Towards a Practical Lipreading System

Ziheng Zhou, Guoying Zhao and Matti Pietikäinen
Machine Vision Group, Computer Science and Engineering Laboratory, University of Oulu
P.O. Box 4500, FI-90014, Oulu, Finland
{ziheng.zhou,guoying.zhao,mkp}@ee.oulu.fi

Abstract

A practical lipreading system can be considered either as subject dependent (SD) or subject-independent (SI). An SD system is user-specific, i.e., customized for some particular user while an SI system has to cope with a large number of users. These two types of systems pose variant challenges and have to be treated differently. In this paper, we propose a simple deterministic model to tackle the problem. The model first seeks a low-dimensional manifold where visual features extracted from the frames of a video can be projected onto a continuous deterministic curve embedded in a path graph. Moreover, it can map arbitrary points on the curve back into the image space, making it suitable for temporal interpolation. Based on the model, we develop two separate strategies for SD and SI lipreading. The former is turned into a simple curve-matching problem while for the latter, we propose a video-normalization scheme to improve the system developed by Zhao et al. We evaluated our system on the OuluVS database and achieved recognition rates more than 20% higher than the ones reported by Zhao et al. in both SD and SI testing scenarios.

1. Introduction

Recognizing what people speak is one of the major tasks in machine vision. It is known that speech perception is a multimodal process which involves information not only from what we hear (audio) but from what we see (visual) [11]. Although visual information is insufficient to distinguish languages completely, when combined with audio, it could significantly improve the performance of speech recognition [13]. When there are a limited number of utterances to be identified, it is possible to use visual information only to do speech recognition. The technique, so-called lipreading, is an important alternative to traditional speech recognition technology in human-machine interactions, especially when audio is unavailable or seriously corrupted by background noise.

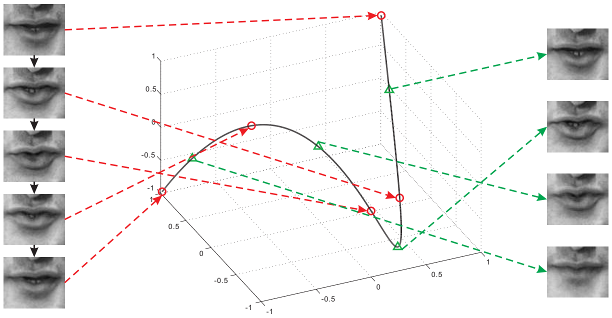


Figure 1. An example of a speech video (left) being projected onto a low-dimensional curve (middle) from which the images (right) are synthesized.

From the practical point of view, a lipreading system can be considered as subject-dependent (SD) or subject-independent (SI). The former is often used in a private environment and customized for a particular user. For instance, to avoid unnecessary distractions, such a system could be built in a smart vehicle to read voice commands from the driver to operate different devices. Since the system is user-specific, the training data is probably provided directly by the user. Considering usability, it is user-unfriendly that each utterance would be repeated by the user more than a couple of times. Therefore, the lipreading algorithm has to deal with a small-training-sample-size (STSS) problem.

On the other hand, a subject-independent system is aimed to serve a large group of users in a public environment. It can be integrated, for example, into a social robot to capture vocal queries from users. Unlike the SD systems, such a system has to cope with various challenges posed by allowing unknown users to access the system. The difficulties include large variations within lip shapes, skin textures around the mouth, varying speaking speeds and different accents, which could significant affect the spatiotemporal appearances of a speaking mouth. The system training is often based on a database that obtains, for each utterance, dozens of speech videos from multiple subjects to represent the variations.

As a brief summary, an SD system deals with one user at a time, which makes the spatiotemporal appearances of the utterances relatively stable. However, it has to tackle

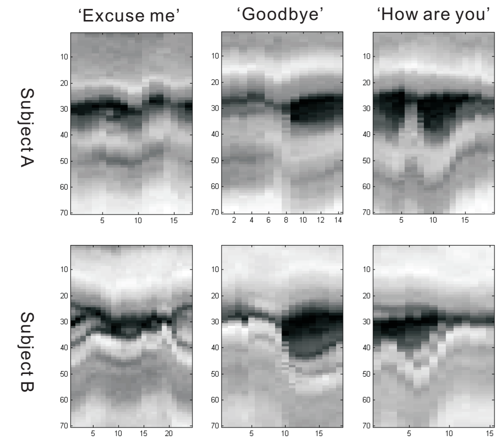


Figure 2. Temporal patterns of the videos of three utterances (top row) spoken by two subjects. The abscissa and the ordinate show the image frame and row indices, respectively. All the patterns are the vertical sections at the 30th column (from left to right).

the STSS problem, making it difficult to use complicated classifiers. On the other hand, an SI system has to cope with large variations within the spatiotemporal appearances of the utterances, making it challenging to extract discriminative features. It is usually trained on a large database. The difference between the two systems is nontrivial and we need to develop different strategies for them. Unfortunately, such an issue has not been addressed explicitly by the existing methods [2, 4, 10, 14, 18, 19], which attempted to do lipreading in a uniform way.

In this paper, we propose a deterministic model for lipreading. Given a speech video, as illustrated in Figure 1, the learned model can map high-dimensional visual features extracted from the images onto a low-dimensional continuous curve determined by a set of trigonometric functions embedded within a path graph, and project the entire curve back into the image domain. Based on the model, we turn the problem of SD lipreading into a simple curve matching problem to tackle the STSS problem. For building the SI system, we propose a novel video-normalization scheme to extract more discriminative spatiotemporal features for recognition. We evaluate the system on the public-available OuluVS database that contains phrases and short sentences and the results are significantly better than the best reported performance in both SD and SI experiments.

2. Related Work

A comprehensive review of lipreading or more generally, audio-visual speech recognition can be found in [13]. In this section, we briefly mention the research work on lipreading (or visual-only speech recognition).

For our problem, the key question is how to classify the temporal changes of the appearance of a mouth in a sequence of images. Inspired by the success in audio speech recognition, researchers considered a visual speech signal as a Markov process and used a hidden Markov model (HMM) or more generally, a dynamic Bayesian network (DBN) to model the video dyanmics [2, 10, 14].

In [14], such a statistical model was demonstrated to be useful for building an SD lipreading system. The system was designed to recognize phrases spoken by three subjects, simulating an in-car stereo controlled by voice commands. Although the authors demonstrated the superiority of their method over the traditional viseme- and HMM-based systems, the best reported recognition rate was below 70%, not accurate enough for a practical SD system.

Recently, Zhao *et al.* [18] proposed to use a spatiotemporal local texture descriptor to capture video dynamics. They considered the whole video sequence as a volume and calculated LBP features from not only the original mouth images, but from the accumulated temporal patterns which were the cross/vertical sections of the volume. Figure 2 illustrates such patterns. It can be seen that although with different temporal resolutions, the patterns of the same utterance show clear similarities meanwhile distinguishable from others. In their work, every video volume was divided, in the same way, into smaller rectangular cuboids from which normalized LBP histograms were computed. The histograms were then concatenated to form a feature vector and classified by SVMs. They demonstrated that significant discriminative power came from the temporal patterns which were less sensitive to speakers. Such a characteristic makes it useful for building an SI system.

One problem of implementing this method is that it requires the input video to be long enough such that the spatiotemporal feature-extraction can be performed. Such a requirement could sometimes stop us from extracting finer multiresolution features that might be highly discriminative for recognition. We will further discuss this issue in Section 3.4 and provide a solution to remove the requirement. Moreover, their system, as mentioned in [18], could not handle the single-training-sample situation.

In [4, 19], graph embedding (GE) was used to tackle the problem of lipreading. Zhou *et al.* [19] demonstrated the effectiveness of their system in the SD scenario. However, they did not show the results for SI lipreading. Similar to the HMM/DBN-based methods, the GE-based approaches extracted visual features only from video frames (not from the temporal patterns). Therefore, we would expect that in an SI scenario, the large inter-speaker variations could significantly downgrade the system performance.

3. Proposed Method

In this section, we describe the motivation (Section 3.1) and construction (Section 3.2) of the proposed deterministic model and the strategies for building SD and SI lipreading systems in Sections 3.3 and 3.4, respectively.

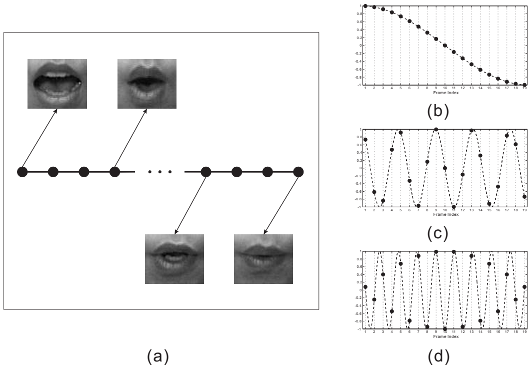


Figure 3. (a) Graph (P_{19}) representation of a video with 19 frames in total and (b)-(d) the 1st, 9th and 18th eigenvectors of the Laplacian of the graph. The dots mark the values of the elements of the eigenvectors. The dash lines show the curves of f_1^{19}, f_9^{19} and f_{18}^{19} on which the eigenvectors lie.

3.1. Graph Representation

If we consider the movement of a talking mouth as a continuous process, a speech video can be viewed as a set of images sampled at a fixed pace along a curve that represents the utterance in the image space, or more generally, the space of the visual features extracted from the images. Typically, such a space has a high dimension and we may assume that there exists a low-dimensional manifold within which the continuous process of uttering can be characterized by a continuous and deterministic function.

In our work, we reveal such a function through representing the input video as a path graph P_n where n is the number of vertices. An example of such a graph representation is given in Figure 3(a). As shown in the figure, each vertex corresponds to a frame and the connections between the vertices can be represented by an adjacency matrix $\boldsymbol{W} \in \{0,1\}^{n \times n}$ where $W_{ij} = 1$ if $|i-j| = 1,\ i,j = 1,2,\ldots,n$ and 0 otherwise. As described in [1], to get the manifold embedded in the graph, we can consider the problem of mapping P_n to a line such that connected vertices stay as close as possible. Let $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$ be such a map and we can obtain $\boldsymbol{y}$ by minimizing

$$\sum_{i,j} (y_i - y_j)^2 W_{ij}, \quad i,j = 1,2,\ldots,n. \tag{1}$$

It is equivalent to calculate the eigenvectors of the graph Laplacian $\boldsymbol{L}$ [3] of P_n. The matrix $\boldsymbol{L}$ is defined as: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is a diagonal matrix with the ith diagonal entry computed as $D_{ii} = \sum_{j=1}^{n} W_{ij}$. According to the definition of $\boldsymbol{L}$, it is not difficult to verify that it has $n-1$ eigenvectors $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n-1}\}$ with non-zero eigenvalues $\lambda_1 < \lambda_2 < \cdots < \lambda_{n-1}$ and the uth element $(u = 1,2,\ldots,n)$ of $\boldsymbol{y}_k$ $(k = 1,2,\ldots,n-1)$ is determined by:

$$\boldsymbol{y}_k(u) = \sin\left(\pi k u/n + \pi(n-k)/2n\right). \tag{2}$$

If we replace u by $t = u/n$ in Equation 2, $\boldsymbol{y}_k$ can be viewed as a set of points on the curve described by functions $f_k^n(t) = \sin\left(\pi k t + \pi(n-k)/n\right),\ t \in [1/n, 1]$ sampled at $t = 1/n, 2/n, \ldots, n/n$. Figures 3(b)-(d) illustrate the 1st, 9th and 18th eigenvectors (black dots) of path graph P_{19} and functions f_1^{19}, f_9^{19} and f_{18}^{19} (dashed curves). It can be seen that the temporal relations between the video frames are governed by the curve, which motivates us to make an assumption that the unseen mouth images occurring in the continuous process of uttering can also be characterized by the curve defined by function $\mathcal{F}^n : [1/n, 1] \rightarrow \mathbb{R}^{n-1}$

$$\mathcal{F}^n(t) = \begin{bmatrix} f_1^n(t) \\ f_2^n(t) \\ \vdots \\ f_{n-1}^n(t) \end{bmatrix}. \tag{3}$$

3.2. Model Construction

The assumption we have made would not be useful unless we can find the way to connect video frames and the curve defined by $\mathcal{F}^n$. Given a video with n frames, we denote the visual features extracted from the frames as $\{\boldsymbol{\xi}_i \in \mathbb{R}^m\}_{i=1}^n$ where m is the dimension of the visual-feature space. Note that when the features are simply defined as the raw pixel values, $\boldsymbol{\xi}_i$ is the vectorized ith frame.

We start from establishing a projection from $\boldsymbol{\xi}_i$ to the points defined by $\mathcal{F}^n(1/n), \mathcal{F}^n(2/n), \ldots, \mathcal{F}^n(1)$. Typically, $n \ll m$ and we assume that vectors $\boldsymbol{\xi}_i$ are linearly independent. The mean $\bar{\boldsymbol{\xi}}$ is calculated and removed from $\boldsymbol{\xi}_i$. The mean-removed vectors are denoted as $\boldsymbol{x}_i = \boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}$. Based on the assumption on $\boldsymbol{\xi}_i$, matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$ has a rank equal to $n-1$.

Recall that we represent the video by graph P_n with adjacency matrix $\boldsymbol{W}$. By using the linear extension of graph embedding [17], we can learn a transformation vector $\boldsymbol{w}$ that minimizes

$$\sum_{i,j} \left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_j\right)^2 W_{ij}, \quad i,j = 1,2,\ldots,n. \tag{4}$$

Vector $\boldsymbol{w}$ can be computed as the eigenvector of the following generalized eigenvalue problem:

$$\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w} = \lambda^{'}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}. \tag{5}$$

He *et al.* [7] solved the above problem using the singular value decomposition [5] on $\boldsymbol{X}$, *i.e.*, $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$ and then turned the problem into a normal eigenvalue problem

$$\begin{aligned} \boldsymbol{A}\boldsymbol{v} &= \lambda^{'}\boldsymbol{v} \\ \boldsymbol{A} &= (\boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}})^{-1}(\boldsymbol{Q}\boldsymbol{L}\boldsymbol{Q}^{\mathrm{T}}) \\ \boldsymbol{Q} &= \boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}. \end{aligned} \tag{6}$$

such that $\boldsymbol{w} = \boldsymbol{U}\boldsymbol{v}$. Since $\boldsymbol{Q} \in \mathbb{R}^{(n-1)\times n}$, $\boldsymbol{A} \in \mathbb{R}^{(n-1)\times(n-1)}$, and they are both of full rank.

Let $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{n-1}$ be the eigenvectors of $\boldsymbol{A}$ with their eigenvalues $\lambda_1^{'} \leq \lambda_2^{'} \leq \cdots \leq \lambda_{n-1}^{'}$. From Equation 6, for each $\boldsymbol{v}_k$ $(k = 1, 2, \ldots, n-1)$ we have:

$$\begin{aligned} (\boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}})^{-1}(\boldsymbol{Q}\boldsymbol{L}\boldsymbol{Q}^{\mathrm{T}})\boldsymbol{v}_k &= \lambda_k^{'}\boldsymbol{v}_k \\ \Rightarrow \boldsymbol{L}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{v}_k &= \lambda_k^{'}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{v}_k \end{aligned} \tag{7}$$

It can be seen that vectors $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{v}_k$ are eigenvectors of L. Therefore,

$$\begin{aligned} \lambda_k^{'} &= \lambda_k \\ \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{v}_k &= m_k \boldsymbol{y}_k \end{aligned} \tag{8}$$

where m_k is a scaling constant. Without loss of generality, m_k can be evaluated as the ratio of the first element of vector $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{v}_k$ to the first element of $\boldsymbol{y}_k$:

$$m_k = \frac{\sum_{i=1}^{n-1} \boldsymbol{Q}_{i1}\boldsymbol{v}_k(i)}{\boldsymbol{y}_k(1)}. \tag{9}$$

Let $\boldsymbol{M}$ be a diagonal matrix with $M_{kk} = m_k$, $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n-1}]$ and $\boldsymbol{\Upsilon} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{n-1}]$. From Equation 8 and $\boldsymbol{Q} = \boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}$, we have

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{\Upsilon} = \left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}\right)^{\mathrm{T}}\boldsymbol{\Upsilon} = \boldsymbol{Y}\boldsymbol{M}. \tag{10}$$

Recall that vectors $\boldsymbol{y}_k$ are determined by a set of trigonometric functions f_k^n (see Equation 2). We can write matrix $\boldsymbol{Y}$ as:

$$\begin{aligned} \boldsymbol{Y} &= \Big[\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n-1}\Big] \\ &= \begin{pmatrix} f_1^n(1/n) & f_2^n(1/n) & \cdots & f_{n-1}^n(1/n) \\ f_1^n(2/n) & f_2^n(2/n) & \cdots & f_{n-1}^n(2/n) \\ \vdots & \vdots & \ddots & \vdots \\ f_1^n(n/n) & f_2^n(n/n) & \cdots & f_{n-1}^n(n/n) \end{pmatrix} \end{aligned} \tag{11}$$

From Equation 3, $\boldsymbol{Y}^{\mathrm{T}} = [\mathcal{F}^n(1/n), \mathcal{F}^n(2/n), \ldots, \mathcal{F}^n(1)]$. The visual features can be projected onto the curve by:

$$\mathcal{F}^n(i/n) = \left(\boldsymbol{M}^{-1}\boldsymbol{\Upsilon}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\right)(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}), \quad i = 1, 2, \ldots, n. \tag{12}$$

Here, we define a function $\mathcal{F}_{\text{map}}$ to describe the projection: $\mathcal{F}_{\text{map}} : \mathbb{R}^m \rightarrow \mathbb{R}^{n-1}$

$$\mathcal{F}_{\text{map}}(\boldsymbol{\xi}) = \left(\boldsymbol{M}^{-1}\boldsymbol{\Upsilon}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\right)(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}). \tag{13}$$

So far, we have found the map from $\boldsymbol{\xi}_i$ to their correspondences on the curve through $\mathcal{F}_{\text{map}}$. Now the question is raised that whether such a map is reversible. Once again, since the mean is removed from $\boldsymbol{\xi}_i$, resulting in $\text{rank}(\boldsymbol{X}) = n-1$, $\boldsymbol{\Upsilon}$ is a $(n-1) \times (n-1)$ square matrix of full rank and hence, $\boldsymbol{\Upsilon}^{-1}$ exists. From Equation 13, we have

$$\boldsymbol{\xi}_i = \boldsymbol{U}\left(\boldsymbol{\Upsilon}^{-1}\right)^{\mathrm{T}}\boldsymbol{M}\mathcal{F}^n(i/n) + \bar{\boldsymbol{\xi}}. \tag{14}$$

It can be seen that the map is reversible. If the visual feature space coincides with the image space, we can establish a function $\mathcal{F}_{\text{syn}}$ that can not only reconstruct but temporally interpolate the input video by controlling a single variable t: $\mathcal{F}_{\text{syn}} : [1/n, 1] \rightarrow \mathbb{R}^m$

$$\mathcal{F}_{\text{syn}}(t) = \boldsymbol{U}\left(\boldsymbol{\Upsilon}^{-1}\right)^{\mathrm{T}}\boldsymbol{M}\mathcal{F}^n(t) + \bar{\boldsymbol{\xi}}. \tag{15}$$

Discussions on Linear-Independence

The proposed model (Equations 13 and 15) is built on the assumption that $\boldsymbol{\xi}_i$ are linearly independent. For a practical lipreading system, the utterances (e.g., voice commands) to be recognized are usually phrases or short sentences and their lengths are no longer than a couple of seconds. Given the feature dimension m much larger than the video length n, the assumption is very likely to hold. We have tested the assumption on more than 800 videos recorded at 25 fps and found it valid for all the time. In case of $\boldsymbol{\xi}_i$ being linearly dependent, we suggest to downsample the video (*e.g.*, using only the odd frames) to make them linearly independent.

3.3. Subject Dependent

In this subsection, we describe the strategy of building a subject-dependent lipreading system. Here, the main challenge is that the number of training videos for each utterance is limited, *i.e.*, the small-training-sample-size problem. We tackle the problem by proposing a robust measurement on the similarity between a test video $\boldsymbol{\Psi}^1$ and a training video $\boldsymbol{\Psi}^0$. Note that $\boldsymbol{\Psi}^0$ only contains frames corresponding to a speaking mouth, while $\boldsymbol{\Psi}^1$ may include irrelevant images of a non-speaking mouth. Since the system is user-specific, the two videos record the mouth movements of the same person. If they belong to the same utterance, we would expect to see in $\boldsymbol{\Psi}^1$ the mouth appearances very similar to those in $\boldsymbol{\Psi}^0$.

We can learn a projection $\mathcal{F}_{\text{map}}$ that maps video frames of $\boldsymbol{\Psi}^0$ onto a deterministic curve within a low-dimensional space. We then project frames of $\boldsymbol{\Psi}^1$ into the space using $\mathcal{F}_{\text{map}}$. The similarity between the projected trajectory from $\boldsymbol{\Psi}^1$ and the curve from $\boldsymbol{\Psi}^0$ can be used to quantify the similarity between the two videos.

From Equation 3, we can see that the curve along each dimension is defined by a trigonometric function f_k^n. Instead of solving a multi-dimensional curve matching problem, we simply calculate the correlation between the projected trajectory and the curve of f_k^n along each dimension. Figure 4 illustrates the process of curve matching. To do that, we first define a sliding window $W(h, t)$ where h is the width and t is its starting position in $\boldsymbol{\Psi}^1$. We let h vary in a range of $L_0(1 \pm 20\%)$ to counter different speaking speed. Here, L_0 is the length of $\boldsymbol{\Psi}^0$. Let $Corr(k, h, t)$ be the correlation between the projected trajectory in W and

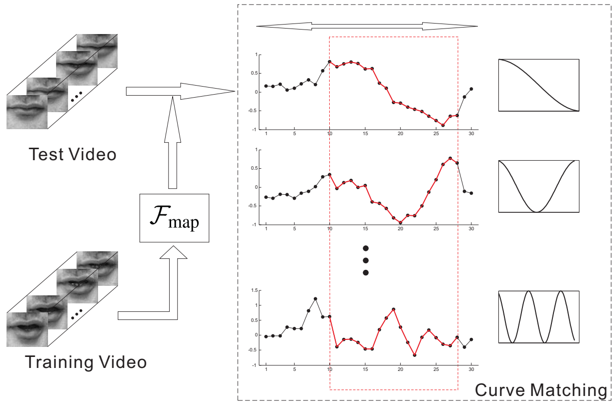


Figure 4. Process of curve matching. A test video is firstly mapped into a low-dimensional space using $\mathcal{F}_{\text{map}}$ learned from a training video. Along each dimension, the projected trajectory inside a sliding window (red solid line) is matched against the curve of a trigonometric function (shown beside the projected curve).

the curve of f_k^n along the kth dimension. We define the similarity $sim(\mathbf{\Psi}^1, \mathbf{\Psi}^0)$ as:

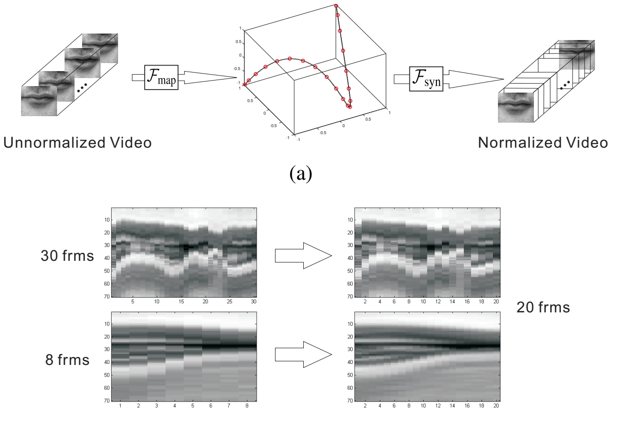
$$sim(\mathbf{\Psi}^1, \mathbf{\Psi}^0) = \max_{h,t} \sum_{k=1}^{K} Corr(k, h, t). \tag{16}$$

Here, we only consider the first $K = 6$ dimensions to limit the effect from noise. When doing lipreading, a test video is compared with training videos of each utterance and the computed similarities are averaged. The maximum mean similarity reveals the identity of the utterance spoken in the video. We can see that the proposed method requires neither multiple samples of the same utterance for training, nor to know the exact boundaries of the utterance in the test video, making it suitable for practical applications.

3.4. Subject Independent

To tackle the large variations within the spatiotemporal appearances of utterances, our strategy for building a subject-independent lipreading system is to first normalize speech videos to have a standard length and to apply the spatiotemporal local texture descriptor (SLTD) [18] to the normalized videos for feature extraction and recognition. As mentioned in Section 2, the SLTD-based method has the advantage in case of the SI lipreading since the descriptor allows us to extract features from temporal patterns, which are less sensitive to variant speakers.

We have also mentioned that the SLTD requires an input video to have a minimum length. Here, we discuss this issue in more details. In [18], they divided a video into 3 segments along the time axis and used an LBP descriptor of radius $R = 3$ pixels to calculate features on its temporal patterns. According to [18], the first and last 3 frames need to be removed from the image volume since the descriptor could not be placed there. To allow each segment to have at least one frame, the video has to contain more than 9 ($= 2 \times$ radius $+$ number of segments) frames. The more frames to be included in each segment, the longer the video has to be. Moreover, if the segments contained only a couple of frames, we would expect the calculated histograms to be statistically less stable.



(a)

(b)

Figure 5. Video normalization: (a) illustrates the normalization process that maps the original video onto a curve and sample a novel video along the curve. (b) shows the temporal patterns of two videos with 8 and 30 frames and their counterparts when both are normalized to be 20-frame long.

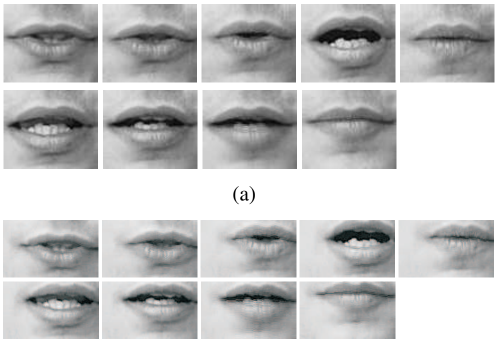
In this work, video normalization is done based on the proposed model. Figure 5 illustrates the process and effect of the video normalization. Given an n-frame video, we simply use the raw pixel values as the visual features. From the video, we can learn function $\mathcal{F}_{\text{syn}}$ that projects the embedded curve back into the image space. Let m be the length of the normalized video. We first sample evenly m points on the curve, or equivalently, choose m variables $t_1, t_2, \ldots, t_m$ that evenly separate interval $[1/n, 1]$. Here $t_1 = 1/n$ and $t_m = 1$. Images are then interpolated by $\boldsymbol{\xi}_i^{\text{syn}} = \mathcal{F}_{\text{syn}}(t_i)$, $i = 1, 2, \ldots, m$. In Figure 5(b), it can be visually seen that the normalized temporal patterns preserve well the characteristics of the original ones. After normalization, the lipreading can be done in the same way as described in [18].

It has to be noticed that given a video for normalization, we have to remove frames corresponding to a non-speaking mouth to maximize system performance. It has been found out that including too many such frames could significantly downgrade the final lipreading performance. In this work, the removal is done by an SVM classifier [14].

4. Experiments and Results

4.1. Data Description

In recent years, a few publicly available databases have been published for researching audio-visual speech recog-


(a)

(b)

Figure 6. An example of the manually (a) and automatically (b) preprocessed videos.

nition, lipreading and multimodal biometric recognition. These databases including the XM2VTSDB [12], VidTIMIT [15], AVletters [10], AVICAR [9], AVTIMIT [6] and OuluVS [18] are all possible candidates for testing our system. However, after checking the details, we found out that the reading materials used in most of the databases were either single digits, single letters or complete sentences. In [6, 15], the sentences read by each subject were even different from speaker to speaker.

In this work, we choose the OuluVS database as our test dataset. Its reading materials consist of ten different phrases (see [18] for details), which can be used to simulate the voice commands users may speak to a machine. Moreover, each of the phrases was read by everyone of the 20 subjects up to 5 times, making it suitable for lipreading experiments. The speakers were from 4 different countries, making the dataset challenging due to their own accents and speaking rates. The video corpus was recorded in an indoor controlled environment. The frame rate was set as 25 fps and the image resolution was 720×576 pixels. Another suitable dataset was found in [14]. However, it is not publicly available yet.

4.2. Preprocessing

In general, a speech video is preprocessed through the following three steps: 1) detecting faces in its frames, 2) locating eyes, and 3) cropping off the mouth region. We found out that Step 2 could significantly affect the overall system performance.

In the experiments, the OuluVS database was first preprocessed manually since we wanted to know the upper bound of the performance of our systems. To do that, we manually located the eye positions every 5 frames and linearly interpolated the positions in the intermediate images. We then resized all the images to make eye distances constant. Finally, a 84×70 mouth region was cropped off each of video frames.

After that, the database was preprocessed automatically. With the help of [18]'s authors, we did the preprocessing in exactly the same way as they did in their experiments. Consequently, we could compare our results directly to theirs. Figure 6 gives an example of the manually and automatically preprocessed videos.

	clean	noisy
$\mathrm{LBP}^{\mathrm{u2}}_{(8,3)}$	95.2%	81.5%
$\mathrm{LBP}^{\mathrm{u2}}_{(16,4)}$	96.3%	83.3%
$\mathrm{LBP}^{\mathrm{u2}}_{(8,3)} + \mathrm{LBP}^{\mathrm{u2}}_{(16,4)}$	**96.5%**	**85.1%**
Zhou *et al.* [19]	90.7%	n/a
Zhao *et al.* [18]	n/a	64.2%

Table 1. Results of the subject-dependent experiments on the clean and noisy data.

4.3. Subject Dependent

We first tested our strategy proposed in Section 3.3 for building a subject-dependent lipreading system. For each of the 20 speakers, the leave-one-video-out cross validation was carried out, that is, one video was used for testing and the rest for training. The utterances spoken in the training video were located using their audio recordings [8] and the frames corresponding to a non-speaking mouth were removed from the videos. Note that such an operation was not done on the test video. After that, we then computed the mean similarity for each phrase as described in Section 3.3.

The multiresolution local binary pattern descriptors were used to extract visual features in the experiments. As mentioned above, we located eyes in two ways, manually and automatically, during preprocessing. We denoted the mouth images cropped based on the manually-found eye coordinates as the '*clean*' data and the other type as the '*noisy*' data. For the clean data, each image was divided into 1×5 (row $\times$ column) blocks. Two descriptors, $\mathrm{LBP}^{\mathrm{u2}}_{(8,3)}$ and $\mathrm{LBP}^{\mathrm{u2}}_{(16,4)}$, were used to extract visual features. See [18] for the details of the descriptors. For the noisy data, we also divided images into 1×5 blocks. However, each block was overlapped with its neighboring blocks to counter the vertical misalignment as illustrated in Figure 6 (b). Following [18], we used an overlap ratio of 70% of the original non-overlapping block size. The same two LBP descriptors were used to extract visual features.

Table 1 gives the recognition rates over all the 20 subjects on the clean and noisy data. Our proposed algorithm achieved high recognition rates on the clean data. We also fused the features by simply concatenating them to form a longer feature vector. The resulting recognition rate is slightly better. Our method was compared with the one described in [19]. The results were comparable since their test protocol coincided with ours and eyes were also located manually in their work. It can be seen that we have achieved

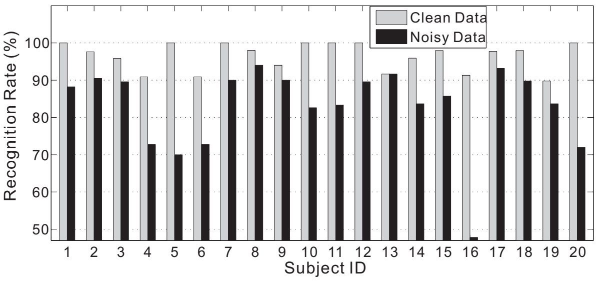


Figure 7. Subject-dependent results for each of the 20 subjects on the clean and noisy data.

more than 5% better performance on top of a 90% recognition rate.

As expected, our method performed worse on the noisy data where there was mouth misalignment. The recognition rates dropped about 10% from the clean to the noisy data. The fusion of the LBP features gave a substantial 2% improvement at this time. Since the noisy data and the test protocol were also used in [18], we quoted their experimental results for comparison here. It can be seen that our system significantly outperforms theirs in case of SD lipreading. In Figure 7, we show the recognition rates for each individual speaker on the fused features. As can be seen, the figures are consistent with the results reported in Table 1.

4.4. Subject Independent

We then tested our subject-independent lipreading strategy. All videos in the database were first preprocessed in the same way as we did in the previous experiments, resulting in the clean and noisy cropped mouth videos. For video normalization, we need to locate the boundaries of the utterance spoken in each video such that frames corresponding to a non-speaking mouth can be removed. In this work, it was done in two different ways. To obtain accurate utterance locations, we used the audio recordings. Since the database was collected indoor, background noise was minimized and therefore, algorithms such as [8] could be used to detect the boundaries with high accuracy. In a practical situation, the audio information might be unavailable, or seriously corrupted by background noise. In the second way, we built up an SVM classifier to classify the speaking and non-speaking mouth images. The classifier-training process was similar to the one described in [14].

To demonstrate the effectiveness of the video normalization, we used the same experimental settings described in [18]. Briefly speaking, the leave-one-subject-out cross validation was adopted for testing. The $\text{LBP}_{(8,3)}$ descriptor was used to calculate the LBP-TOP features, which were labelled as $\text{LBP-TOP}_{8,3}$. In each round of cross validation, an SVM classifier was trained for each pair of phrases. The second degree polynomial kernel was used in the classifier. The majority-voting scheme was implemented to decide which phrase the test video belonged to.

	Unnormalized	Normalized	
		Audio	SVM
$1 \times 5 \times 3$	67.8%	88.9%	79.2%
$1 \times 5 \times 4$	70.6%	90.9%	79.4%
$1 \times 5 \times 5$	70.3%	90.9%	79.0%
MKL Fusion	n/a	**92.8**%	**84.7**%

Table 2. Subject-independent results on the clean data. Here, 'Audio' means that videos are normalized using the audio-based method to locate the utterance boundaries while 'SVM' corresponds to the SVM-based approach.

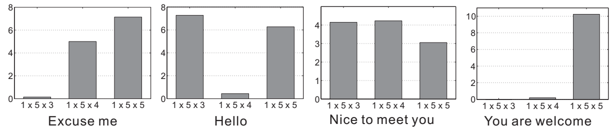


Figure 8. Mean weights corresponding to different types of features, learned by MKL when recognizing particular phrases.

In the experiments, all the videos were normalized to be 20-frame, 30-frame and 40-frame long. The 20-frame videos were divided into $1 \times 5 \times 3$ (row $\times$ column $\times$ time) rectangular cuboids, the 30-frame into $1 \times 5 \times 4$ cuboids and the 40-frame into $1 \times 5 \times 5$ cuboids. We denoted the features extracted using, for instance, the $1 \times 5 \times 3$ division as $\text{LBP-TOP}_{8,3}^{1\times5\times3}$. After discarding all the non-speaking frames, we found out that there were 7 (out of 817) videos containing less than 9 frames. As explained in Section 3.4, we could not calculate $\text{LBP-TOP}_{8,3}^{1\times5\times3}$ features from those videos. To let each time segment have more than two frames, the video had to have more than 12 frames. There were 87 videos found not to meet the requirement. It can be seen that without video normalization, we would be stopped from extracting finer spatiotemporal features from those videos.

Table 2 gives the recognition results on the clean data. Three different types of features were computed from the videos without normalization (unnorm), normalized using the audio-based utterance-locating method (norm + audio) and normalized using the SVM-based method (norm + SVM), respectively. It can be seen that the video normalization significantly boost the system performance (20% up from 'unnorm' to 'norm + audio' and 10% up from 'unnorm' to 'norm + SVM'). We also fused the features using the multiple kernel learning (MKL) [16]. Instead of training a classifier for each pair of phrases, we trained one classifier to distinguish one phrase against the others. The histogram-intersection kernel was used due to its high performance. From the table, we can see that the MKL further improved the performance by 2% on the 'norm + audio' and 5% on the 'norm+SVM' data. Figure 8 shows the weights learned by MKL. It can be seen that the three types of features contribute differently when recognizing different phrases,

	Recognition Rate
LBP-TOP$^{1\times5\times3}_{8,3}$ without normalization [18]	58.6%
LBP-TOP$^{1\times5\times3}_{8,3}$ with normalization (SVM)	76.7%
MKL-Fusion with normalization (SVM)	**81.3**%

Table 3. Subject-Independent results on the noisy data.

showing the advantage of fusing multiresolution features.

We then tested our SI strategy on the noisy data. At this time, only the SVM-based method was used to detect the utterance boundaries so as to make our system fully automatic and using visual information only. Video normalization and feature extraction were carried out in the same way as we did for the clean data. Table 3 lists the recognition results. Using the same LBP-TOP$^{1\times5\times3}_{8,3}$ features, the system achieved 76.7% on the normalized videos and only 58.6% on the unnormalized videos, once again, demonstrating the effectiveness of our proposed strategy. Through fusing variant multiresolution features, the recognition rate went up to 81.3%, more than 20% higher than the rate reported in [18].

Compared with the figures in Tables 2 and 3, it can be seen that the SI lipreading can be significantly affected by the performance of eye detection (during preprocessing) and utterance-boundary detection (during video normalization). It can be found out that the audio-based method outperforms the SVM-based one about 10% and the more accurate eye-detection method gives the system about 3% boost.

5. Conclusions

We have presented a simple model for solving the problems when building a practical lipreading system. Using the model, we can not only map the visual feature extracted from the video frames onto a low-dimensional continuous curve governed by a set of trigonometric functions, but project the curve back into the image space for temporal interpolation. We have proposed separate strategies for SD and SI lipreading. The former is turned into a simple curve-matching problem to counter the STSS problem and the latter is tackled by the introduction of a video-normalization scheme to extract more discriminative features from dataset containing large inter-speaker variations. Our system has been tested on the OuluVS database. The experimental results have demonstrated the superiority of our proposed strategies. The future work includes developing algorithms for accurate mouth detection and utterance-boundary detection to improve overall performance.

6. Acknowledgements

This research was supported by the Academy of Finland.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.

[2] G. Chou and J. Hwang. Lipreading from color video. *TIP*, 6(8):1192–1195, 1997.

[3] F. Chung. *Spectral graph theory*. American Mathematical Society, 1996.

[4] Y. Fu, M. Liu, M. Hasegawa-Johnson, and T. Huang. Lipreading by locality discriminant graph. In *ICIP*, pages 325–328, 2007.

[5] G. Golub and C. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[6] T. Hazen, K. Saenko, C. La, and J. Glass. A segment-based audio-visual speech recognizer: data collection, development and initial experiments. In *ICMI*, pages 235–242, 2004.

[7] X. He, D. Cai, S. Yan, and H. Zhang. Neighborhood preserving embedding. In *ICCV*, volume 2, pages 1208–1213, 2005.

[8] J. Junqua, B. Mak, and B. Reaves. A robust algorithm for word boundary detection in the presence of noise. *TPAMI*, 2(3):406–412, 1994.

[9] B. Lee, M. Hasegawa-Johnson, C. Goudeseune, S. Kamdar, S. Borys, M. Liu, and T. Huang. AVICAR: audio-visual speech corpus in a car environment. In *INTERSPEECH-2004*, pages 2489–2492, 2004.

[10] I. Matthews, T. Cootes, J. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *TPAMI*, 24(2):198–213, 2002.

[11] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[12] K. Messer, J. Matas, J. Kittler, J. Luettin, , and G. Maitre. XM2VTSDB: The extended M2VTS database. In *AVBPA*, pages 72–77, 1999.

[13] G. Potamianos, C. Neti, and G. Gravier. Recent advances in the automatic recognition of audio-visual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003.

[14] K. Saenko, K. Livescu, M. Siracusa, K. Wilson, J. Glass, and T. Darrell. Visual speech recognition with loosely synchronized feature streams. In *ICCV*, pages 1424–1431, 2005.

[15] C. Sanderson. The VidTIMIT database. Technical report, IDIAP Communication 02-06, 2002.

[16] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *JMLR*, 7:1531–1565, 2006.

[17] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *TPAMI*, 29(1):40–51, 2007.

[18] G. Zhao, M. Barnard, and M. Pietikäinen. Lipreading with local spatialtemporal descriptors. *TMM*, 11(7):1254–1265, 2009.

[19] Z. Zhou, G. Zhao, and M. Pietikäinen. Lipreading: a graph embedding approach. In *ICPR*, pages 523–526, 2010.