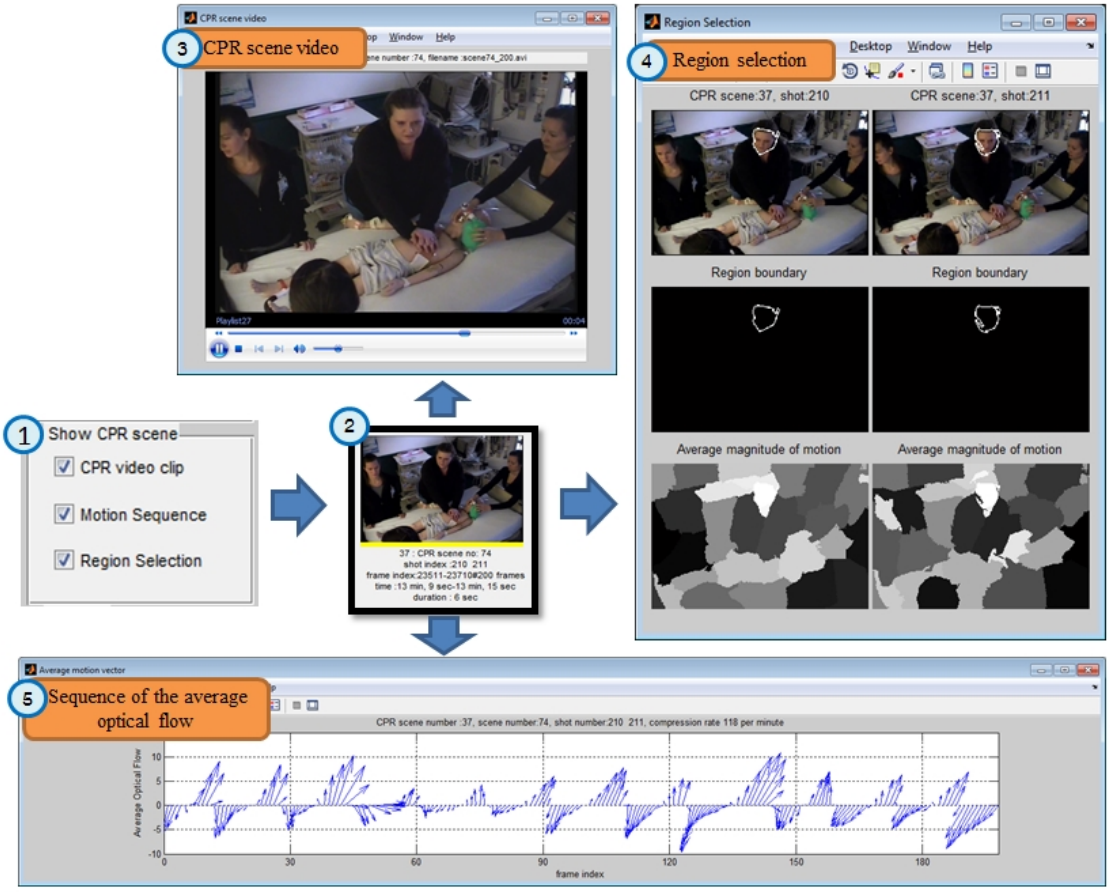


Figure 4.23: Tools to analyse a given CPR scene.

4.30, we display 2 sample CPR scenes. The first one has a high confidence value, while the second one has low confidence value. As it can be seen, the motion in the second scene is not as consistent as the first one.

Panel 4 in figure 4.22 can be used to focus on a specific time interval. The user can select the start and end time. Only CPR scenes within the specified range will be displayed.

Panel 5 of the GUI in figure 4.22 provides the user to retrieve only CPR scenes performed by a specific subject. All subjects in the video will be shown in this panel and the user can select one of them. For instance, in figure 4.31, the users selects the middle person and all CPR scenes performed by this subject and shown on the right side. The detail of query by face is highlighted in figure 4.32.

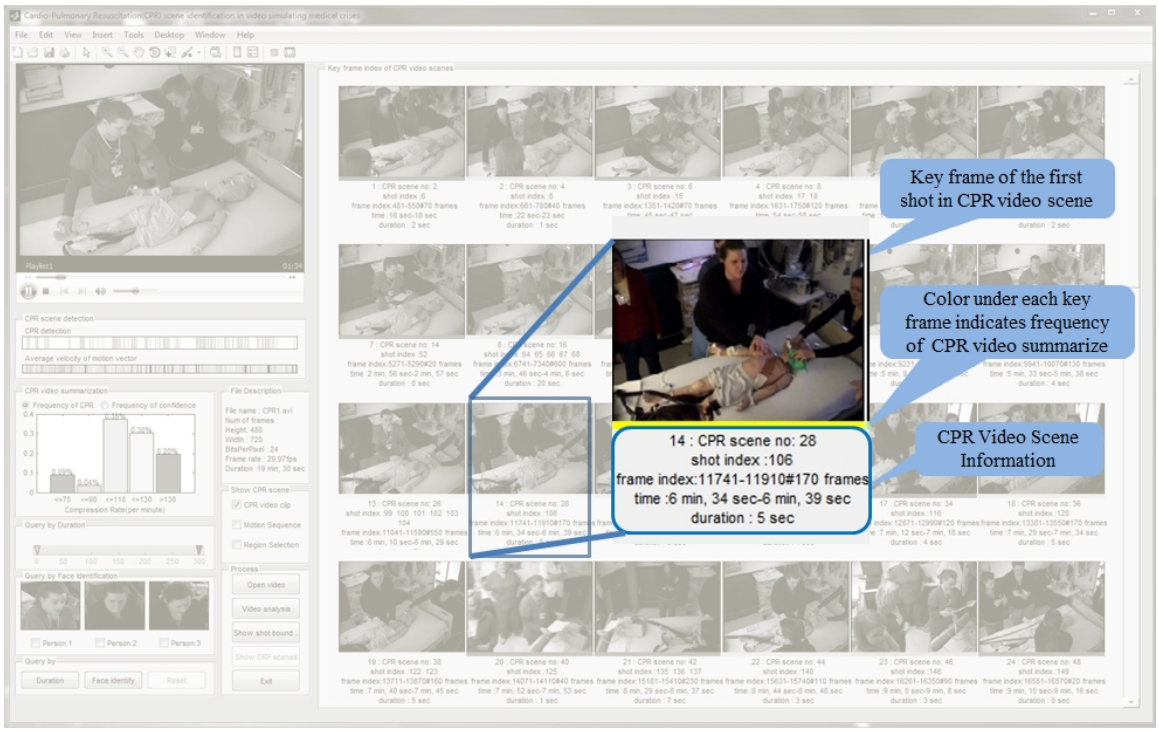


Figure 4.24: Illustration of the retrieval information in each CPR video scene.

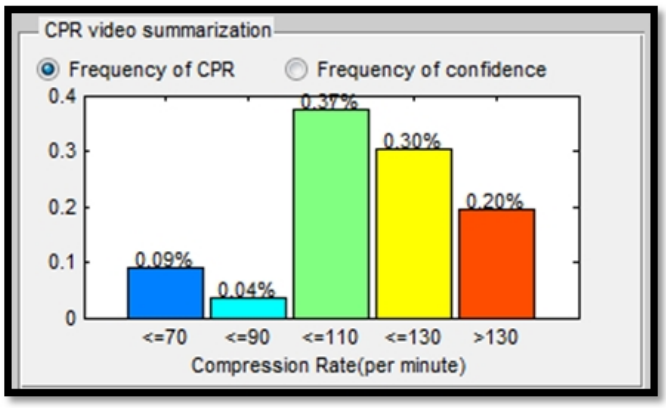


Figure 4.25: Histogram of the frequency of all CPR scenes.

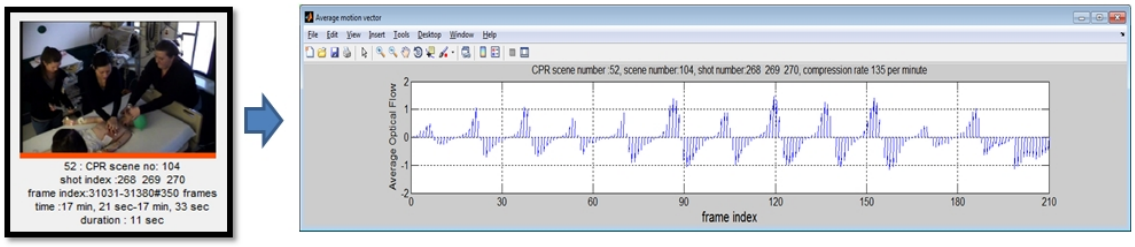


Figure 4.26: A sample CPR scene and its observation sequence that has high chest compression rate.

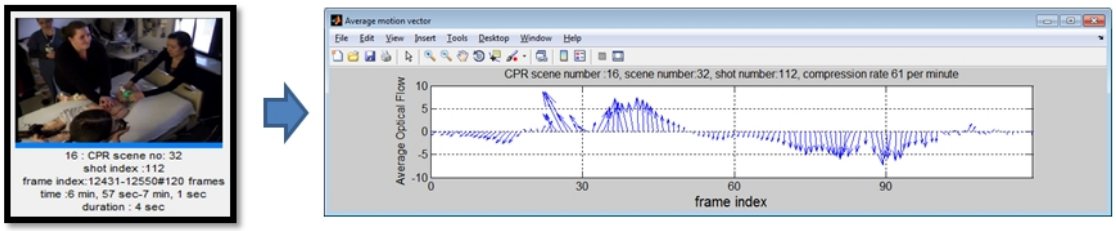


Figure 4.27: A sample CPR scene and its observation sequence that has low chest compression rate.

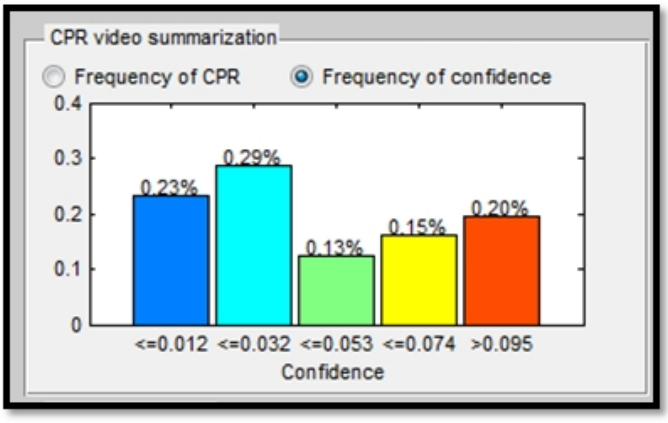


Figure 4.28: Histogram of the confidence values assigned to all CPR scenes by the HMM classifier.

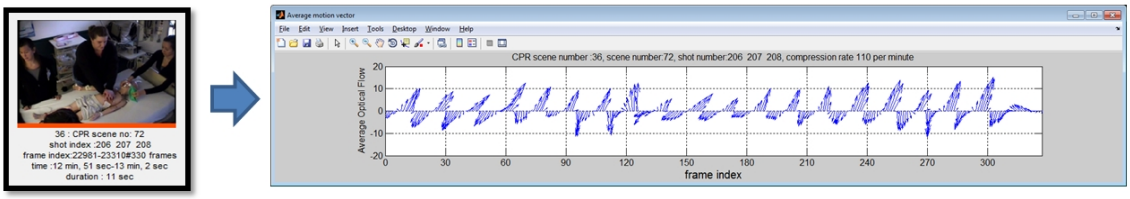


Figure 4.29: A sample CPR scene with high confidence values.

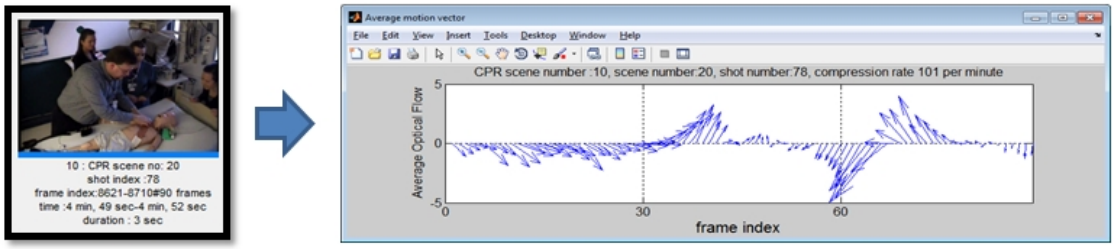


Figure 4.30: A sample CPR scene with low confidence values.

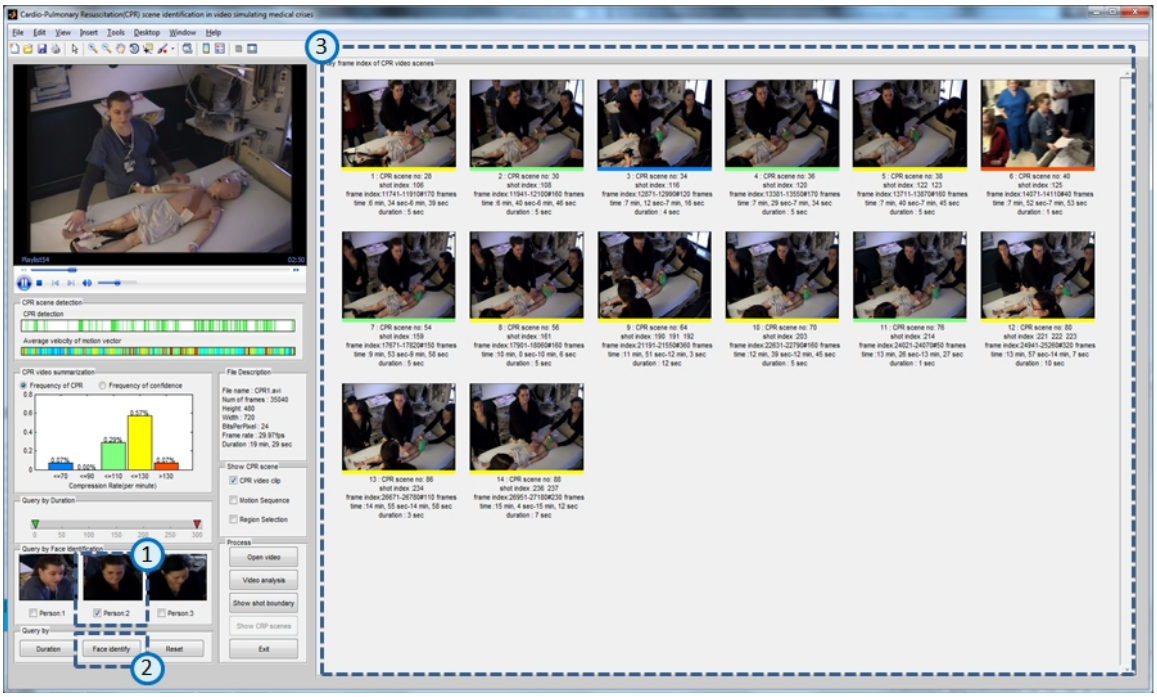


Figure 4.31: Output of query by CPR face.

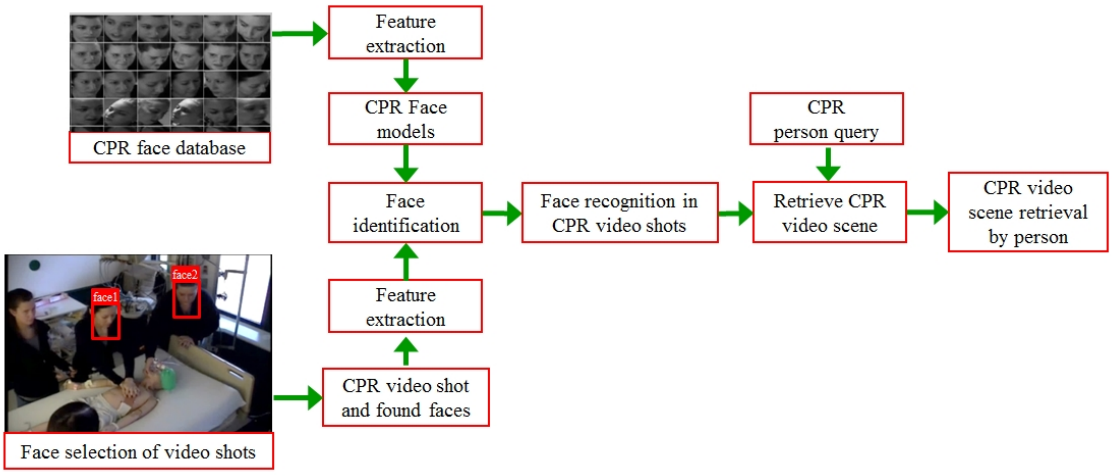


Figure 4.32: Overview of the query by a human face in our approach.

For this task, we need a face database to have a face recognition classifier. For the training data, we extract few faces for each subject from the given video. Then, from each face we extract Gabor features based on a bank of Gabor filters or kernels at different scales and orientations [90]. We extract Gabor features from each image with 5 scales and 8 orientations. This will result in a 40-dimension feature vector in each face. Then, we use the principal component analysis (PCA) [91] to reduce the dimension. Finally, a k-nearest neighbor classifier [90] is used to label each face.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

The main objective of this dissertation was to provide the physician with automated tools to segment, semantically index and retrieve specific scenes, especially in CPR scenes, from a large database of training sessions. These tools are expected to enable the physician to immediately review important sections of the training video with the team. To achieve this objective, we proposed a machine learning based approach to detect and classify scenes that involve rhythmic activities such as CPR from training video sessions simulating medical crises. This applications requires different preprocessing techniques from other video applications. In particular, to identify and track regions of interest, we integrated motion and color information and spatial and temporal constrains. The first step of our approach consists of segmenting the video into shots. This is achieved by extracting features from each frame and identifying locations where consecutive frames are different. The first feature is based on color information. In particular, the color histogram which is an efficient and robust way to describe and encode the content of the video frames. To complement the information provided by the color, we propose using motion information as an additional feature to identify shot boundaries. This additional information is extracted directly from the compressed domain from the MPEG stream. Thus, no additional computation is needed. To identify shot boundaries, we proposed two different methods. The first one is based on simple thresholds (namely SD_FD) while the second one uses unsupervised learning techniques (namely SD_UL). Our experiments have shown that the SD_UL method outperforms the SD_FD method.

The second step of our approach consists of selecting one key frame from each shot

and segmenting it into homogeneous regions. Then, few regions of interest are identified for further processing. These regions are selected based on the type of motion of their pixels and their likelihood to be skin-like regions. The regions of interest are tracked and a sequence of observations, that encode their motion throughout the shot, is extracted. Since CPR sequences can be characterized by rhythmic pattern (an upward motion followed by a downward motion), we train an HMM classifier to learn the characteristics of the CPR actions and to discriminate between regions that involve CPR and other regions. We experiment with both continuous and discrete HMM. The HMM classifiers produced a confidence value by backtracking through the model states using the Viterbi algorithm.

To improve the accuracy of our system, we also detect faces in each key frame, track them throughout the shot, and fuse their HMM confidence with the region's confidence.

We have also developed a CPR scene video retrieval and analysis prototype system with graphic user interface (GUI). This GUI prototype has several desirable features that allow the user to view and analyze the video training session much more efficiently.

To validate our proposed approach to detect CPR scenes, we used one video simulation session recorded by the SPARC group to train the HMM classifiers and learn the system's parameters. Then, we analyzed the proposed system on other video recordings. We showed that our approach can identify most CPR scenes with few false alarms.

5.2 Future Work

Although our system has much proved to be an efficient approach to index, segment, analyze, and retrieve relevant video scenes with CPR activity, there are still some issues that can improve its performance. For instance, shot boundary detection could be improved by including additional features such as audio, facial, and textual features. Similarly, segmentation of the key frames into regions can be improved by using additional features such as texture [92] and other clustering algorithms that can find the optimal number of regions [93].

Currently, our system has been trained using one training session. Testing on differ-

ent videos has been resulted in reduced accuracy. Investigation of our results has revealed that our system is sensitive to the viewing angle, frame resolution, and size of the patient (infant vs toddler). Thus, a larger training collection with diverse settings may be needed to improve the robustness of our system.

REFERENCES

[1] W. Hu, N Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," in *IEEE Transactions on Systems, Man, and Cybernetics*, 2011, vol. 41, pp. 797–819.

[2] R. Brunelli, O. Mich, and C.M. Modena, "A survey on the automatic indexing of video data," *Visual Communication and Image Representation*, vol. 10, no. 2, pp. 78–112, 1996.

[3] N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor, "Applications of video-content analysis and retrieval," *IEEE MultiMedia*, vol. 9, no. 3, pp. 42–55, 2002.

[4] Y. A. Aslandogan and C. T. Yu, "Techniques and systems for image and video retrieval," in *IEEE Trannsions on knowledge and data engineering*, 1999, vol. 11, pp. 56–63.

[5] S. Antani, R. Kasturi, and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video," vol. 35, pp. 945–956, 2002.

[6] P. Geetha and V. Narayanan, "A survey of content-based video retrieval," vol. 4, pp. 474–486, 2008.

[7] D. Brezeale and D. J. Cook, "Automatic video classification: A survey of the literature," in *IEEE Transactions on Systems, Man, and Cybernetics*, 2008, vol. 38, pp. 416–430.

[8] M. K. Geetha and S. Palanivel, "Video classification and shot detection for video retrieval applications," vol. 2, pp. 39–50, 2009.

[9] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The qbic system," *Computer*, vol. 28, no. 9, pp. 23–32, Sept. 1995.

[10] A. Hampapur, A. Gupta, B. Horowitz, C.-F. Shu, C. Fuller, J.R. Bach, M. Gorkani, and R.C. Jain, "Virage video engine," in *SPIE 3022, Storage and Retrieval for Image and Video Databases,*, 1997, vol. 188.

[11] E. Hauptmann, R. Jin, D. Ng, R. Houghton, and S. Thornton, "Video retrieval with the informedia digital video library system," in *the Tenth Text Retrieval Conference*, 2001.

[12] S. C. Chen, M.-L. Shyu, and N. Zhao, "An enhanced query model for soccer video retrieval using temporal relationships," in *International Conference on Data Engineering, ICDE*, 2005, pp. 1133–1134.

[13] B. Li and M. I. Sezan, "Event detection and summarization in american football broadcast video," in *Proc. Storage and Retrieval for Media Databases*, 2002, pp. 132–138.

[14] D. D. Saur, Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge, "Automated analysis and annotation of basketball video," in *SPIE 3022, Storage and Retrieval for Image and Video Databases*, 1997, vol. 176.

[15] T. Nishimura and R. Oka, "Indexing of baseball telecast for content-based video retireval,," in *Proc. Int. Conf. on Automatic Face and Gesture Recognition*, 1996, pp. 318 –322.

[16] J. Raiyn, "Detection of objects in motiona survey of video surveillance," vol. 4, pp. 73–78, 2013.

[17] S. A. Velastina, B. A. Boghossianb, and M.A. Vicencio-Silvac, "A motion-based image processing system for detecing potentially dangerous situations in underground railway stations," vol. 14, pp. 96113, 2006.

[18] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 844–851, 2000.

[19] J. Huang, Z. Liu, and Y. Wang, "Joint scene classification and segmentation based on hidden markov model," in *IEEE Transactions on Multimedia*, 2005, vol. 7, pp. 538–550.

[20] L. Li, N. Zhang, L.-Y. Duan, Q. Huang, J. Du, and L. Guan, "Automatic sports genre categorization and view-type classification over large-scale dataset," in *Proceedings of the 17th ACM International Conference on Multimedia*, New York, NY, USA, 2009, MM '09, pp. 653–656, ACM.

[21] S.-H. Liu, Y. Cao, Y. Li, M. Li, and S. Hu, "Semantic image classification for medical videos," in *IEEE International Conference on Semantic Computing*, 2009, pp. 648–653.

[22] Y. Cao, S-H Liu, M Li, S. Baang, and S. Hu, "Medical video event classification using shared features," in *Tenth IEEE International Symposium on Multimedia*, 2008, pp. 266 – 273.

[23] S. B. Gokturk and C. Tomasi, "A new 3-d pattern recognition technique with application to computer aided colonoscopy," in *IEEE Computer Society on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 93–100.

[24] Y. Cao, D. Li, W. Tavanapong, J. Oh, J. Wong, and P. C. de Groen, "Parsing and browsing tools for colonoscopy videos," in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, 2004, MULTIMEDIA '04, pp. 844–851.

[25] Y. Cao, D. Liu, W. Tavanapong, J Wong, J. Oh, and P.C. de Groen, "Computer-aided detection of diagnostic and theraputic operations in colonoscopy videos," in *IEEE Transactions on Biomedical Engineering*, 2007, pp. 1268–1279.

[26] Y. Cao, D. Liu, W. Tavanapong, J Wong, J. Oh, and P.C. de Groen, "Automatic classification of image with appendiceal orifice in colonoscopy videos," in *IEEE International Conference of the Engineering in Medicine and Biology Society(EMBC)*, 2006, pp. 2349 – 2352.

[27] L. Li, D. Chen, S. Lakare, K. Kreeger, I. Bitter, A. E. Kaufman, M. R. Wax, P. M. Djuric, and Z. Liang, "Image segmentation approach to extract colon lumen through colonic material tagging and hidden markov random field model for virtual colonoscopy," 2002, vol. 4683, pp. 406–411.

[28] J. Han, P. H. N. de With, A. Merien, and G. Oei, "Intelligent trainee behavior assessment system for medical training employing video analysis," *Pattern Recogn. Lett.*, vol. 33, no. 4, pp. 453–461, Mar. 2012.

[29] X. Zhu, J. Fan, M. Hacid, and A. K. Elmagarmid, "Classiminer: miniing medical video for scalable skimming and summarization," pp. 79–80, 2002.

[30] E. Mendi, S. Cecen, E. Ermisoglu, and C. Bayrak, "Automated neurosurgical video segmentation and retrieval system," vol. 3, pp. 618–624, 2010.

[31] Q. Zhang and B. Li, "Video-based motion expertise analysis in simulation-based surgical training using hierarchical dirichlet process hidden markov model," in *Proceedings of the 2011 International ACM Workshop on Medical Multimedia Analysis and Retrieval*, 2011, MMAR '11, pp. 19–24.

[32] M. D. Fabro and L. Bszrmenyi, "State-of-the-art and future challenges in video scene detection: a survey," vol. 19, pp. 427–454, 2013.

[33] J. S. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection techniques," in *Storage and Retrieval for Still Image and Video Database IV*, 1996.

[34] B. Gunsel and A.M. Tekalp, "Content-based video abstraction," in *International Conference on Image processing(ICIP)*, 1998, vol. 3, pp. 128–132.

[35] R. Lienhart, "Comparison of automatic shot boundary detection algorithms," in *Storage and Retrieval for Image and Video Database*, 1999, pp. 290–301.

[36] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *the third ACM international Multimedia'95*, 1995, pp. 189–200.

[37] A. Hauptmann, R. V. Baron, M. y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W. h. Lin, T. Ng, N. Moraveji, C. G. M. Snoek, G. Tzanetakis, J. Yang, R. Yan, and H. D. Wactlar, "Informedia at trecvid 2003: Analyzing and searching broadcast news video," in *In Proc. of TRECVID*, 2003.

[38] A. Divakaran, R. Radhakrishnan, and K.A. Peker, "Motion activity-based extraction of key-frames from video shots," in *IEEE International Conference Image processing*, 2002, vol. 1, pp. 932–935.

[39] C. Wu, Y. He, L. Zhao, and Y. Zhong, "Motion feature extraction scheme for content-based video retrieval," in *Storage and Retrieval for Media Databases (SPIE)*, 2002, vol. 296.

[40] U. Gargi, R. Kasturi, and S.H. Strayer, "Performance characterization of video-shot-change detection methods," in *IEEE Transactions on circuits and systems for video technology*, 2000, vol. 10, pp. 1–13.

[41] M. Cooper, J. Foote, J. Adcock, and S. Casi, "Shot boundary detection via similarity analysis," in *in Proceedings of the TRECVID 2003 Workshop*, 2003, pp. 79–84.

[42] Z.-C. Zhao and A.-N. Cai, "Shot boundary detection algorithm in compressed domain based on adaboost and fuzzy theory," in *Proceedings of the Second International Conference on Advances in Natural Computation - Volume Part II*, Berlin, Heidelberg, 2006, pp. 617–626, Springer-Verlag.

[43] B.-L. Yeo and B. Liu, "Rapid scene analysis on compressed video," in *IEEE Transaction on Circuits System Video Technology*, 2002, vol. 5, pp. 533–544.

[44] K. Tse, J. Wei, and S. Panchanathan, "A scene change detection algorithm for mpeg compressed video sequences," in *Canadian Conference on Electrical and Computer Engineering*, 1995, vol. 2, pp. 827–830.

[45] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of trecvid activity," *Comput. Vis. Image Underst.*, vol. 114, no. 4, pp. 411–418, Apr. 2010.

[46] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, "A formal study of shot boundary detection," in *IEEE Transaction on Circuit and Systems For Video Technology*, 2007, pp. 168–186.

[47] M. Cooper, T. Liu, and E. Rieffel, "Video segmentation via temporal pattern classification," in *IEEE Transactions on Multimedia,*, 2007, vol. 9, pp. 610–618.

[48] J.S. Boreczky and L.D. Wilcox, "A hidden markov model framework for video segmentation using audio and image features," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998, vol. 6, pp. 3741–3744.

[49] S. Chang H., S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieva," in *IEEE Transactions on Circuits and Systems for Video Technology*, 1999, vol. 9, pp. 1269 – 1279.

[50] K.-W. Sze, K.-M. Lam, and G. Qiu, "A new key frame representation for video segment retrieval," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 15, pp. 1148–1155, Sept. 2005.

[51] N. Goela, K. Wilson, F. Niu, A. Divakaran, and I. Otsuka, "An svm framework for genre-independent scene change detection," in *IEEE International Conference on Multimedia and Expo*, 2007, pp. 532 – 535.

[52] Z. Rasheed and M. Shah, "Detection and representation of scenes in videos," *Trans. Multi.*, vol. 7, no. 6, pp. 1097–1105, 2005.

[53] C.-w. Ngo, Y.-f. Ma, and H.-j. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol*, vol. 15, pp. 296–305, 2005.

[54] S. Benini, L. Q. Xu, and R. Leonardi, "Identifying video content consistency by vector quantization," in *International Workshop on Image Analysis for Multimedia Interactive Service*, 2005.

[55] T. Lin, H.-J. Zhang, and Q.-y. Shi, "Video scene extraction by force competition," in *ICME 2001. IEEE International Conference on Multimedia and Expo*, 2001, pp. 753 – 756.

[56] D. Gatica-Perez, A. Loui, and M.-t. Sun, "Finding structure in home videos by probabilistic hierarchical clustering," in *IEEE Transactions on Circuits and Systems for Video Technology,*, 2003, vol. 13, pp. 539 – 548.

[57] M. Rautiainen and D. Doermann, "Temporal color correlograms for video retrieval,," in *16th International Conference on Pattern Recognition*, 2002, vol. 1, pp. 267–270.

[58] S. Zhuang, H. Yan, K. Palaniappan, and Z. Yunxin, "Gaussian mixture density modeling, decomposition, and applications," in *IEEE Transaction on Image Processing*, 1996, vol. 5, pp. 129–1302.

[59] J. Sivic, M. Everingham, and A. Zisserman, "Person spotting: Video shot retrieval for face sets," in *Proceedings of the 4th International Conference on Image and Video Retrieval*, Berlin, Heidelberg, 2005, CIVR'05, pp. 226–236, Springer-Verlag.

[60] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal Of Computer Vision*, vol. 12, pp. 43–77, 1994.

[61] R. Venkatesh Babu and K.R. Ramakrishnan, "Content-based video retrieval using motion descriptors extracted from compressed domain," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2002, vol. 4, pp. 141–144.

[62] B.K. Horn and B.G. Schunck, "Determining optical flow," vol. 17, pp. 185–203, 1981.

[63] S. Fischer, R. Lienhart, and W. Effelsberg, "Automatic recognition of film genres," in *Proceedings of the Third ACM International Conference on Multimedia*, New York, NY, USA, 1995, Multimedia '95, pp. 295–304, ACM.

[64] S. Moncrieff, S. Venkatesh, and C. Dorai, "Horro film genre typing and scene labeling via audio analysis," in *International Conference on Multimedia and Expo(ICME)*, 2003, vol. 2, pp. 193–196.

[65] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings Of The IEEE*, 1989, pp. 257–286.

[66] M. J. Roach, J.S.D. Mason, M. Pawlewski, M. Heath, and I. I. Re, "Video genre classification using dynamics," in *IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP)*, 2001, vol. 3, pp. 1557–1560.

[67] M. Roach, J. S. Mason, and M. Pawlewski, "Motion-based classification of cartoons," in *International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2001, pp. 146–149.

[68] C. Lu, J. Au, and M. S. Drew, "Classification of summarized videos using hidden markov models on compressed chromaticity signatures," in *Proceedings of the ninth ACM international conference on Multimedia*, 2001, pp. 479–482.

[69] L. Xie, P. Xu, S.-f. Chang, A. Divakaran, and B. S. Sun, "Structure analysis of soccer video with hidden markov models," in *Pattern Recognition Letters*, 2002, pp. 767–775.

[70] U. Gargi, R. Kasturi, and S.H. Strayer, "Performance characterization of video-shot-change detection methods," vol. 10, pp. 1–13, 2000.

[71] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, Upper Saddle, New Jersey, 1989.

[72] B. Martin, M. Robine, P. Hanna, and et al., "Musical structure retrieval by aligning self-similarity matrices," in *ISMIR*, 2009, pp. 483–488.

[73] S. Jun and E. Hwang, "Music segmentation and summarization based on self-similarity matrix," in *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, New York, NY, USA, 2013, ICUIMC '13, pp. 82:1–82:4, ACM.

[74] J. Serr, E. Gmez, P. Herrera, and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Trans. on Audio, Speech, and Language Processing*, 2008.

[75] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers and Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.

[76] Y. Taniguchi, A. Akutsu, Y. Tonomura, and H. Hamada, "An intuitive and efficient access interface to real-time incoming video based on automatic indexing," in *Proceedings of the Third ACM International Conference on Multimedia*, New York, NY, USA, 1995, MULTIMEDIA '95, pp. 25–33, ACM.

[77] B. Shahraray and D. C. Gibbon, "Automatic generation of pictorial transcripts of video programs," in *Multimedia Computing and Networking (SPIE)*, 1995, vol. 2417.

[78] S. W. Smoliar and H. Zhang, "Content-based video indexing and retrieval," *IEEE MultiMedia*, vol. 1, no. 2, pp. 62–72, 1994.

[79] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[80] J. Shi, *MATLAB normalized cuts segmentation code.*

[81] C.O. Conaire, N.E. O'Connor, and A.F. Smeaton, "Detector adaptation by maximising agreement between independent data sources," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2007, pp. 1–6.

[82] S. L. Phung, A. Bouzerdoum, and D. Chai, "A novel skin color model in ycbcr color space and its application to human face detection," in *International Conference on Image Processing*, 2002, vol. 1, pp. 289–292.

[83] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001*. IEEE, 2001, vol. 1, pp. I–511.

[84] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, pp. 1612, 1999.

[85] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, 2006.

[86] G. D. Forney, "The viterbi algorithm," in *Proceeding of the IEEE*, 1973, vol. 61, pp. 268–278.

[87] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[88] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," in *Proceedings of the IEEE*, 1985, vol. 73, pp. 1551–1588.

[89] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 12 1966.

[90] V. Štruc and N. Pavešić, "The complete gabor-fisher classifier for robust face recognition," *EURASIP J. Adv. Signal Process*, vol. 2010, pp. 31:1–31:13, Feb. 2010.

[91] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[92] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *Journal of the Optical Society of America*, vol. 7, pp. 923–932, 1990.

[93] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

CURRICULUM VITAE

NAME: Surangkana Rawungyot

ADDRESS: Department of Computer Engineering and Computer Science
Speed School of Engineering
University of Louisville
Louisville, KY 40292

EDUCATION:

Ph.D., Computer Science and Engineering
December 2014
University of Louisville, *Louisville, Kentucky*

M.S., Information Technology
April 2002
King Mongkut′s Institute of Technology Ladkrabang,
Bangkok, Thailand

B.S., Computer Science
March 1998
Chiang Mai University, *Chiangmai, Thailand*

CONFERENCE PUBLICATIONS:

1. H. Frigui, **S. Rawungyot**, and A. Hamdi, *"Identification of Cardio Pulmonary Resuscitation (CPR) Scenes in Video Simulating Medical Crises"*, International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, March 2014.

HONORS AND AWARDS:

1. Doctoral Dissertation Completion Award, Summer 2014.

2. Thai Royal Government Scholarship for studying in Doctoral in computer science and engineering, August 2009.

3. Naresuan University Scholarship for studying in Master of Science degree, June 2001.