CHAPTER 4

Multivariate analysis based learning

4.1. Motivation for the approach

In this chapter it will be explained how from the original primitive features introduced in the previous chapter can be created new and more effective for gesture recognition features by utilizing a multivariate analysis based learning process. The learning process projects the original features from feature space $\boldsymbol{F}$ into a new *discriminant feature space* $\boldsymbol{D}$ which is better suited for discrimination between the different classes of gestures. Although it is possible to attempt classification directly in $\boldsymbol{F}$, this is usually not preferable for the following reasons:

(a) the dimensionality of feature space (given by expression (3-9) for the case of the features introduced in the previous chapter) can easily become too high, leading to analytical intractability and other computational problems because of the "curse of dimensionality", i.e. it is necessary to utilize some technique to reduce dimensionality;

(b) the greater the number of the features which are used, the greater becomes the possibility that some of them are mutually correlated. Because of this, increasing the number of the features with the hope that more information would lead to better recognition does not necessarily lead to increased recognition rates;

(c) when only a limited number of training samples are available, increasing the dimension of F leads to higher classification errors, the so-called Hughes phenomenon (Hughes, 1968);

(d) from the point of view and for the purposes of pattern classification, the input samples usually are not optimally distributed in feature space. It is possible, however, by using certain statistical criteria, to project the input distributions into a new space where much simpler decision functions would lead to better and more reliable pattern classification.

To address the above problems, many different methods have been developed throughout the years following the pioneering work of scientists like F. Galton, K. Pearson, R. A. Fisher, J. Wishart, H. Hotelling, etc., forming the branch of modern statistics known as multivariate analysis (Ito, 1969; Kawaguchi, 1973; Anderson, 1984; Fukunaga, 1990; Ishii et al., 1998). Multivariate analysis methods have been successfully applied to various problems, first in biology, psychology, agriculture, and more recently to meteorology, operations research, engineering, etc. For the problem of gesture recognition, which is of primary interest for us here, we have found that multivariate analysis based methods (and especially the linear discriminant analysis method described in the next section) provide very satisfactory solutions in terms of good recognition rates (good generalization abilities), simplicity and speed of processing, allowing real-time performance on general-purpose computers.

The rest of this chapter is organized in the following order. First, the method employed in the current implementation for projection of primitive feature space to discriminant feature space is described in section 4.2. The criterion for a classification decision is explained in section 4.3, and in section 4.4 it is shown how a better and more reliable

gesture recognition can be achieved by using an ensemble of classifiers operating at different resolution scales.

4.2. Projection of primitive feature space to discriminant feature space

In the *learning* stage (Fig. 1-1) described in this section, from the original primitive features obtained in the *feature extraction* stage (described in chapter 3), new and more effective features are created on the basis of multivariate analysis using linear discriminant analysis (LDA) (Fisher, 1936; Duda and Hart, 1973; Otsu, 1981), and input data is projected from feature space $\boldsymbol{F}$ into discriminant feature space $\boldsymbol{D}$ which is of significantly lower dimension and much better suited for the discrimination between the different classes of gestures. For example, if the primitive features (3-7) are represented by

$$x = (x_1,...,x_N)^T, \tag{4-1}$$

where N is the primitive feature space's dimension given by (3-9), the new features

$$y = (y_1,...,y_C)^T \tag{4-2}$$

can be obtained as linear combinations of features x with weights $A = [a_{ij}]$ and constants $b = (b_1,...,b_C)^T$:

$$y = A^T x + b. \tag{4-3}$$

Then the optimal parameters are determined so as to optimize a criterion function which is set to evaluate the performance of the linear model for the task at hand from the

learning samples. If we have K classes of gestures $\{\omega_k\}_{k=1}^{K}$, then the *within-class* and the *between-class* covariance matrices of the primitive features are computed from the training samples as:

$$S_W = \sum_{k=1}^{K} P(\omega_k) S_k , \tag{4-4}$$

$$S_B = \sum_{k=1}^{K} P(\omega_k)(\bar{x}_k - \bar{x}_T)(\bar{x}_k - \bar{x}_T)^T , \tag{4-5}$$

where $P(\omega_k)$, $\bar{x}_k$ and $\bar{x}_T$ denote the *a priori* probability of class ω_k (set here to be equal to *1/K*), the mean vector of class ω_k and the total mean vector, respectively, and S_k is the covariance matrix of class ω_k, defined as

$$S_k = \frac{1}{n_k} \sum_{x \in \omega_k} (x - \bar{x}_k)(x - \bar{x}_k)^T , \tag{4-6}$$

where n_k is the number of the samples for class ω_k. The within-class and between-class covariance matrices for the new features y can be calculated from (4-3) – (4-5) as

$$\hat{S}_W = A^T S_W A , \tag{4-7}$$

$$\hat{S}_B = A^T S_B A . \tag{4-8}$$

The following discriminant criterion

$$J(A) = tr(\hat{S}_W^{-1} \hat{S}_B) = tr\{(A^T S_W A)^{-1}(A^T S_B A)\} \tag{4-9}$$

is used to evaluate the performance of the discrimination of the new features y, and is maximized in order to obtain the optimal coefficient matrix A, i.e. A is obtained by solving the following generalized eigen-value problem, obtained after taking the derivative of (4-9) with respect to A and equating it to zero

$$S_B A = S_W A\Lambda \qquad (A^T S_W A = I), \tag{4-10}$$

where $\Lambda = diag(\lambda_1,...,\lambda_L)$ is the Lagrange multipliers matrix (for the constraint $\hat{S}_W = I$) having the eigenvalues λ_i on its principal diagonal, and I denotes the unit matrix. The j-th column of A is the eigenvector corresponding to the j-th largest eigenvalue. Thus, the C new features y_j are evaluated in their importance for discrimination by the eigenvalues. The maximum number C is bounded by min(K-1, M). Also, b in (4-3) is determined so that the total mean is mapped to the origin of y, i.e.

$$b = -A^T \bar{x}_T. \tag{4-11}$$

As the feature functions (3-7) are projected by LDA in the lower dimensional discriminant feature space $\boldsymbol{D}$, the trajectories $\Phi(r, s, t)$ in primitive feature space $\boldsymbol{F}$ will be projected to corresponding trajectories $\Psi(r, s, t)$ in $\boldsymbol{D}$, where for each gesture class, a class representative trajectory can be computed from the trajectories of each training sample. Thus, recognition of a given test sample can be performed in $\boldsymbol{D}$ by utilizing a similarity measure, defined, for instance, by the distance of the test sample's trajectory to each of the class-representative trajectories and classifying the test sample to the class for which the similarity measure is maximal (i.e. the corresponding distance minimal). Fig.4-1 shows the actual trajectories obtained for 4 classes of gestures, together with the trajectories of test samples for each class. The minimal distance be-

tween a test sample's trajectory and the learnt class average trajectories is obtained in the way explained in the following section.

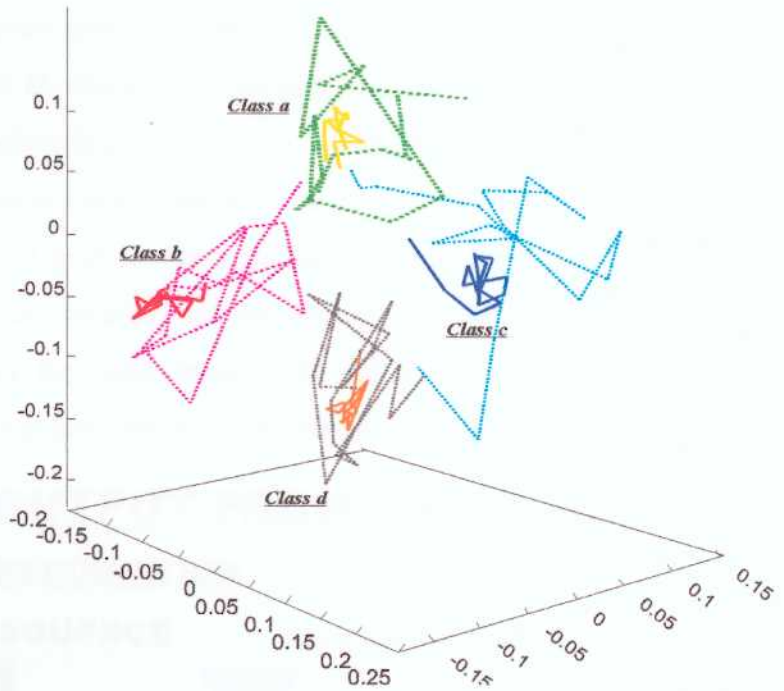

Figure 4-1. Gesture trajectories in discriminant feature space for 4 classes of gestures. The class representative trajectories for each class are displayed by solid lines, while the trajectories of the corresponding test samples (which have not been used during the training process) are shown in dashed lines

4.3. Output from the classifier

As it has been already explained in the last two chapters, each three consecutive image frames (an ordered image triplet) from the input image sequence are first projected by the feature extraction process to a point in feature space $\boldsymbol{F}$, and after that to a point in discriminant feature space $\boldsymbol{D}$ by LDA. During the learning process, from the training samples s for each gesture class r (which depict trajectories $\Psi(r, s, t)$ in $\boldsymbol{D}$), class representative trajectories $\Psi(r, t)$ can be constructed in $\boldsymbol{D}$ by averaging the sam-

ple trajectories for each class. The class representative trajectories for several different gesture classes (A, B, … , Z) are symbolically represented as strings of different color in Fig. 4-2. The index after the class name indicates the successive "frame number" in the class representative sequence of points from the trajectory in ***D***. In a similar manner, when an unknown gesture sequence is input to the system, it will be mapped to a trajectory $X(t)$ in ***D***, which is symbolically represented in Fig. 4-2 as a string in gray color and yet unclassified "frames" X. Although the length of the class representative strings of frames is known (shown to be 20 frames for all classes, but it is not necessary that all classes of gestures have the same length) the length and contents of the string for the test gesture sequence are not known, because there is no information yet how to segment/classify the input stream. The following approach for online recognition is proposed here and used for the experiments in chapter 6.

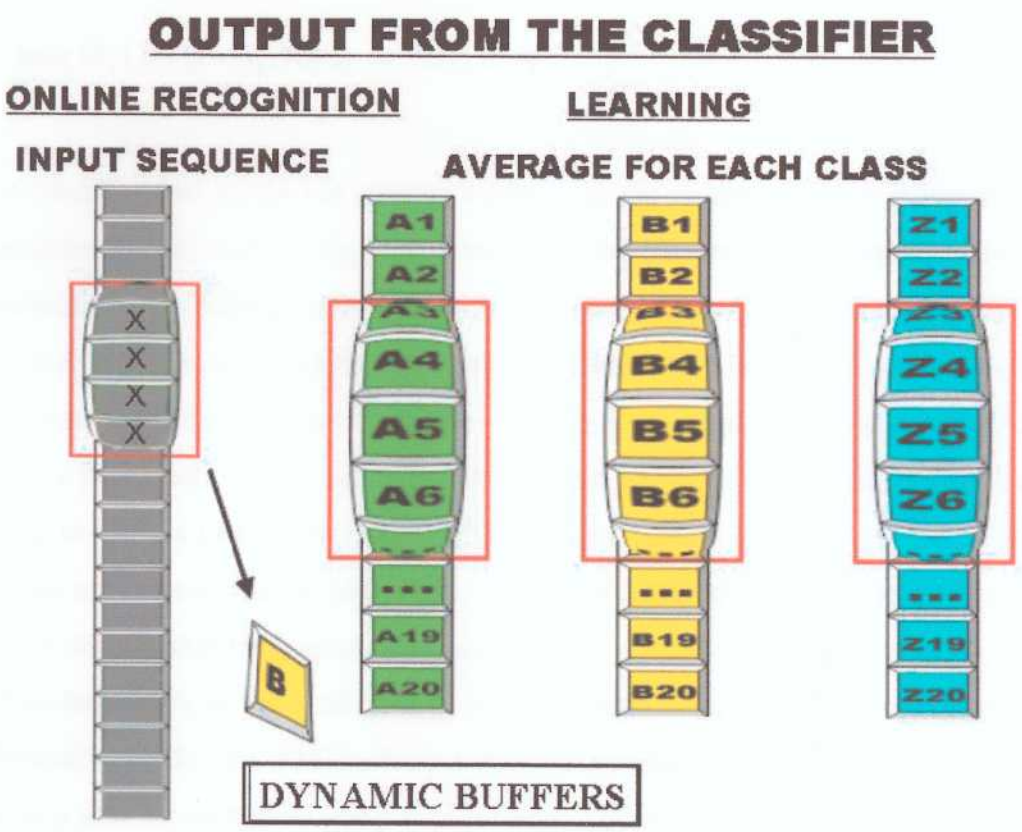

Figure 4-2. Output from the classifier (see the text for details)

A "time window" W of k consecutive points (shown as a red rectangle in Fig. 4-2) from the test gesture string is successively compared to each k consecutive points from the class-average trajectories for each learnt gesture class. The comparison is based on the following similarity measure

$$S_r(\mathrm{X}(t)) = \sum_{t \in W} \left\| \mathrm{X}(t) - \Psi(r,t) \right\|^2 \tag{4-12}$$

i.e. the Euclidean distance between the corresponding points inside the windows W (which are sequentially shifted over each class's representative strings) are calculated, and the decision for classification is based on the following rule

$$\max_{r=1,\cdots,K} \{S_r(\mathrm{X}(t)\} = S_k(\mathrm{X}(t)) \Rightarrow \mathrm{X}(t) \in \omega_k . \tag{4-13}$$

The classification of the test gesture's frames in the current window is carried out by evaluating (4-12) and (4-13), after which the classifier outputs a symbol (represented symbolically as a falling yellow frame in Fig. 4-2) corresponding to the class having the minimal distance to the current test sample's window. Thus, each time a new frame becomes available from the input, the window W is shifted to accommodate the new frame's projection in $\boldsymbol{D}$, a new symbol is output from the classifier and sent for processing to the last part of the system - the dynamic buffers structure - which will be described in the next chapter (the exact contents of the symbol output from the classifier will be described at the beginning of section 5-2).

The basic idea of the method used in this section is similar to the subspace method (Watanabe, 1978; Oja, 1983) where a lower-dimensional subspace is created from the training data for each class using the Karhunen-Loeve (KL) expansion (Fukunaga, 1990) and subsequent discrimination is performed by evaluating which class's subspace

provides the best approximation of the test sample's data. However, although the subspaces formed in this way provide an optimal representation of the data for the corresponding class, they do not necessarily provide an optimal space for class discrimination, since the distributions of the other classes are not taken into consideration for their formation. Thus, especially in the cases when the distributions for the different classes are overlapping (as is generally the case with gestures, where different classes of gestures very often contain overlapping motion patterns), it might be expected that a KL-based learning algorithm would perform worse than a LDA-based one. For the purposes of gesture *classification* (rather than gesture *representation*) it would be expected that better features are the ones for which the difference in the class means is large relative to the standard deviations (as in LDA), not the ones for which the standard deviations are large (as in KL). We verified these expectations by observing much better recognition rates for the method proposed in this chapter compared to a KL-based version.

4.4. Combining ensembles of classifiers

In this section it will be shown how a better and more reliable gesture recognition can be achieved by using an *ensemble of classifiers* (rather than a single classifier) based on the same decision rule (4-13) but operating at different resolution scales. The basic idea of combining ensembles of classifiers comes from the observation that if different classifiers are used for the same task, the patterns misclassified by each classifier would not necessarily overlap. This means that if a suitable combination of classifiers is used, each classifier would reflect a slightly different aspect of the input patterns, and combining the "opinions" of an ensemble of classifiers would potentially lead to better performance in comparison with the case when only a single classifier is used at a time.

When combining of classifiers is considered, one of the most important issues is how to determine the combination rule. Many different strategies have been proposed in the literature (Hansen and Salamon, 1990; Franke and Mandler, 1992; Xu et al., 1992; Ho

et al., 1994; Bagui and Pal, 1995), and recently a common theoretical framework has been outlined by Kittler et al. (Kittler at al., 1998) who show that under certain assumptions, some of the most common schemes can be considered as special cases of only two rules, the *product rule* and the *sum rule*. The product rule for assigning a certain pattern Z to class ω_j is defined as

$$P^{-(R-1)}(\omega_j)\prod_{i=1}^{R}P(\omega_j \mid x_i) = \max_{k=1}^{K} P^{-(R-1)}(\omega_k)\prod_{i=1}^{R}P(\omega_k \mid x_i) \quad \Rightarrow \quad Z \in \omega_j \tag{4-14}$$

and the sum rule as

$$(1-R)P(\omega_j)+\sum_{i=1}^{R}P(\omega_j \mid x_i) = \max_{k=1}^{K}[(1-R)P(\omega_k)+\sum_{i=1}^{R}P(\omega_k \mid x_i)] \quad \Rightarrow \quad Z \in \omega_j \tag{4-15}$$

where x_i ($i = 1, \ldots, R$) is the decision output from the i-th classifier, and $P(\omega_k \mid x_i)$ are the a posteriori probabilities yielded by the respective classifiers. Many of the most commonly used combination rules like the *max rule*, the *min rule*, the *median rule* and the *majority vote rule* can be developed from the *product* and *sum* rule if appropriate assumptions (see Kittler et al., 1998 for details) are made about the a posteriori probabilities in (4-14) and (4-15). In the gesture recognition system proposed here we have chosen to utilize the majority vote rule which can be obtained from (4-15) if the a posteriori probabilities $P(\omega_k \mid x_i)$ are hardened to produce the following binary functions

$$\Delta_{ki} = \begin{cases} 1 & \text{if } P(\omega_k \mid x_i) = \max_{j=1}^{K} P(\omega_j \mid x_i) \\ 0 & \text{otherwise} \end{cases} \tag{4-16}$$

so that the resulting rule is given by

$$\sum_{l=1}^{R}\Delta_{jl} = \max_{k=1}^{K}\sum_{l=1}^{R}\Delta_{kl} \quad \Rightarrow \quad Z \in \omega_j . \tag{4-17}$$

The sum on the right hand side of (4-17) effectively counts the votes received for this hypothesis from the different classifiers and the class which receives the majority of votes is selected as the final decision. For the purposes of the implementation proposed here, one way to obtain the a posteriori probabilities of the different gesture classes would be to use (4-13) directly, i.e. to assume the a priory probability to be equal to unity for the class selected by (4-13) and zero for all other classes. An alternative solution, which has been used in the calculations for experiment 6.2.2 in chapter 6, is to output a separate decision

$$O_i(t) = \begin{cases} 1 & \text{if } \arg\max_{r=1}^{K}\{\left\|\mathrm{X}(t) - \Psi(r,t)\right\|^2\} = i \quad \text{for } \forall t \in W \\ 0 & \text{otherwise} \end{cases} \tag{4-18}$$

for each frame in the current window W (see Fig. 4-2; notation is same as in (4-13)) and estimate the a posteriori probability of the classes for that classifier as

$$P(\omega_i \mid x) = \frac{\sum_{t \in W} O_i(t)}{\sum_{t \in W}\sum_{k=1}^{K} O_k(t)}. \tag{4-19}$$

When an ensemble of classifiers is considered, another problem which has to be addressed is how to select a criterion by which the classifiers will differ from each other in some important and meaningful features or parameters, so that performance can be improved by utilizing these differences in "viewing point". We have chosen to construct a

resolution pyramid from the original binary motion images in (3-4) (see also Fig.3-7), such that each next level of the pyramid is obtained by subsampling the image at the previous level using the following algorithm. A 2x2 pixels neighborhood in the source image is replaced by a single pixel in the output image so that if the number of '1's in the 2x2 area is less than 2, the value of the corresponding output image's pixel is '0', and if the number of the '1's in the 2x2 area is greater or equal to 2, the value of the corresponding output image's pixel is '1'.

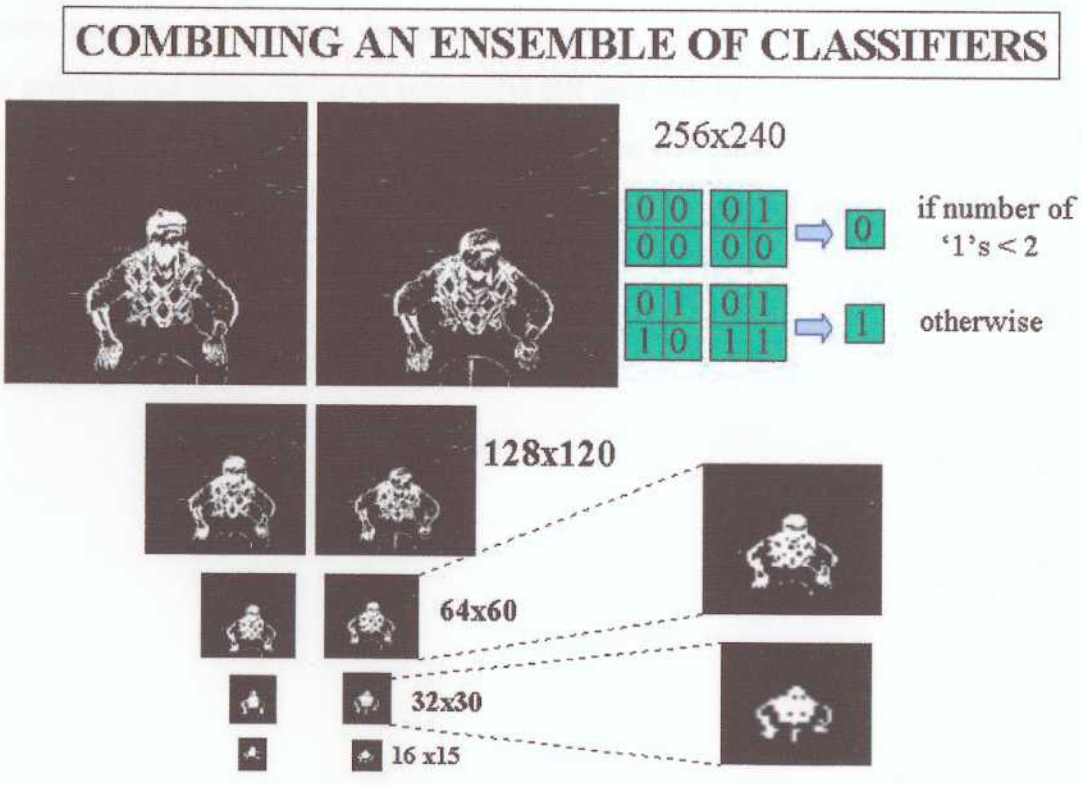

Figure 4-3. The mechanism for combining an ensemble of classifiers operating at different resolution scales for better and more reliable recognition. At the left part of the figure, two consecutive binary motion images from the gesture "bow" are displayed for several different resolutions. The images at scales 64x60 and 32x30 are shown magnified to the right. One possible algorithm for obtaining of the resolution pyramid is shown at top right (see also text for details).

An example of the resulting resolution pyramid for two frames from the gesture "bow" is shown in Fig.4-3. As can be seen from the examples, the effect of the subsampling algorithm is to progressively "blur" the binary motion patterns, so that the higher the level in the pyramid the less binary detail is present (i.e. the result is somewhat similar to a Gaussian filtering of gray-scale images). A separate classifier is associated with each subsequent level of the resolution pyramid, judging the input from a different "resolution point of view" and the final decision is based on (4-17).

An implementation of the method proposed in this section has been created (which presently operates only offline, but a parallel version for online performance can be easily created) and some results on real-world test data will be reported in chapter 6. As expected, higher recognition rates are obtained for the ensemble of classifiers proposed here, compared to only a single classifier's output (an increase of about 3% above the best single classifier's output has been observed).