Chapter 1 Introduction

Human activity recognition from video streams has a wide range of application such as human-machine interface, security surveillance, home care system, etc.

Recently, there were many transformation methods which are used to decrease the size of data. Huang *et al.* [1], [2] applied EigenSpace Transformation (EST) and Canonical Space transformation (CST) to human gait recognition and face recognition. They recognized human by temporal templates of their gaits in canonical space. Although their method was not designed for human activity recognition, we can modify it for thesis. An action is composed of the same (or similar) pose templates done by different people within a tolerable difference.

However, Jobson and Woodell [3], and Rahman [4] developed photo enhancement method called retinex. Although we need not enhance our image, retinex guides us to build the background and to extract the foreground subjects by frame ratio. Park and Aggarwal [5] subtracted foreground pixels from background by computing Mahalanobis distance in each pixel in HSV color model. Their method requires more computational costs. Leung and Yang [6] built a human body outline labeling system. Haritaoglu *et al.* [7] built a human body part labeling system. They all try to find out the real poses a human did by human body outline or by silhouettes.

There were some similar works. Yamato *et al.* [8] use Hidden Markov Model (HMM) to recognize human action in time sequential images. They turn images into symbol sequence and used HMM to tell what a tennis player is dong (smash, serve, etc). Bobick and Davis [9] recognized the human movement (activity) by comparing temporal templates with motion-energy images and motion-history images. Bodor *et al.* [10] tracked people and recognized their activities by analyzing the pedestrian velocity and path. Masoud and Papanikolopoulos [11] recognized the human activities via PCA in eigenspace.

There have been some significant projects on detecting, tracking people and recognizing their activities. W^4 [7], [12] is one of them. W^4 can detect people (single person or people in group) by an adaptive background model and identify the activities by find the body parts on the silhouette boundary. It constructed dynamic models of people's movements to answer questions about What they are doing, Where and When they act, and tracks people with relative identity, i.e., Who. That's why so called W^4. In [13]–[16], the pose of body parts was represented by a set of normalized body part angles. Models of human activity were represented by a sequential arrangement of sets of multidimensional vectors that correspond to sampled angular poses of body parts over the entire time interval. These vectors were then divided into a set of subvectors and then form a set of hash tables, each of which corresponds to an individual body part. Namely, they turned each body parts into a vector by angles and form hash tables, and then recognized the activity of image sequence by looking up the hash table. Multidimensional indexing used complex EXpansion Matching (EXM) to find the angles of main body parts. It is quite different from our method. We map the binary images of subject into canonical space by the transformation matrix which has been generated at the training stage.

The objective of this thesis is to provide a human-like system to auto-surveillance and to track people and identify their activities. This system can tell where the foreground subject is in an image, and what the subject is doing. This system tracks and identifies people via a monochromatic video camera. The reason why we use a monochromatic video camera instead of a color one is if we can do well in a monochromatic format, we can still succeed at night when a special illumination source is used, for example, infrared. This system is designed for indoor surveillance tasks such as home watching, security surveillance, home care, etc.

Fig. 1 is a video example showing our proposed system. One can classify the activity just by looking at Fig. 1, and determine that the activity of the subject in Fig. 1 is walking. Our proposed system can work as we humans do. It needs not a full video sequence and only three or four video image samples of an activity period are sufficient to classify the activity. Therefore, based on this human vision experience, an activity is represented by several essential templates in our system. In video processing, the size of image sequence usually extremely large and the activity recognition algorithm is usually complex. If a few essential templates are enough to represent an activity, we can recognize the activity of the subject by down sampling the image sequence instead of all consecutive image frames in order to reduce the recognition complexity, decrease the computational load, and improve the recognition performance.

There are four processing steps in our system: 1. foreground subject extraction, 2. data transformation (mapping), 3. image frame classification, and 4. activity recognition. In the first processing step, the system locates the foreground pixels in an image by a statistic background model. In the second processing step, the system project the image sequence into points in canonical space by EST based on Principal Component Analysis (PCA) and then CST based on Canonical Analysis (CA) [5], [7]. In the third processing step, the system classifies the transformed image sequence in canonical space by referring a database we generated. In the fourth processing step, the system evaluates the current frame matching grade as well as those of the passed four frames to determine the activity type of the subject. Fig. 2 shows the building blocks of our proposed human activity recognition system.

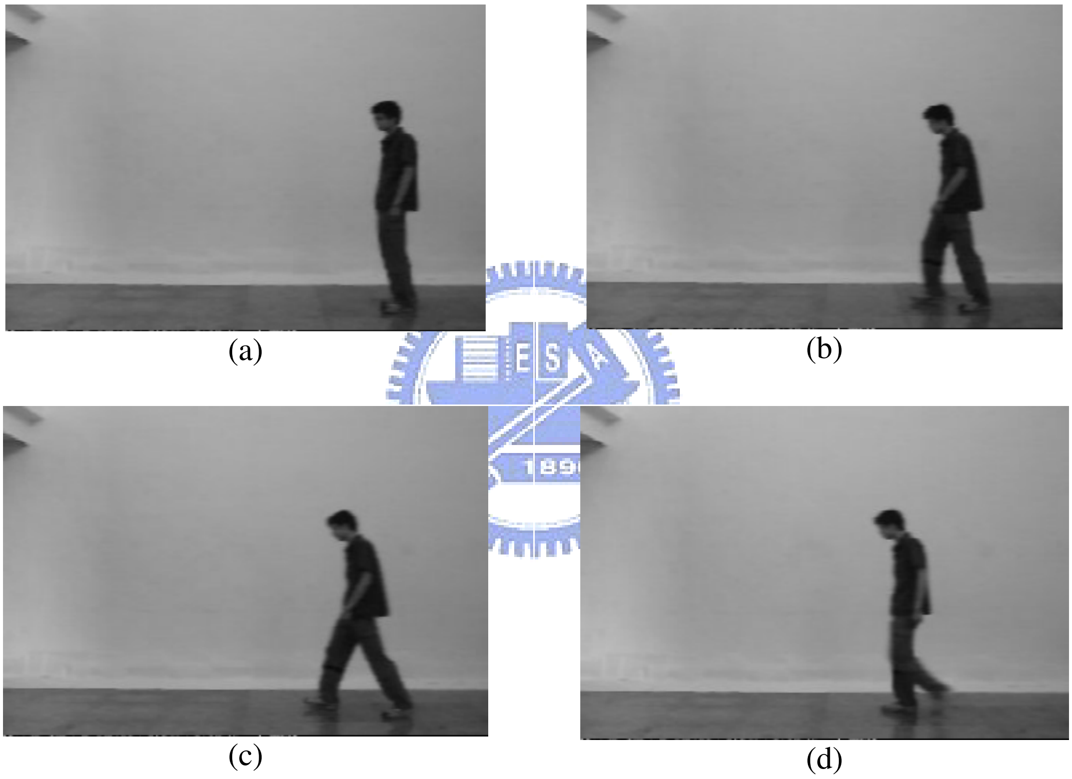

(a) (b) (c) (d)

Fig. 1. Four sampled video of a man walking. (a) Frame 26, (b) frame 36, (c) frame 42, and (d) frame 50.

Before recognition, the system needs a background video to generate a background model for foreground extraction. This background model is a simple statistical model. Details will be discussed in the following chapters.

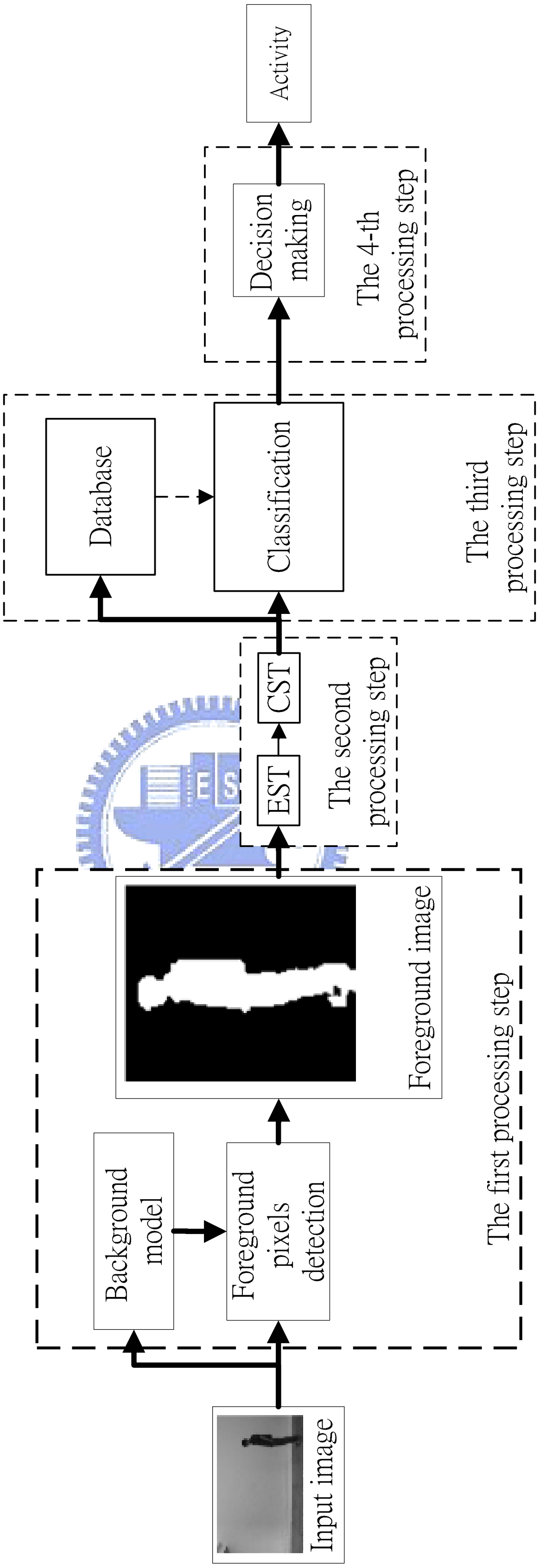


Fig. 2. The building blocks of human activity recognition system.

Eigenspace transformation based on Principle Component Analysis and canonical space transformation based on Canonical Analysis are widely used in classification problems and pattern recognition problems. EST based on PCA can convert the high-dimensional images to low-dimensional eigenspace while maintaining the representative information. CST based on CA can again reduce the dimension of images and optimize the class separability by maximizing the between class distance and minimizing the within class distance to improve the classification performance.

After transforming image sequence into points in canonical space, equivalently, the activity recognition problem would be cast as a sequence matching and then tracing problem or state transfer problem. Fig. 3 is a states transfer diagram we have developed of the activity “walking from right to left.” The transformed image sequence in canonical space will show up from state 1 to 2, to 3, to 4 … to 9 in one period, and again to state 1 for a new period. Thus we classify the activity type of the foreground subject, and hence we can tell what he is doing according to the state transfer sequence.

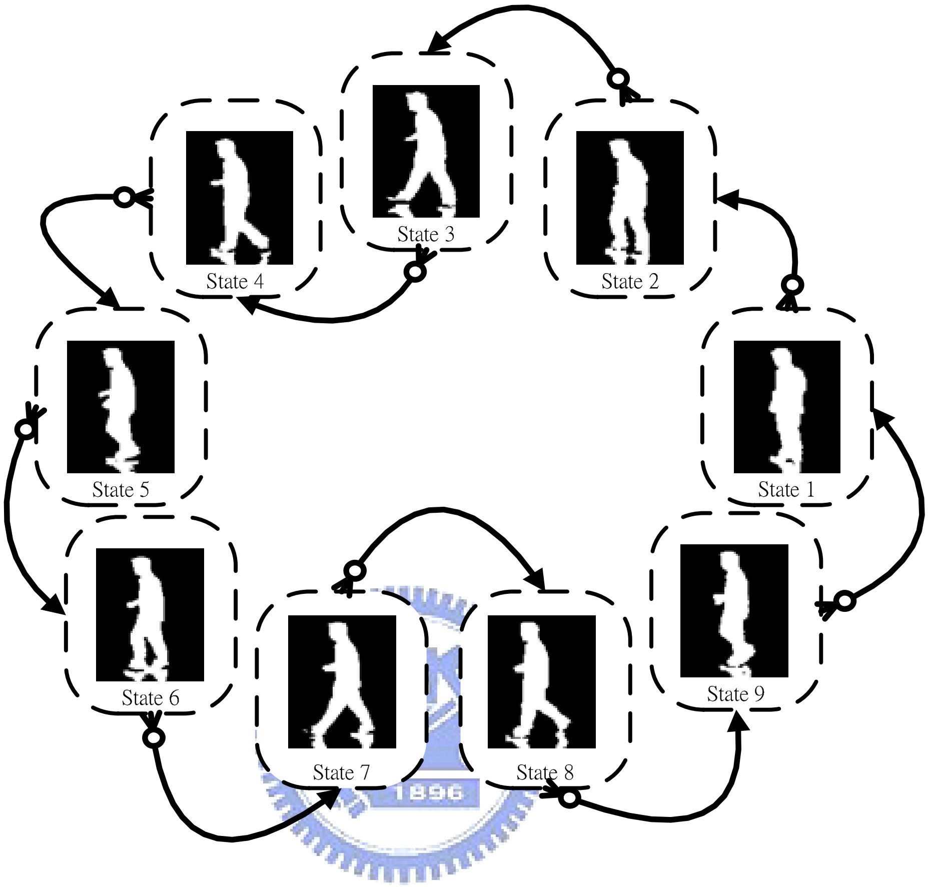


Fig. 3. The state transfer diagram for activity recognition.

Chapter 2 Fundamentals of Eigenspace and Canonical Space Transformation

In video processing, the size of image sequence is usually extremely large. There are many well-known transformation methods to reduce the size of data such as PCA, wavelet, and so on. In our system, we use eigenspace transformation for reducing the size of image sequence. In activity recognition, image sequence matching is the key step to be investigated further. In our system, we use canonical space transformation to improve the recognition performance.

If we compare the image sequence with the templates by the original size, the computation load for a real-time constraint will be a problem (the image size is 128×96 pixels). We need to reduce the template image size. Eigenspace transformation based on Principle Component Analysis is potent to reduce the data size while maintaining representative information. Canonical space transformation based on Canonical Analysis again reduces the data size, if there are fewer classes than principle components, optimizes the class separability, and improves the classification performance. After EST and CST, each image is converted to an one-dimensional canonical vector. Recognition is accomplished in the canonical space.

Assume that there are c training classes (23 essential templates in our case) to be learned. Each class represents same poses that different persons performed. $\mathbf{x}'_{i,j}$ 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

$$[\mathbf{x}'_{1,1}, \cdots, \mathbf{x}'_{1,N_1}, \cdots, \mathbf{x}'_{2,1}, \cdots, \mathbf{x}'_{c,N_{c1}}], \tag{1}$$

where each $\mathbf{x}'_{i,j}$ is a rasterly-scanned vector from the original of size 128×96 pixels.

First, the intensity of each sample image needs to be normalized by

$$\mathbf{x}_{i,j} = \frac{\mathbf{x}'_{i,j}}{\|\mathbf{x}'_{i,j}\|}. \tag{2}$$

Then we need the mean pixel value for training image set. That is

$$\mathbf{m}_{\mathbf{x}} = \frac{1}{N_T} \sum_{i=1}^{c} \sum_{j=1}^{N_i} \mathbf{x}_{i,j} \tag{3}$$

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

$$\mathbf{X} = [\mathbf{x}_{1,1} - \mathbf{m}_{\mathbf{x}}, \cdots, \mathbf{x}_{1,N_1} - \mathbf{m}_{\mathbf{x}}, \cdots, \mathbf{x}_{c,N_c} - \mathbf{m}_{\mathbf{x}}] \tag{4}$$

2.1 Eigenspace Transformation (EST)

Basically EST is used widely 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 for information loss. EST uses the eigenvalues and eigenvectors generated by the data covariance matrix to rotate the original data coordinates along the direction of maximum variance.

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

$$\lambda_i\, \mathbf{e}_i = \mathbf{R}\,\mathbf{e}_i, \qquad i = 1, 2, \cdots, K, \tag{5}$$

where $\mathbf{R} = \mathbf{XX}^{\mathrm{T}}$ and R is a square, symmetric matrix. Now we need to solve Eq. (5) for EST, but $\mathbf{R}$ is too large in dimensionality to solve. Based on singular value decomposition theory [17], we can compute another matrix $\tilde{\mathbf{R}}$ to get the eigenvalues and eigenvectors, that is

$$\tilde{\mathbf{R}} = \mathbf{X}^{\mathrm{T}}\mathbf{X}, \tag{6}$$

where $\tilde{\mathbf{R}}$ is a $N_T \times N_T$ matrix, and is much smaller than R in dimensionality. Assume that the matrix $\tilde{\mathbf{R}}$ 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 can be related to those in $\mathbf{R}$ by

$$\begin{cases} \lambda_i = \tilde{\lambda}_i \\ \mathbf{e}_i = \left(\tilde{\lambda}_i\right)^{-\frac{1}{2}} \mathbf{X}\, \tilde{\mathbf{e}}_i, \end{cases} \tag{7}$$

where $i = 1, 2, \cdots, K$.

These K eigenvectors are used as an orthogonal basis to span a new vector space. Every image can be converted to a point in this K-dimensional space. Based on the theory of PCA, each image can be represented by taking only $k \le K$ largest eigenvalues $|\lambda_1| \ge |\lambda_2| \ge \cdots \ge |\lambda_k|$ and the associated eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k$. Namely we project the image $\mathbf{x}_{i,j}$ into $\mathbf{y}_{i,j}$ in eigenspace by

$$\mathbf{y}_{i,j} = \left[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k\right]^{\mathrm{T}} \mathbf{x}_{i,j}, \tag{8}$$

where $i = 1, 2, \cdots, c$ and $j = 1, 2, \cdots, N_c$. We named this matrix $\left[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k\right]^{\mathrm{T}}$ eigenspace transformation matrix. If this transformation matrix is used in face analysis, the eigenvectors in eigenspace transformation matrix are also known as eigenfaces.

2.2 Canonical Space Transformation (CST)

Based on canonical analysis [18], we can suppose $\{\phi_1, \phi_2, \cdots, \phi_c\}$ represents the classes of transformed vectors by eigenspace transformation and $\mathrm{y}_{i,j}$ is the *j*-th vector in class *i*. The mean vector of entire set can be written as

$$\mathbf{m}_y = \frac{1}{N_T} \sum_i \sum_j \mathbf{y}_{i,j} \tag{9}$$

where $i = 1, 2, \ldots, c$ and $j = 1, 2, \ldots, N_i$. The mean vector of the *i*-th class can be presented by

$$\mathbf{m}_i = \frac{1}{N_i} \sum_{\mathbf{y}_{i,j} \in \phi_i} \mathbf{y}_{i,j} \, . \tag{10}$$

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

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

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 subjective is, as we said before, to minimize $\mathbf{S}_{\mathrm{w}}$ and maximize $\mathbf{S}_{\mathrm{b}}$ simultaneously. That is known as the generalized Fisher linear discriminant function and is given by

$$J(\mathbf{W}) = \frac{\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathrm{b}}\mathbf{W}}{\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathrm{w}}\mathbf{W}}. \tag{11}$$

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

$$\frac{\partial J}{\partial \mathbf{W}} = 0. \tag{12}$$

Suppose $\mathbf{W}^{*}$ be the optimal solution $\mathbf{w}_i^{*}$ be its column vector which is a generated eigenvector and correspond to the *i*-th largest eigenvalues λ_i. According to [18], Eq. (12) can be solved and represented as

$$\mathbf{S}_{\mathrm{b}}\mathbf{w}_i^{*} = \lambda_i \mathbf{S}_{\mathrm{w}}\mathbf{w}_i^{*}. \tag{13}$$

After Eq. (11) is solved, we will obtain c-1 nonzero eigenvalues and their corresponding eigenvectors $[\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_c]$ that create another orthogonal basis and span a (c-1)-dimensional canonical space. By these bases, each point in eigenspace can be projected to another point in canonical space by

$$\mathbf{z}_{i,j} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{c-1}]^{\mathrm{T}} \mathbf{y}_{i,j} . \tag{14}$$

We named this orthogonal basis $[\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{c-1}]^{\mathrm{T}}$ the canonical space transformation matrix.

By merging Eqs. (8) and (14), each image can be projected onto a point in the new (c-1)-dimensional space, that is

$$\mathbf{z}_{i,j} = \mathbf{H}\,\mathbf{x}_{i,j} , \tag{15}$$

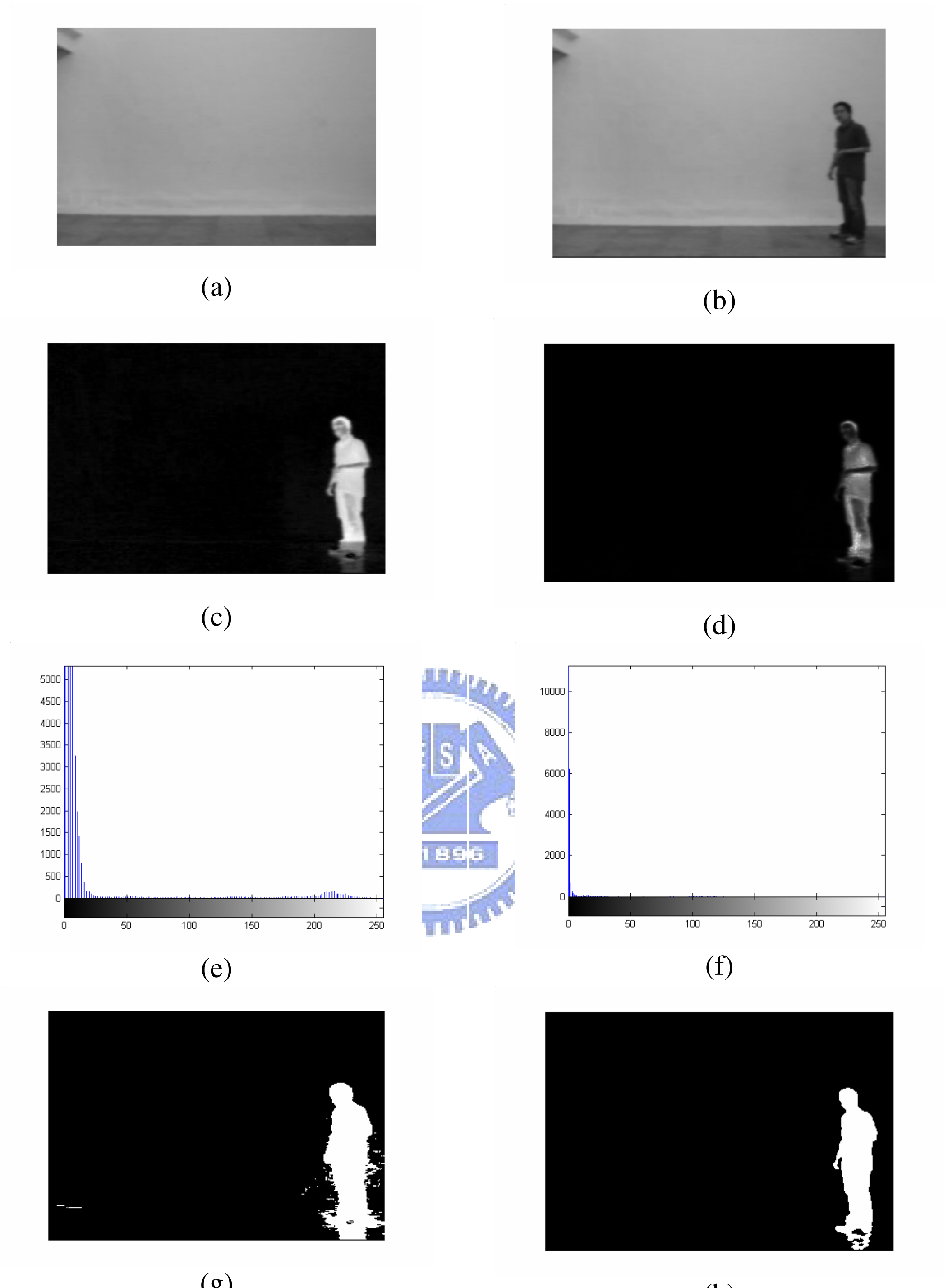
where $\mathbf{H} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{c-1}]^{\mathrm{T}} [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k]^{\mathrm{T}}$.

Chapter 3 Human Activity Recognition System

The first step in a human activity recognition system is background model construction. There are many well-known background models. The most common one is frame difference with threshold. W^4 is a typical one with some modifications. It records the maximum and minimum grayscale and the maximum interframe difference in every position of a frame in a background video. Then every image frame subtracts the maximum and minimum grayscale at each position. If the pixel's absolute value of the subtraction operation is over the maximum interframe difference, the pixel is a foreground one. W^4 admits some rules make the background model an adaptive one. Here we describe the background scene as a computational statistical model. We need a background video to learn by calculating the maximum and minimum gray level in each position in an image, and also the maximum frame ratio.

3.1.1 Background Modeling by Frame Ratio

Let us take a look at Fig. 4. Fig. 4 shows a simple example between frame ratio and frame difference. Fig. 4(a) is a background image. Fig. 4(b) is an image frame. After operating, we can get Fig. 4(c) and Fig. 4(d), and scale them to [0, 255]. Fig. 4(e) is a histogram of frame difference, and we can find there are some noises in low gray level region. Fig. 4(f) is histogram of frame ratio, and we can find that there is a little noise in the low gray level region. Fig. 4(g) and Fig. 4(h) is the binary image after simple thresholding at gray level equal to 15. One can see that there is less noise in the binary image after frame



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 4. The comparison between frame ratio and frame difference. (a) Background image, (b) image frame, (c) frame difference, (d) frame ratio, (e) histogram of frame difference, (f) histogram of frame ratio, (g) foreground pixels of frame difference after simple thresholding, and (h) foreground pixels of frame ratio after simple thresholding.

ratio (Fig. 4(h)) than frame difference (Fig. 4(g)). If we compare Fig. 4(e) with Fig. 4(f), we can find that there is more noise caused by frame difference than by frame ratio. And we can set the threshold at low gray level, 15 in this case, to get a clean binary image.

We develop a method that can be robust to the illumination changes. It is better to utilize frame ratio instead of frame difference. Usually the illumination of a place changes smoothly, but the smooth change will still affect when a long duration is taken. We assume the scene captured by camera can be described as

$$I_i(x, y) = S_i(x, y) r_i(x, y), \tag{16}$$

where I_i is the intensity of the scene, S_i is the spatial distribution of source illumination, r_i is the distribution of scene reflectance, (x,y) is the pixel location in the scene image, and i is the image sequence index. Now we can compare the difference caused by illumination change between frame difference and frame ratio. If we hold the camera still and there are no foreground subjects pass by, namely, background scene only, the reflectance of this background scene should be the same at any time. That is,

$$r_i(x, y) = r(x, y). \tag{17}$$

Although the reflectance is not change, the effect of illumination is still going on. The frame difference and frame ratio between two consecutive frames can respectively be written as

$$\begin{aligned} I_i^d(x, y) - I_{i-1}^d(x, y) &= S_i^d(x, y)r(x, y) - S_{i-1}^d(x, y)r(x, y) \\ &= \left(S_i^d(x, y) - S_{i-1}^d(x, y)\right)r(x, y), \end{aligned} \tag{18}$$

$$\begin{aligned} \log\left(\frac{I_i^r(x, y)}{I_{i-1}^r(x, y)}\right) &= \log\left(\frac{S_i^r(x, y)r(x, y)}{S_{i-1}^r(x, y)r(x, y)}\right) \\ &= \log\left(\frac{S_i^r(x, y)}{S_{i-1}^r(x, y)}\right) \\ &= \log\left(S_i^r(x, y)\right) - \log\left(S_{i-1}^r(x, y)\right), \end{aligned} \tag{19}$$

where I^d is the intensity of scene captured by camera of frame difference, S^d is the spatial distribution of source illumination of frame difference, and I^r and S^r is of frame ratio. From (18) and (19), we can see that if we hold the camera still and there are no foreground subjects passing by, the reflection of the scene will be the same and there are only illumination changes. $r(x,y)$ will affect the frame difference signal because $r(x,y)$ effect is cancelled in Eq. (4) of frame ratio approach.

3.1.2 Background Model

Here we use a simple method for our background model. Each pixel of background scene can be characterized by three statistics: its minimum intensity value $n(x,y)$ and maximum intensity value $m(x,y)$, and the maximum interframe ratio $d(x,y)$ in a background video. Thus we need a background video (for about 10 or 20 seconds) for background model training. Let I be any array containing N consecutive images, and $I^i(x,y)$ be the intensity at pixel location (x,y) in the i-th frame of I. The background model at pixel location (*x,y*) [*m*(*x,y*), *n*(*x,y*), *d*(*x,y*)] can be obtained as

$$\begin{bmatrix} m(x, y) \\ n(x, y) \\ d(x, y) \end{bmatrix} = \begin{cases} \begin{bmatrix} \max_i \{I^i(x, y)\} \\ \min_i \{I^i(x, y)\} \\ \max_i \{I^i(x, y)/I^{i-1}(x, y)\} \end{bmatrix} & \text{if } I^i(x, y)/I^{i-1}(x, y) \geq 1, \\ \begin{bmatrix} \max_i \{I^i(x, y)\} \\ \min_i \{I^i(x, y)\} \\ \max_i \{I^{i-1}(x, y)/I^i(x, y)\} \end{bmatrix} & \text{otherwise.} \end{cases} \tag{20}$$

where $i \in \{1, 2, \ldots, N\}$.

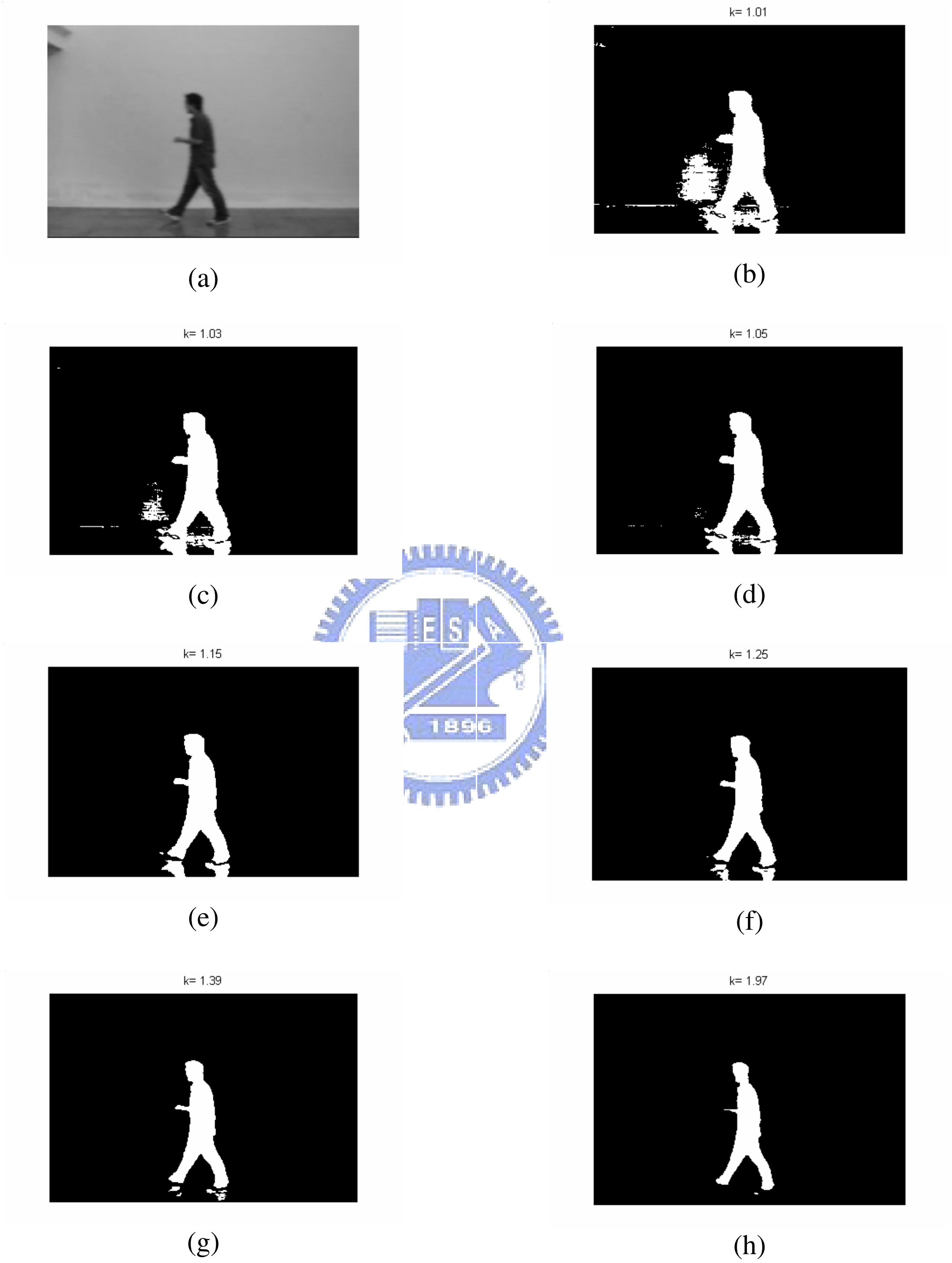
3.1.3 Foreground Subject Extraction

Foreground subjects can be segmented from background scene in every frame of the video stream. Each pixel is classified to either a background or a foreground pixel by comparing with the background model we build in Sec. 2.2. Given the maximum intensity *m*(*x,y*), the minimum intensity *n*(*x,y*), and the maximum interframe ratio *d*(*x,y*) that represent the background scene *B*(*x,y*), a pixel at location (*x,y*) from image frame I^i is a foreground pixel if

$$B(x, y) = \begin{cases} 0, \quad \text{a background pixel if } \begin{cases} I^i(x, y)/m(x, y) < kd(x, y) \\ \text{or} \\ I^i(x, y)/n(x, y) < kd(x, y) \end{cases} \\ \\ 1, \quad \text{a background pixel otherwise.} \end{cases} \tag{21}$$

We ran a series of experiments to determine the best threshold constant *k*. From Fig. 5 we can see the lower *k* causes more noise and the higher *k* filters the foreground pixels that we want to keep. According to our experiments shown in Fig. 5, the best *k* for our data is 1.25.

After analyzing the background scene, we get a binary background image *B*(*x,y*). From *B*(*x,y*), we can extract the foreground region to minimize the following processing region. Foreground region extraction can be accomplished by simply thresholding on X and Y direction. Fig. 6 shows an example of foreground region extraction. Fig. 6(a) is an image frame. Fig. 6(b) is the binary image after background model analysis. Figs. 6(c) and 6(d) show the projection of Fig. 6(b) image onto the X and Y directions, respectively. We can find the coordinates of foreground region in the image by simply thresholding the X and Y projection of Fig. 6(c) and 6(d). And we resize every extracted foreground region to 128×96 pixels by maintaining a normalized height-to-width ratio.



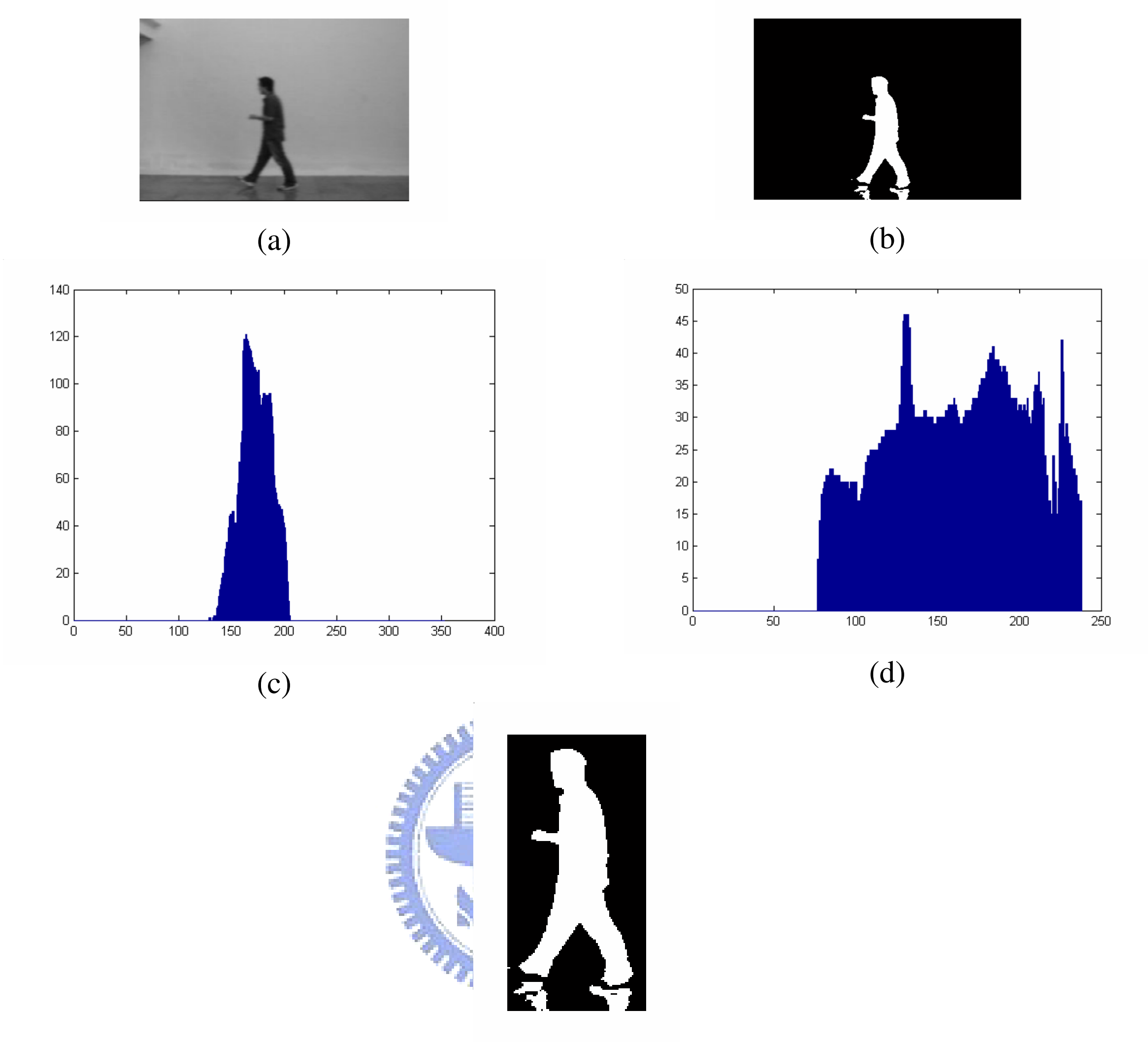
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 5. An example of foreground region extraction for different threshold values. (a) An image frame, (b) k = 1.01, (c) k = 1.03, (d) k = 1.05, (e) k = 1.15, (f) k = 1.25, (g) k = 1.39, and (h) k = 1.97.



(a)

(b)

(c)

(d)

(e)

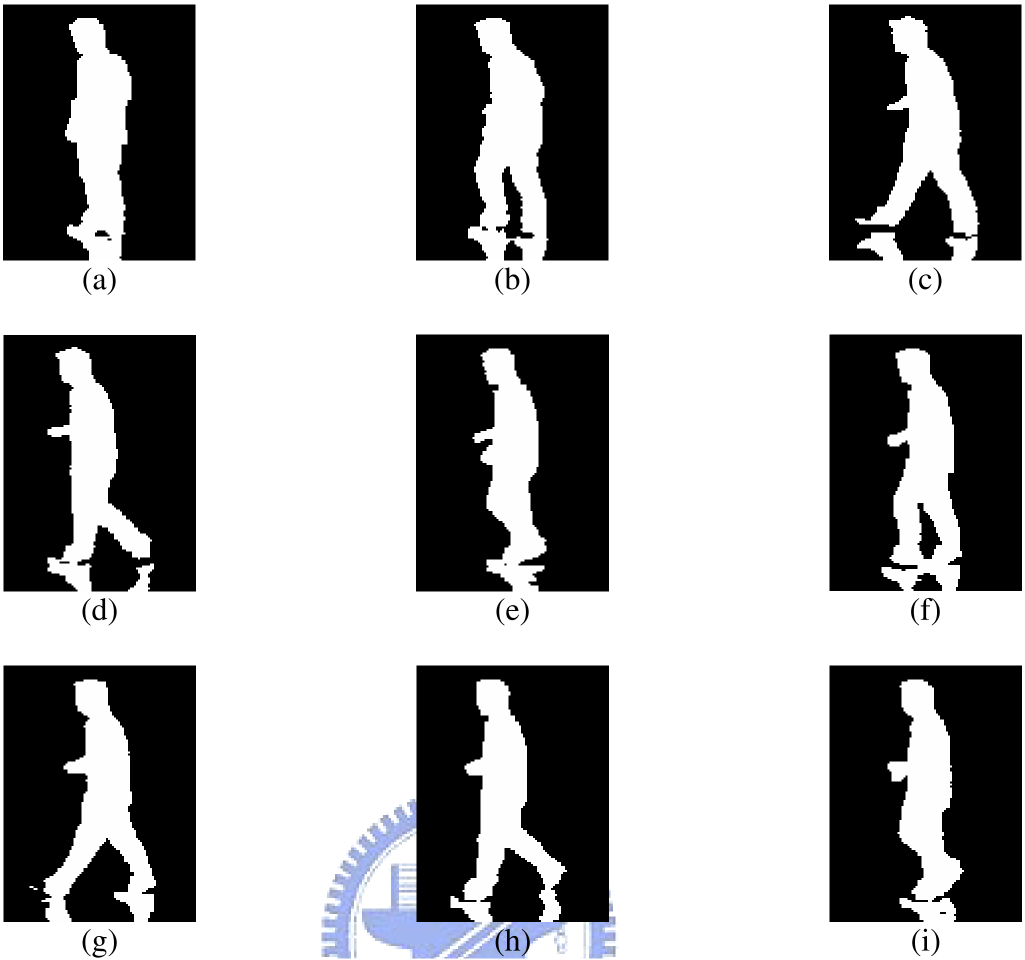
Fig. 6. An example of foreground region extraction. (a) An image frame, (b) binary image after background analysis, (c) projection of (b) onto X direction, (d) projection of (b) onto Y direction, (e) foreground region extracted.

3.2 Activity template selection

As described in Chapter 1, we try to develop a human-like, activity recognition system. In Fig. 7, we can determine that the person in these 8 frames is walking from right to left. According to human vision experience, a few sampled frames of an activity are enough to recognize what a person is doing. We represent each activity by several essential templates and transform these essential templates by Eigenspace transformation (EST) and canonical space transformation (CST).

Eigenspace transformation (EST) based on Principal Component Analysis (PCA) has been demonstrated to be a potent metric in pattern recognition and data analysis. And we combine EST with canonical space transformation (CST) based on Canonical Analysis (CA). This combined method can be used to reduce the data dimensionality, to optimize the separability, and improve the classification performance.

We choose several frames in an activity video stream to represent this activity. Fig. 7 shows the nine templates we choose for the activity "walking from right to left." This representation of an activity is helpful to reduce greatly the size of the video frame images because we recognize the activities only by essential templates instead of whole image frames.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 7. The essential templates of activity “walk from right to left.”

If we observe the contiguous essential frames for certain time interval in an activity, we can see that the poses in these frames are observed very similar each other. Fig. 8 shows the six contiguous essential frames, and it can be seen that the poses of these image are very similar each other. These similar poses after data transformation like EST will converge to one essential template (or a point in canonical space). We consider these templates to be the common pose that everybody will perform. When a man walks, for example, first he lifts his right foot up (Fig. 7(b)) and step it forward (Fig. 7(c)). Then he transfers his weight to his right foot (Fig. 7(d)). And he lifts his left foot up (Fig. 7(e) and 7(f)) then step it forward (Fig. 7(g)), and so on in a periodic manner. Different people perform the same activity may have almost the same poses with a miner difference.

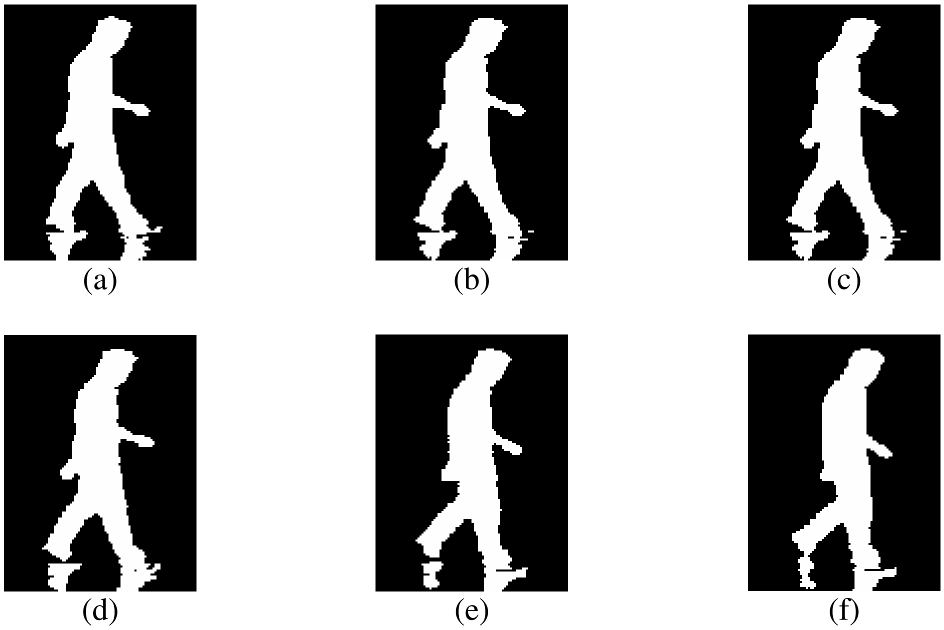


Fig. 8. The six contiguous frames. (a) Frame 28, (b) frame 29, (c) frame 30, (d) frame 31, (e) frame 32, (a) frame 33.

Our scheme to recognize the human action is described below. There are four activities to be recognized in our experiment. They are "walking from right to left," "walking from left to right," "jumping," and "crouching." We choose nine essential templates for "walking from right to left" and "walking from left to right" respectively, three for "common state," one for "jump" and "crouch" respectively. Common state means the common poses between "jump" and "crouch." Fig. 9 shows that when people jump or crouch, the preparing action contains the same or similar poses. We call the preparing action "common state." When a man jumps or crouches, he stands at first (Fig. 9(a)). Then he will crouch his knees slowly (Fig. 9(b) and 9(c)). If he jumps, he will look like Fig. 9(d) afterward. On the other hand he will then look like Fig. 9(e) if he crouches. In summary, we use only the above 23 essential templates, i.e., 23 classes, to recognize the four activity type from the video sequence.

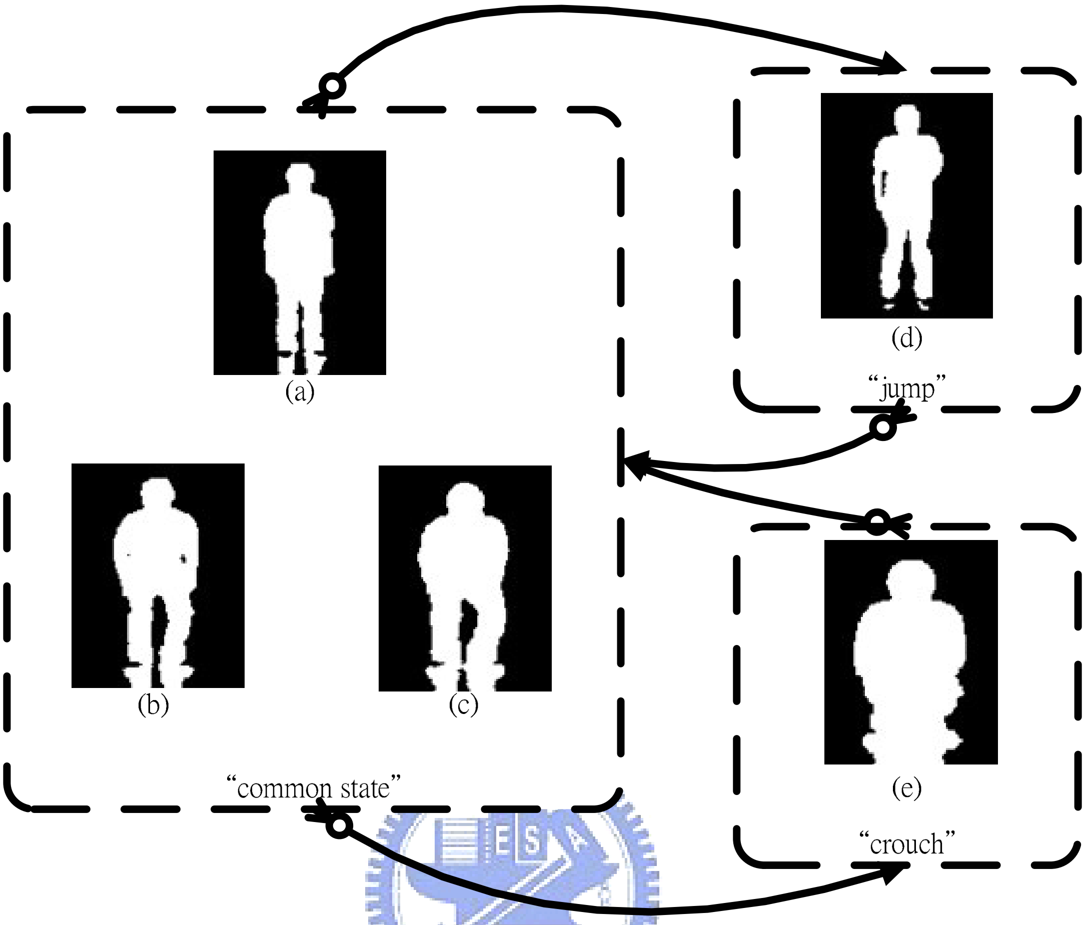


Fig. 9. "Jumping" contains "common states" (a), (b), (c), and (d) while "crouching" contains (a), (b), (c) and (e).

3.3 Classification Algorithm

Every image frame is transformed into canonical space by transformation matrix **H**. The system compares the converted vector, image frame, with the training set **z,** and then makes an activity decision for output. There are many classification algorithms, such as *K*-means clustering, neural networks, *K* nearest neighbors, and so on. Our proposed system applies nearest neighbor classifier and maximum likelihood method to compare the image frames with the training set **Z**.

Let $\mathbf{x}'_k$ be the binary image of subject where *k* is the image sequence index. Like the process of generating the training set **z**, the binary image of subject $\mathbf{x}'_k$ needs to be normalized and to subtract the mean vector (Eq. (3)) in order to become the standardized vector $\mathbf{x}_k$. That is,

$$\mathbf{x}_k = \frac{\mathbf{x}'_k}{\left\|\mathbf{x}'\right\|} - \mathbf{m}_x . \tag{22}$$

After the binary image of subject is normalized, $\mathbf{x}_k$ will be converted into the vector $\mathbf{t}_k$ in canonical space by the transformation matrix **H**. That is,

$$\mathbf{t}_k = \mathbf{H}\mathbf{x}_k \tag{23}$$

Then the system compares this $\mathbf{t}_k$ with the training set **z** to make a classification class by Nearest neighbor algorithm and maximum likelihood algorithm.

3.3.1 Nearest Neighbor Algorithm

This algorithm is memory-based. Given a query vector $\mathbf{t}_k$, the system finds one training vector $\mathbf{z}_{i,j}$ closest in distance to $\mathbf{t}_k$, where $\mathbf{z}_{i,j}$ is the *j*-th column vector in class *i*, and then classify $\mathbf{t}_k$ to class *i*. The measure of distance is Euclidean distance in canonical space:

$$\mathrm{d}_{i,j} = \left\| \mathbf{z}_{i,j} - \mathbf{t}_k \right\| \tag{24}$$

where $\mathrm{d}_{i,j}$ is the Euclidean distance from $\mathbf{z}_{i,j}$ to $\mathbf{t}_k$.

The query vector $\mathbf{t}_k$ is classified to *p*-th class if

$$(p, q) = \arg\min_{(i,j)} \mathrm{d}_{i,j}\,. \tag{25}$$

Details and recognition rate will be described in the next chapter.

3.3.2 Maximum Likelihood Algorithm

Suppose $\mathbf{t}_k$ is a Gaussian vector, and each of its components is independent. We can write $\mathbf{z}_{i,j}$ and $\mathbf{t}_k$ as

$$\mathbf{z}_{i,j} = \left[\mathrm{z}_{i,j}^{1}, \cdots, \mathrm{z}_{i,j}^{m}\right]^{\mathrm{T}}, \tag{26}$$

$$\mathbf{t}_k = \left[\mathrm{t}_k^{1}, \cdots, \mathrm{t}_k^{c-1}\right]^{\mathrm{T}}. \tag{27}$$

where $\mathrm{z}_{i,j}^{1}, \cdots, \mathrm{z}_{i,j}^{m}$ are the elements in $\mathbf{z}_{i,j}$, and $\mathrm{t}_k^{1}, \cdots, \mathrm{t}_k^{c-1}$ are the elements in $\mathbf{t}_k$.

We can estimate the mean and variance of the *m*-th dimension in class *i* by

$$\mu_{i,m} = \frac{\sum_{j=1}^{N_i} z_{i,j}^{m}}{N_i}, \tag{28}$$

$$\sigma_{i,m}^2 = \frac{\sum_{j=1}^{N_i} \left(z_{i,j}^{m} - \mu_{i,m}\right)^2}{N_i - 1}. \tag{29}$$

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

Let C_i denote the *i*-th class, and $\mathbf{C}_i$ denote the covariance matrix of the *i*-th class, and the likelihood function is given by

$$
\begin{aligned}
L(\mathbf{t}_k|\mathrm{C}_i) &= \frac{1}{(2\pi)^{\frac{c-1}{2}}\det^{\frac{1}{2}}(\mathrm{C}_i)}\exp\left[-\frac{1}{2}\mathbf{t}_k{}^{\mathrm{T}}\mathbf{C}_i\mathbf{t}_k\right] \\
&= \prod_{m=1}^{c-1}\frac{1}{\sqrt{2\pi}\sigma_{i,m}}\exp\left[-\frac{1}{2}\frac{(\mathrm{t}_k^m-\mu_{i,m})^2}{\sigma_{i,m}{}^2}\right],
\end{aligned}
\tag{30}
$$

The query vector $\mathbf{t}_k$ is classified to the *p*-th class if

$$
p = \arg\max_i L(\mathbf{t}_k|\mathrm{C}_i). \tag{31}
$$

Details and recognition rate will be described in next chapter.

3.4 Recognition Algorithm for Implementation with Real-Time Consideration

Computational load for image processing and pattern recognition is usually extremely large and hence an important factor to be considered. We propose two recognition algorithms based on the description that mentioned in Sec. 3.3. Algorithm 1 uses the current classification that generated by the nearest neighbor algorithm and the past four classifications to determine the activity type by the majority vote for the current instant.

Algorithm 2 uses the current classification that generate by the maximum likelihood algorithm and the past four classifications to determine the final activity type by summing the grades of the likelihood function in the same class and then the class is determined to be the class having the maximum grade. The recognition rate of these two algorithms will be described in the next chapter.

3.4.1 Algorithm 1

As we mentioned previously, Algorithm 1 determines the activity type by the results of nearest neighbor classification algorithm. Fig. 10 shows the structure of this algorithm. Every image frame is first transformed by EST and CST, and then has a classification result by nearest neighbor classification algorithm. Then system determines the activity type by the current and the past 4 classification results by the majority vote. For example, if the current image frame, t-th, and (t-1)-th image frame are classified to “walking from left to right” while the (t-2)-th, (t-3)-th, and (t-4)-th are classified to “common state,” the activity type for this instant is “common state.”

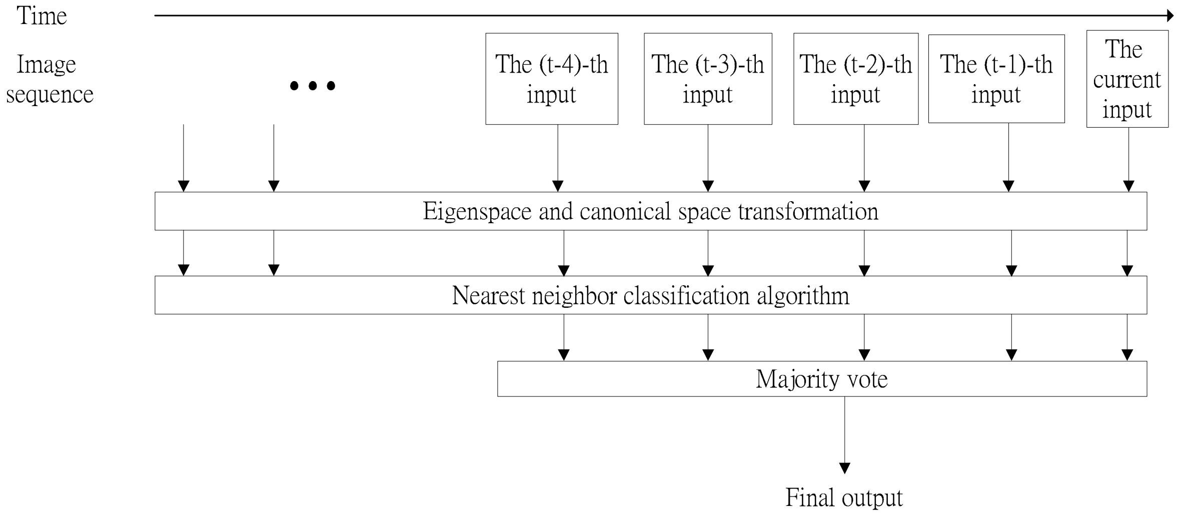


Fig. 10 The structure of algorithm 1.

3.4.2 Algorithm 2

Algorithm 2 determines the activity type by the likelihood function in Eq. (30). Let $g_{k,i}$ denote the grade in class i of k-th image frame. That is

$$\begin{aligned} g_{k,i} &= L(\mathbf{t}_k | \mathrm{C}_i) \\ &= \prod_{m=1}^{c-1} \frac{1}{\sqrt{2\pi}\sigma_{i,m}} \exp\left[-\frac{1}{2} \frac{(\mathrm{t}_k^m - \mu_{i,m})^2}{\sigma_{i,m}{}^2} \right], \end{aligned} \tag{32}$$

where $\mathbf{t}_k$ is the image frame that is transformed by EST and CST, C_i denote the i-th class. As we mentioned previously, Algorithm 2 sums the grades in the same class and finds which class has the maximum grade. The grading function G_i that includes the current and the past four image frames is given by

$$G_i = \sum_{r=k-4}^{k} g_{r,i} \, . \tag{33}$$

Then the activity type for the current instant is classified to *p*-th class by

$$p = \arg\max_i G_i \, . \tag{34}$$

Fig. 11 shows the structure of Algorithm 2. Equivalently, every image frame is first transformed by EST and CST, and then has grades the represent how close it is to certain classes. Then the system gathers the current and the past four grades to generate an activity type for output.

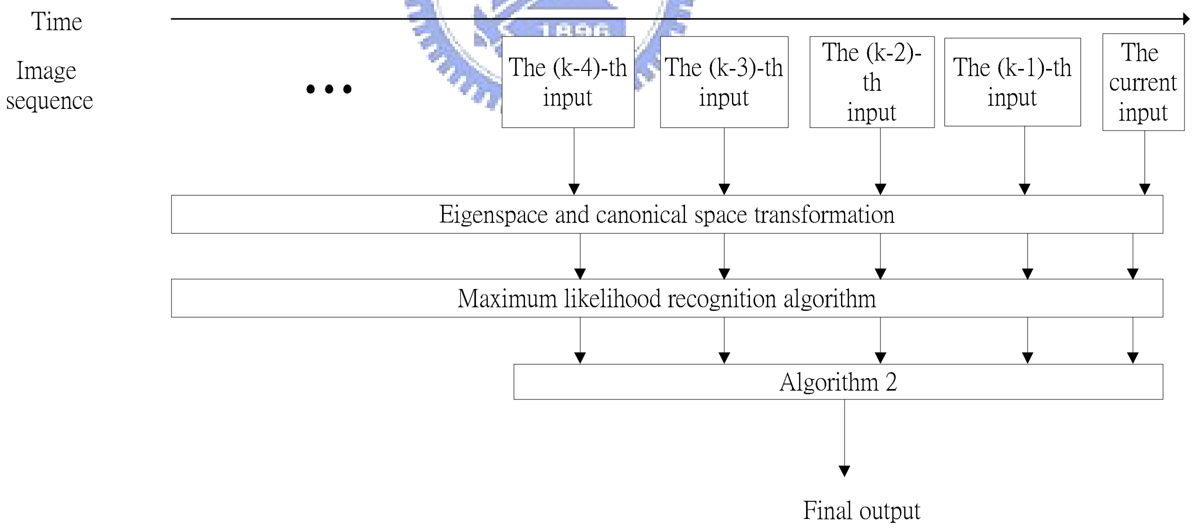
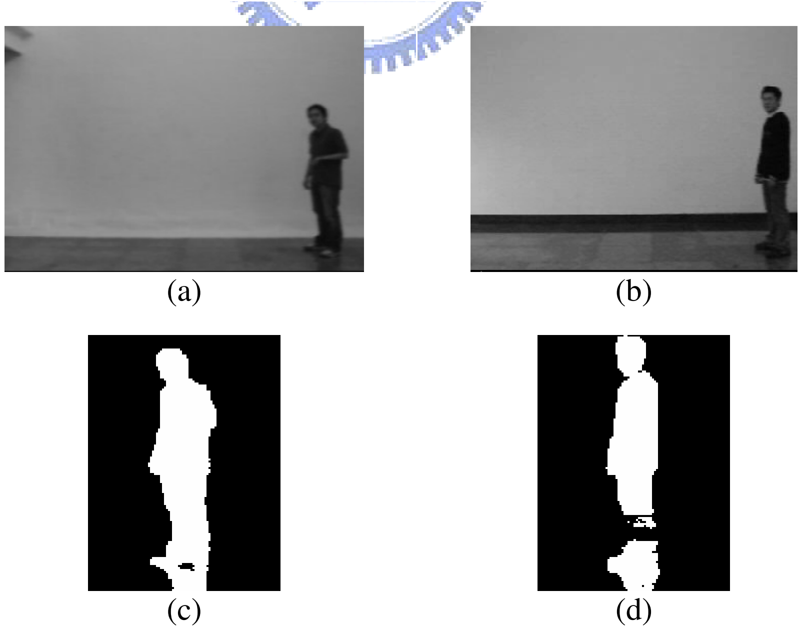


Fig. 11. The structure of Algorithm 2.

Chapter 4 Experimental Results

In our experiment, we test our proposed system on real time-sequence images. And we have two kinds of images in the background: one is with a skirting board on the wall, and another is without skirting board because we cover it on purpose. The skirting board affects seriously the foreground subject extraction. Fig. 12 shows the effects of the skirting board on subject extraction. Fig. 12(a) is an image sequence with a skirting board while Fig. 12(b) is without a skirting board. After subject extraction, we can see that part of the foreground subject overlapping with the skirting board can not be extracted, no matter how we adjust the threshold. We tested the system by these two kinds of images, and moreover applied a cross validation to these images, that is, the images with skirting board be the training set and the images without skirting board be the test sets, and vice versa.


(a) (b)

(c) (d)

Fig. 12. Effects of a skirting board. (a) Image with a skirting board, (b) image without a skirting board, (c) binary image after subject extraction of (a), (d) binary image after extraction of (b).

Recognition was done by the two classification algorithms mentioned in Sec. 3.3, and by two recognition algorithms mentioned in Sec. 3.4. The recognition rate of the nearest neighbor classification algorithm was compared with that of the maximum likelihood classification algorithm while the recognition rate of Algorithm 1 was compared with that of Algorithm 2. By observing the essential templates of "walking from right to left" and "walking from left to right," we found that the interval between these templates is about five frames. Therefore, for Algorithm 1 and 2, we assume the executing speed is six frames per second, namely, 5:1 down sampling is taken on image sequence. To compare the difference in different start position of image sequence, there are five columns in the recognition rate table showing the five different sampling start positions. Recognition rate was computed as the ratio of number of image sequence sample classified correctly to the total number of image sequence sample used. Here we used the leave one out cross validation on these images. The size of image sequence is 640×480, 8-bit grey levels and the size of extracted foreground subjects is 128×96, binary images.

4.1 Experiment 1

In Experiment 1, we used the images without a skirting board to test our proposed system. There were seven persons that performed four activities which are "walking from right to left," "walking from left to right," "jumping," and "crouching." As mentioned in Sec. 3.2, the "jumping" and "crouching" need a "common state" to represent the common poses between jumping and crouching. Therefore we classified images into five groups, i.e., four activity type and "common state", to represent the four activities, and the number of total class i.e., the essential templates is 23. For nearest neighbor and maximum likelihood classification algorithms, there were 2871 images to be classified. For Algorithm 1 and 2, we assume the execute speed is six frames per second; namely 5:1 down sampling on image frame is taken.

The comparison on nearest neighbor and maximum likelihood classification algorithms is shown in Table I. The recognition rates of Algorithm 1 and 2 are shown in Table II and Table III.

Table I

The recognition rates of nearest neighbor and maximum likelihood recognition algorithm in Experiment 1.

	Nearest neighbor	Maximum likelihood
Person 1	87.6	90.3
Person 2	77.6	89.3
Person 3	92.9	93.9
Person 4	92.0	89.5
Person 5	93.0	94.7
Person 6	78.1	80.2
Person 7	93.3	96.4
Total	87.8	90.2

Table II

The recognition rates of Algorithm 1 in Experiment 1.

	Start at sample 1, i.e., 1, 6, 11, 16, 21	Start at sample 2, i.e., 2, 7, 12, 17, 22	Start at sample 3, i.e., 3, 8, 13, 18, 23	Start at sample 4, i.e., 4, 9, 14, 19, 24	Start at sample 5, i.e., 5, 10, 15, 20, 25
Person 1	92.7	96.3	100.0	100.0	96
Person 2	87.5	93.7	85.7	88.7	86.9
Person 3	98.4	96.9	100.0	96.8	95.1
Person 4	98.6	98.6	95.8	97.2	98.6
Person 5	100.0	100.0	97.4	98.7	98.7
Person 6	86.3	87.7	89.0	83.1	78.6
Person 7	100.0	100.0	95.1	95.1	100
Total	94.9	96.1	94.6	94.1	93.3

Table III

The recognition rates of Algorithm 2 in Experiment 1.

	Start at sample 1 i.e., 1, 6, 11, 16, 21	Start at sample 2 i.e., 2, 7, 12, 17, 22	Start at sample 3 i.e., 3, 8, 13, 18, 23	Start at sample 4 i.e., 4, 9, 14, 19, 24	Start at sample 5 i.e., 5, 10, 15, 20, 25
Person 1	96.4	94.4	94.4	98.1	98
Person 2	98.4	98.4	96.8	95.2	100
Person 3	100.0	100.0	100.0	100.0	100
Person 4	100.0	100.0	90.1	85.9	92.9
Person 5	100.0	100.0	100.0	100.0	100
Person 6	89.0	93.2	91.8	87.3	85.7
Person 7	100.0	100.0	100.0	98.4	100
Total	97.7	98.1	96.1	94.8	96.5

4.2 Experiment 2

In Experiment 2, we used the images with a skirting board to test our proposed system. As we mentioned previously, a skirting board will affect the extraction of the foreground subject. As in Experiment 1, there were also seven persons that performed four activities. But these seven persons differed from the seven persons in Experiment 1. As in Experiment 1, we classified images into five groups to represent the four activities, and the number of total class is 23. For nearest neighbor and maximum likelihood classification algorithms, there were 2871 image sequences to be classified. For Algorithm 1 and 2, we assume the execute speed is up-sampled by processing only six frames per second.

The comparison of nearest neighbor and maximum likelihood classification algorithms is shown in Table IV. The recognition rates of Algorithm 1 and 2 are shown in Table V and Table VI.

Table IV
The recognition rates of nearest neighbor and maximum likelihood recognition algorithms in Experiment 2.

	Nearest neighbor	Maximum likelihood
Person 1	97.8	98.9
Person 2	93.7	97.4
Person 3	82.4	86.1
Person 4	67.4	62.8
Person 5	94.8	95.0
Person 6	97.5	98.3
Person 7	92.3	89.2
Total	89.3	89.7

Table V

The recognition rates of Algorithm 1 in Experiment 2.

	Start at sample 1 i.e., 1, 6, 11, 16, 21	Start at sample 2 i.e., 2, 7, 12, 17, 22	Start at sample 3 i.e., 3, 8, 13, 18, 23	Start at sample 4 i.e., 4, 9, 14, 19, 24	Start at sample 5 i.e., 5, 10, 15, 20, 25
Person 1	100.0	100.0	100.0	100.0	98.1
Person 2	90.7	100.0	100.0	100.0	96.1
Person 3	82.1	80.3	81.8	80.0	81.5
Person 4	65.5	70.4	64.8	66.0	70.6
Person 5	96.6	100.0	96.5	98.2	100
Person 6	100.0	100.0	100.0	100.0	100
Person 7	100.0	100.0	100.0	100.0	91.7
Total	90.5	92.7	91.6	91.7	91.1

Table VI

The recognition rates of Algorithm 2 in Experiment 2.

	Start at sample 1 i.e., 1, 6, 11, 16, 21	Start at sample 2 i.e., 2, 7, 12, 17, 22	Start at sample 3 i.e., 3, 8, 13, 18, 23	Start at sample 4 i.e., 4, 9, 14, 19, 24	Start at sample 5 i.e., 5, 10, 15, 20, 25
Person 1	100.0	100.0	100.0	100.0	100
Person 2	100.0	100.0	100.0	100.0	100
Person 3	100.0	98.5	93.9	95.4	95.4
Person 4	63.6	57.4	53.7	58.5	66.7
Person 5	100.0	100.0	100.0	98.2	100
Person 6	100.0	100.0	100.0	100.0	100
Person 7	96.1	100.0	100.0	93.8	93.8
Total	94.5	93.9	92.6	92.5	94

4.3 Experiment 3: Cross Validation Check

In Experiment 3, we used the images without a skirting board to be training set, and the images with a skirting board to be the test sets. That is, let the seven persons in Experiment 1 be the training set, and the seven persons in Experiment 2 be the test patterns. The comparisons of nearest-neighbor and maximum likelihood classification algorithms are shown in Table VII. The recognition rates of Algorithm 1 and 2 are shown in Table VIII and Table IX.

Table VII

The cross-validation recognition rate comparisons of nearest neighbor and maximum likelihood recognition algorithms in Experiment 3.

	Nearest neighbor	Maximum likelihood
Person 1	86.0	94.4
Person 2	74.1	86.8
Person 3	82.9	89.2
Person 4	67.1	82.7
Person 5	80.9	86.9
Person 6	63.7	89.8
Person 7	62.5	81.5
Total	74.3	87.5

Table VIII

The cross-validation recognition rates of Algorithm 1 in Experiment 3.

	Start at sample 1 i.e., 1, 6, 11, 16, 21	Start at sample 2 i.e., 2, 7, 12, 17, 22	Start at sample 3 i.e., 3, 8, 13, 18, 23	Start at sample 4 i.e., 4, 9, 14, 19, 24	Start at sample 5 i.e., 5, 10, 15, 20, 25
Person 1	96.5	92.9	87.3	92.7	92.6
Person 2	83.3	72.2	75.9	79.6	78.8
Person 3	77.6	80.3	87.9	90.8	83.1
Person 4	69.1	72.2	74.0	69.8	66.7
Person 5	81.0	89.5	84.2	87.7	80.7
Person 6	69.0	63.8	71.9	64.8	68.5
Person 7	76.0	68.0	58.3	54.1	66.7
Total	79.0	77.2	77.7	78.0	77.2

Table IX

The cross-validation recognition rates of Algorithm 2 in Experiment 3.

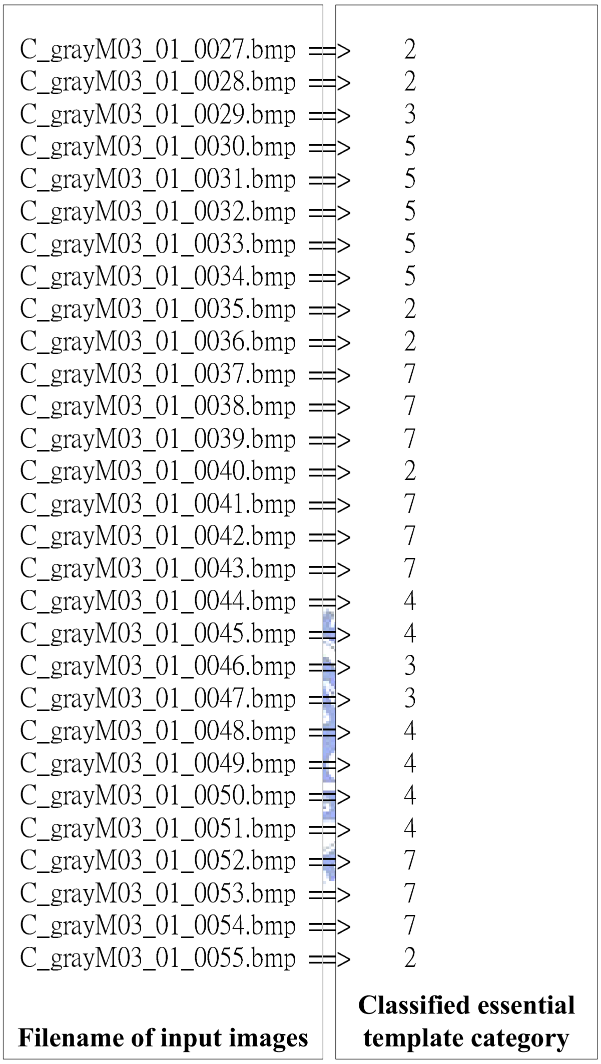
	Start at sample 1 i.e., 1, 6, 11, 16, 21	Start at sample 2 i.e., 2, 7, 12, 17, 22	Start at sample 3 i.e., 3, 8, 13, 18, 23	Start at sample 4 i.e., 4, 9, 14, 19, 24	Start at sample 5 i.e., 5, 10, 15, 20, 25
Person 1	94.7	100.0	100.0	100.0	100
Person 2	81.5	81.5	83.3	83.3	86.5
Person 3	94.0	97.0	97.0	96.9	95.4
Person 4	80.0	81.5	81.5	81.1	80.4
Person 5	93.1	89.5	87.7	93.0	84.2
Person 6	84.5	89.7	91.2	90.7	88.9
Person 7	92.2	82.0	91.7	85.4	85.4
Total	88.8	89.1	90.5	90.4	89

4.4 Summary

As mentioned in Chapter 1, we can recognize the activity of the subject by down sampling the image sequence instead of all consecutive image frames in order to reduce the recognition complexity, decrease the computational load, and improve the recognition performance. By observing the Table II, Table III, Table V, Table VI, Table VIII, and Table IX, the recognition rates remain good enough even though down sampling is taken.

Fig. 13 shows partial classification result of maximum likelihood recognition algorithm. As we mentioned in Sec. 3.2, the similar poses (consecutive frames) after EST and CST converge to an essential template (class) shown in Fig. 13. However, in a sequence matching problem, one needs to know the start and end position of a sequence in order to compare with the target sequences. In our proposed system, an activity is represented by several essential templates and the similar pose images after EST and CST converge to one essential template (class). Therefore, we do not need to know the start and end position of image sequence. Fig. 14 shows partial classification result of implement consideration Algorithm 2. From Fig. 14, we can find that the activity "crouching" can be represented by our system.

In summary, the maximum likelihood method performs better in recognition rate than nearest neighbor method and Algorithm 2 shows better recognition rate than Algorithm 1.



```
C_grayM03_01_0027.bmp ==>     2
C_grayM03_01_0028.bmp ==>     2
C_grayM03_01_0029.bmp ==>     3
C_grayM03_01_0030.bmp ==>     5
C_grayM03_01_0031.bmp ==>     5
C_grayM03_01_0032.bmp ==>     5
C_grayM03_01_0033.bmp ==>     5
C_grayM03_01_0034.bmp ==>     5
C_grayM03_01_0035.bmp ==>     2
C_grayM03_01_0036.bmp ==>     2
C_grayM03_01_0037.bmp ==>     7
C_grayM03_01_0038.bmp ==>     7
C_grayM03_01_0039.bmp ==>     7
C_grayM03_01_0040.bmp ==>     2
C_grayM03_01_0041.bmp ==>     7
C_grayM03_01_0042.bmp ==>     7
C_grayM03_01_0043.bmp ==>     7
C_grayM03_01_0044.bmp ==>     4
C_grayM03_01_0045.bmp ==>     4
C_grayM03_01_0046.bmp ==>     3
C_grayM03_01_0047.bmp ==>     3
C_grayM03_01_0048.bmp ==>     4
C_grayM03_01_0049.bmp ==>     4
C_grayM03_01_0050.bmp ==>     4
C_grayM03_01_0051.bmp ==>     4
C_grayM03_01_0052.bmp ==>     7
C_grayM03_01_0053.bmp ==>     7
C_grayM03_01_0054.bmp ==>     7
C_grayM03_01_0055.bmp ==>     2
```

Filename of input images
Classified essential template category

Fig. 13. A partial list of classification by maximum likelihood recognition algorithm.

```
***********************************
The 5 temporal sampled image frames are 1 6 11 16 21
This instant is classified to group  “Common State”
***********************************
The 5 temporal sampled image frames are 6 11 16 21 26
This instant is classified to group  “Crouch”
***********************************
The 5 temporal sampled image frames are 11 16 21 26 31
This instant is classified to group  “Crouch”
***********************************
The 5 temporal sampled image frames are 16 21 26 31 36
This instant is classified to group  “Crouch”
***********************************
The 5 temporal sampled image frames are 21 26 31 36 41
This instant is classified to group  “Crouch”
***********************************
The 5 temporal sampled image frames are 26 31 36 41 46
This instant is classified to group  “Crouch”
***********************************
The 5 temporal sampled image frames are 31 36 41 46 51
This instant is classified to group  “Common State”
***********************************
The 5 temporal sampled image frames are 36 41 46 51 56
This instant is classified to group  “Common State”
***********************************
The 5 temporal sampled image frames are 41 46 51 56 61
This instant is classified to group  “Common State”
***********************************
The 5 temporal sampled image frames are 46 51 56 61 66
This instant is classified to group  “Common State”
***********************************
The 5 temporal sampled image frames are 51 56 61 66 71
This instant is classified to group  “Common State”
***********************************
The 5 temporal sampled image frames are 56 61 66 71 76
This instant is classified to group  “Common State”
```

Fig. 14. A partial list of classification by implement consideration Algorithm 2.

Chapter 5 Conclusion

Human activity recognition finds application in the field of security surveillance, home care, etc. In this thesis, we present a system for video-based human activity recognition by transforming the images into canonical space. In our system, foreground subject is first extracted as the binary image by a statistical background model using frame ratio, and then transformed by EST and CST, and recognition is done in canonical space. Without referring any geographic information such as location, path, and velocity of the subject, our proposed system uses only the binary images of subject to recognize the activity and works very well. By using several essential templates to represent an activity, our proposed system can recognize the activity of the subject by down sampling the image sequence instead of all consecutive image frames in order to reduce the recognition complexity, decrease the computational load, and improve the recognition performance.

In summary, we propose here a representation for human activity which can correctly describe the four activities, and develop a robust system for activity recognition. This system decreases the dimensionality efficiently and is robust to the illumination change.

References

[1] 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 Analysis and Interpretation*, 1998, pp. 180–185.

[2] P. S. Huang, C. J. Harris, and M. S. Nixon, “Human gait recognition in canonical space using temporal templates,” *Vis. Imag. Signal Process.*, vol. 146, no. 2, pp. 93–100, 1999.

[3] D. J. Jobson and G. A. Woodell, “Properties of a center/surround retinex part two: surround design”, *NASA Technical Memorandum #110188*, 1995.

[4] Z. Rahman, “Properties of a centerhrround retinex part one: signal processing design”, *NASA Contractor Report #198194*, 1995.

[5] S. Park, J. K. Aggarwal, “Segmentation and Tracking of Interacting Human Body Parts under Occlusion and Shadowing,” in *Proc. of the Workshop on Motion and Video Computing*, pp.105, Dec. 05–06, 2002

[6] M. K. Leung and Y. H. Yang, “First sight: A human-body outline labeling system,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 4, pp. 359-377, Apr. 1995.

[7] I. Haritaoglu, D. Harwood, and L. Davis, “Ghost: A Human Body Part Labeling System Using Silhouettes,” in *Proc. Int'l Conf. Pattern Recognition*, 1998.

[8] J. Yamato, J. Ohya, K. Ishii, “Recognizing Human Action in Time-Sequential Images using Hidden Markov Model,” In *Proc. IEEE CVPR*, pp. 379–385, 1992.

[9] A. F. Bobick and J. W. Davis, “The Recognition of Human Movement Using Temporal Templates,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, Mar. 2001.

[10] R. Bodor, B. Jackson and N. Papanikolopoulos, “Vision-Based Human Tracking and Activity Recognition,” in *Proc. of the 11th Mediterranean Conf. on Control and Automation*, June 18–20, 2003

[11] O. Masoud and N. Papanikolopoulos, “Recognizing human activities,” in *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance*, Miami, FL, Jul. 2003, pp. 157–162.

[12] I. Haritaoglu, D. Harwood and L. S. Davis, “W4: Real-Time Surveillance of People and Their Activities,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.22, no. 8, pp. 809–830, August 2000.

[13] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, “Human Activity Recognition Using Multidimensional Indexing,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp.1091–1105, August 2002.

[14] K.R. Rao and J. Ben-Arie, “Multiple Template Matching Using the Expansion Filter,” *IEEE Trans. Video Technology*, vol. 4, no. 5, pp. 490–504, 1994.

[15] K R Rao and J. Ben-Arie , “Optimal Edge Detection Using Expansion Matchingand Restoration,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13,no. 12, pp. 1169–1182,Dec. 1994.

[16] J. Ben-Arie and KR Rao , "A Novel Approach for Template Matching by Nonorthogonal Image Expansion," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 3,no. 1, pp. 71–84, 1993.