國 立 交 通 大 學

電 控 工 程 研 究 所

碩 士 論 文

人體動作辨識之推論與取樣頻率研究

Inference and Down-sampling Rate study for Video-based Human Action Recognition

研 究 生：劉 育 誠

指 導 教 授：張 志 永

中 華 民 國 一百零一 年 七 月

人體動作辨識之推論與取樣頻率研究

Inference and Down-sampling Rate study for Video-based Human Action Recognition

學　　生：劉育誠　　Student : Yu-Cheng Liu

指導教授：張志永　　Advisor : Jyh-Yeong Chang

國立交通大學

電控工程研究所

碩士論文

A Thesis

Submitted to Department of Electrical Engineering

College of Electrical Engineering

National Chiao-Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master in

Electrical and Control Engineering

July 2012

Hsinchu, Taiwan, Republic of China

中 華 民 國 一百零一 年 七 月

人體動作辨識之推論與取樣頻率研究

學生:劉育誠　　指導教授: 張志永 博士

國立交通大學電機與控制工程研究所

摘要

人體動作辨識系統在電腦視覺領域一直是很熱門的研究與應用目標。在居家監控系統中最常見的方式是，使用固定式的攝影機，對室內的人物進行追蹤與動作辨識。為了達到即時監控之目標，處理的演算法必須快速，而且又必須能夠有效的分析影像。

在本論文中，動作辨識的目標是人體，為了更正確的擷取出人體部份，我們同時使用灰階域與 HSV 色彩空間，建立兩個背景模型，提升消除影像中陰影部分之影響，使得前後景之分離結果能夠更完整。取得即時影像，擷取出的前景部份，經過特徵空間轉換與標準空間轉換後，累積三張動作影像後，藉由預先學習而建立之模糊法則與時序動作姿態比對，完成人體動作之辨識。

研究對於較短周期的動作其取樣頻率改變是否獲得更多資訊，更多的訊息可以使人體動作辨識更加的準確，並且對判斷相同動作的規則，取其最大或者前三大、前五大、前七大和前九大相似度的動作法則平均值，藉由更多規則決定目前輸入的影像與判別動作之間的相似度，確能更加準確判斷人體動作。

Inference and Down-sampling Rate study for Video-based Human Action Recognition

STUDENT: Yu-Cheng Liu ADVISOR: Dr. Jyh-Yeong Chang

Institute of Electrical and Control Engineering
National Chiao-Tung University

ABSTRACT

Human activity recognition system is now a very popular subject for research and application. Using a fixed camera to track a person and recognize his (her) activity is widely seen in home surveillance. For real-time surveillance, the embedded algorithms must be efficient and fast to meet the real-time constraint.

In the thesis, we build two background models, one is grayscale another is HSV color space that extract the human region correctly, and we also reduce the shadowing effect. For better efficiency, the binary image is transformed to a new space by eigenspace and canonical space transformation. After that, we gathered three consecutive down-sampled images to recognize the human actions by fuzzy rules.

We utilize different down-sampling rate for short-period action to obtain more information which is useful for the human action recognition. Furthermore, we investigate to the average value of maximal top-3, top-5, top-7 and top-9 firing strength of rules with the same action to recognize the human action. Using more rules to determine the similarity between the inputs and rules that can be more accurately determine human action.

Acknowledgment

I would like to express my sincere gratitude to my advisor, Dr. Jyh-Yeong Chang for valuable suggestions, guidance, support and inspiration he provided. Without his advice, it is impossible to complete this research. Thanks are also given to all the people who assisted me in completing this research.

Finally, I would like to express my deepest gratitude to my family for their concern, supports and encouragements.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation of this research

Recognizing human actions in monocular video is an important scene understanding issue for applications [1] such as automatic surveillance, content-based video search, human-computer interaction, intelligent environment, and many others. Our society is becoming increasingly aging. Thus, home nursing is getting more and more important. However, the price of most home nursing care service is very expensive. Besides, the trained nurses are limited and they cannot take care of the elderly 24 hours a day. Therefore, automatic home caring system plays an important role to this trend. For example, when automatic surveillance system recognizes one's human activity is dangerous, the alarm will be triggered. Nevertheless, there is no well-defined structure which is effective to recognize the human actions to data. Therefore, this makes human action recognition become a challenging task.

A number of human action recognition methods have been proposed in the past few years. Carlsson and Sullivan *et al*. [6] proposed an action recognition method by shape matching to key frames from edge data which obtained from canny edge detection. Luke and Keller *et al.* [5] utilized using fuzzy logic to model human activities from voxel person. Cohen and Li [4] presented a 3D visual-hull constructed from a set of silhouettes to infer the human postures. W^4 [2] can detect single person or several persons in group by using an adaptive background model and identify the activities by finding the body components on the silhouette boundary. Bobick *et al*. [3], recognized human activities by comparing motion's energy and history.

In our research, we have designed a robust method that makes use of shape features to recognize the human actions. It is know that, when people do a specific action, which are composed of similar posture sequences in the time axis. Therefore, we can down-sample the frame sequence to recognize the human actions. Then, we use the 5:1, 3:1 and 2:1 down-sampling rate to classify the three consecutive key postures and then use the maximum, the average value of maximal top 3, top 5, top7 and top 9 firing of the action rule to recognize the human action.

The human action recognition system is composed of three modules. The first module is foreground extraction. The second module is the posture classification module which will transform the image data to a smaller dimension for computational and storage efficiency. Then the foreground image will be classified to the key postures of actions. The third module is the inference module which reasons the using the fuzzy rules to classify the three consecutive posture sequences to recognize the human action. The human action recognition system is showing in Fig. 1.1.

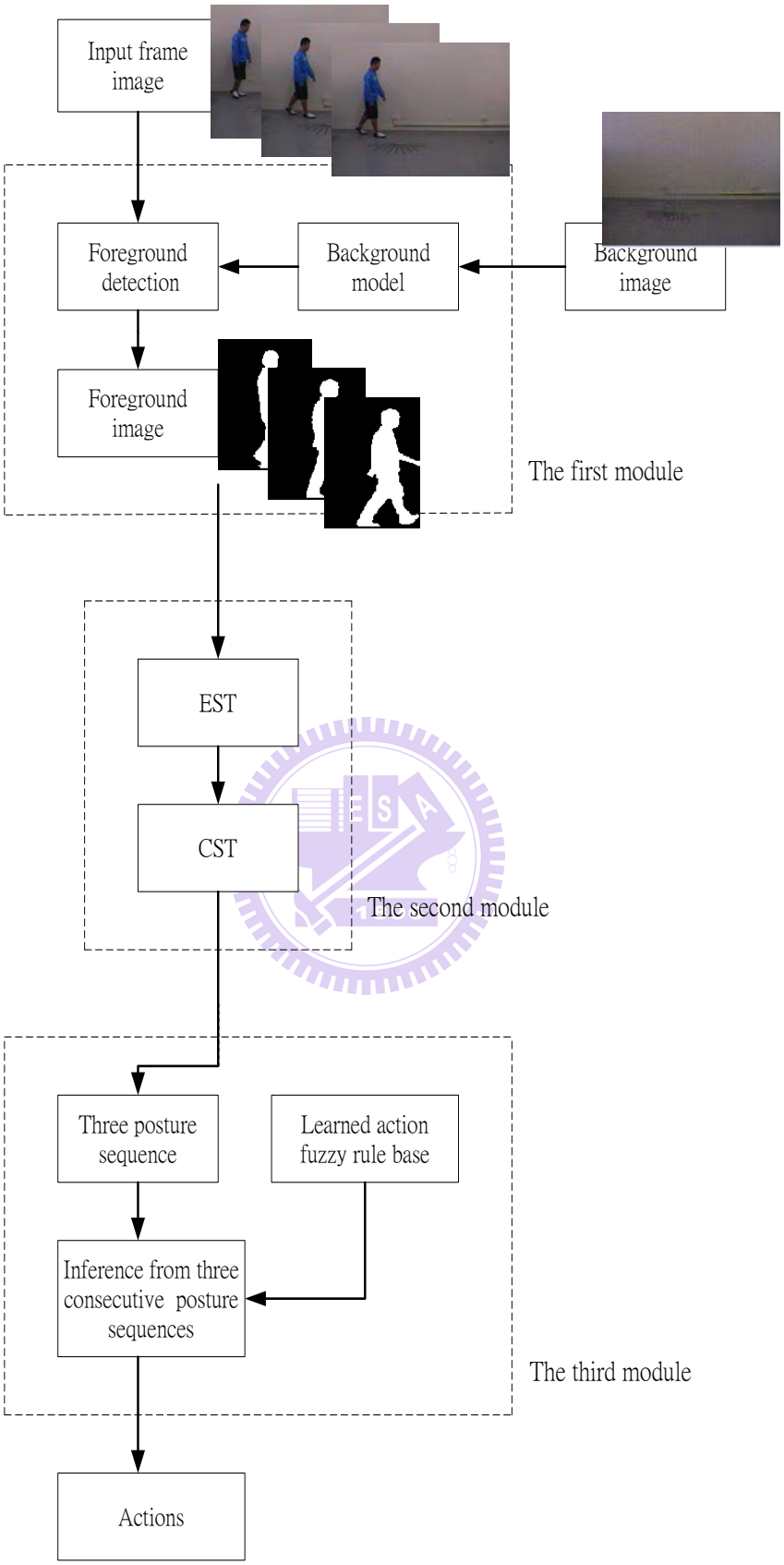


Fig. 1.1 Block diagram showing the human action recognition system.

1.2 Foreground extraction

The foreground subject extraction is the first step of human action recognition. We need to construct a background model. Background subtraction is a method typically used to segment moving regions in image sequences taken from a static camera by comparing each new frame to a model of the scene background [7]. There are many methods to build background models. In Piccardi [8], a review of the most relevant background subtraction methods were presented. W^4 [2] is such a popular example that using frame-difference with a threshold. In addition, foreground subject extraction is commonly affected by the additional inclusion of shadows. A lot of attempts have been developed to tackle the shadow suppression. Horprasert *et al.* [9] and Cucchiara *et al.* [10] utilized the rationale that shadows have similar chromaticity, but lower brightness than the background model. In our system, we build two background models. The first one is based on grayscale value and the second is based on HSV color space. In short, the background model should be adoptive to the represent the scene change. A subject enters scene and then leaves some things in the scene. After the subject leaves the scene, the background model will be updating accordingly.

After building the background model, we can extract the foreground subject from video frames. Subtracting each pixel of background model from current image frames will produce foreground subject. Then, the resulting image is converted to a binary image which contains the foreground subject by setting a threshold. Therefore, we can extract the most possible region of a person to a rectangle binary image. The rectangle image is resized to the specified resolution for normalization.

1.3 Eigenspace and Canonical Space Transformation

In most of video and image processing, the size of frame is usually very large and it usually contains a great deal of redundancy. The redundancy wastes the resources greatly in computation and storage aspects. Hence, some space transformations are introduced to reduce the redundancy of an image by reducing the data size of the image. The first step of redundancy reduction often transforms an image from a high-dimensional space into a low-dimension space. The transformation can use fewer dimensions to approximate the original image. There are many well-known transformation methods such as Fourier Transformation, Wavelet Transformation, Principal Component Analysis, Multi Dimensional Scaling (MDS) and Locally Linear Embedding (LLE). Our transformation method combines eigenspace transformation and canonical space transformation which are described as follows.

The Eigenspace Transformation (EST), which is based on Principal Component Analysis (PCA), has been demonstrated to be a potent scheme used for automatic face recognition [11], [12], gait analysis [13] and action recognition [14]. The subsequent transformation, Canonical Space Transformation (CST) based on Canonical Analysis, is used to reduce data dimensionality and to optimize the class separability and improve the classification performance. Unfortunately, CST approach needs long computation efforts when the image is large. Therefore, we combine EST and CST in order to improve the classification performance while reducing the dimension. Thus each image can be projected from a high-dimensional spatiotemporal space to a low-dimensional canonical space. In this low-dimension space the recognition of human activities becomes much simpler and easier.

1.4 Image Frame Classification and Activity Recognition

In this thesis, images are transformed into an image feature vector by extracting features from images. We extract image features by using eigenspace transformation and canonical space transformation. Because of the cameras usually capture image frames with a frequency of 30 frames per second. There is not much difference between two consecutive image frames. Thus, down-sampling the input image stream is necessary and that can reduce the computation load and complexity. We group three contiguous down-sampled images and transform them to three consecutive feature vectors. Then, the time-sequential images are converted to a posture sequence by using these three feature vectors. The posture sequence is signified by the number of the templates. In the learning phase, we build a transition model in terms of three consecutive posture sequences which are the category symbol of the posture template. For human action recognition, the model which best matches the observed posture sequence is chosen as the recognized action category.

One of the famous methods to model the time-sequential data's transition model is Hidden Markov Models (HMMs). The basic concept of Hidden Markov Models is described in [15]. HMMs have been used in speech recognition [16] and hand gestures recognition [17]. After transforming image frames to eigenspace and canonical space domain, we greatly reduce the image data size. We adopt the fuzzy rule-based techniques to classify human activities, not by the shape-based features of the images. Therefore our activity analysis system is tolerant of dissimilarity, uncertainty, ambiguity and irregularity which exist in the action video. Relevant articles using the fuzzy theory in action recognition are described as follows. Wang and Mendel [18] proposed that fuzzy rules to be generated by learning from examples.

Su [19] presented a fuzzy rule-based approach to spatio-temporal hand gesture recognition.

In our system, we propose a fuzzy rule-base approach for human activity recognition. Each action is represented in the form of fuzzy IF-THEN rules, extracted from the posture sequences of the training data. Each IF-THEN rule is fuzzified by employing an innovative membership function in order to represent the degree of the similarity between a time-based three posture pattern and the corresponding antecedent to infer the subject's action. When our system classifies an unknown action video, we match the three contiguous down-sampled images to the precedent part of each fuzzy rule. The rule's consequent with maximal accumulated similarity measure associated with these three consecutive postures defines the subject's activity type.

1.5 Thesis Outline

The thesis is organized as follows. In Chapter 2, we introduce the basic concepts concerning eigenspace transform, canonical space transform, and the HSV color space. In Chapter 3, we describe that utilizing different down-sampling rate to select the key postures and investigating to the average value of maximal top-3, top-5, top-7 and top-9 firing strength of rules with the same action to recognize the human action. In Chapter 4, the experiment results of our recognition systems are shown. At last, we conclude this thesis with a discussion in Chapter 5.

Chapter 2 Basic Concept

In this chapter, we briefly explain the basic concepts of eigenspace and canonical space transform. Then HSV color space concept is introduced.

2.1 Fundamentals of Eigenspace and Canonical Space Transform

In video and image processing, the dimensions of image data are often extremely large. It is common to transform the image from high-dimensional space into a low-dimension one to discover a small set of composite features for action recognition. There are many well-known transformation methods such as Fourier Transformation, Wavelet Transformation, Principal Component Analysis (PCA), Multi Dimensional Scaling (MDS) and Eigenspace Transformation (EST). However, PCA based on the global covariance matrix of the full set of image data is not very effective to the class structure existent in the data. In order to enhance the discriminatory power of various activity features, Etemad and Chellappa [20] introduced Linear Discriminant Analysis (LDA), also called canonical analysis (CA) [21], which can be used to optimize the posture class separability of different activity classes and improve the classification performance. The features are obtained through maximizing between-class and minimizing within-class variations. Here we call this approach canonical space transformation (CST). To benefit from these two transforms, combining EST based on PCA and CST based on CA. Therefore, our approach reduces the data dimensionality and optimizes the posture class separability of different activity classes.

Image data in high-dimensional space are converted into low-dimensional eigenspace using EST. Then, the obtained vector is further projected to a smaller canonical space using CST. Action Recognition is accomplished in the canonical space.

Assume that there are *c* training classes to be learned. Each class represents a specific posture, which assumes of testers various forms existing in the training image data. $\mathbf{x}'_{i,j}$ is the *j*-th image in class *i*, and N_i is the number of images in the *i*-th class. The total number of images in training set is $N_T = N_1 + N_2 + \cdots + N_c$. This training set can be written as

$$\left[\mathbf{x}'_{1,1}, \cdots, \mathbf{x}'_{1,N_1}, \cdots, \mathbf{x}'_{2,1}, \cdots, \mathbf{x}'_{c,N_c} \right] \tag{2.1}$$

where each $\mathbf{x}'_{i,j}$ is an image with n pixels.

At first, the intensity of each sample image is normalized by

$$\mathbf{x}_{i,j} = \frac{\mathbf{x}'_{i,j}}{\left\| \mathbf{x}'_{i,j} \right\|}. \tag{2.2}$$

Then, the mean pixel value for the training set is given by

$$\mathbf{m}_{\mathrm{x}} = \frac{1}{N_T} \sum_{i=1}^{c} \sum_{j=1}^{N_i} \mathbf{x}_{i,j}. \tag{2.3}$$

The training set can be rewritten as an $n \times N_T$ matrix **X**. And each image $\mathbf{x}_{i,j}$ forms a column of **X, t**hat is

$$\mathbf{X} = \left\lfloor \mathbf{x}_{1,1} - \mathbf{m}_{\mathrm{x}}, \cdots, \mathbf{x}_{1,N_1} - \mathbf{m}_{\mathrm{x}}, \cdots, \mathbf{x}_{c,N_c} - \mathbf{m}_{\mathrm{x}} \right\rfloor. \tag{2.4}$$

2.1.1 Eigenspace Transformation (EST)

Basically EST is widely used to reduce the dimensionality of an input space by mapping the data from a correlated high-dimensional space to an uncorrelated low-dimensional space while maintaining the minimum mean-square error to avoid information loss. EST uses the eigenvalues and eigenvectors generated by the data covariance matrix to retain the original data coordinates along the directions of maximal variance sequentially.

If the rank of the matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ is *K*, then *K* nonzero eigenvalues of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$, $\lambda_1, \lambda_2, \cdots, \lambda_K$, and their associated eigenvectors, $\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_K$, satisfy the fundamental relationship

$$\lambda_i \mathbf{e}_i = \mathbf{R}\mathbf{e}_i, \qquad i = 1, 2, \cdots, K \tag{2.5}$$

where $\mathbf{R} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$ and $\mathbf{R}$ is a square, symmetric $n \times n$ matrix. In order to solve Eq. (2.5), we need to calculate the eigenvalues and eigenvectors of the $n \times n$ matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$. But the dimensionality of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ is the image size, it is usually too large to be computed easily. Based on singular value decomposition, we can get the eigenvalues and eigenvectors by computing the matrix $\tilde{\mathbf{R}}$ instead, that is

$$\tilde{\mathbf{R}} = \mathbf{X}^{\mathrm{T}}\mathbf{X} \qquad \mathbf{X}: \text{ data matrix} \tag{2.6}$$

in which the matrix size of $\tilde{\mathbf{R}}$ is $N_T \times N_T$ which is much smaller than $n \times n$ of R. Then the matrix $\tilde{\mathbf{R}}$ still has *K* nonzero eigenvalues $\tilde{\lambda}_1, \tilde{\lambda}_2, \cdots, \tilde{\lambda}_K$ and *K* associated eigenvectors $\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \cdots, \tilde{\mathbf{e}}_K$ which are related to those in $\mathbf{R}$ by

$$\begin{cases} \lambda_i = \tilde{\lambda}_i \\ \mathbf{e}_i = \left(\tilde{\lambda}_i\right)^{-\frac{1}{2}} \mathbf{X}\, \tilde{\mathbf{e}}_i \end{cases} \qquad i = 1, 2, \cdots, K \tag{2.7}$$

These *K* eigenvectors are used as an orthogonal basis to span a new vector space. Each image can be projected to a point in this *K*-dimensional space. Based on the theory of PCA, each image can be approximated by taking only the largest eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_k|$, $k \leq K$, and their associated eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k$. This partial set of *k* eigenvectors spans an eigenspace in which $\mathbf{y}_{i,j}$ are the points that are the projections of the original images $\mathbf{x}_{i,j}$ by the equation

$$\mathbf{y}_{i,j} = \left[\, \mathbf{e}_1,\ \mathbf{e}_2, \cdots,\ \mathbf{e}_k \right]^{\mathrm{T}} \mathbf{x}_{i,j} \qquad i = 1, 2, ..., c \ ;\ j = 1, 2, ..., N_c \tag{2.8}$$

We called this matrix $\left[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k\right]^{\mathrm{T}}$ the eigenspace transformation matrix. After this transformation, each image $\mathbf{x}_{i,j}$ can be approximated by the linear combination of these *k* eigenvectors and $\mathbf{y}_{i,j}$ is a one-dimensional vector with *k* elements which are their associated coefficients.

2.1.2 Canonical Space Transformation (CST)

Based on canonical analysis in [22], we suppose that $\{\phi_1, \phi_2, \cdots, \phi_c\}$ represents the classes of transformed vectors by eigenspace transformation and $\mathbf{y}_{i,j}$ is the *j*-th vector in class *i*. The mean vector of entire set can be written as

$$\mathbf{m}_y = \frac{1}{N_T} \sum_i \sum_j \mathbf{y}_{i,j} \quad i = 1,\ 2, \ldots, c;\ j = 1,\ 2, \ldots, N_i \tag{2.9}$$

The mean vector of the *i*-th class can be presented by

$$\mathbf{m}_i = \frac{1}{N_i} \sum_{\mathbf{y}_{i,j} \in \Phi_i} \mathbf{y}_{i,j}. \tag{2.10}$$

Let $\mathbf{S}_{\mathrm{w}}$ denote the within-class matrix and $\mathbf{S}_{\mathrm{b}}$ denote the between-class matrix, then

$$\mathbf{S}_{\mathrm{w}} = \frac{1}{N_T} \sum_{i=1}^{c} \sum_{\mathbf{y}_{i,j} \in \varphi_i} \left(\mathbf{y}_{i,j} - \mathbf{m}_i\right)\left(\mathbf{y}_{i,j} - \mathbf{m}_i\right)^{\mathrm{T}}$$
$$\mathbf{S}_{\mathrm{b}} = \frac{1}{N_T} \sum_{i=1}^{c} N_i \left(\mathbf{m}_i - \mathbf{m}_y\right)\left(\mathbf{m}_i - \mathbf{m}_y\right)^{\mathrm{T}}$$

where $\mathbf{S}_{\mathrm{w}}$ represents the mean of within-class vectors distance and $\mathbf{S}_{\mathrm{b}}$ represents the mean of between-class distance vectors distance. The objective is to minimize $\mathbf{S}_{\mathrm{w}}$ and maximize $\mathbf{S}_{\mathrm{b}}$ simultaneously, which is known as the generalized Fisher linear discriminant function and is given by

$$\mathbf{J}(\mathbf{W}) = \frac{\mathbf{W}^{\mathrm{T}} \mathbf{S}_{\mathrm{b}} \mathbf{W}}{\mathbf{W}^{\mathrm{T}} \mathbf{S}_{\mathrm{w}} \mathbf{W}}. \tag{2.11}$$

The ratio of variances in the new space is maximized by the selection of feature transformation **W** if

$$\frac{\partial \mathbf{J}}{\partial \mathbf{W}} = 0. \tag{2.12}$$

Suppose that $\mathbf{W}^*$ is the optimal solution where the column vector $\mathbf{w}_i^*$ is a generated eigenvector corresponding to the *i*-th largest eigenvalues λ_i. According to the theory presented in [22], we can solve Eq. (2.12) as follows

$$\mathbf{S}_{\mathrm{b}} \mathbf{w}_i^* = \lambda_i \mathbf{S}_{\mathrm{w}} \mathbf{w}_i^*. \tag{2.13}$$

After solving (2.11), we will obtain c–1 nonzero eigenvalues and their corresponding eigenvectors $[v_1, v_2, \cdots, v_{c-1}]$ that create another orthogonal basis and span a (c–1)-dimensional canonical space. By using these bases, each point in eigenspace can be projected to another point in canonical space by

$$\mathbf{z}_{i,j} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{c-1}]^T \mathbf{y}_{i,j} \tag{2.14}$$

where $\mathbf{z}_{i,j}$ represents the new point and the orthogonal basis $[\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{c-1}]^T$ is called the canonical space transformation matrix. By merging equation (2.8) and (2.14), each image can be projected into a point in the new (c —1)-dimensional space by

$$\mathbf{z}_{i,j} = \mathbf{H}\,\mathbf{x}_{i,j}. \tag{2.15}$$

in which $\mathrm{H} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{c-1}]^T [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k]^T$.

2.2 The HSV color space

The HSV (hue, saturation and value) color space corresponds closely to the human perception of color. Conceptually, the HSV color space is a cone as shown in Fig. 2.1. Viewed from the circular side of the cone, the hues are represented by the angle of each color in the cone relative to the 0^o line, which is traditionally assigned to be red. The saturation is represent as the distance from the center of the circle. Highly saturation color are on the outer edge of the cone, whereas gray tones (which have no saturation) are at the very center. The value is determined by the colors vertical position in the cone. At the point end of the cone, there is no brightness, so all colors are blacks. At the fat end of the cone are the brightness colors.

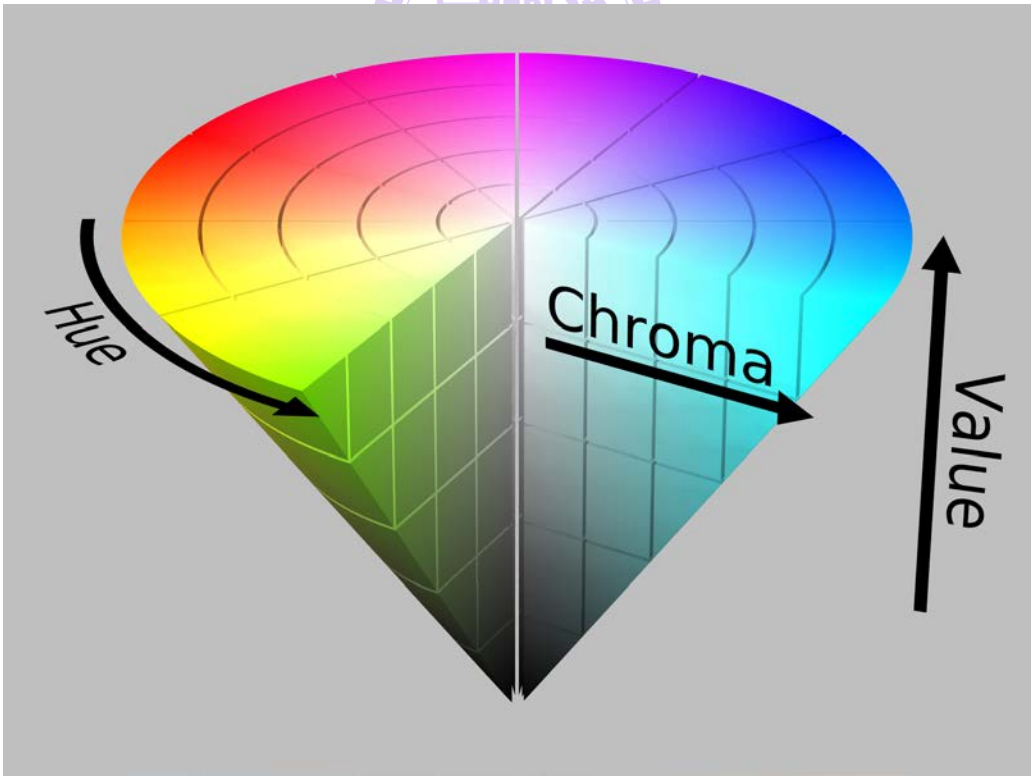


Fig. 2.1 The HSV Cone

The formula of RGB transfers to HSV is defined as below:

$$H = \begin{cases} 0°, & \text{if } max = min \\ 60° \times \dfrac{G-B}{max-min} + 0°, & \text{if } max = R \text{ and } G \geq B \\ 60° \times \dfrac{G-B}{max-min} + 360°, & \text{if } max = R \text{ and } G < B \\ 60° \times \dfrac{B-R}{max-min} + 120°, & \text{if } max = G \\ 60° \times \dfrac{R-G}{max-min} + 240°, & \text{if } max = B \end{cases}$$

$$S = \begin{cases} 0, & \text{if } max = 0 \\ \dfrac{max-min}{max}, & \text{otherwise} \end{cases}$$

$$V = max \tag{2.16}$$

where $max = \max(R, G, B)$ and $min = \min(R, G, B)$.

The hue parameter is the value which represents color information without brightness. Therefore, the hue is not affected by change of the illumination brightness and direction. Although hue is the most useful attribute, there are three problems in using hue attribute for color segmentation: 1) hue is meaningless when the intensity value is very low, 2) hue is unstable when the saturation is very low, and 3) saturation is meaningless when the intensity value is very low [18]. Accordingly, Ohba *et al*. [23] use three criteria (*intensity value*, *saturation*, and *hue*) to obtain the hue value reliably.

- **Intensity Threshold Value:**

  If $V < V_t$, then $H = 0$, where V, V_t, and H are an intensity value, the intensity threshold value, and a hue value, respectively. If measured color is not bright enough, the color is discarded. Then, the hue value is set to a predetermined value, i.e., 0.

- **Saturation Threshold Value:**

  If $S < S_t$, then $H = 0$, where S, S_t, and H are an saturation value, the saturation threshold value, and a hue value, respectively. Using this equation, measured color close to gray is discarded in the image.

- **Hue Threshold Value:**

  If $0 < H < H_t$ or, $2\pi - H_t < H < 2\pi$ then $H = 0$. The range of hue value is from 0 to 2π, and it has discontinuity at 0 and 2π. We use the phase threshold value ΔP_t to avoid the discontinuity effect.

Chapter 3 Human Activity Recognition System

The first step of human activity recognition system is foreground subject extraction. We need to construct a background model. There are many well-known methods to build background models. The most common one is that applying frame difference with a threshold. W^4 [2] is such a famous example with some modifications. It records the maximum and minimum grayscale values and the maximum inter-frame difference of each pixel in a background video. Then each pixel in the image frames subtracts the maximum and minimum grayscale values. If the pixel's absolute value of the subtraction operation is larger than the maximum inter-frame difference, the pixel is classified as the foreground. We cannot detect reliably those foreground pixel whose luminance component close to background pixel. In order to solve this problem, we build another background model in the HSV color space. The HSV color space corresponds closely to the human perception of color. We can have the luminance information and the chromatic information simultaneously. Hue is unreliable in some condition, so we use the three criteria (*intensity value*, *saturation*, and *hue*) described in Chapter 2 to obtain the hue value reliably.

3.1 Object Extraction

3.1.1 Background Model

A. Grayscale Value Background Model

In the grayscale value background model, each pixel of background scene is

characterized by three statistics: minimum grayscale value $n^{gray}(x,y)$, maximum grayscale value $m^{gray}(x,y)$ and maximum inter-frame difference $d^{gray}(x,y)$ of a background video. Because these three values are statistical, we need a background video without any moving objects, for background model training. Let *I* be an image frame sequence and contains *N* consecutive images. $I_i^{gray}(x,y)$ is the grayscale value of a pixel which is located at (x,y) in the *i*-th frame of *I*. The grayscale value for background model, $[m^{gray}(x,y), n^{gray}(x,y), d^{gray}(x,y)]$, of a pixel is obtained by

$$\begin{bmatrix} m^{gray}(x,y) \\ n^{gray}(x,y) \\ d^{gray}(x,y) \end{bmatrix} = \begin{bmatrix} \max_i \left\{ I_i^{gray}(x,y) \right\} \\ \min_i \left\{ I_i^{gray}(x,y) \right\} \\ \max_i \left\{ \left| I_i^{gray}(x,y) - I_{i-1}^{gray}(x,y) \right| \right\} \end{bmatrix} \tag{3.1}$$

where $i = 1, 2, \ldots, N$.

B. HSV Color Space Background Model

We build another background model with the minimum value ($[n^H(x,y), n^S(x,y), n^V(x,y)]$) and maximum value ($[m^H(x,y), m^S(x,y), m^V(x,y)]$) in each HSV domain. Then, we also record the inter-frame ratio in the brightness information and the inter-frame different in the chromatic information. Similarly, we use the same background video to build another background model. Suppose the observed image frame sequence that contains *N* consecutive images. $I_i^H(x,y)$ is the pixel's hue value at (x,y) of the *i*-th image frame. $I_i^S(x,y)$ is the pixel's saturation value at (x,y) of the *i*-th image frame. $I_i^V(x,y)$ is the pixel's brightness value at (x,y) of the *i*-th image frame. The background model of a pixel is obtained by

$$\begin{bmatrix} m^H(x,y) \\ n^H(x,y) \\ d^H(x,y) \end{bmatrix} = \begin{bmatrix} \max_i \{I_i^H(x,y)\} \\ \min_i \{I_i^H(x,y)\} \\ \max_i \{ |I_i^H(x,y) - I_{i-1}^H(x,y)| \} \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} m^S(x,y) \\ n^S(x,y) \\ d^S(x,y) \end{bmatrix} = \begin{bmatrix} \max_i \{I_i^S(x,y)\} \\ \min_i \{I_i^S(x,y)\} \\ \max_i \{ |I_i^S(x,y) - I_{i-1}^S(x,y)| \} \end{bmatrix} \tag{3.3}$$

$$\begin{bmatrix} m^V(x,y) \\ n^V(x,y) \\ d^V(x,y) \end{bmatrix} = \begin{cases} \begin{bmatrix} \max_i \{I_i^V(x,y)\} \\ \min_i \{I_i^V(x,y)\} \\ \max_i \{ |I_i^V(x,y) / I_{i-1}^V(x,y)| \} \end{bmatrix} & \text{if } I_i^V(x,y) / I_{i-1}^V(x,y) \geq 1 \\ \begin{bmatrix} \max_i \{I_i^V(x,y)\} \\ \min_i \{I_i^V(x,y)\} \\ \max_i \{ |I_{i-1}^V(x,y) / I_i^V(x,y)| \} \end{bmatrix} & \text{if } I_i^V(x,y) / I_{i-1}^V(x,y) < 1 \end{cases} \tag{3.4}$$

where $i = 1, 2, ..., N$

3.1.2 Foreground Object Extraction

Fig. 3.1 shows the framework we apply to foreground subject extraction. Our framework of foreground subject extraction is composed of four components. The first component is foreground subject extraction in the grayscale value and the HSV color space background models. The second component is the shadow suppression. The third component is the object segmentation. And the finally component is the foreground image compensation to recover the foreground pixels those are wrongly

classified to the background.

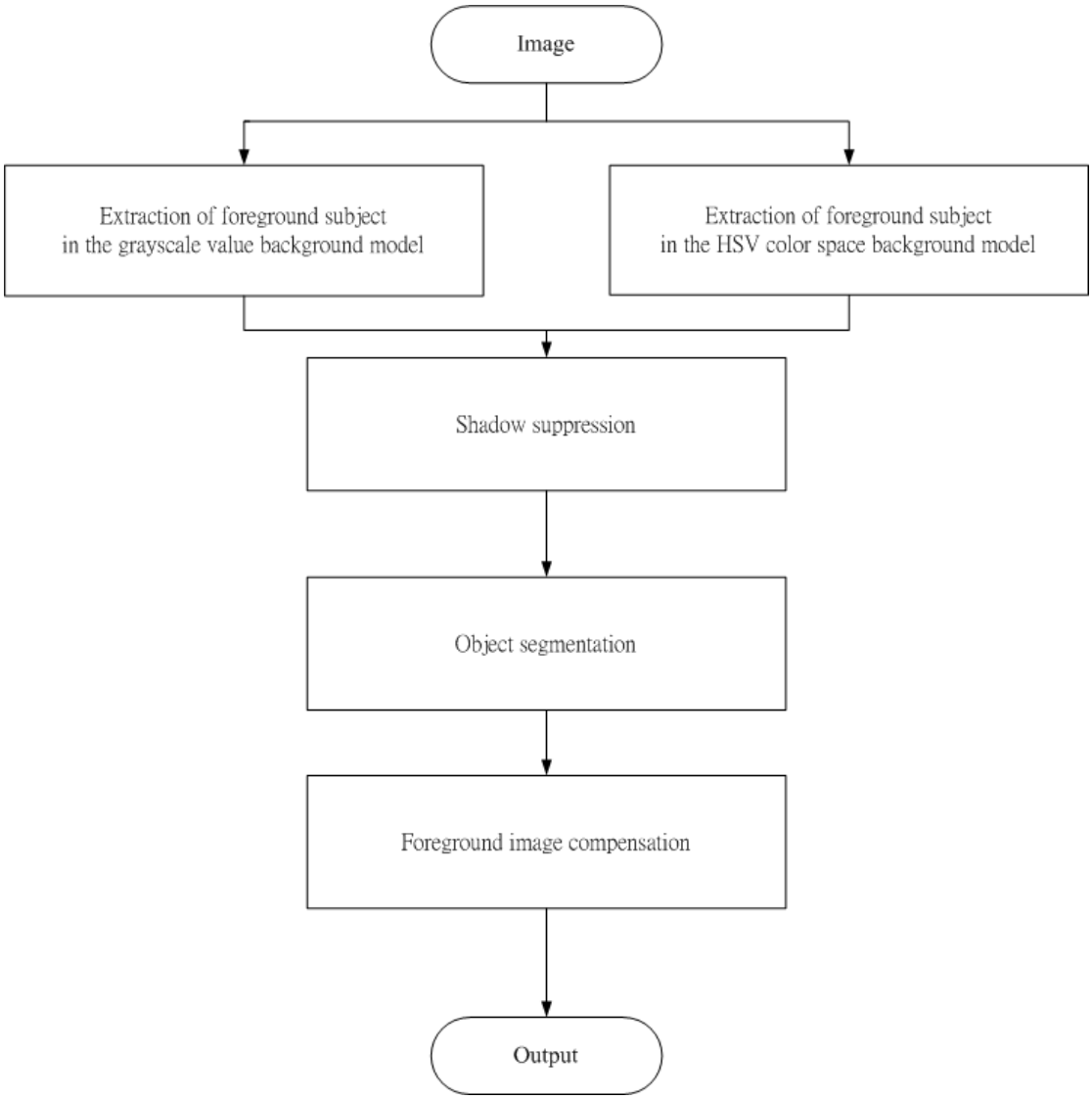


Fig. 3.1 The framework we apply to foreground subject extraction.

Foreground objects can be segmented from every frame of the video stream. Each pixel of the video frame is classified to either a background or a foreground pixel by the difference between the background model and a captured image frame. First, we utilize the maximum grayscale value $m^{gray}(x, y)$, minimum grayscale value $n(x, y)$ and maximum inter-frame difference $d^{gray}(x, y)$ of the grayscale value background

model to segment a foreground by

$$I^{1}{}_{foreground}(x,y)=\begin{cases}0, & \text{if } I_i^t(x,y)<(m^{gray}(x,y)+k\mu)\\ & \text{and } I_i^t(x,y)>(n^{gray}(x,y)-k\mu)\\ 255, & \text{otherwise}\end{cases} \tag{3.5}$$

where $I_i^t(x,y)$ is the intensity of a pixel which is located at (x,y), $I^{1}{}_{foreground}(x,y)$ is the gray level of a pixel in binary image, μ is the median of all $d^{gray}(x,y)$, and k is a threshold. Threshold k is determined by experiments according to different environments. The value of k affects the amount of information retained in binary image $I^{1}{}_{foreground}(x,y)$.

On the other hand, we utilize the maximum value $m^V(x,y)$, minimum value $n^V(x,y)$ and maximum inter-frame value ratio $d^V(x,y)$ of the HSV color space background model to segment the foreground pixel by

$$I^{2}{}_{foreground}(x,y)=\begin{cases}0, & \text{if } I_i^V(x,y)/m^V(x,y)<k_V d^V(x,y)\\ & \text{or } I_i^V(x,y)/n^V(x,y)<k_V d^V(x,y)\\ 255, & \text{otherwise}\end{cases} \tag{3.6}$$

where $I_i^V(x,y)$ is the intensity of a pixel which is located at (x,y), $I^{2}{}_{foreground}(x,y)$ is the gray level of a pixel in a binary image, k_V is a threshold, determined by light sufficiency of the scene. k_V will be reduced for in-sufficient light condition and increased otherwise.

3.1.3 Shadow Suppression

The pixels of the moving shadows are easily detected as the foreground pixel in normal condition. Because the shadow pixels and the object pixels share the important visual feature: motion model. For this reason, the moving shadows cause object merging and object shape distortion. Therefore, we need to remove the shadow by using the shadow filter. We assume that the observed intensity of shadow pixels is directly proportional to incident light. Consequently, shadowed pixels are scaled versions (darker) of corresponding pixels in the background model [24].

In the first place, we build the shadow filter in the grayscale value. Let $B(x, y)$ be the background image formed by temporal median filtering, and $I(x, y)$ be an image of the video sequence. For each pixel (x, y) belonging to the foreground, consider a 3×3 template T_{xy} such that $T_{xy}(m,n) = I(x+m, y+n)$, for $-1 \le m \le 1, -1 \le n \le 1$ (i.e. T_{xy} corresponds to a neighborhood of pixel (x, y)). Then, the NCC between template T_{xy} and image B at pixel (x, y) is given by:

$$NCC(x,y) = \frac{ER(x,y)}{E_B(x,y)E_{T_{xy}}} \tag{3.7}$$

where

$$ER(x,y) = \sum_{n=-1}^{1}\sum_{m=-1}^{1} B(x+m, y+n)T_{xy}(m,n)$$

$$E_B(x,y) = \sqrt{\sum_{n=-1}^{1}\sum_{m=-1}^{1} B(x+m, y+n)^2} \tag{3.8}$$

$$E_{T_{xy}} = \sqrt{\sum_{n=-1}^{1}\sum_{m=-1}^{1} T_{xy}(m,n)^2}$$

If a pixel (x, y) is in a shadowed region, the NCC in a neighboring region T_{xy} should be large, and the energy $E_{T_{xy}}$ of this region should be lower than the energy $E_B(x, y)$ of the corresponding region in the background images. There, we get

$$S^1(x,y) = \begin{cases} \text{shadow}, & NCC(x,y) \geq L_{ncc} \text{ and } E_{T_{xy}} < E_B(x,y) \\ \text{foreground}, & \text{otherwise} \end{cases} \tag{3.9}$$

where $S^1(x, y)$ is the binary image, and L_{ncc} is a fixed threshold. If L_{ncc} is low, several foreground pixels may be misclassified as shadow pixels. On the other hand, selecting a large value of L_{ncc}, then the shadow pixels may not be detected.

We know that the shadow pixels have similar chromaticity but lower brightness than the background model. Therefore, we can detect the shadow in the HSV color space. We analyze the points which are possible moving object that are detected above. Building another shadow filter S^2 for each (x, y) point as follows:

$$S^2(x,y) = \begin{cases} \text{shadow}, & \text{if } I_i^V(x,y) - n^V(x,y) < 0 \\ & \text{and} \left| I_i^H(x,y) - m^H(x,y) \right| < k_H d^H(x,y) \\ & \text{and} \left| I_i^S(x,y) - m^S(x,y) \right| < k_S d^S(x,y) \\ \text{foreground}, & \text{otherwise} \end{cases} \tag{3.10}$$

where $I_i^H(x, y)$, $I_i^S(x, y)$, and $I_i^V(x, y)$ are respectively the HSV channel of a pixel located at (x, y), and $S^2(x, y)$ is one of the shadow filter to class the pixel in the moving shadow. Values k_S and k_H are selected threshold values that used to measure the similarities of the hue and saturation between the background image and the current observed image.

We extract the foreground objects from the two background models. Setting a hard threshold for each background model, we obtain the foreground objects which have less noise, but missing some foreground objects. Therefore, using the union is better than the intersection. Because of using the union can increase the foreground with less noise. Finally, the foreground subject is defined as:

$$I_{foreground}(x,y) = S^1(x,y) \vee S^2(x,y) \tag{3.11}$$

3.1.4 Object Segmentation

According to the binary image $I_{foreground}$ segmented by above, we extract the region of foreground object as minimum as possible. Foreground region extraction can use a simply method by setting a threshold on the histograms in X-axis and Y-axis. Fig. 3.2 shows an example of foreground region extraction. We utilize the binary image that the binary image is projected on X-axis and Y-axis. The region we interest in that has higher counts in the histogram. We obtain the boundary coordinates x_1, x_2 of X-axis and y_1, y_2 of Y-axis from the projection histogram. We can use these boundary coordinates as corners of a rectangle to extract foreground region.

3.1.5 Foreground Image Compensation

It is difficult to detect all the foreground pixels and remove all the shadows in the same time. When we want to remove shadow pixels, some foreground information

will be lost and that makes the foreground image be broken. In order to solve the problem, we will repair the foreground image by opening filter and closing filter.

After the four components, we extract the foreground objects. The rectangular image which contain foreground objects will be normalized to 128×96. Fig. 3.3 is the extracted foreground region.

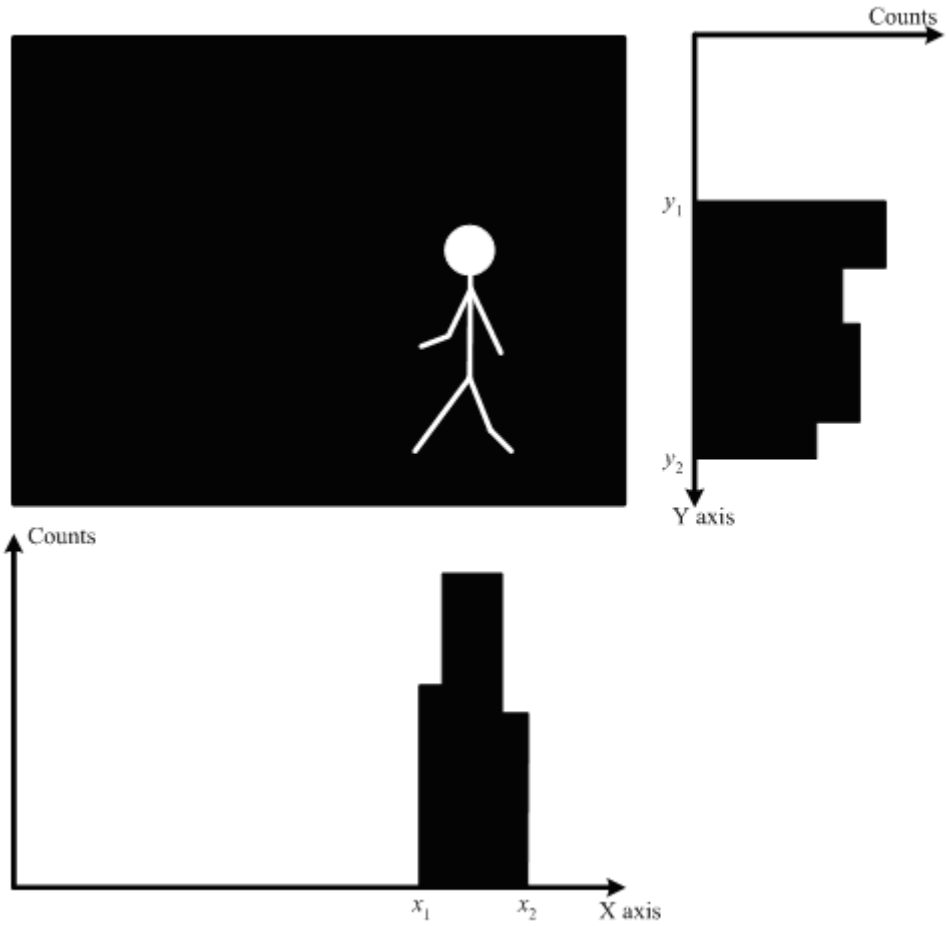


Fig. 3.2 The binary image is projected on X-axis and Y-axis.

Fig. 3.3 The binary image of extracted foreground region.

3.2 Background Update

If we move indoor facilities, they will be detected as foreground pixels and the activity recognition will be misclassified. Therefore, we have to update background models in order to avoid above state occurring. Background models will be updated if the video does not vary for a long time and there is nobody in the scene. By Eq. (3.12), we calculate how many times the binary values are unchanged.

$$update(x,y)=\begin{cases} update(x,y)+1, & \text{if } I^{t-1}_{foreground}(x,y)=I^{t}_{foreground}(x,y) \\ update(x,y), & \text{otherwise} \end{cases} \tag{3.12}$$

where $I^{t}_{foreground}(x,y)$ is the gray level of a pixel in binary image and it is located at (x,y). Value $update(x,y)$ is a record of how many times $I^{t}_{foreground}(x,y)$ remains unchanged.

By skin color detection, we can discriminate that there are someone or not. First, the input image is transfer to the normalized RGB color space by:

$$r=\frac{R}{R+G+B} \tag{3.13}$$

$$g=\frac{G}{R+G+B} \tag{3.14}$$

According to Soriano and Martinkauppi [25], the boundary of skin tone in the r-g plane is defined as follow:

$$f_{upper}(r) = -1.3767r^2 + 1.0743r + 0.1452 \quad (3.15)$$

$$f_{lower}(r) = -0.7760r^2 + 0.5601r + 0.1766 \quad (3.16)$$

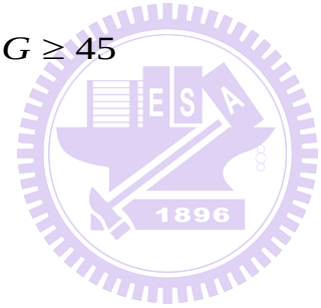
If a pixel satisfies the following four conditions, it will be labeled as skin pixel. Therefore, we know there is a person or not.

$$g > f_{lower}(r) \text{ and } g < f_{upper}(r) \quad (3.17)$$

$$(r - 0.33)^2 + (g - 0.33)^2 \geq 0.0004 \quad (3.18)$$

$$R > G > B \quad (3.19)$$

$$R - G \geq 45 \quad (3.20)$$



3.3 Down-sampling the Video Stream

Physical constraints on the architecture of human bodies' induce rhythmic and repetitive patterns of motion limited within a certain frequency. Because cameras usually capture image frames in high frequency, i.e., 30 frames /sec. There is almost no difference between two consecutive image frames for a normal recording of 30 frames per second. Hence, we can down-sample the video frame instead of using all the 30 frames captured in a second. Down-sampling can also reduce the intensive computation and memory load. It is difficult to select the key posture as a result of different actions with different cycles.

The action cycle is defined as backing to repeat the same attitude time. In our daily life, some of the most universal and frequently performed actions are walking, jog and running. According to the statistics shown in KTH datasets, the mean of the walking cycle is 1.1s, jog cycle is 0.86s and running cycle is 0.67s. We observed running is short-period and walking is long-period. Fig. 3.4 is the walking video sequences. Fig. 3.5 is the jog video sequences, and Fig. 3.6 is the running video sequences.

Fig. 3.4 The walking video sequences.

Fig. 3.5 The jog video sequences.

Fig. 3.6 The running video sequences.

In our research, we compare using the same down-sampling rate with using the different down-sampling rate for different actions. Because of utilizing the 5:1 down-sampling rate for short-period actions, the key postures may be the same that cannot show the short-period action. Therefore, we will see whether down-sampling rate affects the human action recognition or not. The first method, we utilize 5:1 down-sampling rate for all actions to select the essential template image. The second method, we utilize 5:1 down-sampling rate for walking, 3:1 down-sampling rate for jog and 2:1 down-sampling rate for running to select the essential template image. Fig. 3.7 shows using 5:1 down-sampling rate to select the essential template images.

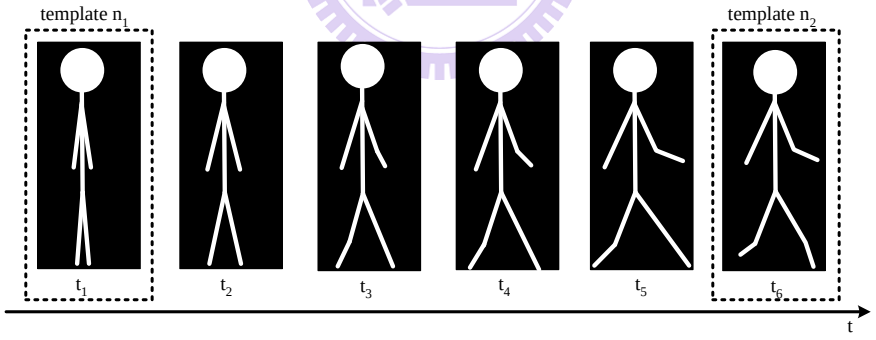


Fig. 3.7 Using 5:1 down-sampling rate to select the essential template images.

These essential templates are transformed to a new space by eigenspace transformation (EST) and canonical space transformation (CST). The approximation will lose slight information of image with little differences, but it can decrease massive data dimensions. However, two similar image frames will converge to two

near points after eigenspace and canonical space transformation. The images of similar postures done by difference people also barely converge to one point. Consequently, we select only essential templates rather than use all sequences for human activity recognition.

Combining both EST and CST, each image frame is transformed to a (c–1)-dimensional vector [26]. Assume that there are n training models and c clusters in the system. Therefore, we have N_t templates, where N_t is equal to n multiplied by c. Let $\mathbf{g}_{i,j}$ be a vector of template image of the j-th training model and the i-th category and $\mathbf{t}_{i,j}$ be the transformed vector of $\mathbf{g}_{i,j}$. $\mathbf{t}_{i,j}$ is computed by

$$\mathbf{t}_{i,j} = \mathbf{H} \cdot \mathbf{g}_{i,j}, \qquad i = 1, 2, \cdots, c; \quad j = 1, 2, \cdots, n \tag{3.21}$$

where **H** denotes the transformation matrix which combine EST with CST and n is the total number of posture images in the i-th cluster. $\mathbf{t}_{i,j}$ is a (c–1)-dimensional vector and each dimension is supposed to be independent. Hence, $\mathbf{t}_{i,j}$ is rewritten as

$$\mathbf{t}_{i,j} = \left[t_{i,j}^{1}, t_{i,j}^{2}, \cdots, t_{i,j}^{c-1}\right]^{\mathrm{T}} \tag{3.22}$$

The transformation of each training model's templates is treated as a mean vector. That is,

$$\boldsymbol{\mu}_i = \frac{1}{n}\sum_{j=1}^{n} \mathbf{t}_{i,j} \tag{3.23}$$

where *i* is the number of template categories.

The standard deviation vector of the *m*-th dimension is computed by

$$\sigma_m = \sqrt{\frac{\sum_{i=1}^{c}\sum_{j=1}^{n}\left(t_{i,j}^{m} - \mu_i^m\right)^2}{N_t - 1}} \tag{3.24}$$

where $m = 1,\ 2, \ldots, c-1$.

3.4 Construction of Fuzzy Rules from Video Stream

For human activity classification, temporal relationships of postures in video sequence are important information. Human's actions may have similar postures in two different activity sequences. Therefore, only one image frame is utilized to classify the action that is prone to wrong. For example, the actions of "jumping" and "crouching" both have the same postures called common states as shown in Fig. 3.8. Besides, the posture sequence of each activity is dissimilar in different people.

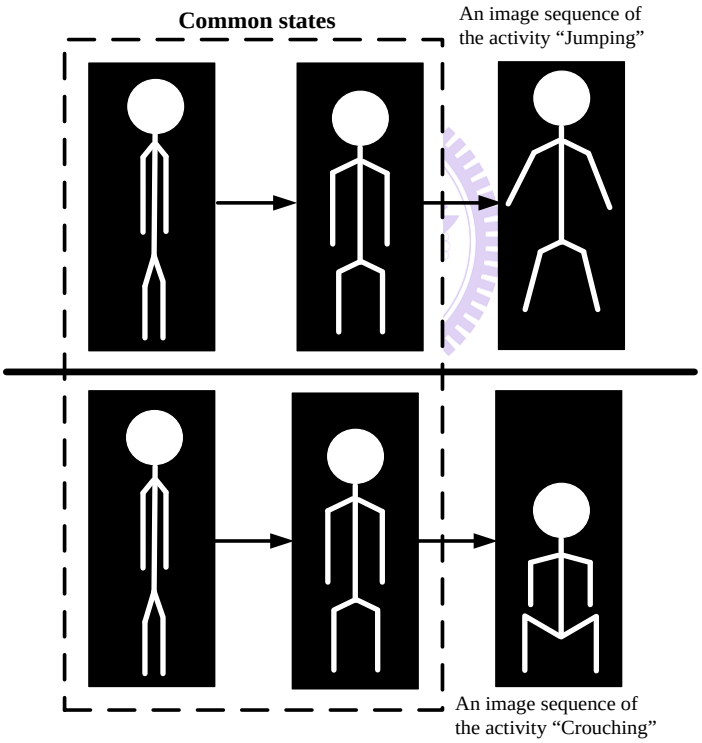


Fig. 3.8 Common states of two different activities.

Hence, we propose a method which not only combines temporal sequence information for recognition but also is tolerant to variation of actions done by different people. We utilize the fuzzy rule-base approach to design our system. The fuzzy rule-base approach also has been proposed in gesture recognition in [19].

We use the membership degree to represent the feature's possibility of each cluster. We choose the Gaussian type membership function to represent the key posture's features, because the Gaussian type membership function can reflect the similarity of the input feature vector to a key posture template vector.

Firstly, when the *k*-th training image frame $\mathbf{x}_k$ is inputted, the feature vector $\mathbf{a}_k$ is extracted by

$$\mathbf{a}_k = \mathbf{H}\,\mathbf{x}_k . \tag{3.25}$$

where $\mathbf{H}$ denotes the transformation matrix and $\mathbf{a}_k$ can be rewritten as

$$\mathbf{a}_k = \left[a_k^1, a_k^2, \cdots, a_k^{c-1}\right]^{\mathrm{T}}. \tag{3.26}$$

If we assume that the dimensions of the feature vectors are independent, then we can compute the similarity between the training vector $\mathbf{a}_k$ and each template vector. Let $\mathbf{\Sigma}$ denote the covariance matrix of all essential template vectors and C_i denote the *i*-th class of essential templates. The membership function is given by

$$\begin{aligned} r_{i,k} &= \mathrm{M}(\mathbf{a_k}|C_i) \\ &= \frac{1}{(2\pi)^{\frac{c-1}{2}}\left|\mathbf{\Sigma}\right|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{a_k}-\boldsymbol{\mu})^T\,\mathbf{\Sigma}^{-1}(\mathbf{a_k}-\boldsymbol{\mu})\right] \\ &= \prod_{m=1}^{c-1}\frac{1}{\sqrt{2\pi}\sigma_m}\exp\left[-\frac{1}{2}\frac{(a_k^m-\mu_{i,j}^m)^2}{\sigma_m^2}\right] \end{aligned} \tag{3.27}$$

where *j* is the training model number, $r_{i,k}$ denotes the grade of membership function in category *i* of the *k*-th image frame and σ_m is the *m*-th dimensional variance of the covariance matrix. After that we can obtain which category the image belongs to by

$$p_k = \arg\max_i r_{i,k} \tag{3.28}$$

We obtain which category the image belongs to, but that is a single image. Recognizing the human action is using three consecutive posture sequences instead of a single posture. Therefore, we combine three consecutive posture sequences to a group (I_1, I_2, I_3) and transfer the group to the feature vector (a_1, a_2, a_3). Assume we have c categories. There are c^3 combinations of the feature vector. By Eq. (3.28), the feature vector (a_1, a_2, a_3) is represented to (p_1, p_2, p_3).

In [18], fuzzy rules are generated by learning from examples. The generated rules are the follow form:

"**IF** antecedent conditions hold, **THEN** consequent conditions hold."

The number of antecedent conditions equals the number of features and the antecedent conditions are connected by "**AND**". For example, an image sequence (Image 1, Image2, Image3) with its category is D_1. We express that in vector format. Eq. (3.29) is showing the vector format.

$$[P_1, P_2, P_3; D_1] \tag{3.29}$$

Suppose that Image 1, Image2 and Image 3 belong to category 1, category 2 and category 3 respectively. Therefore, the image sequence (Image 1, Image2, Image3) is transferred to (P_1, P_2, P_3). Then, a rule is generated by the image sequence.

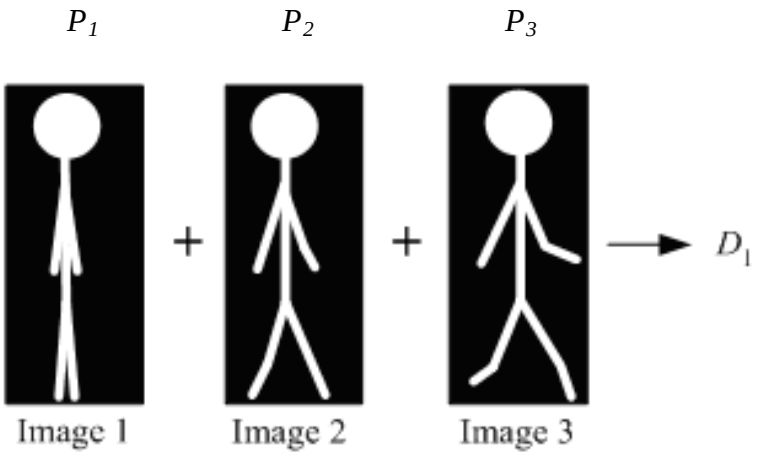


Fig. 3.9 A fuzzy rule learned to classify action.

Rule 1. IF I_1 is P_1 AND I_2 is P_2 AND I_3 is P_3 , THEN the activity is D_1.

After the learning step of different actions, some conflicting rules may be generated. The conflicting rules have the same image sequence but different activity. Therefore, we have to choose one from a set of conflicting rules. To this end, we choose the rule that is supported by a maximum number of training examples. Furthermore, to prune redundant or inefficient fuzzy rules, if the number of examples supporting a rule is less than a threshold, the rule is excluded from the set of rules.

3.5 Classification Algorithm

To obtain the action for an input video stream, we utilize the background model to extract foreground objects from the image frames. Then, we use down-sampling rate to classify the three consecutive postures. These images will be obtained by the following procedures:

1. Foreground subject extraction
2. Normalization
3. Transformation by EST and CST

After these procedures and constructing the rule base, we can compute the similarity between current image sequences (I_{k-2}, I_{k-1}, I_k) and each rule in the rule base by the membership function which is given by

$$\begin{aligned} r_{i,k} &= \mathrm{M}(s_{\mathbf{k}} | C_i) \\ &= \frac{1}{(2\pi)^{\frac{c-1}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2} (\mathbf{s_k} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{s_k} - \boldsymbol{\mu}) \right] \\ &= \prod_{m=1}^{c-1} \frac{1}{\sqrt{2\pi}\sigma_m} \exp\left[-\frac{1}{2} \frac{(s_k^m - \mu_{i,j}^m)^2}{\sigma_m^2} \right] \end{aligned} \tag{3.30}$$

where $\boldsymbol{\Sigma}$ denote the covariance matrix of all essential template vectors, C_i denote the i-th class of essential templates and j is the training model number. σ is the standard deviation of all essential templates. $r_{i,k}$ denotes the grade of membership function in category i of the k-th image frame. After the membership function, we obtain the membership degree from current image sequences.

For example, given a rule, "IF $\mathbf{I}_{k-2}$ is P_{n1} AND $\mathbf{I}_{k-1}$ is P_{n2} AND $\mathbf{I}_k$ is P_{n3}, THEN the action is *Dn*." we compute the similarity degree of each image. We obtain the membership degrees ($r_{k-2,n1}$, $r_{k-1,n2}$, $r_{k,n3}$) by Eq. (3.30). Then, we have to calculate the firing strength (FS) of the rule. The sum is used to compute the firing strength that is defined as follows:

$$FS = r_{k-2,n1} + r_{k-1,n2} + r_{k,n3} \tag{3.31}$$

Hence, we can compute the firing strength of each fuzzy rule which is in the rule base. Moreover, we will also investigate to the average value of maximal top-3, top-5, top-7, and top-9 firing strength of the rules with the same action to recognize the human action. Fig. 3.10 shows that take top-3 as an example, the similarity between three consecutive down-sampled images and each action that we average the maximal top-3 firing strength of the rules which have the same actions. After that, the action which has the highest average value of similarity is selected. Because of the major factors to separate the walking, jog and running is not templates but the velocity, therefore we include the velocity factor to recognize the human actions. We utilize the velocity to determine the actions and the fuzzy rules to determine the direction.

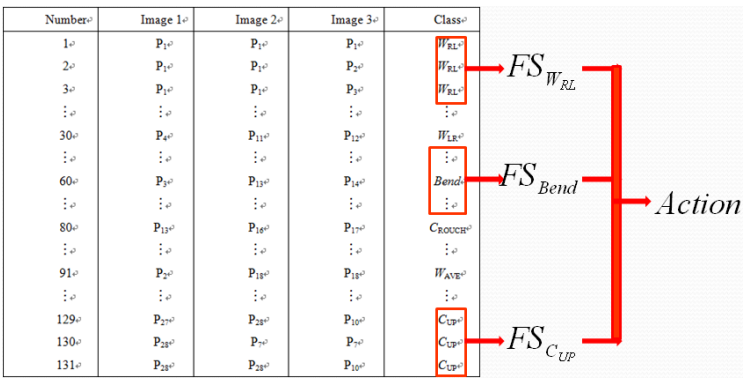


Fig. 3.10 Utilizing the average value of maximal top-3 firing strength.

Chapter 4 Experimental Results

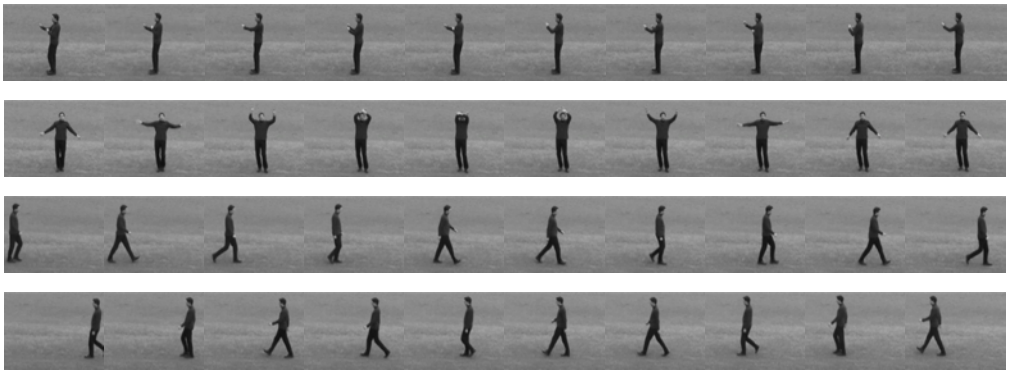
In our experiment, we tested our system on videos. We took the Weizmann databases, KTH databases and our LAB databases. We took our LAB databases at the 5th Engineering Building in NCTU campus. The light source is fluorescent lamp and stable. The background is not complex and we equip a table in the scene. The camera has a frame rate of thirty frames per second and image resolution is 320×240 pixels.

Fig. 4.1 One of the environment in our LAB databases.

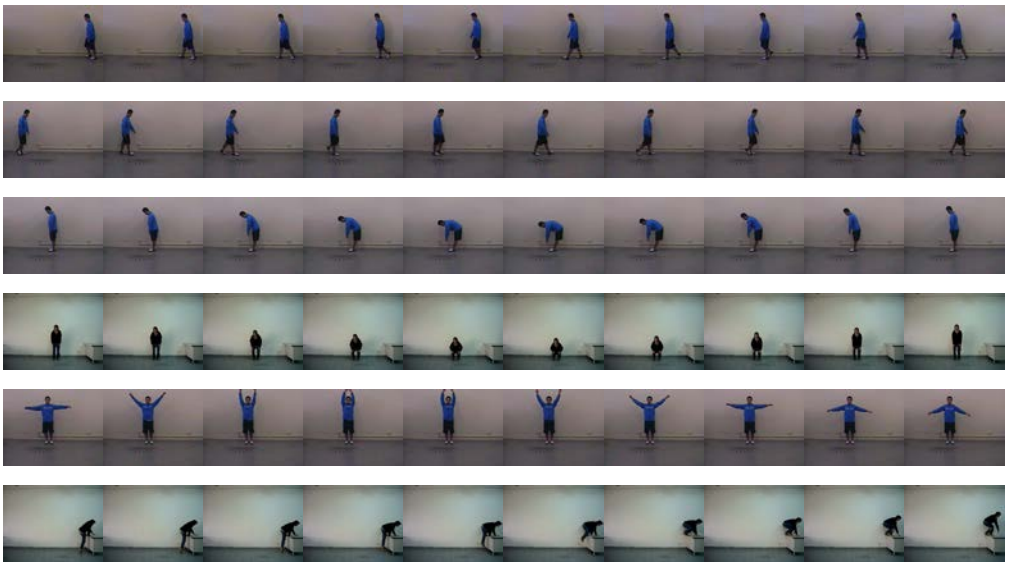
Fig. 4.2 Another environment in our LAB databases.

Fig. 4.1 and Fig. 4.2 show the environment of our LAB databases. In our LAB databases, the person performed several actions: “walking from left to right,” “walking from right to left,” “bend,” “crouch,” “climbing up,” and “waving.” The action “climbing up” is to climb up on the table from the ground. Fig. 4.3 shows the examples of actions from Weizmann databases, KTH databases and our LAB databases.

(a)



(b)



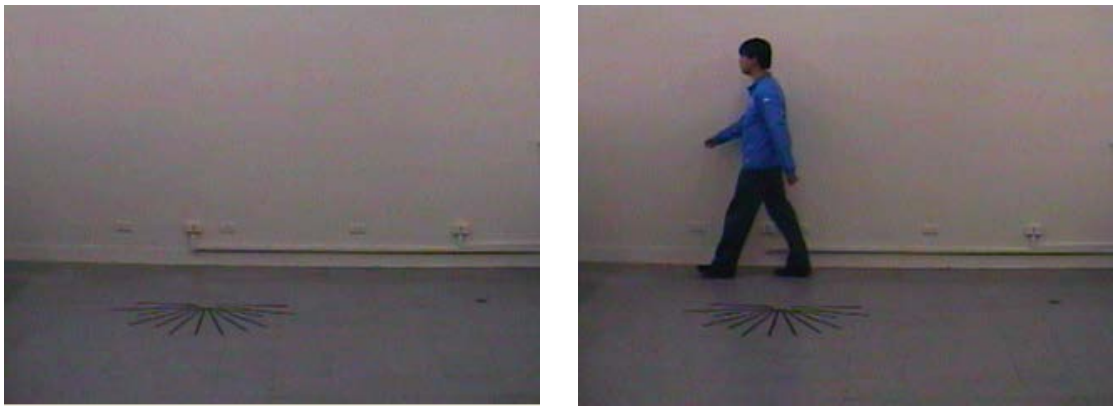
(c)

Fig. 4.3 Example video sequence used in our experiments. (a) Typical video sequences of Weizmann. From top to bottom: walk, bend, jack, jump, side, wave1 and wave2, respectively. (b) Typical video sequences of KTH. From top to bottom: boxing, wave, walking from left to right and walking from right to left, respectively. (c) Typical video sequences for actions of our LAB. From top to bottom: walking from right to left, walking from left to right, bend, crouch, wave and climbing up, respectively.

4.1 Background Model and Object Extraction

A background model is used for segmenting the foreground subject or object. In our system, we first record a video with no subject in environment to build the background models. If the grayscale value and the HSV color space background models are complete, we will extract the foreground pixels by using Eq. (3.5) and Eq. (3.6) in Section 3.1.2.

In order to get the optimal result of object extraction, we have to adjust the threshold in our system. In the grayscale value and the HSV color space background models, we set $k = 2.3$ in Eq. (3.5) and $k_v = 1.4$ in Eq. (3.6) to extract foreground pixels. Fig 4.4 shows an example of foreground extraction. Fig. 4.4(a) is a frame which obtained from background video. Fig. 4.4(b) is an image frame of the video stream. Fig. 4.4(c) is the result of the image frame transferred to grayscale value. Fig. 4.4(d) is the binary image after using shadow filter, closing filter and opening filter. Fig. 4.4(e) is the extracted foreground region.



(a) (b)

(c) (d)

(e)

Fig 4.4 Showing an example of foreground extraction. (a) Background image. (b) Input image. (c) Grayscale value image. (d) Binary image. (e) Extraction foreground image.

4.2 Fuzzy Rule Construction for Action Recognition

We construct the template model and the fuzzy rule database with the training data. We first utilize 5:1 down-sampling rate for walking, jog and running to select the essential templates. On the other hand, we utilize 5:1 down-sampling rate for walking, 3:1 down-sampling rate for jog and 2:1 down-sampling rate for running to select the essential templates. Thus, compare using the same down-sampling rate with using the different down-sampling rate, and we will know whether down-sampling rate affects the human action recognition. Fig. 4.5 is an example of some templates that using the same down-sampling rate. Fig. 4.6 shows using the different down-sampling rate to select the essential templates.

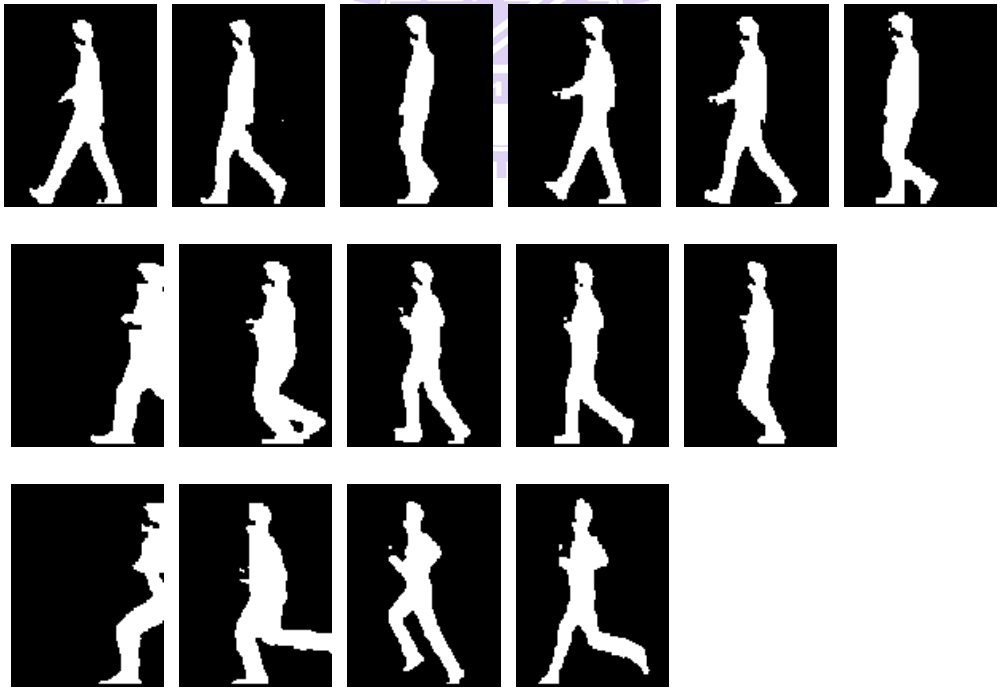


Fig. 4.5 Some essential templates use the same down-sampling rate. From top to bottom: walking, jog and running, respectively.

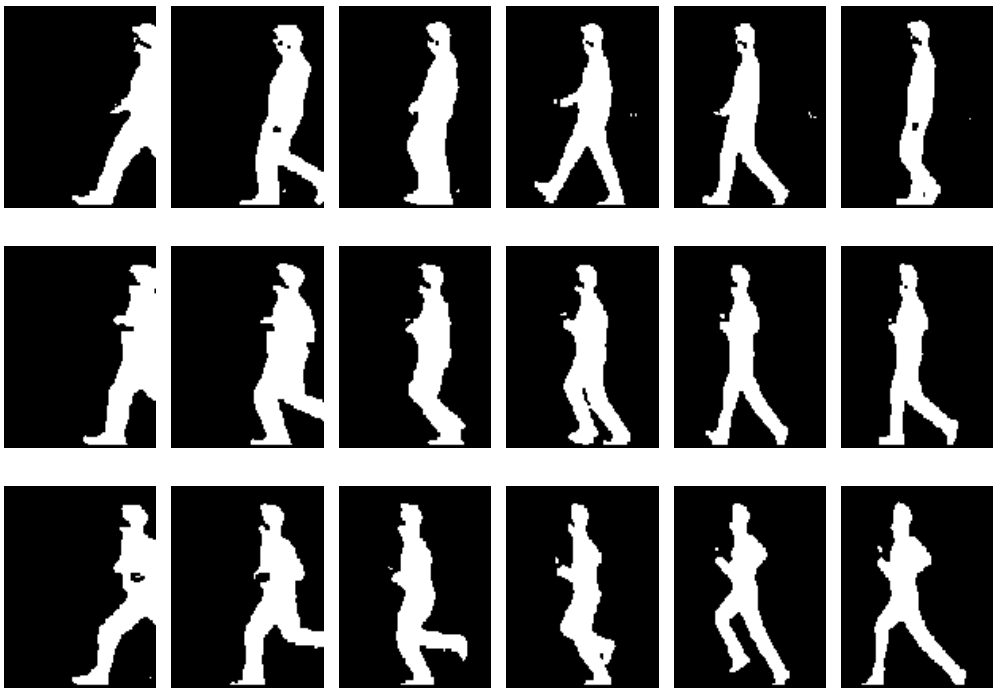


Fig. 4.6 Some essential templates use the different down-sampling rate. From top to bottom: walking, jog and running, respectively.

Furthermore, we chose five kinds of essential templates for “walking from right to left,” “walking from left to right,” “bend,” “crouch,” “climbing up,” “waving,” “boxing,” “jack,” respectively; four for “side,” “wave1,” “wave2” and three for “jump.” There are totally 30 kinds of essential templates in Weizmann databases, 20 kinds of essential templates in KTH databases and 30 kinds of essential templates in our LAB databases. Each essential template is a cluster with four similar key postures which are selected from four different training persons. Fig 4.7, Fig. 4.8, and Fig. 4.9 are the examples of essential templates of respective datasets.

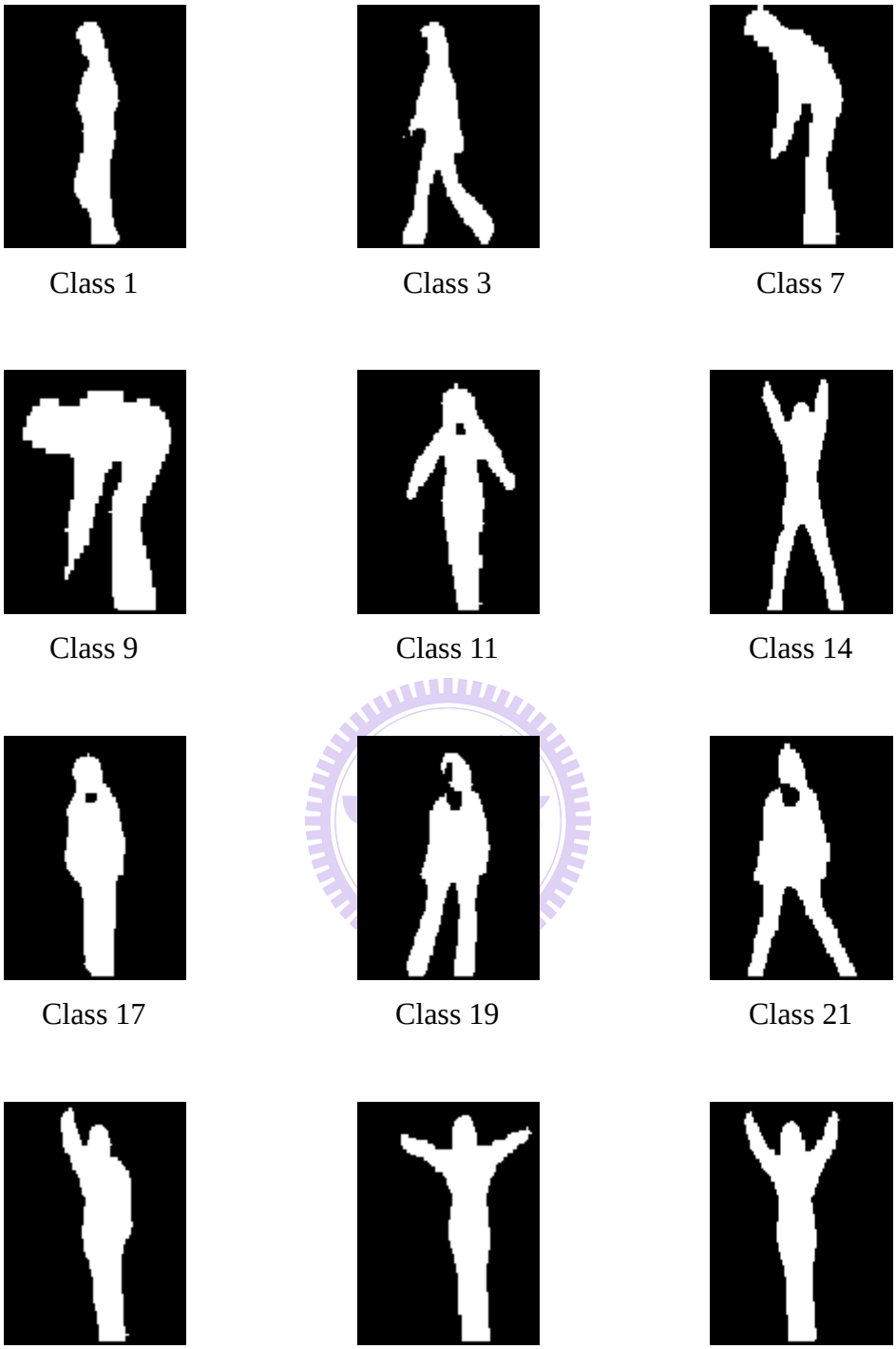


Fig. 4.7　30 essential templates for Weizmann databases.

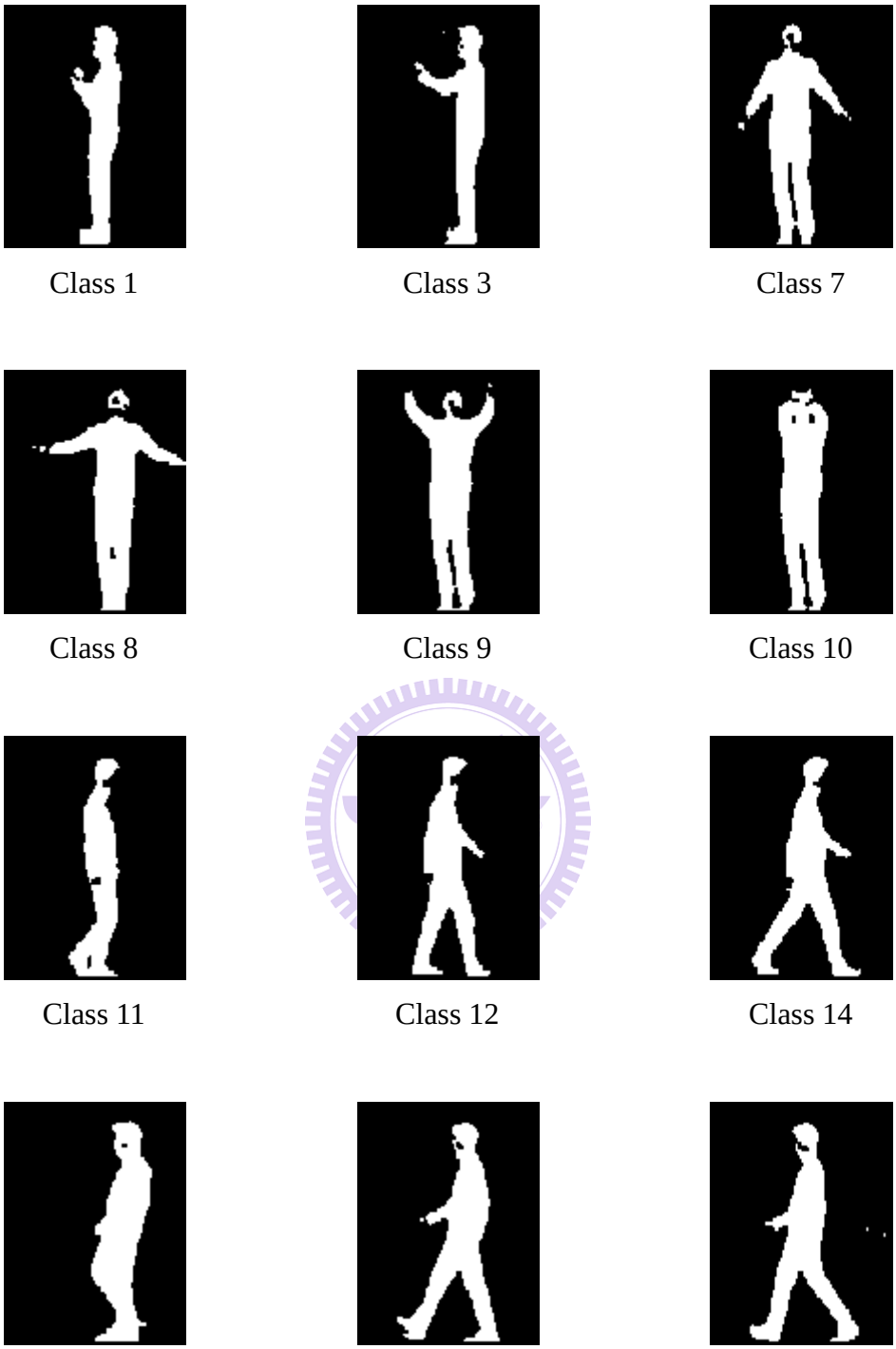


Fig. 4.8 20 essential templates for KTH databases.

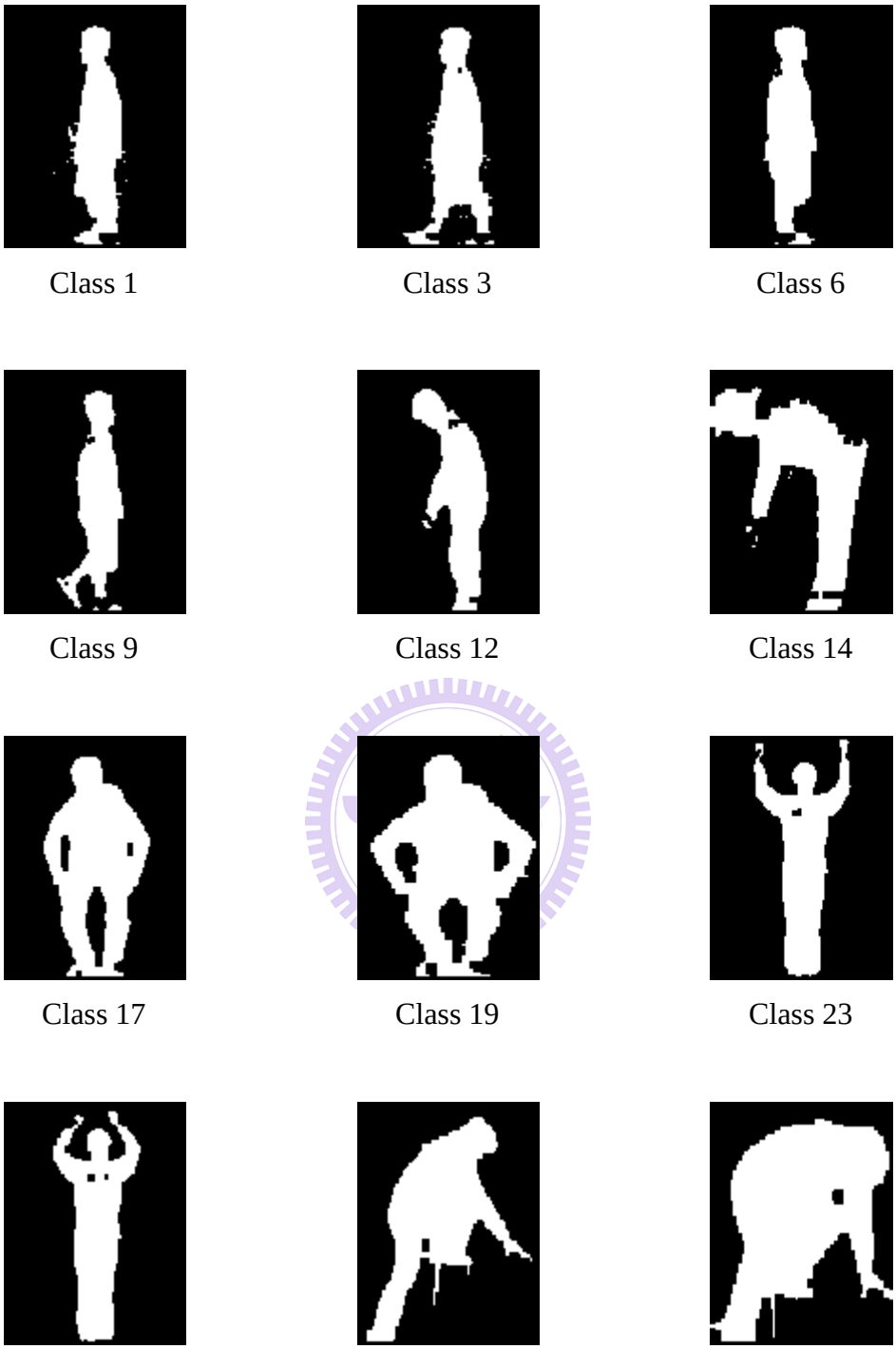


Fig. 4.9 30 essential templates for our LAB databases.

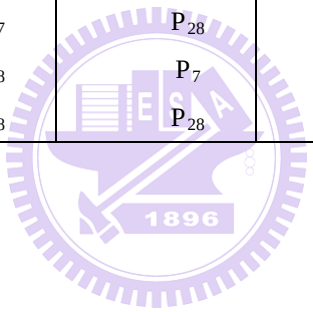
After determining the standard deviation vectors, the corresponding training video frames are inputted. The relationship between each image frame and each template is calculated by using Eq. (3.27) in Section 3.4. We gathered three consecutive down-sampled images as a group in order to include temporal information. The interval between each of these three images is determined by the down-sampling rate which is the same as in template selection phase and learning phase. Therefore, we gathered three images from different start points to train fuzzy rules. Taking 5:1 down-sampling rate for examples: the first frame, the 6-th frame and the 11-th frame are gathered together as an input training data; the second frame, the 7-th frame and the 12-th frame are gathered together as another input training data; the third frame, the 8-th frame and the 13-th frame are gathered together as another input training data, *etc*. Different start points of image frames are used for training fuzzy rules in our experiment, because the starting posture of testing video may not be the same. By utilizing different start points, the system is able to learn and then classify the actions at any time instant.

The group of the three images is converted to the posture sequence which has the maximum summation of three membership function values in Eq. (3.27). Each posture sequence will the consequent action of the rule with maximal value. If the corresponding rule is not existent, a new rule is built in the form of **IF-THEN** which is represented in Section 3.4.

Table I

Some of the Obtained Fuzzy Rule Base

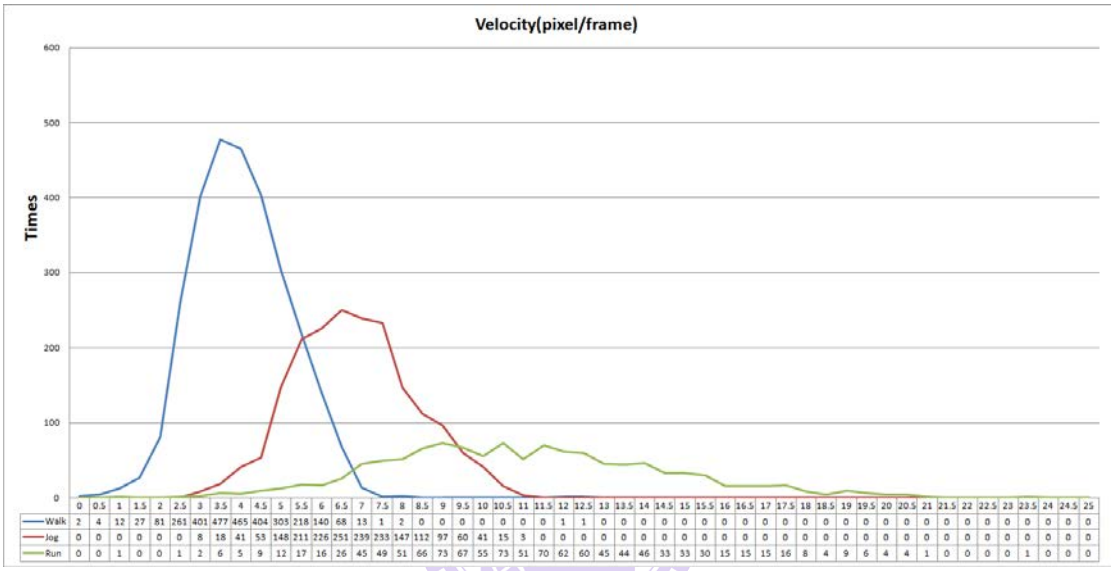
Number	Image 1	Image 2	Image 3	Class
1	P_1	P_1	P_1	W_{RL}
2	P_1	P_1	P_2	W_{RL}
3	P_1	P_1	P_3	W_{RL}
⋮	⋮	⋮	⋮	⋮
30	P_4	P_{11}	P_{12}	W_{LR}
⋮	⋮	⋮	⋮	⋮
60	P_3	P_{13}	P_{14}	*Bend*
⋮	⋮	⋮	⋮	⋮
80	P_{13}	P_{16}	P_{17}	C_{ROUCH}
⋮	⋮	⋮	⋮	⋮
91	P_2	P_{18}	P_{18}	W_{AVE}
⋮	⋮	⋮	⋮	⋮
129	P_{27}	P_{28}	P_{10}	C_{UP}
130	P_{28}	P_7	P_7	C_{UP}
131	P_{28}	P_{28}	P_{10}	C_{UP}



4.3 The Recognition Rate of Activities

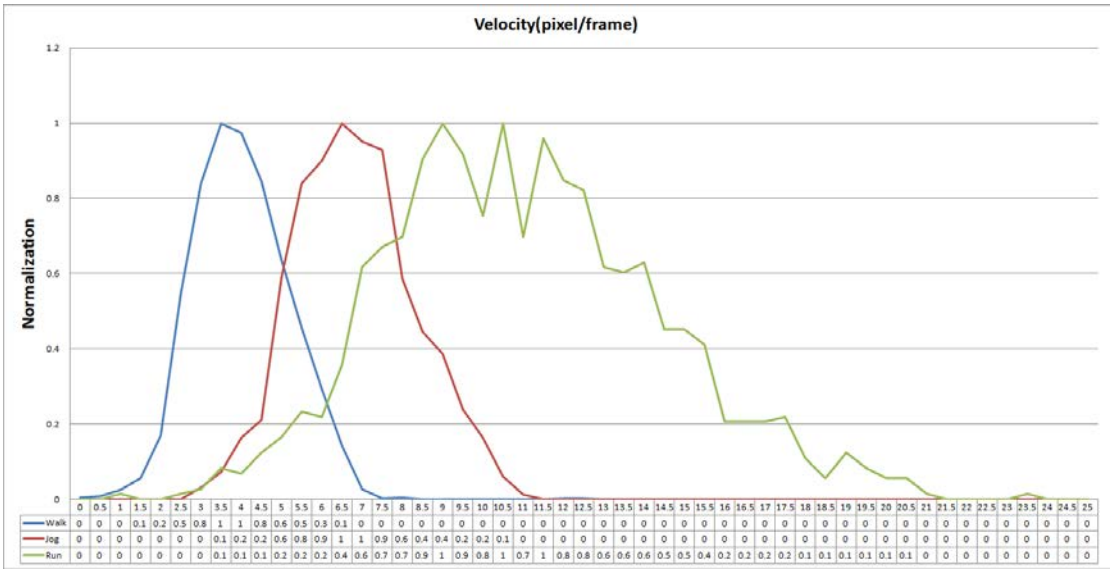
In order to calculate the recognition rate of activities, we use videos to test the human action recognition system. Each of video includes several actions in our experiment. Then, we input the testing video from different starting frames which is similar to the way for the training fuzzy rules. Namely, we recognize the video from the first frame, the second frame, the third frame and the fourth frame, *etc.* with the sampling intervals of five frames. Hence, there are many video databases for testing. The W_{RL} is the activity "walking from right to left," W_{LR} is the activity "walking from left to right," J_{RL} is the activity "jog from right to left," J_{LR} is the activity "jog from left to right," R_{RL} is the activity "running from right to left," R_{LR} is the activity

"running from left to right," and C_{UP} is the activity "climbing up." The frame based accuracy is the total number of correct recognition divide by the total number of recognitions done. The following tables show the accuracy by using the video bases. Fig. 4.10 shows the statistic velocity of walking, jog and running from the KTH dataset and we normalize the statistic result. Fig. 4.11 shows the regularization result.



	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10	10.5	11	11.5	12	12.5	13	13.5	14	14.5	15	15.5	16	16.5	17	17.5	18	18.5	19	19.5	20	20.5	21	21.5	22	22.5	23	23.5	24	24.5	25
Walk	2	4	12	27	81	261	401	477	465	404	305	218	140	68	15	1	2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Jog	0	0	0	0	0	0	8	18	41	53	148	211	226	251	239	233	147	112	97	60	41	15	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Run	0	0	1	0	0	1	2	6	5	9	12	17	16	26	45	49	51	66	73	67	55	73	51	70	62	60	45	44	46	33	33	30	15	15	15	16	8	4	9	6	4	4	1	0	0	0	0	1	0	0	0

Fig. 4.10 The statistic velocity of walking, jog and running from the KTH dataset.



	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10	10.5	11	11.5	12	12.5	13	13.5	14	14.5	15	15.5	16	16.5	17	17.5	18	18.5	19	19.5	20	20.5	21	21.5	22	22.5	23	23.5	24	24.5	25
Walk	0	0	0	0.1	0.2	0.5	0.8	1	1	0.8	0.6	0.5	0.3	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Jog	0	0	0	0	0	0	0	0.1	0.2	0.2	0.6	0.8	0.9	1	1	0.9	0.6	0.4	0.4	0.2	0.2	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Run	0	0	0	0	0	0	0	0.1	0.1	0.1	0.2	0.2	0.2	0.4	0.6	0.7	0.7	0.9	1	0.9	0.8	1	0.7	1	0.8	0.8	0.6	0.6	0.6	0.5	0.5	0.4	0.2	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0	0	0	0	0	0	0	0	0

Fig. 4.11 Normalizing the statistic velocity.

Table II

Action Recognition Use 5:1 Down-Sampling Rate on KTH dataset.

Input / Output	5:1 Down-Sampling Rate					
	W_{RL}	W_{LR}	J_{RL}	J_{LR}	R_{RL}	R_{LR}
W_{RL}	379	57	114	6	21	3
W_{LR}	34	374	10	124	7	22
J_{RL}	57	7	147	1	59	0
J_{LR}	11	33	0	109	0	65
R_{RL}	11	0	11	0	39	0
R_{LR}	7	17	0	37	1	40
Accuracy	75.95%	76.64%	52.13%	39.35%	30.71%	30.77%

Total frame based accuracy: 60.34%

Table III

Action Recognition Use 5:1 for Walking, 3:1 for Jog and 2:1 for Running Sampling Rate.

Input / Output	Different Down-Sampling Rates					
	W_{RL}	W_{LR}	J_{RL}	J_{LR}	R_{RL}	R_{LR}
W_{RL}	863	94	316	11	74	6
W_{LR}	70	735	16	299	10	76
J_{RL}	65	16	231	7	95	1
J_{LR}	9	111	1	265	2	129
R_{RL}	13	4	81	0	199	2
R_{LR}	6	25	2	37	1	137

Accuracy	94.11%	74.62%	35.70%	42.81%	52.23%	39.03%

Total frame based accuracy: 60.61%

Table IV

Action Recognition Use 5:1 for Walking, 3:1 for Jog and 2:1 for Running Sampling Rate with Velocity factor.

Input / Output	Different Down-Sampling Rates					
	W_{RL}	W_{LR}	J_{RL}	J_{LR}	R_{RL}	R_{LR}
W_{RL}	708	119	194	3	26	4
W_{LR}	61	663	8	219	2	35
J_{RL}	53	11	273	3	49	3
J_{LR}	6	80	6	265	0	85
R_{RL}	5	0	60	0	179	12
R_{LR}	2	11	2	25	3	168
Accuracy	84.79%	75.00%	50.28%	51.46%	69.11%	54.72%

Total frame based accuracy: 67.48%

Table V

Action Recognition Use the Maximum Firing Strength on Weizmann databases.

Output \ Input	Top-1						
	W_{RL}	$Wave_2$	*Bend*	*Jack*	*Jump*	*Side*	$Wave_1$
W_{RL}	292	0	0	5	0	0	0
$Wave_2$	0	295	0	2	0	0	11
Bend	5	0	259	3	10	0	4
Jack	0	1	0	337	2	3	1
Jump	0	0	0	2	185	7	4
Side	2	0	0	1	2	163	1
$Wave_1$	0	8	1	7	0	0	218
Accuracy	97.66%	97.04%	99.62%	94.40%	92.96%	94.22%	91.21%

Total frame based accuracy: 95.52%

Table VI

Action Recognition Use the Average Value of maximal Top-3 Firing Strength on Weizmann databases.

Output \ Input	Top-3						
	W_{RL}	$Wave_2$	*Bend*	*Jack*	*Jump*	*Side*	$Wave_1$
W_{RL}	295	0	0	5	0	0	0
$Wave_2$	0	288	0	2	0	0	0
Bend	1	0	258	2	0	0	1
Jack	0	1	0	337	0	0	0
Jump	0	0	0	2	199	6	6

Side	3	0	0	0	0	167	0
$Wave_1$	0	15	2	9	0	0	232
Accuracy	98.66%	94.74%	99.23%	94.40%	100%	96.53%	97.07%

Total frame based accuracy: 97.00%

Table VII

Action Recognition Use the Average Value of maximal Top-5 Firing Strength on Weizmann databases.

Input / Output	Top-5						
	W_{RL}	$Wave_2$	*Bend*	*Jack*	*Jump*	*Side*	$Wave_1$
W_{RL}	295	0	0	5	0	0	0
$Wave_2$	0	289	0	2	0	0	0
Bend	1	0	258	2	0	0	0
Jack	0	0	0	337	0	0	0
Jump	0	0	0	3	199	7	6
Side	3	0	0	0	0	166	0
$Wave_1$	0	15	2	8	0	0	233
Accuracy	98.66%	95.07%	99.23%	94.40%	100%	95.95%	97.49%

Total frame based accuracy: 97.05%

Table VIII

Action Recognition Use the Average Value of maximal Top-7 Firing Strength on Weizmann databases.

Input / Output	Top-7						
	W_{RL}	$Wave_2$	*Bend*	*Jack*	*Jump*	*Side*	$Wave_1$
W_{RL}	295	0	0	4	0	0	0
$Wave_2$	0	291	0	2	0	0	0
Bend	1	0	258	2	0	0	0
Jack	0	0	0	343	0	0	0
Jump	0	0	0	2	199	8	6
Side	3	0	0	0	0	165	0
$Wave_1$	0	13	2	4	0	0	233
Accuracy	98.66%	95.72%	99.23%	96.08%	100%	95.38%	97.49%

Total frame based accuracy: 97.43%

Table IX

Action Recognition Use the Average Value of maximal Top-9 Firing Strength on Weizmann databases.

Input / Output	Top-9						
	W_{RL}	$Wave_2$	*Bend*	*Jack*	*Jump*	*Side*	$Wave_1$
W_{RL}	295	0	0	3	0	0	0
$Wave_2$	0	290	0	2	0	0	0
Bend	1	0	258	2	0	0	0
Jack	0	0	0	342	0	0	0

Jump	0	0	0	4	199	8	6
Side	3	0	0	0	0	165	0
$Wave_1$	0	14	2	4	0	0	233
Accuracy	98.66%	95.39%	99.23%	95.80%	100%	95.38%	97.49%

Total frame based accuracy: 97.32%

Table X

Action Recognition Use the Maximum Firing Strength on KTH databases.

Input / Output	Top-1			
	Boxing	W_{LR}	*Handwaving*	W_{RL}
Boxing	1773	2	22	1
W_{LR}	18	465	39	50
Handwaving	94	2	2146	8
W_{RL}	7	5	19	411
Accuracy	93.71%	98.10%	96.41%	87.45%

Total frame based accuracy: 94.73%

Table XI

Action Recognition Use the Average Value of maximal Top-3 Firing Strength on KTH databases.

Input / Output	Top-3			
	Boxing	W_{LR}	*Handwaving*	W_{RL}
Boxing	1863	3	22	0
W_{LR}	0	464	3	11
Handwaving	29	1	2195	2
W_{RL}	0	6	6	457
Accuracy	98.47%	97.89%	98.61%	97.23%

Total frame based accuracy: 98.36%

Table XII

Action Recognition Use the Average Value of maximal Top-5 Firing Strength on KTH databases.

Input / Output	Top-5			
	Boxing	W_{LR}	*Handwaving*	W_{RL}
Boxing	1868	2	22	0
W_{LR}	0	467	1	9
Handwaving	24	1	2201	1
W_{RL}	0	4	2	460
Accuracy	98.73%	98.52%	98.88%	97.87%

Total frame based accuracy: 98.70%

Table XIII

Action Recognition Use the Average Value of maximal Top-7 Firing Strength on KTH databases.

Input / Output	Top-7			
	Boxing	W_{LR}	*Handwaving*	W_{RL}
Boxing	1872	0	22	0
W_{LR}	0	469	1	6
Handwaving	20	1	2203	1
W_{RL}	0	4	0	463
Accuracy	98.94%	98.95%	98.97%	98.51%

Total frame based accuracy: 98.91%

Table XIV

Action Recognition Use the Average Value of maximal Top-9 Firing Strength on KTH databases.

Input / Output	Top-9			
	Boxing	W_{LR}	*Handwaving*	W_{RL}
Boxing	1872	0	24	0
W_{LR}	0	469	1	4
Handwaving	20	0	2201	2
W_{RL}	0	5	0	464
Accuracy	98.94%	98.95%	98.88%	98.72%

Total frame based accuracy: 98.89%

Table XV

Action Recognition Use the Maximum Firing Strength on LAB databases.

Input / Output	Top-1					
	W_{RL}	W_{LR}	*Bend*	*Crouch*	*Wave*	C_{UP}
W_{RL}	1286	14	40	0	92	0
W_{LR}	125	1084	0	0	85	0
Bend	20	0	2269	0	2	82
Crouch	0	14	2	875	118	62
Wave	4	1	0	0	1149	3
C_{UP}	2	1	138	2	2	411
Accuracy	89.49%	97.31%	92.65%	99.77%	79.35%	73.66%

Total frame based accuracy: 89.74%

Table XVI

Action Recognition Use the Average Value of maximal Top-3 Firing Strength on LAB databases.

Input / Output	Top-3					
	W_{RL}	W_{LR}	*Bend*	*Crouch*	*Wave*	C_{UP}
W_{RL}	1462	3	0	0	0	0
W_{LR}	82	1278	0	0	0	0
Bend	14	0	2529	0	0	23
Crouch	0	0	0	929	0	5
Wave	10	20	0	0	1533	4
C_{UP}	0	0	17	2	0	533
Accuracy	93.24%	98.23%	99.33%	99.79%	100%	94.34%

Total frame based accuracy: 97.87%

Table XVII

Action Recognition Use the Average Value of maximal Top-5 Firing Strength on LAB databases.

Input / Output	Top-5					
	W_{RL}	W_{LR}	*Bend*	*Crouch*	*Wave*	C_{UP}
W_{RL}	1495	7	0	0	0	0
W_{LR}	43	1272	0	0	0	0
Bend	19	0	2526	0	0	13
Crouch	2	0	0	929	0	9
Wave	9	20	0	0	1533	5
C_{UP}	0	2	20	2	0	538

Accuracy	95.34%	97.77%	99.21%	99.79%	100%	95.22%

Total frame based accuracy: 98.21%

Table XVIII

Action Recognition Use the Average Value of maximal Top-7 Firing Strength on LAB databases.

Input / Output	Top-7					
	W_{RL}	W_{LR}	*Bend*	*Crouch*	*Wave*	C_{UP}
W_{RL}	1499	9	0	0	0	0
W_{LR}	30	1266	0	0	0	0
Bend	25	0	2520	0	0	3
Crouch	3	0	0	929	0	9
Wave	11	23	0	0	1533	5
C_{UP}	0	3	26	2	0	548
Accuracy	95.60%	97.31%	98.98%	99.79%	100%	96.99%

Total frame based accuracy: 98.24%

Table XIX

Action Recognition Use the Average Value of maximal Top-9 Firing Strength on LAB databases.

Input / Output	Top-9					
	W_{RL}	W_{LR}	*Bend*	*Crouch*	*Wave*	C_{UP}
W_{RL}	1498	10	0	0	0	0
W_{LR}	25	1261	0	0	0	0
Bend	28	1	2516	0	0	3
Crouch	3	1	0	929	0	12
Wave	14	25	0	0	1533	8
C_{UP}	0	3	30	2	0	542
Accuracy	95.54%	96.93%	98.82%	99.79%	100%	95.93%

Total frame based accuracy: 98.05%

Chapter 5 Conclusion

In this thesis, a novel method for human action recognition was proposed. Firstly, a foreground subject is extracted and converted to a binary image. For better efficiency and separability, the binary image is transformed to a new space by eigenspace and canonical space transformation. After down-sampling, we gather three image sequences to recognize the human actions. By template matching, we can infer the actions from fuzzy rules. Fuzzy rules combine not only temporal sequence information for recognition but also the tolerant to variation of actions done by different people.

Experimental results have shown that using same down-sampling rate is similar to using different down-sampling rate. This is because that, the three image sequences which selected by different down-sampling are similar. It is difficult to recognize actions with similar posture sequences. However, by combining the important speed rule base, we can improve the similar walking, jog and running action recognition by about 67.48%. Moreover, using the average value of maximal top-7 firing strength to recognize the human action is better than similar attempts. The best frame based accuracy is obtained by using the average value of maximal top-7 firing strength in Weizmann databases, KTH databases and our LAB databases.

References

[1] K. Schindler and L. Van Gool, “Action Snippets: How many frames does human action recognition require?,” in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 1–8, Jun. 2008.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis, “W^4: Real-time surveillance of people and their activities,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[3] F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[4] I. Cohen and H. Li, “Inference of human postures by classification of 3D human body shape,” in *Proc. IEEE Int. Workshop on Anal. Modeling of Faces and Gestures*, pp. 74–81, Oct. 2003.

[5] Robert H. Luke and James M. Keller, “Modeling human activity from voxel person using fuzzy logic,” *IEEE Transactions on fuzzy systems*, vol. 17, no.1, pp. 39–49, Feb. 2009.

[6] S. Carlsson and J. Sullivan, “Action recognition by shape matching to key frames,” in *Proc. IEEE Comput. Soc. Workshop Models versus Exemplars in Comput. Vision*, pp. 263–270, Dec. 2002.

[7] A. Elgammal, D. Harwood, and L. Davis, “Non-Parametric Model for Background Subtraction,“ *Proc. Sixth European Conf. Computer Vision*, vol. II, pp. 751–767, June 2000.

[8] M. Piccardi, “Background subtraction techniques: a review,” in *Proc. IEEE Int. Conf. SMC.*, vol. 4, pp. 3099–3104, Oct. 2004.

[9] T. Horprasert, D. Harwood, and L.S. Davis, “A Statistical Approach for Real-Time Robust Background Subtraction and Shadow Detection,” in *Proc. IEEE ICCV’99*, 1999.

[10] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, “Improving Shadow Suppression in Moving Object Detection with HSV Color Information,” in *Proc. IEEE Intelligent transportation System Conference*, pp. 334 – 339, 2001.

[11] H. Saito, A Watanabe, and S Ozawa, “Face pose estimating system based on eigenspace analysis,” in *Proc. Int. Conf. Image Processing*, vol. 1, pp. 638–642, 1999.

[12] J. Wang, G. Yuantao, K. N. Plataniotis, and A. N. Venetsanopoulos, “Select eigenfaces for face recognition with one training sample per subject,” *in Proc.8th Cont., Automat. Robot. Vision Conf., ICARCV 2004*, vol. 1, pp. 391–396, Dec. 2004.

[13] P. S. Huang, C. J. Harris, and M. S. Nixon, “Canonical space representation for recognizing humans by gait or face,” in *Proc. IEEE Southwest Symp. Image Anal. Interpretation*, pp. 180–185, Apr., 1998.

[14] M. M. Rahman and S. Ishikawa, “Robust appearance-based human action recognition,” in *Proc. the 17th Int. Conf. Pattern Recog.*, vol. 3, pp. 165–168, 2004.

[15] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257 – 286, 1989.

[16] S. Yoshizawa, N. Wada, N. Hayasaka, and Y. Miyanaga, “Scalable architecture for word HMM-based speech recognition,” in *Proc. the 2004 Int. Symposium Circuits Syst., ISCAS 2004*, vol. 3, pp. III–417 – 420, 2004.

[17] L. Nianjun, B. C. Lovell, and P. J. Kootsookos, “Evaluation of HMM training algorithms for letter hand gesture recognition,” in *Proc. the 3rd IEEE Int.*

Symposium Signal Processing Inform. Technol., ISSPIT 2003, pp.648 – 651, 2003.

[18] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man Cybern*, vol. 22, no. 6, pp. 1414–1427, Dec. 1992.

[19] M. C. Su, "A fuzzy rule-based approach to spatio-temporal hand gesture recognition," in *IEEE Trans. Sys., Man Cybern*, vol. 30, no. 2, pp. 276 – 281, 2000.

[20] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images," in *Proc. ICASSP*, pp. 2148–2151, 1997.

[21] P. S. Huang, C. J. Harris, and M. S. Nixon, "Canonical space representation for recognizing humans by gait or face," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation*, pp. 180–185, Apr. 1998.

[22] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition, 1300 Boylston Street Chestnut Hill, Massachusetts USA: Academic Press, 1990.

[23] K. Ohba, Y. Sato, and K. Ikeuchi, "Appearance-based visual learning and object recognition with illumination invariance," in *Machine Vision and Applications*, Vol. 12, No. 4, pp. 189–196, 2000.

[24] J. C. S. Jacques Jr., C. R. Jung, and S. R. Musse, "Background subtraction and shadow detection in grayscale video sequences," in *Proc. the 18th Brazilian Symp. Computer Graphics and Image Processing, SIBGRAPI 2005*, pp. 189–196, 2005

[25] Soriano M, Huovinen S, Martinkauppi B, Laaksonen M. "Using the skin locus to cope with changing illumination conditions in color-based face tracking," in *IEEE Nordic Signal Processing Symposium, kolmarden, Sweden*, pp. 383 – 386,

Jun. 2000.

[26] P. S. Huang, C. J. Harris, and M. S. Nixon, “Canonical space representation for recognizing humans by gait or face,” in *Proc. IEEE Southwest Symp. Image Anal. Interpretation*, pp. 180–185, Apr. 1998.

[27] Y. C. Luo, “Extracting the Foreground Subject in the HSV Color space and Its Application to Human Activity Recognition System,” *Master Thesis*, Elect. and Con. Eng. Dept., Chiao Tung Univ., Taiwan, 2007.