Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 453064, 15 pages
doi:10.1155/2010/453064

Research Article

Recognizing Human Actions Using NWFE-Based Histogram Vectors

Cheng-Hsien Lin, Fu-Song Hsu, and Wei-Yang Lin

Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan

Correspondence should be addressed to Fu-Song Hsu, hfs95p@cs.ccu.edu.tw

Received 15 December 2009; Revised 18 March 2010; Accepted 11 May 2010

Academic Editor: ChangIck Kim

This study presents a novel system for human action recognition. Two research issues, namely, motion representation and subspace learning, are addressed. In order to have a rich motion descriptor, we propose to combine the distance signal and the width feature so that a silhouette can be characterized in more detail. These two features provide complementary information and are integrated to yield a better discriminative power. The combined features are subsequently quantized into mid-level features using k-means clustering. In the mid-level feature space, we apply the Nonparametric Weighted Feature Extraction (NWFE) to construct a compact yet discriminative subspace model. Finally, we can simply train a Bayes classifier for recognizing human actions. We have conducted a series of experiments on two publicly available datasets to demonstrate the effectiveness of the proposed system. Compared with the existing approaches, our system has a significantly reduced complexity in classification stage while maintaining high accuracy.

1. Introduction

Recognizing human actions from video sequences is an important area of research in computer vision. This technology has many practical applications such as video surveillance [1–3], human-computer interaction [4], entertainment [5], sports video analysis [6], and smart rooms [7]. However, machine understanding of human behaviour has remained a challenging and sophisticated task due to several reasons. For example, there is a problem originating from the diversity in the way an action is performed by different people. Moreover, actions often last for various lengths of time.

To resolve the above-mentioned issues, one needs a reliable representation that can deal with spatial-temporal scaling variations associated with human actions. The chosen representation must also encapsulate the unique characteristics of an action performed by different persons. After obtaining an action descriptor, the other key issue is how to develop a classification strategy in the resultant feature space. In the last decade, Hidden Markov Model (HMM) is arguably the most popular approach for modelling and classifying human actions. However, there is no systematic method to determine the structure of an HMM. More importantly, one usually does not have sufficient training data for learning the model parameters.

According to the above analysis, we devote our attention to two research challenges in human action recognition, namely, the representation scheme and how to conduct a meaningful learning with limited number of training samples. To put it differently, the proposed framework is basically composed of the following two modules: motion representation and subspace learning.

For the first part, a continuous human action is considered as a sequence of discrete codewords. In particular, background subtraction is applied to extract body silhouette and their boundaries are obtained by contour tracing. Next, the distance signal feature [8] and width feature [9] are computed from the pose contour. These two features are concatenated to yield a more discriminative pose representation. We can then represent a human action as a sequence of symbols, that is, a string, by quantizing the combined features into codewords. The k-means clustering is utilized to construct a codebook due to its simplicity. Once actions have been converted into strings, the matching between two video

sequences is reduced to the problem of measuring string distance. There are many metrics available for assessing the distance between two strings.

Given a string distance metric, we can perform action recognition by using the nearest neighbour rule, which classifies the input data to the category of its nearest neighbour. However, nearest neighbour searching is very time-consuming, especially when the database is large. Hence, we propose to use a subspace learning method, called Nonparametric Weighted Feature Extraction (NWFE) [10], to project the original high-dimensional feature space to a low-dimensional subspace. By exploiting the low-dimensional structure embedded in high-dimensional data, we can not only achieve high-recognition rate but also significantly reduce the computational complexity for classification.

The rest of this paper is organized as follows: Section 2 provides a brief review on the related work. The proposed system is described in detail in Section 3. In Section 4, we conduct several experiments to evaluate the performance of the proposed method. Comparisons with many state-of-the-art systems are included. Finally, We make concluding remarks in Section 5.

2. Related Work

Ways in which a video sequence can be compared to pre-stored instances of actions for recognition were developed. Similar to the surveys by [11, 12], some related studies are summarized under the following groups.

2.1. Template Matching. As stated in [13], human action recognition can be considered to classify time varying feature data. For example, one action can be denoted by one feature vector, obtained from the whole action sequence. Through these vectors, the distance with minimum value is selected as the criterion for recognition. In the early work by Bobick and Davis [5], the temporal template is used to characterize each action with the binary motion energy image (MEI) and motion-history image (MHI). The 7 Hu moments [14] are applied to compute the statistical descriptions of these temporal templates to generate feature vectors. To recognize an input action, the Mahanalobis distance is calculated between the feature vector of the input and each labeled action entry. To extend this, Weinland et al. [15] proposed Motion History Volumes (MHV) as a free-viewpoint representation of human actions. Hsieh and Hsu [16] present a novel string representation scheme to transform each action sequence into a set of symbols, and therefore they can adopt the string matching algorithm to realize human action recognition.

The advantage of template matching is low computational complexity. However, it is usually suffering from the noise and various length of human movements. In addition, a common theme of all these approaches [5, 13, 15, 16] is that the matching process by employing the nearest neighbour classifier with the simple similarity measurements (e.g., Euclidean distance, Mahanalobis distance). However, the performance of the matching process can still be improved due to the fact that the computational cost of the nearest neighbour rule will be proportional to the number of sequences in database.

2.2. State-Space Approaches. The approach consists of some states from the predefined static postures. Any action can be considered as a set through various states. The criterion for recognition is carried out by selecting the maximum value of joint probability through these sets. Some practical applications have been reported that are based on the state-space models [3, 6, 17–20]. Our present work is more related to the approaches of converting a continuous human action into a discrete symbol sequence over the state-space that represents human motion [17, 20]. Specifically, Yamato et al. [6] proposed the first HMM-based human action recognition system to classify six tennis strokes. Chen et al. [20] used star skeleton as a representative descriptor of human posture to recognize human behavior. Liang et al. [17] attempted to learn and recognize human actions through atomic actions.

Rather than using the simple similarity measurement as the criterion, the state-space approach usually has a better result. However, it usually involves complex iterative computation. Meanwhile, research focusing on how to select the proper number of states is still scarce. To overcome some of these difficulties, in this paper we propose an effective and efficient framework that combines the advantages of the template matching and state-space approach. We hope that the proposed strategy can improve the accuracy of recognition rate and reduce computational complexity. The details are described in the following sections.

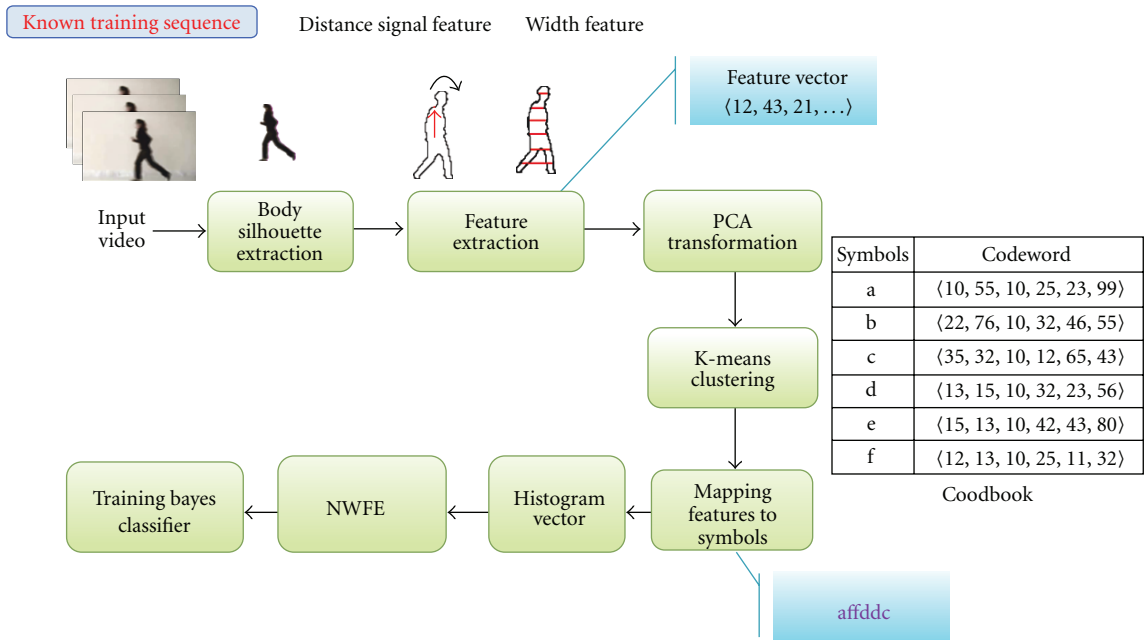
3. Proposed Method

3.1. System Overview. The proposed system basically consists of four parts, including body silhouette extraction, feature extraction, mapping features to symbols, and action recognition. The flowcharts of training and recognition processes in our system are shown in Figure 1.

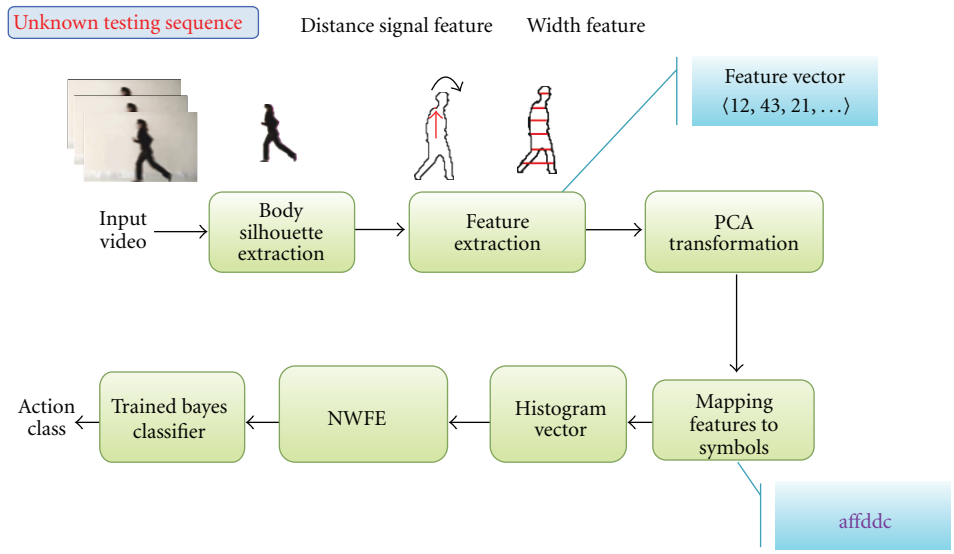
For body silhouette extraction, we assume that videos are captured by a stationary camera. Because the camera is static, we can create a background image of the captured videos by the Gaussian model [21]. Then, background subtraction is applied to segment foreground objects.

In the next step, we first obtain the pose contour from a body silhouette. Then, the distance signal feature [8] and the width feature [9] are computed from the resulting pose contour. Both of them are concatenated into one column vector for the latter processes. To achieve fast recognition, we apply the Principal Component Analysis (PCA) to reduce the dimensionality of the combined feature.

Up to this point, we have represented each human action as a pose feature sequence, which is subsequently converted into a symbol sequence. In particular, we use the k-means clustering method to create a codebook where each codeword is the mean of one cluster. Afterwards, we represent an extracted pose feature vector by the nearest codeword. Notice that each codeword can be denoted by



(a) Process flow of training



(b) Process flow of recognition

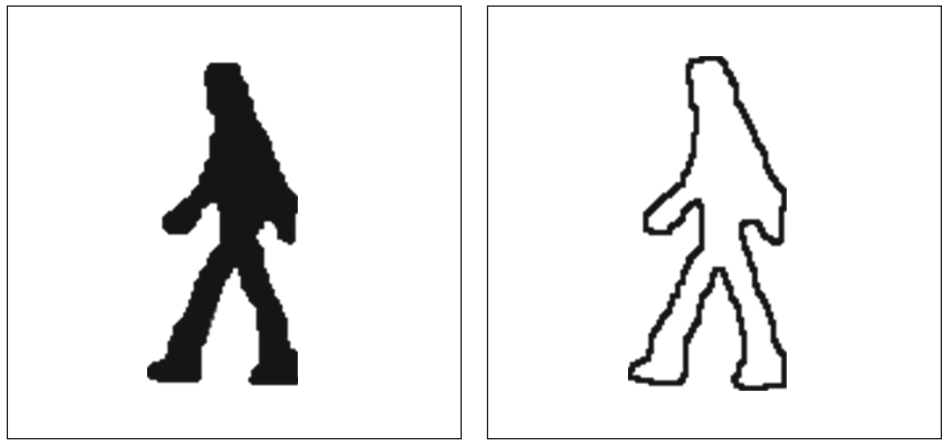
Figure 1: Flowchart of the proposed system.

a symbol. As a result, a human action sequence is represented by a symbol sequence, that is, a string.

Since we convert action sequences into strings, the action recognition simply boils down to string matching. That is, the category of an unknown human action can be determined by measuring some string distance. In this setting, the nearest neighbour rule seems to be a natural choice for classifying an unknown action. However, the speed of this scheme, especially when the database is large, is slow because it typically has to match the input with all the training samples. In this study, we propose a novel method based on the Nonparametric Weighted Feature Extraction (NWFE) [10] to tackle this problem.

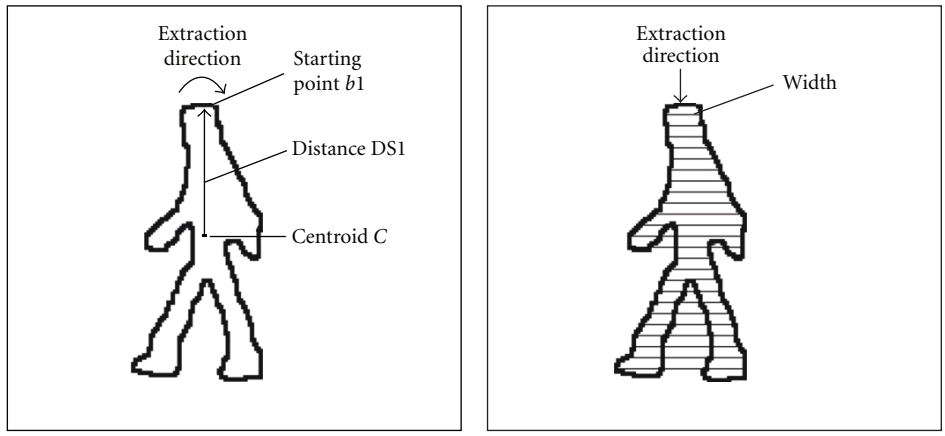
The details are described in the following sections.

3.2. Body Silhouette Extraction. Given a video, it is important to derive foreground objects from background. In our system, we use the background subtraction method [21] to segment foreground objects. The basic concept of background subtraction is to obtain foreground image by thresholding the difference between the current image and a background image. In our system, the running average is



(a) (b)

Figure 2: Human binary silhouette and its contour.



(a) distance signals (b) width features

Figure 3: The illustration of feature extraction.

used as a background model and the corresponding update equations are given by

$$\begin{aligned} \alpha_{t+1} &= \beta\alpha_t + (1-\beta)\, p_{t+1}, \\ \sigma_{t+1} &= \beta\sigma_t + (1-\beta)\,|\alpha_{t+1} - p_{t+1}|, \end{aligned} \tag{1}$$

where p_t is a pixel value in the tth frame, β is the learning rate, and α_t and σ_t are the running average and the standard deviation associated with p_t. The interested reader is referred to [21] for more details. Notice that our system requires detailed silhouettes to be extracted, which might be a challenging task given real-world noisy videos.

3.3. Feature Extraction. Figure 2 shows a human silhouette and its contour. Two features, the distance signal [8] and the width features [9], are extracted from pose contour. These two features can be combined to yield better recognition results. However, doing so will inevitably increase the dimensionality of the feature space. A common way to resolve this dilemma is to use dimension reduction techniques. We describe these steps in more detail in the following subsections.

3.3.1. The Distance Signal Feature and the Width Feature. The first step in generating the distance signal is to calculate the centroid $\boldsymbol{\xi}$ of a silhouette, which is given by

$$\boldsymbol{\xi} = \left(\frac{1}{N_s}\sum_{i=1}^{N_s} x_i, \frac{1}{N_s}\sum_{i=1}^{N_s} y_i\right), \tag{2}$$

where (x_i, y_i) denotes the coordinate of ith pixel in the silhouette and N_s is the number of pixels in the silhouette. Next, let $\{\mathbf{b}_i\}_{i=1}^{N_b}$ be the contour of the silhouette that contains N_b points ordered from top center point in clockwise direction. The distance signal $\{d_i^{\mathrm{DS}}\}_{i=1}^{N_b}$ is defined as

$$d_i^{\mathrm{DS}} = \|\mathbf{b}_i - \boldsymbol{\xi}\|, \tag{3}$$

where $\|\cdot\|$ denotes the L^2 norm.

The distance signal should be normalized because silhouettes are of varying sizes. One should realize that even the

silhouette size of the same person changes from frame to frame. Let ζ denote a predefined constant. The length of a distance signal is standardized as follows:

$$\hat{d}_i^{\mathrm{DS}} = d^{\mathrm{DS}}_{\lfloor i(N_b/\zeta)\rfloor}, \quad \forall i \in [1 \ldots \zeta], \tag{4}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. Its amplitude is normalized by

$$\overline{d}_i^{\mathrm{DS}} = \frac{\hat{d}_i^{\mathrm{DS}}}{\sum_{j=1}^{\zeta} \hat{d}_j^{\mathrm{DS}}} \tag{5}$$

so that the elements of the normalized distance signal must sum to one. Figure 3(a) shows an illustration of computing the distance signal.

Cherla et al. [9] demonstrate that the width of a silhouette contour carries valuable information for recognizing actions. As shown in Figure 3(b), the width feature is simply the distance between the left-most and right-most contour pixels, calculated from top to bottom at different heights. Since different people have different silhouette size, the width features must also be normalized to reduce the effect of scale variations. The normalization of the width feature is the same as the normalization of the distance signal feature (see (5)).

3.3.2. The Combined Feature. We have introduced how to extract distance signal and width features from a human silhouette in the previous sections. These two features could be combined in various ways. Here, we simply concatenate these two features into one column vector, called the combined feature. The effectiveness of the combined feature will be evaluated in Section 4.

In order to perform fast recognition, we need to make the dimensionality of the feature space as small as possible while maintaining the recognition accuracy. Here, we construct a lower-dimensional subspace by using the PCA. The feature vectors obtained in the previous step are projected onto this space.

Consider a set of feature vectors $\{\mathbf{x}_k \in \mathbb{R}^d \mid k = 1, \ldots, K\}$; the corresponding mean vector $\mathbf{m}$ and covariance matrix $\mathbf{C}$ are given by

$$\begin{aligned} \mathbf{m} &= \frac{1}{K}\sum_{k=1}^{K} \mathbf{x}_k, \\ \mathbf{C} &= \frac{1}{K}\sum_{k=1}^{K} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T. \end{aligned} \tag{6}$$

The eigendecomposition of $\mathbf{C}$ takes the form of

$$\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}^T, \tag{7}$$

where $\mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_d]$ is the orthogonal matrix whose columns are the eigenvectors of $\mathbf{C}$, and $\mathbf{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues.

So far, we have used the eigen decomposition to build a basis $\{\mathbf{v}_1 \cdots \mathbf{v}_d\}$ for the original feature space. Our goal, however, is to perform dimension reduction. The reduced dimensionality d' is determined by choosing the d' largest eigenvalues, so that

$$\frac{\sum_{j=1}^{d'} \lambda_j}{\sum_{i=1}^{d} \lambda_i} > \theta. \tag{8}$$

In our experiments, the threshold θ is set to 0.95 to preserve the most important information contained in the original data.

Then, we can construct a $d \times d'$ projection matrix $\mathbf{\Phi}$ where the ith column contains the eigenvector $\mathbf{v}_i$. The projection of the original feature vector $\mathbf{x}_k$ in the d'-dimensional subspace is given by

$$\mathbf{p}_k = \mathbf{\Phi}^T(\mathbf{x}_k - \mathbf{m}). \tag{9}$$

3.4. Mapping Features to Symbols. The image sequence of a human action has been converted to the sequence of feature vectors. To further simplify the analysis, we transform the extracted feature sequence into a symbol sequence. This is accomplished by constructing a codebook for vector quantization. In our system, we use the k-means clustering to create a codebook where each codeword is the mean of one cluster. Once a codebook is built, each codeword is treated as a symbol and thus each feature vector can be converted into a symbol by finding the nearest codeword. Up to this moment, a continuous human action has been discretized as a sequence of feature vectors, and each feature vector can then be represented by a symbol. Consequently, a human action composed of a sequence of symbols can be regarded as a string.

3.5. Action Recognition. We can now exploit the string representation to model different human behaviours. In Section 3.5.1, we will briefly introduce a popular approach for computing string distance. In Section 3.5.2, we will describe the inherent difficulty in training a classifier and present our solution to the problem. Finally, a Bayes classifier for multiclass classification problem is described in Section 3.5.3.

3.5.1. String-to-String Distance Measure. There are many existing approaches for measuring string distance. We have implemented three popular distance measures, namely, the edit distance [22], the Longest Common Subsequence (LCS) [23], and the histogram [8], in this study. The experimental results show that the histogram distance measure yields the highest recognition rate. A description about how to compute the distance between two strings using their histograms is given below.

Firstly, a string is converted into a histogram by counting the number of each symbol in the string (see Figure 4). The number of the bins in the histogram is determined by the codebook size. Furthermore, a histogram is normalized so that it sums to one. The normalized histogram $\overline{h}$ is given by

$$\overline{h}(i) = \frac{h(i)}{\sum_{i=1}^{Q} h(i)}, \tag{10}$$

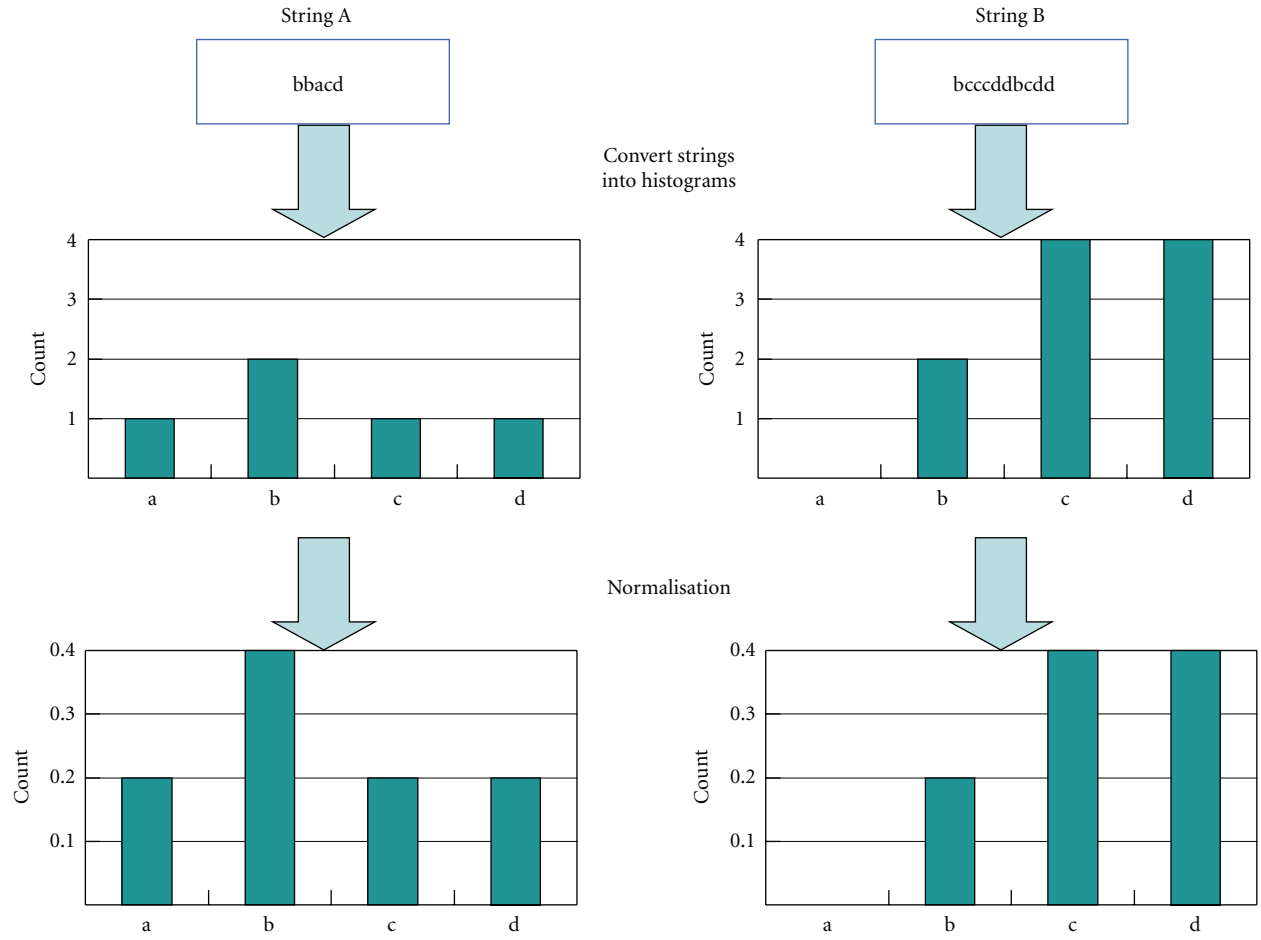


Figure 4: The illustration of converting strings to histograms.

where Q is the number of bins, and $h(i)$ denotes the ith bin of the original histogram. Figure 4 shows the steps of converting strings to normalized histograms. In Figure 4, two strings, "bbacd" and "bcccddbcdd", are initially converted into histograms and the results are shown in the 2nd row. These histograms are subsequently normalized (see the 3rd row of Figure 4).

Thus, the distance between two strings can be defined as the distance between the corresponding normalized histograms. The distance between two normalized histograms, $\overline{h}_a$ and $\overline{h}_b$, is calculated as follows:

$$d_{\text{Hitogram}}\left(\overline{h}_a, \overline{h}_b\right) = \sum_{i=1}^{Q} \left| \overline{h}_a(i) - \overline{h}_b(i) \right|. \tag{11}$$

3.5.2. Histogram-Based Vector Space. In this section, we will illustrate why dimension reduction is necessary for training a classifier and state the reason for choosing NWFE rather than conventional dimension reduction methods, such as PCA or Linear Discriminant Analysis (LDA). Then, a brief introduction to the NWFE will be provided.

The normalized histograms obtained in Section 3.5.1 are represented as vectors. As shown in Figure 5, we simply put all bin contents into one column vector. The resulting feature dimension is equal to the total number of bins in the normalized histogram. In other words, the dimension of the histogram vector is determined by the codebook size, that is, the number of clusters in k-means clustering. This would suggest an intuitive way to increase recognition performance, which is to use a larger codebook size when we convert a human action into a string. Unfortunately, it is observed in our experiments that, beyond a certain point, the increase in the codebook size leads to worse rather than better performance. The source of this difficulty can be traced to the fact that we have a fixed amount of training samples and thus the classifier parameters estimated in high dimensional feature space are not accurate. As a result, the obtained classifier may not be reliable. The use of dimension reduction techniques before training a classifier has been proposed in order to mitigate this problem [24, 25]. The dimension reduction will enable the parameter estimation to be more accurate for classification purpose.

The PCA is arguably the most frequently cited technique for dimension reduction in the literature. Recall that the PCA is intended to maximize the total scatter in a projection subspace despite the availability of category information. On the other hand, the LDA makes use of category information

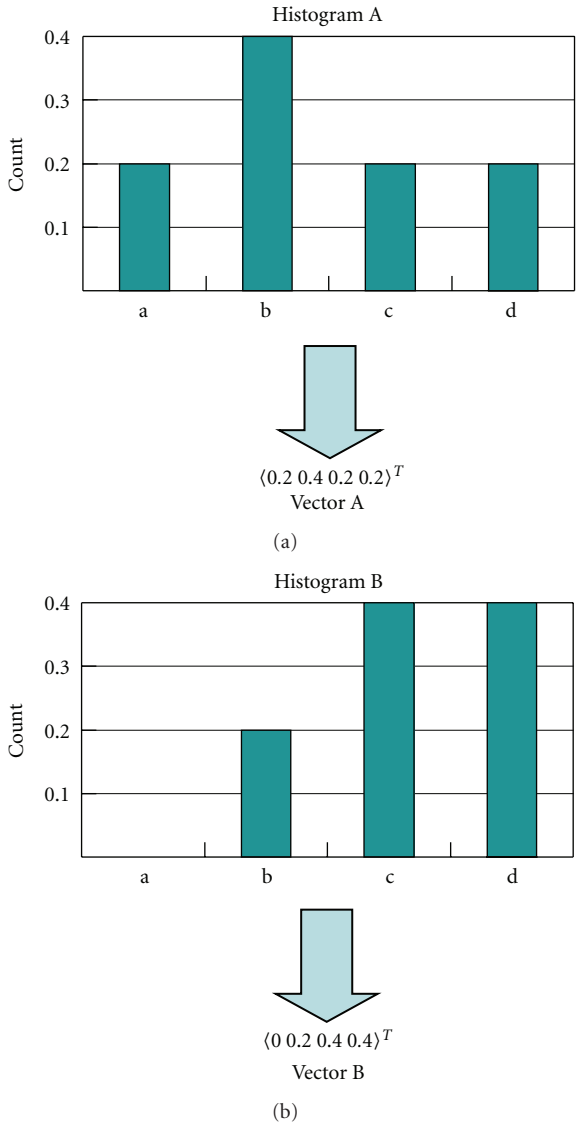


Figure 5: The illustration of representing histograms as vectors.

associated with training samples, to maximize the separablility between different categories in a projection subspace. Let us examine an example, which demonstrates how the PCA and LDA behave differently in dimension reduction. In Figure 6, the ellipses denote the probability distributions of two classes. We perform dimension reduction on the two-dimensional feature space and obtain one-dimensional subspaces, the PCA subspace and the LDA subspace. It is obvious that the class separability in the LDA subspace is better than that in the PCA subspace. Thus, the LDA generally gives better classification results than the PCA does.

However, the LDA has its own limitations which restrict its direct applicability to practical problems. Kuo and Landgrebe [10] argue that there are three disadvantages in LDA. One is that it usually does not work well if the class-conditional distributions are not Gaussian-like distributions. The second disadvantage is that the rank of the between-class scatter matrix is $N_c - 1$, where N_c is the number of

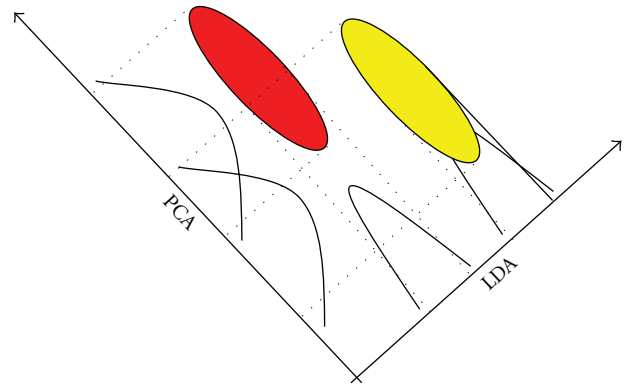


Figure 6: The projection of PCA and LDA.

classes. So, the dimension of an LDA subspace is at most $N_c - 1$. In real-world applications, the data distributions are often complicate and therefore the LDA will have poor classification performance in such a low-dimensional subspace. The third one is that the within-class scatter matrix usually becomes singular and thus the LDA fails.

The Nonparametric Weighted Feature Extraction has been introduced by Kuo and Landgrebe [10], which preserves the desirable characteristics of the LDA while avoiding its shortcoming. The main concept of the NWFE is to calculate the "weighted means" by assigning different weights on every sample. Let $\mathbf{x}_k^{(i)}$ denote the kth sample of class i. The weighted mean of $\mathbf{x}_k^{(i)}$ in the class j is defined by

$$\mathbf{m}_j\left(\mathbf{x}_k^{(i)}\right) = \sum_{\ell=1}^{n_j} w_{k\ell}^{(i,j)} \mathbf{x}_\ell^{(j)}, \tag{12}$$

where the weights on the samples in class j are given by

$$w_{k\ell}^{(i,j)} = \frac{\left\|\mathbf{x}_k^{(i)} - \mathbf{x}_\ell^{(j)}\right\|^{-1}}{\sum_{t=1}^{n_j}\left\|\mathbf{x}_k^{(i)} - \mathbf{x}_t^{(j)}\right\|^{-1}}. \tag{13}$$

Based on the weighted mean, the between-class scatter matrix is defined as

$$\mathbf{S}_b^{NW} = \sum_{i=1}^{N_c} P_i \sum_{\substack{j=1,\\ j\neq i}}^{N_c} \sum_{k=1}^{n_i} \frac{\lambda_k^{(i,j)}}{n_i} \left(\mathbf{x}_k^{(i)} - \mathbf{m}_j\left(\mathbf{x}_k^{(i)}\right)\right) \times \left(\mathbf{x}_k^{(i)} - \mathbf{m}_j\left(\mathbf{x}_k^{(i)}\right)\right)^T, \tag{14}$$

and the within-class scatter matrix is defined as

$$\mathbf{S}_w^{NW} = \sum_{i=1}^{N_c} P_i \sum_{k=1}^{n_i} \frac{\lambda_k^{(i,j)}}{n_i} \left(\mathbf{x}_k^{(i)} - \mathbf{m}_i\left(\mathbf{x}_k^{(i)}\right)\right)\left(\mathbf{x}_k^{(i)} - \mathbf{m}_i\left(\mathbf{x}_k^{(i)}\right)\right)^T, \tag{15}$$

where n_i is the size of class i, P_i is the prior probability for class i, and the scatter matrix weight $\lambda_k^{(i,j)}$ is given by

$$\lambda_k^{(i,j)} = \frac{\left\|\mathbf{x}_k^{(i)} - \mathbf{m}_j\left(\mathbf{x}_k^{(i)}\right)\right\|^{-1}}{\sum_{\ell=1}^{n_i}\left\|\mathbf{x}_\ell^{(i)} - \mathbf{m}_j\left(\mathbf{x}_\ell^{(i)}\right)\right\|^{-1}}. \tag{16}$$

Table 1: Action recognition using different features.

Features	Recognition rate (%)
distance signal feature (codesize = 173)	95.5
width feature (codesize = 122)	94.4
combined feature (codesize = 115)	100

Table 2: Action recognition using different distance measures.

Distance Measures	Recognition rate (%)
Edit distance (codesize = 54)	94.4
LCS (codesize = 64)	97.7
histogram (codesize = 158)	100

Table 3: Reported results on Weizmann dataset.

Methods	Recognition rate (%)
Scovanner et al. [30]	82.6
Ali et al. [29]	92.6
Niebles et al. [28]	95
Wang and Suter [32]	97.78
Blank et al. [26]	99.61
Weinland and Boyer [27]	100
Our method	**100**

Table 4: The classification times of the nearest-neighbour rule.

Distance Measures	Average classification time (msec.)
Edit Distance	66.9
LCS	63.6
Histogram	0.5

The feature vector in the d''-dimensional subspace is obtained by projecting the histogram vector to the subspace spanned by d'' eigenvectors, associated with the largest d'' eigenvalues of $(\mathbf{S}_w^{NW})^{-1}\mathbf{S}_b^{NW}$.

3.5.3. Classifier. After performing the NWFE on the histogram vectors, the remaining task is to construct a classification rule in the d''-dimensional feature space. In the spirit of starting simple, we attempt to utilize Bayes classifier with the conventional assumption of Gaussian distributions. More precisely, we model a class $\mathcal{C}_k$ with the class-conditional density $p(\mathbf{x}|\mathcal{C}_k)$. Because of the Gaussian assumption, the density for class $\mathcal{C}_k$ is

$$p(\mathbf{x} \mid \mathcal{C}_k) = \frac{1}{(2\pi)^{d''/2}} \frac{1}{|\mathbf{\Sigma}_k|^{1/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_k\right)^T \mathbf{\Sigma}_k^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_k\right)\right\} \tag{17}$$

and the maximum likelihood estimate of $\boldsymbol{\mu}_k$ is then given by

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} \mathbf{x}_i, \tag{18}$$

Table 5: The classification times of Bayes rule.

Input feature	Average recognition time (msec.)
Histogram Vector	0.5
Histogram Vector + NWFE	0.1

Table 6: Complexity analysis of recognition time.

Methods	Complexity
Edit distance	$T_{\text{recognition}} = T_{\text{pca}} + T_{\text{mapping}} + T_{\text{EdDistance}} = T_{\text{pca}} + T_{\text{mapping}} + S \times \mathcal{O}(F^2)$
LCS	$T_{\text{recognition}} = T_{\text{pca}} + T_{\text{mapping}} + T_{\text{LcsDistance}} = T_{\text{pca}} + T_{\text{mapping}} + S \times \mathcal{O}(F^2)$
Histogram	$T_{\text{recognition}} = T_{\text{pca}} + T_{\text{mapping}} + T_{\text{ConvertToHistogram}} + T_{\text{HistogramDistance}} = T_{\text{pca}} + T_{\text{mapping}} + O(F) + S \times \mathcal{O}(N)$
Our method	$T_{\text{recognition}} = T_{\text{pca}} + T_{\text{mapping}} + T_{\text{ConvertToHistogram}} + T_{\text{NWFE}} + T_{\text{Bayes}} = T_{\text{pca}} + T_{\text{mapping}} + O(F) + M \times \mathcal{O}(N) + \mathcal{O}(CM^2)$

F: Average number of frames in input as well as database sequence.
N: Total number of clusters (same as the dimensionality of histogram).
S: Total number of sequences in the database.
M: Dimensionality of the NWFE subspace.
C: Number of classes.

where $\{\mathbf{x}_i\}_{i=1}^{n_k}$ denotes the training samples in class $\mathcal{C}_k$. Notice that the maximum likelihood estimate of $\mathbf{\Sigma}_k$ would be singular if the number of training samples in class $\mathcal{C}_k$ was smaller than the feature dimension d''. Hence, the maximum likelihood estimate of $\mathbf{\Sigma}_k$ takes the form

$$\hat{\mathbf{\Sigma}}_k = \frac{1}{n_k}\sum_{i=1}^{n_k}\left(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k\right)\left(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k\right)^T + \epsilon\mathbf{I}, \tag{19}$$

where $\mathbf{I}$ is an identity matrix and ϵ is the regularization coefficient that prevents singularity.

After obtaining the class-conditional densities $p(\mathbf{x} \mid \mathcal{C}_k)$, we can compute the posterior probability $p(\mathcal{C}_k \mid \mathbf{x})$ using Bayes' theorem

$$p(\mathcal{C}_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}, \tag{20}$$

where $p(\mathcal{C}_k)$ denotes the prior probability for class $\mathcal{C}_k$. Thus, we have the following classification rule:

$$\text{Decide } \mathcal{C}_k \text{ if } p(\mathcal{C}_k \mid \mathbf{x}) > p(\mathcal{C}_\ell \mid \mathbf{x}), \quad \forall \ell \neq k. \tag{21}$$

If we had equal prior probabilities, then the prior probabilities would certainly provide no information about the category of the input $\mathbf{x}$. In other words, the classification would hinge entirely on the class-conditional densities. By eliminating the prior probability from (20), we obtain the following simplified classification rule:

$$\text{Decide } \mathcal{C}_k \text{ if } p(\mathbf{x} \mid \mathcal{C}_k) > p(\mathbf{x} \mid \mathcal{C}_\ell), \quad \forall \ell \neq k. \tag{22}$$

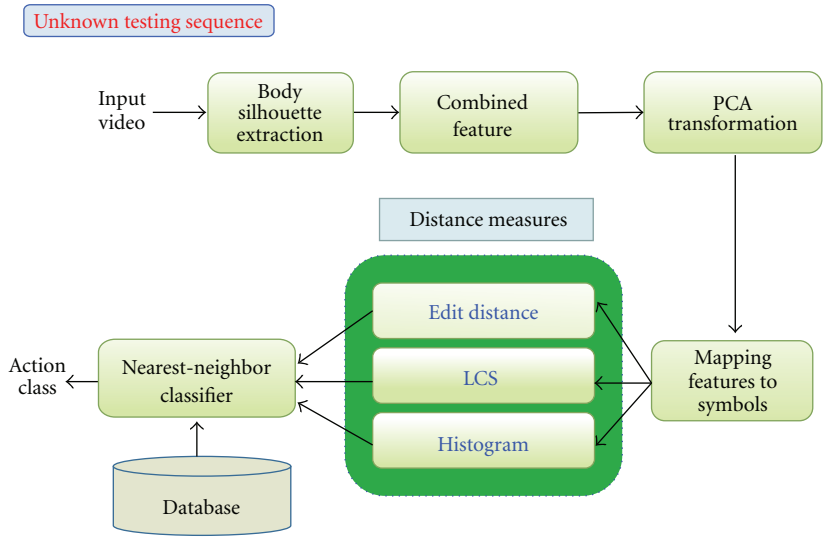


Figure 7: The flowchart for the 3 compared distance measures.

4. Experimental Results

In this section, we conduct a series of experiments to evaluate the performance of the proposed system. In particular, we investigate the influence of using different features and string distance measures. After finding the best configuration, we compare the recognition accuracy and computation time of the proposed system with those of existing approaches.

Experiments are firstly conducted on Weizmann dataset [26], which is a popular benchmark dataset in the literature [27–33]. It contains 10 actions performed by 9 persons, selected frames from the dataset are shown in Figure 8. These 10 actions are bend, walking (walk), running (run), jumping-jack (jack), jumping forward on one leg (skip), jumping forward on two legs (jump), jumping in place on two legs (pjump), jumping sideways (side), waving on hand (wave1), waving two hands (wave2).

Here, we adopt leave-one-out scheme for computing all recognition rates. That is, use 8 out of the 9 persons in the dataset to train the classifier and the 9th person is utilized in the test phase. Repeat this procedure for all 9 persons and the resulting recognition rates are then averaged.

Experimental results on another publicly available dataset [17] are provided to illustrate the accuracy, efficiency and generality of the proposed method.

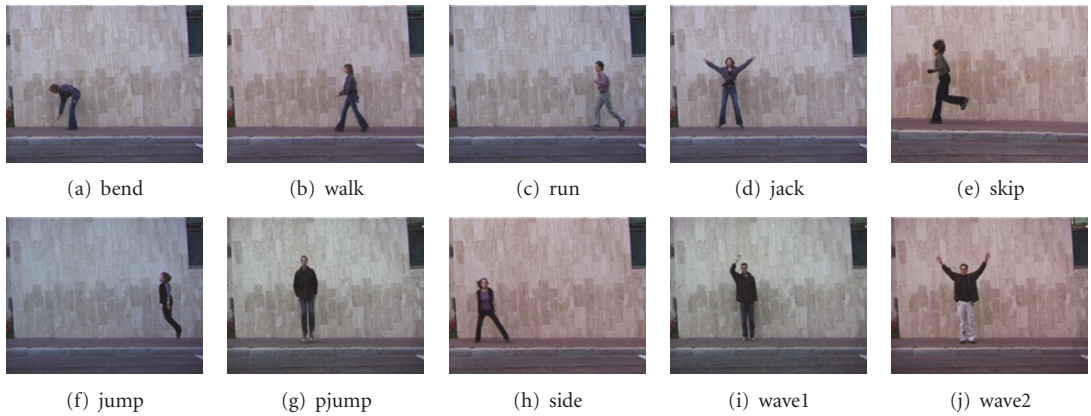
4.1. Comparison of Different Features. In any pattern recognition problems, the choice of discriminative features is a crucial step and depends largely on the problem domain. Hence, we have conducted experiments to evaluate the discrimination capability of three different features, namely, the distance signal feature [8], the width feature [9], and the combined feature. The flowchart of the experimental setup is shown in Figure 9. The three features utilized in this experiment are highlighted by the dotted block.

The confusion matrices of the distance signal feature and the width feature are shown in Figures 10(a) and 10(b), respectively. The results indicate that these two features are complementary. For instance, the action "Pjump" tends to be misclassified by the width feature, but not by the other. Hence, we can achieve a synergetic effect by properly combining these two features. Table 1 summarizes the recognition rates of using the distance signal, the width feature, and the combined feature. Clearly, the combination feature performs better than the individual features.

Table 7: The results of action recognition in Academia Sinica Dataset.

	C01	C02	C03	C04	C05	C06	C07	C08	C09	C10
C01	5	0	0	0	0	0	0	0	0	0
C02	0	5	0	0	0	0	0	0	0	0
C03	0	0	5	0	0	0	0	0	0	0
C04	0	0	0	5	0	0	0	0	0	0
C05	0	0	0	0	5	0	0	0	0	0
C06	0	0	0	0	0	5	0	0	0	0
C07	0	0	0	0	0	0	5	0	0	0
C08	0	0	0	0	0	0	0	5	0	0
C09	0	0	0	0	0	0	0	0	5	0
C10	0	0	0	0	0	0	0	0	0	5

4.2. Comparison of Different Distance Measures. As mentioned in Section 3.5, a continuous human action can be treated as a string. By using the string representation, the problem of action recognition is reduced to a string matching problem. That is, the procedure of recognition is to match the string representing an unknown action to the strings representing training samples. There are many existing approaches for computing string distances. Here, we evaluate the effect of using different string-to-string distance measures, namely, the edit distance [22], the LCS [23], and



(a) bend (b) walk (c) run (d) jack (e) skip

(f) jump (g) pjump (h) side (i) wave1 (j) wave2

Figure 8: Selected frames from Weizmann dataset.

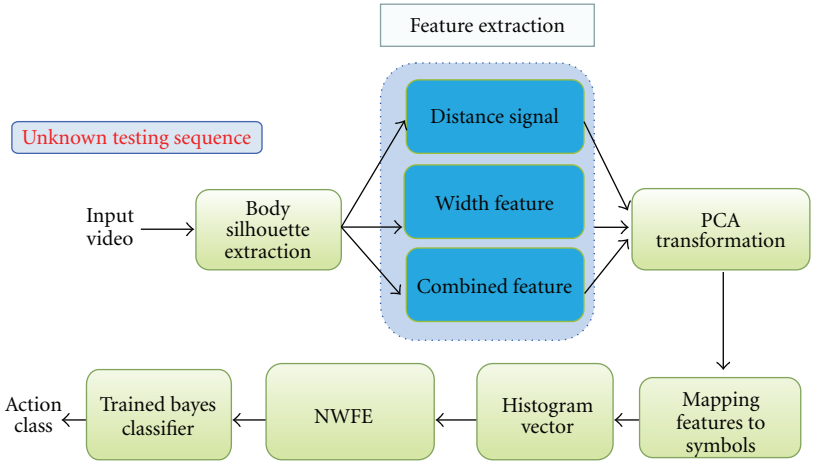


Figure 9: The flowchart for the comparison of three feature extraction methods.

the histogram [8]. The flowchart of the experimental setup is shown in Figure 7, where distance measures are highlighted by the dotted block.

Table 2 summarizes the results of using different string distance metrics. We observe that the histogram [8] achieves the best recognition rate. The primary advantage of using the histogram lies in the fact that it can remove time related information, such as speed or length of an action sequence. On the contrary, the edit distance is not robust to temporal variations and exhibits the worst recognition performance.

4.3. Dimensionality Reduction of Histogram Vectors. To visualize the effect of different dimension reduction methods, we first project training samples onto a two-dimensional subspace. The training samples projected by PCA are shown in Figure 11(a). In the PCA subspace, the data points belonging to different action categories are mixed together. As a result, PCA is not a suitable dimension reduction method for classification problems. On the other hand, the training samples projected by LDA and NWFE are shown in Figures 11(b) and 11(c), respectively. As opposed to the PCA projection, class separability is well maintained by either LDA or NWFE. From the experimental results in Figure 11, we may conclude that LDA and NWFE are more appropriate dimension reduction techniques for classification problems. Nevertheless, the dimension of the LDA subspace is at most $N_c - 1$ (see Section 3.5). Since Weizmann data set contains 10 action categories, the dimension of the corresponding LDA subspace is at most 9. It is difficult to achieve high recognition accuracy using such a low-dimensional subspace. Because the NWFE does not have such a limitation, we thus choose it as the dimension reduction method in the proposed system.

Next, we construct a Bayes classifier from the histogram vector space (see Figure 12, the upper path in the dotted block). The resulting recognition rates versus codebook size are shown as the dashed line in Figure 13(b). Here, we observe a significant performance drop when the codebook size is large. Without NWFE, the Bayes classifier is essentially performing random guess when the codebook size is larger than 60. The observed failures indicate a challenging problem in classifier design.

It seems that the high dimensionality of feature space, that is, large codebook size, should increase the accuracy

	1. Bend	2. Jack	3. Jump	4. Pjump	5. Run	6. Side	7. Skip	8. Walk	9. Wave1	10. Wave2
1. Bend	9	0	0	0	0	0	0	0	0	0
2. Jack	0	9	0	0	0	0	0	0	0	0
3. Jump	0	0	9	0	0	0	0	0	0	0
4. Pjump	0	0	0	9	0	0	0	0	0	0
5. Run	0	0	0	0	7	1	0	1	0	0
6. Side	0	0	0	0	0	9	0	0	0	0
7. Skip	1	0	1	0	0	0	7	0	0	0
8. Walk	0	0	0	0	1	1	0	7	0	0
9. Wave1	0	0	0	0	0	0	0	0	9	0
10. Wave2	0	0	0	0	0	0	0	0	0	9

(a)

	1. Bend	2. Jack	3. Jump	4. Pjump	5. Run	6. Side	7. Skip	8. Walk	9. Wave1	10. Wave2
1. Bend	7	2	0	0	0	0	0	0	0	0
2. Jack	0	9	0	0	0	0	0	0	0	0
3. Jump	0	0	9	0	0	0	0	0	0	0
4. Pjump	0	2	1	5	0	0	0	0	1	0
5. Run	0	0	0	0	9	0	0	0	0	0
6. Side	0	0	0	0	0	9	0	0	0	0
7. Skip	0	0	0	0	0	0	9	0	0	0
8. Walk	0	0	0	0	0	0	0	9	0	0
9. Wave1	0	0	0	0	0	0	0	0	7	2
10. Wave2	0	0	0	0	0	0	0	0	1	8

(b)

Figure 10: (a) Confusion matrix of the distance signal feature using 128 codewords. (b) Confusion matrix of the width feature using 128 codewords.

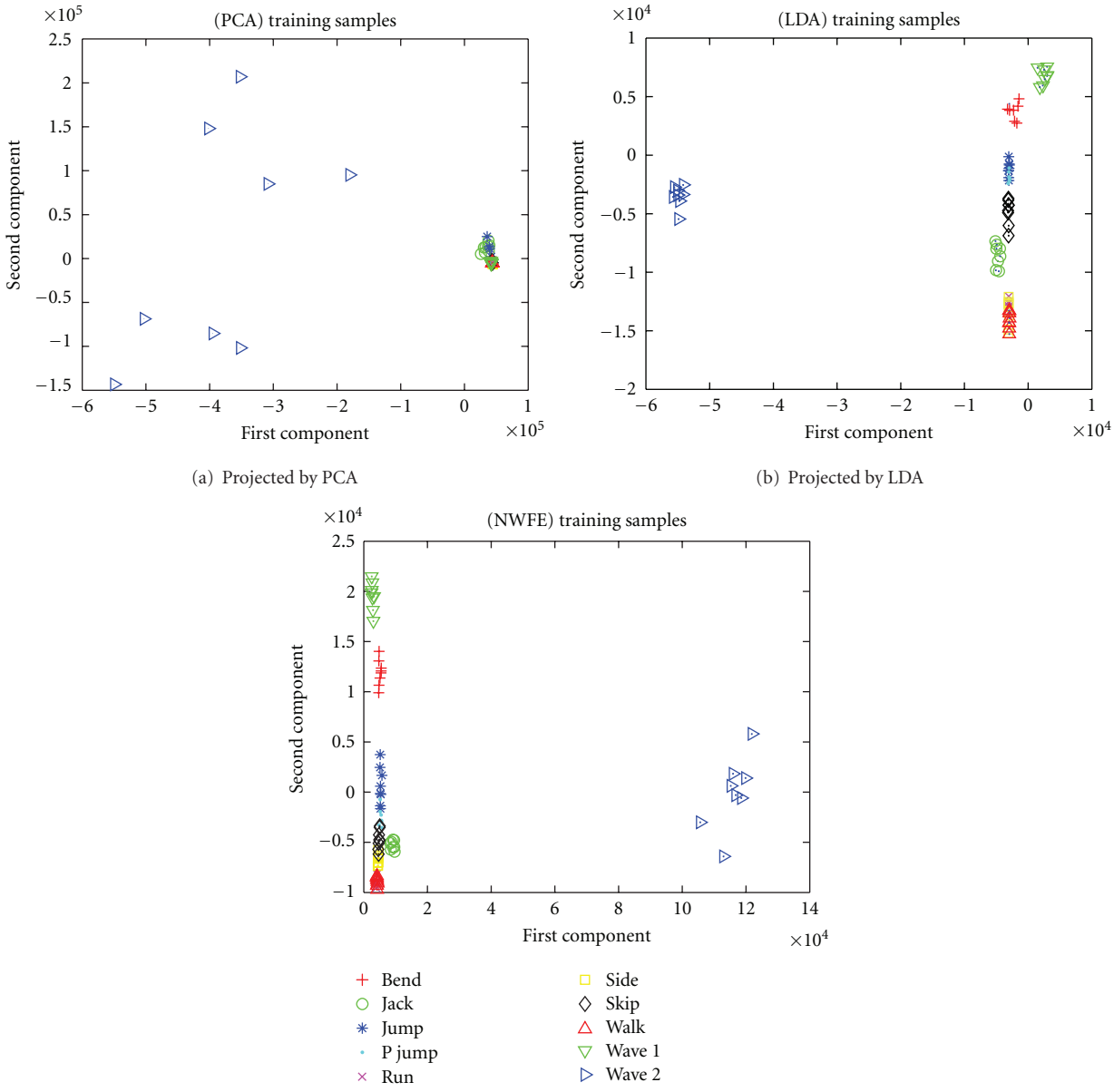
in classifying human actions. In Figure 13(b), the dashed line first ascends and then descends rapidly as the size of codebook increases. In other words, problems will arise if too many codewords are specified in the codebook with limited training samples. It has been shown that the amount of training data required for a Gaussian distribution-based Bayes classifier increases quadratically with respect to the increase of feature dimension [34]. If we do not have sufficient training samples, the parameter estimation for the Bayes classifier becomes inaccurate and unreliable. Thus, the classification accuracy declines as we have observed in Figure 13(b).

Hughes [35] demonstrates the behavior of classification accuracies with respect to the number of training samples and feature dimension, the so-called Hughes phenomenon. As suggested by Hughes phenomenon, the number of training samples required for training a classifier should increase as the dimensionality increases. However, we are usually confronted with fixed number of training samples in practice. Hence, it is necessary to develop another method to deal with high-dimensional data. An effective way is to reduce the dimensionality of feature space. In the proposed system, this is done by performing the Nonparametric Weighted Feature Extraction [10] that can extract the most discriminative features from the histogram vectors (see Figure 12, the lower path in the dotted block). The resulting performance of action recognition is shown as the solid line in Figure 13(b). It is apparent that the recognition accuracy increases stably with respect to the increase of codebook size. The proposed system can achieve the recognition rate of 100% when the codebook size is 115 and the corresponding confusion matrix is shown in Figure 13(a).

There are many recent efforts to the problem of human action recognition. Some recently published results on Weizmann dataset are summarized in Table 3.

4.4. Comparison of Classification Time. In this section, the aim is to compare the computational times of different classification methods. Basically, we have implemented two types of classifiers, namely, the nearest-neighbour and Bayes classifier. These classification methods are implemented using Visual Studio C++ 2008 and evaluated on an Intel Core2 CPU-6320 1.86 GHz machine with 2 GB RAM. Because all these methods take strings as the input, we only consider the time required for classifying the input strings.

For the nearest-neighbour classifier, we utilize three different distance measures. The resulting classification times are presented in Table 4. Evidently, the histogram-based



(a) Projected by PCA

(b) Projected by LDA

(c) Projected by NWFE

Figure 11: Training samples are projected onto a two-dimensional subspace by different methods.

distance measure [8] has a significantly lower computational cost than the other distance measures. For the Bayes classification, we perform classifier training in two different feature spaces, that is, the histogram vector space and the subspace obtained by the NWFE. The resulting classification times are summarized in Table 5. Since the NWFE reduces the dimensionality of the histogram vectors, the average recognition time of Bayes classifier is thus reduced from 0.5 to 0.1 msec. More importantly, the reduction in computational complexity is achieved without sacrificing classification accuracy. In addition, Table 6 shows the complexity analysis of recognition time. It is worth noting that the time complexity of the proposed method is not affected by the size of dataset.

4.5. Results on Academia Sinica Dataset. In this section we describe the other experiment on the Academia Sinica Dataset [17]. This dataset consists of 10 action categories from ten different people, selected frames of 10 action categories and then sequentially named C01–C10 as shown in Figure 14. In this dataset, each person performs each action five times. Therefore, the dataset includes 500 action sequences.

First we select the sequences of 1st human person as the test sequences to evaluate the performance of our method. Accordingly, the train sequences contained rest action sequences by the 2nd–10th person. Despite these train sequences, a Bayes classifier can be trained to perform action recognition. The experimental result is given in Table 7.

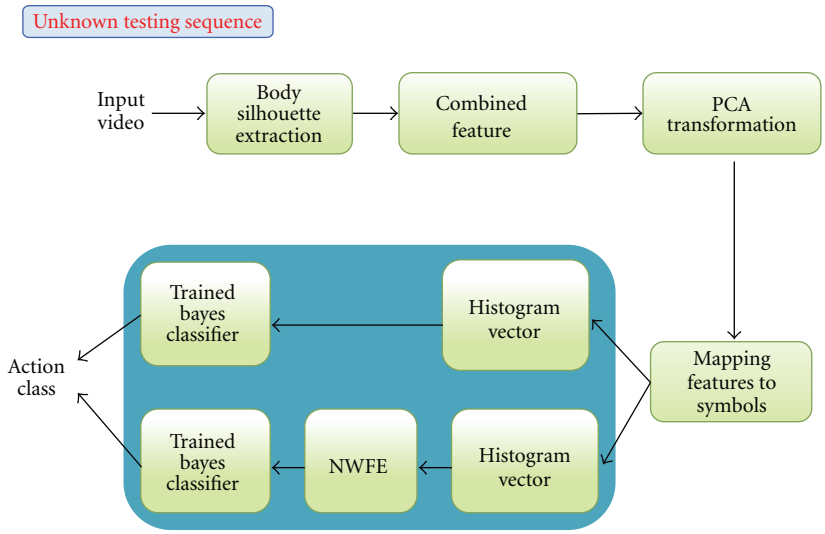


Figure 12: The flowchart of the proposed system, with and without performing NWFE.

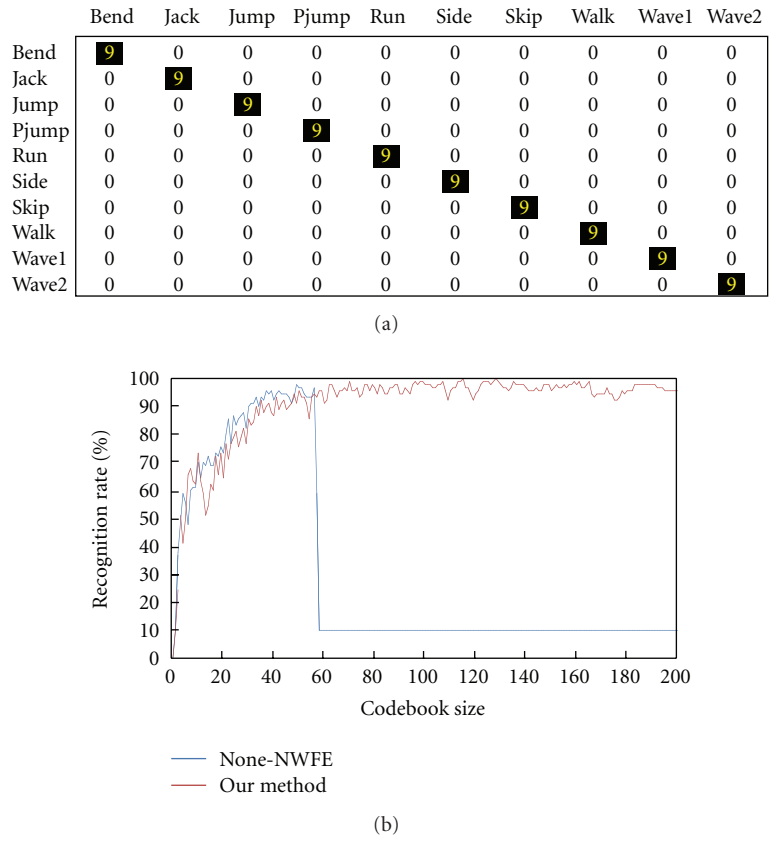
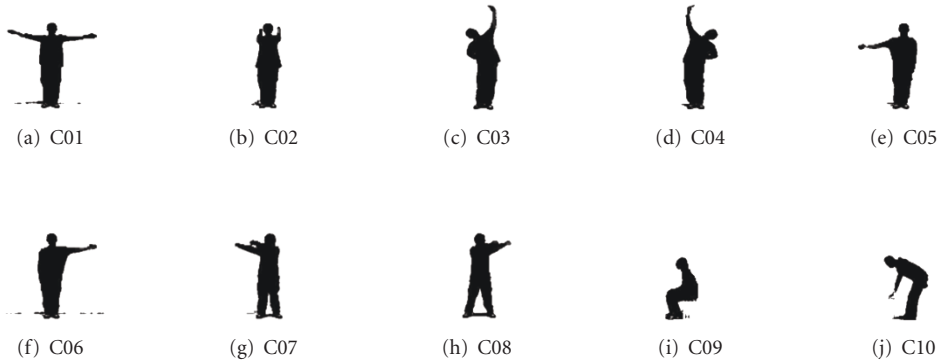


Figure 13: (a) Confusion matrix using 115 codewords. (b) Recognition rates versus codebook size.

Overall, the results have been very positive. We also use the leave-one-out scheme to further verify the recognition results. In this case, use total action sequences of ten different people to compute all recognition rates. Namely, take 49 out of the 50 subjects of each category to train the classifier. Next, the 50th subject is utilized to be test sequence. The results collected all action categories by using this procedure as shown in Figure 15. The results show the method can achieve a high recognition in line with previous experiment. In order to observe the system performance, the detailed



(a) C01 (b) C02 (c) C03 (d) C04 (e) C05

(f) C06 (g) C07 (h) C08 (i) C09 (j) C10

Figure 14: The selected frames from ten categories of the new public database.

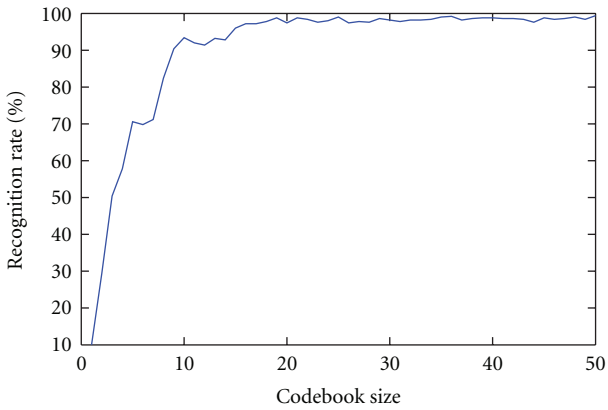


Figure 15: Recognition rates of Academia Sinica Dataset.

results as shown in Figure 15. In terms of the relationships among recognition rates and codebook sizes, the results are what is expected.

5. Conclusions

The system described in this paper proposes a method to recognize human action. The system first performs a combined feature, which integrates the signal distance feature and the width feature extracted from a human pose silhouette. From the experimental results, we observe that the combined feature is more discriminative than individual features for human action recognition. The system is efficient because we have employed PCA to reduce the dimensionality of feature vectors and the k-means algorithm can be applied to construct a codebook. Therefore, we do not need to select key pose manually for codebook formation. Besides, We utilize Bayes classifier to label NWFE-based histogram vectors. This scheme is computationally faster than the nearest-neighbour classifiers. The experimental results demonstrated the accuracy of the system, but also, showed that it equals or outperforms a state-of-the-art system.

Acknowledgment

The authors have been partially supported by the National Science Council, Taiwan (Grant no. 98-2221-E-194-039-MY3).

References

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 34, no. 3, pp. 334–352, 2004.

[2] N. M. Oliver, B. Rosario, and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[3] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852–872, 2000.

[4] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," in *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, pp. 296–301, 1995.

[5] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[6] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379–385, 1992.

[7] A. F. Bobick, S. S. Intille, J. W. Davis et al., "The kidsRoom: a perceptually-based interactive and immersive story environment," *Presence*, vol. 8, no. 4, pp. 369–393, 1999.

[8] Y. Dedeoğlu, B. U. Töreyin, U. Güdükbay, and A. E. Çetin, "Silhouette-based method for object classification and human action recognition in video," in *Proceedings of the European Conference on Computer Vision in Human Computer Interaction*, vol. 3979 of *Lecture Notes in Computer Science*, pp. 64–77, Springer, 2006.

[9] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian, "Towards fast, view-invariant human action recognition," in *Proceedings of IEEE Computer Society Conference on Computer*

Vision and Pattern Recognition Workshops (CVPR '08), pp. 1–8, June 2008.

[10] B.-C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 5, pp. 1096–1105, 2004.

[11] J. K. Aggarwal and Q. Cai, "Human motion analysis: a review," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, 1999.

[12] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[13] A. Mokhber, C. Achard, and M. Milgram, "Recognition of human behavior by space-time silhouette characterization," *Pattern Recognition Letters*, vol. 29, no. 1, pp. 81–89, 2008.

[14] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[15] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 249–257, 2006.

[16] J.-W. Hsieh and Y.-T. Hsu, "Boosted string representation and its application to video surveillance," *Pattern Recognition*, vol. 41, no. 10, pp. 3078–3091, 2008.

[17] Y.-M. Liang, S.-W. Shih, A. C.-C. Shih, H.-Y. M. Liao, and C. C. Lin, "Learning atomic human actions using variable-length markov models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 1, pp. 268–280, 2009.

[18] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, October 2007.

[19] F. Huang and G. Xu, "Action recognition unrestricted by location and viewpoint variation," in *Proceedings of the 8th IEEE International Conference on Computer and Information Technology Workshops (CIT '08)*, pp. 433–438, July 2008.

[20] H.-S. Chen, H.-T. Chen, Y.-W. Chen, and S.-Y. Lee, "Human action recognition using star skeleton," in *Proceedings of the ACM Conference on Video Surveillance and Sensor Networks*, pp. 171–178, 2006.

[21] J. Zhou and J. Hoang, "Real time robust human detection and tracking system," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pp. 149–149, 2005.

[22] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2001.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 2001.

[24] B.-C. Kuo and D. A. Landgrebe, "Hyperspectral data classification using nonparametric weighted feature extraction," in *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS '02)*, vol. 3, pp. 1428–1430, June 2002.

[25] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804–3814, 2008.

[26] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proceedings of IEEE International Conference on Computer Vision*, vol. 2, pp. 1395–1402, 2005.

[27] D. Weinland and E. Boyer, "Action recognition using exemplar-based embedding," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR '08)*, pp. 1–7, June 2008.

[28] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[29] S. Ali, A. Basharat, and M. Shah, "Chaotic invariants for human action recognition," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, pp. 1–8, October 2007.

[30] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th ACM International Conference on Multimedia (MM '07)*, pp. 357–360, September 2007.

[31] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, October 2007.

[32] L. Wang and D. Suter, "Recognizing human activities from silhouettes: motion subspace and factorial discriminative graphical model," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, June 2007.

[33] J. C. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, June 2007.

[34] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.

[35] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 1968.