University of Alberta

Robust Tracking and Human Activity Recognition

by

Meghna Singh ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

Department of Electrical and Computer Engineering,

University of Alberta, Edmonton, Canada.

Fall 2004.

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-95854-X
Our file Notre référence
ISBN: 0-612-95854-X

This thesis is dedicated to my parents.

Acknowledgements

I would like to thank Dr. Mrinal Mandal and Dr. Anup Basu for their direction, assistance and guidance during the course of my studies. I would like to particularly express my gratitude to Dr. Mandal for his patience in editing this thesis. A special note of thanks is due to all my student colleagues, who helped me in many ways.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
ASL	American Sign Language
ATM	Automatic Teller Machine
CCD	Charge Coupled Device
CDF	Cumulative Distribution Function
CMM	Combined Motion Model
CRR	Correct Recognition Rate
DOF	Degrees of Freedom
DPIV	Digital Particle Image Velocimetry
DTW	Dynamic Time Warping
DV	Directional Vector
ENN	Nearest neighbor with class exemplar
fps	Frames per second
FW	Feature Window
GMM	Global Motion Model
HAR	Human Activity Recognition
HMA	Human Motion Analysis
HMI	Human Machine Interaction

HMM	Hidden Markov Model
IMM	Individual Motion Model
IPAN	Image and Pattern Analysis Group
JPDAF	Joint Probability Data Association Filter
LED	Light Emitting Diode
LoG	Laplacian of Gaussian
LHS	Left Hand Side
MEI	Motion Energy Image
MHI	Motion History Image
MHT	Multiple Hypothesis Tracker
MLD	Moving Lights Display
NN	Nearest Neighbor
OOI	Object of Interest
PCA	Principal Component Analysis
PIV	Particle Image Velocimetry
RHS	Right Hand Side
ROC	Receiver Operating Characteristics
SSD	Sum of Squared Differences
TP	True Positive
VLMM	Variable Length Markov Model

Chapter 1

Introduction

Over the last decade there has been considerable increase in the interaction between humans and machines, leading to growing research in the fields of Human machine interaction (HMI), Human activity recognition (HAR) and Human motion analysis (HMA).

In order to achieve the objective of a higher degree of interaction, a machine should be able to identify it current environment. It should be able to recognize the presence of humans and track them. Finally, it should be able to understand through a variety of inputs what it is being commanded to do. These inputs can be of various modalities such as visual gestures and voice commands. It is also desirable that the machine be able to adapt its response to cues even when specific commands are not present. This will make the machine 'intelligent.' A major component in this entire scenario is the need for a machine to view the world around it and to derive relevant information from the plethora of data it has to process. Current research in the area of computer vision aims to fulfill this challenge.

For the past few decades, computer vision researchers have been striving to bring machine vision at par with its human counterpart, and have achieved varying success with that goal. The human brain develops over years (especially in childhood) and sets neural pathways in response to all stimuli. We, therefore develop over time the inherent capability to detect and track moving objects. Over a period of time, our brain also gets trained to recognize the activity being performed, much like supervised learning. We can interpret easily the direction of movement and can derive a reasoning of what activity is being performed. The same, however, is not as easy to accomplish with a machine.

1.1 Motivation

Advances in this domain (HAR) of computer vision are driven by a wide range of promising applications in fields such as robotics, smart rooms [68], surveillance, medical diagnostics and kineisiology to name a few. Since understanding of human activities will also lead to better synthesis of the same, related research in fields such as computer graphics, virtual worlds and avatars is also influenced by new theories in HAR. Other promising applications being looked into involve compression technology, and data indexing and retrieval systems.

Most of the work done in this area is specific to the application that the algorithm is designed for and is limited by the assumptions made for the purpose of recognition. For some of the research, the data-capturing environment too is made unrealistically simple, for example, the subject only performs activities parallel to a camera, or the background is a static green making it easier to separate movement by chromakeying. Since this domain is just emerging from infancy, the activities (such as walking and sitting) being studied for recognition at this time are also simplistic. Most existing systems are not robust to noise and clutter, or poor preprocessing leads to a complete failure all together. Robust techniques that have been developed are complex and difficult to implement in real time.

The potential applications of research in this area and the current lack of a universal, robust method of recognition are the main motivation to develop algorithms for robust tracking and recognition of human activities.

1.2 Objective

The objective of any vision based HAR system is to be able to detect, track and identify people or objects and their actions. The interaction of a vision system with its environment can be either passive, where the environment is simply monitored, or it can be active, where the visual system has the ability to control various parameters of the acquisition tool (camera) in response to the requirement of the task and external stimuli. 'Active vision' has also been used to refer to vision techniques where structured light is projected onto the scene for enhanced capture, however, that is not our intent here. While the vision system for action recognition can be intrusive, taking data directly from sensors attached to a person's body, a non-intrusive approach is generally preferred. A non-intrusive (non-contact) approach maintains a generality that makes it suitable for applications where sensory contact is not feasible, such as surveillance. Data acquired by the vision system can be suitably manipulated for the purpose of recognition. We anticipate that future research in the domain of computer vision would involve developing robust, generalized algorithms for activity recognition and a higher-level understanding of human behavior from these activities. The objective of this thesis is to develop a robust, non-intrusive, passive and generalized algorithm for human activity recognition.

1.3 Major Contributions

This thesis addresses the issue of robust tracking and HAR. The major contributions of this thesis in the area of computer vision and pattern recognition are:

1) We propose a novel non-intrusive algorithm based on silhouette directionality for recognition of human activities. This algorithm works with monocular monochrome images and is fairly view invariant, independent of color of clothing and robust to background clutter.
2) We propose the incorporation of Gaussian and Laplacian of Gaussian (LoG) weighting functions into a feature-based tracker, to enhance the noise immunity of the tracking algorithm.

1.4 Organization of Thesis

The thesis is organized as follows. We review previous research works done on HMA in Chapter 2. We also discuss the structure of a basic HMA framework, and organize our review according to the methodology adopted by various authors corresponding to each step of the basic framework. In Chapter 3, we review feature-based tracking with particular emphasis on the Kanade-Lucas-Tomasi (KLT) feature-based tracking algorithm. We briefly summarize the objectives of this research and present the problem statements that we aim to solve, in Chapter 4. In Chapter 5 we present the proposed algorithm for HAR. The experimental setup used to evaluate the proposed algorithm is also discussed. In Chapter 6 we present the proposed Gaussian and LoG weighting functions to enhance tracking performance. Chapter 7 presents the conclusions of this thesis, followed by some suggestions for future work.

Chapter 2

Review of Human Motion Analysis

Over the past few years, significant research has been carried out for developing intelligent human machine interaction (HMI). Natural language understanding, knowledge databases, sophisticated tools for reasoning have all contributed towards the goal of designing machines that behave more ‘human like.’ A truly intelligent machine should be able to extract information from the environment that it is embedded in, without the need for any external agent to supply this information. The key aspect to interact in a human inhabited environment is the ability of a machine to recognize humans and their activities. Recognizing activities will allow different modalities such as hand gestures, facial expressions and lip movements to be used for communication with the machine, thus greatly expanding the scope for human–machine interfacing. Understanding of activities and how they are performed will allow the machine to emulate them better, thus providing better synthesis for computer graphics and virtual world applications. Being able to segment body structure, track the joints and determine the underlying structure of motion of humans will also help in athletic performance analysis and medical diagnostics. With the creation of large digital libraries, activity recognition will also allow activity-based video retrieval and storage.

We begin this chapter with an introduction to a few selected potential applications in the area of human activity recognition (HAR) in Section 2.1. In Section 2.2 the framework of a generic motion analysis system is discussed. In Section 2.3-2.5 we review in detail the various past approaches for each step of the generic motion analysis system.

In Section 2.6 we discuss the assumptions that are generally made in most contemporary work related to human motion analysis (HMA). These limitations will provide us with an insight into potential areas of future work. We also compare a few methods and their performances. The chapter is summarized in Section 2.7.

2.1 Applications

The tremendous potential in research on motion analysis becomes apparent when we look at the applications that have benefited or are expected to benefit from research in this domain. Table 2.1 lists some of these applications, which are discussed further in this section.

Kinesiology: Kinesiology, the study of biomechanics, involves developing models of the human body in order to study it at a mechanical level, aiming to improve the efficiency of movement. This is also used for clinical studies of orthopedic patients. For such studies, detailed information about movement of body parts and joints is needed, and this information is mostly gathered in an intrusive manner by placing retro-reflective markers [43] or small sources of light such as LEDs [8] on the human body. Bobick and Johnson [44] use a magnetic sensor based motion capture framework to obtain three-dimensional position and orientation information of the limbs of subjects.

Choreography: HMA in particular can also be used to develop a high level description of movements of dance [66], ballet and theatre. However there has been little consensus on what the general description should be.

Computer Graphics and Animation: Motion analysis has also been used to study and then synthesize realistic motion patterns for virtual world humans [66][70]. Facial expression mapping, gait mapping have been performed to create realistic appearance of graphic characters and 'avatars.' In addition, applications of motion analysis in crash simulations have been explored.

Table 2.1: Applications of Human Motion Analysis.

General Domain	Specific Application
Kinesiology	➢ Biomechanics ➢ Clinical orthopedic studies ➢ Personalized training
Choreography	➢ Ballet moves
Computer Graphics	➢ Virtual reality ➢ Games ➢ Animation ➢ Teleconferencing– Avatars
Surveillance	➢ Pedestrian detection ➢ Parking lot surveillance ➢ Supermarket, ATM surveillance ➢ Biometric measure
Video Indexing and Retrieval	➢ Search engines
Human Machine Interfacing	➢ Gesture driven control ➢ Social interfacing ➢ Lip reading ➢ Sign language translation
Video Compression	➢ Teleconferencing ➢ Digital movies ➢ Video phones

Psychology: Some classical studies in psychology have tried to understand the human perception of motion. Pioneering work in this area was done by Johansson [8]. He showed through a moving light display (MLD) experiment that humans are able to recognize biological motion patterns even when presented with only a few moving dots.

This led to the question whether structural information was indeed required to recover motion. Further research by Boyd *et al.* [18], showed that recognition is possible even by non–structural means using the global shape of motion features. Troje *et al.* at Ruhr University [43] have also studied and demonstrated that biologically and socially relevant information about a person is conveyed in biological motion patterns. Their demonstration reflects how personality traits and emotional expression are conveyed in biological motion and how the dynamic part of motion contains more information about gender, than motion mediated structural clues. They proposed that their framework can be used not only for analysis but also for synthesis of new motion patterns.

Surveillance: Current surveillance techniques generally perform a post-operation task and provide detection only after the act has been committed. At times, these techniques need manual intervention for real time detection. However, with activity detection systems incorporated into 'smart surveillance systems' real time detection will be possible without manual intervention. Such smart surveillance systems can be introduced in parking lots [15], supermarkets and department stores. However, the possible conflicts of such surveillance applications with privacy could be a deterministic factor governing the implementation of such systems in daily life.

Video indexing and retrieval: With the advent of large video databases, researchers are looking into simpler and more effective ways to query videos in the database. A particular approach to indexing and retrieval is to query the database using action descriptors like 'videos where person steps into a car and drives away.' Retrieval of this kind will be possible when each video in the database is indexed according to features derived from the activities being performed.

Human machine interfacing: Activity recognition will also allow for better human-machine interfacing by allowing various input modalities to be developed such as gesture driven control of machines or robots [80]. It will also allow communication in noisy environments, such as airports and factories where visual clues are essential for communication.

2.2 Generic Motion Analysis System

Several works have been done in the area of HMA. These works can be classified using various criteria such as dimensionality of space being analyzed (2D *vs*. 3D), sensor multiplicity (single camera *vs*. multiple cameras), sensor modality (visible light *vs*. infra-red) and the type of model used. But, in order to get a better insight into the algorithms developed in previous work, it is prudent to discuss a generic motion analysis system and review existing techniques for each step of the generalized system.

A motion analysis system will typically have the following three basic steps.

1. Detection: It involves finding the answer to the question: '*Is there motion (corresponding to a human) present in the scene?*' It essentially involves low level processing of images and is usually the first step in all motion analysis algorithms.
2. Tracking: It answers the question: '*Where is the human moving?*' Tracking is of prime importance to systems that involve some kind of history to be maintained for the purpose of recognition of action and is usually understood to be an intermediate-level processing step. However, at times there is considerable overlap between detection and tracking algorithms.

3. Behavior Understanding: It is a high-level vision step, which involves interpreting the information derived in the first two steps in order to answer the question: '*What is the human doing?*' We anticipate that it is this step of the vision system, which will be the focus of much of the future work in this area.

A schematic of these three steps is shown in Figure 2.1. In the next section we will review work related to these three primary steps.

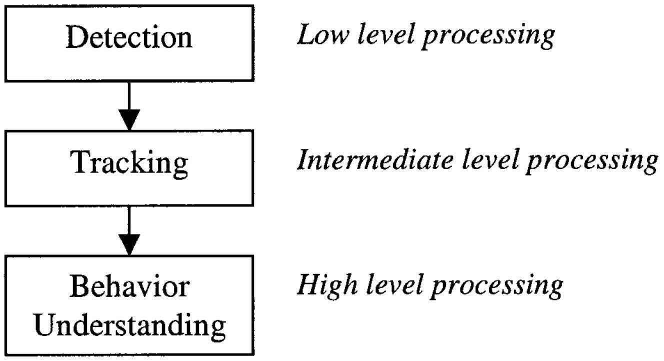


Figure 2.1: Structure of a generic motion analysis system.

2.3 Detection

Nearly every vision-based motion analysis or activity-recognition system starts with detection, much like human vision that detects movement foremost before it recognizes activities. Detection involves segmenting regions of interest, such as moving car, person, or any object of interest, from the background. This background is often referred to as clutter. Detection is an important preprocessing step since it initializes all the subsequent steps of the algorithms. Hence, it becomes imperative that the detection algorithm be able to segment the motion of the object of interest (OOI) accurately. Once the motion has been segmented it also needs to be classified as belonging to animate or inanimate object. For example, we may need to determine if the motion corresponds to a moving car, a person or a pet. In addition when multiple entities exist in the motion field, detection should also be able identify multiple presences. Thus detection involves motion segmentation and object identification (see Figure 2.2) and is classified as a low-level vision process. In the following sub-sections we will discuss motion segmentation and object identification in detail.

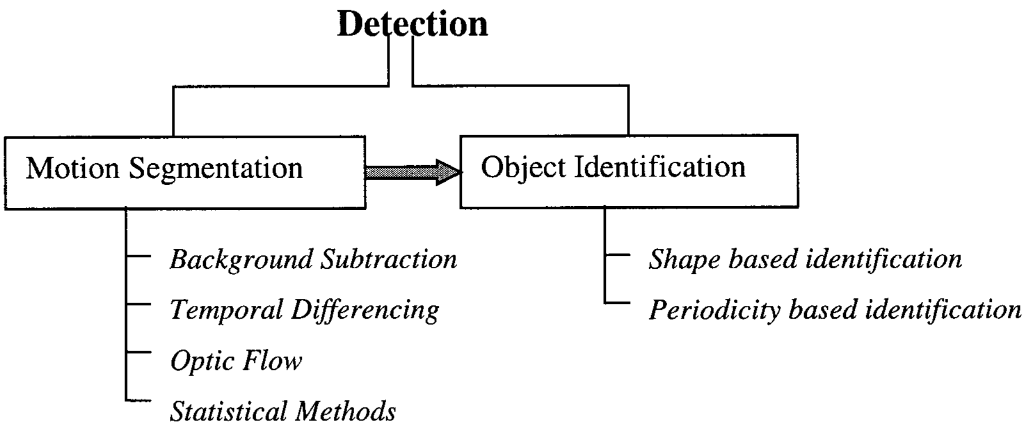


Figure 2.2: Overview of various approaches of detection.

2.3.1 Motion Segmentation

Motion segmentation involves separating moving OOI from the background image. A segmentation algorithm is required to be robust to noise and changes in the background and illumination. Some contemporary techniques are discussed below.

Background subtraction: Background subtraction is a simple solution to motion segmentation. A static image not containing the OOI is taken as a background model and a pixel-by-pixel difference between the successive frames and the background model gives the motion image. This method however, is not suitable when the background is dynamic and involves movement itself. Variations in this technique include different ways of computing the background model. The simplest background model can be created by a time average of static background frames. Instead of the time average, a median value of color or gray scale value of each pixel has also been used to create the background model. The median value computation has been found to be more robust to changes in the background illumination [48].

Temporal differencing: Temporal differencing involves a pixel-wise difference between lengths of consecutive frames [50]. The idea originated as two frame differencing and progressed to three and multiple frame differencing. Temporal differencing is adaptive to changing environments since the history of the background is just a few frames old.

Optic Flow: Optic flow techniques [51][52][53] are based on the assumption that the intensity of pixels in a sequence of images is unchanged. With optic flow it is, however, not possible to determine the image velocity in the direction perpendicular to the image intensity gradient. This ambiguity is referred to as the 'aperture problem'. Optic flow is computationally very complex and requires the inter frame motion of features, derived on the OOI, to be small. It is also difficult to implement in real time and often requires special hardware. Optic flow, however, has the advantage that it can segment moving objects even in the presence of camera motion. Optic flow can also distinguish between rigid and non-rigid motion since rigid motion presents little residual flow.

Statistical Methods: The statistical methods for motion segmentation are generally derived from the more basic background subtraction technique. The statistical methods compute the statistics of individual pixels or a group of pixels and use that information to classify regions in an image as background or foreground. Wren *et al.* [12] and Stauffer *et al.* [49] have used a mixture of Gaussians to model each pixel in an image and have used a dynamic approximation to update the model. Haritaoglu *et al.* [10] have used the maximum and minimum intensity values, and maximum variance in these values over consecutive frames as statistical parameters to model a background. This technique has been found to be more robust to changes in the background conditions.

2.3.2 Object Identification

Object identification is important in cases where multiple objects are in motion. Some times the identification has to differentiate between inanimate objects and humans, such as motion of cars and pedestrians. At other times when dealing with multiple subjects of the same kind, object identification is needed to associate an identity with each individual subject, so that they can be tracked and their activities studied independently. Objects have been identified based on two criteria, the shape of the object and the type of motion being detected.

Shape based identification: Shape based identification [50] is primarily used to differentiate between objects of different shapes, such as cars and people. Once the object motion 'blob' or region has been identified, parameters such as aspect ratio, position of extremities and skeletal representation of the blob, are used to classify the object. Shape based object identification works well when dealing with rigid body shapes.

Periodicity based identification: Periodicity based identification [54][55][56] is primarily used to differentiate between objects of the same kind that exhibit periodic motion. For example, the motion of arms and legs of a person during walk exhibits periodic repetition and can be used as characteristics to differentiate and recognize people based on how they walk. Thus a time-frequency analysis is able to determine the classification of the object. Often the self-similarity observed in periodic motion can be used to classify animate and inanimate motion.

To achieve higher identification performance a hybrid combination of both the shape based and periodicity-based classification is typically used. In addition, factors such as constraints of the structure of the human body and constraints on the movement of vehicles can be incorporated to obtain better motion detection.

2.4 Tracking

When an object of interest has been segmented and identified (detection), we may need to track it over a period of time. This is essential for most recognition algorithms that require a history of the motion to be maintained. Tracking belongs to the intermediate level of vision, and involves finding coherent relations between image features in consecutive frames with respect to color, texture, velocity and position. Tracking algorithms are mostly application dependant, for example, they depend on whether hand gestures, facial expressions, whole body parts, vehicles or pedestrians need to be tracked. Over a period of time a multitude of mathematical tools have been developed for tracking. Some of the more promising ones are Kalman filtering [57], condensation algorithm [60][58] and dynamic Bayesian networks [59]. Tracking can also involve single/multiple views, monocular/stereo cameras and singular/multiple subjects. In situations where congestion of subjects is expected, multiple cameras may be used to reduce ambiguity and improve the reliability of data. The use of multiple cameras requires optimal data fusion in order to determine the best camera view. The most relevant classification of tracking algorithms is based on whether or not the algorithms use *a priori* shape models to track objects, *i.e.* model based and non-model based tracking (Figure 2.3). As far as HMA is concerned both the model-based and non-model based approaches have evolved from simple 2D to more complex 3D volumetric analysis. Tracking scenario is often assumed to be restricted to a 'closed world', in which all possible objects present in the image sequence are known. This assumption simplifies the recognition algorithm significantly.

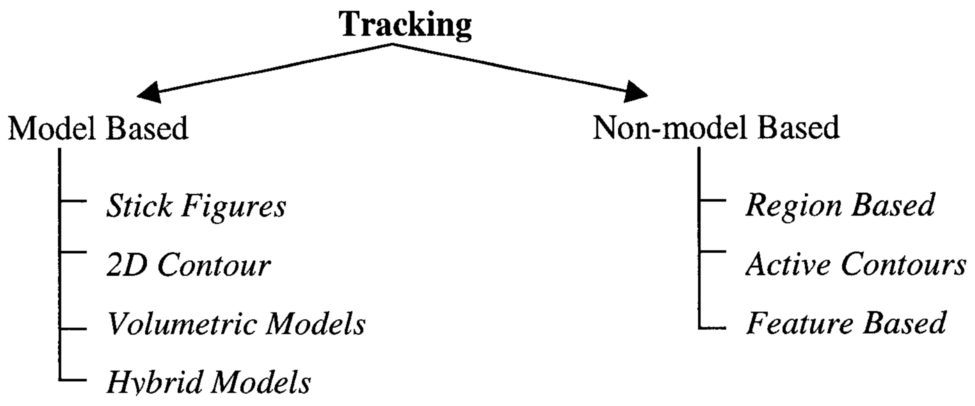


Figure 2.3: Overview of methods for tracking.

2.4.1 Model based tracking

Model based tracking generally uses a predetermined model of the subject for tracking. Features are extracted from the image and mapped to the model's structure and motion. However, mapping image features to the model can be a computationally complex task and also requires a strong segmentation of the moving object from its background. Such techniques are therefore difficult to implement for blurred sequences. Models can be created for tracking the entire human body or for more specific parts such as hands and faces. Rehg and Kanade [45] created a model of the human hand with 27 degrees of freedom (DOF) (see Figure 2.4). Trackers then attempted to align the projected model lines to the finger edges extracted locally from the image against a static solid background. While hand gestures remain fairly similar over wide range of people (differing primarily in skin color), body poses vary significantly from one person to the other and also within the same person, depending on his/her moods and clothing. We shall now discuss in detail the various traditional models used for the human body.

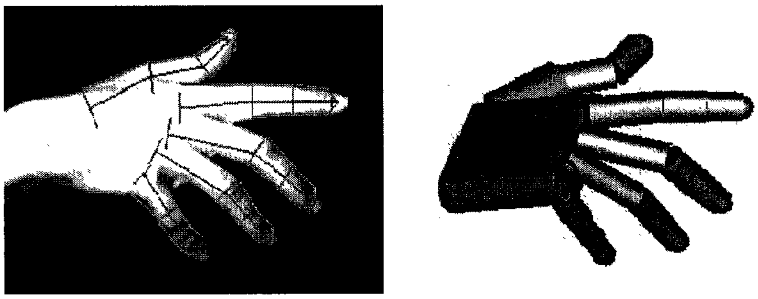


Figure 2.4: Hand tracking [45]

Stick figures

Stick figure based tracking [61][47] (see Figure 2.5) is based on the fact that the inherent structure for motion of the human body is the skeleton. Thus the motion of the legs, torso, arms and head can be approximated by the motion of the corresponding line segments. Stick figure based models have been developed for the entire human body, and for parts of it, *e.g.* legs and hands. Ali and Aggarwal [21] approximated line segments (corresponding to the stick figure) in the image by skeletonization of the whole body, and they then computed the angle between the line segments corresponding to the torso and legs for further recognition. Cunado *et al.* [46] computed the Hough transform to extract the lines that represented legs in a sequence of images. Stick figures have also been used to compute the relative angle between the joints of the limbs for recognition of activities.

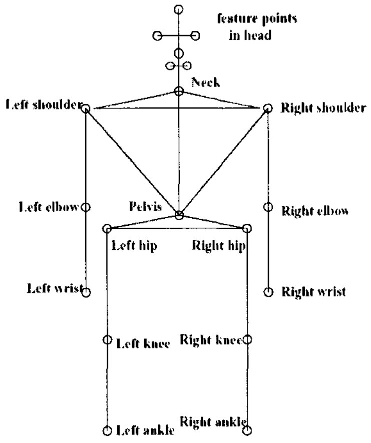
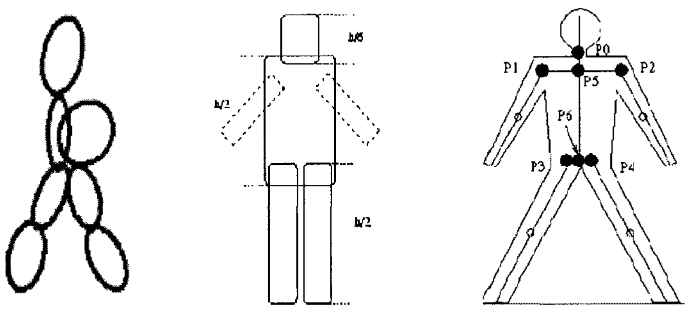


Figure 2.5: Stick figure human model [47]

Two Dimensional (2D) contours

The 2D contours (Figure 2.6) are a close representation of the projection of the subject onto the image plane. These contours were initially termed as ‘cardboard people’ model in which limbs of humans were represented as connected rectangular pieces of cardboard. Human motion constraints, such as anatomical joint-angle limits, body part inter-penetration and equilibrium positions, are also applied to the joints of these 2D contours, like their stick figure counterparts. However 2D contours are restricted to the angle of view of the camera.

Lee and Grimson [62] and Shakhnarovich *et al.* [63] created 2D models based on ellipses fitted to various body segments of the extracted silhouette (see Figure 2.6 (a)). They then derived feature vectors such as centroid location of each ellipse, aspect ratio of minor and major axis of the ellipse and the orientation of the major axis. These moment based feature vectors were then used for recognition. Haritaoglu *et al.* [10] (see Figure 2.6(b)) used cardboard model of person standing upright to predict the location of the head, torso, legs and feet of the individual in the image. Intille *et al.* [64] computed blobs corresponding to each person in the image and then matched objects to these blobs.



(a) (b) (c)

Figure 2.6: 2D contour modeling of the human body. (a) Elliptical model [62], (b) Cardboard model - W^{4}S [72] and (c) 2D contour human model [71].

Volumetric models

Volumetric models (Figure 2.7), although being computationally more complex, allow the tracking model to be independent of the camera viewpoint and better tracking results are achieved. In these models the geometric representation of the human body is more detailed and precise. Elliptical cylinders, cones and spheres are used to represent the various body parts. The surface of the human body is represented as a polygonal mesh. The primary advantage of volumetric models is their ability to handle occlusion. There is a tradeoff between the accuracy of representation and the number of parameters of the model. Gavrila *et al.* [66] developed a full body model using tapered superquadrics with twenty-two DOF: six for torso and four for each arm and leg; and captured images from four calibrated cameras (see Figure 2.7(a)). The subjects, however, were required to wear a tight-fitting body suit with contrasting colors for each limb.

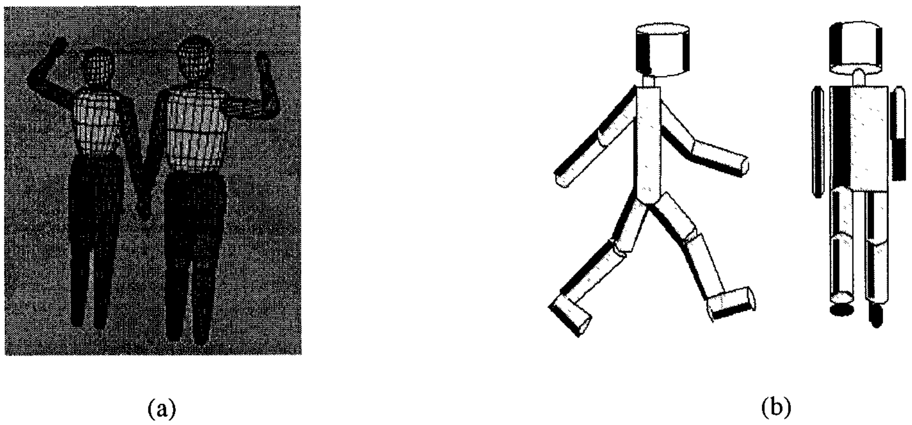


Figure 2.7: Volumetric Models. (a) Superquadrics model [66] and (b) Volumetric human [73].

Hybrid models

Several hybrid models have been proposed in the literature. Green *et al.* [13] have developed a hybrid model called the clone-body model that combines 2D models and volumetric models. The clone-body model involves tracking both the edge and the region of each subject in a sequence of frames and then dynamically mapping size and texture on to the clone model. This approach contrasts previous works, which involved either edge or region tracking for mapping a human image to a model. Green and Guan [13] have used particle filtering (condensation algorithm) to calculate the joint angles. Other authors have used the Kalman filter, instead of particle filters, although it is a considerably more complex algorithm. The output of the particle filter at a given time-step is an approximation of the probability distribution of likely joint angles, while the output of the Kalman filter is a single estimate of the position and covariance of joint angles. This gives particle filtering the advantage of maintaining multiple hypotheses and therefore being more robust to clutter. Once an estimate of the joint angles in the next frame is computed, HMMs can then be used to infer the human movement that could have produced the observed set of joint angles.

Most model based techniques encounter the problem of matching a human image extracted from the video sequence frame to its abstract representation by models of varying complexity. This problem in itself is non trivial, and is governed by the number of model parameters and the efficiency of the segmentation algorithm.

2.4.2 Non model based tracking

The highlight of non-model based tracking [69] is the idea that structural information is not always required to track an object and complexity can be reduced by using some other method of tracking. We will now discuss in detail the various non-model based tracking techniques.

Region based tracking

The region based tracking algorithms identify a 'blob' or connected region in space that is associated with each OOI and track it over time using a similarity measure or a cross correlation parameter. This method however suffers from two major drawbacks. Firstly, shadows often result in incorrect 'blob' representation. Secondly, in situations where multiple subjects are present together, congestion and occlusion leads to merging of the blobs and individuality is lost. A potential solution of this problem is the use of multiple cameras.

Active Contour based tracking

Active contours or 'snakes' are based on direct extraction of the bounding representation of the subject. Snakes are splines that possess an internal energy function, defined by their configuration and an external energy configuration, defined by the image energy. Given an initial set of points on the snake, the snake tries to achieve a position that results in a local maxima of the energy functions. Though active contours are less computationally complex than region based tracking they need to be initialized as separate contours for each individual subject. This initialization can be difficult for complex objects. Niyogi and Adelson [74] have used snakes to recover the fronto-parallel walker's body contours. These recovered contours can be further used to build a stick model of the walker and recognize activities based on the joint angles computed from the stick model.

Feature-based tracking

The foundation of feature-based tracking is the idea that computational complexity is reduced when tracking prominent features of the object, instead of the whole region of the object or its contours. Feature-based tracking involves feature extraction and feature matching. Parameters such as corners, color information and texture have been used as features for the purpose of tracking. There are two broad approaches to feature-based tracking — dynamic and static feature tracking. Feature tracking is termed static when features are extracted in each frame *a priori* and the algorithm computes the optimal correspondence between them. Various parameters such as entry and exit of features, and cost functions for trajectory smoothness have been investigated for static feature-based tracking. Particle Image Velcoimetry (PIV) [36] is one such application where features undergo random rapid movement and is best solved by static feature-based tracking. In dynamic feature-based tracking the features are determined and tracked over consecutive frames dynamically, estimating motion of the feature and searching for it in the next frames. Popular algorithms such as the KLT feature-based tracker have been developed on the principle of dynamic feature-based tracking. In Chapter 3, we discuss feature-based tracking, with particular emphasis on the KLT algorithm.

An issue with feature-based tracking is the tradeoff between feature complexity and tracking efficiency. Lower level features such as coordinate locations of edges are easier to extract but relatively more difficult to track since it is difficult to establish a one to one correspondence between them. Higher-level features such as blobs and 3D volumes are easier to track but difficult to extract.

2.5 Behavior Understanding

Although behavior is a term generally associated with humans in computer vision research, it can also be associated with more inanimate objects such as vehicles and robots. The path-plan of the vehicle while driving on a highway or while doing parallel parking is also indicative of the behavior of the vehicle. As such, we will concentrate more on reviewing human motion patterns to derive behavioral understanding. Behavior understanding involves two steps: Action recognition and description. We will now discuss selected related works in these two steps.

2.5.1 Action recognition

In simple words, action recognition is a classification problem between unidentified test sequences and pre-stored typical activity sequences. Some important requirements of an action recognition algorithm are to make the training and recognition robust to small spatial and temporal variations. Some techniques for activity recognition are discussed below.

Dynamic Time Warping

Dynamic time warping (DTW) is a technique that was developed primarily for speech recognition in the early days. It has since been applied effectively for HMA [78]. DTW uses an optimum time expansion and compression function to perform a non-linear time alignment of two signals. This technique can be explained best as a ‘temporal template’ based dynamic matching of patterns and is conceptually simple and robust. When dealing with images it has the advantage of being able to work with inconsistent time scales of the test frames and the reference frames. Myers *et al.* [67] have proposed that given a test pattern (sequence of frames) of length Q and reference patterns of P templates, an optimal match between test and reference patterns can be found by dynamic programming. The order of complexity of matching is $O(PxQ^2)$, if no constraints on pattern matching are applied. The complexity reduces to O(PxQ), if the pattern matching is constrained. Rao *et al.* [23] have computed the spatio-temporal curvatures of trajectories of moving objects in the videos and used DTW to match these trajectories based on a view invariant similarity measure.

Neural Networks

Neural networks are also promising for analyzing time varying data. However, a large set of data is generally required to train the network. Several enhancements to neural networks have been made to incorporate time-delayed functionality. Yang and Ahuja [77] have applied time delayed neural networks for hand gesture recognition and achieved a high recognition rate.

Principal Component Analysis

Principal component analysis (PCA) is often used to reduce the dimensionality of the feature space, which in turn reduces the complexity of computation. It is a mathematical technique that transforms possibly correlated variables to smaller number of uncorrelated variables. PCA is, essentially, a linear basis transformation that decomposes original data into a number of components. The first few principal components account for most of the relevant information and thus result in a compact representation of the features extracted. Mathematically, the principal components are the Eigen vectors of the covariance matrix of the original data. PCA can capture redundancy in data only when the data lies within a linear subspace (of the original data) of low dimensionality. If the data is low dimensional but exists in a non-linear manifold, then PCA will not be able to reduce redundancy in data. PCA has been used on data sets for pedestrian detection [65] (see Figure 2.8), hand gesture tracking and in the Bio-walker project [43].

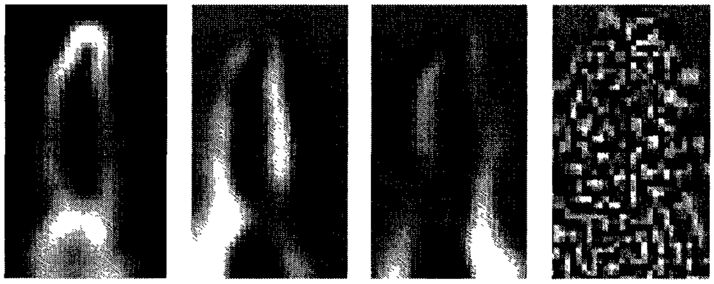


Figure 2.8: The first few Eigen vectors generated from PCA analysis of a pedestrian ([65]).

Fourier Analysis

Fourier analysis uses complex exponential, orthogonal basis functions, and can lead to data decomposition much like PCA. Cunado *et al.* [46] extracted lines corresponding to the legs of human subjects from a sequence of images depicting walking. They then treated the motion of these lines as simple harmonic motion of two pendulums. Fourier analysis of this harmonic motion revealed frequency components that were used further as a gait biometric. Both the Fourier magnitude and phase spectra can be used to enhance the analysis. For other periodic activities such as jumping, running and skipping also, a Fourier analysis of data allows matching with predetermined patterns of trained activities. Fujiyoshi *et al.* [22] have extracted a star representation from the extremities of the human silhouette and performed Fourier analysis of this representation for gait cycle detection and recognition.

Template matching

In this technique, an image sequence is converted into a static shape pattern, which is compared to pre-stored prototypes for recognition. Template matching leads to a loss of the dynamic characteristics of motion and is more sensitive to variance in the duration of movement. Bobick *et al.* [20] have created two component temporal templates: Motion History Images (MHI) and Motion Energy Images (MEI) to recognize human movements. The MEIs (see Figure 2.9) are binary images that represent where motion has occurred in a sequence whereas the MHI are scalar images where intensity at each location is a function of the local motion history at that location. These templates are view specific and were matched against models of views of different movements. Collins *et al.* [75] used temporal templates to establish a baseline method for human identification. They classified body shape and gait by performing nearest neighbor matching among correlation scores.

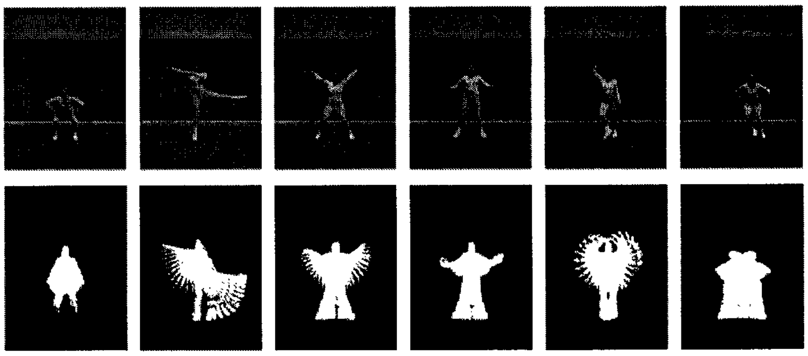


Figure 2.9: MEI of six aerobic exercises [20].

State Space approach

Image sequences depicting motion can be noisy. The activities can be occluded or the background can be too cluttered for a good segmentation. In such cases recognition calls for a probabilistic framework. In such a framework the recognition decision is made in favor of the activity with the highest probability of occurrence. State representations and Hidden Markov Models (HMM) are probabilistic in their approach. The state space approach defines each static pose of the body as a state, *e.g.* states s1, s2 and s3 in Figure 2.10. The states are connected to each other by certain state transition probabilities, *e.g.* probabilities p2, p4, p5 and p6 in Figure 2.10. Any motion sequence is considered to be a set of directional transitions between states. This approach does not have a time duration issue since a long activity could be represented as a state repeatedly transiting to itself, *e.g.* transitions corresponding to probabilities p1, p3 and p7 in Figure 2.10.

HMM are variants of stochastic state machines. Motion can be modeled as an HMM with a finite set of output probability distributions. Galata *et al.* [76] used variable length Markov models (VLMM) for modeling human behavior. They used VLMM because of their more powerful encoding of temporal dependencies. Yamato [82] developed low-level silhouettes of human movement (tennis strokes) and vector quantized them to use as inputs to the HMMs. Bregler [81] proposed a probabilistic decomposition of human motion at various levels of abstraction and showed how probability distributions propagate across space, time and levels of abstraction. Several other HMM based approaches have also been proposed for modeling gait and for human activity recognition [79] [16].

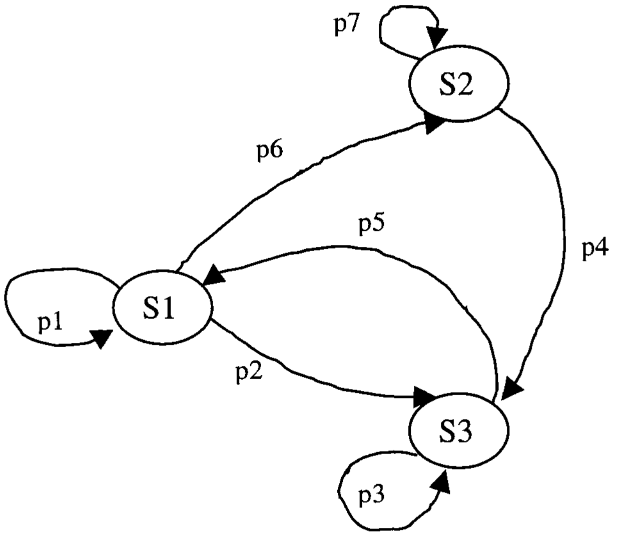


Figure 2.10: An illustration of state diagram and associated transition probabilities.

Pattern Classifier

Pattern classification is a widely used method for recognition of motion and activities. It attempts to measure the similarity between standard patterns and patterns derived from the test cases. Nearest neighbor algorithm (NN), Nearest Neighbors with class exemplars (ENN) and the K- means clustering algorithm are some of the standard pattern classification algorithms that have been used in past works. These algorithms use similarity measures such as Euclidean distance, Procrustes distance [27] and Mahalanobis distance [20].

2.5.2 Action description

Researchers are also involved in building a grammar of movements, much like the development of phonemes for processing speech and American Sign Language recognition (ASL) [83]. The smallest contrastive dynamic units of movement have been termed by researchers as 'movemes' [81], 'dynemes' [13], and sign language motion as 'cheremes' [84]. The entire recognition process in a semantic system becomes a transitional set from one dyneme to another, with the constraints of motion governing possible combinations. Green *et al.* [13] have developed a skill model, representing each movement as a linear sequence of dynemes. Ivanov and Bobick [15] have developed a stochastic parsing method in which they divided the basic problem of recognition into two levels. At the first level, they used independent probabilistic event detectors to predict the possible 'low-level' event. The output of this low level provides the input for a stochastic context-free grammar parsing mechanism. The grammar parsing level allows prior knowledge and temporal constraints to be included in the recognition framework. They demonstrate the effectiveness of their framework with surveillance and hand gesture recognition experiments.

2.6 Assumptions in related work

Most of the related work in HMA is based on certain limiting assumptions. We will discuss these assumptions briefly in order to understand the potential areas of further research in this field. Molesund *et al.* [3], in their survey of computer vision based motion capture, classified typical assumptions into two broad categories: assumptions related to movement and assumptions related to appearance. The first category is related to the movement of the subject (or the camera) in the relevant environment. The second category is related to the appearance of the environment and the subject. We list these assumptions in Table 2.2

Table 2.2: Typical assumptions made in related works on human motion analysis.

Assumptions related to movement	Assumptions related to appearance
1. Subject is always present in the view space. 2. The camera is stationary or in motion at constant speed. 3. Only one subject in view at one time. 4. Subject faces camera at all times. 5. Movements are parallel to the camera plane. 6. No occlusion, self or otherwise 7. Movements are slow. 8. Motion pattern of subject is known.	**Environment** 1. Constant lighting (well lit). 2. Stationary background. 3. Uniform (chroma-key) background. 4. Known camera parameters. 5. The ground plane is flat. **Subject** 1. Known start pose. 2. Known subject and model. 3. Markers placed on subject. 4. Special colored clothes. 5. Tight-fit clothing. 6. Subject within a specified age group.

The first three movement-related assumptions are almost universal to all HMA algorithms. The fourth assumption of the subject facing the camera is related more to applications such as face tracking and lip reading, where it is important that the face be visible to the camera at all times. The fifth assumption of the movements being parallel to the camera plane is related to applications such as gait analysis and biomechanics, where the most important features of movement are visible only in the lateral plane. The sixth and seventh assumptions are also universally used as they simplify the analysis of motion. Applications where the analysis of motion, and not recognition, is the primary motive, *a priori* knowledge of subject motion pattern simplifies the problem, which is the final assumption related to movement.

Assumptions made with respect to appearance can be related to either the environment or the subject. The first assumption related to the appearance of the environment is that the lighting in the area of capture should be constant. This basically holds for indoor sequences and implies that the scene is well lit. The second and third assumptions of a stationary uni-colored background also relate practically to indoor scenes. A stationary, uniform background results in ease of segmentation but may not always be true in real life; therefore this is a major limiting assumption. Assumptions four and five simplify calculation of the depth of subject and image registration. Subject appearance related assumptions are the most limiting factors of an algorithm, as these reduce the scope of real life implementation. The first two subject appearance related assumptions help in initializing the recognition algorithms by specifying some model parameters *a priori*. The third assumption makes tracking and data capture more tractable, but is intrusive and therefore undesirable for many applications. The fourth and fifth assumptions are related to the subject's clothing and are probably the most significant limiting factor of any algorithm. Practically, it cannot be assumed that the subject will always wear clothes of a certain kind or fit. The last subject appearance related assumption limits the age of the subject to be with a certain range. Most of the human motion analysis algorithms are designed for humans in the age group of 20-50. However, it is important to note that the movement of the subject is greatly influenced by his/her age. Most researchers, however, often neglect this factor.

It would be prudent at this stage to tabulate the activity recognition rates achieved by researchers so far with the various approaches to detection, tracking and recognition of activities (see Table 2.3). It is also worth mentioning that most of the applications developed so far relate to human identification based on gait and only a few works have been done for recognition of generic activities.

Table 2.3: Activity recognition rates achieved so far.

Authors	**Activity being recognized**	**Primary technique**	**Recognition rates**
Cunado *et al.* [46]	Gait	Fourier Analysis	80-90%
Benabdelkader *et al.* [93]	Gait	PCA	93%
Ali *et al.* [21]	Sit, Stand, Stoop	Joint angle computation	71-89%
Sun *et al.* [16]	8 Martial arts poses	HMM	30-90%
Ben-Arie *et al.*[17]	Walking, jumping	3D Model based, multidimensional indexing	70-97%
Tanawongswan *et al.*[90]	Gait	Joint angle, DTW, PCA	73%
Oliver *et al.* [92]	Office related, phone conversation, presence in cubicle.	HMM	72-99%
Niu *et al.* [92]	Stalking behavior	Statistical properties of trajectories	80-100%

2.7 Summary

In this chapter we presented a comprehensive review of the state of the art techniques for HAR. We also discussed the limiting assumptions made by these techniques. It is these assumptions that limit the application of contemporary methods as a universal solution for HAR. The development of a universal solution still eludes researchers and it is a motivating factor for further research in this domain.

Chapter 3

Review of Feature Based Tracking Algorithms

Object tracking, while a simple task for humans, is monumentally more challenging for computer vision systems. We have mentioned the potential applications for object tracking in Chapter 2 and also discussed the past approaches that include region based, contour based and feature-based tracking algorithms. In this chapter we will review feature-based tracking, followed by a detailed discussion of a standard, often used algorithm called the KLT (Kanade-Lucas-Tomasi) algorithm.

As discussed in Chapter 2, feature-based tracking has been used for applications involving human tracking. Besides tracking human motion, feature-based tracking has been used for applications such as digital particle image velocimetry DPIV [36]. Particle image velocimetry involves illuminating a flow of fluid or gas with pulsed laser and tracking the movement of the flow. Researchers have used optic flow techniques and feature-based tracking methods for accomplishing particle image velocimetry (PIV) digitally.

3.1 What is a Feature?

The foremost and fundamental question to be answered at this point is, 'what is a feature?' Features are sections of the image that are easily highlighted for the purpose of detection and tracking. Verestoy *et al.* [30] have defined features as local regions of interest. Features can be selected based on some measure of texture, edge sharpness, color and corners. Comaniciu *et al.* [41] have used color as the target feature in their algorithm to track ‘non-rigid’ objects using mean shift. In the following section we look at the difference between tracking rigid and non-rigid bodies.

3.2 Tracking of Rigid and Non-Rigid bodies

Rigid bodies themselves do not change shape as they undergo motion, thus rigid body motion is simpler to track since the transformation of the body itself is a tractable problem. For relatively slow motion it is easier to compare local features, such as edges on a rigid body. However, the assumption of shape constancy is not always valid in cases where there are large changes from one frame to the other. Tracking non-rigid bodies is a more complicated task because of the lack of consistent features to track. Often object segmentation for non-rigid bodies is unreliable thus making it difficult to even extract local features, let alone track them. Halevi *et al.* [85] have tracked the motion of water flowing down a stream and the motion of ants (non-rigid bodies) using ‘disturbance maps’. These disturbance maps are obtained by a linear subtraction of the temporal average of the previous frames from the new frame.

3.3 Static *vs.* Dynamic methods

As mentioned in Section 2.4.2, feature-based tracking techniques can be broadly classified into static and dynamic methods. Static feature-based techniques aim at finding the best features in all frames of a sequence and then establish the optimal correspondence between them. They are also referred to as feature linking algorithms. Dynamic methods on the other hand find the best features only in the first few frames (it could also be a manual initialization) and then attempt to dynamically find the displacement of these features in the next frames. A popular dynamic image registration technique was proposed by Lucas and Kanade [37], which makes use of the spatial intensity gradient of the images to iteratively find a good match between frames. Later, Shi and Tomasi [38] extended this technique incorporating an affine transform to handle rotation, scaling, and shearing of objects. This algorithm is popularly known as the KLT algorithm and has been widely used since its inception.

Static algorithms are preferred when tracking a dense field of similar objects. The IPAN tracker developed by Verestoy and Chetverikov [30] is one such non-iterative algorithm that advocates static feature-based tracking. Other static tracking algorithms developed by Rangarajan and Shah [34], Sethi and Jain [35] are also based on the feature linking technique and differ primarily in the trajectory optimization strategies and cost functions.

All feature-based tracking algorithms encounter the non-trivial problem of motion correspondence. Researchers attempt to apply 1-1 mapping constraints to resolve motion correspondence. Static algorithms can be further classified, according to the method used to resolve correspondence, as statistical methods, heuristic methods and qualitative methods. We briefly discuss these methods in the following sections.

3.3.1 Statistical Methods

Statistical methods of feature tracking represent the location of feature points as probability density functions and not as specific locations. Some of the existing statistical methods include the Multiple Hypothesis Tracker (MHT) [86] and the Joint Probabilistic Data-Association Filter (JPDAF) [87]. The strength of the statistical approach lies in its robustness to clutter, however, it has several limitations. Firstly, this approach assumes that feature points move independently, which is not always true. Secondly, these techniques typically involve several probabilistic parameters, such as *a priori* probabilities for false measurements and missed detections. It is not a trivial task to determine these parameters optimally when the designed systems are sensitive to parameter settings. Finally, these methods are computationally very demanding with their complexity increasing exponentially with the number of points being tracked.

3.3.2 Heuristic Methods

Other researchers have attempted to solve the motion correspondence problem with deterministic solutions. The most common approach in these methods is the use of a greedy exchange algorithm. Not only are the heuristic methods computationally simpler, but they also have a smaller set of parameters to be investigated. It is easy to incorporate additional constraints such as motion velocity and smoothing cost functions to the heuristics. Shafique and Shah [42] presented a non-iterative greedy algorithm for multi-frame point correspondence, using the KLT algorithm to detect the features, and a weighed digraph to formulate the framework for the greedy algorithm.

3.3.3 Qualitative Modeling

Veenman *et al.* [40] incorporated available motion knowledge to build motion models, which resolve the motion correspondence to a certain degree. They proposed three motion models: individual motion models (IMM), combined motion models (CMM) and global motion models (GMM). They also discussed different strategies to satisfy these models. IMMs, as the name suggests, represent the motion of individual features. Properties such as inertia and rigidity were incorporated in the individual models. Motion smoothness constraint was imposed on a set of points to develop the

CMMs, and was extended over the whole sequence to develop the GMMs. These models made it easier to find specific strategies for optimal solutions among the large number of candidate solutions.

3.4 KLT Tracking algorithm

The KLT algorithm, introduced in Section 3.3, is a pioneering algorithm in feature-based tracking literature. The fact that it still finds extensive use ten years after its inception speaks well for the underlying theory. Over time, the algorithm has been modified and enhanced to suit specific applications. Gonzalez *et al.* [39] have used the KLT algorithm to extract feature points in an image. These feature points were then used to initialize a point wise human motion tracker. The coherence between the displacements of the features was used as a constraint to guide their tracker. Bourel *et al.* [88] and Sarris *et al.* [89] have used the KLT algorithm to pick features on a human face and used facial geometry constraints to make the tracking robust.

The development of the KLT algorithm started off as an image registration technique to find the best match between two images and later developed into a definition of good features to track. We present the details of the KLT algorithm in the following section.

3.4.1 The algorithm

The original algorithm proposed by Kanade and Lucas [37] was developed based a Newton–Ralphson style iterative search method. At that time the tracking methods in use were generally based on the following two criteria: image correlation or sum of squared differences (SSD). If there was small inter-frame displacement in the image then feature windows could be tracked by optimizing some defined matching criterion. Kanade and Lucas recognized that there were two main issues with selecting features.

1) If the selected features are poor, such as the edge of a light reflection on a glossy surface, then the tracking will not be optimal, and such poor features will be lost or will give erroneous tracks.

2) The inherent projection of features from the 3D real world to a 2D image plane, may result in occlusion of features and the tracker can drift away from the original target.

Shi and Tomasi modified the algorithm developed by Kanade and Lucas to incorporate affine motion such as linear warping and translation. The main objective of the KLT algorithm is to extract features in the first frame of an image sequence and track them in the next frame. There are two major concepts in the KLT algorithm, namely 1) track-ability and 2) dissimilarity, which are discussed below.

3.4.2 Track-ability

The KLT algorithm defines a good feature as a textured patch that has high intensity variations in both the x and y directions. The best features to track, such as edges and corners, can be determined by analyzing multiple pixel areas called 'feature windows' (FW) of size $X \times Y$, in the vicinity of the edge. They define the intensity function of a textured patch as g_x, g_y and compute the local intensity variation matrix as:

$$Z = \begin{bmatrix} \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} g_x^2 & \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} g_x g_y \\ \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} g_x g_y & \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} g_y^2 \end{bmatrix}$$

Let the eigen values of this Z matrix be denoted by λ_1 and λ_2. A good feature is determined as one whose minimum eigen value exceeds a predetermined threshold λ.

$$\min(\lambda_1,\lambda_2)>\lambda$$

3.4.3 Dissimilarity

The KLT algorithm also defines a dissimilarity metric, which quantifies the change of appearance of a feature between the first and the current image frame. Let B(x,y,t) be the first frame in which the FW is initialized, and A(x+Δx,y+Δy,t+τ) be the next frame in which the FW is being tracked. The displacement $d=[\Delta x \quad \Delta y]$ of these FWs is approximated with a translation. The KLT algorithm attempts to iteratively find 'd' such that the dissimilarity over the feature window in the second frame is minimized. The dissimilarity 'E' is computed as the sum of squared differences (SSD), and is calculated using the following equation:

$$\textit{Minimize}\text{: } E=\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[A(x+\Delta x, y+\Delta y, t+\tau) \quad -B(x,y,t)]^2 w(x,y) \qquad (3.1)$$

where, the standard KLT uses $w(x,y)=\begin{cases}1 & (x,y)\in FW \\ 0 & elsewhere\end{cases}$ (3.2)

In practical implementations of KLT, if 'E' is high, the feature is termed lost. Using the first order Taylor expansion and the image derivatives ($g_i, i=x,y$), Eq. 3.1 is rewritten as:

$$Zd=a \qquad (3.3)$$

where

$$Z = \begin{bmatrix} \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} g_x^2 & \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} g_x g_y \\ \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} g_x g_y & \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} g_y^2 \end{bmatrix}, \; a = \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[A(x,y) - B(x,y)]\,[g_x, g_y]^T$$

The tracking algorithm is implemented to find the solution to Eq. 3.3.

3.4.4 Limitations

The KLT algorithm implements an affine matching, also referred to as feature template matching, approach for motion correspondence. However, large inter-frame displacements of features and noisy image sequences may lead to failure of the KLT tracking algorithm.

3.5 Summary

In this chapter we reviewed various state of the art techniques for feature-based tracking and also discussed in detail the KLT tracking algorithm. The KLT tracker has a noise limitation and cannot track features in noisy image sequences. We will propose a solution to this limitation in Chapter 6.

Chapter 4

Problem Statement

The goal of this thesis is to develop a robust algorithm for object tracking and human activity recognition. We define our primary and secondary objectives as follows:

1. Primary Objective- Efficient Human Activity Recognition.

2. Secondary Objective- A Noise Immune Tracking System.

4.1 Problem Statements of Primary Objective

A simplified framework for human activity recognition (HAR) system has been discussed in Chapter 2. The foremost task performed in all current techniques is the separation of the moving subject of interest from the background. In Section 2.3 we investigated the existing techniques for background-foreground separation. Although these techniques are robust and efficient, most of them are computationally expensive. This leads us to the formulation of our first problem statement.

Problem Statement 1: Which of the existing methods for background-foreground separation meets our requirement of a robust, efficient and computationally simple technique?

We discuss the solution to this problem statement in Chapter 5 (Section 5.1), where we implement a background-foreground separation technique, suitably modified to meet our requirements of low complexity and high efficiency.

Various existing techniques of human activity recognition have been discussed in Chapter 2. These techniques are limited in their approach by assumptions regarding clothing, as they often require tight form fitting clothing of specific colors to be worn by the subjects. Some of the techniques only work with clean uniform colored backgrounds with the test videos shot indoors. Other factors such as limited camera view angle also reduce the scope of most contemporary work to recognition of activities that are performed parallel to the camera view plane. These assumptions limit the feature vectors that can be derived from the images. The limited feature vectors are used further for the purpose of recognition. The assumptions discussed above lead to the formulation of the second problem statement.

Problem Statement 2: Is it possible to develop HAR algorithm, which is independent of factors such as clothing, background, view angle and zoom depths, and is not specific to individuals?

We address this problem in Chapter 5, where we propose a novel human activity recognition algorithm, which does not involve special clothing, background or viewpoint limitations. We also discuss the theoretical aspects of our proposed algorithm.

The third problem statement involves the deliverable element of our research.

Problem Statement 3: Are their potential applications that could benefit from a direct implementation of this research work?

We present a few applications that will benefit greatly from our research, in Chapter 5 Section 5.8.

4.2 Problem Statement of Secondary Objective

We have reviewed feature-based trackers in detail in Chapter 3 and also discussed the KLT algorithm. It has been found that the standard implementation of the KLT algorithm fails when the input image sequences are noisy. Noisy sequences may be caused by conditions such as poor camera exposure and poor lighting conditions. In these noisy cases the tracking algorithm will lose the feature in the next frame or choose an erroneous correspondence between features. This drawback leads us to formulate our fourth and last problem statement.

Problem Statement 4: Is it possible to enhance the performance of feature-based trackers in noisy environment?

We discuss the performance enhancement achieved by the incorporation of two proposed weighting functions with feature-based tracking in Chapter 6.

Chapter 5

Human Activity Recognition

As humans we have the inherent ability to easily interpret actions from silhouettes. This ability is demonstrated during various dances and theatrical plays where only the performer's shadow is displayed on a screen. The audience can easily recognize various actions from the silhouettes without seeing any structural details. Using a similar principle, in this chapter, we propose an algorithm that will allow a machine to recognize activities based on silhouettes, without the need to compute motion of individual body parts.

We present the various steps of the algorithm in Sections 5.1 through Section 5.5. The experimental setup and the performance evaluation of the proposed algorithm are described in Section 5.6 and Section 5.7, respectively. Potential applications of the algorithm are discussed in Section 5.8. A summary of this chapter is presented in Section 5.9.

An overview of the proposed algorithm is shown in Figure 5.1. The algorithm can be divided into three steps based on low-level, intermediate-level and high-level vision. The low-level vision step includes video data acquisition, background-foreground separation, silhouette extraction and representation. The video data acquired in our experiment is monocular, and an adaptive background-foreground separation algorithm extracts motion information as foreground from the video data. However, a foreground is often corrupted by noise and may consist of disconnected components. Therefore, we use morphological operations and connected component analysis to extract a clean connected silhouette. We then represent the contour information of the silhouette as a chain code from which the directional vectors (DVs) are extracted in the subsequent (intermediate level) step. Note that direct silhouette matching with a template of activities is not always optimal, as silhouette shape generally changes non-rigidly depending on clothing, activity and is also specific to individuals. In order to achieve scale invariance, we normalize the extracted directional vectors. At the high-level vision step, we perform vector space analysis and clustering of the DVs to compute the activity decisions for each frame, and smooth these decisions over time to maintain smooth activity transitions. The details of each step are presented in the following section.

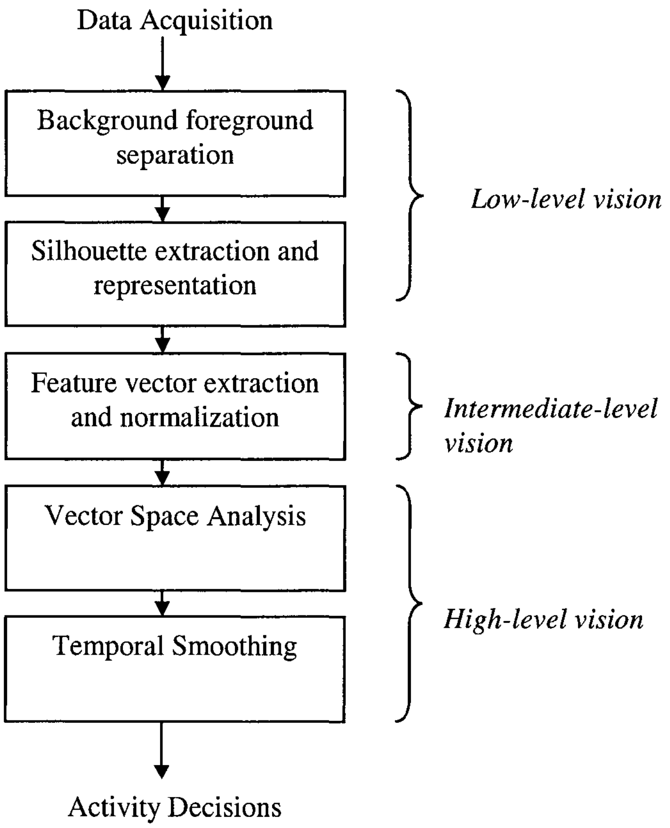

Data Acquisition
Background foreground separation
Silhouette extraction and representation
Feature vector extraction and normalization
Vector Space Analysis
Temporal Smoothing
Activity Decisions
Low-level vision
Intermediate-level vision
High-level vision

Figure 5.1: Flowchart of the proposed algorithm.

5.1 Background-Foreground Separation

Separation of the background and foreground is the first step of the proposed algorithm. In Section 2.3 we reviewed several methods such as background subtraction, optic flow and statistical models for separating the moving foreground from a relatively static background. In the proposed algorithm we use the statistical background modeling method. Here, we generate a statistical background model by computing the mean and variance of intensity of each pixel over a set of initial frames. This method has been found to be more robust to noise, shadow and change in light conditions than simple background subtraction or optic flow. We assume a relatively stationary background and use an adaptive threshold $\tau(x, y)$ for each pixel $p(x, y)$, assuming noise at each pixel to be time varying. Each pixel of the current frame is thesholded against the corresponding pixel of the background model to extract foreground information. We restrict our algorithm to monochrome images (if the original images are in color, then they are converted to intensity images). The mean intensity $\mu(x, y)$ at location (x, y), corresponding to the 'N' initial frames is computed as:

$$\mu(x, y) = \frac{1}{N}\sum_{i=1}^{N} p(x, y; i)$$

where, $p(x, y; i)$ is the pixel value at location (x, y) in the 'i^{th}' frame.

The threshold for each pixel in the background model is calculated using the following equation.

$$\tau(x, y) = \max\{|\mu(x, y) - p(x, y; i)|\} \quad \text{for} \quad (1 \le i \le N)$$

To obtain the foreground we classify each pixel in frame 'k' ($k > N$) into the foreground bitmap $F(x, y : k)$ according to the following inequality:

$$\begin{aligned}
&if \quad |p(x,y;k)-\mu(x,y)| < \tau(x,y)\\
&\qquad \textit{pixel is background} \;\; F(x,y:k)=0\\
&else\\
&\qquad \textit{pixel is foreground} \;\; F(x,y:k)=1
\end{aligned}$$

where $F(x,y:k)=\begin{cases}0; & \textit{background}\\ 1; & \textit{foreground}\end{cases}$

5.2 Silhouette Extraction and Representation

Subsequent to background-foreground separation, each video frame is represented as a bi-level foreground-background image. In order to extract the silhouette, noise present in the foreground is removed by performing morphological operations such as erosion and dilation. In our experiments, we did not encounter cases with unconnected foreground. However, in cases where the foreground is broken, connected component analysis can be used to link the broken components. Subsequently, the silhouette contour is obtained by filtering the foreground bitmap frame with a Laplacian of Gaussian filter. The results of these operations are illustrated in Figure 5.2.

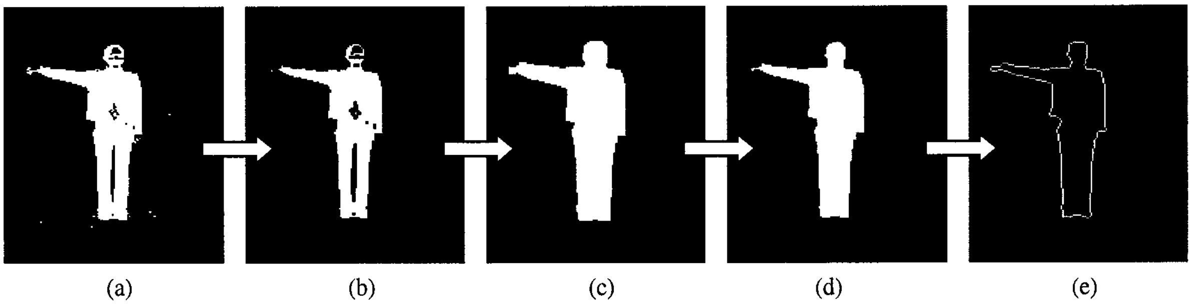


Figure 5.2: Illustration of silhouette extraction. (a) Separated foreground, (b) First erosion, (c) Dilation, (d) Second erosion, (e) Edge extraction.

Since the silhouette is often used to derive the feature vectors for classification, the method of silhouette representation is of particular importance. In the past, silhouette contours have been represented using various techniques. Cuntoor *et al.* [14] represented the silhouette contour as distance of the contours from reference vertical and horizontal lines. Wang *et al.* [27] computed the complex coordinate representation of the silhouette contour using the centroid of the silhouette as the origin of a complex coordinate system. We represent the silhouette contour as a chain code. We assume that each pixel is connected to its eight neighbors, and therefore eight chain code vectors are used (see Figure 5.3(d)). We traverse the silhouette contour from the highest-leftmost point in a clockwise direction to generate a chain-code signature of the silhouette contour (see Figure 5.3). Since the chain code is cyclic in nature, the starting point of the code does not affect the results of the algorithm.

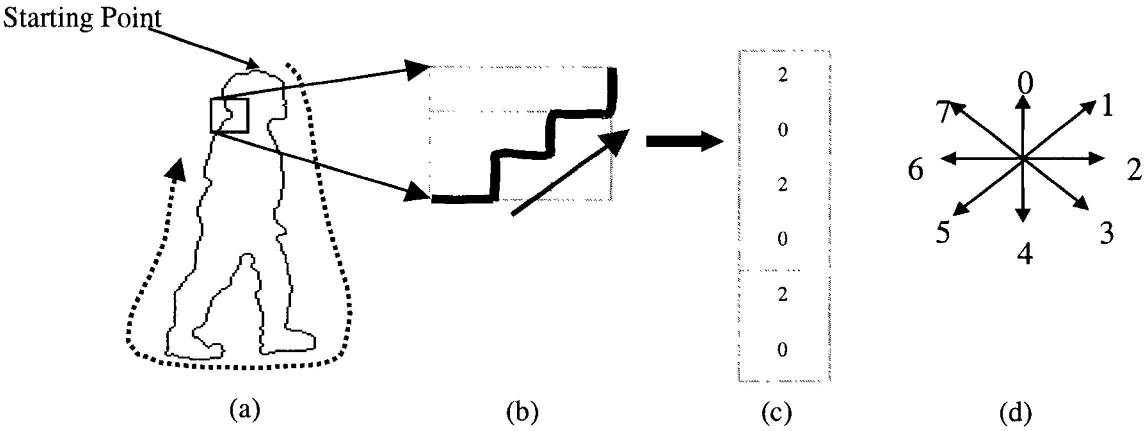


Figure 5.3: Illustration of silhouette representation. (a) Silhouette contour (color inverted), (b) Pixelized section of contour, (c) Chain code, (d) Chain code vectors used.

5.3 Feature Vector Extraction and Normalization

Classification requires that relevant features be extracted from the chain code representation of the silhouette contour. In order to determine a generic feature to extract from any silhouette representation, let us consider a relation $\Re$ of the silhouette representation. This relation $\Re$ can have n attributes X_j $(j = 1,2,..,n)$ such as directionality, color and texture. The value set $V_j = (v_j(1), v_j(2)...v_j(\kappa))$ of attribute X_j is the set of 'κ' values of attribute set X_j that are represented in $\Re$. For example, if the attribute in question is palletized color (with pallet size of 256), then the value set V_j will be the set of 256 colors present in the relation $\Re$. The individual elements of the data distribution should always belong to a value set. The frequency $f_j(k)$ of a value $v_j(k)$ is the number of tuples in $\Re$ with attribute $X_j = v_j(k)$. The silhouette representation generated from each frame can be approximated as a data distribution D_j. In the case where the silhouette is represented as a chain code, the data distribution D_j can be evaluated as the histogram of the chain code generated for each video frame. Note that the distribution of an attribute X_j will be the set of pairs shown below:

$$D_j = \{[v_j(1), f_j(1)], .. [v_j(\kappa), f_j(\kappa)]\} \tag{5.1}$$

We construct a data distribution on the 'directionality' attribute (henceforth ignoring the subscript '*j*') by partitioning the data distribution of X into β (=8) mutually disjoint subsets. When grouping into dyads (*i.e.,* by considering two chain code vectors at a time) and triads (three chain code vectors at a time), β increases depending on the constraints applied. A serial partition class is used with frequency as the sort parameter to obtain the distribution of chain code vectors. This distribution of chain code vectors is the directional vector derived from the silhouette contour for each frame.

An effective activity recognition algorithm needs to be invariant to scale changes that are caused by motion of a subject towards the camera or vice versa. Polana *et al.* [19] achieved spatial scale invariance by measuring the size of the object through successive frames and estimating spatial scale parameters. These scale parameters are then used to compensate for changes in scale. This approach is computationally more expensive. Furthermore, Polana *et al.* assumed that the height of the object of interest does not change over time, an assumption that is true only for a few activities such as walking and running. In our proposed algorithm, spatial scale invariance is achieved by normalizing the directional vectors, such that the mean is zero and all values lie between [-1,1]. If $f(i)$ is the frequency of the 'i^{th}' tuple of the data distribution D (see Eq. 5.1) (or in other words the i^{th} component of the directional vector), the normalized frequency (or normalized directional vector component) can be written as:

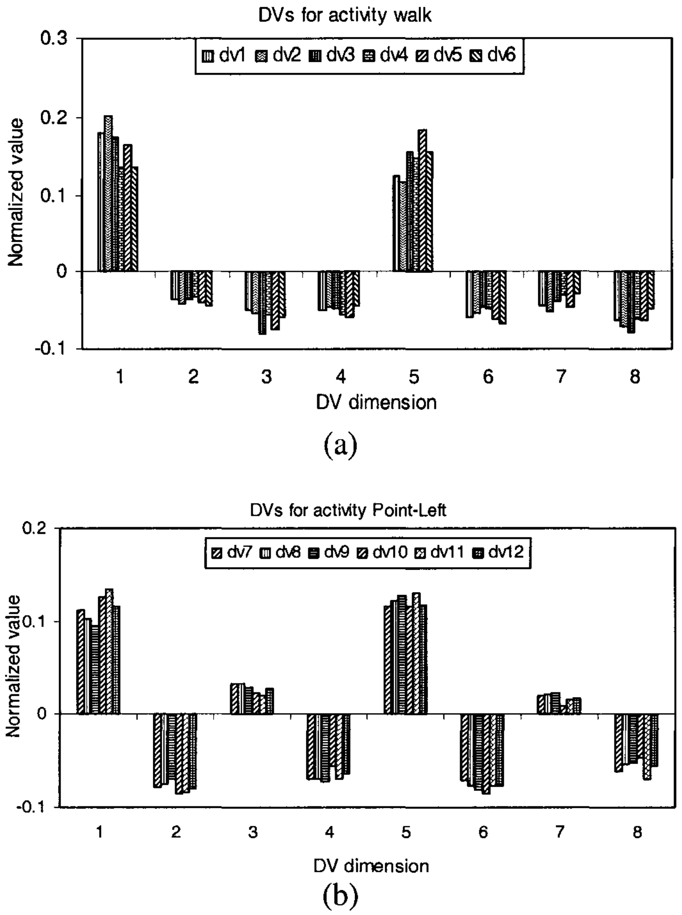
$$\phi_i = \frac{f(i)}{\sum_{i=1}^{\kappa} f(i)} - \frac{1}{\kappa} \qquad \text{for} \qquad 1 \leq i \leq \kappa \qquad (5.2)$$

The proposed algorithm is based on the following assumptions of the normalized directional vectors.

1. The normalized directional vectors, for different subjects performing the same activities at the same distance from the camera, have small variance.
2. The variance in the normalized directional vectors, for different subjects performing the same activities at varying distances from the camera (implying scale invariance) and with varying backgrounds, is also small.

3. The variance in the normalized directional vectors for different activities is high.

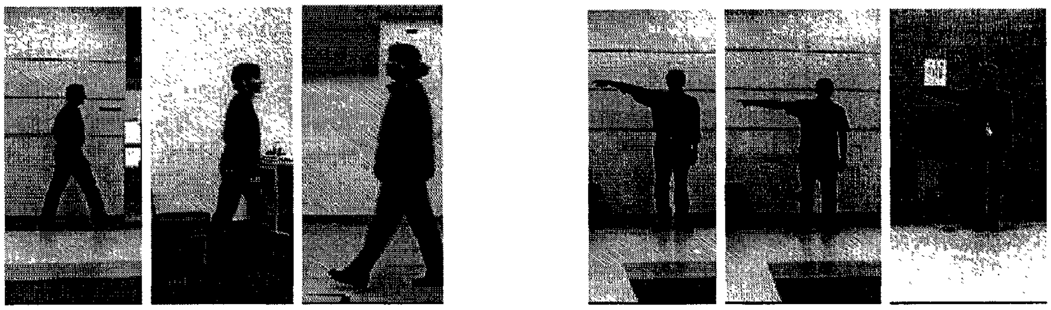
In order to validate Assumptions 1 and 2 we compute the variance between the DVs derived from different frames that consist of the same activity. The DVs for activity 'walk' and 'point left' are shown in Figure 5.4(a) and (b), respectively. Note that we designate left and right as directions with respect to the viewer and not the subject. Some of the frames, represented in Figure 5.4 as dvl-12, are shown in Figure 5.5. It can be seen from these plots that the variance within DVs for the same activity (in this case walk and point) is small. We also present the validation of Assumptions 1 and 2 for activities stand, sit, point right, and lie down in Figures 5.6-5.9.



(a)

(b)

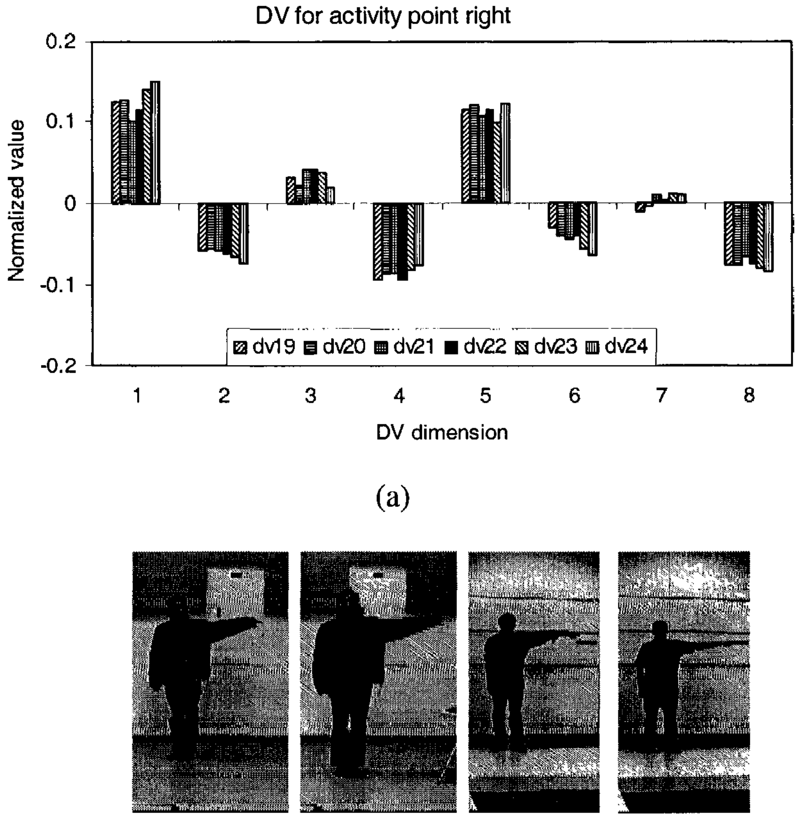
Figure 5.4: Plots of normalized DVs for six frames of activity: (a) walk and (b) point left.



(a)

(b)

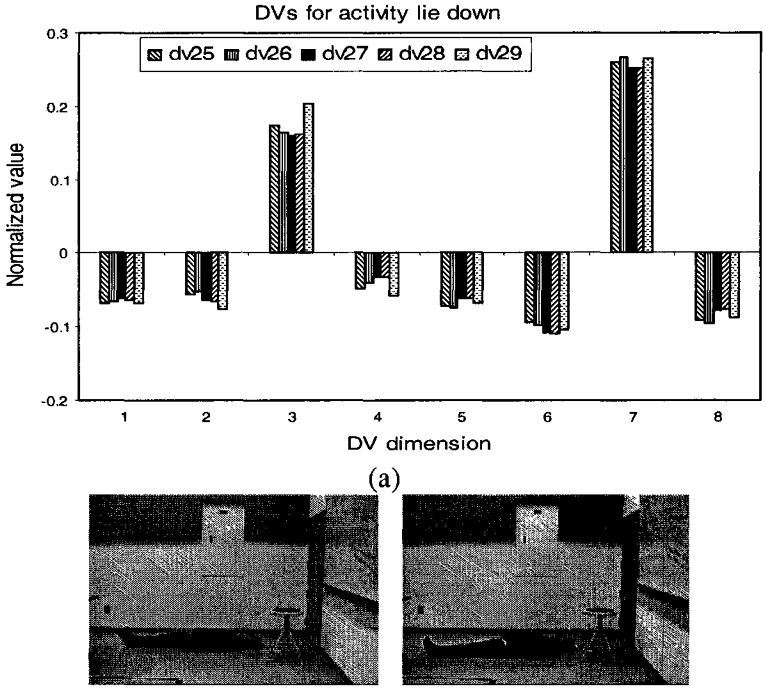
Figure 5.5: Sample frames from different data sets, for which DV plots are illustrated in Figure 5.4: (a) dv1, dv3, dv4 corresponding to Figure 5.4(a), and (b) dv9, dv10, dv11 corresponding to Figure 5.4(b).



(a)

(b)

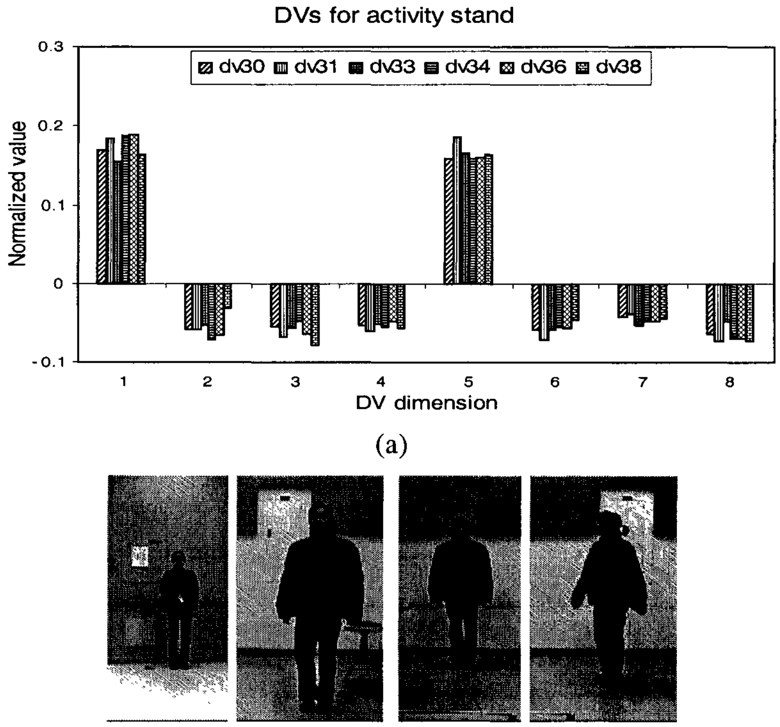
Figure 5.6: Low variance in DVs for activity point right. (a) Plots of normalized DVs for six frames of activity point right, and (b) Sample frames used to compute Figure 5.6(a).



(a)

(b)

Figure 5.7: Low variance in DVs of activity lie down. (a) Plots of normalized DVs for five frames of activity lie down, and (b) Sample from frames used to compute Figure 5.7(a).

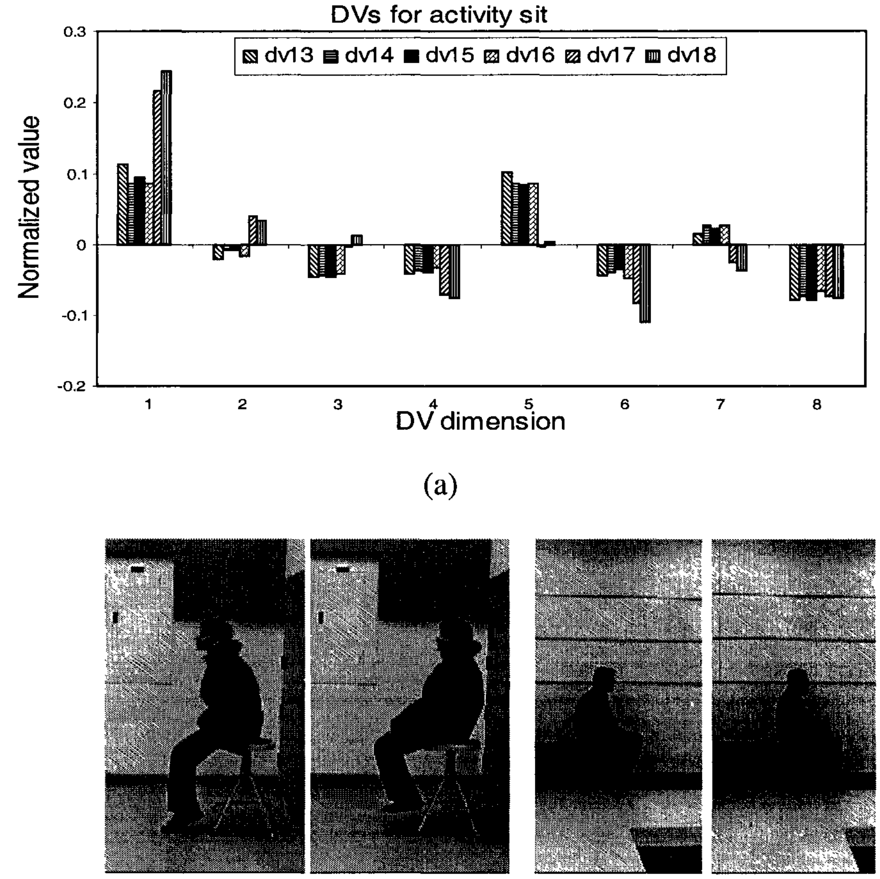


(a)

(b)

Figure 5.8: Low variance in DVs of activity stand. (a) Plots of normalized DVs for nine frames of activity stand, and (b) Sample from frames used to compute Figure 5.8(a).

Note that the normalized DV plot for activity sit (see Figure 5.9(a)) has two somewhat distinctive DV patterns. These two patterns correspond to the DVs generated from frames with the subject sitting facing left and subject sitting facing right (see Figure 5.9(b)).



(a)

(b)

Figure 5.9: Low variance in DVs of activity sit. (a) Plots of normalized DVs for six frames of activity sit. (b) Sample from frames used to compute Figure 5.9(a).

In order to validate Assumption 3, we compute the mean DVs for each of the five different activities and then analyze the variance within these mean directional vectors. Figure 5.10 is a plot of the mean directional vectors for five different activities. It can be seen from Figure 5.10 that the DV patterns are quite distinct for different activities.

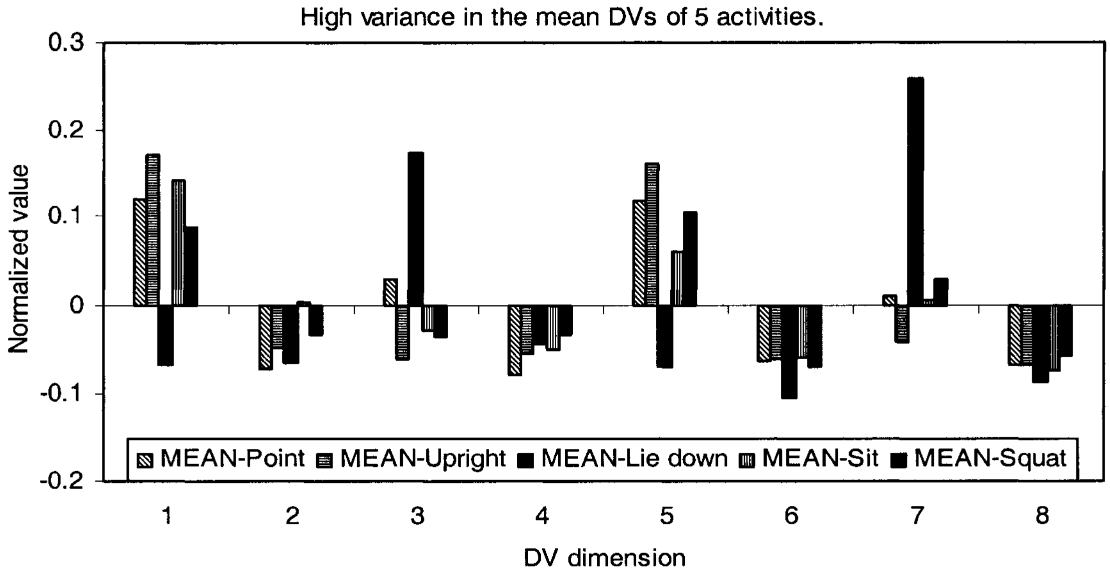


Figure 5.10: Plot of mean normalized DVs for five activities: Point, Upright, Lie-down, Sit and Squat.

5.4 Vector Space Analysis

We represent the normalized directional vectors extracted from the chain code of the silhouette contour as eight dimensional vectors in 'activity space.' Let us consider a sequence of 'N' frames of a video. For each frame, we extract and normalize a DV (see Section 5.3) $d_i = [w_{i0}, w_{i1},..w_{i(X-1)}]$ where w_{ij} implies coordinate of directional vector 'i' in the 'j$^{\text{th}}$' dimension ($0 \leq j \leq X-1$) and X is the dimensionality of vector space. Thus, a length 'N' video sequence can be represented as a set of vectors $\Psi = [d_1, d_2 ..., d_N]$. The extracted DVs are analyzed by defining an angular distance parameter Ω_d between two directional vectors. Other parameters such as Mahalanobis distance (Bobick *et al.* [20]) and the Procrustes distance (Wang *et al.* [27]) have been used in the past for feature vector analysis. The Procrustes distance [28] between two complex vectors u_1 and u_2 is defined as:

$$P_f = 1 - \frac{\left|u_1^* u_2\right|}{\left(u_1 \times u_1^*\right)\left(u_2 \times u_2^*\right)} \tag{5.3}$$

where, $u_1^* = \left|\overline{u_1}\right|^T$ and $\overline{u}_1$ is the complex conjugate of u_1.

The angular distance parameter Ω_d between two directional vectors $\vec{d} = \left[w_{d0}, w_{d1} \ldots \; , w_{dX-1}\right]$ and $\vec{q} = \left[w_{q0}, w_{q1} \ldots. \; , w_{q(X-1)}\right]$ where $d, q \in \Psi$, is determined as:

$$\Omega_d = 1 - \frac{\vec{d} \cdot \vec{q}}{\left\|\vec{d}\right\| \left\|\vec{q}\right\|} = 1 - \cos(\alpha) \tag{5.4}$$

where, $\cos\left(\alpha\right) = \frac{\vec{d} \cdot \vec{q}}{\left\|\vec{d}\right\| \left\|\vec{q}\right\|}$ and α is the angle between the vectors. The angular distance parameter Ω_d in Eq. 5.4 can be represented as:

$$\Omega_d = 1 - \cos(\alpha) = 1 - \frac{\sum_{i=0}^{X-1} w_{di} w_{qi}}{\sqrt{\sum_{i=0}^{X-1} w_{di}^2} \sqrt{\sum_{i=0}^{X-1} w_{qi}^2}} \tag{5.5}$$

Because we normalized the directional vectors (see Eq. 5.2), $\left\|\vec{d}\right\| = \left\|\vec{q}\right\| = 1$, and hence Eq. 5.5 reduces to:

$$\Omega_d = 1 - \cos(\alpha) = 1 - \sum_{i=0}^{X-1} w_{di} w_{qi} \tag{5.6}$$

We cluster frames with similar activities based on the angular distance between the directional vectors derived for each frame. The vectors are clustered using an eight dimensional K-means clustering algorithm [94], such that for all pairs of vectors

$[(d_1, d_2),\ (d_1, d_3) \ldots\ (d_1, d_N)]$, Ω_d is minimized. The clustering of activities is hierarchical (see Figure 5.11), and activity resolution increases with increasing level.

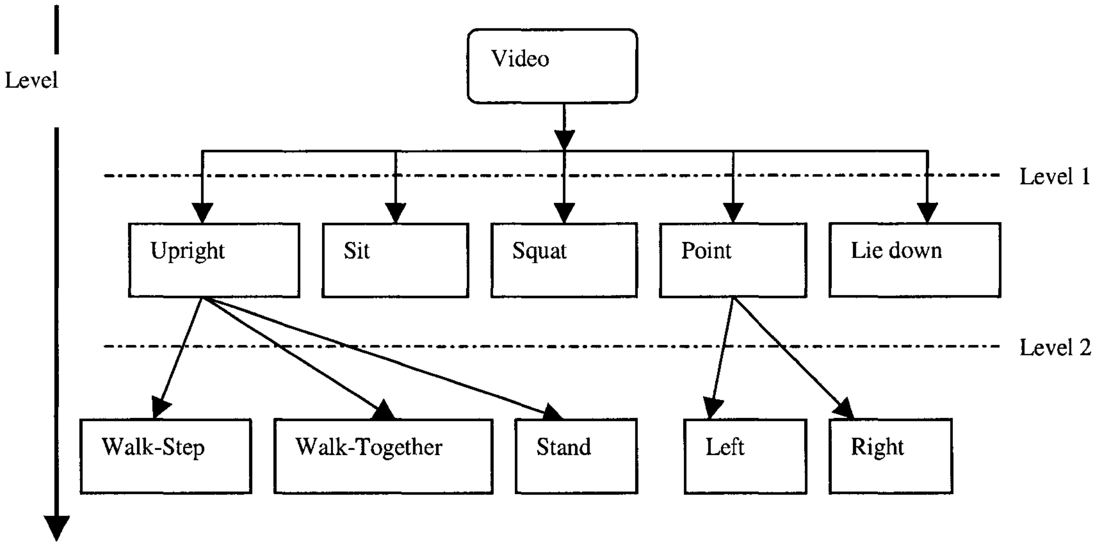


Figure 5.11: Hierarchical clustering levels for higher activity recognition resolution.

A known issue with K-means clustering algorithm is that it can be sensitive to the initial centers, and the search for the optimum center locations may result in poor local minima. We perform ten iterations of the K-means clustering, and choose clusters for which the angular distance parameter computed in the iterations is minimum. This resolves the problem of poor local minima convergence. The cluster centroids for each activity can be obtained from the key frames (see Figure 5.12) of the training set. These centroid locations can be used as seeds for clustering in the recognition stage. However, this method of solely using the cluster centroids for classification will not allow the system to learn independently from each sequence. We therefore use the cluster centroids only to identify an activity, and not to segment it from the video. At the end of this analysis, an activity decision ' Λ ' is associated with each frame of the video sequence.

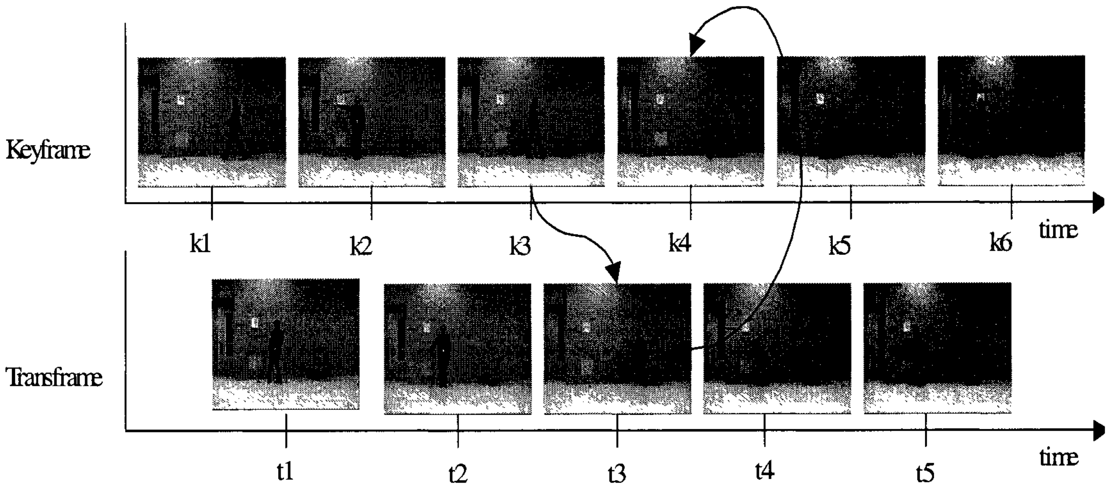


Figure 5.12: Sequential representation of key frames and transitional frames. t3 is one of the transitional frames between k3 depicting 'walk activity' and k4 depicting 'sit activity'.

5.5 Temporal Smoothing

The proposed algorithm can make error in recognizing non-rigid human activities due to poor foreground-background separation in some frames. This poor foreground extraction can lead to large variance in the directional vector of neighboring frames resulting in incorrect decisions. In order to overcome this problem, we use the dynamic characteristics of human motion and assume that activities cannot change suddenly from one frame to another and must undergo a smooth transition. This assumption may not be valid for very low frame rate video capture; in such cases, activities can change from one frame to the immediate next. Thus, for low frame rate videos temporal smoothing does not give significant performance enhancement and may actually lead to deterioration. We found that for capture rate of 6 fps and above, temporal smoothing increases the correct recognition rate of activities.

Table 5.1: Numeric values assigned to activities

Activity	Lie Down	Squat	Sit	Upright	Point
Numeric Value	1	2	3	4	5

In order to achieve temporal smoothing we associate a numeric value (as shown in Table 5.1) with each activity decision. The smoothing is performed in two steps. In the first step we mark frames that could potentially be in error. In the second step, we compute the mean decision over a larger window size, and based on the numeric value closest to this mean we correct the activity-decisions. Let $\Lambda(i)$ be the numeric value of the decision made at frame i. In the first step, we determine frames that have random activity transitions (using a filter length of *2M+1*) and mark them as potential abnormalities by flagging them by ξ, as shown in Table 5.2. The parameter ξ as such does not have any value, and is simply a flag. In the second step, we use a larger filter window *(2K+1)* where $K \geq M$ to compute the true decision of frames that have been flagged by ξ, as shown in Table 5.3. We also associate decision weights $W=\{w_{-K},..w_0..,w_K\}$ with each filter tap. These weights determine the influence a decision has on preceding or succeeding decisions. Recursive execution of these two iterations of the temporal smoothing filter eliminates local peaks and valleys.

Table 5.2: Algorithm for temporal smoothing, iteration 1.

```
Iter1:
if $\Lambda(i) \notin \{\Lambda(i-M+p)\}$ where $p \neq M, 0 \leq p \leq 2M$
%implies decision made for 'i' is different from its '2M' nearest neighbors.
then {
    $\Lambda(i)=\frac{1}{2M+1}\sum_{p=0}^{2M}\Lambda(i-M+p);$
    $\Lambda_{decision}(i)=\xi$ }
% possible error
```

Table 5.3: Algorithm for temporal smoothing, iteration 2.

```
Iter2:
if $\Lambda_{decision}(i) = \xi$
then
    $\Lambda(i) = \frac{\sum_{n=i-K}^{n=i+K} w_n \Lambda(n)}{2K+1}$ for i>K
```

For example, let the decisions made for a sequence of 10 frames be as shown in Table 5.4, with an incorrect decision having been reached for Frame 4. Table 5.4 also lists the results obtained after the first and second iteration of temporal smoothing (M=1, K=2, w_n=1). It can be seen that smoothing has successfully been able to correct the erroneous decision previously reached at Frame 4.

Table 5.4: An example to illustrate temporal smoothing of activity decisions.

Frame	1	2	3	4	5	6	7	8	9	10
Activity	Point	Point	Point	Squat	Point	Point	Point	Point	Point	Point
Numeric value Λ	5	5	5	2	5	5	5	5	5	5
Iter-1	5	5	5	4(ξ)	5	5	5	5	5	5
Iter-2	5	5	5	5	5	5	5	5	5	5

Figure 5.13 shows a plot of the four cluster level 1 activities recognized in each frame of a test video captured at 10 fps. The un-smoothed plot shows local peaks, which represent misclassified frames, while the smoothed plot exploits the dynamics of motion to reclassify frames and reduce recognition error using temporal smoothing.

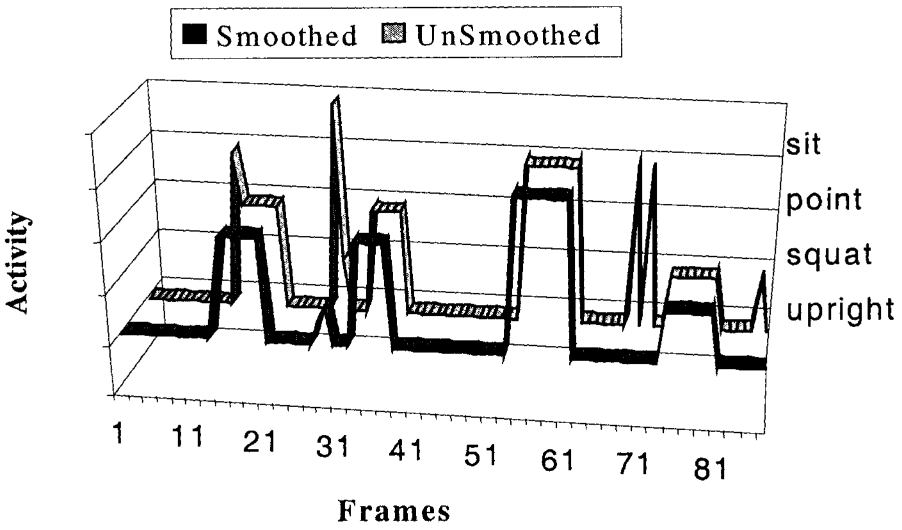


Figure 5.13: A sample 3D plot illustrating the removal of local peaks by temporal smoothing.

5.6 Experimental Setup

In this section, we discuss the experimental setup used to evaluate the performance of the proposed algorithm. In Section 5.6.1, we give a brief overview of the databases used in our experiments. The data sets created from these databases are explained in Section 5.6.2. The set-up of the experiments and numeric values of the various parameters used are presented in Section 5.6.3.

5.6.1 Databases

Several gait databases have been developed for gait-based human identification. However, there is no standard database available for the purpose of HAR. We have used two gait databases (University of Southampton UoS-HID [24] and Carnegie Mellon University CMU-Mobo [25]) originally developed for human identification based on gait analysis. We now briefly discuss the characteristics of the databases used in our experiments.

University of Southampton Database UoS-HID: The UoS-HID database consists of video sequences of subjects walking in front of a uniform stationary background. The average sequence length is 60 frames; captured at a frame rate of 25 frames per second. It is primarily a gait database and we randomly choose some key frames from these sequences to train the proposed algorithm for activity-walk.

Carnegie Mellon University CMU-Mobo: The CMU-Mobo database consists of video sequences of subjects walking on a treadmill. The average sequence length is 300 frames; captured at a frame rate of 30 frames per second. As with the UoS-HID, we randomly choose some key frames from this database to train the algorithm for activity-walk.

University of Texas at Austin UoT-DB: We used an activity database provided by the University of Texas at Austin (UoT-DB) that consists of image frames depicting the following activities: sit, squat, stoop and walk. The sequences from this database vary in length from 60-80 frames, captured at 12-15 frames per second and consist of four activities: sit, squat, stoop and walk.

University of Alberta UoA-DB: We have created our own activity database (UoA-DB), comprising videos of six activities: standing, walking, sitting, squatting, lying down and pointing.

Capturing Device: We used a SONY DCR-PC100 CCD camera to capture indoor and outdoor activity videos with frame size of 780x480 at 30 frames per second. The UoA-DB was developed with subjects of different physical builds; wearing indoor as well as bulky out-door clothing; moving in front of a stationary camera with static lighting conditions and a relatively static background (outdoor sequences had background movement). The videos were captured with varying zoom depths and backgrounds. Actions were performed such that the view angle changed frequently and limited occlusion occurred.

Transfer Protocol: A fire wire IEEE 1394 interface was used to transfer video sequences captured in the DV format to the computer.

Video Software: A commercial software package 'UleadStudio' was used to convert the captured files into uncompressed avi format. This conversion was required because MATLAB only supports uncompressed avi file format. The UleadStudio software was also used to temporally down-sample the video sequences to 6-15 frames per second and spatially decimate each frame to size 360x240. This step not only reduced the size of the data set (thus enabling faster computation) but also provided limited blurring, which smoothed the derived silhouette contour to an extent.

Software: The entire algorithm was implemented in MATLAB version 6.5. Some of the functions of the image processing toolbox were used to perform the file I/O and morphological operations.

We constructed four sequence sets (UoA-DS1, UoA-DS2, UoA-DS3 and UoA-DS4) of video sequences from the UoA-DB. We will now discuss them briefly.

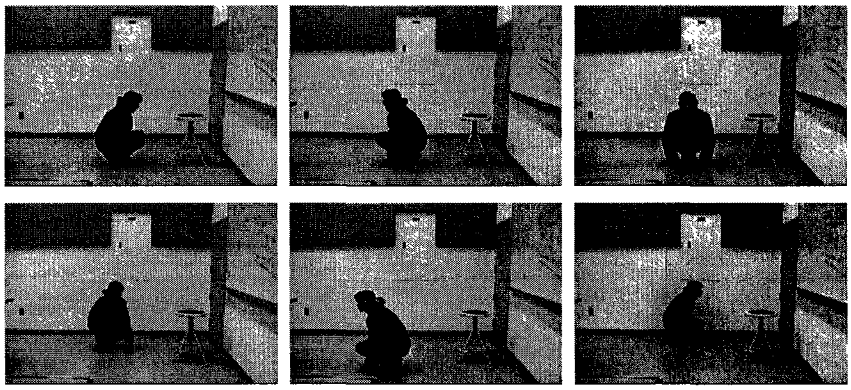
1. Sequence set UoA-DS1 consists of two video sequences of length 133 and 138 frames. These sequences were captured indoors and depict subjects performing activities walk, stand and sit.

2. Sequence set UoA-DS2 consists of five video sequences captured indoors, having a general sequence of activities but not limited to: walk→ stand→ point-

left→ stand→ point-right→ walk→ sit→ walk→ squat→ walk. We also sampled these video sequences at varying frame rates of 6, 10 and 15 fps, consequently the sequences range in length from 48 to 286 frames.

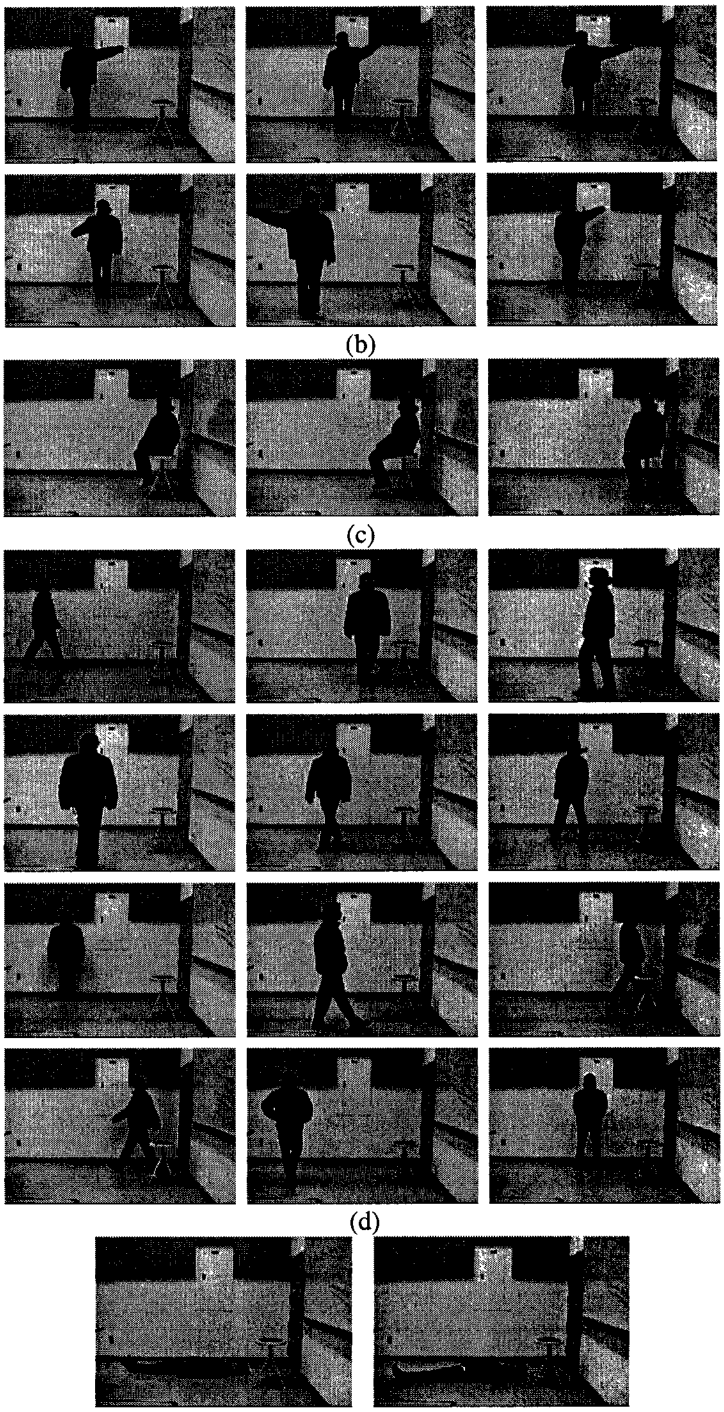
3. Sequence set UoA-DS3 consists of a 7130 frame long video sequence captured indoors at 15 fps. The sequence consists of 65 random repetitions of 5 (level1) activities captured from varying viewpoints (Figure 5.14). A new activity, lie down, is also introduced in order to observe the behavior of the algorithm on encountering an activity for which training data is not available.

4. Sequence set UoA-DS4 consists of two video sequences captured outdoors at 15 and 6 fps. The sequences consist of random repetitions of three activities: Upright, point and squat. This set is used to test the robustness of the directional vectors in outdoor sequences.



(a)

Figure 5.14: Sample frames from UoA-DS3 illustrate the varied view angle and zoom depth of subject from camera for the activities: (a) squat.



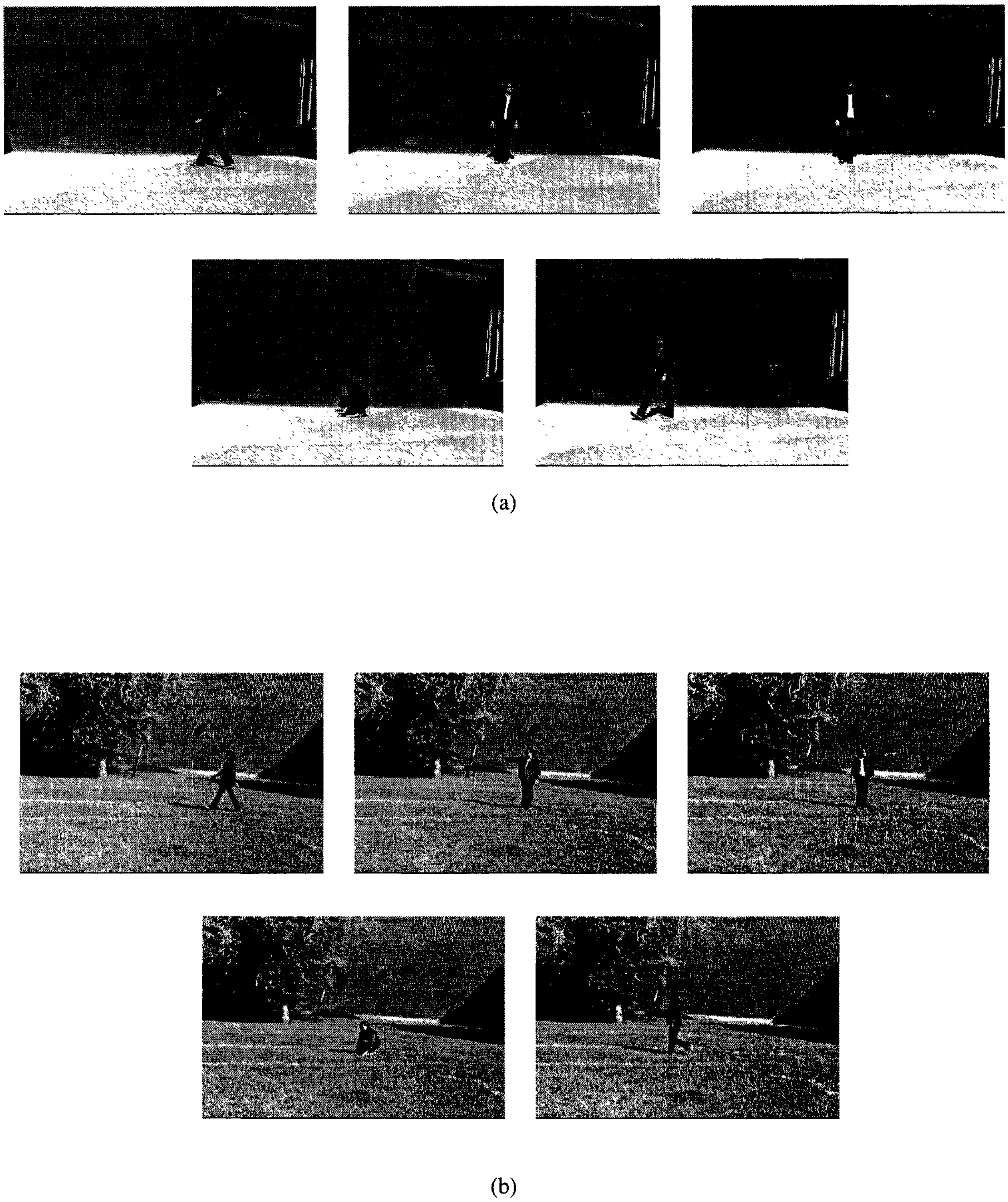
(b)

(c)

(d)

(e)

Figure 5.14: (continued) Sample frames for the activities: (b) point, (c) sit, (d) upright and (e) lie-down.



(a)

(b)

Figure 5.15: Sample frames from the outdoor sequences of UoA-DS4 data, (a) Sequencel and (b) Sequence2.

5.6.2 Data sets

We create four data sets from the databases mentioned in Section 5.6.1. These sets are created such that data used for training and recognition remains mutually exclusive. This ensures that the recognition phase of the algorithm remains unbiased.

Data Set 1

The Data Set-1, used exclusively for training, was comprised of the key frames derived from UoS-HID, CMU-Mobo and UoA-DS1 (see Table 5.5). The key frames from UoS-HID and CMU-Mobo are used to train the algorithm for upright (walk) activity, while key frames from UoA-DS1 are used to train for activities sit and upright (walk-stand).

Table 5.5: Details of Data Set-1

Database	Type of activity	Length (frames)	fps	Frame size	Comments
University of Southampton (UoS-HID)	Walk	60	25	384x288	Gait ID database, use only key frames
Carnegie Mellon University (CMU-Mobo)	Walk	300	30	640x480	Gait ID database, use only key frames
University of Alberta (UoA-DS1)	Walk, Stand, Sit	138	30	360x240	Activity database, use only key frames
	Walk, Stand, Sit	133	30	360x240	

Data Set 2

The Data Set-2 was comprised of five video sequences from UoA-DS2 (see Table 5.6).

Table 5.6: Details of Data Set-2

Database	Sequence	Type of activity	Length (frames)	fps	Frame size	Comments
University of Alberta (UoA-DS2)	Sequence 1	Walk, Sit, Squat, Point	76	6	360x240	Activity database, varied frame rate.
			210	15		
	Sequence 2	Walk, Sit, Squat, Point	58	6	360x240	Activity database, varied frame rate.
			96	10		
			286	30		
	Sequence 3	Walk, Sit, Squat, Point	65	6	360x240	Activity database.
	Sequence 4	Walk, Sit, Squat, Point	48	6	360x240	Activity database.
	Sequence 5	Walk, Sit, Squat, Point	61	6	360x240	Activity database.

Data Set 3

The Data Set-3 was comprised of video sequences from UoT-DB and UoA-DS3 (see Table 5.7). Data Set-2 and Data Set-3 are used as mutually exclusive training and recognition sets for each other.

Table 5.7: Details of Data Set-3

Database	Type of activity	Length (frames)	fps	Frame size	Comments
University of Texas at Austin (UoT-DB)	Walk, Sit, Stoop, Squat	60-80	12-15	640x480	Activity database.
University of Alberta (UoA-DS3)	Walk, Sit, Squat, Point, Lie down	7130	15	360x240	Activity database.

Data Set 4

Data Set-4 was comprised of outdoor video sequences from UoA-DS4 and is used exclusively for recognition only (see Table 5.8).

Table 5.8: Details of Data Set-4

Database	Sequence	Type of activity	Length (frames)	fps	Frame size	Comments
University of Alberta (UoA-DS4)	Sequence 1	Walk, Squat, Point	227	15	360x24 0	Activity database, outdoor sequence.
	Sequence 2	Walk, Squat, Point	68	6	360x24 0	Activity database, outdoor sequence.

5.6.3 Experiments

We have performed our experiments in two phases — training and recognition. In the training phase, we generate cluster centers of activities from the key frames of the corresponding training data. In the recognition phase, we cluster the normalized directional vectors from all the frames of the test videos (key as well as transitional frames). The cluster centers generated in the training phase are then used to map an activity to the cluster centers generated in the recognition phase. Recognition begins only when the whole body is visible in the frame, and hence the DVs corresponding to the preceding frames are discarded. We used a filter window of size $M = 1, K = 2$, with uniform decision weights $\{w_{-K}, ...w_0...w_K\} = 1$ for the temporal smoothing filter.

In order to evaluate the performance of the proposed algorithm we performed three experiments. The experiments were conducted such that the datasets acted as complementary training and recognition data for each other.

Experiment 1

In the first experiment, we trained the algorithm by generating cluster centers from video sequences of Data Sets 1 and 3. We then clustered the DVs derived from video frames of the test set (Data Set 2) and used the training data to map activities. We also tested the performance of the algorithm with varying frame rates of the sequences in Data Set 2.

Experiment 2

In the second experiment, we generated the training data from Data Sets 1 and 2, and recognized activities in Data Set 3.

Experiment 3

The third experiment dealt with outdoor video frames. Data Sets 1, 2 and 3 were used for the purpose of training and Data Set 4 was tested for recognition. The training-recognition correspondence between the data sets is illustrated in Table 5.9.

Table 5.9: Recognition sets and training sets for different experiments.

	Training Set	Recognition Set
Experiment–1	Data set 1 & 3	Data set 2
Experiment–2	Data set 1 & 2	Data set 3
Experiment–3	Data set 1,2 & 3	Data set 4

5.7 Performance Evaluation

We use two criteria to evaluate the performance of the proposed algorithm — correct recognition rates and confusion matrix. The correct recognition rate (CRR) is defined as the percentage of frames recognized correctly from all the tested input frames of the video sequence. A five-class confusion matrix is also used to compare the actual and predicted classifications. Table 5.10 reports the CRR obtained by our algorithm before temporal smoothing in Experiment 1 and for the UoT-DB sequence in Experiment 2. At low frame rates the algorithm makes more recognition errors as there are fewer cluster points to learn from and generate cluster centers. However, applying a smoothing filter in the temporal domain removes the local error peaks and valleys, leading to a 100% CRR. This filtering is not optimal when the frame rate is low as the influence that each decision has, on its predecessors and successors, is significantly reduced.

Table 5.10: CRR without temporal smoothing for Experiment 1 (L1 and L2 correspond to activities at level 1 and 2, respectively, of Figure 5.11).

Video	Length (frames)	Fps	CRR L1	CRR L2-Point	CRR L2-Upright
Sequence 1	76	6fps	98.68	100	76
Sequence 2	58	6fps	96.55	100	84.21
Sequence 3	65	6fps	100	100	91
Sequence 4	48	6fps	96.77	100	93.75
Sequence 5	61	6fps	100	100	82.5
Sequence 1	210	15fps	100	100	98.11
Sequence 1	76	6fps	98.68	100	76
Sequence 2	286	30fps	98.6	100	96.88
Sequence 2	96	10fps	96.87	100	96.55
Sequence 2	58	6fps	96.55	100	84.21

Table 5.11: Confusion matrix for UoA-DS3 Experiment 2 (overall efficiency= 95.5%)

		Recognized Activities						
		Upright	Sit	Lie down	Point	Squat	TP rate	FP rate
Actual Activities	Upright	3919(a)	47(b)	8(c)	152(d)	21(e)	98.7	0.0168
	Sit	45(f)	1010(g)	0(h)	2(i)	1(j)	94.3	0.0101
	Lie down	1(k)	0(l)	847(m)	1(n)	33(o)	99.1	0.0012
	Point	1(p)	13(q)	0(r)	334(s)	10(t)	67.1	0.0247
	Squat	6(u)	1(v)	0(w)	9(x)	669(y)	91.1	0.0101
	Total	3972	1071	855	498	734		

The recognition results of the algorithm for Experiment 2 with Data Set UoA-DS3 are presented in Table 5.11, where the confusion matrix [26] for five activities is shown. The true positive (TP) or recall rates of recognition are defined as the percentage of positive cases that are correctly identified. Note that the TP values are calculated using Eq. 5.7. The accuracy of the proposed algorithm can be determined from the confusion matrix using Eq. 5.8.

$$TP_{upright} = \frac{True \quad walks}{All \quad positive \quad cases \quad of \quad walks} = \frac{a}{a+b+c+d+e} \tag{5.7}$$

$$Overall \quad Efficiency = \frac{True \quad activities}{All \quad activities} = \frac{a+g+m+s+y}{\sum confusion \quad matrix} \tag{5.8}$$

In our experiments, out of the 65 random instances of activities in the UoA-DS3 set, 63 were correctly recognized, resulting in an activity CRR of 96.92%. Since the classifier used is non-parametric, a receiver operating characteristics (ROC) plot will comprise of points and not curves, and hence the ROC plot has been omitted from this discussion.

The CRR for Experiment 2 and Experiment 3 are reported in Table 5.12. In Experiment 2, 61 out of 63 frames from the video sequence UoT-DB are recognized correctly, resulting in a CRR of 96.8%. In the same experiment with the UoA-DS3 video sequence of length 7130, 6779 frames are correctly recognized, resulting in a CRR of

95.5%. In Experiment 3, with outdoor video sequences, the CRR evaluated are 100% (all 227 frames recognized correctly) and 98.5% (1 frame out of 68 was incorrectly recognized) for Sequence 1 and 2 respectively.

Table 5.12: CRR without temporal smoothing for Experiment 2 & 3.

	Video	Length (frames)	fps	CRR L1
Experiment2	UoT-DB	63	Not Known	96.8
	UoA-DS3	7130	15	95.5
Experiment3	UoA-DS4			
	Sequence1	227	15	100
	Sequence2	68	6	98.5

5.8 Potential Applications

We have identified two promising areas that would benefit directly from an automated computer vision system for recognizing select human activities, *i.e.* potential areas of direct application of this work. Further work and enhancements to the algorithm, which are mentioned as future work in Chapter 7, will open more avenues for application. The first area of application is an activity recognition system for geriatric residences. In such residences it is often difficult to maintain a proper vigilance over the activities of seniors in order to prevent injuries due to accidents. It has been reported by Health Canada (see Appendix A for a detailed report) that with the growing aging population of Canada one in every four citizens will be a senior by the year 2041. In the late 1990s falls were responsible for 65% of injuries, 84% of injury-related hospital admissions, and 58% of injury-related deaths among the seniors population. The total health care costs due to seniors' falls are estimated at $1 billion annually. Unfortunately, often the kind of injuries sustained by seniors makes it difficult for them to seek immediate help themselves. In such a situation is will be highly beneficial for a vision system to be present, which recognizes and monitors the activities performed by the seniors at all times. Such a system can alert the appropriate authorities in case of a mishap. Another application is the automated monitoring of people and their activities in busy shopping mall areas for crime prevention.

5.9 Summary

Research in computer vision constantly strives to come to par with its human vision counterpart. The endeavor to develop a universal method for human activity recognition continues to challenge researchers. In this chapter, we have presented a novel, non-model silhouette directionality based algorithm for human activity recognition assuming limited occlusion. The algorithm captures both the static and dynamic (transitional) characteristics of human activity, unlike most contemporary works that deal with template matching of static pre-stored activity poses. Our approach is efficient in terms of storage, since each activity is stored and indexed as an eight dimensional vector. In addition, the computational load caused by computing motion for each body part or template matching is avoided, as we deal only with the silhouette contour. The algorithm can handle changes in view angle, scale, background and clothing and is translation independent. It can also deal with limited occlusion of the subject. Experimental results show promising recognition rates, with rare misclassifications arising mainly due to poor foreground-background separation. The CRRs obtained in our experiments range from 85% to 99%, for eight activities when viewed without temporal smoothing. The CRR increases with frame rate and 100% recognition is achieved when temporal smoothing is applied. The ease of implementation is an indication of the potential of the algorithm. The proposed algorithm can be used to maintain tracks of multiple identities and to recognize activities of individuals in geriatric or special care homes.

Chapter 6

Robust Feature Based Tracking

A comprehensive review of object tracking algorithms was presented in Chapter 3. It was mentioned that the KLT algorithm is a widely used tracking algorithm. However, the KLT tracking algorithm is not robust to noise. In this chapter we propose a weighting function based approach to enhance the noise immunity of the KLT tracker. In Section 6.1, we present an analysis of integrating weighting functions in the affine matching model for feature tracking. In Section 6.2 we describe the experimental setup used to evaluate the performance of the weighting functions and the results obtained are presented in Section 6.3.

Weighting functions have been discussed in the past in the context of estimation. They have been used to emphasize regions in space that are more reliable for analysis than others. In relation to the KLT algorithm, any weighting function that reduces the dissimilarity metric (Eq. 3.1) will provide improved performance. In the following section, we explore two such weighting functions. We also present a comprehensive analysis of the relative error functions to establish conditions of optimality.

6.1 Gaussian and LoG weighting functions

Digital images, now used frequently for data acquisition, are susceptible to a variety of noises. Noise can be introduced in several ways, such as poor film grain, poor CCD exposure and lossy electronic transmission. The averaging weighting function used by the KLT algorithm, described in Section 3.4.3, fails for image sequences that have been corrupted by noise. However, it is possible to improve tracking performance by assigning different weights to pixels depending on their location within the tracking window. Until now, the Gaussian and LoG weighting functions have been used for a plethora of applications such as image enhancement, edge detection, image filtering and memory-based learning. We propose the use of these weighting functions to improve the performance of tracking algorithms in noisy environments. The 2D Gaussian and LoG weighting functions can be expressed as:

$$w_{Gauss}(x, y) = \begin{cases} \dfrac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2+y^2}{2\sigma^2}} & (x, y) \in FW \\ 0 & elsewhere \end{cases} \tag{6.1}$$

$$w_{LoG}(x, y) = \begin{cases} \dfrac{1}{\pi\sigma^4}\left[1 - \dfrac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} & (x, y) \in FW \\ 0 & elsewhere \end{cases} \tag{6.2}$$

We discussed in Chapter 3, that good features are those that can be easily tracked. Generally these features are corners or edges of the object in motion. It is therefore important that we analyze the edge characteristics in an image. Synthetically generated objects or sharp edged objects generate very sharp transitions in the edge characteristics. On the other hand, real objects or objects in motion generate blurred transitions in the edge characteristics. We represent the edge characteristics of objects in an image by edge signals. Although, the edges in an image are two-dimensional signals we represent them as separable one-dimensional edge signals. This representation reduces the complexity of the analysis. A sharp edge is represented by a step function, $u(x)$. As shown in Figure 6.1(a), $u(x)$ represents a sharp edge centered in the feature window (FW). The mathematical representation of $u(x)$ is as follows:

$$u(x) = \begin{cases} 1 & x \geq t \\ 0 & x < t \end{cases} \tag{6.3}$$

A blurred edge $v(x)$ is represented by a concatenation of an exponential function followed by a step function. As shown in Figure 6.1(b), $v(x)$ represents a blurred edge centered in the FW. The mathematical representation of $v(x)$ is as follows:

$$v(x) = \begin{cases} \exp\left(-\frac{(x-t)^2}{2\sigma_s^2}\right) & x \leq t \\ 1 & x > t \end{cases} \tag{6.4}$$

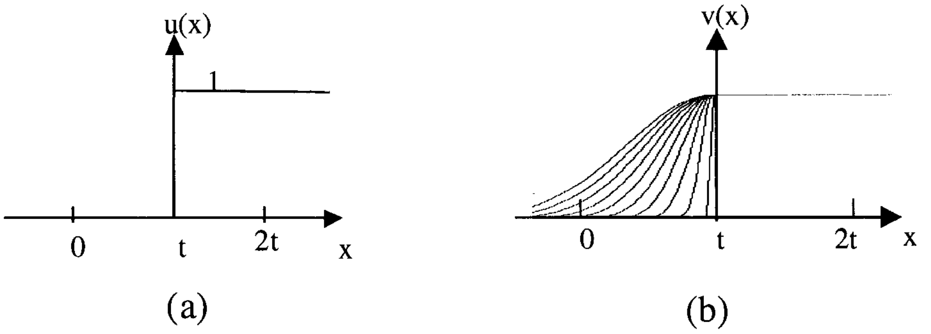


Figure 6.1: 1D Edge functions. (a) Step function, (b) Blur function

The parameter σ_s of the exponential function for the blurred edge $v(x)$, will determine the extent of blurring of the edge. Analysis of other blur edge models, such as the cumulative distribution function (CDF) of a normal distribution, increases the complexity of the problem and has been omitted from this discussion. We also assume that the edges are centered in the FW. This assumption holds since the KLT tracking algorithm is implemented by traversing all image pixels sequentially, hence every edge in the image will be centered in the *FW* at least once.

The ‘dissimilarity’ metric (E) of the KLT algorithm was discussed in Section 3.4.3. The dissimilarity is computed as the sum of squared differences (SSD), and is calculated using the following equation:

Minimize: $E = \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[A(x+\Delta x, y+\Delta y, t+\tau) \quad - B(x,y,t)]^2 w(x,y)$ where, the standard KLT uses $w(x,y) = \begin{cases} 1 & (x,y) \in FW \\ 0 & elsewhere \end{cases}$. We use the dissimilarity metric as our evaluation criteria. Since the dissimilarity metric calculates the error in matching the FWs, we term it as the error function. If $f(x,y)$ is the square differential image and $t = sizeof(FW)/2$, the error function computed by using weighting function ‘w’ with an edge characterized by ‘e’ can be written as:

$$E_e^w = \int_0^{2t}\int_0^{2t} f(x,y)w(x,y)dxdy \tag{6.5}$$

Eq. 6.5 is the continuous space version of the discrete space KLT Eq. 3.1. Since the dissimilarity or error $'E_e^w'$ determines if a feature is worth tracking, reduction in this error directly affects the tracking performance. The error function (Eq. 6.5) and the weighting function (Eqs. 6.1 and 6.2) can be represented in 1D as:

$$E_e^w = \int_0^{2t} f(x)w(x)dx \tag{6.6}$$

$$w_{Gauss}(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{6.7}$$

$$w_{LoG}(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\left[1-\frac{x^2}{\sigma^2}\right]\exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{6.8}$$

We now present two propositions. The first proposition is an analysis of the error functions computed when tracking a sharp edge with the Gaussian and LoG weighting functions. Note that $t = sizeof(FW)/2$.

Proposition 1: $E_u^g > E_u^l$ if $t < (3.35\sigma)$ (6.9)

Implications: The above proposition implies that while tracking an object with a sharp edge, the error obtained with a LoG weighting function, E_u^l, will be less than the error obtained when using a Gaussian weighting function while $t < (3.35\sigma)$. We now present a proof of Proposition 1.

Proof:

The right-hand-side of Eq. 6.9 can be written as follows:

$$R.H.S = \int_t^{2t}\left\{\frac{1}{\sqrt{2\pi\sigma^2}}\left[1-\frac{x^2}{\sigma^2}\right]\exp\left(-\frac{x^2}{2\sigma^2}\right)\right\}dx$$

The left-hand-side of Eq. 6.9 can be written as follows:

$$L.H.S = \int^{2t}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x^2}{2\sigma^2}\right)dx$$

$L.H.S > R.H.S$ if $L.H.S - R.H.S > 0$

Solving the integrals using the mathematical tool 'Mathematica5.0' results in the following relation for $L.H.S - R.H.S$:

$$L.H.S.-R.H.S = \frac{1}{2\sqrt{2\pi}\sigma}\left(2e^{-\frac{2t^2}{\sigma^2}}\left(2-e^{\frac{3t^2}{2\sigma^2}}\right)t+\sqrt{2\pi}\sigma Erf\left(\frac{t}{\sqrt{2}\sigma}\right)-\sqrt{2\pi}\sigma Erf\left(\frac{\sqrt{2}t}{\sigma}\right)\right) \quad (6.10)$$

A graphical analysis of Eq. 6.10 is performed with varying values of t and σ, to determine the relation between t and σ for which Eq. 6.9 holds true. The mesh plot of the computation is shown in Figure 6.2.

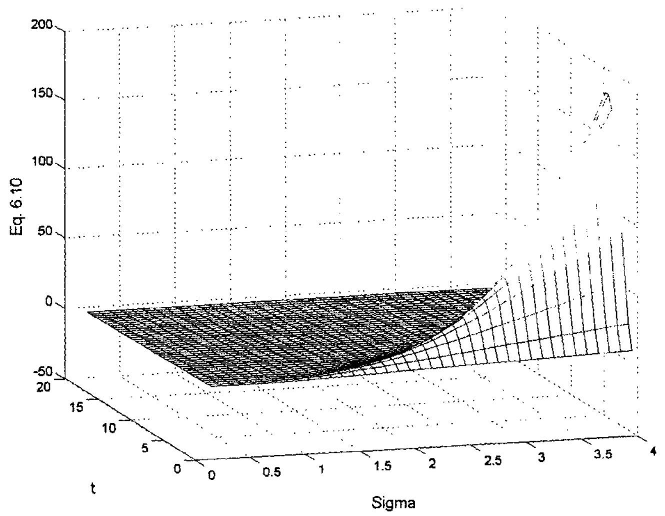


Figure 6.2: Mesh plot of Eq. 6.10 for various values of *t* and σ (sigma).

In order to evaluate a deterministic relation, we modify the mesh plot (Figure 6.2) to reflect values greater than zero as 1, and values less than zero as –1. The modified mesh plot (Figure 6.3) clearly demarcates the linear relation between t and σ for which Eq. 6.10 holds true. A geometrical analysis determines the slope of the line to be 3.35, thus for $t < (3.35\sigma)$ Eq. 6.9 holds true.

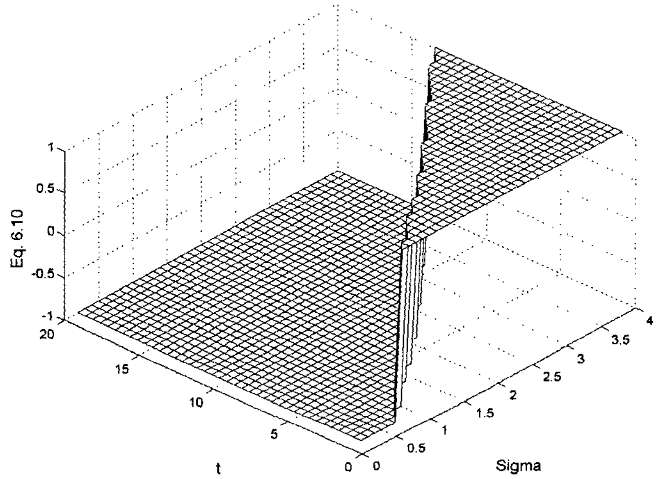


Figure 6.3: Mesh plot reflecting modifications made to Figure 6.2.

The second proposition is an analysis of the error functions generated when tracking a blurred edge with the Gaussian and LoG weighting functions.

Proposition 2: $E_v^g < E_v^l$ if the edge is sufficiently blurred (6.11)

Implications: The above proposition implies that when tracking an object with a blurred edge, a Gaussian weighting function will provide better performance than a LoG weighting function. The blurring of the edge is determined by the value of σ_s. For small values of σ_s, the function for a blurred edge *v(x),* resembles a unit step function. Thus we also prove that Proposition 1 can be derived directly as a limiting case of Proposition 2. We now present a proof of Proposition 2.

Proof:

$$L.H.S = E_v^g = \int_0^t \left\{ \exp\left(-\frac{(x-t)^2}{2\sigma_s^2} \right) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2} \right) \right\} dx + \int_t^{2t} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2} \right) dx$$

$$R.H.S = E_v^l = \int_0^t \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \left[1 - \frac{x^2}{\sigma^2} \right] \exp\left(-\frac{x^2}{2\sigma^2} \right) \exp\left(-\frac{(x-t)^2}{2\sigma_s^2} \right) \right\} dx$$
$$+ \int_t^{2t} \frac{1}{\sqrt{2\pi\sigma^2}} \left[1 - \frac{x^2}{\sigma^2} \right] \exp\left(-\frac{x^2}{2\sigma^2} \right) dx$$

$R.H.S > L.H.S$, if $L.H.S - R.H.S < 0$

The above definite integrals can be solved to obtain the following equation for $L.H.S - R.H.S$, where σ_s = s:

$$L.H.S - R.H.S = A + B \tag{6.12}$$

where,

$$A = -\frac{e^{-\frac{2t^2}{\sigma^2}}\left(-2 + e^{\frac{3t^2}{2\sigma^2}}\right)\sqrt{\frac{2}{\pi}}t - \sigma Erf\left(\frac{t}{\sqrt{2}\sigma}\right) + \sigma Erf\left(\frac{\sqrt{2}t}{\sigma}\right)}{2\sigma}$$

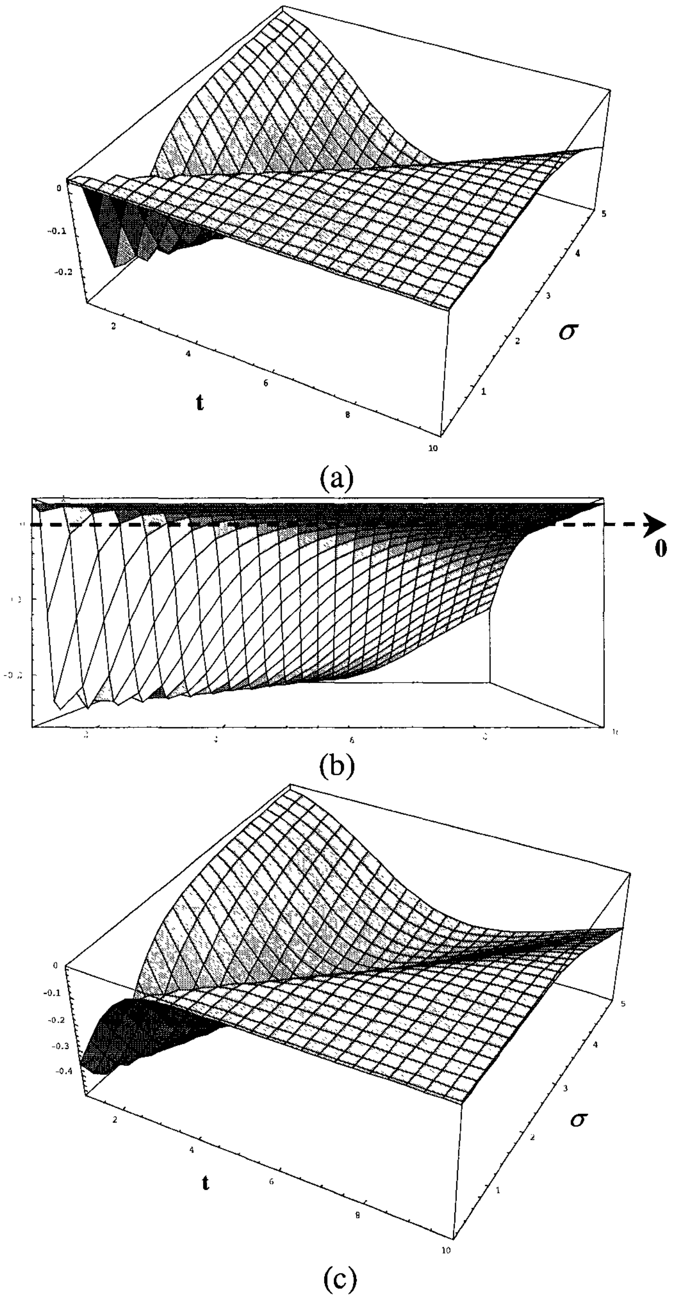
$$B = -\frac{1}{2\sqrt{2\pi}\sigma\left(s^2+\sigma^2\right)^{5/2}}\left(s\left(C+D\right)\right)$$

Expressions for C and D are as follows:

$$C = e^{\frac{t^2\left(s^2+\sigma^2\right)}{2s^2\sigma^2}}\left(-2e^{\frac{t^2}{2s^2}}st\sqrt{s^2+\sigma^2}\left(s^2+2\sigma^2\right)+e^{\frac{t^2\left(1+\frac{\sigma^4}{s^2\left(s^2+\sigma^2\right)}\right)}{2\sigma^2}}\sqrt{2\pi}\sigma\left(s^4+s^2\sigma^2+t^2\sigma^2\right)Erf\left(\frac{st}{\sqrt{2}\sigma\sqrt{s^2+\sigma^2}}\right)\right)$$

$$D = e^{\frac{t^2}{2s^2}}\sigma\left(2st\sigma\sqrt{s^2+\sigma^2}+e^{\frac{t^2\sigma^2}{2s^2\left(s^2+\sigma^2\right)}}\sqrt{2\pi}\left(s^4+s^2\sigma^2+t^2\sigma^2\right)Erf\left(\frac{t\sigma}{\sqrt{2}s\sqrt{s^2+\sigma^2}}\right)\right)$$

When 's=0', Eq. 6.12 reduces to Eq. 6.10 as the edge function resembles a sharp edge. For small values of edge variance 's', graphical plots of Eq. 6.12 (see Figure 6.4) show positive values for $E_v^g - E_v^l$. This implies that error with a Gaussian weighting function is higher than error with LoG weighting function, which validates Proposition 1. As the edge variance increases (*i.e.* the edge becomes blurred) graphical plots of Eq. 6.12 show increasingly negative values for $E_v^g - E_v^l$. This implies that with a blurred edge, the LoG has higher error values than the Gaussian weighting function.



(c)

Figure 6.4: Mesh plot Eq. 6.12 for various values of t and σ. (a) Eq. 6.12 , $\sigma_s = 0.01$, (b) Lateral view of (a) highlighting values higher than 0, (c) Eq. 6.12 , $\sigma_s = 1.0$. Note the negative values corresponding to low values of t and σ.

These propositions can be easily extended to a two-dimensional case as the weighting function and the edge characteristics have been considered to be separable. The weighting functions in two-dimensions are illustrated in Figure 6.5.

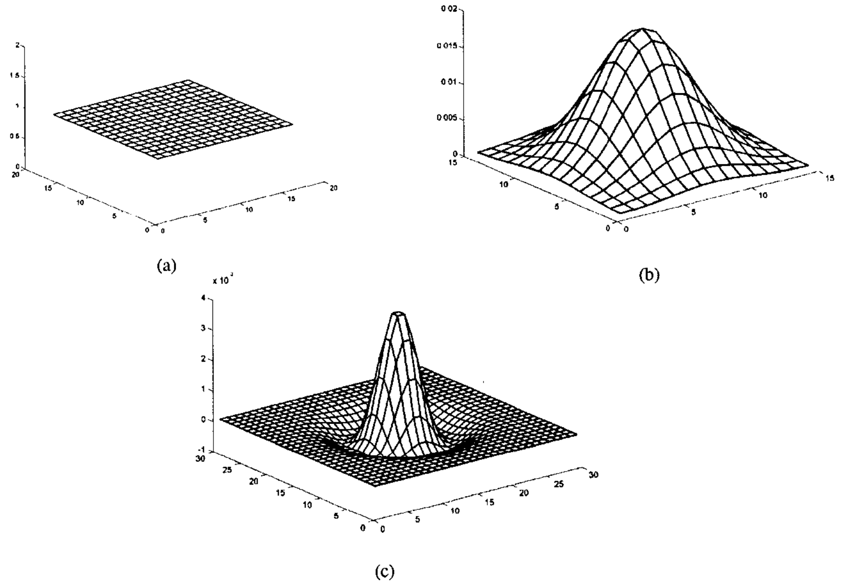


Figure 6.5: Pictorial representation of 2D weighting function. (a) Averaging function, (b) Gaussian function sigma=3), (c) LoG function (sigma=3).

6.2 Experimental Setup

The proposed weighting functions are applied to two sets of sequences: sequences with no noise reduction applied prior to tracking and sequences with noise reduction applied prior to tracking. In our experiments, we use a search range of 12 pixels, and (when not mentioned specifically) a FW of size 5 x 5 (*i.e.*, t = 2.5) is used. In order to validate Proposition 1 and 2 (Eq. 6.9 and Eq. 6.11 respectively), we also evaluate the proposed weighting functions with varying FW size and σ. The size of the tracking window limits the true FW size; hence, for an accurate comparison the FW size and tracking window size are considered to be equivalent. To simulate a real noisy test environment we add controlled amount of salt and pepper noise with a uniform probability density function to the test images. We also assume that an addition of ‘n%’ noise implies adding random noise to n% pixels of the original image. We use four test sequences — Synthetic square sequence, Synthetic car sequence, Real car sequence and a Hybrid workbench sequence.

6.2.1 Synthetic Square Sequence

The synthetic test images of size 600 x 600 (Figure 6.6) consist of a simple bright square of size 240 x 240 on a dark background. The bright square is displaced by 12 pixels in both the horizontal and vertical directions in the next frame.

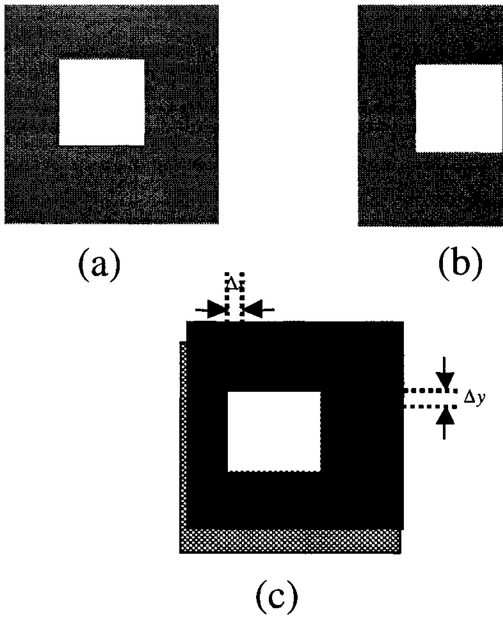


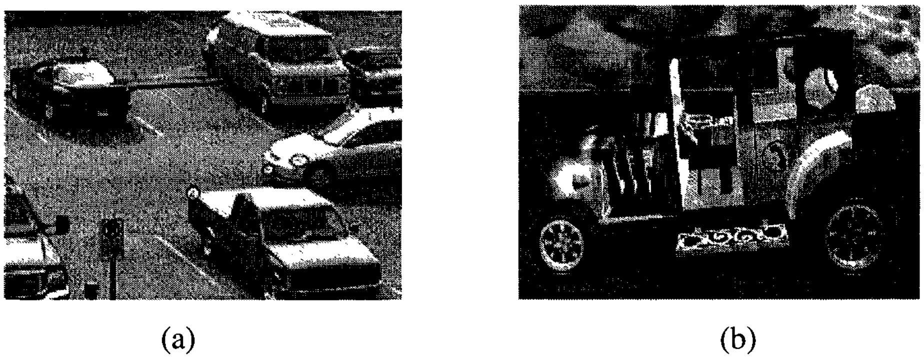
Figure 6.6: Generation of synthetic square sequence. Simulated Noiseless Video Frames (a) First frame, (b) Second Frame and (c) The objects overlapped (pictorial representation of the displacement of the object with respect to the first frame).

6.2.2 Real Car Sequence

The real car sequence (Figure 6.7(a)) was captured at the University of Alberta and depicts a vehicle entering into a parking lot. The presence of multiple vehicles in the background provides a rigorous test of the weighting functions and the tracking algorithm.

6.2.3 Synthetic Car Sequence

Using a screen shot from a popular Internet car racing game, 'Buzzing Cars', we generated an image sequence that consists of a synthetically created car. The car object (Figure 6.7(b)) in this sequence has sharper edge definitions than our real car image sequence.



(a) (b)

Figure 6.7: Car sequences. (a) Real car frames and (b) Synthetic car frames.

6.2.4 Hybrid Workbench Sequence

A hybrid test sequence (Figure 6.8) was captured in the Multimedia Computing and Communications Laboratory. The camera was moved to simulate motion, which leads to blurring of the edges of the objects in the image. The objects of interest (OOI) are the chair and the computer monitor. While the features of the chair are left intact (*i.e.*, blurred), the computer monitor is manually enhanced to have sharper features. Thus, this test sequence gives us an opportunity to test Propositions 1 and 2 simultaneously.

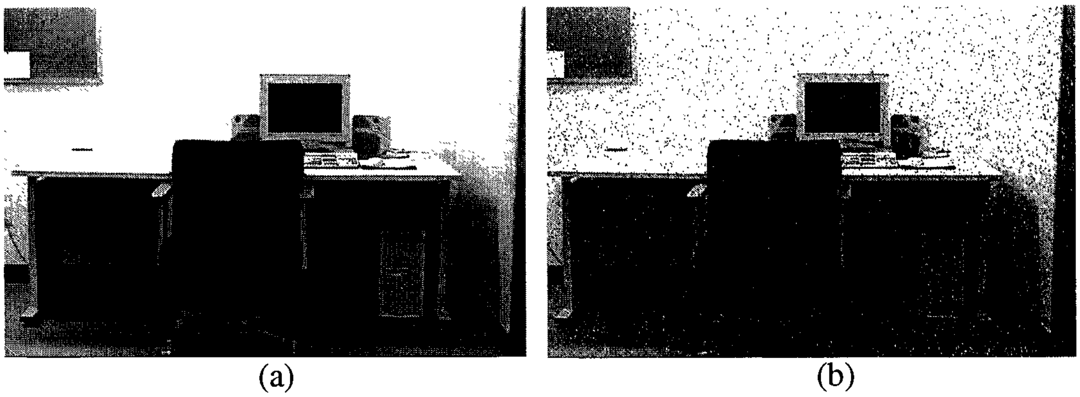


Figure 6.8: Hybrid video frames. (a) Noiseless test image of the workbench. (b) Noisy image with 5% noise.

6.2.5 Pre-Noise removal

We also performed experiments in which noise removal was applied prior to tracking. To generate the test sequences for noise reduction, we experimented with three standard algorithms: average filtering, Wiener adaptive filtering and median filtering. Figure 6.9 shows the results obtained with noise removal algorithms applied to the synthetic square sequence. It is observed that a 3x3 median filter resulted in maximum noise removal and hence we used median filtering for the pre-noise removal evaluation of the weighting functions.

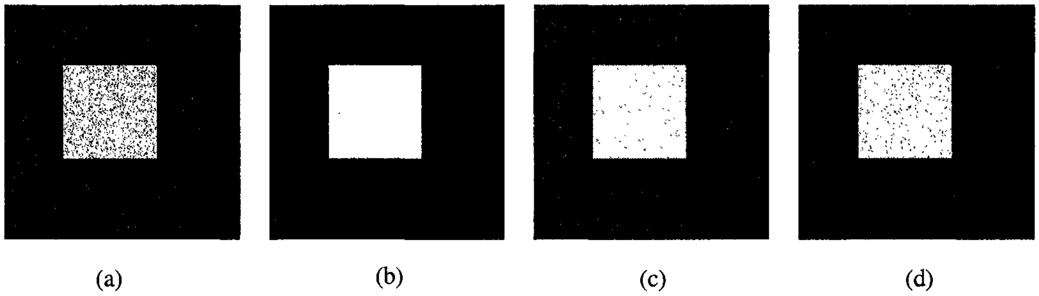


Figure 6.9: Synthetic square sequence pre noise removal. (a) Simulated 20% noisy frame; (b) Median filtering noise reduction applied to (a); (c) Average filtering noise reduction applied to (a); (d) Wiener adaptive filtering noise reduction applied to (a).

6.3 Performance Evaluation

We base our evaluation on two criteria — the Euclidean distance between actual and tracked features (similar to our theoretical analysis) and the tracking efficiency. The Euclidean error is computed and averaged over features that have been correctly tracked and the remaining features are ignored. When a feature cannot be tracked successfully we assign a global maximum error for that test case. In cases where the correspondence between the feature points is not known, the ratio of the features that are correctly tracked in the successive frame to the features that are initialized in the first frame is used as an evaluation criterion. We define the tracking efficiency (η) of the weighting functions as:

$$\eta = \frac{Number \;\; of \;\; successfully \;\; tracked \;\; features}{Total \;\; number \;\; of \;\; features \;\; initialized} \times 100\ \%$$

Figure 6.10 shows the performance of the weighted tracking algorithm for the synthetic car and hybrid workbench sequences. It is observed that for all test sequences the averaging KLT algorithm fails to track the object even at low levels of noise, while the weighted tracking algorithm continues to track until 25 % noise is added (Figure 6.10(a)). We find that in the hybrid sequence, for blurred edges, the Gaussian weighting function has lower Euclidean error than the LoG weighting function (Figure 6.10(b)). For sharper edges the LoG weighting function has lower Euclidean error (Figure 6.10(c)), which experimentally validates Propositions 1 and 2 (Eq. 6.9 and Eq. 6.11).

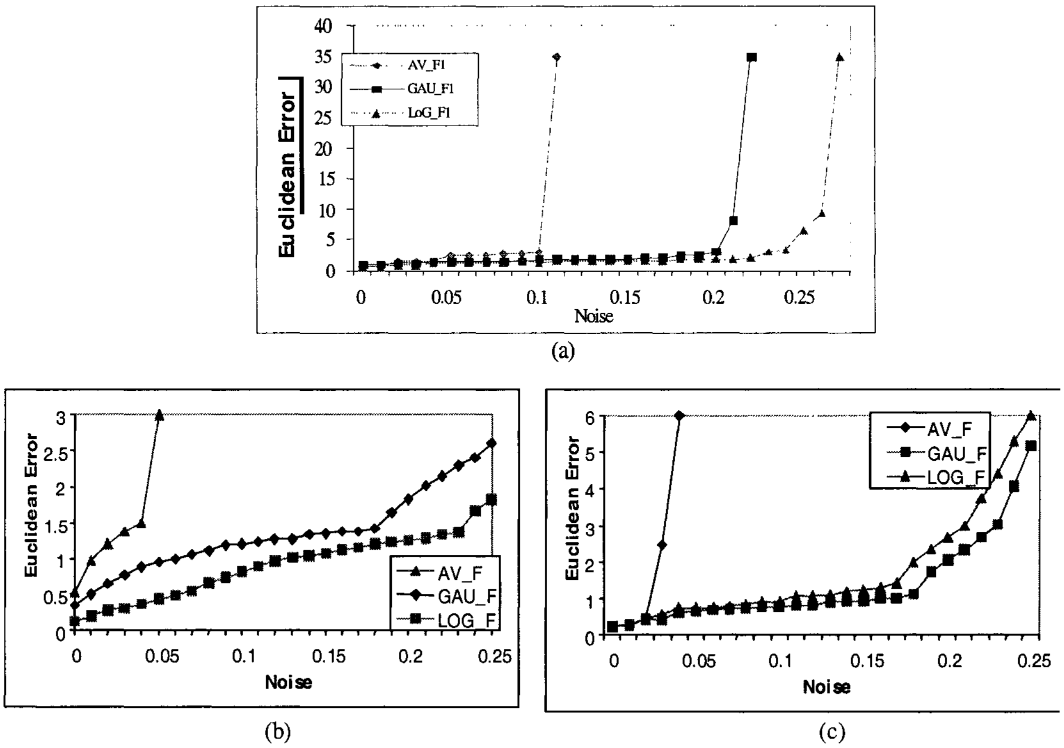
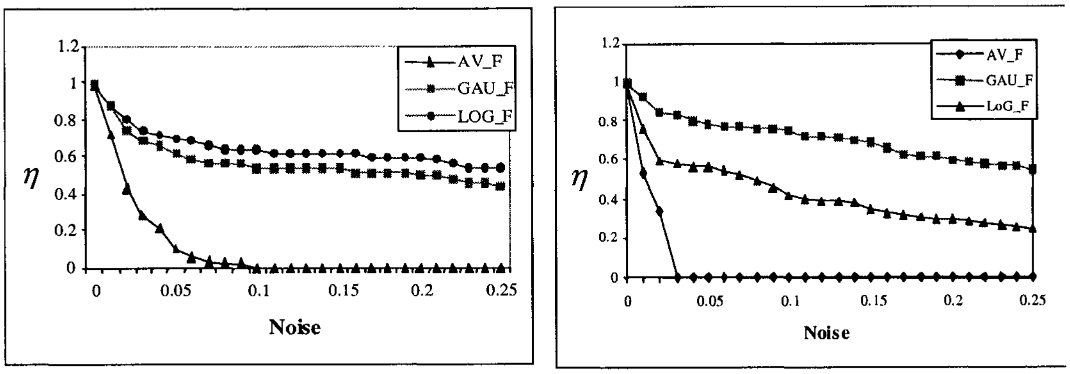


Figure 6.10: Performance of weighting functions for (a) synthetic square sequence (4 features), (b) hybrid workbench sequence, tracking the sharp corners of the computer monitor object (4 features) and (c) hybrid workbench sequence, tracking the chair object (17 features).

Figure 6.11 shows the performance evaluation of the weighted KLT tracker with synthetic and real car sequences. It is observed that LoG weighting function has higher tracking efficiency than the Gaussian and averaging weighting function (Figure 6.11 (a)) when tracking the synthetic car. This observation is in accordance with Proposition 1 (Eq. 6.9). Tracking efficiency of Gaussian and LoG weighting function for the real car sequence is shown in Figure (b). It is observed that for the real car sequence with blurred edges, the Gaussian weighting functions perform better than the LoG weighting functions. This validates Proposition 2 (Eq. 6.11).



(a) (b)

Figure 6.11: Performance of weighting functions with car sequences. (a) Synthetic car sequence, 50 features have been initialized and are being tracked. And (b) Real image car sequence, 50 features have been initialized and are being tracked.

Figure 6.12 shows the performance evaluation with pre-noise removed test sequence. When the image sequences are preprocessed to remove noise, the performance plots translate along the noise axis. The weighting functions demonstrate performance trends (at higher noise levels) to be similar to no prior noise removal at low noise levels, as shown in Figure 6.12.

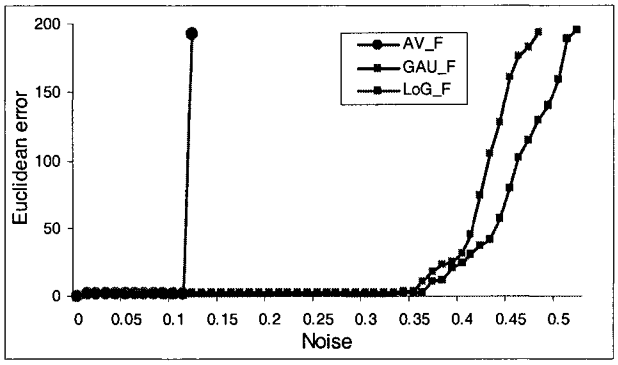


Figure 6.12: Performance of weighting function with synthetic square sequence when noise reduction is applied prior to tracking.

Experiments have also been performed to verify the relations between 't' and 'σ' which were proposed in Section 6.1 (Eq. 6.9). Tabulated results (as shown in Table 6.1) concur with the relations developed between the weighting function standard deviation 'σ' and feature window size '$2t$' for both the proposed weighting functions. Based on the experiments performed, we observe that when the image has sharp boundaries and edges, the LoG weighting function, which has a sharper parametric curve, performs better than the Gaussian. In real images, the edges are smoothed because of factors such as environmental conditions, camera resolution and pixel discretization. For such images the Gaussian weighting function performs better than the LoG weighting function.

Table 6.1: Additional experimental results validating Propositions 1.

Edge Characteristic	t	σ of Weighting Function	Percentage of Noise	E_e^g	E_e^l	Condition	Result
Sharp	2.5	1.0	15%	1.6638	2.2744	$t < 3.35\sigma$	$E_e^g < E_e^l$
Sharp	2.5	0.5	15%	1.9334	1.6078	$t > 3.35\sigma$	$E_e^g > E_e^l$
Sharp	2.5	1.0	15%	1.6638	2.2744	$t < 3.35\sigma$	$E_e^g < E_e^l$
Sharp	3.5	1.0	15%	2.9051	2.7577	$t > 3.35\sigma$	$E_e^g > E_e^l$
Sharp	4.5	1.0	15%	5.1819	4.6884	$t > 3.35\sigma$	$E_e^g > E_e^l$
Sharp	5.5	1.0	15%	5.6862	4.7515	$t > 3.35\sigma$	$E_e^g > E_e^l$

6.4 Summary

In this chapter, we proposed two weighting functions to improve the performance of the KLT tracking algorithm. We presented the mathematical analysis of our propositions. We have established that for typical test sequences, both the proposed weighting functions (Gaussian and LoG weighting function) consistently perform better tracking than the KLT average weighting function. This is especially true in a noisy environment. Additionally, we also established minimum limits to parameters and conditions under which one of the weighting functions has better tracking performance than the other. Real world tracking sequences are expected to be noisy, and therefore are ideal for application of these weighting functions. The LoG tends to give a sharper emphasis on the central FW pixels and is suitable for images with well-defined and sharp image corners like the synthetic test images. Interestingly, for real images where the edges may not be as sharply defined, the Gaussian weighting function performs better.

Chapter 7

Conclusions and Future Work

Human activity recognition (HAR) is one of the leading areas of research in computer vision. A plethora of applications would benefit from continuing research in this area. Past works in this area are limited by the complexity of their approaches and the lack of a universally applicable algorithm. In addition, often the presence of noise degrades the recognition efficiency of the algorithms. This thesis addressed two issues. We proposed a novel algorithm based on silhouette directionality for HAR. The algorithm extracts contour directionality based feature vectors from the frames of a video sequence and clusters them in vector space based on an angular distance parameter. The proposed algorithm was evaluated on a wide range of indoor and outdoor video sequences, and performance comparisons with related non-model based, non-intrusive work showed promising high recognition rates.

We also investigated the effects of incorporating weighting functions in tracking algorithms. We proposed the use of the Gaussian and LoG weighting functions and determined conditions for optimal performance of the two weighting functions. It was validated experimentally with noisy image sequences that the Gaussian weighting function provides better tracking performance of an object with a blurred edge. The LoG weighting functions on the other hand provides a better tracking performance of objects with sharp edges, such as computer generated image sequences.

Although the proposed algorithm for human activity recognition shows promising results, there are still several issues that need to be addressed. A few potential future works are presented below.

1. Robust foreground-background subtraction techniques. Although the current application of the foreground-background separation in our work is not limited to plain uni-colored backgrounds, it has not been extended to more dynamic backgrounds with larger amount of clutter. Techniques for a more robust foreground and background separation are still an important issue for future work.
2. Activity recognition for multiple subjects. We have assumed in our work, that only one subject is present in the field of view at a time. This assumption limits the scope of application of our work. An identified area of work is to incorporate extraction of feature vectors for multiple individuals. The incorporation of multiple subjects would also lead to a new direction of research that would study the interaction between different subjects vis-à-vis single activities.
3. Training and testing the activity recognition algorithm for a larger set of activities. We also need to broaden the experiment set used for HAR to include more activities such as jumping, swinging and falling.
4. Incorporation of color. This would extend our HAR work to multiple attribute feature vectors vis-à-vis the current directionality based feature vectors.

5. Real time system implementation. The current implementation of the proposed technique has been done offline. We have however already begun work on a real time implementation of the same.

References

[1] J. K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review", *Computer Vision and Image Understanding*, pp 428-440,1999.

[2] D. M. Gavrila, "The Visual Analysis of Human Movement: A Survey", *Computer Vision and Image Understanding*, 73(1), pp 82-98, 1999.

[3] T. B. Molesund and E. Granum, "A survey of Computer Vision Based Human Motion Capture", *Computer Vision and Image Understanding*, pp 231-268, 2001.

[4] L. Wang, W. Hu and T. Tan, "Recent Developments in Human Motion Analysis", *Pattern Recognition*, pp 585-01, Vol. 36, 2003.

[5] T. C. C. Henry, E. G. R. Janapriya and L. C. deSilva, "An Automatic System for Multi Human Tracking and Actions Recognition in Office Environment", *In Proc. of ICASSP*, 2003.

[6] J. Krumm, S. Harris, B. Meyers, B. Brummit, M. Hale, S. Shafer, "Multi-Camera Multi-Person Tracking for Easy Living", *Proc of 3rd IEEE International Workshop on Visual Surveillance*, 2000.

[7] S. Dagtas, W. A. Khatib, A. Ghafoor, R. L. Kashyap, "Models for Motion-Based Video Indexing and Retrieval", *IEEE Trans. on Image Processing*, pp 88-101, vol 9(1), Jan 2000.

[8] G. Johansson, "Visual Motion Perception", *Scientific American*, pp 76-88, June 1975.

[9] K. Akita, "Image Sequence Analysis of Real World Human Motion", *Pattern Recognition*, 17(1), pp 73-83, 1984.

[10] I. Haritaoglu, D. Harwood and L. S. Davis, “W4: real-time surveillance of people and their activities”, *IEEE Trans. on PAMI*, Volume: 22, Issue: 8, pp 809 – 830, 2000.

[11] A. Bottino and A. Laurentini, “A Silhouette Based Technique for the Reconstruction of Human Movement”, *Computer Vision and Image Understanding*, 83, pp 79-95, 2001.

[12] C. R. Wren, A. Azarbayejani, T. Darrel and A. Pentland, “Pfinder: Real Time Tracking of the Human Body”, *IEEE Transactions on PAMI*, pp 780-785, 1997.

[13] R. D. Green and L. Guan, “Quantifying and Recognizing Human Movement Patterns from Monocular Video Images- Part I: A New Framework for Modeling Human Motion”, *IEEE Trans. on CSVT*, Vol. 14, No.2, pp 179-189, Feb 2004.

[14] N. Cuntoor, A. Kale and R. Chellappa, “Combining Multiple Evidences for Gait Recognition”, *In Proc. of ICASSP*, 2003.

[15] Y. A Ivanov and A. F. Bobick, “Recognition of Visual Activities and Interactions by Stochastic Parsing”, *IEEE Trans. on PAMI*, Vol.22, No.8, 2000.

[16] X. Sun, C. W. Chen and B. S. Manjunath, “Probabilistic Motion Parameter Models for Human Activity Recognition”, *In Proc. of ICPR*, 2002.

[17] J. Ben-Arie, Z. Wang, P. Pandit, S. Rajaram, “Human Activity Recognition Using Multidimensional Indexing”, *IEEE Trans. on PAMI*, Vol. 24, No. 8,2002.

[18] J. E. Boyd and J. J. Little, “Global versus Structured Interpretation of Motion: Moving Light Displays”, *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, 1997.

[19] R. Polana and R. Nelson, “Detection and Recognition of Periodic, Nonrigid Motion”, *Computer Vision*, vol. 23, pp. 261-282, 1997.

[20] A. F. Bobick and J. W. Davis, “The Recognition of Human Movement Using Temporal Templates”, *IEEE Trans. on PAMI*, 2001.

[21] A. Ali and J. K. Aggarwal, “Segmentation and Recognition of Continuous Human Activity”, *IEEE Workshop on Detection and Recognition of Events in Video*, 2001.

[22] H. Fujiyoshi, A. J. Lipton and T. Kanade, “Real Time Human Motion Analysis by Image Skeletonization”, *IEICE Trans. INF. & SYST.*, pp 113-120, Vol. E87-D, No.1, Jan 2004.

[23] C. Rao, A. Yilmaz and M. Shah, “View-Invariant Representation and Recognition of Actions”, *International Journal of Computer Vision*, 50(2), pp 203-226, 2002.

[24] J. D. Shutler, M. G. Grant, M. S. Nixon and J. N. Carter, “On a Large Sequence-Based Human Gait Database”, *Proc. of 4th International Conference on Recent Advances in Soft Computing*, pp 66-71, 2002.

[25] R. Gross and J. Shi, “The CMU Motion of Body (MoBo) Database, Technical Report CMU-RI-TR-01-18”, Robotics Institute, Carnegie Mellon University, 2001.

[26] R. Kohavi and F. Provost, “Glossary of Terms”, Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process, Vol. 30, pp 271-274, 1998.

[27] L. Wang, H. Ning, T. Tan and W. Hu, “Fusion of Static and Dynamic Body Biometrics for Gait Recognition”, *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 14, No.2, pp 149-158, Feb 2004.

[28] K. V. Mardia and P. Jupp. “Directional Statistics”, John Wiley and Sons Ltd., 2nd edition, 2000.

[29] J. Verestoy and D. Chetverikov, ‘Experimental Comparative Evaluation of Feature Point Tracking Algorithm’, Proc. *Workshop on Evaluation and Validation of Computer Vision Algorithms*, pp 183-194, 1999.

[30] J. Verestoy and D. Chetverikov, ‘Tracking Feature Points: A New Algorithm’, Proc. of *14th International Conf. on Pattern Recognition*, pp. 1436-1438, Australia, 1998.

[31] J. Shan, ‘Feature Extraction Based on Multiresolution Analysis’, extended abstract in *Comm III, International Society for Photogrammetry and Remote Sensing* (ISPRS), July 1998.

[32] V. Hwang, ‘Tracking Feature Points in Time Varying Images using an Opportunistic Selection Approach’, *Pattern Recognition*, pp 247-256,1989.

[33] V. Salari and I.K. Sethi, ‘Feature Point Correspondence in the Presence of Occlusion’, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, pp 87-91, Jan 1990.

[34] K. Rangarajan and M. Shah, ‘Establishing Motion Correspondence’, *CVGIP: Image Understanding*, pp 103-108,1991.

[35] I. K. Sethi and R. Jain, ‘Finding Trajectories of Feature Points in Monocular Image Sequence’, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 9, No. 1, pp 56-73, Jan 1987.

[36] J. Verestoy and D. Chetverikov, ‘Digital PIV: A Challenge for Feature Based Tracking’, *Proc. 23rd Workshop of the Austrian Pattern Recognition Group*, pp 165-174,1999.

[37]B.D. Lucas and T. Kanade, 'An Iterative Image Registration Technique with an Application in Stereo Vision', Proc of *7th International Joint Conference on Artificial Intelligence*, pp 674-679, Vancouver, Canada, 1981.

[38] J. Shi and C. Tomasi, 'Good Features to Track', Proc. *IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR94), pp 593-600, Seattle, WA, USA, June 1994.

[39] J. J. Gonzalez, I. S. Lim, P. Fua and D. Thalmann, 'Robust Tracking and Segmentation of Human Motion in an Image Sequence', *IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), Vol. 3, Hong Kong, April 2003.

[40] C. J. Veenman, M.J.T. Reindeer and E. Backer, 'Resolving Motion Correspondence for Densely Moving Points', *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp 54-72, Vol. 23, Jan 2001.

[41] D. Comaniciu, V. Ramesh and P. Meer, 'Real-time Tracking of Non-rigid Objects Using Mean Shift', *Proc. of Conference on Computer Vision and Pattern Recognition,* pp. 42-149, June 2000.

[42] K. Shafique and M. Shah, 'A Non-Iterative Greedy Algorithm for Multi-frame Point Correspondence', *Proc. of 9th International Conference on Computer Vision*, 2003.

[43] N. F. Troje, 'Decomposing biological motion: A framework for analysis and synthesis of human gait patterns', *Journal of Vision*, pp. 371-387, Vol. 2, Sept 2002.

[44] A. F. Bobick and A. Johnson, 'Gait recognition using static activity-specific parameters', *Proceedings of IEEE Computer Vision and Pattern Recognition*, Kauai, Hawaii, Dec 2001.

[45] J. Rehg, T. Kanade, 'Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking', *Proceedings of the 3rd European Conference on Computer Vision,* pp. 35-46, Vol. 2, May 1994.

[46] D. Cunado, M. S. Nixon and J. N. Carter, 'Using Gait as a biometric, via Phase-Weighted Magnitude Spectra,' *Proceedings of 1st International Conference on Audio and Video Based Biometric Person Authentication,* pp 95-102, 1997.

[47] Z. Chen and H. J. Lee, 'Knowledge-Guided Visual Perception of 3D Human Gait from a Single Image Sequence', *IEEE Transactions on Systems, Man and Cybernet,* pp 336-342, 22(2), 1992.

[48] Yang, Y.H. and M.D. Levine, "The background primal sketch: An approach for tracking moving objects," *Machine Vision and Applications*, pp. 17-34, Vol. 5, 1992.

[49] C. Stauffer and W. Grimson, 'Adaptive background mixture models for real-time tracking', *Proceedings of Conference on Computer Vision and Pattern Recognition,* pp. 246-252, Vol. 2, 1999.

[50] R. T. Collins *et al.*, 'A system for video surveillance and monitoring: VSAM final report', CMU-RI-TR-00-12, Technical Report, Carnegie Mellon University, 2000.

[51] A. Verri, S. Uras and E. DeMicheli, 'Motion segmentation from optic flow', *Proceedings of the Fifth Alvey Vision Conference,* pp. 209-214, 1989.

[52] A. Meygret and M. Thonnat, 'Segmentation of optical flow and 3D data for the interpretation of mobile objects', *Proceedings of the International Conference on Computer Vision*, Dec 1990.

[53] J. Barron, D. Fleet and S. Beauchemin, 'Performance of optical flow techniques', *International Journal of Computer Vision*, pp. 42-77, Vol. 12(1), 1994.

[54] R. Cutler and L. S. Davis, 'Robust real-time periodic motion detection, analysis and applications', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 781-796, Vol. 22(8), 2000.

[55] A. J. Lipton, 'Local application of optic flow to analyse rigid versus non-rigid motion', In the website http://www.eecs.lehigh.edu/FRAME/Lipton/iccvframe.html.

[56] A. Selinger and L. Wixson, 'Classifying objects as rigid or non-rigid without correspondences', *Proceedings of the DARPA Image Understanding Workshop*, pp. 341-358, Vol. 1, 1998.

[57] G. Welch and G. Bishop, 'An Introduction to the Kalman Filter', from http://www.cs.unc.edu, UNC-Chapel Hill, TR95-041, Nov 2000.

[58] H. Sidenbladh, M. J. Blake and D. J. Fleet, 'Stochastic tracking of 3D human figures using 2D image motion', *Proceedings of the European Conference on Computer Vision*, 2000.

[59] V. Pavolic, J. M Rehg, T. J. Cham and K. P. Murphy, 'A Dynamic Bayesian network approach to figure tracking using learned dynamic models', *Proceedings of the International Conference on Computer Vision*, pp. 94-101, 1999.

[60] M. Isard and A. Blake, 'Condensation- conditional density propagation for visual tracking', *International Journal of Computer Vision*, pp. 5-28, Vol. 29(1), 1998.

[61] Y. Guo, G. Xu and S. Tsuji, 'Understanding Human motion patterns', *Proceedings of the International Conference on Pattern Recognition*, pp. 325-329, 1994.

[62] L. Lee and W.E.L. Grimson, 'Gait Analysis for Recognition and Classification', *Proceedings of 5th International Conference on Automatic Face and Gesture Recognition*, 2002.

[63] G. Shakhnarovich, L. Lee and T. Darrel, 'Integrated Face and Gait Recognition From Multiple Views', *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2001.

[64] S. S. Intille, J. W. Davis and A. F. Bobick, 'Real-Time Closed World Tracking', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 697-703, June 1997.

[65] U. Franke, D. M. Gavrila, S. Gorzig, F. Lindner, F. Paetzhold and C. Wohler, 'Autonomous Driving Approaches Downtown', *IEEE Intelligent Systems,* pp. 40-48, Vol. 13, 1998.

[66] D. Gavrila and L. Davis, 3-D Model Based Tracking of Humans in Action: a Multiview Approach', *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,* pp. 73-80, 1996.

[67] C. Myers, L. Rabinier and A. Rosenberg, 'Performance tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition', *IEEE Transactions on ASSP,* pp. 623-635.

[68] A. Pentland, 'Smart Rooms', *Scientific American,* pp 54-62, 274(4), 1996.

[69] R. Polana and R. Nelson, 'Low Level Recognition of Human Motion', *Proceedings of IEEE Workshop on Motion of Non-Rigid and Articulated Objects,* 1994.

[70] M. Turk, 'Visual Interaction with Life-like Characters', *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition,* pp. 368-373, 1996.

[71] M. Leung and Y. Yang, 'First Sight: A Human Body Outline Labelling System', *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 359-377, Vol. 17(4), 1995.

[72] I. Haritaoglu, D. Harwood and L. Davis, 'W4S: A Real Time System for Detecting and Tracking People in 2.5D', *Fifth European Conference on Computer Vision,* June 1998.

[73] D. Hogg, 'Model-Based Vision: A Paradigm to See a Walking Person', *Image and Vision Computing,* pp. 5-20, Vol. 1(1), 1983.

[74] S. Niyogi and E. Nelson, 'Analyzing gait with spatio-temporal surfaces', *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,* pp 750-753, 1994.

[75] R. T. Collins, R. Gross and J. Shi, 'Silhouette Based Human Identification from Body Shape and Gait', *Fifth International Conference on Automatic Face and Gesture Recognition,* May 2002.

[76] A. Galata, N. Johnson and D. Hogg, 'Learning Variable-Length Markov Models of Behaviour', *Computer Vision and Image Understanding,* pp. 398-413, Vol. 81(3), 2001.

[77] M. H. Yang and N. Ahuja, 'Recognizing Hand Gesture Using Motion Trajectories', *Proceedings of the IEEE Conference on Computer Vision and Image Understanding,* 1999.

[78] N. Cuntoor, A. Kale and R. Chellappa, 'Combining Multiple Evidences for Gait Recognition', *ICASSP,* 2003.

[79] A. Kale, A.N. Rajagopalan, N. Cuntoor and V. Krueger, 'Gait based Recognition of Humans Using Continuous HMMs', *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition,* May 2002.

[80] C. Hu, X. Wang, M. K. Mandal, M. Meng, and D. Li, 'Efficient Face and Gesture Recognition Techniques for Robot Control,' *Proc. of the Canadian Conference on Electrical and Computer Engineering (CCECE),* Montreal, Canada, May 4-7, 2003.

[81] C. Bregler, 'Learning and Recognizing Human Dynamics in Video Sequences', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* June 1997.

[82] J. Yamato, J. Ohya and K. Ishii, 'Recognizing Human Action in Time Sequential Images Using Hidden Markov Models', *Proceedings Computer Vision and Pattern Recognition,* pp. 379-385, 1992.

[83] C. Vogler and D. Metaxas, 'Towards Scalability in ASL Recognition: Breaking Down Signs into Phonemes,' *Gesture Workshop,* pp. 211-224, 1999.

[84] S. Tamura and S. Kawasaki, 'Recognition of Sign Language Motion Images', *Pattern Recognition,* pp. 343-353, Vol.21, 1988.

[85] G. Halevi and D. Weinshall, 'Motion of Disturbances: Detection and Tracking of multi-Body non-Rigid Motion', *Proceedings of Computer Vision and Pattern Recognition,* 1997.

[86] D. B. Reid, ' An Algorithm for Tracking Multiple Targets', *IEEE Transactions on Automatic Control,* pp. 843-854, Vol. 24(6), Dec 1979.

[87] T.E. Fortmann, Y. Bar-Shalom and M. Sheffe, 'Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association', *IEEE Journal on Oceanic Engineering,* pp. 173-184, Vol. 8(3), 1998.

[88] N. Sarris, D. Makris, M. G. Strintzis, 'Three Dimensional Model Based Rigid Tracking of a Human Head', *International Workshop on Intelligent Communication Technologies and Applications with Emphasis on Mobile Communications*, May 1999.

[89] F. Bourel, C. C. Chibelushi, A. A. Low, 'Robust Facial Feature Tracking', *Proceedings of the Eleventh British Machine Vision Conference*, pp. 232 – 241, Vol. 1, Sept. 2000.

[90] R. Tanawongsuwan and A. Bobick, 'Gait Recognition from Time-normalized Joint-angle Trajectories in the Walking Plane', *Proceedings of International Conference on Computer Vision and Pattern Recognition,* Dec. 2001.

[91] W. Niu, J. Long, D. Han and Y. F. Wang, 'Human Activity Detection and Recognition for Video Surveillance', *Proceedings of the International Conference on Multimedia and Expo,* June 2004.

[92] N. Oliver, E. Horvitz and A. Garg, 'Layered Representations for Human Activity Recognition', *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces,* Oct. 2002.

[93] C. BenAbdelkader, R. Cutler, H. Nanda and L. S. Davis, 'EigenGait: Motion-Based Recognition of People Using Image Self-Similarity', *Proceedings of 3rd International Conference on Audio and Video-Based Biometric Person Authentication,* June 2001.

[94] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

Appendix A

Senior Citizen Health Care: Facts and Figures

The HAR algorithm proposed in this thesis can be applied to a wide range of applications. In Section 5.8, we discussed geriatric residence surveillance as one such application where early detection of falls could be highly beneficial to the seniors. In order to reinforce our claim we discuss briefly some facts and figures related to senior citizen health care, which were presented by the National Advisory Council on Aging and by Health Canada.

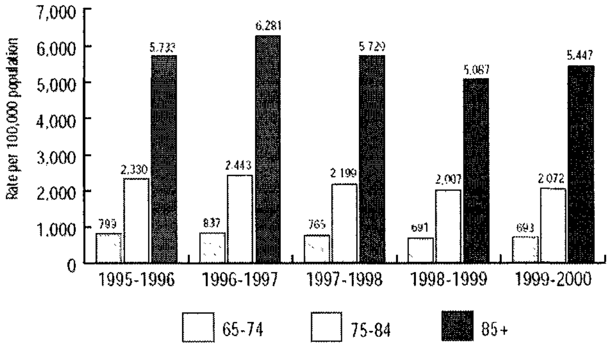
Quotes from the *Interim Report Card*, National Advisory Council on Aging: 2003

1. Of the 30 million Canadians in 2001, 3.9 million were 65 and older. Seniors are the fastest-growing age group in the country: the increase in their numbers since the 1996 census (about 360,000) is enough to populate a mid-size Canadian city, such as London, Ontario, or Halifax, Nova Scotia. Issues that concern seniors should be high on every government's agenda.

2. With respect to health status, the 2001 Report Card: Seniors in Canada had identified three areas needing improvement: injury prevention, promotion of physical activity, and suicide prevention – especially for men.

3. Council is however concerned about the persistently high rate of falls among older seniors.

CHART 1. Falls requiring hospitalization have decreased for all seniors since 1995-1996, but remain very high for seniors over 85



Source: Canadian Institute for Health Information. "National Trauma Registry". Ottawa: 1998-2002 and Health Canada. "Canadian Injury Data: Mortality-1997 and Hospitalizations -1996-97". Ottawa: October 1999.

Figure 7.1: Chart illustrating the percentage of fall related injuries in seniors.

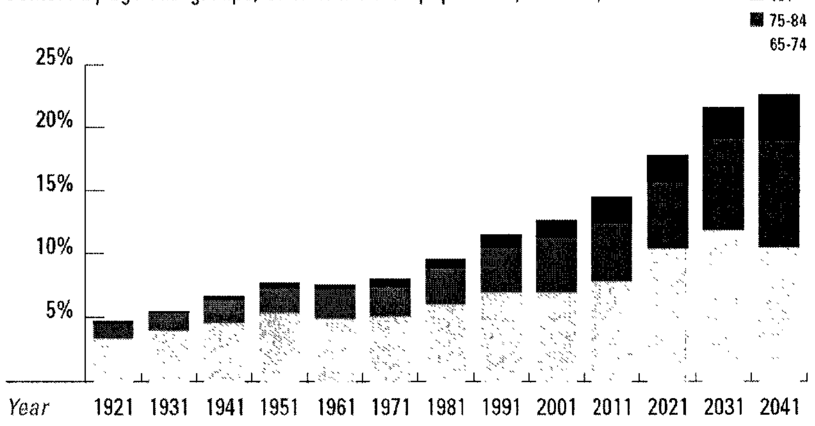
Quotes from '*Canada's Aging Population*': A report prepared by Health Canada in Collaboration with the Interdepartmental Committee on Aging and Seniors Issues; 2002.

Seniors* constitute the fastest growing population group in Canada. In 2001,it was estimated that 3.92 million Canadians were 65 years of age or older, a figure that is two thirds more than in 1981. During the same period, the overall Canadian population increased by only one quarter. The proportion of seniors in the overall population has gone from one in twenty in 1921, to one in eight in 2001. As the "baby boomers" (born between 1946 and 1965) age, the seniors population is expected to reach 6.7 million in 2021 and 9.2 million in 2041(nearly one in four Canadians). In fact, the growth of the seniors' population will account for close to half of the growth of the overall Canadian population in the next four decades.

The fastest growth in the seniors' population is occurring among the oldest Canadians. In 2001, over 430,000 Canadians were 85[1] years of age or older –more than twice as many as in 1981, and more than twenty times as many as in 1921. The proportion of Canadians aged 85 or more is expected to grow to 1.6 million in 2041 – 4% of the overall population. (See Figure 7.2 below.)

Chart 1

Seniors by age sub-groups, as % of the total population, Canada, 1921-2041



Source: See *Appendix 1*

* Please note that in this document, the terms "senior" and "older Canadian" refer to adults 65 years of age or more.

3

Figure 7.2: Chart illustrating the growing population of seniors in Canada.

Injuries

Injuries among seniors are a key concern, because of the sharp increase in the rate of injuries and injury-related deaths with age. In 1996-97, seniors aged 85 and over were 70% more likely than seniors aged 65 to 74 to suffer an injury that limited their activities. Senior women are nearly 60% more likely than senior men to suffer an injury. Falls are

[1] * Please note that in this document, the terms "senior" and "older Canadian" refer to adults 65 years of age or more.

the main cause. In the late 1990s, falls were responsible for 65% of injuries, 84% of injury-related hospital admissions, and 58% of injury-related deaths among the seniors population. The total health care costs due to seniors' falls are estimated at $1 billion annually.