MKPLS: Manifold Kernel Partial Least Squares for Lipreading and Speaker Identification

Amr Bakry and Ahmed Elgammal
Computer Science Department, Rutgers University,
110 Frelinghuysen Rd, Piscataway, NJ 08854, USA
{amrbakry, elgammal}at cs.rutgers.edu

Abstract

Visual speech recognition is a challenging problem, due to confusion between visual speech features. The speaker identification problem is usually coupled with speech recognition. Moreover, speaker identification is important to several applications, such as automatic access control, biometrics, authentication, and personal privacy issues. In this paper, we propose a novel approach for lipreading and speaker identification. We propose a new approach for manifold parameterization in a low-dimensional latent space, where each manifold is represented as a point in that space. We initially parameterize each instance manifold using a nonlinear mapping from a unified manifold representation. We then factorize the parameter space using Kernel Partial Least Squares (KPLS) to achieve a low-dimension manifold latent space. We use two-way projections to achieve two manifold latent spaces, one for the speech content and one for the speaker. We apply our approach on two public databases: AVLetters and OuluVS. We show the results for three different settings of lipreading: speaker independent, speaker dependent, and speaker semi-dependent. Our approach outperforms for the speaker semi-dependent setting by at least 15% *of the baseline, and competes in the other two settings.*

1. Introduction

Audio visual speech recognition (AVSR) has been investigated intensively in the last few decades [19]. Specially after bimodal fusion of audio and visual stimuli in perceiving speech has been demonstrated by the *McGurk* effect [15]. For example, when the spoken sound /ga/ is seen as /ba/, then most people perceive the sound as /da/ [15]. Good survey for work on AVSR can be found in [19]. In the last two decades, with the advances in computer vision, visual speech recognition (VSR), also called lipreading, have attracted research attention [25]. VSR systems gain importance with the need for controlling machines verbally in a noisy environment. Example of such an environment is the car, where the noise (*e.g.* from motor and radio) makes it very hard for audio speech recognition. Another potential example is to control robot in the outer space where there is no media for audio transmission. Nevertheless, visual speech recognition is a challenging problem, due to confusion between visemes [1]. Specially, when using information only from plan marker-less and real life images.

Several approaches have been adopted for solving the lipreading problem. Two main approaches are commonly used in VSR literature: a Hidden Markov Model (HMM) based approach and classifier based approach. In the HMM approach, after choosing suitable descriptor for the visual unit (usually visemes) corresponding to every node, this descriptor employs as observations for the model. Then HMM model is trained using Baum-Welch algorithm for encoding the stochastic temporal relationship between these observations [14]. Consequently, the Viterbi algorithm [20] is used for classification. The classifier based approach is based on extracting a single feature vector for the whole clip of uttered phrase (usually single word, or short sentence), and train a classifier (usually SVM) based on that [27, 6]. The proposed approach in this paper belongs to the latter category.

Speaker identification and authentication are tightly coupled with speech recognition [13, 23, 25]. Speaker identification is defined as the ability to identify the speaker within a group of users from solely speech related features, like voice or mouth motion. Meanwhile, speaker authentication is the ability to authenticate users. We tackle the former problem in this paper. Speaker identification is related to several research fields such as automatic access control, biometrics, and personal privacy issues.

In this paper, we present a new approach for embedding of manifolds in a low-dimensional latent space. We ini-

[1] Viseme is the visual phoneme. It is defined as the smallest discriminative unit for visual speech

tially parameterize each manifold using a nonlinear mapping from a unified manifold representation, similar to [5]. However, unlike [5], where factorization of the manifold parameterization is achieve using unsupervised subspace projection, we factorize the parameterization space in a supervised way. We propose to use kernel partial least square (KPLS) on the mapping coefficient space to achieve a supervised low-dimensional latent space for manifold parameterization. We use two-way projections to achieve two manifold latent spaces, one for the speech content and one for the speaker. The resulting low-dimensional parameterization can be considered as a global spatio-temporal descriptor for each speech sequence, which can be effectively used for speech recognition and speaker identification.

The contribution of the paper can be contrasted in two ways. From learning point of view, we propose a new way to learn a low-dimensional supervised parameterization of manifolds where each manifold is represented as a point in a latent space. From the visual-speech point of view, we propose a new approach for projecting visual speech features into dual latent spaces that are capable of discriminating speech and speaker.

In this work, we use cosine similarity as a kernel on the parameterization space. Moreover, we use two different techniques for classifying new speech clip: one of them is SVM, we learn multi-class SVM based on the projected manifolds. The other one uses KPLS regression for classification on the latent space.

To test the effectiveness of our approach, empirically, we show that our approach outperform previous approaches applied on two databases: AVLetters [14] and OuluVs [27]. We tackle three different lipreading problems: speaker independent, speaker dependent, and speaker semi-dependent. In both databases, our approach outperforms for speaker semi-dependent setting by at least 15% over the baseline [27], and competes in the other two settings.

This paper is organized as follow: after this introduction, the related work will be reviewed in Section 2. The problem statement will be defined clearly and the manifold parameterization will be described in Section 3. Synopsis for KPLS is presented in Section 4. Thereafter, the proposed framework will be presented in details in two sections: first the manifold parameterization is described in Section 5, and the manifold embedding using KPLS is presented in Section 6. Section 7.1 lists the used datasets, and reveals all technical details used in the experiments. Experimental results will be shown in Section 7.

2. Related Work

Encoding the dynamics of speech video as a descriptor has a long history within lipreading research. Graphical models have been used extensively in VSR and AVSR. In [14], HMM was used for encoding the visual dynamics of speech using Active Shape Model (ASM) and Active Appearance Model (AAM). A more general Dynamic Bayesian Network(DBN) model has been used in [22] with different visual articulation units called articulatory features. Graph embedding has been used in [28] for estimating the curve that represent the dynamics in video. These methods try to capture the smooth temporal changes between the used visual units, but they may loose some visual information that may be crucial for discriminating small speech chunks like single letter utterance.

On the other hand, the work in [27] is based on extracting a single spatio-temporal feature vector for representing the visual and temporal information for the whole speech video. In [24] optical flow was used for extracting the whole word features. These two approaches outperform in the case of small size videos but it might be sensitive to frame outliers.

In our method, we care about smoothness, since we extract the geometric deformation of the lip-moving manifold and at the same time use all the appearance information for learning a parameterization for this manifold. We test our model on two databases, one contains small clip (AVLetters) and the other database contains slightly longer clips (OuluVs). As the best of our knowledge, we are the first to use homeomorphic manifold analysis and KPLS in the field of visual speech recognition.

3. Problem Definition and Framework Overview

We have a set of images sequences representing different activities. Let us denote the k-th sequence by $S_k = \{\mathbf{x}_i^k \in \mathbb{R}^D, i = 1 \cdots n_k\}$, where the images are represented using suitable features of dimensionality D. Let y_k represents the class labels for the k-th sequence. In this paper, for the particular case of speech recognition and speaker identification, $y_k \in \{c_1, \cdots, c_K\} \times \{p_1, \cdots p_L\}$. Here c_i is the activity class label (speech unit), while p_j is the performer class label (speaker). Each sequence lies on a low-dimensional manifold, denoted by $\mathcal{M}_k$, embedded in the feature space $\mathbb{R}^D$. We will denote these manifolds by *instance manifolds*. The basic assumption is that all these manifolds are topologically equivalent, however each of them has different geometry in $\mathbb{R}^D$. In other words, all these manifolds are deformed instances of each others. This assumption is fairly met in the domain of activity recognition. For example, periodic locomotive activities intuitively lie on one-dimensional closed manifolds, and hence topologically equivalent. For instance, sequence of features representing a Viseme, starting from a neutral pose and reaching a peak pose, lies on a one-dimensional manifold (curve) in the feature space.

The goal is to achieve a low-dimensional latent space of instance manifolds. In that space each manifold is represented by a single point. Based on that space, instance

classification can be achieved. We learn two classification functions $f_{speech}(S)$ and $f_{speaker}(S)$ based on two latent spaces for speech and speaker respectively.

The first step in our framework is to parameterize these manifolds to obtain a descriptor for each of them. The manifold parameterization we use is based on [5, 11]. We learn a regularized mapping function from a unified (average) low-dimensional embedded representation of all manifolds to each input manifolds. These mapping functions encode the geometric deformations between the unified representation and the original data manifolds. Therefore, the space of coefficients of these mapping functions provides a parameterization of the input manifold.

The obtained parameterization is high-dimensional, which makes it hard to learn classification functions that can generalize well. In [5] subspace analysis was used to obtain a latent representation of the manifold parameterization space. However such approach does not benefit from available class labels. Alternatively, we propose a supervised way to achieve a low-dimensional latent manifold parameterization space, which benefits from the class labels. Given the instance manifold parameterization, we propose two alternative manifold kernels based on the parameterization space. Given a manifold kernel, we use KPLS in the parameterization space to obtain a latent low-dimensional manifold parameterization space. We apply KPLS independently for the speech and speaker factors.

It worth mentioning that the unified manifold representation is supposed to be topological equivalent to each instance manifold. This can not simply be obtained by traditional Dimensionality Reduction (DR) on the whole input data. This is because the goal of DR approaches is to find an embedding that preserves the local (or global) geometry of the data. In contrast, the unified manifold representation is a collapsing of all instance manifolds to one average manifold. There are various ways that can be used to achieve this. In [5] individual manifolds are embedded and warped to compute an average embedding. Alternatively, if the topology of the manifold is known, a conceptual representation can be imposed; for example a unit circle can be used as topologically equivalent representation of all closed one-dimensional manifolds [11]. Another alternative is to use manifold alignment (*e.g.* [7]) to learn a unified embedding. In this paper, we work on top of such unified representation, independent of the approach used to achieve it.

4. Background: Kernel Partial Least Squares

Projection of data to a low-dimensional latent space is widely used in pattern classificaiton problems. The most common techniques for projection to a latent spaces are PCA and LDA [4]. Another technique that is widely used in chemometric pattern recognition is Partial Least Squares (PLS) [26, 21, 2]. Projection using PCA tends to keep most of the variance of the input space. In contrast, LDA tends to increase the clustering ability between different classes by maximizing the interclass and minimizing the intraclass distances [4]. PLS compromises by creating orthogonal components (in the latent space) using the existing correlations between explanatory variables (in the input space) and corresponding labeling, while keeping most of the variance of the points in the input space. A good interpretation for PLS and its relationship with iterative PCA can be found in [12, 2]. Additionally, PLS has been proven to be useful in situations where the number of the explanatory variables (dimensionality of the input space) exceeds significantly the number of observations and/or a high level of multicollinearity[2] among those variables.

For understanding the PLS, synopsis for PLS analysis [2] is presented here. PLS is a least squares regression-based technique. Like PCA regression (PCR), PLS finds a regressor $\mathbf{w}$, so that, $y_i \approxeq \mathbf{x_i}^\top \mathbf{w}, \forall i$, where $\mathbf{x_i}$ is the observation and y_i is its response (output). If we put that in a matrix form, the objective is to minimize the least squares error $\|\mathbf{XW} - \mathbf{y}\|^2$. Bennett [2] showed that

$$\|\mathbf{XW} - \mathbf{y}\|^2 \leq \|\mathbf{X} - \mathbf{yW}\|^2 .$$

Therefore, if we minimize $\leq \|\mathbf{X} - \mathbf{yW}\|^2$, we satisfy the objective. Then, he shows that

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{yW}\|^2 \propto \max_{\mathbf{W}} cov(\mathbf{XW}, \mathbf{y}), s.t. \mathbf{W}^\top \mathbf{W} = \mathbf{I}, \tag{1}$$

where cov stands for covariance. The solution of the Eq 1 has been shown to be

$$\mathbf{W} = \frac{\mathbf{X}^\top \mathbf{y}}{\mathbf{y}^\top \mathbf{X}\mathbf{X}^\top \mathbf{y}}, \tag{2}$$

which provides a closed form for $\mathbf{W}$.

However, for the high-dimensional observation space, Eq 2 is not robust and computationally inefficient. On the other hand, the NIPALS algorithm [26] is an iterative robust procedure for solving eigen-values and eigen-vectors problem, see Algorithm 1. Then NIPALS has be used later for PLS solution [26].

Henceforward, Lewis proves in [12] that we can get the same results by using the variance-covariance matrix $\mathbf{X}\mathbf{X}^\top$ instead of $\mathbf{X}$, which is significantly more computationally efficient than NIPALS in the case of dimensionality of the input space exceeds the number of observations. Moreover, he presents NIPALS-PLS algorithm for solving PLS in an iterative efficient way.

Then, Rosipal et al. [21] used the kernel trick[3] for inducing nonlinear version of the PLS (called KPLS). The KPLS

[2]Multicollinearity refers to a situation in which two or more explanatory variables in a multiple regression model are highly linearly related.

[3]Proposed in [1]. the kernel trick is commonly used technique in pattern recognition (*e.g.* KPCA and KSVM).

Algorithm 1 NIPALS algorithm - Single iteration

Ramdomly initialize $\mathbf{t}$
repeat
$\mathbf{p} \leftarrow \mathbf{X}^\top \mathbf{t}$
$\mathbf{t} \leftarrow \mathbf{X}\mathbf{p}$
$\mathbf{t} \leftarrow \frac{\mathbf{t}}{\|\mathbf{t}\|}$
until Convergence of $\mathbf{t}$ ▷ the resulting $\mathbf{t}$ is a single eigen-vector of $\mathbf{X}$.
$\mathbf{X} \leftarrow \mathbf{X} - \mathbf{t}\mathbf{t}^\top \mathbf{X}\mathbf{y}$ ▷ Data deflation

algorithm 2 is based on NIPALS-PLS, however, it uses the kernel form $\mathbf{K} = \Phi(\mathbf{X})\Phi(\mathbf{X})^\top$ instead of $\mathbf{X}\mathbf{X}^\top$.

Algorithm 2 KPLS algorithm

for $i \leftarrow 1 \rightarrow m$ **do** ▷ *m-dim latent space*
Ramdomly initialize u_i
repeat
$\mathbf{t}_i \leftarrow \mathbf{K}\mathbf{u}_i$
$\mathbf{t}_i \leftarrow \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|}$ ▷ *normalize vetor* $\mathbf{t}$
$\mathbf{u}_i \leftarrow \mathbf{y}^\top \mathbf{t}_i$
$\mathbf{u}_i \leftarrow \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|}$ ▷ *normalize vetor* $\mathbf{u}$
until Convergence in $\mathbf{t}_i$
$\mathbf{K} \leftarrow (\mathbf{I} - \mathbf{t_i}\mathbf{t_i}^\top)\mathbf{K}(\mathbf{I} - \mathbf{t_i}\mathbf{t_i}^\top)$ ▷ *Kernel deflation*
end for
$\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_m]$
$\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_m]$

5. Individual Manifold Parameterization

In this section we briefly describe parameterizing instance manifold. Let $\{\mathbf{x}_i^k \in \mathbb{R}^D, i = 1, \cdots, n_k\}$ be the input images for instance manifold $\mathcal{M}_k$, represented in a D-dimensional feature space. Let $\{\mathbf{z}_i^k \in \mathbb{R}^e, i = 1, \cdots, n_k\}$ be the corresponding embedded representation in an e-dimensional Euclidean space, which lie on the unified manifold $\mathcal{U}$. Notice that the number of points in each sequence (manifold) does not need to be equal.

We learn mapping functions $\gamma^k(\cdot) : \mathbb{R}^e \rightarrow \mathbb{R}^D$, which maps from $\mathcal{U}$ to each instance manifold $\mathcal{M}_k$. To learn such mappings, we learn individual functions $\gamma_l^k : \mathbb{R}^e \rightarrow \mathbb{R}$ for the l-th dimension in the feature space. Each of these functions minimizes a regularized loss functional in the form

$$\sum_i^{n^k} \left\|\mathbf{x}_{il}^k - \gamma_l^k(\mathbf{z}_i^k)\right\|^2 + \lambda\, \Omega[\gamma_l^k], \tag{3}$$

where $\|\cdot\|$ is the Euclidean norm, Ω is a regularization function that enforces the smoothness in the learned function, and λ is the regularizer that balances between fitting the training data and smoothing the learned function. When $\lambda \rightarrow 0$, the regression function over-fits the training data. From the representer theorem [9, 18] we know that such mapping functions admit a representation in the form of a linear combination of kernel basis functions in the embedding space $\mathbb{R}^e$. To achieve a common parameterization space of all the manifold, we use the same set of basis functions $K(\cdot, w_i), i = 1 \cdots n$, where $w_i \in \mathbb{R}^e$. The whole mapping can be written in the matrix form as

$$\gamma^k(\mathbf{z}) = \mathbf{C}_k \psi(\mathbf{z})$$

where $\mathbf{C}_k$ is a $D \times n$ matrix, and the vector $\psi(\mathbf{z}) = [K(\mathbf{z}, \mathbf{w}_1), \cdots, K(\mathbf{z}, \mathbf{w}_n)]$ represents a nonlinear kernel map from the embedded representation to a kernel induced space. The solution of Eq 3 is shown [18] to have closed form as

$$\mathbf{C}_k^\top = (\mathbf{A}_k^\top \mathbf{A}_k + \lambda \mathbf{G})^{-1} \mathbf{A}_k^\top \mathbf{X}_k^\top, \tag{4}$$

where $\mathbf{A}_k$ is an $n_k \times n$ matrix with $\mathbf{A}_{(ij)} = K(\mathbf{z}_i, \mathbf{w}_j)$ and $\mathbf{G}$ is an $n \times n$ matrix with $\mathbf{G}_{(ij)} = K(\mathbf{w}_i, \mathbf{w}_j)$. $\mathbf{X}_k$ is the $n_k \times D$ data matrix for instance k. Solution for $\mathbf{C}$ is guaranteed under certain conditions on the basis functions [18]. In this paper, we use Gaussian Radial Basis Function (Gaussian-RBF) for the kernel $K(\cdot, \cdot)$.

6. Manifold KPLS

6.1. Manifold Kernels

Given the manifold parameterization described above, a kernel in the space of manifolds can be defined as a kernel between their parameterizations, *i.e.*

$$K_{manifold}(\mathcal{M}_i, \mathcal{M}_j) \doteq K_{parameterization}(\mathbf{C}_i, \mathbf{C}_j). \tag{5}$$

Therefore, we need to define kernels over the space of parameterizations, which consequently, measure the similarity between manifolds in terms of their geometric deformation from the common manifold representation. We can use any valid kernel, in this section we propose using a kernel based on cosine similarity.

Cosine-manifold kernel:

Since each parameterization point $\mathbf{C}_k$ represents n-dimensional subspace in $\mathbb{R}^D$. Therefore, we can use cosine the angle between the two subspaces as a similarity in parameterization space. Therefore, the cosine-manifold kernel can be defined as

$$K_{\cos}(\mathbf{C}_i, \mathbf{C}_j) = \frac{tr(\mathbf{C}_i \mathbf{C}_j^\top)^2}{||\mathbf{C}_i||_F ||\mathbf{C}_j||_F}, \tag{6}$$

where $\|\cdot\|_F$ is matrix Frobenius norm.

In next section, we discuss the discriminant analysis for those parameterizations.

6.2. Manifold Latent Space

In our framework, we have a set of manifolds represented by $\{(\mathbf{C}_k, y_k), k = 1 \cdots N\}$. y_k is the categorical labeling of the manifold. We need to find nonlinear projection function $\mathcal{F} : \mathbb{C} \rightarrow \mathbb{R}^m$, where $\mathbb{C}$ is the space of all coefficient matrices, and $\mathbb{R}^m$ is a low-dimensional Euclidean space ($m \ll D$), so that $\mathcal{F}$ satisfies the objective

$$\min_{\mathcal{F}} \left\| \mathbf{C} - \mathcal{F}^{-1}(\mathcal{F}(\mathbf{C})) \right\|,$$
$$\max_{\mathcal{F}} cov(\mathcal{F}(\mathcal{C}), \mathbf{y})$$

where $\mathcal{C}$ is the set of parameterizations and $\mathbf{y}$ is the set of responses. We can write $\mathcal{F}$ in a nonlinear regression form as

$$\hat{y} = \Phi(\mathbf{C})\mathbf{B} - \mathbf{E} \tag{7}$$

where $\mathbf{B}, \mathbf{E}$ are the regression coefficients and residuals respectively.

For solving Eq 7, we can use kernel-PCA (KPCR) or kernel-Ridge Regression (KRR). However, using KPLS [21], produces embedding that maximizes the correlation with the response $\mathbf{y}$. KPLS Algorithm 2 finds projection function that embeds the parameterizations $\{\mathbf{C}_k, k = 1 \cdots N\}$ into a low-dimensional latent space $\mathbb{R}^m$, as $\{\mathbf{t}_k \in \mathbb{R}^m, k = 1 \cdots N\}$. The result of KPLS regression is

$$\hat{y} = \mathbf{KU}(\mathbf{T}^\top \mathbf{KU})^{-1}\mathbf{T}^\top \mathbf{y} \tag{8}$$

Let $\mathbf{R} = \mathbf{U}(\mathbf{T}^\top \mathbf{KU})^{-1}$. $\mathbf{R}$ works as the projection matrix[21]. Then, the matrix T, of all embedded points, can be written as

$$\mathbf{T} = \mathbf{KR} \tag{9}$$

For a new manifold $\mathcal{M}_\nu$, represented by its parameterization $\mathbf{C}_\nu$ and label y_ν (unknown), the corresponding embedded point can be given by

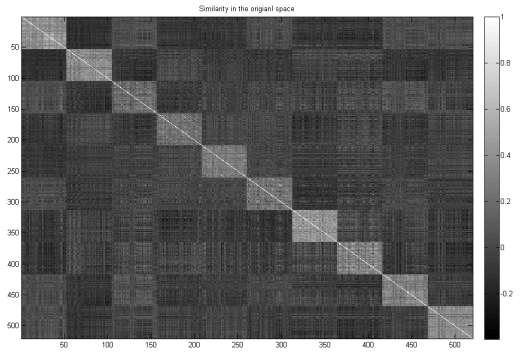
$$\mathbf{t}_\nu = \mathbf{v}_\nu \mathbf{R}. \tag{10}$$

Where $\mathbf{v}_\nu = K_{cos}(C_\nu, .)$ (Eq 6) is an N-dimensional row vector representing the similarity with all training manifold parameterizations $\{\mathbf{C}_k, k = 1 \cdots N\}$.

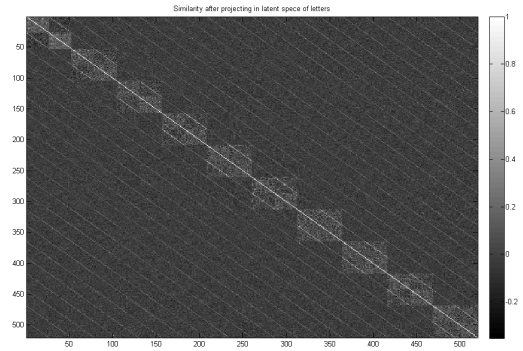
6.3. Multifactor Embedding

As aforementioned, we have set of labeled manifold parameterizations $\{(\mathbf{C}_k, y_k);\ k = 1 \cdots N\}$. Consider the case where we have multiple labeling for the same manifold. Therefore, we need to deal with different classification tasks. In this paper, we have two simultaneous tasks: speech recognition and speaker identification.

For phrase/speech recognition, the input manifolds have labeling $y_k^h, k = 1 \cdots N$. We can learn projection matrix $\mathbf{R}^h$ for embedded points $\mathbf{T}^h$ (**A**lgorithm 2).



(**a**)



(**b**)

Figure 1. AVletters: Similarity among points in the manifold parameterization original space (**a**), and after projection into the letters' latent space (**b**).

For any new manifold $\mathcal{M}_\nu$, $\mathbf{C}_\nu$ is compute (Eq 4), then get the corresponding embedded point by Eq 10, as $\mathbf{t}_\nu^h = \mathbf{v}_\nu \mathbf{R}^h$.

For speaker identification, we have different labeling $y_k^p, k = 1 \cdots N$. Similarly, we learn the projection matrix $\mathbf{R}^p$ and the embedded points $\mathbf{T}^p$. For new manifold $\mathcal{M}_\nu$, we compute the parameterization $\mathbf{C}_\nu$, then get the corresponding embedded point by $\mathbf{t}_\nu^p = \mathbf{v}_\nu R^p$.

Figure 1 shows the affect of projecting into the letters' latent space in the AVLetters database (see Section 7.1). In Figure 1**(a)** , the similarity between speaker dominates the similarity between letters. However in Figure 1**(b)** , the similarity between letters (represented by diagonals) dominates the similarity between speakers. In the same time, self-similarity between speakers still exist which means that the projection preserves the topological relationships in the original space.

6.4. Manifold Classification

At this point, we have a set of labeled low-dimensional representations for manifolds $\{(\mathbf{t}_k, y_k) \in \mathbb{R}^m \times \mathbb{R};\ k =$

$1 \cdots N\}$. Given a new manifold, parameterized by $\mathbf{C}_\nu$, we need to classify it, *i.e.* to get its class label $\hat{y}_\nu$. For achieving this goal, we use two alternative approaches:

Regression for classification (RfC) Use regression results of KPLS [21]

$$\hat{y}_\nu = \mathbf{t}_\nu \mathbf{T}^\top \mathbf{y}$$

where t_ν is computed from Eq 10.

Support vector machines (SVM) Learn one-vs-all SVM classifier for every class on the latent space, and use it for classifying the new embedded point $\mathbf{t}_\nu$, to get $\hat{y}_\nu$.

7. Experimental Results

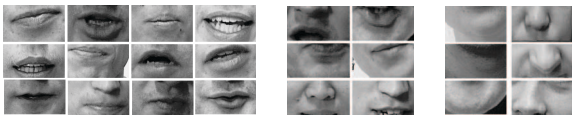
7.1. Databases

There are many databases available for AVSR, such as AVLetters [14], AVLetters 2 [3], AVICAR [10], AV-TIMIT [8], GUAVE [17] and OuluVS [27]. All AVSR databases can be used for VSR research by simply ignoring the audio information. Our choice is based on several factors. First, we are looking for recent work using solely visual data to compare with. Second, we need to test on different length spoken units. Third, reasonable image resolution. We find that the most adequate databases are AVLetters [14] and OuluVs [27] for speech recognition and speaker identification. In all experiments, the recognition rate is measured as the ratio between the correctly recognized clips and the total number of clips.

AVLetters database [4] [14] has ten subjects. Each speaker repeats every English letter ($A \cdots Z$) exactly three times, with a total of 780 video sequences. The speaker was requested to start and end utterance of every letter in a neutral state (mouth closed). No head motion/rotation is allowed from speakers. Every frame is a 60×80 pixel image of the mouth area. This database is very challenging for VSR. The best achieved accuracy for recognizing the spoken letter has been on this database is about 62% [27]. We use the following setting: For $\mathbf{LBP}$ features, we tried many configuration. The results is reported in terms of two of them: single cell eight-resolutions ($\mathbf{LBP}_{1:8\times8}$) and 3×4 cell-grid with four-resolutions ($_{3\times4}\mathbf{LBP}^{u_2}_{1:4\times8}$). For more details about **LBP**, reader is referred to [16].

OuluVS database [27] it consists of ten different everyday phrases. Each phrase is uttered by 20 subjects up to five times. The frame rate was set to 25 fps. The dataset contains sequence of images for mouth area with average resolution of 120×60 pixels. This database is less constrained than AVLetters, so that limited rotation and shift was allowed in the recording time, Figure 2(**a**). Not all sequences are perfectly segmented, so that, some sequences have few frames with partial-mouth (Figure 2(**b**)) or non-mouth frames (Figure 2(**c**)). Some of the outlier sequences (that contain very few mouth/partial-mouth frames) are excluded from the experiment. Consequently, we exclude four speakers with very few sequences remaining (*P004*, *P005*, *P010* and *P016*). The feature configurations used on this database are ($\mathbf{LBP}_{1:8\times8}$) and ($_{1\times2}\mathbf{LBP}^{u_2}_{1:8\times8}$).

[4] Public version is available on `http://www.ee.surrey.ac.uk/Projects/LILiR/datasets/avletters1/index.html`



(**a**) (**b**) (**c**)

Figure 2. OuluVs: (**a**) Regular frames, (**b**) Partial mouth area frames, (**b**) Non-mouth area frames.

7.2. Visual speech recognition

We adopt three test protocols for visual speech recognition: speaker independent, speaker dependent and speaker semi-dependent. To present a fair comparison, we restrict ourselves by the configuration specified in [27].

Speaker Independent VSR (SI): the challenge here is to recognize the uttered phrase, independent completely of the speaker. By this configuration, we show that our framework generalizes to users is not seen before in the training set. In this experiment, we use one-speaker-out technique.

Speaker Semi-Dependent VSR (SSD): here we test on one part of the available videos and train based on the remaining set of videos. With one condition that all speakers and phrases have to be presented in the training set. The challenge here is to classify the phrase/expression correctly regardless the user identity.

Speaker Dependent VSR (SD): this experiment tests how far our approach is adequate for use with limited data available. For every speaker, we left one video out for test, and trained based on the remaining videos for the same speaker.

Table 1 and Table 2 show the SI speech recognition accuracy for OuluVs and AVLetters, respectively. We can see that for solving speaker independent problem, we need a low-dimensional latent space (about 15 for OuluVs and 25 for AVLetters).

Table 3 and Table 4 show SSD results. In this case, good results need higher dimensional latent space (about 100 for both databases) than in the SI case. This is expected, because in SSD case, almost all variational parameters have been learned already in the training phase, therefore, slightly over-fitting the training data is needed. While in SI case, new variability (*e.g.* new speaker) is presented in testing, therefore, smoothing the projection function is required.

Table 1. Subject independent (**SI**) results on **OuluVs** database

	${}_{1\times1}\mathbf{LBP}^{u_2}_{1-8\times8}$		${}_{1\times2}\mathbf{LBP}^{u_2}_{1-8\times8}$	
m	**SVM**	**RfC**	**SVM**	**RfC**
10	58.28	55.15	57.18	54.53
15	61.09	62.18	**62.18**	**58.59**
20	60.93	60.46	54.68	57.65
25	**61.56**	**62.34**	56.09	57.50
30	59.06	61.56	55.93	58.28
40	55.62	59.37	56.71	**58.91**
50	58.75	60.46	56.87	58.75

Table 2. Subject independent (**SI**) on **AVLetters** database

	${}_{3\times4}\mathbf{LBP}^{u_2}_{1-3\times8}$		$\mathbf{LBP}^{u_2}_{1-8\times8}$	
m	**SVM**	**RfC**	**SVM**	**RfC**
10	32.44	33.46	28.85	29.23
15	38.46	34.87	29.74	32.31
20	41.79	38.85	30.38	33.85
25	**42.69**	**39.87**	28.97	33.59
30	40.77	41.03	31.92	37.82
40	38.33	**42.82**	29.87	**39.36**
50	37.69	41.67	33.08	36.03

Table 3. Subject semi-dependent (**SSD**) on **OuluVs** database.

	${}_{1\times1}\mathbf{LBP}^{u_2}_{1-8\times8}$		${}_{1\times2}\mathbf{LBP}^{u_2}_{1-8\times8}$	
m	**SVM**	**RfC**	**SVM**	**RfC**
90	84.68	**83.90**	81.25	**81.56**
100	**84.84**	83.75	**81.87**	81.56
130	84.22	83.75	81.71	81.56
150	84.37	83.75	81.56	81.56
180	84.06	83.75	81.71	81.56
200	84.21	83.75	**82.03**	81.56
220	83.90	83.75	81.40	81.56
250	83.59	83.75	81.71	81.56

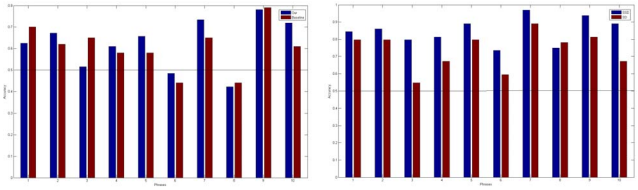
Table 4. Subject semi-dependent (**SSD**) on **AVLetters** database

	${}_{3\times4}\mathbf{LBP}^{u_2}_{1-3\times8}$		$\mathbf{LBP}^{u_2}_{1-8\times8}$	
m	**SVM**	**RfC**	**SVM**	**RfC**
80	64.36	62.56	62.31	61.92
90	64.10	63.59	63.08	62.56
100	64.23	63.85	62.31	62.18
130	**65.64**	64.87	62.44	61.79
150	65.38	64.49	62.44	61.67
180	65.00	64.10	61.67	61.79
200	64.87	64.10	62.31	61.79
220	65.00	64.10	62.05	61.79
250	64.74	64.10	62.44	61.79

Table 5 shows that our framework outperforms the baseline for SSD and compete for SI setting. The third column in Table 5 refers to the results of [28], a recent extension to [27]. The results for [28] are based on what is called *n*ormalized and clean version of OuluVs, while we use the *n*oisy version of OuluVs. Even though, we can compete in the recognition rate. Moreover, the most practical settings SSD is not presented in this paper. In addition, Figure 3 shows more results for OuluVs dataset. Figure 3(**a**) shows per-phrase comparison between our results and the results reported in [27], for SI settings. While Figure 3(**b**) shows per-phrase comparison between our framewrok performance in both SSD and SD settings.

Table 5. Comparative for **OuluVs** database.

	Ours	[27]	[28]
SI	62.34	62.4	**70.6**
SSD	**84.84**	64.2	na
SD	73.59	na	**85.1**



(**a**) (**b**)

Figure 3. On **OuluVs**: (**a**) comparing **SI** results for our approach (blue) and approach used in [27] (red) . (**b**) comparison between **SSD** results (blue) and **SD** results (red) of our approach.

Table 6. Comparative results for **AVLetters** database.

	Ours	[27]	[14]
SI	42.83	**43.46**	na
SSD (third fold)	**64.23**	58.82	57.3
SSD (total)	**65.26**	62.82	44.6

Table 6 shows comparison between our results for AVLetters database and the results in [27] and [14]. In this dataset, even though the confusion among the letters clips is high, our approach outperform both approaches, specially in the SSD setting.

7.3. Speaker recognition:

The goal in this experiment is to find the speaker within the register set of users. The challenge is to find the speaker from the limited available information in the mouth area. Moreover, we want to prove that although the manifold parameterization encodes mainly the geometric deformation from the unified manifold to the original data manifold, parameterization also hold speaker-related information. The testing protocol used here is the same as in SSD setting, since we take one repetition out for testing, and we train over all other repetitions. In both databases, we use the same configuration ($\mathbf{LBP}^{u_2}_{1-8\times8}$), and the results in both datasets is about 100% regardless of the dimension latent space. That was expected for two reasons: first, we have limited number of speaker (10 in AVLetters and 16 in OuluVs). Second, since we use solely visual informa-

tion, then the variability due to different speakers is significantly dominating the variability of speech, as shown in Figure 1(**a**).

8. Conclusion

We proposed a framework that utilized the homeomorphic manifold analysis and KPLS for manifold classification. We tackled two related classification problems speaker identification and speech recognition. We use supervised latent low-dimensional space embedding for solving the simultaneous multi-factor classification problem. We presented three different configurations of lipreading speaker independent, speaker semi-dependent and speaker dependent. The results show that our approach outperform in the semi-dependent setting which we consider the most realistic configuration and perform well in the other two settings.

Acknowledgments: This work was partly supported by the National Science Foundation award number 0923658. This work was also partly supported by the Office of Navel Research grant N00014-12-1-0755.

References

[1] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[2] K. Bennett and M. Embrechts. An optimization perspective on kernel partial least squares regression. *Nato Science Series, Sub-Series III: computer and System Sciences*, 190:227–249, 2003.

[3] S. Cox, R. Harvey, and Y. Lan. The challenge of multispeaker lip-reading. *International Conference on Auditory-Visual Speech Processing*, 2008.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001.

[5] A. Elgammal and C. Lee. Separating style and content on a nonlinear manifold. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:478–485, 2004.

[6] Y. Fu and X. Zhou. Lipreading by locality discriminant graph. *IEEE International Conference on Image Processing*, pages 325–328, 2007.

[7] J. Ham, L. Daniel, and L. Saul. Semisupervised alignment of manifolds. *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, 2005.

[8] T. Hazen, K. Saenko, C.-h. La, and J. Glass. A segment-based audio-visual speech recognizer: Data collection, development, and initial experiments. *Proceedings of the 6th international conference on Multimodal interfaces. ACM*, 2004.

[9] G. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 1970.

[10] B. Lee and et al. AVICAR: Audio-visual speech corpus in a car environment. *Proc. Int. Conf. Spoken Lang. Process*, 2004.

[11] C. Lee and A. Elgammal. Homeomorphic manifold analysis: Learning decomposable generative models for human motion analysis. *IEEE International Conference on Computer Vision*, 2005.

[12] P. J. Lewi. Pattern recognition, reflections from a chemometric point of view. *Chemometrics and Intelligent Laboratory Systems*, 28(1):23–33, Apr. 1995.

[13] J. Luettin, N. Thacker, and S. Beet. Speaker identification by lipreading. *International Conference on Spoken Language Processing*, pages 1–4, 1996.

[14] I. Matthews and T. Cootes. Extraction of visual features for lipreading. *PAMI*, 24(2):198–213, 2002.

[15] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264(23 December):746–748, 1976.

[16] T. Ojala. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002.

[17] E. Patterson and S. Gurbuz. Moving-talker, speaker-independent feature study, and baseline results using the CUAVE multimodal speech corpus. *EURASIP Journal on Appl. Signal Process.*, 2002(1110-8657):1189–1201, 2002.

[18] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, pages 1481–1497, 1990.

[19] G. Potamianos and C. Neti. Audio-visual automatic speech recognition: An overview. *Issues in Visual and Audio-Visual Speech Processing*, 2004.

[20] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[21] R. Rosipal and L. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *The Journal of Machine Learning Research*, 2:97–123, 2002.

[22] K. Saenko and K. Livescu. Visual speech recognition with loosely synchronized feature streams. *IEEE International Conference on Computer Vision*, 2005.

[23] C. Sanderson and K. Paliwal. Identity verification using speech and face information. *Digital Signal Processing*, 14(5):449–480, Sept. 2004.

[24] A. Shaikh, D. Kumar, and W. Yau. Lip Reading using Optical Flow and Support Vector Machines. *IEEE International Congress on Image and Signal Processing*, 1:327–330, Oct. 2010.

[25] D. Shiell and L. Terry. Audio-Visual and Visual-Only Speech and Speaker Recognition: Issues about Theory, System Design, and Implementation. *Visual speech recognition: lip segmentation and mapping*, pages 1–38, 2009.

[26] H. Wold. Soft Modeling by Latent Variables; the Nonlinear Iterative Partial Least Squares Approach. *Perspectives in Probability and Statistics. Papers in Honour of M. S. Bartlett*, pages 520 – 540, 1975.

[27] G. Zhao. Lipreading with local spatiotemporal descriptors. *IEEE Transactions on Multimedia*, pages 1–11, 2009.

[28] Z. Zhou, G. Zhao, and M. Pietikainen. Towards a practical lipreading system. *Computer Vision and Pattern Recognition*, 2011.