The Dissertation Committee for Qingfeng Yu
certifies that this is the approved version of the following dissertation:

Human Extremity Detection and Its Applications in Action Detection and Recognition

Committee:

J.K. Aggarwal, Supervisor

Ross Baldick

Alan Bovik

Wilson S. Geisler

Choudur Lakshminarayan

Human Extremity Detection and Its Applications in Action Detection and Recognition

by

Qingfeng Yu, B.S., M.Eng.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2009

To my family!

Acknowledgments

I thank the multitude of people who helped me during my graduate study in the University of Texas.

My advisor Professor J.K. Aggarwal provided me the opportunity to pursue a Ph.D. in the Computer and Vision Research Center. I am most grateful for his patience on my progress and his financial support whenever possible. I will not be able to finish the degree without his consistent guidance during my six years in CVRC. His enthusiasm on research, critical thinking on ideas, caution on implementations, and strictness on writings have showed me thoroughly what it takes to produce good research works.

I also thank other members in my committee, including Professor Ross Baldick, Professor Alan Bovik, Professor Wilson S. Geisler and Dr. Choudur Lakshminarayan, who are always ready to help when I am in need. I sincerely appreciate the support in all aspects from Dr. Choudur Lakshminarayan in my most difficult times.

I worked with quite a few other professors in the department as a TA or grader. I thank Professor William Bard, Professor George Cardwell, Professor John Davis and Professor Nina Telang for giving me jobs. It was my pleasure to work with all of them.

I thank all my friends in the past decade. Some friends at UT also

contributed to this dissertation by helping with the video taking of certain activities, which I used as a part of my data. I thank Selina Keilani for proofreading my previous papers and this dissertation.

My parents are not doing very well in the recent years and I thank them for forgiving my long time inability to support the family physically or financially. I got married with my wife in the summer and wish we will live happily ever after.

Human Extremity Detection and Its Applications in Action Detection and Recognition

Publication No. _____

Qingfeng Yu, Ph.D.
The University of Texas at Austin, 2009

Supervisor: J.K. Aggarwal

It is proven that locations of internal body joints are sufficient visual cues to characterize human motion. In this dissertation I propose that locations of human extremities including heads, hands and feet provide powerful approximation to internal body motion.

I propose detection of precise extremities from contours obtained from image segmentation or contour tracking. Junctions of medial axis of contours are selected as stars. Contour points with a local maximum distance to various stars are chosen as candidate extremities. All the candidates are filtered by cues including proximity to other candidates, visibility to stars and robustness to noise smoothing parameters.

I present my applications of using precise extremities for fast human action detection and recognition. Environment specific features are built from

precise extremities and feed into a block based Hidden Markov Model to decode the fence climbing action from continuous videos. Precise extremities are grouped into stable contacts if the same extremity does not move for a certain duration. Such stable contacts are utilized to decompose a long continuous video into shorter pieces. Each piece is associated with certain motion features to form primitive motion units. In this way the sequence is abstracted into more meaningful segments and a searching strategy is used to detect the fence climbing action. Moreover, I propose the histogram of extremities as a general posture descriptor. It is tested in a Hidden Markov Model based framework for action recognition.

I further propose detection of probable extremities from raw images without any segmentation. Modeling the extremity as an image patch instead of a single point on the contour helps overcome the segmentation difficulty and increase the detection robustness. I represent the extremity patches with Histograms of Oriented Gradients. The detection is achieved by window based image scanning. In order to reduce computation load, I adopt the integral histograms technique without sacrificing accuracy. The result is a probability map where each pixel denotes probability of the patch forming the specific class of extremities. With a probable extremity map, I propose the histogram of probable extremities as another general posture descriptor. It is tested on several data sets and the results are compared with that of precise extremities to show the superiority of probable extremities.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The purpose of computer vision is to have machines "see". Since machines are built to serve people, many videos in computer vision focus on people. Human motion analysis has become a critical part of modern computer vision. In general, it involves detection and tracking of human beings and interpretation of human behaviors from videos. While detecting and tracking human figures are very important, they are often regarded as intermediate rather than final results of motion analysis. For many motion analysis systems in practice, behavior understanding is the goal and end product.

The importance of human behavior understanding owes to the increasing demand from all kinds of applications. In battle fields, as displayed in Figure 1.1(a), unmanned aerial vehicles take videos of military personnel from high above to identify actions such as planting mines. In public transport environments including subway stations and airports, numerous cameras are set up to monitor abnormal human behavior such as leaving baggages unattended in Figure 1.1(b). In shopping malls, store owners employ cameras to cover valuable items in the hope of preventing theft as in Figure 1.1(c). With the rapid growth of internet media, content based video retrieval becomes more

desirable than ever.

Figure 1.1: (a) Planting IED taken by UAV; (b) i-LIDS bag detection challenge; (c) Group theft in an Apple store.

Human behavior understanding is a general and loosely defined term. To be more specific, given the input data as a video stream or an image sequence, one has to temporally segment it into pieces and recognize each piece as a predefined action. A consecutive sequence of certain actions constitute a certain kind of semantic activity. In short, the behavior understanding problem consists of three parts, including temporal segmentation, action recognition, and semantic description. Sometimes temporal segmentation is not explicitly performed and action recognition is done on continuous videos. In such circumstances, it is called action detection. In this dissertation, I present my perspective on the visual understanding of human behavior, and focus on fast action detection and recognition with a particular kind of visual cue, e.g. the human extremities.

1.1 Challenges

Most of the behavior understanding in the above mentioned applications are completed by human operators. Despite various efforts from researchers, there are still plenty of difficulties before fully automated analysis is possible in practice.

There are some fundamental problems facing the entire computer vision community. For example, to recover the lost three-dimensional information from two-dimensional images is the primary difficulty in vision. According to Shah [60], the shape from stereo problem has almost been solved, while shape from motion and other similar problems have proved difficult or less interesting. Image segmentation is another well known difficulty which has not been overcome yet.

Beyond those common difficulties, there are challenges native to the human motion analysis task. First, the human body is non-rigid, its motion is articulated and body parts may have different motions. Second, under different camera views, body parts may be self-occluded and have different appearances.

To narrow down the issues further, there are specific difficulties in each of the three parts of human behavior understanding. Videos are too lengthy to be manually broken into shorter pieces. Those from surveillance cameras require fast processing to produce real time responses. In action recognition, the action labels are often predefined in a closed world. Such labels are more

human language oriented and they may turn out to be fuzzy or ill posed.

The definitions of actions overlap sometimes. For example, the KTH data set [59] has walking, jogging, and running as different categories, as shown in Figure 1.2. However, it is hard to draw a clear line between jogging and running even for human beings. Jogging is slow running in essence. How slow is slow? Is slow running not running? Therefore, except that the term "jogging" is used often in an exercise context, there is no real distinction between them.

Figure 1.2: The definitions of jogging and running overlap.

Even for the same action, there are great intra-class variations. For instance, there is a lot of variation in climbing fences, as shown in Figure 1.3. The fences may differ in height and style. The height of fences greatly affects the specific climbing action. People can easily jump over a short fence enclosing cows in a ranch, and they have to really climb a fence when it is as tall as they are. The style of fences is less critical but still important. Fences with barb or razor wires on the top greatly increase the climbing difficulty. For visual surveillance purposes, privacy fences are quite different from picket or split rail fences since it completely blocks the view on the other side. Different persons

may climb in a distinct style. Even the same person may climb in a different way occasionally.

Figure 1.3: There are huge variations in climbing fences.

1.2 Motivation

Since there are various difficulties in human behavior understanding, researchers usually work only on a part of it. For human action recognition, one traditional approach is to represent each frame with certain features. With the frame descriptors one may classify the entire action either with sequential analysis methods or by simple majority voting. In such circumstances, the representation of human figures inside a frame greatly determines how effective the entire action classification system will be. So what is an effective and efficient human posture representation?

Johansson [33] demonstrated that locations of human body joints are effective visual cues for human recognition of activities. He attached lights on human body joints and took videos of human actions in the dark. As shown in Figure 1.4, the set of points in an image does not really follow any Gestalt

principle, which is often used in psychology to group scattered cues. But when viewing the points in an image sequence, observers can quickly find a vivid human figure in action. In essence, human visual systems can recover object information from very sparse inputs such as a set of points in motion. This phenomenon is known as biological motion in the biological vision literature.

Figure 1.4: Biological motion: human visual systems can recognize actions from inputs as sparse as a set of body joints.

Stimulated by Johansson's experiments, Webb and Aggarwal [72] proposed to estimate the structure of jointed objects from motion, where jointed objects have two visible points on each rigid part. Consistent with Johansson's method, modern motion capture systems generally have performers wear suits with distinct markers to identify such body joints. In the past, there have been extensive studies following Johansson's moving light displays (MLD), as reviewed by Cedras and Shah [9].

However, body joints are not easily available from videos or images directly. Can one replace body joints with other points to represent a human body? In Figure 1.5, I display a few images of another set of points, instead

of body joints. For human observers, it is as easy to identify the action as the same "jumping jack" as in Figure 1.4. This new set of points includes heads, hands and feet, which I call extremities.

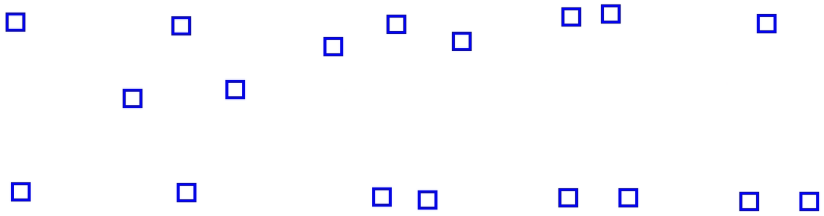

Figure 1.5: Recognizing human actions from their extremities including head, hands and feet. For display purpose, each extremity is drawn as a square.

1.3 My approach

In this dissertation I present the unique idea of using human extremities, including heads, hands and feet, as a powerful cue for fast action detection and recognition.

I propose to extract precise extremities from contours. Starting with the star-skeleton representation as a baseline comparison, I propose the two-star-skeleton and the variable-star-skeleton representations. I utilize the concept of a star polygon from the computer graphics community to illustrate why the number and locations of star points matter for extremity detection. I present experimental results on a set of 1000 images taken from videos of persons climbing fences to verify the variable-star-skeleton performs best in

detecting extremities.

With the precise extremities, I propose to generate both task specific and general purpose action descriptors. For the specific task of fence climbing from continuous videos, which consists of mixed actions, I define a set of features from the relative spatial configuration of extremities against fences. Such features are employed in a block-based Hidden Markov Model (HMM) to decode the fence climbing action from continuous videos. For general action recognition, where each action is already temporally segmented, I define a spatial histogram of the extremities for each frame. All the unique histograms form the observation symbol set for the HMM training and testing.

With the precise extremities, I further propose the concept of stable contacts, which are those extremities that do not move for a certain amount of time. The change in the number of stable contacts indicates a pose change in human actions. By monitoring such changes, I decompose the continuous videos into smaller pieces where each piece has a fixed number of stable contacts, which usually correspond to a phase in human action. I describe how to take advantage of such temporal segmentations and search different actions in continuous videos.

When contours are not available, I propose to model the extremity as an image patch instead of a single point on the contour, which helps overcome the segmentation difficulty and increase the detection robustness. Extremity patches are represented with Histograms of Oriented Gradients. The detection is achieved by window based image scanning combined with the integral

histograms. The result of approximate detection is a probability map where each pixel denotes the probability of the patch forming the specific class of extremities.

With such a probability map, I propose the histogram of probable extremities, as a compact human posture representation. A Support Vector Machine is used in classifying individual actions. I present experimental results on a few data sets to show the effectiveness of the proposed action descriptor.

1.4 My contributions

In short, my contribution in this dissertation is to propose and validate the effectiveness of extremities for fast human action detection and recognition.

In overview, my contributions are listed as follows.

1. **Human extremity detection [81]:** To the best of my knowledge, there is no prior work that detects head, hands and feet simultaneously. A vast amount of research has been devoted to face detection, as reviewed by Yang et al. [75]. Some researchers work on head detection and even hand detection. One significant difference between previous and my work is the resolution at which the videos are taken. In other words, the field of view is different. For example, Kölsch and Turk [36] applied the method from Viola and Jones [67] to detect hands. As their application is hand gesture recognition, their videos cover mostly the hands and no feet are visible. It is similar in face detection where videos usually focus only on

the upper body. In my work, the resolution is significantly coarser, since the video has to cover the entire human body at various postures.

2. **Extremities for fast human action detection [78–80] and recognition [81]:** There are at most five points in my precise extremity definition, which provides very limited and restricted information. However, I show that when used properly these extremities are powerful cues for action detection and recognition. My experiments on various data sets demonstrate that the accuracy on action classification is comparable to the most state-of-the-art algorithms, which is very impressive considering that those other algorithms employ significantly more input information and hence consume more computation resources.

In addition, I invent some novel techniques.

1. Variable-star-skeleton [81]. When the segmentation yields reasonably clean contours, the variable-star-skeleton provides accurate localization of extremities on the contour. I develop the technique in three stages. First, I propose the two-star-skeleton and observe that it is better than the single-star-skeleton. Second, I use the concept of the star polygon to explain that the appropriate number and locations of the stars help increase the detection accuracy, since in this way the human silhouette can be approximately decomposed into star polygons. Third, I propose to have junctions of medial axis as stars, extract candidates from these

stars and filter them out with the robustness, visibility and proximity criteria.

2. Histogram of Probable Extremities. When clean contours are not available, I model the extremities as image patches. Each patch is represented by a feature vector through available techniques such as Histogram of Oriented Gradients. By window scanning the image with integral histograms, I build a probability map where each pixel denotes how likely it is the center of an extremity patch. Then I lay a set of spatial cells over the map and compute the histogram of probability over each extremity class and cell. The resulting histogram is a vector capturing the spatial distribution of probable extremities in an image.

3. Stable contacts [79, 80]. Precise extremities not only tells the spatial configuration of human body parts, it also provides temporal information on human actions. I define stable contacts to be those extremities that do not move over a certain time. The durations of those stable contacts are used to form primitive intervals where each interval corresponds to a phase in human actions. In this way, I can group frames in an image sequence into frame blocks and achieve action detection faster and in a more meaningful way.

1.5 Outline

The rest of the dissertation is organized as follows. I review relevant works in Chapter 2. The detection of precise extremities from contours is presented in Chapter 3, which features the development of the variable-star-skeleton and its comparison with two previous works. Next in Chapter 4, I describe how to make the full use of precise extremities in action detection and recognition. In action recognition, the precise extremities are used to generate both task specific and general purpose action descriptors. I present the detection of probable extremities in Chapter 5. In the same chapter the probable extremity map is converted into a histogram as a general purpose action descriptor. Finally the conclusion is given in Chapter 6.

Chapter 2

Relevant Works

As human motion analysis is such a broad topic, in this chapter I concentrate on the research that either has something to do with my own research, or helps strengthen my understanding of the area. I briefly discuss the different kinds of human behaviors and introduce typical actions researchers are interested in. Shape and motion are the two cues used most often in human motion analysis, so I describe a few papers that used silhouette or its equivalent, contour, as input of the approaches. In particular, I collect all works that involve usage of a star-skeleton representation. While silhouette or contour is an explicit shape representation, it is difficult to obtain sometimes. In comparison, Histograms of Oriented Gradients (HOG) is a popular technique that represents shapes implicitly. Hence, I also introduce those works involving the usage of HOG. Next, I review those works that explicitly use optical flow to capture motion information. As my own approach uses extremities only, I also review other part based methods and discuss their advantage against holistic methods. Finally, I introduce the emerging trend of using interest point detectors and descriptors as an unstructured representation, which is different from my or other part based methods where parts have a structure.

For more complete coverage, I list some excellent review papers as follows. Aggarwal and Cai [1] focused on three major components of human motion analysis, including segmentation of body parts in images and reconstruction of the 3-dimensional body structure from trajectories of such body parts, tracking human beings with multiple cameras without identifying body parts, and recognition of human movements. Gavrila [26] discussed various methods grouped in 2-dimensional approaches without explicit models, 2-dimensional approaches with explicit models and 3-dimensional approaches. Shah [60] gave possible reasons for slow progress in human behavior understanding, presented their work on human tracking, representation and recognition, and commented on promising future solutions. Wang et al. [69] organized their reviews in a hierarchy according to the general framework of human motion analysis, with the emphasis on grouping methods on each task within the framework. Pantic [49] narrowed down the definition of human behavior as affective and social signaling, and discussed how far we are from embedding computers into human centered daily lives.

2.1 Types of human behaviors

According to Bobick [8], machine perception may focus on one of three levels: movement, activity, and action, ranked by their complexity. In English, the word "activity" sometimes sounds more complex than "action". In computer vision literature, people often use the two words interchangeably. In my understanding, human behavior refers to observations of certain patterns

of human actions over a relatively long time, although some researchers use it equivalently with actions and activities. Hence, a better ranking is: movement, action, activity, behavior, with increasing complexity in time or involved subjects. Anyway, the differences are very subtle and there is no real standard definition.

By using the relationship between humans and environment, one may broadly divide human actions into three types: (a) single person actions, such as walking, bending and sitting [2], where the action is performed by a single person and involves no interaction with the environment; (b)interactions between persons, such as following and leaving [58], hugging and punching [51], greeting and fighting [57]; (c) actions involving inanimate objects, such as opening a file cabinet [56] and digging [39]. In this dissertation I actually assume there is only one person in the video, hence I do not need any tracking part and all actions involved are single person actions.

2.2 Silhouette or contour for explicit shape feature

Many earlier works utilized silhouettes (blobs) or contours as a starting point for human representation.

Davis and Bobick [17] represented human movements by temporal templates, which are vector images wherein each pixel records some function of the movement at that pixel. In the two component temporal template, one component of the vector is a binary value representing the occurrence of motion, and the other is a recency function that describes how recently the motion

occurred. In other words, the temporal template can be split into the Motion History Image (MHI) and Motion Energy Image (MEI), wherein the MHI is formed by stacking time weighted foreground masks and the MEI is its binarized version. They test the matching algorithm on sequences of 18 aerobic exercises. The temporal segmentation is achieved by approximately searching over a wide range of the movement duration parameter. An example of temporal template is shown in Figure 2.1 computed on a short sequence of walking.

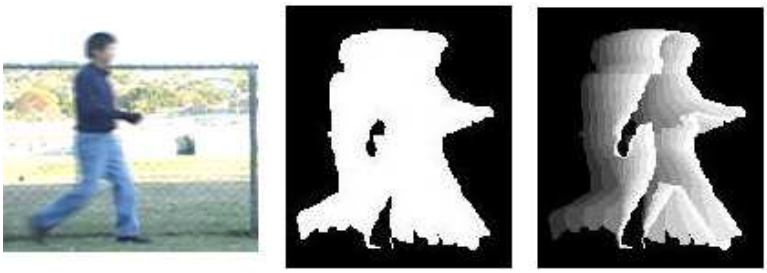

Figure 2.1: Shown in the left is one out of about 12 frames of a person walking. Shown in the middle is the Motion Energy Image, while on the right is the Motion History Image.

A later extension called Motion History Volumes by Weinland et al. [73] generalized the temporal templates to three-dimension for free viewpoint action recognition. For human figures in each of the multiple camera views, the silhouette is obtained by background subtraction. From these silhouettes, the visual hull is constructed and accumulated over time to form the Motion History Volume (MHV). They further transform these MHVs into cylindrical coordinates around the vertical axes of a human visual hull, and extract

view-invariant features with Fourier analysis.

Blank et al. [6] built a space-time three-dimensional action shape induced from silhouettes. Given the silhouette of a human figure, each pixel inside the contour is associated with a value, which is the average time that it takes for the pixel to randomly walk into a contour point. A Poisson equation is used to model such a measure. Solutions to all Poisson equations are stacked together to form a space-time shape, from which a set of action features are extracted, including space-time saliency, orientations, etc.

Yilmaz and Shah [76] presented a similar approach where human contours are stacked to form a spatiotemporal object, which they call spatial-temporal volume (STV) in the (x,y,t) space. It differs from other works that stack entire frames in that they segment the contour, find point correspondence between contours and stack contours according to the point correspondence. They analyze the STV with differential geometric surface properties including peak, ridge, saddle ridge, flat, minimal, pit, valley and saddle valley. The set of such points is called an action sketch. As each action sketch consists of a set of 3-dimensional points, the action classification becomes a problem of point matching as formulated in epipolar geometry.

All these works require foreground segmentation as precise as up to the blob or contour level.

2.2.1 Works involving star-skeleton

Fujiyoshi and Lipton [25] proposed a star-skeleton model (SS) to analyze human motion. The center of mass of a human silhouette is extracted as the star. Distances from contour points to the star are computed as a function of indices of clockwise sorted contour points. Their initial goal is to use such a representation for feature extraction to recognize cyclic human actions such as walking and running. Their features include the angle between the left leg and the vertical axis passing through the human blob centroid and the angle between the line from head to the star and the vertical axis, as shown in Figure 2.2.

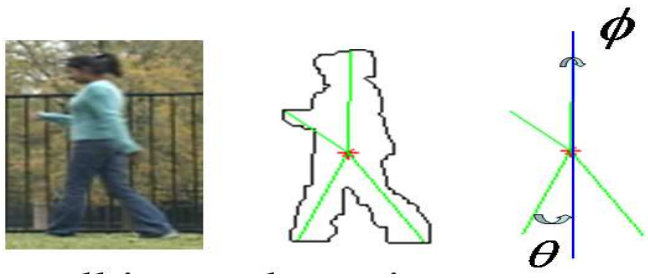

Figure 2.2: Displayed from left to right are the source image, the segmented contour with the star and skeletons, and the features extracted from the representation.

Petkovic et al. [52] used the star-skeleton to find out parts of the human body that stick out. However, they only consider those parts that fall within a pre-defined portion centered around the body, to emphasize the hand movements.

Peursum et al. [53] used a modified star-skeleton in each of the multiple views of an action and fused the 2-dimensional star-skeletons into a 3-dimensional one. They modified the star so that it is no longer the blob centroid but the "shoulder" point of the body, which is defined as the point at one third from the head to the centroid. This creates the problem of finding the head, which is solved by designating the highest extreme point from the star-skeleton as the head. However, the head is not always the highest point. Chen et al. [11] employed the same modified star-skeleton in their work on estimating 3-dimensional body pose.

To utilize structure information available in the star-skeletons, Chen et al. [12] defined a distance function between two star-skeletons. Each star-skeleton is converted into a vector of five extremities. If less than five are detected, they fill the rest with zero. If more than five are detected, they increase the noise smoothing level to remove extra extremities. The distance function is defined as the sum of Euclidean distances between five matched pairs of skeletons. Such a distance function is used in their HMM-based action recognition system.

2.3 HOG for implicit shape feature

Dalal and Triggs [15] proposed a Histogram of Oriented Gradients (HOG) for human detection in still images. First, the image gradients are computed. Then, the image is split into a dense grid of spatial cells. Inside each cell the gradients are grouped into orientation bins, with the gradient

magnitudes as weights. The concatenation of all local histograms forms the final descriptor.

Even before the HOG appears, there is a partially identical work that uses oriented gradients for histograms to describe regions of a human body. Shashua et al. [62] divided the area of interest, which is the bounding box of a human body, into a fixed set of 9 regions. Among the 9 regions, region 1,2,3 represents the head, upper body, and lower body respectively, while region 4,6,8 represents the left,right and middle of the upper body respectively, region 5,7,9 represents the left, right and middle of the lower body respectively. Some pairs of these regions also form the additional region 10,11,12,13. Each of the 9 regions is further divided into 2 by 2 sub-regions with 8 orientation bins, hence represented as a 32-element vector. Up to here, the approach is very similar to the HOG technique. Then a procedure called ridge regression is applied, to assign a discriminant value for each region, where the value is the inner product between a region and a weight vector. The entire area of interest is converted into a $13 * 9$ element feature vector.

Later, Wang and Suter [71] employed a similar partition over human figures. They divided each silhouette image into $h * w$ non-overlapping sub-blocks. For each sub-block, the number of foreground pixels is divided by the maximum number of foreground pixels over all sub-blocks, to produce a normalized value as the representation of how much the block is covered by foreground.

In addition to human detection in images and videos, HOG is quickly

extended to non-human object detection as well. As a part of the 2005 PASCAL Visual Object Classes Challenge [20], Dalal and Triggs used HOG to detect cars, motorbikes and persons and won the detection competition for the car and person classes.

Furthermore, researchers started to represent human poses with HOG features for action recognition. Thurau [64] divided each detector window of size $40 * 80$ into evenly distributed cells of size $10 * 10$, and represented each window as a vector of size $4 * 8 * 9$ for further clustering to produce action primitives. Different from [64], Hatun and Duygulu [30] computed the HOG in a radial grid structure for each frame.

2.4 Optical flow for motion feature

Efros et al. [19] proposed a novel optical flow based motion descriptor, for recognizing actions of human figures about 30 pixels tall. They first track and stabilize human figures. Then they compute the optical flow between two adjacent frames with the Lucas-Kanade [42] algorithm. The optical flow field F is split into F_x and F_y, corresponding to the horizontal and vertical components. The two components are further half-wave-rectified into the four channels, F_x^+,F_x^-,F_y^+,F_y^-, so that each channel has only positive values. These four channels are further smoothed with a Gaussian, to form the final motion descriptor. The distance between two action sequences are defined as the normalized correlation between motion descriptors.

Similar to Efros et al. [19], Fathi and Mori [21] added one more channel

called zero motion, F_0 by computing the L_2 norm of the four channels. Then they treat $\hat{F}_c(p)$ as a low level feature, which is the value of channel c for the pixel at location p. They partition the spatial-temporal volume of each action sequence as evenly distributed cuboids. With each cuboid as the mid-level motion feature of a weak classifier, they apply the AdaBoost algorithm to train a strong classifier which is a linear combination of those weak classifiers.

As opposed to the two representative works above, some researchers chose to build histograms out of raw optical flow fields to have a more compact and robust motion feature.

Dalal et al. [16] proposed the Motion Boundary Histograms, which is essentially a replicate of the HOG descriptor on optical flow fields. Since optical flow fields have two channels including horizontal and vertical directions, each channel is treated separately like an image. For each channel, the local gradients are computed. The gradient magnitudes and orientations are used for magnitude weighted votes in the orientation histogram of local neighborhoods, where each neighborhood is a spatial cell in a block, just as the standard Histogram of Oriented Gradients on gray scale images. In their work, the goal is to detect human beings from images.

Laptev and Pérez [38] adopted the Motion Boundary Histograms technique with a different name, histograms of optical flow, to represent motion. In their work to detect actions such as "drinking" from movies, the histograms have 5 bins with four corresponding to four discrete motion directions and the last bin corresponding to no motion.

Although the name is "histograms of optical flow" in the two works [16, 38], the histograms are in fact built out of derivatives of optical flows, with the consideration that relative motion is more important in discriminating actions from each other. In contrast, Ikizler et al. [31] built the histograms of optical flow literally. Their histograms have only 4 bins corresponding to the directions of 0,90,180,270. Each spatial bin corresponds to a cell inside a block. For each cell, the optical flow associated with each pixel is projected into the four directions and summed over the entire cell. The histograms from different cells are concatenated to form the histogram for each pair of adjacent frames.

Li [40] used oriented histograms of optical flow field in his Hidden Markov Model based framework for action recognition. However, it is not clear whether the work treats the histogram of optical flows as a global feature over an image, or a local feature within a spatial cell, as the paper never mentions spatial bins inside the image. Ignoring the spatial configuration of optical flows might significantly hurt the overall effectiveness of the motion feature.

Chaudhry et al. [10] proposed to abbreviate Histogram of Oriented Optical Flow as HOOF. They further modify the histogram in [31] by changing the four orientation bins to the four regions symmetric along the vertical axis, in order to allow actions in reverse directions. Then they generalize the Binet-Cauchy kernels to nonlinear dynamical systems for action recognition.

2.5 Part based v.s. holistic

In general, an object can be described by breaking it down into multiple parts and specifying the spatial relationships between parts. Such part based representations are mostly used in object recognition, including detection and localization.

Fischler and Elschlager [24] proposed the pictorial structure model. The basic idea is to model an object through a collection of parts arranged in a deformable configuration. The appearance of each part is modeled separately, and the deformable configuration is represented by spring-like connections between pairs of parts. These models allow for qualitative descriptions of visual appearance, and are suitable for generic recognition problems.

Felzenszwalb and Huttenlocher [23] presented a statistical framework for modeling the appearance of objects with the pictorial structure models [24]. Their contribution is to present efficient algorithms in both finding instances of an object in an image and training tree structured object models from training images.

Crandall et al. [13] proposed the k-fan models for more general object classes that do not necessarily have tree structures. When $k = 0$, there is no dependence between locations of parts. When $k = 1$, the structure becomes the star-skeleton representation. When $k = n - 1$ where n is the number of parts, there are dependencies between all pairs of parts. The models are tested on detecting airplanes and motorbikes.

In human behavior understanding, some approaches have an explicit part based model for the human body.

Ju et al. [34] proposed the "cardboard people" model, where the limbs of a person are represented by a set of connected planar patches. The motion of the limb is estimated from optical flow fields by treating the limb as a chain structure of rigid objects. Their experiments are conducted on "walking" with only two legs visible.

Haritaoglu et al. [27] developed a real time system to estimate human body pose and detect body parts from silhouettes. The system uses a silhouette-based body model which consists of 6 primary body parts (head, hands(2), feet(2), and torso) and 10 secondary parts(elbows(2), knees(2), shoulders(2), armpits(2), hips, and upper back). It first compares the human body contour with predefined templates to estimate body posture. Then the head position is detected and other body parts are estimated with the topology of the estimated body posture. Their work was later included in the W^4 system [28].

Park and Aggarwal [50] used a hierarchical human body model, where a body is divided into the head, the upper body and the lower body. Furthermore, the head has hair and a face, the upper body has hands and torso, and the lower body has legs and feet. A maximum a posterior (MAP) classifier is employed to assign each blob into a body part.

There are a few advantages of part based v.s. holistic representations.

1. Better representation power: Since a part based model usually just has a few fixed parts and connection between parts, it has a very flexible structure and can represent much more object classes than those holistic methods such as template based methods. For human beings with articulated motions, this is particularly useful, since human postures have huge intra-class variations.

2. More robust to occlusions: When objects are occluded partially, the holistic methods usually cannot work as well, since there is missing data in the representation. However, as long as the key parts are still visible, it should have no influence on part based methods. In some cases, some parts may still get occluded, but it is not as severe for part based methods as for holistic methods.

2.6 Interest points as unstructured representation

In holistic methods, object structures are implicitly coded into the algorithm. In part based methods, parts are explicitly detected and their spatial relationships are also modeled. Unlike those methods, there are a considerable amount of works in recent years that ignore the structure inside an object.

Probably the most important reason for this phenomenon is the success of the "bag of words" model in the text mining community. For example, before the search engine Google appears, companies such as Yahoo and AskJeeves were attempting to give structured and semantically meaningful answers to

queries submitted by Internet users. Such efforts proved in vain, since manually sorting out Internet documents is prohibitively expensive. Nowadays, it has become almost standard to just represent each document by the bag of words model. For each document, it is represented as a frequency vector where each element denotes the frequency of a certain word in the document. There is not any structural information kept in such a sparse feature vector. Such a basic representation is further enhanced by the Term-Frequency-Inverse-Document-Frequency (TFIDF) weighting scheme.

Researchers in computer vision are borrowing the model and its accompaning techniques such as Latent Semantic Analysis and its variants. The model and techniques are first replicated in the object recognition area and later extended into the human motion analysis as well. In order to build visual words out of images or videos, researchers have tried different techniques for feature detectors and descriptors.

Vogel and Schiele [68] proposed a two-stage system for content based image retrieval. In the first stage, an image is divided into small patches of equal sizes and a classifier is employed to determine which class a patch is from. In the second stage, all decisions over these small patches are accumulated to represent frequency of occurrence for each patch class. In this way the patches are regarded as visual words and the images are documents in the bag of words model.

Vidal-Naquet and Ullman [66] selected informative fragments to represent images. They first cropped a large set of image patches of different sizes at

random locations, then computed the optimum threshold for each fragment to be determined present at the image, and finally selected a set of such patches that convey the maximum amount of information about the class. Maree et al. [43] proposed a similar strategy on building visual words from random sampling.

Barnard et al. [4] built visual words out of image segments. They first segmented images with normalized cuts, and then selected 8 largest segments. Each segment is represented by a set of 40 features that reflect size, shape, texture, position, color, etc.

The most popular way of building visual words out of images or videos might be due to the interest point detectors, such as Harris corner detector [29], the saliency detector [35] and Lowes DoG [41]. With interest points detected, one can choose different descriptors for the image patches centered around them, such as SIFT [41].

Fei-Fei and Perona [22] selected local patches from images with different strategies including evenly sampled grids, random sampling, the saliency detector and the DOG detector. For each detector, two different descriptors are used, including normalized gray scale intensities and the SIFT descriptor. Furthermore, the patches are clustered to yield codewords and all unique codewords form the visual vocabulary.

In addition to works originally designed for object detection in images, there are also interest point detectors for action recognition in videos. As an

extension of the Harris corner detector, Laptev and Lindeberg proposed the space time interest point [37]. Dollár et al. [18] proposed sparse spatiotemporal features to recognize human and rodent behavior.

Niebles et al.[45] modeled the action as a bag of visual words, ignoring the spatial and temporal relationships among the words. Both static and motion features are computed. For static features, a set of points are sampled along the edges and a shape context descriptor [5] is computed around each sampled point. For motion features, the separable linear filter [18] is used to capture human movement characteristics.

Chapter 3

Precise Extremities from Contours

In this dissertation human extremities refer to human heads, hands and feet, which provide useful information about human movements. There are different ways to define the extremities in details such as their locations, scales and representations. In the simplest case, extremities are modeled as points along the body contour. In this chapter, I present how to detect extremities as points precisely from contours.

3.1 Extracting contours

In video analysis for human behavior understanding, a given video is often decomposed into an image sequence first. Working with image sequences taken under unrestricted settings poses many challenges for successful segmentation. For simplicity, in this dissertation I assume there is only one person in an image sequence. Under such settings, there is no need for tracking, as long as the area of interest is detected from each frame.

For images sequences taken by a stationary camera, the common approach is to build a statistical background model for background subtraction, where each pixel follows a normal distribution. This is followed by thresh-

olding and binary morphological operations. To ensure there is just one blob extracted, I extract only the largest blob and ignore all smaller ones after connected component analysis. The contour is obtained from the blob with a border following algorithm. Shown in Figure 3.1 is an example of such procedures applied on a frame.

Figure 3.1: Extracting contours from frames by background subtraction.

When image sequences are taken by a moving camera, the background subtraction method is not applicable any more. The problem here can be formulated as contour tracking. For details, readers can refer to Yilmaz et al. [77].

3.2 The star-skeleton representation

My first attempt to extract extremities is to use the star-skeleton model to represent the human body. The contour points are sorted clockwise by their indices. With the blob centroid as the star, the distances between contour points and the star are computed. In this way, the distance from the star to a contour point is a function of the index of the point. The function is then smoothed by a Gaussian to extract points where the distance is the largest in

its contour neighborhood. These contour points whose distance reach a local maximum are regarded as extremities.

3.3 The two-star-skeleton representation

Later, I proposed to have a two-star-skeleton representation for detecting extremities. First two stars are chosen. The first star point is the blob centroid, and the second star point is the highest contour point. For each star, distances between the star and all contour points traversing clockwise from the highest contour point are computed. After the distances are smoothed with a Gaussian kernel, two curves of distances varying along the contour are obtained. For better understanding, an example is shown in Figure 3.2 and Figure 3.3.

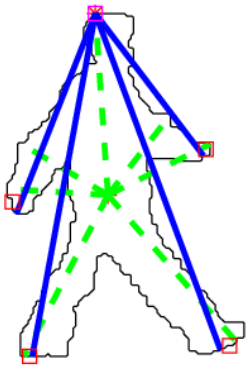

Figure 3.2: An example of the two-star-skeleton representation, where the two stars are the centroid and the highest contour point. Blue solid lines represent skeletons from the second star, while green dash lines represent skeletons from the first star.

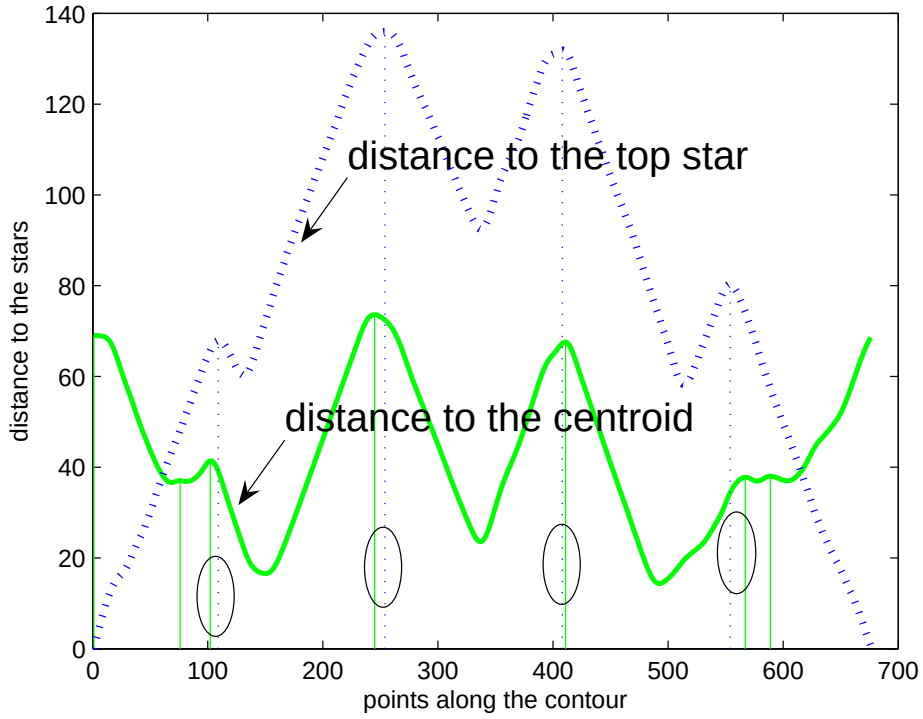

Figure 3.3: An example of computing the extremities from the distances between the stars and the contour points.

Next, I extract all those local peaks where the distances reach the largest values among their local contour neighborhood and record the two sets of indices of corresponding contour points. I then group the two sets of indices into pairs by proximity and use the mean index of each pair as the index of an extreme point.

In order to get the best pairing, I compute a cost function for each possible pairing between the two sets, and search exhaustively as explained

below.

Given a contour with perimeter NC (number of contour points), I have two sets of numbers: $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$. Each number is an index of a local extreme point with respect to a star along the clock-wise contour. For each number in one set, it can either have a one-one correspondence in the other set or be left unmatched. So in a possible pairing, I split all the numbers into three portions, including unmatched numbers in A denoted as A', unmatched numbers in B denoted as B', and numbers of A or B with one-one correspondence denoted as AB. I define a cost function $C(A', B', AB)$ as

$$C(A', B', AB) = C(A') + C(B') + C(AB) \tag{3.1}$$

$$C(A') = |A'| \cdot (0.5 \cdot \alpha \cdot NC - c) \tag{3.2}$$

$$C(B') = |B'| \cdot (0.5 \cdot \alpha \cdot NC - c) \tag{3.3}$$

$$C(AB) = \sum_{\forall \langle a_i, b_i \rangle \in AB} \min\{|a_i - b_i|,\ NC - |a_i - b_i|\} \tag{3.4}$$

In equation 3.1, I compute the overall cost function as the sum of costs of both un-matched points and matched pairs. In equations 3.2 and 3.3, $|X|$ is the set cardinality, and $\alpha \cdot NC$ ($\alpha = 0.05$ in my experiments) acts as a threshold to judge if a pair should be kept or broken apart. Note that c represents an arbitrarily small value to break the tie when a pair of points are exactly at a distance of $\alpha \cdot NC$ from each other. One can also choose randomly without

using c, when there is a tie. In equation 3.4, I sum the circular distances across all pairs as the cost. The minimum function is to select between the two distances, since there are always two distance between two points on a closed contour.

The idea here is to form pairs of numbers if they are close enough and leave them apart if they are far away from each other. In my experiment, as all of my sequences are taken in very similar camera and scenario settings, sizes of human blobs do not vary much among the frames. Hence I didn't change α much in my experiments. When I change α in a reasonably small range, it yields very similar results. If I change α too much, for example from 0.05 to 0.5, the result does not make sense as I group two candidates that are half of the contour away from each other.

I search over all possible pairings and compute the cost function for each to find the optimum pairing in the sense of minimizing the cost function, as implemented in four steps.

1. Build a matrix D of size $m{\cdot}n$, where $D_{ij} = \min\{|a_i - b_i|,\ NC - |a_i - b_i|\}$. Each row represents a point in A, and each column represents a point in B.

2. Thresholding D by $\alpha \cdot NC$ to produce an indicator matrix E, where $E_{ij} = 1$ if $D_{ij} < \alpha \cdot NC$, and 0 otherwise.

3. Without loss of generality, I iterate through columns of matrix E to compute the total number of possible pairings as $\prod_j (E_j + 1)$ where E_j is the number of 1's in column j. Note that having no 1 entry in a column of E

means the point in set B is left unmatched. One possible pairing corresponds to selecting none or one point from A for each point in B.

4. For each possible pairing, I compute the cost function as defined in equation 3.1. The best pairing is the one with the minimum cost.

3.4 On the number and position of stars

Why should I propose the two-star-skeleton over the simple star-skeleton? The simple answer is that the number and positions of the stars matter. In this section, I first illustrate my motivation for analyzing the number and position of stars with an example in Section 3.4.1. Next I connect my observation with the visibility and star polygon concepts in Section 3.4.2. Then I analyze both the single and two star skeleton representations with the concepts in Section 3.4.3.

3.4.1 Observation

If there is only one star in a star-skeleton representation, the position of the star greatly effects, if not determines, whether a contour point could be a local peak in the contour neighborhood hence be a possible human extremity.

An example is given in Figure 3.4 showing a climbing person. I focus on detecting the left hand here. The part of the contour around the left hand is highlighted with a green solid line, while the other parts are shown with a black dash line. For illustration purposes, four stars are chosen as shown with blue asterisks and numbered. The detected hand from each star is shown with

a solid red square and numbered accordingly in Figure 3.4. From this example, it is obvious that the fourth star provided the best approximation, the second star made a close one, and the other two produced incorrect extremities.

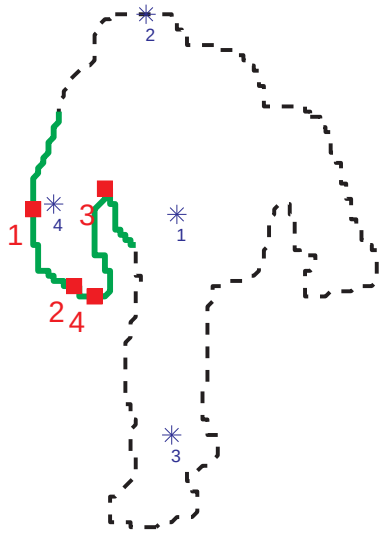

Figure 3.4: The four stars are shown with blue asterisks and their respective detections of left hand shown with solid red squares with corresponding numbers.

3.4.2 The star polygon concept

Before I proceed, I briefly review some concepts from the computer graphics community to make the paper self-contained.

Given a human contour represented by a set of clockwise sorted contour points, I treat it as a simple polygon P. In geometry [61], a *simple polygon* is a polygon whose sides do not intersect unless they share a vertex. A point in the polygon (including interior and boundary) is *visible* with respect to another

point in the polygon if their line segment falls completely within the polygon. For example, in Figure 3.5, point D is visible to point E, F, G while not visible to H.

According to Shapira and Rappoport [61], if there exists a point $v \in P$ that is visible from any other point inside P, then P is a *star polygon* and v is a star point. Since not every polygon is a star polygon, they further defined the star skeleton to decompose a simple polygon as a star set and the associated skeleton. Simply speaking, the star set is a set of star polygons such that each shares at least one edge with another star polygon; the skeleton is a tree that connects star points and mid points of the shared edges.

Shown in Figure 3.5 is the same simple polygon decomposed into two star polygons in (a) and into three in (b). In Figure 3.5(a), points A, C are star points and the connection ABC is the skeleton. In Figure 3.5(b), points D, F, H are star points and the connection $DEFGH$ is the skeleton. Obviously the star-polygon decomposition is not unique.

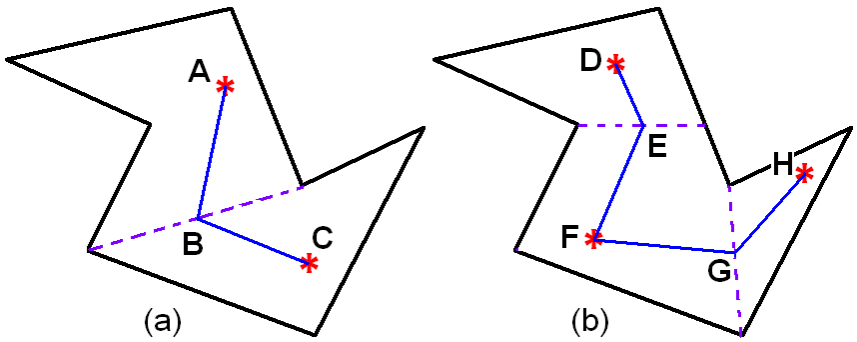

Figure 3.5: Two different decomposition on the same simple polygon.

3.4.3 Characteristics of star skeleton representations

Why do different stars produce different approximations of the left hand in Figure 3.4? There are many possible explanations such as distance, scale, and visibility. Among all the factors, I regard visibility as the most important one. The reason that the second and fourth star perform better is because the left hand is visible to them, while not visible to the other two.

Fujiyoshi and Lipton [25] considered the human centroid as a single star. As human contours are usually not star polygons, a single star cannot be visible to all contour points. Hence the star-skeleton will easily miss true human limbs or produce false alarms. In extreme conditions, the centroid may not even be inside the human silhouette.

In my previous improvement [78], I added the highest contour point as the second star. It can be interpreted as an intention to make all those points not visible to the center of mass visible to the second star. This way, it is hoped that most contour points will be visible to at least one of the two stars. This strategy is intuitive and reasonable; however, its practical effect is weakened in two aspects. First, it is a problem whether or not to treat the highest contour point as an extremity. In most human postures, the highest contour point is the head, hence it is desirable to include the second star as one of the detected extremities. When the assumption is violated, the inclusion might produce false alarms. Second, two detected limbs (each from a different star) are paired up and averaged, which means a good detected extremity is compromised by a bad one. I would rather have the algorithm select the good

ones and discard the bad ones.

Using a frame of a person climbing fence, I show in Figure 3.6 the visibility of each contour point with respect to the center of the mass and the highest contour point. Details of computing such visibility are described later in Section 3.5.3. It is obvious that with only the center of mass as the single star, a considerable portion of the contour is not visible. With the addition of the second star, more contour pieces are covered, while there is still a significant portion not visible.

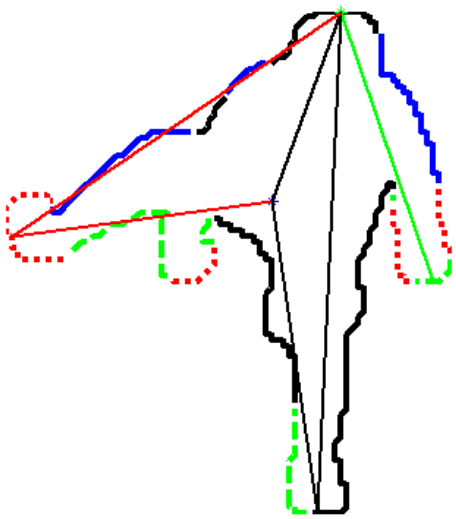

Figure 3.6: The two stars are in blue and green respectively. Contour points visible only to the center of mass are shown with a blue solid line, visible only to the highest contour point shown with a green dash line, visible to both shown with a black solid line, visible to neither shown with a red dotted line. Best viewed in color.

Note that in both works [25, 78], the so-called stars are just approximations of star points as defined in Section 3.4.2. Considering human contours

as simple polygons, my ultimate desire is to choose an appropriate number of "stars" and their positions so that many contour points are visible to at least one "star", e.g. making the approximation as good as possible.

3.5 The variable-star-skeleton representation

In this section, I develop a variable star skeleton (VSS) representation, motivated by observing that more and well positioned stars make contour points more visible. Although built upon previous works [25, 78], my new representation is considerably different in two aspects, including finding stars and producing extremities out of multiple sets of candidates. I take as stars, junction points in the medial axis of the human silhouette, which may be regarded as a rough approximation of human body joints. Each star will produce a set of extreme points, as previously done in SS and 2SS. As a candidate, each extreme point will be processed according to its robustness to noise smoothing, visibility to the generating star, and proximity to its neighbors.

3.5.1 Detecting junctions of a medial axis

For contours, a medial axis is the union of all centers of inset circles that are tangent to at least two contour points. In order to compute the medial axis, I choose the augmented Fast Marching Method by Telea and Wijk [63] among many existing algorithms such as [7, 47]. There is a threshold t controlling how short each branch of the medial axis may be. Shown in Figure 3.7 is the computed medial axis in magenta dotted line with $t = 10$ and $t = 30$

respectively.

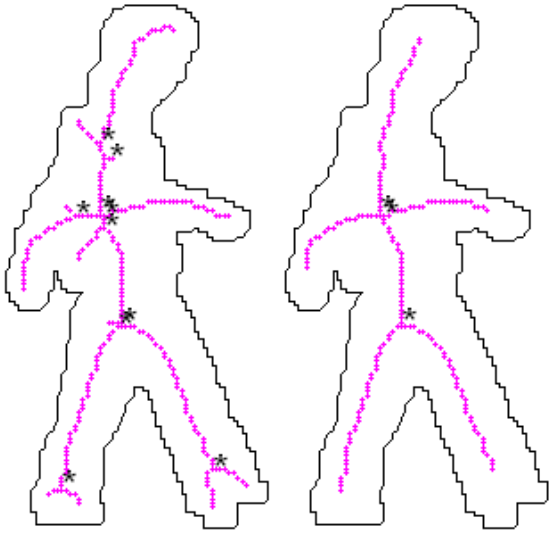

Figure 3.7: The left image shows in magenta line the medial axis obtained with $t = 10$, while the right one with $t = 30$. Each detected junction point is annotated with a black asterisk.

In order to find the junction points, I employ a lookup table (LUT) in the 3 by 3 neighborhood of every pixel on the medial axis. As each cell in the neighborhood take binary values, I have 256 total possible combinations of the 8 connected neighbors. For each combination, I determine if the center pixel is a junction point, as denoted by a black asterisk in Figure 3.7. One may notice in the figure that sometimes two junctions are too close together; in such cases, I merge those junctions that are closer than a threshold (w) and use their mean as the estimated junction. In rare cases, the parameter t is too strict to produce any junction from the medial axis; I opt to use the center of the mass as the single star, although I can also choose to reduce t until there

is at least one junction point.

3.5.2 Generating candidate extreme points

Suppose there are N stars denoted as $star_j$ $(j = 1, 2...N)$. Starting with the highest contour point, each point in the contour of length NC is sorted clockwise, and denoted as P_i $(i = 1, 2...NC)$. As in previous works [25, 78], I compute the Euclidean distance from $star_j$ to P_i as a function $dist_j(i)$. The function is then smoothed by a one-dimensional Gaussian kernel with standard deviation δ. Contour points with a local peak are chosen as candidate extreme points.

In order to find the local peaks from the smoothed distance function, I proceed with the following steps.

1. Modify the computed distance $dist_j(i)$ to $D_j(k)$ by removing repeating values so that there are no identical values adjacent to each other in $D_j(k)$. Now that the length of $D_j(k)$ should be reduced from NC to another number denoted as NK. Keep the indices Ind_k $(k = 1, 2...NK)$ updated, so that for each chunk of identical distance values, their common index is the middle of the interval. The main purpose of this step is to accommodate those contour pieces where every point has the same distance to the star.

2. For each k, check if $D_j(Ind_k) > D_j(Ind_{k-1})$ and $D_j(Ind_k) > D_j(Ind_{k+1})$. If both are satisfied, it is output as a candidate extremity. Note here $k-1$

and $k+1$ are both modulo NK arithmetic.

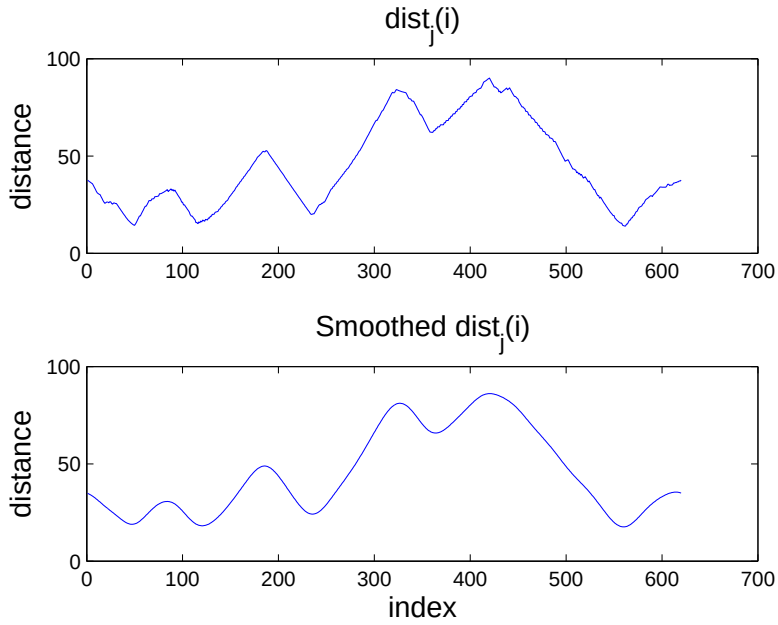

Figure 3.8: The top plot shows distance from sorted contour points to the red star, while the bottom plot shows its smoothed version.

Using the contour and junctions from Figure 3.9, Figure 3.8 shows the plots of a distance function and its smoothed version with respect to the top red star. Those with respect to the bottom green star are similar. The detected candidates are drawn as red or green crosses in Figure 3.9 accordingly.

3.5.3 Filtering

In this section, I determine if a candidate extreme point is kept, discarded or merged with a nearby candidate. I first associate each candidate

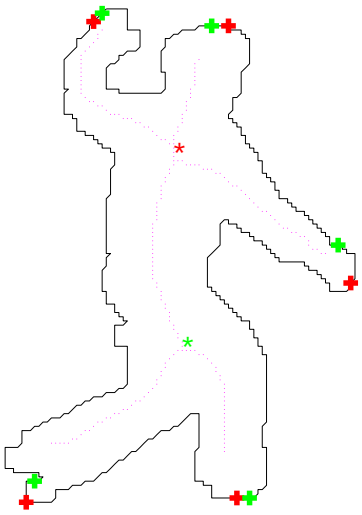

Figure 3.9: The medial axis is shown with a magenta line, junctions as asterisks, and candidate extremities as crosses in the same color as the corresponding star.

with two properties, including robustness to the smoothing parameter and visibility to the generating star.

The robustness R may be viewed as a measurement of how much a possible human limb protrudes out of the torso. As I have located all the local peaks from the distance function D_j described above, I can easily modify it to locate all the local valleys as well. Given a local peak with value $D_j(Ind_K)$ at position Ind_K, it must have an adjacent valley both on the left and on the right. Suppose the higher adjacent valley has value $D_j(Ind_{K'})$ at position $Ind_{K'}$, I define robustness R associated with the candidate extreme point P_{Ind_K} in the following equation, also illustrated in Figure 3.10.

$$R = \frac{D_j(Ind_K) - D_j(Ind_{K'})}{|Ind_K - Ind_{K'}|} \tag{3.5}$$

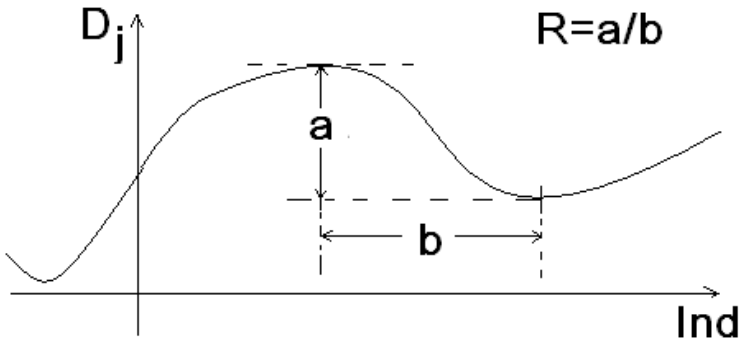

Figure 3.10: The definition of robustness of an extremity candidate.

I connect from the candidate to the star generating it, to form a line segment. The visibility V is computed as a proportion of the line segment that lies inside a silhouette. Given two points, I use the basic raster algorithm [48] on line drawing to produce the set of points between them. Then the intersection of the set with the binary human silhouette produces line points inside the silhouette.

With these properties, I proceed with the following procedure where the input is all those candidates as generated in Section 3.5.2, and the output is the detected extremities.

1. **Select candidates chosen by more than one star**. I group all those candidates by hierarchical agglomerative clustering with single linkage, so that any two candidates whose indices are closer than w are put into

one group. The means of all those clusters with more than one members form set A, and all the single member clusters form set B.

2. **Select candidates with better visibility and robustness.** Select from B all those candidates with R bigger than threshold $MaxR$ and V bigger than $MaxV$ into set A.

3. **Discard bad candidates** from B with R smaller than threshold $MinR$ or V smaller than $MinV$.

4. **Make at most 5 extremities.** I denote the number of elements of A as $|A|$. If $|A| > 5$, sort A by product of R and V, stop and output the top 5 only. If $|A| <= 5$, sort B by product of R and V. Select the top $min|B|, 5 - |A|$ candidates from B into set A, stop and output A.

3.6 Experiments on extremity detection

In order to compare the performance of detecting extremities from contours with the three kinds of star-skeleton representations, I built a data set from 50 sequences of persons climbing fences. Shown in Figure 3.11 are sample frames of a sequence. I collect 20 frames evenly distributed from each sequence to form a data set of 1000 frames. It is checked manually to test if the proposed VSS performs better than previous methods including SS and 2SS.

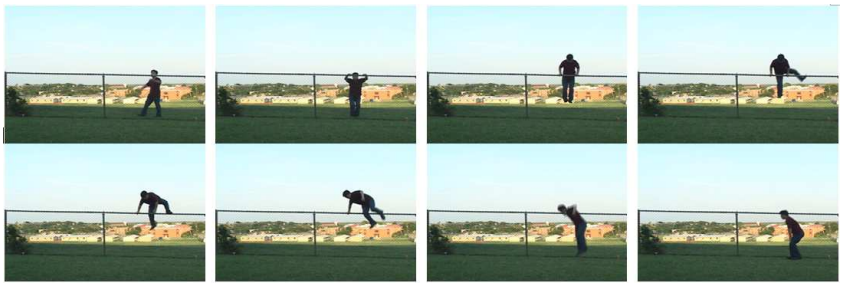

Figure 3.11: Sample frames of a fence climbing sequence.

	Ground truth	True positive	False alarm
SS	3691	3107/84.2%	779/21.1%
2SS	3691	3381/91.6%	146/4.0%
VSS w/o robustness	3691	3617/98.0%	705/19.1%
VSS w/o visibility	3691	3580/97.0%	384/10.4%
VSS	3691	3440/93.2%	98/2.7%

Table 3.1: Results from the three representations on the data set.

3.6.1 Comparison

For each frame in the data set, I have all three star skeleton representations performing detection of extremities as an approximation of head and human limbs. I manually check the results and determine the number of ground truth extreme points, true positives and false alarms. To empirically validate the relative importance of visibility and robustness criteria for human extremities, I also did experiments on the data set without the visibility or robustness criteria. Comparisons are shown in Table 1.

3.6.2 Parameter selection

There are several parameters involved in all the three star skeleton representations. The common parameter among the three is the Gaussian smoothing factor δ. There is a trade off between detecting more global or more local extreme points when selecting different scales of smoothing parameters. I used $\delta = 10$. The t threshold is set as 30, which yields a reasonable medial axis for most binary blobs. I usually get one, two or three junctions from a medial axis. I set $w = 10$ for both merging junctions and clustering candidates. The two thresholds for R in the filtering process are set as 0.6, 0.1, and the two thresholds for V are set as 0.9, 0.5. All the parameter values are chosen empirically and used throughout the experiments.

3.6.3 Discussion

From Table. 3.1, I conclude that the two-star-skeleton (2SS) can considerably improve detection accuracy from the single star skeleton. The variable star skeleton (VSS) performs best. From detection results over the 1000 frames, I have the following observations.

When there is no junction point detected, the VSS is reduced to the single-star-skeleton, except that there is the filtering process. Fortunately this does not occur often due to proper selection of t. When there is only one junction point, the VSS is more different with the single SS than without any junction point. An example is shown in Figure 3.12(a), where the VSS can successfully detect the two hands while both SS and 2SS fail. The difference

lies in that the single star is usually closer to human body joints instead of being the center of mass. Hence it has better visibility to the ground truth extreme points including the head and limbs.

The 2SS improves on the SS by fixing the highest contour point as the second star and making it automatically one of the final detections. As displayed in the left image of Figure 3.12(c), the head is detected as the highest contour point. This implicit assumption of the highest contour point being the head does not always hold. When it holds, 2SS could perform better than VSS in some cases. As shown in the right image of Figure 3.12(c), the head is missed by the VSS. This shows that increased complexity might cost us due to the difficulty in finding a set of parameters suitable all the time.

If the assumption holds, the VSS can also perform better than the 2SS, as shown in Figure 3.12(d). In this example, the left corner of a cloth is a false alarm for 2SS while it is correctly removed by VSS. When the assumption does not hold, the VSS easily wins over the 2SS. Figure 3.12(b,e) shows the hand is higher than the head, and Figure 3.12(f) shows the back is higher than the head.

3.7 Summary

In this chapter I presented how to find extremities precisely from contours with a variable-star-skeleton representation. With the concept of a star polygon, I concluded the variable-star-skeleton is a better approximation of decomposing the human contour as star polygons. Furthermore, I experimentally

validated its superiority over the previous star-skeleton and two-star-skeleton models.

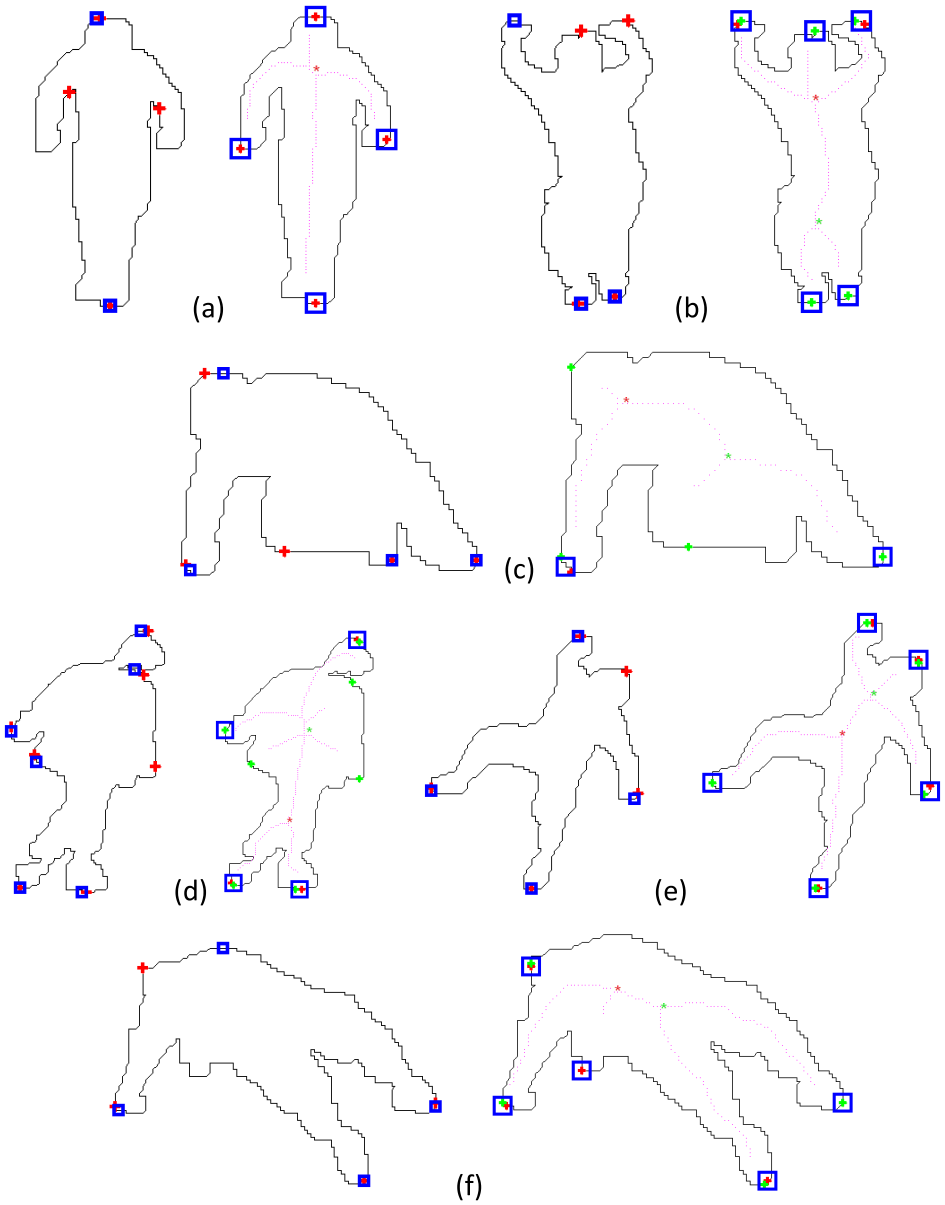

Figure 3.12: For each pair of images, the image on the left shows the result of SS in red crosses, the result of 2SS in blue squares; the image on the right shows the result of VSS in blue squares. In the images on the right, stars are shown in colors and their associated extremity candidates are shown in the same color crosses.

Chapter 4

Action Detection and Recognition with Precise Extremities

After extracting precise extremities from contours, I seek various ways to put them into usage in this chapter. As stated in Chapter 1, the main goal is to understand human behavior from videos with extremities only. When the video is long enough to consist of multiple actions, it is necessary to decompose the video into shorter pieces so that each piece is a single predefined action. For research purpose, a long video with multiple actions is often decomposed manually and the focus is then the classification of each individual piece, which is called action recognition. If the long video is not manually broken, the task of finding and labeling each consisted action is then called action detection.

In this chapter, I present first an application of using extremities to generate environment specific features for detection of fence climbing. Next, I present the idea of stable contact, which is used to abstract the image sequence into primitive motion units. Finally, I develop a general purpose human posture descriptor so that the actions to be recognized are not limited to fence climbing or those with stable contacts.

4.1 Detection of fence climbing from continuous videos

Usually fences or walls surround important infrastructures or facilities such as airports, power plants, national borders and military zones. For example, on 10/26/2006 President Bush authorized the construction of a fence along 700 miles of the U.S.-Mexico border. In order to prevent persons leaving or entering such special territories by climbing, security staff patrol around the area regularly. With such background motivation, I wish to develop an algorithm that will help monitor people climbing fences.

For now, I focus on two types of fences including flat top fences with vertical iron bars and chain link fences with slightly "barbed" wires, as shown in Figure 4.1. The main reason for choosing such fences is due to a performer's physical capability to climb. These fences are simple enough for an amateur to climb with a modest amount of effort. The camera is positioned so that the fence is in the front-back view instead of in the side view.

By continuous videos, I mean such videos are long enough to consist of multiple types of actions. Since there are mixed walking and climbing actions in a fence climbing video, I develop a Hidden Markov Model (HMM) based framework to decode the video into an action sequence.

4.1.1 Task specific features

There are three cues for identification of a human climbing fences. The first one is the coordinates of the blob centroid. A change in the y-coordinate indicates a possible climbing action. The second cue is the extreme point


Figure 4.1: An wrought iron fence with a flat top surrounding a swimming pool, and a chain link fence with slight barbed wires separating a school playground.

configuration relative to the fence, which is a coarse approximation of the position of the human hands and feet. The third cue is the height of the fence which is either known a priori or obtained by doing a simple horizontal line extraction.

Five features are computed as shown in Table 4.1 and Figure 4.2.

Table 4.1: The environment specific features for detecting fence climbing

Feature	Explanation
1	centroid x-coordinate changes?
2	centroid y-coordinate up, down, or not
3	centroid y-coordinate above fence?
4	2 or more extreme points above fence?
5	2 or less extreme points under fence?

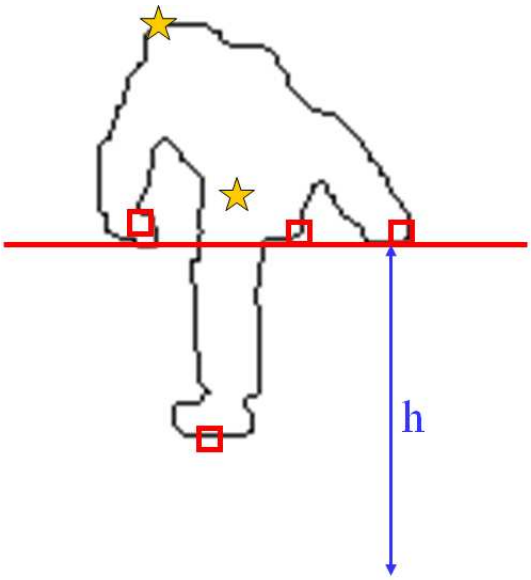

Figure 4.2: Extremities for detecting fence climbing. In the figure, the extremities are shown in red squares, and the fence is shown as the red horizontal line.

4.1.2 A block based HMM

I define that each fence climbing sequence consists of a few basic actions, including walking, climbing up, crossing over the top of the fence, and dropping down. A generalization to include more actions is straightforward. So the desired HMM has the four actions as the hidden states. The Viterbi algorithm of the HMM decoding problem is employed to infer the action sequences.

After training a discrete HMM for each of the four basic actions, there are four sets of HMM parameters $\{P_i, T_i, O_i\}$ where P_i is the prior state distribution vector, T_i is the state transition matrix, and O_i is the observation distribution matrix, $i \in \{walk, up, cross, down\}$. These parameters are con-

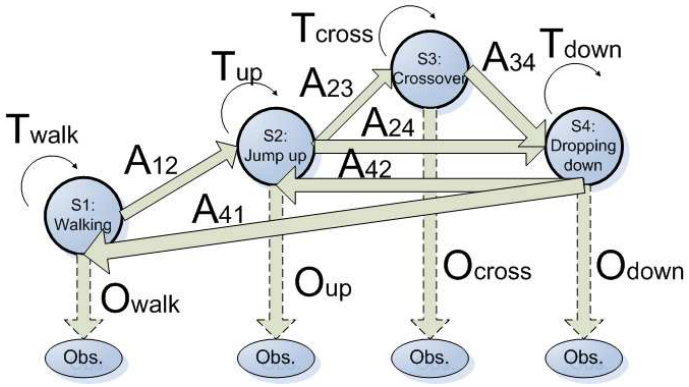

Figure 4.3: The block based HMM assembled from the four individual HMMs

catenated to form a block based HMM, as shown in the following equations and Fig. 4.3.

$$Prior = \left[\begin{array}{cccc} P_{walk} & 0 & 0 & 0 \end{array} \right] \tag{4.1}$$

$$Trans = \left[\begin{array}{cccc} a_1 T_{walk} & A_{12} & 0 & 0 \\ 0 & a_2 T_{up} & A_{23} & A_{24} \\ 0 & 0 & a_3 T_{cross} & A_{34} \\ A_{41} & A_{42} & 0 & a_4 T_{down} \end{array} \right] \tag{4.2}$$

$$Obs = \left[\begin{array}{c} O_{walk} \\ O_{up} \\ O_{cross} \\ O_{down} \end{array} \right] \tag{4.3}$$

Note that the zeros mean appropriate size matrices with all zero values. Comparing the three equations above with Figure 4.3, there come the following

interpretations. Equation 4.1 means that one always starts the sequence by walking. Equation 4.2 comes into being as usually it is assumed that only certain transitions between actions are possible, where A blocks are the random matrices with fixed weight and have the same meaning as in Figure 4.3, and a values mean weights to sum every row up to 1. Equation 4.3 shows that each block state may observe all the observation symbols, hence the observation matrices are concatenated by rows.

4.1.3 Decoding HMM

As illustrated in Fig. 4.4, to detect a climbing action, I first decode the observation sequences into hidden state sequences, and then generalize them into block sequences since each block of hidden states corresponds to one of the basic actions. Qualitatively, climbing is determined if there is a consecutive triple {up, cross, down} where each lasts for a long enough frame period.

Furthermore, I implement a quantitative measurement to judge if the detected action sequence is the same as the ground truth. It consists of two steps. In the first step, I remove noise and merge adjacent labels if necessary. In the second step, I determine if the detected action sequence is the same as the ground truth by judging if they have the exact same labels and similar duration for each label.

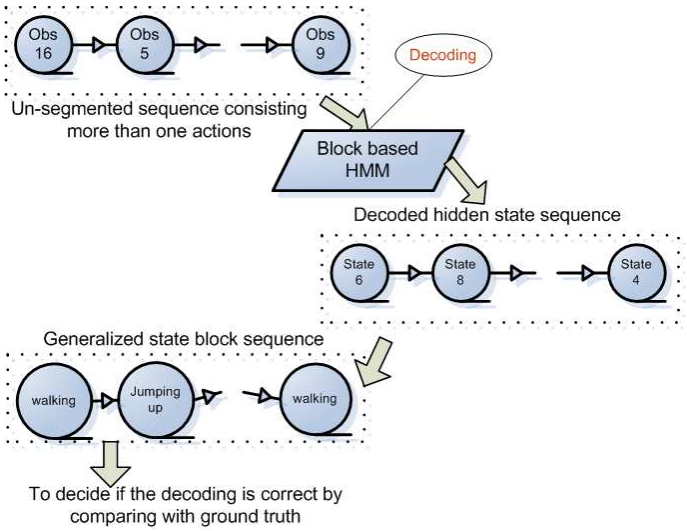

Figure 4.4: Decoding on the block based HMM to infer the action sequence

4.1.4 Experiments

I collected fence-climbing videos from six men and a woman in two scenarios, including a swimming pool surrounded by a flat-top fence and a school playground separated by a chain-link fence with "barbed" wires, as shown in Fig. 4.5. I dumped videos into image sequences, where each frame is a 24 bit RGB bitmap file of size 360 by 240 pixels and the FPS rate is 30. Overall, there are 50 sequences consisting of mixed actions of walking and fence-climbing.

I manually segmented temporally the mixed action sequences of walking and climbing (split into three actions) according to a manually determined ground truth. I tested the classification accuracy of the four trained individual

Figure 4.5: Sample frames of climbing two fences.

Table 4.2: The accuracy of four individual HMMs under two different star skeleton representations.

HMM	my two-star-skeleton	star-skeleton
walking	18/18	18/18
jumping up	7/10	0/10
crossing over	10/10	7/10
dropping down	10/10	8/10

HMMs each with 3 hidden states, using my two-star skeleton representation. The results are shown in the middle column of Table 4.2.

On the same data, I also tested classification accuracy with the representation of Fujiyoshi and Lipton [25]. The results are shown in the right column of Table 4.2. It is clear that their representation cannot recognize accurately any jumping up action (the first component action of climbing). The main reason is that when the human jumps up in the front/back view, the shoulders or elbows are easily but incorrectly detected as desired extreme points in the single-star-skeleton representation. However, this problem is greatly reduced with my two-star-skeleton representation, with seven out of 10 correctly classified.

I checked the decoding accuracy of the proposed block based HMM.

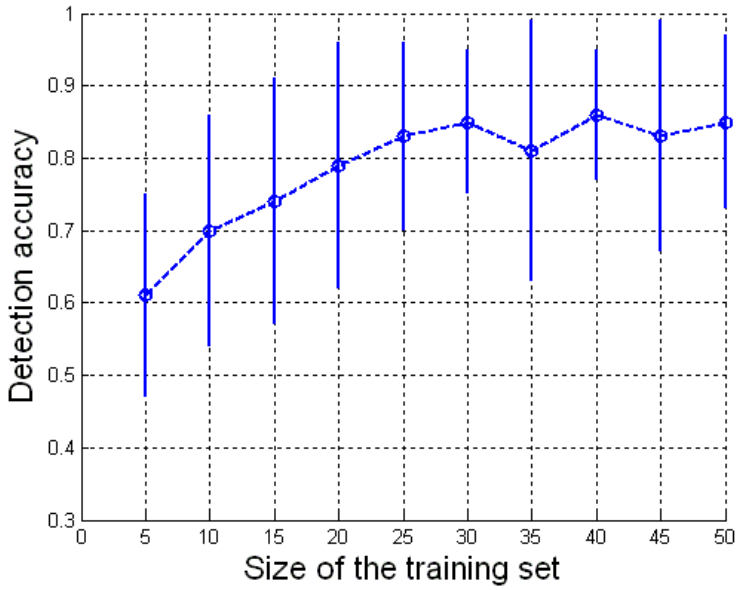

Figure 4.6: Continuous recognition accuracy of the frame level analysis.

After assembling the parameters of the four HMMs into a block based HMM, I used the proposed quantitative measurement to judge if a sequence is correctly decoded. The experiment was done on different sizes of a training set of the 50 mixed action sequences. I increased the training size from 5 to 50, and used the whole set as the testing set. For each training size, I randomly chose training sequences and computed the testing accuracy. Random selection for each training size was repeated 50 times, and the mean and standard deviation were computed, as shown in Fig. 4.6.

I further validated the decoding approach with those sequences consisting of no climbing actions. I fed 25 walking sequences into the decoding implementation with the two-star skeleton and got zero false alarms of a fence

climbing action.

4.2 Detection of stable contacts for motion analysis

The concept of stable contact comes from a close observation of human actions. In most time instants, there has to be at least one part of the body in stable contact with the surrounding environment; otherwise the human body must be in rare moving conditions such as falling or swimming. The number and positions of stable contacts themselves give a lot of information about human actions. For example, when climbing there are always stable contacts between the human body and the environment, involving either the hands or feet. When the number of stable contacts does not change, one may consider the period to be in a relatively static state. I call this period a primitive interval and further define a primitive motion unit (PMU) as what happens in that period including both stationary and motion information. By this definition, PMUs are formed to break a long continuous action or activity into multiple parts. In other words, a frame sequence is abstracted as a PMU sequence, which provides a new perspective for the detection and recognition phase.

For each type of action to be recognized, a discrete HMM is trained with associated PMU sequences. In order to continuously recognize activities, I search over the time axis stepped by PMUs, by varying the duration of a candidate PMU sequence and judging how well it fits among all the trained models. The overall architecture is shown in Fig. 4.7. One block shows the processing by frames, and the other block shows the analysis by PMUs. There

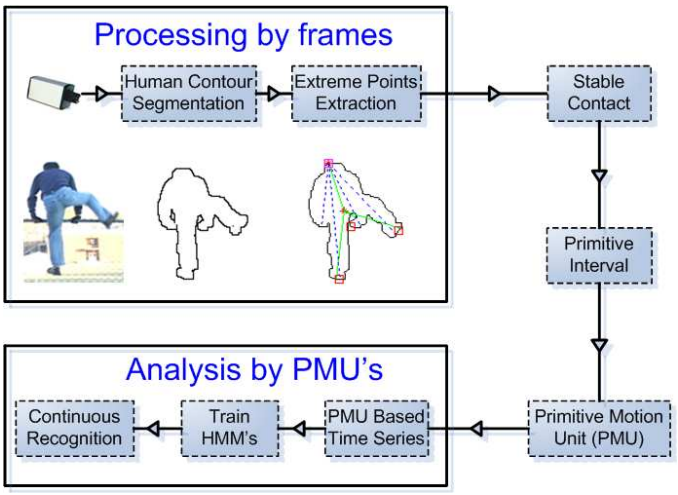

Figure 4.7: The architecture of the second approach.

are three steps necessary to abstract frames into PMUs, including the detection of stable contacts, the construction of primitive intervals, and the extraction of PMU attributes.

4.2.1 Detection of stable contacts

Intuitively a stable contact is any body part or region that is in contact with the environment for a period longer than a minimum threshold τ. I call this period the duration with respect to the stable contact. By broad definition the stable contact is a surface region where the body part is in contact with a large area of the environment. However, human body parts always come into contact in three-dimensional space from a point to a gradually increasing region until it is stable, or in the reverse situation when the

Input

List A

Frame	Coordinates	
	x	y
5	20	-220
5	100	-50
6	120	-100
6	21	-223
7	140	-60
7	18	-219
8	22	-221
9	19	-222

Intermediate result

List B

Stable Contact Candidate ID	Coordinates		Frame
	x	y	
1	20	-220	5
1	21	-223	6
1	18	-219	7
1	22	-221	8
1	19	-222	9
2	100	-50	5
3	120	-100	6
4	140	-60	7

Output

List C

Stable Contact ID	Mean Coordinates		Frame Interval	
	x	y	start	end
1	20	-221	5	9

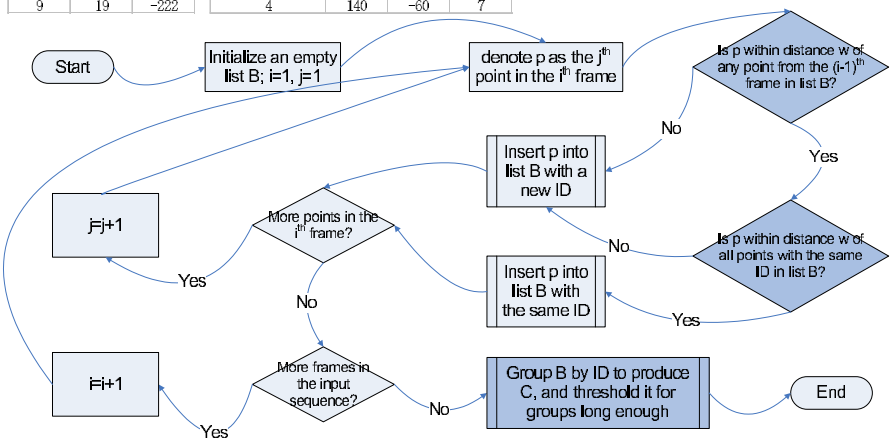

Figure 4.8: My implementation of detecting stable contacts.

stable contact is disappearing, the contact region gradually decreases into a single point and then null. So the whole stable contact surface is abstracted as a single point, which should be the 'starting point' when the stable contact surface appears, and the 'ending point' when it disappears. This representation also fits the common practice of working on image sequences consisting of only 2-dimensional information. How to detect stable contacts and their associated durations in image sequences depends highly on the human body representation. In my case, I have extracted sets of extreme points from human contours.

Since extreme points are usually human hands and feet, stable contacts

are formed from those stationary extreme points. Hence stable contacts are detected by checking if any extreme point remains in the same place for a long enough period τ. A maximum deviation tolerance parameter w is also applied in order to tolerate the detection inaccuracy of extreme points. In order to find such extreme points, consider the following two equations, as explained below.

$$j - i + 1 \geq \tau \tag{4.4}$$

$$\forall m, n (i \leq m, n \leq j) \quad |p_{m,x_m} - p_{n,x_n}| \leq w \tag{4.5}$$

Given a length l sequence of extreme point sets, denoted as $\langle \mathcal{P}_1, \ldots, \mathcal{P}_l \rangle$, where each point set $\mathcal{P}_r (1 \leq r \leq l)$ consists of all extreme points $\{p_{r,1}, p_{r,2}, \ldots, p_{r,|\mathcal{P}_r|}\}$ in the r^{th} frame. Note that $|\mathcal{P}_r|$ represents the cardinality of the set $\mathcal{P}_r$. The goal is to find any consecutive sub-sequence of extreme points $\langle p_{i,x_i}, \ldots, p_{k,x_k}, \ldots, p_{j,x_j} \rangle$, where $\forall k,\ p_{k,x_k} \in \mathcal{P}_k\ (1 \leq i \leq k \leq j \leq l)$, such that the two equations (4.4) and (4.5) hold. Then I output the mean coordinate of each sub-sequence of extreme points as the stable contact position and the associated frame interval as the duration of the stable contact.

Fig. 4.8 shows my implementation of the stable contact detection algorithm, with simplified illustrative data from a real sequence. Note that the shaded diamond box implies usage of the parameter w, while the shaded rectangle box implies usage of the parameter τ.

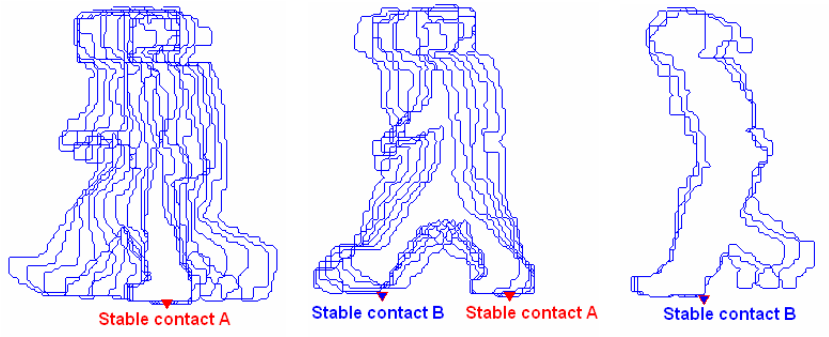

Figure 4.9: Detected stable contacts from a sequence of 21 frames shown in three primitive intervals respectively.

Fig. 4.9 shows examples of two detected stable contacts for a total period of 21 frames in a walking sequence. For each image in the figure, all the associated frames are stacked to demonstrate that a stable contact is in fact the extreme point that stays in contact with the environment for a reasonable length of time.

Fig. 4.10 shows an example of all detected stable contacts on a real sequence. Each triangle represents a stable contact point, and all detected stable contacts are plotted on the last frame of the sequence, regardless of their associated durations. The sequence starts with a person walking, followed by his climbing over the fence. These positions of the stable contacts are formed by either his hands or feet, displaying approximately the activity trajectories.

Although I experiment with limited kinds of activities, this stable contact concept is designed as a generic abstraction tool for human actions. The only limitation of the concept may be that it needs as many visible stable

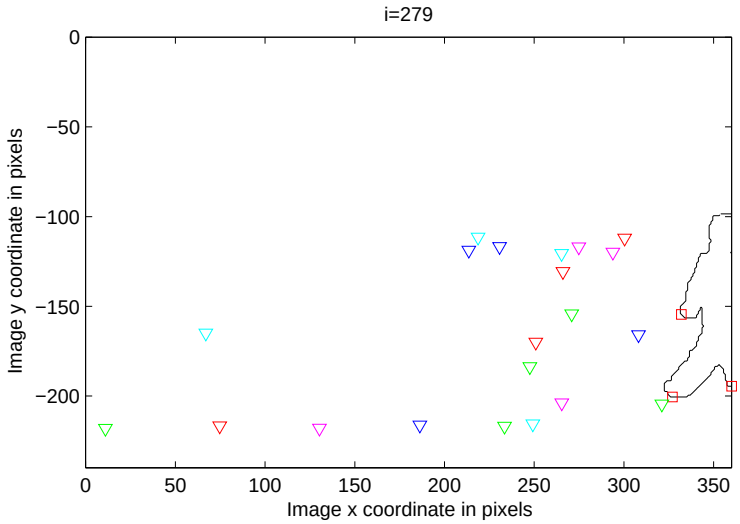

Figure 4.10: An example of detected stable contacts shown as triangles in a sequence of walking and fence-climbing. Best viewed in color.

contacts as possible in order to achieve the best results.

For the stable contact, its detection is not as sensitive as that for the extreme points. The reason is that I am looking for the consecutive extreme points appearing in nearby positions. In my practice, I ensure the recall of all ground truth extreme points while tolerating false alarms. As long as those false alarms are not consistent in consecutive frames, they will not get detected as stable contacts. Stable contact detection will only yield false alarms, if the noise produces persistent extreme points in the same neighborhood.

This parameter w determines how much a group of consecutive extreme points can deviate from each other before they are not regarded as a stable

contact. My strategy in selecting it is to allow a relatively loose standard, which means a bigger w. This will allow more tolerance of errors produced from getting the extreme points.

4.2.2 Primitive Intervals

In general, there are a number of stable contacts associated with each frame, and their durations overlap. Hence temporal segmentation is achieved through changes in the number of stable contacts (NSC), instead of using the durations of stable contacts. More precisely, I segment a new block of consecutive frames from the image sequence whenever there is a change in the NSC. The frame block covers a period when there is a consistent number of stable contacts. I call such a period a primitive interval to distinguish it from the duration of a stable contact. Therefore a primitive interval may be associated with an arbitrary number of stable contacts, and all durations of those stable contacts may intersect to produce the primitive interval, as shown in Fig. 4.11.

Fig. 4.11 uses the same data as in Fig. 4.9. The video has 21 consecutive frames of a person walking. Since there are two stable contacts, of which the first is from frame 1 to 18 and the second is from frame 12 to 21, their durations overlap during the period from frame 12 to 18. Hence these two stable contacts form three primitive intervals, covering frames from 1 to 11, from 12 to 18, and from 19 to 21 respectively, also shown in Fig. 4.9.

Fig. 4.12 shows an example of temporally segmenting the same image

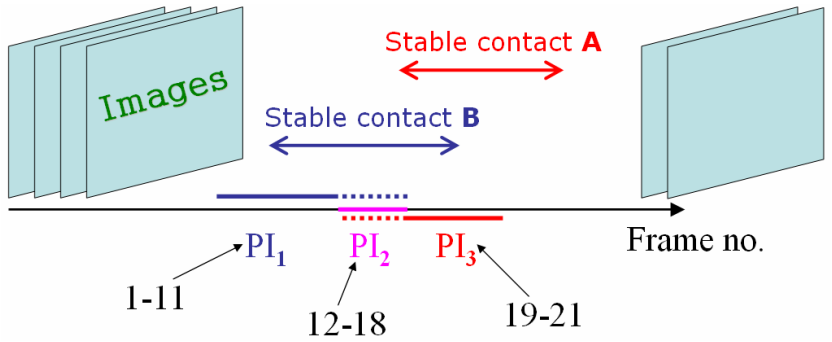

Figure 4.11: An example of three primitive intervals from two stable contacts.

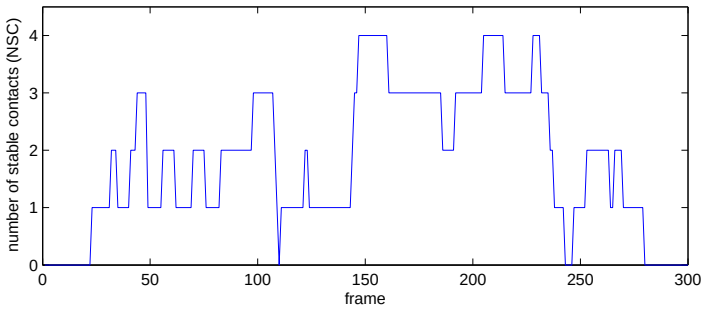

Figure 4.12: An example of temporal segmentation by change in the number of stable contacts.

sequence as used in Fig. 4.10, including walking in side view and climbing a fence in the front-back view. For example, soon after frame 100, there is a jumping up action when the NSC is detected as zero since hands are occluded. Just before frame 250, the person jumps down, causing the NSC zero again. Before or after the climbing action, there is only walking, hence the NSC fluctuates between one and two, with an exception when an immobile hand is detected as a false positive stable contact. Every primitive interval has a consistent NSC, which will be my time unit for analysis.

4.2.3 Primitive Motion Units

Over each primitive interval, I define a PMU to represent the movement during the period. I use three attributes to summarize low level information about the movement, as shown in Table 4.3. The first attribute is the NSC, with category values *0*, *1*, *2*, and *more than 2*. The second attribute involves the duration of the primitive interval, with category values *short*, *medium* and *long*. The third attribute is related to the approximate motion direction, with category values *left*, *right*, *up*, *down* or *stationary*. By using the joint attributes, I get a total of $4 \times 3 \times 5 = 60$ different types of PMUs. Note that not all types appear in the experimental sequences.

Table 4.3: Attributes for a PMU.

Attribute	Description
Number of stable contacts	0,1,2,more than 2
Duration of primitive interval	short, medium, long
Direction of blob centroid	left, right, up, down, stationary

The number of stable contacts (NSC) itself is a perfect cue for the movement categorization. For example, the NSC normally alternates between one and two in walking, but alternates between zero and one during running. I define three category values on the second attribute of a PMU. A short duration normally means that the primitive interval is a transition period for the action. A medium duration tends to be the actual phase of performing the action. A long duration implies there is very slow movement or no movement at all. I choose thresholds from trials to determine the category values. I obtain the value of the third attribute using the velocity of the blob centroid. Briefly

speaking, I take the average velocity of the blob centroid in the primitive interval and categorize it with appropriate thresholds.

4.2.4 Searching with trained HMMs

I evaluate the three attributes listed in Table 4.3, hence forming a feature vector for each PMU. I then build a code book to convert each discrete feature vector into a symbol. The original image sequence is now summarized by a time series of PMUs. A typical PMU sequence may contain any number of PMUs, depending on the length of the original frame sequence. The goal is to model an action with a discrete HMM, and the observation of the HMM is the symbol of PMU attributes. I train a discrete HMM for each action to be classified, including human walking and climbing fences. The training data is manually segmented from continuous activity sequences so that each piece of training data consists of only one action. Note that here I regard the three actions (climbing up, crossing over and dropping down) considered in the first approach together as a single climbing action.

I recognize continuous activities from acquired image sequences by searching over the time axis. The approach is motivated partially by works from Davis and Bobick[17] and Min and Kasturi.[44] Here, I work on PMU sequences instead of frame sequences.

The searching task is as follows. Given two trained HMMs representing walking and climbing, and a length l PMU sequence $\langle PMU_1, PMU_2, \ldots, PMU_l \rangle$, I have to form a temporal partition P which breaks the PMU sequence into r

parts with the i^{th} part p_i having l_i consecutive PMUs. I also need to classify each part as either walking or climbing. There are $\sum_{r=1}^{l} \binom{r-1}{l-1} \times 2^r$ possible combinations of partitions and classifications.

The searching task is formulated as an optimization problem by defining an objective function $\Psi(P, C)$, with the partition $P = \langle p_1, \ldots p_r \rangle$ and classification $C = \langle c_{p_1}, \ldots c_{p_r} \rangle$ as variables. Having trained c HMMs, I can derive the likelihood of an action c_{p_i} on any part p_i of a partition, which can be denoted as $L(p_i, c_{p_i})(1 \leq i \leq r)$. Note that I use c_{p_i} to imply that this action class is for the part p_i in the partition, and the actual action class is from the two trained HMM classes $\{C_{walking}, C_{climbing}\}$. Hence I re-write the likelihood as $L(p_i, C_j)(1 \leq i \leq r, j \in \{walking, climbing\})$. The objective function is formally defined as in equation 4.6, for the case of two classes only, where C_j means the other class of the two.

$$\Psi(P, C) = \sum_{i=1}^{r} L(p_i, C_{p_i}) - L(p_i, C_j) \tag{4.6}$$

My goal is to maximize the objective function. I choose a greedy style optimization strategy. As shown in equation 4.7, I break the objective function into the sum of local objective functions as defined in equation 4.8. I get a solution by maximizing the local objective functions one by one.

$$\Psi(P, C) = \sum_{i=1}^{r} \psi(p_i) \tag{4.7}$$

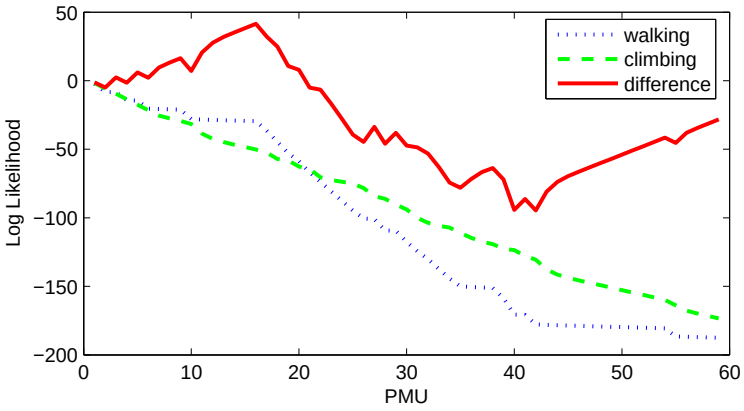

Figure 4.13: An example of searching for the maximum relative likelihood over the time axis by PMUs.

$$\psi(p_i) = L(p_i, C_{p_i}) - L(p_i, C_j) \tag{4.8}$$

I start from the first ($i = 1$) PMU, and search for the last PMU of part p_1 in the partition with a minimum duration m and a maximum duration M. Every search will result in a fixed part duration and classification of that part into one of the two activities. The next search for part p_i ($i \geq 2$) starts from the PMU immediately after the last PMU of part p_{i-1}. The number of parts r increases by one after every search until the search stops when there are no more PMUs left in the sequence.

An example from real sequences is given in Figure 4.13. The dotted line represents the log likelihood of a walking action covering the duration at the PMU level. The dashed line represents the log likelihood of a climbing action. The solid line represents the difference between the two classes. From the

graph, I can see the difference first reaches a local maximum at the 16^{th} PMU. Hence I determine that the duration from the 1^{st} to the 16^{th} PMU represents a walking action. The next search starts from the 17^{th} PMU.

4.2.5 Experiments

Using the exact same data set as used in the first application, I check the accuracy of the continuous recognition by searching at the PMU level. Similar to the first approach, I stepped the training set from 5 to 50 and used the whole set to test. The mean and standard deviation of 50 runs at each partition are shown as the red line in Fig. 4.14, where results from Fig. 4.6 are also shown as blue lines shifted one unit to the left for comparison.

From Fig. 4.14, I can see that the frame level analysis (the first approach) performs worse than the PMU level analysis (the second approach), when the size of the training set is not big enough. The reason is that the block-based HMM needs to be accurate enough for the decoding to work properly, while searching in the PMU level can be more robust since it involves only the relative difference between individual HMMs instead of depending on a global HMM.

Both approaches achieve about 80 percent accuracy when the size of the training set is big enough. When the training set becomes larger, the PMU level analysis has a smaller standard deviation in accuracy than the frame level analysis. There are two explanations for this increased stability. On one hand, when I extract features based on PMUs instead of on frames, the accuracy is

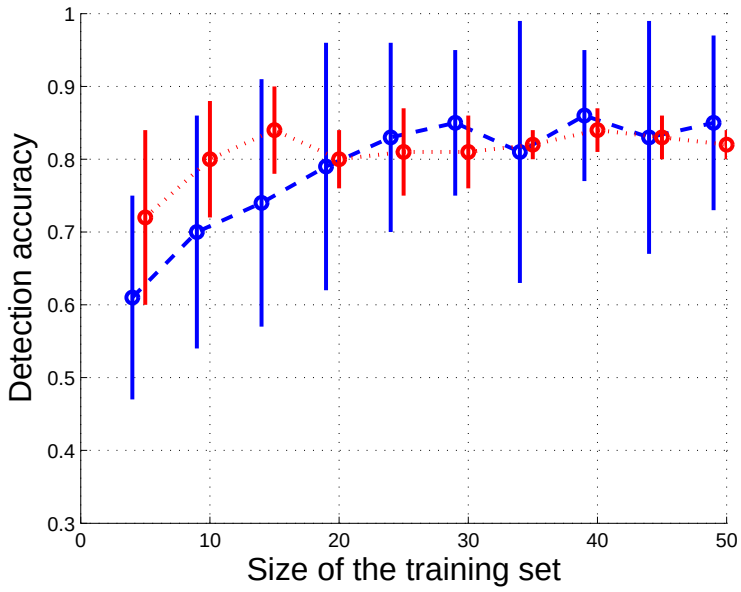

Figure 4.14: Comparing the two approaches, with accuracy of the second approach shown in red line and that of the first approach shown in blue line shifted one unit to the left for comparison.

improved since features extracted across a few frames are less noisy than those extracted from a single frame. On the other hand, in PMU level, the number of possible partitions of a sequence is greatly reduced since primitive intervals have already abstracted the frame sequence, hence improving the accuracy.

4.2.6 Comparison between the two approaches

Overall I conclude from Fig. 4.14 that there is no significant difference in terms of performance with the two approaches. However, the two approaches differ in their applicability to practical problems.

The first approach depends on the availability of fence height to build feature vectors from each frame. From Table 4.1, I can see that feature 3, 4 and 5 cannot be obtained without the height of the fence. Moreover, providing the height of the fence as a single value to the approach implicitly assumes that the fence forms a horizontal line in the field of view of the camera. This may not always be true. The second approach does not require any such context.

In another aspect, the first approach trains a global HMM to model switching between climbing and walking, while the second approach trains two HMMs, one for walking and one for climbing. Hence if the switching pattern between walking and climbing is relatively fixed (for example, people always walk in, climb, and walk away), it may be enough to train only once in the first approach. But if the switching pattern changes often (for example, suddenly all people walk in and out without climbing), it requires frequent updates for the global model to accurately decode an observation sequence.

In terms of algorithm complexity, the second approach involves more steps and is much more complicated. So when a given scenario is simple enough (with the height of the fence is available and there is not much fluctuation in the pattern of walking and climbing), one may still prefer the first approach for its simplicity.

4.3 General purpose posture descriptor

In the last two sections, I introduced how to utilize extremities to produce features for detecting fence climbing and how to detect the stable contacts

from extremities. Both approaches are focusing on detecting certain actions from continuous videos. In this section I will focus on classifying temporally segmented videos into predefined classes. I propose a circular histogram of extremities to abstract a frame into a 12-element feature vector.

4.3.1 Histogram of extremities

With detected body extremities, one can recognize a variety of common human actions by using the discrete HMM technique [55]. As each action is represented with an image sequence or video, the key procedure is to convert each frame to an observation symbol so that each action may be represented by an observation sequence. Note that I use only a set of human extremities for each frame. Motivated by the shape context descriptor proposed by Belongie et al. [5], I use a simple circular histogram to build a feature vector for each frame. As shown in Figure 4.15, we find the relative coordinates of each extremity with respect to the center of mass of the human silhouette. The entire plane is evenly divided into N ($N = 12$) sectors, and the histogram is a N-element vector with each element indicating if there is an extremity in the sector.

In order to reduce the number of observation symbols, vector quantization is commonly employed to cluster the feature vectors. The cluster label of each feature vector acts as the observation symbol for HMM usage. However, it is not always necessary if there are a limited number of unique features. In my experiments, I simply use the index of each feature vector in the unique

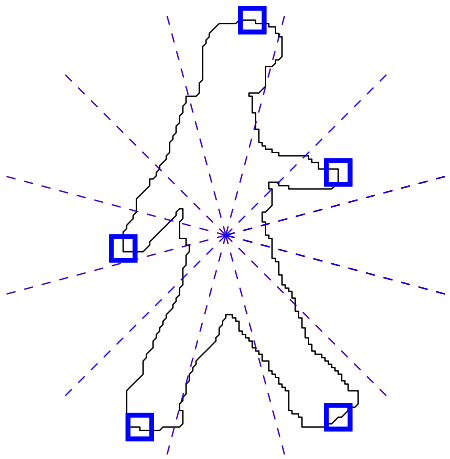

Figure 4.15: A simple histogram to extract feature vectors from frames.

feature vector set as the observation symbol.

4.3.2 Experiments

I tested action recognition on four different data sets with the same strategy. For each data set, I build a feature vector from each frame with the simple histogram. The procedure itself can be viewed as vector quantization as well, since the number of unique feature vectors is much less. I adopted the leaving-one-out cross-validation strategy in our HMM classification framework. In each iteration, we just pick one test sequence in turn, and use all the rest as a training set to train for each class a HMM with 2 hidden states. Finally, each sequence is used exactly once as a test sequence, and the confusion matrix is produced.

The experiments on the four data set is listed in the following sub-

sections, according to the relative size of the human figures in videos. The summarized results are displayed in Table 4.4 to show the classification accuracy on the four data sets. It gives an idea about how good the histogram of extremity descriptor is with respect to the size of human figures.

Data	Figure size	# of class	# of seq.	Accuracy
Fence climbing	130 pixels	2	140	97.9%
Weizmann [6]	70 pixels	10	93	93.6%
Tower	30-40 pixels	5	60	86.7%
Soccer [19]	40 pixels	7	66	63.6%

Table 4.4: Performance of the histogram of extremity descriptor on different data sets.

4.3.2.1 On the fence climbing data set

There are a total of 12652 frames in the 50 *climbing* and 90 *walking* sequences. After feature extraction, there are 685 unique feature vectors. After 140 iterations, the confusion matrix produced only 3 misclassifications, e.g. the overall accuracy is 97.9%. I found all three misclassifications are due to their very short durations, including 16, 12, and 19 frames. Since the frame rate is 30 (fps), these short sequences do not even show a full step, as validated by manual inspection.

As a baseline comparison, my previous work [78] reported 3 misclassifications on 18 walking and 10 climbing test sequences, which is approximately 5 times our error rate. In that work, I used 2SS to find extremities and built features such as how many extremities are above or under the fence, in ad-

dition to motion features including the direction of the centroid velocity. In comparison, the histogram approach involves no explicit motion features.

4.3.2.2 On the Weizmann data set

Figure 4.16: Sample images of the 10 actions, including *bend, jack, jump, pjump, run, side, skip, walk, wave1 and wave2.*

In this data set [6], there are 93 sequences of 9 persons performing 10 different actions, as shown in Figure 4.16. Using the provided human silhouette, I extracted the human extremities for all 5687 frames. To get a flavor of the accuracy of the proposed VSS on this particular data set, I manually checked all 701 frames from 10 sequences performed by one person (*Daria*). The VSS detected 1889 (96.1%) out of 1966 ground truth extremities, while making only 18 false alarms.

There are only 179 unique feature vectors. After 93 iterations, the confusion matrix is produced as in Figure 4.17. There are 2 misclassifications

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	9									
jack		9								
jump			7	1			1			
pjump			1	8						
run					10					
side						9				
skip					1		9			
walk						1		9		
wave1									8	1
wave2										9

Figure 4.17: The confusion matrix of action recognition on the Weizmann data set.

between *jump* and *pjump*, as they are essentially the same action taken in different views. Among these actions, *jump* and *pjump* are essentially the same action which is taken from different views; *walk* and *run* are hard to differentiate without considering the speed factor.

The overall accuracy is 93.6%, as compared in Table 4.5. Although we didn't achieve the perfect recognition rate, our methodology is the fastest in the sense that our VSS to detect extremities is linear in the number of contour points; and our feature extraction procedure is very simple.

Note that the first four papers [3, 6, 32, 46] worked on the old version of the data set without the *skip* action. In Blank et al. [6], each sequence is further split into cubes and classification is done per cube. Their algorithm has linear time complexity in the number of space-time points, e.g. the total number of

Method	Accuracy	Data
Blank et al. [6]	99%	81 seq. no *skip*, chopped as cubes
Ali et al. [3]	92.6%	81 seq. no *skip*
Niebles and Fei-fei [46]	72.8%	83 seq. no *skip*
Jhuang et al. [32]	98.8%	81 seq. no *skip*
Fathi and Mori [21]	100%	93 seq. 10 actions
Wang and Suter [70]	100%	93 seq. 10 actions
Ours	93.6%	93 seq. 10 actions

Table 4.5: Comparison of different methods on the Weizmann data set.

pixels inside all silhouettes. Ali et al. [3] assume the six body joints including head, belly, hands and feet are available for further action recognition. In their experiment, they used the end points of a medial axis as an approximation of body joints, which is very close to our idea of using junctions as stars. Both Niebles and Fei-fei [46] and Jhuang et al. [32] have the advantage of avoiding the difficult segmentation step, but their time complexity is at least linear in the number of all pixels in a video. In Fathi and Mori [21], a computation of the optical flow is necessary for each frame as the first step. Considering they have tracked the human figure as a rectangle, the time complexity is linear in the number of all pixels in the tracked region. In Wang and Suter [70], the module of dimension reduction by Locality Preserving Projection (LLP) has square time complexity in the number of frames, as the construction of the adjacency matrix need to find K nearest neighbors for each frame.

4.3.2.3 On the tower data set

In comparison with the Weizmann data set, the tower data are in lower resolution. For this data set, the camera is mounted on a tower around 70 meters tall and actors perform in a garden under the tower. Human figures in the frames are only around 40 pixels tall. There are 6 actors each performing 5 actions twice, including *carrying, running, jumping, waving one hand, and waving both hands*. There are 60 sequences and 2406 frames in total. We show five sample frames of each action in Figure 4.18.

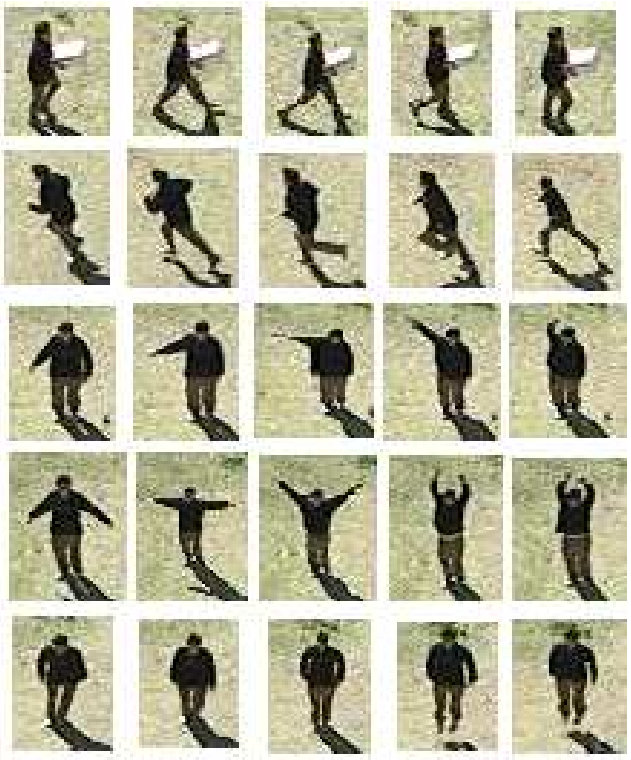

Figure 4.18: Five sample frames of each action in the tower data set.

As the resolution is low, the segmentation is not as good as in the first

two data sets. The overall classification accuracy is 86.7%.

4.3.2.4 On the soccer data set

The soccer data set provided by Efros et al. [19] proves very difficult for action recognition. Its difficulty is due to two aspects. First, the action classes are very similar to each other. Among the seven classes, running left and running left at 45 degrees look very similar to each other, and so do running right and running right at 45 degrees. Walking left and running left are almost the same, and so are walking right and running right. The only distinct class is walking/running in/out. These similar classes are even more similar when there is only relative motions between body parts available. Second, the video is in low resolution and quite noisy. The sample frames are shown in Figure 4.19.

Figure 4.19: One sample frame of each action in the soccer data set. From left to right, the seven actions are: walking/running in/out, running left, running left at 45 degrees, running right, running right at 45 degrees, walking left, walking right.

Amazingly, the histogram descriptor performed well and obtained 63.6% classification accuracy, in comparison with the 67% in Efros et al. [19]. Considering that they compute optical flows for each pair of images, our method is significantly faster.

4.4 Summary

In this chapter I utilized extremities to detect climbing actions from continuous videos. It is done first by generating environment specific features and decoding a trained block based HMM to infer the action sequence. It is then accomplished by another approach, where I detect stable contacts to abstract frames into primitive motion units. The searching strategy is used to find the duration and label for each piece of action along the time axis. In other words, this second application of extremities can also be regarded as a kind of temporal segmentation.

Furthermore, I defined a posture descriptor with the histogram of extremities for general action recognition. I reported the experimental results on four data sets, where the size of human figures decreases from about 130 pixels to only 30 pixels. The results proved the effectiveness of precise extremities as long as the segmentation is not an big issue.

Chapter 5

Probable Extremities

In previous chapters, I introduced how to extract precise extremities from contours and utilize such extremities for action detection and recognition. I conducted extensive experiments to prove the effectiveness of extremities as a compact representation of human postures. Although the result is excellent, there is an inherent limitation to the work, e.g. the contour has to be provided.

In order to relax the limitation, I propose to model human extremities as image patches instead of points on the contour. In Section 5.1, I explain the advantage of modeling extremities as patches. With extremities as patches, I elaborate in Section 5.2 how to represent them with Histogram of Oriented Gradients. In Section 5.3, I describe how to extract a set of training examples for extremities, train a classifier for patches and detect the extremities from an image. In this way, each frame is represented by a probability map, where each pixel is associated with the probability of the patch centering around it as an extremity type. In Section 5.4, I describe the integral histograms for fast feature computation without losing accuracy. In Section 5.5, I propose the histogram of probable extremities descriptor to summarize the probability map as another compact representation of human postures, in comparison to

the one introduced in Section 4.3.1. In Section 5.6, I explain the pipeline for action recognition with the probable extremities. In Section 5.7, the new posture descriptor is applied into several data sets, to validate its superiority over the old one.

5.1 Advantage of extremities as patches

Human extremities refer to heads, hands, and feet in this dissertation. The simplest way to model human extremities is to define them as the points on the human body contour that produce the maximum distance from the corresponding star. Such an approach is described in details in Chapter 3.

However, extremities are not really single points in practice. Instead, human extremities often cover an image region and are more appropriately modeled as image patches. There are the following advantages for the image patch model:

1. **To overcome the segmentation difficulty:** Although there are different methods to compute human contours from images or videos, the contour segmentation remains a challenging problem in practice. Since segmentation is a fundamental difficulty in computer vision, there is a tendency among researchers to bypass the segmentation step. For my extremity detection and applications, it is more practical to work directly on images or videos, instead of on segmented contours. Such a strategy can also be explained from the perspective of information theory. When

there is less step in a system, there is less information loss. The essence of bypassing the contour segmentation step is not to ignore the detection of extremities, but to minimize the detection steps.

2. **To increase the detection robustness:** When the extremities are modeled as image patches, there are many more representations available for extremities besides their locations, since a region can easily provide more powerful and rich descriptors than a single point. With better representations, the detection is more accurate.

5.2 Representing patches

Given an image patch, there are various ways to represent it as a feature vector. The simplest way is to concatenate all the pixel intensities. For example, Turk and Pentland [65] proposed the Eigenfaces approach for face recognition, with the face images represented by concatenations of all pixels in the images. What exact representation one should use for patches depends on the specific types of patches.

Among the three types of extremities, heads are in general upright, feet are mostly upright, while hands may have various orientations. The HOG descriptor can capture edge orientations at different spatial local neighborhoods. As shown in Figure 5.1, given an extremity patch, its gradients are computed first. With a set of m by n spatial cells imposed on the gradient image, the occurrences of each of the $nBin$ edge orientations ($nBin = 9$) in the local

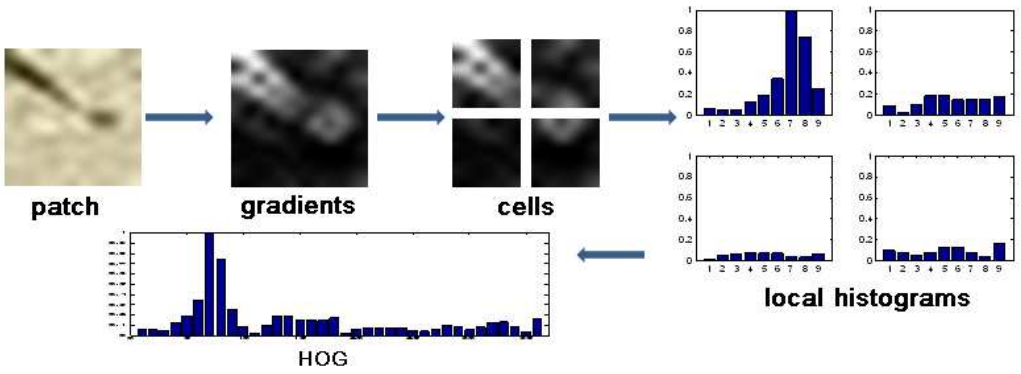

Figure 5.1: The HOG representation of an extremity patch.

cells are counted. All the local histograms are concatenated and normalized to form the final representation as a feature vector of length $m * n * nBin$.

5.3 Predicting a patch as an extremity class

Now the task is to train a classifier so that one can predict how likely a testing patch is from one of the extremity classes. There are mainly two steps, including collecting the training set of extremities and training a classifier to predict a testing patch.

5.3.1 Collecting extremities

In order to identify a patch with a classifier, one needs both positive and negative extremity patches. I designate a set of images as the training set, and extract positive and negative patches from all images in the set. For positive extremity patches, I manually collect example patches at a fixed size and label them as heads, hands or feet. For negative examples, I have written a

program to automatically collect two sets of patches according to the locations of positive patches from the images.

As illustrated in Figure 5.2, given an image in (a), I manually collect four patches in (b), including the head, two hands, and the feet. The corresponding masks are displayed in (c) as white squares. Next my program randomly selects points out of those contours of white squares as patch centers. The resulting patches may cover both a part of the true extremity and the background, as shown in (d). Then my program randomly selects points out of the black regions so that the resulting patches entirely cover the background, as shown in (e). Both sets of patches in (d) and (e) are treated as negative examples, for better identification of extremities from backgrounds. In Figure 5.3 some samples of the three extremity and negative classes are shown.

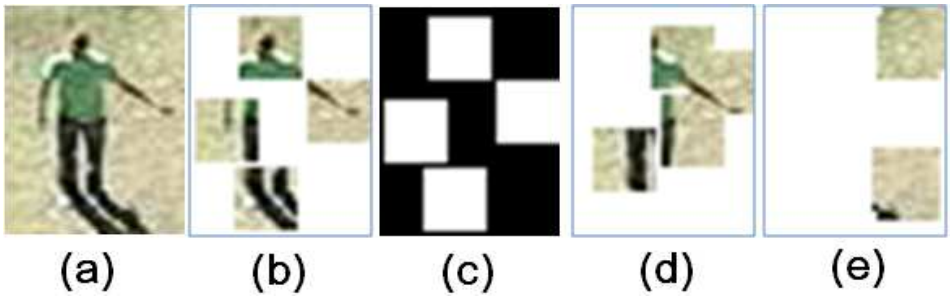

Figure 5.2: To collect extremity patches from frames for training.

5.3.2 Training a classifier to predict

Given the training set of image patches represented by feature vectors, there are many choices for a classifier. The support vector machine (SVM)

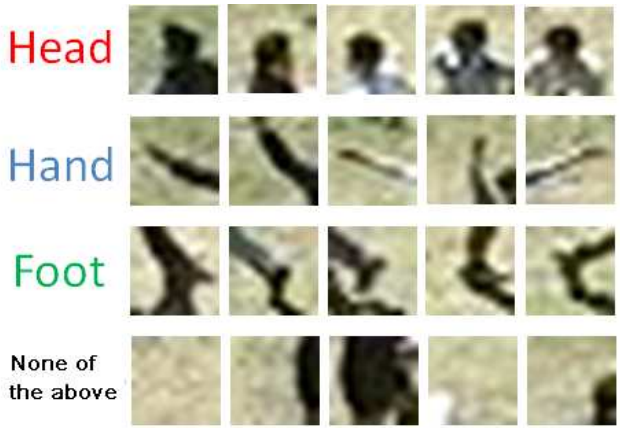

Figure 5.3: Samples of the collected patches for training.

technique is chosen due to its capability of reducing the generalization error through margin maximization. I train and validate with a SVM to see if a patch can be reliably classified as having positive or negative extremities.

It is a multi-class classification problem. In my work, there are three extremity classes plus the negative (not an extremity) class. While the original SVM is for binary classification, it can be extended for multi-class classification. Suppose there are c classes and there are $\binom{c}{2}$ class pairs. For each pair, a binary classifier is trained to assign the testing instance to one of the two classes and the vote for the assigned class is increased by one. In this one-versus-one approach, finally the class with most votes determines the instance classification.

The name "probable extremities" comes from the fact that the classifier produces a probability over each extremity patch. The motivation for having such soft decisions is to improve robustness by capturing more accurate

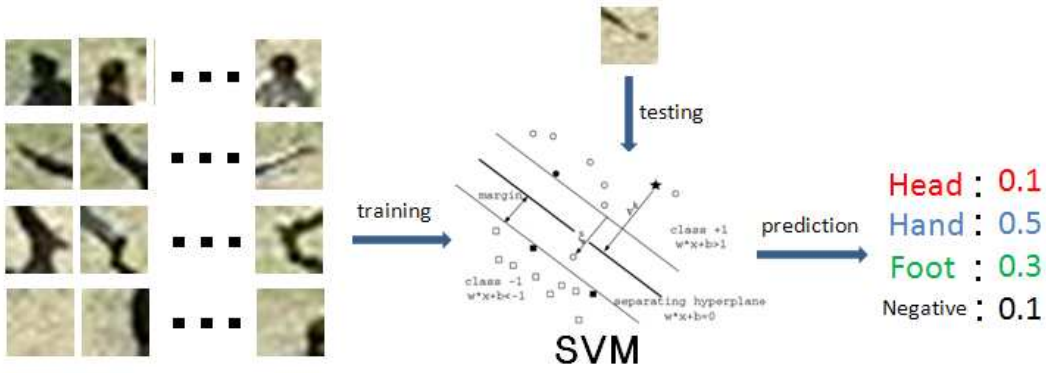

Figure 5.4: Probability estimate of an image patch as extremities or negative.

information and delaying the hard decision to later steps. The binary decision over a patch being an extremity is prone to errors, due to complicated image backgrounds. With a probability describing how likely a patch is an extremity, better representation is achieved.

To predict the probability of an image patch as one of the extremity classes or negative, I utilize the algorithm proposed by Wu et al. [74]. As opposed to the traditional SVM that produces only a binary classification result, they make a probability estimate for multi-class classification by pairwise coupling. For a given test image patch, a number between 0 and 1 is produced for each class as the probability of the patch forming a head, hands, feet or negative. The basic flow is shown in Figure 5.4.

5.4 Detection of probable extremities

Described in the last section is how to predict a patch as an extremity class. However, the task is to detect probable extremities in an image. Hence

Figure 5.5: Building the probable extremity map, which includes three channels for heads, hands, feet.

one needs to search over the image to find all the possible locations for an extremity. Intuitively the solution is to do an exhaustive search over all possible locations. At each location, a patch is cropped and feed into the classifier to produce the probability. In this way, a probable extremity map is built, as shown by the flow chart in Figure 5.5.

5.4.1 Integral histograms

Instead of cropping a rectangle region and computing the histogram repeatedly, there is a faster method to compute histograms for all possible locations over an image. The method is called integral histogram, as first proposed by Porikli [54]. It originates from the integral image idea by Viola and Jones [67], which in turn dates back to the summed area table by Crow [14]. In order to compute the Haar-like rectangle features efficiently, they first compute the integral image of the image to search and then each rectangle feature can be evaluated efficiently with only a constant number of array access and arithmetic operations.

A histogram is a set of numbers, where each number is the frequency of a

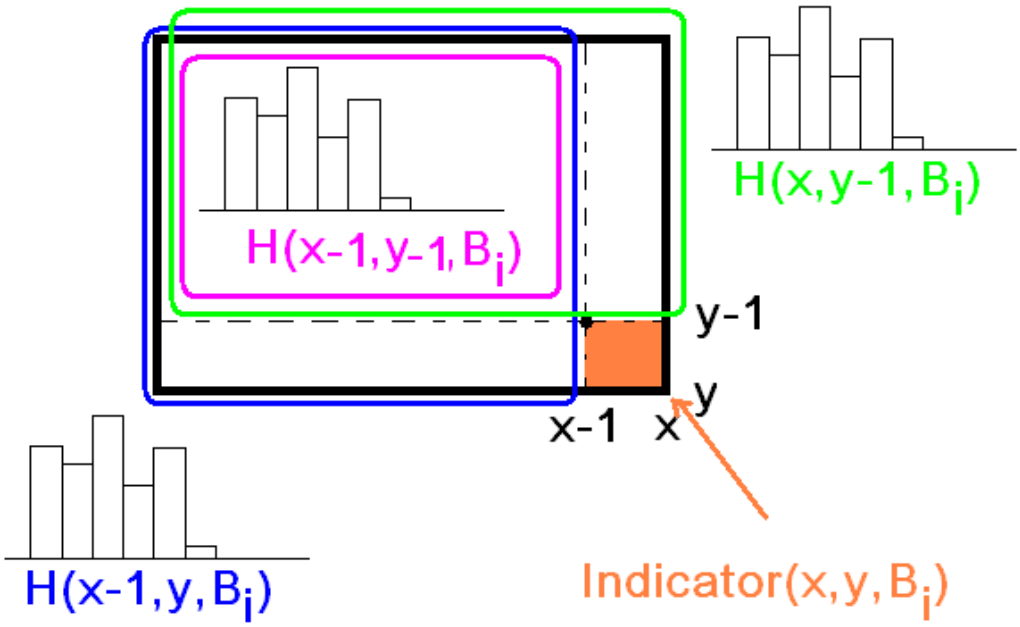

Figure 5.6: The computation of integral histograms for one bin.

range of values in the given data. Such a range of values are called bins. For the HOG descriptor, there are 9 bins (8 directions plus no gradients). The integral histogram is defined as in Equation 5.1, where the function $Indicator(x, y, B_i)$ returns 1 if the value at pixel (x,y) falls within bin B_i or 0 else.

$$H(x, y, B_i) = H(x-1, y, B_i) + H(x, y-1, B_i) - H(x-1, y-1, B_i) + Indicator(x, y, B_i) \quad (5.1)$$

The computation of integral histograms is further illustrated in Figure 5.6. Note in the figure, only one bin is shown. It is easy to verify that

one scan across all the pixels is sufficient to compute the integral histograms. When the integral histograms are ready, it only takes four array accesses, one summation and two subtractions to compute the histogram over any randomly chosen image patch with the following Equation 5.2.

$$\begin{aligned} & h(x_{left} : x_{right}, y_{top} : y_{bottom}, B_i) \\ & \quad = H(x_{left} - 1, y_{top} - 1, B_i) + H(x_{right}, y_{bottom}, B_i) \\ & \quad\quad - H(x_{left} - 1, y_{bottom}, B_i) - H(x_{right}, y_{top} - 1, B_i) \end{aligned} \quad (5.2)$$

5.5 Histogram of probable extremities

With the probable extremity map built, I propose to build a histogram out of it hence the name "Histogram of Probable Extremities" and its abbreviation HOPE. In the case of precise extremities from contours, the histogram of extremities is built with a radical circle centered around the blob centroid. In the case of probable extremities, no blob centroid is available and the histogram is built in the style of the HOG descriptor.

Briefly speaking, a grid of M by N cells are imposed on the probable extremity map. In each cell, the average of probabilities across the cell is computed for each of the three extremity classes. Hence each histogram in the cell is a feature vector of length 3. All the histograms from cells are concatenated and normalized to form a feature vector of length $M * N * 3$.

5.6 Action classification

With each image represented as a HOPE descriptor of length $M*N*3$, I can employ any classifier for action recognition. Each block of T consecutive frames are treated as the most basic action unit, and their HOPE descriptor are further combined to form a feature of length $M*N*3*T$. Adjacent blocks may have OT frames in overlapping. Considering that the block descriptor may be very long, I use Principle Component Analysis for dimension reduction when necessary. Beyond the SVM for estimating extremity probabilities in Section 5.3, I train another multi-class SVM to classify whether a block of consecutive frames belongs to one of the predefined action classes. The sequence label is assigned to the class that gets the most votes from the block based classification.

5.7 Experiments

The framework is applied to two data sets, including the Weizmann data [6] and the Tower data. For each data set, the histograms of probable extremities are built out of frames, and the leave-one-out cross-validation is utilized to compute an overall classification accuracy on sequences. As a baseline comparison, I reported the performance of both the algorithm based on probable extremities and the algorithm based on precise extremities in Table 5.1.

	Weizmann	Tower
Based on precise extremities	93.6%	86.7%
Based on probable extremities	95.7%	98.3%

Table 5.1: Comparison of the two types of extremities on two data sets.

5.7.1 On the Weizmann data

Refer to Section 4.3.2.2 for details on the data set. For those images in Figure 4.16, the corresponding vector images of probable extremities are shown in Figure 5.7. Our best result is achieved with $M = 8$,$N = 6$,$T = 15$,$OT = 5$.

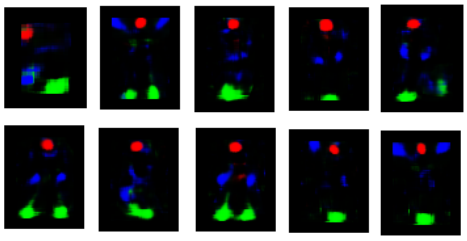

Figure 5.7: Corresponding vector images of the probable extremities. Best viewed in color.

5.7.2 On the Tower data

The same data is described in Section 4.3.2.3. The first 5 frames of each action and the corresponding vector images of probable extremities are shown in Figure 5.8.

5.7.3 Discussion

On the Weizmann data, the probable extremities performed better than the precise extremities , also with the advantage of avoiding contour segmen-

tation. On the Tower data, I obtained significantly better results. From my observation, when the extremities are well detected by the baseline algorithm, my probable extremity approach does not significantly improve the classification accuracy, since it is an approximation to the precise extremity after all. This is exactly what happens with the Weizmann data set. For the tower data, since there is quite some shade associated with human figures in lower resolution, contour segmentation is not a easy job. In such cases, the probable extremities clearly outperform the precise extremities.

5.8 Summary

In this chapter, I present how to compute the probable extremity map from an image and how to summarize it with the histogram of probable extremities descriptor. I achieve reduction of the computation load by implementing the integral histograms. The experimental results are presented to prove the superiority of the probable extremities over the precise extremities.

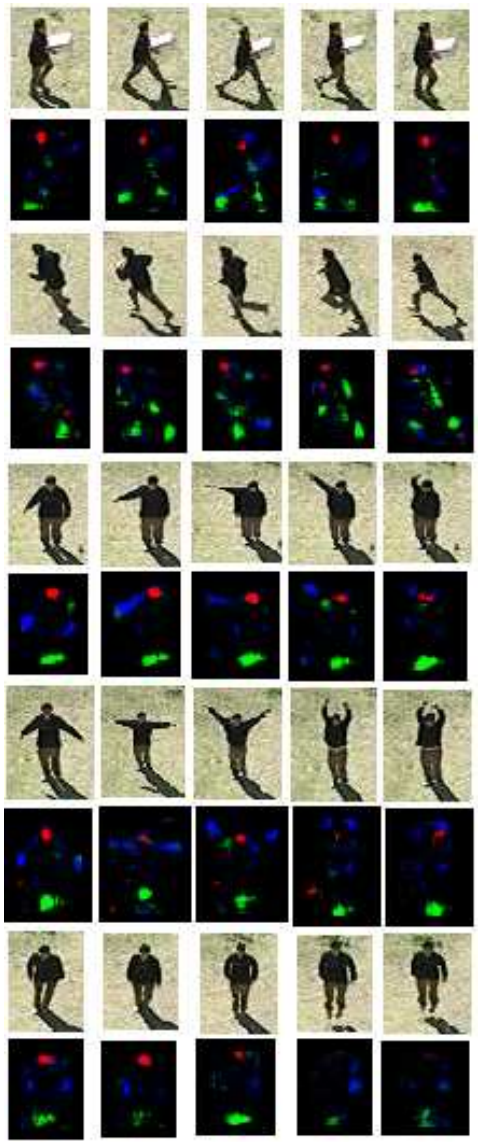

Figure 5.8: The first 5 frames of each action and their corresponding vector images of the probable extremities. Best viewed in color.

Chapter 6

Conclusions

In this dissertation, I presented my unique perspective on analyzing human behaviors with human extremities only.

For precise extremities, the detection goes from the simplest star-skeleton to the most complicated variable-star-skeleton, with more and more comprehensive understanding on what can be seen as an extremity on a contour. After detection, the human body information is essentially reduced to only a few points. It is amazing that so little information can do so much. The applications also go from restricted to more flexible usage. Based on the precise extremities, I derived several features to describe a human posture, including the environment specific features for fence climbing actions, the stable contacts for abstracting continuous videos, and the histogram of extremities descriptor as a general purpose posture descriptor.

For probable extremities, the detection is more or less relaxed to detect the extremities and its nearby body parts. These regions are the most descriptive regions for identifying human posture, in comparison with torsos. In order to detect probable extremities from raw images, I utilized the Histograms of Oriented Gradients for patch representation. At the same time, I implemented

integral histograms for fast computation without sacrificing accuracy.

The major advantage of applying human extremities in action detection and recognition is its compact representation and fast computation while maintaining comparable accuracy against other methods. Furthermore, the usage of human extremities is not to claim that it is the best and only cue for motion analysis. Instead, I promote that it is so powerful that it should be used together with other common cues such as optical flow in motion analysis. More and more state-of-the-art approaches are combining multiple cues into one system, in the hopes of achieving better results with more inputs.

Although my current applications of human extremities are mainly centered around interpreting human actions from either continuous or temporally segmented videos, the detected human extremities may find other usage beyond motion analysis. For example, in the motion capture industry, researchers and developers are trying hard to go markerless in an affordable way. Finding extremities directly from videos is a solid intermediate step toward such a goal.

6.1 Future work

There are a few directions one can go to further push the detection of extremities.

First, the relationship between the detection of extremities and recognition of actions are not fully exploited. Right now, the detection and recognition of actions benefit from detection of extremities, but not vice versa.

Some researchers worked on unifying the segmentation and recognition into one framework, and the same philosophy is applicable in my task. A simple way to go this direction is to build a prior probability on extremity locations for each kind of action, and look for extremities with the appropriate emphasis at different locations under a particular action hypothesis.

Second, in comparison with individual images, videos provide richer information. Working on temporal relationships between image features should help yield better results on extremity detection. For example, human extremities usually alternate between two types of stances. In one type of stance, the extremity exhibits some movement; in the other type of stance, it becomes the stable contact.

Third, using multiple cameras is the best way to reduce object occlusion and self-occlusion among human body parts. With two cameras, the depth may be estimated to differentiate the left and right hand. With more cameras, it is possible to construct a human visual hull to fit a three-dimensional extremity model.

Bibliography

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999.

[2] A. Ali and J. K. Aggarwal. Segmentation and recognition of continuous human activity. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 28–35, Vancouver, Canada, 2001.

[3] Saad Ali, Arslan Basharat, and Mubarak Shah. Chaotic invariants for human action recognition. In *IEEE International Conference on Computer Vision*, Rio de Janeiro, 2007.

[4] K. Barnard, P. Duygulu, N. de Freitas, F. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI.*, 24(4):509–522, 2002.

[6] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *IEEE International Conference on Computer Vision*, pages 1395–1402, Beijing, 2005.

[7] Harry Blum and Roger N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167–180, 1978.

[8] Aaron F. Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Phil. Trans. R. Soc. Lond. B*, 352:1257–1265, 1997.

[9] C. Cedras and M. Shah. A survey of motion analysis from moving light displays. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 214–221, Seattle, 1994.

[10] Rizwan Chaudhry, Avinash Ravichandran, Gregory Hager, and Rene Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[11] Daniel Chen, Pi-Chi Chou, Clinton B. Fookes, and Sridha Sridharan. Multi-view human pose estimation using modified five-point skeleton model. In *International Conference on Signal Processing and Communication Systems*, 2007.

[12] Hsuan-Sheng Chen, Hua-Tsung Chen, Yi-Wen Chen, and Suh-Yin Lee. Human action recognition using star skeleton. In *International Workshop on Visual Surveillance & Sensor Networks*, pages 171–178, Santa Barbara, 2006.

[13] David Crandall, Pedro Felzenszwalb, and Daniel Huttenlocher. Spatial priors for part-based recognition using statistical models. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 10–17, 2005.

[14] Franklin Crow. Summed-area tables for texture mapping. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 207–212, 1984.

[15] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 886–893, June 2005.

[16] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*, 2006.

[17] James W. Davis and Aaron F. Bobick. The representation and recognition of human movement using temporal templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[18] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005.

[19] Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, pages 726–733, Nice, France, 2003.

[20] Mark Everingham, Andrew Zisserman, Christopher K. I. Williams, Luc Van Gool, Moray Allan, Christopher M. Bishop, Olivier Chapelle, Navneet

Dalal, Thomas Deselaers, Gyuri Dork, Stefan Duffner, Jan Eichhorn, Jason D. R. Farquhar, Mario Fritz, Christophe Garcia, Tom Griffiths, Frederic Jurie, Daniel Keysers, Markus Koskela, Jorma Laaksonen, Diane Larlus, Bastian Leibe, Hongying Meng, Hermann Ney, Bernt Schiele, Cordelia Schmid, Edgar Seemann, John Shawe-taylor, Amos Storkey, Or Szedmak, Bill Triggs, Ilkay Ulusoy, Ville Viitaniemi, and Jianguo Zhang. The 2005 pascal visual object classes challenge, 2006.

[21] Alireza Fathi and Greg Mori. Action recognition by learning mid-level motion features. In *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, 2008.

[22] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 524–531, 2005.

[23] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61:2005, 2003.

[24] M.A. Fischler and R.A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22(1):67–92, 1973.

[25] H. Fujiyoshi and A. Lipton. Real-time human motion analysis by image skeletonization. In *IEEE Workshop on Applications of Computer Vision*, pages 15–21, Princeton, 1998.

[26] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999.

[27] Ismail Haritaoglu, David Harwood, and Larry S. Davis. Ghost: A human body part labeling system using silhouettes. In *International Conference on Pattern Recognition*, pages 77–82, 1998.

[28] Ismail Haritaoglu, David Harwood, and Larry S. Davis. w^4: Real-time surveillance of people and their activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[29] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[30] K. Hatun and P. Duygulu. Pose sentences: A new representation for action recognition using sequence of pose words. In *International Conference on Pattern Recognition*, 2008.

[31] N. Ikizler, R.G. Cinbis, and P. Duygulu. Human action recognition with line and flow histograms. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[32] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *IEEE International Conference on Computer Vision*, Rio de Janeiro, 2007.

[33] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14(2):201–211, 1973.

[34] Shanon X. Ju, Michael J. Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *International Conference on Automatic Face and Gesture Recognition*, 1996.

[35] T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[36] Mathias Kölsch and Matthew Turk. Robust hand detection. In *International Conference on Automatic Face and Gesture Recognition*, pages 614–619, 2004.

[37] Ivan Laptev and Tony Lindeberg. Space-time interest points. In *IEEE International Conference on Computer Vision*, pages 432–439, 2003.

[38] Ivan Laptev and Patrick Pérez. Retrieving actions in movies. In *IEEE International Conference on Computer Vision*, 2007.

[39] M. Leo, T. D'Orazio, I. Gnoni, P. Spagnolo, and A. Distante. Complex human activity recognition for monitoring wide outdoor environments. In *International Conference on Pattern Recognition*, volume 4, pages 913–916, Cambridge, 2004.

[40] X. Li. Hmm based action recognition using oriented histograms of optical flow field. *Electronics Letters*, 43(10):560–561, 2007.

[41] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[42] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, 1981.

[43] R Maree, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 33–40, 2005.

[44] J. Min and R. Kasturi. Extraction and temporal segmentation of multiple motion trajectories in human motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 118–126, Hawaii, 2001.

[45] Juan Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, September 2008.

[46] Juan Carlos Niebles and Li Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, 2007.

[47] R. L. Ogniewicz and O. Kubler. Hierarchic voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.

[48] Joseph O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.

[49] Maja Pantic, Alex Pentland, Anton Nijholt, and Thomas Huang. Human computing and machine understanding of human behavior: a survey. In

The 8th International Conference on Multimodal interfaces, pages 239–248, New York, NY, USA, 2006. ACM.

[50] Sangho Park and J.K. Aggarwal. Segmentation and tracking of interacting human body parts under occlusion and shadowing. In *IEEE Workshop on Motion and Video Computing*, 2002.

[51] Sangho Park and J.K. Aggarwal. A hierarchical bayesian network for event recognition of human actions and interactions. *Multimedia Systems*, 10(2):164–179, 2004.

[52] M. Petkovic, W. Jonker, and Z. Zivkovic. Recognizing strokes in tennis videos using hidden markov models. In *International Conference on Visualization, Imaging and Image Processing*, Marbella, Spain, 2001.

[53] P. Peursum, H. Bui, S. Venkatesh, and G. West. Human action recognition with an incomplete real-time pose skeleton. *Technical Report 2004/1*, May 2004.

[54] Fatih Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 829–836, 2005.

[55] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 1989.

[56] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[57] Michael S. Ryoo and J. K. Aggarwal. Semantic representation and recognition of continued and recursive human activities. *International Journal of Computer Vision*, 82(1):1–24, 2009.

[58] Koichi Sato and J.K. Aggarwal. Temporal spatio-velocity transform and its application to tracking and interaction. *Computer Vision and Image Understanding*, 96(2):100–128, 2004.

[59] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *International Conference on Pattern Recognition*, volume 3, pages 32–36, 2004.

[60] Mubarak Shah. Understanding human behavior from motion imagery. *Machine Vision Applications*, 14(4):210–214, 2003.

[61] Michal Shapira and Ari Rappoport. Shape blending using the star-skeleton representation. *IEEE Computer Graphics and Applications*, 15(2):44–50, 1995.

[62] Amnon Shashua, Yoram Gdalyahu, and Gaby Hayun. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 1–6, 2004.

[63] Alexandru Telea and Jarke J. van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *ACM Symposium on Data Visualization*, pages 251–260, Aire-la-Ville, Switzerland, 2002. Eurographics Association.

[64] C. Thurau. Behavior histograms for action recognition and human detection. In *Human Motion Workshop*, 2007.

[65] M. Turk and A. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[66] Michel Vidal-Naquet and Shimon Ullman. Object recognition with informative features and linear classification. In *IEEE International Conference on Computer Vision*, 2003.

[67] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2002.

[68] Julia Vogel and Bernt Schiele. On performance characterization and optimization for image retrieval. In *European Conference on Computer Vision*, pages 49–63. Springer, 2002.

[69] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.

[70] L. Wang and D. Suter. Analyzing human movements from silhouettes using manifold learning. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Sydney, 2006.

[71] L. Wang and D. Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *International Conference on Pattern Recognition*, 2007.

[72] Jon A. Webb and J. K. Aggarwal. Structure from motion of rigid and jointed objects. *Artificial Intelligence*, 19:107–130, 1982.

[73] Daniel Weinland, Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2006.

[74] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.

[75] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:34–58, 2002.

[76] A. Yilmaz and M. Shah. Actions sketch: a novel representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–989, San Diego, 2005.

[77] Alper Yilmaz, Xin Li, and Mubarak Shah. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:1531–1536, 2004.

[78] Elden Yu and J.K. Aggarwal. Detection of fence climbing from monocular video. In *International Conference on Pattern Recognition*, pages 375–378, Hong Kong, 2006.

[79] Elden Yu and J.K. Aggarwal. Detection of stable contacts for human motion analysis. In *ACM Multimedia Workshop on Visual Surveillance & Sensor Networks*, pages 87–94, Santa Barbara, 2006.

[80] Elden Yu and J.K. Aggarwal. Detecting persons climbing fences. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(7):1309–1332, November 2009.

[81] Elden Yu and J.K. Aggarwal. Human action recognition with extremities as semantic posture representation. In *IEEE CVPR Workshop on Semantic Learning and Applications in Multimedia*, Miami Beach, 2009.

Vita

Qingfeng Yu (Elden) received his B.S. degree in Automatic Control from University of Science and Technology of China in 1997, and his Master of Engineering degree in Pattern Recognition and Artificial Intelligence in 2000. He got re-admitted into The University of Texas at Austin and re-instated into F-1 status in 2004, after switching the area of study and taking leave of absence for half a year. He finally graduated in December 2009, as the 42nd and the most persistent Ph.D. that ever existed in the Computer and Vision Research Center.

Permanent address: 5 Bin Jiang Da Dao
Building 4, Unit 3, Room 401
Xiangfan, Hubei, 441021
P.R. China

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.