

# 國立交通大學

## 資訊科學與工程研究所

### 碩士論文

以輪廓特徵學習之人類姿勢辨識



Silhouette Feature Detection Using AdaBoost with Applications  
to Human Posture Recognition

研究生：李啟銘

指導教授：廖弘源 教授

中華民國九十五年六月

以輪廓特徵學習之人類姿勢辨識  
Silhouette Feature Detection Using AdaBoost with Applications to  
Human Posture Recognition

研究生：李啟銘

Student : Chi-Ming Lee

指導教授：廖弘源

Advisor : Hong-Yuan Mark Liao

國立交通大學  
資訊科學與工程研究所  
碩士論文



Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

# 以輪廓特徵學習之人類姿勢辨識

學生: 李啓銘

指導教授: 廖弘源 教授

國立交通大學 資訊工程學系 (研究所) 碩士班

## 摘 要

對一個成功的人類行為分析系統而言，人類姿勢辨識是其中最重要的一部分。在本篇論文中，我們提出以輪廓及機器學習為基礎的方法，以發展出一個有效率且正確性高的人體姿勢辨識系統。在所提出的系統中，我們首先由人體輪廓抽取出其中具有高度識別性的特徵，並利用名為 AdaBoost 的機器學習方法來建構我們的辨識系統。我們根據人類姿勢中獨有的特性來選擇並修改 AdaBoost 所使用的特徵。藉由我們所使用的特徵所具的高度描述能力，我們將呈現此系統會較使用傳統步驟的方法有更高的效能。使用者可藉由我們所提出的架構來達到一個快速且有效的人類姿勢辨識。

# Silhouette Feature Detection Using AdaBoost with Applications to Human Posture Recognition

Student : Chi-Ming Lee Advisor : Dr. Hong-Yuan Mark Liao

Institute of Computer Science and Engineering  
National Chiao Tung University

## Abstract

Human posture analysis is one of the most important steps towards successful human behavior analysis. In this thesis, a silhouette-based learning approach is proposed to develop an efficient and effective human posture recognizing system. Discriminating features from human body silhouette are first extracted and then AdaBoost algorithm is employed for training a recognition system. The features in AdaBoost are selected and modified according to the specific characteristics of human postures. Depending on the describing ability of our features, we demonstrate that our system operates higher performance than the traditional approaches. Users can recognize human postures in an efficient and effective manner using the proposed framework.

# Contents

<b>Abstract in Chinese</b>	<b>i</b>
<b>Abstract in English</b>	<b>ii</b>
<b>Contents</b>	<b>iii</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Background</b>	<b>3</b>
2.1 Human Motion Analysis .....	3
2.1.1 Human Body model .....	4
2.1.2 Tracking .....	5
2.1.3 Human Activity Recognition .....	6
2.2 Human Posture Detection .....	7
2.2.1 Foreground Extracted Method .....	8
2.2.2 Contour Information .....	8
2.2.3 Shape Context Method .....	9
<b>3. Our Approach</b>	<b>12</b>
3.1 Learning Approach .....	14
3.2 Feature Selection .....	18
3.3 Training Process .....	24
3.3.1 Integral Gradient Orientation Map .....	24
3.3.2 Training Algorithm .....	26
3.3.3 Training Cascade of Classifiers .....	31

3.3.4	Directly Feature Assignment .....	32
<b>4.</b>	<b>Experiment</b>	<b>33</b>
4.1	Experimental Result .....	33
4.2	Discussion .....	35
<b>5.</b>	<b>Difficulties</b>	<b>36</b>
5.1	Background Subtraction .....	36
5.2	The difficulty in constructing data base .....	37
5.3	Occlusion Problem .....	38
<b>6.</b>	<b>Conclusion</b>	<b>39</b>
	<b>References</b>	<b>40</b>



# List of Figures

2.1 Relationship among three areas of human motion analysis .....	4
2.2 Human body representation models .....	5
2.3 Activity recognition based on motion trajectory which is generated by optical flow tracking .....	6
2.4 (a)HMM Concept (b)An example of human action recognition using HMM .....	7
2.5 The human posture detecting approach in $W^4$ .....	8
2.6 Using the contour information to estimate body parts .....	9
2.7 Abnormal postures .....	9
2.8 The Concept of Shape Context .....	10
2.9 Use shape context matching to construct the human body configuration .....	11
3.1 Basic flow chart of our approach .....	12
3.2 Learning Stage: For each posture in data base, we generate a classifier by employing AdaBoost training process .....	13
3.3 Detecting Stage: Use the classifier cascade generated by training stage to detect the input image frame .....	13
3.4 Harr rectangle features .....	14
3.5 The first two features select by AdaBoost .....	14
3.6 The detection cascade of AdaBoost .....	15
3.7 Implementation of face detection using AdaBoost approach .....	15
3.8 The Contour fragment representation of a horse .....	16
3.9 Use Harr rectangle feature to train significant feature cascade to the silhouette of human postures.....	17
3.10 Viola and colleagues' work for automatic animating a dancing partner from	

the configuration of captured human pose .....	17
3.11 Gradient curvature of an image.....	19
3.12 Tensor representation .....	20
3.13 Local descriptor of SIFT key .....	21
3.14 (a) Two different rectangle. (b) Body silhouette of two persons with different physiques .....	22
3.15 (a) Original image. (b) The extracted foreground of the image. (c) The gradient orientation along the verge point .....	23
3.16 The feature used in our approach .....	24
3.17 The value of the integral image map .....	24
3.18 The representation of integral image.....	24
3.19 The accumulated gradient orientation within a block can be computed with four references .....	26
3.20 Example of selecting threshold for weak classifier .....	30
3.21 Weak Classifier Examples .....	30
3.22 Example of pre-assigned feature approach .....	32
4.1 Human Posture data base in our experiment .....	33
4.2 Divide a motion as several postures .....	34
4.3 Example of perturbation in a posture class .....	34
4.4 Trade-off between false positive rate and correct detection rate .....	36
5.1 The situation of imperfect background subtraction .....	37
5.2 Poser: a software for animating and controlling human motion .....	38



# List of Tables

Table 1	The Comparison between the experimental results of gradient orientation feature and Harr rectangle feature .....	35
---------	--	----



## 1. Introduction

Human posture analysis is one of the most important steps towards successful human behavior analysis. The difficulty of human posture analysis is twofold. First, the movement of a human body is an articulated motion. Therefore, it is obvious that the issue to be addressed is a problem with high dimensionality and complexity. Second, characterization of human behavior is equivalent to dealing with a sequence of video frames that contain both spatial and temporal information. The most challenging issue is how to properly characterize spatial-temporal information and then facilitate subsequent comparison/retrieval tasks. The objective of this study is to propose a methodology for systematic human posture analysis, not simply conducting a thorough human behavior analysis. Therefore, the detailed descriptors, such as the movement of hands, feet or torso, are not the main subjects of this research. The posture classification systems proposed in the past can be categorized into two classes, i.e., 2-D based or 3-D based, depending on the types of human body model adopted [4]. Most of the existing approaches made a basic assumption: a complete action is composed of a consecutive sequence of posture changes. And, the starting and the ending time spots for the sequence of posture changes are pre-defined. Therefore, one has to identify the type of posture in a frame before behavior (action) analysis is conducted. In [1], Haritaoglu et al. proposed a W4 system which is able to identify the type of postures. They categorized the type of posture into four kinds: standing, sitting, lying down, and crawl-bend. They designed a state transition diagram to identify different postures at different time instants. The main drawback of their approach is twofold: (1) their system can only recognize four postures; (2) the details of posture changes are not characterized. In [12], Leung and Yang did a thorough study on the issue of posture analysis. Their work is a fairly early effort in comparison with the advanced techniques developed recently. However, their posture analysis based on the silhouette of a subject did inspire significantly a number of subsequent researchers. One of the main drawbacks of their work is the complexity of their algorithm. In nowadays, sufficient

efficiency is indispensable for real-time applications. Bobick and Davis [15] proposed a temporal template built by stacking a set of consecutive frames. The proposed temporal template characterized human motion by using motion energy images (MEI) and motion intensity images (MHI). Moment based features were extracted from MEI and MHI and they used these moment based features to conduct template matching. The idea of this representation is excellent. However, the design of its early image processing stage is not stable. The unstable versions grabbed from different frames result in a very unstable representation outcome. There are quite number pieces of works dealing with the human-motion related topics. People who are interested in the related topics may refer to [9,10]. It is also possible to perform human behavior analysis based on human silhouette analysis. In [26], the projection histograms of each person were computed and compared with the probabilistic projection maps stored for each posture during the training phase. For the purpose of posture classification reliability, the obtained posture was further validated exploiting the information extracted by a tracking module. In addition to 2-D model-based systems introduced above, there are also some existing 3-D model-based systems. Boulay et al. [5] first computed projections of moving pixels on a reference axis and learned 2-D posture appearances through PCA. Then, they employed a 3-D model of posture to make the projection-based method independent of the camera position. Zhao et al. [11] used a 3-D human model to verify whether a moving region detected is a person or not. The verification process was done by walking recognition using an articulated human walking model. However, due to the need of developing low-cost systems, complex computations and expensive 3-D solutions will not be considered in this study. In order to achieve the goal of building an efficient and automated human behavior analysis system, the first important step is to identify the significant postures from a human behavior video sequence systematically and automatically. In this thesis, we try to present an accurate and efficient human posture detecting system which can best achieve this target. Based on the experience learned from the

face detection problem [6] , we would like to use learning approach for building a real-time system for this application. We will demonstrate the whole structure of our algorithm in this these.

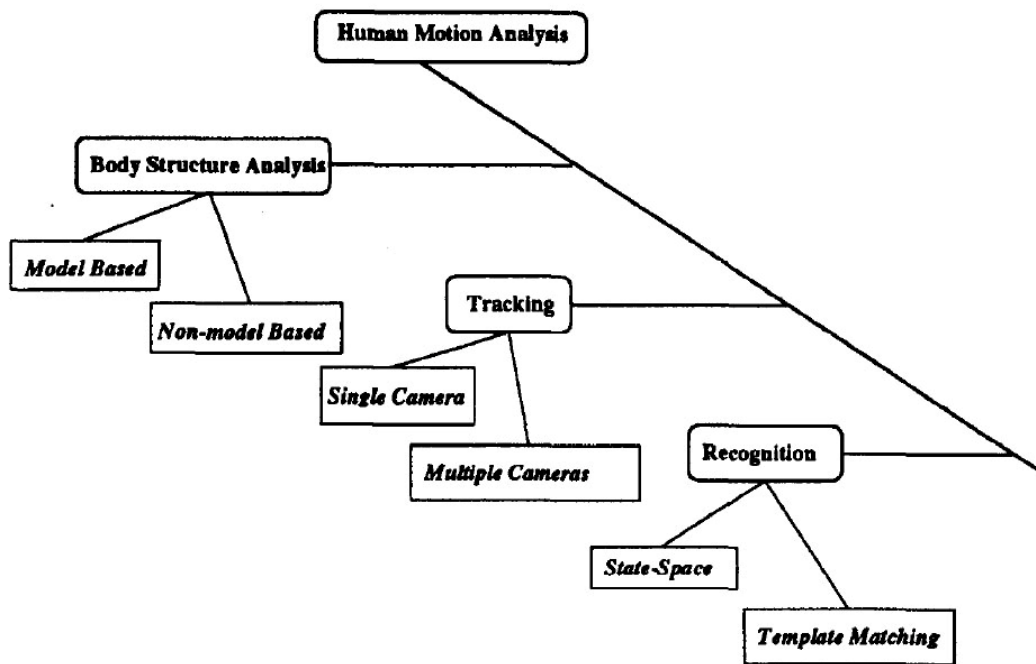
The remainder of this thesis is organized as follows: Section 2 describes the background of our research. We will briefly describe the work of human motion analysis, and review some previous human posture detection methods. In Section 3, we demonstrate our approach, including the selecting features and the structure of our learning algorithm. Section 4 shows the experimental result of our implementation and discusses the performance of our system. In Section 5, we discuss the problems occurred when implementing this work. And last we give a brief talk about extending researches after what we have done and conclude our work in Section 6.



## **2. Background**

### **2.1 Human motion analysis**

In this chapter, we will briefly introduce the development of human motion analysis and then describe the motivation of our work on human posture detection. Aggarwal and Cai [9] have provided a review on human motion analysis. Human motion analysis can be divided into three research areas from low level to high level: (1) human body structure, (2) tracking approach and (3) activity recognition approach. (See Fig. 2.1)

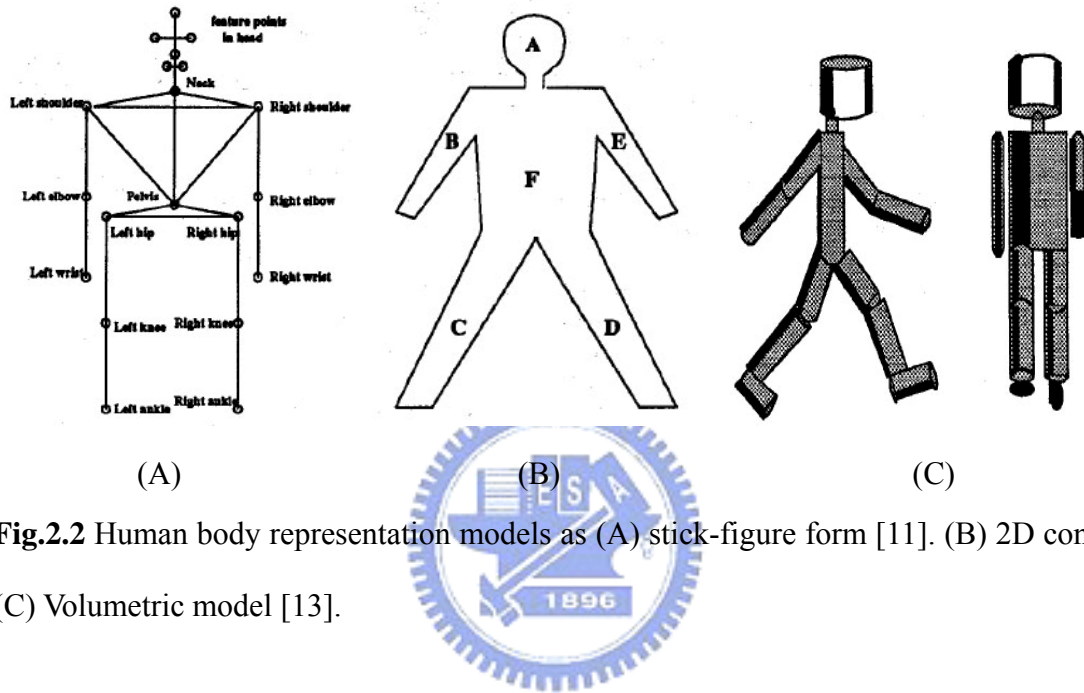


**Fig. 2.1** Relationship among three areas of human motion analysis addressed in [9]

### 2.1.1 Human body model

Conventionally, human body can be represented as stick figures, 2D contours or volumetric models [10] (See Fig. 2.2). In those approaches of analyzing human motion using the information of body structure, human body is first segmented to several body parts like head, limbs and torso. The movement information of these body parts presents the key to the estimation and recognition of human motions. This kind of methodology follows a general framework that it first extracts features, and then establishes the feature correspondence for the information of body parts, and then performs high-level processing at last for analyzing human motion. Depending on whether information of object shape is employed, there are two kinds of strategies: model-based and non-model based method. Non-model based method simply extracts features first, such as lines and joints in stick-figure representation or 2D ribbons and blobs in 2D contour representation, and then the feature correspondence is built later using heuristic assumptions. Other than that, in the model-based approach, a human body model should be constructed first, such as stick-figure form or the projection of 2D contour

form. Feature correspondence in a model-based method is easily to achieve once that the matching between the images and the model is established. This kind of approach performs basic and low-level human motion analysis method. Efficiency and accuracy are what we need to concern. Detailed description can be seen in [11, 12, 13, 14].

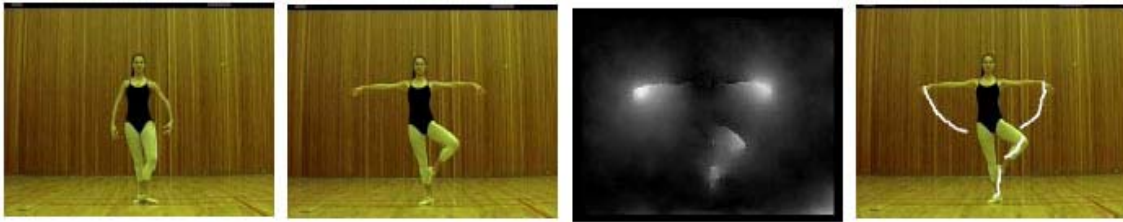


**Fig.2.2** Human body representation models as (A) stick-figure form [11]. (B) 2D contour [12]. (C) Volumetric model [13].

### 2.1.2 Tracking Approach

Unlike previous method, tracking-based approach doesn't extract each body part explicitly in consecutive frames. It simply uses low-level visual features to track or to recognize motions, such as shape, color or texture. The target of this approach is to build the correspondence of image structure between consecutive frames. Iconic model and structural model are two general correspondence models, which use correlation templates and image features respectively. However, there are some problems in realizing tracking-based approaches. First, features may disappear under tracking in some motions which contain self-occluded posture, and thus tracking process would fail under this circumstance. Second, pre-label is needed for our interested features, which is impractical for an automatic long-term

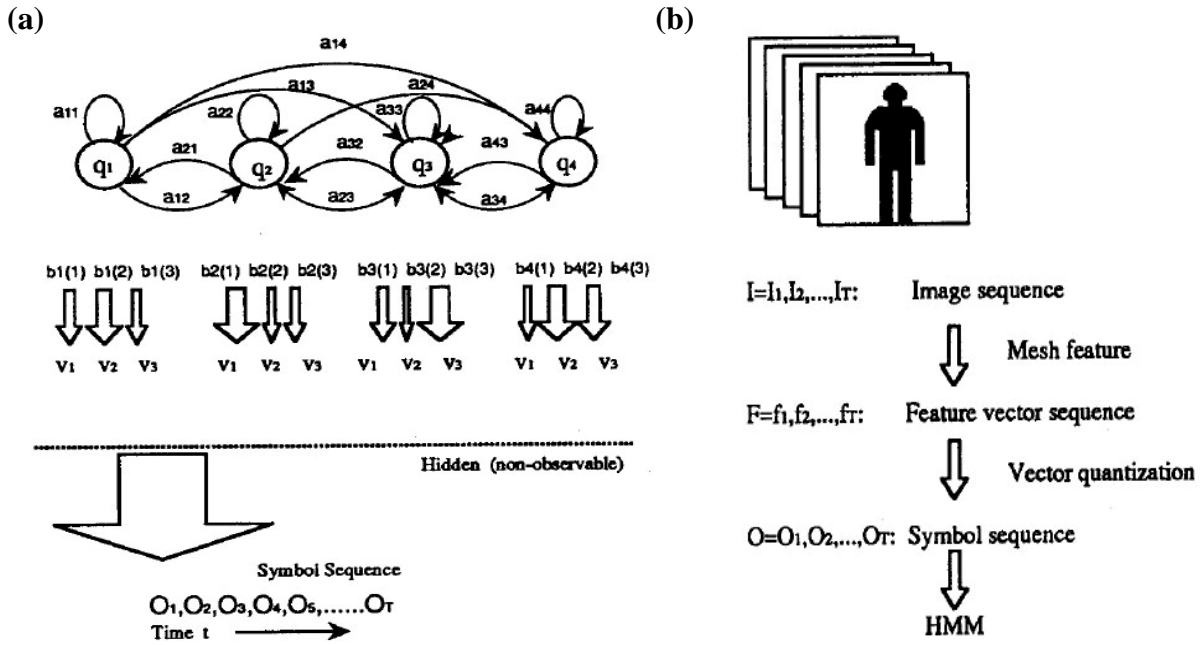
surveillance application..



**Fig. 2.3** Activity recognition based on motion trajectory which is generated by optical flow tracking [18].

### 2.1.3 Human activity recognition

High level methods of human motion analysis can be divided into two categories, template matching approach and state-space approach. Bobick and Davis [15] use the optical flow approach developed by Polana and Nelson's work [16] to build a template matching approach for human motion recognition. They represent human actions as MEI (Motion Energy Images) and MHI (Motion History Images) in sequential image, and recognize motions by matching those templates. However, the design of its early image processing stage is not stable. The unstable visions grabbed from different frames result in a very unstable representation outcome. On the other hand, state-space is a most popular approach for human activity recognition. It defines each posture as a state, and connects each state with probabilities. Each motion can be represented as a state sequence. Numerous kind of state-space models are used for analyzing human actions. Yamato and colleagues [2] use Hidden Markov Model for this task, and following this, Hogg and colleagues [17] use another structure namely Variable-Length Markov Model to solve human motion analysis problem.



**Fig. 2.4** (a) HMM Concept; (b) an example of human action recognition using HMM. [2]

To analyze human motions, the major task is to establish feature correspondence between consecutive frames, i.e., to compute the similarity between consecutive human postures. A good human posture recognition system will help us for solving the problem of analyzing human motion. By perfectly detection of human posture in each image frame, we can easily get information of body structure and feature correspondence. With the success of posture recognition, it can be used as the support for tracking-based approaches to labeling body parts. Also, for a state-space method, the recognized posture in each image can be directly used as the high-level information in each state. Therefore, in this work we make our efforts on develop an efficient and accurate system of human posture recognition.

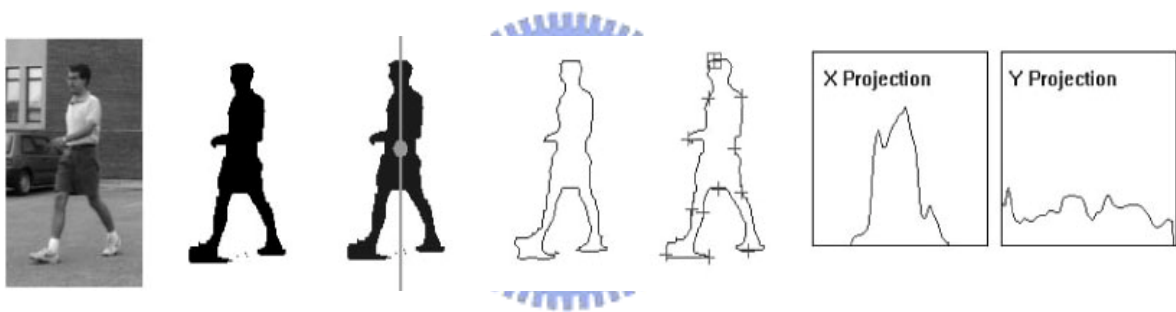
## 2.2 Human posture detection

In this chapter, we will review some previous works of recognizing human postures.



### 2.2.1 Foreground Extracting Method

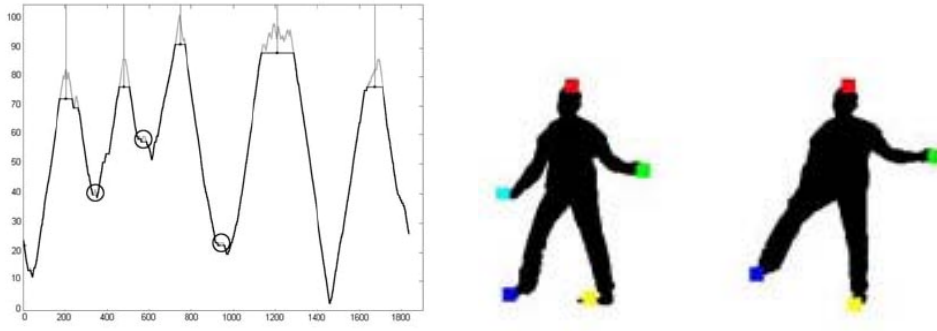
Haritaoglu, Harwood, and Davis [1] integrate a set of techniques for building a real-time visual surveillance system called  $W^4$ . In the human posture detection part, they first extract the foreground image, determine a major axis of the foreground region, and then use the X projection and Y projection of the image as a feature template. Extreme points or curvature maxima of the silhouette boundary such as corners or convex/concave hulls are used to represent the features of body parts (See Fig. 2.5). The X, Y projection of human image is a rough presentation of human body, and for more complicate pose of human or for more huge data base, the matching will become difficult and the accuracy may turns down. So, for our request, we need a discriminating feature to represent human body.



**Fig. 2.5** The human posture detecting approach in  $W^4$ . Last two figures present the characteristic of human body silhouette in the vertical and horizontal domain.[1]

### 2.2.2 Contour information

Without pre-constructed data base, some researches estimate posture directly from the information of human body silhouette. It use the contour structure to evaluate the protrusion of human body, and identify them as body parts like head or limbs. It may starts from the center of mass of the foreground image, and calculates the correlated value such as Euclidean distance or curvature from the center to the edge point of contour. The local maximum of the correlated value may be used to estimate the protrusion of body as body parts. ( See Fig. 2.6)



**Fig. 2.6** Using the contour information to estimate body parts. [25]

Problem of this kind of approach is that it merely extracts the protrusion parts, and estimate those body parts by heuristic assumptions. If some abnormal posture occurs, it may puzzles about the position of body parts, like Fig. 2.7. Also it is sensitive to noise and small perturbation of the contour, so it is unreliable in a real implementation.

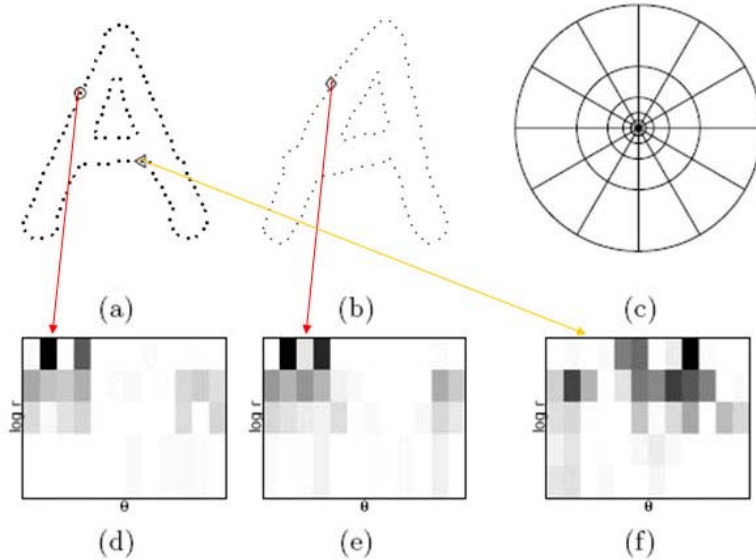


**Fig. 2.7** Abnormal posture may confuse the approach which merely uses the contour information as assumption.

### 2.2.3 Shape Context method

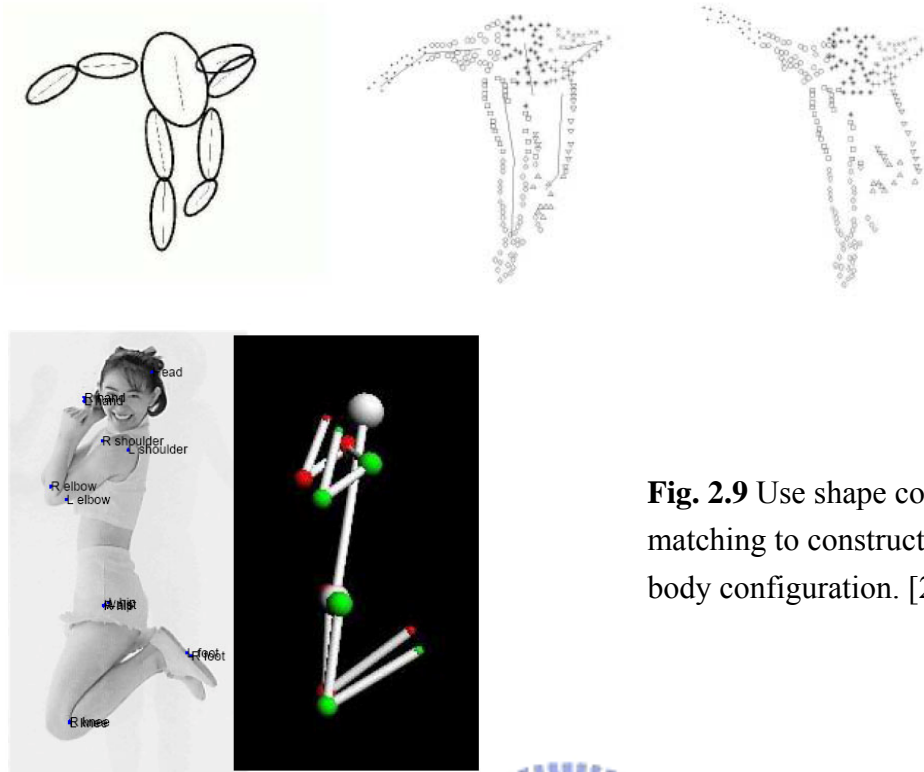
Shape Context is an object recognition method developed by Belongie, Puzicha and Malik [19]. It uses the information of shape, and builds a log-polar histogram over each sample point along the contour. Recognition is treated as the similarity matching of histograms between two images. Because that the log-polar bin contains a small region of shape information, it is tolerant with small geometrical distortions, occlusion and presence of

outliers. By normalizing the log-polar circle, it can also achieve the invariance of scaling. Fig. 2.8 shows the concept of Shape Context.



**Fig. 2.8** The Concept of Shape Context. (a) and (b) present sampled points of two shapes. (c) is the diagram of log-polar histogram bins. (d), (e), and (f) is shape context of the samples in (a) and (b). Each histogram is generated by applying the log-polar diagram at the sample point, and accumulates samples in each bin. We see that if points have similar structure, their shape context histogram will also be similar [19].

Mori and Malik apply the shape context approach to the human body configuration problem [20]. By matching two human posture images with shape context, we can easily decide the similarity of two images, and then we can pick the best match as the result for the detected frame. By the matching process, Mori and Malik build a system that can easily construct the human body configuration by matching the interest frame to a pre-labeled posture data base. (See Fig. 2.9)

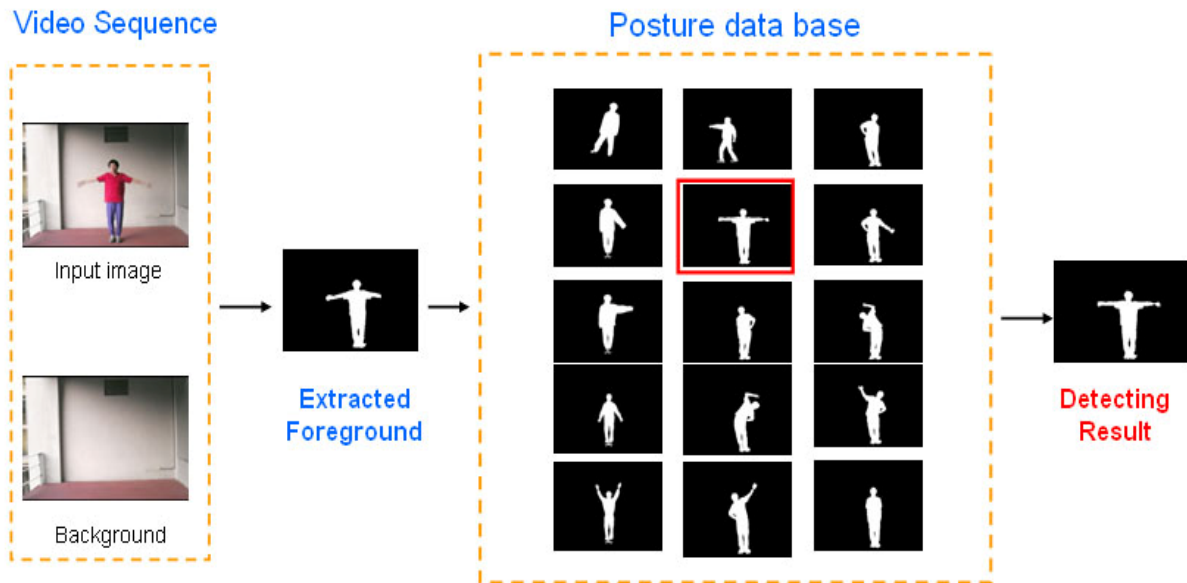


**Fig. 2.9** Use shape context matching to construct the human body configuration. [20]

The representation of shape context can precisely describe the silhouette structure of human body, and it performs a robust process in matching shape. But the major drawback of shape context is that it is consuming in computational cost. It spends a lot of time in matching histograms point by point. For the task of finding best match in a huge posture data base, the cost would be high and thus obviously does not meet our requirement of computation efficiency.

Through the result of above discussions, we need to find an approach which is accurate in representing the body structure, and also efficient in computation time. We choose learning approach which is employed in the robust face detecting application and achieves excellent performance [6]. Using learning approach means the need of building large data base and the cost of huge training time, it is reasonable for huge computation in the off-line pre-processing task. We will demonstrate the whole approach in next section.

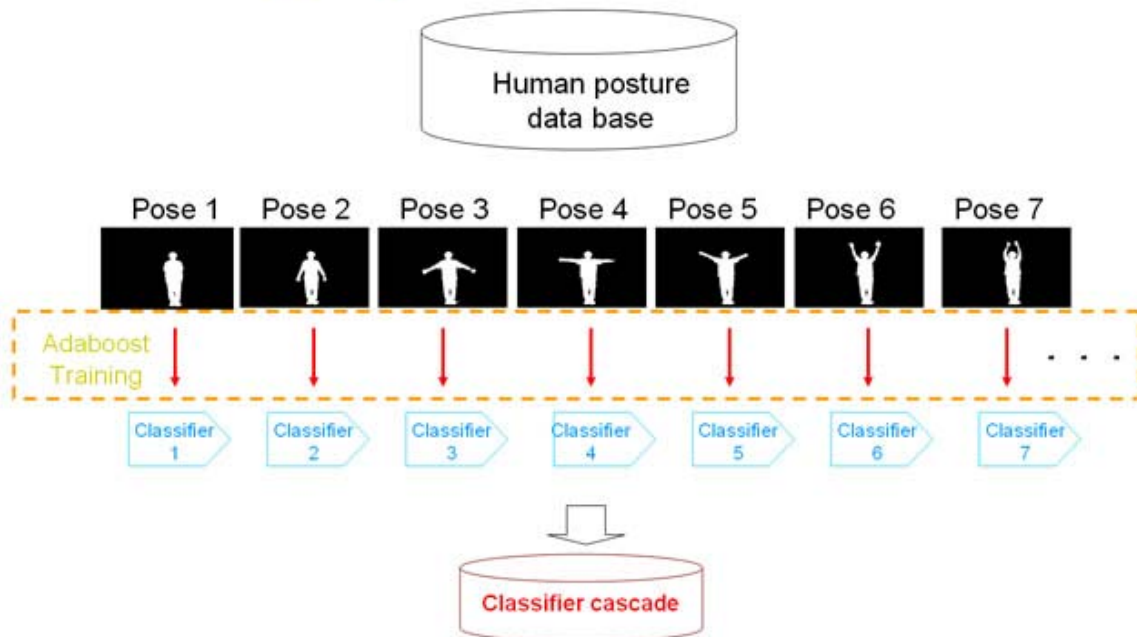
### 3. Our approach



**Fig. 3.1** Basic flow chart of our approach.

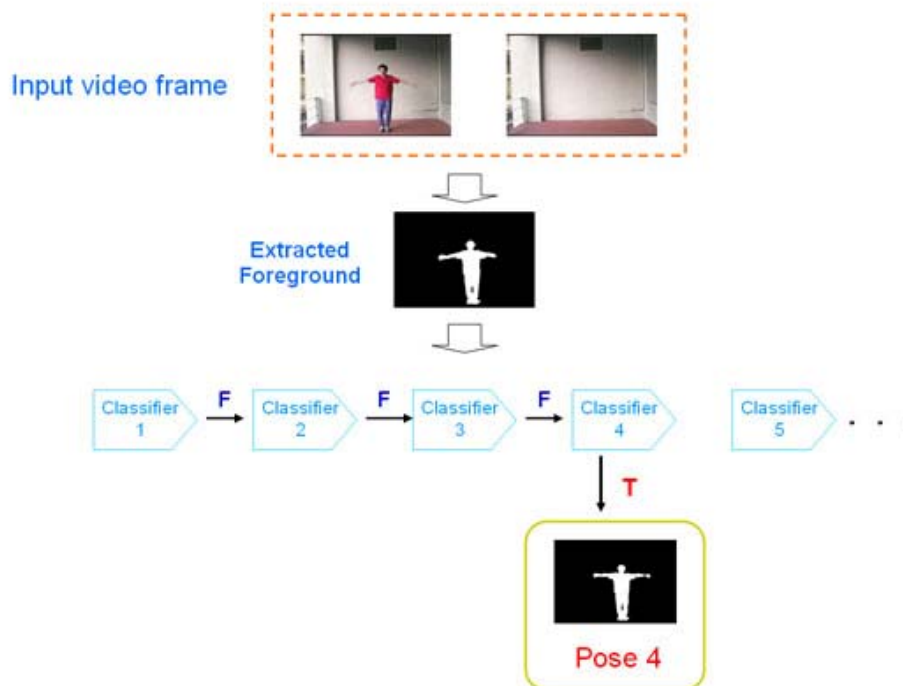
For a given video sequence, we first extract the foreground region each image frame, and then compare the extracted foreground to the posture data base. Finally, the best match is treated as the detecting result. We could simply use shape context as our matching method, but as we mentioned above, the computational time will be a huge consumption. Therefore, we use a learning approach namely AdaBoost which is developed by Viola and Jones in the work of real-time face detection [6]. The AdaBoost approach uses numerous simple features to represent the character of an object, and uses training approach to select amounts of most discriminating features as the detecting cascade. In the case of face detection task, since we have only one target - human face, the training process needs to generate only one feature classifier to identify it. But in the case of human posture detection, our target is numerous different human poses. Thus, we need to perform a “Class Training” process to generate a detecting cascade as a classifier for each pose. Our proposed method is shown in Fig. 3.2 and Fig. 3.3.

## • Learning stage



**Fig. 3.2** Learning Stage: For each posture in data base, we generate a classifier by employing AdaBoost training process.

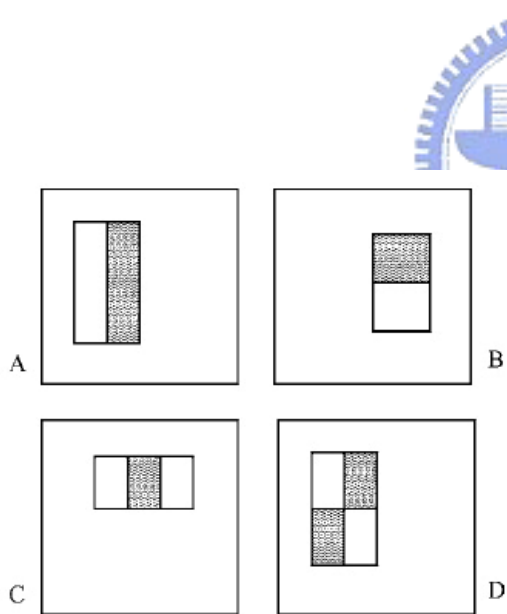
## • Detecting stage



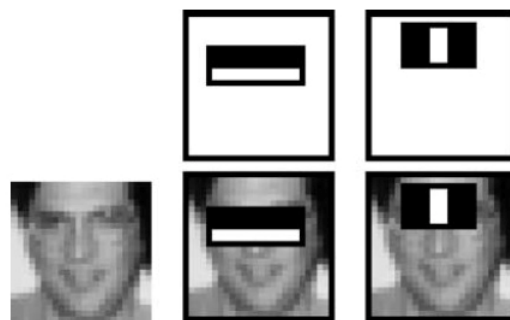
**Fig. 3.3** Detecting Stage: Use the classifier cascade generated by training stage to detect the input image frame.

### 3.1 Learning approach

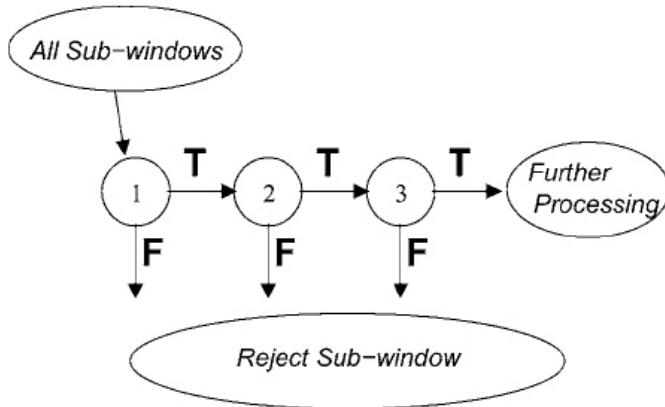
Boosting has been developed as a simple yet powerful technique for building an accurate classifier from a set of “weak learners”. AdaBoost is a feature-based learning approach which has been applied in the real-time face detection and operates excellent performance. It uses large amount of rectangle features called Harr-like feature (See Fig. 3.4 and Fig.3.5) to train a series of feature cascaded as a detecting chain as shown in Fig.3.6. Each feature contains local information and an adapted threshold to distinguish objects from non-objects. By a boosting approach, the most discriminating features will be assigned larger weights to give more powerful influence to the detecting result. By fast eliminating false positive with cascaded features, the system can obtain the candidate efficiently. It rapidly discards a lot of false sub-window, and keeps candidates which pass all classifiers as the final detection.



**Fig. 3.4** Harr rectangle features. The sum of pixels within white rectangles is subtracted from the sum of pixels in black rectangles.

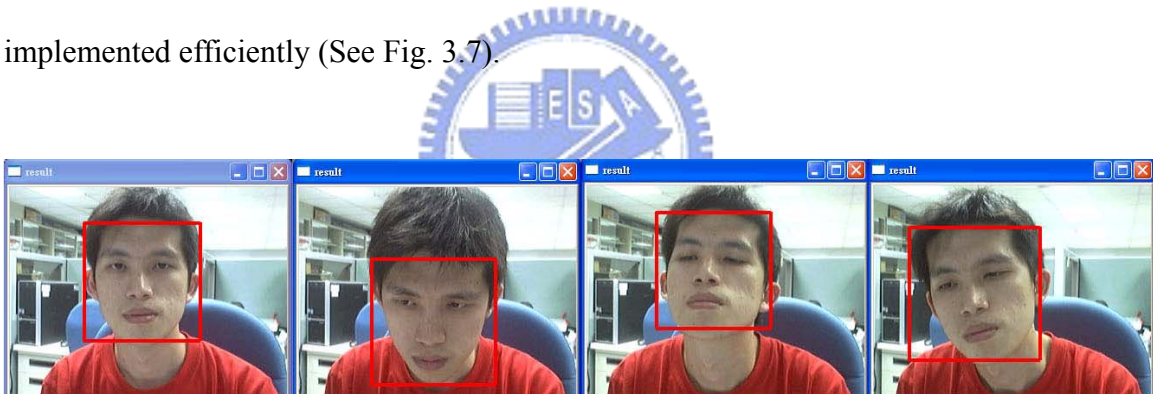


**Fig. 3.5** The first two features select by AdaBoost. Both of them represent significant structure of human face.



**Fig. 3.6** The detection cascade of AdaBoost [6].

The reason that Viola and Jones use feature-based model rather than pixel-based one is that: First, feature-based approach preserves more domain knowledge. In addition, feature-based system operates much faster than a pixel-based one. The system can be implemented efficiently (See Fig. 3.7).

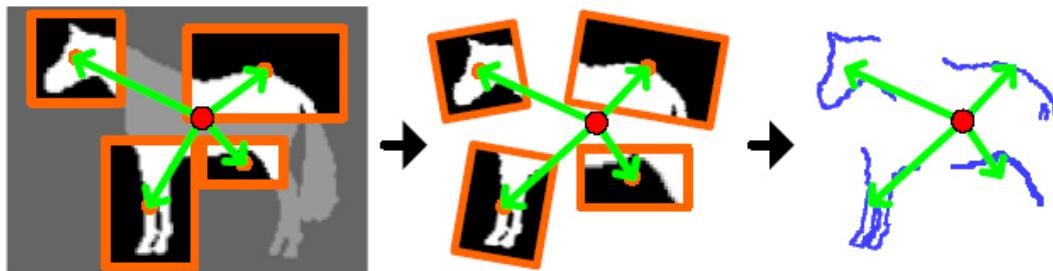


**Fig. 3.7** Implementation of face detection using AdaBoost approach.

Shotton et al. [21] evolve another learning approach for object detection. Unlike the work of Viola and Jones, they construct the system which not merely towards a specific object, but for the recognition of multi-object classes. Their method extracts fragments of object contour, and use “gentle-boosting” to train those fragment features. Star skeleton with contour fragments of the object is built after training, and is considered as a class-specific model for this object (See Fig.3.8). In the recognition part, they use oriented chamfer matching to compare the similarity of contour fragment model. The contribution of their work is that they learn from a small set of supervised object data to generate a rudimentary detector, and train



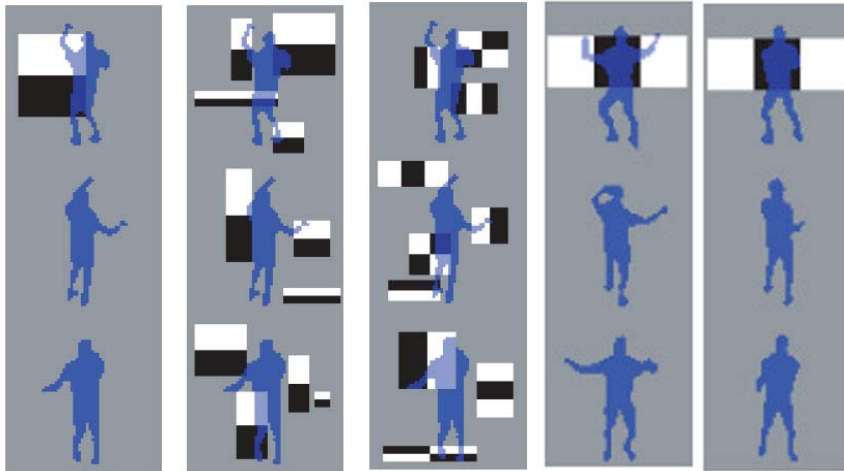
in second stage from a larger unsupervised data set to get the final detector. Therefore, unlike traditional learning approach, they save large amount of computation time in supervising a large data base. However, because of their chamfer matching method in detection process, the recognition time will be very slow in their system (about 10 second per frame in their experiment), so that it is not suitable for a real-time application.



**Fig. 3.8** The Contour fragment representation of a horse presented in [21]. They use gentle-boosting to train the most discriminating fragments of the object's contour, and build a star-skeleton model according to the center of mass.

For the purpose of real-time detection, we choose the AdaBoost training approach as the underlying technique and extend it to an object class recognition method for suitable our task. Viola et al. [8] has expanded their work with colleagues to a vision-based performance interface for controlling animated human character. In this work, they use AdaBoost to train features of silhouette for different postures for the purpose of real-time animating and controlling the configuration of human body(as shown in Fig. 3.9, Fig. 3.10). In our opinion, Harr rectangle feature is just a simple feature with less ability to describe structure. In the case of face detection, the texture is complicated and the contrast of brightness is strong on human face. Hence, we can represent it efficiently with amounts of Harr rectangle features efficiently. In the case of human body silhouette, the representation of contour form is too rough, and thus we need more features to describe each object (in this case, posture) more precisely. However, higher complexity of features would result ing higher computation cost in the

training process of AdaBoost. Rather than using Harr rectangle features, we propose an approach to construct an efficient system not only in detection but also in training.



**Fig. 3.9** Use Harr rectangle feature to train significant feature cascade to the silhouette of human postures. [8]

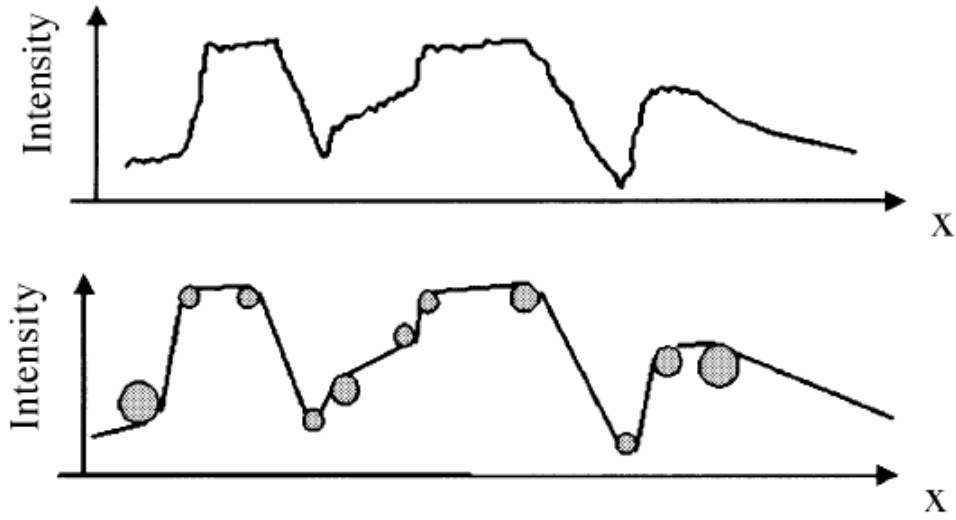


**Fig. 3.10** Viola and colleagues' work for automatic animating a dancing partner from the configuration of captured human pose. [8]

### 3.2 Feature selection

In the research domain of object recognition or retrieval, we often use three kind of features to describe an object: color, texture and shape. In our work, we only extract the shape information from the silhouette of human body. Color feature cannot suffice for this application since different dressing of people would significantly change the characteristic in color domain, but the posture would be similar. Texture may be useful in the representation of some specific body part like face or hand, but for the view of global body structure, it isn't such helpful. Besides, it will be strongly influenced by dressing. And thus it is the reason why we choose shape feature as the description of human posture.

In the perception of human eyes, we used to interpret moving figure as a priori shape model learned from our previous experience which strongly impress us. Since edge is a strong feature for human perception, for this reason it is used in many image recognition researches. In our work, we choose image gradient orientation as our feature rather than using edge directly. Image gradient exhibits the direction that the image intensity curvature changes steeply. Image gradient preserves the same character as edge, and is more sensitive to the human vision perception since that it present the region information instead of a line only when intensity changes. Like edge, it is robust to light condition because illumination changes may result in significant changes to gradient magnitude but is likely to have relatively less influence on gradient orientation.



**Fig. 3.11** Gradient curvature of an image. The above fig presents the image intensity along some axis line. The bottom fig shows the characteristic of image gradient. It use circle to represent the change of intensity curvature in the verge point, and the radius shows that the intensity changes steeply or smoothly. [22]

Fig. 3.11 shows the gradient curvature along some crossing line over an image. In our task, we need to calculate the intensity curvature change in a 2D image, which means a calculation of gradient curvature orientation for a 3D model is needed (contains X, Y and intensity domain of the image). Eq. (1) is conducted by Ngo, Pong et al. [23]. They regard image gradient orientation as “Structure Tensor” and formulize it as:

$$\theta = \frac{1}{2} \tan^{-1} \frac{2J_{xy}}{J_{yy} - J_{xx}} \quad (1)$$

, where

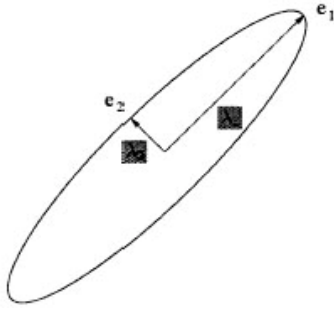
$$J_{xx} = \sum_{x', y' \in w} \hat{H}_x^2(x - x', y - y')$$

$$J_{xy} = \sum_{x', y' \in w} \hat{H}_x(x - x', y - y') \hat{H}_y(x - x', y - y')$$

$$J_{yy} = \sum_{x', y' \in w} \hat{H}_y^2(x - x', y - y') \quad ,$$

where H is a slice of image, and  $\hat{H}_x$  means the smoothed image slice with Gaussian kernel,

and  $w$  is a  $3 \times 3$  sub-window. It calculate the gradient orientation of an image region.



**Fig. 3.12** Tensor representation. The major axis of the ellipse estimates the orientation of local structure while the shape. [23]

In [24], Lowe develops an object recognition approach called SIFT (Scale-Invariant Feature Transform) to extract features which is robust under scaling condition. It first employs DoG (Difference of Gaussian ) filter to search features of the image which is stable across all possible scales in scale space, and then generates local descriptor to represent the local character of those features each. For each potential interest point, it pre-computes the gradient magnitude and orientation at the same scale using Eq. (2):

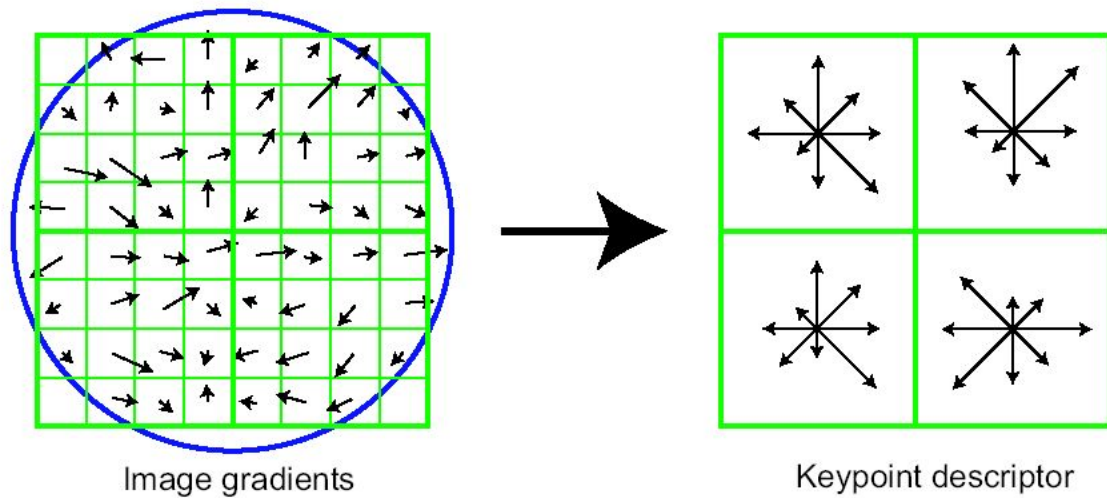
$$m(x, y) = \sqrt{(A_{xy} - A_{x+1,y})^2 + (A_{xy} - A_{x,y+1})^2}$$

$$\theta(x, y) = \tan^{-1} \frac{(A_{x,y+1} - A_{xy})}{(A_{xy} - A_{x+1,y})} , \quad (2)$$

where  $A$  means the intensity of the image.

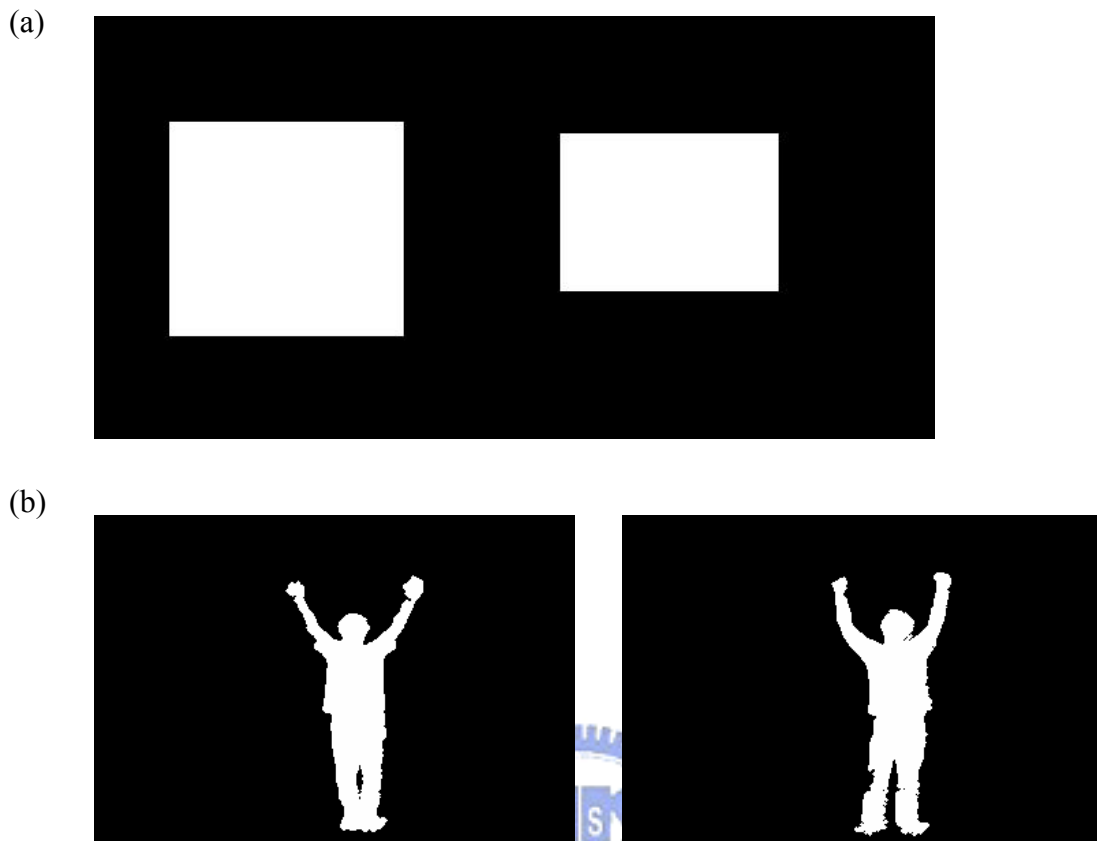
It builds orientation histogram with 36 bins from the region around the keypoint. Each gradient is weighted by its magnitude and a Gaussian circular window, and then is added to

the corresponding bin as demonstrate in Fig. 3.13.



**Fig. 3.13** Local descriptor of SIFT key. It uses orientation assignment to describe the local characteristic around the SIFT key.

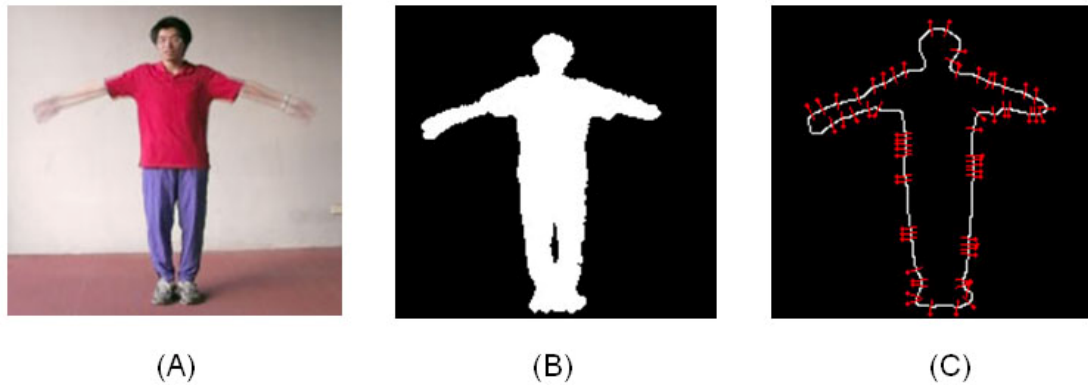
The reason why we mention this work here is that the feature descriptor of SIFT feature gathers local information in a small region. The image gradient orientation of a region forms the local character in the view of a higher scale according to their region structure, which is similar to human perception. In the perspective of neuroscience, people find that there exists complex neurons in primary visual cortex that respond to a gradient at a particular direction and spatial frequency, but the location of the gradient on the retina is allowed to shift over a small receptive field. It means that for human perception, we are sensitive to the existence of gradient for its orientation and spatial frequency, but insensitive to its position. It would result in a less tolerance to the shift of image gradients. For example, human eyes recognize rectangle as a collection of high frequency salient gradient which forms in the combination of lines and right angle corners such as Fig. 3.14 (a). Although there may be some shift of these gradients, it forms similar structure to human perception. Another example is shown in Fig. 3.14 (b), the shape of two persons with different physiques will give us similar impression if they perform similar posture.



**Fig. 3.14** (a) Two different rectangle. (b) Body silhouette of two persons with different physiques.

Therefore, we construct our feature as a local descriptor with image gradient orientation. For a given image, we first extract its silhouette by performing a background subtraction process, and then smooth the silhouette by a  $5 \times 5$  Gaussian filter to reduce the noise. Second, we calculate the gradient orientation around the whole foreground region. By applying the work of Ngo et al [23] and Lowe [24], we find that their methods performs similar characteristic to represent the gradient orientation in our implementations, so we choose Lowe's method (See Eq. (2)) which is more efficient as our approach to calculate the gradient orientation. Since we treat the whole foreground as an object, the magnitude part in Lowe's work doesn't mean a lot to us. We only adopt the gradient orientation part for our system. After doing this, we performs Canny edge detector to extract the verge of the silhouette. We

use the verge points as our interest point and compute local descriptor only on them.

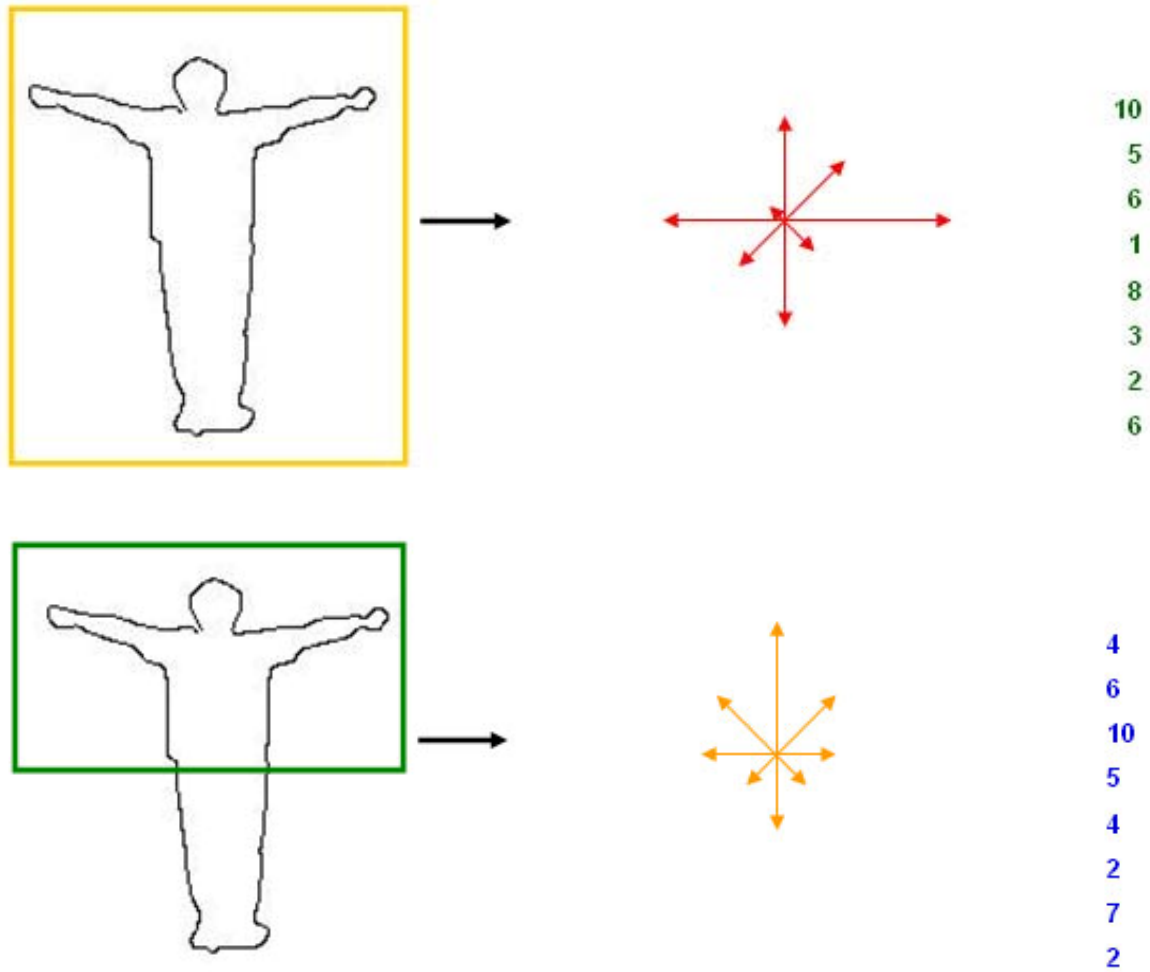


**Fig. 3.15** (A) Original image. (B) The extracted foreground of the image. (C) The gradient orientation along the verge point. We merely show some of the gradient orientation (red lines) to simplify the representation.



Since we only process the silhouette, the change of gradient is simple and monotonous. As a result, we can arrange our gradient orientation to 8 bins (from  $0$  to  $2\pi$  with an interval of  $\pi/4$ ). In the end, we represent our feature as a block, which accumulate the edge gradient orientation with 8 bins. We normalize the bin value from  $0$  to  $10$  according to the largest one for the purpose of efficient implementation in training stage. Fig. 3.16 shows the form of our feature.





**Fig. 3.16** The feature used in our approach. With an arbitrary rectangle block, we accumulate the edge gradient orientation with 8 bins within it, and generate a feature vector by the normalized bin value.

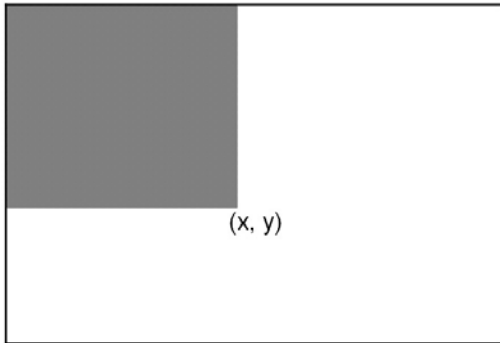
### 3.3 Training process

In our training stage, we use AdaBoost with modifications of feature selection to meet our requirement.

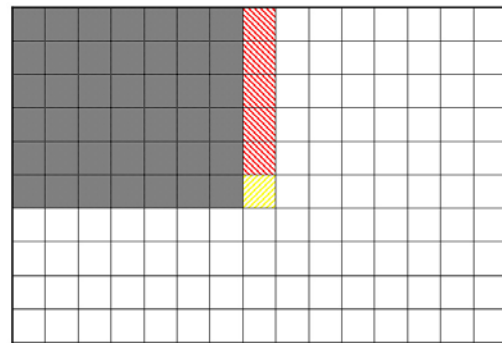
#### 3.3.1 Integral Gradient Orientation Map

In AdaBoost approach, since its feature shows the difference between the sum of pixels within two kinds of rectangle regions (black region for positive, and white region for

negative), it needs an efficient computation for fast summarizing pixels in a specific region. AdaBoost constructs an integral image map, which is the accumulation information of image pixels. For each point in the integral image map, it presents the accumulation of pixels in the block region from the origin (0, 0) to it, as shown in Fig. 3.17.



**Fig. 3.17** The value of the integral image map at (x, y) is the accumulation of pixel above and to the left.



**Fig. 3.18** The representation of integral image.



Fig. 3.18 shows the integral image map. The integral value of a point is summed by the integral value of its left point (black region) and the cumulative row sum of the point above it (red region). In our approach, the feature contains eight possible directions, and thus we build eight integral orientation maps following the above approach for each direction bin. After this process, for any point in the image, we can get the accumulated orientation map by the reference those eight maps. A formulation of AdaBoost is modified as follows:

$$OO_i(x, y) = \sum_{\substack{x' \leq x, y' \leq y, \\ O(x', y') = i}} O(x', y'), \quad i = 0..7$$

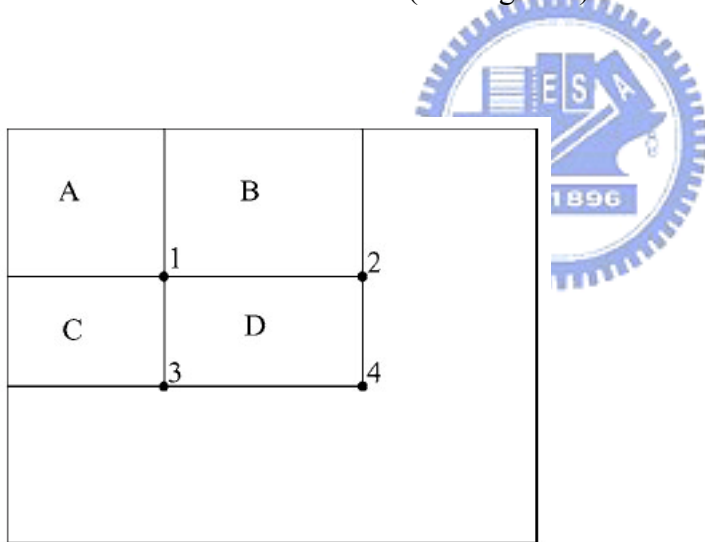
where  $O(x, y)$  is the gradient orientation of the point (x, y), and  $OO_i(x, y)$  is the integral orientation for bin i in that point. We calculate it using the following recurrences:

$$S_i(x, y) = \begin{cases} S_i(x, y-1) + O(x, y), & \text{if } O(x, y) = i \\ S_i(x, y-1) & , \text{ else} \end{cases} \quad (3)$$

$$OO_i(x, y) = OO_i(x-1, y) + S_i(x, y) \quad , \quad (4)$$

where  $S_i(x, y)$  is the cumulative row sum for bin  $i$ .  $S_i(x, -1) = 0$  and  $OO_i(-1, y) = 0$  for each bin  $i$ .

After the integral orientation map is built, we can easily get gradient orientation within a block by simply refer it. In the case of AdaBoost approach, since it contains several kind of feature, it may need six, eight or nine references for two-rectangle feature, three-rectangle feature or four rectangle feature. In our case, our feature is a block region, so we only need four references for each feature (See Fig.3.19).



**Fig. 3.19** The accumulated gradient orientation within a block can be computed with four references. If we want to know the feature information in D, we can simply get it by the computation of  $4+1-(2+3)$ .

### 3.3.2 Training Algorithm

After we construct our feature, we want to develop an approach to select best features from all possible ones. We call each feature trained by a weak learning algorithm a weak

learner. And the weak learning algorithm is built to choose a single weak learner which best separates positive and negative examples. After training, we can get an optimal threshold classification function which minimizes the number of false classification. Following AdaBoost approach, we construct our weak classifier as below:

$$h(x, f, p, \theta) = \begin{cases} 1, & \text{if } pf(x) < p\theta \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

where  $x$  is the image slice, and  $f$  is a feature,  $f(x)$  represents the response of  $x$  over feature  $f$ ,  $\theta$  is the adapted threshold choosing by the training algorithm, and  $p$  indicates the direction of the inequality.

The major approach of a boosting algorithm is that it chooses good classifiers by weighting. It first gives each training example a weight. For any selected classifier, it performs a re-weighting approach to give those examples which are correctly classified a decreasing weight, and the other false classified ones an increasing weight. The purpose of the re-weighting approach is to give those examples which are not correctly classified by the previous weak feature a more significant influence, so that they can be properly segmented in next round. Those classifiers that have minimum cost of weighted error will be chosen from the boosting algorithm (we will discuss the efficient implementation later). By combining weak classifiers together with confidence weight, a final strong classifier is generated. The AdaBoost training algorithm is shown below.

- Given example images  $(x_1, y_1), \dots, (x_n, y_n)$  where  $y_i = 0, 1$  for negative and positive examples respectively.
- Initialize weights  $w_{1,i} = 1/2m, 1/2l$  for  $y_i = 0, 1$  respectively, where  $m$  and  $l$  are the number of negatives and positives respectively.
- For  $t = 1, \dots, T$ : ( T hypothesis )

1. Normalize the weights,  $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$

2. Select the best weak classifier with respect to the weighted error

$$\varepsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$$

We will discuss it later for an efficient implementation.

3. Define  $h_t(x) = h(x, f_t, p_t, \theta_t)$  where  $f_t, p_t,$  and  $\theta_t$  which minimize  $\varepsilon_t$ .

4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where  $e_i = 0$  if example  $x_i$  is classified correctly,  $e_i = 1$  otherwise,

and  $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$ .

- The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{i=1}^T \alpha_i h_i(x) \geq \frac{1}{2} \sum_{i=1}^T \alpha_i \\ 0 & \text{otherwise} \end{cases}$$

Where  $\alpha_i = \log \frac{1}{\beta_i}$

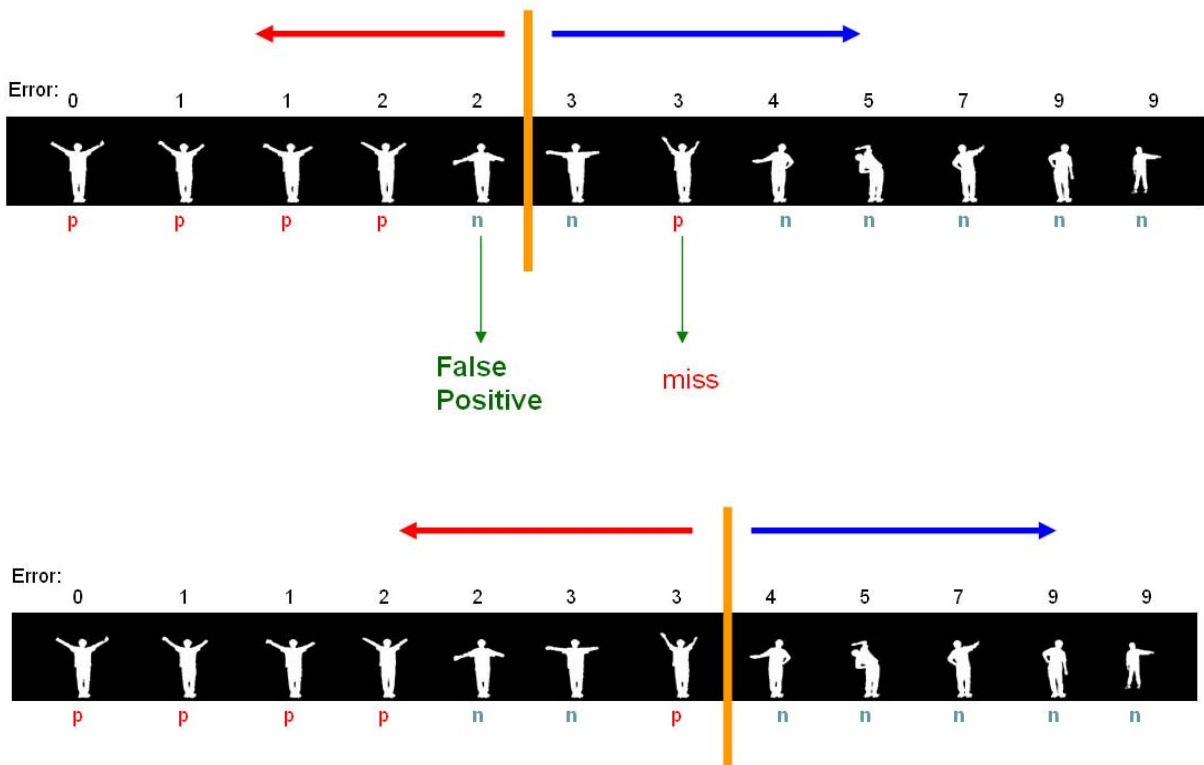
In our proposed approach, we directly assign the hypothesis T in the boosting algorithm as 8 to combine the whole block as a classifier since each feature contains 8 bins.

The weak classifier selection algorithm can be efficiently implemented as described in Viola and Jones's work. For each feature, we first sort the examples according to the value of feature response. Then the optimal threshold for this feature which can minimize the weighted error can be found in a single pass over the sorted list. For each example sorted list, we concern about four sums: (1)  $T^+$  for the total sum of positive example weight. (2)  $T^-$  for the total sum of negative examples. (3)  $S^+$  for the sum of positive weights below the current example. (4)  $S^-$  for the sum of negative weights below the current example.

It can be formulated as:

$$e = \min(S^+ + (T^- - S^-), S^- + (T^+ - S^+)) \quad , \quad (6)$$

where  $e$  is the weighted error in this adapted threshold. It means that we try to find the minimum weighted error between the weight of false positive and the weight of miss-detecting. For a given threshold in the example sorted list,  $S^+ + (T^- - S^-)$  means the sum of false positive weight and miss-detecting weight when we set the example with feature response larger than the threshold as positive. On the contrary,  $S^- + (T^+ - S^+)$  represents the sum of false positive weight and miss-detecting weight when we set the example with feature response smaller than the threshold as positive. The threshold of the classifier will be adapted according to minimum error cost between  $S^+ + (T^- - S^-)$  and  $S^- + (T^+ - S^+)$ . It decides the  $p$  value used in Eq. (5). And after the pass of tracking the sorted example list, we can find the optimal threshold for this feature. By selecting the minimum error of all features, we can find the best weak classifier at last. Fig. 3.20 is an example of selecting threshold for weak classifier.



**Fig. 3.20** Example of selecting threshold for weak classifier. We select the right part as positive because that it classifies the example well (small weighted error). And for different threshold, we can compare the sum of weight error of false positive and miss, so that we can select the optimal one.

The final weak classifier may presents as Fig. 3.21:

		Value	Threshold	p	Error weight
Bin 1	→	7	1	false	0.0118739205
Bin 2	↗	3	0	true	0.00413026278
		⋮			
		⋮			
		⋮			
Bin 8					

**Fig. 3.21** Weak Classifier Examples. It contains 8 bin values, thresholds for the feature tolerance, p for the direction of classifier, and error weight as the feature confidence.

### 3.3.3 Training Cascade of Classifiers

Based on the AdaBoost training approach, two methods can be used for building our system. First we use the classifier cascade method evolved by Viola and Jones. Their experiment results show that although the detecting accuracy between a monolithic feature classifier and a feature cascade of several small classifiers may not be significantly different, but the speed of cascade classifier is almost 10 times faster. The reason is that by it can perform a fast pruning approach which throws out most false targets cascading small set of classifiers. So we can adopt it to construct our system. The algorithm to generate a cascaded detector in [6] is shown below:

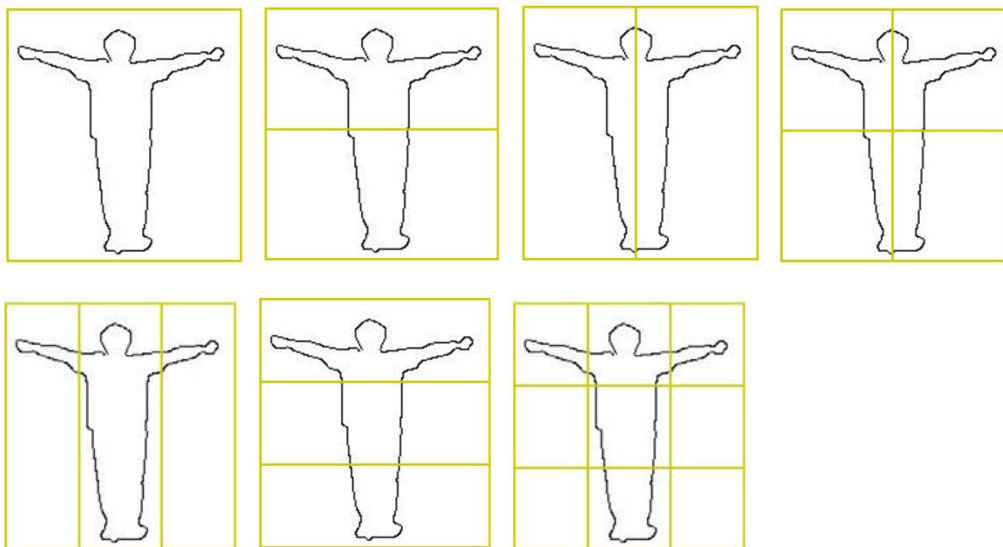
- User selects values for  $f$ , the maximum acceptable false positive rate per layer and  $d$ , the minimum acceptable detection rate per layer.
- User selects target overall false positive rate,  $F_{target}$ .
- $P$  = set of positive examples
- $N$  = set of negative examples
- $F_0 = 1.0$ ;  $D_0 = 1.0$
- $i = 0$
- while  $F_i > F_{target}$ 
  - $i \leftarrow i + 1$
  - $n_i = 0$ ;  $F_i = F_{i-1}$
  - while  $F_i > f \times F_{i-1}$ 
    - \*  $n_i \leftarrow n_i + 1$
    - \* Use  $P$  and  $N$  to train a classifier with  $n_i$  features using AdaBoost
    - \* Evaluate current cascaded classifier on validation set to determine  $F_i$  and  $D_i$ .
    - \* Decrease threshold for the  $i$ th classifier until the current cascaded classifier has a detection rate of at least  $d \times D_{i-1}$  (this also affects  $F_i$ )
  - $N \leftarrow \emptyset$
  - If  $F_i > F_{target}$  then evaluate the current cascaded detector on the set of non-face images and put any false detections into the set  $N$



By assigning a target value of false positive rate, the training algorithm trains classifier by AdaBoost training (see Table 1.), and recursive generates cascading classifiers if the false positive rate doesn't achieve our target.

### 3.3.4 Direct Feature Assignment

In the approach of section 3.3.4, we arbitrary choose features from all possible rectangle features to train the best classifier. The number of all possible rectangles is huge (about 160000 for Harr-rectangle feature, and about 20000 for our feature), and hence we must spent large time in selecting features. However, based on the describing ability of the edge gradient orientation feature, we can get lot of distinguishing features from some significant body structure, such as global body structure, above half body, bottom half body, etc, as shown in Fig. 3.22. Based on this property, we may efficiently eliminate large number of false candidates, and as a result we may save large amount of training time. We found in experiment that it not only performs good detecting accuracy as the cascade classifier method, but also the results have excellent performance in saving training time. We will show it in Section 4.



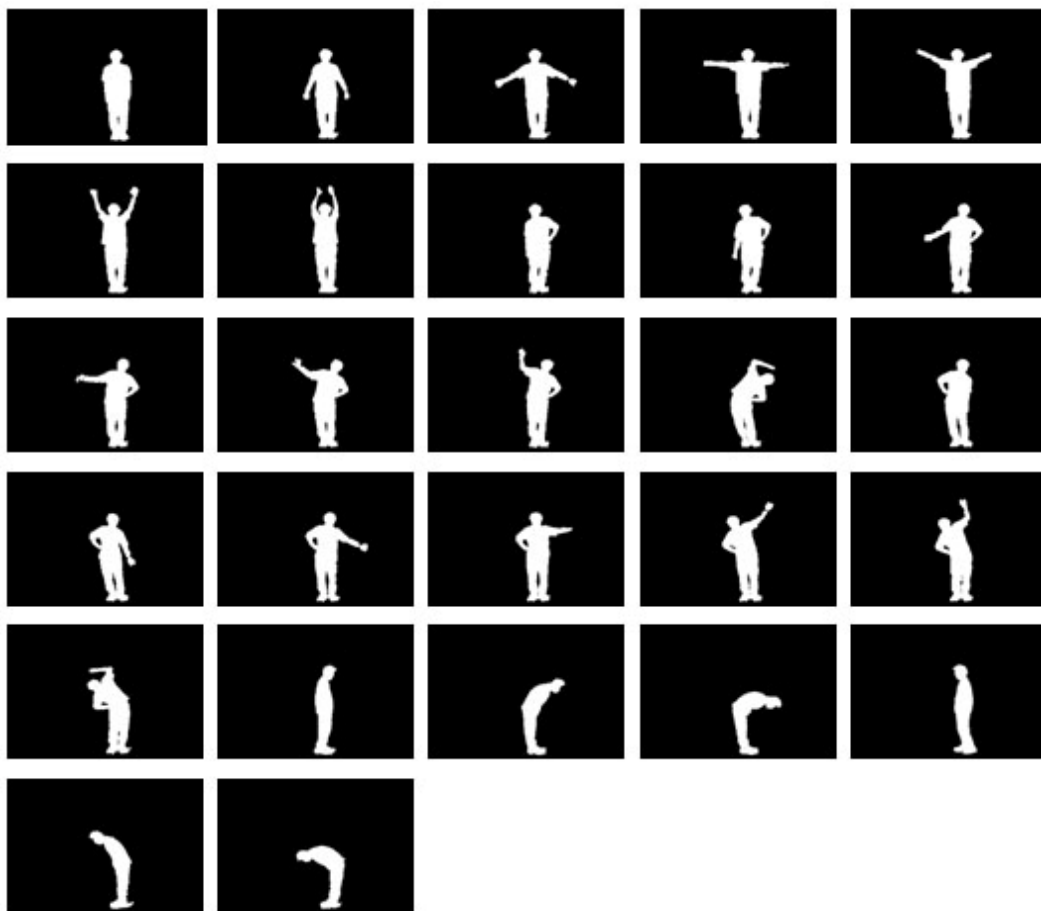
**Fig. 3.22** Example of pre-assigned feature approach. Features of those blocks will be trained by AdaBoost to get significant information for the body representation.

## 4 Experiment

### 4.1 Experimental result

We construct a data base with 27 postures of 5 motions. Each category of posture in the data base contains about 40 frames. And our experiment is based on a system of Intel P4 3GHz processor, 1GB Ram, Widows XP .

Our data base is shown as follows:



**Fig. 4.1** Human Posture data base in our experiment.

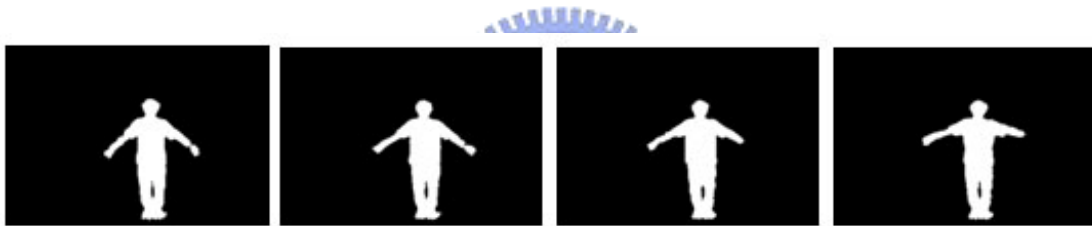
For each motion, we divide it into several postures according to the body structure. For example, a motion that raising both hands up to head is divided in seven postures according to

the raising degree as shown in Fig. 4.2.



**Fig. 4.2** Divide a motion as several postures.

And, for each class, the data base contains a small perturbation for the same posture as shown in. It is because that our posture is defined in a small region of motion and the motion of human may not always be the same. The training process will help us to give a tolerance for those small perturbations.



**Fig. 4.3** Example of perturbation in a posture class. We see that the position of the hand may be a little high or lower, but we still treat it as the same posture. Also some parts of silhouette may disappear because of the imperfect background subtraction, but if the distortion doesn't huge, we still see it as the same.

We evaluate the performance of the implementing system by a motion sequence which contains 1934 frames. By assigning the target false positive rate as about 0.2%, we compare the method that uses Harr-rectangle features and gradient-orientation approach, and their performances are shown in Table 1.

**Table 1.**

	Avg. Feature Number	Avg. Training Time	Detection Rate
Gradient orientation feature	About 32	4mins for each posture	96% ~ 99%
Harr rectangle feature	About 298	10mins for each posture	95% ~ 98%

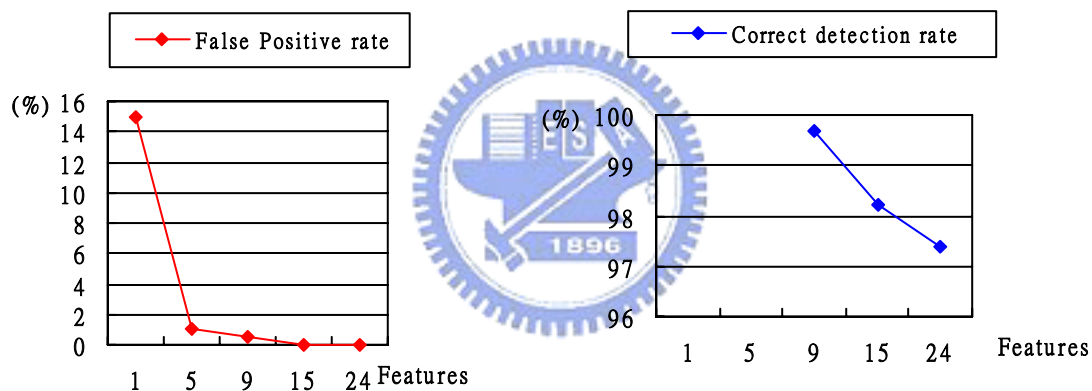
We can observe that they have similar detection rate, but for the training time, the gradient orientation method is much faster.

The experiment conducted is based on the approach of Section 3.3.3. The time for training all posture in data base is about 1.4 hours in gradient orientation approach and 4.5 hours in the Harr-rectangle approach. By applying the approach we demonstrated in Section 3.3.4, we found that we can achieve a higher performance by only using few significant features. Using 24 features shown in Fig. 3.22, it achieves about 97% ~ 99% in detection rate, and that is similar to the approach above. But the most important is that the cost of training time is only 5 second, that is extremely efficient than the previous approaches. But, we also find that the problem of this approach may happen when the data base turns huge. In that situation, those significant features may be not enough (although we can divide them more detailed), so that we still need to use the cascade classifier training approach.

## **4.2 Discussion**

In the experiments, we found that there exists trade-off between correct detecting rate and false positive rate. In our system, we build a small data base for each posture and expect to detect body silhouettes which belong to this category including those not in the data base.

Since it exists perturbations for each class of posture, for the purpose of not missing the detection of correct data we need tolerance in each classifier. The more we want to correctly detect to those not in data base (which means higher correctly detection rate), the more tolerance we will give for our classifier (which means high false positive rate ). On the contrary, if we want to precisely distinguish each class of posture from each other, we will have strict constraint in our classifier. It means that we may miss some correct detection for those not existing in our data base. The only solution is to adjust them till that both the correct detection rate and false positive rate are acceptable.



**Fig. 4.4** Trade-off between false positive rate and correct detection rate. When features increase, it means more strict constraint for the classifier, which will lower the false positive rate. But on the other hand, it decreases the tolerance, which result in a lower correct detection rate.

## 5. Difficulties

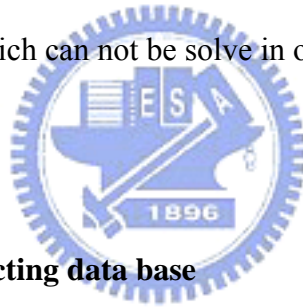
### 5.1 Background Subtraction

As mentioned in [1], the overall performance of the silhouette-based technique is largely depending on the accuracy of the foreground extracting method. Since that our approach is

built based on the silhouette information of human body, our system will be strongly influenced by the extracted foreground. Unfortunately, there still not an ideal method that can extract the foreground perfectly over all situations. This may be the most difficult problem to our system. Fig. 5.1 shows some example of incomplete foreground extraction, includes occlusion and shadow.



**Fig. 5.1** The situation of imperfect background subtraction. It contains shadow and incomplete body silhouette, which can not be solve in our system.



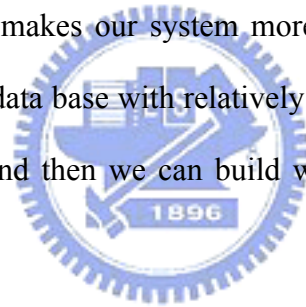
## **5.2 The difficulty in constructing data base**

Another problem in human posture recognition approaches is that we must enumerate all possible human postures. In normal environment, it is very difficult to construct a complete perfect posture data base because the problem of shadow, occlusion, lighting, dressing, etc. In our experiment, we only build a small set of posture data base, which is apparently not sufficient in a real world case. We may get help from some human-model animation software like Poser (See Fig. 5.2), which can generate all kind of possible human pose from animated human motion.



**Fig. 5.2** Poser: a software for animating and controlling human motion, and can synthesis the human body model as contour form.( E frontier America, Inc)

Another idea is that we may construct our system for the application of detecting abnormal motion, which may makes our system more feasible and useful. By generating a small set of abnormal posture data base with relatively large background posture data, we can construct the system easily. And then we can build with other approaches for an abnormal human motion analysis system.



### 5.3 Occlusion problem

The occlusion problem is similar to the problem of incomplete foreground extraction, which is hard to solve in our hypothesis. The training process generates a category classifier which represents the characteristic and structure to the class of posture. And for those silhouettes which suffer occlusion problem, the characteristic and structure change, so that they will be recognized as different kind of class. To solve it, we may use the experience of learning-based face detection approach. It solves the occlusion problem by building another data base which represent the class of incomplete face, and recognize them as a face. For example, for an occulted face which failed in the detection of a complete face detector, we perform another incomplete face detector to classify it, so that it may be detected. Along this

vein, we may construct an incomplete data base for each posture to generate an incomplete posture classifier so that we solve the problem. But, it will turn our problem more complicated.

## **6. Conclusion**

In this thesis, we have constructed a real-time human posture recognizing system. We have adapted the AdaBoost training approach and modified the feature with edge gradient orientation feature rather than Harr rectangle feature. Depending on the significant discriminating ability of our feature, the system recognized the silhouette information more efficient and more effective than the traditional AdaBoost approach. Additionally, our approach could directly construct a lot of distinguishing features from significant body structure rather than training all possible features, and thus extremely reduced the training time. From the experiment results, we have showed the efficiency and accuracy of our system. Although there still remained some difficulties, we have constructed practical system for real-time human posture recognition.



## References

- [1] Ismail Haritaoglu; David Harwood; Larry S. Davids; "W<sup>4</sup>: real-time surveillance of people and their activities ". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 22, No. 8, August 2000.
- [2] Junji Yamato; Jun Ohya; Kenichiro Ishii; "Recognizing human action in time-sequential images using hidden Markov model ". IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun 1992.
- [3] Aphrodite Galata; Neil Johnson; David Hogg; "Learning structured behaviour models using variable length Markov models ". IEEE International Workshop, pages 15-102, September 1999.
- [4] T. Moeslund and E. Granum, "A Survey of Computer Vision-based Human Motion Capture," *Computer Vision and Image Understanding*, Vol.81, no.3, pp.231-268, Mar. 2001.
- [5] B. Boulay, F. Bremond, and M. Thonnat, "Human Posture Recognition in Video Sequence," in Proc. *IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 23-29, 2003.
- [6] Paul Viola; Michael J. Jones; "Robust Real-Time Face Detection ". IEEE International Conference on Computer Vision, 2001.
- [7] Geogory Shakhnarovich; Paul Viola; Trevor Darrell; "Fast Pose Estimation with Parameter-Sensitive Hashing ". Proceedings of Ninth IEEE International Conference on Computer Vision, 2003.
- [8] Liu Ren; Gregory Shakhnarovich; Jessica K. Hodgins; Hanspeter Pfister; Paul Viola; "Learning Silhouette Features for Control of Human Motion ". ACM Transactions on Graphics, Vol.24(4), October, 2005.
- [9] J. K. Aggarwal; Q. Cai; "Human Motion Analysis : A Review ". IEEE Proceedings on

Nonrigid and Articulated Motion Workshop , pages:90 – 102. June 1997.

- [10] J. K. Aggarwal; Q. Cai; W. Liao; B. Sabata; “Articulated and elastic non-rigid motion: a review ”. IEEE Workshop on Motion of Non-Rigid and Articulated Objects , pages:2 – 14, Nov. 1994.
- [11] Z. Chen; H. J. Lee; “Knowledge-guided visual perception of 3D human gait from a single image sequence ”. IEEE Transactions on Systems, Man, and Cybernetics,22(2), pages:336-342, 1992.
- [12] M. K. Leung; Y. H. Yang; “First sight: A human body outline labeling system “. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 359-377, 1995.
- [13] D. Hogg. “Model-based vision: a program to see a walking person ”. Image and Vision Computing, pages 5-20, 1983.
- [14] R. F. Rashid. “Towards a system for the interpretation of moving light display ”. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 574-581, November 1980.
- [15] A. F. Bobick; J.Davids; “Real-time recognition of activity using temporal templates ”.IEEE Computer Society Workshop Applications on Computer Vision, pages 39-42, 1996.
- [16] R. Polana; R.Nelson; “Low level recognition of human motion (or how to get your man without finding his body parys) ”. IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects, pages 77-82, 1994.
- [17] Aphrodite Galata; Neil Johnson; David Hogg; “Learning Variable-Length Markov Models of Behavior”. IEEE Computer Vision and Image Understanding, pages 398-413, 2001.
- [18] Junghye Min; Rangachar Kasturi; “Activity Recognition Based on Multiple Motion Trajectories”. IEEE International Conference on Pattern Recognition, 2004.
- [19] Serge Belongie; Jitendra Malik; Jan Puzicha; “Shape Matching and Object Recognition

- Using Shape Contexts”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002.
- [20] Greg Mori; Jitendra Malik; “Estimating Human Body Configurations Using Shape Context Matching”. European Conference on Computer Vision, pages 666-680, 2002.
- [21] Jamie Shotton; Andrew Blake; Reberto Cipolla; “Contour-Based Learning for Object Detection ”. IEEE International Conference on Computer Vision, pages 503-510, 2005.
- [22] SJ Wang; LC Kuo; HH Jong; ZH Wu;” Representing images using points on image surfaces”. IEEE Transactions on Image Processing, VOL. 14, NO. 8, AUGUST 2005.
- [23] Chong-Wah Ngo; Ting-Chuen Pong; Hong-Jiang Zhang; “Motion Analysis and Segmentation Through Spatio-Temporal Slices Processing ”.IEEE Transactions on Image Processing, Vol. 12, NO 3, March 2003.
- [24] David. G Lowe; ” Object recognition from local scale-invariant features ”, IEEE International Conference on Computer Vision, pages: 1150-1157, 1999.
- [25] Correa, P.; Cxyz, J.; Umeda, T.; Marques, F.; Marichal, X.; Macq, B.; “Silhouette-based probabilistic 2D human motion estimation for real-time applications”, IEEE International Conference on Image Processing, pages 836-839 2005.
- [26] I. C. Chang and C. L. Huang, “The Model-based Human Body Motion Analysis System,” *Image Vision Computation*, Vol.18, no.14, pp.1067-1083, 2000.