Shape Classifier based on Generalized Probabilistic Descent Method with Hidden Markov Descriptor

Ninad Thakoor
University of Texas at Arlington
Electrical Engineering Department
ninad.thakoor@uta.edu

Jean Gao
University of Texas at Arlington
Computer Science and Engineering Department
gao@cse.uta.edu

Abstract

The goal of this paper is to present a weighted likelihood discriminant for minimum error shape classification. Different from traditional Maximum Likelihood (ML) methods, in which classification is based on probabilities from independent individual class models as is the case for general Hidden Markov Model (HMM) methods, proposed method utilizes information from all classes to minimize classification error. The proposed approach uses a HMM for shape curvature as its 2-D shape descriptor. In this contribution we introduce a weighted likelihood discriminant function and present a minimum error classification strategy based on Generalized Probabilistic Descent (GPD) method. We believe our sound theory based implementation reduces classification error by combining HMM with GPD theory. We show comparative results obtained with our approach and classic ML classification alongwith Fourier descriptor and Zernike moments based classification for fighter planes and vehicle shapes.

1. Introduction

Object recognition is classic problem of computer vision. Among others, object recognition based on shape is widely used. The problem of general shape recognition can be classified into two categories:

1. Cluster shapes which are similar to each other.
2. Classify the given shape into one of the pre-defined classes.

Category 1 and Category 2 will be referred as *shape recognition* and *shape classification* problem, respectively. The fundamental difference between the two is that classification problem has a set of pre-defined classes while recognition problem does not. In this paper, we will concentrate on shape classification.

First step towards the design of a shape classifier is feature extraction. Shape can be represented either by its contour or by its region [17]. Curvature, chain codes, Fourier descriptors, etc. are contour based descriptors while medial axis transform, Zernike moments, etc. are region based features. Contour based descriptors are widely used as they preserve the local information which is important in classification of complex shapes.

Feature extraction is followed by shape matching. In recent years, dynamic programming (DP) based shape matching is being increasingly applied [11],[12],[7],[1]. DP approaches are able to match the shapes part by part rather than point by point, and are robust to deformation and occlusion. Hidden Markov Models (HMMs) are also being explored as one of the possible shape modeling and classification frameworks [2],[3],[4],[8],[5]. Apart from having all the properties of DP based matching, HMM also provides a probabilistic framework for training and classification.

The authors in [8] were the first to apply HMM to shape classification. They used autoregressive model for shape representation. Results were presented for stationary as well as non-stationary HMMs with 2 to 6 states. Arica and Vural [2] applied a circular HMM topology with 8 states to model the shape. The shape was expressed in terms of 8-directional Freeman's code. This model topology is insensitive to starting point and the sequence length. Their work presented results for Content Based Image Retrieval (CBIR) rather than shape classification.

Cai and Liu [4] applied a Fourier descriptor based HMM topology to classify the shapes. They modified HMM parameter re-estimation procedure to deal with the proposed HMM structure. Recently Bicego and Murino [3] proposed a curvature descriptor based HMM. Curvatures are treated as mixtures of Gaussian and consequently an ergodic HMM is developed. The approach also applied Bayesian Inference Criterion (BIC) to select optimum number of HMM states. Their work provides comprehensive results for classification with deformation, noise and occlusion. These approaches presented classification results for very dissimilar shapes. However, in practical situations shapes to be classified are generally very similar. To handle such situation modifications to existing approaches is mandatory.

Apart from the obvious advantages of HMM like robustness and time warping capability, it provides two levels of descriptions. Hidden state sequence can be considered as a simple description and the likelihood description can be considered as a more detailed description. We use the state sequence description for the invariant starting point detection and likelihood description for the classification. The HMM approaches discussed above apply maximum likelihood (ML) as their classification criterion. Due to good generalization property of HMM, applying ML criterion to similar shapes does not provide good classification. Also, ML criterion is evaluated using information from only one class and does not take advantage of information from the other classes. Generally shapes can be discriminated using only parts of the boundaries rather than comparing whole boundary. ML criterion does not provide such mechanism.

To overcome these shortcomings, we propose a weighted likelihood discriminant for shape classification. The weighting scheme emulates comparison of parts of shape rather than the whole shape. The weights are estimated by applying Generalized Probabilistic Descent (GPD) method. Unlike ML criterion, GPD uses information from all the classes to estimate the weights. As GPD method is designed to minimize the classification error, the proposed classifier gives good classification performance with similar shapes.

This paper is organized as follows: Section 2 and Section 3 give overview of Hidden Markov Model and Generalized Probabilistic Descent method, respectively. The shape description phase of the proposed method is discussed in Section 4, while Section 5 formulates the weighted likelihood discriminant function. GPD method based training algorithm for the proposed discriminant function is described in Section 6. Experimental results are presented in Section 7 and the paper ends with the conclusions in Section 8.

2. Hidden Markov model

HMM is a stochastic signal model widely used in the field of speech processing. Recently many researchers have applied ideas of HMM to shape recognition [2], [3], [4], [8], [5]. HMM can explain an observation sequence $O = O_1O_2 \dots O_T$ in terms of an underlying state sequence $Q = q_1q_2 \dots q_T$. In this section we review HMM briefly. The details of HMM and its applications can be found in [13].

HMM is characterized by following parameters:

1. S, set of states. $S = \{S_1, S_2, \dots, S_N\}$, where N is number of states. State of HMM at instance t is denoted by q_t.
2. A, state transition probability distribution. $A = \{a_{ij}\}$, a_{ij} denotes the probability of changing the state from S_i to S_j.
$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \qquad 1 \leq i, j \leq N. \quad (1)$$
3. B, observation symbol probability distribution. $B = \{b_j(o)\}$, $b_j(o)$ gives probability of observing the symbol o in state S_j at instance t.
$$b_j(o) = P[o \text{ at } t | q_t = S_j], \qquad 1 \leq j \leq N. \quad (2)$$
4. π, initial state distribution. $\pi = \{\pi_i\}$, π_i gives probability of HMM being in state S_i at instance $t = 1$.
$$\pi_i = P[q_1 = S_i], \qquad 1 \leq i \leq N. \quad (3)$$

For convenience, HMM λ can be compactly denoted as,

$$\lambda = (A, B, \pi). \quad (4)$$

In order to apply HMMs to a real world situation, following three problems must be solved:

1. Likelihood problem: Given a model λ, how to efficiently calculate probability of the observation sequence $O = O_1O_2 \dots O_T$, i.e., $P(O|\lambda)$?
2. Optimal path problem: For a given model λ, how to choose a state sequence $Q = q_1q_2 \dots q_T$, which best explains the observation sequence O?
3. Training problem: How to estimate the parameters of model λ such that $P(O|\lambda)$ is maximized?

The likelihood problem is solved by “forward procedure” while the optimal path problem is resolved with “dynamic programming”. Due to a finite number of observations, there is no optimal way to solve the training problem. The parameters are chosen to locally maximize $P(O|\lambda)$ by applying iterative techniques such as “Baum-Welch algorithm” which is a special case of Expectation Maximization (EM) algorithm.

Consider a classification problem to be solved with HMM. An unknown observation sequence O can be classified into one of the M classes, $C_j, j = 1, 2, \dots, M$ as,

$$C(O) = C_i, \quad \text{iff } i\text{= arg} \max_j P(O|\lambda_j), \quad (5)$$

where λ_j is HMM for class C_j. This can be expressed in terms of basic problems of HMM. First λ_j can be estimated by solving the training problem. Then $P(O|\lambda_j)$ can be determined by solving the likelihood problem.

EM based training procedure for HMM utilizes information from single class to train the HMM parameters. This allows one to easily add more classes to the classifier without any need to train the entire classifier again. Training for only the newly added class is necessary to incorporate the class. On the other hand, ML trained HMM classifier cannot discriminate between similar shapes due to its ability to generalize. In practical situations, classification problems generally involve similar shape classes. A better training strategy is thus required to handle such situations. In the following section we give an introduction to the GPD method for training of a discriminant function which performs better in this condition.

3. Generalized probabilistic descent method

The formulation of GPD [9], [10] is closely related to the concept of minimum classification error learning. Consider a problem of classifying an observation vector O into one of the M classes, $C_j, j = 1, 2, \ldots, M$, by discriminant function approach. According to probabilistic descent theorem, the classifier parameters Λ can be iteratively re-estimated to minimize a cost function. This cost function is representation of classification error, which means minimization of the cost function leads to the minimum classification error.

Minimum classification error formulation is a three step procedure. The first step is to select a discriminant function $g_j(O; \Lambda)$. This can be any conventional discriminant function such as a distance measure, maximum likelihood or a discriminant function can be designed to suit the classification scheme under consideration. A misclassification measure is introduced in second step to embed the decision process in a continuous differentiable functional form. One way of defining such measure is,

$$d_j(O; \Lambda) = -g_j(O; \Lambda) + \left\{ \frac{1}{M-1} \sum_{k, k \neq j} g_k(O; \Lambda)^{\eta} \right\}^{\frac{1}{\eta}} \tag{6}$$

where η is a positive smoothing factor. In Eq.(6), η controls the degree to which maximum of the discriminant function for incorrect classes dominates the expression in the bracket. Negative g_j indicates correct classification and positive g_j indicates classification error. In extreme case, when $\eta \to \infty$,

$$d_j(O; \Lambda) = -g_j(O; \Lambda) + g_{k'}(O; \Lambda). \tag{7}$$

Here k' is the maximum of the discriminant functions for incorrect classes. This is equivalent to ideal classification similar to shown in Eq.(5).

In the third step, a cost function is defined which maps the misclassification measure between zero and one. Given below is one of the possible cost functions,

$$l_j(d_j) = \frac{1}{1 + e^{-\xi d_j}}, \qquad \xi > 0. \tag{8}$$

It is required that the discriminant function, the misclassification measure, and the cost function are continuous and differentiable functions of Λ. This ensures that numerical methods like gradient search can be applied to optimize the parameters Λ. According to probabilistic descent theorem, parameter re-estimation rule for above formulation is given as,

$$\Lambda_{n+1} = \Lambda_n - \varepsilon U \nabla l_j(O; \Lambda) \tag{9}$$

where U is a positive definite matrix and ε is small real number. ε is called learning factor and it controls the speed and accuracy of convergence of the parameters.

GPD method uses information from all the classes for training and can be directly used to train the HMM discriminant function i.e. the ML criterion. Other training approaches like Maximum Mutual Information (MMI) based training which use information from all the classes, also exist. However, the classification performance of properly designed and ML trained HMM cannot be improved significantly with MMI or GPD training of HMM [10]. Therefore in our paper, we stay with the optimally designed HMM as described in Section 4 and make our contributions in designing a robust discriminant functions to achieve minimum error.

4. Shape description with HMM

4.1. HMM topology

To achieve good classification results, ML approaches need carefully designed HMM topology with a large number of Gaussian mixtures. For the proposed approach in this paper, the description phase employs HMM topology proposed by Bicego and Murino [3]. The curvature of the shape is used as the descriptor. Any shape can be assumed to be formed by various segments, each of which has a constant curvature. Any deviation from the constant curvature can be due to the noise or due to the details of the shape. Each of these segments are treated as the states of the HMM. Each state is modeled as Gaussian distribution with mean representing the constant curvature of the segment and standard deviation representing the deviation from constant curvature of the segment. In the rest of the section we discuss the formulation of HMM for the shape.

The shape is first filtered with large variance Gaussian filter to reduce the effect of noise in curvature estimation. The filtered shape is normalized to a fixed length to make the curvature invariant of the scale. Let the normalized shape be indicated by $D = \{D_n\}$ and $D_n = (x_n, y_n)$ for $1 \leq n \leq T$, where T is the normalized length of the shape, and D_n indicates the coordinates of n^{th} point of the shape. Finally, approximate curvature at each point is calculated as the turn angle at that point. The turn angle θ_n at point D_n is defined as,

$$\theta_n = \arctan \frac{y_n - y_{n-1}}{x_n - x_{n-1}} - \arctan \frac{y_n - y_{n+1}}{x_n - x_{n+1}} \tag{10}$$

The turn angle θ_n is treated as the observation O_n for the HMM. Each shape class is modeled by a N-state ergodic HMM and observation symbol probability distribution, i.e., b_j of each state is modeled by a one-dimensional Gaussian distribution. Gaussian Mixture Model (GMM) [14] for N clusters estimated from unrolled values of curvature of L samples of the shape, is used to initialize $B = \{b_j\} = \{(\mu_j, \sigma_j)\}$. Baum-Welch algorithm is then applied to estimate the parameters of the HMM $\lambda_j = (A_j, B_j, \pi_j)$. For every example sequence of same class, separate HMMs are built. These models are combined to form a single HMM to represent the shape class. Assuming each observation sequence has equal probability, we can express the combined HMM parameters as,

$$\pi_j = \sum_{i=1}^{L} \pi_j^i / L, \tag{11}$$

$$A_j = \sum_{i=1}^{L} A_j^i / L, \tag{12}$$

$$\mu_j = \sum_{i=1}^{L} \mu_j^i / L, \tag{13}$$

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{L} \left(\sigma_j^i\right)^2}{L} + \frac{\sum_{i=1}^{L} \left(\mu_j^i\right)^2}{L} - \left(\mu_j\right)^2}, \tag{14}$$

where $\lambda_j^i = (A_j^i, B_j^i, \pi_j^i)$ represents the HMM for i^{th} example from L examples of j^{th} class. In addition to these parameters, the number of HMM states N, is another important parameter. As N increases, computational complexity involved in the training and the classification increases. As likelihood of the model increases with N, ML criterion cannot be used to select optimal N. We choose optimum N for the HMM by applying the Bayesian Inference Criterion (BIC) [15]. BIC penalizes the likelihood of the HMM according to its complexity. In [3] BIC is applied to GMM to select optimal N, but this gives optimal N for the GMM and not for the HMM. In our approach, BIC is applied to the HMM to ensure proper model selection. For the HMM topology discussed, BIC can be written as,

$$\text{BIC}_N(\lambda_j) = \text{log}P(O|\lambda_j) - \frac{N^2 + 2N - 1}{2}\text{log}(T). \tag{15}$$

Penalty term in Eq.(15) is derived from number of free parameter of the model and the observation sequence length. Number of states N is selected to maximize the $\text{BIC}_N(\lambda_j)$.

4.2. Invariant starting point detection

Before we propose the weighted discriminant function, it is important to detect the starting point of a closed shape invariantly as formulation depends on it. For open shapes this step is not required. Major axis based rotation similar to that of He and Kundu [8] can be used to achieve this. However in case of some shapes, slight change in the shape results in a large change in the major axis. Shapes with high degree of symmetry cannot be aligned properly with the method. This method cannot detect reflection of shape. To overcome this, we apply state sequence based rotation to detect the starting point of the shape. Recall that the shape is modeled as constant curvature segments. Comparison between these segments gives the criterion for alignment for our method. These segments are nothing but optimum state sequence for the HMM.

Consider a closed shape $\bar{O} = \bar{O}_1\bar{O}_2 \dots \bar{O}_T$ modeled by HMM $\lambda = (A, B, \pi)$ and another closed shape $O = O_1 O_2 \dots O_T$ to be aligned with the shape $\bar{O}$. The best path sequence for $\bar{O}$ is given by $\bar{Q} = \bar{q}_1\bar{q}_2 \dots \bar{q}_T$.

$$\bar{Q} = \text{arg} \max_{q_1 q_2 \cdots q_T} P(q_1 q_2 \dots q_T | \bar{O}, \lambda) \tag{16}$$

Similarly, best path $Q = \tilde{q}_1\tilde{q}_2 \dots \tilde{q}_T$ for O is,

$$Q = \text{arg} \max_{q_1 q_2 \cdots q_T} P(q_1 q_2 \dots q_T | O, \lambda) \tag{17}$$

Mismatch between these descriptions is defined as,

$$\Delta Q(k) = \sum_{t=1}^{T} \delta(t, h(t-k)) \tag{18}$$

where,

$$\delta(n_1, n_2) = \begin{cases} 0, & \text{if } \bar{q}_{n_1} = \tilde{q}_{n_2}; \\ 1, & \text{if } \bar{q}_{n_1} \neq \tilde{q}_{n_2}. \end{cases} \tag{19}$$

and

$$h(n) = \begin{cases} n, & \text{if } n > 0; \\ n + T, & \text{if } n \leq 0. \end{cases} \tag{20}$$

Let $O' = \{O'_1 O'_2 \dots O'_T\}$ indicate the reflection of shape O and $\Delta Q'$ be the corresponding mismatch with respect to the reference shape. The aligned shape is given as,

$$O^* = \begin{cases} O_{k^*+1} O_{k^*+2} \dots O_T O_1 O_2 \dots O_{k^*}, \\ \qquad \text{if } \Delta Q(k^*) < \Delta Q'(k^*); \\ O'_{k^*+1} O'_{k^*+2} \dots O'_T O'_1 O'_2 \dots O'_{k^*}, \\ \qquad \text{otherwise.} \end{cases} \tag{21}$$

where,

$$k^* = \text{arg} \min_{0 \leq k \leq T-1} \{\min(\Delta Q(k), \Delta Q'(k))\} \tag{22}$$

Any of the example shapes can be treated as the reference shape for starting point detection. In the following sections, we assume that starting point is already detected using this method and use the obtained HMM description for minimum error classification formulation.

5. Discriminant function formulation

In this section, we formulate a minimum error classifier with a weighted likelihood discriminant function. The discriminant function is derived from the intuitive idea that similar shapes can be discriminated by comparing the parts of their boundaries. Meaning, some parts of the shape contour play important role in classification than the others. We signify the importance of part of shape in classification by assigning weights to it. The weights introduced in the discriminant function will be trained with GPD method.

Consider observation sequence to be classified, $O = O_1 O_2 ... O_T$. We model the sequence with HMM $\lambda = (A, B, \pi)$. One of the possible state sequences for the given observation sequence is given by, $Q = q_1 q_2 ... q_T$. The probability of observation sequence O given the state sequence Q and model λ is,

$$P(O|Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \dots b_{q_T}(O_T). \tag{23}$$

Probability of state sequence Q can be written as,

$$P(Q|\lambda) = \pi_{q_1} \cdot a_{q_1 q_2} \dots a_{q_{T-1} q_T}. \tag{24}$$

Then probability of the both O and Q occurring simultaneously is given by,

$$P(O, Q|\lambda) = P(O|Q, \lambda) \cdot P(Q|\lambda)$$

The probability of O is obtained by summing the above joint probability over all possible state sequences.

$$\begin{aligned} P(O|\lambda) &= \sum_{all\ Q} P(O|Q,\lambda) \cdot P(Q|\lambda) \\ &= \sum_{q_1,q_2\ldots q_T} \pi_{q_1} b_{q_1}(O_1)\, a_{q_1 q_2} b_{q_2}(O_2) \ldots \\ &\qquad \ldots a_{q_{T-1} q_T} b_{q_T}(O_T). \end{aligned} \tag{25}$$

Eq.(25) is nothing but ML criterion which is widely applied in classification problems. However, our goal is to weight the observations individually and the individual observation probabilities cannot be extracted from Eq.(25). To achieve this, we express the ML criterion in terms of a forward variable $\alpha_t(i)$. Forward variable is defined as,

$$\alpha_t(i) = P(O_1 O_2 \ldots O_t, q_t = S_i|\lambda). \tag{26}$$

Values of the forward variable can be calculated with the forward procedure [13]. Probability of partial observation sequence $O_1 O_2 \ldots O_t$ given the model λ can be written as,

$$P(O_1 O_2 \ldots O_t|\lambda) = \sum_{i=1}^{N} \alpha_t(i). \tag{27}$$

For $t = 1$ above equation reduces to,

$$P(O_1|\lambda) = \sum_{i=1}^{N} \alpha_1(i). \tag{28}$$

Similarly for $t = 2$,

$$P(O_1 O_2|\lambda) = \sum_{i=1}^{N} \alpha_2(i). \tag{29}$$

As observations O_1 and O_2 are independent, from Eqns.(28) and (29),

$$P(O_2|\lambda) = \frac{\sum_{i=1}^{N} \alpha_2(i)}{\sum_{i=1}^{N} \alpha_1(i)}. \tag{30}$$

Repeating the above procedure, it can be shown that for $t > 1$,

$$P(O_t|\lambda) = \frac{\sum_{i=1}^{N} \alpha_t(i)}{\sum_{i=1}^{N} \alpha_{t-1}(i)}. \tag{31}$$

Now, logarithm of the ML criterion can be expressed in terms of observation probabilities as,

$$\log P(O|\lambda) = \sum_{t=1}^{T} \log P(O_t|\lambda). \tag{32}$$

This function gives equal importance to every point of the shape in the classifications. Hence, we introduce a new discriminant function which weights the curvature likelihood of shape points according to their importance in classification.

The new discriminant function, g_j is given by,

$$g_j = \sum_{t=1}^{T} w_j(t).\log P(O_t|\lambda_j), \tag{33}$$

where w_j is weighting function for class C_j. w_j provides additional discrimination among the classes. These weights will be tuned by applying GPD method to minimize the classification error. Weighting function at individual observation can be estimated by applying GPD to the current formulation. But due to the large number of parameters (equal to T), the convergence of GPD will be slower and will need a large number of observation sequences for training. As mentioned previously in the section, to discriminate between similar shapes, comparison between parts of their contour is sufficient. As a result, shape can be weighted segment by segment instead of being weighted pointwisely. Following this idea, weighting functions are chosen to be windows which can adapt their position, spread and height. Although any smooth window function can be selected, our approach uses weighting function given in Eq.(34), which is sum of S Gaussian shaped windows.

$$w_j(t) = \sum_{i=1}^{S} h_{i,j} \cdot e^{-\frac{(t-\mu_{i,j})^2}{s_{i,j}^2}}. \tag{34}$$

Parameter $h_{i,j}$ governs the height, $\mu_{i,j}$ controls the position, while $s_{i,j}$ determines spread of i^{th} window of j^{th} class. In this case, we have only $3S$ parameters to estimate. The discriminant function can now be written as,

$$g_j = \sum_{t=1}^{T} \sum_{i=1}^{S} h_{i,j} \cdot e^{-\frac{(t-\mu_{i,j})^2}{s_{i,j}^2}} \cdot \log P(O_t|\lambda_j). \tag{35}$$

In the next section, GPD method is applied to above discriminant function to formulate the training algorithm.

6. GPD algorithm

To complete the formulation of GPD, we introduce a misclassification measure for observation sequence of j^{th} class as,

$$d_j = -g_j + \frac{1}{\eta} \log \left(\frac{1}{M-1} \sum_{k,k \neq j} e^{\eta \cdot g_k} \right) \tag{36}$$

and corresponding cost function as,

$$l_j = \frac{1}{1 + e^{-\xi \cdot d_j}} \tag{37}$$

As discussed in Section 3, the probabilistic descent re-estimation rule for parameters Λ is given as,

$$\Lambda_{n+1} = \Lambda_n - \varepsilon U \nabla l_j. \tag{38}$$

For the proposed method, U is chosen to be identity matrix and the learning factor, ε is chosen to be a small number compared to the dynamic range of the parameter. The re-estimation rules in iteration n, for i^{th} window parameters of k^{th} class when C_j is the correct class are given by,

$$h_{i,k}^{n+1} = h_{i,k}^{n} - \varepsilon_h \cdot \frac{\partial l_j}{\partial h_{i,k}}, \tag{39}$$

$$\mu_{i,k}^{n+1} = \mu_{i,k}^{n} - \varepsilon_\mu \cdot \frac{\partial l_j}{\partial \mu_{i,k}}, \tag{40}$$

$$s_{i,k}^{n+1} = s_{i,k}^{n} - \varepsilon_s \cdot \frac{\partial l_j}{\partial s_{i,k}}, \tag{41}$$

for $1 \leq i \leq S$, $1 \leq k \leq M$. Partial derivatives appearing in Eqns.(39)-(41) can be calculated by chain rule as,

$$\frac{\partial l_j}{\partial h_{i,k}} = \frac{\partial l_j}{\partial d_j} \cdot \frac{\partial d_j}{\partial g_k} \cdot \frac{\partial g_k}{\partial h_{i,k}}, \tag{42}$$

$$\frac{\partial l_j}{\partial \mu_{i,k}} = \frac{\partial l_j}{\partial d_j} \cdot \frac{\partial d_j}{\partial g_k} \cdot \frac{\partial g_k}{\partial \mu_{i,k}}, \tag{43}$$

$$\frac{\partial l_j}{\partial s_{i,k}} = \frac{\partial l_j}{\partial d_j} \cdot \frac{\partial d_j}{\partial g_k} \cdot \frac{\partial g_k}{\partial s_{i,k}}, \tag{44}$$

where,

$$\frac{\partial l_j}{\partial d_j} = \frac{\xi e^{-\xi . d_j}}{(1 + e^{-\xi . d_j})^2}, \tag{45}$$

$$\frac{\partial d_j}{\partial g_k} = \begin{cases} -1, & j = k; \\ \frac{e^{\eta \cdot g_k}}{\sum_{k', k' \neq j} e^{\eta \cdot g_{k'}}}, & j \neq k. \end{cases} \tag{46}$$

$$\frac{\partial g_k}{\partial h_{i,k}} = \sum_{t=1}^{T} e^{-\frac{(t-\mu_{i,k})^2}{s_{i,k}^2}} \log P(O_t|\lambda_k), \tag{47}$$

$$\frac{\partial g_k}{\partial \mu_{i,k}} = \sum_{t=1}^{T} \frac{2h_{i,k}(t-\mu_{i,k}) e^{-\frac{(t-\mu_{i,k})^2}{s_{i,k}^2}}}{s_{i,k}^2} \log P(O_t|\lambda_k), \tag{48}$$

$$\frac{\partial g_k}{\partial s_{i,k}} = \sum_{t=1}^{T} \frac{2h_{i,k}(t-\mu_{i,k})^2 e^{-\frac{(t-\mu_{i,k})^2}{s_{i,k}^2}}}{s_{i,k}^3} \log P(O_t|\lambda_k). \tag{49}$$

Note that in above formulation log$P(O_t|\lambda_k)$ is treated as a constant, as the HMM parameters are not affected by the change in Λ.

7. Experimental results

As mentioned before, we focus on shape classification problem where shapes are similar. This is an application of wide interest for surveillance e.g. separating vehicle types (civilian surveillance application), classifying aeroplane or tank views (military surveillance applications) etc. Most public benchmark databases are designed to test the shape clustering or similarity based shape retrieval. As we deal with the shape classification problem where shape classes are predefined and are very similar, these data sets are not appropriate for this problem . The proposed method was tested with two different data sets specially designed to test our method. First data set was aeroplane shapes and the second data set included shapes of vehicles.

7.1. Fighter plane shapes

The fighter aeroplane shape database included Mirage, Eurofighter, F-14, Harrier, F-22 and F-15. Since F-14 has two possible shapes, one when its wings are closed and another when its wings are opened, total number of shape classes are seven. Each class includes 30 shape samples. Shape database was created by taking digital pictures of die-cast replica models of these aeroplanes from top. Pictures were captured at 640×480 resolution, and were segmented using Spedge and Medge [6] color image segmentation algorithm. Contours of the segmented planes were used for training and testing of the classifier. Figure 1 shows the extracted shapes for different classes. The extracted shapes exhibit deformation due to varying view points and noise due to automatic segmentation.

Shapes were filtered with Gaussian filter (standard deviation = 10) and shape length was normalized to 512 points. The normalized shapes were split randomly into training and testing samples. For the training samples of each class, HMM was built as explained in Section 4. Optimum number of HMM states were selected by applying BIC to models with 3 to 6 states. Sum of 20 Gaussian windows was used for formulation and training of the discriminant function. The window parameters were initialized to spread the windows uniformly over the shape. The training vectors were used to train the classifier with $\xi = 1$ and $\eta = 10$. Once the training was complete, testing samples were used to determine the classification performance. For comparison, ML classification was carried out with optimal HMM after application of BIC. Also Fourier descriptor (FD) and Zernike moments (ZM)(up to order 30) based classification was carried out to compare our method with conventional shape classification methods.

Training and classification of the shapes was carried out with 15 training samples and 15 testing samples per class. The classification results were averaged over 20 runs of the classifier design, each time with different combination of training and testing samples. Figure 2 shows the ML discriminant functions and weighted likelihood discriminant functions for the test vectors. The test vectors are grouped in sets of 15. Labels just above the x-axis indicate the correct class for the test vector and dotted lines separate the correct classes. For correct classification, discriminant function of correct class should be maximum. Difference between the discriminant function of correct class and the other classes is not clear in ML for all the classes. As a result, the classification accuracy is not satisfactory. For weighted likelihood discriminant, this difference is large and clearly separable. This large difference results into very high accuracy for the proposed classifier.

Table 1 compares classification accuracy GPD based weighted likelihood classification (WtL) with Fourier de-

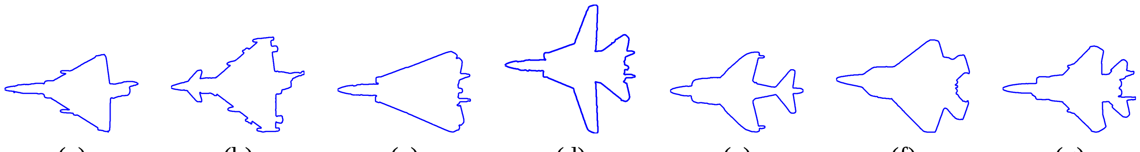


Figure 1. Aeroplane shape classes: (a) Mirage, (b) Eurofighter, (c) F-14 wings closed, (d) F-14 wings opened, (e) Harrier, (f) F-22, (g) F-15.

scriptor (FD) classification, Zernike moments (ZM) classification and HMM classification with maximum likelihood (ML).

Table 1. Classification accuracy in % for aeroplane shapes

Class	FD	ZM	ML	WtL
Mirage	87.00	83.00	49.33	100.00
Eurofighter	34.00	97.00	88.67	99.00
F-14 Close	56.00	90.33	100.00	98.67
F-14 Open	100.00	100.00	26.67	96.67
Harrier	75.33	78.33	92.00	100.00
F-22	72.67	94.67	96.00	100.00
F-15	60.33	76.33	71.67	99.00

7.2. Vehicle shapes

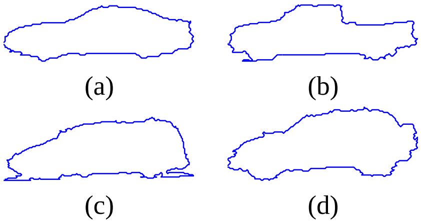


Figure 3. Vehicle shape classes: (a) Sedan, (b) Pickup, (c) Minivan, (d) SUV.

In the second experiment, we classified the vehicles shapes extracted from traffic videos using the motion information. Approach discussed in [16] was implemented and applied to outdoor videos to extract these shapes. The vehicles were classified into one of the four class: sedan, pickup, minivan or SUV.

Videos were captured at resolution of 320×240. As object extraction approach used does not deal with shadows, the extracted car shapes are distorted in the bottom half due to shadow. 30 samples for each class were extracted from the video. Extracted shapes were filtered with Gaussian filter (standard deviation = 5) to reduce the effect of the noise and shape length was normalized to 128 points. The normalized shapes were split randomly into training and testing samples similar to first experiment. After training HMM for all the classes, optimum number of HMM states were selected by applying BIC to models with 3 to 6 states. Sum of 16 Gaussian windows was used for formulation and training of the discriminant function with $\xi = 10$ and $\eta = 10$. Once the training was complete, testing samples were used to determine the classification performance. The process was repeated 20 times.

Comparative classification performance for individual class can be seen in Table 2. In this experiment, the classification accuracy was lower than the first experiment due the reasons, (1) Shape samples within the class show larger variation, as shape of vehicles of different makes and models vary. (2) The contours extracted show higher degree of deformation due to the shadow problem in object extraction. Despite of these, overall classification accuracy of the scheme was found to be 87% compared to ML classification accuracy of 72%.

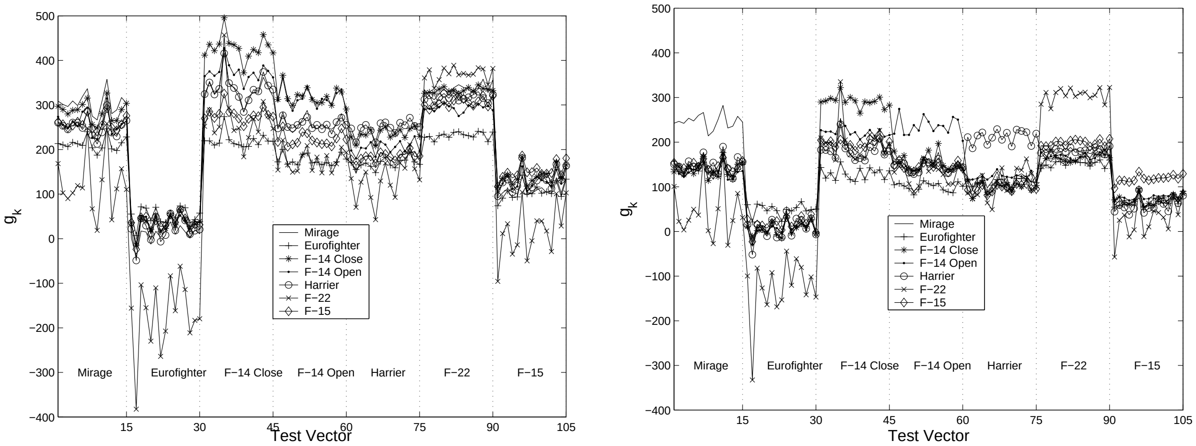
Table 2. Classification accuracy in % for vehicle shapes

Class	FD	ZM	ML	WtL
Sedan	58.33	98.00	93.00	92.33
Pickup	62.33	29.67	69.33	87.33
Minivan	25.33	46.00	90.33	94.33
SUV	24.67	53.67	33.33	75.00

8. Conclusion

In this paper, we proposed a weighted likelihood discriminant function for shape classification by the combination of GPD theory and HMM. HMM was applied as a robust descriptor for individual classes and the weighted likelihood discriminant function was used to discriminate amongst them. The weighting emulates feature selection scheme which selects features required to correctly classify the shapes. A training algorithm based on GPD method to estimate the optimal weights to minimize the classification error was formulated.

The performance of the proposed shape classification scheme was tested with two different shape data sets. As these data sets were not generated synthetically, the results obtained for the classification are reliable in practical scenarios. In the first experiment, ML based classification accuracy of 75% was improved to 99% by proposed method and in second experiment it was improved from 72% to 87%. Improvement over conventional classification with



(a) (b)

Figure 2. (a) ML discriminant functions, (b) Weighted likelihood discriminant functions.

Fourier descriptors and Zernike moments is also seen from the results.

Currently the weighting windows are spread uniformly over the shape contour. But these windows can used as expert input to the classification system. One such example would be, weighting only top parts of vehicle shapes as bottom parts of the vehicles are very similar and thus are not important for classification. In our experiments however we avoid this to show generality of our approach.

References

[1] T. Adamek and N. E. O'Connor. A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):742–753, May 2004.

[2] N. Arica and F. Yarman-Vural. A shape descriptor based on circular hidden markov model. *Proceedings of IEEE International Conference on Pattern Recognition*, 1:924–927, 2000.

[3] M. Bicego and V. Murino. Investigating hidden markov models' capabilities in 2d shape classification. *IEEE Transactions on Pattern Recognition Machine Inteligence*, 26(2):281–286, Feb 2004.

[4] J. Cai and Z.-Q. Liu. Hidden markov models with spectral features for 2d shape recognition. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 23(12):1454–1458, Dec. 2001.

[5] A. Fred, J. Marques, and P. Jorge. Hidden markov models vs. syntactic modeling in object recognition. *Proceedings of IEEE International Conference of Image Processing*, 1:893–896, 1997.

[6] J. Gao, A. Kosaka, and A. Kak. Interactive color image segmentation editor driven by active contour model. *Proceedings of International Conference on Image Processing*, 3:245–249, Oct. 1999.

[7] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1312–1328, Dec. 1999.

[8] Y. He and A. Kundu. 2-d shape classification using hidden markov model. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 13(11):1172–1184, Nov. 1991.

[9] S. Katagiri, B.-H. Juang, and C.-H. Lee. Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method. *Proceedings of IEEE*, 86(11):2345–2373, Nov. 1998.

[10] E. McDermott. *Handbook of Neural Networks for speech processing*, chapter 5, pages 159–216. Artech House, 2000.

[11] E. Milios and E. G. M. Petrakis. Shape retrieval based on dynamic programming. *IEEE Transactions on Image Processing*, 9(1):141–147, Jan. 2000.

[12] E. G. M. Petrakis, A. Diplaros, and E. Milios. Matching and retrieval of distorted and occluded shapes using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1501–1516, Nov. 2002.

[13] L. R. Rabiner. A tutorial on hidden markov models and selected application in speech recognition. *Proceedins of IEEE*, 77(2):257–286, Feb. 1989.

[14] D. A. Reynolds and R. C. Rose. Robust text-independant speaker identification using gaussian mixture models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, Jan. 1995.

[15] G. Schwarz. Estimating the diamension of a model. *The Annals of Statistics*, 6(6):461–464, 1978.

[16] N. Thakoor and J. Gao. Automatic video object detection and recognition with camera in motion. *To be publised in proceedings of IEEE International Conference on Image Processing*, 2005.

[17] D. Zhang and G. Lu. Review of shape representation and description technique. *Pattern Recognition*, 37(1):1–19, Jan. 2004.