國 立 交 通 大 學

電 控 工 程 研 究 所

碩 士 論 文

全天候之人臉與動作辨識及其於睡著與清醒偵測

Day and Night Face and Action Recognition and Its Application to Sleep/Awake Detection

研 究 生：歐 瑞 賢

指 導 教 授：張 志 永

中 華 民 國 一百零一 年 七 月

全天候之人臉與動作辨識及其於睡著與清醒偵測

Day and Night Face and Action Recognition and Its Application to Sleep/Awake Detection

學　　生：歐瑞賢　　Student : Rui- Xian Ou

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

全天候之人臉與動作辨識及其於睡著與清醒偵測

學生:歐瑞賢　　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

本論文實現了一套結合人臉辨識、動作辨識與清醒或睡著判別的自動化居家看護系統。首先的人臉與動作辨識工作，待測影像是分別藉由背景相剪法與 Haar 疊層分類器產生。為了能抽取出更完整的前景影像，我們分別在灰階與 HSV 空間建立背景模型。Haar 疊層分類器是一種基於特徵運算的演算法，這種演算法比基於逐點運算的更快速。接著影像將藉由特徵空間與標準空間轉換被投影到一個讓不同類別影像的區別性更大且維度較小的空間。

動作與人臉辨式分別利用模糊法則推論與 FisherFace 方法來實現。為了將時間軸上的資訊包含進來，我們結合從動作視訊 5:1 減低抽樣連續三張影像來訓練建立動作辨識模糊法則，並用之推論動作辨識工作。在清醒判別系統中，影像首先會藉由照度隨中心遞減公式來校正。接著利用移動估測方法來量化測試者在睡眠中的活動程度並進一步判定他的清醒/睡著狀態。

Day and Night Face and Action Recognition and Its Application to Sleep/Awake Detection

STUDENT: Rui-Xian Ou ADVISOR: Dr. Jyh-Yeong Chang

Institute of Electrical and Control Engineering
National Chiao-Tung University

ABSTRACT

In this thesis, we implement an automatic home health care system that combines the face, action and sleep/awake recognition of a person in day and night. The test images are extracted by background subtraction embedded in an action recognition system and then by Haar cascade classifier for face recognition. We build two background models in grayscale and HSV color space to extract the foreground images correctly. Haar cascade classifier for face is a feature-based algorithm that works much faster than the pixel-based algorithm. Then, the test images are transformed to a new space by eigenspace and canonical space projection for better efficiency and separability.

Face and action and recognition is implemented by using FisherFace method and fuzzy rule inference, respectively. We gather three consecutive images 5:1 down-sampled from activity video to construct fuzzy rules inference for containing temporal information to recognize the action. In sleep/awake detection, the LCD NIR images will be rectified by using the function of illumination variation firstly. Then, the motion estimation is utilized to quantify the activity degree of a sleeper to determine one's sleep/awake state.

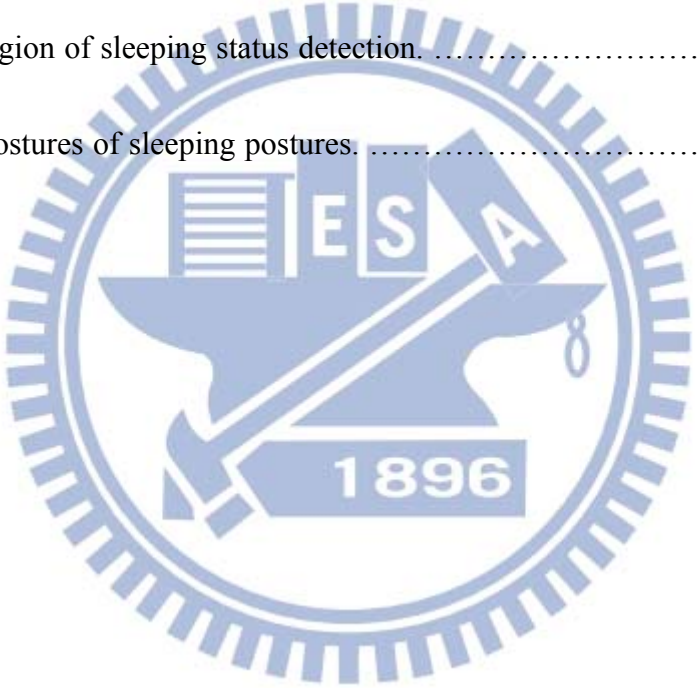
ACKNOWLEDGEMENTS

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

The importance of home nursing care increases with the coming of aged society and the trend of fewer children. Most of the home nursing care service is provided by professional people, such as nurse. However, the service cost is maybe expensive and the nurse cannot look after elderly people in 24 hours. Therefore, the home automatic health care system become a popular research area in recent years. The care system not only can record user's information, which is a reference of diagnosis but also make a response in time when emergency happened.

In this thesis, we design a home health care system which includes the following: face and activity recognition system and video-based sleep/awake detection system in one's home environment. 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 or dangerous human activities. Finally, we combine face identification with activity recognition system to enhance its effectiveness. We hope that the system can recognize a person in his home and also recognize and record his activity in the daily living environment.

Our life in daytime is affected by the quality of sleep during the previous night. The working efficiency will be decreased when the sleep is disturbed. Sleep disorders also causes many different diseases. Therefore, sleep quality can be an index of personal health both physically and mentally. The system in this thesis can detect sleep/awake status of a user by video frames. Cameras are usually utilized to study sleep disorders in recent years because it is cheaper and less intrusive than traditional devices, such as [1], [2].

1.2 Face and Activity Recognition System

The first step of activity recognition system is foreground subject extraction. The method for subject extraction exploited in this thesis is the background subtraction. It 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 [3] where many different approaches were proposed. In our system, we construct two background models for more correct subject extraction; one is based on grayscale value and the other is based on HSV color space. After subtracting each pixel value of background model from that current image frame, the resulting image is converted to a binary image by setting a threshold. 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. Then, the rectangle image is resized to the specified resolution for normalization.

In most of video and image processing, the size of image frame is usually very

large and an image frame usually contains a great deal of redundancy. 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 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 and Principal Component Analysis. Our transformation method combines eigenspace transformation and canonical space transformation which are described as follows.

The eigenspace transformation (EST), which uses principal components analysis (PCA) for dimensionality reduction, generates projection directions that maximize the total scatter across all classes. It has been demonstrated to be a potent scheme used for automatic face recognition [4], [5] and action recognition [6]. 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 point in a low-dimensional canonical space.

Then, we group three contiguous 5:1 down-sampled images and transform them to three consecutive feature vector by EST and CST. The time-sequential images are classified to a posture sequence by using these three feature vectors. 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 that best matches the observed posture sequence is chosen as the

recognized action category.

We propose a fuzzy rule-base approach for human activity recognition, in which we employ the fuzzy rule based learning of Wang and Mendel [7] for action recognition. In our activity recognition system, activity rule base is represented in the form of fuzzy IF-THEN rules, extracted from the posture sequences of the training data of activity video. Each IF-THEN rule is fuzzified by employing an innovative membership function in order to represent the degree of the similarity between a three-posture pattern and the corresponding antecedent part of action type. When our system classifies an unknown action, it will match the three consecutive sampled images of the video frames by each fuzzy rule learned before. The rule with the largest accumulated similarity measure associated with the above three consecutive postures is selected and then its consequent part of action type defined the current action type of the person. The system flowchart is depicted in Fig. 1.1.

The face detection is an important step before face recognition. The purpose of face detection is to localize and extract the face region from the extracted foreground. Firstly, Haar cascade classifier of OpenCV [10] is employed to detect the face region. The classifier is proposed by Viola et al. [8]. Skin detection is also utilized to locate the face position in our system. Then we recognize face images by Eigenface and Fisherface method [9]. The rationale behind Eigenface and Fisherface are eigenspace and canonical space transformation. Finally, the class of the most similar training face is the system output. The system flowchart is depicted in Fig. 1.2.

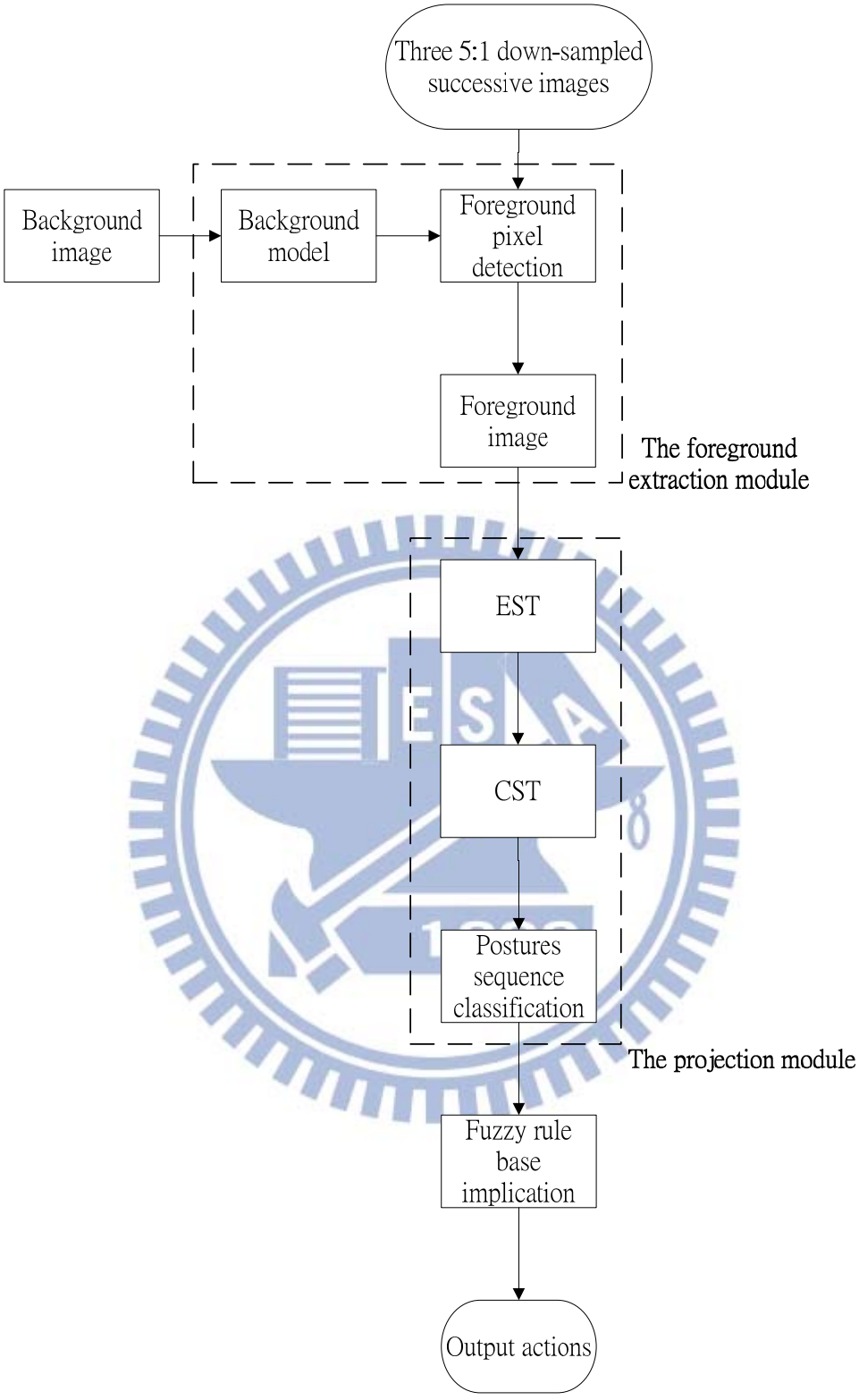


Fig. 1.1 The block diagram of human action recognition system.

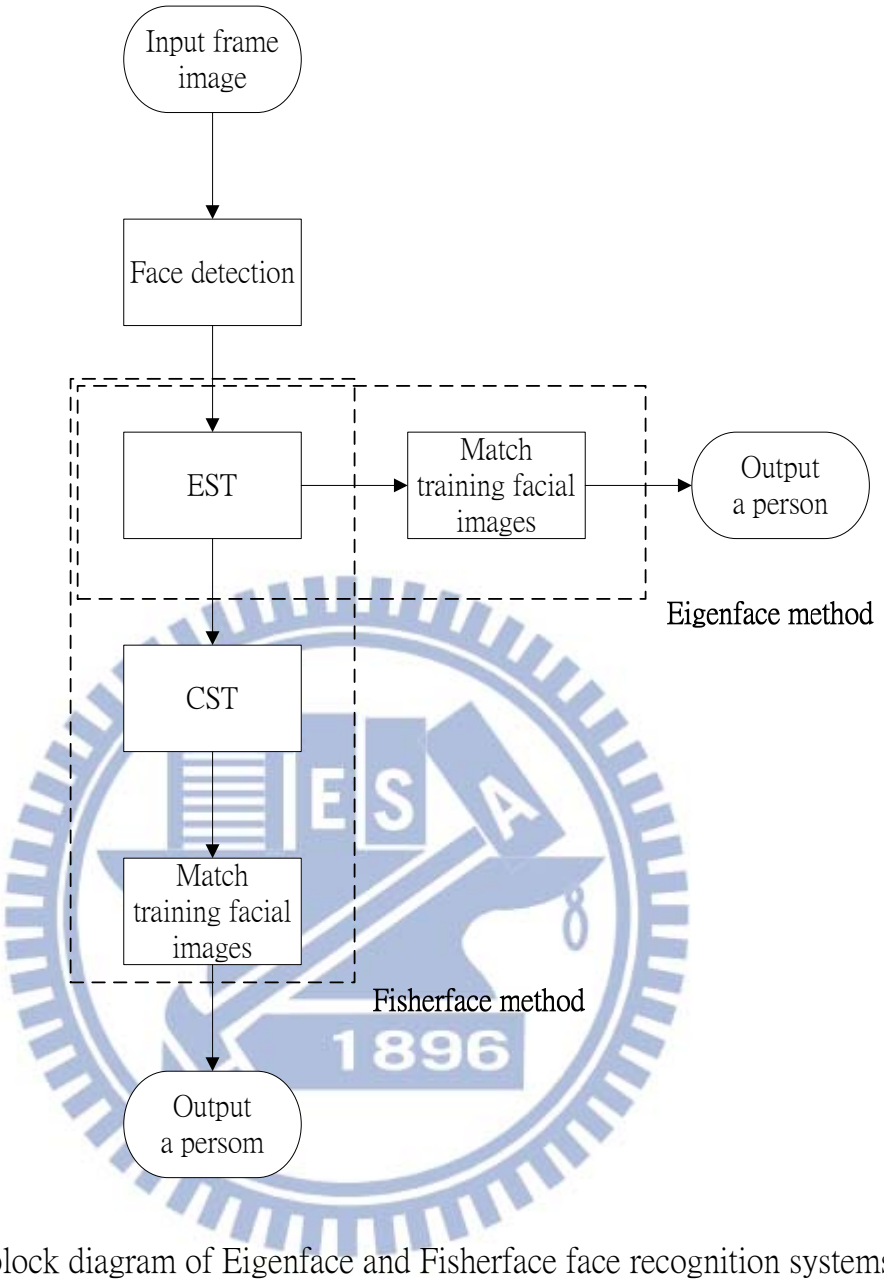


Fig. 1.2 The block diagram of Eigenface and Fisherface face recognition systems.

1.3 Video-Based Sleep/Awake Detection System

Common cameras cannot capture the image with useful information in total darkness. In order to solve above problem, the near-infrared (NIR) cameras are utilized to capture the image in sleep studies. LEDs around the camera lens emit near-infrared light toward target objects, and then the lens collects the reflected light

to form the image. Capturing images from NIR cameras in total darkness not only contain much noise than in the lightness but also exhibit non-uniformity due to irregular illumination. Therefore, we must improve the image quality by reducing the random noise and rectifying for non-uniform illumination before recognizing sleep/awake status. In the next step, our system determines sleeper's status (sleep or awake) in thirty seconds by calculating degrees of frame differences because it is proportional to human activity levels of moving in sleeping. The system flowchart is depicted in Fig. 1.3.

The sleeping postures can also reveal useful information about sleep quality and/or diseases. The automatic posture classification methods are usually based on pressure sensor array on the bed and video image frames, and the latter is realized in this paper. We will use the action recognition system, which is described in chapter 1.2, to recognize sleeper's postures.

1.4 Thesis Outline

This thesis is organized as follows. In Chapter 2, we introduce our face and action recognition system in detail. In Chapter 3, we describe the sleep/awake detection system that includes 1) image rectification for non-uniform illumination, 2) sleep/awake status detection, 3) noise removal and 4) sleeping posture recognition. In Chapter 4, the experiment results of our system are shown. At last, we conclude this thesis with a discussion in Chapter 5.

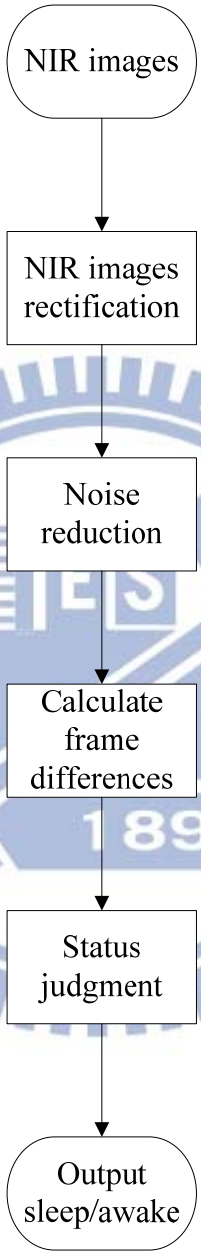


Fig. 1.3 The block diagram of sleep/awake detection system.

Chapter 2 Face and Action Recognition System

2.1 Foreground Extraction

We extract foreground subject by using background model methods. There are many well-known background models. W^4 is such a typical example with some modifications [11]. It records the maximum, minimum and maximum inter-frame difference grayscale of each pixel in background video frames. If the pixel's grayscale is in interval between maximum and minimum grayscale with toleration, the pixel is classified to a foreground one. The toleration is a median of the maximum inter-frame difference grayscales over the entire image. The toleration is usually adjusted to multiple of median according to environments. We build two background models in grayscale and HSV domain to detect reliably foreground pixels [18]. The HSV color space corresponds closely to the human perception of color.

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

$$\begin{bmatrix} m^{gray}(x,y) \\ n^{gray}(x,y) \\ d^{gray}(x,y) \end{bmatrix} = \begin{bmatrix} \max_{i}\left\{I_i^{gray}(x,y)\right\} \\ \min_{i}\left\{I_i^{gray}(x,y)\right\} \\ \max_{i}\left\{\left|I_i^{gray}(x,y) - I_{i-1}^{gray}(x,y)\right|\right\} \end{bmatrix} \tag{2.1}$$

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

Similarly we build another background model like grayscale value background model in each HSV dimension, hue, saturation and brightness [18]. The HSV background model of a pixel is obtained by

$$\begin{bmatrix} m^{H}(x,y) \\ n^{H}(x,y) \\ d^{H}(x,y) \end{bmatrix} = \begin{bmatrix} \max_{i}\left\{I_i^{H}(x,y)\right\} \\ \min_{i}\left\{I_i^{H}(x,y)\right\} \\ \max_{i}\left\{\left|I_i^{H}(x,y) - I_{i-1}^{H}(x,y)\right|\right\} \end{bmatrix} \tag{2.2}$$

$$\begin{bmatrix} m^{S}(x,y) \\ n^{S}(x,y) \\ d^{S}(x,y) \end{bmatrix} = \begin{bmatrix} \max_{i}\left\{I_i^{S}(x,y)\right\} \\ \min_{i}\left\{I_i^{S}(x,y)\right\} \\ \max_{i}\left\{\left|I_i^{S}(x,y) - I_{i-1}^{S}(x,y)\right|\right\} \end{bmatrix} \tag{2.3}$$

$$\begin{bmatrix} m^V(x,y) \\ n^V(x,y) \\ d^V(x,y) \end{bmatrix} = \begin{cases} \begin{bmatrix} \max_i \{I_i^V(x,y)\} \\ \min_i \{I_i^V(x,y)\} \\ \max_i \{|I_i^V(x,y)/I_{i-1}^V(x,y)|\} \end{bmatrix} & \text{if } I_i^V(x,y)/I_{i-1}^V(x,y) \geq 1 \\ \begin{bmatrix} \max_i \{I_i^V(x,y)\} \\ \min_i \{I_i^V(x,y)\} \\ \max_i \{|I_{i-1}^V(x,y)/I_i^V(x,y)|\} \end{bmatrix} & \text{if } I_i^V(x,y)/I_{i-1}^V(x,y) < 1 \end{cases} \quad (2.4)$$

where $I_i^H(x,y)$, $I_i^S(x,y)$ and $I_i^V(x,y)$ are respectively intensity of each HSV dimension at (x,y) of the *i*-th image frame, $i = 1, 2, ..., N$. Specifically, $d^V(x,y)$ is the inter-frame ratio instead of inter-frame different in the brightness information.

2.1.2 Extraction of Foreground Object

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

$$I^1{}_{foreground}(x,y) = \begin{cases} 0, & \text{if } I_i^t(x,y) < (m^{gray}(x,y) + k\mu) \\ & \text{and } I_i^t(x,y) > (n^{gray}(x,y) - k\mu) \\ 255, & \text{otherwise} \end{cases} \quad (2.5)$$

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 $d^{gray}(x,y)$ in the entire image, and k is a threshold. Threshold k is determined by experiments according to difference environments.

In 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

$$I^2{}_{foreground}(x,y) = \begin{cases} 0, & \text{if } I_i^V(x,y)/m^V(x,y) < k_V d^V(x,y) \\ & \text{or } I_i^V(x,y)/n^V(x,y) < k_V d^V(x,y) \\ 255, & \text{otherwise} \end{cases} \quad (2.6)$$

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.

2.1.3 Shadow suppression

The shadows of the object are easily classified as foreground pixels in normal condition. The situation causes object merging and object shape distortion in the binary foreground image. Therefore, we need to remove the shadow by using a shadow filter. The rationale behind the filter is that shadows have similar chromaticity,

but lower brightness than the background model. The shadows filter in the HSV color space is intuitively designed as follows:

$$S^1(x,y)=\begin{cases}\text{shadow}, & \text{if } I_i^V(x,y)-n^V(x,y)<0 \\ & \text{and} \left|I_i^H(x,y)-m^H(x,y)\right|<k_H d^H(x,y) \\ & \text{and} \left|I_i^S(x,y)-m^S(x,y)\right|<k_S d^S(x,y) \\ \text{foreground}, & \text{otherwise}\end{cases} \tag{2.7}$$

We analyze only points belonging to possible moving object that are detected in the former step. 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 $S^1(x,y)$ is the result of foreground extraction in HSV domain.

In the grayscale domain, we utilize the estimate of Normalized Cross-Correlation (NCC) [12] to quantify the similarity between the background image and an image of the video sequence. The NCC estimate method is described as follows. 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:

$$NCC(x,y)=\frac{ER(x,y)}{E_B(x,y)E_{T_{xy}}} \tag{2.8}$$

where

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

$$E_B(x,y)=\sqrt{\sum_{n=-1}^{1}\sum_{m=-1}^{1}B(x+m,y+n)^2} \tag{2.9}$$

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

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

$$S^2(x,y)=\begin{cases}\text{shadow}, & NCC(x,y)\geq L_{ncc} \text{ and } E_{T_{xy}}<E_B(x,y)\\ \text{foreground}, & \text{otherwise}\end{cases} \tag{2.10}$$

where $S^2(x,y)$ is the result of foreground extraction in grayscale domain, and L_{ncc} is a fixed threshold. If L_{ncc} is low, several foreground pixels corresponding to moving objects may be classified as shadow pixels. Otherwise, choosing a large value of L_{ncc}, then the actual shadow pixels may not be detected. Finally, the foreground subject is defined as:

$$I_{foreground}(x,y)=S^1(x,y)\vee S^2(x,y) \tag{2.11}$$

2.1.4 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. 2.1 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 96×128 . Fig. 2.2 is the extracted foreground region.

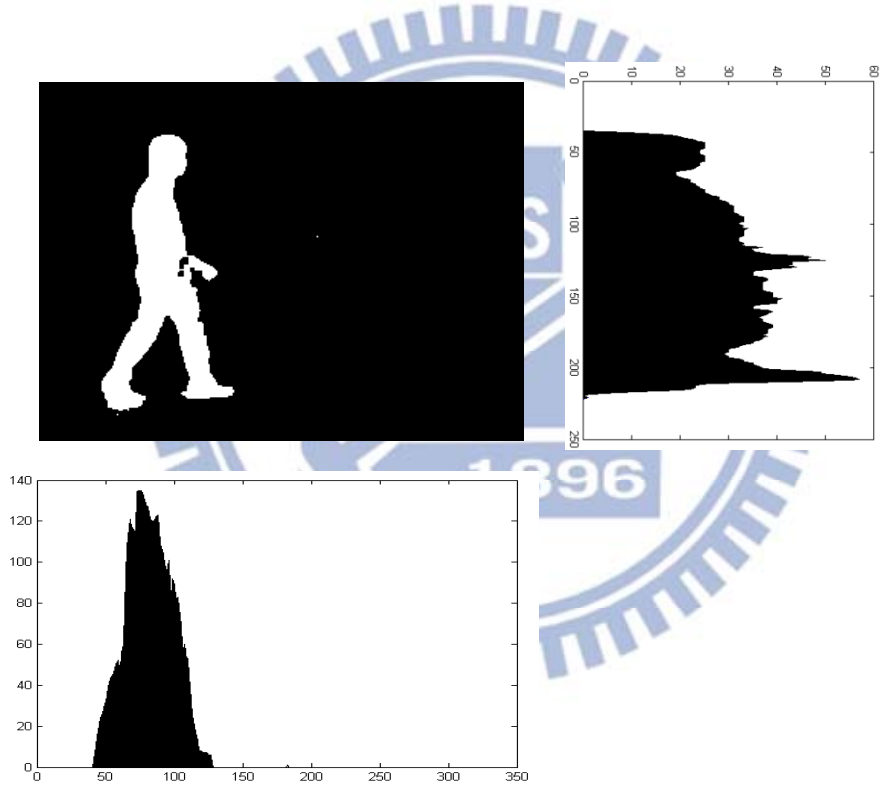


Fig. 2.1 Histogram of binary image projection in X and Y direction.

Fig. 2.2 The binary image of extracted foreground region.

2.2 Face Extraction

2.2.1 Haar Cascade Classifier

We use the classifier that is proposed by Viola et al. [8] to detect face regions. The classifier is based on the value of simple features. The feature-based algorithm works much faster than the pixel-based algorithm. The algorithm utilizes three kinds of features, *two-rectangle feature*, *three-rectangle feature* and *four-rectangle feature* to classify facial region and not facial region (see Fig. 2.3). The sum of the pixels which lie within the white rectangles is subtracted from the one within the gray rectangles, and then the value is considered as a feature.

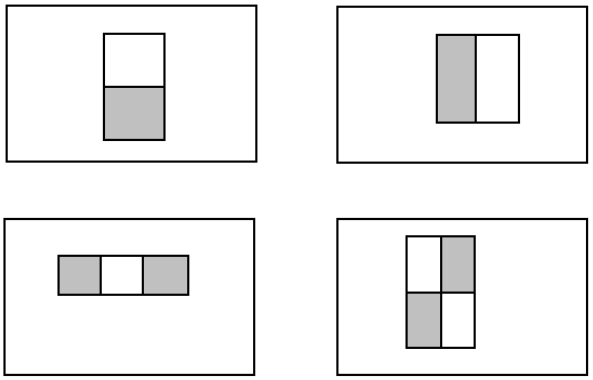


Fig. 2.3 Rectangle features shown relative to the enclosing detection widow

The cost of calculation of rectangle features can be reduced by using the integral image. The integral image intensity at location (*x,y*) is the sum of the pixels above and to the left of (*x,y*), the mathematical description as follows:

$$ii(x, y) = \sum_{x' \le x,\, y' \le y} i(x', y') \tag{2.12}$$

where $ii(x, y)$ is the integral image and $i(x', y')$ is the original image (see Fig. 2.4).

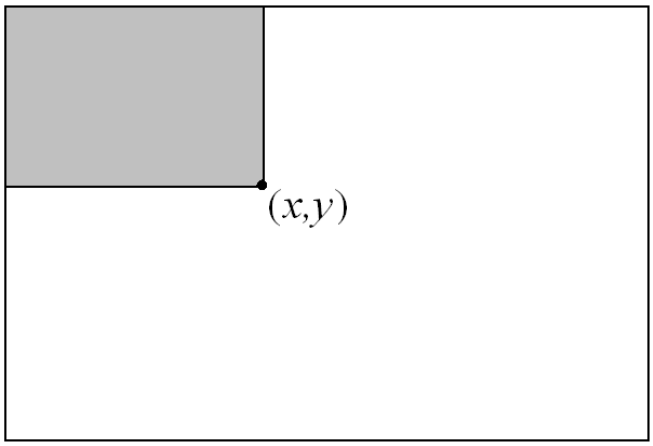


Fig. 2.4 Sum of all pixels marked is the integral image intensity at (*x,y*)

The integral image can be computed in just one pass over the original image by using the following pair of recurrences:

$$s(x, y) = s(x, y-1) + i(x, y) \tag{2.13}$$

$$ii(x, y) = ii(x, y-1) + s(x, y) \tag{2.14}$$

where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$ and $ii(-1, y) = 0$. Any rectangular sum can be computed in four array references (see Fig. 2.5). The sum of pixels in rectangle A is the integral image intensity at location 1. The sum of A+B is at location 2, A+C is at location 3 and A+B+C+D is at location 4. Therefore, the sum of

pixels in rectangle D can be computed as 4+1－(2+3).

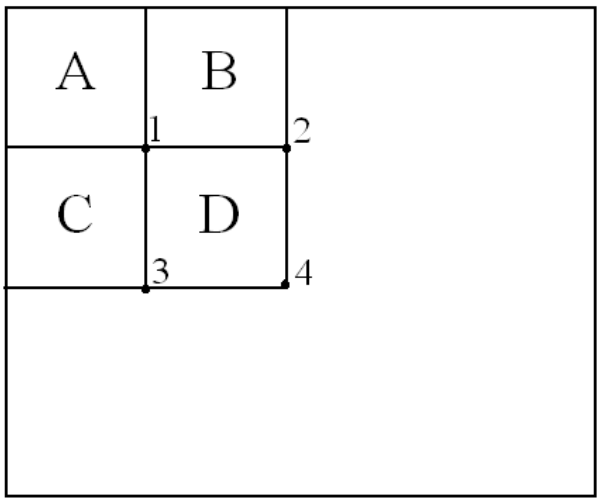


Fig. 2.5 The sum of pixels in rectangle D can be computed as 4+1－(2+3)

A variant of AdaBoost is used to select the features and train the classifier. The objective of the AdaBoost algorithm is to form a stronger classifier by combining a collection of weak classification functions. If the correct rate of a weak classifier is above 50%, it is a good weak classification function. Finally, the Haar cascade classifier is built by stringing strong classifiers for detecting face region more accurately.

2.2.2 Skin Detection

A skin locus model is proposed by Soriano et al. [13]. They sample skin pixels form image in 16 conditions (4 illuminant under 4 camera calibrations) and transform RGB space to Normalized Color Coordinates (NCC) *r-g* space to reduce illumination brightness dependence. The NCC *r-g* components are obtained by

$$r = \frac{R}{R+G+B}$$
$$g = \frac{G}{R+G+B} \tag{2.15}$$

where R、G and B are three components in RGB space. The skin locus is limited by a pair of quadratic functions defining the upper and lower bound of the cluster. The maximum and minimum g for each r is utilized to find the upper and lower quadratic functions by using least squares estimation. The upper and lower boundary curves are estimated as follows:

$$f_{\text{lower}}(r) = -0.776\, r^2 + 0.5601\, r + 0.1766 \tag{2.16}$$

$$f_{\text{upper}}(r) = -1.3767 r^2 + 1.0743 r + 0.1452 \tag{2.17}$$

$$\text{R1}: g > f_{\text{lower}}(r) \quad \text{and} \quad g < f_{\text{upper}}(r) \tag{2.18}$$

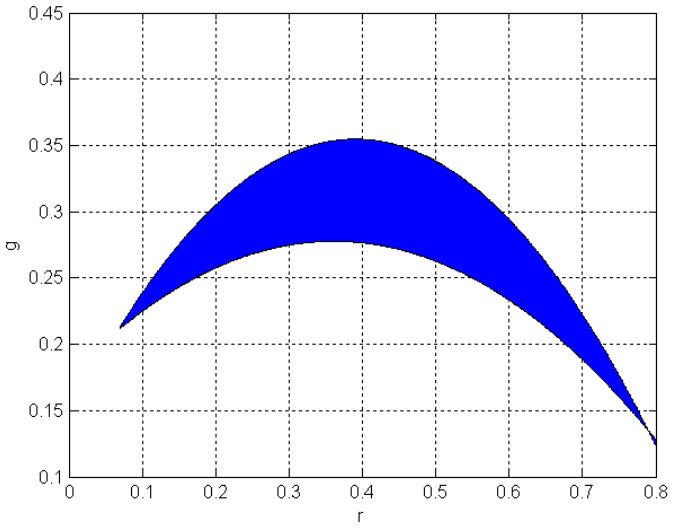


Fig. 2.6 Skin locus in the NCC *r*-*g* space [13]

However, the white pixel ($r = g = 0.33$) is within the skin locus model. To avoid that whitish pixels are labeled as skin, a circle of radius 0.02 is drawn around the white

point and pixels falling within the circle are excluded from skin model. The second condition is obtained by

$$\mathrm{R2}: W = (r - 0.33)^2 + (g - 0.33)^2 \geq 0.0004 \tag{2.19}$$

Because the skin regions are not accurately extracted with conditions mentioned previously, we add two conditions to remove the wrong skin pixels.

$$\mathrm{R3}: R > G > B \tag{2.20}$$

$$\mathrm{R4}: R - G \geq 45 \tag{2.21}$$

Finally, the skin locus model is obtained as follows:

$$\mathrm{S} = \begin{cases} 1, & \text{if all R1, R2, R3 and R4 are true,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.22}$$

where S = 1 expresses that the pixel is in skin region.

2.3 Fundamentals of Eigenspace and Canonical Space Transform

Egenspace transform is a linear projection that reduces dimensionality and maximizes the scatter of all projected data, but it is not sensitive to the class structure existent in the dada. In order to increase the recognition rate of various actions, Etemad and Chellappa [14] used linear discriminant analysis (LDA), also called Canonical Analysis (CA), which can be used to optimize the class separability of

different posture classes and improve the classification performance. If c is the number of class, the method will find a (c–1)-dimensional space in which the data are 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 classes.

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

$$\left[\mathbf{x}_{1,1},\ldots,\mathbf{x}_{1,N_1},\ldots,\mathbf{x}_{2,1},\ldots,\mathbf{x}_{c,N_c}\right] \tag{2.23}$$

where each $\mathbf{x}_{i,j}$ is an image with n pixels. Then, the mean pixel value for the training set is given by

$$\mathbf{m}_{\mathrm{x}} = \frac{1}{N_T}\sum_{i=1}^{c}\sum_{j=1}^{N_i}\mathbf{x}_{i,j} \tag{2.24}$$

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

$$\mathbf{X} = \left[\mathbf{x}_{1,1} - \mathbf{m}_{\mathrm{x}},\ldots,\mathbf{x}_{1,N_1} - \mathbf{m}_{\mathrm{x}},\ldots,\mathbf{x}_{c,N_c} - \mathbf{m}_{\mathrm{x}}\right] \tag{2.25}$$

2.3.1 EigenSpace Transformation (EST)

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 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$, $\lambda_1, \lambda_2, \ldots, \lambda_K$, and their associated eigenvectors, $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K$, satisfy the fundamental relationship

$$\lambda_i \mathbf{e}_i = \mathbf{R}\mathbf{e}_i, \qquad i = 1, 2, \ldots, K \ , \tag{2.26}$$

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. (2.26), 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

$$\tilde{\mathbf{R}} = \mathbf{X}^{\mathrm{T}}\mathbf{X} \qquad \mathbf{X}: \text{ data matrix} \tag{2.27}$$

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 $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_K$ and *K* associated eigenvectors $\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \ldots, \tilde{\mathbf{e}}_K$ which are related to those in $\mathbf{R}$ by

$$\begin{cases} \lambda_i = \tilde{\lambda}_i \\ \mathbf{e}_i = \left(\tilde{\lambda}_i\right)^{-\frac{1}{2}} \mathbf{X}\,\tilde{\mathbf{e}}_i \end{cases} \qquad i = 1,\ 2, \ldots, K \ . \tag{2.28}$$

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 $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_k|$, $k \leq K$, 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

$$\mathbf{y}_{i,j} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]^{\mathrm{T}} \mathbf{x}_{i,j} \quad i = 1,2, \ldots, c\,; j = 1,2, \ldots, N_c \tag{2.29}$$

We called this matrix $[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]^{\mathrm{T}}$ 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 vector with *k* elements which are their associated coefficients.

2.3.2 Canonical Space Transformation (CST)

Based on canonical analysis in [15], 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

$$\mathbf{m}_y = \frac{1}{N_T} \sum_i \sum_j \mathbf{y}_{i,j} \quad i = 1,\ 2, \ldots, c;\ j = 1,\ 2, \ldots, N_i \tag{2.30}$$

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

$$\mathbf{m}_i = \frac{1}{N_i} \sum_{\mathbf{y}_{i,j} \in \Phi_i} \mathbf{y}_{i,j} \tag{2.31}$$

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

$$\begin{aligned} \mathbf{S}_\mathrm{w} &= \frac{1}{N_T} \sum_{i=1}^{c} \sum_{\mathbf{y}_{i,j} \in \varphi_i} (\mathbf{y}_{i,j} - \mathbf{m}_i)(\mathbf{y}_{i,j} - \mathbf{m}_i)^\mathrm{T} \\ \mathbf{S}_\mathrm{b} &= \frac{1}{N_T} \sum_{i=1}^{c} N_i (\mathbf{m}_i - \mathbf{m}_y)(\mathbf{m}_i - \mathbf{m}_y)^\mathrm{T} \end{aligned} \tag{2.32}$$

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

$$\mathbf{J}(\mathbf{W}) = \frac{\mathbf{W}^\mathrm{T}\mathbf{S}_\mathrm{b}\mathbf{W}}{\mathbf{W}^\mathrm{T}\mathbf{S}_\mathrm{w}\mathbf{W}} \tag{2.33}$$

The ratio of variances in the new space is maximized by the selection of feature transformation $\mathbf{W}$ if

$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{W}} = 0 \tag{2.34}$$

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 [15], we can solve Eq. (2.34) as follows

$$\mathbf{S}_{\mathrm{b}}\mathbf{w}_i^* = \lambda_i \mathbf{S}_{\mathrm{w}}\mathbf{w}_i^*. \tag{2.35}$$

After solving (2.33), we will obtain c–1 nonzero eigenvalues and their corresponding eigenvectors $[\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_{c-1}]$ 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

$$\mathbf{z}_{i,j} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{c-1}]^{\mathrm{T}} \mathbf{y}_{i,j} \tag{2.36}$$

where $\mathbf{z}_{i,j}$ represents the new point and the orthogonal basis $[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{c-1}]^{\mathrm{T}}$ is called the canonical space transformation matrix. By merging equation (2.29) and (2.36), each image can be projected into a point in the new $(c-1)$-dimensional space by

$$\mathbf{z}_{i,j} = \mathbf{H}\,\mathbf{x}_{i,j}. \tag{2.37}$$

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

2.4 Activity Template Selection

Cameras usually capture image frames in high frequency (30 frames / sec), but human action transforms are much slower than the camera capturing speed. There are only few changes between two consecutive postural image frames. Therefore, we select some key frames, called as essential template images, from a sequence with a fixed interval to represent an action. We select an essential template image every 5

frames in this thesis and the schematic diagram is shown in Fig. 2.8. The number of essential template images about an action is depended on the period of the action. The long period action has more template images than the short period one. Template images of the action, “right to left walking”, are shown in Fig. 2.9. The period of the action is about 26 frames, so we choose 5 frames to represent the action.

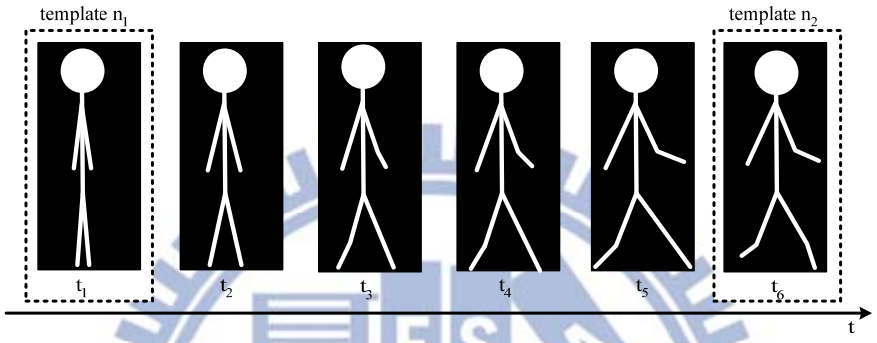


Fig. 2.7 An essential template image is selected every 5 frames.

Fig. 2.8 Template images of the action, “right to left walking”.

These essential templates are transformed to a (c－1)-dimensional vector by EST and CST methods. 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

$$\mathbf{t}_{i,j} = \mathbf{H} \cdot \mathbf{g}_{i,j}, \qquad i = 1, 2, \cdots, c; \quad j = 1, 2, \cdots, n \tag{2.38}$$

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

$$\mathbf{t}_{i,j} = \left[t_{i,j}^{1}, t_{i,j}^{2}, \cdots, t_{i,j}^{c-1}\right]^{\mathrm{T}} \tag{2.39}$$

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

$$\boldsymbol{\mu}_i = \frac{1}{n}\sum_{j=1}^{n}\mathbf{t}_{i,j} \tag{2.40}$$

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

$$\sigma_m = \sqrt{\frac{\sum_{i=1}^{c}\sum_{j=1}^{n}\left(t_{i,j}^{m} - \mu_i^{m}\right)^2}{N_t - 1}} \tag{2.41}$$

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

2.5 Construction of Fuzzy Rules from Video Stream

Transitional relationships of postures in a temporal sequence are important information for human activity classification. If we only utilize one image frame to recognize actions, it may be not sufficient to obtain high correct rate because human's actions may have similar postures in two different action sequences. For example, the actions of "jumping" and "crouching" both have the same postures called common states as shown in Fig. 2.10.

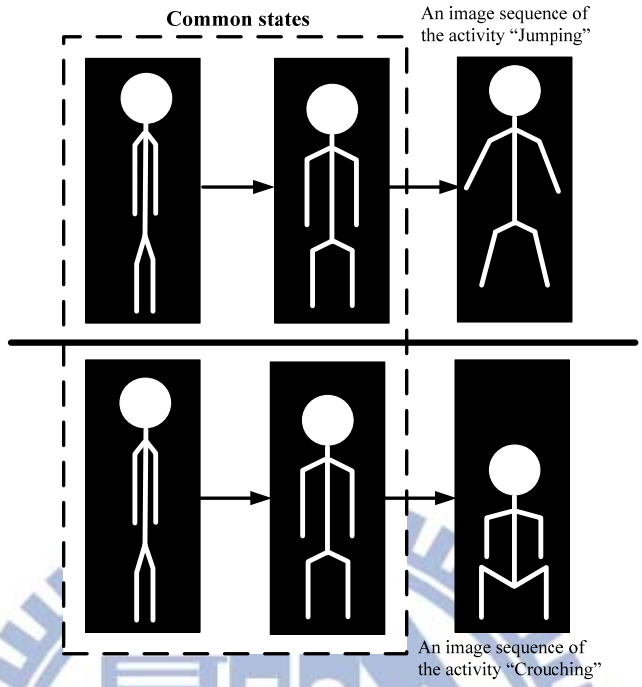


Fig. 2.9 Common states of two different activities.

We use the fuzzy rule-base approach to solve aforesaid problem. The approach not only combines temporal sequence information for recognition but also is tolerant to variations of different people. The Gaussian type membership function is represented the possibility of each cluster in this thesis because the 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 vector $\mathbf{a}_k$ is extracted by

$$\mathbf{a}_k = \mathbf{H}\,\mathbf{x}_k \tag{2.42}$$

where $\mathbf{H}$ denotes the transformation matrix of EST and CST. As the same as $\mathbf{t}_{i,j}$ in

Eq.(2.39), $\mathbf{a}_k$ can be rewritten as

$$\mathbf{a}_k = \left[a_k^1, a_k^2, \cdots, a_k^{c-1}\right]^{\mathrm{T}} \tag{2.43}$$

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 $\boldsymbol{\Sigma}$ 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

$$\begin{aligned} r_{k,i,j} &= \mathrm{M}(\boldsymbol{a_k}|C_i) \\ &= \frac{1}{(2\pi)^{\frac{c-1}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{a_k}-\boldsymbol{t})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{a_k}-\boldsymbol{t})\right] \\ &= \prod_{m=1}^{c-1} \frac{1}{\sqrt{2\pi}\sigma_m} \exp\left[-\frac{1}{2}\frac{(a_k^m - t_{i,j}^m)^2}{\sigma_m^2}\right] \end{aligned} \tag{2.44}$$

where j is the training model number. $r_{i,j}$ 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

$$P_k = \arg\max_{\mathrm{i}} r_{k,i,j} \tag{2.45}$$

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

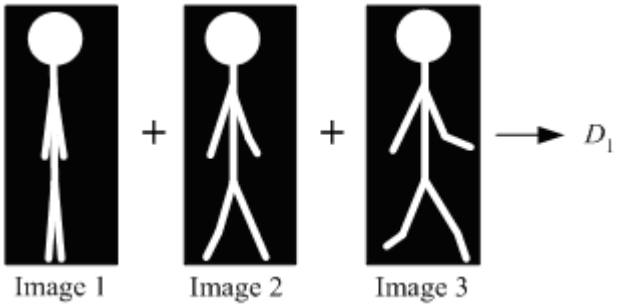
labels. Here, we combine three contiguous images to a group (I_1, I_2, I_3) and the interval of itself and next is 5 frames. The transformation of the image group can form a feature vector ($[\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}]$).There are c^3 combinations of the feature vector. Each combination represents the possible transition states of the three images. We use Eqs. (2.44) and (2.45) to class each image frame. Hence, we can represent the feature vector ($[\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}]$) by linguistic label sequence $([P_1, P_2, P_3])$. An image sequence with linguistic label sequence is associated with its output of corresponding action.

As developed by Wang and Mendel [7], fuzzy rules can be generated by learning from training data. 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

$$[P_1, P_2, P_3; D_1] \tag{2.46}$$



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 $[\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3]$, 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 action is D_1. (2.47)

Sometimes conflicting rules may be generated; they have the same image sequence but refer to different activity. Therefore, we have to choose one from the two or more conflicting rules. To this end, we choose the rule that is supported by a maximum number of training data. Furthermore, to prune redundant or inefficient fuzzy rules, if the supporting actions of a rule are less than a threshold, the rule is excluded from defining an **IF-THEN** rule.

2.6 Classification algorithm

After constructing the rule base, we can grade the input image sequence with each fuzzy rule by grade of membership function. First, each image $\boldsymbol{s}$ in the test image sequence $[\boldsymbol{s}_{k-2}, \boldsymbol{s}_{k-1}, \boldsymbol{s}_k]$ can generate a membership function lookup table R_k (k-th image frame) between image $\boldsymbol{s}$ and each template image $\boldsymbol{t}_i$ by using Eq.(2.44). In order to calculate the similarity between image sequence and each postural sequence in the training data base, we take out the membership function values from the table 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. (2.45). 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 base in the same manner. Consequent condition of the rule which has the highest value of similarity is considered as the action at the moment.

Chapter 3 Video-Based Sleep/Awake Detection System

3.1 Image Rectification for Non-uniform Illumination

NIR cameras can generate better images than common cameras because LEDs around the lens provide the near infrared wave in total darkness environment. Sometimes, LEDs are too little to supply sufficiently uniform illumination for the scene to produce good quality NIR images. In this situation, NIR images will be better presented around the image center and thus usually cause serious non-uniform illumination problem (See Fig 3.1). The region around the center of the image is bright and the brightness of the pixels, will decreases gradually. If we use the NIR image directly to recognition sleep/awake status of a person, it may produce poor results. Therefore, We rectify for non-uniform illumination before recognizing sleep/awake status of a person.

Fig. 3.1 NIR image with non-uniform illumination.

We utilize the on-axis and off-axis illumination relationship that is described by Kang and Weiss in [16] to rectify the non-uniform illumination of NIR image. The image illumination decreases across field of view in proportion with the fourth power of the cosine of the field angle. The function of illumination variation is derive as follows:

The illuminance on-axis (see Fig. 3.2) at the image point indicated by dA' is

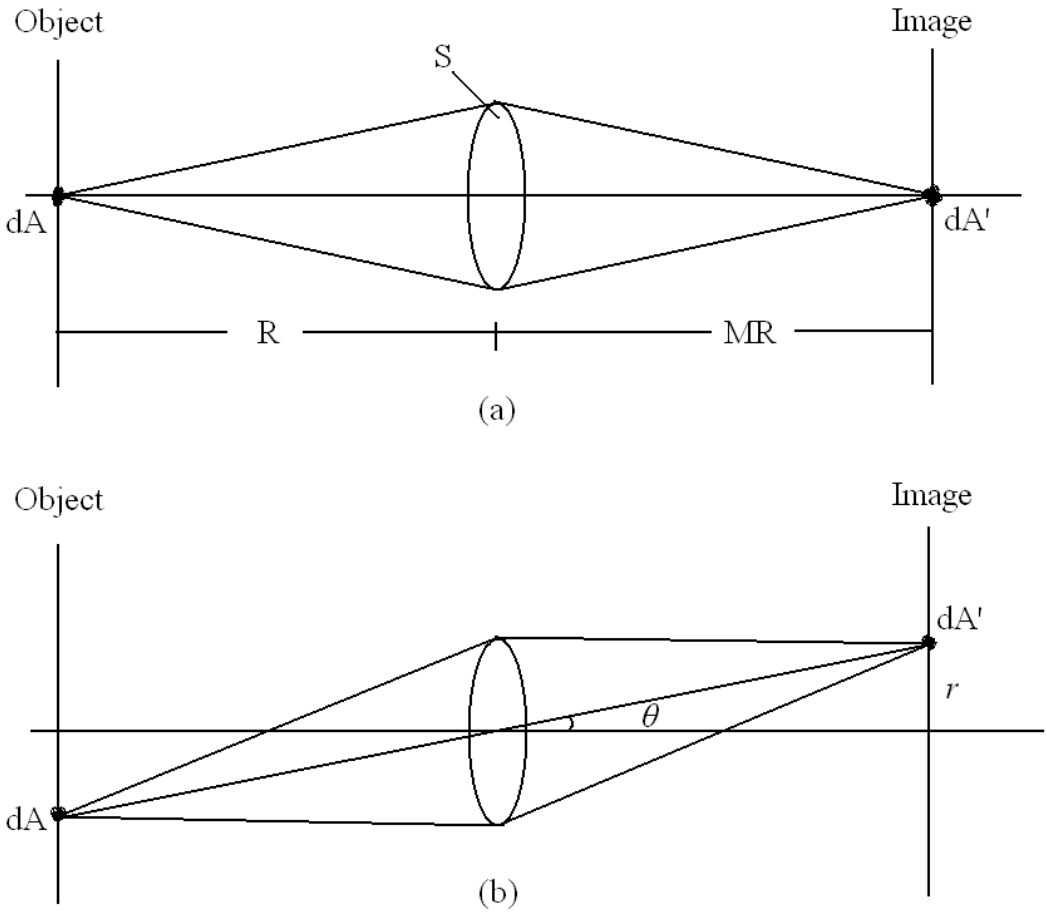


Fig. 3.2 The object projection on images (a) On-axis, (b) Off-axis at entrance angle θ.

The ellipses represent the lens for the image plane.

$$I_0{}' = \frac{L\,S}{(MR)^2} \tag{3.1}$$

L is the radiance of the source at dA, i.e., the emitted flux per unit solid angle, per unit projected area of the source. S is the area of the pupil normal to the optical axis, M is the magnification ($dA' = M^2 dA$), and R is the distance of dA to the entrance lens. The flux Φ is related to the illuminance by the equation

$$I' = \frac{d\Phi}{dA'} \tag{3.2}$$

Now, the flux for the on-axis case (see Fig. 3.2(a)) is

$$d\Phi_0 = \frac{L\, dA\, S}{R^2} \tag{3.3}$$

However, the flux for the off-axis case (see Fig. 3.2(b)) is

$$d\Phi = \frac{L(dA\cos\theta)(S\cos\theta)}{(R/\cos\theta)^2} = dA\frac{L\,S}{R^2}\cos^4\theta = dA'\frac{L\,S}{(MR)^2}\cos^4\theta \tag{3.4}$$

As a result, the illuminance at the off-axis image point will be

$$I'(\theta) = I_0'cos^4\theta \tag{3.5}$$

If f is the effective focal length and the area dA' is at image position (x, y) relative to the center of image, then

$$I'(\theta) = I_0'(\frac{f}{\sqrt{f^2+x^2+y^2}})^4 = I_0'\frac{1}{(1+(r/f)^2)^2} = \beta I_0' \tag{3.6}$$

where I' and I_0' can be considered as the intensity in original and rectified NIR images, and $r^2 = x^2 + y^2$.

3.2 Sleep/Awake Status Detection

When a person often rolls over in sleeping, we generally think that his sleep quality is poor. Therefore, motion estimation is used to detection sleep/awake status of a person in our system [19]. The current image frame is partitioned into non-overlapping and fixed-size rectangular blocks in this method. The size of the blocks called as *macroblocks* is often 4×4 to 16×16. The motion vector of each macroblock is measured by finding the closest block in the previous (or subsequent) video frame (called the *reference frame*) according to a similarity criterion (see Fig. 3.3).

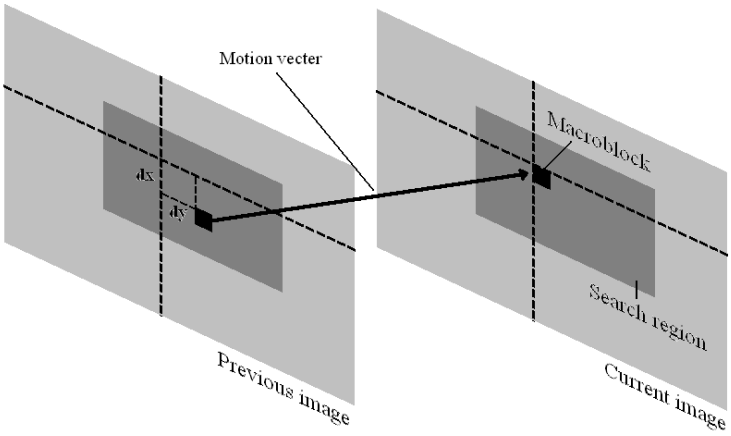


Fig 3.3 The motion vector of a macroblock

One of the most commonly used error measures is *Mean Absolute Distortion (MAD)* [19].

$$MAD(x,y) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}|f(x+i,y+j) - p(x+i+dx,y+j+dy)| \qquad (3.7)$$

where x and y are the coordinates of the upper-left pixel of the m×n macroblock in current frame *f*, *dx* and *dy* are displacements from the reference frame, and *p* is array of macroblock pixels value in the reference frame. Typically, *dx* and *dy* must fall within a limited search region that is usually ±8 to ±64 pixels for each macroblock. Motion estimation is performed by searching for the *dx* and *dy* that minimize *MAD(x,y)* over the allowed range.

However, we just want to know whether the person move in sleeping instead of the direction of motion (i.e., is the direction of motion a vector (0,0) ?). Therefore, we calculate the *MAD(x,y)* at $dx = dy = 0$ for measuring the activity degree of sleeper. If the value of *MAD(x,y)* is 0, the macroblock that corresponds to the *MAD(x,y)* is consider as a static block. Contrariwise, the macroblock is consider as a dynamic block. Activity degree in sleeping can be quantified as follows:

$$\text{ADI (Activity Degree Index)} = N_{\text{dyn}} \tag{3.8}$$

where N_{dyn} is the number of dynamic blocks in a frame. The multiplier of thirty seconds is often a period for recording data in most sleep researches. Therefore, we will average every ADI in thirty seconds, and then determine sleeper's status (sleep or awake) of a person by a threshold from training data.

$$\text{MADI (Mean of Activity Degree Index)} = \frac{\sum_{i=1}^{N} \text{ADI}_i}{N} \tag{3.9}$$

where N is the number of ADI and ADI_i is the *i*-th activity degree index in 30 seconds. Sometime a person is awake, but he do not move in the view of the NIR

camera. Therefore, we will make the second and third judgment for more correct result. In the second judgment, sleeper's status in current interval (30 seconds) is associated with former 9 intervals. If sleeping status of one in former 9 intervals is awake, sleeping status in the current interval will be awake.

3.3 Noise Removal

The MAD of a static block is not 0 if the video is recorded with random noise. That will make static blocks be regarded as dynamic blocks. To avoiding the situation, we have to find the MAD range of the noise to set a threshold. When the MAD of a macroblock lies in the noise range, we set the MAD of the macroblock to 0. To determine the range of the random noise, we calculate the MAD of each block in frames that belonging to a background video. Because no object moves in the background video, the MAD can be regarded as effect of random noise.

Fig. 3.4 is the histogram of MAD of macroblocks in background video. The background video had 24 frames and each frame has 901 macroblocks (24 × 901= 21624 macroblocks). The grayscale difference can be regard as effect of noise. We can find that the range of MAD interfered by noise is about 0 to 3.6 from Fig. 3.4.

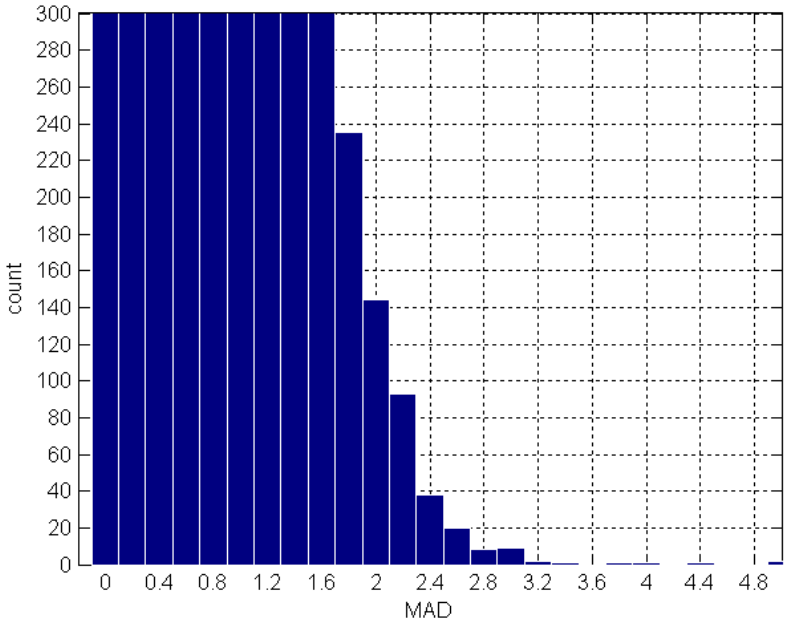


Fig. 3.4 MAD of macroblocks in background frame

3.4 Sleeping Posture Recognition

Sleeping postures can also reveal the posture likeness and sleeping quality as well of a person in sleeping. For example, maybe the people has a poor sleep quality if he often changes his sleeping posture all night. Moreover, some diseases, such as bedsore and obstructive sleep apnea, have a close relationship with one's sleep postures. We define four kinds of sleeping postures, *log, star-fish, right-foetus, left-foetus,* (see Fig. 3.5) [17] to implement sleeping posture recognition. Sleeping postures will are classified by the approach in line with actions recognition system that is described in Chapter 2.

(a)

(b)

(c)

(d)

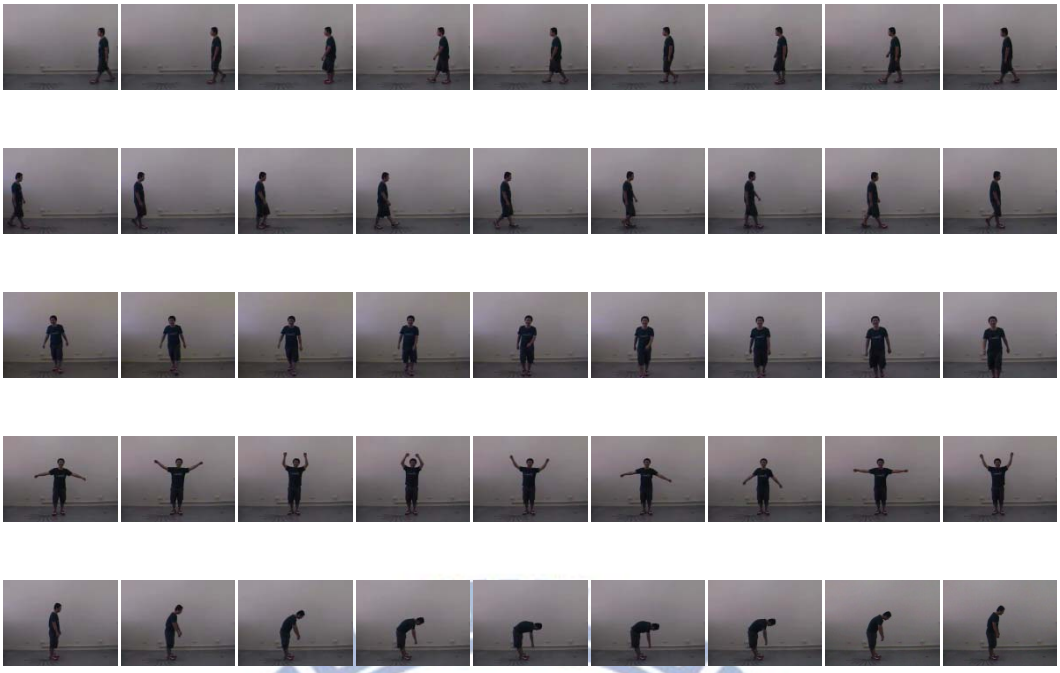
Fig. 3.5 Sleeping postures (a) log, (b) star-fish, (c) right-foetus, (d) left-foetus.

Chapter 4 Experimental Results

In our experiment, we test our system on video captured by NIR camera in bright and dark environments. The experimental environment is in our laboratory at the 5th Engineering Building in NCTU campus. The NIR camera with a lens of 6.0mm focus is set up at the location that is far from the object about 4 meter. This camera has a frame rate of 30 frame per second and image resolution is 320×240 pixel. The background of the experiment environment is simple and illumination of the environment is 524 Lux in the day (fluorescent lamps are on.) and 0.07 Lux in the night respectively. The scenes in bright and dark environments are shown in Fig. 4.1. We choose five actions: "walking from right to left," "walking from left to right," "walking straightly," "waving" and "bending" to recognize in our system. Fig. 4.2 shows the examples video sequence form our LAB databases.

(a) (b)

Fig. 4.1 (a) The experiment environment in the day, (b) The experiment environment in the night



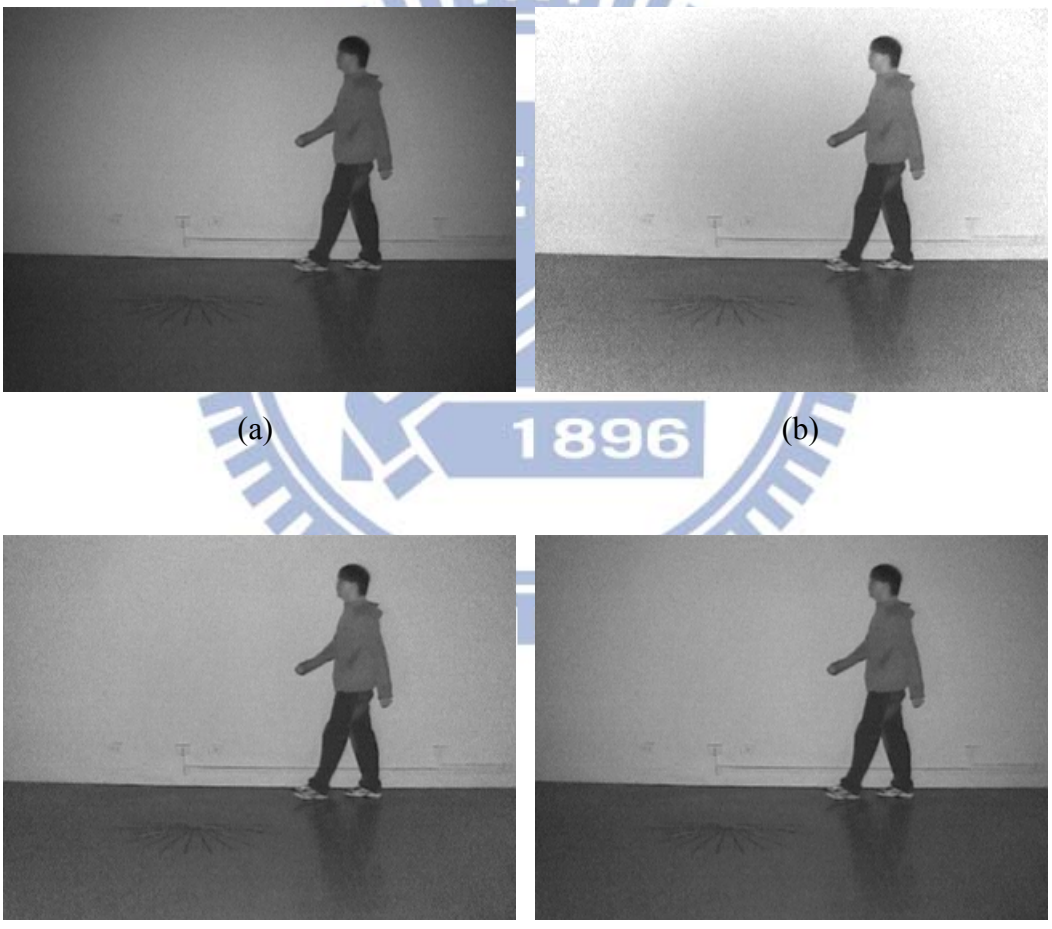
(a)

(b)

Fig. 4.2 Example video sequences used in our experiments. (a) and (b) are typical video sequences for actions of LAB in bright and dark environments. From top to bottom: “walking from right to left, walking from left to right, walking straightly, waving and bending respectively.

4.1 Image Rectification Result

All frames of video captured in total dark environment must be rectified by using Eq. (3.6) in Section 3.1. Fig. 4.2(a) is a frame from the action recognition training data and it is transformed to gray level. Results of rectifying NIR images with different f are shown in Fig. 4.2. We can find that f = 300 is a better parameter for rectifying the NIR image to be a uniform illumination image.



(a) (b)

(b) (d)

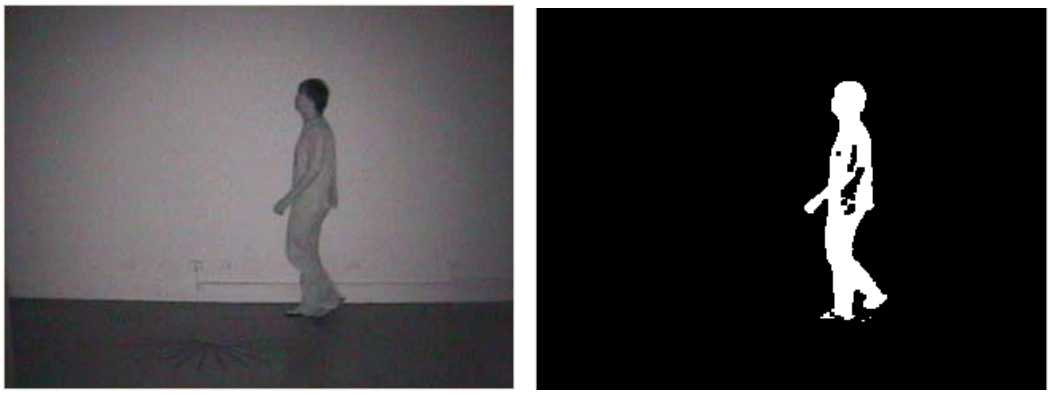
Fig. 4.3 Results of rectifying NIR image with different f (a) brfore rectifying, (b) f = 200, (c) f = 300, (d) f = 400.

4.2 Background Model and Foreground Object Extraction

For constructing the background model, we first record a video of pure background (like Fig. 4.1) about 2 second in bright and dark environments. After building the grayscale value and the HSV color space background models, we will extract the foreground pixel by using Eq. (2.5) and Eq. (2.6) in Section 2.1.2. Then we continue to emend the former foreground image by shadow filter.

In order to get the optimal result of object extraction, we have to adjust some parameters in our system. We set $k = 2.3$ and $k = 2.0$ for the grayscale value background models and $k_v = 1.4$ and $k_v = 1.1$ for the HSV color background models in bright and dark environments respectively. The same parameter is used in bright and dark environments for shadow filter. We set $L_{ncc} = 0.95$ in the grayscale value space and $k_H = 1.3$ and $k_s = 1.3$ in the HSV color space to detect shadow pixels. Fig. 4.3 shows results of foreground extraction in bright and dark environments.

(a) (b)



(c) (d)

Fig. 4.4 Results of foreground extraction (a) an image frame in the bright environment, (b) foreground extraction from (a), (c) an image frame in the dark environment (d) foreground extraction from (c).

Finally, we simply introduce a threshold on the histograms in X and Y direction to minimize the size of foreground images, and then resize the images to 96×128 for normalization. That is described in Section 2.1.2. The threshold in X and Y direction is about 10 pixels in our experience.

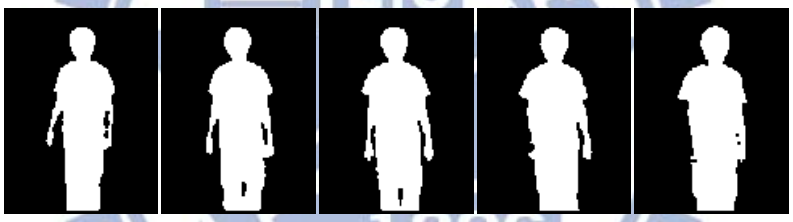
4.3 The Day and Night Face and Action Recognition

4.3.1 Fuzzy Rule Construction

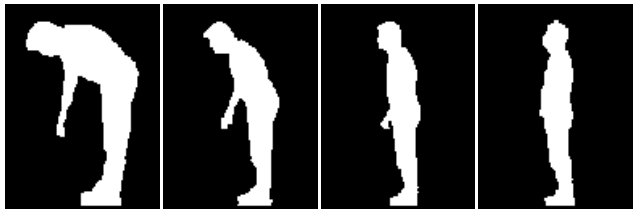
We construct the template model matrix and the fuzzy rule database with the training data. Firstly, we choose key posture images as essential templates from each action, and the number of each action is according to its period. Key posture images of each action for one person (one model) are shown in Fig. 4.4. We will regard each posture as one class.

(a)

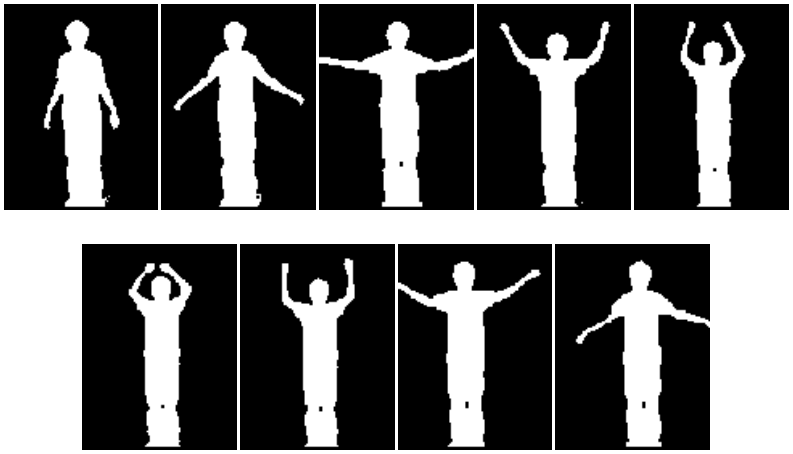
(b)



(c)



(d)



(e)

Fig 4.5 Key postures of the actions (a) walking from right to left, (b) walking from left to right, (c) walking straightly, (d) bending, (e) waving.

Fuzzy rules are constructed in off-line situation. 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, *etc*. 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. (2.44). 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 2.5.

4.3.2 The Recognition Rate of Actions

In order to calculate the recognition rate of actions, we use off-line videos 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 and the third frame, *etc.* Table I and Table II show the recognition rate in bright and dark environments respectively, four folds cross validation, of each action of each model. If we test these videos in Person 1, we will constructed the templates and fuzzy rules by used the order three persons. That is, the testing video was not used for constructing templates and fuzzy rules.

In the tables, W_{RL} is the action "walking from right to left," W_{LR} is the action "walking from left to right," W_S is the action "walking straight," W_{AVE} is the action "waving," B_{END} is the action "bending." Here, the recognition rate is the number of correct recognition divide by the total number of recognition for each video.

Table I

The recognition rate of each activity in the light environment

	Person 1	Person 2	Person 3	Person 4
W_{RL}	93.1% (108/116)	99.1% (105/106)	92.9% (118/127)	90.3% (121/134)
W_{LR}	95.0% (95/100)	100% (110/110)	98.4% (112/124)	96% (96/100)
W_S	100% (81/81)	88.8% (87/98)	92.3% (36/39)	94.0% (47/50)
W_{AVE}	100% (83/83)	95.6% (43/45)	100% (107/107)	98.1% (53/54)
B_{END}	100% (48/48)	89.2% (66/74)	100% (200/200)	100% (74/74)
Average	95.7% (1790/1870)			

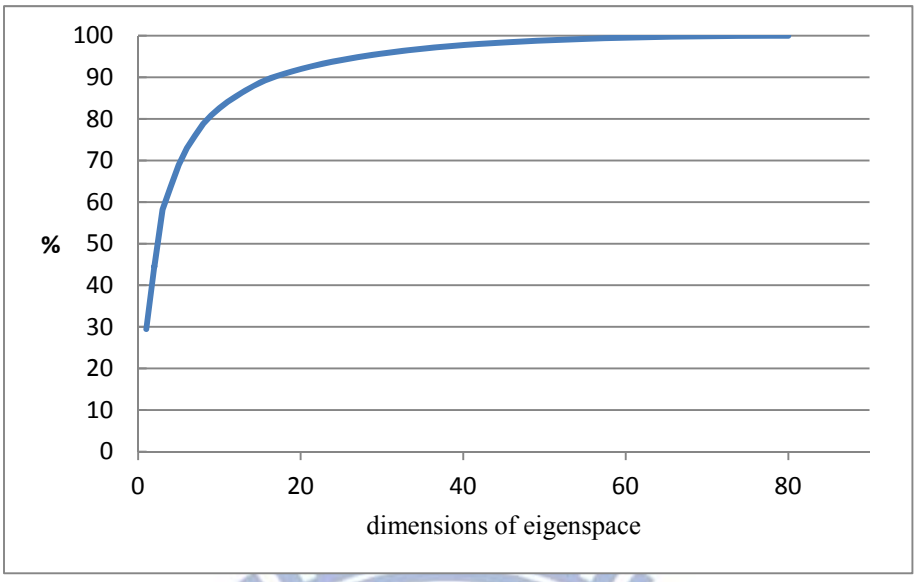
Table II

The recognition rate of each activity in the dark environment

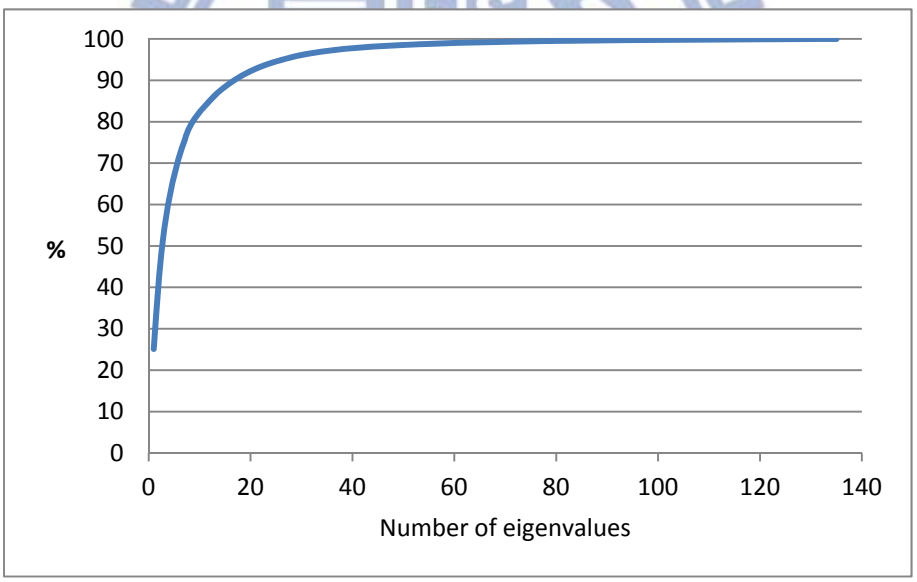
	Person 1	Person 2	Person 3	Person 4
W_{RL}	89.5% (68/76)	90.9% (60/66)	95.5% (64/67)	93.3% (56/60)
W_{LR}	100% (80/80)	95.6% (43/45)	100% (66/66)	93.9% (46/49)
W_{S}	100% (83/83)	100% (49/49)	100% (44/44)	100% (32/32)
W_{AVE}	82.7% (62/75)	100% (94/94)	93.8% (45/48)	100% (55/55)
B_{END}	100% (86/86)	100% (70/70)	97.6% (80/82)	94% (63/67)
Average	96.5% (1249/1294)			

4.3.3 The Recognition Rate of Faces

In our face recognition experiment, we take face images of 8 persons and 9 persons in bright and dark environments respectively to obtain the accurate rate of face recognition. The size of face images is 50×60 for training and testing. Firstly, the face images are project to eigenspace by using EST transformation. Then, we utilize CST transformation to project former images to FisherFace space and implement face recognition. The test image is compared to every training data by L_2 norm to find the most similar one. The numbers of training and testing images are 15 and 45 for each person in the darkness. In the lightness, the numbers of training and testing images are 10 and 100 for each person. Fig. 4.5 shows the curve of accumulative eigenvalues. Accumulative eigenvalues contain 98% information of images when the number of eigenvalues is about 50. Fig. 4.6 shows the correct rate of face recognition by using FisherFace method for different dimension in eigenspace. The best correct rate of face recognition in bright and dark environments are recorded in Table III and Table IV respectively.

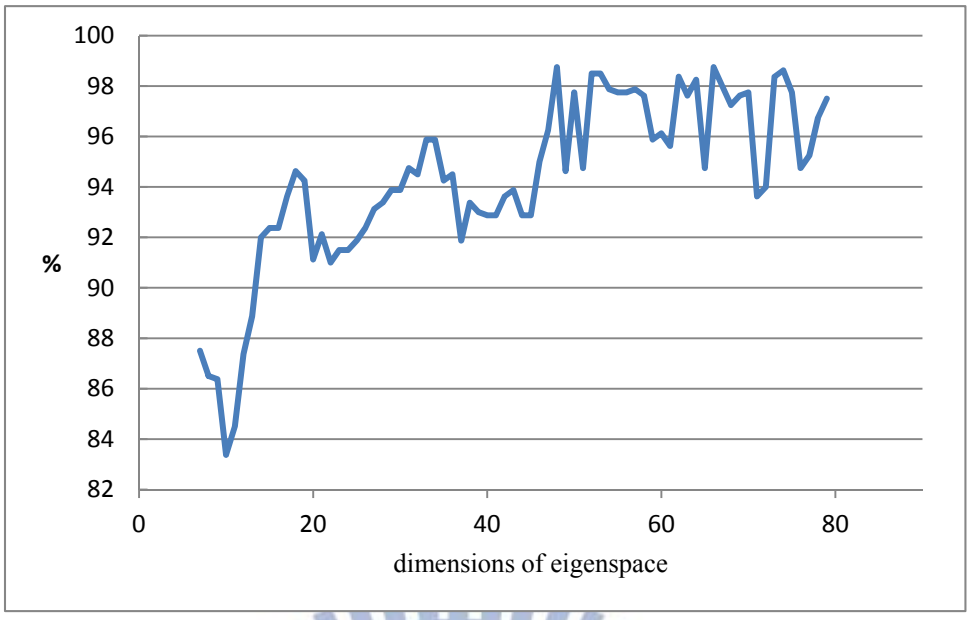


(a)



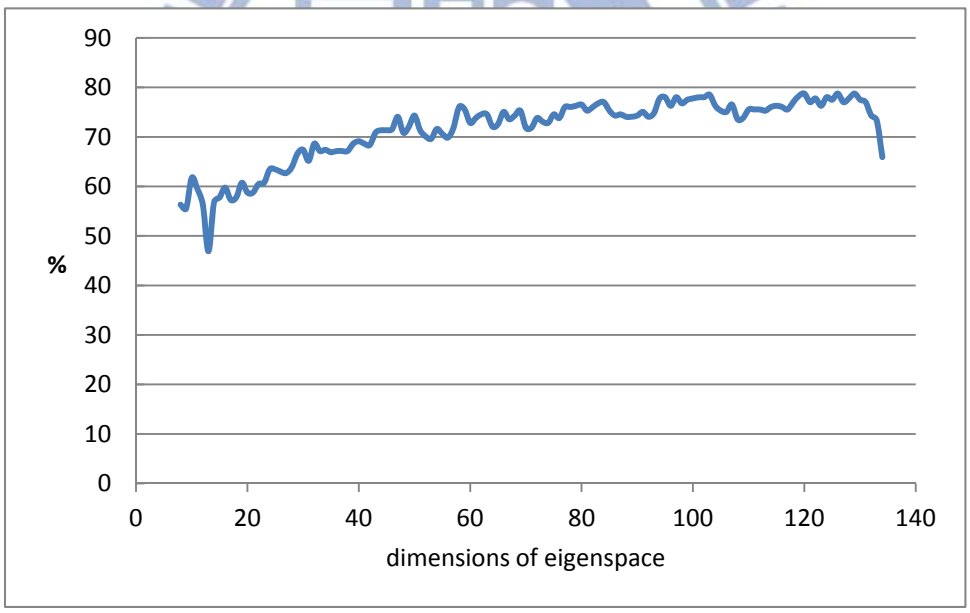
(b)

Fig. 4.6 The Curves of accumulative eigenvalues (a) in bright environment (b) in dark environment.



;

(a)



(b)

Fig. 4.7 The curves of face recognition rate versus dimensions of eigenspace used in the (a) bright environment; (b)dark environment.

Table III

The correct rate of face recognition in the light environment

Judge \ Test	**Person 1**	**Person 2**	**Person 3**	**Person 4**	**Person 5**	**Person 6**	**Person 7**	**Person 8**
Person 1	100	0	0	1	0	0	2	0
Person 2	0	100	0	0	0	0	0	0
Person 3	0	0	99	0	0	0	0	0
Person 4	0	0	0	99	0	0	0	0
Person 5	0	0	0	0	100	0	2	0
Person 6	0	0	1	0	0	100	0	4
Person 7	0	0	0	0	0	0	96	0
Person 8	0	0	0	0	0	0	0	96
individual Accuracy rate	100%	100%	99%	99%	100%	100%	96%	96%

The total accuracy rate is 98.7%.

Table IV

The correct rate of face recognition in the dark environment

Judge \ Test	**Person 1**	**Person 2**	**Person 3**	**Person 4**	**Person 5**	**Person 6**	**Person 7**	**Person 8**	**Person 9**
Person 1	42	0	4	0	0	0	0	5	0
Person 2	2	44	0	0	0	0	1	1	0
Person 3	0	1	23	2	9	2	15	0	7
Person 4	0	0	0	27	0	0	0	0	0
Person 5	0	0	0	0	36	0	0	0	2
Person 6	0	0	0	0	0	43	0	0	0
Person 7	0	0	18	0	0	0	29	0	0
Person 8	0	0	0	0	0	0	0	39	0
Person 9	1	0	0	16	0	0	0	0	36
Individual Accuracy rate	93.3%	97.8%	51.1%	60.0%	80.0%	95.6%	64.4%	86.7%	80.0%

The total accuracy rate is 78.8%

4.4 Sleep/Awake Detection

In the sleep/awake detection system, the detected region is divided into hundreds of macroblocks if the size we choose of macroblocks is 5×5 pixels (see Fig. 4.7). Dimensions of the region in the rectangle with red edges are 265×85 pixels (i.e., 53× 17 = 901 macroblocks). Common sample rate of NIR camera is 30 frames per second, and it will waste the spaces for data if we capture image data by using the sampling rate. Because the human activity is not active in sleeping, we reduce the sampling rate to 2 frames per second for our records.

Fig. 4.8 The region of sleep/awake detection.

Table IV show the result of sleep/awake detection. An interval represents a sleep or awake video of 30 seconds. The threshold of MADI is 6 that is set by training data. When a person is awake, our system will output 1, otherwise 0.

Table V

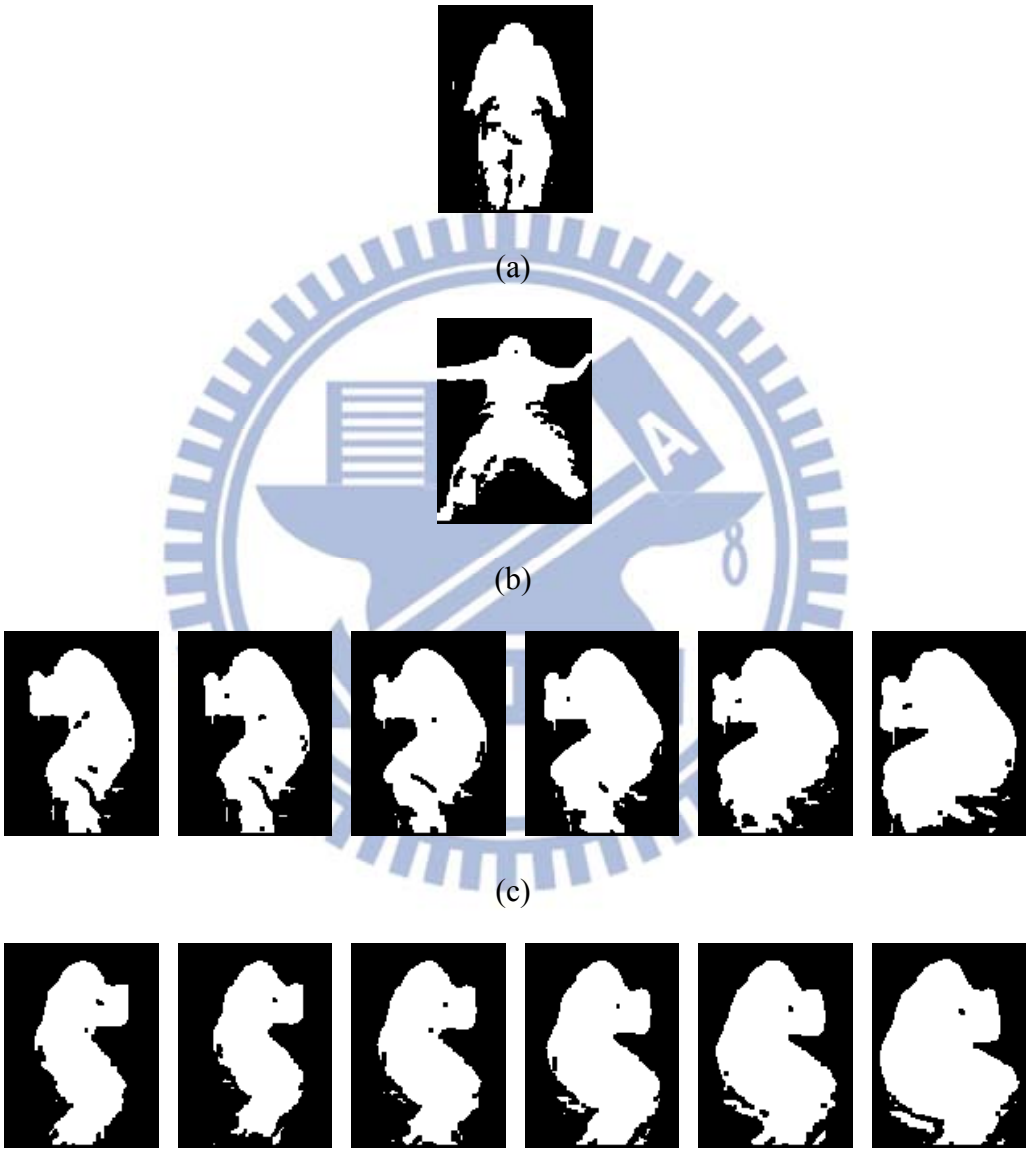
The result of sleep/awake detection

	Awake			Sleep		
Interval	**MADI**	**Judge 1**	**Judge 2**	**MADI**	**Judge 1**	**Judge 2**
1	20.57	1	1	3.82	0	0
2	6.30	1	1	2.93	0	0
3	15.97	1	1	3.63	0	0
4	2.82	0	1	4.68	0	0
5	4.12	0	1	3.30	0	0
6	43.78	1	1	3.53	0	0
7	4.73	0	1	3.02	0	0
8	75.52	1	1	4.77	0	0
9	93.83	1	1	4.45	0	0
10	3.38	0	1	4.23	0	0
11	2.62	0	1	3.62	0	0
12	3.28	0	1	4.00	0	0
13	13.37	1	1	3.98	0	0
14	2.95	0	1	3.02	0	0
15	2.42	0	1	2.58	0	0
16	3.75	0	1	4.37	0	0
17	2.88	0	1	4.35	0	0
18	2.80	0	1	3.40	0	0
19	3.10	0	1	2.95	0	0
20	2.75	0	1	3.47	0	0

4.5 Sleeping Posture Recognition

Actions recognition system is utilized to classify sleeping postures in this thesis. We set $k = 2.0$ for the grayscale value background models and $k_v = 1.1$ for the HSV color background models. In the HSV color space, we set $L_{ncc} = 0.95$ in the grayscale value space and $k_H = 1.3$ and $k_s = 1.3$ to detect shadow pixels. Fig. 4.3

shows results of foreground extraction in bright and dark environments. Key posture images of four sleeping posture are show in Fig. 4.8. We select different postures as templates according to degree of shrinking feet in sleeping postures, right and left foetus. Table VI show the correct rate of sleeping posture.



(a)

(b)

(c)

(d)

Fig. 4.9 Key postures of sleeping postures (a) log, (b) star-fish, (c) right-foetus, (c) left-foetus.

Table VI

The recognition rate of each sleeping posture

	Log	Star-fish	Right-foetus	Left-foetus
Person 1	100% (79/79)	100% (95/95)	96.1% (73/76)	98.2% (54/55)
Average	98.7% (301/305)			

5. Conclusion

In this thesis, we implement the automatic home health care system that combine the face, action and sleep/awake recognition of a person in day and night. The test images are extracted by background subtraction in action recognition system and by Haar cascade classifier in face recognition system. Then, the test images are transformed to a new space by eigenspace and canonical space projection for better efficiency and separability. Because actions are dynamic unlike face, we gather three images with fixed interval to construct fuzzy rules for containing temporal information. In sleep/awake detection, the NIR images will are rectified by using the function of illumination variation firstly. Then, the motion estimation is utilized to quantify the activity degree of sleepers.

NIR images look similar to gray-level image. The NIR image has less information of hue and saturation components than color images. Therefore, the correct rate of face recognition in dark environment is much lower than in the bright environment. However, the correct rates of action recognition in bright and dark environment are not that different because information provided by NIR images is sufficient to extract almost complete foreground images. In the sleep/awake detection system, we also obtain very good by using motion estimation. In the future, it is necessary to find a new a new face recognition algorithm to improving the correct rate in darkness environment.

References

[1] W. H. Liao and C. M. Yang , “Video-based activity and movement pattern analysis in overnight sleep studies,” ICPR, pp.1-4, 2008.

[2] Y.T. Peng, C.Y. Lin, M.T. Sun, and C.A. Landis, "Multimodality Sensor System for Long-Term Sleep Quality Monitoring," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 1, no. 3, pp.217–227, 2007.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] P. Viola and M. Jones, “Robust Real-Time Face Detection,” *IJCV*, vol. 57, no. 2, pp. 137–154, Mar. 2004.

[9] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection,” *IEEE Trans. Pattern*

Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711-720, 1997.

[10] "OpenCV 1.0, Open Source Computer Vision Library," http://www.intel.com/technology/computing/opencv/, 2006.

[11] 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, August 2000.

[12] J. C. S. Jacques Jr., C. R. Jung, S. R. Musse, "Background subtraction and shadow detection in grayscale video sequences." In *Proc. SIGGRAPH*, pp. 189–196, 2005.

[13] M. Soriano, B. Martinkauppi, S. Huovinen and M. Laaksonen, "Using the skin locus to cope with changing illumination conditions in color-based face tracking," in *Proc. IEEE NORSIG*, Kolmarden, Sweden, pp. 383–386, 2000.

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

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

[16] S. B. Kang and R. Weiss, "Can we calibrate a camera using an image of a flat textureless Lambertian surface?" in *Proc. ECCV*, vol. 2, pp. 640–653, 2000.

[17] W. Huang, A. Phyo Wai, S. Fook Foo, J. Biswas, C. Hsia and K. Liou, "Multimodalsleeping posture classification," in *Proc. ICPR*, pp. 4336-4339, Aug., 2010.

[18] 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.

[19] R. Gonzales and R. Woods, *Digital Image Processing*, 3rd ed. Pearson Education International, pp. 589–591, 2008.