UNIVERSITY OF CALIFORNIA

SANTA BARBARA

Motion Activity for Video Indexing

A dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

In

Electrical and Computer Engineering

by

Xinding Sun

Committee in charge:

Professor B.S. Manjunath, Chair

Professor Yuan-Fang Wang

Professor Shivkumar Chandrasekaran

Dr. Jonathan Foote

June 2004

The dissertation of Xinding Sun is approved

B. S. Manjunath

Yuan-Fang Wang

Shivkumar Chandrasekaran

Jonathan Foote

February 2004

Motion Activity for Video Indexing

By

Xinding Sun

Dedicated to my family

Acknowledgement

I am deeply indebted to my research advisor and committee chair Prof. B. S. Manjunath. It is his constant support and encouragement that make it possible for me to finish my dissertation. He is not only my research advisor but also a great mentor in my life. Working and interacting with him was truly an invaluable learning experience.

My sincere thanks go to Dr. Jonathan Foote. Part of my dissertation work was inspired by my interaction with Dr. Foote during my summer internship at the FX Palo Alto Lab. I learned many research skills by working with him. I would like to thank Professor Yuan-Fang Wang and Professor Shivkumar Chandrasekaran for serving on my committee and for constructive discussions about my dissertation work.

I would like to thank the graduate students in Image Processing and Vision Research Lab, Sitaram Bhagavathy, John A. Berger, Zhiqiang Bi, Jong-jin Chae, Ching-Wei Chen, Kapil Chhabra, Onkar Dabeer, Yining Deng, Motaz EI-Saban, Myléne Farias, Daryl Fortney, Gabriel Gomez, Serkan Hatipoglu, David E. Lane, Mike Moore, Debargha Mukherjee, Shawn Newsam, Christian Schmidt, Kaushal Solanki, Ken Sullivan, Baris Sumengen, Jelena Tesic, Lei Wang, Jay Winkeler, Peng Wu, Marco Zuliani. Working with them makes my life joyful.

Finally, I would thank my wife Liru for her love and support.

Vita

Education

1991, Bachelor of Engineering, in Department of Automation, Nantong Institute of Technology,

1994. Master of Science, in Department of Automation, Tsinghua University

2004, Doctor of Philosophy in Department of Electrical and Computer Engineering, University of California, Santa Barbara.

Professional Experience

1995-1997 Engineer, Department of Automation, Tsinghua University

1997-1998 Research scholar, Kent Ridge Digital Labs.

2000 Intern, Fuji-Xerox Palo Alto Lab

2001 Intern, Fuji-Xerox Palo Alto Lab.

2003 Visiting student, Microsoft Research Asia.

1998-2004 Research assistant, Department of Electrical and Computer Engineering, University of California.

Publication

1. M. A. El Saban, X Sun, B. S. Manjunath and K. Kamath, " Multiscale Edge-based Tracking of Microtubles," submitted to ICIP.
2. Xinding Sun, Jonathan Foote, Don Kimber, B. S. Manjunath, "Extraction of Region of Interest and Virtual Camera Control Based on Panoramic Capturing," accepted, IEEE Transactions on Multimedia.
3. Xinding Sun, Don Kimber, Jonathan Foote, B. S. Manjunath, "Detecting Path Intersections in Panoramic Video," ICME, 2002.
4. Xinding Sun, B. S. Manjunath, "Panoramic Capturing and Recognition of Human activity," ICIP,2002.
5. Xinding Sun, B. S. Manjunath, and Divakaran Ajay, "Representation of motion activity in hierarchical levels for video indexing and filtering," ICIP,2002.
6. Xinding Sun, Ching-Wei Chen, B. S. Manjunath, "Probabilistic Motion Parameter Models for Human Activity Recognition," ICPR,2002.

7. Xinding Sun, Jonathan Foote, Don Kimber, B. S. Manjunath, “Panoramic Video Capturing and Compressed Domain Virtual Camera Control,” Proc. ACM Multimedia, pp. 329-338, 2001.
8. Xinding Sun, Jonathan Foote, Don Kimber, B. S. Manjunath, “Recording the Region of Interest from FlyCam Panoramic Video,” Proc. ICIP, 2001.
9. Xinding Sun, Ajay, Divakaran, B.S. Manjunath, “A Motion Activity Descriptor and Its Extraction in Compressed Domain,” Proc. Pacific-Rim Multimedia, 2001.
10. Xinding Sun, Mohan. S. Kankanhalli, “Video Summarization Using R-Sequences” Journal of Real Time Imaging ,6, pp: 449-459, 2000
11. A. Divakaran, H. Sun, H. Kim, C.S. Park, B.S. Manjunath, X. Sun, H. Shin, V.V. Vinod, et al., “Report on the MPEG-7 CE on the Motion Activity Feature,” *ISO/IEC MPEG99/m5030*, 2000
12. Xinding Sun, B. S. Manjunath, Peng Wu, Yining Deng, “Motion Quantized Alpha Histogram as a Video Unit Descriptor, ISO/IEC JTC1/SC29/WG11/P75,” MPEG7 group,1999.
13. Xinding Sun, Mohan S. Kankanhalli, Yongwei Zhu, Jiankang Wu, “Content-Based Representative Frame Extraction For Digital Video,” Proc. IEEE Multimedia Computing and Systems'98 pp:190-193, 1998.
14. Xinding Sun, et al. “Research on Spatial Non-Homogenous Kalman Filter and Its Application in Cephalometric Image,” Proc. IEEE ICSP' 96, pp: 1078-1081, 1996.
15. Xinding Sun, Yan Wang, Zhenmin Xu, Changsheng Xu, “Adaptive Kalman Filtering Approach of Color Noise in Cephalometeric Image,” Proc. IEEE ICSP' 96, pp:622-625,1996.
16. Changsheng Xu, Zhengming Xu, Zhaoying Zhou, Xinding Sun,”Research on Image Processing Using Bidimensional Kalman Filtering Approach,” Proc. IEEE IMTC'96,pp: 119-121,1996.

Patent Pending

1. Methods for Intersection Detection in Panoramic Video, with J. Foote, D. Kimber, and J. Adcock. Application no: 20040004659.
2. Modeling and Recognition of Human Activity , with B. S. Manjunath, C.-W. Chen, et al. Application no: 20020085092.
3. Motion Activity Descriptor , with A. Divakaran, B. S. Manjunath, et al. Application no: 20040022317, 20040022317, 20030026340.

Field of Study

Major fields: Image and Video Processing, Multimedia System Indexing.

Abstract

Motion Activity for Video Indexing

By
Xinding Sun

Video indexing based on motion is an emerging research area. While most previous work focused on video indexing using motion vectors, a detailed quantitative characterization of the spatial and temporal change of motion vectors in a video has not received much attention. We characterize motion in terms of motion activity and propose novel methods for motion activity description. In the particular context of human motion activity analysis, we propose new algorithms for motion activity capture and recognition.

Two new motion activity descriptors are introduced for low level video indexing. The first one, the motion intensity descriptor, represents the degree of change in motion in a scene. The second descriptor, the motion intensity histogram, represents the temporal statistics of motion intensity. The motion activity information is extracted in compressed domain based on MPEG macroblock type information. We present a system for capturing panoramic video of human motion activity and a novel method for virtual camera control. The proposed method integrates region of interest (ROI) detection, tracking, and virtual camera control, and works on both uncompressed and compressed video streams. Finally, we present a unified approach for human motion activity recognition. The panoramic camera capturing system is used for video capture. The virtual camera control parameters are used for the recognition of activities such as

walking, and the motion parameters of each frame are used for the recognition of other activities like turning around, sitting down and getting up. For motion parameter based recognition, the likelihood of the motion parameters is represented using a multivariate Gaussian model and their temporal change is characterized using a continuous density hidden Markov model (HMM). Detailed experimental results are provided to demonstrate the efficiency and effectiveness of the proposed descriptors and motion based activity recognition.

In summary, the research presented in this dissertation advances the current state of the art in video indexing by proposing new methods for characterizing motion activity at the low level, using motion intensity and motion intensity histogram, and at the semantic level for annotating some of the common human motion activities.

Table of Contents

List of Figures

List of Tables

1 Introduction

Digital video is playing an increasingly important role in our daily life. In recent years, the development of software and hardware technology has enabled the creation of a large amount of digital video content. Due to the rapid increase in the size of digital video databases, users now have access to a very broad selection of video content, and thus they require more flexible as well as powerful video indexing tools. Therefore, development of advanced video indexing tools is a very important area of research on video applications.

Motion is a salient feature in video, in addition to other typical image features such as color, texture and shape. Motion represents two dimensional temporal change of video content. Motion information is used in many applications such as motion based segmentation, and structure from motion. In this dissertation, we focus on indexing video using motion.

In a sports video such as the American football, the game starts from very little or no motion at the start of each play to high motion with the progress of the play. In this case, we want to characterize the low level motion into different "intensity levels" or motion intensity for video indexing. The temporal distribution of motion intensity can also be used. The motion intensity and motion intensity histogram form the low level motion activity feature and is the focus of the first part of our research. Specifically, we

investigate the use of motion intensity and motion intensity histogram for video search and browsing applications.

Another typical scenario that this dissertation is concerned with is that of a speaker giving a lecture in a classroom/seminar or teleconference. The speaker may move around, and turn his body. We are interested in capturing and annotating the motion activity of this speaker. We would like to capture the region of interest that includes the speaker, and refer to this as motion-based human activity capture. Then, based on the captured video, we would like to do automatic recognition of the speaker activity. This motion-based human activity capture and recognition is the focus of the second part of our research.

1.1 Related Work

Since the early nineties, the problem of video data management has received considerable attention. Some examples of video management systems that have been developed include: CueVideo from IBM [108], VideoLogger from Virage [114], Video Manga [94] from FXPAL, Netra-V from UCSB , VideoSeek from Columbia [110], and Stars from UIUC [75], among others. A milestone of this so-called "content-based research" is the standardization of the video feature descriptors and description schemes in MPEG-7 [112]. A typical demonstration of these systems allows one to search the system for "objects that have similar colors to a given picture of a rose." In that case, most systems would produce results that would include red flowers and red cars. Generally, content-based systems can deal with simple applications where the

query is related to low level features such as color, texture, shape, and motion. While audio features in digital video are equally important, this dissertation will focus on the use of visual features alone. Even though in some cases we can derive some semantics directly from these low level features, the derivation is not so straightforward in most cases. The power of these content-based systems is thus quite limited. For example, one cannot expect to query for events such as start of the play in a sports game, or a person walking in a street.

Generally, we can divide video processing into two stages: pre-stage recording, and post content analysis. In the pre-stage, our main concern is to find the regions of interest. For example, given a panoramic input, we want to find the focus of attention (*FOA*) [80], or we simply want to track the region of interest (*ROI*). In post processing, both low-level and high-level video content analyses are required for a successful video indexing and summarization system. While the focus of our low-level video content analysis is on motion activity analysis, the high-level video content analysis is focused on human activity capture and recognition.

While motion vectors have been used for video coding or indexing in previous research, a detailed quantitative characterization of the spatial and temporal change of motion vectors in a video has not received much attention. Information about how many regions have changed in a given frame and how the changes are distributed within a period of time can be very useful for video indexing. For example, it can facilitate search for a segment of sports video with a typical motion activity in mind. By introducing the concepts of motion intensity – the degree of scene motion change, and

motion intensity histogram –the temporal distribution of motion intensity, we provide the user a description of video in terms of low level motion activity.

For human motion activity capture, there are some commercial as well as research systems that attempt to provide a solution for capturing events. Sony's EVI-D30 camera [113] can be used to track moving objects, but it is often not robust. Campbell and Bobick [19] use tokens to track object for event analysis. Token detection is more robust, but it also suffers from the same drawback. Systems that stitch multi-camera video sequences have been designed to capture events. The advantage of this kind of panoramic systems is that the speaker never leaves the camera's field of view. Chen and Williams [22] and many others [89] have developed systems that compose existing still images into a panorama that can be dynamically viewed. They are computationally expensive and thus limit their application to real-time video. While there are many other panoramic camera systems [67], [68], [88], we choose FlyCam [37] since it is fast and targets low cost and general-purpose hardware for video capture. For human motion activity capture, previous efforts on tracking date back to the early 1980s. An example work on personal tracking is by O'Rourke and Badler [73] on 2D kinematic modeling. In a more recent work, Darrell et al. [29] integrate stereo, color, and face detection with person tracking. A more detailed review of the related work on camera control and panoramic capturing can be found in chapter 4, where we present a simple model based on region of centroids to capture the region of interest.

Motion-based recognition has been well studied in the literature for recognition of motion activities. A review of some of the early work in this field can be found in [20],

[91]. Typically, feature points [19], [59], [76], [99], region features [9], [14], and global motion fields [30], [55], [69], are used in characterizing the motion information. Many of the above mentioned approaches use the Hidden Markov Models[72] for temporal characterization. While in most previous work capture and recognition of human motion activity are discussed separately; there is very little research that has tried to combine the two together. In chapter 5, we provide a detailed review of related work on human activity recognition, and present a unified approach to capturing and recognition of some human activities.

1.2 Objectives

The objective of this dissertation is to develop tools based on motion activity for video indexing. We aim to provide tools for low level motion activity description, and capture and recognition of human motion activity.

- Low level motion activity descriptors

For low level motion activity description, we propose descriptors for video scene indexing. They are (i) motion intensity that represents the degree of change in motion in a scene, and (ii) motion intensity histogram that represents the temporal statistics of motion intensity.

- Design a general structure for human motion activity capture

For capture of human motion activity, we propose a general architecture for human speaker video capture. A panoramic video capture system is used to cover the scene where the speaker moves. The objective is to produce a smooth ROI video which follows the speaker.

- Propose methods for human motion activity recognition

For recognition of human motion activity, we propose a method that unifies the human motion activity capture and recognition process. The objective of the work is to combine virtual camera control parameters and motion parameters for human motion activity recognition.

1.3 Approach

Our approach to motion intensity and motion intensity histogram extraction is based on compressed domain motion information. To extract motion activity information, an MPEG (MPEG-1/2) video is first adaptively segmented into levels with fixed percentage of original video length based on P-frame macroblock motion information. The motion intensity and motion intensity histogram are then used for feature clustering and video retrieval.

Our approach to video capture of human motion activity is based on the FlyCam panoramic video system that is designed to produce high resolution and wide-angle video sequences by stitching the video pictures from multiple stationary cameras. The proposed approach integrates region of interest detection, tracking, and virtual camera

control, and works on both uncompressed and compressed domains. It first detects the ROI of the whole video stream. The ROI is tracked using a Kalman filter. The Kalman filter estimation results are used for virtual camera control that simulates human controlled video recording.

Our approach to human motion activity recognition unifies virtual camera control and motion-based human motion activity recognition. Given a ROI sequence, the virtual camera control parameters are used for the recognition of activities such as walking, and the motion parameters of each frame are used for the motion-based recognition of other activities such as turning around, sitting down and getting up.

1.4 Summary of Contributions

The main contributions of this dissertation are as follows:

- Two new motion activity feature descriptors are introduced. The proposed descriptors are *motion intensity* and *motion intensity histogram*. The motion activity descriptors are computed in compressed domain and therefore the extraction process is efficient. Experimental results demonstrate their effectiveness.

- Design of a system for capturing of human motion activity: The proposed approach is based on the FlyCam panoramic video system. The proposed approach integrates region of interest detection, tracking, and virtual camera control, and works on both uncompressed and compressed videos. The system has no physical camera motion

and the virtual camera parameters are readily available for video indexing. Experimental results show that the system is fast and reliable.

- Design of a unified approach for human motion activity recognition: The panoramic camera capturing system is used for video capture. Virtual camera control outputs the region of interest video that covers the speaker. Given a ROI sequence, the virtual camera control parameters are used for the recognition of activities such as walking, and the motion parameters of each frame are used for the recognition of other activities such as turning around, sitting down and getting up. For motion parameter based recognition, the likelihood of the motion parameters is represented using a multivariate Gaussian model, and the change of the likelihood is characterized using a continuous density HMM. Experimental results prove the system works effectively.

1.5 Dissertation Outline

The rest of the dissertation is organized as follows:

In chapter 2, we provide background review material for the rest of the dissertation. First, we introduce motion estimation techniques. Second, we introduce motion compensation for MPEG video coding. They are the basis for motion activity analysis.

In chapter 3, we propose two new motion activity features. Motion intensity represents the degree of change in motion in a scene, and motion intensity histogram represents the temporal statistics of motion intensity. First, we introduce how to extract motion

intensity in compressed domain. Second, we introduce video segmentation based on motion intensity. Then, we introduce motion intensity histogram for video segments. Last, we introduce the applications of the two descriptors for video retrieval and indexing.

In chapter 4, we present a system for capturing panoramic video and a novel method for virtual camera control. First, we introduce the FlyCam panoramic video system that is designed to produce high resolution and wide-angle video sequences by stitching the video pictures from multiple stationary cameras. Second, we discuss ROI detection in both uncompressed domain and compressed domain. Third, we introduce Kalman tracking of ROI and virtual camera control. Experimental results are provided.

In chapter 5, we present a unified approach to human motion activity capturing and recognition. First, we introduce the general system architecture of the system. Second, we introduce motion activity recognition based virtual camera control parameters. Third, we discuss representation of motion parameters using a Gaussian model. Fourth, the temporal change of the motion parameters is characterized using a continuous density Hidden Markov Model. Experimental results are shown.

In chapter 6, we provide conclusions and future research directions.

2 Motion Estimation and Motion Compensation for MPEG Video Coding

Motion reflects the temporal change of video content. Estimation of motion has many important applications in the areas of computer vision and video processing. One direct application of motion estimation technique is motion compensation based video coding. Since motion activity analysis is based on motion features, we discuss motion estimation in this chapter. A brief discussion of motion compensation is also provided as it is crucial in compressed domain video analysis.

2.1 Motion Estimation

Many efforts have been made in the past two decades on motion estimation, and it is still one of the most active research areas in video analysis. Motion estimation methods include optical flow estimation, model based estimation, and feature tracking. In this dissertation, we are primarily interested in optical flow and model based motion estimation as they are the basis for motion activity analysis. While optical flow computation methods provide the technology for panoramic video capturing and virtual camera control, the model based motion estimation technique is the basis for our work on human motion activity recognition.

2.1.1 Optical Flow Computation

The issue of optical flow computations has been addressed in different research fields, for example, physiology, psychology, and computation vision. Optical flow was introduced by Gibson [39] in 1950 to describe the relationship between the temporal variations of the intensity and the motion of the camera and the motions and shapes of the objects. Early physiological studies include Hubel and Wiesel [52], Bridgeman [13], and Grusser and Gursser [42]. Recently, there have been many efforts in psychology and computational vision to estimate optical flow from video sequences. According to Barron et al. [5], optical flow computation methods can be characterized into four categories: Differential Techniques, Region-Based Matching methods, Energy-Based methods, and Phase-Based Techniques.

Differential Techniques

One common idea for differential techniques is using the first order derivatives of image intensity field, see Fennema and Thompson [35], Horn and Schunck [51], Nagel [65]. If a point at (x, y) at time t moves to $(x+v_x, y+v_y)$ at $(t+\delta t)$, assuming that the intensity of the original pixels remains the same, we have the following:

$$I(x, y, t) = I(x+v_x, y+v_y, t+\delta t) \tag{2-1}$$

where I is the image intensity, and $V = [v_x, v_y]^t$ is the optical flow vector. By a Taylor expansion of the above equation, we get the so called gradient constraint equation:

$$I_x v_x + I_y v_y + I_t = 0 \tag{2-2}$$

Where $I_x = \frac{\partial \mathrm{I}}{\partial x}, I_y = \frac{\partial \mathrm{I}}{\partial y}, I_t = \frac{\partial \mathrm{I}}{\partial t}$ are the partial derivatives of I along $x, y,$ and t.

The above equation provides one local constraint for optical flow, but it has two velocity variables v_x, v_y, therefore one more constraint is needed to solve the equation. However, the *normal velocity* V_n can be determined from the equation. It is equal to the perpendicular distance of the motion constraint line from the origin of (v_x, v_y) space. It can be computed as:

$$V_n = \frac{-I_t}{\sqrt{I_x^2 + I_y^2}} \tag{2-3}$$

Combining the gradient constraint with a global smoothness term to constrain the velocity field, Horn and Schunck [51] proposed a method for computing optical flow by minimizing:

$$\int_D (I_x v_x + I_y v_y + I_t)^2 + \gamma^2 (\| \nabla v_x \|_2^2 + \| \nabla v_y \|_2^2) dx dy \tag{2-4}$$

Where D is the region surrounding the pixel, γ is the smoothness term, $\| \nabla v_x \|_2^2 = (\frac{\partial v_x}{\partial x})^2 + (\frac{\partial v_x}{\partial y})^2$, $\| \nabla v_y \|_2^2 = (\frac{\partial v_y}{\partial x})^2 + (\frac{\partial v_y}{\partial y})^2$, and $\frac{\partial v_x}{\partial x}, \frac{\partial v_x}{\partial y}, \frac{\partial v_y}{\partial x}, \frac{\partial v_y}{\partial y}$ are the derivatives of v_x and v_y in the x and y directions respectively. There have been significant amount of work on extending this original idea. Generally, the approaches constraint the velocity field in different formats. For a complete reference, please refer to Barron et al. [5].

Region-Based Matching Techniques

Because of noise, or small number of frames available, or severe aliasing in the image acquisition process, numerical differentiation becomes impractical in some cases. For this reason, region based methods are often preferred. Anandan [2], Burt et al. [15], Glazer et al. [40], and Little and Verri [56] propose methods where the velocity can be computed as the shift of a region. The idea is to find the best match between image regions in one frame with surrounding regions in the other frames. Finding the best match then becomes minimizing the distance between reference region and current region. Distance measure such as the sum of squared difference (SSD) can be used:

$$SSD(\mathbf{x};\mathbf{V}) = \sum_{j=-n}^{n}\sum_{i=-n}^{n} W(i,j) * \left[I_1(\mathbf{x}+(i,j)) - I_2(\mathbf{x}+\mathbf{V}+(i,j)) \right]^2 \qquad (2\text{-}5)$$

Where $\mathbf{x} = [x, y]^t$ and W is a 2D smoothing filter.

Variant methods have been applied to the above measures to compute the optical flow. One example is Anandan [2] which uses Laplacian pyramid to compute the region optical flow at different resolution of the image sequence. Because this method is reasonably robust, it is the basis of motion compensation based coding for MPEG2, which we will discuss in section 2 of this chapter.

Energy-Based Methods

Adelson and Bergen [3], Barman et al. [4], Bigun et al. [8], Haglund [44], Heeger [48], Jahne [53], propose optical flow techniques based on the output energy of velocity-tune

filters. This is based on the observation that the Fourier transform of a 2D translation pattern is:

$$\bar{I}(\mathbf{K}, w) = \bar{I}_{t0}(\mathbf{K})\delta(w + \mathbf{V}^T\mathbf{K}) \tag{2-6}$$

where $\bar{I}_{t0}(\mathbf{K})$ is the Fourier transform of the original 2D image pattern $I(\mathbf{x}, t0)$, $\delta(k)$ is the dirac delta function, w denotes temporal frequency, and $\mathbf{K} = [k_x, k_y]^t$ is the spatial frequency.

One example of the work is by Heeger [48]. He proposed a method based on Gabor-energy filter tuned to frequency (k_x, k_y, w) to extract local energy at different spatial orientations and temporal frequencies. The optical flow is computed using least squared fit of spatiotemporal energy.

Phase-Based Techniques

The phase-based technique derives optical flow in terms of phase behavior of band-pass filter outputs. Barron et al. [5] classified zero-crossing techniques by Buxton and Buxton [17], Duncan and Chou , Hildreth [49], Waxman et al.[97] as phase–based methods. Fleet and Jepson [36] proposed a generalized phase-based method for optical flow computation.

Fleet and Jepson [36] used band pass filters to decompose the input signal according to scale, speed and orientation. Each filter output is complex valued:

$$R(\mathbf{x}, t) = \zeta(\mathbf{x}, t)e^{i\varphi(x,t)} \tag{2-7}$$

Where $\zeta(\mathbf{x},t)$ and $\varphi(x,t)$ are the amplitude and phase parts of R. The component of 2D velocity in the direction normal to level-phase contours can then be computed based on the phase information $\varphi(x,t)$. Finally, given the normal velocity estimates from different filter bands, a linear velocity model is fit to each local region to get the optical flow.

2.1.2 Model-Based Motion Estimation

The optical flow computation generally involves estimating displacement at pixel level in an image. In other applications, one may want to characterize a global motion field using models with a small number of unknown parameters. The motion models can also be used locally for a smooth surface in a scene. The model parameters provide a more compact and better representation of the motion feature of the objects than optical flow.

One of the early works on model based motion estimation was proposed by Bergen et al. [7]. The basic assumption behind the model-based method is intensity constancy, which is the essentially the same as (2-1):

$$I(\mathbf{x},t) = I(\mathbf{x}-\mathbf{V},t-1) \tag{2-8}$$

Where $\mathbf{x}$ is the position vector of a pixel and $\mathbf{V}$ is its motion vector. The SSD error measure minimization for flow field within a region is then:

$$E(\{\mathbf{V}\}) = \sum_{\mathbf{x}} (I(\mathbf{x},t) - I(\mathbf{x}-\mathbf{V},t-1))^2 \tag{2-9}$$

If we model $\mathbf{V}$ as $\mathbf{V}(\mathbf{x};\mathbf{z})$, where $\mathbf{z}$ represents the motion model parameters such as the affine model discussed in the next section. Gauss-Newton method can be used to estimate $\mathbf{V}$ based on $\mathbf{z}$. If $\mathbf{V}_i$ is the current estimate of the flow field during the i th iteration, the incremental estimate $\{\delta\mathbf{V}\}$ can be obtained by minimizing the quadratic error measure:

$$E(\{\delta\mathbf{V}\}) = \sum_{\mathbf{x}} (\Delta I + \nabla I \cdot \delta\mathbf{V})^2 \tag{2-10}$$

where $\Delta I = I(\mathbf{x},t) - I(\mathbf{x}-\mathbf{V}_i, t-1)$, and $\nabla I = (I_x, I_y)$ is the spatial gradient of I.

There are several motion parameter models used in research such as affine, planar surface, and rigid body model. Among them the affine model is of interest to us and is introduced below. The application of the affine motion model is discussed in chapter 5.

Affine Model

Affine model [7] usually applies when the distance between the background surfaces and the camera is large. It is formulated as:

$$\mathbf{V} = \mathbf{U}\boldsymbol{\kappa} \tag{2-11}$$

where $\mathbf{U} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix}$, and $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \kappa_3, \kappa_4, \kappa_5, \kappa_6)^t$. Here $\boldsymbol{\kappa}$ is the affine model parameter vector.

Based on (2-10), we can obtain the error measure:

$$E(\{\delta\boldsymbol{\kappa}\}) = \sum_{x} (\Delta I + \nabla I \cdot (\mathbf{U}\delta\boldsymbol{\kappa}))^2 \tag{2-12}$$

Suppose current estimate of $\boldsymbol{\kappa}$ is $\boldsymbol{\kappa}_i$, the incremental estimate $\delta\boldsymbol{\kappa}$ can then be solved by the following equation [7]:

$$[\sum_{\mathbf{x}} \mathbf{U}^T \nabla I \nabla I^T \mathbf{U}]\delta\boldsymbol{\kappa} = -\sum_{\mathbf{x}} \mathbf{U}^T \nabla I \Delta I \tag{2-13}$$

Then, we have $\boldsymbol{\kappa}_{i+1} = \boldsymbol{\kappa}_i + \delta\boldsymbol{\kappa}$. Note that there is a close relationship between model-based motion estimation and optical flow computation. When the surface window is small, affine model can be used to compute the optical flow of the center of the surface window around a pixel. In this case, we can use (κ_1, κ_4) to represent optical flow at the surface center. The whole optical flow field of an image can be computed by shifting the window to cover the pixels one by one.

2.2 Motion Compensation for MPEG Video Coding

Motion estimation technique helps to code video at interframe level. It exploits the interframe redundancy of video content and helps reduce the bit rate of video significantly. In MPEG 1/2 video standard, it takes the form of motion compensation. In this dissertation we use compressed domain motion compensation information for motion activity description, and compressed domain virtual camera control. Therefore, we briefly review the motion compensation concepts here. For simplicity, we use MPEG to represent both MPEG1 and MPEG2. We also assume the MPEG streams are progressive, and refer to each image in a video sequence as a frame.

2.2.1 MPEG Frame Type

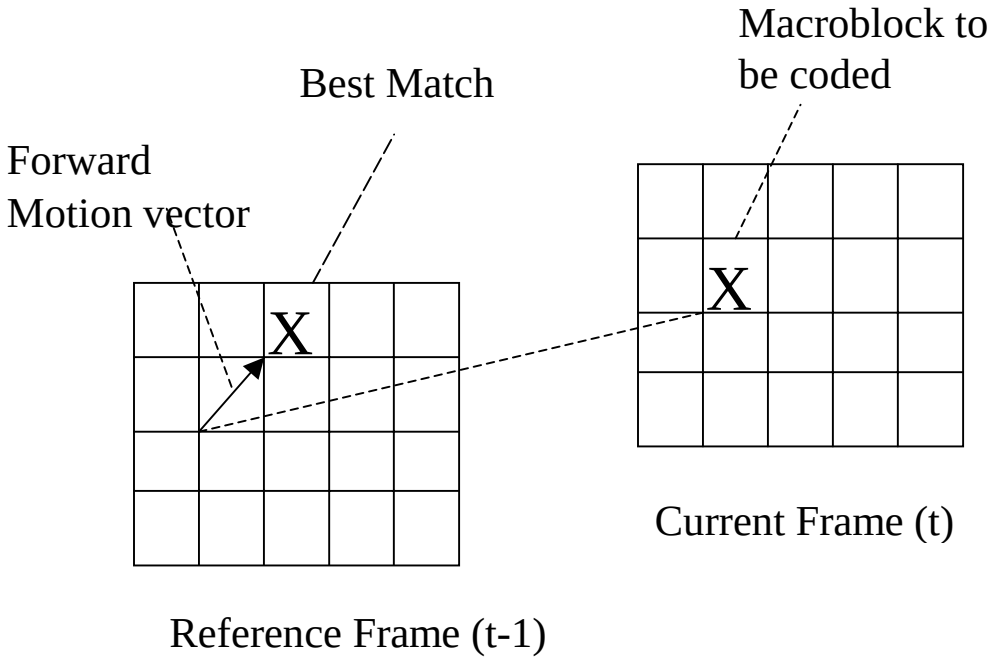


Figure 2-1. Forward motion–compensation prediction scheme for P frame. Each macroblock is 16x16 pixels.

Since video is a sequence of still images, each image can be compressed to reduce spatial redundancy. Such compression methods are called intraframe coding. In an MPEG stream, frames that are coded only using intraframe coding are called the I frames.

To exploit temporal redundancy, MPEG algorithms compute an interframe difference called prediction error. Motion compensation is used to correct the prediction for motion. The macroblock based approach is used for motion compensation. In MPEG, a macroblock is a 16x16 block of pixels that partition the whole image into small equal sized regions, as shown in Figure 2-1. In computing the motion vector, a typical MPEG algorithm tries to minimize:

$$PE(i,j) = \frac{1}{MN} \sum_{|m| \le \frac{M}{2}, |n| \le \frac{N}{2}} \sum [(I_{mn}, t) - (I_{m+i,n+j}, t-1)]^2 \qquad (2\text{-}14)$$

where M and N are macroblock sizes and are usually set to 16x16. Comparing (2-5) and (2-14), we see that the two essentially achieve the same objective. The motion vector (i, j) of (2-14) is the same as $\mathbf{V}$ in (2-5). However, (2-14) is used at the macroblock level. The computation of motion vector at macroblock level is essentially an application of region-based matching in motion estimation techniques. Each macroblock has only one motion vector to represent the whole macroblock. Suppose we have a frame of size (width, Height), then it has width*height/256 macroblock motion vectors. The macroblock motion vectors provide a sub-sampled motion field.

In unidirectional motion estimation, a target macroblock is coded based on its best match, called prediction macroblock, in a past frame called reference frame. The forward prediction error is then computed as the difference between the target macroblock in the current and the prediction macroblock in the reference frame. This is illustrated in Figure 2-1. Frames coded this way are called P frames. MPEG also applies bidirectional temporal prediction method for frame coding and referred to as motion-compensation interpolation. In bidirectional prediction, a frame is coded with two reference frames, one in the past, the other in the future. This is illustrated in Figure 2-2. A target macroblock in a B frame can be coded from the reference frame (Forward prediction), or from the future frame (Backward prediction), or the average of the two. So a macroblock of B frame can have up to two motion vectors. In an MPEG stream, a video sequence is comprised of Group of pictures (GOP) which supports

random access of the video frames. However, the order of frame types in a GOP is not fixed.

Figure 2-3 shows one typical example of a GOP that is used in our later experiments.

2.2.2 P Frame Macroblock Type

Of all the MPEG frame types, P frame is used more often for video analysis than I and B frames. Figure 2-4 lists the P frame macroblock types. For a given macroblock, the encoder first determines whether it is Motion Compensated (MC) or Non Motion Compensated (NO_MC). The input video frame is analyzed by a motion compensation estimator/predictor. For each macroblock, a scheme is used to determine whether the current block is intra/inter coded based on the prediction error. The scheme can be quite complex, but the general idea is to code the difference between target macroblock and reference macroblock when the prediction error is small, otherwise, intra-code the macroblock. In a special situation when the prediction error (for perfect match) is zero, the macroblock is not coded using prediction error and is skipped.

In general, the condition for No_MC inter-coding is as follows:

$$\sum_{\mathbf{x}} \left(I_c(\mathbf{x}) - I_r(\mathbf{x}) \right)^2 < \sigma \tag{2-15}$$

Where $\mathbf{x} = (x, y)^t$ is the position of a point in the macroblock, σ is the threshold, I_c is the current marcroblock, and I_r is the reference macroblock which is either an I frame or a P frame.

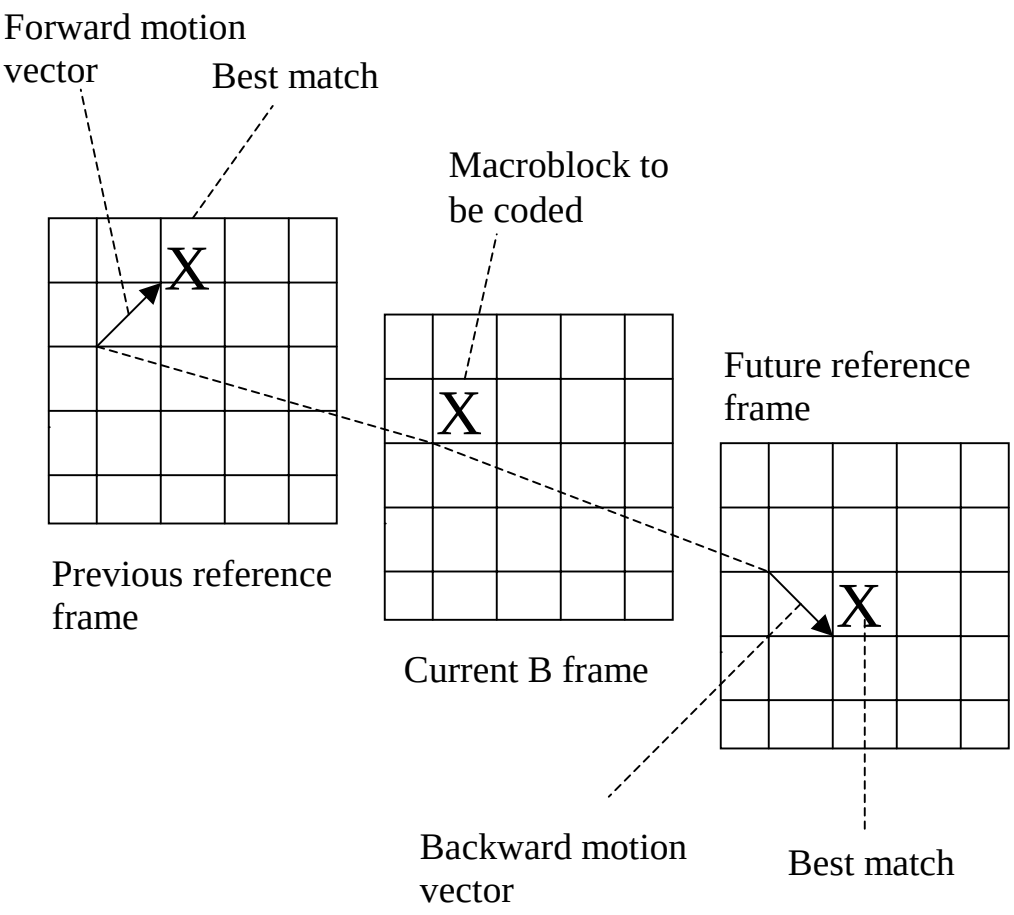


Figure 2-2. Bidirectional motion–compensation scheme for B frame.

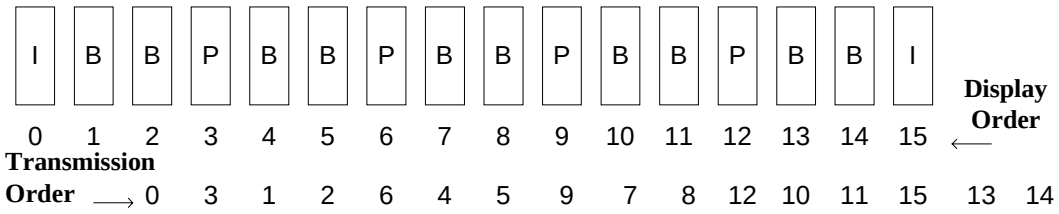


Figure 2-3. An example of MPEG group of picture coded in different types.

.

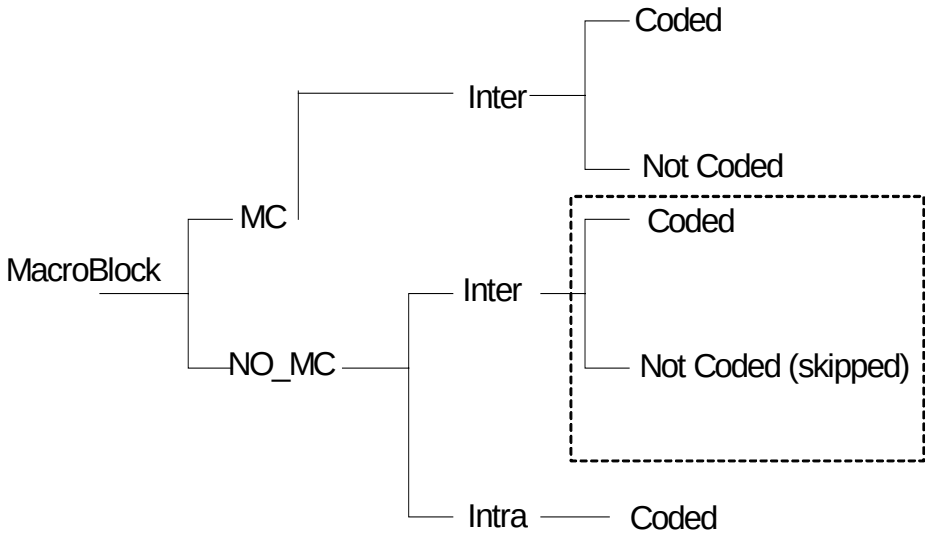


Figure 2-4. P frame macroblock type.

However, if we let $\mathbf{V} = \mathbf{0}$ in (2-9), we have a minimization function:

$$E(\{\mathbf{V}\}) = \sum_{\mathbf{x}} (I_c(\mathbf{x}) - I_r(\mathbf{x}))^2 \tag{2-16}$$

where $I_c(\mathbf{x}) = I(\mathbf{x}, t)$, $I_r(\mathbf{x}) = I(\mathbf{x}, t-1)$. Comparing (2-15) and (2-16), we see that inter-coded No_MC macroblock is basically the background with zero motion. Inter-coded No_MC macroblock is shown in the dashed region of Figure 2-4.

In Chapter 3 we will use the inter-coded No_MC macroblock information for video scene intensity description. In Chapter 4 we will use this information for compressed domain region of interest detection.

2.3 Summary

In this chapter we briefly reviewed motion estimation techniques such as model based motion estimation, differential optic flow computation, region based matching, energy-based method, and phase-based methods. Then, we introduced the MPEG motion compensation based coding techniques. The MPEG macroblock information will be used in motion activity computations in the following chapters.

3 Motion Activity for Low Level Video Indexing

In this chapter we explore the use of motion estimation for video content description. One approach to video scene classification is based on the amount of motion activity in a scene – scenes that have large motion activity and those that have minimal motion activity, as well as the spatial and temporal distribution of motion activity. The motion activity descriptors presented in this chapter aim at providing such a description of the video. We first introduce the concept of motion intensity, and then discuss the motion intensity histogram and spatial motion activity. We also discuss the computation of the descriptors in compressed domain since much of the digital video is archived in a compressed format. In the end, we discuss their applications to video indexing and video filtering.

3.1 Introduction

Motion as a visual feature has been widely used in content-based video retrieval [32], [65], [108], [110], [114]. One example application is to provide information about the dominant motion in a given region of interest encoded for search and retrieval. Another example is to provide the motion vectors directly for video content indexing. While motion vector has been used for video coding or indexing in previous research, a detailed quantitative characterization of the spatial and temporal change of motion vectors in a video has not received much attention. Information about how many regions have changed in a given frame and how the changes are distributed within a

period of time can be very useful for video indexing. For example, it can facilitate search for a segment of sports video with a typical motion activity in mind. By introducing concepts of motion intensity – the degree of scene motion change, and motion intensity histogram – the temporal distribution of motion intensity, we provide the user a description of video in terms of low level motion activity.

The concept of motion intensity comes from our observation of video content change. The level of motion change of a scene can increase or decrease within a temporal sequence. In a football video, for example, the scene change goes from motionless at the start of a play to gradually intensified play with considerable motion. Characterizing the activity level of motion – motion intensity – will be useful in describing such vide sequences. Just like music that can be described by its rhythm, changes in motion intensity can be used to characterize a video segment. We propose a "motion intensity histogram" to characterize the temporal change in motion intensity that can be used to compare and classify video sequences.

To extract the motion intensity and motion intensity histogram, we process the MPEG encoded video in the compressed domain. The motion intensity is extracted based on P frame macroblock information of the MPEG video. There are two reasons for using P frame macroblock information. First, digital video possesses redundant information; therefore, P frames are good temporal samples of the original video, and have been used in many applications [88]. Second, as discussed in Chapter 2, P frames are encoded with macroblock information that can be processed quickly. To obtain motion intensity histogram, a given video scene is partitioned into a number of units (segments), coarse

to fine, with a "fine" segment containing very few frames, and a "coarse" segment containing a large number of frames. We adaptively segment video into units with fixed percentages (1% to 20%) of original video length using the method proposed in [86]. This is based on the observation that the pattern of motion change can be across the shots, while within a shot the motion pattern can also change significantly. For example, the motion in a sports video shot can vary from motionless during the pause of the game to significant large motion when many players are moving. The video units with fixed percentages with respect to the whole video length also provide an effective way of comparing video segments quantitatively.

The motion activity descriptors at different unit lengths can be obtained once the video is segmented. Motion intensity and motion intensity histogram can then be used for video classification and retrieval for any given video units. They can also be combined with other motion activity features such as spatial motion activity descriptor [33], introduced in section 3.5, for a variety of applications. An example of video filtering is discussed in section 3.7.

3.2 Motion Intensity

The intensity of motion—the level of motion activity, and the change in it, is the first feature used for motion activity description. In this section we discuss how to extract motion intensity in the compressed domain. We use motion compensation information at marcroblock level in an MPEG video to extract the motion intensity feature.

Therefore, as mentioned in chapter 2, the following discussion applies to only P-frames in an MPEG coded video.

In order to reduce the bit rate, some macroblocks in the MPEG P frames are coded using their differences with the corresponding reference macroblocks. Note that the No_MC macroblocks in MPEG video have no motion compensation. The No_MC macroblock can also be categorized into two types, one is intra-coded and the other is inter-coded. In a special case, when the macroblock perfectly matches its reference, it is skipped and not coded at all. To simplify the illustration, the skipped frames are categorized the same as No_MC inter-coded frames. According to (2-15) and (2-16), the No_MC inter-coded macroblock has zero motion.

According to the definition of inter-coded No_MC, when the content of video changes are not too significant, and thus many macroblocks match their references, the inter-coded No_MC macroblock numbers would be high in a P frame. For example, pauses in sports games often coincide with small object motion and the corresponding inter-coded No_MC macroblock numbers would be very high. On the other hand, when the content of the video changes rapidly, and thus many macroblocks cannot be matched by their reference frames, the inter-coded No_MC macroblock numbers would be small in a P frame. The change in the number of the inter-coded No_MC macroblocks therefore depends on the video motion content change, and matches our goal to use motion intensity to describe scene changes. To provide a normalized description, we define the α-ratio of a P frame as:

$$\alpha = \frac{Number\ of\ \ inter\ No\_MC\ Macroblocks}{Total\ Number\ of\ \ Frame\ Macroblocks} \tag{3-1}$$

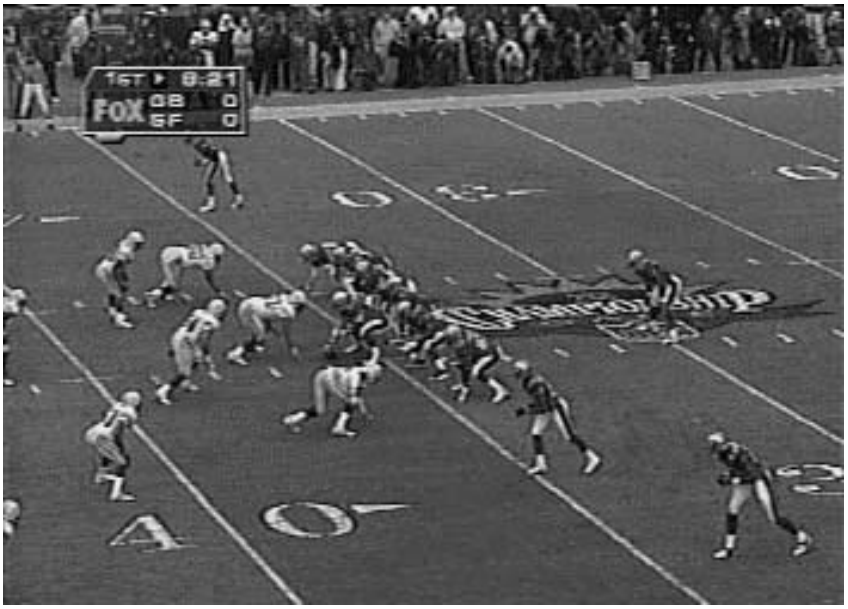
From our observation, we find that this ratio is a good measure of scene motion intensity change and it conforms with human perception very well. The higher the ratio is, the lower the scene motion intensity change is.

Figure 3-1 shows two frames from the same shot in a football video. The first one is extracted from the start of a play that is motionless. Therefore, it has a high α = 86%. The second one corresponds to the play in progress that has a significant amount of motion taking place in the scene. Therefore, it has a low α =0.05%. The two frames are from the same video shot though their motion characteristics are significantly different.

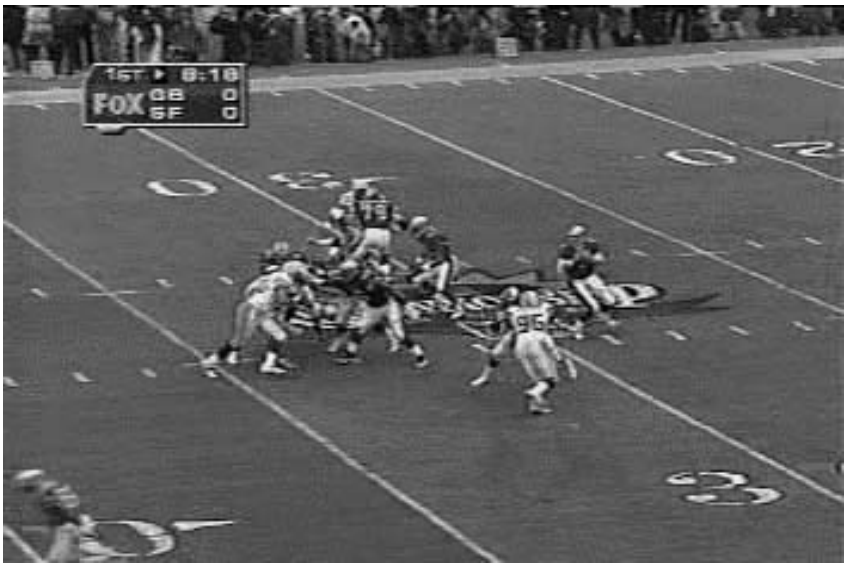
As our objective is to find motion change levels, it is not necessary to use α-ratio directly for video motion description. A logarithmic compandor is used to quantize α-ratio. From our experience, such a this non-linear scaling matches subjective motion perception reasonably well. By using this method, we can keep the quantization step higher for high ratio values. The ratio is compressed using the μ-law characteristic:

$$G_u(\alpha) = q\frac{\log(1+u\alpha/q)}{\log(1+\mu)}\ ; \qquad \alpha \le q \tag{3-2}$$

Where the parameter μ is set to 255 and q is set to 1. $G_u(\alpha)$ is used to represent motion intensity.



(a) α=0.86



(a) α=0.05

Figure 3-1. Two frames with different inter-coded No_MC ratios from the same shot. The first one is almost motionless and the second one has significant amount of motion.

3.3 Video Segmentation Based on Motion Intensity

After the motion intensity of the P frame is computed, it can be used for a statistical analysis (motion intensity histogram) of the given video sequence. A video can be segmented into a video scene, a video shot, or an even shorter video sequence. The features such as color, texture, and motion may change differently at different granularities of video. For example, while the changes of color and motion are usually both very high at the shot boundaries of a sports video, there is usually not much color change but significant amount of motion change at intra-shot level. Therefore, it is of interest to investigate one single feature throughout the video segmentation. In particular, we investigate motion intensity feature for temporal video segmentation.

Different methods have been proposed for video segmentation [10], [23], [31]. In the following we use the method proposed by Sun et al. [86] for video segmentation. The general idea is to segment a given video into different lengths that can be pre-determined by a user. The formulation of video segmentation is stated in Figure 3-2.

The ratio ϑ is set to 1%, 5%, 10%, 20%, of the total number of frames in the original video. These representative frames can be used for video summarization and indexing [85]. On the other hand, once these representative frames are extracted, their temporal positions in the video basically segment the whole video into smaller units. Correspondingly, the numbers of these units are 1%, 5%, 10%, 20%, of the number of frames in the video. These units provide a hierarchical representation of the video. For example, units at 10% (finer) level are a subset of those units at 5% (coarser) level. The

Given:

1. an ordered set of input digital video sequence $\mathbf{F}$ with cardinality N. $\mathbf{F}=\{F_1, F_2, ..., F_N\}$, where $F_1, F_2, ..., F_N$ are the frames of $\mathbf{F}$.
2. ratio ϑ such that $0<\vartheta<1$.
3. low-level content f of {color, motion intensity, ...}.

To extract:
a set of output frames $\mathbf{F}'$ with cardinality of N'.

$$\mathbf{F}'=\{R_{p1}, R_{p2}, ...R_{pN'}\}$$

where

- $N' = N * \vartheta$.
- $R_{p1}, R_{p2}, ...R_{pN'} \in \mathbf{F}$, are the representative frames of $\mathbf{F}$ with respect to feature f.
- $\mathbf{F}' \in \mathbf{F}$

Figure 3-2. Formulation of temporal video segmentation.

advantage of such segmentation is that we can compare video content change quantitatively in terms of their duration. This is especially important when we compare two sequences using their motion intensity histogram discussed in the next subsection.

3.3.1 Segmentation Criterion

A video segment can be modeled as a trajectory of multidimensional feature points in a multidimensional space. The nature of the spatial distribution of the points that represent their corresponding video frames can be described as clusters connected by abrupt or gradual changes. This nature of the distribution of points provides a sound basis for clustering techniques. Since we are analyzing the trajectories of feature points in temporally localized units, it is possible to use the change in consecutive

representative frames to represent the change within a unit. Given a unit U_i and a selected feature P, we define the *unit change* as follows:

$$Change(U_i) = D_c(R_{pi}, R_{p(i+1)}) \tag{3-3}$$

Where $R_{pi}, R_{p(i+1)}$ are the consecutive representative frames that are the boundaries of the unit, $D_c(\bullet)$ computes the difference of two frames with respect to a selected feature.

From an optimization point of view, given a selected feature, the objective of representative frame extraction is to divide a video into units that have very similar unit changes. This can be formulated as the minimization of:

$$\sum_{i=1}^{N'-2} \sum_{i=j+1}^{N'-1} \left| Change(U_i) - Change(U_j) \right| \tag{3-4}$$

Where N' is the number of representative frames.

Even though not normalized by the MPEG standard, the positions of P frames in a video stream usually take fixed positions in a Group of Pictures (GOP) [46], and consequently fixed positions in a video. Therefore, they are good temporal samples of the original video. Thus, instead of processing the video frames one by one, we process only the P-frames for temporal segmentation of the MPEG video stream. Note that the feature we choose is the motion intensity. Therefore, based on (3-2), the unit change becomes:

$$D_c(R_{pi}, R_{p(i+1)}) = \left| G_u(\alpha_i) - G_u(\alpha_{i+1}) \right| \tag{3-5}$$

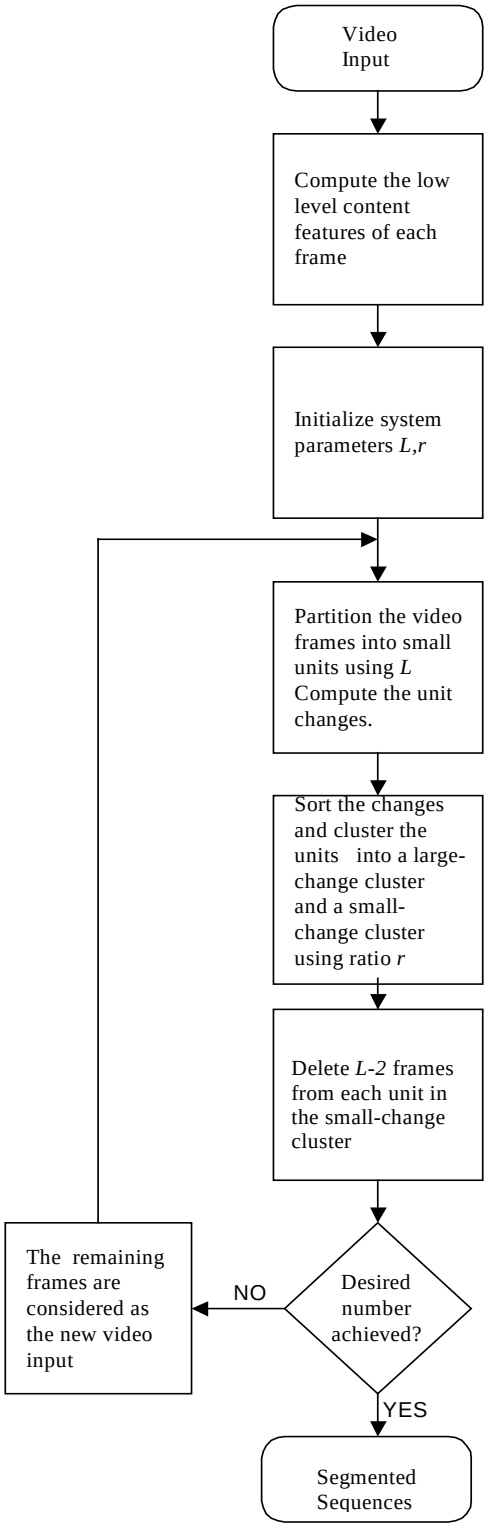


Figure 3-3. Adaptive video segmentation.

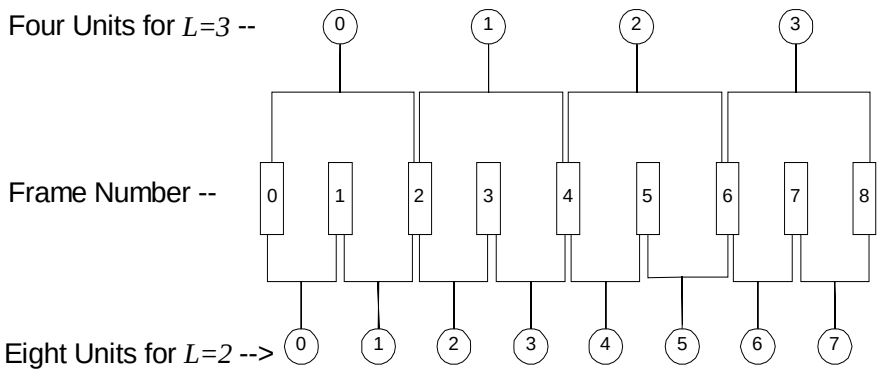


Figure 3-4. Sequence partitions.

3.3.2 Adaptive Segmentation

For a given video V with length N, suppose we want to extract N' representative frames. The motion intensity feature of each frame in V is computed first. The clustering algorithm works in an iterative fashion. We start initially with all the frames of the video and iteratively drop frames till the desired result is obtained. A schematic of our clustering algorithm is shown in Figure 3-3. We have three steps during each *iteration*.

Step 1: The sequence of the video frames is segmented into small units whose length are all L. All the units are temporally consecutive. Figure 3-4 shows the segmentation with $L = 2$ and $L = 3$ respectively. The units for $L = 3$ are {(0,1,2), (2,3,4), (4,5,6), (6,7,8)}. In each unit, the unit change is computed, which is the distance between the first frame and the last frame of the unit computed as in (3-5).

There are total of $K = \lceil N/(L-1) \rceil$ units. The computed unit changes represent the units and these changes construct an array of length K. Since our objective is to extract representative frames according to frame content changes, the changes do reflect the actual degree of content change in all the units. This is because the distance metric is computed in a temporally localized region.

Step 2: By sorting the unit changes in an ascending manner, we get an array that represents the general content change of the video. The elements which are located in the beginning part of the array represent the frames where there are small changes, while the units in the later part consist of frames having large changes.

By selecting a ratio $0 < r < 1$ we cluster the array into two clusters according to the value of unit change. The first cluster comprises of the smallest elements of the array and its length is $K * r$. Here we refer to it as the *small-change cluster*. The rest of the elements comprise the *large-change cluster*.

Step 3: If the change of a unit belongs to the currently large-change cluster, then we take all of its frames as part of the current extracted representative frames. If the change of a unit belongs to the small-change cluster, then we will delete all the frames except the first and the last frames from the unit. The first and the last frames are retained as part of the current extracted representative frames. After the deletion process, $K * r * (L-2)$ frames will be deleted.

Suppose the number of frames left is N''. If $N' \geq N''$, then our desired result is obtained and we can stop the iteration. If $N' < N''$, we can dynamically regroup all the retained frames as a new video and repeat step 1 to step 3.

With the decrease in the number of frames for comparison, small units are consequently clustered together. A unit will physically span across more frames in the original video. So it will adaptively represent a larger range of frame changes in the original video. The smaller the number N' we desire, the more times the algorithm would adaptively repeat the clustering procedure. After each iterative process, there will be frames deleted from the sequence, so the overall number of frames left will decrease each time, and the method will eventually converge to the desired number.

3.3.3 Parameter Selection

As the whole extraction process is basically unsupervised, the result will depend on the proper selection of the parameters L and r.

Selection of L

If $L = 2$, the distance is in fact consecutive frame difference. Consecutive frame difference has been successfully applied for shot detection in the past, but it is not suitable for finding representative frames.

In general, if we use a large L, the algorithm will converge fast. At the beginning of the process, a large L will not degrade the result. However, if the required number of units is very small, a large number of iterations will be needed. After many iterations, the unit will physically span across many frames in the video. In this case, if we just keep the

first and last frame of the units in the small-change cluster, the video content will not be well represented by the remaining frames. Therefore a large L will degrade result in the end. In the experiments, we usually set L to a small value such as 3 or 5.

Selection of r

If $L = 3 \text{ or } 5$, then $1 \text{ or } 3$ frames in each unit of the small-distance cluster will be deleted after the execution of each loop of the iteration discussed in section 3.3.2. Accordingly, if before the iteration the retained number is $N"$, then after the iteration, approximately the following number of frames will be deleted.

$$\begin{aligned} N"/2*r*1 &= N"*r*(1/2), \quad \text{for } L = 3, \\ N"/4*r*3 &= N"*r*(3/4), \quad \text{for } L = 5, \end{aligned} \tag{3-6}$$

In many cases, it is really not critical that the number of extracted representative frames is strictly equal to the required number. Assume that the maximum allowed error is 20%. Then we can calculate that the maximum allowed r as:

$$\begin{aligned} r &= 0.2/(1/2) = 0.4 \quad , \text{ for } L = 3, \\ r &= 0.2/(3/4) \approx 0.3 \ , \text{for } L = 5, \end{aligned} \tag{3-7}$$

Since the larger the ratio r, the faster the algorithm converges, we try to use the largest r that we can possibly use in our algorithm. In practice, we select $r = 0.3$.

3.4 Motion Intensity Histogram

Assume a video has been segmented into temporal video units, where the video units can be a video sequence, a shot, or small temporal segments. Then the temporal

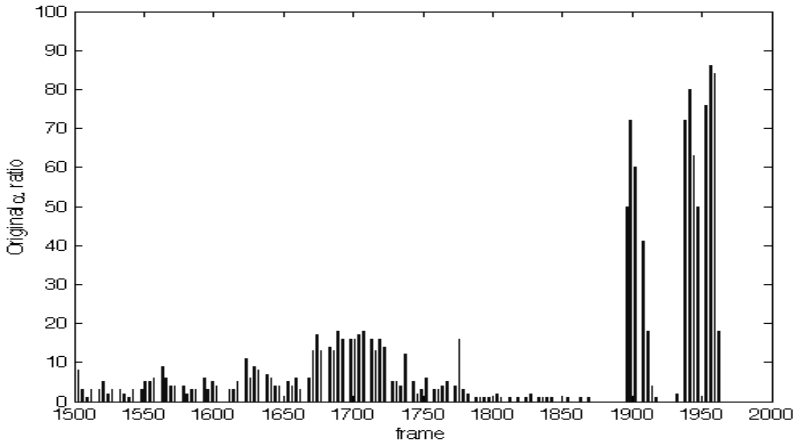
histogram of the motion intensity can be used to characterize the units' temporal motion intensity distributions. Note that the histogram is not dependent on the video segment size, therefore it can be easily scaled to multiple video levels and to support hierarchical video content description.

We need to quantatize motion intensity into levels before further computing the motion intensity histogram. We use vector quantization methods to transform $G_u(\alpha)$ into N_l quantized intensity levels. The codebook of N_l entries is extracted from the $G_u(\alpha)$ data set first, then $G_u(\alpha)$ values are indexed using the code book. When describing a scene intensity, it is reasonable to use several degrees like very low, low, medium, high, very high. As a result, in our experiments, we set $N_l = 5$. Figure 3-5 shows quantatized motion intensity of part of a soccer video.

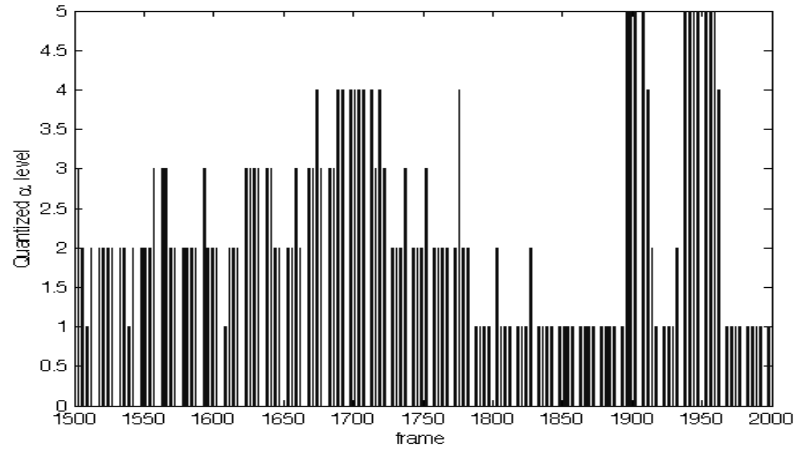
Given a video unit, we define our temporal descriptor as the corresponding motion intensity histogram of the unit: $MIH = [p_1, p_2, p_3, \ldots p_{N_l}]^t$, where p_i is the percentage of the quantized motion corresponding to the *i*-th quantization level , and $\sum_{i=1}^{N_l} p_i = 1$. We use $N_l = 5$ in the experiment.

3.5 Spatial Motion Activity Descriptor

While motion intensity histogram characterizes the dynamic change of motion activity along the temporal direction, we need to combine it with the spatial distribution of motion activity to form powerful video analysis tools [88]. To motivate the discussion



a) Original α-Ratio scaled by 100



b) Quantized α Level

Figure 3-5. α-ratios and their quantized levels from a part of soccer video. (MPEG 7 Test Data V18)

in the experimental results, we introduce the spatial motion activity descriptor proposed by Divakaran and Sun [33]. They use the magnitude of motion vectors with a run-length framework to form a descriptor this descriptor.

To extract this spatial motion descriptor, the average motion vector magnitude per macro-block of the frame/object C_{mv}^{avg} is computed as: $C_{mv}^{avg} = \frac{1}{MN} \sum_{i=0}^{M} \sum_{j=0}^{N} C_{mv}(i, j)$ where M and N are the width and height in macroblocks in the fame. This C_{mv}^{avg} is used as a threshold on C_{mv} to get the matrix:

$$C_{mv}^{thresh}(i, j) = \begin{cases} C_{mv}(i, j), \text{ if } C_{mv}(i, j) \geq C_{mv}^{avg} \\ 0, \text{ otherwise} \end{cases}$$

Then lengths of runs of zeroes in the above matrix are computed using a raster-scan order. The run-lengths are classified then into three categories, short, medium and long and they are normalized with respect to the object/frame width. In this case, the short runs are defined to be 1/3 of the frame width or lower, the medium runs to be greater than 1/3 the frame width and less than 2/3 of the frame width, and the long runs to be all runs that are greater than or equal to the width. N_{sr} is the number of short runs, with N_{mr}, N_{lr} similarly defined. Such a "quantization" of runs can be used to get some invariance with respect to translation and reflection.

The spatial descriptor can then be constructed as $(C_{mv}^{avg}, N_{sr}, N_{mr}, N_{lr})$. Note that the descriptor indirectly expresses the number and size and shape of distinct moving objects in the frame and their distribution across the frame. For a frame with a single large object such as a talking head, the number of short run-lengths is high, whereas for a frame with several small objects such as an aerial shot of a soccer game the number of short run-lengths is lower.

3.6 Similarity Measure

In order to compare feature vectors, we need to provide a similarity measure. Generally, the feature vector components proposed above have correlations. Therefore, the Mahalanobis distance metric is better suited for comparing the motion activity descriptors. The Mahalanobis distance between two feature vectors Q_1 and Q_2 is given by:

$$D_M(Q_1, Q_2) = [Q_1 - Q_2]\boldsymbol{\Sigma}^{-1}[Q_1 - Q_2] \tag{3-8}$$

Where $\boldsymbol{\Sigma}$ is the covariance matrix of the feature vector. Since $\boldsymbol{\Sigma}^{-1}$ is symmetric, it is a semi or positive definite matrix. So we can diagonalize it as $\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\Lambda}\boldsymbol{\Phi}$. Where $\boldsymbol{\Lambda}$ is a diagonal matrix, and $\boldsymbol{\Phi}$ is an orthogonal matrix. The computation of (3-8) can be simplified in terms of Euclidean distance as,

$$D_M(Q_1, Q_2) = D_E\left(\sqrt{\boldsymbol{\Lambda}}\boldsymbol{\Phi}Q_1,\ \sqrt{\boldsymbol{\Lambda}}\boldsymbol{\Phi}Q_2\right) \tag{3-9}$$

Since $\boldsymbol{\Lambda}$ and $\boldsymbol{\Phi}$ can be computed directly from $\boldsymbol{\Sigma}^{-1}$, the complexity of the computation of the vector distance can be reduced from $O(n^2)$ to $O(n)$, where n is dimension of the feature vector.

3.7 Experimental Results and Applications

The motion activity descriptors can be used in a wide range of applications. Here, we discuss how to apply these low level descriptors for video indexing and filtering. To

extract the descriptors, we first compute the motion intensities for each P frame for a given MPEG video. Then the video is segmented into small temporal units as discussed in section 3.3. The numbers of units are 1%, 5%, 10%, and 20% of the original video length. The spatial activity descriptor is computed for each P frame in each video unit. The motion intensity histogram is computed for each unit.

3.7.1 Video Indexing

Classification of video unit for browsing

To test the effectiveness of motion activity descriptor, we use motion intensity histogram to classify the video. Figure 3-6 shows our classification result on a video of American football mixed with commercials. The segmented units of frames are classified into five categories and are shown in the figure. From left to right there are five columns that represent the five clusters. In each column there are 4 starting r-frames selected to represent corresponding category. The demonstration allows the user to browse through the video based on motion intensity histogram.

The first, second, and fourth image of the first row in Figure 3-6 are three representative frames from the first shot of the video and they correspond to three consecutive video units. These frames are actually classified into three clusters. The quantized motion intensity change of the first shot is shown in Figure 3-7. If we use a shot based analysis similar to that of [106], we cannot get this finer level information. So, by segmenting a video into finer temporal units of frames, we can characterize the detail change patterns of a video or even a shot.

Figure 3-6. Video unit classification based on motion intensity histogram. The video units are classified into five clusters. Representative frames from each unit is displayed to show the content of the unit.

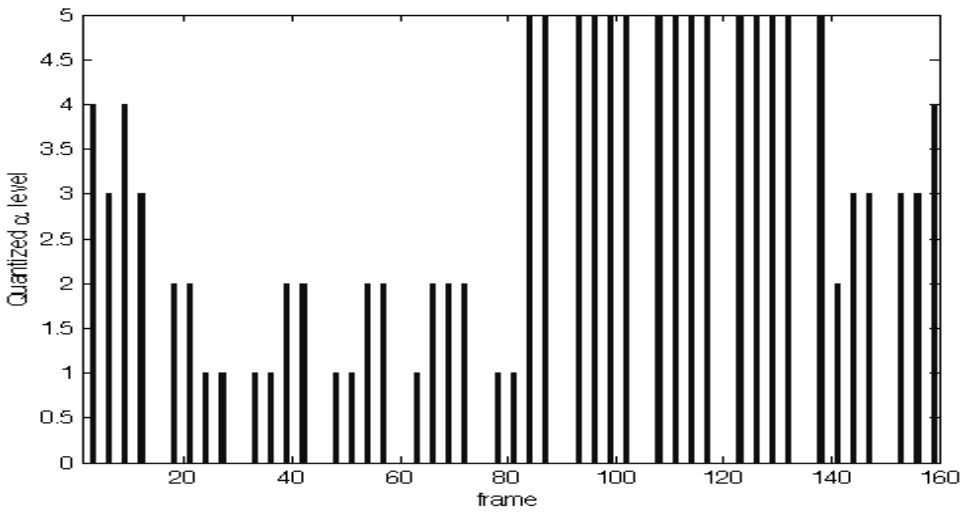


Figure 3-7. Motion intensities of the first shot in the football and commercial video.

Figure 3-8 and Figure 3-9 shows two other examples of browsing – one from the American football video and the other from the soccer video from the MPEG-7 set. Watching the video units, it is clear that segments of the video with similar motion (such as pauses in the game, start of the play, close-up shots) are all clustered together. The clusters shown in Figure 3-8 and Figure 3-9 correspond to slow or very little motion (large α-ratio).

Commercial Detection

It is observed that the change of motion vectors in commercial video is neither extremely high nor extremely low [60]. As shown in Figure 3-7, the first several commercial video units are classified into cluster 3, whose content concentrates on medium level motion intensity. This motivates us to use motion intensity histogram as a feature for detecting commercials. The video is segmented into units with 5% of its

length. The total number of units is 2061, corresponding to a 20-minute video. The first half of the units is used for training and the other half is used for test. Two popular classification methods, K-means [58] and Support vector machine (SVM) [43] with radial basis function (RBF), are used. The result is shown in Table 3-1. Note that with this single feature, the methods can achieve an error rate to 26.77%. While it is expected the detection error could be further reduced when combined with other techniques such as black frame detection, static scene detection, text location detection [1], research in this direction is not our goal in this dissertation and therefore not discussed further.

Subjective Tests

To further verify the effectiveness of the descriptors, ninety-two units from 18 videos of news, sports, and drama from MPEG-7 data set are used for subjective test. They are chosen to be 5% of their original video lengths, and are categorized into five groups based on motion intensity histogram. Six subjects take the test to categorize them as well. In the case of spatial descriptor, the first frame of each unit is used for spatial descriptor computation.

Table 3-2 lists the subjective test results. The results indicate that descriptors such as the motion intensity histogram and spatial descriptors are best used with sports and news content. This is consistent with the fact that the semantics and motion features are strongly correlated in sports, moderately correlated in news, and not well correlated in drama. Therefore, it is important to choose the right domain for the application of the motion activity descriptors.

Figure 3-8. Cluster 5 of football and commercial

Figure 3-9. Cluster 5 of soccer(MPEG-7 test data)

Method	K-means	SVM(RBF)
Training Segments	1-1030	1-1030
Test Segments	1031-2061	1031-2061
Football Error Rate	283/728	62/728
Commercial Error Rate	94/303	214/303
Total Error Rate	36.56%	26.77%

Table 3-2. Separation of commercial and football video.

Units	Precision	
	MIH	Spatial
News	63%	60%
Sports	80%	75%
Drama	30%	20%

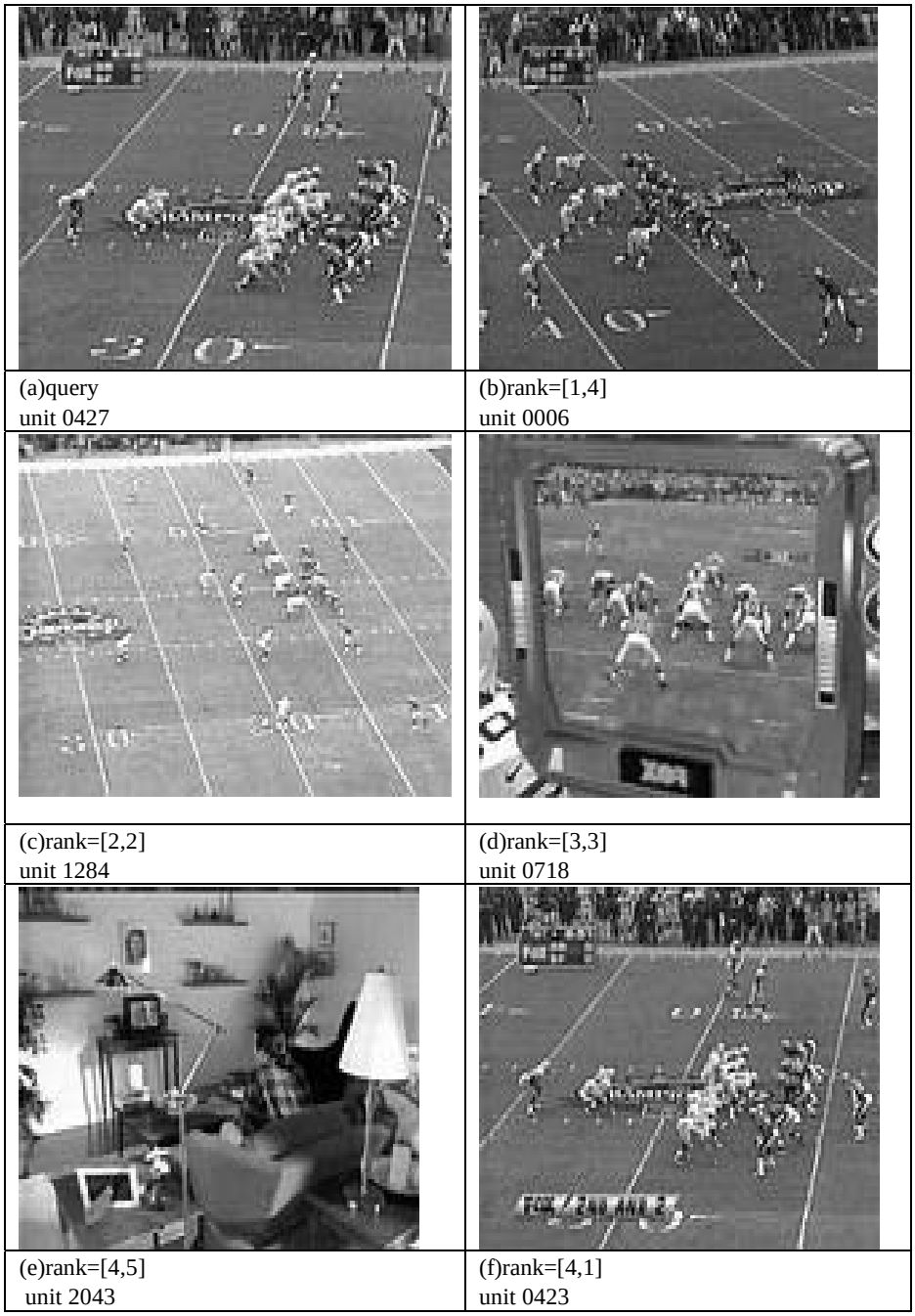
Table 3-1. Subjective test results for video retrieval based on MIH and spatial motion activity.

Descriptor/Percentage	1%	5%	10%	20%
Motion Intensity Average	Poor	Poor	Poor	Good
Motion Intensity Histogram	Good	Good	Good	Poor
Spatial descriptor	Poor	Good	Good	Good

Table 3-3. Expected performance of the motion activity descriptors at different granularities.

3.7.2 Video Filtering

It is intuitive that different attributes are suitable at different granularities of the video. For instance, motion intensity and the spatial attribute are best used to describe a locally homogenous unit of video and is not meaningful when applied to, say, an hour of video. We describe the utility of each attribute at each level in Table 3-3.



(a)query unit 0427 | (b)rank=[1,4] unit 0006

(c)rank=[2,2] unit 1284 | (d)rank=[3,3] unit 0718

(e)rank=[4,5] unit 2043 | (f)rank=[4,1] unit 0423

Figure 3-10. Motion intensity histogram and spatial descriptor for indexing and filtering. The first number in each bracket gives query result based on MIH. The second one gives query result after spatial descriptor processing.

The motion activity descriptor provides a novel way to segment content into semantically distinct units, and thus enables the user to get to the desired content by filtering or skipping over undesired content at different granularities. For example, sitcoms can be easily distinguished from high-motion-content sports scenes such as a soccer game. The temporal histogram of an hour of soccer video would have a high percentage of high action, which would help rule it out as typical drama content. Similarly the motion intensity descriptor helps us to directly access the high action parts of a sports video or to skip over them as needed. However, the motion activity descriptor is not so useful for intra-program browsing, such as searching within a sitcom program, since the gross motion characteristics of drama content do not change much.

For a sports or news video, motion intensity histogram can serve to filter at the coarser levels of video. Once we have located the program of interest, the spatial attribute is useful in effectively locating similar activities among the video segments. Figure 3-10 shows the result of video filtering based on such strategy. The first number in each rank bracket indicates the filtering result after using motion intensity that helps to extract the five candidate units. Then, they are used for further spatial filtering. The spatial descriptor for all P frames in all candidate units and the query unit are computed. The distance between a candidate and the query is computed as the smallest distance between two spatial descriptors, one from the query P frames and the other from the candidate P frames. Then we can reorder the candidates based on their distances to the query. The second number in each rank bracket indicates the result after the spatial

filtering. Unit_0423 is our target, but it is the last one among the MIH filtering results. After further spatial processing, it moves to rank 1 as expected.

3.8 Summary

In this chapter, we proposed two new low level motion activity descriptors, motion intensity and motion intensity histogram. Motion intensity represents the degree of change in motion in a scene, and the motion intensity histogram represents the temporal statistics of motion intensity. The motion intensity and motion intensity histogram are then used for feature clustering and video filtering. These two descriptors have been accepted as part of ISO/MPEG-7 motion activity descriptors. While it is true that that the semantics and motion features are significantly correlated in sports and news video, the motion activity descriptor is still a low level descriptor. Some of the applications that are currently being developed in MPEG-7 use this descriptor together with other features such as color and texture for effective semantic level video indexing.

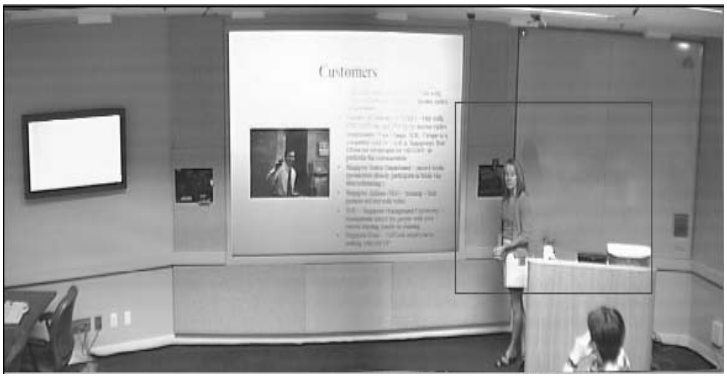
4 Virtual Camera Control based on Human Motion Activity

Automatic capturing of speakers in a lecture/conference room environment is of much interest in many applications, including video indexing. During a lecture, the speaker may move around in the front of the lecture room. The general idea of human activity capture discussed in this chapter is to produce a smooth region of interest (ROI) video output. The ROI video covers the speaker and it is much smaller than the original scene size. Therefore, the produced ROI video saves much bandwidth for delivery. The issues discussed in this chapter include the design of a panoramic capture system, modeling and detection of the speaker, filtering of the ROI, and simulating the human controlled video recording process.

4.1 Introduction

A typical scenario that this chapter is concerned with is that of a speaker giving a lecture in a classroom/seminar or teleconference. The speaker may move around, stop, turn his body, or perform some gestures. One would like to obtain the ROI video sequence in the scene that includes the speaker.

The first problem considered here is the design of a system to capture the events. It is natural to use a panoramic camera system to capture the whole scene. Processing the panoramic video will obtain the ROI. The advantage of a panoramic system is that the



(a) Panoramic scene view (b) ROI output

Figure 4-1. An example of a panoramic scene and its ROI. The rectangle region of the ROI in (a) is displayed in (b) as output.

speaker is always in the scene when s/he moves during the lecture. In our research, the FlyCam [37] system is used to produce panoramic video. Figure 4-1(a) shows the frame from a panoramic video, and its corresponding ROI is shown in Figure 4-1(b).

The second problem considered here is the control of the virtual camera to output smooth ROI sequence. In the case of real time virtual camera control, the panoramic video is usually in raw image format. It requires virtual camera control in uncompressed domain. After the panoramic video is stored and delivered in a compressed format, it usually requires virtual camera control in compressed domain. The MPEG video compression format is considered in this discussion. Our objective is to provide a fast and robust solution for virtual camera control in both the compressed and uncompressed domain.

A novel method is proposed to process the panoramic video to produce ROI video output. The method integrates ROI detection, tracking, and virtual camera control. In uncompressed domain, it detects ROI based on motion and color information. In

compressed domain, it first detects the ROI based on P picture macroblock information. Then, it up-samples the detection results to obtain the ROI of the whole video stream. The ROI is then tracked using a Kalman filter. The Kalman filter output is used to steer a "virtual camera" for displaying or recording ROI video. The Kalman filter output is smoothed to simulate the response of a human camera operator (this will be discussed in later sections). Since the panoramic camera is statically mounted, no physical camera control is needed.

4.2 Related Work

Research on automatically capturing lectures or conferences can be categorized into two areas. The first involves active camera tracking and the second involves virtual camera control based on panoramic video capturing.

Active camera control has been investigated by Zheng et al. [105] for robot control. Zobel et al. [107] design camera control method for the purpose of visual tracking. Sony's EVI-D30 camera [113] can be used to track moving objects and has the basic functions needed for the application presented in this chapter. However, in our experiment, we find this kind of steerable camera suffers from the drawback that the objects are difficult to track once they drift out of the camera's field of view. Mukhopadhyay and Smith [64] use infrared beacons for tracking, which also suffer from the same problem.

Chen and Williams [22] and many others have developed systems that compose existing still images into a panorama that can be dynamically viewed. Teodosio and Bender [91] have developed a system that composites successive video frames into a still panorama. Nayar [67] has developed an omnidirectional digital camera using curved mirrors. A conventional camera captures the image from a parabolic mirror, resulting in a hemispherical field of view. Majumder et al. [60] use 12 video cameras and a mirror apparatus to form a spherical panoramic image using texture-mapping hardware. Swaminathan and Nayar [89] have taken a similar approach, using an array of board cameras. Instead of piecewise image warping, a table lookup system directly warps each image pixel into the composite panorama. In a more recent work by Nicolescu and Medioni [68], a camera array is used for panoramic image composition. There is also commercially available low-resolution panoramic camera used for meeting recording, by Lee et al. [54] . Other commercially available systems include BeHere [109] and IPIX [111]. Other recent systems that stitch multiple camera input include RingCam [27] developed by Cutler et al. for meeting recording. In our work, we use the FlyCam [37] system to capture panoramic video. FlyCam stitches video from multiple cameras to create a high-resolution output. While in this work we use FlyCam for panoramic video capturing, in general any kind of panoramic video can be used with our system.

Speaker tracking is needed to generate the best ROI video from the captured panoramic video. Previous person-tracking efforts date back to the early 1980s. An example is O'Rourke and Badler's [72] work on 2D kinematic modeling. Other vision-based techniques include skin-color-based tracking by Stiefelhagen et al. [80], motion-based

tracking by Cutler and Turk [26], and shape-based tracking by Baumberg and Hogg [6]. Darrell et al. [29] integrate stereo, color, and face detection with person tracking. Wang and Brandstein [95] combine image and audio data (from a microphone array) for the purpose of face tracking. Wang and Chang [96] have developed a system that can detect a face in an MPEG video based on DCT coefficients, avoiding the expense of decompression. Their face detection rates reportedly approach 90%. Depending on the application, the tracking system can be complex, for example, Tao et al. [90] use a layered representation for multiple object tracking. As we will see in sections 4.5 and 4.6, complex speaker tracking models are not needed for the specific application presented in this paper.

Since the main objective of the above systems is tracking or detection, the output of these systems is usually an object outline. Using this kind of raw tracking results to steer ROI selection usually produces objectionable jitter in the video output. Therefore, the ROI output must be processed for optimal control of the virtual camera. Examples include the 3D virtual cinematographer by He et al. [47], 3D animation by Gleicher and Witkin [41], and fovea area view by Wei and Li [98]. Since the purpose of work here involves moving a small ROI rectangle inside a large panoramic video, our virtual camera control is equivalent to controlling a moving camera in a 2D image plane. Our main concern is to design a new method that simulates the response of a human operator for lecture capturing, as discussed in section 4.8.

4.3 The FlyCam Panoramic Video System

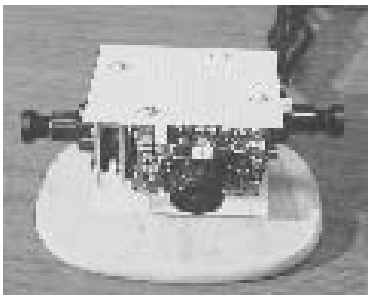
To motivate camera control discussion, we first introduce the FlyCam panoramic video system designed by Foote and Kimber [37]. The system generates panoramic video from multiple adjacent cameras in real time. In generating the panoramic images, lens distortions are corrected and the images are stitched by digital warping.

4.3.1 Hardware Construction

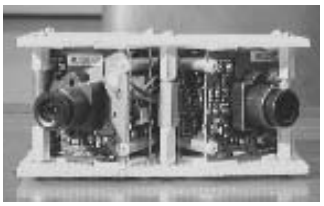
Figure 4-2 shows examples of the FlyCam prototype constructed from multiple video cameras. The cameras do not share a common center of projection. Thus, it is not necessary to align or optically calibrate the cameras in any way as long as their fields of view overlap slightly. The system shown in Figure 4-2(a) can capture a panoramic view of 360º and the system shown in Figure 4-2(b) can capture a panoramic view of 180º. Depending on the application, different number of cameras can be used to capture different kinds of scenes. The panoramic view captured in Figure 4-1(a) is produced by using the system shown in Figure 4-2(b).

4.3.2 Piecewise Image Stitching

To merge images from adjacent cameras, piecewise perspective warping of quadrilateral regions is used to correct lens distortion and map images from adjacent cameras. For digital warping, a number of image registration points are manually identified. In practice, the system chooses the corners of a grid of squares as registration points. The four corners of each square form a quadrilateral "patch" as shown in Figure 4-3. The



(a) 360º view FlyCam.



(b) 180º view FlyCam.

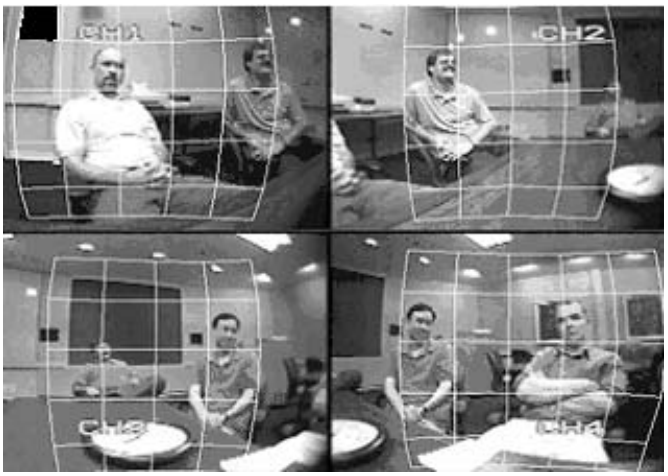
Figure 4-2. FlyCam panoramic video system

patches are warped back to a square and tiled with their neighbors to form the panoramic image.

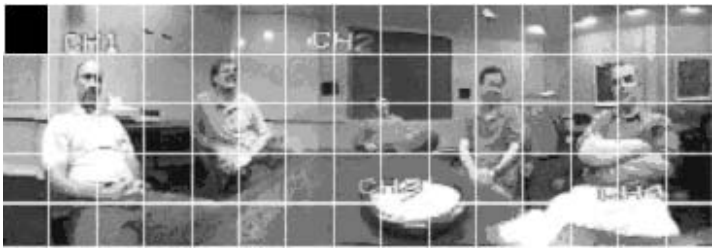
Bilinear transformations are used to warp the patches. Each patch is then mapped into a square "tile" in the panoramic image. As shown in Figure 4-3, the black quadrangle at the upper-left corner of Figure 4-3(a) is mapped to the black rectangle tile at the upper-left corner of Figure 4-3(b). For this discussion, the original coordinate system is assumed to be (u,v) and the warped coordinate system to be (x,y). Given that the tiles are square, with corners at known coordinates, the transformation from (u,v) coordinate system to the warped coordinate system (x,y) is given by:

$$[x\ y] = \begin{bmatrix} uv & u & v & 1 \end{bmatrix} \begin{bmatrix} \eta_3 & \beta_3 \\ \eta_2 & \beta_2 \\ \eta_1 & \beta_1 \\ \eta_0 & \beta_0 \end{bmatrix} \tag{4-1}$$

Where $\eta_i, \beta_i, i = 0,1,2,3$ are the bilinear transformation parameters. To calculate a pixel value in the warped coordinate system (x,y), the above equation is inverted by solving



(a) Raw camera images showing patches.



(b) Composite panoramic video frame

Figure 4-3. Raw camera images and composite panoramic video frame. The images are obtained from [37].

for (u,v) in terms of (x,y). This allows for what is termed as "inverse mapping." Details can be found in [100]. For every pixel in the warped coordinate system, the corresponding pixel in the unwarped system is found and its value is copied.

4.3.3 Border Patch Cross-fading

Since there is luminance difference between adjacent cameras, cross fading is introduced to minimize the problem. The pixel value in a patch is given by a linear combination of the component patches. In the panoramic image, the pixels on the left come from the left camera, pixels on the right in the panoramic image come from the right camera, and pixels in the middle are a linear mixture of the two.

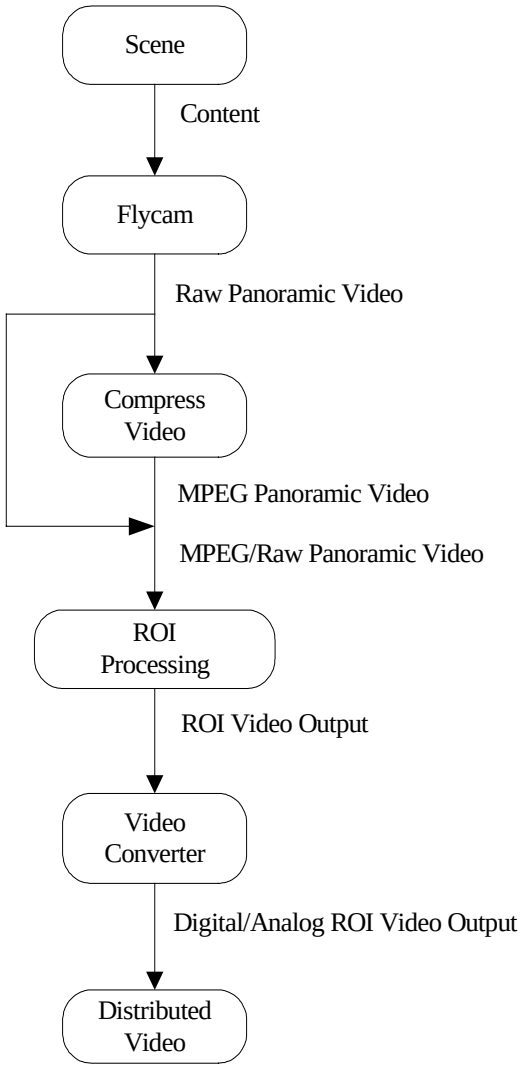


Figure 4-4. General system architecture for speaker tracking and recording system using FlyCam.

The cameras are not aligned to a common center of projection. Therefore, the panoramic image will have imperfections due to disparity between the cameras. Blending the border patches reduces the presence of disparity artifacts. For the classroom or teleconference applications presented here, subjects do not get close enough to the FlyCam that disparity is noticeable.

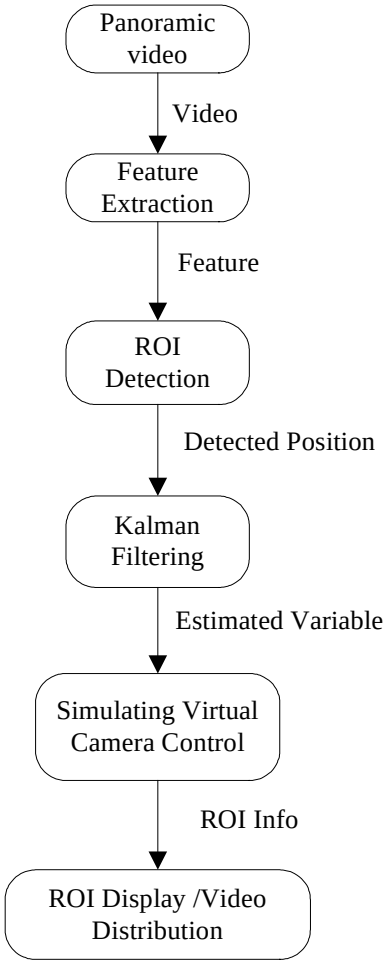


Figure 4-5. ROI detection and virtual camera control components in uncompressed domain.

4.4 System Architecture

4.4.1 General System Architecture

Figure 4-4 shows the general structure of the proposed person tracking and recording system using an 180º FlyCam. Each FlyCam component camera produces an NTSC video that is digitized using a frame grabber. The videos are stitched into panoramic video. The panoramic can be compressed into MPEG video or kept in raw format before ROI processing. After ROI processing, the output digital video can be recorded or distributed, for example, over the web. Additionally, the ROI video can be converted back to an analog signal for recording or distribution. After image stitching, the core part of the tracking system is the ROI processing, as discussed in the following sections.

4.4.2 ROI Detection and Virtual Camera Control Component

Figure 4-5 shows a general schematic of this ROI detection and virtual camera control in uncompressed domain. The input to the ROI detection component is the stitched high-resolution wide-angle video from the FlyCam. In the detection phase, the position of the ROI is detected from computed visual features. The detected position is then fed into a Kalman filter for position tracking. Estimation results from the output of the detection process are then smoothed to simulate virtual camera control and thus produce the ROI output video.

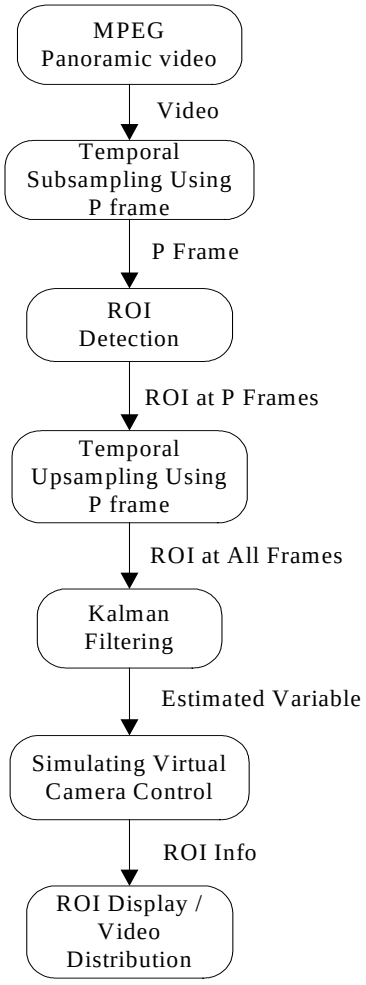


Figure 4-6. ROI detection and virtual camera control components in compressed Domain.

A schematic of the virtual camera control process in the compressed domain is shown in Figure 4-6. It shares two components with uncompressed domain processing: Kalman filtering and simulating virtual camera control. The input is a panoramic video in MPEG-2 format. First, the ROI is detected using the P frames in the MPEG-2 stream. This includes detecting the ROI in a P frame, and then propagating the results to neighboring frames by up-sampling. Next, the output of the ROI detection is fed into a Kalman filter. The Kalman filter estimates the speed and position of the speaker. These estimated parameters are used for virtual camera control. The ROI output can be used to display the video. It can also be used to extract the ROI video from the original panoramic video for storage purposes.

4.5 ROI Detection in Uncompressed Panoramic Video

Many methods have been proposed for object tracking. Since our primary objective is to capture a single speaker in a panorama, complex models such as those introduced in related work are not needed. The speaker is modeled as a point object corresponding to the centroid of the speaker's motion. The ROI output is a rectangular region of predetermined size, for example 200x200, surrounding the centroid point. Thus, ROI detection reduces to detecting the centroid of the moving part of the body.

4.5.1 Feature Extraction

Two principal features are considered for tracking: normal flow and color. The proposed solution is based on the overall confidence of motion and color change at

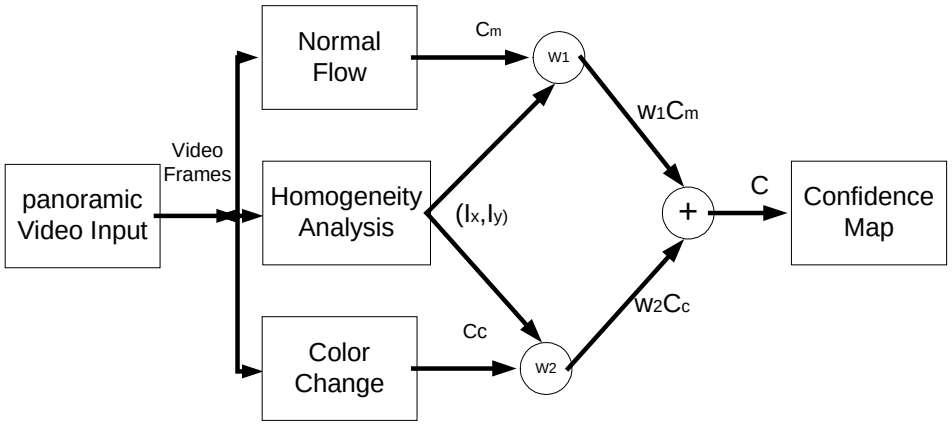


Figure 4-7. Building the confidence map.

each pixel. The confidence value is computed as a weighted sum of the color and motion features.

For motion score computation, we use the normal optical flow that is defined by (2-3) as: $V_n = \frac{I_t}{\sqrt{I_x^2 + I_y^2}}$. This value is then normalized to [0,1], and is taken as the confidence of motion at the pixel. The normalized values are denoted by $C_m(x, y)$.

In addition to motion, color provides important information about the scene. While any of the traditional color spaces (such as RGB, HSV and LUV) can be used for the computations, it is observed that the HSV space is better suited for computing the color changes. Separating hue from the saturation and brightness adds robustness under most lighting variations. For example, Bradski [11] uses the distribution of the hue value for tracking. The pixel-wise hue difference between two consecutive frames is normalized. The normalized value is defined as color confidence and denoted by $C_c(x, y)$.

The overall confidence value of motion and color change at each pixel can then be computed as a weighted sum of $C_c(x,y)$ and $C_m(x,y)$:

$$C(x,y) = w_1 C_m(x,y) + w_2 C_c(x,y) \tag{4-2}$$

Though weights in (4-2) can be fixed, a better way to combine the motion and color information is to use a spatially varying weight according to the homogeneity of the image. This can be obtained directly from the spatial derivatives of the image as shown in Figure 4-7. We choose w_1 to be $Max(I_x, I_y)$, where I_x and I_y are the normalized (0-1) spatial derivates at x and y directions, and set $w_2 = 1 - w_1$. If the spatial derivatives at a given pixel are very small, (2-3) tends to create large errors for normal flow estimation. In this case w_1 can be set to zero. This choice of w_1 is observed to work well in our experiments. Optimal selection of the weights requires extensive data training that is not discussed here. After the confidence value at each pixel is computed, a confidence map for a given video frame is obtained. This confidence map is then used for feature tracking. Figure 4-8(b) shows the confidence map of a frame from a panoramic video shown in Figure 4-8(a).

4.5.2 Centroid Detection

Thresholding the confidence map separates the moving part of the body from non-moving region (background plus non-moving part of the body) of the scene. Figure 4-8(c) shows an example of such a moving part obtained from Figure 4-8(b). The white pixel is the non-moving region, and the black pixel is the moving part of the body. The

centroid of the moving part of the body (and thus the ROI) can be located from the first order spatial moment of the moving part.

Figure 4-8(d) shows the manually segmented moving part. It is observed that when the speaker's clothes do not have much texture as shown in Figure 4-8(a), the detected moving part tends to be located at the edges of the body. Since the body is symmetric in most cases, the detected centroid will not drift much (have much error) in the x direction but it will drift more in the y direction. Nevertheless, in general the drift (the difference between the computed centroid and ground truth) is very small (shown in Table 4-1) compared to the ROI output that is as large as 200x200. The drift in y direction will not significantly change the viewing result and this is also observed in the experiments.

4.6 ROI Detection in Compressed Domain

In some applications, the panoramic video has to be stored as digital video data. In other applications, it has to be delivered to a client, and thus, the ROI detection and virtual camera control has to be performed on the client-side. These videos are usually available as compressed video streams. A straightforward solution to this problem is to decompose the video and process it in the uncompressed domain, but it is not efficient. In the following, an efficient ROI detection method is introduced.

(a) A frame from a panoramic video.

(b)Confidence map.

(c)Moving part detection based on Confidence map.

(d)Manually segmented moving part

(e) Moving part detection of the object in the compressed domain after median filtering.

Figure 4-8. Detection of moving part for a frame in a panoramic Video.

Compressed domain video processing can achieve fast speeds. While Zhang et al. [104] and many others use compressed domain features for video data segmentation and indexing, very few efforts have been made to use them for detection purposes. An example of compressed domain face detection is given in [96]. The method proposed here is based on our previous work on motion activity detection [88] which is also discussed in chapter 3. The ROI is detected using the P frames in the MPEG stream. This includes extracting P frames from the video and detecting the ROI in a P frame. The detected ROI position is then propagated to neighboring frames by temporal up-sampling. Note that even though we discuss the algorithms based on recorded video, they can be applied to real-time encoded MPEG stream the same way.

4.6.1 No_MC Inter-Coded Macroblocks as Background

Of all the P frame macroblock types, Inter-coded No_MC macroblock is of interest to us. The inter-coded No_MC macroblock type is shown in the dashed region of Figure 2-4. From the analysis of chapter 2, we know that No_MC inter-coded macroblocks correspond to zero motion regions in a scene. Therefore it can be used to represent the background of the scene which has no motion.

4.6.2 Detection of the ROI Centroid in P Frames

As discussed in uncompressed domain ROI detection in section 4.5, the speaker is modeled as a point object corresponding to the centroid of the moving part of the body. The ROI output is a predetermined rectangular region that surrounds this point. Thus, the ROI detection basically detects the centroid of the moving region of the body that is in the foreground of the scene.

Spatial Sub-sampling and Up-sampling of P Frames

The MPEG motion compensation scheme borrows its strategy from traditional region-based optic flow estimation, even though the motion vectors it provides are not the same as optic flow. In the case of a video where there is only one moving object, this motion compensation information becomes especially important.

If the center of a macroblock is to represent the whole block, then a sub-sampled image of the original frame can be obtained. Since the macroblock size is 16x16, the height and width of the sub-sampled image are 1/16th of the original frame height and width respectively. If an estimation of the centroid of the ROI in the sub-sampled image is obtained, it can then be up-sampled, i.e. the estimated centroid position (x, y) can be scaled by 16 to get the estimation of the original frame. Sub-sampling tends to create aliasing effects when there are high frequency signals in the original image. That is why traditional motion estimation methods usually filter the images with a Gaussian filter before sub-sampling.

Detection of the ROI Centroid.

In a scene captured in panoramic video, the region where there is motion is considered to be the ROI. In the compressed domain processing, the non moving region is first detected using the P frame macroblock motion information. The ROI is detected by taking the complement of the non moving region.

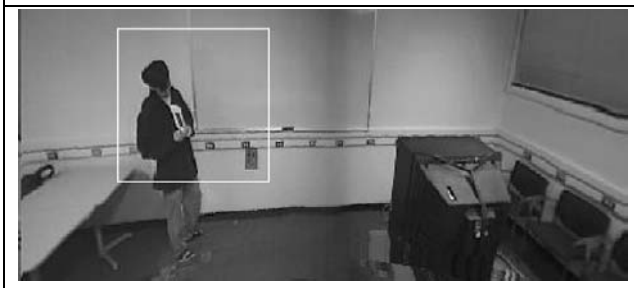
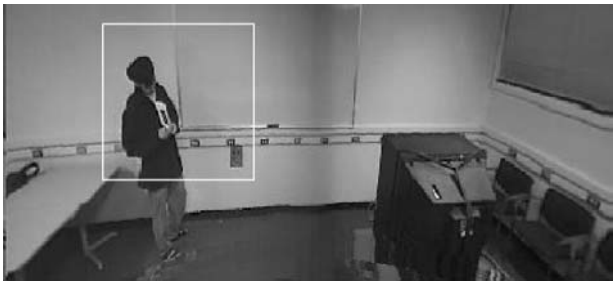
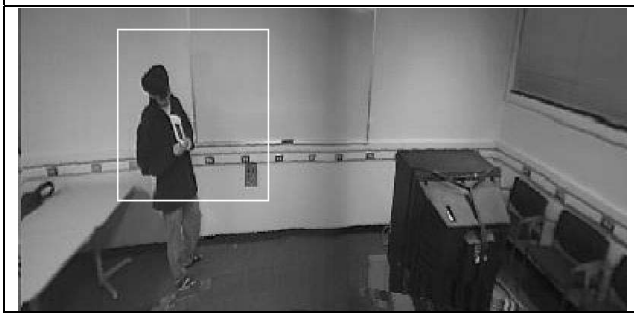
Frame	Frame Information
	(a) P Frame Frame Size: 800x352 ROI Size: 200x200 ROI Centroid (x,y) : (231, 125)
	(b) B Frame The ROI centroid of (a) is applied here.
	(b) B Frame The ROI centroid of (a) is applied here
	d) I Frame The ROI centroid of (a) is applied here.

Figure 4-9. Four consecutive frames in different frame types in an MPEG-2 video.

Figure 4-8(a) shows an example of a P frame in an MPEG-2 video. Figure 4-8(e) shows the foreground detection results based on No_MC coding information, where the white region is the non moving region, and the black macroblock is the moving region of the foreground. Since the region that covers the speaker's body is connected, a median filter is used to improve the detection result. The median filter used is of size 3x3. In practice, the upper body is more important than the lower body as ROI output. Therefore, if the ROI size is small, the detected ROI can be shifted upward to center it on the upper body.

After the foreground object is detected, the centroid of the object can be computed easily in the sub-sampled image domain. Since sub-sampling using the macroblocks scales the image to 1/16th of its original height and width, the computed result is then scaled by 16 times to get the estimation of the ROI centroid position in the original video frame. In theory, when the speaker's clothes do not contain much texture, the compressed-domain centroid estimate will be close to that of the uncompressed domain. However, as shown in Figure 4-8, after spatial up-sampling to the original video size, we find the detected motion region is larger than that from the uncompressed domain. Motion detection is generally more robust for macroblocks than for pixels because the larger macroblocks tend to average out noise.

Temporal Sub-sampling and Up-sampling Using the P Frame

In a typical video, the 29.97 frames per second frame rate is higher than is necessary for the ROI detection discussed here. Figure 4-9(a-d) show four consecutive frames in an MPEG video in a seminar room setting. Note that the frame-to-frame motion of the

speaker is quite small. If only the centroid of the ROI, which covers the speaker, is considered, it moves only several pixels in each direction. Therefore, even if the ROI centroid of Figure 4-9(a) is applied to the following frames in Figure 4-9(b) to Figure 4-9(d), there is no noticeable difference.

Therefore it is reasonable to use the P picture to sub-sample the video sequence first. After the ROI is detected in the P frames, it is then up-sampled to obtain ROI positions of neighboring I and B frames in the original video sequence. As shown in Figure 2-3, the P frames, numbered 3, 6, 9, and 12, sub-sample the displayed video sequence. Depending on the organization of the frame types, the distance between two P frames varies. However, it was found in the experiments that sub-sampling using P frames is very effective in video sequences coded as in Figure 2-3.

4.7 Tracking using a Kalman Filter

The detection of the ROI centroid coordinates is generally a noisy process. The noise may come from sub-sampling, and lighting changes. If the noise is assumed to be Gaussian, then it can be handled by using an extended Kalman filter. The centroid has a trace in 2D trajectory. The trajectory in the x direction can be modeled by the second-order Taylor series expansion of the form:

$$x(k+1) = x(k) + v_x(k)T + a_x(k)T^2/2 + h.o.t. \tag{4-3}$$

$$v_x(k) = a_x(k)T + h.o.t. \tag{4-4}$$

Where $x(k)$ is the centroid coordinate of the ROI in the x direction, $v_x(k)$ is the corresponding velocity, $a_x(k)$ is the corresponding acceleration, T is the time interval, and *h.o.t.* are higher order terms. Similarly, the same model applies to the trace in the y component of the trajectory. Combining these gives the centroid system model:

$$\mathbf{F}(k+1) = \mathbf{\Theta F}(k) + \mathbf{\Gamma w}(k) \tag{4-5}$$

Where $\mathbf{F}(k) = \left[x(k), y(k), v_x(k), v_y(k) \right]^t$, $y(k)$ is the centroid coordinate in the y direction, $v_y(k)$ is the corresponding velocity. In the above equation $\mathbf{w}(k)$ models the effect of the acceleration ($a_x(k), a_y(k)$ in equations (4-3) and (4-4)) as additive Gaussian noise.

$$\mathbf{\Theta} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{\Gamma} = \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix}$$

Higher order Taylor series expansions can be applied to the centroid system model, which would lead to higher model orders. However, in our experiments, we found it did not appreciably improve the results. Additionally, the system variables provide enough information for virtual camera control as discussed in the following section.

Since the speaker is modeled as a simple point, the location measurement can be modeled as:

$$\mathbf{Z}(k) = H\mathbf{F}(k) + \mathbf{n}(k) \tag{4-6}$$

Where $\mathbf{Z}(k)$ is the measurement, $\mathbf{n}(k)$ is the Gaussian measurement noise, and H is the measurement transfer function, in this case a scaling factor.

The covariance form of Kalman filtering is used to recursively update the prediction based on the innovation information at each step. The prediction at each update is output for further ROI virtual control purposes. The predicted or estimated variable used to control the recording process is $\hat{\mathbf{F}}(k) = [\hat{x}(k), \hat{y}(k), \hat{v}_x(k), \hat{v}_y(k)]^t$.

4.8 Virtual Camera Control

Kalman filtering reduces most of the noises inherent in the tracking estimate, and suffices for most purposes. However, if the tracking result is used to control the ROI window directly for video output, the quality of the output video is often jittery. The resulting motion is less smooth than that of a physical camera which has inertia due to its mass. Therefore an additional filtering step is taken to produce smooth and pleasant ROI video output.

The method proposed here for virtual camera control is based on the following observation. When an experienced camera operator records a lecture, if the speaker is not moving or moving only within a small region, the operator usually does not move the camera ("stabilization control"). When the speaker changes his position by a large distance, the operator must move the camera to catch up with the speaker ("transition

control"). After the speaker has been centered, the operator then follows further movement ("following control"). Accordingly, the virtual camera control operates in three similar regimes.

Stabilization control is based on the Kalman filter estimates of position and velocity. The initial centroid position is registered first, denoted as $\mathbf{X}_R(k) = [x_R(k), y_R(k)]^t$, where $x_R(k), y_R(k)$ correspond to its coordinates in the x direction and y direction respectively. Then at each frame, the estimated speed and position are checked. They can be obtained from $\hat{\mathbf{F}}(k)$ during the Kalman filter update process. If the following two conditions are satisfied, the virtual camera is fixed and the registered position is used as a position output. Firstly, the new position must be within a specified distance of the registered position in a given direction. Secondly, the estimated speed must be below a specified threshold at a given direction. Otherwise, the virtual camera control is changed to the "transition" regime. The stabilization control conditions can be formalized as:

$$\begin{aligned} &\mathbf{Y}(k) = \mathbf{X}_R(k) \\ &\textit{if} \quad |\hat{x}(k) - x_R(k)| < \sigma_1, \ |\hat{y}(k) - y_R(k)| < \sigma_2 \\ &\textit{and} \quad |\hat{v}_x(k)| < \sigma_3, \ |\hat{v}_y(k)| < \sigma_4 \end{aligned} \tag{4-7}$$

Where $\sigma_1, \sigma_2, \sigma_3, \textit{ and } \sigma_4$ are thresholds, and $\mathbf{Y}(k)$ is the ROI output. Recall that $x_R(k), y_R(k)$ are the registered coordinates.

In the transition regime, a lowpass filter is used to update the virtual camera location. For this purpose, a first order lowpass infinite impulse response (IIR) filter is used:

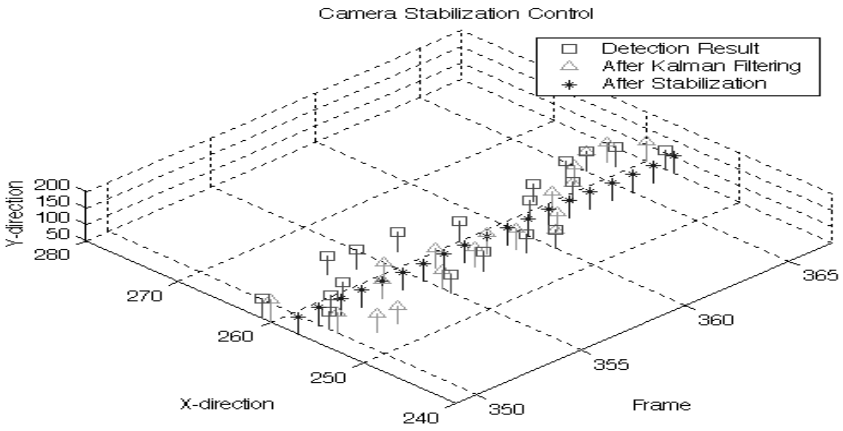
$$\mathbf{Y}(k+1) = \theta_1 \mathbf{Y}(k) + \theta_2 \hat{\mathbf{X}}(k) \tag{4-8}$$

Where $\theta_1 + \theta_2 = 1, \theta_1, \theta_2 > 0$, and $\hat{\mathbf{X}}(k) = [\hat{x}(k), \hat{y}(k)]^t$ is the estimated centroid from the Kalman filter, and serves as the input to the IIR filter. The virtual camera now follows $\mathbf{Y}(k)$, which is smoother than the Kalman filter output. It also helps to reduce the noise in the case of abrupt changes that the Kalman filter does not handle well. Experiments show that values of $\theta_1 = 0.8, \theta_2 = 0.2$ give a reasonable simulation of human camera operation.

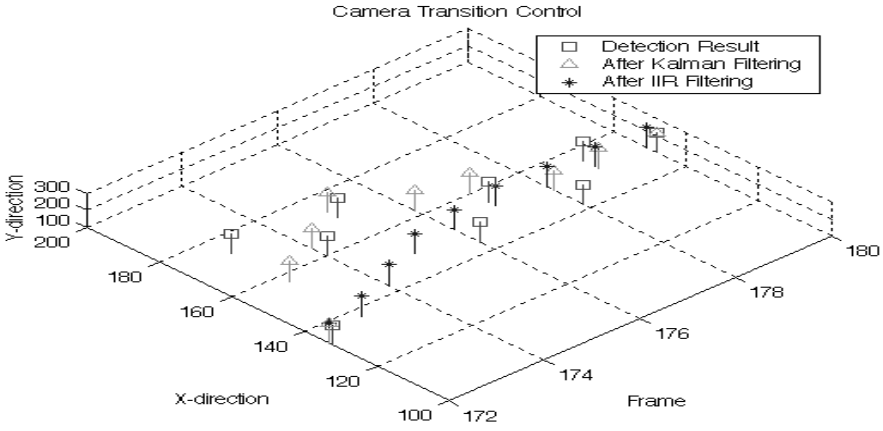
Since the IIR filter (4-8) tends to create delays in the output, the number of steps in the virtual camera "transition" stage is limited. After a certain time in the transition regime, for example 0.5 seconds, the camera control is switched to the "following" regime. Updating the ROI position directly from the Kalman filter output realizes this objective:

$$\mathbf{Y}(k) = \hat{\mathbf{X}}(k) \tag{4-9}$$

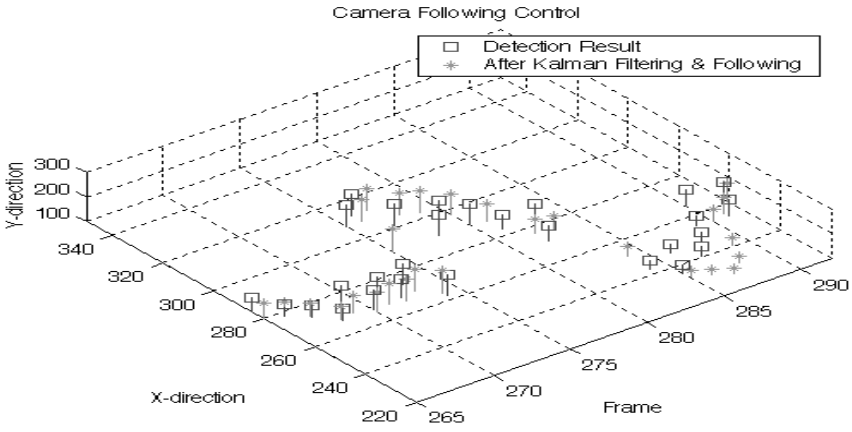
Note that this is equivalent to setting $\theta_1 = 0, \theta_2 = 1,$ in (4-8).



(a)

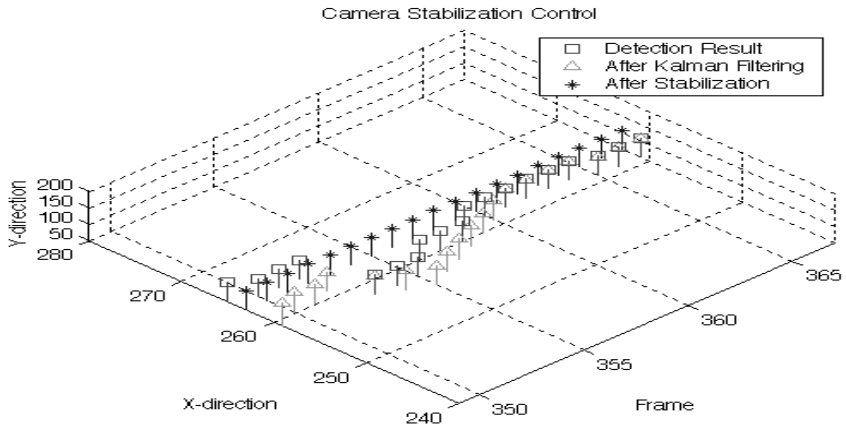


(b)

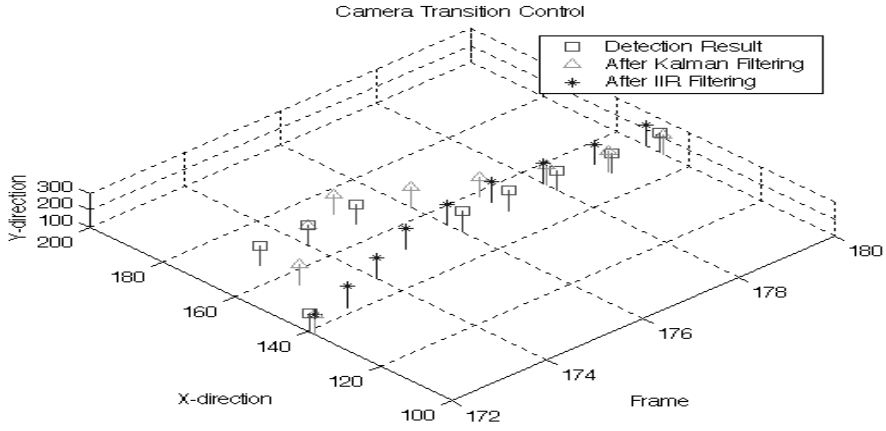


(c)

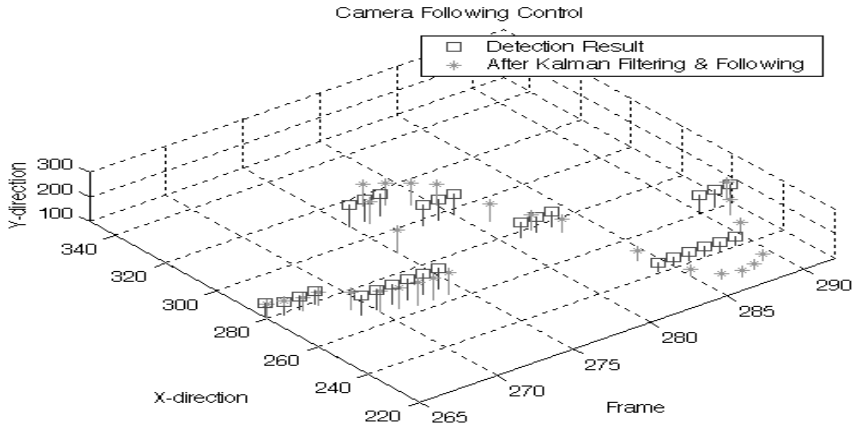
Figure 4-10. Simulation of three types of camera control in uncompressed domain.



(a)



(b)



(c)

Figure 4-11. Simulation of three types of camera control in compressed domain.

.

Figure 4-10 and Figure 4-11 show the results for three kinds of camera control in uncompressed and compressed domain for a video recorded. The view angle is chosen to emphasize the control process in the x direction. Figure 4-10(a) and Figure 4-11(a) show the "stabilization control" which fixes the virtual camera for a given small noise. The fixed positions are shown as "after stabilization." Figure 4-10(b) and Figure 4-11(b) show the "transition control" which uses IIR filter to smooth the output of Kalman filter. The smooth curves shown as "after IIR filtering" clearly display the transition process. Figure 4-10(c) and Figure 4-11(c) show "Following control" which steers the virtual camera according to Kalman filter output. No further virtual control is needed in this case.

The Kalman filter assumes environmental noise is Gaussian. It can handle lighting change and occlusion very well. But many noises are not Gaussian. For example, the projection display and the audience both can produce constant noise in fixed regions, as can be seen in Figure 4-1. This knowledge can be incorporated into the tracking system to improve performance, especially as the panoramic video cameras are fixed with respect to the background. Configuration parameters allow some part of these regions to be ignored. By offering this kind of flexibility, the tracking technology can be easily adapted to different environments

4.9 Experimental Results

Experimental evaluation of our tracking system was performed on panoramic video taken in a seminar room during seminars and presentations. The speaker moves around

at the front of the seminar room during a lecture. Panoramic video can be produced in real time at around 15 fps. To ensure frame rate, the panoramic video was stitched off-line and stored. Five video sequences up to 30 minutes were recorded and compressed to MPEG format using the following settings: 4MBit/s, 29.97fps, 800x352 pixels/frame.

A 200-frame video sequence showing a speaker moving from right to left in the seminar room was used for testing. The moving part of the body was manually segmented for each frame, and its centroid served as ground truth for ROI detection and tracking experiments. Though precise segmentation is difficult, the precision is not absolutely critical for virtual camera control. As long as the speaker is contained in the ROI output, the manual segmentation result is good enough as a reference ground truth.

First, ROI detection is processed for P frames from the test MPEG video and the corresponding frames from the uncompressed test video. The drifts (errors) between computed centroids and those of ground truth are calculated. As shown in Table 4-1, the drifts for uncompressed domain and compressed domain are quite close. Since ROI detection in uncompressed domain has a higher resolution than that (spatially sub-sampled using marcroblocks) in compressed domain, the uncompressed domain detection performs better in the x direction. On the other hand, as discussed in section 4.6, ROI detection in compressed domain is more stable even though it has a lower resolution. Therefore no significant difference can be seen between the performance of ROI detection for both compressed and uncompressed domain in y direction.

		ROI detection result for frames at P frame only		ROI tracking result for all the frames	
		Centroid Drift (pixel)	Drift Standard Deviation (pixel)	Centroid Drift (pixel)	Drift Standard Deviation (pixel)
Uncompressed Domain	*x* direction	9	5	8	5
	y direction	22	9	23	8
Compressed Domain	*x* direction	23	16	21	15
	y direction	18	9	16	8

Table 4-1. Statistics of ROI detection and tracking result.

The drifts between computed centroids after Kalman filtering and those of ground truth are also calculated. Kalman filtering generally improves the performance for both uncompressed domain and compressed domains except for the y direction in uncompressed domain. This can be explained the same way as discussed in section 4.5.2 about the centroid detection in uncompressed domain. Note that temporal up-sampling of P frame ROI detection result in general does not affect overall result after tracking. Since a much larger ROI window (200x200) is used as output, the average drifts and their standard deviations indicated in Table 4-1 in general ensures that the speaker is covered in the ROI output.

We note that the speaker need not exactly be at the center of the ROI video for most purposes. It is also observed that there is no single standard for when and by how much to move the virtual camera as far as the smoothness is concerned. Therefore, to benchmark the system we only determine whether the speaker is covered in the ROI

video output and whether the video is smooth. The results turn out to meet the requirements in general.

Software was developed to view ROI video in both compressed and uncompressed domains. The uncompressed-domain processing software is developed based on the Intel Image Processing Library. A video player was also developed based on the MPEG player distributed by the MPEG Software Simulation Group [112] to view compressed-domain output video. ROIs of size 200x200 are displayed to judge the tracking results. Since the upper body is more important in viewing, the ROI is shifted upward by a fixed number of pixels when playing. Experiments show that after initialization, the software controls the virtual camera to follow the speaker. We tested the programs by enabling and disabling the ROI processing. In terms of computation complexity, we observe that the delay created by virtual camera control is not noticeable. Since the size of the P frame macroblock information is insignificant compared to the entire video stream, the compressed domain ROI processing can be done extremely rapidly.

4.10 Summary

In this chapter a new method is presented for recording the region of interest in a scene. The FlyCam panoramic camera system is used to capture the scene. The proposed method integrates detection, tracking and recording processes, and simulates human camera control. This processing is done in both uncompressed and compressed domain. The entire process is fully automated and experimental results show that it is robust and fast enough for real time application.

Since the panoramic camera is stationary, the tracking information also provides indexing features for the video content. Spatial data about the lecture environment can be combined with the tracking information to provide a descriptive indexes about lectur activity. Since the region of interest is isolated from other objects in the scene, the recorded result may also be useful for object based coding, such as in MPEG-4. Other research possibilities include virtual camera control for multiple objects, synchronizing the ROI output with PowerPoint slides, analyzing speaker activity, or using the ROI image as a basis for gesture tracking or face recognition.

5 Recognition of Human Motion Activity

In a lecture, the speaker may walk, turn around, sit down or stand up. While chapter 4 provides a general architecture for capturing such events, automatic recognition of such events is the key for semantic indexing of video content. In this chapter we discuss the combination of virtual camera control with motion analysis for video event recognition. The issues discussed in this chapter include the use of virtual camera control parameters for activity recognition, and the development of a probabilistic model to model the motion parameter change, both spatially and temporally, for activity recognition.

5.1 Introduction

Analysis of typical activities such as a speaker walking, turning around, sitting down on a chair, and getting up from a chair in a classroom setting, is of interest in video indexing. It has many potential applications in the areas of indexing classroom and seminar presentations, and in human computer interfaces. While recognition of these activities is important, capturing of the regions of interest corresponding to these activities is essential to the success of the overall application. While most previous work discuss the two problems separately, we present a framework that integrates the capturing and recognition processes. We discuss how to use both virtual camera control parameters and motion parameters for the purpose of human activity recognition.

While different camera systems have been designed to capture the human motion, there is not much research on human motion activity recognition based on the capture

feature of the systems. Traditionally a static camera is used for capturing human activities [20], [91]. In this case, the person has to move within a small constrained area. There have been quite a few systems that use active cameras (for example, Sony's EVI camera), or combine it with other wide-angle cameras for seminar capturing [74]. However, these systems involve camera motion which is not helpful in the recognition process. As we have discussed in the previous chapter, the FlyCam [37] panoramic system fits well for large area human activity capturing. For the FlyCam system, we have designed virtual camera control methods [83] to output ROI video that covers the speaker. The advantage of this system is that it is automatic, and the speaker is always in the scene. There is no physical camera motion in the system and the virtual camera parameters are readily available for recognition purposes. Figure 4-1 shows an example of such capturing result.

5.2 Related Work

While it is possible to use the virtual camera control parameters for the recognition of some human activities such as walking toward left/right as discussed in section 5.4, for other activities such as bending body, turning body, we need motion based activity recognition. Motion-based recognition has been well studied in the literature for recognition of motion activities. A review of some of the early work on motion-based recognition can be found in Cedras and Shah [20] and Tsotsos [91]. Typically, feature points, region features, view appearance, and global motion fields are used in characterizing the motion information for recognition.

Some examples of feature-based tracking include Madabhushi and Aggarwal [59], Song et al. [78], and Wilson and Bobick [99]. In [99], the centroid of a hand is used as the feature for tracking. In general, automatic detection and tracking of feature points from a human body has proven to be difficult. Consequently, very few attempts have been made to recognize complex human activities based on feature points. Campbell and Bobick [19], Gavrila and Davis [38] discuss the recognition of activities by tracking tokens attached to parts of a human body. However, tokens are not always easily available in practice.

Considerable work has been done on tracking region features as well. Blake and Yacoob [9] use different motion parametric models of regions for recognizing facial expressions. Bregler [14] uses the Expectation-Maximization (EM) algorithm to segment the human body into regions, and then uses the Hidden Markov Model (HMM) to characterize the dynamic change of those regions. Region based methods have shown some promising results, but they rely on region segmentation. Like feature detection, segmentation is often a difficult problem, particularly when the background scene is complex.

Given the complexity of human body motion, techniques that do not require explicit image feature detection or segmentation are of much interest. Some gesture recognition work has been done based on stored views [28], [45] without motion estimation. In [28], Darrell and Pentland use dynamic warping to match grey level image sequence with leaned templates by correlation. As pointed out in [20], if optical flow can be reliably extracted, it should perform better. One obvious reason why optical flow performs better is that optical flow computation is not affected by background change.

For motion recognition without feature detection and region segmentation, some previous work has focused on global motion field. Among the early work is Polana and Nelson [70], wherein they propose temporal textures for activity recognition. Their initial experiments involve recognizing events such as water flow or the fluttering of leaves. They use first and second order statistical representations of optic flow. Davis and Bobick [30] use temporal templates for human movement recognition. Their method requires less computation, but is sensitive to variances in the movement. Little and Boyd [55] use the moments of moving points to represent the optic flow for the purpose of periodic human gait recognition. Hoey and Little [50]use the Zernike moment of optic flow to represent motion. Their focus is on facial expression recognition and lip reading.

Since actions and gestures have typical temporal pattern, some temporal models have been proposed for recognition. Polana and Nelson [70] use temporal textures of optical flow for the recognition of simple motion like ripples of water. Davis and Bobick [30] use temporal templates to recognize human actions. However, as founded in their experiment, the temporal templates usually do not work well when the activities have similar sub-process such as group 3 activities introduced in section 5.3. While Darrell and Pentland use dynamic warping for the recognition of some gestures, it is recognized that Hidden Markov Model (HMM) works better in handling the statistic feature of actions [69]. Therefore, we use it for temporal characterization of human motion activity.

In signal processing, HMM has been extensively used for the recognition of spoken words independent of their duration and variation in pronunciation. Similarly, human actions have changeable duration and varied gestures. Therefore, Yamato [102] and many others use HMM for the recognition of human actions and gestures. Even though the human actions and gestures are simpler than the human motion activities discussed in this chapter, similar techniques can also be applied to human motion activities.

HMM characterizes temporal sequence using a doubly stochastic process, a probabilistic network with hidden states which are observable. At each time instance or frame, a hidden state is observed. The hidden states have an initial distribution. The transition between the states is controlled by a transition matrix. In speech recognition, one unit of speech is represented using a HMM. In human action recognition, one action can be represented using a HMM. In speech recognition the observable states take the values of linear prediction cepstrum coefficients. In action recognition, the observable states can take the features of an image sequence that can be computed based on geometric moments, Zernike moments, etc. In the learning phase, the model parameters of HMM are modified so that the model describes the temporal dynamic of dataset optimally. It involves the use of expectation-maximization (EM) procedure. To recognize a given action, the features of image sequence can be tested over the set of trained HMMs in order to decide which action it belongs to. The probability of the action being produced by each HMM is evaluated using the Viterbi algorithm [71].

In section 5.5.3, we will provide details of the HMM modeling. Other related work on HMM includes [12], [76] and [79]. The references here are by no means complete, and

we refer to Pavlovic et al. [69] for a review of temporal modeling, in particular application to gesture recognition.

Our proposed method integrates the capturing and recognition processes. The virtual camera control parameters are used for the recognition of activities such as walking, and the motion parameters of each frame are used for the recognition of other activities such as turning around, sitting down, and getting up. Similar to those using global motion fields that do not require image feature tracking or segmentation, we introduce a multivariate Gaussian model to represent the likelihood of the motion parameters. The temporal change of the likelihood is characterized using a HMM for activity recognition. Motion parameter based recognition of activity is then posed as a maximum likelihood parameter estimation problem. The virtual camera control parameters and HMM are designed to work on different types of activities. Experimental results show that the method works well in recognizing such complex human body activities.

5.3 Panoramic Capturing and Recognition of Human Motion Activities

The FlyCam panoramic system [37] described in Chapter 4 is used to capture the speaker. The camera system is fixed and covers all of the area where the speaker activities take place, and produces real time panoramic video output. While we can compress the panoramic video first and extract ROI video from compressed domain for later activity recognition [83], here we choose to extract ROI video in real time [83]

Group	Term	Description
Group One	WL	walking toward left
	wr	walking toward left
Group Two	l2f	turning of the body from left to front
	f2l	turning of the body from front to left
	f2r	turning of the body from front to right
	r2f	turning of the body from right to front
Group Three	su	standing up
	sd	sitting down
	bu	starting to sit down but returning to the standing position without sitting down
	bd	starting to get up but returning to the sitting position without getting up

Table 5-1. Ten types of activities for recognition.

from the panoramic video. By doing this, we can avoid storing extra large amount of redundant data outside the ROI area in the panoramic video and still do not lose any information about the activities. Figure 5-1 shows the general system architecture for activity capturing and recognition. The motion parameters computed from ROI video output and the associated virtual camera control parameters are used for activity recognition.

Our experiments consist of ten activities as shown in Table 5-1. We separate these activities into three groups.

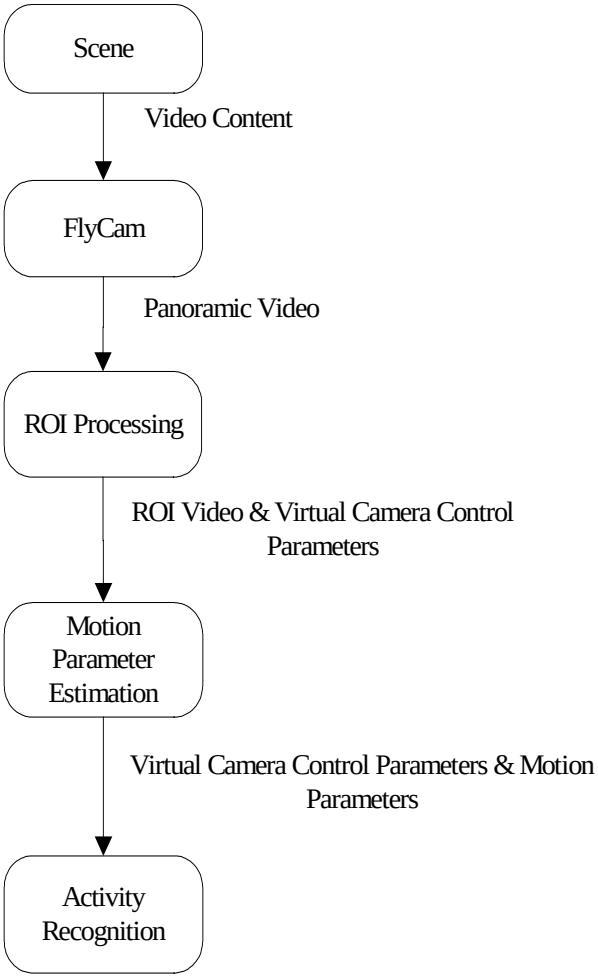


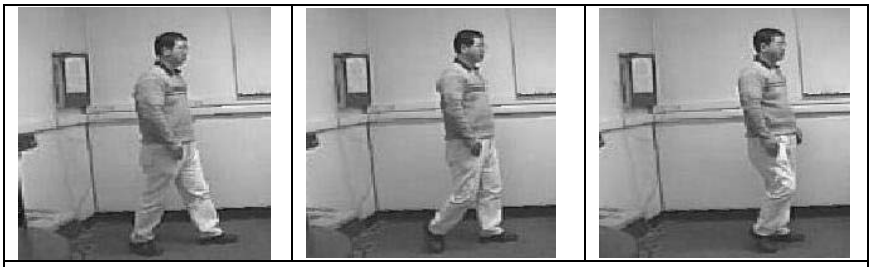
Figure 5-1. General system architecture for activity capturing and recognition.

In the first group, we have: walking toward left ("wl") and walking toward right ("wr"). In the second group, we have: turning of the body from left to front ("l2f"), front to left ("f2l"), front to right ("f2r") and right to front ("r2f"). In the third group we have: standing up ("su"), sitting down ("sd"), starting to sit down but returning to the standing position without sitting down ("bu"), and starting to get up (from a sitting position) but returning to the sitting position without getting up ("bd"). The third group is designed in such a way that the sequences have similar sub-processes. For example,

the ‘sd’ sequence has similar sub-process as that of ‘bd’ sequence. In the experimental results we show why HMM works well for these activities. Figure 5-2 shows representative frames (r-frames) from ROI video sequences of different kinds of human motion activities.

The speaker is modeled as a point object corresponding to the moving region of the body. The ROI output is a predetermined rectangular region that surrounds this point. Thus, the ROI basically tracks the centroid of the body’s moving region. As discussed in the previous chapter, we have a simple centroid model $\mathbf{F}(k) = \left[x(k), y(k), v_x(k), v_y(k) \right]^t$, where $x(k), y(k),$ are the positions of the centroid, and $v_x(k), v_y(k)$ are the velocities of the centroid in x and y direction respectively. The ROI detection results are processed through a Kalman filter.

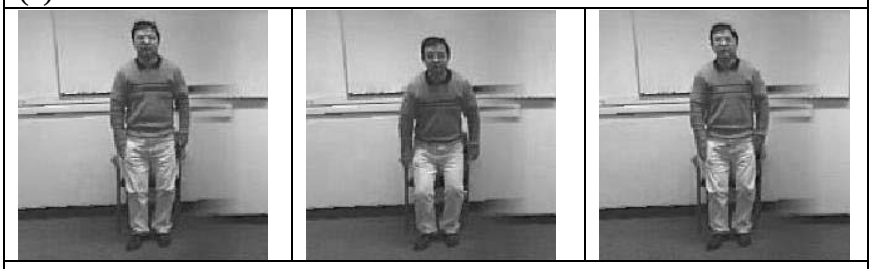
The Kalman filter output $\hat{\mathbf{F}}(k) = [\hat{x}(k), \hat{y}(k), \hat{v}_x(k), \hat{v}_y(k)]^t$ is then used to steer a virtual camera to create smooth ROI video output. From the discussion in chapter 4, the virtual camera control has three regimes. When the speaker is motionless or moving only in a small region, ROI is kept at the same position (stabilization control). When the speaker changes his position by a large distance, an infinite impulse response (IIR) filter is used to steer ROI to catch up with the speaker (transition control). After the speaker has been centered, ROI is changed according to the estimate $[\hat{x}(k), \hat{y}(k)]^t$ (following control). In the following section, we use the these virtual camera control information for the recognition of the group 1 activities.



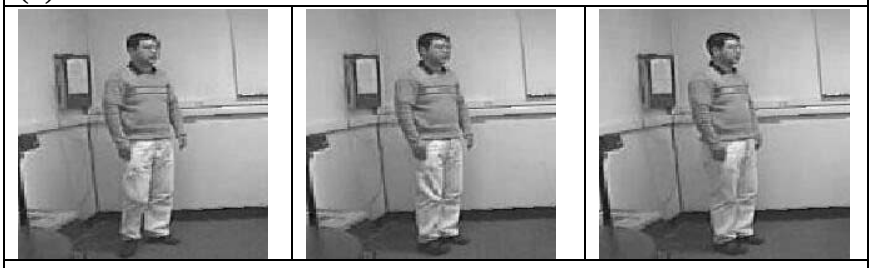
(a)Walking toward right

(b)Walking toward left

(c)“bd”



(d)“bu”



(e)“f2l”

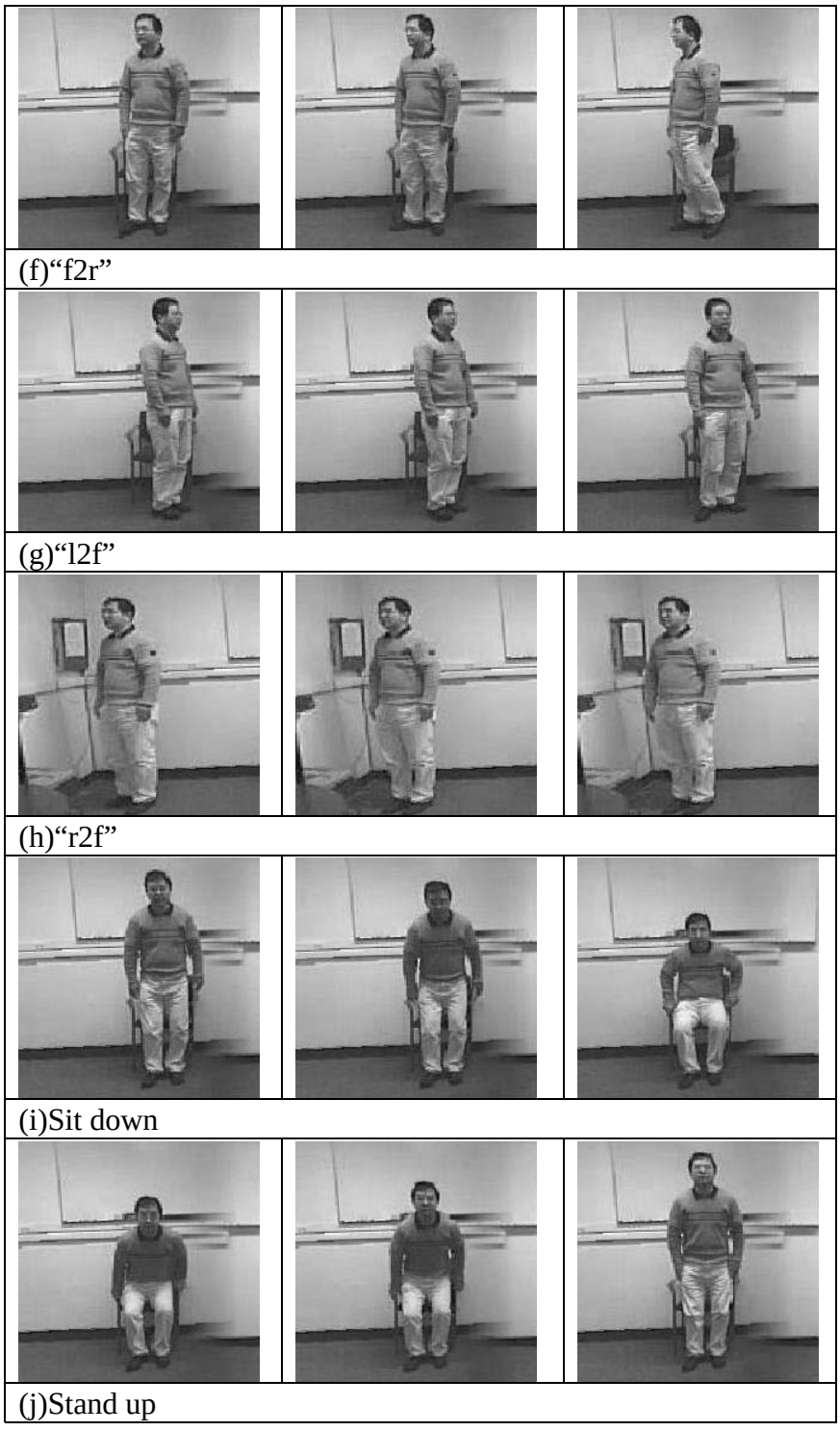


Figure 5-2 Representative frames (R-frames) of different human activities.

5.4 Recognition Based on Virtual Camera Parameters

Further observation of the activities described in the last section show that the first group of activities basically correspond to the virtual camera control in the "transition control" and "following control" regimes in x direction, while the second and the third group correspond to the "stabilization control" in x direction. Figure 5-2(a),(b) show representative frames of the walking sequences. Since walking is a periodic process, previous work has focused on modeling the shape of motion [55] and periodicity of the motion [25]. However, from above observation, we conclude that a decision on pattern of walking activity can be made if virtual camera control process falls into the categories of "transition control" and "following control" in the x direction. For others we have "stabilization control" in the x direction.

5.5 Human Motion Activity Recognition Based on Motion Parameters

The second and the third group of activities correspond to virtual camera motions that are not consistent in one direction. It is not straightforward to do activity recognition directly based on the virtual camera parameters as discussed in section 5.4. Therefore, we propose to use a probabilistic model to characterize these types of activities.

5.5.1 Motion Parameter Estimation

The first step in activity detection is motion estimation. Here we use a model-based approach. Model-based motion estimation techniques have been used in a variety of

vision research topics. In the area of 3D reconstruction, for example, the projective parametric model is often used. In [14], the affine motion model is used for body part segmentation, and in [9], both affine and planar region motion parameters are used for facial motion recognition.

In chapter 2 we have introduced model-based motion estimation method proposed by Bergen et al. [7]. While projective, planar, affine model can be used for motion estimation, our interest is in the affine model. Affine model usually applies when the distance between the object surfaces and the camera is large. It is formulated as (2-11):

$$\mathbf{V} = \mathbf{U}\boldsymbol{\kappa}, \text{ where } \mathbf{U} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix}, \text{ and } \boldsymbol{\kappa} = [\kappa_1, \kappa_2, \kappa_3, \kappa_4, \kappa_5, \kappa_6]^t$$

Note that κ_1, κ_4 correspond to the translation, and $\kappa_2, \kappa_3, \kappa_5, \kappa_6$ correspond to the deformation of the surface. If we ignore $\kappa_2, \kappa_3, \kappa_5, \kappa_6$ of $\boldsymbol{\kappa}$, then we get $\mathbf{V} = (\kappa_1, \kappa_4)$ which is the traditional optic flow vector.

Figure 5-3 shows few frames from a video sequence where the motion corresponds to a person standing up from an initial sitting position. Figure 5-3(d) shows the object inside of the window drawn in Figure 5-3(b). The motion inside this window is of interest. Figure 5-3(e)-(f) show the corresponding motion along the x and y directions, respectively, for this region of interest. In computing this motion, we smooth the motion vectors during the estimation process. Note that it is generally not necessary to use the motion parameters of the whole ROI video frame for activity

recognition. Instead, an even smaller region that covers the object, called object window, is chosen in our experiments.

5.5.2 Representation of Motion Parameters

Consider a motion parameter $\mathbf{z} = (z_1, z_2, ...z_d)^t$ computed at each pixel location. $\mathbf{z}$ could be the 6D affine parameter set as described in the previous section or a 2D optic flow vector. These parameter values are then organized into a vector by row scanning the image. Let ℓ be the number of pixels in an object window in a frame (ordered according to a row scan). Let

$$\mathbf{Z} = (z_1^1, z_1^2 .....z_1^\ell, z_2^1, z_2^2 .....z_{2,}^\ell .....z_d^1, z_d^2 .....z_d^\ell)^t \tag{5-1}$$

Note that $\mathbf{Z}$ is a $d \times S$ dimensional vector. We model $\mathbf{Z}$ as a multivariate Gaussian. Let the mean of this Gaussian be $\mathbf{m}$ and covariance $\mathbf{\Sigma}$. Then, given $\mathbf{Z}$ from an observation class $\mathbf{\Omega}$, we can write the conditional probability $P(\mathbf{Z} \mid \mathbf{\Omega})$ as:

$$P(\mathbf{Z} \mid \Omega) = \frac{\exp(-\frac{1}{2}(\mathbf{Z}-\mathbf{m})^{\mathbf{T}}\mathbf{Q}^{-1}(\mathbf{Z}-\mathbf{m}))}{(2\pi)^N \left|\mathbf{\Sigma}\right|^{1/2}} \tag{5-2}$$

This approach to modeling the observation is similar to the work in [62] where the observation vector is the image intensity and the application is object recognition. In the following discussion, we refer to $\mathbf{Z}$ as the *parametric motion object* (PMO).

The Karhunen-Loeve transform (KLT) is used to simplify the computation of (5-2).

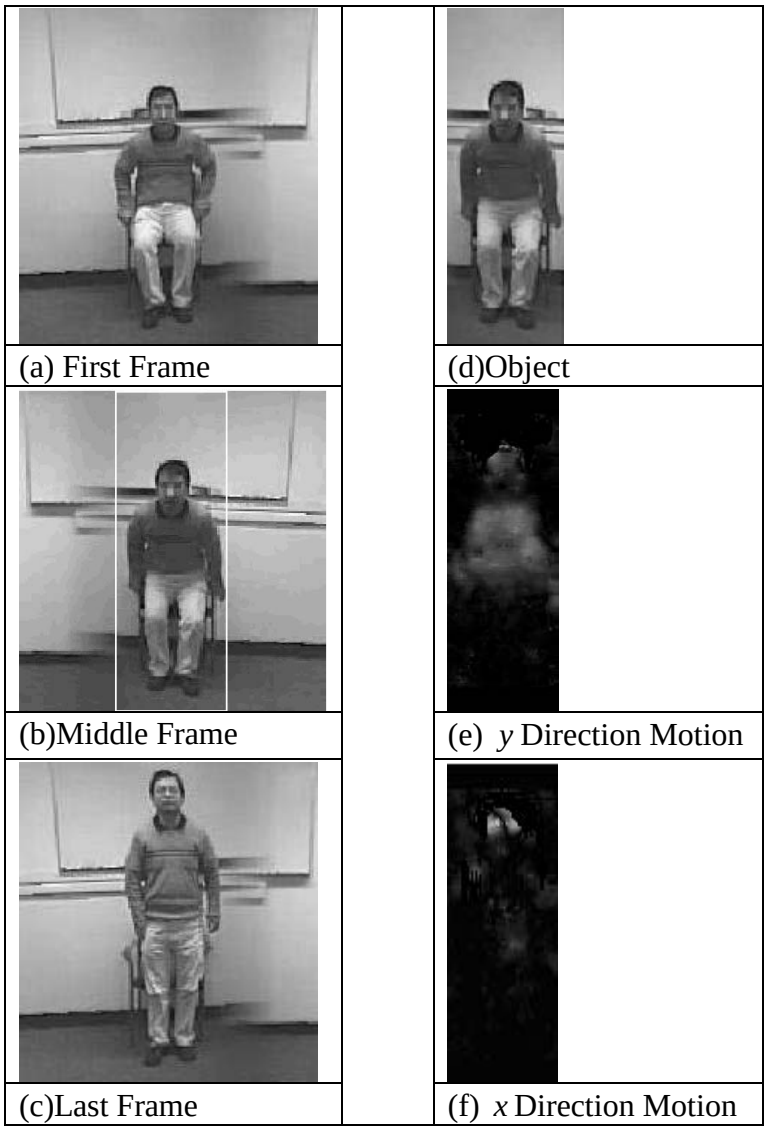


Figure 5-3. Motion estimation: optic flow visualized as a normalized image. (a)-(c) show the video frames from a ROI sequence corresponding to a person standing up from an initial sitting position. (d) shows the object window in (b). (e) and (f) show the optic flow images of (d) along the y and x directions, respectively.

Let $\tilde{\mathbf{Z}} = \mathbf{Z} - \mathbf{m}$. The covariance matrix can be decomposed as:

$$\mathbf{\Sigma} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{\mathbf{T}} \tag{5-3}$$

where the columns of $\mathbf{\Phi}$ are the orthonormal eigenvectors of $\mathbf{\Sigma}$, and $\mathbf{\Lambda}$ corresponds to the diagonal eigenvalue matrix of $\mathbf{\Sigma}$. Let

$$\mathbf{S} = \mathbf{\Phi}^{\mathbf{T}}\tilde{\mathbf{Z}} \tag{5-4}$$

Following [62] , we can compute (5-2) using

$$P(\mathbf{Z}\,|\,\Omega) = P_{P}(\mathbf{Z}\,|\,\Omega)P_{\bar{P}}(\mathbf{Z}\,|\,\Omega) \tag{5-5}$$

where
$$P_{p}(\mathbf{Z}\,|\,\Omega) = \left[\frac{\exp(-\frac{1}{2}\sum_{1}^{M} s_i^2/\lambda_i)}{(2\pi)^{M/2}\prod_{1}^{M}\lambda_i^{1/2}}\right], \text{ and} \tag{5-6}$$

$$P_{\bar{P}}(\mathbf{Z}\,|\,\Omega) = \left[\frac{\exp(-\frac{1}{2}\sum_{M+1}^{N} s_i^2/\lambda_i)}{(2\pi)^{(N-M)/2}\prod_{M+1}^{N}\lambda_i^{1/2}}\right] \tag{5-7}$$

M is the number of principal components, s_i is the i-th component of $\mathbf{S}$, λ_i is the i-th eigenvalue of $\mathbf{\Sigma}$.

The first part of (5-5) represents the principal subspace of the object. Note that the basic idea of using a subspace for computing the object features for recognition has been used by many researchers (see Turk and Pentland [93] for face recognition, and Hoey and Little [50] for motion recognition). As noted in [62] and also observed in our

experiments, the second component $P_{\bar{P}}(\mathbf{Z}|\Omega)$, which represents the complementary orthogonal subspace of the principal component, plays an important role in the recognition process.

Since direct computation of $P_{\bar{P}}(\mathbf{Z}|\Omega)$ is too expensive, in practice we use the following approximation suggested by Moghaddam and Pentland [62]:

$$\hat{P}_{\bar{P}}(\mathbf{Z}|\Omega) \approx \left[\frac{\exp(-\frac{1}{2}\sum_{M+1}^{N} s_i^2 / 2\rho)}{(2\pi\rho)^{(N-M)/2}} \right] \tag{5-8}$$

ρ can be obtained my minimizing a suitable cost function $J(\rho)$. This cost function should be Kullback-Leibler divergence [24] between $P_{\bar{P}}(\mathbf{Z}|\Omega)$ and its estimate $\hat{P}_{\bar{P}}(\mathbf{Z}|\Omega)$ from an information-theoretic point of view. It can be formulated as:

$$J(\rho) = \int P_{\bar{P}}(\mathbf{Z}|\Omega) \log \frac{P_{\bar{P}}(\mathbf{Z}|\Omega)}{\hat{P}_{\bar{P}}(\mathbf{Z}|\Omega)} d\mathbf{Z} = E\left[\log \frac{P_{\bar{P}}(\mathbf{Z}|\Omega)}{\hat{P}_{\bar{P}}(\mathbf{Z}|\Omega)} \right] \tag{5-9}$$

Note that $E[s_i^2] = \lambda_i$, so we can get a simple form of $J(\rho)$ as:

$$J(\rho) = \frac{1}{2}\sum_{i=M+1}^{N} \left[\frac{\lambda_i}{\rho} - 1 + \log \frac{\rho}{\lambda_i} \right] \tag{5-10}$$

The optimal weight of ρ^* can then be found by solving the equation $\frac{\partial J}{\partial \rho} = 0$, which yields:

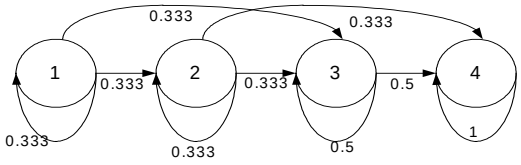
$$\rho^* = \frac{1}{N-M} \sum_{M+1}^{N} \lambda_i \,. \tag{5-11}$$

5.5.3 Modeling Human Motion Activity Using HMM

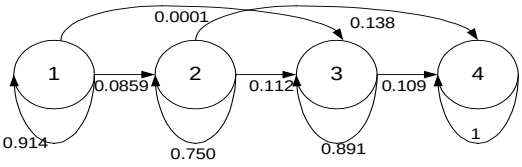
To model the temporal pattern of motion, we use HMM. A generic HMM [72] can be represented as $\Psi = \{\Xi, A, B, \pi\}$, where $\Xi = \{q_1, q_2, ...q_{N'}\}$ denotes the N' possible states, $A=\{a_{ij}\}$ denotes the transition probabilities between the hidden states, $B=\{b_j(.)\}$ denotes the observation symbol probability corresponding to the state j, and π denotes the initial state distribution. Given a video sequence $O = O_1, O_2, ...O_N$, where N is the length of the sequence, we then want to find one model Ψ_i from a given dictionary $\{\Psi_1, \Psi_2, ...\Psi_c\}$ which maximizes the likelihood $P(O/\Psi)$.

Model

We choose continuous density HMM for activity recognition here. Figure 5-4 gives an example of an HMM model before and after training. The number of states is empirically determined to be four and we observed that increasing the number of states did not result in any performance gains on our initial data sets. Therefore, hidden states are $\Xi = \{1, 2, 3, 4\}$.



(a). HMM for 'bd' before training



(b). HMM for 'bd' after training

Figure 5-4. An example HMM for the 'bd' sequence.

The state observation model B is based on the Gaussian representation of motion parameters as discussed in section 5.5.2. We can obtain $\mathbf{m}, \boldsymbol{\Sigma}$ and consequently $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ from the training data. Then (5-5) can be used to compute the probability for a given frame based on a typical hidden state model. The state transition is initially set to uniform, i.e.

$$A=\begin{pmatrix} 0.333 & 0.333 & 0.333 & 0 \\ 0 & 0.333 & 0.333 & 0.333 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The state transition matrix after training becomes:

$$A=\begin{pmatrix} 0.914 & 0.0859 & 0.0001 & 0 \\ 0 & 0.750 & 0.112 & 0.138 \\ 0 & 0 & 0.891 & 0.109 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The activity always starts at the first hidden state. Therefore, we have $\pi_1 = p(q_1 = 1) = 1$, and $\pi_i = p(q_i = i) = 0, \; for \; i \neq 1$.

Another observation is that the HMM here has a typical left to right graph structure. The left to right type of HMM is a special class of HMMs. This type of HMM has an additional property that the state index is non-decreasing as the time increases, i.e.

$$a_{ij} = 0, \quad \text{for} \quad i > j \tag{5-12}$$

As shown in Figure 5-4, the left to right HMM can effectively model the time-dependent property in an activity sequence. In all of the models trained here, every state has a self-transition loop.

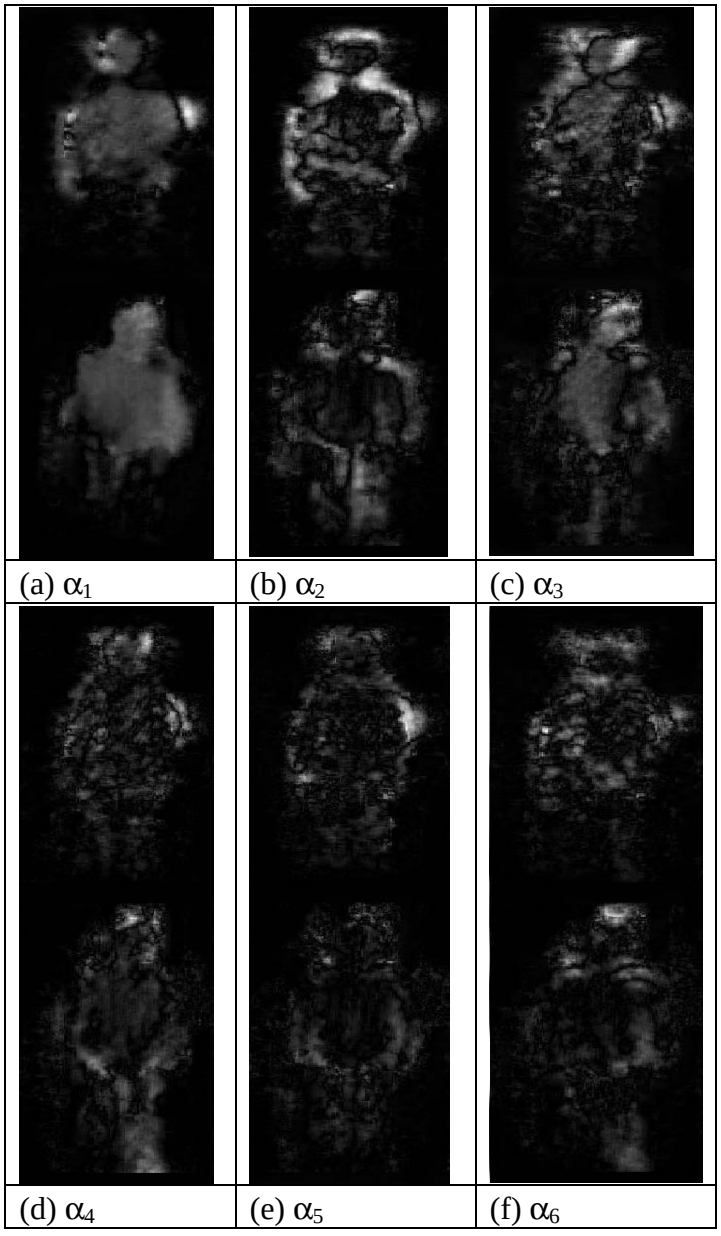
Training

The first step in HMM training is to obtain the observation model B. For simplicity, we only use one multivariate Gaussian instead of the mixture of multivariate Gaussian to model each state of the HMM. This is chosen based on the fact that the motion field is stable within a given small interval and therefore one Gaussian model is enough to characterize the distribution of the motion parameters at a given pixel.

A good initialization of B can be obtained in a number of ways, as discussed in [72]. They include 1) manual segmentation of the observed video sequences into states and averaging the observations within states, 2) maximum likelihood segmentation of observations and averaging the observations, 3) segmental k-means segmentation with clustering. We combine 1) and 3) for the training of B. Intuitively, we can divide the

sequence into temporal segments where each segment corresponds to a state. Based on the above assumption, we uniformly segment each training sequence into 4 segments before clustering. Each segment is assigned a state number that is the same as its segment order in the sequence. This provides a good initialization clustering of the states. The position of PMO of each frame is manually selected around the moving subject. Then we can compute $\mathbf{m}, \boldsymbol{\Sigma}$ and consequently $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ for each state. After this step, we can follow the conventional K-means clustering method to iteratively classify the frames based on its likelihood computed using (5-5). After the initial parameters of B is obtained, it can be used for maximum likelihood segmentation of sequences. The segmented sequences can be used for segmental K-means clustering.

Figure 5-5 shows the first 6 eigenvectors of $\boldsymbol{\Sigma}$ for state 1 of 'sd' activity. In computing this, we only use the optic flow components, and we put y direction first and x direction second in the corresponding PMO structure. Figure 5-6 shows the normalized likelihood of each frame from one of the 'sd' sequences based on four different state models obtained from training data. The transition from one state to the next is clearly evident. The likelihood takes highest value for the first state model in the beginning (frames 1 to 5). Then, the highest likelihood goes to the second state model (frames 6 to 11). Later, the highest likelihood goes to the third state model (frames 12 to 16). Finally, the highest likelihood goes to the fourth state model.



(a) α_1 (b) α_2 (c) α_3

(d) α_4 (e) α_5 (f) α_6

Figure 5-5. The first 6 eigenvectors for state 1 of the "*sd*" activity using optical flow PMO. The absolute value of each pixel is scaled to 0-255.

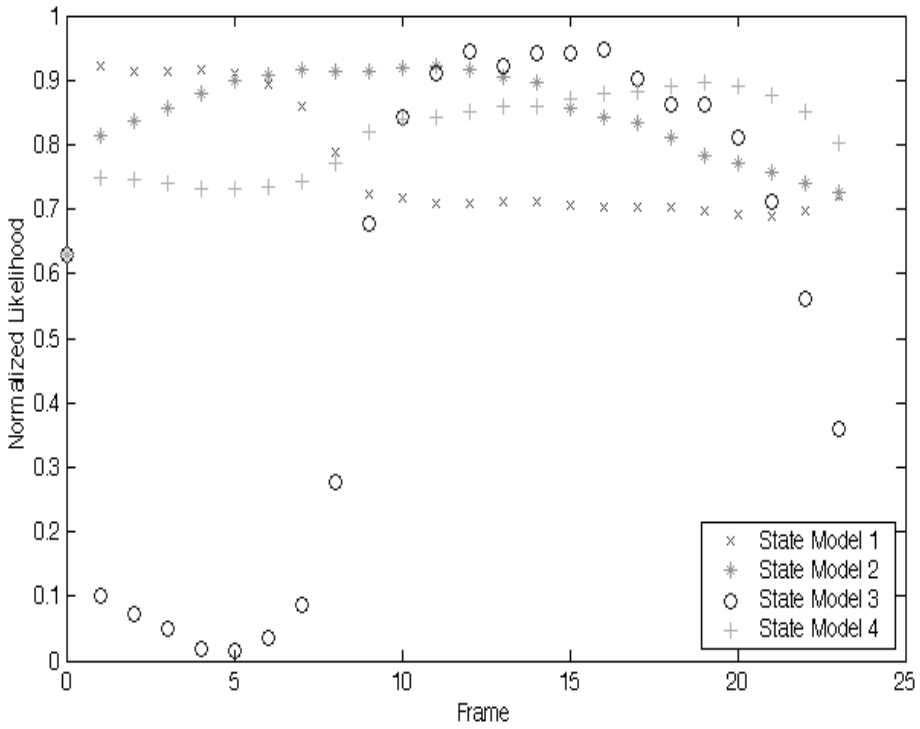


Figure 5-6. Normalized (0-1) likelihood of one *sd* sequence computed based on four different state models corresponding to the *sd* activity. Each curve corresponds to one state model.

At this stage we have the observation model B with $\mathbf{m}, \boldsymbol{\Sigma}$ computed. The likelihood of the observation for a given frame can be computed based on (5-5) and the model parameters $\mathbf{m}$ and $\boldsymbol{\Sigma}$. The next step is to obtain the state transition matrix A. A is initialized in an uniform way, as shown in Figure 5-4(a). For a typical four state HMM model, while the intermediate transitions such as those from state 2 to state 3 and state 2 to sate 4 are possible, it is generally not possible for transition from state 1 to state 4. This is because the whole activity sequence is a process that smoothly progresses from the beginning to the end. A transition that skips all the intermediate states is therefore

not very likely. Therefore, during initialization A, we set $a_{14} = 0$. This significantly simplifies the model computation. The update of A is based on the EM algorithm [72]. Note that we do not need to compute π as in our model we always start in state 1. The trained HMM structure for the 'bd' activity is shown in Figure 5-4(b).

Recognition

Suppose we have a test video sequence $O = O_1 O_2 O_3 ... O_N$, where O_i represents the i-th frame. If the trained HMM model parameters are $\Psi_i = \{\Xi_i, A_i, B_i, \pi_i\}$, the problem of recognizing the sequence is equivalent to deciding from which of the trained models the sequence O is observed. We first compute the motion parameters of each frame. A window of the same size as training PMO is moved around the video frame to find the position where the maximum likelihood is obtained based on (5-5). The likelihood is $b_j(.)$ for a given state model j. A Kalman filter can be applied to track the window in order to speed the search process.

It is possible to compute $P(O/\Psi_i)$ based on each HMM parameters, which is the probability of the observation sequence O given the HMM model parameters for each of the trained activity class. According to the maximum likelihood principal, we pick the activity model Ψ_{i*} that maximizes the probability,

$$i* = \arg\max_{1 \le i \le c} [P(O/\Psi_i)] \tag{5-13}$$

where c is the number of trained classes. The computation of the probability based the given models involves the well-known Viterbi algorithm [71].

5.6 Experimental Results

The FlyCam system is used to capture the speaker activities and produces panoramic video of size 800x352 in pixel resolution. The output ROI window size is 200x200. The size of object widow for motion parameter based recognition is 64x160. We collect 20 sequences for each activity. Each sequence contains 20 to 30 frames. Half of the video sequences are used for training, while the other half are used for evaluation. For simplicity, the subjects are asked not to wave hands or make other gestures while recording the video. They also pause for a while between two consecutive activities. This creates artificial zero motion frames in the video, and thus simplifies the segmentation of activities. Therefore, it makes the recognition similar to isolated-word instead of connect-word recognition in speech processing [72].

In recognizing the activities, the ROI sequences are first segmented into smaller sequences containing one single activity each based on the temporal position of zero motion frames. The recognition of walking activity is processed using the virtual camera control parameters first. For the rest of the video sequences, motion parameter based recognition method is used. The affine parameters and optic flow vectors of a ROI frame are computed first based on [7] to obtain the PMO. The PMOs are normalized to a zero-mean unit-norm.

Table 5-2 summarizes recognition results. 5% step size is used for accuracy computation.

We first use virtual camera control parameters to distinguish group 1 from group 2 and group 3 activities. Then the directions of virtual camera control parameters for group 1 activities are used to distinguish walking toward right and left. As expected, group 1 activity recognition is stable.

The activities in group 2 are similar to those presented in [30] which use temporal templates for recognition. As expected, the recognition based on HMM model can achieve high recognition rate of 90% for these activities. While statistical results for the recognition in [30] is not given, we believe HMM can handle simple activities as well as or even better than temporal templates.

Group 3 activities share similar sub-processes, making their estimation more difficult. Also, group 3 activities are more complex. For example, the first state of "su" is the same as the first state of "bd". In addition, the transitions in "bu" and "bd" are also more complicated than those in group 2. As pointed in [30], temporal templates based method usually does not perform well for these kinds of activities. However, we can still achieve a high recognition rate (85%) by using HMM based on motion parameters.

It is surprising to note that the optic flow based modeling performs better than the more informative affine model. One possible explanation is that the affine motion parameters are more sensitive than the optic flow, and the variations are not well

Activity			**Group1**	**Group2**	**Group3**
Virtual Camera Control Parameters			100%	----	----
Optic Flow	PCA	6 bases	----	75%	60%
		10 bases	----	80%	75%
	PMO	6 bases	----	90%	80%
		10 bases	----	90%	85%
Affine Model	PCA	6 bases	----	40%	30%
		10 bases	----	45%	40%
	PMO	6 bases	----	50%	50%
		10 bases	----	60%	50%

Table 5-2. Experimental results on the test sequences.

captured within the four state HMM used in our experiments. Also, the training dataset is perhaps too small for accurate model training for the affine model.

In general, larger dimensions of principal subspaces perform better than smaller ones, but we did not observe significant differences here between six and ten dimensions. PCA based method is also tested for comparison. Note that for the PCA based method, $P(\mathbf{Z}|\Omega)$ is approximated using $P_P(\mathbf{Z}|\Omega)$. It can be seen from experiment that in general PMO method outperforms the PCA method.

5.7 Summary

In most previous work, the tasks of capturing and recognition of human motion activities are separated. In this chapter we present an approach that integrates camera capture information and motion information for human motion activity recognition. We discuss the use of virtual camera control parameters for the recognition of some common activities, and the motion parameters of each frame for the recognition of other activities. Experimental results show that the approach works well in recognizing such complex human motion activities.

Model selection and representation are important issues in activity recognition. While significant amount of research work has been made on hand, lip, face motion pattern recognition, large human body activity remains challenging. The inaccuracies in motion estimation, a critical first step in this process, further complicates the problem. Our continuing investigation includes better ways of estimating the model parameters. Future research efforts could involve a more general Bayesian Network [18], [63] for even more complex human activities.

6 Conclusions and Future Directions

In this dissertation we introduced motion activity descriptors for both low level and high level video indexing. In chapter 3 we introduced the motion intensity and the motion intensity histogram descriptors. In chapter 4, we introduced the method for panoramic video capturing and virtual camera control. We proposed a method for human motion activity recognition in chapter 5. The proposed algorithms and systems have many potential applications to video event analysis. We summarize here our main contributions and suggest potential future research directions.

6.1 Conclusions

The main objective of this dissertation is to address issues related to motion activity for both low level and high level video indexing. At the low level, we discussed the use of motion activity descriptors for content-based video retrieval and browsing. Moving toward a semantic analysis, we address issues concerning the capture and recognition of human motion activities.

In chapter 3 we proposed two new motion activity feature descriptors, motion intensity and motion intensity histogram, for low level motion activity description. A detailed quantitative characterization of motion activity enables the user to effectively browse through video using motion information. Since the two descriptors, the

motion intensity and the motion intensity histogram, are extracted in compressed domain, the extraction process is also quite efficient. When combined with other low level features such as spatial motion activity descriptor, the two descriptors also prove to be effective in video filtering.

In chapter 4 we presented a system for capturing panoramic video of human motion activity. The general idea of the system is to use a panoramic camera to capture a static scene with a person moving around, and then create a ROI video which is part of the panoramic video. The proposed system design is based on the FlyCam panoramic video system. The proposed method integrates region of interest detection, tracking, and virtual camera control, and works on both uncompressed and compressed domains. To create a smooth ROI video output, the ROI is tracked using a Kalman filter, and the Kalman filter estimation results are used for virtual camera control that simulates human controlled video recording. While the whole process can be done in real time in uncompressed domain, it can also be implemented efficiently in compressed domain. The system has no physical camera motion and the virtual camera parameters are readily available for video indexing. Experimental results show that the methods in both compressed domain and uncompressed domain are quite promising.

In chapter 5 we described a unified approach for human motion activity recognition. Our interests are activities such as a speaker walking, turning around, sitting down and getting up from a chair in a static scene. A general system is designed for the recognition of the above activities. The FlyCam panoramic camera capturing system is

used to capture the scene. Virtual camera control outputs the region of interest video that covers the speaker. We use the virtual camera control parameters for the recognition of activities such as walking, and the motion parameters of each frame for the recognition of other activities such as turning around, sitting down and getting up. For motion parameter based recognition, the likelihood of the motion parameters is represented using a multivariate Gaussian model, and the temporal change of the likelihood is characterized using a continuous density HMM. Experimental results show that this unified approach works quite effectively in recognizing the above mentioned human motion activities.

6.2 Future Directions

Some ideas for further extending the three aspects of current work are listed in the following:

6.2.1 Semantic Analysis of Video

In Chapter 3 we have proposed methods for motion activity extraction for video indexing and filtering. To extract motion activity information, an MPEG (MPEG-1/2) video is first adaptively segmented into hierarchical levels based on P-frame motion information. The motion intensity and the motion intensity histogram are then computed to represent different levels of video. While it is true that that the semantics and motion features are significantly correlated in sports and news video, the motion activity descriptor is still a low level descriptor. Integration of motion with other low

level features such as color, shape and texture is needed towards creating a semantic level description of video.

6.2.2 ROI Output for Video Coding and Streaming

A new method is presented for recording the region of interest in a scene in Chapter 4. The FlyCam panoramic camera system is used to capture the scene. After the video is compressed, the proposed method integrates detection, tracking and recording processes, and simulates human camera control. This processing is done in the compressed domain. The entire process is fully automated and experiments show that it is robust and fast enough for real time applications.

Provided there is only one speaker in the scene, this method can be applied to a panoramic view of up to 360º using the system. For typical lectures, the speaker remains at roughly the same distance from the camera, thus zooming is not necessary. However, digital zooming could be achieved by scaling the ROI for applications, as discussed in [79] and [95]. Physical zooming of panoramic cameras is difficult if not impossible, thus the highest resolution of the ROI depends on the resolution of the original panorama. Zooming in at an even higher resolution is an interesting research problem.

The cropped ROI video can be placed on the Internet for streaming as part of on-line learning software interface. It can also be sent to devices such as smart phones for wireless access. In these cases, the resolution of ROI video can be predetermined.

Adaptation of ROI video to fit different resolutions like those mentioned above and others standardized in industry is one possible research direction.

Since in the FlyCam system the cameras are stationary, the tracking information also provides a feature description of the video content. This feature information is useful for content-based retrieval applications. Also, since the region of interest is isolated from other objects in the scene, the recording result may be useful for object based coding, such as that in MPEG-4. Other research possibilities include virtual camera control for multiple objects, synchronizing the ROI output with presentation slides, or using the ROI image as a basis for gesture tracking or face recognition.

6.2.3 Complex Human Motion Activity Recognition

While most previous methods solve the problem of capturing and recognition of human activities separately, in this dissertation we present a unified approach that integrates the capturing and recognition processes. For simplicity, we have worked on activity sequences that have explicit shot boundaries. An extension of current work is to apply dynamic programming method for the recognition of activity sequences without explicit boundaries. The use of Bayesian Bayesian Networks for more complex human activity recognition is another interesting research direction to pursue.

6.2.4 Video Indexing and Summarization

One research possibility that integrates the methods and systems proposed in this dissertation is video indexing and summarization for classroom video management. The panoramic video capture system can be utilized for lecture recording. Low level motion

activity descriptors can be extracted based on recorded lecture video for the purpose of video segmentation and feature indexing. These human activities can be recognized based on the methods proposed for activity recognition. The post-processing results from the human activity recognition and the low level motion activity description can be used for high and low level video indexing and summarization. One application of the research is an on-line education system that allows a user to view and browse the lecture video.

In conclusion, we believe that there are many potential opportunities for continuing research on video indexing, both at the low level and for human motion activity recognition. By integrating at the sensor level with application specific activity recognition, significant progress can be made.

7 References

[1]. L. Agnihotri, N. Dimitrova, T. McGee, S. Jeannin, D. Schaffer, J. Nesvadba., "Evolvable visual commercial detector," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* v2, pp. 79-84, Madison, June 2003.

[2]. P. Anadan, "A computational framework and algorithm for the measurement visual motion," *Int. J. Computer Vision,* 2, pp. 283-310, February 1989.

[3]. E. H. Adelson and J. R. Bergen, "The extraction of spatiotemporal energy in human and machine vision," in *Proc. IEEE Workshop on Visual Motion*, pp. 151-156, Charleston, 1986.

[4]. H. Barman, L. Haglund, H. Knutsson, and G. Granlund, "Estimation of velocity, acceleration and disparity in time sequences," in *Proc. IEEE Workshop on Visual Motion*, pp. 44-51, Princeton, October 1991.

[5]. J. L. Barron, D. J. Fleet and S. S. Beauchemin, "Systems and Experiment Performance of Optical Flow Techniques," *Int. J. of Computer Vision*, 12:1, pp. 43-47, February 1994.

[6]. A. Baumberg and D. Hogg, "An Efficient Method for Contour Tracking using Active Shape Models," in *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 194-199, Austin, November 1994.

[7]. J. R. Bergen, P. Anandan, K. J. Hanana, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," in *Proc. European Conf. Computer Vision*, pp. 237-252, Santa Margherita, May 1992.

[8]. J. Bigun, G. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Trans. Pattern. Analysis and Machine Intelligence,* 13, pp. 775-790, August 1991.

[9]. M. J. Black, and Y. Yacoob, "Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion," *Int. J. Computer Vision*, 25(1), pp. 23-48, October 1997.

[10]. J. S. Boreczky and L. D. Wilcox, "A Hidden Markov Model Framework for Video Segmentation Using Audio an Image Features," in *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing*, pp. 3741-3744,Seattle, May 1998.

[11]. G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Proc. IEEE Workshop on Applications of Computer*, pp. 214-19, Princeton, October 1998.

[12]. M. Brand, N. Oliver, and S. Pentland, "Coupled hidden Markov models for complex action recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 994-999, San Juan, Puerto Rico, June 1997.

[13]. B. Bridgeman, "Visual receptive fields to absolute and relative motion during tracking," *Science*, 187, pp. 1106-1108, 1972.

[14]. C. Bregler. " Learning and recognizing human dynamics in video sequences," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 568-574, San Juan, Puerto Rico, June 1997.

[15]. P. J. Burt, and E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. On Communications,* 31, pp. 532-540, April 1983.

[16]. P. J. Burt, C. Yen, and X. Xu, "Multiresolution flow-through motion analysis," in *Proc. IEEE Conf. Computer vision and Pattern Recognition*, pp. 246-252, Washington, June 1983.

[17]. B. Buxton and H. Buxton, "Computation of optical flow from the motion of edge features in image sequences," *Image and Visual Computing*, 2, pp. 59-74, 1984.

[18]. H. Buxton and S. Gong, "Advanced Visual Surveillance using Bayesian Networks," in *Proc. International Conf. on Computer Vision Workshop on Context-Based Vision*, pp. 111-123, Cambridge, June 1995.

[19]. L. Campbell and A. Bobick, "Recognition of Human Body Motion Using Phase Space Constraints," in *Proc. International Conf. on Computer Vision*, pp. 624-630, Cambridge, June 1995.

[20]. C. Cedras and M. Shah, "Motion-based recognition: a survey," *Image and Vision Computing*, 13(2), pp. 129-155, March 1994.

[21]. S. –F. Chang, W. Chen, H. Meng, H. Sundaram and D. Zhong, "A Fully Automated Content-Based Video Search Engine Supporting Spatiotemoral

Queries," *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5), pp. 602-615, September 1998.

[22]. S. Chen, and L. Williams, "View Interpolation for Image Synthesis," in *Proc. ACM Conf. on Computer Graphics*, pp. 279-288, Anaheim, August 1993.

[23]. M. Christel, T. Kanade, M. Mauldin, R. Reddy, M. Sirbu, S. Stevens, and H. Wactlar, "Informedia digital video library," *Comm. ACM*, 38(4), pp. 57--58, April 1995.

[24]. M. Cover and J. Thomas, "Elements of Information Theory," John Wiley & Son, 1994.

[25]. Cutler, R. and Davis, L. "Robust Real-Time Periodic Motion Detection, Analysis, and Applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8), pp. 781-796, August 2000.

[26]. R. Cutler and M. Turk, "View-based Interpretation of Real-time Optical Flow for Gesture Recognition," in *Proc. IEEE Conf. Automatic Face and Gesture Recognition*, pp. 416-421, Nara, April 1998.

[27]. R. Cutler, et al. "Distributed Meetings: A Meeting Capture and Broadcasting System," in *Proc. ACM Multimedia*, pp. 503 – 512, Juan-les-Pins, November 2002.

[28]. T. Darell and A. Pentland, "Space-time Gestures," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 335-340, New York, June 1993.

[29]. T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated person tracking using stereo, color, and pattern detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 601-608, Santa Barbara, June 1998.

[30]. J. W. Davis and A. F. Bobick, " The representation and recognition of human movement using temporal templates," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 928-34, Puerto Rico**,** June 1997.

[31]. D. DeMenthon, V. Kobla, D. Doermann, "Video Summarization by Curve Simplification," in *Proc. ACM Multimedia*, pp. 211--218, Bristol, August 1998.

[32]. Y. Deng and B. S. Manjunath, "NeTra-V: toward an object-based video representation," *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5), pp. 616-27, September 1998.

[33]. A. Divakaran and H. Sun, "A Descriptor for spatial distribution of motion activity," in *Proc. SPIE Conf. on Storage and Retrieval from Image and Video Databases,* pp. 24-28, San Jose, January 2000.

[34]. J. H. Duncan and T. C. Chou, "Temporal edges: the detection of motion and the computation of optical flow," in *Proc. IEEE Conf. Computer Vision*, pp. 374-382, Tarpon Springs December 1988.

[35]. C. Fennema and W. Thompson, "Velocity determination in scenes containing several moving objects," *Comput. Graph. Image Process*. 9: pp. 301-315, April 1979.

[36]. D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *Int. J. Computer Vision,* 5, pp. 77-104, August 1990.

[37]. J. Foote, and D. Kimber, "FlyCam: practical panoramic video and automatic camera control," in *Proc. IEEE International Conf. on Multimedia and Expo*, pp. 1419-1422, New York City, August 2000.

[38]. D. M. Gavrila and L. S. Davis, "3-D Model-Based Tracking of Human Upper Body Movement: A Multi-View Approach," in *Proc. Symposium on Computer Vision*, pp. 253-258, Coral Gables, November 1995.

[39]. J. J. Gibson, "The perception of the Visual World, " Houghton Mifflin, 1950.

[40]. F. Glazer, G. Reynolds, and P. Anandan, "Scene matching through hierarchical correlation," in *Proc. Conf. Computer Vision Pattern Recognition,* pp. 432-441, Washington, June 1983.

[41]. M. Gleicher and A. Witkin, "Through-the-lens camera control," in *Proc. ACM Conf. on Computer Graphics*, pp. 331-340, Chicago, July 1992.

[42]. O. J. Grusser and U. Grusser, "Neuronal mechanisms of visual motion perception, " in R. Jung, edt. *Handbook of Sensory Physilogy*, 7, Pt.3, pp. 332-429, Springer-Verlag, 1973.

[43]. S. Gunn, "Support Vector Machine for Classification and Regression," ISIS Technical Report, 1998.

[44]. L. Haglund, "Adaptive multidimensional filtering," Ph.D. dissertation, Dept. Electrical Engineering, Uinv. Of Linkoping, 1992.

[45]. R. Hamdan, F. Heitz, L. Thoraval, "Gesture Localization and Recognition using Probabilistic Visual Learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 98-103, Ft. Collins, June 1999.

[46]. B. G. Haskell, A. Puri and A. N. Netravali, "Digital Video: An Introduction to MPEG 2," Chapman and Hall, 1997

[47]. L. He, M. Cohen, and D. Salesin, "The Virtual Cinematographer:A Paradigm for Automatic Real-Time Camera Control and Directing," in *Proc. ACM Conf. on Computer Graphics*, pp. 217-224, New Orleans, August 1996.

[48]. D. J. Heeger, "Optical flow using spatiotemporal filters," *Int. J. Comput. Vis*. 1, pp. 279-302, January 1988.

[49]. E. C. Hildreth, "The computation of the velocity field," *Prof. Roy. Soc. London B*, 221, pp. 189-220, 1984.

[50]. J. Hoey, J. J. Little, "Representation and recognition of complex human motion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 752-759, South Carolina, June 2000.

[51]. B. K. P Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, 17, pp. 185-204, August 1981.

[52]. D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," ***J. Physilogy***, 195, pp. 215-243, March 1968.

[53]. B. Jahne, "Image sequence analysis of complex physical objects: nonlinear small scale water waves," in ***Proc. IEEE Conf. Computer Vision***, pp. 191-200, London, June 1987.

[54]. D. Lee, B. Erol, J. Graham, J. J. Hull, and N. Murata, "Portable Meeting Recroder," in ***Proc. ACM Multimedia***, pp. 493-502, Juan-les-Pins, November 2002.

[55]. J. J. Little and J. Boyd, "Recognizing People by Their Gait: the Shape of Motion, " ***Videre***, 1(2), the MIT press, pp. 2-32, Winter 1998.

[56]. J. J.Little, and A. Verri, "Analysis of differential and matching methods for optical flow," in Proc. ***IEEE Workshop on Visual Motion***, pp. 173-80, Irvine, March 1989.

[57]. W. Y. Ma and B. S. Manjunath, "NETRA: A toolbox for navigating large image databases," in ***Proc. IEEE Conf. on Image Processing***, pp. 568-571, Washington DC, October 1997.

[58]. J. MacQueen, "Some methods for classification and analysis of multivariate observations," in ***Proc. Fifth Berkeley Symposium on Mathematical statistics andprobability***, pp. 281-297, Berkeley, 1967.

[59]. A. Madabhushi and J. K. Aggarwal, "A Bayesian approach to human activity recognition," in ***Proc. Second IEEE Workshop on Visual Surveillance ,*** pp. 25-32, Fort Collins, June 1999.

[60]. A. Majumder, W. B. Seales, M. Gopi, and H. Fuchs, "Immersive teleconferencing: a new algorithm to generate seamless panoramic video imagery," in *Proc ACM Multimedia*, pp. 169-178, Orlando, November 1999.

[61]. T. McGee and N. Dimitrova, "Parsing TV Programs for Identification and Removal of Non-story Segments," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 24, January1999.

[62]. B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation," *IEEE Trans. On Pattern Analysis and Machine Intelligence,* 19(7), pp. 696-710, July 1997.

[63]. D.J. Moore, I.A. Essa and M.H. HayesIII, "Exploiting human actions and object context for recognition tasks," in *Proc. IEEE. Conf. Computer Vision,* pp. *80-86, Kerkyra*, September 1999.

[64]. S. Mukhopadhyay, and B. Smith, "Passive Capture and Structuring of Lectures," in *Proc. ACM Multimedia*, pp. 477-487, Orlando, November 1999.

[65]. H. H. Nagel, "Displacement vectors derived from second-order intensity variations in image sequences," *Comput. Graph. Image Process.* 21, pp. 85-217, January 1983.

[66]. H. R. Naphide, T.S. Huang, "A probabilistic framework for semantic video indexing, filtering, and retrieval," *IEEE Trans. on Multimedia*, 3(1), pp. 141-51, March 2001.

[67]. S. Nayar, "Catadioptric omnidirectional camera," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 482-488, Ft. Collins, June 1999.

[68]. M. Nicolescu and G. Medioni, "Electronic pan-tilt-zoom: a solution for intelligent room systems, " in *Proc. IEEE International Conf. on Multimedia and Expo*, pp. 1581-1584, New York City, August 2000.

[69]. V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual Interpretation of Hand Gestures for Human-computer Interaction: A Review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7), pp. 677-695, July 1997.

[70]. R. Polana, and R. Nelson, "Recognition of Motion from Temporal Texture," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 129-134, Champaign, June1992.

[71]. J. G. Proakis, "Digital Communications," McGraw-Hill, 1995.

[72]. L. R. Rabiner,"A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. of the IEEE*, 77(2), pp. 257-286, February 1989.

[73]. J. O'Rourke and N. I. Badler., "Model-based image analysis of human motion using constraint propagation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(6), pp. 522-536, November 1980.

[74]. Y. Rui, L. He, A. Gupta, and Q. Liu, "Building an intelligent camera management system," in *Proc. ACM Multimedia,* pp. 2-11, Ottawa, October 2001.

[75]. Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, " Relevance Feedback: A Power Tool in Interactive Content-Based Image Retrieval," *IEEE Tran on Circuits and Systems for Video Technology,* Special Issue on Segmentation, Description, and Retrieval of Video Content, 8, No. 5, pp. 644-655, September 1998.

[76]. J. Schlenzig, E. Hunter, and R. Jain, **"**Recursive identification of gesture inputs using hidden Markov models," in *Proc. IEEE Workshop on Applications of Computer Vision*, pp. 187-194, Sarasota, December 1994.

[77]. M. A. Smith, T. Kanade, "Video Skimming for Quick Browsing based on Audio and Image Characterization," *Technical Report No. CMU-CS-95-186*, School of Computer Science, Carnegie Mellon University, 1995.

[78]. Y. Song, Y., Feng, and P. Perona, "Towards Detection of Human Motion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 810-817, South Carolina. June 2000.

[79]. Starner, T., and Alex Pentland (1996) "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models" in *Proc. IEEE Conf. Computer Vision*, pp. 265 –270, Boston, June 1995.

[80]. R. Stiefelhagen, J. Yang, and A. Waibel, "Modeling Focus of Attention for Meeting Indexing," in *Proc. ACM Multimedia*, pp. 3-10, Orlando, November 1999.

[81]. Xinding Sun, Ching-Wei Chen, B. S. Manjunath, "Probabilistic Motion Parameter Models for Human Activity Recognition," in *Proc. International Conf. on Pattern Recognition*, pp. 443-446, Quebec City, August 2002.

[82]. Xinding Sun, Ajay, Divakaran, B.S. Manjunath, "A Motion Activity Descriptor and Its Extraction in Compressed Domain," in *Proc. Pacific-Rim Multimedia*, pp. 450-457, Beijing, October 2001.

[83]. X. Sun, J. Foote, D. Kimber, B. S. Manjunath, "Recording the Region of Interest from FlyCam Panoramic Video," in *Proc. IEEE Conf. Image Processing,* pp. 409-412, Thessaloniki, September 2001.

[84]. X. Sun, J. Foote, D. Kimber, B. S. Manjunath, "Panoramic Video Capturing and Compressed Domain Virtual Camera Control," in *Proc. ACM Multimedia*, pp. 329-338, Ottawa, October 2001.

[85]. Xinding Sun, Mohan. S. Kankanhalli, "Video Summarization Using R-Sequences" *Journal of Real Time Imaging*, 6, pp. 449-459, December 2000.

[86]. X. Sun, M. Kankanhalli, Y. Zhu and J. Wu, "Content-Based Representative Frame Extraction for Digital Video," in *Proc. International Conf. on Multimedia Computing and Systems*, pp. 190-194, Austin, July 1998.

[87]. Xinding Sun, B. S. Manjunath, "Panoramic Capturing and Recognition of Human activity," in *Proc. IEEE Conf. Image Processing*, pp. 813-16, Rochester, September 2002.

[88]. Xinding Sun, B. S. Manjunath, and Divakaran Ajay, "Representation of motion activity in hierarchical levels for video indexing and filtering," in *Proc. IEEE Conf. Image Processing,* Rochester, pp. 149-152, September 2002.

[89]. R. Swaminathan, and S. Nayar, "Non-metric Calibration of Wide-angle Lenses and Polycameras," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 413-419, Ft. Collins, June 1999.

[90]. H. Tao, H. S. Sawhney, R. Kumar, "Dynamic Layer Representation with Applications to Tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 134-141, South Carolina, June 2000.

[91]. L. Teodosio, and W. Bender, "Salient Video Stills: Content and Context Preserved," in *Proc. ACM Multimedia*, pp. 39-46, Anaheim, August 1993.

[92]. J. K. Tsotsos, "The Scope of Research on Motion: Sensations, Perception, Representation and Generation," in *Motion: Representation and Perception*, edited by N. I. Badler and J. K. Tsotsos, pp. 20-26, 1986.

[93]. M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, 3(1), pp. 71-86, March 1991.

[94]. S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, "Video Manga: Generating Semantically Meaningful Video Summaries," in *Proc. ACM Multimedia,* pp. 383-392, Orlando, October 1999.

[95]. C. Wang, and M. S. Brandstein, "A Hybrid Real-Time Face Tracking System," in *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing*, pp. 3737-3740, Seattle, May 1998.

[96]. H. Wang, and S-F. Chang, "A Highly Efficient System for Face Region Detection in MPEG Video," *IEEE Trans. Circuits and Sys. for Video Tech,* 7(4), pp. 615-628, August 1997.

[97]. A. M. Waxman, J. Wu, and F. Bergholm, "Convected activation profiles and receptive fields for real time measurement of short range visual motion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 771-723, Ann Arbor, June 1988.

[98]. J. Wei and Z. N. Li, "On Active Camera Control and Camera Motion Recovery with Foveate Wavelet Transform," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 23(8), pp. 896-903, August 2001.

[99]. A. Wilson, and A. F. Bobick "Learning Visual Behavior for Gesture Analysis," in *Proc. IEEE Symposium on Computer Vision*, pp. 229-234, Coral Gables, November 1995.

[100]. G. Wolberg, "Digital Image Warping," *IEEE Computer Society Press*, 1992.

[101]. Y. Yacoob, M.J. Black, **"**Parameterized modeling and recognition of activities,**"** in *Proc. IEEE. Conf. Computer Vision*, pp. 120-127, Kerkyra, September 1999.

[102]. I. Yamato, I.Ohya, and K. Ishii, "Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 379-385, Urbana, June 1992.

[103]. H. J. Zhang, A. Kankanhalli, S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, 1(1): pp. 10-28, June1993.

[104]. H. Zhang, C. Y. Low, and S. W. Smoliar, "Video parsing and browsing using compressed data," *Multimedia Tools and Applications*, 1(1): pp. 89-111, March 1995.

[105]. J. Y. Zheng, F. Kishino, Q. Chen, and S. Tsuji, "Active Camera Controlling for Manipulation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 413-418, Lahaina, June 1991.

[106]. D. Zhong, H. J. Zhang, S.-F. Chang, "Clustering Methods for Video Browsing and Annotation," in *Proc. SPIE Conf. on Storage and Retrieval for Image and Video*, pp. 239-246, San Jose, February 1996.

[107]. M. Zobel, J. Denzler, and H. Niemann, "Entropy based camera control for visual object tracking," in *Proc. IEEE Conf. Image Processing,* v3, pp. 901-904, Rochester, September 2002.

[108]. http://www.almaden.ibm.com/cs/cuevideo.

[109]. http://www.behere.com/.

[110]. http://www.ctr.columbia.edu/advent.

[111]. http://www.ipix.com.

[112]. MPEG: http://www.mpeg.org.

[113]. Sony EVI-D30: www.sony.com.

[114]. http://www.virage.com.