ACOUSTIC BASED SKETCH RECOGNITION

A Thesis

by

WENZHE LI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2012

Major Subject: Computer Science

ACOUSTIC BASED SKETCH RECOGNITION

A Thesis

by

WENZHE LI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Tracy Hammond
Committee Members,	Thomas Ieorger
	Tony Cahill
Head of Department,	Duncan Walker

August 2012

Major Subject: Computer Science

ABSTRACT

Acoustic Based Sketch Recognition. (August 2012)
Wenzhe Li, B.S., Nankai University
Chair of Advisory Committee: Dr. Tracy Hammond

Sketch recognition is an active research field, with the goal to automatically recognize hand-drawn diagrams by a computer. The technology enables people to freely interact with digital devices like tablet PCs, Wacoms, and multi-touch screens. These devices are easy to use and have become very popular in market. However, they are still quite costly and need more time to be integrated into existing systems. For example, handwriting recognition systems, while gaining in accuracy and capability, still must rely on users using tablet-PCs to sketch on. As computers get smaller, and smart-phones become more common, our vision is to allow people to sketch using normal pencil and paper and to provide a simple microphone, such as one from their smart-phone, to interpret their writings. Since the only device we need is a single simple microphone, the scope of our work is not limited to common mobile devices, but also can be integrated into many other small devices, such as a ring. In this thesis, we thoroughly investigate this new area, which we call *acoustic based sketch recognition*, and evaluate the possibilities of using it as a new interaction technique. We focus specifically on building a recognition engine for acoustic sketch recognition. We first propose a dynamic time warping algorithm for recognizing isolated sketch sounds using MFCC(Mel-Frequency Cesptral Coefficients). After analyzing its performance limitations, we propose improved dynamic time warping algorithms which work on a hybrid basis, using both MFCC and four global features including skewness, kurtosis, curviness and peak location. The proposed approaches provide both robustness and

decreased computational cost. Finally, we evaluate our algorithms using acoustic data collected by the participants using a device's built-in microphone. Using our improved algorithm we were able to achieve an accuracy of 90% for a 10 digit gesture set, 87% accuracy for the 26 English characters and over 95% accuracy for a set of seven commonly used gestures.

ACKNOWLEDGMENTS

My greatest thanks to the members of the Sketch Recognition Lab for their continued support and help in the research work covered in this thesis. This thesis would not have been possible without their support. In addition, I would like to give extra thanks to my advisor Dr. Tracy Hammond, as well as to my committee members Dr. Thomas Ieorger and Dr. Tony Cahill for their valuable sage advice.

Partial of this thesis is based on author's prior work "Recognizing T ext Through Sound Alone" [27] that is published by AAAI.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Keyboard and mouse are traditional interacting tools that are being widely used in daily life. The interfaces require users to explicitly manipulate the objects on the screen through keyboard input and button presses. Although these types of interactions work well in many environments, they have a poor mapping to visual mediums.

Modern interaction techniques provide a more natural way for humans to communicate with computers. Sketch recognition is one such growing field that allows people to interact with devices using multi-touch surfaces. These devices typically require users to use their stylus or fingers to draw on the screen as if using normal pen and paper. The computer can then automatically recognize hand-drawn shapes or diagrams. One important advantage of using sketch-based systems is that people can continuously interact with computers and get instant feedback from them, making them especially useful for engineering design.

While these interaction techniques are easy to use and becoming very popular, they are still quite costly and need more time to be integrated into existing systems with high accuracy. For example, handwriting recognition systems, while gaining in accuracy, rely on users to use a tablet-PC to sketch on. As computers get smaller and smart-phones become more common, our vision is to allow people to sketch using a normal pencil and paper and to provide a simple microphone, such as one from their smart-phone to interpret their writings. We call this new field of research *acoustic based sketch recognition.*

The journal model is *IEEE Transactions on Automatic Control.*

(a) Wacom display (b) Tablet PC

Fig. 1.: Examples of pen-based devices. (a) Wacom display. (b) Tablet PC.

A. Sketch Recognition Systems

Sketch recognition systems enable people to interact with computers through a multitude of pen-based input devices such as a Wacom or a tabletPC, which are shown in Figure 1. These devices collect pen-data as a series of time-stamped coordinate points in (x, y, t) tuples. Points are collected as soon as the pen is pressed down onto the digitizing or touch-sensitive screen, and the recording stops once the user lifts his her pen.

Sketch-based systems are useful for engineering design and education. Figure 2 shows a system called *Mechanix*, which has been deployed in a freshman engineering class [48]. The system can recognize hand-drawn truss and free-body diagrams, and provide interactive feedback to students to help them solve homework problems. The system works by comparing the correct instructor drawn answers already stored in the database to the one given by the student. By using this system, instructors or TAs only need to draw the correct answers and store them in the database, which greatly reduces their workload.

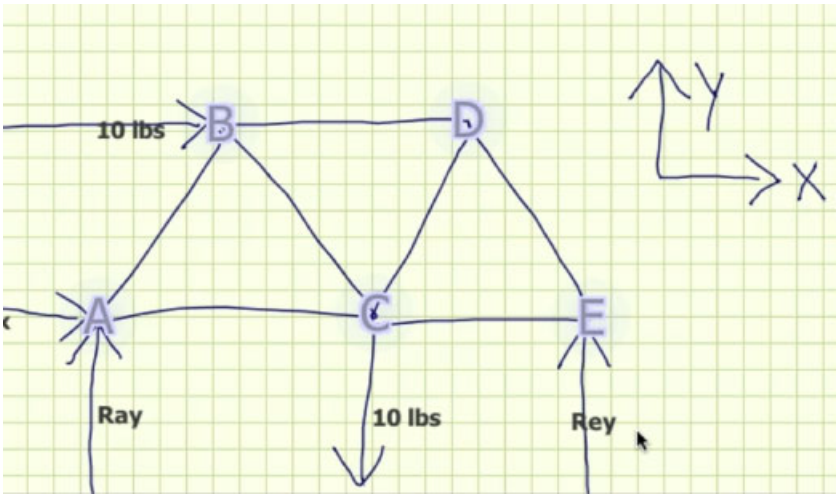


Fig. 2.: Mechanix: A sketch-based tutoring system for statics courses [48]. The system recognizes hand-drawn truss and free-body diagrams and is already deployed in freshman engineering classes at two different universities.

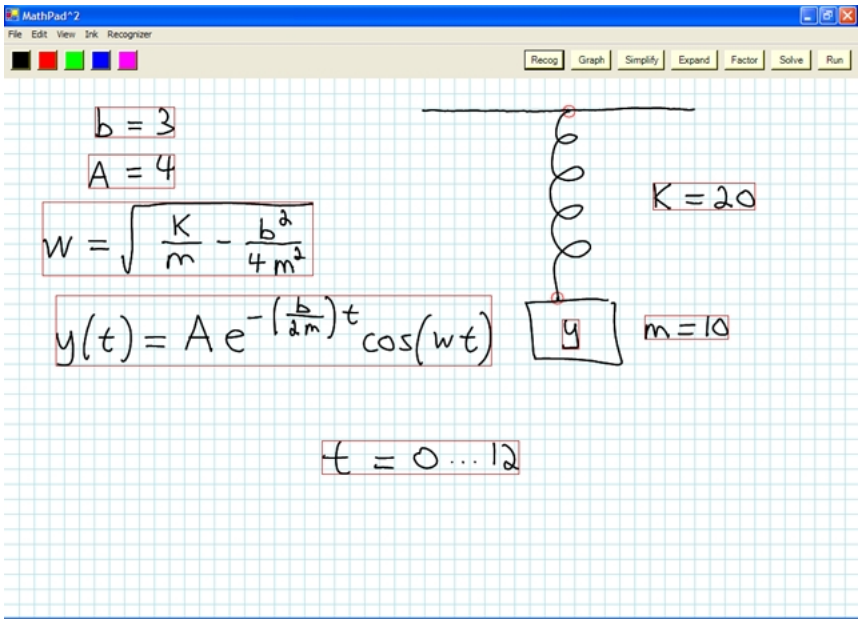


Fig. 3.: Mathpad: a system for the creation and exploration of mathematical sketches [26].

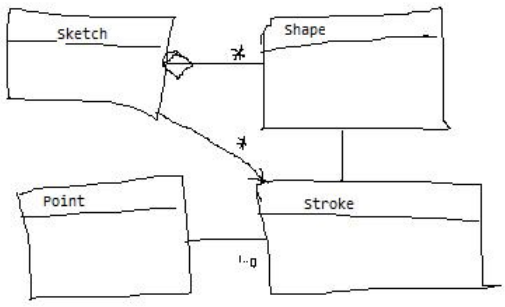


Fig. 4.: Tahuti: A Geometrical Sketch Recognition System for UML Class Diagrams [14].

Another good example is that using Microsoft office to edit mathematical expressions or diagrams is painful. Sketch recognition techniques are especially useful for these domains. MathPad [26], which is shown in Figure 3, is a sketch-based system that is able to recognize hand-drawn mathematical expressions. Using such a system, people do not need to switch back and forth to edit formulas using buttons, which can be both time-consuming and non-intuitive. UML class diagram design [14] (shown in Figure 4) is another example that shows the power of sketch recognition techniques.

Sketch-based systems can be easy to use and are becoming more common to most people. People can interact with them as if they were using a pen and paper. However, these devices are still quite costly and need more time to be integrated into existing systems. For example, handwriting recognition systems, while gaining accuracy, rely on users to sketch on a tablet-PC. In addition, in order to interact with these devices, we need a screen of suitable size. If the screen size is too small, it is hard to sketch on. Thus, as computers get smaller, and smart-phones become more common, our vision is to allow people to sketch using a normal pencil and paper and to provide a simple microphone, such as one from their smart-phone to interpret

Fig. 5.: Device built-in microphone. It is very cheap, just for few dollars!

their writings. We called this technique *acoustic based sketch recognition*, since our intuition comes from sketch recognition research.

B. Acoustic Based Sketch Recognition

First, let's give our formal definition:

Acoustic based sketch recognition is a technique to recognize a sketch through sound.

Using acoustic based sketch recognition, the only device we need is a simple microphone, which is shown in Figure 5. This new field of research provides new possibilities as the scope of its usage is not only limited to the common mobile devices, but can also be integrated into many other devices that do not contain a screen, since the interaction is done only through the microphone. The technique is very cheap, and one can buy a normal microphone for only a few dollars on Amazon.com. In terms of potential applications, the technique can be used by a small device such as a wrist watch, shown in Figure 6, the interaction modality may be useful for building multi-model systems that use both vision and sound to get higher accuracy.

In this thesis, we only use a single microphone. By using multiple microphones, it is possible for the system to be able to localize the sound source by measuring the

Fig. 6.: One example application of acoustic based sketch recognition technique - wrist watch.

time difference to reach the target [1]. However, in that case, we have to deal with the configuration problems before using the system every time the microphones are moved. This makes such a design impractical or at least less practical than using a tablet. Using one microphone is a potential solution that makes the system both cheap and simple.

One might ask the question: "Is it possible to recognize a sketch through sound alone?" It depends. If one wants the system to recognize large set of gestures only through the sound, it might be very hard. But on the other hand, if one wants the system to recognize limited number of common gestures, we show in the thesis that it is quite possible. The intuition of why this works comes from a well-known children game shown in Figure 7, where person A uses a fingernail or key to sketch certain shapes or word on a table without allowing person B to watch it. Then, person B tries to guess what person A has drawn by carefully listening to the sound that A has made. This game shows the feasibilities of our technique from an interesting point of view.

As far as we know, limited work has been done in this area. And until now, there

Fig. 7.: Guess what is being sketched!

have few, if any, well-studied algorithms or applications created. The fundamental problem here is to build a robust recognizer for acoustic sketch sound. At a first glance, the problem may look similar to the traditional speech recognition problem, since they both use the acoustic signal for recognition. However, our problem is much harder not only because it is greatly affected by environmental noise, but also because users writing styles and changes of drawings significantly affect the quality of the sound. People may write the same shape differently each time, changing the sound signal significantly. Additionally, some features that are useful for speech recognition do not work for our problems.

Keeping these problems in mind, we simplify the problem so that we make initial progress in the area by making the following assumptions. 1) We only allow for consistent background noise, i.e, the person should not be speaking at the same time. 2) The system is user dependent and must be trained a data from a particular user before use. This is a common first step in the development of new human action

recognition technology. After solving the user dependent case, we hope to be able to move into more complicated use independent case. 3) We define the gesture set only containing reasonable number of gestures (specifically looking at gesture sets of size $7 \sim 26$).

C. Contributions

We make the following contributions in this work:

- We propose a novel and cheap solution for interacting with devices. The solution can be used for any device with or without a screen. The idea is novel and interesting, which can be explored more in the future work.
- We propose robust dynamic time warping algorithms for sound recognition. The proposed methods may be applied to many other domains to solve practical problems.
- We propose a novel feature set for effectively summarizing the major properties of sketched sound. We show that by combing traditional time series features with our global new features, it is possible to both improve system performance and reduce the computational cost.

D. Organizations

The subsequent chapters are organized as follows: We first describe what has been done in terms of prior work. We then propose a dynamic time warping algorithm for recognizing sound. We show two disadvantages of using template-based approach, those of computational cost and sensitivity to the noise. In order to cope with these problems, we propose improved versions of dynamic time warping algorithms which

are more robust to the noisy data. We come up with four novel global features, including skewness, kurtosis, curviness and peak location. The algorithms work by taking a probability measure into account using these features. To measure the probability, we use a copula classifier, which provides great flexibility by modeling each marginal density allows for any distribution, not just gaussian. We then show its advantages over using a quadratic classifier. Finally, we evaluate our algorithms using acoustic data collected by the participants using a device's built-in microphone. Using our improved algorithm we were able to achieve an accuracy of 90% for a 10 digit gesture set, 82% accuracy for the 26 English characters and over 95% accuracy for a set of seven commonly used gestures.

CHAPTER II

BACKGROUND

In this section, we will review some relevant work and techniques. Since the major part of our algorithm is based on template matching, we begin this section with basic background of the template matching technique. Then we will review some common techniques which used for recognizing acoustic sound.

A. Previous Work

Scratch Input [16] is probably the earliest work that is closely related to ours, and can be considered as seminal work in this area. This work shows the feasibility of recognizing various scratch gesture made on physical surfaces (see Figure 8). Scratch Input is an acoustic-based input technique that relies on the unique sound produced when a fingernail is dragged over the surface of a textured material; it employs a digital stethoscope for sound recording. They conducted a study that shows users can perform six different gestures, with the system obtaining about 90% accuracy. They employed a shallow decision tree primarily based on peak count and amplitude variation. The simple algorithm recognizes six different gestures. Their use of a stethoscope ensures high quality sound. To contrast our method, we use only a standard PC microphone, which can contain lots of noise. We chose to use only a standard microphone to allow for more flexibility and greater ease of use.

Kim [25] developed a system to detect the gesture drawn on non-electronic surface. Their recognition method requires the use of a special textured pad which produce different sound depending on the direction that the user scratches, combined with a wrist mounted microphone (see Figure 9). Their system is designed for small input surfaces, such as the back of an mp3 player. Their system is able to detect

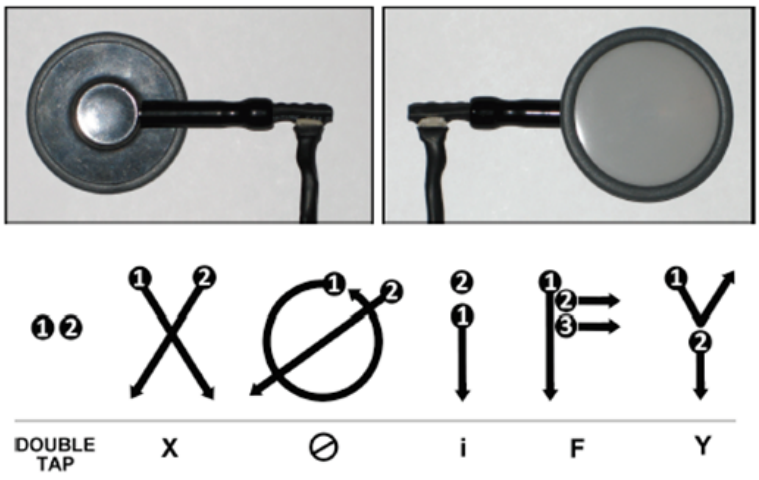


Fig. 8.: Scratch Input: Creating Large, Inexpensive, Unpowered and Mobile finger Input Surfaces. Scratch Input is an acoustic-based input technique that relies on the unique sound produced when a fingernail is dragged over the surface of a textured material, such as wood, fabric, or wall paint [16].

single-stroke representations of Arabic numbers, triangles, rectangles, and bent lines. The authors also outline ways to create wearable input surfaces such as the described wrist-mounted textured pad. One major disadvantage of their system is that people need a special textured pad to sketch on, comared to our system which works on any surface.

Seniuk [41] analyzed the feasibility of recognizing handwritten cursive text through an analysis of acoustic emissions, shown in Figure 10. Their recognition algorithm can recognize isolated lowercase cursive characters and a sketch whole words. They evaluated their approaches on data set collected from 9 users and showed 70% accuracy for the 26 written of the cursive alphabet, and 90% accuracy for a selected of 26 whole cursive words. For whole word recognition, they did not apply any segmentation method, instead, their method needs to be trained and compared to the entire

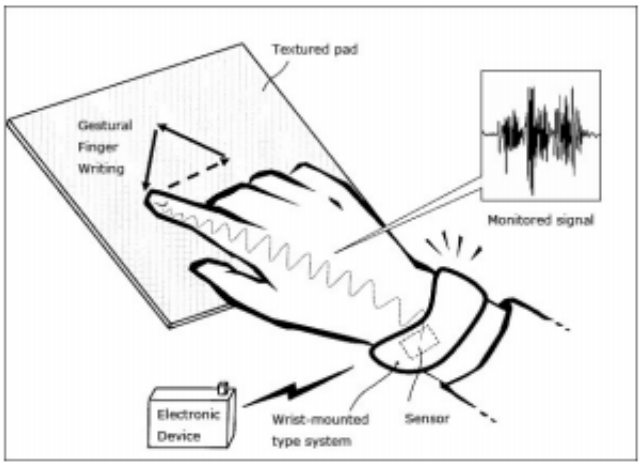


Fig. 9.: A system that allows for gestural scratch input on a textured pad using a wrist mounted microphone [25]

word. Given that the 26 words as of various length, it is not surprising that their accuracy increased dramatically, since length of drawings time will be a powerful feature. However, given the number of similar lengthed words in the English language, this method will have significant scalability issues.

Another very different, but interesting work that also uses sound for recognition is that of Gupta et al [13]. They use the doppler effect to recognize hand gestures (Figure 11). The system allows people to use hand gestures to do simple tasks like navigating web pages. The cool part of their system is, instead of using a camera to recognize hand gestures, it uses sound to recognize them. The concept is rather simple: moving hands in front of the microphone can affect certain frequency range of the sound, which is out of range of human hearing (Figure 12). Their system works pretty well in a noisy environment, but currently only has the capability to recognize a handful of gestures. In fact, our system work in different manner.

Fig. 10.: Pen Acoustic Emissions for Text and Gesture Recognition [41]

Fig. 11.: SoundWave uses the doppler effect to sense gestures [13]

.

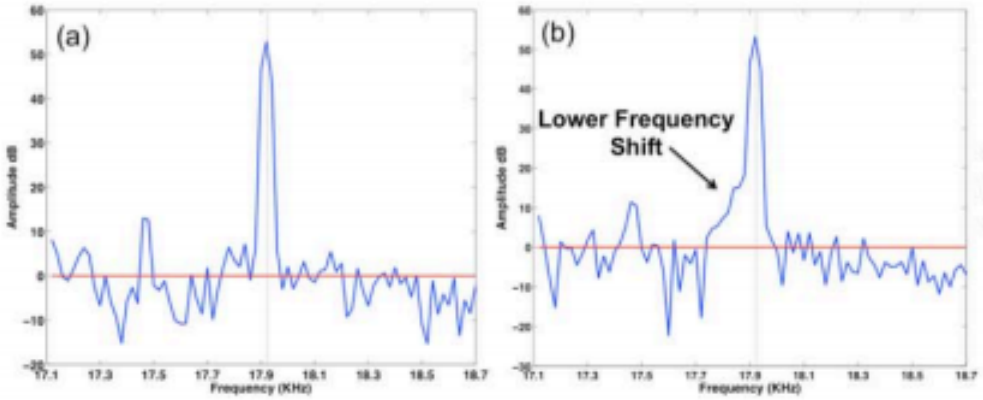


Fig. 12.: Doppler effects. (a) the figure shows the frequency when there are no hand movements. (b) shows the frequency change after making hand movements. It is easy to notice that the frequency within the certain range has changed.

B. Nearest Neighbor Instance-based Pattern Recognition

Instance-based recognition methods, such as the k-nearest neighbor algorithm, are widely used in sketch and speech recognition. In instance-based learning, samples are classified by comparing it with other samples that are already stored in the database. Because classification is performed by direct comparison of the gesture in the training set, the system can easily scaled by simply adding new gesture to our training set. Thus, scalability is a major advantage of instance-based learning, since the model does not have to be retrained. However, instance-based learning (or lazy learning) has two significant disadvantage, that of computational cost (since the running time increases as the size of our training set increases) and that of sensitivity to noise. We will discuss both of these issues in Chapter 3 in more detail.

Formally, each sample s_i can be written as $s_i = \{p_1, ..., p_{n_i}\}$, where n_i is the number of features s_i has. The major computation for nearest neighbor approach is the distance measure between two samples (objects). There are several commonly

used distance metrics.

1. Euclidean Distance

Euclidean distance is one of the most widely used distance metrics. Euclidean distance between point p and q is the length of the line segment connecting these two points in Euclidean space. Assume $p = (p_1, ..., p_k)$ and $q = (q_1, ..., q_k)$ are two points in Euclidean k-space, then the distance is given by:

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + ... + (q_k - p_k)^2} = \sqrt{\sum_{i=1}^{k} (q_i - p_i)^2} \tag{2.1}$$

In two dimensional Euclidean space, the distance between two points can be calculated as $d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$.

Now, let's back to our problem where we use Euclidean distance as distance metric. Assume we have two sequence of points A and B such that $A = (a_1, ..., a_n)$, $B = (b_1, ..., b_n)$. By using Euclidean distance metric, the distance between two objects A and B can be written as :

$$D(A, B) = \sum_{i=1}^{n} d(a_i, b_i) \tag{2.2}$$

where a_i, b_i are the points in Euclidean space, and $d(a_i, b_i)$ can be calculated using equation (2.1).

This simple distance measure is powerful tool for many pattern recognition problems. $1[50] and $N[2] recognizers use this method. $1 recognizer can recognize single stroke gesture and $N recognizer can recognize multi-stroke ones, both of which have high accuracy rate. One thing we should notice is that in sketch recognition and image classification, each sample is represented as sequence of points (or pixels) in 2-dimensional space. In order to calculate the distance between two objects in two dimensional space, we first need to align them together, which is one of the most

difficult problems. This problem can be solved by iteratively rotating one shape by a certain angle at a time, relative to another one, until we find an optimal angle. This approach was used in [50, 2]. Another approach might use some optimization techniques to directly find the optimal rotation angles. Some relevant work can be found in [12, 18, 7]. Due to the complexity of finding such optimal rotation angles, some researchers try to represent samples using rotation invariant features such as moment invariant [20, 5, 28, 19] and eigenvalues of covariance matrix [43].

2. Mahalanobis Distance

Euclidean distance treats all dimensions equally. However, since the dimensions often represent feature values, and since different feature values can have markedly different variances and correlation, it is important to take this into account. Mahalanobis distance takes into account the covariance of the various features. We assume the point p and q are drawn from a group of values with mean $\mu = (\mu_1, ..., \mu_k)$ and covariance matrix S. Then the mahalanobis distance between these two points can be written as:

$$d(p, q) = \sqrt{(p - q)^T S^{-1} (p - q)} \tag{2.3}$$

One may easily notice that if the covariance matrix is the identity matrix, the Mahalanobis distance simply reduces to the Euclidean distance. And, if the covariance matrix is diagonal, it reduces to the normalized Euclidean distance which can be written as:

$$d(p, q) = \sqrt{\sum_{i=1}^{k} \frac{(p_i - q_i)^2}{s_i^2}} \tag{2.4}$$

where s_i is the standard deviation of the p_i and q_i over the sample set. For sound recognition, most algorithms involve the extraction of multiple features from each time window (in our work, we extract 12 features from each time window), in which

case the mahalanobis distance metric becomes especially useful.

3. Hausdorff Distance

Hausdorff distance is another important distance metric that has been widely used to evaluate image-template matches. The formal definition is shown in (2.5), where A and B denote sequence of points. As we can see from the equation, the distance between sets of points A and B is calculated as the maximum of the minimum neighbor distances between a point $a \in A$ and all points in B. In other words, $h(A, B)$ finds the maximum distance bound for a point $a \in A$ to be away from a point $b \in B$.

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b|| \tag{2.5}$$

Huttenlocher et al [21] use the Hausdorff distance model to compare complex 2D images. Kara and Stahovich [24] also use Hausdorff distance to recognize hand-drawn shapes.

4. Objects With Different Length

When two objects have the same number of points, the distance can be calculated using one-to-one direct mapping between each pair of points. But in most of cases, two objects may have different number of points. In general, there are two ways to cope with this problem.

1. **Normalization**. We can first make the length equal. In sketch recognition, this process is called resampling. In [50], they resample the points by making two adjacent points have the equal distance. This resampling process can remove the effects from different devices and shown to improve the recognition accuracy. In speech recognition, the sound is usually represented as time series data, in which case we can normalize the signal through the time axis by using transformation (e.g.,Z-

transformation).

2. **Dynamic Time Warping**. This technique makes the matching possible for two objects with different length. Instead of doing one-to-one mapping between each pair of points, dynamic time warping (DTW) allows the mapping become rather flexible. For example, one point can map to multiple points on another object, and vise versa. When we design a dynamic time warping algorithm, one important thing we need to do is to appropriately define some mapping constraints. We will show more design details in the subsequent chapters.

C. Hidden Markov Model

Beside template-based approach, hidden markov model(HMM) is another widely used approach in speech community. HMM is a powerful tool for modeling time dependent signal. Figure 13 shows the structure of this model. As shown in [37], there are two strong reasons why we use HMM. First the models are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications. Second the models work pretty well in practice for many applications.

Mathematically, we can represent HMM as $\lambda = (A, B, N, M, \pi)$.

N : the number of states in the model $S = s_1, ..., .s_N$

M : the number of discrete observation symbols $V = v_1, ..., v_M$

A : $A = \{a_{ij}\}$, the state transition probability

$$a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$$

B : $B = \{b_j(k)\}$, the observation or emission probability distribution

$$b_j(k) = P(o_t = v_k | q_t = s_j)$$

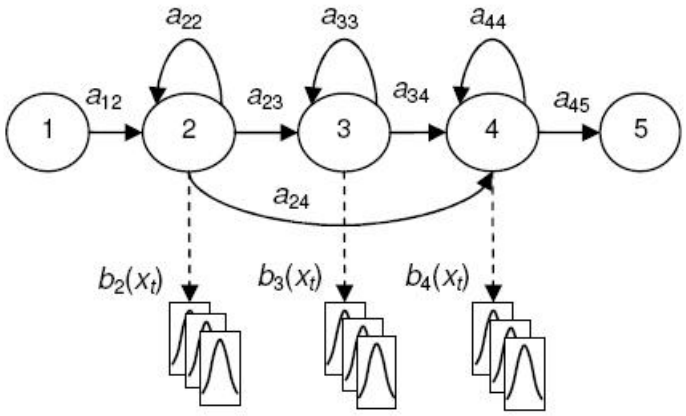


Fig. 13.: Hidden Markov Model

π : the initial state distribution

$$\pi_j = P(q_1 = s_j)$$

For compact representation, we can write the model as $\lambda = (A, B, \pi)$. In order to apply HMM to the practical problems, we need to solve three major problems.

- Problem 1. Given observation sequence $\{O = o_1, o_2, ...\}$ and model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the likelihood of the observation sequence given the model? The solution is given by the Forward and Backward procedures.

- Problem 2. Given observation sequence $\{O = o_1, o_2, ...\}$ and model λ, how do we choose a state sequence $Q = \{q_1, q_2, ...\}$ that is optimal? We can use Viterbi algorithm to solve this problem.

- Problem 3. Parameter estimation. The problem is to estimate the model parameters given observations. This problem can be solved by finding the parameters which maximize the likelihood $P(O|\lambda)$. We can use Baum-Welch re-estimation procedure.

More information about HMM can be found in [37]

CHAPTER III

PROBLEM

The fundamental problem is to recognize the acoustic sound. The recognition process is pretty straightforward. We use our microphone (either from Mac or mobile phone) to record the sound while people are sketching shapes on the table. As soon as people finish sketching, our recognizer tempt to recognize what people have drawn.

The core idea of our algorithm is instance-based learning, by which we compare new sample with all other candidate samples already stored in the database. We record each sound and saved it in a ".wav" file. The sampling rate we chose is 8kHz, and each sample is 16bits, and we use mono channels. The reason for choosing lower sampling rate is for the purpose of reducing computational cost. However, lower sampling rate might reduce the recognition accuracy, since it contains less information comparing to the sound with high sampling rate. But we finally figure out that 8kHz sampling rate works pretty well in practice while only affecting the performance a little bit.

Now, let's give formal definition of our problem. All the notations will be used frequently throughout the thesis. We use $\mathcal{W}$ to denote the set of candidate samples stored in the database, such that $\mathcal{W} = \{w_1, ...w_n\}$, where w_i is ith sample. We also have a label set and denote it as $\mathcal{C}$, such that $\mathcal{C} = \{c_1, ..., c_k\}$, where k is the number of labels in $\mathcal{C}$. Each sample w_i is associated with a label c_i choosing from the label set, $\mathcal{C}$. Our goal is to assign the most suitable label to a new query sample w_{new}, given all the data (or training samples) which is already stored in the database.

This is a classical machine learning problem, which we will focus on in the subsequent chapters. Please note that we call each sound file a *sample*.

CHAPTER IV

DYNAMIC TIME WARPING

In this chapter, we will look at the problem of recognizing isolated acoustic sketch. As we've shown before, each sound was recorded and saved in a ".wav" file. Before trying to recognize it, we first preprocess the sound to make it contain less noise. After that, we can extract feature values from each sound. Finally, the recognition algorithm can be applied to get the final result.

A. Preprocessing

The sound is recorded using device built-in microphone. Thus, it is likely that it contains some noise which can affect the system performance significantly. In order to make the problem easier, we only assume that there is consistent environmental noise, which stays almost the same over the time. The preprocessing contains noise reduction and silence detection steps. We will show them in detail.

1. Noise Reduction

Noise reduction is required even when we expect our working environment to be relatively quiet. Those noise either comes from environment or device itself (but anyway, we assume it is consistent). Let's first review some typical approaches for removing noise.

- *Energy Based Approach.* This may be the simplest approach that we can apply. It works as follows: given acoustic signal S, we assume the signal from the first 10 to 100 milliseconds is noise and sample it. Then we calculate the average energy for this 'noisy frame', denoted it as $\hat{\mu_{noise}}$. After that, we scan through

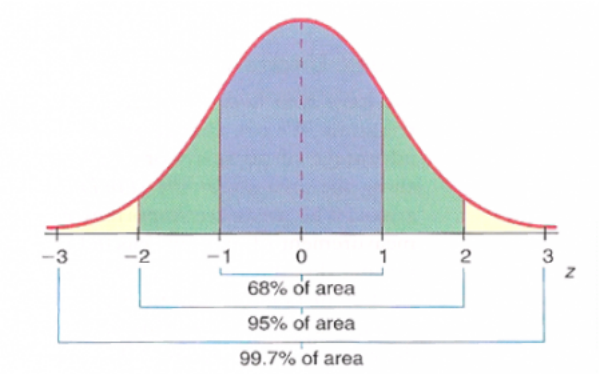


Fig. 14.: Gaussian distribution with zero mean and unit standard deviation. The figure shows that the probability of the value resides between ± three times standard deviation is over 99.7%.

.

the remaining body of signal by examining one frame at a time. If the average energy $\overline{\mu}$ for that frame is larger than $c\hat{\mu_{noise}}$, such that $\overline{\mu} > c\hat{\mu_{noise}}$, we mark that frame as 'noisy'. Here, c is the constant threshold.

- *Gaussian Noise Removal.* It is similar to the *energy based approach.* The only difference is that we explicitly model the noise using gaussian distribution, $G_{noise} \sim N(\hat{\mu_{noise}}, \hat{\sigma^2_{noise}})$. From the beginning 10 to 100 millisecond signal, we calculate the $\hat{\mu_{noise}}$ and $\hat{\sigma_{noise}}$. Then we scan through all the remaining frames and calculate the average energy μ_i, then we use the formula $|\mu_i - \hat{\mu_{noise}}|/\hat{\sigma_{noise}} < 3$ to decide whether it is valid frame or noisy frame. The threshold 3 corresponds to 99.7% in normal distribution, which is shown in Figure 14. In [8], they showed that this simple approach works well in practice. However, this approach sometimes turns out to mark the valid frame as 'silence'. This is simply because the silence within the acoustic sketch signal is not necessarily the noisy signal in our context.

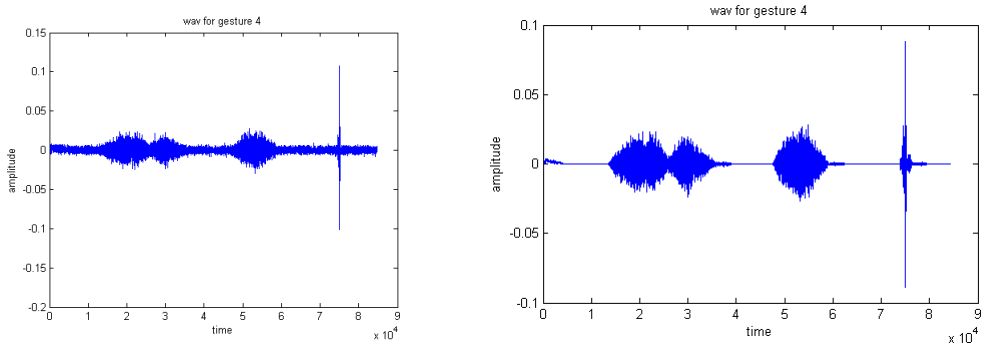
- *Fourier Based Approach.* The idea comes from fourier transformation. Analyzing the signal in frequency domain provides much information about the signal itself. The approach works as follows: 1. We first transform the original signal into frequency domain. 2. Analyze the signal in frequency domain and find out the frequency range where most of noise resides. 3. Set all the corresponding frequency values to 0, then transform back to the original time domain using inverse fourier transformation. Besides fourier transform [15], wavelet transformation [54] is another good technique for filtering the noise out.

- *Spectral Subtraction.* This is an old technique, but works very well for removing background noise. It is proposed by [4]. The algorithm offers computationally efficient, processor-independent approach to remove consistent background noise. It suppresses stationary noise from speech by subtracting the spectral noise calculated from non-speech frames. Then the second step is applied to attenuate the residual noise left after subtraction. We have applied Boll's method to our work, and Figure 15 shows the performance of spectral subtraction method. Figure (a) shows the original signal before noise reduction, which we got from the microphone recording. Figure (b) shows the clean signal after we apply Boll's spectral subtraction method.

2. Silence Detection

After removing the background noise, the next thing we need to do is to remove silence parts of the signal. We denote each silence part as "invalid" signal. One simple way to remove all the silence parts is to use gaussian noise removal [8]. It works as follows:

- We assume the beginning part of the signal as "invalid".



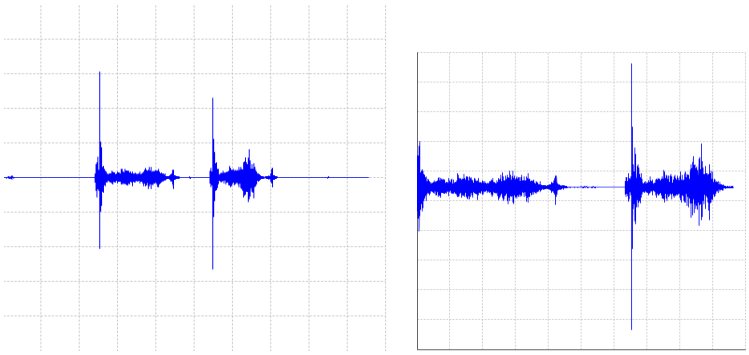
(a) Before (b) After noise reduction

Fig. 15.: Noise reduction using Boll's spectral suppression. (a) shows the original signal before noise reduction. The original signal contains consistent background noise. (b) clean signal after removing the noise.

- Sample those parts of the signal and model it by using gaussian distribution $\mathcal{N} \sim (\mu, \sigma^2)$.
- Mark each remaining frame as "valid" or "invalid" based on $|x - \mu|/\sigma < 3$.
- Discard all the "invalid" frames and concatenate the remaining ones.

This approach can work for traditional speech recognition. However, for acoustic-based sketch recognition, the silence does not necessarily mean that it is invalid signal. One good example is that when we sketch multi-stroke shapes on the table, the signal may contain the silence part when the pen-up action occurs. But in this case, we want to treat them as valid signal without cutting them off. In order to handle this problem, we propose an improved version of silence detection algorithm.

The algorithm only removes the beginning and end parts of the signal, which equivalent to the problem of detecting the end points. The searching procedure starts from the beginning and end of the signal, respectively. The algorithm checks the



(a) Before (b) After silence removal

Fig. 16.: Silence removal for gesture 4. (a) is the original signal before removing silence parts. (b) is the resulting signal after applying our silence removing algorithm

consecutive silence segments each time to calculate the ratio based on equation (4.1).

$$R = \frac{\#\text{of silence segments}}{\text{total\# of segments}} \tag{4.1}$$

By comparing the ratio R with predefined threshold, we can decide whether to stop the search or not. Figure 16 shows how our improved silence detection algorithm works. As we can see in the figure, our algorithm only removes the silence at the beginning and end of the signal.

B. Feature Extraction

1. Several Commonly Used Features

Now we are ready for extracting features from the sound. After carefully looking at the wave of each sound file, we noticed that these sound samples can be distinguished from the shapes of their sound waves. This gives us a clue that using short time fourier transform(STFT) features can help distinguish among them. These features include spectral centroid [9], spectral bandwidth [47], spectral flatness measure [17, 22], shan-

non entropy [32], renyi entropy [3], Mel-frequency cepstral coefficients(MFCC) [29, 52] and etc. Brief introduction of these STFT features are shown in the following [38].

- *Spectral Centroid.* The feature measures the center of gravity of the magnitude spectrum of the STFT and is a measure of spectral shape.
- *Spectral Bandwidth.* The feature measures weighted average of the distance between the spectral components and the spectral centroid.
- *Spectral Flatness Measure.* The features measures the flatness of the spectrum and distinguishes between noise and valid signal.
- *Shannon Entropy.* The feature measures the spectral distribution of the signal.
- *Renyi Entropy.* The feature also measures the spectral distribution of the signal. Renyi entropy is a generalization of shannon entropy.
- *Mel-Frequency Cepstrum Coefficient.* It is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency [30].

2. Mel-frequency Cepstrum Coefficients

Among all these possible options, we finally end up choosing MFCC features for acoustic sound recognition which work well for our problems. MFCCs is one of the most widely used feature set in signal processing, and characterizes the dynamic change of the digital signal [53]. The overall process of calculating MFCCs is as follows:

1. Take the Fourier transform of (a windowed excerpt of) a signal.

2. Map the powers of the spectrum obtained above onto the mel scale using triangular overlapping windows.

3. Take the logs of the powers at each of the mel frequencies.

4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.

5. The MFCCs are the amplitudes of the resulting spectrum.

In actual implementations, there are two major issues we need to consider:

- Choosing the appropriate window size. We set the window size as 256 when we use 8kHz sampling rate, and set the size as 1024 when we use 44.1kHz sampling rate.

- Choosing the appropriate number of coefficient values. We used the first 13 coefficients excluding the first one, since this parameterization has been shown to be quite effective [36].

Note that this feature extraction is essentially corner finding, and searches for periods of slow movement in the stroke. The output of feature extraction is a number of partial strokes that are segmented based on silence/slow movement that likely represents strong corners.

3. Representation

MFCC features are extracted from each time window(as we've shown above, we only chose the first 13 coefficients while discarding the first one, which makes 12 in total), and the number of windows depends on the length of sound signal. Thus, after using MFCC to represent each sound sample, we get multi-dimensional time series

data. Formally we can represent each sample as $\text{MFCC}(w_i) = \{f_i^{[1]}, ..., f_i^{[n_i]}\}$, where $f_i^{[j]} = (f_{i,1}^{[j]}, ..., f_{i,12}^{[j]})$, we use upper case $[j]$ to denote the timing index.

C. Algorithm Design

In this section, we will show our algorithm design in detail.

1. Similarity Measure

The idea is based on similarity measure. Usually we need to calculate the distance between two objects and use that to measure how similar two objects are. One advantage of using instance-based approach is its scalability. You can simply add more samples into the database without changing the core part of algorithm. However, as the number of templates grows, the time complexity grows as well, which is one big disadvantage of instance-based approach. The computational cost generally can be reduced by using parallel computation.

Since we represent each sound sample using MFCC features, calculating the distance between two sound samples equals to calculating the distance between two MFCCs feature vectors. Formally, we can write it as:

$$\textbf{dist}(w_i, w_j) = \textbf{dist}(\text{MFCC}(w_i), \text{MFCC}(w_j)) = \textbf{dist}(\{f_i^{[1]}, ..., f_i^{[n_i]}\}, \{f_j^{[1]}, ..., f_j^{[n_j]}\}) \tag{4.2}$$

where n_i and n_j are the length of feature vectors, $\text{MFCC}(w_i)$ and $\text{MFCC}(w_j)$, which **may not be the same**.

2. Grid Representation

For easy illustration, let's first slightly simplify the notations without changing any meanings. We can formalize the problem as calculating the distance between two

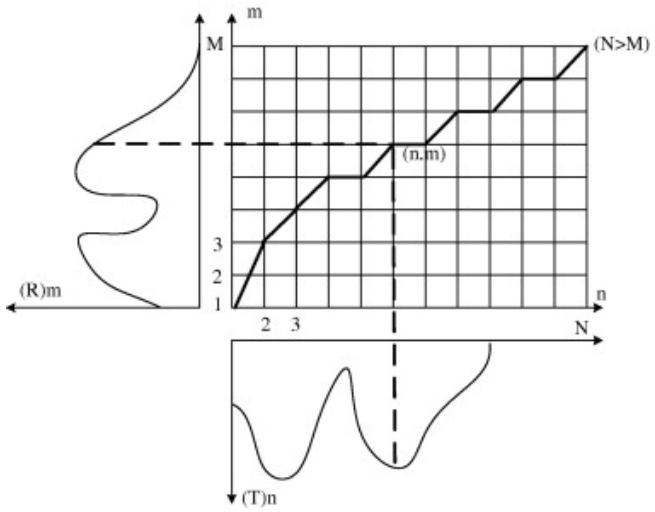


Fig. 17.: Grid representation of dynamic time warping

time dependent sequences, A and B.

$$A = a_1, ..., a_n, \qquad B = b_1, ..., b_m$$

Figure 17 shows the grid representation of designing dynamic time warping algorithm. As we can see from the figure, x-axis represents the signal T, and y-axis represents the signal R, both are time dependent. The goal is to find the distance between these two signals. The problem can be visually interpreted as finding an optimal path from the original coordinate (1,1) to the destination (N,M). Actually, there are lots of possible paths from the original point to the destination, but the path we want to find is the one that leads to the minimum distance between two signals. The figure shows one such possible path. It is easy to see that classical template matching is actually a special case of dynamic time warping, which measures the distance between two objects with different length. In classical template matching problems, the path should strictly follow diagonal direction.

3. Design Constraints

Assume that we want to calculate the distance between query $A = a_1, a_2, ...a_n$ and each template $B = b_1, b_2, ...b_m$. Equation (4.3) shows the formal calculation of dynamic time warping.

$$D(A, B) = \frac{1}{N} min_F [\sum_{k=1}^{K} d(c(k))w(k)] \tag{4.3}$$

where $c(k)$ is the actual mapping between candidate and template at time index k, $w(k)$ is the weighting function, and N is the normalized value. Dynamic time warping will find the optimal path F to minimize $D(A, B)$ [33]. In order to design well performed algorithm, there are several constraints we need to keep in mind.

- *Endpoint Constraint*: The endpoint constraint decides where the mapping starts and ends. We start mapping at $(1, 1)$ and end at (N, M).

- *Local Continuity Constraint*: In order to avoid excessive compression or expansion of the time scales, neither the query nor the template can skip more than two frames at a time when matching. The five possible movements to get to (m, n) are $(m-1, n), (m, n-1), (m-1, n-1), (m-2, n-1), (m-1, n-2)$, which is shown in figure 18.

- *Global Path Constraint.* Because the signal is time dependent, we set the boundary to control for each mapping range at time index k such that $|A(k) - B(k)| \leq R$. We choose R equals to 20.

- *Axis Orientation*: There are two variations. We can either put a candidate on the X-axis and the template on the Y-axis or put a template on the X-axis and a template on the Y-axis. However, [33] shows that putting the query on the

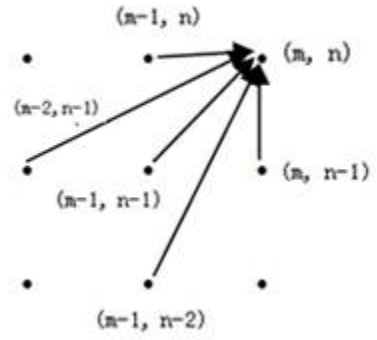


Fig. 18.: Local constraints. This figure shows five possible movements in each location. These possible locations to get to (m, n) are $(m-1, n), (m, n-1), (m-1, n-1), (m-2, n-1), (m-1, n-2)$.

X-axis generally gives better result, thus we adopted that technique for this work

- *Weighting Function*: We choose a symmetric weighting function, $w(k) = A(k) - A(k-1) + B(k) - B(k-1)$, so that the weighting value characterizes how many steps are from the current point to the previous one.

- *Distance Measure*: We use Euclidean distance for our distance measurement. One tricky part is to properly calculate the distance between two MFCCs features, which are multi-dimensional values. There are 12 dimensions for each feature value in MFCCs.

4. Final Form of Algorithm

Finally, we can write our DTW algorithm as follows:

$$D(n,m) = min \begin{cases} D(n-1,m-1) + 2d(n,m) \\ D(n,m-1) + d(n,m) \\ D(n-1,m) + d(n,m) \\ D(n-2,m-1) + 3d(n,m) \\ D(n-1,m-2) + 3d(n,m) \end{cases} \tag{4.4}$$

One thing we want to mention is that the features we extracted from each time window is 12 dimensional feature values. Thus, calculating the distance between two feature vectors is another sub-problem. In fact, we can either use Euclidean distance or Mahalanobis distance metric to calculate it. But after realizing that different coefficient has largely different variance, we decided to use Mahalanobis distance instead, where the covariance matrix is the diagonal matrix (each diagonal entry corresponds to the variance).

At the end of this chapter, we want to show some possible ideas to further improve the current template matching algorithm.

- *Feature weighting.* Since each feature has different degree of importance regarding the prediction accuracy, setting different weight for each feature might be helpful.

- *Control the size and quality of the templates stored in the database.* Storing more templates means can cover more variances from users. However, it also causes the problem that template stored in one class looks similar to the one in another class. Thus, it is important to control the number of templates per class to balance the tradeoff between quality and quantity.

- *Removing noisy templates.* We guess this can be done by clustering techniques

or dimensionality reduction techniques like MDS. The purpose is to remove the outliers which affect the system performance.

CHAPTER V

IMPROVED DYNAMIC TIME WARPING

In the previous chapter, we show a dynamic time warping algorithm for acoustic sound recognition. However, this simple approach suffers from two major problems. In order to make the algorithm more robust, we make some improvements, which is the focus of this chapter.

A. Two Major Problems

Dynamic time warping algorithm suffers from two major problems, in general.

- **Sensitivity to Noise.** Since the method relies on the samples already stored in the database, the quality of the samples will greatly affect the algorithm performance [45]. However there are many potential problems in maintaining high quality samples. Firstly, if the number of samples in the database is large, it makes one hard to analyze the data and pick those with high quality. In fact, the definition of "high quality" itself is ambiguous. If we only choose clean data, it will be likely to lead us over-fitting problems, which makes the training samples no longer good representations of the incoming new samples. But anyway, it is always good to remove the outliers.

- **Computation Cost**. This seems much clear, since we have to compare each new sample with all others already stored in the database. As the number of samples grows, the computational cost grows accordingly. There are generally two ways to handle this problem in classification context. Firstly, it is possible to only choose few prototypes from each category, and consider them as training samples while discarding all the remaining ones, which we call "prototype

selection" [35, 42, 10]. The other way is to reduce the computational cost by improving the search strategy. In this way, we only try to search for subset of space. This approach needs some probability measure to determine where to search and where not.

The major focus of the subsequent sections are for solving these two problems. But before getting into details, let's first look at one simple way to remove the outliers.

B. Removing the Noisy Samples Explicitly

One naive way is to visually check all the samples and to see whether each of them is noisy or clean one. However, as the size of data set becomes larger, it is impossible for us to manually point out them. We have to think about some automated way for finding such noisy samples. In fact, visualization technique provides great potential for such problems. In machine learning community, there are some tools are especially useful for visualizing or summarizing data set, including principal component analysis(PCA) [51], linear discriminant analysis(LDA) [31] and multi-dimensional scaling(MDS) [46]. Besides, boxplot [49] and functional boxplot [44] are both great information exploratory tools that are widely used in statistic community.

PCA and LDA are mainly for the data set which has large number of attributes, while boxplot and functional boxplot are mainly for summarizing the statistical features of the given data sets. After considering both advantages and disadvantages of each technique, we decided to use MDS as our information visualization tool. One important reason for choosing MDS is that it needs to calculate the pair-wise distance matrix. In fact, this distance matrix can be easily calculated using our proposed dynamic time warping approach.

The core idea of MDS is that it transforms the original data set into lower dimen-

sion, which is characterized by new vector basis, to keep the pair-wise relationship between data points.

The computation for MDS is straightforward:

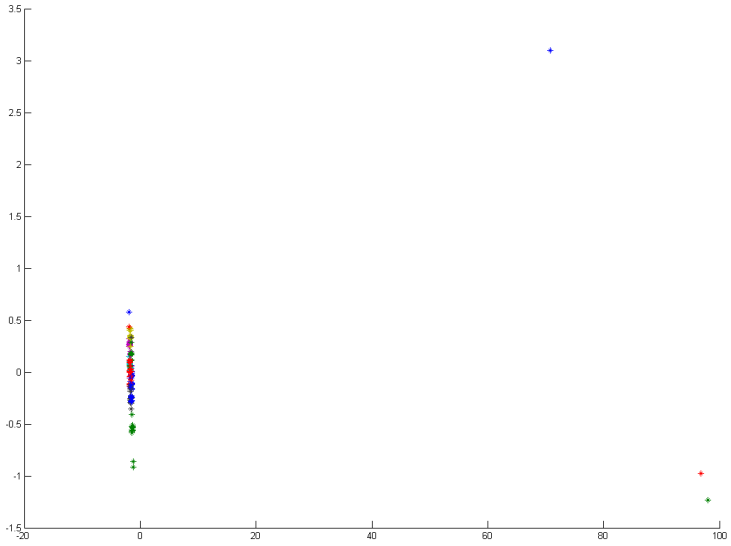
Given $\mathcal{W} = \{w_1, ..., w_n\}$, we first need to calculate the pairwise distance matrix.

$$S_{m,n} = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,n} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & \cdots & d_{n,n} \end{pmatrix}$$

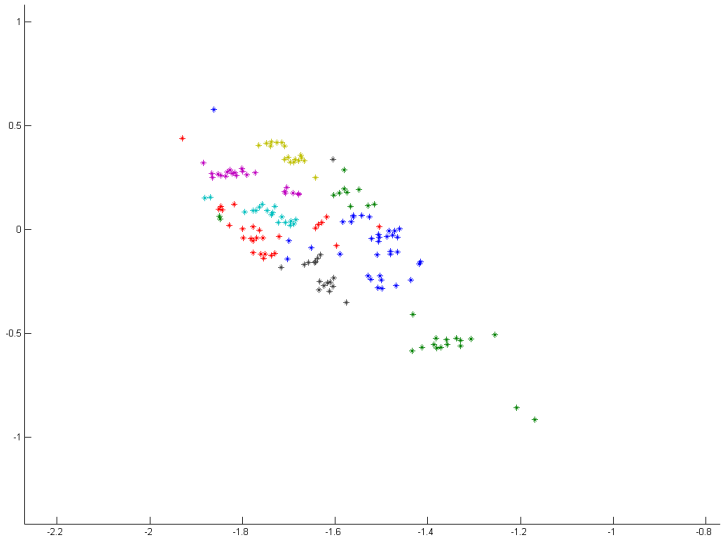
The goal is to find new vectors $x_1, ..., x_n \in \mathbb{R}^N$ such that $||x_i - x_j|| \approx S_{i,j}$ for all i, j, where $d_{i,j} = \mathbf{dist}(w_i, w_j)$. For visualization purpose, we set $N = 2$. By applying optimization techniques, we can easily find out the solution. For more information about MDS, please refer to [46].

In order to clearly show the idea of using MDS to remove the noisy samples. Let's look at the Figure 19. The figure shows the MDS plots for "0-9" gesture set. The samples with the same color mean that they have the same class labels. Figure (a) shows the original MDS plot for the 10 gesture data set while Figure (b) shows the resulting MDS plot after removing some outliers (those outliers can be easily seen on the right side of the figure). But after moving those outliers, it becomes hard to further figure out additional noisy samples. That's the motivation of our subsequent work.

Keeping these problems in mind, we propose to use some strategies to improve the robustness of the recognizer. The first proposed algorithm is based on average distance (more information will be given). Another approach is hybrid basis, which effectively takes probability measure into account. The hybrid solution works by



(a) Original MDS plot



(b) MDS plot after removing the outliers

Fig. 19.: MDS plots for "0-9" gesture set. The samples with the same color have the same labels. Figure (a) shows the original MDS plot. (b) shows the MDS plot after removing the outliers (the most obvious ones).

aggregating results from probabilistic classifier and template based approach. It can:

- Increase the robustness to the noisy samples (outliers)
- Reduce the computational cost by some degree.

C. DTW with Average Distance

The original dynamic time warping algorithm tries to find the best matching candidate sample from the database. Once we guarantee that there is no noisy sample, this algorithm should work well. One may already notice that the proposed algorithm is actually 1-nearest neighbor. The generalization of 1 nearest neighbor is K nearest neighbor [6], where K is the number of nearest samples we need to compare. When we set value(larger than 1) for K, one simple approach is to use majority voting to do classification. There is some tradeoff between choosing larger K and smaller K value. For more information, please refer to some relevant work [6].

The algorithm we proposed is different from the classical K nearest neighbor algorithm. What we are trying to do is to average the distance. More specifically, once new sample w_{new} arrives, we want to know which label we should assign to it. Instead of finding the best candidate sample from the database, we want to calculate the distance between w_{new} and class label i. The equation (5.1) shows the formal definition of distance measure between sample and class label.

$$D(w_{new}, c_i) = \frac{1}{N} \sum_{i=1}^{N} D(w_{new}, w_i), \qquad w_i \in S_{c_i} \tag{5.1}$$

where S_{c_i} is the subset containing all the samples with class label i.

Finally, we assign the new sample with the best class label between which they

have smallest distance. Formally, we can write it as:

$$\text{best label} = \arg\min_i D(w_{new}, c_i), \quad \text{where} \quad i \in \{1, ..., K\}.$$

We will give some experimental results to show how it performs in practice.

D. Hybrid Approach

The next proposed solution is based on the hybrid approach. The core idea is that: 1. Use some probabilistic classifier to calculate the likelihood of a class label given by a new sample. 2. Use these probabilities to filter out some candidate samples and also incorporate these values into distance measure. We can consider the first step as brief classification while considering the second step as refined classification.

In order to build a probabilistic classifier, we propose a new feature set which we call them "global features". These features are inherently different from spectral features like MFCC, since they are timing independent and evolving along the time axis. Thus, the next question is how to extract such global features from each sound?

Again, we use the MFCC features. But instead of using the whole 12 channels(coefficients), we only choose the first channel in this case. We then extract global features from the first channel. The reason is that the first channel contains more general information about the wave shape. This notion is similar to the fourier transform or moments invariant. If we do fourier transformation, the lower order coefficient is the value characterizes the overall properties of the function while the higher order coefficient can provide more refined information about the shape. We actually tested each individual MFCC channel and plot the accuracy graph for our data set, which is shown in Figure 20. From this figure, we can see that the first MFCC channel gives higher accuracy than other higher order channels.

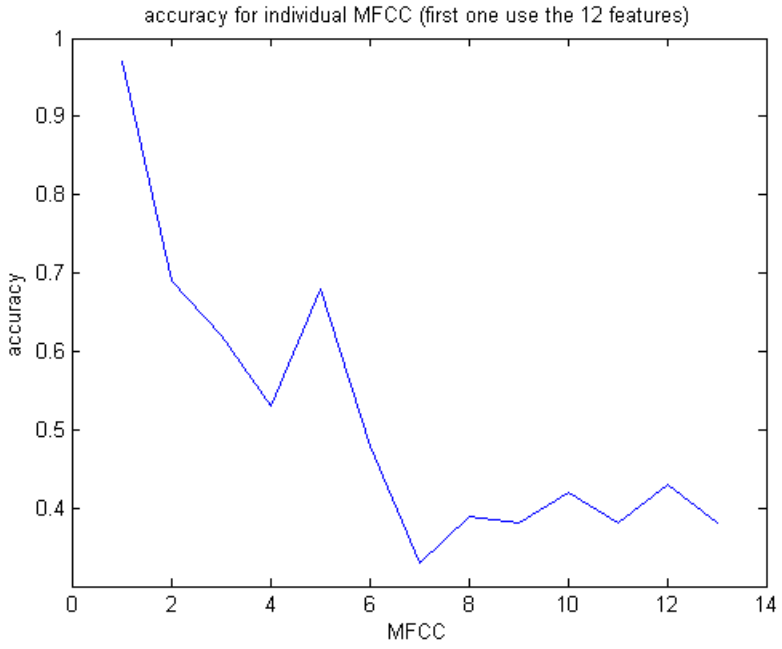


Fig. 20.: Accuracy for individual MFCC channel. Data set we used here is "0-9" gesture set. We can see from the figure that when we use all the MFCC features, we can get over 90% accuracy, and when we use only the first MFCC channel, we can get around 67% accuracy, and so on.

1. Global Feature Extraction

Let's rewrite the representation of a sample using MFCCs. MFCC(w_i) = ($\{f_i^{[1]}, ..., f_i^{[n_i]}\}$), where $f_i^{[j]} = (f_{i,1}^{[j]}, ..., f_{i,12}^{[j]})$. We extract four global features from MFCCs(consists of 12 coefficients in our case). As we've shown before, the first coefficient contains more general information about the sound. For this reason, we extract features from the first coefficient only.

Now, let's only use the first coefficient to represent the sample, we can write it as $\text{MFCC}_1(w_i) = (f_{i,1}^{[1]}, ..., f_{i,1}^{[n_i]})$. Figure 21 shows the plot of first coefficient vector representation for each gesture, "0" to "9". For simple notation, we write this feature vector as $(f^{[1}, ..., f^{[n]})$, by only keeping the timing index. The four features we defined are skewness, kurtosis [23], curviness and peak location.

Skewness. The feature measures the property of symmetry. Defined as:

$$x_1 = \frac{\sum_{i=1}^{n}(f^{[i]} - \overline{f})^3}{(n-1)s^3}$$

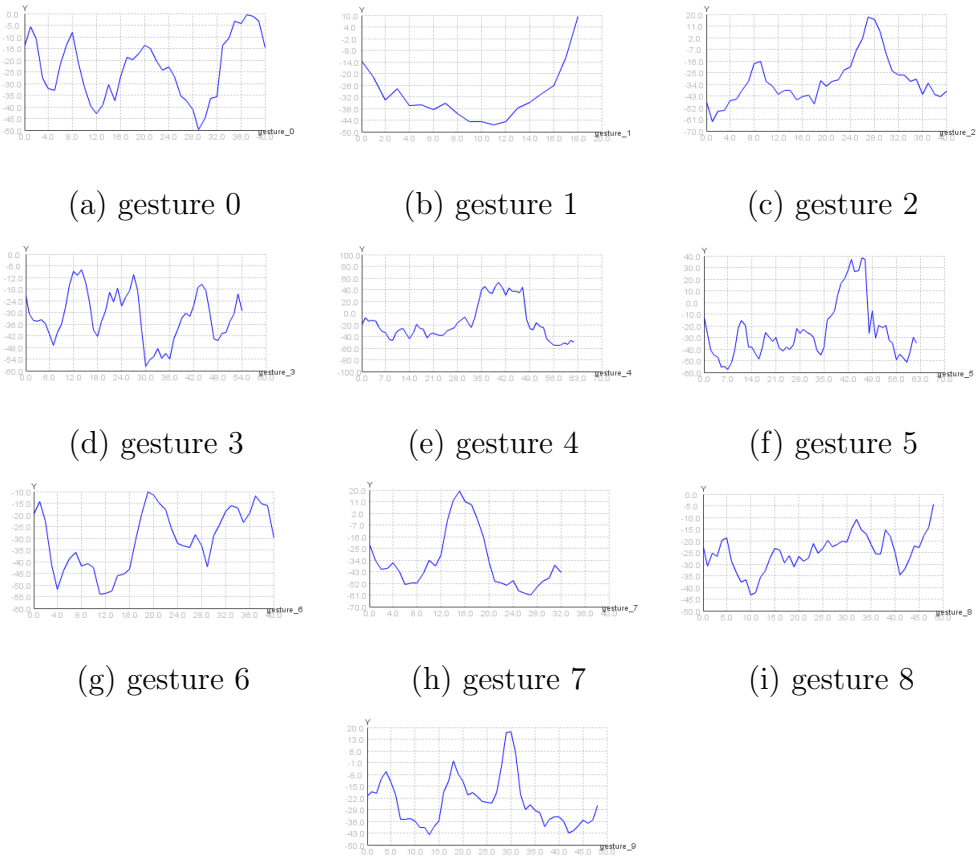
where s is the standard deviation, and $\overline{f}$ is the mean, and n is the number of coefficients. Skewness is one of the most important summary statistics for the distribution. More generally, it is the third order moment for the distribution function.

Kurtosis. Kurtosis refers to the weight of the tails of a distribution, can be defined as:

$$x_2 = \frac{\sum_{i=1}^{n}(f^{[i]} - \overline{f})^4}{(n-1)s^4}$$

where the meanings of the variables $s, \overline{f}, n$ are the same as above.

Curviness. This feature measures the jerkiness of the shape by counting the



(a) gesture 0 (b) gesture 1 (c) gesture 2

(d) gesture 3 (e) gesture 4 (f) gesture 5

(g) gesture 6 (h) gesture 7 (i) gesture 8

(j) gesture 9

Fig. 21.: First channel of MFCCs for gesture "0-9".

local spike. The computation is straightforward:

$$x_3 = \sum_{i=2}^{n-1} \mathbb{1}\{f^{[i]} > f^{[i-1]} \text{ and } f^{[i]} > f^{[i+1]}\}$$

where $\mathbb{1}$ is the indicator function. From Figure 21, we can see that some of them have more jerkiness than others. For example, gesture 1 and gesture 7 have more curvature than others.

Peak location. The feature measures the relative position where the peak appears. This feature can be computed as:

$$x_4 = \frac{\arg\max_i\{i|\forall j : f^{[j]} \leq f^{[i]}\}}{n}$$

This feature is another intuitive one, which can be seen from the plots.

Before extracting these features, we first remove the small parts from the beginning and end of the signal, respectively. Because we notice that these signal is likely to contain some noise. The ratio we choose here is 10%, which means we remove the first 10% of the signal and the last 10% of the signal.

2. Representation

After extracting global features, we can represent each sound sample using four feature-vector such that $w_i = (f_1, f_2, .., f_4)$. where f_1 is the skewness feature, f_2 is the kurtosis feature and so on. The next thing we need to do is to build probabilistic classifier using these four features. We propose two probabilistic classifiers in this work, and combine each of them with dynamic time warping, which we will show in the following sections.

E. Combining Quadratic Classifier with DTW

The first approach we proposed is the one by combining quadratic classifier(QDA) with dynamic time warping. One of the most simplest discriminant method is quadratic classifier, which is solely based on gaussian assumption.

1. Quadratic Discriminant Analysis

The core idea is to model samples within the same class using multivariate gaussian distribution. Then we calculate the probability of a class label given by a new sample to figure out which is the best label we should assign to the that new sample.

The multivariate normal probability density function is

$$f_X(x) = (2\pi)^{-N/2}|\Sigma|^{-1/2}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

where N is the number of variables in the feature vector, Σ is the covariance matrix and μ is the mean vector.

Using Bayes rule,

$$\begin{aligned} g_i(x) = P(w_i|x) &= (P(w_i p(x|w_i))/p(x) \\ &= (2\pi)^{-N/2}|\Sigma_i|^{-1/2}e^{-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)}P(w_i)/p(x) \end{aligned}$$

After eliminating constant terms, we can get

$$g_i(x)|\Sigma_i|^{-1/2}e^{-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)}P(w_i)$$

Then take the natural logs, we can get the final form:

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) - \frac{1}{2}log|\Sigma_i| + logP(w_i).$$

The final form of this probability shows the likelihood of sample x belongs to the

class i.

2. Recognition

Now, the distance between w_{new} and $w_i, i \in \{1, ..., n\}$ can be computed as

$$\text{Distance}(w_{\text{new}}, w_i) = \underbrace{a/\{\log(1/g(i)|w_{\text{new}}))\}}_{\textit{adjustment factor}} \times \underbrace{\textbf{dist}(\text{MFCCs}(w_{\text{new}}), \text{MFCCs}(w_i))}_{\textit{distance between two samples}} \tag{5.2}$$

where a is the constant factor that controls the relative importance of the first term. The subsequent recognition steps are the same as the classical dynamic time warping we showed in chapter 4.

3. Filtering Out Training Samples

Given a new sample w_{new}, we use quadratic discriminant analysis to find out the most likely subset of labels, $\mathcal{B} \subseteq \mathcal{C}$. The computation process is:

1. Calculate $g(c_1|w_{\text{new}}), ..., g(c_k|w_{\text{new}})$.
2. Sort these values. Then we can get order statistics $g_{[1]}, ...g_{[k]}$, where $g_{[1]} \geq g_{[2]} \cdots \geq g_{[k-1]} \geq g_{[k]}$
3. Calculate s=$\arg\min_j \sum_{i=1}^{j} g_{[i]} \geq 0.8$
4. Select the labels corresponding to $g_{[1]}, ...g_{[s]}$, and only consider the samples whose labels are in $\mathcal{B}$

F. Combining Copula Classifier with DTW

In this section, we propose another hybrid approach by combining copula classifier with DTW. Copula theory is the widely studied area in statistics, but still are not

common to the computer science field. It provides great potentials for classification and stochastic modeling. Copula theory has been widely used in financial area [34, 11, 40].

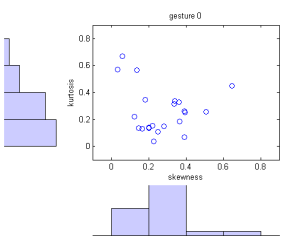
1. Motivation

Quadratic classifier assumes everything is gaussian. It models random vector using multivariate gaussian distribution and models each marginal distribution using gaussian as well. The model is very efficient since the gaussian nature has great benefits for problem formulation and computation. The clear advantages is that marginal, joint, conditional density are all gaussian. Gaussian process [39] is one extension that fully utilizes these gaussian properties and have great performance in prediction and modeling stochastic process.

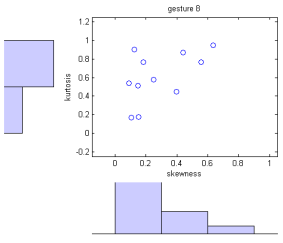
Let's back to our problem. Before deciding which classifier to choose, let's first look at how these four global features are distributed. The Figure 22 and 23 show the plot. From the figure, we can see that these features are not distributed as gaussian, at least not close to gaussian from what we've seen from the data set. Since it is shown as non-gaussian, it's best find other ways to deal with the classification problem, this is the motivation where our copula model comes into. By using copula model, we no longer need to explicitly assume the marginal density as gaussian.

2. Copula Discriminant Analysis

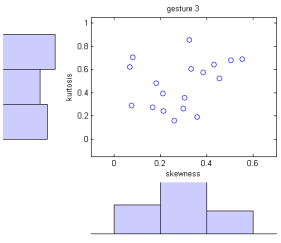
The core ideal is similar to the quadratic discriminant analysis. We need to calculate the probability of label c given a new sample w_i, that is, $P(c|w_{\text{new}})$. However, by using the copula theory, we don't need to explicitly assume any gaussian marginal. The beauty of the copula theory is allowing us to separate the choice of the marginal and that of the dependence structure of between random variables which is expressed



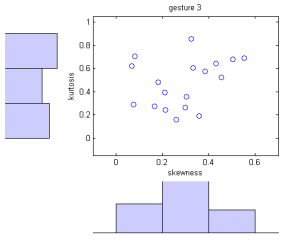
(a) gesture 0



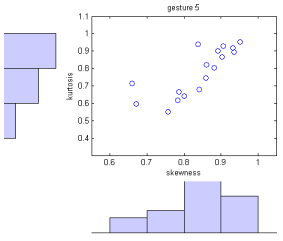
(b) gesture 1



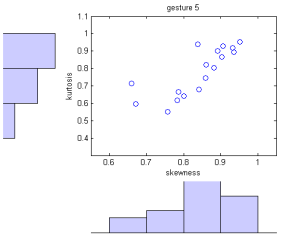
(c) gesture 2



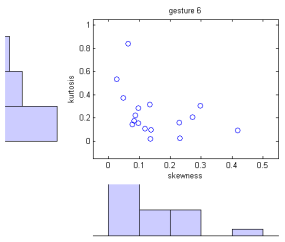
(d) gesture 3



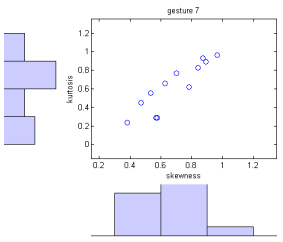
(e) gesture 4



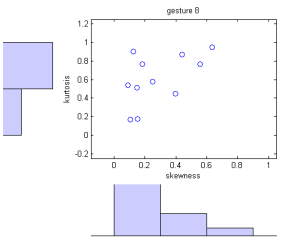
(f) gesture 5



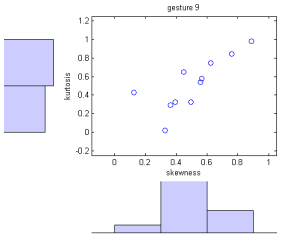
(g) gesture 6



(h) gesture 7

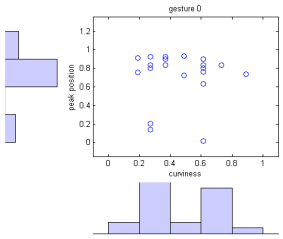


(i) gesture 8

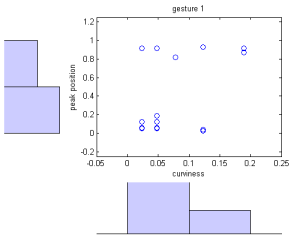


(j) gesture 9

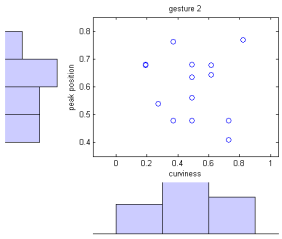
Fig. 22.: Skewness and Kurtosis for gesture "0-9"



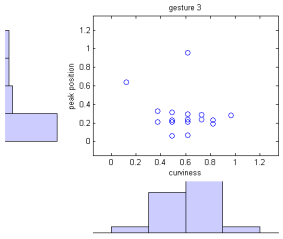
(a) gesture 0



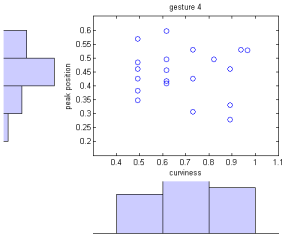
(b) gesture 1



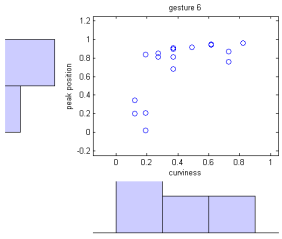
(c) gesture 2



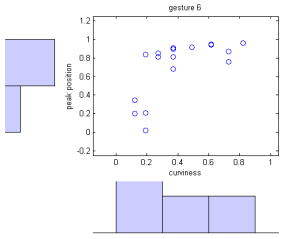
(d) gesture 3



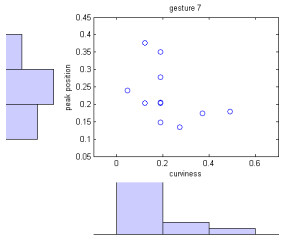
(e) gesture 4



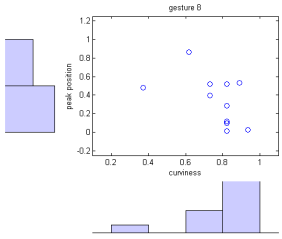
(f) gesture 5



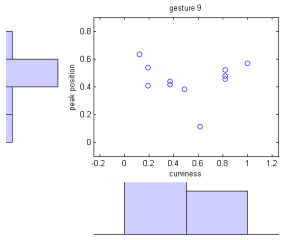
(g) gesture 6



(h) gesture 7



(i) gesture 8



(j) gesture 9

Fig. 23.: Curviness and Peak location for gesture "0-9"

in copula function C [7,8]. This provides great flexility of modeling each marginal using any distribution or via kernel density. Figure 22 and 23 show the marginal distribution of each variable (feature) for "0-9" gestures. From these two figures, we can see that simply modeling using gaussian is not a wise idea.

Let $\mathcal{X} = \{X_1, ..., X_N\}$ be a finite set of real-valued random variables and let $F_{\mathcal{X}}(\mathbf{x}) \equiv P(X_1 \leq x_1, ..., X_n \leq x_N)$ be a CDF (cumulative distribution function) over $\mathcal{X}$.

Definition 1. *A copula C is a joint distribution function of standard uniform random variables.*

$$C(u_1, ..., u_N) = P(U_1 \leq u_1, ...U_N \leq u_d),$$

where $U_i \sim U(0,1)$ for $i = 1, .., N$.

Following is an important result showing how a copula function is related to a joint distribution function.

Theorem 1. ***[Sklar 1959]***. *Let $F(x_1, ..., x_N)$ be any multivariate distribution over real-valued random variables, then there exists a copula function such that*

$$F(x_1, ...x_N) = C(F_1(x_1), ...F_N(x_N))$$

if $F_1(x_1), ...F_N(x_N)$ are all continuous, then C is unique [9].

According to Sklar's theorem, any joint distribution function F with continuous marginal $F_1, ..F_N$ has associated a copula function C. Another important result is that the joint density f and the marginal densities $f_1, ..., f_N$ are also related [7]:

$$f(x_1, ..., x_N) = \frac{\partial^N C(F_1(x_1), ...F_N(x_N))}{\partial F_1(x_1), ..., F_N(x_N)} \prod_i f_i(x_i) = C(F_1(x_1), ...F_N(x_N)) \prod_i f_i(x_i)$$

where c is the density of the copula C. The density equation shows that the product of marginal densities and a copula density gives a N-dimensional joint density. One

beautiful thing here is, the marginal densities can be different, and can be estimated either through parametric or non-parametric way, separately.

3. Copula Functions

One of the most popular copula functions is gaussian copula, which we used in this work.

Definition 2. *The copula associated to the joint standard Gaussian distribution is called Gaussian copula. Gaussian copula has the following expression:*

$$C(\Phi(u_1), ..., \Phi(u_N); \Sigma) = \int_{-\infty}^{\Phi^{-1}(u_1)} ... \int_{-\infty}^{\Phi^{-1}(u_N)} \frac{e^{-\frac{1}{2}t'\Sigma^{-1}t}}{(2\pi)^{n/2}|\Sigma|^{1/2}} dt_N ... dt_1$$

where Φ is the cumulative distribution function of the marginal standard Gaussian distribution and Σ is a symmetric matrix which denotes pairwise correlations between variables Z_i and Z_j , for $i, j = 1, ..., N$. The correlation matrix has $N(N-1)/2$ parameters, which can be estimated using the maximum likelihood method.

Now, let's turn to our first problem, to estimate $P(c_i|w_{\text{new}})$, where c_i is a class label, and w_{new} is new sample to be classified. Suppose we have K classes, $c_1, ..., c_k$. Using Bayes' theorem, we have

$$P(c_i|w_{\text{new}}) = \frac{P(w_{\text{new}}|c_i) \times P(c_i)}{P(w_{\text{new}})}$$

where $P(c_i|w_{\text{new}})$ is the posterior probability, and $P(c_i)$ is the prior probability. Here, we assume the prior probability is identical for all the labels, such that $P(c_i) = 1/K, \forall j \in \{1, ...K\}$. And by removing the constant factor, $P(w_{\text{new}})$, we can estimate the posterior probability by calculating $P(w_{\text{new}}|c_i)$. (Since we remove all the constant factors, after getting all the probability measure for each label, we need to normalize them to make them sum up to 1)

By incorporating Gaussian copula function, we finally estimate the probability using:

$$P(c_i|w_{\text{new}}) = c(F_1(x_1), ..., F_N(x_N)|\Sigma, c_i) \times \prod_{s=1}^{N} f_s(x_s|c_i) \tag{5.3}$$

4. Recognition

Now, the distance between w_{new} and $w_i, i \in \{1, ..., n\}$ can be computed as

$$\text{Distance}(w_{\text{new}}, w_i) = \underbrace{a/\{\log(1/P(c_i|w_{\text{new}}))\}}_{\textit{adjustment factor}} \times \underbrace{\textbf{dist}(\text{MFCCs}(w_{\text{new}}), \text{MFCCs}(w_i))}_{\textit{distance between two samples}} \tag{5.4}$$

where a is the constant factor that controls the relative importance of the first term. The subsequent steps are the same as above.

5. Filtering Out Samples

The process is the same as the one we mentioned in quadratic case. Given a new sample w_{new}, we use copula discriminant analysis to find out the most likely subset of labels, $\mathcal{B} \subseteq \mathcal{C}$. The computation process is:

1. Calculate $P(c_1|w_{\text{new}}), ..., P(c_k|w_{\text{new}})$.
2. Sort these values. Then we can get order statistics $P_{[1]}, ...P_{[k]}$, where $P_{[1]} \geq P_{[2]} ... \geq P_{[k-1]} \geq P_{[k]}$
3. Calculate s=$\arg\min_j \sum_{i=1}^{j} P_{[i]} \geq 0.8$
4. Select the labels corresponding to $P_{[1]}, ...P_{[s]}$, and only consider the samples whose labels are in $\mathcal{B}$

G. Some Other Possible Extensions

Besides all these approaches we proposed already, there are another two possible extensions we can try. We briefly summarize them here, and give some experimental results in the next chapter.

1. Complete Two-Stage Approach

As the name indicates, the method works as two independent stages. At the first stage, the approach uses probabilistic classifier to filter out some portions of samples. And in the second stage, we use dynamic time warping to recognize the sound. The difference between this approach and the one we proposed above is that in the latter case we combine the probability with distance measure while in the first case we consider these two stages separately. Thus, in this case, the probabilistic classifier only performs with the purpose of filtering out examples. More experimental results will be shown in later section.

2. Combining Global Features with Distance Measure

This is another interesting approach we've tried in this work. This is not hybrid approach since it only uses probabilistic classifier. It works by combining global features with distance measure calculated using dynamic time warping. In this case, we have total 5 features, where four of them are global features and the remaining one is distance measure we got from the dynamic time warping. Using these five features, we can build probabilistic classifier for the sound. The process is the same as the first phase in hybrid approach. In terms of classification method, we can choose to use either quadratic classifier or copula classifier. The results are shown in the evaluation section.

CHAPTER VI

EVALUATION

In this section, we will show our comprehensive experimental studies. We conducted evaluation from different aspects. We tested our algorithm using three gesture sets, and we also tested the performance of each global feature. More information will be given in the subsequent sections.

A. Three Gesture Sets

All the data is collected using our microphone embedded in the android phone, and normal tables. We only allow consistent background noise. We evaluated our algorithm using three different gesture sets. The first gesture set consists of "0-9" 10 digit numbers, and the second gesture set consists of 26 English Characters, and the last one contains seven commonly used gestures. These three gesture sets are shown in Figure 24, 25, 26.

B. Summary of Performance

Using the data we collected, we evaluated six versions of our system

DTW uses only dynamic time warping algorithm

CDA uses only copula discriminant analysis

QDA+DTW uses quadratic classifier and dynamic time warping

CDA+DTW uses copula discriminant analysis and dynamic time warping.

Average Distance use the average distance approach

Five Features combine four global features with distance measure

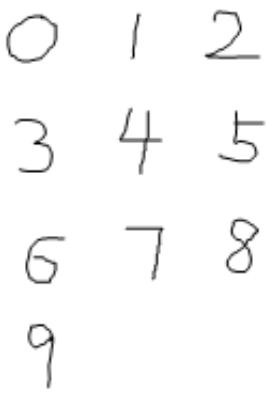


Fig. 24.: A gesture set contains 10 digit numbers. Most of these gestures contain high curves, which makes the problem more difficult.

1. "0-9" Digits

For this evaluation, we have total 5 participants, where each of them was asked to sketch each digit for 20 times on the table, so we have total $5 \times 20 \times 10 = 1000$ samples. We performed user-dependent, 4-fold cross validations. For experimental results, we average the accuracy across these five participants. The overall accuracy is shown in Table I. As we can see in the table, combining copula approach has positive impact on improving the accuracy, as we expected. Before combining copula classifier, we found that most of misclassifications are due to one or two samples only. When either of them was stored in the database, misclassification is likely to occur. However, if we combine copula discriminant analysis with distance measure, we implicitly decrease the probability of choosing that sample as best candidate. For combining QDA with DTW, we cannot see the clear improvements, but it gives relatively good performance comparing to using DTW only. Anther observation is that by adding distance measure into original four global features, the accuracy is improved a little bit, around 80%.

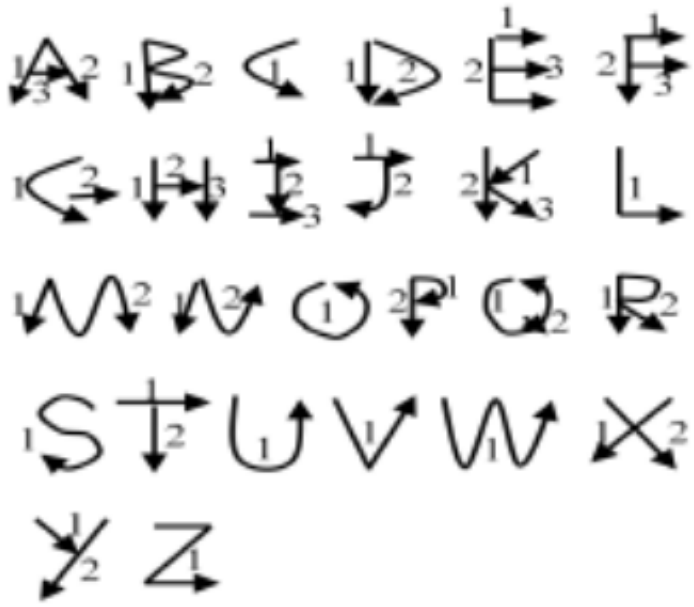


Fig. 25.: A gesture set contains 26 upper case English characters. We assume each individual writes the character using the same order every time. The figure shows one such possible orderings.

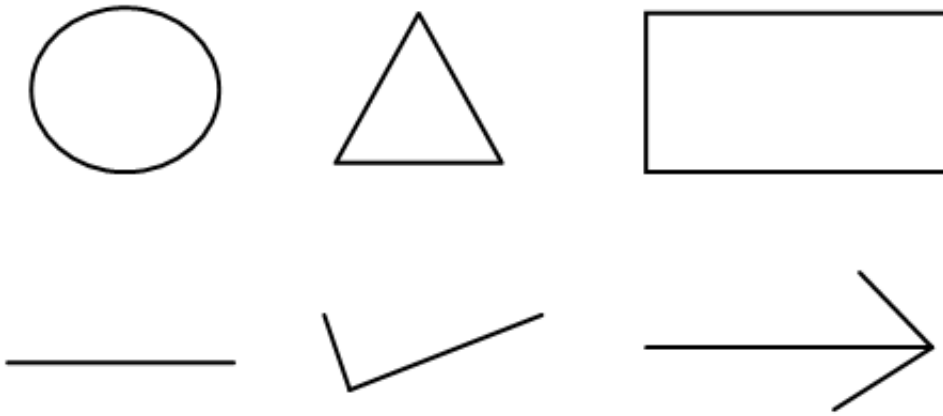


Fig. 26.: A gesture set contains seven different gestures. From left to right, up to down are circle, triangle, rectangle, line, check and arrow. The one that is not shown here is double tap gesture, which makes seven gestures in total.

However, using average distance here actually gives poorer performance comparing to using one nearest neighbor only. The possible reason is that when there is noisy data in the database, the distance value becomes very huge, almost approaches to the infinity. Thus, after averaging these values, the mean is still very large, which cause the misclassifications. We also give a confusion matrix for all five participants, which is shown in Table II.

2. "A-Z" 26 English Characters

For the second evaluation, we collected data from 6 participants. Each participant wrote each English character for 15 times. so we have total $6 \times 15 \times 26 = 2340$ samples. Please note that we let the user write upper case character. We performed user-dependent, 3-fold cross validation. Since some letters are multi-stroke, we encouraged the participant to write these strokes using the same order every time. We got the

Table I.: Accuracy for 10 digits

Case	Accuracy
DTW	0.883
CDA	0.703 (within top 1). 0.931 (within top 3)
QDA+DTW	0.891
CDA+DTW	**0.905**
Average Distance	0.828
Five Features	0.802

similar result pattern with the first evaluation, as shown in Table III. CDA+DTW has the highest accuracy rate, followed by QDA+DTW and DTW. Using five features, we can still improve the accuracy a little bit, but it is still lower than when we use dynamic time warping. In this case, the average distance approach has relatively good performance comparing to the other approaches. An important question is, is 83% accuracy good enough for building real world systems? In fact, if we classify individual character, it is hard to get above 90% accuracy simply because the sound profile from certain pairs of characters are almost the same. Table IV shows some pairs of characters are easily misclassified. (due to the space limit, we show this table instead of showing actual confusion matrix, which has 26 by 26 entries). In fact, more than 90% misclassifications are caused by these pairs of characters.

3. Seven Gestures

For this evaluation, we do not have much data available. We only have two participants, where each of them sketch each gesture for 20 times. Thus, we have total $2 \times 7 \times 20 = 280$. We again use 4-fold cross validations. The accuracy is shown in

Table II.: Confusion matrix for 10 digit (for five participants) and (CDA+DTW)

	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
'0'	81	0	0	0	0	0	11	0	2	6
'1'	2	86	0	0	0	0	0	12	0	0
'2'	0	0	95	5	0	0	0	0	0	0
'3'	0	1	2	90	0	0	5	0	0	2
'4'	0	0	0	0	100	0	0	0	0	0
'5'	0	0	0	6	0	92	2	0	0	0
'6'	8	0	0	1	0	0	86	0	3	2
'7'	0	4	3	0	0	0	0	92	0	1
'8'	1	0	0	0	0	0	4	0	95	0
'9'	6	0	0	2	0	0	1	0	3	88

Table V. In this case, the average distance approach outperforms the combination of CDA and DTW. We guess the main reason is that there are no extreme samples contained in the database. More work can be done to further investigate the performance of the average distance approach. Since we only have seven gestures here, and the defined gestures can be easily distinguishable between each other, the accuracy is relatively high comparing to the first two experiments. In fact, in most of the cases, we only need such small number of gestures to manipulate common tasks. We also give the confusion matrix, which is shown in Table VI. By default, the confusion matrix is based on the combination of CDA and DTW.

Table III.: Accuracy for 26 upper case English characters.

Case	Accuracy
DTW	0.870
CDA	0.760(within top 3) 0.306 (within top 1)
QDA+DTW	0.876
CDA+DTW	**0.878**
Average Distance	0.821
Five Features	0.502

C. Word Recognition

What we discussed so far are for isolated sound recognition. In this experiments, we tested our recognizer on continues sound. The only additional step we add into the recognizer is the sound segmentation. For simplicity, we assume people pause for some time before moving to another character. This makes the segmentation can be easily done by searching for long period silence. We set this threshold to 1 second. The reason for setting such a large threshold is that we want to distinguish the silence between two characters and the one within the individual sound (we already showed that silence within the individual sound does not necessarily mean it is invalid signal). After that, we can simply treat each segment independently, and use probability model like Naive Bayes to combine them together.

In fact, the experiment shows one important observation that the ambiguities can be relieved by recognizing sequence of characters in context through the use of a dictionary. We did two independent evaluations.

Table IV.: Pairs of characters are easily misclassified

Character 1	Character 2
A	F
A	H
D	P
J	T
C	U
X	Y

1. Recognizing Names

We let the participants write down any person name one at a time, and write down total 20 names. All the sound was recorded and stored as ".wav" file. Please note that these people already provided training samples for individual character. All the names they wrote are already in the dictionary, which contains total 50 people's names. The result shows that there are only 3 of them are incorrectly classified, which equals to the 92.5% accuracy rate.

2. Recognizing Commonly Used Words

This experiment again is based on the character recognition. But instead of writing names, we allow participant to write any word from the dictionary, which contains 500 most commonly used words. Unfortunately, we only have one participant who helped collect 100 samples. Finally, our algorithm correctly classified 87 of them, which is 87% accuracy rate.

One thing to note is that most of the misclassification for these two cases are due to either segmentation error or silence detection error. For example, in the first case,

Table V.: Accuracy for seven gestures

Case	Accuracy
DTW	0.95
CDA	0.896
QDA+DTW	0.961
CDA+DTW	**0.961**
Average Distance	0.964
Five Features	0.929

all of three misclassifications are caused by segmentation error, while in the second case 10 of 17 misclassifications are cased by segmentation and silence detection errors.

D. Effect of Different Materials

One interesting question is: does sketching tools matter? The answer is "Yes". We conducted experiment to figure out the effect of using different sketching tools. The experimental setup is rather simple. We let 6 participants provide another 10 samples for each character. They wrote 5 samples using pen and another 5 samples using fingernail. Then we use cross validation to get the result which is shown in Table VII. As we can see from the table, using key and pen make the system have higher performance.

E. Analysis of Four Global Features

In this section, we analyze the performance of each global feature. And we also show how much computational cost can be reduced by using our first step probability

Table VI.: Confusion matrix for seven commonly used gestures (CDA+DTW)

	circle	triangle	rectangle	line	check	arrow	double tap
circle	36	0	4	0	0	0	0
triangle	2	37	1	0	0	0	0
rectangle	1	0	39	0	0	0	0
line	0	0	0	39	0	1	0
check	0	0	0	0	40	0	0
arrow	0	1	0	0	2	37	0
double tap	0	0	0	0	0	0	40

Table VII.: Accuracy for different sketching tools.

	Key	Pen	Finger
Key	0.859	—	—
Pen	—	0.,868	—
Finger	—	—	0.783

classifier. By default (if no further mention), all these subsequent experiments are based on copula discriminant analysis.

1. Performance of Each Global Features

We tested the performance of individual feature using two gesture sets, digit and commonly used gesture set, respectively. For character set, the accuracy for individual feature is similar to the random guess, so we don't list it here. Figure 27 shows the accuracy for 10 digit gesture set, while the Figure 28 shows the accuracy for seven

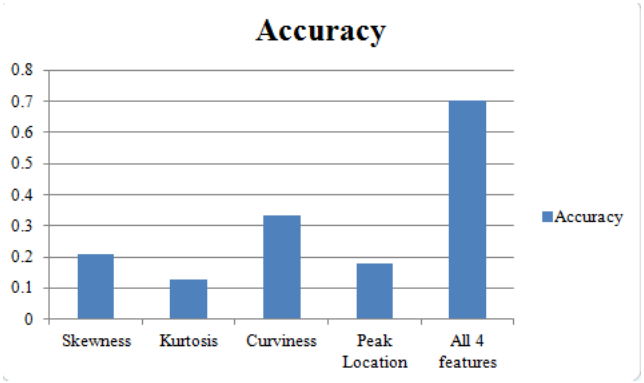


Fig. 27.: Accuracy for individual feature. (10 digits)

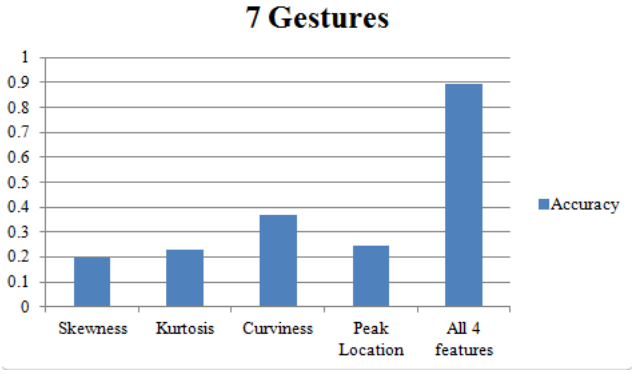


Fig. 28.: Accuracy for individual features. (7 gestures).

commonly used gestures. From this accuracy figure, we can see that the curviness feature performs best, while the kurtosis feature performs worse.

2. Reduced Computational Cost

Based on the probability measure we got from the first step, we filter out the samples whose labels are out of subset $\mathcal{B}$. We found that this strategy can reduce the time complexity by $30 \sim 70\%$, depends on the probability distribution. Actually, one can control the confidence threshold (which we set 80%) to control the portion of samples

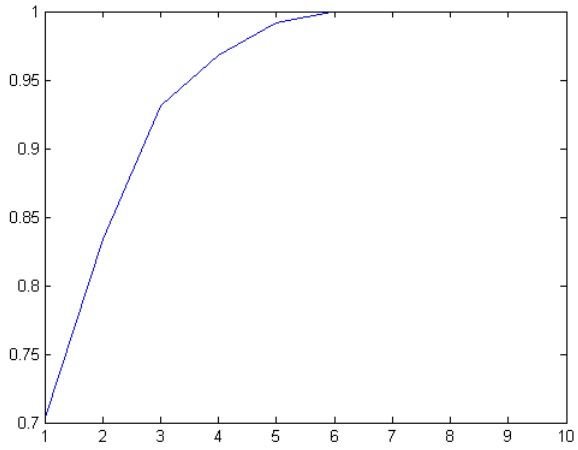


Fig. 29.: CDA accuracy for 10 digit gestures (top N accuracy). From this figure, we can see that when we choose the top 6, it is almost contains the correct label.

to filter out. One question might be, is there any possibility that the samples that are filtered out already have the true label with the new sample? The answer is yes, by reducing the search space, we sacrifice the accuracy by accidently filtering out these samples.

Thus, the next question is how many classes are enough for getting like over 95% accuracy. The goal of this analysis is that we are trying to avoid to filter out the samples which have the same label with the new sample. The Figure 29, 30, 31 show the quartile graphs for three different gesture sets. From these figures, we can easily see that if we cut off the 50% of the samples, we almost get the same accuracy as when we include the whole sample set. Finally, we show all these three graphs in the same one based on the percentage of classes we choose, which is shown in Figure 32.

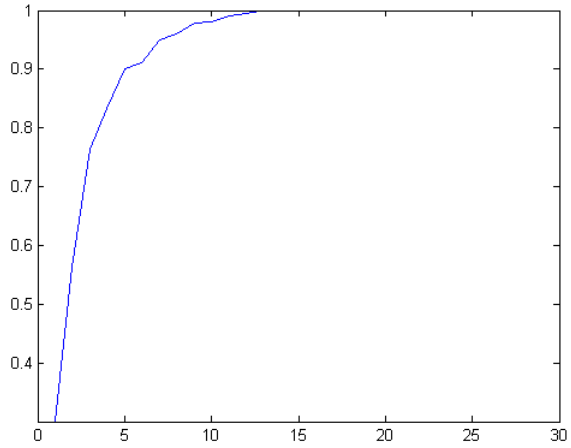


Fig. 30.: CDA accuracy for 26 characters (top N accuracy).From this figure, we can see that when we choose the top 10, it is almost contains the correct label.

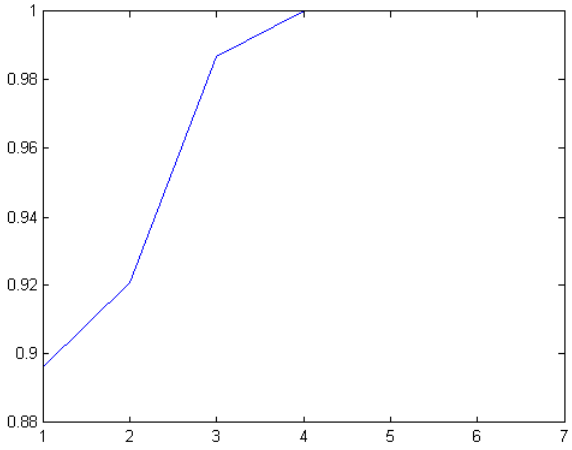


Fig. 31.: CDA accuracy for seven commonly used gestures (top N accuray). From this figure, we can see that when we choose the top 4, it is almost contains the correct label..

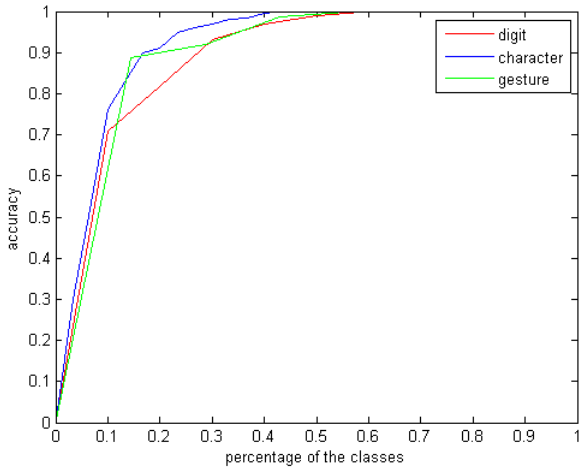


Fig. 32.: Accuracy based on percentage of classes we choose. This is the combined plot of figure 29,30,31.

F. Analysis of Complete-Two-Stage Approach

Finally, we evaluate complete-two-stage approach, which consists of two independent stages. At the first stage, we use copula classifier to cut off the half of the training samples, and in the second stage, we use dynamic time warping to find the best label. The second stage is exactly same as the one we showed in chapter 4. The result is shown in Table VIII.

Table VIII.: Accuracy for three gesture sets using complete-two-stage approach

Gesture Set	Accuracy
10 Digit	0.891
26 English Character	0.876
7 Gestures	0.961

CHAPTER VII

DISCUSSION AND FUTURE WORK

In this chapter, we will review some common questions and then give potential future directions of our work.

Why we restrict the system as user-dependent? is it too conservative? We think this is the natural process for investing new research problems. Since this is pretty new research area and there is little thing we can learn from the past. What we are trying to do is to build a prototype system and just make it work for simple problems. In the future, we can extend it to make it work for user-independent cases. What we want to emphasis is that this is just beginning of the research.

What is the major weakness of our work? Why not develop more elegant signal processing techniques? We think the major weakness of our work is the part dealing with signal processing. As we've shown in the evaluation part, most of the misclassifications for continuous words are caused by segmentation and silence detection errors. Perhaps in the future, once we incorporate more robust signal processing techniques, we may get much better performance. But anyway, the major focus of this work is not for developing new signal processing techniques.

Why not try other method like HMM? Hidden markov model is one of the most widely used statistical model in speech community. Many of today's speech recognition systems are built using HMM. In fact, HMM is the most efficient way for modeling continues speech. However, the main focus of this work is recognizing isolated sound, and dynamic time warping is good for solving this problem. If we decide to build robust system for continues sound, we may change to HMM instead.

Since this is pretty new research area, there are many potential directions we can investigate in the future. We briefly summarize them as follows:

- Use better signal processing techniques.
- Further investigate how different tools can affect the system performance. One interesting research might be identifying different tools automatically using machine learning techniques.
- Designing editing gestures. This is very helpful for building interactive systems. In fact, using gestures to accomplish common editing tasks like delete, undo and redo is very intuitive.
- Collect more data. We definitely need more data to build robust system. However, the data collection is very expensive for our problems. If we have more data, it will be very helpful.
- Comparing with other methods like GMM. For individual sound recognition, we can choose to compare the performance between DTW and GMM.
- Applications. Our techniques are beneficial for many other research problems and applications. One interesting way to think about the problem is to build multi-model interaction system. As we know, solely using camera to detect gestures sometimes do not work well. How we can combine the sound with computer vision techniques might be an intersecting research topic.

CHAPTER VIII

CONCLUSION

In this thesis, we propose a novel interaction technique, which is very cheap and can be work for any device. We also propose novel algorithms to recognize the sound. The propose algorithms can work for many other domains, are not just restricted to our problems. In summary:

- We propose a novel interaction technique, called acoustic based sketch recognition.
- We propose a dynamic time warping algorithm for recognizing acoustic sound.
- After realizing some disadvantages of using dynamic time warping directly, we propose improved versions of dynamic time warping algorithms.
- Besides, we also propose four novel features for sound. The four features can effectively summarize the properties of the sound. The intuition come from sketch and statistical domains.

REFERENCES

[1] P. Aarabi, "The fusion of distributed microphone arrays for sound localization," *EURASIP J. Appl. Signal Process.*, vol. 2003, pp. 338–347, Jan. 2003.

[2] L. Anthony and J. O. Wobbrock, "A lightweight multistroke recognizer for user interface prototypes," in *Proceedings of Graphics Interface 2010*, ser. GI '10. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2010, pp. 245–252.

[3] E. Beadle, J. Schroeder, B. Moran, and S. Suvorova, "An overview of renyi entropy and some potential applications," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, oct. 2008, pp. 1698 –1704.

[4] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 2, pp. 113 – 120, apr 1979.

[5] C.-C. Chen, "Improved moment invariants for shape discrimination," *Pattern Recognition*, vol. 26, no. 5, pp. 683 – 686, 1993.

[6] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21 –27, january 1967.

[7] K. Fredriksson, V. Mkinen, and G. Navarro, "Rotation and lighting invariant template matching," In Proc. 6th Latin American Symposium on Theoretical Informatics (LATIN 2004), LNCS 2976, Tech. Rep., 2003.

[8] S. S. G. Saha, Sandipan Chakroborty, "A new silence removal and endpoint detection algorithms for speech and speaker recognition applications," in *Proceedings of NCC*, 2005.

[9] B. Gajic and K. K. Paliwal, "Robust feature extraction using subband spectral centroid histograms," in *in Proc. ICASSP*, 2001, pp. 85–88.

[10] S. Garca, J. R. Cano, and F. Herrera, "A memetic algorithm for evolutionary prototype selection: A scaling up approach," *Pattern Recognition*, vol. 41, no. 8, pp. 2693 – 2709, 2008.

[11] C. Genest, M. Gendron, and M. Bourdeau-Brien, "The advent of copulas in finance," *The European Journal of Finance*, vol. 15, no. 7-8, pp. 609–618, 2009.

[12] J. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, pp. 33–51, 1975.

[13] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 1911–1914.

[14] T. Hammond and R. Davis, "Tahuti: a geometrical sketch recognition system for uml class diagrams," in *ACM SIGGRAPH 2006 Courses*, ser. SIGGRAPH '06. New York, NY, USA: ACM, 2006.

[15] C. Hampton, B. Lian, and W. McHarris, "Fast-fourier-transform spectral enhancement techniques for -ray spectroscopy," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 353, no. 1C3, pp. 280 – 284, 1994.

[16] C. Harrison and S. E. Hudson, "Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces," in *Proceedings of the 21st annual*

ACM symposium on User interface software and technology, ser. UIST '08. New York, NY, USA: ACM, 2008, pp. 205–208.

[17] J. Herre, E. Allamanche, and O. Hellmuth, "Robust matching of audio signals using spectral flatness features," in *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*, 2001, pp. 127 –130.

[18] J. H. Hongsheng Li, "Optimal Object Matching via Convexi?cation and Composition," in *ICCV*, 2011.

[19] H. Hse and A. R. Newton, "Sketched symbol recognition using zernike moments," in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1 - Volume 01*, ser. ICPR '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 367–370.

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

[21] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the hausdorff distance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 9, pp. 850 –863, sep 1993.

[22] z. Izmirli, "Using a spectral flatness based feature for audio segmentation and retrieval." in *ISMIR*, 2000.

[23] D. N. Joanes and C. A. Gill, "Comparing measures of sample skewness and kurtosis," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 47, no. 1, pp. 183–189, 1998.

[24] L. B. Kara, "An image-based trainable symbol recognizer for sketch-based interfaces," in *in AAAI Fall Symposium Series 2004: Making Pen-Based Interaction*

Intelligent and Natural. AAAI Press, 2004, pp. 99–105.

[25] J.-E. Kim, J. Sunwoo, Y.-K. Son, D.-W. Lee, and I.-Y. Cho, "A gestural input through finger writing on a textured pad," in *CHI '07 extended abstracts on Human factors in computing systems*, ser. CHI EA '07. New York, NY, USA: ACM, 2007, pp. 2495–2500.

[26] J. J. LaViola, Jr. and R. C. Zeleznik, "Mathpad2: a system for the creation and exploration of mathematical sketches," in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 432–440.

[27] W. Li and T. A. Hammond, "Recognizing text through sound alone," in *AAAI*, 2011.

[28] Y. Li, "Reforming the theory of invariant moments for pattern recognition," *Pattern Recognition*, vol. 25, no. 7, pp. 723 – 730, 1992.

[29] A. Martin, D. Charlet, and L. Mauuary, "Robust speech/non-speech detection using lda applied to mfcc," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 1, 2001, pp. 237 –240 vol.1.

[30] B. Martin and V. Juliet, "Extraction of feature from the acoustic activity of rpw using mfcc," in *Recent Advances in Space Technology Services and Climate Change (RSTSCC), 2010*, nov. 2010, pp. 194 –197.

[31] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Mullers, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, aug 1999, pp. 41 –48.

[32] H. Misra, S. Ikbal, H. Bourlard, and H. Hermansky, "Spectral entropy based feature for robust ASR," IDIAP, Martigny, Switzerland, Idiap-RR Idiap-RR-56-2003, 0 2003, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2004.

[33] C. Myers, L. Rabiner, and A. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 6, pp. 623 – 635, dec 1980.

[34] A. J. Patton, "Copula based models for financial time series," in *Handbook of Financial Time Series*, T. Mikosch, J.-P. Krei, R. A. Davis, and T. G. Andersen, Eds. Springer Berlin Heidelberg, 2009, pp. 767–785.

[35] E. Pekalska, R. P. Duin, and P. Paclk, "Prototype selection for dissimilarity-based classifiers," *Pattern Recognition*, vol. 39, no. 2, pp. 189 – 208, 2006.

[36] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[37] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 –286, feb 1989.

[38] A. Ramalingam and S. Krishnan, "Gaussian mixture modeling using short time fourier transform features for audio fingerprinting," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, july 2005, pp. 1146 –1149.

[39] C. Rasmussen, "Gaussian processes in machine learning," in *Advanced Lectures on Machine Learning*, ser. Lecture Notes in Computer Science, O. Bousquet,

U. von Luxburg, and G. Rtsch, Eds. Springer Berlin / Heidelberg, 2004, vol. 3176, pp. 63–71.

[40] J. C. Rodriguez, "Measuring financial contagion: A copula approach," *Journal of Empirical Finance*, vol. 14, no. 3, pp. 401 – 423, 2007.

[41] A. G. Seniuk and D. Blostein, "Pen acoustic emissions for text and gesture recognition," in *ICDAR*, 2009, pp. 872–876.

[42] J. Snchez, F. Pla, and F. Ferri, "Prototype selection for the nearest neighbour rule through proximity graphs," *Pattern Recognition Letters*, vol. 18, no. 6, pp. 507 – 513, 1997.

[43] T.-H. Sun, C.-S. Liu, and F.-C. Tien, "Invariant 2d object recognition using eigenvalues of covariance matrices, re-sampling and autocorrelation," *Expert Syst. Appl.*, vol. 35, no. 4, pp. 1966–1977, Nov. 2008.

[44] Y. Sun and M. G. Genton, "Functional boxplots," *Journal of Computational and Graphical Statistics*, vol. 20, no. 2, pp. 316–334, 2011.

[45] S. L. Tanimoto, "Template matching in pyramids," *Computer Graphics and Image Processing*, vol. 16, no. 4, pp. 356 – 369, 1981.

[46] W. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, pp. 401–419, 1952.

[47] P. Tsiakoulis, A. Potamianos, and D. Dimitriadis, *Short-time instantaneous frequency and bandwidth features for speech recognition.* IEEE, 2009.

[48] S. Valentine, F. Vides, G. Lucchese, D. Turner, H. Kim, H. Kim, W. Li, J. Linsey, and T. A. Hammond, "Mechanix: A sketch-based tutoring system for statics

courses," in *The Twenty-Fourth Conference on Innovative Applications of Artificial Intelligence*, 2012.

[49] D. F. Williamson, R. A. Parker, and J. S. Kendrick, "The box plot: a simple visual method to interpret data." *Annals of Internal Medicine*, vol. 110, no. 11, pp. 916–921, 1989.

[50] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes," in *Proceedings of the 20th annual ACM symposium on User interface software and technology*, ser. UIST '07. New York, NY, USA: ACM, 2007, pp. 159–168.

[51] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1C3, pp. 37 – 52, 1987, ¡ce:title¿Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists¡/ce:title¿.

[52] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. Huang, "Comparing mfcc and mpeg-7 audio features for feature extraction, maximum likelihood hmm and entropic prior hmm for sports audio classification," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 5, april 2003, pp. V – 628–31 vol.5.

[53] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian, "Hmm-based audio keyword generation," in *Advances in Multimedia Information Processing - PCM 2004*, ser. Lecture Notes in Computer Science, K. Aizawa, Y. Nakamura, and S. Satoh, Eds., vol. 3333. Springer Berlin / Heidelberg, 2005, pp. 566 – 574.

[54] Y. Xu, J. Weaver, D. Healy, and J. Lu, "Wavelet transform domain filters: a spatially selective noise filtration technique," *Image Processing, IEEE Transactions*

on, vol. 3, no. 6, pp. 747 –758, nov 1994.

VITA

Name: Wenzhe Li

Email: liwenzhe@cse.tamu.edu

Education: B.S., Computer Science, Nankai University, 2009

M.S., Computer Science, Texas A&M University, 2012

Address: Teague Building, Room 327, 3112 Texas A&M University,

College Station, TX 77843, USA