MASTER'S THESIS

Lip password-based speaker verification system with unknown language alphabet

Zhou, Yichao

Date of Award:
2018

HONG KONG BAPTIST UNIVERSITY

Master of Philosophy

THESIS ACCEPTANCE

DATE: August 31, 2018

STUDENT'S NAME: ZHOU Yichao

THESIS TITLE: Lip Password-based Speaker Verification System with Unknown Language Alphabet

This is to certify that the above student's thesis has been examined by the following panel members and has received full approval for acceptance in partial fulfillment of the requirements for the degree of Master of Philosophy.

Chairman:	Dr Tong Tiejun Associate Professor, Department of Mathematics, HKBU (Designated by Dean of Faculty of Science)
Internal Members:	Dr Tam Hon Wah Associate Professor, Department of Computer Science, HKBU (Designated by Head of Department of Computer Science)
	Prof Cheung Yiu Ming Professor, Department of Computer Science, HKBU
External Members:	Prof Zhang Qingfu Professor Department of Computer Science City University of Hong Kong

Issued by Graduate School, HKBU

Lip Password-based Speaker Verification System with Unknown Language Alphabet

ZHOU Yichao

A thesis submitted in partial fulfillment of the requirements

for the degree of

Master of Philosophy

Principal Supervisor: Prof. CHEUNG Yiu-ming

Hong Kong Baptist University

August 2018

Declaration

I hereby declare that this thesis represents my own work which has been done after registration for the degree of MPhil at Hong Kong Baptist University, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications.

I have read the University's current research ethics guidelines, and accept responsibility for the conduct of the procedures in accordance with the University's Committee on the Use of Human & Animal Subjects in Teaching and Research (HASC). I have attempted to identify all the risks related to this research that may arise in conducting this research, obtained the relevant ethical and/or safety approval (where applicable), and acknowledged my obligations and the rights of the participants.

Signature: Zhou Yichao

Date: August 2018

Abstract

The traditional security systems that verify the identity of users based on password usually face the risk of leaking the password contents. To solve this problem, biometrics such as the face, iris, and fingerprint, begin to be widely used in verifying the identity of people. However, these biometrics cannot be changed if the database is hacked. What's more, verification systems based on the traditional biometrics might be cheated by fake fingerprint or the photo.

Liu and Cheung (Liu and Cheung 2014) have recently initiated the concept of lip password, which is composed of a password embedded in the lip movement and the underlying characteristics of lip motion [26]. Subsequently, a lip password-based system for visual speaker verification has been developed. Such a system is able to detect a target speaker saying the wrong password or an impostor who knows the correct password. That is, only a target user speaking correct password can be accepted by the system. Nevertheless, it recognizes the lip password based on a lip-reading algorithm, which needs to know the language alphabet of the password in advance, which may limit its applications.

To tackle this problem, in this thesis, we study the lip password-based visual speaker verification system with unknown language alphabet. First, we propose a method to verify the lip password based on the key frames of lip movement instead of recognizing the individual password elements, such that the lip password verification process can be made without knowing the password alphabet beforehand. To detect these key frames, we extract the lip contours and detect the interest intervals where the lip contours have significant variations. Moreover, in order to

avoid accurate alignment of feature sequences or detection on mouth status which is computationally expensive, we design a novel overlapping subsequence matching approach to encode the information in lip passwords in the system. This technique works by sampling the feature sequences extracted from lip videos into overlapping subsequences and matching them individually. All the log-likelihood of each subsequence form the final feature of the sequence and are verified by the Euclidean distance to positive sample centers. We evaluate the proposed two methods on a database that contains totally 8 kinds of lip passwords including English digits and Chinese phrases. Experimental results show the superiority of the proposed methods for visual speaker verification.

Next, we propose a novel visual speaker verification approach based on diagonal-like pooling and pyramid structure of lips. We take advantage of the diagonal structure of sparse representation to preserve the temporal order of lip sequences by employ a diagonal-like mask in pooling stage and build a pyramid spatiotemporal features containing the structural characteristic under lip password. This approach eliminates the requirement of segmenting the lip-password into words or visemes. Consequently, the lip password with any language can be used for visual speaker verification. Experiments show the efficacy of the proposed approach compared with the state-of-the-art ones.

Additionally, to further evaluate the system, we also develop a prototype of the lip password-based visual speaker verification. The prototype has a Graphical User Interface (GUI) that make users easy to access.

Keywords: visual speaker verification, lip password, language alphabet

Acknowledgements

First of all, I would like to present great thanks to my supervisor Prof. Yiu-ming Cheung, for providing me with continuous supervision, professional guidance, and generous support. I believe that I will benefit from what I have learned from him throughout my career in the future.

I would like to give my sincere appreciation to Dr. Li Chen (my co-supervisor) and all the other faculty members for their help and advice. I also give my thanks to all the administrative and technical staffs in our department for their warm help.

I would like to thank Dr. Xin Liu, who helps me a lot in this research. I also have learned a lot from the discussions with Dr. Qinmu Peng, Dr. Hong Jia, Dr. Fangqing Gu, Mr. Jian Lou, Mr. Sheungwai Chan, Mr. Yang Lu and Mr. Yiqun Zhang. Additionally, I want to thank all my friends in HKBU who come along with me through the years.

I am also thankful to Xinyuan Zhang, Ziyuan Lin, Xiao Liu, and all my friends that provide research advice to me, give me mental support, and guide me when I am confused during my study. It is not possible to mention all of them in this thesis, but I extremely appreciate what they did for me.

Finally, I give my heartfelt thanks to the most important people in my life, my parents, parents-in-law, and my beloved husband. Without their endless love and continuously encouragement, I could definitely not finish this thesis.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background & Motivations

1.1.1 Lip Password-based Visual Speaker Verification

The traditional security systems that verify the identity of users based on password usually face the risk of leaking the password contents. To solve this problem, biometrics such as the face, iris, and fingerprint, begin to be widely used in verifying the identity of people. However, these biometric cannot be changed if the database is hacked. What's more, verification systems based on the traditional biometrics might be cheated by fake fingerprint or the photo.

Most recently, Liu and Cheung [26] have initiated a new concept, named "lip password", composed of a password embedded in the lip movement and the underlying characteristics of lip motion. The lip password increases the security of the speaker verification by providing a double-check on both the speaker's behavioral biometrics of lip motions and the embedded password.

Lip password-based visual speaker verification aims at verifying the speaker by both of the password information and the biometrics of lip motions simultaneously. The target speaker saying the wrong password or an impostor who knows the correct password will be detected and rejected. Compared with other biometric, such as the face, iris, and fingerprint, the password contents can be changed by users if the

passwords database is hacked. Also, the system is hard to be cheated with fake fingers or photos. Acoustic-based speaker verification systems not only are sensitive to the background noise but also can be easily deceived by a sound recorder [22], while the lip password based on only the visual information that can be used without sound. What's more, compared with other visual speaker verification, lip password provides double-security on both speakers identity and the contents of the password.

As shown in [26], lip password has at least four advantages:

1. The modality of lip motion is insusceptible to the background noise;
2. The acquisition of lip motions is insensitive to the distance to a certain degree;
3. Lip password can be performed silently;
4. It is applicable to speech impaired people.

It has been shown [26] that a lip password-based system can successfully detect not only a target speaker saying the wrong password, but also an impostor who knows the correct password will be detected and rejected. That is, only a target user saying the correct password can be accepted.

In [26], the visual feature sequences extracted from lip video are segmented according to the mouth open and close and then recognized individually. As a result, the lip password contents in [26] can be made up by given elements only with a priori knowledge of language alphabet, e.g. English digits, which limits the security and extensibility of lip password. Actually, from the practical perspective, it is natural that different speakers saying the lip passwords in different languages.

In this thesis, we aim at building a visual speaker verification system based on lip password without knowing the password alphabet beforehand. From a practical perspective, such a system should have the following properties: **Arbitrariness**: the system should allow the user to choose his private password without strict limitation on lexical content. **Rejection**: the system should reject not only the wrong password spoken by the target user, but also need to reject the impostor that saying the correct password [26]. **Robustness**: the system should be robust to environments

such as illumination variations [56]. **Convenience**: a user-friendly system should not require the user to repeat his lip password too many times. Therefore, the system should be able to learn the lip passwords from a few training examples.

1.1.2 Lip Password-based Visual Speaker Verification with Hidden Markov Model

Instead of recognizing the elements of password individually, we compare the whole lip movements to verify if the utterance is the target user or impostors like most text-dependent speaker verification system do but focus on studying not only the different behavioral biometrics between target user and impostors but also the effect of password contents. The difference between our lip-password speaker verification and most other speaker verification is that the latter tries to make the verification independent with the text to improve the robustness but the former focus exactly the opposite. Compared with lip-reading that do not wish the feature contain information of the hugely different personal speech style, we hope this difference can be as large as possible. Instead of utilizing the visual features for learning with traditional methods of acoustic speaker verification, such as Gaussian Mixture Model (GMM) [24] or Hidden Markov Model (HMM) [48] like most visual speaker verification systems do [34], we consider the visual signal as a series of actions of lips, and the lip password verification task can be solved like action recognition tasks.

In this thesis, we treat the lip movement as a series of "actions", and the verification task can be treated as action recognition in the video. So the lip sequence cannot be segmented but see it as a whole. In the meantime, the key action when speaking is found and select with some criteria to train the speaker's lip model. We propose a new concept, named "interest interval", which represents the key frames of lip movement. The interest intervals are detected first and then matched with the model trained by the user's data. According to the matching scores, the testing utterance is then accepted/ rejected by the system. Since the lip password is treated as some kind of actions, there is no limitation in language and lexical contents of

the password.

Further, we design a novel method, termed as overlapping subsequence matching, to encode the information of password content. In contrast to previous related methods of detecting the mouth status and segment the whole lip sequences into several independent subsequences with the risk of cutting across potentially discriminating features, our method allows each subsequence have overlap between successive subsequences and does not rely on the detection of mouth status, which is computationally heavy and sensitive to environments such as illumination.

Experiments show the promising results of the proposed algorithms.

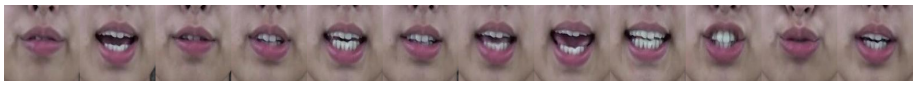
1.1.3 Lip Password-based Visual Speaker Verification with Sparse Coding

The objective of this work is to propose a novel approach to lip password-based speaker verification with arbitrary lip password, i.e. the password content to be a phrase of any languages, based on sparse coding. Since the language alphabet of the lip password is unknown, we design a representation of lip sequences which preserve the temporal order by employ a diagonal-like mask in pooling stage and build a pyramid structure to form the spatiotemporal lip representation containing the structural characteristic under lip password.

Currently, most speaker verification methods only focus on verifying the identity of the speaker, ignoring the temporal information of password content. As shown in Figure. 1.1, since Figure. 1.1 (a), (b) and (c) are spoken by the same person, the spatial-based features of these three sequences are similar. The proposed lip representation can keep this temporal order and improve the accuracy of verifying the password contents significantly. To further improve the accuracy of verifying the lip passwords, a pyramid structure of lips is proposed to contain more information in the spatial domain.

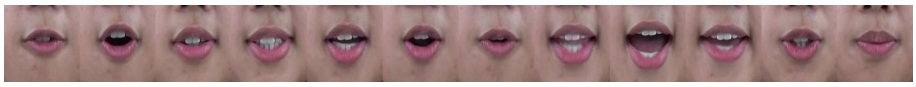
In contrast to the previous related methods of detecting the mouth status and segmenting the whole lip sequences into several independent subsequences with

(a) “Zhi Ma Kai Men” of user A



(b) “5683” of user A

(c) “4092” of user A



(d) “4092” of user B

Figure 1.1: Images of each row are sampled from the lip password sequences. Subfigure (a∼c) are different lip passwords spoken by the same user. If (c) is set as the correct lip password, then (a) and (b) are the case that the wrong password spoken by the target user, and (d) is the case that the impostor speaking the correct password. For a lip password-based speaker verification system, (a), (b) and (d) are all regarded as the impostor data and rejected.

the risk of cutting across potentially discriminating features [26, 45], the proposed method directly utilizes the temporal order structure of sparse representation and does not rely on the detection of mouth status, which is computationally heavy and sensitive to environments such as illumination.

In order to evaluate the benefits of the proposed lip representation in the lip password-based speaker verification system, we collect a database which contains both English and Chinese lip passwords. We empirically investigate the ability of state-of-the-art spatiotemporal lip features to verify the lip contents and speakers identity. Experimental results show that the proposed lip feature outperforms the state-of-the-art ones in all scenarios we have tried so far, especially when verifying the lip password contents.

1.2 Thesis Outline

The thesis is organized as follows. Chapter 2 gives the literature review of related work on visual speaker verification and lip password-based visual speaker verification. Chapter 3 describes two novel approaches, Interest Interval Model and Overlapping Subsequence Matching, for lip password-based visual speaker verification with Hidden Markov Models. Chapter 4 further develops a lip password-based speaker verification system with unknown language alphabet based on sparse coding. We propose a novel diagonal-like pooling method and the pyramid structure of lips based on sparse coding as the representation of the lip password. Chapter 5 presents the framework and implementation of a lip password-based visual speaker verification system, which is based on the algorithms of this thesis and designed for verifying the identity of the speaker based on their lip password without the limitation of language. Chapter 6 describes the conclusion of the thesis and discusses the future work.

Chapter 2

Related Works

In Chapter 1, we have introduced the development of speaker verification and introduce the concept of lip password-based visual speaker verification. In this chapter, we first present the existing visual feature representation modalities in 2.1. Then, Section 2.2 surveys some recent works with respect to the visual speaker verification systems. Finally, we give a brief introduction to the lip password-based speaker verification in Section 2.3.

2.1 Visual Feature Extraction

The visual feature of lip region has been demonstrated to encode rich information for lip motion. Compared with traditional biometric features such as the face, iris or fingerprint, lip features contain not only physiological information but also behavioral information [48]. There are several ways to categorize visual feature extraction methods of lips. In general, they can be roughly divided into two group: static methods and dynamic methods, which use static or dynamic information of lip region, and their combination which capture both types of information.

2.1.1 Static Methods

Static methods extract the information from a single frame of lip region that corresponding to a shape model of the lip contour or the texture of the mouth area.

There are mainly two types of features: contour-based features and appearance-based features.

For the contour-based features, the geometric shape parameters are used as the visual features[7]. The contour-based features are usually described with the parameters of the geometric properties of lip shapes, such as the height and width of mouth, contour perimeter, and lip area. To obtain this type of features, the extraction of lip contours is often the first step by applying the lip modeling techniques. In [39], Petajan extracts the contour-based features from the binary thresholded lip images. Chiou *et al.* [10] utilize the Active Contour Model (ACM) to model the lip shapes by eight radial vectors as the geometric parameters of the shape of lips. Luettin *et al.* [30] employ an Active Shape Model (ASM) to extract the contour of lips as the shape features for the lipreading system, while Matthews [32] utilize the Active Appearance Model (AAM). Different from those works that use a deformable template to obtain the shape of lips, Zhang *et al.* [53] and Broun *et al.* [6] extract the lip contours by segmenting the lip images and then derived the geometric parameters. In [19], Kaynak *et al.* conducts a comprehensive investigation of lip contour features.

For the appearance-based features, the textures of the mouth region, including lips, teeth, and tongue, have shown the effectiveness. Compared with the contour-based features, the appearance-based features represent the low-level information of the lip region, which provides the complementary information to the contour-based features. The raw pixel values of lip images can be utilized as the appearance-based features without losing any information [5]. However, such a feature contains too much redundant information and the dimensionality is always very high. To solve this problem, one way is to reduce the dimensionality of the raw pixel features by employing the transform coefficients of the images to compact the main appearance characteristics within a smaller dimension. There are some works use the transforming coefficients such as Principal Component Analysis (PCA), Independent Components Analysis (ICA) and two-dimensional Discrete Cosine Transform (2D-DCT) to

extract the textures of the mouth region [7, 47, 33]. Another way is to extract the texture features from lip images. Since the features that are directly extracted from the raw pixels are usually not invariant to the changes of illumination variations or angle alteration, it is more often to utilize the features extraction methods which are invariant to those changes, such as Histogram of Oriented Gradient (HOG) [12] and Local Binary Patterns (LBP) [54].

2.1.2 Dynamic Methods

Compared with the static methods which extract the features from one single lip image, dynamic methods extract the features from the subsequent frames of lip movements, which contain the intrinsically temporal information.

Optical flow[28, 1] is one of the most straightforward methods to reveal the temporal changes of lip pixels. Mase *et al.* [31] utilize the statistic characteristics of optical flows within four windows as the visual features for the lipreading system. In [7], Cetingul *et al.* compute the 2D-DCT coefficients extracted from the optical flow for visual speaker verification. In [15], Faraj *et al.* calculate the 2D velocity vectors of optical flows corresponding to the structure tensor for the visual speaker verification system.

2.1.3 Hybrid Methods

In the literature, some works have utilized the spatiotemporal feature directly too. For instance, Zhao *et al.* [55, 56] proposed to extract the Local Binary Patterns (LBP) features with Three Orthogonal Planes (TOP) to generate local spatiotemporal descriptors for lipreading. Chan *et al.* [8] proposed the Local Ordinal Contrast Pattern with Three Orthogonal Planes (LOCP-TOP) for lip-based speaker verification. The LOCP is a texture descriptor that encodes the appearance of lip images. Meanwhile, by using the LOCP in TOP, it makes the final lip representation keep the dynamic information of lip movement. In [37], the spatiotemporal histogram of oriented gradients is extracted as the features for lipreading system which contain

both spatial and temporal information of lip motions.

2.2 Visual Speaker Verification

Visual speaker verification aims to verify whether the speaker is the one who he or she claims to be [21], according to the visual information. There are several studies in this area.

2.2.1 Statistical Models

Several works have been done on visual speaker verification, which regard the lip motion as the sequential data and employ statistic methods to train the speaker's model. For example, Saeed [43] extracted the static and dynamic lip features based on lip contour detection and utilized the Gaussian Mixture Model (GMM) to model the lip movements for person recognition. Wang and Liew [48] utilized the GMM to train the static features and Hidden Markov Model (HMM) for dynamic features. Also, Cetingul *et al.* [7] evaluated the explicit lip motion information comparing with lip intensity and geometry features through HMM for speaker identification. However, in most of those works, the users pronounce the same phrase for verification, which is the most constrained case as both duration and text are fixed. Moreover, the scenario of target user saying the phrase incorrectly is not considered.

2.2.2 Sparse Coding

In [21], Lai *et al.* used sparse coding under a hierarchical spatiotemporal structure to form the lip representation. The dictionary is learned from all users and max-pooling in the hierarchical structure is performed in the sparse representation of lip subsequences. These lip representation densely extract lip features over time and space domains to avoid the segmentation and modeling of lip sequences, leading to a better generative performance comparing with the model-based approaches [21]. Nevertheless, the world model used in visual speaker authentication [21] is

not realistic for a speaker verification system that has new users enrolling. Also, the training of dictionary is also time-consuming. In contrast, our method uses fixed dictionary which is chosen from the training data and thus does not require dictionary learning on the data of whole users.

2.3 Lip password-based Speaker Verification

Most recently, Liu and Cheung [26, 25] have proposed a novel concept of lip password, which is composed of a password embedded in the lip movement and underlying characteristic of lip motion. They proposed a method based on appearance and contour features and multi-boosted HMMs learning approach for the lip password-based speaker verification system. Experiments in [26] have shown that their method performs favorably well to verify both of the private password information and the lip biometrics of speaker.

Furthermore, to improve the performance of speaker verification system, some works in the literature perform detection approaches on the mouth status of "closing" and segment the whole lip sequences into several subsequences on those places [27, 46]. For instance, Liu and Cheung [26] extracted the lip contour by tracking [9] first and detect mouth status according to the area of lips. Karlsson *et al.* [18] used the optic flow to perform lip motion events detection and lip segmentation. Shaikh *et al.* [45] proposed a segmentation method based on the pair-wise pixel comparison of consecutive lip images. However, these methods partition the lip sequences into independent subsequences with the risk of cutting across potentially discriminating features, like the transformation between password units. What is more, the computations are somewhat laborious and the results are sensitive to the environment such as illumination. In addition, some words with the mouth close deeply, e.g. "me", are departed.

Chapter 3

Interest Interval Model and Overlapping Subsequence Matching Approach to Lip Password-based Visual Speaker Verification with Hidden Markov Models

3.1 Introduction

In this chapter, we aim at building a visual speaker verification system based on lip password without knowing the password alphabet beforehand. Instead of recognizing the elements of password individually, we compare the whole lip movements to verify if the utterance belongs to the target user or impostors. We focus on studying not only the different behavioral biometrics between target user and impostors but also the effect of password contents. The difference between our lip-password speaker verification and most other speaker verification is that the latter tries to

make the verification independent with the text to improve the robustness but the former focus exactly the opposite. Compared with lip-reading that do not wish the feature contain information of the hugely different personal speech style, we hope this difference can be as large as possible. Instead of utilizing the visual features for learning with traditional methods of acoustic speaker verification, such as Gaussian Mixture Model (GMM) [24] or Hidden Markov Model (HMM) [48] like most visual speaker verification systems do [34], we consider the visual signal as a series of actions of lips, and the lip password verification task can be solved like action recognition tasks.

In this chapter, we treat the lip movement as a series of "actions", and the verification task can be treated as action recognition in the video. As a result, the lip sequence is no need to be segmented but seen as a whole. In the meantime, the key actions during the speech are found and selected with specific criteria to train the model of the speaker's lip password. We propose a new concept, named "interest interval", which represents the key frames of lip movement. The interest intervals are detected first and then matched with the model trained by the user's data. According to the matching scores, the testing utterance is then accepted/ rejected by the system. Since the lip password is treated as some kind of actions, there are no limitations in language and lexical contents of the password.

We further design a novel method, termed as "Overlapping Subsequence Matching", to encode the information of password contents. In contrast to previous related methods of detecting the mouth status and segment the whole lip sequences into several independent subsequences with the risk of cutting across potentially discriminating features, our method allows each subsequence have overlap between successive subsequences and does not rely on the detection of mouth status, which is computationally heavy and sensitive to environments such as illumination. To achieve this, we first sample the feature sequence extracted by HOG descriptor [12] into overlapping subsequences with fixed sampling length and overlapping rate. Each subsequence is then treated as a single unit to train the corresponding HMM.

The verification result depends on the matching scores over all of those subsequences predicted by corresponding HMMs.

Experiments on a Chinese lip password database show the promising results of the proposed algorithms.

The remaining part of this chapter is organized as follows: Section 3.2 gives an overview of the related work. Then, the proposed Interest Interval Model approach is described in detail in Section 3.3. After that, the Overlapping Subsequence Matching algorithm is provided in Section 3.4. Finally, we draw a summary in Section 3.5.

3.2 Overview of Hidden Markov Model

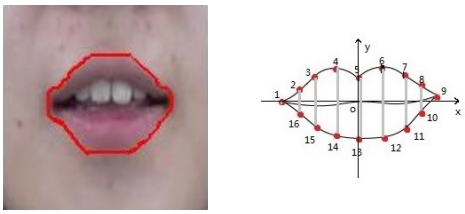
To distinguish the feature vector $\boldsymbol{F}$ extracted from the video, the verification of lip password becomes a typical binary classification task of time series.

Hidden Markov Model (HMM) is one of the most popular methods to be utilized in speaker verification domain. The dataset of target speaker saying correct password is utilized to train an HMM with parameter λ to be learned using Baum-Welch algorithm [40]. The classification of data is performed based on the log-likelihood (LL):

$$\begin{aligned} \mathrm{LL}(\boldsymbol{F}) &= \log P(\boldsymbol{F}|\lambda), \\ if\ \mathrm{LL}(\boldsymbol{F}) \geq \tau &: accepted. \\ Otherwise &: reject. \end{aligned} \tag{3.1}$$

$\boldsymbol{F}$ is the feature sequence of data, and λ can be calculated by the forward algorithm [40]. In general, the feature $\boldsymbol{F}$ will be rejected if $\mathrm{LL}(\boldsymbol{F})$ is less than the threshold, and vice versa.

HMM performs outstandingly on acoustic signal applying in speaker verification and speech recognition [13]. In the visual speaker verification system, it is also very common to utilize HMM-based methods[26]. However, its performance on visual features is usually not as good as that in the acoustic domain [50]. One plausible reason is that the appropriate visual features would not have been found to con-



(a) (b)

Figure 3.1: The 7-2-7 lip model

vey enough information comparing with the popular acoustic features, e.g. Mel frequency cepstral coefficients (MFCC). Another reason is that the structure of the HMM might not be an appropriate model to describe the true data process, especially when the data sequence is long. In fact, the acoustic signals are often cut into small subsets in silence moments to shorten the sequence length. By contrast, visual signals, however, do not have such significant status in general. In the literature, Liu and Cheung [26] have separated the sequence into small pieces when the mouth is "closed". Also, Karlsson and Bigun [18] utilized the optical flow to detect the mouth events of opening and closing. However, the lip movement is highly relative to the former words. Furthermore, some words with mouth close deeply, e.g. "me", are departed, resulting in incorrect recognition results.

3.3 Interest Interval Model with HMM

To verify a password embedded in the lip movement as a whole, the features in those frames that contain significant changes of lips can be the unit of lip password model instead of the linguistic contents. In the following, we show how to detect those units, namely "interest intervals", as well as the model matching and training algorithms of the proposed Interest Interval Model (IIM), for representing lip password.

3.3.1 Feature Extraction

The appearance features that contain every pixel in the lip area are not effective due to the high dimension and sensitiveness to image noise and illumination [26], as well as the intensity difference between different datasets [7]. As we need to measure the movement of lips, it is reasonable to assume the information mostly lies in the shape of lips and its dynamic changes over speaking. To this end, the contour of lips should be extracted first to represent the lip image in a more effective way.

First of all, one of the most popular image segmentation algorithms, GrabCut [42, 4, 3, 20], is applied to extract lip contours frame by frame. An example of contour detection result is shown in Figure. 3.1(a). Secondly, we employ a 7-2-7 lip model (i.e. 7 points averagely divide the upper and lower lip contours, respectively, and 2 points represent the lip corners) to describe the lip contours, which is modified from the 5-2-7 model proposed by Wang [48]. After that, the shape features are normalized to have the same width as shown in Figure. 3.1(b).

The normalization process is shown as follows [48]. For the shape features of one frame $\{x_1, x_2, \ldots, x_{16}, y_1, y_2, \ldots, y_{16}, \}^\intercal$, where (x_i, y_i) is the position of the ith point, the center and width of lips are calculated by

$$\begin{aligned} x_c = \frac{(x_1 + x_9)}{2}, y_c = \frac{(y_1 + y_9)}{2}, \\ s = \frac{\sqrt{(x_1 + x_9)^2 + (y_1 + y_9)^2}}{2}, \end{aligned} \tag{3.2}$$

where (x_1, y_1) and (x_9, y_9) are left and right corners of lips. The normalized points are calculated by

$$\begin{aligned} x'_i = \frac{(x_i - x_c)}{s}, \\ y'_i = \frac{(y_i - y_c)}{s}, i = 1, 2, \ldots, 16. \end{aligned} \tag{3.3}$$

After the normalization process, y_1 and y_9 are fixed at 0. All x values are also unchanged between frames, which has little information about the shape. Hence, we select the position of y axis of 14 points representing upper and lower lips contours, respectively, i.e. $\boldsymbol{f} = \{y_2, \ldots, y_8, y_{10}, \ldots, y_{16}\}^\intercal$. Define $\boldsymbol{f}(t)$ as the tth contour feature, the final feature vector extracted from a lip video with N frames is

$\boldsymbol{F} = \{\boldsymbol{f}(t)\}, t = 1, 2, \ldots, N$. To avoid some mistakes made by contour detection of several frames, the low-pass Butterworth filter [41] is utilized to smooth the feature sequences.

3.3.2 Interest Intervals

Analogous to action recognition, we would like to detect those key frames with high discriminative power and utilize them to formulate the model for lip password verification. We assume that the key frames are those frames with rapid changes of feature $\boldsymbol{F}$. We define the intervals corresponding to these key frames as "interest intervals". We will introduce how to find them as follows.

Consider the position of y_2 as an example, denoting as $f(t) = y_2(t), t = 1, 2, \ldots, N$, we define the change c of $f(t)$ produced by a shift Δt at point t:

$$c(t; \Delta t) = \sum_{t_i} w(t_i)[f(t_i + \Delta t) - f(t_i)]^2 \tag{3.4}$$

where $w(t)$ is a Gaussian window and t_i is the points in the window w centered on t. According to Taylor series, we have

$$f(t_i + \Delta t) \approx f(t_i) + f'(t_i) \cdot \Delta t \tag{3.5}$$

where $f'(t_i)$ is the derivative at t_i. Substituting approximation (3.5) into Eq. (3.4),

$$\begin{aligned} c(t; \Delta t) &= \sum_{t_i} w(t_i)[f(t_i + \Delta t) - f(t_i)]^2 \\ &\approx \sum_{t_i} w(t_i)[f(t_i) + f'(t_i) \cdot \Delta t - f(t_i)]^2 \\ &= \sum_{t_i} w(t_i)[f'(t_i) \cdot \Delta t]^2 \\ &= (\Delta t)^2 \cdot \sum_{t_i} w(t_i) f'(t_i)^2 \\ &= (\Delta t)^2 \cdot C(t) \end{aligned} \tag{3.6}$$

where $C(t) = \sum_{t_i} w(t_i) f'(t_i)^2$ captures the change structure of function $f(t)$. To obtain the total change of the 14 points, we sum up $C(t)$ of all 14 points:

$$C^*(t) = \sum_{y_j \in \boldsymbol{f}} C_{y_j}(t) \tag{3.7}$$

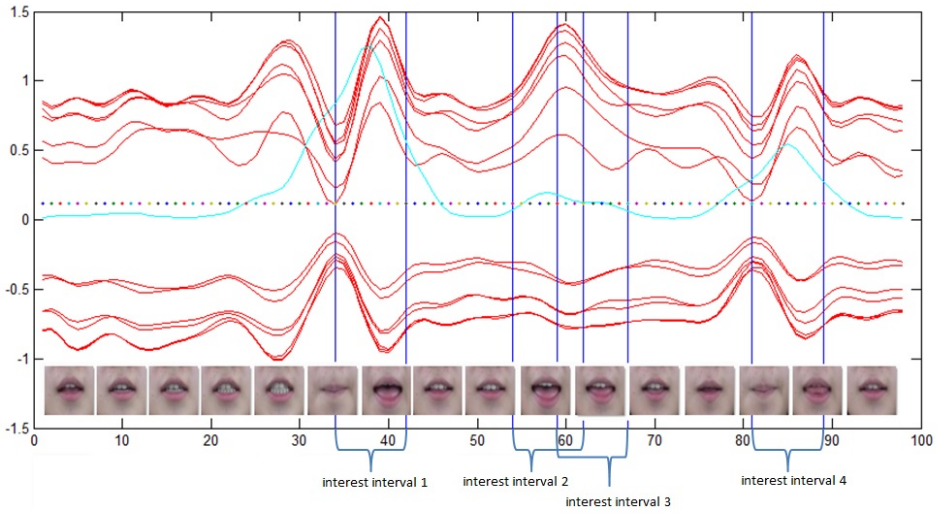


Figure 3.2: An example of interest intervals detection results. The red lines denote the lip shape features, the cyan line denotes $C^*(t)$ and the dotted line denotes threshold. Four interest intervals are detected in those frames with rapid changes of lip shapes. The overlap between interest intervals are allowed in order to keep the whole lip movements.

The local maximum of $C^*(t)$ is the center position of key frames of lip movement.

After calculating $C^*(t)$ and its local maximum, we set a threshold to avoid the influence of slight changes and noise produced by contour detection. For one selected maximum at position p with the Gaussian window of variance σ^2, the region of interest interval is defined as $t \in [p - 2\sigma, p + 2\sigma]$. Subsequently, the corresponding feature in this interval is defined as $\boldsymbol{I}(p) = \{\boldsymbol{f}(t)|t \in [p-2\sigma, p+2\sigma]\}$. One example of the detection result of interest intervals is shown in Figure 3.2. Then, we can utilize those interest intervals to train an HMM and use Eq. (3.1) to calculate the probability of the interest intervals produced by the HMM. Consequently, the interest interval can be classified into the HMM with largest log-likelihood.

3.3.3 Interest Interval Model

To obtain a model of speaker's lip password, we define the structure of the model based on the Interest Interval, named Interest Interval Model (IIM), first. Given n interest intervals found in training sequences, we define the model by summarizing all sets of similar intervals that appear in every training sequence $M = \{m_i = \{p_i, \lambda_i, c_i\} | i = 1, \ldots, k\}$, k is the number of interval sets. The position p_i, the corresponding trained HMM parameters λ_i and the log-likelihood center c_i are trained and calculated by the intervals belonging to the ith interval set.

In the following, we describe the procedure of computing matching scores between the model and a data first, and then of model training.

Computation of Matching Scores

Given all the interest intervals detected from a data sequence $T = \{\boldsymbol{I}_j^t = \boldsymbol{I}(p_j^t) | j = 1, \ldots, l\}$, l is the number of interest intervals, the match between the model $M = \{m_i = \{p_i^m, \lambda_i, c_i\} | i = 1, \ldots, k\}$ and the data is defined by a weighted sum of similarity s between the model features and the data's intervals

$$S(T|M) = \sum_{i=1}^{k} s(\boldsymbol{I}_j^t | m_i) e^{-(p_i^m - p_j^t)^2/\omega} \tag{3.8}$$

where s is the similarity score between $\boldsymbol{I}_j^t$ and m_i, ω is the variance of the exponential weighting function that gives more importance to the intervals that appear in closer place. For the ith interval set of the model, j is chosen by

$$j = \arg\max s(\boldsymbol{I}_j^t | m_i) e^{-(p_i^m - p_j^t)^2/\omega}. \tag{3.9}$$

We define the similarity score of interest interval in data $\boldsymbol{I}^t$ corresponding to the features in the model $m = \{p^m, \lambda, c\}$ by:

$$s(\boldsymbol{I}^t | m) = \frac{1}{c - \log P(\boldsymbol{I}^t | \lambda)}, \tag{3.10}$$

where $P(\boldsymbol{I}^t | \lambda)$ is the probability of the interest interval in data produced by the corresponding HMM in training model.

To find the best match between the model and the data sequence, we align every interval in the model with the data's intervals. We do this by searching the interest intervals of the data that maximize the matching score $S(T|M)$ for every m_i in the model according to Eq. (3.9). The details of computing matching scores are given in Algorithm 1.

Algorithm 1 Computation of Matching Scores

Input: The training model $M = \{m_i = \{p_i^m, \lambda_i, c_i\} | i = 1, \ldots, k\}$ and interest intervals of a data sequence $T = \{\boldsymbol{I}_j^t = \boldsymbol{I}(p_j^t) | j = 1, \ldots, l\}$.

Output: Matching score $S(T|M)$.

1: **for** $i = 1$ to k **do**

2: Choose the jth interest intervals of the data that maximize the matching score via Eq. (3.9).

3: Add the maximum score to the total matching score.

4: **end for**

Model Training

In order to train the model using several training data of target speaker's lip password, the model should be matched with those training data very well. Hence, we try to find the most similar interest intervals that appear in every sequence to represent the lip password model. With those interval sets founded, all the interest intervals in the same interval sets are utilized to train the HMM parameters λ and calculate the average positions and log-likelihoods.

The training model is firstly initialized with the interest intervals of the first training data sequence. All the interest intervals are included in the initial model. After that, we utilize the following data sequences to readjust the model one by one. To make the new model match training data with a larger score, the alignment between data and model which is applied in model matching is also utilized here. We align every interval in the data to the interval set of model. After the alignment, those interval sets without any match with data are discarded and the intervals

in the data that match with the same interval set are also discarded except the interval with the maximum similar score. After that, the interval set is updated by calculating $\{p_i^m, \lambda_i, c_i\}$ again with the new interest interval. The final model will be formed by the most representative interest intervals that all the training data of lip passwords can be matched with the high similarity score. The corresponding algorithm is presented in Algorithm 2.

Algorithm 2 Model training

Input: N training sequences of the target user speaking correct password.

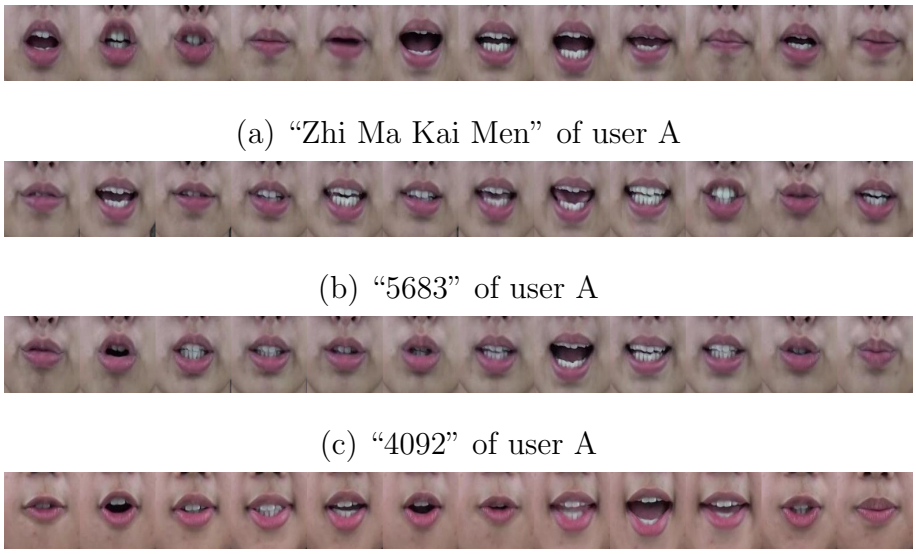
Output: The Interest Interval Model M.

1: Initialize the model M^1 with the interest intervals of the first training data sequence.
2: **for** $i = 2$ to N **do**
3: Find the interest intervals of the ith training data: T^i.
4: **for** $k = 1$ to N_{T^i} **do**
5: Choose the jth interval sets of M^{i-1} that

$$j = \arg\max s(\boldsymbol{I}_k^{T^i} | m_j) e^{-(p_j^m - p_k^{T^i})^2/\omega} \tag{3.11}$$

6: Classify the kth interval of training sequence T^i to the jth interval sets of model M^{i-1}.
7: **end for**
8: Discard the interval sets of model with no classified intervals found.
9: Select one interval among the matching intervals for every interval sets in the model M^{i-1} with max similarity score under the weight of time.
10: Training the HMM and calculate new parameters with new interest interval set as new model M^i
11: **end for**

With the training model, we can then calculate the similarity score of those correct lip password and impostors by Algorithm 1.



(a) “Zhi Ma Kai Men” of user A

(b) “5683” of user A

(c) “4092” of user A

(d) “4092” of user B

Figure 3.3: Images of each row are sampled from the lip password sequences. Subfigure (a∼c) are different lip passwords spoken by the same user. If (c) is set as the correct lip password, then (a) and (b) are the case that the wrong password spoken by the target user, and (d) is the case that the impostor speaking the correct password. For a lip password-based speaker verification system, (a), (b) and (d) are all regarded as the impostor data and rejected.

3.3.4 Experiments

The Database

In the visual speaker verification domain, existing databases such as XM2VTS [16] and MVGL [7] have been widely utilized [26]. However, these databases are incompetent for the study on lip password for the insufficient variability of pass-phrase, which makes it impossible for any study on the effect of password variability [22]. Therefore, we constructed a database consisting of Chinese lip passwords to evaluate the proposed algorithm. Some samples from the database is shown in Figure. 3.3. Each utterance contains about 90 lip images of size 131×131 lasting for about 3 second. The length of utterance is controlled by the users. The duration variations over all password from all speakers are illustrated by the histogram as shown in

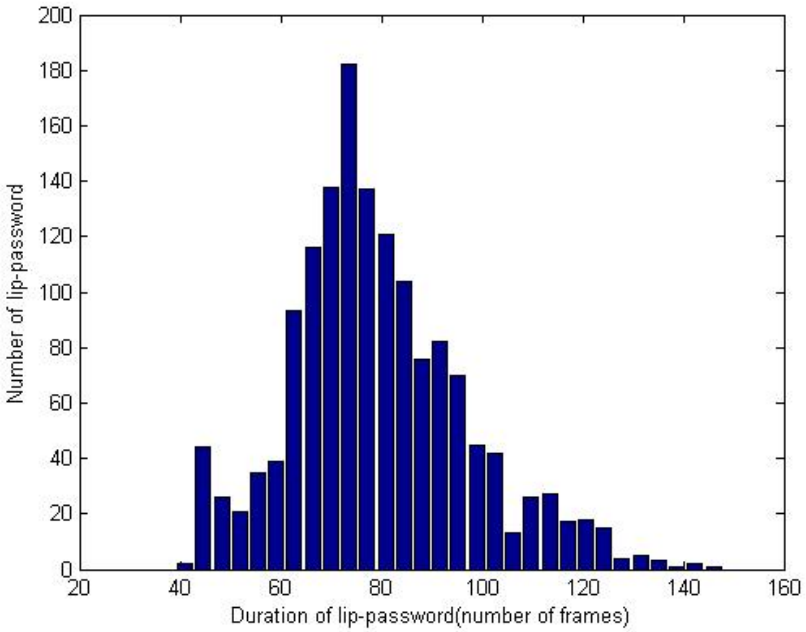


Figure 3.4: Distribution of utterance duration for all 35 sentences from 43 speakers in database.

	Correct Password	Wrong Password
Target User	**Target-Correct**	Target-Wrong
Impostor	Impostor-Correct	Impostor-Wrong

Table 3.1: Different types of scenarios in the database.

Figure 3.4.

The database can be categorized into four types according to if the utterances are spoken by the target users or impostor with correct or wrong passwords, as summarize in Table 3.1.

Experimental Setting

To evaluate the performance of speaker verification systems, the database is randomly divided into two parts for training and testing. The overall impostor dataset includes three scenarios: Target-Wrong, Impostor-Correct and Impostor-Wrong.

The false acceptance rate (FAR) and the false rejection rate (FRR) are calculated

Scenario	Evaluation	Our method	HMM	GMM
Sum-Impostor	AUC	0.957 ± 0.012	0.876 ± 0.014	0.905 ± 0.012
	EER(%)	6.26 ± 1.38	18.39 ± 1.34	15.01 ± 1.39
	FRR(%)	5.62 ± 1.66	26.72 ± 2.72	18.18 ± 2.80
Target-Wrong	AUC	0.886 ± 0.012	0.760 ± 0.019	0.758 ± 0.017
	EER(%)	16.43 ± 1.44	29.56 ± 1.86	27.49 ± 1.71
	FRR(%)	18.68 ± 2.30	52.85 ± 3.46	65.54 ± 2.86
Impostor-Correct	AUC	0.957 ± 0.011	0.862 ± 0.014	0.896 ± 0.014
	EER(%)	6.82 ± 1.23	18.87 ± 1.30	15.73 ± 1.68
	FRR(%)	5.97 ± 1.55	28.05 ± 3.02	18.76 ± 3.00
Impostor-Wrong	AUC	0.963 ± 0.010	0.886 ± 0.013	0.913 ± 0.011
	EER(%)	5.66 ± 1.29	17.56 ± 1.43	14.13 ± 1.35
	FRR(%)	5.19 ± 1.45	25.56 ± 2.95	17.79 ± 2.78

Table 3.2: Means and standard deviations of AUC, EER and the value of FRR when FAR is fixed to 10% are calculated over ten times experiment results.

as follows:

$$\text{FAR} = \frac{N_{FA}}{N_{IM}} \times 100\% \tag{3.12}$$

$$\text{FRR} = \frac{N_{FR}}{N_{CL}} \times 100\% \tag{3.13}$$

where N_{CL} and N_{IM} are the total number of testing client and impostor samples, N_{FA} is the number of testing impostor samples being falsely accepted and N_{FR} is the number of testing client samples being falsely rejected. The experiments of the overall impostors as well as three subset scenarios are evaluated separately to show the performance of the proposed algorithm.

Experimental Results

The HMM and GMM algorithms are chosen as the baseline to compare with the proposed algorithm using the PMTK toolkit [35]. In the experiments, the Gaussian

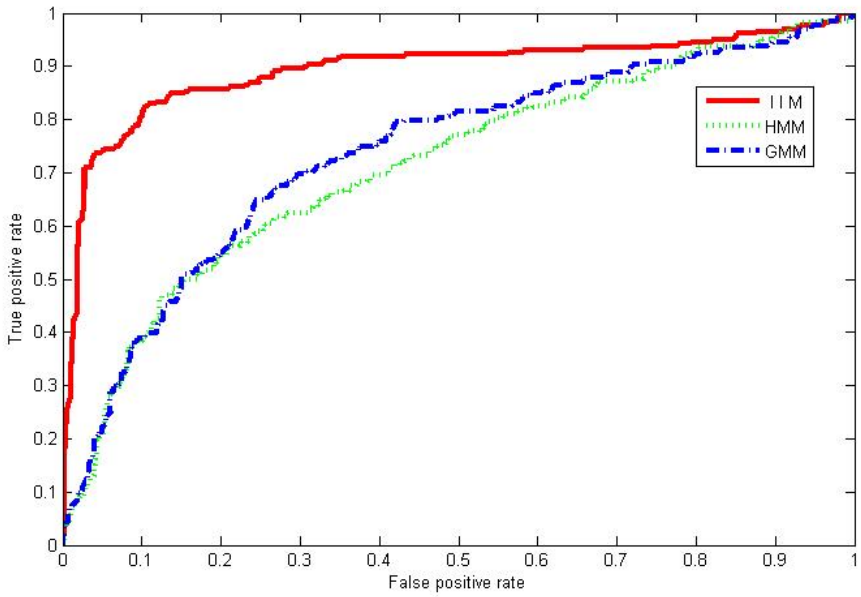


Figure 3.5: The ROC curve of the first scenario that the target user speaking the wrong passwords are considered as the impostor samples.

window of interest intervals is set at $\sigma = 2$ and the threshold is set at 0.12, the weight of time when calculating the similarity scores is set at $\omega = 100$. The performance of these algorithms are compared in terms of Receiver Operating Characteristic (ROC) curve, Area Under the Curve (AUC) of ROC, Equal Error Rate (EER), and the value of FRR when FAR is fixed to 10%.

With the same client samples of the target speaker saying the correct password, the following 4 kinds of datasets are selected to be the impostor samples: (1) the target speaker saying wrong passwords; (2) the impostor saying the correct password; (3) the impostor saying the wrong passwords, and (4) all those three kinds of impostors.

Analysis of Different Scenarios As we mentioned above, the algorithms are evaluated in four scenarios according to the impostor types.

The ROC curve of the first scenario result is shown in Figure 3.5, where the target speaker saying the wrong passwords is considered as the impostors. The

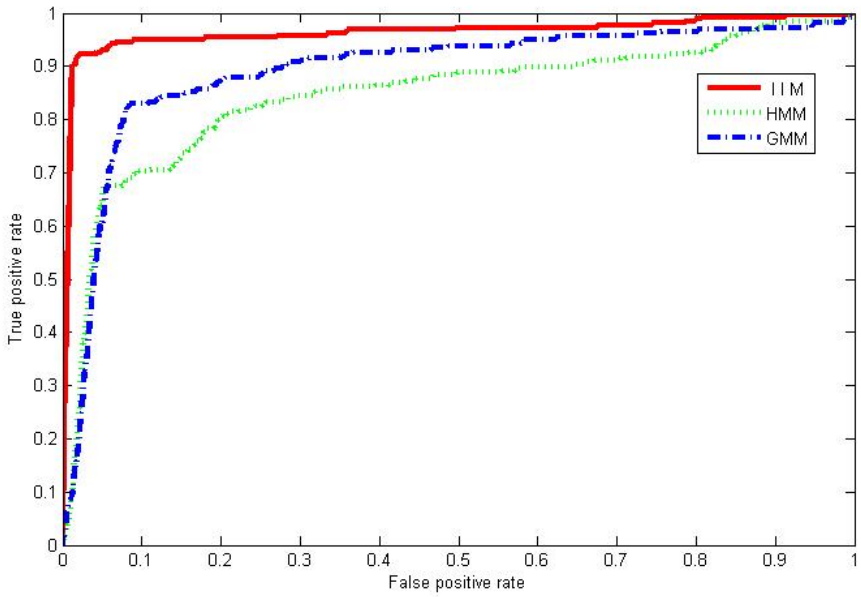


Figure 3.6: The ROC curve of the second scenario that the impostors speaking the correct passwords are considered as the impostor samples.

correct passwords and all the five kinds of wrong passwords spoken by the same speakers are selected to evaluate the performance of the three speaker verification algorithms. It can be seen that the proposed algorithm significantly outperforms the baselines in this scenario. It was also found that HMM and GMM would degrade rapidly when the passwords are similar, such as the correct password "Zhi Ma Kai Men" and wrong password "Kai Men Jian Shan", in which two Chinese characters are the same. In the proposed algorithm, the intervals with the large information about the lip movement are detected and utilized for training the model.

The ROC curve of the second scenario result is shown in Figure 3.6, where the impostors saying the correct passwords are considered as the impostors. For every person in the database, all other 42 people saying correct password are selected as impostor samples. The EERs of all the three algorithms are lower than 20%. It implies that even the impostors know the correct password, the underlying behavioral characteristics are different between different people. In this scenario, the proposed algorithm is also significantly better than the baseline, which EER is less than 10

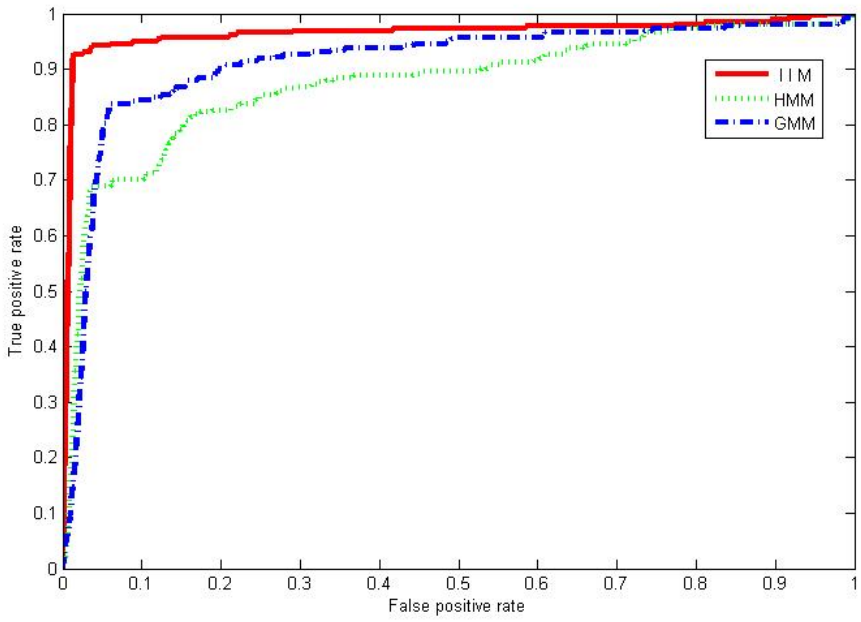


Figure 3.7: The ROC curve of the third scenario that the impostors saying the wrong passwords are considered as the impostor samples.

%. The experiment results show that the proposed algorithm can successfully discriminate between people.

The ROC curve of the third scenario result is shown in Figure 3.7, where the impostors saying the wrong passwords are considered as the impostors. For every person in the database, all other 42 people speaking five kinds of wrong passwords are selected as impostors. This is the most simple scenario because the difference lies in not only the password contents but also the speaker's identity.

The ROC curve of the final scenario result is shown in Figure 3.8, where all the first three kinds of impostors are considered. The verification algorithm will set the threshold according to the FAR and FRR of the whole dataset, which includes all the situations that may occur from a practical viewpoint. The experiment results show the total performance of the proposed algorithm and the baselines. Once again, the proposed algorithm outperforms the other two methods.

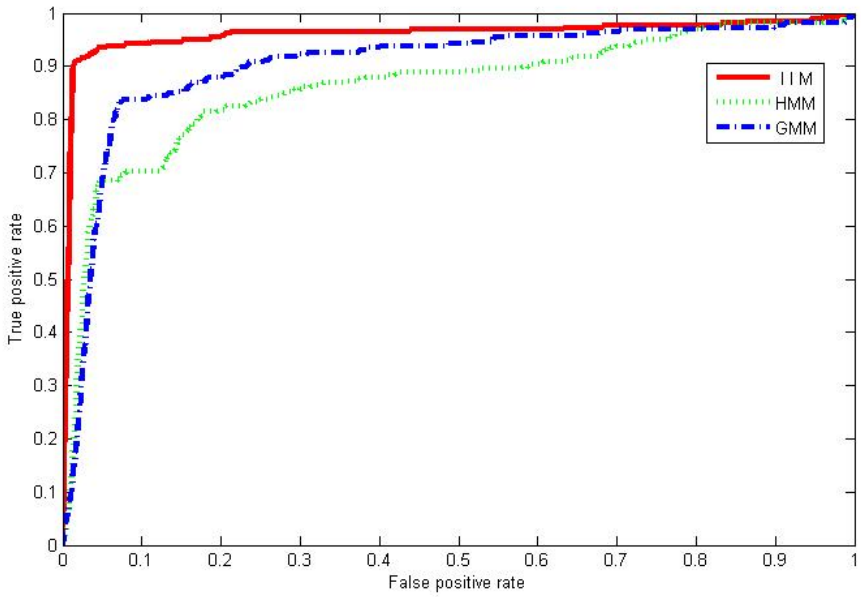


Figure 3.8: The ROC curve of the fourth scenario that all kinds of impostor are considered as the impostor samples.

Statistical Comparison with the Baselines To compare the proposed algorithm statistically with the two baselines algorithms in the four scenarios, the paired two-sample t-test is used to show the statistical difference of results, at 0.01 significance level.

The experiment is run for ten times with randomly choosing the training and testing datasets. AUC, EER and the value of FRR when FAR is fixed to 10% are calculated from the ROC curves to quantify the performance of the algorithms. Table 3.2 present their means and standard deviations over ten times experiment results.

The experiment results show that our algorithm statistically outperforms the baselines and reaches the EER of 6.26% in total.

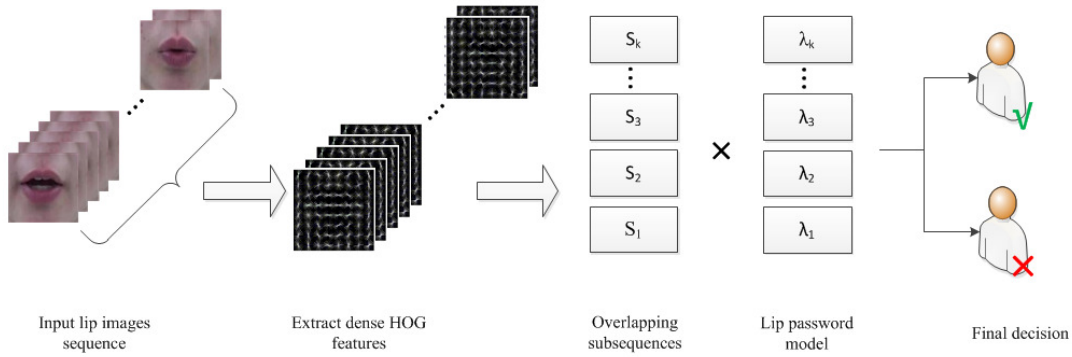


Figure 3.9: The stages of our speaker verification pipeline. For a testing lip images sequence shown in the leftmost, the dense HOG feature is extracted first to generate a feature sequence. The feature sequence is then partitioned into overlapping subsequences and matched with corresponding HMM in lip password model trained in enrollment step. Finally, the final decision depends on the similarity of every subsequence.

3.4 Overlapping Subsequence Matching with HMM

This section will present a speaker verification system via arbitrary lip passwords based on the proposed overlapping subsequence matching approach, as shown in Figure. 3.9. We first describe the details of lip feature extraction in our system and then introduce our overlapping subsequence matching approach.

3.4.1 Feature Extraction

There are mainly three kinds of lip features: texture features, shape features and motion features. Texture features that contain not only lips but also teeth and tongue appearing during speaking obtained by using Principal Component Analysis (PCA) and two dimensional Discrete Cosine Transform (2D-DCT) to impress the raw images of lips, and contour feature extracted by a lip tracking method are used in [26] as the features of lip password. However, the texture features extracted from PCA and 2D-DCT is sensitive to illumination, and lip tracking results might not accurate in some frames and causing noise on features.

In [38], Pei et al. use HOG features to describe the patches in the contours around lips and jaws obtained by an Active Appearance Models tracker for lip reading. HOG is based on the distribution of intensity gradients and edge orientations which make it very suitable to describe the shape of objects and insensitive to illumination. To avoid the noise caused by the failure of lip tracking, we use a dense HOG descriptor on lip video frame by frame. This dense HOG descriptor is of high dimension, which contains much redundant information and increases the computational cost of the HMM training in the enrollment step. Therefore, we perform PCA to reduce the dimension of the dense HOG descriptor as the features of lip images. All the features from the lip video are then concatenated into a feature sequence.

For a lip video with T frames, the feature sequence is denoted as $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$, where the dimension of $\boldsymbol{x}_i, i \in [1, T]$ is denoted as dim.

3.4.2 Overlapping Subsequence Matching

Overlapping subsequence matching works by sampling the feature sequences extracted from lip videos into overlapping subsequences and matching them individually by HMMs. We first describe the overlapping sampling and then introduce subsequence matching of our method.

Overlapping Sampling

Sampling technique with overlap is a common method to extract the observation sequences of HMM. For instance, in [11], the audio features, Mel-Frequency Cepstral Coefficients (MFCCs), are calculated from a 20ms window sampling on the original audio signal with a 50% overlap as the observation feature sequences of HMM for automatic speech recognition. Sampling with no overlap implies a risk of cutting across potentially discriminative features and requires accurate alignment [11]. In our method, the overlapping subsequences are obtained by similar sampling technique not for generating feature sequences but to partition the feature sequences into several overlapping subsequences.

Let L be the ratio of subsequence length to sequence length and α be the overlapping rate. The number of subsequences n is given by:

$$n = \phi\left(\frac{1-L}{L(1-\alpha)}\right) + 1 \tag{3.14}$$

where $\phi(x)$ is the largest inter r such that $r \leq x$.

To apply our method to speaker verification system via arbitrary lip passwords, the overlapping rate is set at 50% and $L = 1/k$, where k is the number of password units. For a feature sequence $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$ extracted from the lip video with T frames, the subsequence length is T/k. The number of subsequence is $2k+1$ according to Eq. 3.14. The set of feature subsequences is then denoted as $S = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_{2k+1}\}$, where the $\boldsymbol{s}_{2i+1}, i \in [1, k]$ is approximately the feature sequence of the ith password unit, and $\boldsymbol{s}_{2i}, i \in [1, k-1]$ is the feature sequence of the transition between the ith and the $(i+1)$th password unit.

Noted that if $\alpha = 0$, it becomes a non-overlapping sampling method, which cuts the feature sequence into several subsequences equally. It is similar to some previous works that segment the input sequence into subunits. As we all know, the audio signal between two words are silent, so it is reasonable to segment the sequence in those silent place. However, when the user says a phrase as the lip password, the lip motion between two words is usually not frozen. The transitions between different words are quite different, which embed potential distinctive power for verifying the password contents. The overlapping sampling strategy not only consists of the password units but also the transition between consecutive units, leading to the simplicity but efficiency of our method.

Subsequence Matching

Before explaining the subsequence matching method, we describe the matching between the whole sequences based on HMM first, which is a special case of our method when $L = 0$ in sampling step.

In training step, the whole feature sequences extracted from lip password training samples, which are collected by target user saying correct password, are used to train

an HMM with parameter λ to be learned using Baum-Welch algorithm [40]. The HMM then is used to calculate the log-likelihood of feature sequences, which is calculated by the forward algorithm [40].

After training the HMM, the average of log-likelihood of each training samples is defined as the center of this lip password model, denoted as c. The lip password model is then formed by c and λ. Then, the distance between the feature sequence X and lip password model is calculated by the Euclidean distance between the center and log-likelihood of X:

$$d(X|\lambda, c) = ||c - \log P(X|\lambda)||^2. \tag{3.15}$$

The feature sequence X will be accepted or rejected according to a threshold on its distance to lip password model $d(X|\lambda, c)$.

Next, considering a subsequence matching with n subsequences partitioned from one sequence.

After getting the sets of feature subsequences from all the lip password training samples, the ith subsequences among all the training samples are used to train ith HMM with parameter λ_i. The set of those parameters is denoted as $M = \{\lambda_1, \cdots, \lambda_n\}$.

After obtaining M from training, the log-likelihoods of all the subsueqences with corresponding HMM are denoted as $D = \{d_1, \cdots, d_n\}$, where $d_i = \log P(s_i|\lambda_i)$. The matching results of subsequence become a vector in n dimension space.

Similar to baseline algorithm, we calculate the matching vectors among all the training samples. The center of them is denoted as $C = \{c_1, \cdots, c_n\}$. The lip password model is formed by C and M.

Then, the Euclidean distance between the feature sequence X and lip password model is calculated by:

$$d(X|C, M) = ||C - D||^2. \tag{3.16}$$

The feature sequence X will be accepted or rejected according to a threshold on its distance to lip password model $d(X|C, M)$.

Table 3.3: List of the passwords contents recorded by each speaker in the database.

Language	Password content	Note
English	Four-Zero-Nine-Two	E1
	Five-Six-Eight-Three	E2
Chinese	Zhi-Ma-Kai-Men	CO
	Kai-Men-Jian-Shan	CI1
	You-Qi-Wu-Li	CI2
	Gong-Xi-Fa-Cai	CI3
	Dong-Fang-Ming-Zhu	CI4
	Wu-Jing-Da-Cai	CI5

In our method, when $\alpha = 50\%$ and $L = 1/k$ during the overlapping sampling, the number of subsequences are $2 * k + 1$. A set of feature subsequences $S = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_{2k+1}\}$ can be obtained by overlapping sampling from the lip video for subsequences matching. Noted that if we set $\alpha = 0$, the number of subsequences is k, and we can also use the proposed subsequences matching approach on the non-overlapping subsequences.

3.4.3 Experiments

The Database

Similar to Section 3.3.4, we use a database consisting of two kinds of 4-digit English phrases and six kinds of 4-character Chinese phrases as lip passwords to evaluate the proposed algorithm. The password contents are listed in Table 3.3. Each utterance contains about 90 lip images of size 131×131 lasting for about 3 seconds, which is controlled by the users.

Experimental Setting

To evaluate the performance of lip password-based speaker verification systems, we utilized the videos recording all 8 kinds of passwords as listed in Table 3.3. In each section, the password spoken by the target users are considered as the authenticated user samples, while the rest of them are considered as the impostor samples. Each section is randomly divided into two parts: 40% for training and 60% for testing.

In the lip password-based speaker verification context, four types of scenarios (summarized in Table 3.1) can be considered according to whether the spoken utterance is the correct password or not and whether it is spoken by target user or not. With the same collections of target speaker's correct password (denoted as Target-Correct), three impostor types are evaluated separately to show the performance of proposed algorithm:

1. Target-Wrong: The target speaker saying the incorrect password;

2. Impostor-Correct: The impostor saying the correct password;

3. Impostor-Wrong: The impostor saying the incorrect password.

The False Accept Rate (FAR) and the True Accept Rate (TAR) are calculated as follows:

$$\text{FAR} = \frac{N_{FA}}{N_{IM}} \times 100\% \tag{3.17}$$

$$\text{TAR} = \frac{N_{TA}}{N_{CL}} \times 100\% \tag{3.18}$$

where N_{CL} and N_{IM} are the total number of testing client and impostor samples, N_{FA} is the number of testing impostor samples being falsely accepted and N_{TA} is the number of testing client samples being correctly accepted. As an application with high security, a FAR above 1.0% is acceptable [17]. Equal Error Rate (EER) may not always be a suitable metric since it is independent of the specific FAR requirement of the application [17]. In our experiments, TAR achieved at 1% FAR is chosen as the accuracy metric.

Experimental Results

In all experiments, the size of each cell is 16×16 pixels and the number of cells in each block is 2×2 to extract HOG descriptor. Dimensions of features (denoted as dim) reduced by PCA are set as 10, 20, 50, 100 respectively to evaluate the performance of the system under different kinds of features.

To assess the effectiveness of the proposed method, we compare the baseline with non-overlapping subsequence matching (denoted as NSM) and overlapping subsequence matching (denoted as OSM) as follows:

- **Baseline**: We first build the system with a single HMM with 16 hidden states and 3 continuous density Gaussian mixtures with spherical covariance matrix output as the baseline. That is a special case of our method when $L = 1$.

- **NSM**: We then add the subsequence matching method to the system. L is set as 1/2, 1/4, 1/8, and 1/16 with corresponding numbers of hidden states 8, 4, 2, and 1 of HMM with 3 continuous density Gaussian mixtures with spherical covariance matrix output. That is a special case of our method when the overlapping rate α is set as 0.

- **OSM**: We combine the overlapping sampling and subsequence matching method together with 50% overlapping rate. All the other setting is the same as NSM to evaluate the impact of overlap.

Target-Wrong Table 3.4 shows detailed results of verification experiments under the scenario where the target speaker saying the incorrect password.

First, let us examine the behavior of overlapping subsequences matching. For all the three kinds of features, most of the results increase dramatically as we go from $L = 1$ to $L = 1/4$ and then decrease from $L = 1/8$. When the subsequence length become too small, the sequence is too finely subdivided resulting to the drop in performance. Though non-overlapping subsequences verification accounts for most of the improvement, using overlapping subsequences even further improve

Table 3.4: TAR at 1% FAR under the Target-Wrong scenario. For each kind of features, the highest results are shown in bold and the lowest results underline. The best baseline performance is 71.03% when dim = 20 and the best performance among all the result is 83.82% achieved by OSM with $L = 1/4$ when dim = 20.

	Dim = 10		Dim = 20		Dim = 50		Dim = 100	
L	NSM	OSM	NSM	OSM	NSM	OSM	NSM	OSM
1	<u>68.77%</u>		<u>71.03%</u>		<u>56.59%</u>		<u>63.59%</u>	
1/2	77.01%	79.12%	77.98%	81.18%	75.70%	78.08%	76.02%	76.77%
1/4	80.84%	**83.04%**	81.66%	**83.82%**	79.05%	**80.94%**	79.19%	**80.69%**
1/8	79.17%	81.23%	80.35%	81.61%	78.27%	80.14%	79.85%	78.39%
1/16	75.00%	76.09%	75.90%	73.16%	74.13%	72.75%	71.27%	68.31%

the performance for the most part. When $L = 1/4$, the subsequences in OSM consists of the 4 units (i.e. 4 English digits and 4 Chinese characters) of passwords and 3 transformations between the 4 units, totally 7 subsequences. Matching over these 7 overlapping subsequences outperform NSM with only 4 units of passwords (L = 1/4) and larger numbers of non-overlapping subsequences (L = 1/8 and $L = 1/16$), which confirm the benefit of overlaps.

Next, let us examine the performance of features in different dimensions. The best baseline performance is 71.03% when the features are reduced to 20-dimension by PCA. The best performance of our methods is 83.82%, which is also obtained under the same features. The results imply that the baseline performance is heavily depended on a proper choice of dimension, whereas the performance of our method is much more stable. This is the main advantage of our overlapping subsequence matching method. Because it contains more distinctive information, we can use weak features (e.g. dim = 10) to accelerate the speed of algorithm with little drop in performance.

Impostor-Correct Table 3.5 shows detailed results of verification experiments under the scenario where the impostor saying the correct password.

Table 3.5: TAR at 1% FAR under Impostor-Correct scenario. For each kind of features, the highest results are shown in bold. The results of OSM with $L = 1/4$, which achieves the best results in Target-Wrong (see Table 3.4), are shown in underline. The best performance among all kinds of features is 99.42% achieved by the baseline with dim = 20.

	Dim = 10		Dim = 20		Dim = 50		Dim = 100	
L	NSM	OSM	NSM	OSM	NSM	OSM	NSM	OSM
1	**98.93%**		**99.42%**		**99.42%**		**98.76%**	
1/2	98.59%	98.74%	99.25%	98.98%	98.72%	98.64%	97.96%	97.63%
1/4	97.87%	<u>97.55%</u>	98.04%	<u>97.94%</u>	96.85%	<u>95.95%</u>	95.28%	<u>94.55%</u>
1/8	95.57%	95.47%	96.05%	95.74%	94.65%	92.73%	92.56%	89.95%
1/16	92.98%	92.13%	94.69%	97.41%	90.26%	86.94%	84.89%	84.45%

In this scenario, the best performance is obtained by the baseline with dim = 20. The subsequence matching, no matter whether with overlap or not, decrease the performance of baseline. The reason might be that comparing with one matching score obtained by the whole sequences, the matching scores of subsequences change the distribution characteristic of different people in a higher dimension. Nevertheless, our method with $L = 1/4$ when dim = 20, which achieve the best performance in Target-Wrong (see Table 3.4), only decrease about 1.5% of the best baseline performance from 99.42% to 97.94%, which is still acceptable in practice.

Impostor-Wrong Table 3.6 shows detailed results of verification experiments under the scenario where the impostor saying the incorrect password.

In this scenario, the best performance is obtained by the baseline with dim = 20. For the same reason of Impostor-Correct scenario, the subsequence matching also decreases the performance of baseline here. In spite of that, our method with $L = 1/4$ when dim = 20, which achieve the best performance in Target-Wrong (see Table 3.4), only decrease about 1.1% of the best baseline performance from 99.49% to 98.35%, which is also acceptable in practice.

Table 3.6: TAR at 1% FAR under the Impostor-Wrong scenario. For each kind of features, the highest results are shown in bold. The results of OSM with $L = 1/4$, which achieves the best results in Target-Wrong (see Table 3.4), are shown in underline. The best performance among all kinds of features is 99.49% achieved by the baseline with dim = 20.

	Dim = 10		Dim = 20		Dim = 50		Dim = 100	
L	NSM	OSM	NSM	OSM	NSM	OSM	NSM	OSM
1	**99.15%**		**99.49%**		**99.39%**		**98.86%**	
1/2	99.05%	99.01%	99.30%	99.25%	98.79%	98.74%	98.11%	97.60%
1/4	98.57%	<u>98.57%</u>	98.52%	<u>98.35%</u>	96.83%	<u>96.15%</u>	95.42%	<u>94.60%</u>
1/8	97.62%	96.80%	97.02%	96.44%	94.74%	92.93%	92.73%	91.45%
1/16	96.29%	95.76%	95.66%	97.75%	90.65%	87.72%	94.28%	95.47%

To summarize, our method has improved the performance of the Target-Wrong scenario significantly without much impact on the performance of Impostor-Correct and Impostor-Wrong scenarios.

3.5 Summary

In this chapter, we first present an approach to lip password-based speaker verification without knowing the password alphabet beforehand, thus enhancing the security of a lip password-based system for speaker verification. In our method, a concept of "interest intervals" has been presented to describe the lip movement. Subsequently, the IIM has been designed for the lip password to verify the speaker. Experiments have shown the efficacy of the proposed approach in comparison with the GMM and HMM, which are widely utilized in visual speaker verification systems.

Further, we design a novel overlapping subsequence matching approach to encode the information in lip passwords in the system. This technique works by sampling the feature sequences extracted from lip videos into overlapping subsequences and matching them individually. In each subsequence, we use dense Histogram of Ori-

ented Gradient (HOG) descriptor to encode the information of lip images, following with Principal Component Analysis (PCA) to reduce the dimension of HOG descriptor. Subsequently, the feature sequence is used to train the HMMs and produce the loglikelihood. All the loglikelihood of each subsequence form the final feature of the sequence and are verified by the Euclidean distance to positive sample centers. Our method does not require accurate alignment of feature sequences or detection on mouth status which is computationally expensive, and the overlap between consecutive subsequences shows significantly improved performance in the scenario where the target user saying the wrong password without much impacts on the performance in the other two impostor scenarios. We achieve 97.94% TAR at 1% FAR under the scenario where the impostor saying the correct passwords and 83.82% TAR at 1% FAR when the target user saying the wrong passwords, which shows the promise towards this novel speaker verification approach.

Chapter 4

Diagonal-like Pooling and Pyramid Structure of Lips Based on Sparse Coding

4.1 Introduction

The objective of this chapter is to propose a novel approach to lip password-based speaker verification with arbitrary lip password, i.e. the password content to be a phrase of any languages, based on sparse coding. Since the language alphabet of the lip password is unknown, we design a representation of lip sequences which preserve the temporal order by employ a diagonal-like mask in pooling stage and build a pyramid structure to form the spatiotemporal lip representation containing the structural characteristic under lip passwords.

Currently, most speaker verification methods only focus on verifying the identity of the speaker, ignoring the temporal information of password content. The proposed lip representation can keep the temporal order and improve the accuracy of verifying the password contents significantly. To further improve the accuracy of verifying the lip passwords, a pyramid structure of lips is proposed to contain more information in the spatial domain.

In contrast to the previous related methods of detecting the mouth status and segmenting the whole lip sequences into several independent subsequences with the risk of cutting across potentially discriminating features [26, 45], the proposed method directly utilizes the temporal order structure of sparse representation and does not rely on the detection of mouth status, which is computationally heavy and sensitive to environments such as illumination.

In order to evaluate the benefits of the proposed lip representation in the lip password-based speaker verification system, we collect a database which contains both English and Chinese lip passwords. We empirically investigate the ability of state-of-the-art spatiotemporal lip features to verify the lip contents and speakers identity. Experimental results show that the proposed lip feature outperforms the state-of-the-art ones in all scenarios we have tried so far, especially when verifying the lip password contents.

The remaining part of this chapter is organized as follows: Section 4.2 gives an overview of the related work. Then, the proposed approach is described in detail in Section 4.3. Experimental results and analysis are provided in Section 4.4. Finally, we draw a summary in Section 4.5.

4.2 Overview of Sparse Coding Learning Framework

Sparse coding is a generative model that describes the input signal as the linear combination of signals in a pre-learned dictionary [21]. When it is used in the time sequence classification, one way is to calculate the reconstruction coefficients on the dictionary sequences [36]. As described in [52], the input sequence will most receive strong coding responses from those closely related dictionaries, while the reconstruction coefficients on the unrelated dictionary will be smaller. Thus the sparse coding can be used to perform time sequence classification by using the reconstruction error.

As describe in [36], we denote the set of dictionary sequences from the jth class by $X^{(j)} = [X_{j_1}, X_{j_2}, \dots]$. X_{j_i} denotes the ith sequence in $X^{(j)}$ that belongs to the jth class. We assume there are C classes, $j \in \{1, \dots, C\}$. We denote $\alpha^{(j)}$ as the corresponding reconstruction coefficients for $X^{(j)}$. Following [52], the predicted class label is the one with the lowest total reconstruction error:

$$j_{opt} = \arg\min_j \|Y - X^{(j)}\alpha^{(j)}\|_F^2 \tag{4.1}$$

In the literature, some works have utilized the spatiotemporal feature directly too. For instance, Chan *et al.* [8] proposed the Local Ordinal Contrast Pattern with Three Orthogonal Planes (LOCP-TOP) for lip-based speaker verification. The LOCP is a texture descriptor that encodes the appearance of lip images. Meanwhile, by using the LOCP in TOP, it makes the final lip representation keep the dynamic information of lip movement. In [21], Lai *et al.* used sparse coding under a hierarchical spatiotemporal structure to form the lip representation. The dictionary is learned from all users and max-pooling in the hierarchical structure is performed in the sparse representation of lip subsequences. These lip representation densely extract lip features over time and space domains to avoid the segmentation and modeling of lip sequences, leading to a better generative performance comparing with the model-based approaches [21]. Nevertheless, the world model used in visual speaker authentication [21] is not realistic for a speaker verification system that has new users enrolling. Also, the training of dictionary is also time-consuming. In contrast, our method uses fixed dictionary which is chosen from the training data and thus does not require dictionary learning on the data of whole users.

4.3 The Proposed Method

For a lip password-based speaker verification, it is of great concern to extract lip features which are discriminative against different password contents and speaker's identity. In this chapter, we propose a novel lip representation using the diagonal-like pooling based on sparse coding and a 3-layer pyramid structure designed for the

lips. In the following, we first illustrate the details of the sparse learning scheme and the diagonal-like pooling method in our approach in Section 4.3.1. Then in Section 4.3.2 we introduce the 3-layer pyramid structure of lips.

4.3.1 Diagonal-like Pooling

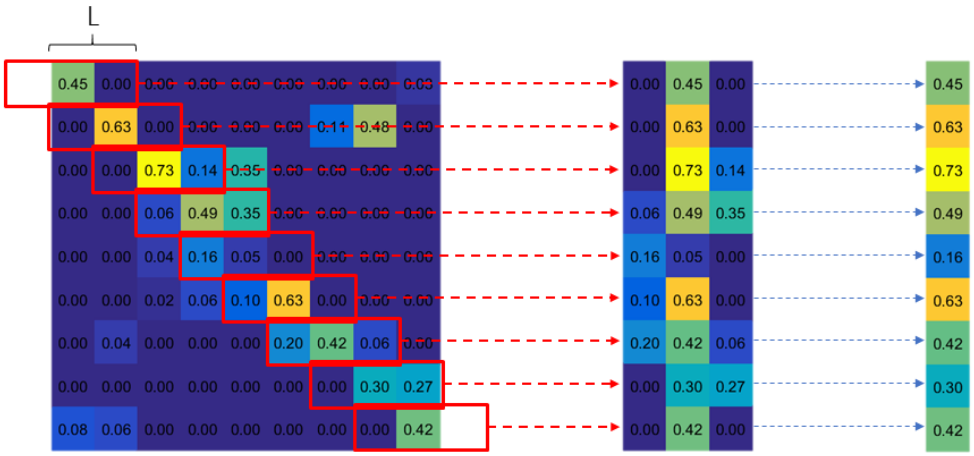
Sparse coding is a generative model that describes the input signal as the linear combination of signals in a pre-learned dictionary [21]. Previous work in constructing lip representation by sparse coding segments the lip sequences into very small cells and trains an elementary dictionary of all users [21]. This approach eliminates the location of sparse response, which is critical for lip password, and needs large computation when training the dictionary. What's more, the dictionary trained by the existing users are not representative enough for the new users. In contrast, we choose the original training data as the password-specified dictionary and thus don't require dictionary learning on the data of whole users.

Given the correct lip password training data, the password-specified dictionary is formed as $D = \{\boldsymbol{d}_1, \cdots, \boldsymbol{d}_T\}$, where $\boldsymbol{d}_i, i \in [1, T]$ is the vectorized ith frame in D. For a lip video with T frames, the input data is denoted as $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$, where $\boldsymbol{x}_i, i \in [1, T]$ is the vectorized ith frame in X. With the password-specified dictionary D, the sparse code of $\boldsymbol{x}_i$ can be obtained by Lasso algorithm [14]. We can get a matrix of coefficients $A = [\alpha_1, \ldots, \alpha_T]$ such that for every $\boldsymbol{x}_i$, the corresponding column α_i is the solution of

$$\min_{\alpha_i} \|x_i - D\alpha_i\|_2^2, \text{ s.t.} \|\alpha_i\|_1 \leq \lambda, \tag{4.2}$$

where λ is the sparse regularization parameter.

The elements in this sparse matrix A can reflect the relation between input data X and dictionary D. The sparse values in α_i represent the weight of linear combination in D to construct $\boldsymbol{x}_i$. The larger α_{ij} means the more similar $\boldsymbol{d}_i$ and $\boldsymbol{x}_j$ is. If the input data X is the correct password, the sparse values in its sparse matrix A should be near the diagonal the matrix. A special case is when $X = D$, then $A = I$ is the solution of Equation (4.2), where I is the identity matrix. On



(a) (b) (c)

Figure 4.1: The key idea of diagonal-like pooling. (a) $T \times T$ sparse representation of a video sequence with the matched lip password sequence as the dictionary, (b) diagonal structure extraction using sliding window, and (c) max pooling over time.

the other hand, if the input data X is the wrong password spoken by the target user, that means X and D may also have many large values in A but might not all near the diagonal. If we simply do max-pooling [49, 44] over time, the location information will all be lost.

To address this problem, we introduce the diagonal-like pooling method as shown in Fig. 4.1, where we can observe that if the input data is matched with the dictionary, the sparse representation gets most responses near the diagonal of the matrix. The key idea is, by using the sliding windows through the diagonal elements, the value near the diagonal, which represents the relation between input data and dictionary in very close time t, is collected and other mismatched elements are removed. Then, doing max pooling over time can generate the features which keep the order information.

For more convenient calculation, let L denote the length of windows, we define

a binary $T \times T$ matrix W, where the element is calculated as:

$$w_{ij} = \begin{cases} 1, & \text{if } |i-j| \leq L, \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

For example, if $T = 5$ and $L = 2$, the structure of W is:

$$\mathbf{W} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{4.4}$$

This matrix can be regarded as the "mask" that marks the diagonal-like structure as "1" and others as "0". Then, the final feature $\boldsymbol{f} = \{f_1, ..., f_T\}$ contained by diagonal-like pooling process can be written as:

$$f_i = \max_{j=1}^{T} (\alpha_{ij} w_{ij}). \tag{4.5}$$

Accordingly, the dictionary is selected from the training data that is most suitable for representing the lip password. That is to select the sample with the smallest reconstruction error when used as the dictionary.

4.3.2 The Pyramid Structure of Lips

The diagonal-like pooling method introduced in Section 4.3.1 preserves the temporal order of features. To further enhance the representativeness of proposed lip feature, we build a pyramid spatial structures according to the characteristic of lip movement.

As shown in Fig. 4.2 (a), the origin lip images are departed into three block group. The first layer is the whole mouth area. The second layer contains the upper lip, lower lip and the middle of the mouth containing teeth and tongue. It is observed that the lip movement has more vertical symmetry than horizontal symmetry. What is more, the shapes of the upper lip and lower lip contain the information of the speaker's identity. The third layer is the center and four corners of the lip images. The center of the lip images, including teeth and tongue, can also

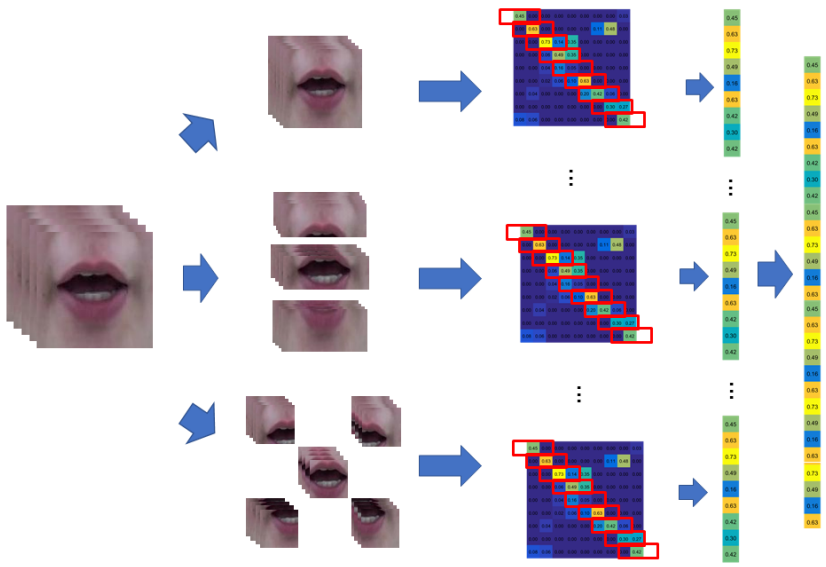


Figure 4.2: The framework of the proposed approach. (a) Origin lip images are departed into (b) three groups of blocks; (c) learning $T \times T$ sparse representations of each block; (d) diagonal structures extraction using sliding window and max pooling over time; (e) the final lip feature.

provide some information about the password contents. These areas have overlap with each other.

All these blocks are regarded as the input sequential data and get the features by sparse coding and the proposed diagonal-like pooling shown in Section 4.3.1. Finally, the lip representation feature is a combination of those features of the blocks, as shown in Fig. 4.2. Denote $\boldsymbol{D}_k^l$ and $\boldsymbol{f}_k^l$ as the dictionary and corresponding feature of the kth block in the l layer using Equation (4.5), the final feature can be written as:

$$\boldsymbol{F} = \{\boldsymbol{f}_1^1, \boldsymbol{f}_1^2, \ldots, \boldsymbol{f}_3^2, \boldsymbol{f}_1^3, \ldots, \boldsymbol{f}_5^3\}. \tag{4.6}$$

The proposed approach is summarized in Algorithm 3.

Algorithm 3 The proposed approach to extract lip representation

Input: The dictionary $\{\boldsymbol{D}_1^1, \boldsymbol{D}_1^2, \ldots, \boldsymbol{D}_3^2, \boldsymbol{D}_1^3, \ldots, \boldsymbol{D}_5^3\}$ of all 9 block in 3 layers, the mask W generated by Equation (4.3) and the input data X.

Output: Lip representation feature $\boldsymbol{F}$.

1: **for** $l = 1$ to 3 **do**
2: **for** $k = 1$ to n_l **do**
3: Use Equation (4.2) to calculated A_k^l.
4: Use Equation (4.5) to calculated f_k^l.
5: **end for**
6: **end for**
7: Final feature F is calculated from Equation (4.5)

4.4 Experiments

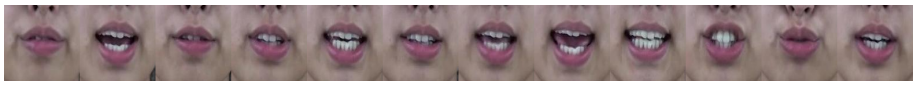
4.4.1 Database

Most existing databases such as XM2VTSDB [29] and MVGL-AVD [7], despite their popularity in serving as benchmarks for traditional visual speaker verification tasks, are incompetent for our lip password-based system, because these databases contain the limited types of different password contents and languages. Under the circumstances, we constructed a database consisting of 8 kinds of different lip passwords, including 2 kinds of English digits and 6 kinds of Chinese phrases, to evaluate the proposed algorithm. Some samples from the database are shown in Fig. 4.3. Each utterance contains about 90 lip images of size 131×131 lasting for about 3 seconds. In our experiment, each lip sequence is sampled to 50 frames with size 50×50 in order to reduce the computational complexity. The size of the second layer in pyramid structure is set as 20×50 and the size of the third layer is 30×30.

4.4.2 Experiment Protocol

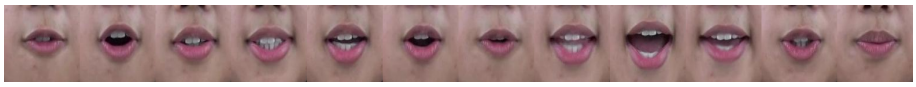
Different from the speaker verification system that only detects and rejects the impostor with a fixed phrase, the lip password-based speaker verification system

(a) “Zhi Ma Kai Men” of user A



(b) “5683” of user A

(c) “4092” of user A



(d) “4092” of user B

Figure 4.3: Images of each row are sampled from the lip password sequences. Subfigure (a∼c) are different lip passwords spoken by the same user. If (c) is set as the correct lip password, then (a) and (b) are the case that the wrong password spoken by the target user, and (d) is the case that the impostor speaking the correct password. For a lip password-based speaker verification system, (a), (b) and (d) are all regarded as the impostor data and rejected.

faces the challenge of three kinds of impostor scenarios, which are summarized as follows:

- **Target-Wrong**: The target speaker saying the incorrect password;

- **Impostor-Correct**: The impostor saying the correct password;

- **Impostor-Wrong**: The impostor saying the incorrect password.

Three impostor types above and the holistic impostor data (denoted as **All-Impostor**) are evaluated to show the performance of proposed algorithm.

Similar to the protocol in [29], we employ a protocol specific to lip password-based speaker verification. In each section, the password spoken by the target users are considered as the authenticated user samples, while the rest of them are considered

as the impostor samples. The final accuracy is the average of the results by running 10 times.

Linear SVM [2] is adopted as the classifier. The half total error rate (HTER) is adopted to evaluate the performance of verification algorithms. HTER is obtained by setting the threshold of SVM to obtain Equal Error Rate (EER) in the evaluation set, and then calculated by the False Accepted Rate (FAR) and False Rejected Rate (FRR) in the testing set according to HTER=(FAR+FRR)/2 [21]. As the EER does not reflect the practical system performance when the testing data is unseen [29], the HTER is a more reasonable measurement of the performance for speaker verification system.

FAR and FRR are calculated as follows:

$$\text{FAR} = \frac{N_{FA}}{N_{IM}} \times 100\% \tag{4.7}$$

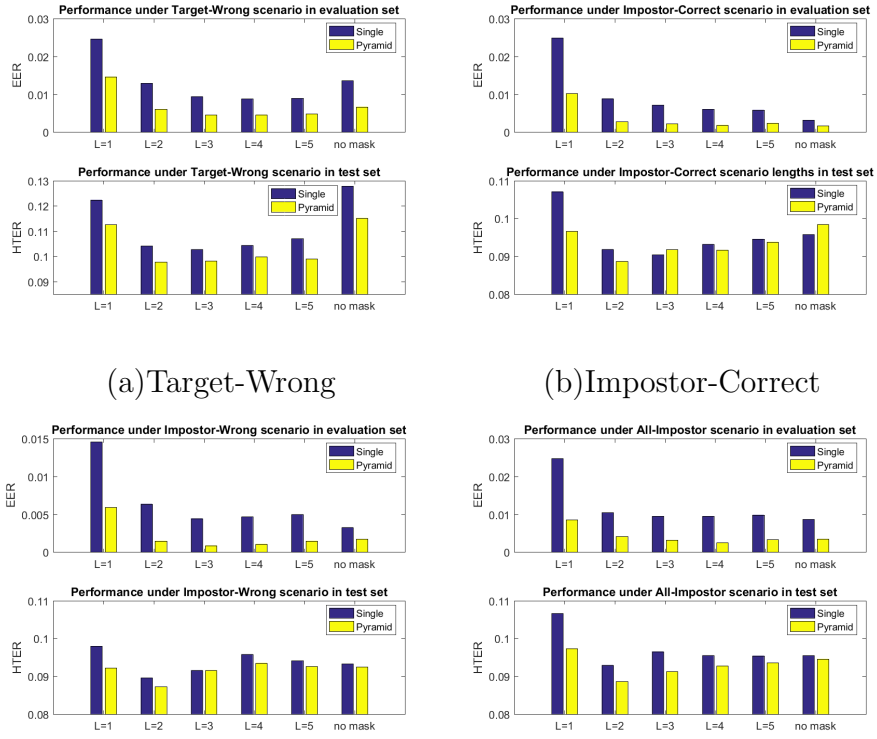
$$\text{FRR} = \frac{N_{FR}}{N_{TU}} \times 100\% \tag{4.8}$$

where N_{TU} and N_{IM} are the total number of target user and impostor samples, N_{FA} is the number of impostor samples being falsely accepted and N_{FR} is the number of target user's correct password samples being false rejected. The EER is the value of FAR or FRR when they are equal (i.e. FAR=FRR).

4.4.3 Experiment Results

To evaluate the influence of the choice of window size L and the utility of proposed pyramid structure, we extract the lip features under the different window sizes with both 3-layer structure and the single layer. The evaluation and testing results are shown in Fig. 4.4.

From Fig. 4.4, we can observe that, when window length is set at 1, both the EER in the evaluation set and HTER in the testing set are very large comparing to the other lengths. That is reasonable because even the same password spoken by the same person, the speed is hard to be the same. When the length of the windows is too small, it faces the problem of under-fitting. When $L = 2$, the error rate



(a)Target-Wrong (b)Impostor-Correct

(c)Impostor-Wrong (d)All-Impostor

Figure 4.4: Eval-set EER and Test-set HTER variation against increasing window length L under four impostor scenario.

decreases rapidly and then slowly increases when increasing the L. That indicates our methods are not very sensitive to the choice of window size as long as it is a reasonable value. Comparing with the feature just using max pooling (denoted as "no mask" in Fig. 4.4), our method also improves the performance, especially in the Target-Wrong scenario, which is very important for the lip password-based speaker verification system.

As observed in Fig. 4.4, the lip feature under the 3-layer pyramid structure outperforms the one under a single layer in almost all cases, especially in the Target-Wrong scenario. The results indicate that the proposed pyramid structure can preserve more discriminative features underlying the second and the third layer of lip sequences.

To sum up, the lip features under single layer and pyramid layers all obtain the

Table 4.1: Feature performance comparison showing EER(Eval) and HTER(Test). For each scenario, the best result is shown in bold and the second one is in underline.

	Tar.-Wrong		**Imp.-Correct**		**Imp.-Wrong**		**All-Imp.**	
	Eval	Test	Eval	Test	Eval	Test	Eval	Test
LOCP-TOP	3.02%	17.39%	0.38%	10.10%	0.19%	10.10%	1.12%	10.68%
SC-HIER	**0.28%**	10.92%	**0.01%**	9.50%	**0.00%**	9.50%	**0.02%**	9.92%
our-single	1.30%	<u>10.41%</u>	0.89%	<u>9.18%</u>	0.64%	<u>9.18%</u>	1.05%	<u>9.29%</u>
our-pyramid	<u>0.61%</u>	**9.77%**	<u>0.27%</u>	**8.87%**	<u>0.15%</u>	**8.87%**	<u>0.41%</u>	**8.86%**

best performance when $L = 2$ and better than the feature just using max pooling. The diagonal-like pooling method significantly improves the performance, especially in the Target-Wrong scenario, which is of importance for the lip password-based speaker verification. What is more, the features formed from the 3-layer pyramid structure is more discriminative than one using a single layer only. The results of All-Impostor show the efficacy of the proposed approach.

4.4.4 Performance Comparison with the State-of-the-art

To assess the effectiveness of the proposed method, two spatiotemporal features widely used in the verification system: the LOCP-TOP feature in [8] (LOCP-TOP in short) and the hierarchical pooling sparse lip representation in [21] (SC-HIER in short) are adopted for comparison. In [8], the LOCP is extracted in TOP to form the final lip representation. In [21], the lip representation is generated by using the hierarchical max pooling on the sparse representation, using a dictionary learned from all the users' data.

For our approach, two kinds of features with windows length sets to $L = 2$ under a single layer (our-single in short) and 3-layer pyramid (our-pyramid in short) structure, are investigated. Comparison results are shown in Table 4.1.

From the table, it can be seen that the features based on sparse coding (our methods and SC-HIER) outperform the LOCP-TOP features. Further, although

HIER has a very low EER (even 0.00% in Impostor-Wrong), the HTER in the testing datasets is even higher than our-single. This is because the lip representation generated by HIER is very sparse and has very high dimension, which makes it easy to be separated by linear SVM and get very low verification error in the evaluation set, leading to over-fitting and get higher HTER in the testing set.

With respect to the performance in different scenarios, Target-Wrong, which is of importance for lip password-based speaker verification, looks more challenging. As shown in Table 4.1, the performance of LOCP-TOP in the Target-Wrong scenario is seriously worse than in the other two scenarios. In all four scenarios, our-pyramid gets the best HTER in the testing set and our-single gets the second best HTER, which outperform the other two kinds of lip representation. The results indicate the discriminative power of the proposed lip representation.

4.5 Conclusion

This chapter has proposed a novel speaker verification approach with unknown language alphabet. The proposed method works by generating lip feature of input data using diagonal-like max pooling on the sparse representation to preserve the temporal order of lip sequences. We also build a pyramid structure to form the spatiotemporal lip representation to catch the structural characteristic under lip password. It need not require the accurate alignment of feature sequences or detection on mouth status, whose computation is laborious. Experiments on different kinds of lip passwords have shown its promising result comparing with the state-of-the-art ones.

Chapter 5

The lip-password based visual speaker verification prototype

5.1 Introduction

In this chapter, we develop a lip password-based visual speaker verification system, which is based on the algorithms of this thesis and designed for verifying the identity of the speaker based on their lip password without the limitation of language. We propose a ubiquitous framework for lip password-based speaker verification system, which contains Graphical User Interface (GUI), the user database and the pipeline that contains feature extraction, model training, and model matching. Besides, we design a GUI that contains registration and verification functions for users to testify the verification system. We implement a pipeline to record the lip password, extract features and train the users' models of their lip passwords. These models are stored with their usernames in the user database. When a user is trying to sign in the system, he/she should provide the username and say a lip password that can be matched with the corresponding model. Only when the target user saying the correct password, he/she could be accepted by the system.

The remainder of this chapter is organized as follows. Section 5.2 presents the system framework & components. Section 5.3 presents the system implementation.

We summarize this chapter in Section 5.4.

5.2 System Framework & Components

Existing visual speaker verification algorithms, including the lip password-based visual speaker verification system proposed in [26], have shown the effectiveness in the experiments. However, they are usually testified by the databases that are recorded in a stable environment not invariant to the changes resulting from illumination variations or view angle alteration of the camera, which might not reflect the performance of these algorithms when applying in the real-world applications. As a result, we propose a ubiquitous framework for lip password-based speaker verification system, which contains Graphical User Interface (GUI), the user database and the pipeline that contains feature extraction, model training, and model matching. The system can train the user data, store the user's model in the database, and verify the user's identity.

In this section, we give an introduction to the system framework and show the structures of the interface and the user database.

5.2.1 System Framework

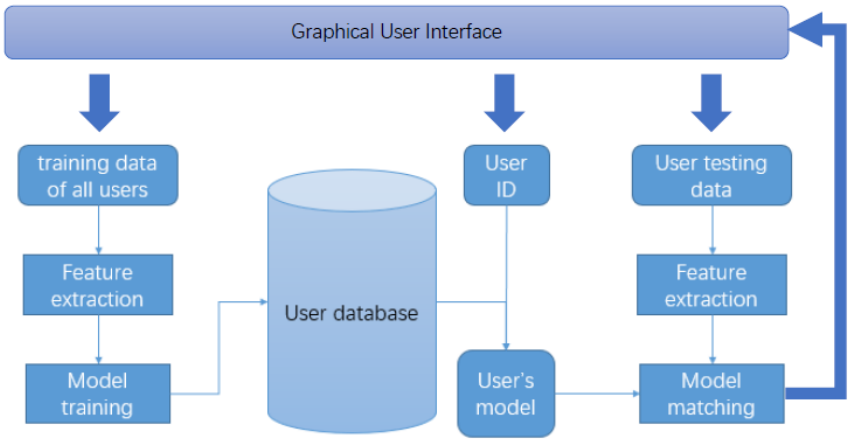


Figure 5.1: The framework of the prototype.

As shown in Figure 5.1, we propose a ubiquitous framework for lip password-based speaker verification system, which contains the Graphical User Interface (GUI), the user database and the pipeline that contains feature extraction, model training and matching. The data is recorded and collected by the GUI and transferred to the internal program as the input. After the processing of the internal program, the result is transferred back to and shown in the GUI. Only when the target user saying the correct password, he/she could be accepted by the system.

5.2.2 Graphical User Interface

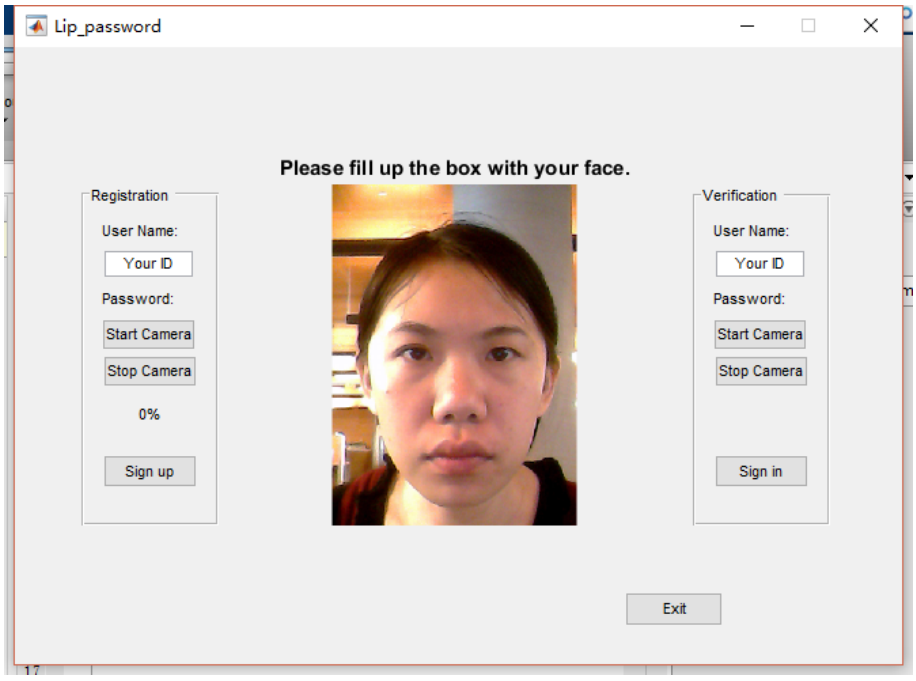
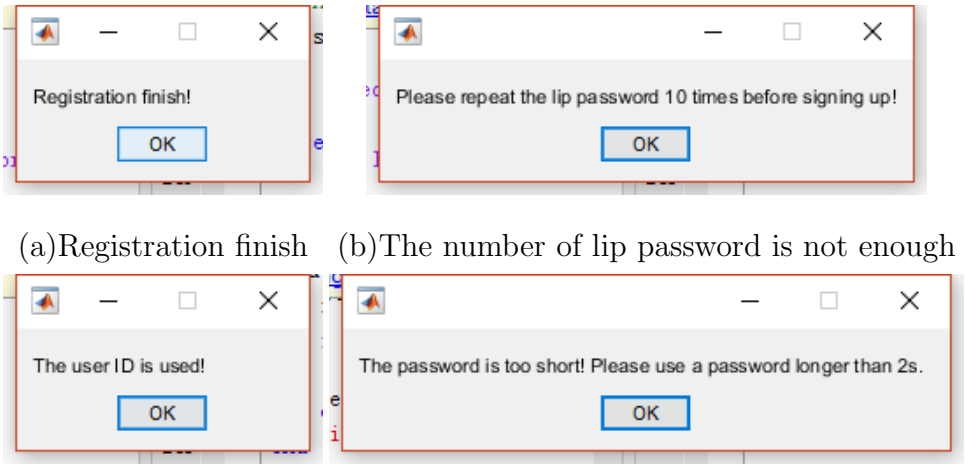


Figure 5.2: The GUI of the prototype.

To make the system easy to interact with the user, we design a GUI that contains registration and verification functions for users to testify the verification system. As shown in Figure 5.2, there are mainly three parts in the GUI: registration, verification and camera display.



(a)Registration finish (b)The number of lip password is not enough

(c)The user ID is used (d)The password is too short

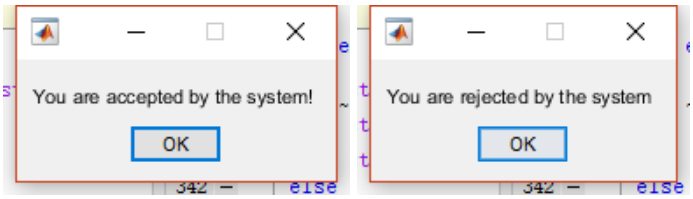
Figure 5.3: (a) Successful and (b)∼(d) different unsuccessful situations when pushing the "Sign up" button.

Registration

The registration area consists of 3 components: (1) the user name text box; (2) the lip password recording area and (3) the "Sign up" button. The user record the data with the following steps:

1. Input the user ID.
2. Push the "Start Camera" button and begin to saying the lip password.
3. After the lip password finished, push the "Stop Camera" button to stop recording.
4. Repeat 2 and 3 until the processing reaches 100%.
5. Push the "Sign up" button.

As shown in Figure 5.3, after the user pushes the "Sign up" button, the system will show a message that states the situation of the registration. If the number of lip password is not enough/ the user ID is used/ the password is too short, the registration is unsuccessful. When the registration is successful, the message box will show "Registration finish!" as shown in Figure 5.3 (a).



(a)Accepted (b)Rejected

Figure 5.4: The massages that show the results of the verification.

Verification

Similar to the registration area, the verification area consists of 3 components: (1) the user name text box; (2) lip password record and (3) the "Sign in" button. The user does the verification with the following steps:

1. Input the user ID.

2. Push the "Start Camera" button and begin to say the lip password.

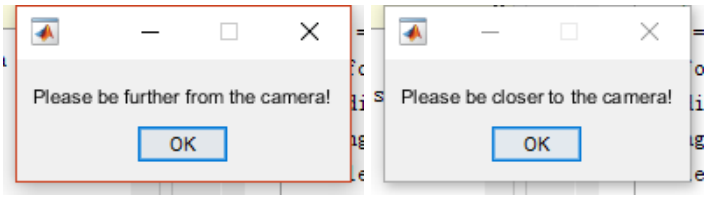
3. After the lip password finished, push the "Stop Camera" button to stop recording.

4. Push the "Sign in" button.

As shown in Figure 5.4, after the user push the "Sign in" button, the system will show a message that states the situation of the verification. Only when the lip password is spoken by the target user and the content is matched with the model, the message box will show "You are accepted by the system!" as shown in Figure 5.4 (a).

Camera Display

This area is used to display the images collected from the camera in real-time. The user is required to fill up the box with the face. If the user is too close to or far from the camera, the GUI will give a warning message, as shown in Figure 5.5.



(a)Too close (b)Too far

Figure 5.5: The warning massages when the user is (a) too close to or (b) too far from the camera.

Table 5.1: The structure of the user database

No.	User ID	Model		
		Dictionary	SVM Structure	Threshold
1	User1	D1	S1	T1
2	User2	D2	S2	T2
3	User3	D3	S3	T3
⋮	⋮	⋮	⋮	⋮

5.2.3 User Database

We propose a structure to store the user IDs and the corresponding models as shown in Table 5.1. The lip password model consists of 3 elements: (1) dictionary, (2) SVM structure and (3) threshold. The dictionary is stored for the sparse coding in Algorithm 3. The SVM structure is obtained in the training step and the threshold is obtained in the validation step during the registration.

5.3 System Implementation

We implement a pipeline to record the lip password, extract features and train the users' models of their lip passwords. These models are stored with their user names in the user database. When a user is trying to sign in the system, he/she should provide the user name and say a lip password that can be matched with the corresponding model. Only when the target user saying the correct password,

he/she could be accepted by the system.

In this section, we describe the algorithm of each step in the pipeline of the system.

5.3.1 Localization

The preprocessing to seek the location of lip region is the key issue in the process of lip password-based visual speaker verification system, particularly under the changing illumination condition [23]. Previous studies have shown that the accuracy of the lip localization is of importance for the recognition rate of a lip reading system and the accuracy of a visual speaker verification system [23]. In [23], they propose a new approach to obtain the mouth area based upon the transformed gray-level image, which solve the problem of lip localization under face illumination with shadow. In [9], Cheung *et al.* used a framework of Localized Color Active Contour Model (LCACM) for lip tracking.

In this system, we employ the algorithms proposed in [51], which utilize a pose-free facial landmark fitting method to localize the position of lip region. After localizing the lip region, the region is cropped from the whole image and the lip image will be normalized a fixed size.

5.3.2 Feature Extraction

As we explain before, this prototype is mainly based on the algorithms proposed in Chapter 4. In Chapter 4, we have presented the feature extraction in Algorithm 3. After the lip images are extracted from the face images, these lip images from one lip password image sequences are used as the input of Algorithm 3. The output is the final representation of the lip password.

In the registration stage, we extract the lip representation from each lip password database for model training. In the verification stage, the testing lip password is also transferred to the feature vector by Algorithm 3 to be verified by the SVM structure and the threshold which the user claims to be.

5.3.3 Verification Algorithms

Similar to the experiment protocol described in Section 4.4, linear SVM [2] is adopted as the classifier.

In the training stage, the input data is separated into the training set and the validation set. The lip representation extracted from the training lip password data by Algorithm 3 is used to train the SVM. The threshold of SVM is the threshold that obtains the Equal Error Rate (EER) in the evaluation set. The EER is obtained when the False Accepted Rate (FAR) is equal to the False Rejected Rate (FRR).

FAR and FRR are calculated as follows:

$$\text{FAR} = \frac{N_{FA}}{N_{IM}} \times 100\% \tag{5.1}$$

$$\text{FRR} = \frac{N_{FR}}{N_{TU}} \times 100\% \tag{5.2}$$

where N_{TU} and N_{IM} are the total number of target user and impostor samples, respectively, N_{FA} is the number of impostor samples being falsely accepted, and N_{FR} is the number of target user's correct password samples being false rejected.

In the verification stage, the lip representation extracted from the testing lip password data by Algorithm 3 is classified by the SVM of the target user. The verification result will be sent back to the GUI for displaying the outcome.

5.4 Summary

This chapter presents the framework and implementation of a lip password-based visual speaker verification system, which is based on the algorithms of this thesis and designed for verifying the identity of the speaker based on their lip password without the limitation of language. We implement a pipeline to record the lip password, extract features and train the users' models of their lip passwords. These models are stored with their usernames in the user database. When a user is trying to sign in the system, he/she should provide the username and say a lip password that can be matched with the corresponding model. Only when the target user saying the correct password, he/she could be accepted by the system.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis has addressed the issue in developing a lip password-based visual speaker verification system with unknown language alphabet.

Firstly, we have presented two approaches based on the Hidden Markov Model to lip password-based speaker verification without knowing the password alphabet beforehand, thus enhancing the security of a lip password-based system for speaker verification. In our first method, a concept of "interest intervals" has been presented to describe the lip movement. Subsequently, the IIM has been designed for the lip password to verify the speaker. Experiments have shown the efficacy of the proposed approach in comparison with the GMM and HMM, which are widely utilized in visual speaker verification systems. We have further designed a novel overlapping subsequence matching approach to encode the information in lip passwords in the system. This technique works by sampling the feature sequences extracted from lip videos into overlapping subsequences and matching them individually. In each subsequence, we use dense Histogram of Oriented Gradient (HOG) descriptor to encode the information of lip images, following with Principal Component Analysis (PCA) to reduce the dimension of HOG descriptor. Subsequently, the feature sequence is used to train the HMMs and produce the loglikelihood. All the loglikelihood of each subsequence form the final feature of the sequence and are verified by the Euclidean

distance to positive sample centers. Our approach does not require accurate alignment of feature sequences or detection on mouth status which is computationally expensive. Experiments have been performed on a database that contains totally 8 kinds of lip passwords including English digits and Chinese phrases. The overlap between consecutive subsequences has shown significantly improved performance on rejecting the target user saying wrong lip password, which is very important for lip password-based speaker verification system. Our proposed algorithms have effectively improved the baseline performance, which shows the effectiveness of our speaker verification system via arbitrary lip passwords.

Subsequently, this thesis has proposed a method and relevant algorithms to build a lip password-based visual speaker verification system with unknown language alphabet based on sparse coding. The proposed method works by generating lip features of input data using diagonal-like max pooling on the sparse representation to preserve the temporal order of lip sequences. We have also built a pyramid structure to form the spatiotemporal lip representation to catch the structural characteristic under lip password. It need not require the accurate alignment of feature sequences or detection on mouth status, whose computation is laborious. Experiments on different kinds of lip passwords have shown its promising result comparing with the state-of-the-art ones.

Finally, we have developed a prototype of the lip password-based visual speaker verification system with unknown language alphabet, which is based on the algorithms of this thesis and designed for verifying the identity of the speaker based on their lip password without the limitation of language.

6.2 Future Work

In our future work, some issues may be further explored along four-fold directions:

(1) *Improving the accuracy of the system*

In this thesis, the novel lip password-based visual speaker verification system has reached an accuracy of over 90%. To further utilize this system in practice, we still need to work on the improvement of the accuracy of this system.

(2) *Robustness of the system*

In this thesis, we use a database that is recorded in a conditional environment which the speaking pace, facial expressions, and illumination are fixed. As a result, the experiments could not evaluate the robustness under these conditions of our system. During the test of the prototype, we found that our system is somewhat sensitive to a great change of illumination, which will limit its utilization in reality. From a practical viewpoint, it is of importance to make our system more robust to the illumination change to a certain degree.

(3) *Fusion with other biometrics*

In this thesis, we mainly investigate the lip motion for the visual speaker verification system. To reach the higher security, the system could fuse with other multi-modal expert systems to improve the robustness and accuracy of speaker verification. Due to the characteristics of lip biometric features, our system can be fused with face recognition and iris biometrics, since these features are all extracted from the face of human users. Future research will focus on combining the lip-password sequence with other biometrics together for the robust visual speaker verification system with higher security.

Bibliography

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[2] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *The Workshop on Computational Learning Theory*, pages 144–152, 1996.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[5] C. Bregler, H. Hild, S. Manke, and A. Waibel. Improving connected letter recognition by lipreading. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 557–560 vol.1, 1993.

[6] C. C. Broun, X. Zhang, R. M. Mersereau, and M. Clements. Automatic speechreading with application to speaker verification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages I–685. IEEE, 2002.

[7] H. E. Cetingul, Y. Yemez, E. Erzin, and A. M. Tekalp. Discriminative analysis of lip motion features for speaker identification and speech-reading. *IEEE Transactions on Image Processing*, 15(10):2879–2891, 2006.

[8] C. H. Chan, B. Goswami, J. Kittler, and W. Christmas. Local ordinal contrast pattern histograms for spatiotemporal, lip-based speaker authentication. *IEEE Transactions on Information Forensics and Security*, 7(2):602–612, 2012.

[9] Y. M. Cheung, X. Liu, and X. You. A local region based approach to lip tracking. *Pattern Recognition*, 45:3336–3347, 2012.

[10] G. I. Chiou and J. N. Hwang. Lipreading from color video. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 6(8):1192, 1997.

[11] S. Cox, R. Harvey, Y. Lan, J. Newman, and B. Theobald. The challenge of multispeaker lip-reading. In *International Conference on Auditory-Visual Speech Processing*, pages 179–184, 2008.

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[13] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga. Evaluation of speaker verification security and detection of HMM-based synthetic speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8):2280–2290, 2012.

[14] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of statistics*, 32(2):407–499, 2004.

[15] M. I. Faraj and J. Bigun. Motion features from lip movement for person authentication. In *International Conference on Pattern Recognition*, pages 1059–1062, 2006.

[16] M. I. Faraj and J. Bigun. Synergy of lip-motion and acoustic features in biometric speech and speaker recognition. *IEEE Transactions on Computers*, 56(9):1169–1175, 2007.

[17] A. Jain, B. Klare, and A. Ross. Guidelines for best practices in biometrics research. In *IEEE International Conference on Biometrics*, pages 541–545, 2015.

[18] S. M. Karlsson and J. Bigun. Lip-motion events analysis and lip segmentation using optical flow. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 138–145, 2012.

[19] M. N. Kaynak, Q. Zhi, A. D. Cheok, K. Sengupta, Z. Jian, and K. C. Chung. Analysis of lip geometric features for audio-visual speech recognition. *IEEE Transactions on Systems Man & Cybernetics Part A Systems & Humans*, 34(4):564–570, 2004.

[20] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[21] J. Y. Lai, S. L. Wang, W. C. Liew, and X. J. Shi. Visual speaker identification and authentication by joint spatiotemporal sparse coding and hierarchical pooling. *Information Sciences*, 373:219–232, 2016.

[22] A. Larcher, K.-A. Lee, B. Ma, and H. Li. Rsr2015: Database for text-dependent speaker verification using multiple pass-phrases. In *INTERSPEECH*, 2012.

[23] M. Li and Y. M. Cheung. Automatic lip localization under face illumination with shadow consideration. *Signal Processing*, 89(12):2425–2434, 2009.

[24] S. Z. Li, D. Zhang, C. Ma, H. Y. Shum, and E. Chang. Learning to boost gmm based speaker verification. In *INTERSPEECH*, 2003.

[25] X. Liu and Y. M. Cheung. A multi-boosted hmm approach to lip password based speaker verification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2197–2200, 2012.

[26] X. Liu and Y. M. Cheung. Learning multi-boosted HMMs for lip-password based speaker verification. *IEEE Transactions on Information Forensics and Security*, 9(2):233–246, 2014.

[27] X. Liu, Y. M. Cheung, and Y. Yan. Lip event detection using oriented histograms of regional optical flow and low rank affinity pursuit. *Computer Vision and Image Understanding*, 148:153–163, 2016.

[28] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[29] J. Luettin and G. Matre. Evaluation protocol for the extended M2VTS database (XM2VTSDB). *IDIAP*, 1998.

[30] J. Luettin, N. A. Thacker, and S. W. Beet. Visual speech recognition using active shape models and hidden markov models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 817–820 vol. 2, 1996.

[31] K. Mase and A. Pentland. Automatic lipreading by optical flow analysis. *Systems & Computers in Japan*, 22(6):67–76, 2015.

[32] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, 2002.

[33] A. Mehra, M. Kumawat, R. Ranjan, and B. Pandey. Expert system for speaker identification using lip features with PCA. In *International Workshop on Intelligent Systems and Applications*, pages 1–4, 2010.

[34] L. L. Mok, W. H. Lau, S. H. Leung, S. L. Wang, and H. Yan. Lip features selection with application to person authentication. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages iii–397, 2004.

[35] K. Murphy and M. Dunham. Pmtk: Probabilistic modeling toolkit. In *Neural Information Processing Systems Workshop on Probabilistic Programming*, 2008.

[36] B. Ni, P. Moulin, and S. Yan. Order preserving sparse coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1615–1628, 2015.

[37] K. Paleek. Lipreading using spatiotemporal histogram of oriented gradients. In *Signal Processing Conference*, pages 1882–1885, 2016.

[38] Y. Pei, T. K. Kim, and H. Zha. Unsupervised random forest manifold alignment for lipreading. In *IEEE International Conference on Computer Vision (ICCV)*, pages 129–136, 2013.

[39] E. Petajan. Automatic lipreading to enhance speech recognition. *Proc IEEE Globcom*, 1984.

[40] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in Speech Recognition*, 77(2):267–296, 1990.

[41] C. M. Rader and B. Gold. Digital filter design techniques in the frequency domain. *Proceedings of the IEEE*, 55(2):149–171, 1967.

[42] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[43] U. Saeed. Person identification using behavioral features from lip motion. In *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, pages 131–136, 2011.

[44] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 994–1000, 2005.

[45] A. A. Shaikh, D. K. Kumar, and J. Gubbi. Automatic visual speech segmentation and recognition using directional motion history images and zernike moments. *The Visual Computer*, 29(10):969–982, 2013.

[46] X. X. Shi, S. L. Wang, and J. Y. Lai. Visual speaker authentication by ensemble learning over static and dynamic lip details. In *IEEE International Conference on Image Processing*, pages 3942–3946, 2016.

[47] S. L. Wang and A. W. C. Liew. ICA-based lip feature representation for speaker authentication. In *International IEEE Conference on Signal-Image Technologies and Internet-Based System*, pages 763–767, 2007.

[48] S. L. Wang and A. W. C. Liew. Physiological and behavioral lip biometrics: A comprehensive study of their discriminative power. *Pattern Recognition*, 45(9):3328–3335, 2012.

[49] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1794–1801, 2009.

[50] P. Yin, I. Essa, and J. M. Rehg. Asymmetrically boosted hmm for speech reading. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–755, 2004.

[51] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1944–1951, 2014.

[52] X. T. Yuan and S. Yan. Visual classification with multi-task joint sparse representation. *IEEE Transactions on Image Processing*, 21(10):3493–3500, 2010.

[53] X. Zhang, R. M. Mersereau, M. Clements, and C. C. Broun. Visual speech feature extraction for improved speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages II–1993–II–1996, 2002.

[54] G. Zhao, T. Ahonen, J. Matas, and M. Pietikainen. Rotation-invariant image and video description with local binary pattern features. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 21(4):1465–1477, 2012.

[55] G. Zhao, M. Barnard, and M. Pietikainen. Lipreading with local spatiotemporal descriptors. *IEEE Transactions on Multimedia*, 11(7):1254–1265, 2009.

[56] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.

CURRICULUM VITAE

Academic qualification of the thesis author, Ms. ZHOU Yichao:

- Received the degree of Bachelor of Engineering from University of Science and Technology of China, China, June, 2014.

August 2018