國 立 交 通 大 學

電 控 工 程 研 究 所

碩 士 論 文

應用人體動作辨識系統於吃藥辨識及日常生活活動

Applying Human Activity Recognition System to Medicine Taking and Activities of Daily Living

研 究 生：蔡 宗 憲

指 導 教 授：張 志 永

中 華 民 國 一百 年 七 月

應用人體動作辨識系統於吃藥辨識及日常生活活動

Applying Human Activity Recognition System to Medicine Taking and Activities of Daily Living

學　　生：蔡宗憲　　Student : Tzung-Shian Tsai

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 2011

Hsinchu, Taiwan, Republic of China

中 華 民 國 一百 年 七 月

應用人體動作辨識系統於吃藥辨識及日常生活活動

學生:蔡宗憲　　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效果，使得前後景之分離結果能夠更完整。我們以 5：1 降低取樣頻率，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，累積三張上述降頻取樣動作影像後，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。

此外，當某人要進行吃藥動作時，我們使用在 HSV 空間中建立好的藥包顏色色彩模型（僅考慮色調）去辨識藥包的顏色。因此，藉由結合藥包顏色色彩模型和人體動作辨識系統，我們就可以得知某人正在吃藥以及他的藥包顏色。最後，我們利用人體動作辨識系統去記錄學生的日常生活。

Applying Human Activity Recognition System to Medicine Taking and Activities of Daily Living

STUDENT: Tzung-Shian Tsai 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, a new person tracking and continuous activity recognition is proposed. We build two background models, in grayscale and HSV color space as well to extract the human correctly, and we could also reduce the shadowing effect well. For better efficiency and separability, the binary image is firstly transformed to a new space by eigenspace and then canonical space transformation, and the recognition is finally done in canonical space. A three image frame sequence, 5:1 down sampling from the video, is converted to a posture sequence by template matching. The posture sequence is classified to an action by fuzzy rules inference. Fuzzy rule approach can not only combine temporal sequence information for recognition but also be tolerant to variation of action done by different people and time.

Moreover, we make use of the hue component to recognize the medical pouch's color when one is taking medicine. By combining with the hue-based pouch's color model and human activity recognition system, we can know someone is taking medicine and its medical pouch's color as well. Finally, we also employ the activity

recognition system to record a student’s activity in the daily living.

ACKNOWLEDGEMENTS

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

Human activity analysis is an open problem that has been studied intensely within the areas of video surveillance, homeland security, and more recently, eldercare. In the video surveillance, human activity recognition from video streams has many applications such as home care system, human-machine interface, automatic surveillance, and smart home applications. For example, an automatic system will trigger an alarm condition when the automated surveillance system detects and recognizes suspicious human activities. Human activity recognition can also be used in extracting semantic descriptions from video clips to automate the process of video indexing. However, there is no rigid syntax and well-defined structure as that of the gesture and sign language which can be used for activity recognition. Therefore, this makes human activity recognition become a challenging task.

Several human activity recognition methods have been proposed in the past few years. Bobick and Davis [1], they recognized human activities by comparing motion-energy and motion-history of template images with temporal images. Carlsson and Sullivan [2], shape was represented by edge data obtained from canny edge detection. Cohen and Li [3] presented a view-independent 3-D shape description for classifying and identifying human activity using SVM. W^4 [4] can detect people (single person or people in group) by adopting an adaptive background model and identify the activities by finding the body parts on the silhouette boundary. Luke and Keller et al. [5], they build a voxel person to model human activity and recognize these activities by fuzzy logic.

In our research, we design a robust method that uses temporal information, which is implicitly inherent in the human activity recognition. People have the same postures and posture sequences when they perform a specific action. Therefore, we use shape features to classify each image frame into postures we defined. Then, we use the frame sequences of key postures to recognize which activity one does. The objective of this thesis is to provide a system to auto-surveillance and to track people and identify their activities.

The human activity recognition system flowchart is illustrated in Fig. 1.1. Our system can be separated into three components. The first component is foreground subject extraction. The second component is the transformation of image data in a space smaller and easier for posture recognition. The third component is the posture classification of an image frame and activity recognition using frame sequences.

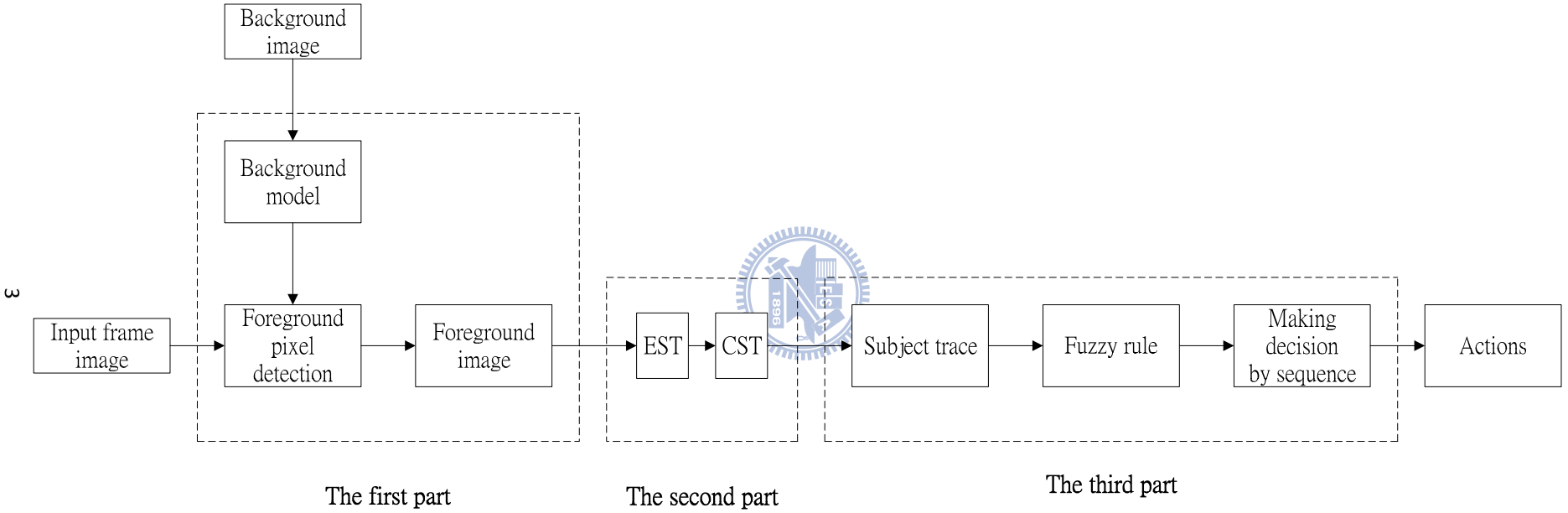


Fig. 1.1 The block diagram of human activity recognition system

1.2 Foreground subject extraction

Background subtraction is widely used for detecting moving objects from image frames of static cameras. The rationale of this approach is to detect the moving objects by the difference between the current frame and a reference frame, often called the "background model." A review is given in [6] where many different approaches were proposed in recent years. In our system, we build two background models; one is based on grayscale value and the other is based on HSV color space. Basically, the background image is a representation of the static scene. We prepare to update the background model until the subject enters the scene. After the subject leaves the scene, we also update the background mode.

After building a background model, we can extract foreground subject from video frames by subtracting each pixel value of background model from that current image frame. Then, the resulting image is converted to a binary image by setting a threshold. The binary image mainly contains foreground subject with only little noise. Therefore, we can set a threshold in the histogram of the binary image to extract a rectangle image, which is the most resemble shape of a person, of the target subjects. The rectangle image is resized to the standard level.

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 exists some redundancy. The redundancy possesses little information of an image. Hence, some space transformations are introduced to reduce redundancy of an image by reducing the data size of the image. The first step of redundancy

reduction often transforms an image from spatiotemporal space to another data 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 and so on. Our transformation method combines eigenspace transformation and canonical space transformation which are described as follows.

Eigenspace transformation (EST), based on Principal Component Analysis, has been demonstrated to be a potent scheme used below: automatic face recognition proposed in [7], [8]; gait analysis proposed in [9]; and action recognition proposed in [10]. 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 high computation efforts when the image is large. Therefore, we combine EST and CST in order to improve the classification performance while reducing the dimension, and hence each image can be projected from a high-dimensional spatiotemporal space to a single point in a low-dimensional canonical space. In this new space the recognition of human activities becomes much simpler and easier.

1.4 Image Frame Classification and Activity Recognition

In this thesis each in a video segmentation, 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. We group three contiguous 5:1 down-sampled images and transform them to three consecutive feature vectors. Then, the three contiguous images are down-sampled and its sample rate is

usually 6 frames per second. Next, the time-sequential images are converted to a posture sequence by using these three feature vectors. The posture sequence is dignified by the number of the templates. In the learning stage, we build a transition model in terms of three consecutive posture sequences which is 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.

After transforming image frames to eigenspace and canonical space domain, we greatly reduce the data (image) size. We make use of fuzzy rule-base techniques to classify human activity, not using the shape of an image. Thus our activity analysis task can be tolerant of dissimilarity, uncertainty, ambiguity and irregularity existent in the data. Relevant articles using the fuzzy theory in action recognition are described as follows. Wang and Mendel [11] proposed that fuzzy rules to be generated by learning from examples.

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 pattern and the corresponding antecedent part in the training data. When our system classifies an unknown action, it will test on three consecutive sampled images of the video frames by each fuzzy rule learned before. The accumulated similarity measure associated with these three consecutive postures is to match the posture sequence representing activity model of the training database, and the unknown action is classified to the action yielding the highest accumulative similarity. Finally, we will build a taking medicine system that is based on the above activity recognition.

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 our taking medicine recognition system that includes “skin color detection,”“medical pouch color recognition” and “activity recognition system.” Then, we also do activities of daily living by only using our “activity recognition system.” 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. There are many well-known transformation methods to reduce the size of data such as Fourier transformation, wavelet, principal component analysis (PCA), eigenspace transformation (EST) and so on. However, PCA based on the global covariance matrix of the full set of image data is not sensitive to the class structure existent in the data. In order to increase the discriminatory power of various activity features, Etemad and Chellappa [12] used linear discriminant analysis (LDA), also called canonical analysis (CA), which can be used to optimize the class separability of different activity classes and improve the classification performance. The features are obtained by maximizing between-class and minimizing within-class variations. Here we call this approach canonical space transformation (CST). Combining EST based on PCA with CST based on CA, our approach reduces the data dimensionality and optimizes the class separability among different activity classes.

Image data in high-dimensional space are converted to low-dimensional eigenspace using EST. The obtained vector thus 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 $\mathbf{X}$ by subtracting $\mathbf{m}_{\mathrm{x}}$. And each image $\mathbf{x}_{i,j}$ forms a column of **X, t**hat is

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

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 rotate 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 [13], 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 $\mathbf{W}$ 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 [13], 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 $[\mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_{c-1}]$ 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

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

where $\mathbf{z}_{i,j}$ represents the new point and the orthogonal basis $[\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{c-1}]^{\mathrm{T}}$ 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 $\mathrm{H} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{c-1}]^{\mathrm{T}} [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k]^{\mathrm{T}}$.

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.

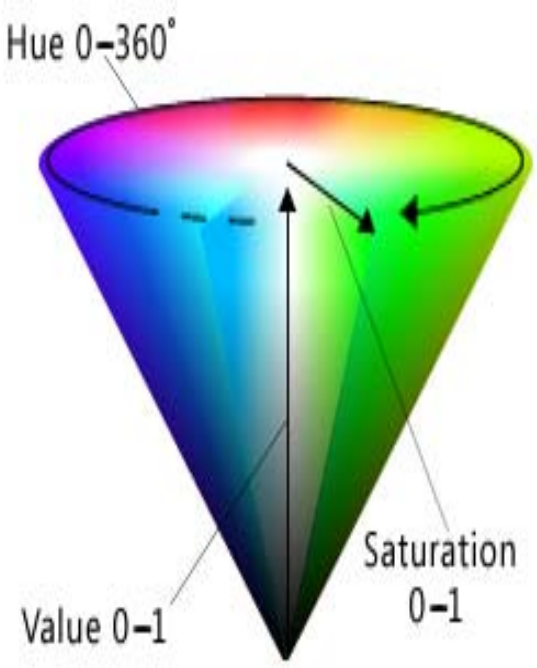


Fig. 2.1 The HSV Cone

The formula of RGB transfers to HSV is defined as :

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

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

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

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 [11]. Accordingly, Ohba *et al*. [14] 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 Taking Medicine Recognition System

The system flowchart is illustrated in Fig. 3.1. Next, we discuss “skin color detection,” “medical pouch color recognition,” and “activity recognition system” in detail.

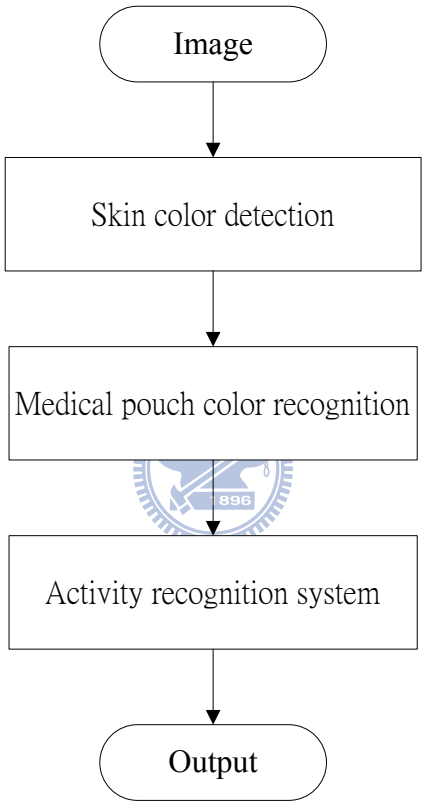


Fig. 3.1 The block diagram of taking medicine recognition system

Firstly, we build the grayscale value and the HSV color space background models in Scene 1, a normal view on medical pouch table. Then, our PTZ camera will zoom-in to become, a zoom-in of Scene 1, Scene 2 quickly because we do medical pouch color recognition. After the color recognition, the camera will return to Scene 1. Finally, we do activity recognition for taking medicine. Fig. 3.2 and Fig. 3.3 represent

Scene 1 and Scene 2, respectively.

Fig. 3.2 Scene 1, normal view on medical pouch table

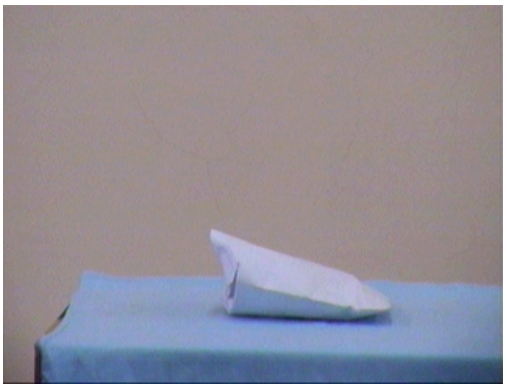


Fig. 3.3 Scene 2, zoom-in of Scene 1 and being used to recognize medical pouch's color.

3.1 Skin Color Detection

Later, we called Scene 1 as s_1 and Scene 2 as s_2. In our system, we have two zooms between s_1 and s_2. If the background models are ok completely, we will have the first scene zoom that is from s_1 to s_2. Then, the scene was still s_2 until our system finished the medical pouch color recognition. Otherwise, we make use of skin color detection to trigger the medical pouch color recognition. Next, we will discuss the skin color detection in detail.

By skin color detection, we can discriminate that there are someone or not anyone in s_2. First, the real-time image is transformed into the normalized RGB color space by

$$r = \frac{R}{R+G+B} \tag{3.1}$$

$$g = \frac{G}{R+G+B} \tag{3.2}$$

According to Soriano and Martinkauppi [15], a boundary condition of skin color in the r-g plane is defined as

$$f_{upper}(r) = -1.3767r^2 + 1.0743r + 0.1452 \tag{3.3}$$

$$f_{lower}(r) = -0.7760r^2 + 0.5601r + 0.1766 \tag{3.4}$$

If a pixel satisfies the following four conditions, it will be labeled as skin pixel; and further, we know there is a person in s_2.

$$g > f_{lower}(r) \ and \ g < f_{upper}(r) \tag{3.5}$$

$$(r-0.33)^2 + (g-0.33)^2 \geq 0.0004 \tag{3.6}$$

$$R > G > B \tag{3.7}$$

$$R - G \geq k_{skin} \tag{3.8}$$

where k_{skin} is a threshold.These detected skin pixels are belonged to hands because our camera focus on subject's hands and medical pouch. Fig. 3.4 shows an original image $I_{original}$ in s_2 that includes subject's hands and a green medical pouch.

Fig. 3.4 Original image $I_{original}$ in s_2

Next, we utilize above equations from Eqs. (3.1) - (3.8) to segment the skin pixels in $I_{original}$. Then, we can get an image I_{skin} from original image $I_{original}$ by

$$I_{skin}(x,y)=\begin{cases}255, & \text{if } I_{original}(x,y) \text{ is detected as skin pixel}\\ 0, & \text{otherwise}\end{cases} \tag{3.9}$$

where $I_{original}(x,y)$ is the intensity of a pixel which is located at (x,y), and $I_{skin}(x,y)$ is the segmented binary image by Eq. (3.9), as shown in Fig. 3.5.

Fig. 3.5 The binary image I_{skin}, in which white.

Since the medical pouch is carried by one's hands, we can first determine the

location of the subject's hands and then the color of medical pouch can be determined. According to the binary image I_{skin} segmented above, we further extract the skin region to minimize the image size to process. Skin region extraction can be accomplished by simply a thresholding on the occupied histograms in the X and Y directions of processing image. Figure 3.6 shows an example of skin region extraction. We project the binary image I_{skin} and to the X and Y directions. The interested section 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 four corners of a rectangle to locate subject's hands, and the medical pouch as well.

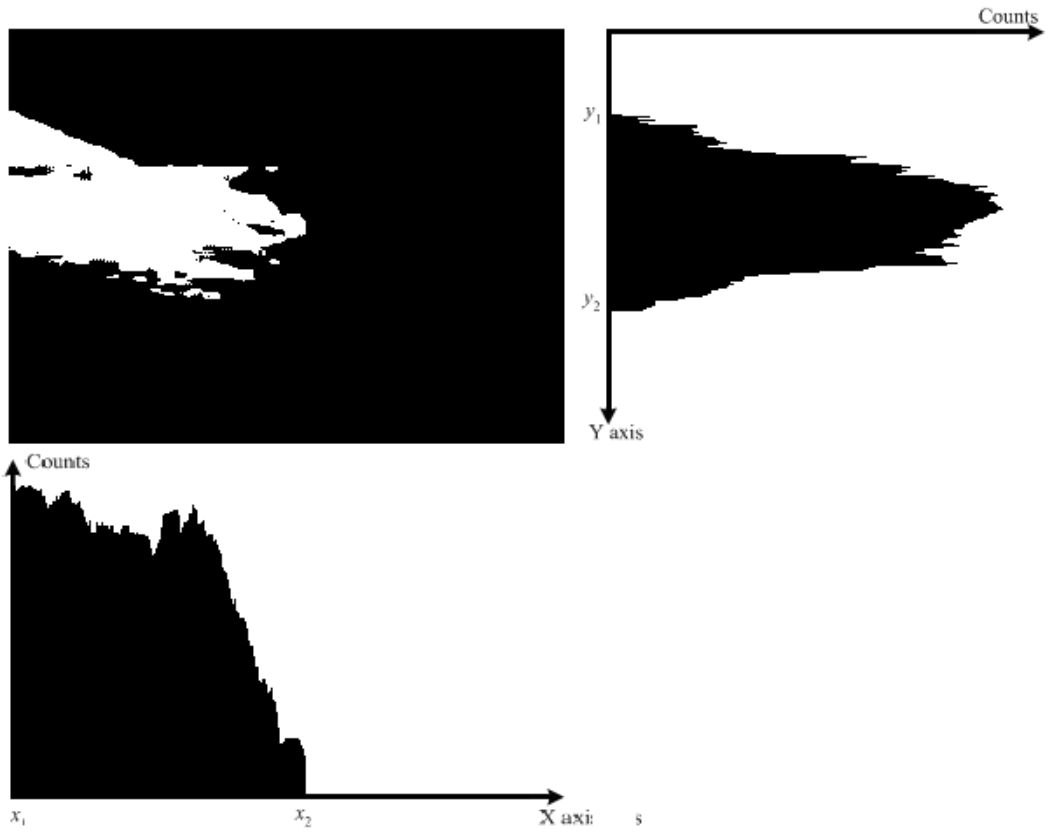


Fig. 3.6 Histogram of binary image I_{skin} projection in the X and Y directions.

3.2 Medical Pouch Color Recognition

That is, it is the location of subject's hands. According to the result of histogram of I_{skin}, Fig. 3.7 shows the region of subject's hands.

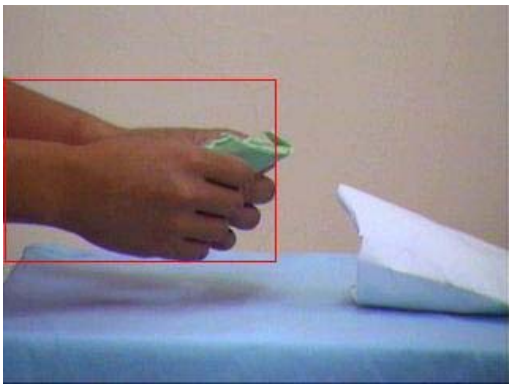


Fig. 3.7 A rectangular region is detected to confine the subject's hands

In Fig. 3.7, we find that the rectangle includes not only subject's hands but also subject's medical pouch. Thus, we can do medical pouch color recognition in the above rectangular region.

We will recognize medical pouch's color in the HSV color space. First, we make use of Eq. (2.16) of chapter 2 to transform pixels in the hand region into the HSV color space. In order to decrease the computation, we do not transform all pixels in the region. Only the pixels not belonged to the skin pixels are transformed.

The hue value can be a reliable clue to discriminate the color of a medical pouch. The colors of our medical pouches are light red, light green and light yellow. We extract image pixels of these color medical pouches, and plot the histogram of hue component, the highly counted regions around red, green, and yellow can specify the threshold boundaries for these three colors, respectively.

To detect the color of the medical pouch of an image, the pixels other than

belonging to the hand in the rectangular hand region are matching to the red, green, and yellow regions obtained above. The color bin with the maximal number of pixels is belonging to specify the color of medical pouch of the image. To be more reliable, we further utilize the dominant medical pouch's color obtained in seven consecutive images to specify the final medical pouch's color of the video clip. Fig. 3.8 shows the structure of the medical pouch color recognition.

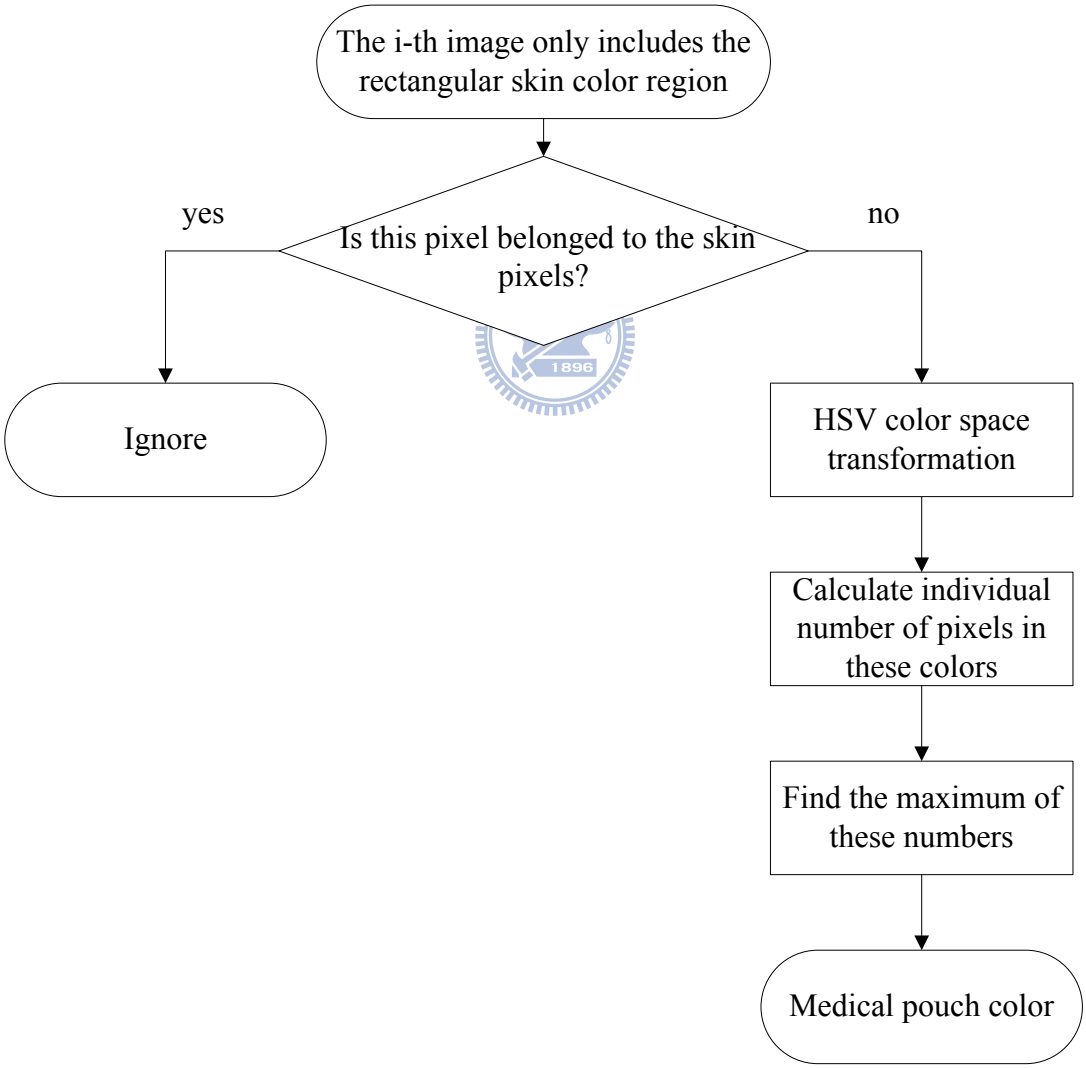


Fig. 3.8 The structure of the medical pouch color recognition in the i-th image

3.3 Human Activity Recognition System

3.3.1 Object Extraction

A. Background Model

First, we only build a grayscale value background model and find out it cannot detect reliably those foreground pixel whose grayscale values close to background pixel. In order to solve this problem, we also 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.

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 background model, $[m^{gray}(x,y), n^{gray}(x,y), d^{gray}(x,y)]$, of a pixel is obtained by

$$\begin{bmatrix} m^{gray}(x,y) \\ n^{gray}(x,y) \\ d^{gray}(x,y) \end{bmatrix} = \begin{bmatrix} \max_i \{I_i^{gray}(x,y)\} \\ \min_i \{I_i^{gray}(x,y)\} \\ \max_i \{|I_i^{gray}(x,y) - I_{i-1}^{gray}(x,y)|\} \end{bmatrix} \quad (3.10)$$

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

In the other hand, 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. Likewise, we use the same background video, for background model training. 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

$$\begin{bmatrix} m^H(x,y) \\ n^H(x,y) \\ d^H(x,y) \end{bmatrix} = \begin{bmatrix} \max_i \{I_i^H(x,y)\} \\ \min_i \{I_i^H(x,y)\} \\ \max_i \{|I_i^H(x,y) - I_{i-1}^H(x,y)|\} \end{bmatrix} \quad (3.11)$$

$$\begin{bmatrix} m^S(x,y) \\ n^S(x,y) \\ d^S(x,y) \end{bmatrix} = \begin{bmatrix} \max_i \{I_i^S(x,y)\} \\ \min_i \{I_i^S(x,y)\} \\ \max_i \{|I_i^S(x,y) - I_{i-1}^S(x,y)|\} \end{bmatrix} \quad (3.12)$$

$$\begin{bmatrix} m^V(x,y) \\ n^V(x,y) \\ d^V(x,y) \end{bmatrix} = \begin{cases} \begin{bmatrix} \max_i \{I_i^V(x,y)\} \\ \min_i \{I_i^V(x,y)\} \\ \max_i \left\{ \left| I_i^V(x,y) / I_{i-1}^V(x,y) \right| \right\} \end{bmatrix} & \text{if } I_i^V(x,y) / I_{i-1}^V(x,y) \geq 1 \\ \begin{bmatrix} \max_i \{I_i^V(x,y)\} \\ \min_i \{I_i^V(x,y)\} \\ \max_i \left\{ \left| I_{i-1}^V(x,y) / I_i^V(x,y) \right| \right\} \end{bmatrix} & \text{if } I_i^V(x,y) / I_{i-1}^V(x,y) < 1 \end{cases} \quad (3.13)$$

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

B. Extraction of Foreground Object

Fig. 3.9 shows the framework we apply to foreground subject extraction. Our framework of foreground subject extraction is composed of four components.

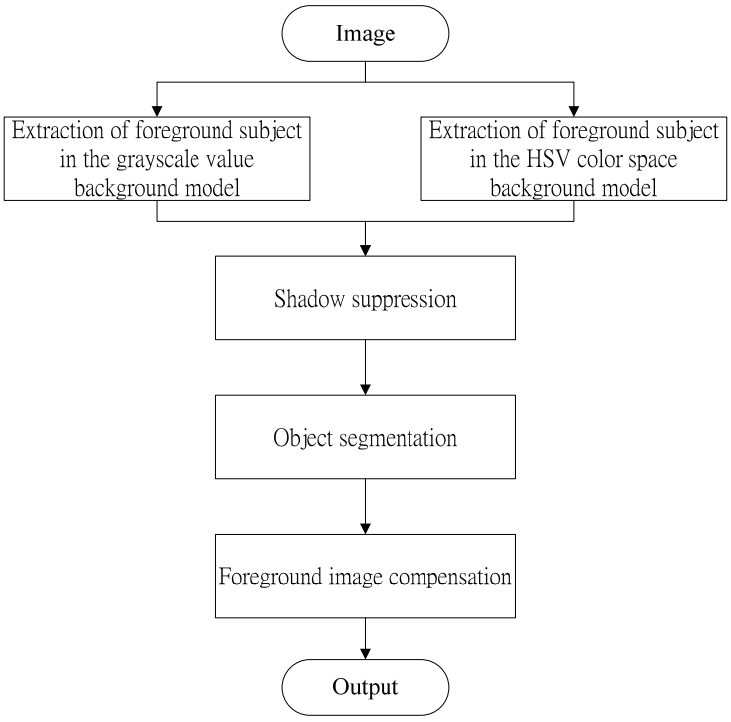


Fig. 3.9 The framework we apply to foreground subject extraction

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.

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

$$I^{1}{}_{foreground}(x,y) = \begin{cases} 0, & \text{if } I_i^t(x,y) < (m^{gray}(x,y) + k\mu) \\ & \text{and } I_i^t(x,y) > (n^{gray}(x,y) - k\mu) \\ 255, & \text{otherwise} \end{cases} \quad (3.14)$$

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 difference environments. The value of k affects the amount of information retained in binary image $I^{1}{}_{foreground}(x,y)$.

In the other hand, we utilize the maximum value $m^{V}(x, y)$, the 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

$$I^2{}_{foreground}(x,y)=\begin{cases}0, & \text{if } I_i^V(x,y)/m^V(x,y)<k_V d^V(x,y)\\ & \text{or } I_i^V(x,y)/n^V(x,y)<k_V d^V(x,y)\\ 255, & \text{otherwise}\end{cases} \quad (3.15)$$

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.

C. Shadow Suppression

The pixels of the moving cast shadows are easily detected as the foreground pixel in normal condition. Because the shadow pixels and the object pixels share two important visual features: motion model and detectability. For this reason, the moving shadows cause object merging and object shape distortion. Therefore, we need to remove the shadow by using a shadow filter. The detail of the shadow filter is in next paragraph.

First, we discuss 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 $-1\le m\le1,-1\le n\le1$ (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:

$$NCC(x,y) = \frac{ER(x,y)}{E_B(x,y)E_{T_{xy}}} \tag{3.16}$$

where

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

$$E_B(x,y) = \sqrt{\sum_{n=-1}^{1}\sum_{m=-1}^{1} B(x+m,y+n)^2} \tag{3.17}$$

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

If a pixel (x,y) is in a shadowed region, the NCC should be large (close to one), 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

$$S^1(x,y) = \begin{cases} \text{shadow}, & NCC(x,y) \geq L_{ncc} \text{ and } E_{T_{xy}} < E_B(x,y) \\ \text{foreground}, & \text{otherwise} \end{cases} \tag{3.18}$$

where $S^1(x,y)$ is the shadow mask to class the pixel in the moving cast shadow, and L_{ncc} is a fixed threshold. If L_{ncc} is low, several foreground pixels may be misclassified as shadow pixels. Otherwise, choosing a large value of L_{ncc}, then the actual shadow pixels may not be detected.

We know that shadow pixels have similar chromaticity, but lower brightness than the background model. Hence, we can detect the shadow from foreground subject in the HSV color space. We analyze only points belonging to possible moving object that are detected in the former step. We define another shadow mask S^2 for each (x,y) point as follows:

$$S^2(x,y)=\begin{cases}\text{shadow}, & \text{if } I_i^V(x,y)-n^V(x,y)<0 \\ & \text{and}\left|I_i^H(x,y)-m^H(x,y)\right|<k_H d^H(x,y) \\ & \text{and}\left|I_i^S(x,y)-m^S(x,y)\right|<k_S d^S(x,y) \\ \text{foreground}, & \text{otherwise}\end{cases} \tag{3.19}$$

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 the shadow mask to class the pixel in the moving cast shadow. Values k_S and k_H are selected threshold values used to measure the similarities of the hue and saturation between the background image and the current observed image. Finally, the foreground subject is defined as:

$$I_{foreground}(x,y)=S^1(x,y)\vee S^2(x,y) \tag{3.20}$$

D. Object Segmentation

According to the binary image $I_{foreground}$ segmented by above, we extract the region of foreground object to minimize the image size. Foreground region extraction can be accomplished by simply introducing a threshold on the histograms in X and Y direction. Fig. 3.10 shows an example of foreground region extraction. We utilize the binary image and project it to X and Y directions. The interested section 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 four corners of a rectangle to extract foreground region and the size of this rectangle is adjusted to 128×96. Fig. 3.11 is the extracted foreground region.

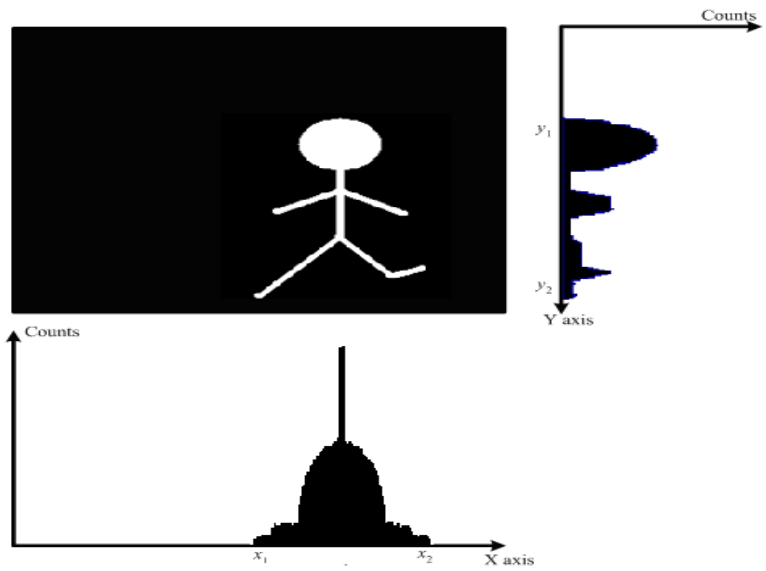


Fig. 3.10 Histogram of binary image projection in X and Y direction.

Fig. 3.11 The binary image of extracted foreground region.

E. Foreground Image Compensation

Detecting all foreground pixels and removing all shadows simultaneously are difficult. When we want to remove shadow pixels, some foreground data will be lost and this makes the foreground image be broken. Therefore, we will repair the foreground image by opening filter and closing filter.

3.3.2 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 this real-time video does not vary for a long time and there is nobody in the scene. By Eq. (3.10), we can calculate how many times the binary values remain unchanged.

$$update(x,y) = \begin{cases} update(x,y)+1, & \text{if } I^{t-1}_{foreground}(x,y) = I^{t}_{foreground}(x,y) \\ update(x,y), & \text{otherwise} \end{cases} \quad (3.21)$$

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.

3.3.3 Activity Template Selection

A human body is a rigid body, thus has individual natural frequency; namely, it has restriction on action speed when doing some specific actions. Because cameras usually capture image frames in a high frequency, i.e., 30 frames /sec, there are few differences between two consecutive postural image frames in a short interval. Therefore, we select some key posture frames from a sequence to describe an activity, i.e., our sample rate is 6 frames /sec. In our approach, we select one image frame, called as the essential template image, with a fixed interval instead of each image, i.e., our interval is 0.167 sec. An example is shown in Fig. 3.12. After selecting the

templates, each action is represented by several essential templates.

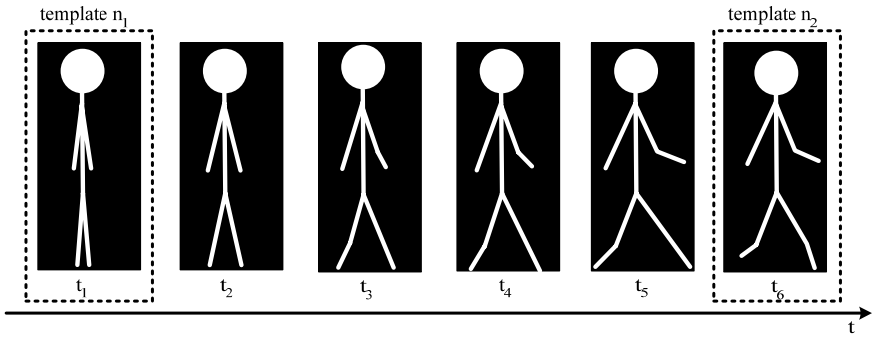


Fig. 3.12 One image frame is selected as template with an interval.

By eigenspace transformation (EST) and canonical space transformation (CST), these essential templates are transformed to a new space. 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 in the new space that is 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.

As described in Section 2.1, each image frame is transformed to a (c–1)-dimensional vector by EST and CST methods. 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

$$\mathbf{t}_{i,j} = \mathbf{H} \cdot \mathbf{g}_{i,j}, \qquad i = 1, 2, \cdots, c; \quad j = 1, 2, \cdots, n \tag{3.22}$$

where **H** denotes the transformation matrix combing EST and 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

$$\mathbf{t}_{i,j} = \left[t_{i,j}^{1}, t_{i,j}^{2}, \cdots, t_{i,j}^{c-1}\right]^{\mathrm{T}} \tag{3.23}$$

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

$$\boldsymbol{\mu}_i = \frac{1}{n}\sum_{j=1}^{n}\mathbf{t}_{i,j} \tag{3.24}$$

where i is the number of template categories. The standard deviation vector of the m-th dimension is computed by

$$\sigma_m = \sqrt{\frac{\sum_{i=1}^{c}\sum_{j=1}^{n}\left(t_{i,j}^{m} - \mu_i^{m}\right)^2}{N_t - 1}} \tag{3.25}$$

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

3.3.4 Construction of Fuzzy Rules form Video Stream

For human activity classification, transitional relationships of postures in a temporal sequence are important information. Human's actions may have similar postures in two different activity sequences, and therefore only using one image frame to classify the action is not sufficient. For example, the actions of "jumping" and "crouching" both have the same postures called common states as shown in Fig. 3.13. Besides, the posture sequence of each activity is dissimilar in different people.

Hence, we propose a method which not only combines temporal sequence information for recognition but also is tolerant to variations of different people. We use the fuzzy rule-base approach to design our system. The fuzzy rule-base approach also has been proposed in gesture recognition in [16]; it has ability to absorb data difference by learning.

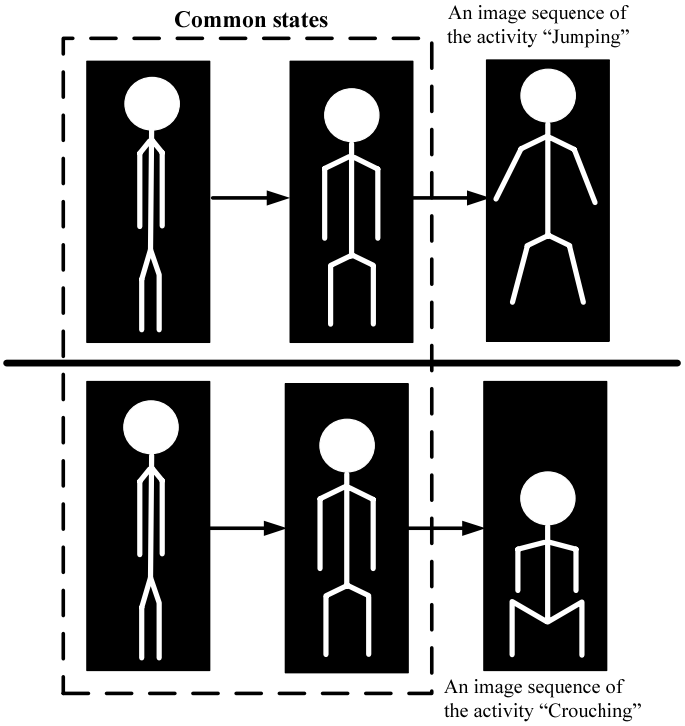


Fig. 3.13 Common states of two different activities.

We use the membership function to represent the feature's possibility of each cluster. We choose the Gaussian type membership function to represent the features because the Gaussian type membership function can reflect the similarity via the first order and second order statistics of clusters and is differentiable.

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

$$\mathbf{a}_k = \mathbf{H}\,\mathbf{x}_k. \tag{3.26}$$

where $\mathbf{H}$ denotes the transformation matrix combing EST and CST. As the same as $\mathbf{t}_{i,j}$ in Eq.(3.21), $\mathbf{a}_k$ can be rewritten as

$$\mathbf{a}_k = \left[a_k^1, a_k^2, \cdots, a_k^{c-1}\right]^{\mathrm{T}}. \tag{3.27}$$

If we suppose the dimensions of the feature vectors are independent, a local measure of similarity between the training vector and each template vectors can be computed. 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

$$\begin{aligned} r_{i,k} &= \mathrm{M}(\mathbf{a_k}|C_i) \\ &= \frac{1}{(2\pi)^{\frac{c-1}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}(\mathbf{a_k}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{a_k}-\boldsymbol{\mu})\right] \\ &= \arg\max_{\mathrm{j}}\left\{\prod_{m=1}^{c-1}\frac{1}{\sqrt{2\pi}\sigma_m}\exp\left[-\frac{1}{2}\sum_{m=1}^{c-1}\frac{(a_k^m-\mu_{i,j}^m)^2}{\sigma_m^2}\right]\right\} \end{aligned} \tag{3.28}$$

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. Besides, we can obtain which category each image belongs to by

$$p_k = \arg\max_i r_{i,k} \tag{3.29}$$

The membership function describes the probability of which one it is like most. But it just contains the information of a single image. Hence, we collect three images to form a basis for temporal information.

Assume we have c linguistic labels, each linguistic label represent a category of essential template. Each image frame can be represented by one of these c linguistic labels. Here, we combine three contiguous images to a group (I_1, I_2, I_3) and the interval of itself and next is 0.167 sec. The transformation of the image group can form a feature vector $[a_1, a_2, a_3]$.There are c^3 combinations of the feature vector. Each combination represents the possible transition states of the three images. We use Eqs. (3.26) and (3.27) to class each image frame. Hence, we can represent the feature vector $([a_1, a_2, a_3])$ by linguistic label sequence $([P_1, P_2, P_3])$. An image sequence with linguistic label sequence is associated with its output of corresponding activity.

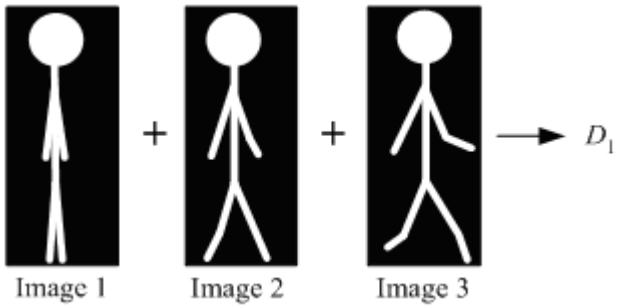
As developed by Wang and Mendel [17], fuzzy rules can be generated by learning from examples. Such image sequence constitutes an input-output pair to be learned in the fuzzy rule base. In this setting, the generated rules are a series of associations of the form

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

The number of antecedent conditions equals the number of features. Note that

antecedent conditions are connected by "**AND**." For example, an image sequence, its transformations of image 1, image 2, image 3 and belonging categories being concatenated as vector format, is given by

$$[P_1, P_2, P_3; D_1] \tag{3.30}$$



Suppose that Image 1, Image2 and Image 3 belong to key posture 1, key posture 2 and key posture 3 respectively. Therefore, we assign the image sequences, whose feature vector is $[\mathbf{a}_1^1, \mathbf{a}_2^1, \mathbf{a}_3^1]$, to the linguistic labels Posture 1, Posture 2 and Posture 3 respectively. Finally, according to the feature-target association implies this image sequence to support the rule of

Rule 1. IF the activity's I_1 is P_1^1 AND its I_2 is P_2^1 AND its I_3 is P_3^1, THEN the activity is D_1. (3.31)

After the learning steps of action video, some rules that obtained enough member of supporting fire strength may be representative to describe an action in video. In this thesis, a rule with at least four supporting input image frames is selected and compiled to constitute the knowledge rule base of our action recognition system. During the training of image sequences, we can compute the mean and standard deviation of each pre-defined activity.

3.3.5 Classification algorithm

After constructing the rule base, we can grade the input image sequence with each fuzzy rule by grade of membership function. Let $\mathbf{\Sigma}$ denote the covariance matrix of all essential template vectors, C_i denote the i-th class of essential templates and $\mathbf{s}_k$ denote the image frame transformed by EST and CST. The membership function is given by

$$\begin{aligned} r_{i,k} &= \mathrm{M}(s_{\mathbf{k}} | C_i) \\ &= \frac{1}{(2\pi)^{\frac{c-1}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2} (\mathbf{s_k} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{s_k} - \boldsymbol{\mu}) \right] \\ &= \arg\max_{\mathrm{j}} \left\{ \prod_{m=1}^{c-1} \frac{1}{\sqrt{2\pi}\sigma_m} \exp\left[-\frac{1}{2} \sum_{m=1}^{c-1} \frac{(s_k^m - \mu_{i,j}^m)^2}{\sigma_m^2} \right] \right\} \end{aligned} \tag{3.32}$$

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. σ is the standard deviation of all essential templates. These membership functions are just the results of one image frame. Therefore, we use two transformed vector of passed image frames, which are called $\mathbf{a}_{k-2}$ and $\mathbf{a}_{k-1}$. Then, we set these three vectors as a feature vector $[\mathbf{a}_{k-2}, \mathbf{a}_{k-1}, \mathbf{a}_k]$ and compute the membership functions of them respectively.

In order to calculate the similarity between image sequence and each postural sequence in the training data base, we take out the membership functions r_{k-2,n_1}, r_{k-1,n_2} and r_{k,n_3} which are corresponding to the three category of linguistic labels, P_{n_1}, P_{n_2} and P_{n_3}, in the rule and have been calculated by Eq. (3.29). The summation of r_{k-2,n_1}, r_{k-1,n_2} and r_{k,n_3} is the similarity between current image sequence and the postural sequence of this rule. We can obtain the similarity related to all fuzzy rules of training data base in the same manner. The rule, which has the highest value of similarity, is selected.

Chapter 4 Experimental Results

In our experiment, we tested our system on videos taken by PTZ camera. We took the video in our laboratory at the 5th Engineering Building in NCTU campus. The light source is fluorescent lamp and is stable. The background is not complex and we equip a table in the scene. The camera is set up at a fixed location and kept stationary. This camera has a frame rate of thirty frames per second and image resolution is 320×240 pixels. Scene 1 is a normal view on medical pouch table, and Scene 2 is the zoom-in of Scene 1. Fig. 4.1 and Fig. 4.2 represent Scene 1 and Scene 2, respectively.

Fig. 4.1 Scene 1, normal view on medical pouch table

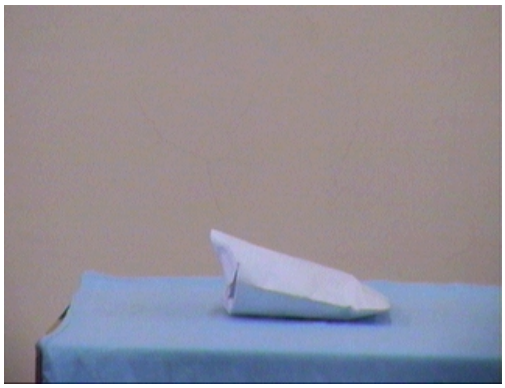


Fig. 4.2 Scene 2, zoom-in of Scene 1 and being used to recognize medical pouch's color.

In our recognition systems, we have eight training actions: “walking from right to left,” “walking from left to right,” “walking straightly,” “reading ,” “using computer,” “sleeping,” “taking medicine,” and “picking up.”

4.1 Skin Color Detection and Medical Pouch Color Detection

4.1.1 Skin Color Detection

Skin color detection is used for segmenting the object’s hands. If we segment the skin region more precisely, we can extract object’s hands more correctly. A threshold k_{skin} is applied in Eq. (3.8) described in Section 3.1 to obtain binary image $B(x, y)$. The value of k_{skin} is chosen by experiment and varies with different environments. Hence, we ran a series of experiments to determine the optical threshold k_{skin} and the corresponding binary images are shown in Fig. 4.3. The threshold k_{skin}=45 was adopted in our experiment.

Hand region is extracted from binary image $B(x, y)$ in order to minimize the size of images. Hand region extraction is accomplished by simply taking a threshold along X and Y directions. Fig. 4.4 shows an example of hand region extraction. Fig. 4.4(a) is a image frame of the video stream. Figure 4.4(b) is the binary image after performing background model analysis. Figures 4.4(c) and 4.4(d) show the projection of Fig. 4.4(b) onto the X and Y directions, respectively. We can find the boundary coordinates of the X and Y directions by observing the projection histogram. We used these boundary coordinates to define a rectangle to extract foreground region from Fig. 4.4(b). Fig. 4.4(e) is the extracted foreground region.

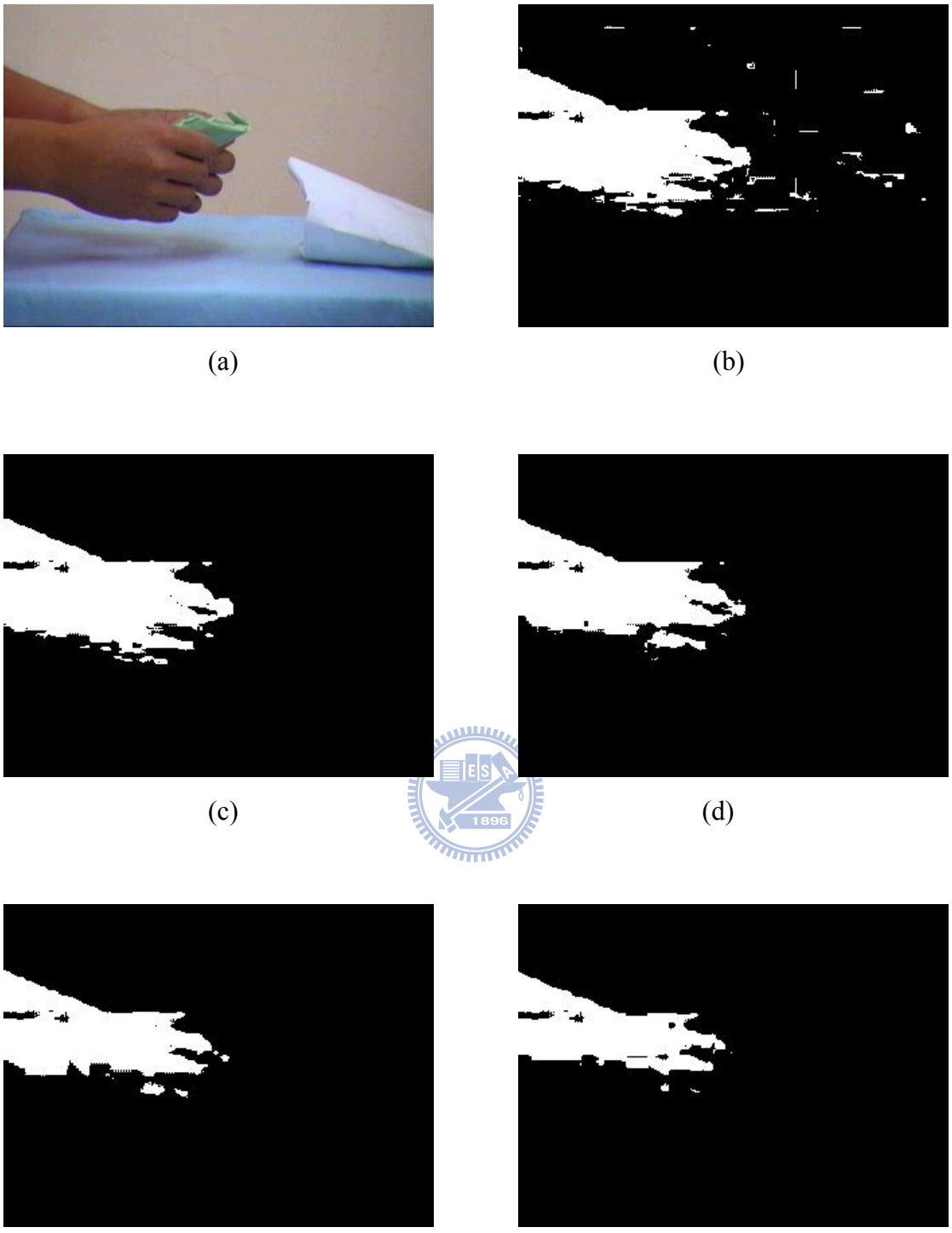
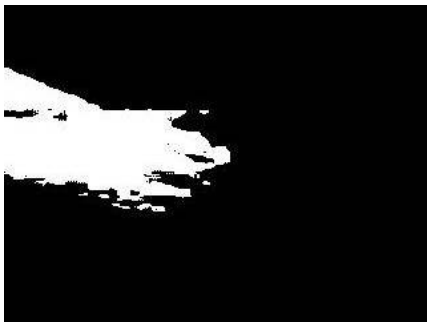
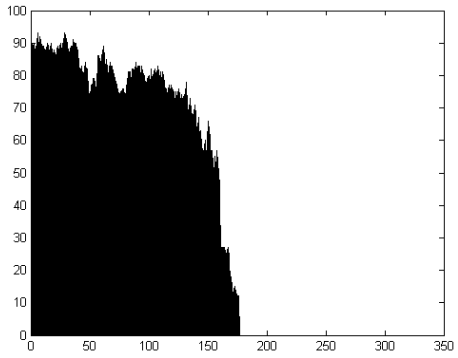


Fig. 4.3 Example of skin color detection at different threshold, k_{skin}, values. (a) An image frame, (b) k_{skin}=40, (c) k_{skin}=45, (d) k_{skin}=50, (e) k_{skin}=55, (f) k_{skin}=60.

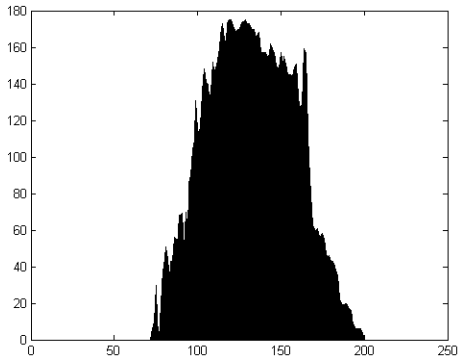
(a)



(b)



(c)



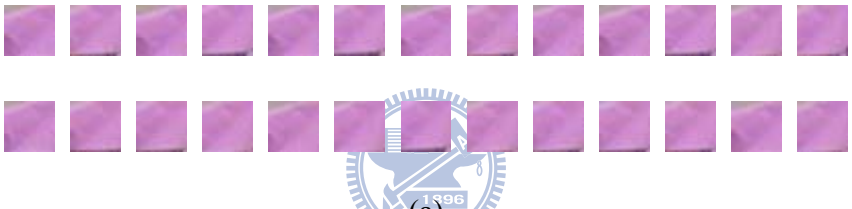
(d)

(e)

Fig. 4.4 An example of hand region extraction. (a) An image frame, (b) binary image after skin color analysis, (c) projection of (b) onto X direction, (d) projection of (b) onto Y direction, (e) hand region extracted.

4.1.2 Medical Pouch Color Detection

For medical pouch color detection, we have to analyse the hue component of red, green, and yellow. Thus, we use three hundred 25×25 images for each color, respectively. First, we plot the histogram of hue component by using all the above images. Then, we eliminate the first 5% and the last 5% in these data, and get a new range for each color. For red, its hue component value is between 296 and 317. For green, its hue component value is between 114 and 169. For yellow, its hue component value is between 37 and 58. We show some images in these analytic data in Fig. 4.5. These histogram plots are shown Figs. 4.6 – 4.8.



(a)

(b)

(c)

Fig. 4.5 Some pouch's color images in analytic phase. (a) red data, (b) green data, (c)yellow data.

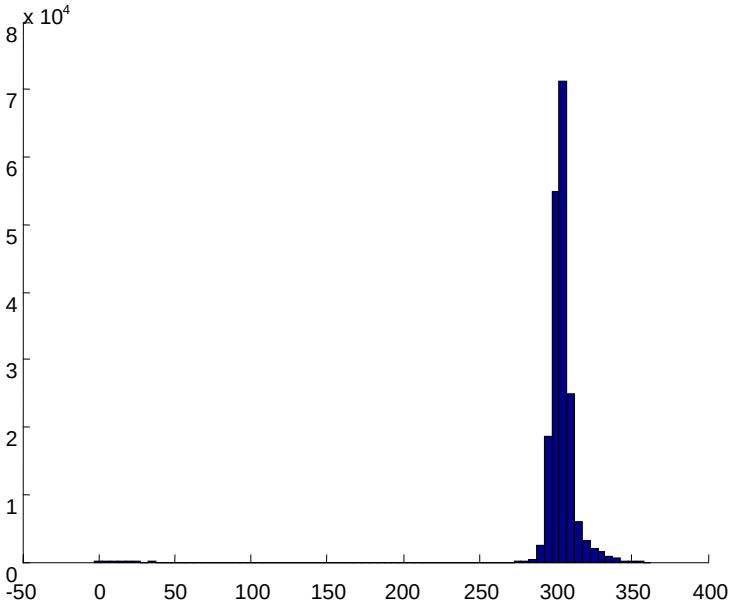


Fig. 4.6 Histogram plot of hue component in the red data.

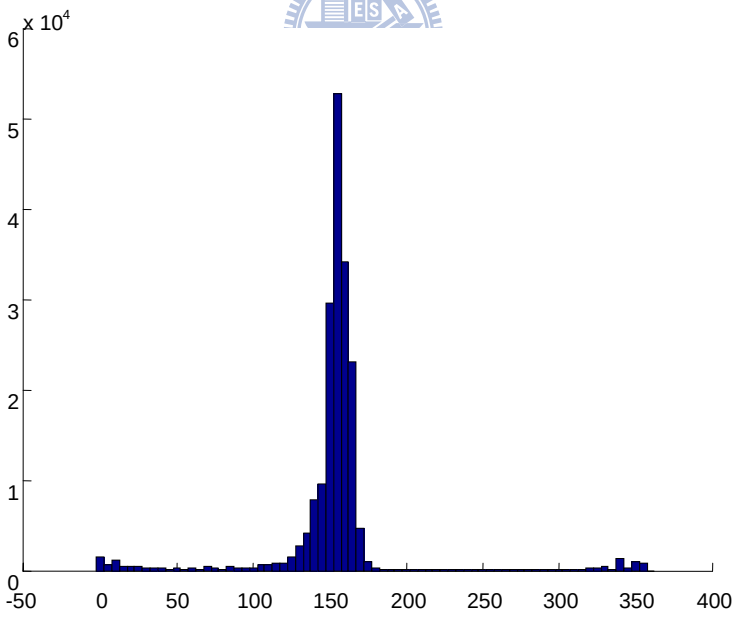


Fig. 4.7 Histogram plot of hue component in the green data.

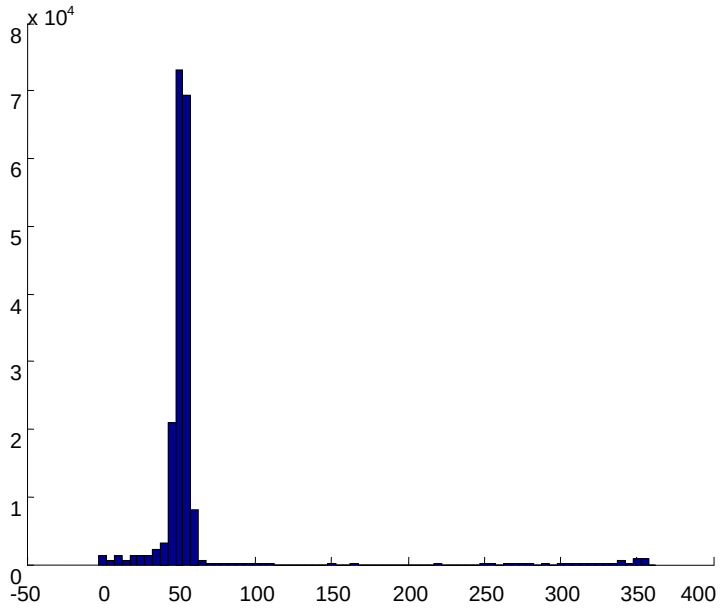


Fig. 4.8 Histogram plot of hue component in the yellow data.

We recognize medical pouch's color twenty times for each color, and we use seven consecutive images for one color recognition. Then, we suppose the recognized time is N_r, and the consecutive time is N_c.We get two recognition rates, the first recognition rate acc_1 is defines as

$$acc_1 = \frac{N_{correct1}}{N_r \times N_c} \tag{4.1}$$

where $N_{correct1}$ is the number of correct color recognition in all recognized images, N_r is twenty, and N_c is seven.

The second recognition rate acc_2, majority voting in the seven consecutive images, is defined as

$$acc_2 = \frac{N_{correct2}}{N_r} \tag{4.2}$$

where $N_{correct2}$ is the number of correct color recognition in twenty recognitions. Table I shows the color recognition result of the recognition rates.

TABLE I

COLOR RECOGNITION RESULT OF THE RECOGNITION RATES

	acc_1	acc_2
Red	79.3%	85.0%
Green	70.7%	80.0%
Yellow	77.9%	85.0%
Average	76.0%	83.3%

4.2 Background Model and Object Extraction

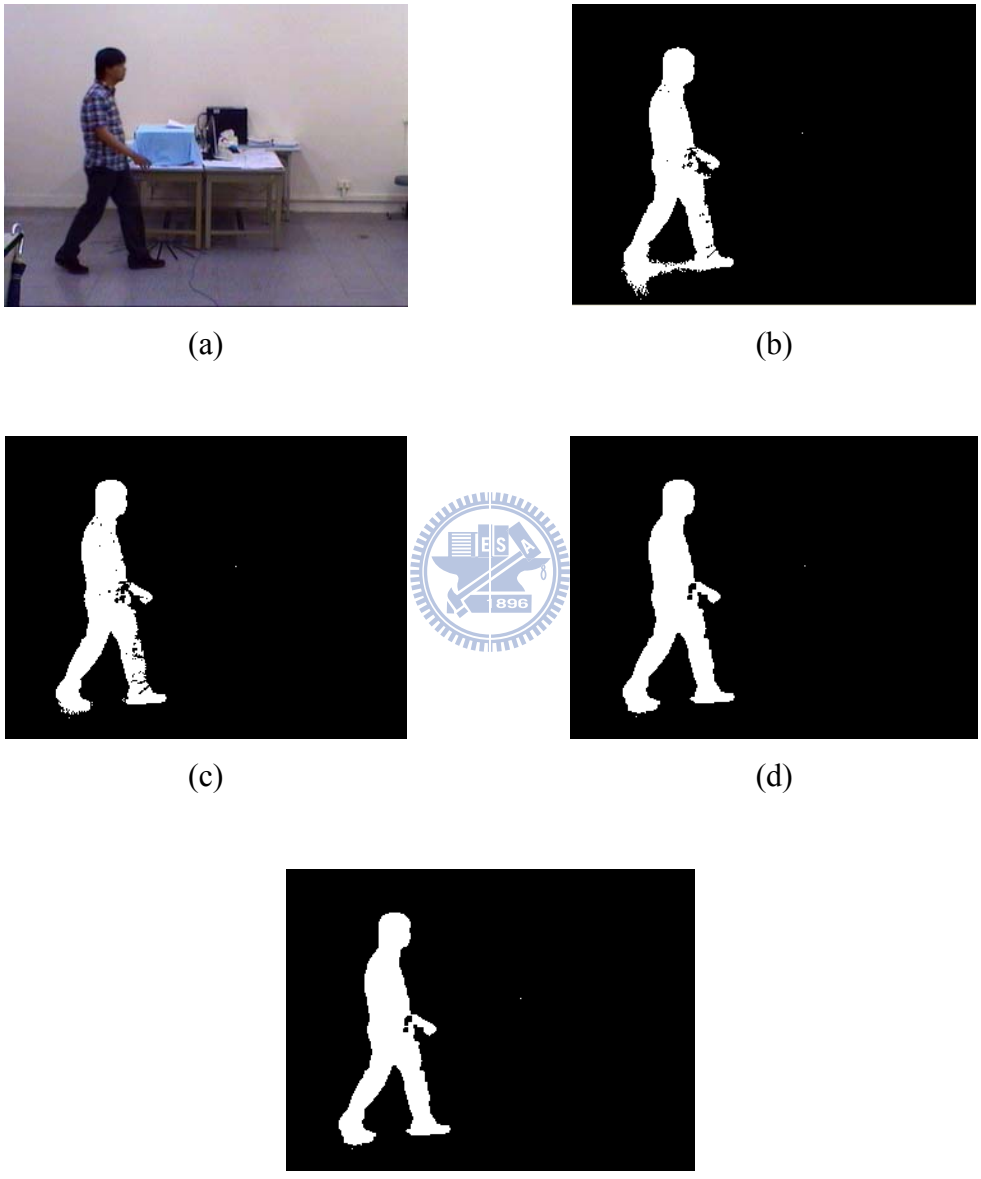
A background model is used for segmenting the foreground subject or object. In our system, we first record a video of pure background with no subject in Scene 1 and it is used as background models. If the grayscale value and the HSV color space background models are ok completely, we will extract the foreground pixels by using Eq. (3.14) and Eq. (3.15) in Section 3.3.1. Then, we continue to emend the former foreground image by using the shadow filter, the closing filter, and the opening filter.

In order to get the optimal result of object extraction, we have to adjust some parameters in our system. In chapter 2, we set thresholding parameters $H_t = 25$, $S_t = 40$, and $V_t = 40$ to distinguish hue component correctly. In the grayscale value and the HSV color space background models, we set $k = 2.3$ in Eq. (3.14) and $k_v = 1.4$ in Eq. (3.15) to extract foreground pixels. In the grayscale value space, we set $L_{ncc} = 0.995$ in Eq. (3.18) to detect shadow pixels. And in the HSV color space, we set $k_H = 1.3$ and $k_s = 1.3$ in Eq. (3.19) to detect shadow pixels.

Figure 4.9 shows an example of foreground extraction. Fig. 4.9(a) is an image frame of the video stream. Fig. 4.9(b) is the binary image after performing background model analysis without shadow filter. So, we detect subject's shadow as foreground pixels. Fig. 4.9(c) shows the result of using shadow filter in Fig. 4.9(b). Figure 4.9(d) shows the result of using closing filter in Fig. 4.9(c). Fig. 4.9(e) shows the result of using opening filter in Fig. 4.9(d) and it is the final result.

Likewise, we minimize the size of foreground images shown as Fig. 4.9(e) by simply taking a threshold along the X and Y directions. Fig. 4.10 shows an example of foreground region extraction. Fig. 4.10(a) is a image frame of the video stream. Fig. 4.10(b) is the binary image after performing background model analysis. Figs. 4.10(c)

and 4.10(d) show the projection of Fig. 4.10(b) onto the X and Y directions, respectively. We can find the boundary coordinates of X and Y directions by observing the projection histogram. We used these boundary coordinates to define a rectangle to extract foreground region from Fig. 4.10(b). Fig. 4.10(e) is the extracted foreground region.



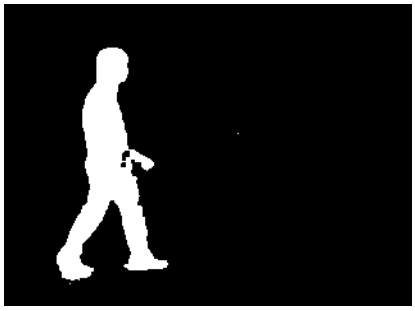
(a) (b)

(c) (d)

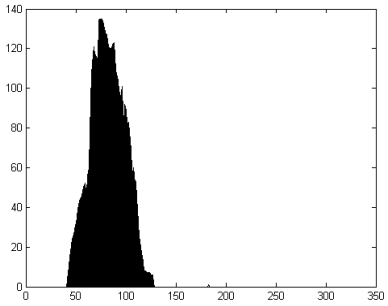
(e)

Fig. 4.9 An example of foreground extraction (a) An image frame, (b) after using background models, (c) after using shadow filter, (d) after using closing filter, (e) after using opening filter.

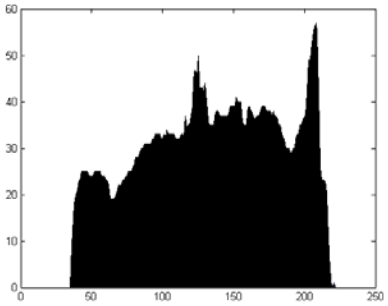
(a)



(b)



(c)



(d)

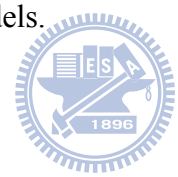
(e)

Fig. 4.10 An example of foreground region extraction. (a) An image frame, (b) binary image after background analysis, (c) projection of (b) onto X direction, (d) projection of (b) onto Y direction, (e) foreground region extracted.

4.3 Fuzzy Rule Construction for Action Recognition

We construct the template model and the fuzzy rule database with the training data. We chose six kinds of essential templates for “walking from right to left,” and “walking from left to right,” respectively; four for “taking medicine,” three for “walking straight,” and “picking up,” respectively; one for “reading,” “using computer,” and “sleeping,” respectively.

There are total 25 kinds of essential templates, and called 25 classes. The essential template numbers of each activity depend on how complexity it takes. Each essential template is a cluster with four template images which are from four different training persons and have similar postures. Fig 4.11 and Fig. 4.12 are two examples of some templates of two training models.



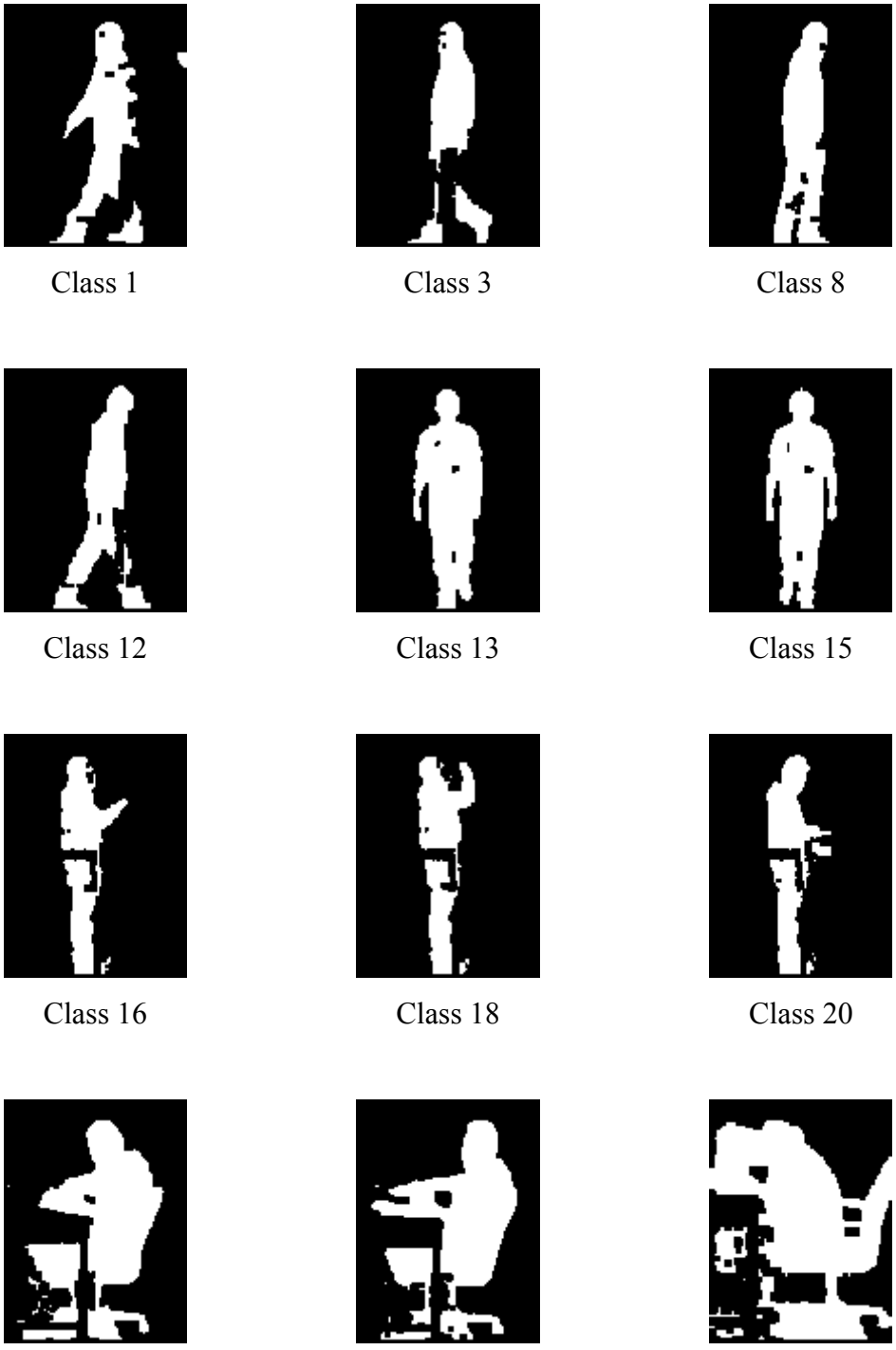


Fig. 4.11 Some “essential templates of posture” of model 1.

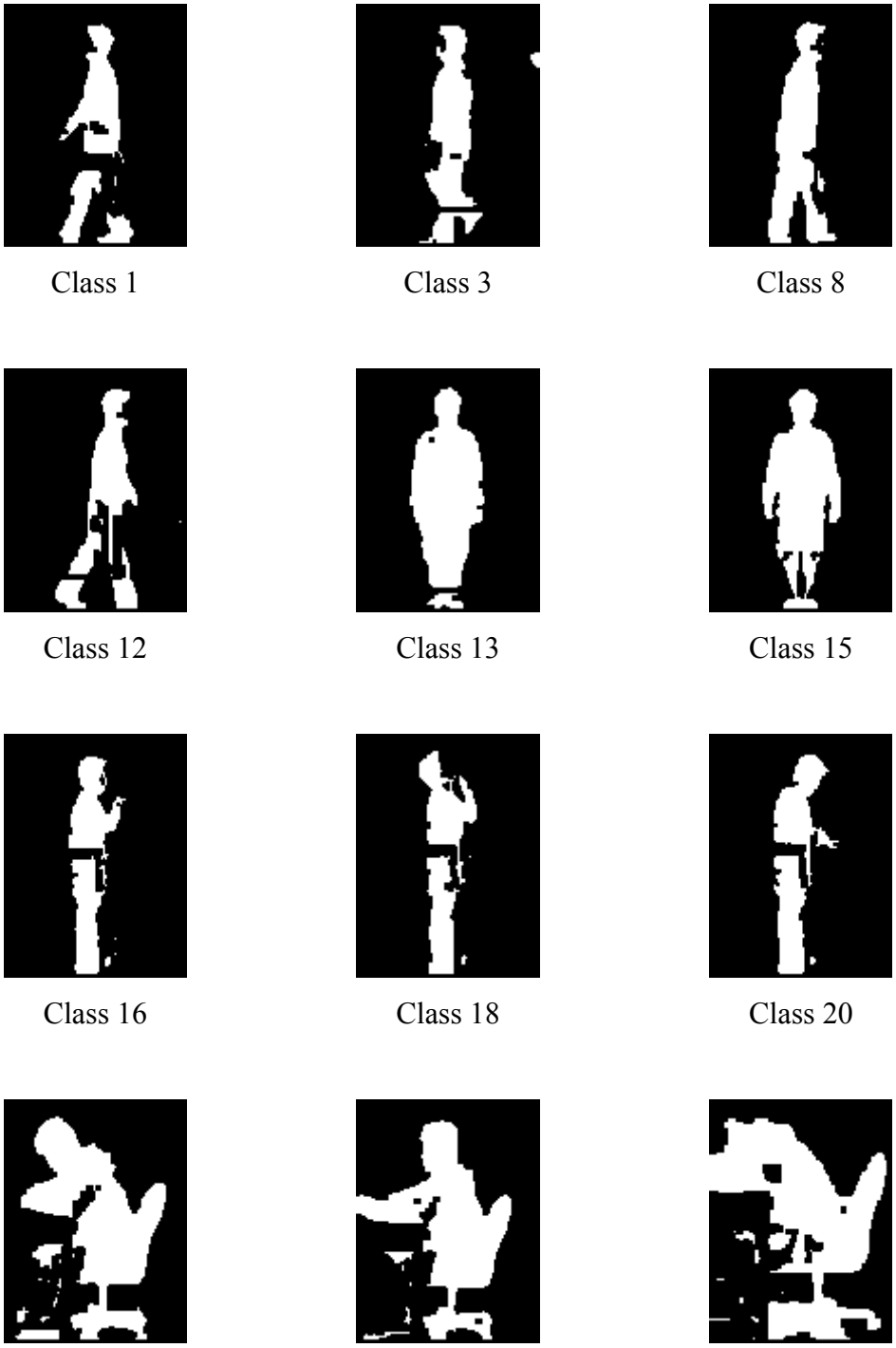


Fig. 4.12 Corresponding “essential templates of posture,” Fig. 4.11, of model 2.

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. (3.28) in Section 3.3.4. We gathered three images as a group in order to include temporal information. The interval between each of these three images is five image frames which is the same as in template selection. Training is accomplished in off-line situation. Therefore, we gathered three images from different start points to train fuzzy rules. For examples: the first frame, the 6-th frame and 11-th frame are gathered together as an input training data; the second frame, the 7-th frame and 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 and of training video may not be the same. By utilizing different start points, the system is able to learn much more combinations of image frames and increase accuracy of fuzzy rules.

The group of the three images is converted to the posture sequence which has the maximum summation of three membership function values in Eq. (3.28). Each posture sequence will trigger a corresponding rule one time. If the corresponding rule is not existent, a new rule is built in the form of **IF-THEN** which is represented in Section 3.3.4.

4.4 The Recognition Rate of Activities

The activity recognition system can operate in not only off-line videos but also real time videos. In order to calculate the recognition rate of activities, we use off-line videos and each of them includes only one activity 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 four video databases for training and testing.

An example of recognition rate of a testing video start from different frames is shown in Table II. In this table, W_{RL} is the activity "walking from right to left," W_{LR} is the activity "walking from left to right," W_S is the activity "walking straight," R_{EAD} is the activity "reading," U_{CP} is the activity "using computer," S_{LEEP} is the activity "sleeping," T_{AKE} is the activity "taking medicine," P_{ICK} is the activity "picking up." Here, the recognition rate is the number of correct recognition divide by the total number of recognition for each video.

TABLE II

The Recognition Rate of Person 2 with Different Starting Frame

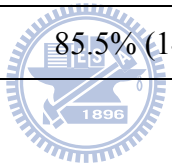
Starting frame	Recognition rate (%)							
	W_{RL}	W_{LR}	W_S	R_{EAD}	U_{CP}	S_{LEEP}	T_{AKE}	P_{ICK}
From the 1st, 6th, … frame	83.3%	58.3%	66.7%	90.0%	80.0%	100%	90.9%	90.0%
From the 2nd, 7th, … frame	66.7%	50.0%	88.9%	100%	100%	100%	90.9%	100%
From the 3rd, 8th, … frame	83.3%	66.7%	88.9%	90.0%	100%	100%	100%	90.0%
From the 4th, 9th, … frame	75.0%	66.7%	100%	100%	90.0%	100%	90.9%	90.0%
From the 5th, 10th, … frame	58.3%	58.3%	88.9%	100%	100%	100%	100%	90.0%

Table III shows the recognition rate, four folds cross validation, of each activity of each model. If we test these videos in Model 1, we will constructed the templates and fuzzy rules by used the order three models. That is, the testing video was not used for constructing templates and fuzzy rules.

TABLE III

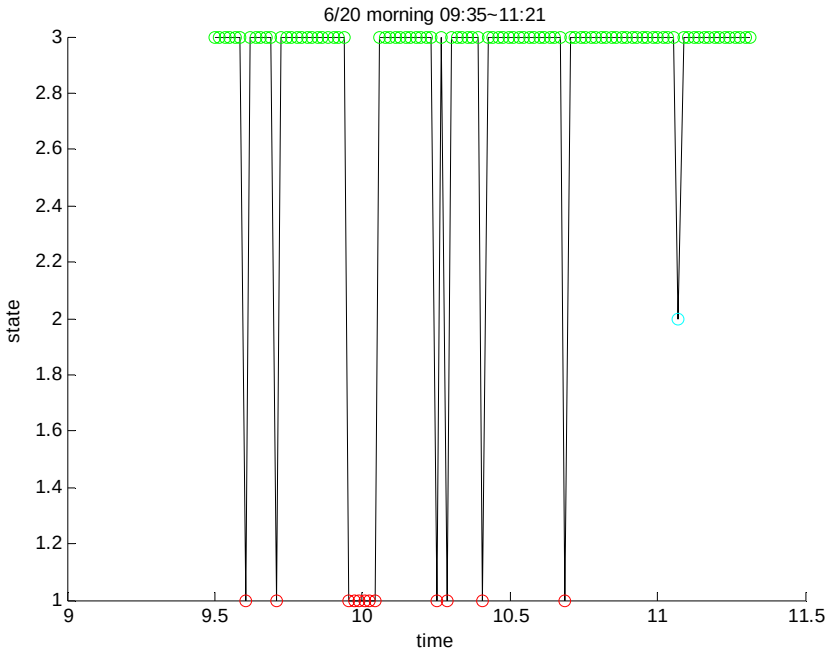
THE RECOGNITION RATES OF FOUR FOLDS CROSS VALIDATION OF EACH ACTIVITY

	Model 1	Model 2	Model 3	Model 4
W_{RL}	77.7% (66/85)	73.3% (44/60)	61.3% (38/62)	63.2% (43/68)
W_{LR}	56.0% (42/75)	60.0% (36/60)	69.1% (38/55)	66.1% (39/59)
W_{S}	86.7% (39/45)	86.7% (39/45)	86.7% (39/45)	93.3% (42/45)
R_{EAD}	100% (50/50)	96.0% (48/50)	94.0% (47/50/)	97.8% (44/45)
U_{CP}	98.0% (49/50)	94.0% (47/50)	96.0% (48/50/)	94.0% (47/50)
S_{LEEP}	100% (45/45)	100% (50/50)	100% (50/50)	100% (45/45)
T_{AKE}	96.0% (48/50)	94.5% (52/55)	90.9% (50/55)	96.0% (48/50)
P_{ICK}	90.0% (45/50)	92.0% (46/50)	90.0% (45/50)	88.9% (40/45)
Average	85.5% (1449/1694)			

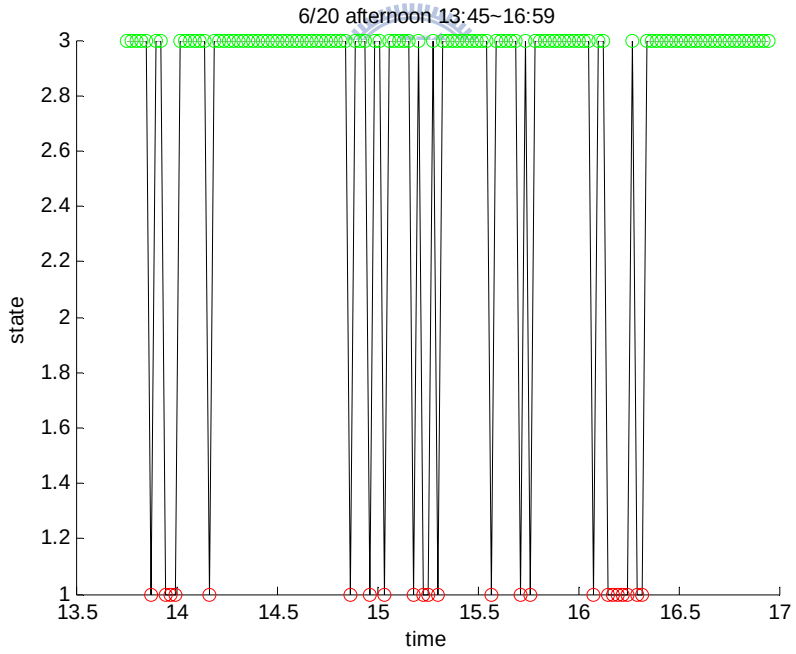


4.5 The Activities of Daily Living

We use the above activity recognition system to record the activities of daily living. In our experiment, we record the daily living of a student in the laboratory, and the common activities we used are “reading,” “walking straight,” “using computer,” and “sleeping.” And we represent the above four activities as state 1, state 2, state 3, and state 4, respectively. We record of activities of daily living for five days and plot these data in Figs. 4.14 – 4.17. In these figures, the abscissa is represented as time and the ordinate is represented as current action. For example, a point (13.5, 3) recording computer represents subject is in state 3 at 13:30; that is, subject is “using computer” at 13:30. Moreover, the interval between two consecutive time points is 1 minute.

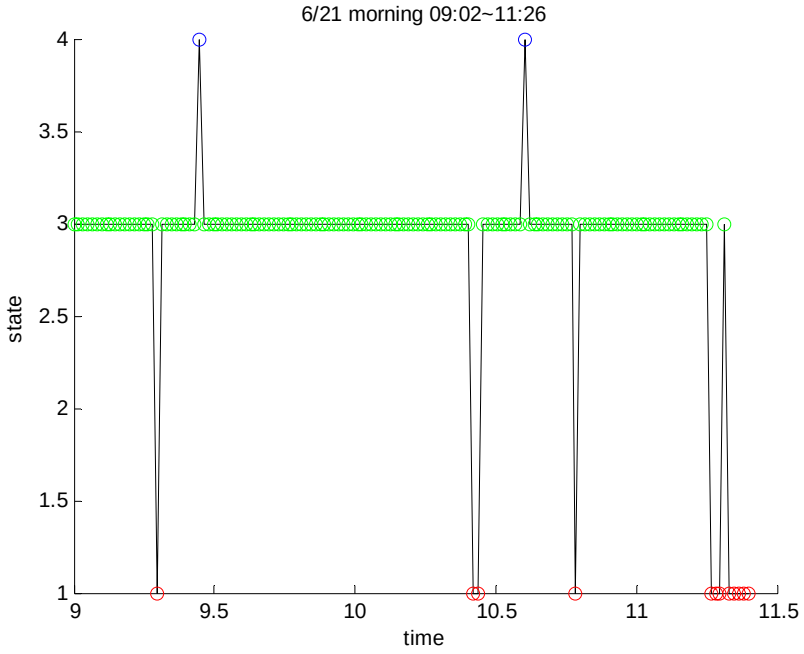


(a)

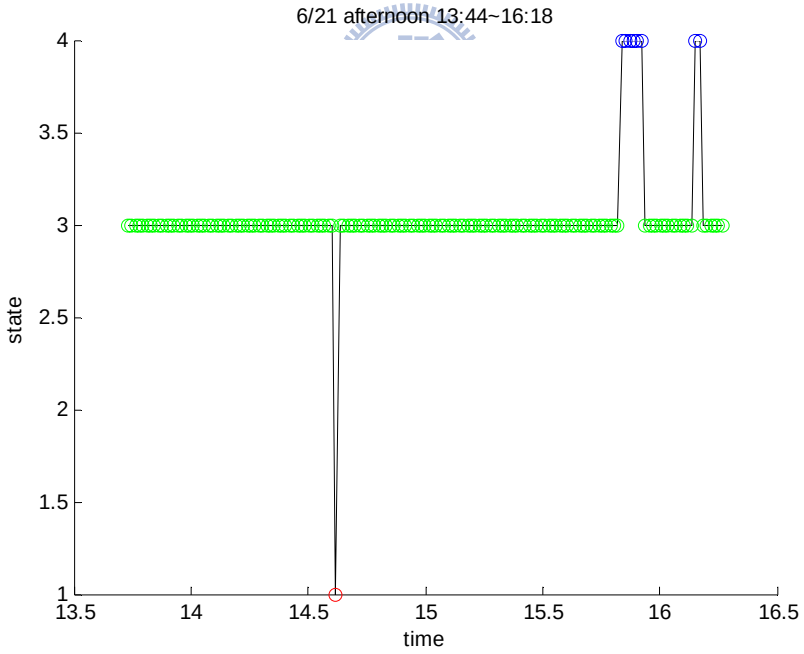


(b)

Fig. 4.13 The activities of daily living in (a) the morning of 6/20, (b) the afternoon of 6/20.

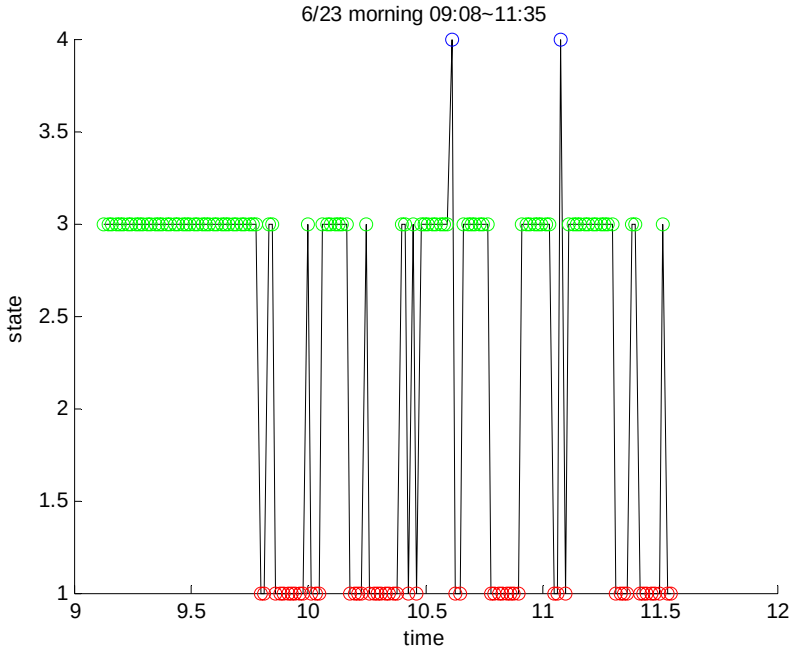


(a)

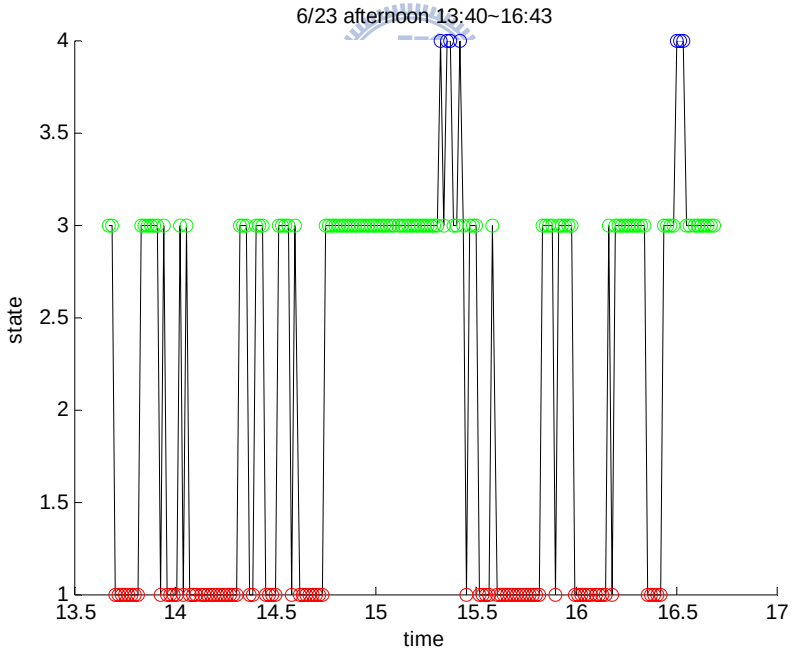


(b)

Fig. 4.14 The activities of daily living in (a) the morning of 6/21, (b) the afternoon of 6/21.

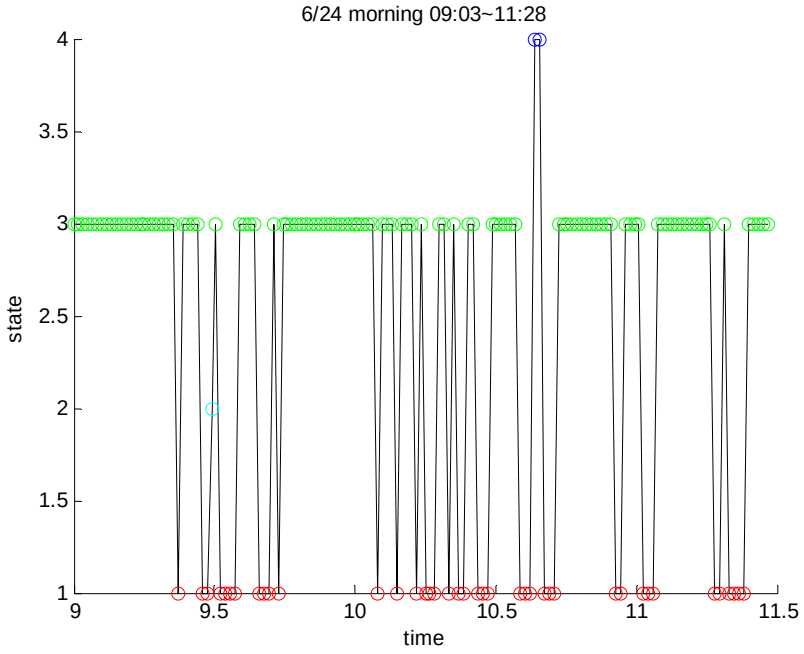


(a)

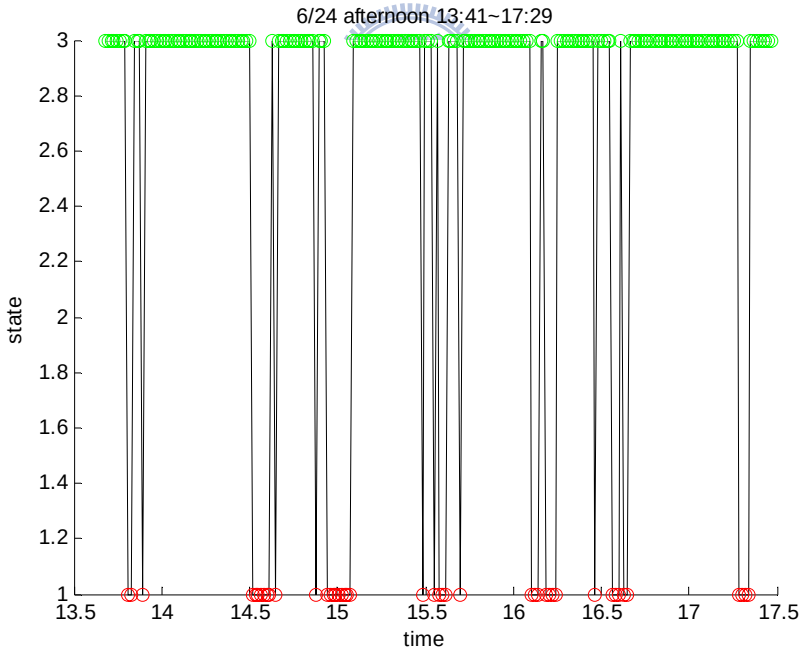


(b)

Fig. 4.15 The activities of daily living in (a) the morning of 6/23, (b) the afternoon of 6/23.

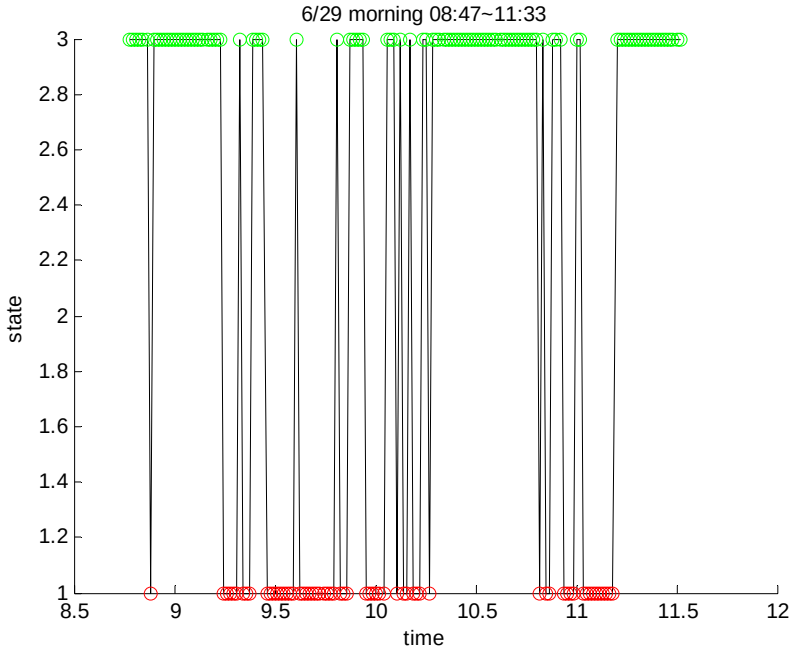


(a)

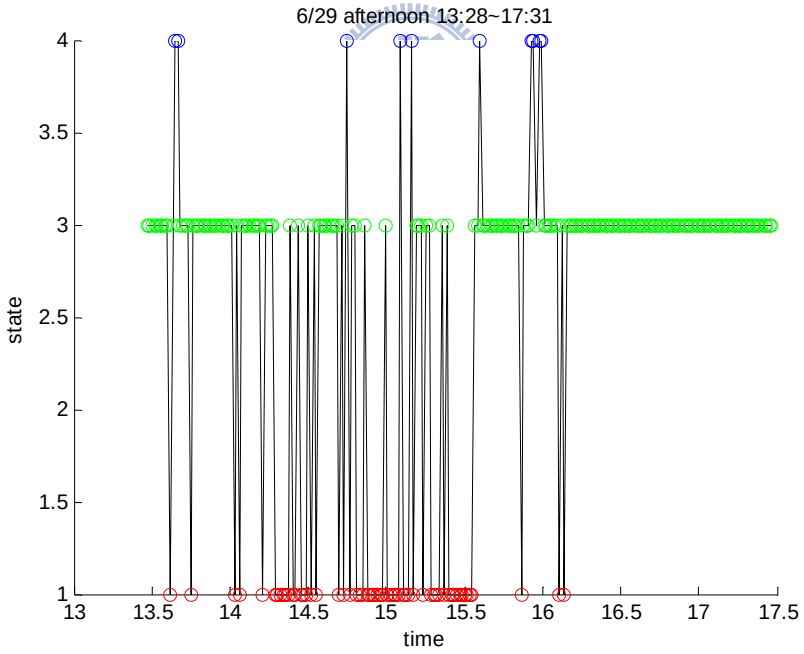


(b)

Fig. 4.16 The activities of daily living in (a) the morning of 6/24, (b) the afternoon of 6/24.



(a)



(b)

Fig. 4.17 The activities of daily living in (a) the morning of 6/29, (b) the afternoon of 6/29.

On view of the ADL results above, we validate with the recording data. In these data, we suppose the record rate is R_{record} per minute and get R_{record} outputs in each minute. Then, we use majority vote to decide which action is in these outputs in the 1 minute. After we get the recognized action in each minute, we calculate the ratio of the action to these R_{record} outputs and let the ratio be accurancy. For example, the recognized action exists Num_{action} times and the accurrancy is $Num_{action} / R_{record}$ in the 1 min. Table IV shows the recognition rate of five days of each activity, and the average of the recording rate is 27.9 per minute (47802 outputs/1711 minutes).

TABLE IV

THE RECOGNITION RATES OF FIVE DAYS DATA OF EACH ACTIVITY

	Recognition rate
R_{EAD}	82.6% (8892/10767)
W_S	53.1% (43/81)
U_{CP}	77.1% (27851/36104)
S_{LEEP}	80.1% (681/850)
Average	78.4% (37467/47802)

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 by a statistical background model based on frame ratio, which is robust to illumination changes. For better efficiency and separability, the binary image is firstly transformed to a new space by eigenspace and then canonical space transformation, and the recognition is finally done in canonical space. A three image frame sequence, 5:1 down sampling from the video, is converted to a key posture sequence by template matching. The key posture sequence is classified to an action by fuzzy rules inference. Fuzzy rule approach can not only combine temporal sequence information for recognition but also absorb deviations due to variations of action done by different people and different time.

Moreover, we make use of the hue component to recognize the medical pouch's color when one is taking medicine. By combining with the hue-based pouch's color model and human activity recognition system, we can know someone is taking medicine and its medical pouch's color as well. Finally, we also employ the activity recognition system to record a student's daily living.

Experimental results have shown that the recognition rate for medical pouch's color classification is 83.3% and the recognition rate for eight actions classification is 85.5%.

Our system is promising in elder care and nursing home. In the activities of daily living, we can use our system to build one's model in one's daily life. Because the activities of elder are slower and simpler, we can use our system to observe the elder's activity of daily living and maybe get some regularities in the record. Then, we can analyze their daily living whether some abnormal activity occur or not. A discrepancy from a normal profile of one's ADL could be a significant signal for one's health. This technique is certainly helpful for elder person automatic taking care and nursing home application.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

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

[15] 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.

[16] M. C. Su, “A fuzzy rule-based approach to spatio-temporal hand gesture recognition,” *IEEE Trans. Syst., Man Cybern*, vol. 30, no. 2, pp. 276–281, May 2000.

[17] 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, Nov

1992.