# HYBRID FEATURE SELECTION FOR GESTURE RECOGNITION USING SUPPORT VECTOR MACHINES

*Yu Yuan and Kenneth Barner*

University of Delaware
Department of Electrical and Computer Engineering
Newark, DE 19716
Email:{yyuan,barner}@udel.edu

**ABSTRACT**

This paper presents an algorithm for extracting and classifying two-dimensional motion in an image sequence based on trajectories. Each gesture signal is represented as a time series in a Principal Component Analysis (PCA) reduced dimensional space. A class of Support Vector Machine (SVM) applicable to sequential-pattern recognition is employed for classification by incorporating a hybrid distance measure into the kernel function that accounts for both the hand shape and movement. The performance of the proposed method is evaluated in continuous tactile hand gesture streams recognition experiments. Results are presented for 9 different gestures performed by 25 subjects at a variety of time scales. Experimental results show that the proposed approach yields high recognition rate for hand gesture motion patterns.

***Index Terms***— Support vector machines, feature selection, gesture recognition

## 1. INTRODUCTION

Hand gesturing is important for communication. Simple human-machine interface (HCI) based on hand gesture recognition is an open problem. In the development of such systems, there are three problems: static recognition of hands form, hand tracking and dynamic gesture recognition.

A gesture interface is an interface where users specify commands by simple gestures, such as drawings and actions. It is common today to control a graphical interface through some mediating device such as the multi-touchpad illustrated in Figure 1. The gestures on a multi-touchpad serve as input to communicate and interact with a computer. That is, time-varying motion patterns (gestures) are used to convey a message to a computer, which can detect and distinguish these motion patterns and respond appropriately. To develop such an interface, the key issues are how to recognize gesture sets, taking into account both trajectory and shape information.

For the solution of tactile gesture recognition, a sequence of images containing a hand gesture is at our main focus. This leads to the following definition of the problem: Given a sequence of images containing a hand gesture, find a gesture in a database of predefined gestures which is the most similar to that. An extensive review of existing hand gesture recognition techniques are given in [1]. Commonly used gesture recognition algorithms fall into two categories: generative and discriminative approaches. It has often been argued that for many application domains, discriminative classifiers often achieve higher test set accuracy than generative classifiers [2], among which SVM is the representative.

**Fig. 1**. Multi-touch pad for gestures

Gesture-based HCIs define a set of gestures suitable for a particular task. If only shape or motion information in the gestures is utilized, the system's discriminative capability is limited, confining the system to a small gesture vocabulary. Since slightly different gestures are allowed to represent different semantic meanings, a framework that can account for both the shape as well as temporal trajectory of the gesticulating hand is desired. Hand shape or trajectory information alone may not be discriminative enough in these cases, which makes recognition a more challenging task. To address this problem, we propose a hybrid feature extraction method that adopts both the parameters representing the shape of the moving hand and motion information, coupled with SVM to classify the unknown gestures utilizing a Gaussian radial basis function kernel.

The remainder of this paper is organized as follows: the method for extracting features in hand shapes and motions is presented in Section 2. Section 3 presents the SVM approach in details. Results are given in Section 4 and conclusions are drawn in Section 5.

## 2. FEATURE EXTRACTION

As using shape or motion trajectories as the single cue may not be sufficient to disambiguate between gestures, use of a multi-cue framework serves to alleviate this problem by falling back on one cue when the other fails to discriminate. Moreover, in the presence of multiple fingertips, common gestures recognition approaches assume that the order of fingertips in input images do not change due to finger crossings. In finger crossing cases, centroid based methods do not capture accurate gesture information. This motivates us to take an entire frame as input and apply PCA to reduce the dimension followed by SVM based classification.

### 2.1. SHAPE FEATURE EXTRACTION

Preprocessing is performed to remove irrelevant variability occurring in the raw coordinate sequence, which include translation and scaling to move the shape centroid to the center of the image and normalize the pattern to a fixed dimension. The feature extractor is fed with the binary image of an isolated hand shape to generate local features. The first feature[3] employed is given by the difference between the sums related to orthogonal directions

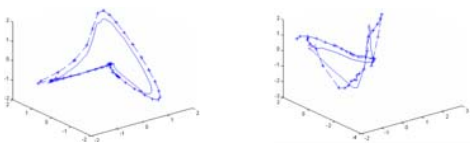
$$d = \frac{1}{2}(1 + \frac{1}{hw^2}\sum_{j=1}^{h} n_j^2 - \frac{1}{h^2 w}\sum_{i=1}^{w} n_i^2) \qquad (1)$$

where $h$ and $w$ are the height and the width in pixels of the hand shape, while $n_i$ and $n_j$are the number of black pixels computed from the $i^{th}$ horizontal and $j^{th}$ perpendicular line. The second feature is the $width/height$ ratio.

### 2.2. MOTION FEATURE EXTRACTION

The key idea of the recognition process is to map gesture frames into an appropriately chosen lower dimensional subspace and perform classification by distance computation. By projecting the original data onto a PCA constructed subspace, the essential structure of the data can be captured in lower dimensions. Figure 2 illustrates two pairs of manifolds of gestures performed by different hand shapes projected into the reduced dimensional subspace, which is formed by using the beginning frames of all the gestures in the training set. Every image is mapped onto a point and sets of similar gestures form clusters in the subspace.

A problem associated with sequence comparison for gesture recognition is the fact that different performances of the same gesture action are seldom realized at the same speed across the entire duration. The temporal properties of the gesture are therefore typically normalized using Dynamic Time Warping (DTW) [4].



(a) (b)

**Fig. 2**. Gestures of the same class projected into PCA lower dimensional space: (a)Two Rectangles and (b)Two Circles.

Consider two manifolds $\mathbf{P}$ and $\mathbf{Q}$, represented by the sequences $(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{T_p})$ and $(\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{T_q})$, respectively, where $\mathbf{p}_i$ and $\mathbf{q}_i$ are parameter vector of the sequence features. We form a feature vector for each $\mathbf{p}_i$ of $\mathbf{P}$ by normalized coordinates and the corresponding tangent slope angles. Specifically, the six local parameters at the $i^{-th}$ sample point $(x_i, y_i, z_i)$ are: $\mathbf{p}_i = (\widetilde{x}_i, \widetilde{y}_i, \widetilde{z}_i, \theta_{xi}, \theta_{yi}, \theta_{zi})^T$. The quantities $\widetilde{x}_i = \frac{x_i - \mu_x}{\sigma_x}$, $\widetilde{y}_i = \frac{y_i - \mu_y}{\sigma_y}$ and $\widetilde{z}_i = \frac{z_i - \mu_z}{\sigma_z}$ are the coordinates normalized by the mean $(\mu_x, \mu_y, \mu_z)$. The feature $\theta_{xi}$, $\theta_{yi}$ and $\theta_{zi}$ are the tangent slope angles at point $i$ approximated by

$$tan^{-1}\frac{x_{i+1} - x_{i-1}}{\sqrt{(y_{i+1} - y_{i-1})^2 + (z_{i+1} - z_{i-1})^2}},$$

$$tan^{-1}\frac{y_{i+1} - y_{i-1}}{\sqrt{(x_{i+1} - x_{i-1})^2 + (z_{i+1} - z_{i-1})^2}}$$

and

$$tan^{-1}\frac{z_{i+1} - z_{i-1}}{\sqrt{(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2}}.$$

Since $\theta_{xi}$, $\theta_{yi}$ and $\theta_{zi}$ are circular measures, they require special treatment necessary when computing the local distance $d(\mathbf{p}_i, \mathbf{q}_j) = \| \mathbf{p}_i - \mathbf{q}_j \|^2$. Circular differences mod $2\pi$ are applied to $\theta_{xi}$, $\theta_{yi}$ and $\theta_{zi}$[4]. The alignment distance is determined by summing the local distances between matched sequences of length $N$ solved by DTW,

$$D_a(\mathbf{P}, \mathbf{Q}) = \frac{1}{N}\sum_{n=1}^{N} d(\mathbf{P}_{\phi(n)}, \mathbf{Q}_{\psi(n)}) \qquad (2)$$

## 3. SUPPORT VECTOR MACHINE CLASSIFICATION

The utilization of support vector machine classifiers has gained immense popularity in recent years. SVMs map the training data into a higher-dimensional feature space via a kernel, and to construct a separating hyperplane with maximum margin between the two classes, which yields a nonlinear decision boundary in the input space.

**Table 1**. Classification accuracy for each pairing of basic gestures with $\beta = 1$. Two rightmost columns give overall mean accuracy for each row's gesture and DDAG results.

| Gesture | Circle | Rectangle | Triangle | P | M | N | W | Bell | Z | Overall | DDAG |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Circle** | - | 93.0% | 94.3% | 97.2% | 99.3% | 99% | 98.6% | 97.5% | 99.4% | 97.3% | 95.7% |
| **Rectangle** | 93.0% | - | 96.2% | 86% | 98.7% | 96.1% | 91.7% | 94.0% | 99.0% | 94.3% | 92.4% |
| **Triangle** | 94.3% | 96.2% | - | 92.7% | 95.1% | 96.8% | 90.2% | 94.6% | 89.3% | 93.7% | 90.8% |
| **P** | 97.2% | 86.0% | 92.7% | - | 93.4% | 99.2% | 97.3% | 94.5% | 98.7% | 94.9% | 91.5% |
| **M** | 99.3% | 98.7% | 95.1% | 93.4% | - | 90.6% | 88.0% | 92.8% | 98.1% | 94.5% | 93.7% |
| **N** | 99.0% | 96.1% | 96.8% | 99.2% | 90.6% | - | 85.6% | 91.7% | 98.5% | 94.7% | 92.9% |
| **W** | 98.6% | 91.7% | 90.2% | 97.3% | 88.0% | 85.6% | - | 93.9% | 94.3% | 92.5% | 91.6% |
| **Bell** | 97.5% | 94.0% | 94.6% | 94.5% | 92.8% | 91.7% | 93.9% | - | 97.0% | 94.5% | 90.7% |
| **Z** | 99.4% | 99.0% | 89.3% | 98.7% | 98.1% | 98.5% | 94.3% | 97.0% | - | 96.8% | 92.9% |

### 3.1. SUPPORT VECTOR MACHINES

The SVM is described in detail by Vapnik [2]. Given a set of linearly separable two-class training data, there are many possible solutions for a discriminative classifier. In the case of the SVM, a separating hyperplane is chosen so as to maximize the margin between the two classes. A SVM classifies a pattern vector **x** based on the training data points $\mathbf{x}_i$ and corresponding labels $y_i$ into classes $\{-1, +1\}$, which corresponds to a nonlinear decision boundary of the form

$$y = sgn(\sum_{i=1}^{L} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b) \tag{3}$$

where $K(\cdot, \cdot)$ is a symmetric positive-definite kernel function. The computation of dot products between vectors without explicitly mapping to another space is performed by a kernel function. The training examples with $\alpha_i \neq 0$ are called *Support Vectors*. Several widely used classifier functions reduce to special valid forms of kernel functions, like $n^{th}$ order polynomial classifiers and radial basis functions (RBFs).

$$K_{RBF}(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \tag{4}$$

The obtained support vectors are found to be almost invariant to the type of kernel functions used [2], which indicates that this choice is not critical to classification performance.

The parameters $\alpha_i$ and $b$ are determined by a linearly constrained quadratic programming problem [5], which can be efficiently implemented by means of a sequence of smaller scale subproblem optimization. The key point for efficient implementation of the SVM is that typically only a small fraction of the $\alpha_i$ coefficients are non-zero.

### 3.2. HYBRID SHAPE MOTION SEQUENCE KERNEL

Once an optimal alignment path is obtained between two sequences, we can define a hybrid kernel to take advantage of the strength of the SVM classifications. The Euclidean distance in the RBFs is replaced by the weighted sum of shape distance and DTW distance between two aligned sequences in the reduced dimensional space:

$$K_{hybrid}(\mathbf{x}, \mathbf{y}) = e^{-\gamma(\beta D_s^2 + (1-\beta)(\frac{1}{N}\sum_{k=1}^{N} \|\mathbf{x}_{\phi(k)} - \mathbf{y}_{\psi(k)}\|^2))} \tag{5}$$

where $D_s$ is the shape distance measure and $\beta$ is the weighting parameter. Thus, for $\beta = 1$ the hybrid kernel is dependent only on the shape information. For $\beta = 0$ the hybrid kernel only considers the trajectory information. For $0 < \beta < 1$, the hybrid kernel is based on both shape and trajectory information.

## 4. EXPERIMENTAL RESULTS

Nine gestures are defined in our experiment: "Circle", "Rectangle", "Triangle", letter "P", letter "M", letter "N", letter "W", "Bell" and letter "Z", whose meanings can be interpreted from the name. Each image sequence of the 9 gestures in the experiment has 80 to 120 frames. For each gesture, a total of 500 instances are collected from 25 subjects with 20 performances each with three different hand shapes. The gestures are randomly and disjointly divided into training and test sets at a ratio of $3:1$.

### 4.1. Binary classification of hand gestures

The Hybrid SVM is trained using the $SVM^{light}$ toolbox. For the following experiment the SVM and kernel parameters are set to $C = 2$ and $\gamma = 2$, respectively. Table 1 gives the individual performance of each one-vs-one classifier without considering the shape information ($\beta = 1$). The results show that the proposed method achieves very high recognition rates.

Table 2 gives the recognition performance with shape information taken into consideration ($\beta = 0.15$). Each gesture is performed with three shapes, namely, singer-finger, double-finger and discontinuous double-finger. The same trajectory

**Table 2**. Recognition rates for various gestures with consideration of shape information ($\beta = 0.15$)

| Gesture | Single-Finger Gestures | | Double-Finger Gestures | | Discontinuous Double-Finger Gestures | |
|---|---|---|---|---|---|---|
| | RNN | Hybrid-SVM | RNN | Hybrid-SVM | RNN | Hybrid-SVM |
| **Circle** | 98.3% | 99.1% | 91.3% | 98.6% | 78.7% | 94.1% |
| **Rectangle** | 98.7% | 98.4% | 89.1% | 96.5% | 73.3% | 92.7% |
| **Triangle** | 98.1% | 99.1% | 87.3% | 95.9% | 74.6% | 90.4% |
| **P** | 96.5% | 97.6% | 88.8% | 96.1% | 71.3% | 91.8% |
| **M** | 95.8% | 96.2% | 91.9% | 94.4% | 82.3% | 90.4% |
| **N** | 97.4% | 97.3% | 90.5% | 96.1% | 83.3% | 95.6% |
| **W** | 95.5% | 96.3% | 92.3% | 94.7% | 83.0% | 91.3% |
| **Bell** | 93.5% | 96.8% | 87.7% | 90.3% | 82.5% | 90.1% |
| **Z** | 97.4% | 97.2% | 96.3% | 95.9% | 86.3% | 90.9% |

with different contact shapes are considered belonging to different classes. The results show that the proposed method achieves much higher recognition results than recurrent neural networks (RNN)[6].

### 4.2. MULTI-CLASS EXPERIMENT

This section will show how SVM-based binary classifiers can be effectively combined to tackle the multi-class classification problem. Many studies have proposed ensemble schemes, which use binary classification algorithms to solve $K$-class classification problems [7].

We employ Decision Directed Acyclic Graphs (DDAG) algorithm presented by Platt [8]. In a $K$-class problem, the DDAG tree includes $K(K-1)/2$ nodes which embed binary classifiers between the $p^{-th}$ and $q^{-th}$ pair of classes. The $K$ leaves of the tree represent predicted class labels of unknown sample. The total depth of the DDAG is $K-1$. Given a test sample $x$, starting at the root node, the binary decision function is evaluated. Then it moves to either left or right depending on the output value. Therefore, we go through a path before reaching a leaf node which indicates the predicted class [7]. During training the DAG scheme is necessarily the same as the One Versus One (OvO) scheme producing $K(K-1)/2$ binary classifiers. However, during evaluation, only $K-1$ decision nodes are traversed. This systematic approach of evaluation in DDAG outperforms the evaluation in OvO scheme by a factor of $K/2$ times in speed. The DDAG recognition results are illustrated in the rightmost column of Table 1.

## 5. CONCLUSIONS

We have presented an SVM-based classification method that uses the DTW to measure the similarity. The key contribution of this paper was the extension of trajectory based approach to handle shape information enabling the expansion of the system's gesture vocabulary. It consists of two steps: converting a given set of frames into fixed-length vectors, and training an SVM from the vectorized manifolds. Using shape information not only lets us discriminate further among various gestures, it also allows us to classify gestures that cannot be characterized solely based on their motion information thus boosting overall recognition scores. Tests show the proposed method yielded satisfying recognition rates on a test set.

## 6. REFERENCES


[1] V.I. Pavlovic, R. Sharma, and T.S. Huang, "Visual interpretation of hand gesturs for hci," July 1997.

[2] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[3] Francesco Camastra, "A svm-based cursive character recognizer," *Pattern Recognition*, vol. 40, pp. 3721–3727, December 2007.

[4] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt, "On-line handwriting recognition using support vector machines - a kernel approach," in *Proc. of the 8th IWFHR*, 2002, pp. 49–54.

[5] B. Scholkopf, C. Burges, and A.Smola, MIT Press, 1998.

[6] Y. Yuan, Y. Liu, and K. Barner, "Tactile gesture recognition for people with disabilities," in *ICASSP 2005*, 2005, vol. 5, pp. 461–464.

[7] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, pp. 415–425, Feburary 2002.

[8] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multicalss classification," in *Advances in neural information processing systems*. 2000, vol. 12, pp. 547–553, MIT Press.