Unsupervised Random Forest Manifold Alignment for Lipreading

Yuru Pei
Peking University
peiyuru@cis.pku.edu.cn

Tae-Kyun Kim
Imperial College London
tk.kim@imperial.ac.uk

Hongbin Zha
Peking University
zha@cis.pku.edu.cn

Abstract

Lipreading from visual channels remains a challenging topic considering the various speaking characteristics. In this paper, we address an efficient lipreading approach by investigating the unsupervised random forest manifold alignment (RFMA). The density random forest is employed to estimate affinity of patch trajectories in speaking facial videos. We propose novel criteria for node splitting to avoid the rank-deficiency in learning density forests. By virtue of the hierarchical structure of random forests, the trajectory affinities are measured efficiently, which are used to find embeddings of the speaking video clips by a graph-based algorithm. Lipreading is formulated as matching between manifolds of query and reference video clips. We employ the manifold alignment technique for matching, where the L_∞-norm-based manifold-to-manifold distance is proposed to find the matching pairs. We apply this random forest manifold alignment technique to various video data sets captured by consumer cameras. The experiments demonstrate that lipreading can be performed effectively, and outperform state-of-the-arts.

1. Introduction

Automatic lipreading plays an important role in communications in noisy environments, e.g. in stadiums and bars where noises overcome speaking signals. The lipreading is traditionally viewed as a supplement to speech recognition [18]. In recent years, more researches are put on lipreading solely from visual channels. Lipreading on a predefined phrase data set has been demonstrated to be effective [1, 5, 13, 15, 24, 25, 26] in speaker dependent and independents scenarios. However, the robust lipreading from visual channels still faces challenges in the following three aspects. First, it is generally unrealistic or uncomfortable to collect large enough stylized data from one subject by asking him or her to repeat the phrases many times. The situation becomes worse when it comes to the lipreading of a set of subjects. Considering the variations of lip shapes and styles related to speaking speeds and intensities, it is often hard to collect a data set covering all possible lip motions. The generalization capacity is desirable in the lipreading tasks. Second, the lipreading system should be real-time considering the communication requirements. Third, the current lipreading systems based on ordinary videos suffer from illumination and texture variations. It is relatively hard to achieve accurate 3D lip motions from monocular color videos.

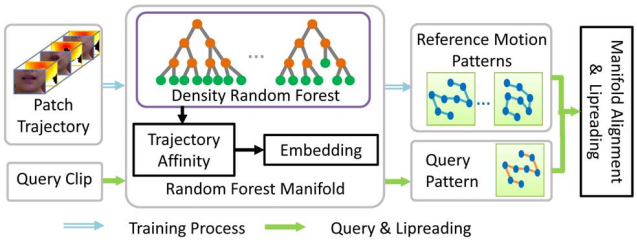

Figure 1. Flowchart of our system.

Confronted with these problems, we propose a novel framework to handle lipreading of phrases by the alignment of random forest manifolds as shown in Fig. 1. The lipreading is performed in the low-dimensional manifold instead of the original feature space, where the video clip represented by a set of patch trajectories is converted to a simplex motion pattern manifold. For the purpose of graph-based embedding, the affinity of patch trajectories are estimated by an unsupervised density random forest, which is known for fast training and online testing, and the generalization capacity. In our experiments, the depth videos are combined together with the color videos to deal with the illumination and texture variations.

Density random forest and rank-deficiency. The random forest often works in a supervised manner, which is trained with labeled data to get a reasonable partition of visual words for image categorization [14], pose estimation [20], and object detection [9]. Different from the supervised random forests, our random forest manifold framework works in an unsupervised manner. The underlying data distribution and affinity are estimated without prior labeling. By introducing a dummy set, a classification random forest was used to cluster unlabeled data set [2]. The

optimal node splitting can also be determined by finding a feature pair to produce the largest variance of feature difference [23]. Recently, the density forest was introduced under a Gaussian distribution assumption in tree nodes [6]. The determinant of the covariance matrix was used to measure the clustering compactness in node splitting. However, the rank-deficiency problem usually exists in the covariance matrices of the high dimensional data set. In that case, the zero-valued determinant can be no longer used to estimate the information gain. We address the rank-deficiency problem by proposing novel criteria in node splitting by combining the trace-based distribution measurement and a scatter index to estimate the optimal node splitting. To the best of our knowledge, this paper is the first attempt to handle the rank-deficiency problem in building unsupervised density forests from the high dimensional data.

Random forest manifold and lipreading. Considering the high efficiency of random forest, it has been used to find data embeddings. Gray et al. [10] employed a supervised classification random forest to derive the distance matrix of medical images and data embedding. Crimisi et al. [6] addressed the forest manifold of low dimensional toy data, where the Laplacian eigenmaps was employed to find the embedding from the affinity matrix produced by the density forest. In this work, a density forest is built to measure affinities of the patch trajectories. Combined with graph-based embedding, the video clips are embedded to low dimensional motion patterns. Once given a query phrase video, the extracted patch trajectories are fed to the random forest for a pairwise affinity matrix and embedding. The lipreading in this paper is formulated as the matching of motion pattern manifolds of the query and those of labeled references by alignments in the embedding space. The manifold pairs with the minimum distance are considered to share the same phrase label.

The main contribution of this paper is to propose novel criteria to handle the rank-deficiency problem in building density forests. We integrate the trace-based cluster compactness measurement together with a scatter index to estimate information gain during node splitting. By virtue of the unsupervised random forest manifold, the lipreading can be performed by matching simplex motion patterns effectively.

1.1. Related Work

Automatic lipreading has been studied in computer vision for several years [18]. Saenko et al. [19] employed the Hidden Markov Model to capture the dynamics of visual speech signals. Zhao et al. [24] proposed a spatiotemporal version of LBP features for lipreading. Aside from the lipreading in the original feature space, automatic lipreading in manifolds has also been investigated. Aharon and Kimmel [1] applied nonlinear dimension reduction techniques to analyze visual lip images. The embedding space with representative key images can be used for lipreading by matching uttering contours. Zhou et al. [25, 26] combined LBP-like features and embedding techniques for lipreading, where the visual feature vectors were mapped to deterministic curves. Different from the above lipreading in color videos solely, we integrate multimodal data and fusions of feature channels to improve the recognition performance, where the density forests and the manifold alignment are employed for an efficient lipreading system.

The manifold alignment provides an approach to establish correspondence between two embedding spaces. Lafon et al. [12] estimated an affine function from predefined landmarks for the correspondence in a diffusion embedding space. A manifold mapping can be defined by linear local maps of the tangent planes [11]. Wang et al. [22] built a mutual embedding space for manifold alignment, where the transformations were solved by eigenvector decomposition of the Laplacian matrix. An extended affine transformation for the non-holistic manifold alignment was proposed in [17]. In this work, the affine transformation is used to align the embedded motion patterns, where the L_∞-based manifold-to-manifold distance is proposed to measure similarities of embedded simplex motion patterns.

2. Patch Trajectory

The patch trajectories are extracted from speaking videos captured by consumer depth cameras (*Kinect*). We employ AAM [4] to track local facial features in the lower faces including lips and jaws from the color videos as shown in Fig. 2. Since the color and depth cameras in *Kinect* are calibrated in advance, the features in depth videos can be located synchronously. The small patches around the lips are extracted and result in patch trajectories.

The patch trajectory features include the trajectory shape s as the difference of patch positions in adjacent frames. The shape vector $s = (\triangle x_i | i = 2, ..., n)$, and $\triangle x_i = x_i - x_{i-1}$, where n is the length of one video clip. The shape vectors are normalized, and $\frac{s}{max||\triangle x_i||} \rightarrow s$.

The HOG feature [7] is known as the distribution of intensity gradients and edge orientations, which is employed to describe the local color and depth patches. Local binary pattern (LBP[16]) is a powerful features for texture classification, and combined with HOG as feature descriptors in our experiments. The shape and texture features are concatenated together to describe the patch trajectory t, and $t = (s, h)$, where h is the combined texture features of HOG and LBP with respect to the color and depth images.

The feature vector of the patch trajectory is relatively high-dimensional, i.e. $n_s + n_h \times n$, where n_s is the dimensionality of the trajectory shape vector s, n_h is the bin number of combined texture features. In this paper, the lipreading is performed in a low-dimensional embedding space for

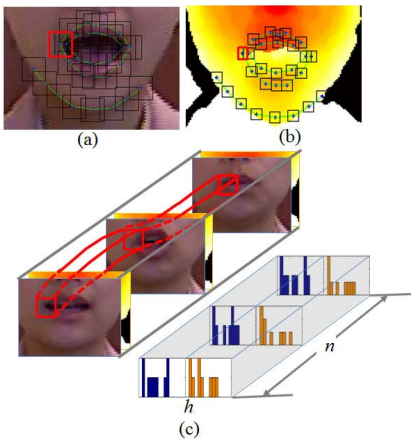

Figure 2. Patch trajectory extraction in a video clip. The green contours around lips and jaws are resulted from an AAM tracker. The patches extracted on color (a) and depth (b) images (black outlined). (c)The patch trajectory and the corresponding texture feature histogram h in the color and depth images.

efficiency instead of in the original feature space. We propose the random forest manifold to represent lip motions.

3. Random Forest Manifold

The random forest manifold technique integrates the unsupervised density forest for affinity estimation and graph-based embedding. The framework takes advantages of an efficient affinity estimation in both the training and query by hierarchical tree structures (See Section 3.2.1).

The density forest with a novel node splitting strategy is introduced to handle the rank-deficiency as described in Section 3.1. Given one data set, the random forest yields an affinity matrix as described in Section 3.2. The graph-based embedding algorithm is employed to find manifolds from the data affinities (Section 3.3).

3.1. Unsupervised Random Forest

Given the unlabeled data set $T = \{t_i | i = 1, ..., n_t\}$, a set of trees are trained independently. Density forest provides an unsupervised method to estimate the underlying data distribution [6] with a Gaussian distribution assumption in tree nodes. The differential of multi-variate Gaussian entropy H is defined by the determinant of the covariance matrix, which can be seen as the volume of the hyperellipsoid that bounds the uncertainty of the data distribution.

$$H = \ln\left((2\pi e)^{\kappa}|\sigma(T_j)|\right), \tag{1}$$

where σ is the covariance matrix of the κ-variant Gaussian distribution. T_j is the data set of the j-th node. $|\cdot|$ is the matrix determinant.

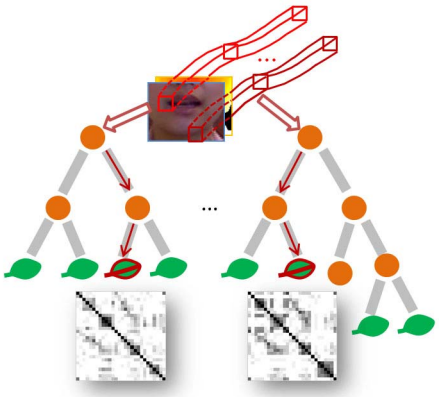

Figure 3. Density forest of patch trajectories. Each tree can yield an affinity matrix of patch trajectories.

The optimal node splitting parameters, including feature channel and random splitting threshold, are obtained by maximizing the information gain. The density forest tends to produce the compact local clusters. It works well for low dimensional toy data with a full-rank covariance matrix. However, when it comes to the rank-deficiency, the whole scheme fails. Unfortunately, it is always the case in the real data sets, where the high dimensionality and the possible intrinsic dependency of the sparse data make the rank of covariance matrix lower than both the instance number and dimensionality. It's deserved to note that the rank-deficiency is not only ubiquitous in high dimensional data, it can be observed in low dimensional data. As shown in Fig. 4, the *corner* data include point clouds on two planes, where the rank of the local data points can be of 2 instead of that of original covariance matrix of 3.

Confronted with this problem, we propose criteria for node splitting as an integration of a trace-based distribution measurement I_1 and a scatter index I_2.

$$I_1 = -\sum_{i=l,r} \frac{m_{T_j^i}}{m_{T_j}} \ln\left(tr\left(\sigma(T_j^i)\right)\right), \tag{2}$$

where $tr(\cdot)$ is the matrix trace, and $m_{T_j^i}$ denotes the size of left or the right children nodes. As described in [21], generally the trace is not a good metric for covariance matrices for lack of invariance to scales and sensitiveness to the parameter units. In our experiments, all shape and texture feature vectors are normalized to avoid the scale problems.

The scatter index I_2 is given as

$$I_2 = \frac{\|\mu_l - \mu_r\|_\infty}{\sum_{i=l,r} \phi(T_j^i, \mu_i)}, \tag{3}$$

where $\phi(T_j^i, \mu_i) = \max_{t \in T_j^i} \|t - \mu_i\|_\infty$. μ_i is the centroids of the children nodes. Note that I_2 is maximized when the data inside each child node are close to the centroid and the

centroids of two children nodes are apart. The binary test φ in node splitting is defined as

$$\varphi(t_u, t_v, \tau)_c = \begin{cases} 1, \text{ if } (t_u - t_v)_c < \tau, \\ 0, \text{ otherwise.} \end{cases} \quad (4)$$

u and v are randomly selected frame indices in the trajectory. $(t_u - t_v)_c$ is the feature difference between the u-th and the v-th frame in feature channel c. In each node splitting, the randomly selected $(u, v, \tau)_c$ splits the data set into two parts. The objective function I is defined as a combination of I_1 and I_2.

$$I(u, v, \tau)_c = I_1 + \lambda I_2. \quad (5)$$

The parameter set $(u, v, \tau)_c$ that maximizes the objective function I is selected for node splitting. The constant λ is empirically set at 50 in all the experiments.

The tree growth is terminated when the point number inside one node is below a predefined threshold or reaches the specific depth.

3.2. Affinity

The forest leaves L define a partition of the training data. When feeding an instance t to one tree, it will finally reach a leaf $\ell(t)$, and $\ell \in L$, after a sequence of binary tests. In case two instances reach the same leaf node, they are assumed to be similar with respect to the tree, where the affinity between the two instances is added by 1, and 0 otherwise. The symmetric affinity matrix $\mathcal{A}$ is obtained in this way (Fig. 3). With an ensemble of density trees, the forest tends to yield a generalized affinity of the data set. The final affinity $\mathcal{A}$ is defined as a weighted combination of $\mathcal{A}_k$ from independent trees.

$$\mathcal{A} = \frac{1}{n_T} \sum_{k=1}^{n_T} \mathcal{A}_k, \quad (6)$$

where n_T is the tree number. Since only points inside one leaf node cluster are considered to be similar, the affinity matrix from the random forest automatically possesses the local neighborship. Thus, $\mathcal{A}$ can be viewed as a geodesic affinity (distance) matrix of the original data set. On the contrary, when using L_2 distance metric, there is no prior on local neighbourship. The kNN-like algorithm is needed to find local neighbourship from the pairwise distance matrix with additional time costs.

3.2.1 Random Forest vs kNN

Compared with the ordinary L_2 norm and kNN for the affinity, our method can greatly reduce the time cost. The time complexity of the kNN-graph algorithm is $O(\rho N^{\alpha})$ [3] for ρ-dimensional data, and $\alpha \in (1, 2)$. N is the instance number. The empirical α can be $1.11 - 1.14$ [8]. In our work, the time complexity of the forest traversal for the affinity is $O(N\nu n_T)$, and the tree depth $\nu = log_2(N/n_l)$. n_l is the leaf size. For a moderate-scale training data set, $\nu \in [5, 9]$. The tree number n_T is set empirically at 17 in all our experiments. More importantly, the time complexity of our method has no relations with the dimensionality, which is desirable for the high dimensional data (see Section 5.4).

It's deserved to note that there is no pairwise distance computation in our method, where the comparison to the thresholds when traversing trees is very fast and is negligible in time. However, in kNN-graph-based method, although the cost of pairwise distance of small subsets or sampled point pairs is much smaller than the dense pairwise distance computation of the original data set, it is still time-demanding considering the complex operations implied, e.g. L_2 and earth mover distance.

3.3. Embedding

The density tree derives the affinity matrix and the neighboring relationship of the data set simultaneously. The multidimensional scaling(MDS) algorithm is employed for the embedding by minimizing the stress function. The original data set is embedded to a low dimensional space in an Isomap-like manner. We apply the random forest manifold method to a set of toy data as shown in Fig. 4. The embeddings based on the random forest affinity is similar to those by the L_2 norm and kNN, while the latter is more time-consuming.

By virtue of the random forest manifold embedding, the original video clip consisting of a set of patch trajectories is converted to a simplex pattern in a low dimensional space, where each point is corresponding to a patch trajectory.

4. Lipreading

In the phrase lipreading scenarios, there is a predefined reference phrase corpus. The lipreading is performed by finding the most similar reference clip and assigning its label to the query clip.

In our experiments, the patch trajectories from the predefined phrase data set serve as the training data for the random forest. Once given the affinity matrix of trajectories, the reference phrases can be embedded to a low dimensional motion pattern set $\Theta = \{P_r\}$. Given the query video clips, the corresponding motion pattern manifold P_q is computed by the proposed method. The lipreading is performed in the embedding space by searching a reference motion pattern P_r that best matches the query P_q.

$$P_r^* = \arg\min_{P_r \in \Theta} d_m(P_r, P_q). \quad (7)$$

$d_m(\cdot)$ is the manifold-to-manifold distance as described in Section 4.1. The embedding preserves the geodesic affinity (distance) of the original data set. However, there is no guarantee that two motion pattern manifolds share the same spatial configuration. For instance, two manifolds can

be of different scales. The direct manifold distance computation by the distance sum of closest point pairs is nonsense. We employ the manifold alignment technique [17] to estimate the pattern correspondence and the manifold-to-manifold distance.

4.1. Manifold Alignment

The goal of manifold alignment is to transform the reference and the query motion patterns, P_r and P_q, to a mutual embedding space.

In case the query and reference video clips share the same patch extraction configurations, the correspondence between the two manifolds are known in advance. Otherwise, the feature correspondence can be obtained with the help of local structures. Each point in the motion pattern is corresponding to a patch trajectory, and the local shape descriptor b_i of the i-th point is defined as $b_i = \{d_{i,j}\}$, where entry $d_{i,j}$ is distance between the i-th point of the motion pattern with its j-th nearest neighbor. The local corresponding Ψ is obtained by locating similar local structures , and

$$\Psi = \{(i^*, j^*) = \arg\min_{i,j} \|b_i^r - b_j^q\|\}, \tag{8}$$

where b_i^r and b_j^q are local shape descriptors of the reference and query motion patterns respectively.

The motion patterns are normalized beforehand to avoid the scale variations across embedding spaces. An affine transformation M is estimated by the least square method based on the local feature matchings, and $P_\Psi^r \approx M \cdot P_\Psi^q$, where M is a $(m_r) \times (m_q + 1)$ dimensional affine transformation matrix. m_r and m_q are the dimensions of the reference and the query motion patterns. Note that there are no requirements that P_r and P_q bear the same dimensionality. The distance d_m between the P_r and P_q is defined as the distance sum of closest point pairs after the transformation of the query pattern P_q.

$$d_m(P_r,\, P_q) = \min \sum_{\substack{i=1,\ p_{r,\,i'} \in P_r \\ p_{q,\,i} \in M \cdot P_q}}^{n_q} \|p_{r,i'} - p_{q,i}\|_\infty, \tag{9}$$

where n_q is the point number of P_q. $p_{r,i'}$ is $p_{q,i}$'s counterpart in P_r. The phrase label of the reference motion pattern with the minimum distance to P_q is assigned to the query clip for lipreading. The dimensionality selection is performed to find the optimal matching pairs. We test a set of manifold matchings with different dimensionalities from 3 to 20. For a query video, the dimensionality with the smallest matching distance is selected.

5. Experiments

5.1. Data Set

We performed the experiments on the following visual uttering data sets, where the first two, KinectVS, and OULUVS, are phrases data sets, and the next two, AVLetters, AVLetters2, are letters data sets, and the last one is a digits data set. In our experiments, 60% extracted trajectories are used to learn the density random forest, and the remaining portion for testing.

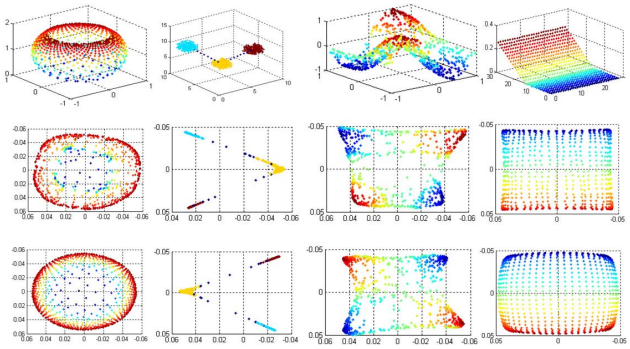

Figure 4. The random forest manifold embeddings of toy data sets. From left to right: *punctured spheres*, *3D clusters*, *twin peaks*, and *corner*. The upper row is the original data set. The middle row is the embedding by our method, while the lower is the embedding based on the L_2 norm.

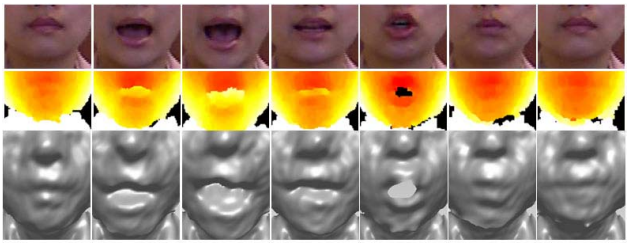

Figure 5. Samples color and depth image sequences of phrase *Hello*. The first row is the color images. The second row is the corresponding depth images. The third row is the 3D meshes related to the depth images.

KinectVS consists 20 subjects uttering 20 phrases (Table 1) six times. The color and depth video data are both captured by *Kinect* at a resolution of 640×480 (Fig. 5). The Laplacian smoothing is applied to the depth images in the preprocessing to remove the device noises.

OULUVS [24] consists 20 subjects uttering 10 phrases (Table 1) five times at a resolution of 720×576.

AVletters [13] has 10 subjects uttering letters A to Z three times. The data set contains pre-cropped lip images of 80×60.

AVLetters2 [5] consists color videos of 5 subjects uttering letters A to Z seven times with a resolution of 1920×1080.

CUAVEsam [1] has two sample videos of digit 0 to 9 at a resolution of 720×480. We use the first eight repetitions with frontal faces.

Table 1. Twenty daily used phrases.

1st part(1-10) ([24])	2nd part (11-20)
Hello; Excuse me; I am sorry; Thank you; Good bye; See you; Nice to meet you; You are welcome; How are you; Have a good time	*Who's calling; Time is up; I agree; I love this game; So far so good; Any thing else; Whats up; So do I; Be careful; Bottoms up*

Table 2. Lipreading performances in subject dependent experiments when using different criteria in node splitting, i.e. trace-based cluster compactness I_1, scatter index I_2, and I.

Criterion	I_1	I_2	I
KinectVS	83.5	78.4	**94.1**
OULUVS	89.2	76.3	**97.3**
AVLetters	61.2	53.2	**69.6**
AVLetters2	87.7	72.2	**91.8**
CUAVEsam	91.5	93.5	**100**

5.2. Subject Dependent (SD)

In the subject dependent experiments, the training and the testing data are from the same set of subjects. As described in Section 3.1, the criterion for the node splitting is a combination of a trace-based cluster compactness I_1 and a scatter index I_2. We compare the fusion with those using I_1, I_2 solely as shown in Table 2. The fused version is better than using each criterion alone. We think the reason is that the scatter index can enhance the uncertainty measurement of the data distribution.

The patch trajectory is composed of multi-feature channels, i.e. trajectory shapes, textures of color and depth image patches. When building random forests, the feature channels and the corresponding thresholds are selected randomly, and then optimized. All feature channels are fused without extra work. The lipreading results based on the shapes (RFMA$_{shape}$), the color (RFMA$_{HOG+LBP(color)}$) and depth patches (RFMA$_{HOG(depth)}$) solely and integration of all feature channels together (RFMA$_{fusion}$) are shown in Table 3. In KinectVS data set, the fusion of all feature channels outperforms those using single modal data.

We have compared our method with the recent lipreading works [24, 25, 26] on OULUVS data set, where the features of the patch trajectories are only extracted from color videos. Table 3 shows accuracies with various feature channels combinations. The fusion of the color patch features (HOG and LBP) and the trajectories shape (RFMA$_{fusion}$) outperformed the reported state-of-the-arts. We have applied our methods to letter sets (Table 4). The performances in the high resolution AVLetters2 is better than that in the low-resolution AVLetters set, and our method produces the higher scores in both data sets.

[1] http://www.clemson.edu/ces/speech/cuave.htm

As we can see, the image resolution has a close relation to the performance (Table 4). The experiments on the low-resolution KinectVS data set (with approx. 45×25 lip regions) can produce a comparable results to those on the OULUVS data set (with approx. 135×80 lip regions) with the help of the depth videos (Table 3). It is a promising way to achieve a powerful lipreading system by virtue of multi-modal data.

5.3. Subject Independent (SI)

In the subject independent experiments, the training and the query data are from different subjects. We employ the leave-one-out strategy, where the patch trajectories from one subject are removed, and the remaining data are used as the training data for the random forest manifolds.

In the data preprocessing, we registered all the faces to avoid the scale problem. For instance, the lip and jaw regions of one subject may be larger than other guys, and the shape vectors would be apparently different from others. As shown in Table 3, the accuracy of subject independent experiment is lower than that of SD experiments. Almost all automatic lipreading literatures reported this problem [1, 24, 26], which comes from the personal characteristics during speaking, and some person-specific texture difference caused by moustache, skin color, lip and teeth shapes. Similar to the SD experiments, the fusion of all feature channels of our method produces an improvement to [24, 26].

5.4. Time Cost for Affinity

Our method can greatly reduce the time cost in affinity estimation (see Fig. 6). We compare the time costs for affinity matrices of 3D *corner* data and 2730-dimensional patch trajectories data by the proposed random forest and the kNN graph-based method [3]. For all varying sizes of data sets tested, our method is faster. Since the forest traversal is extremely fast and has no relations to the data dimensionality, our method is especially superior to the kNN for high dimensional and large data sets.

5.5. Parameter Analysis

Patch Size. In the patch trajectory extraction, a small image region around the salient marker is extracted, and concatenated together as a patch trajectory. It is interesting to note that, the optimal patch size of the color and depth videos are different. As shown in Fig. 7, the video patch size of KinectVS is set at 15×15, while the patch size for depth video is set at 7×7. We think it is due to the noise in the depth images around the trajectories. The larger patch size, the more noise accumulations with an impairment to accuracies. However, in the color video, a comparatively large patch size can hold more texture information. The patch size depends on the resolution and quality of videos.

Table 3. Lipreading performances of subject dependent (SD) and subject independent (SI) experiments on phrase data sets, KinectVS and OULUVS.(† via manual feature extraction)

Data Set	Features	Auto. Tracking	
		SD	SI
KinectVS	RFMA$_{shape}$	72.4	43.5
	RFMA$_{HOG+LBP(color)}$	91.5	82.1
	RFMA$_{HOG(depth)}$	79.4	57.5
	RFMA$_{fushion}$	**94.1**	**87.7**
OULUVS	RFMA$_{HOG}$	89.1	83.4
	RFMA$_{HOG+LBP}$	93.6	86.4
	RFMA$_{fusion}$	**97.3**	**89.7**
	Deterministic Curve [26]	85.1(96.5†)	81.3
	Graph Embedding [25]	n/a(90.7†)	n/a
	Local Spatiotemporal descriptors [24]	64.2(70.2†)	58.6

Table 4. Lipreading performance on letter data sets.

Data Set	Features	Accuracy
AVLetters	RFMA$_{HOG}$	56.3
	RFMA$_{HOG+LBP}$	62.5
	RFMA$_{fushion}$	**69.6**
	Deep Autoencoder [15]	64.4
	Multiscale Spatial Analysis [13]	44.6
	Local Spatiotemporal descriptors [24]	58.85
AVLetters2	RFMA$_{fusion}$	**91.8**
	AAM & Sieves[5]	89.4

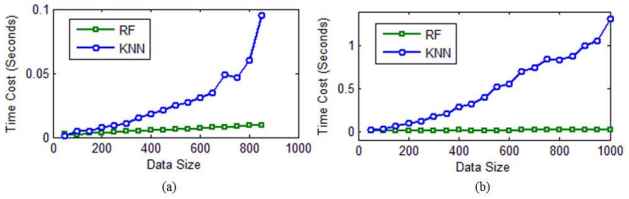

Figure 6. The time cost in computation of affinity matrices by kNN [3] and our random forest of 3D *corner* data (a) and 2730-dimensional patch trajectories (b) of different sizes.

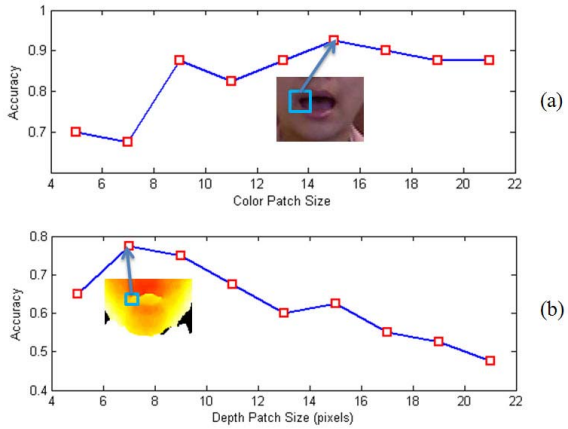

Figure 7. Accuracy variations with different patch sizes in color (a) and depth (b) videos.

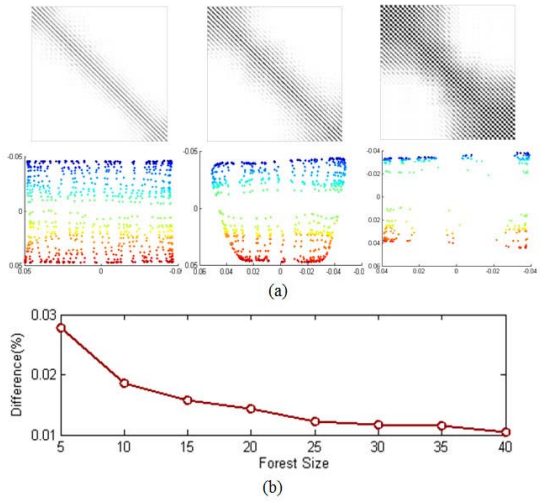

Figure 8. The affinity matrix and the embedding of *corner* data with different leaf sizes of 20, 60 and 100 (from left to right). (b) The difference e between the affinity matrices by our random forest and L_2 norm with different forest sizes in *corner* data.

Leaf Size. One termination condition of tree growth is the minimum leaf size threshold n_l. The nodes stop splitting when the number of points inside reaches the predefined threshold. The larger leaf size, the smaller tree will be built. On the contrary, the small leaf size, e.g. one point in the leaf node, will lead to a very deep tree with n_t leaf nodes and $\log_2 n_t$ levels for a balanced tree. With various leaf size thresholds, the forest tends to yield different affinity matrices. As shown in Fig. 8(a), the small leaf size threshold yields a sparser matrix than that of the large leaf size. It can be explained by the affinity estimation described in Section 3.2. In the tree with a small leaf size, the clusters corresponding to the leaf node could be compact, and the probability of two instances reaching the same leaf node becomes small. In the lipreading experiments, the accuracy reaches a local maximum when n_l is set at 30.

It's deserved to node that, there is one danger with a large leaf size where the points inside the same leaf node are not similar, which could impair the embedding (Fig. 8 (a)).

Forest Size. It is believed that in the random forest the more trees, the more accurate fitting to the original data distribution and affinity estimation. However, the comparatively large computation cost is introduced both in the training and testing processes with increasing forest size. We measure the difference e between the affinity matrices, $\mathcal{A}$ and $\mathcal{A}_{L_2}$, computed by our random forests and the L_2 norm followed by kNN.

$$e = \frac{\|(\mathcal{A} \oplus \mathcal{A}_{L_2})\|_F^2}{n_{\mathcal{A}}}, \quad (10)$$

where $\oplus$ is the *xor* of matrix entries. $n_{\mathcal{A}}$ is the size of $\mathcal{A}$. $\|\cdot\|_F$ is Frobenius norm. We test the $corner$ data with different forest sizes as shown in Fig. 8(b). When the tree number is more than 20, the difference is below 0.015. The difference e decreases when enlarging the forest size. The accuracy reaches a local maximum when the forest size is 17 in lipreading experiments.

6. Conclusions

We have presented a random forest manifold technique and applied it to lipreading in color and depth videos. The video clips represented as a set of patch trajectories are converted to simplex motion patterns in the embedding space. The lipreading is realized by motion pattern matching based on the manifold alignment. The whole process is unsupervised, where the proposed criteria can deal with the rank-deficiency in building density forests. Our framework takes advantage of the efficient training and testing of random forest, especially for affinity estimation, together with the unsupervised manifold distance estimation by the manifold alignment. The proposed method can handle large data set efficiently, and at the same time can perform lipreading in relatively low-resolution videos effectively.

Acknowledgement

This work was supported by NSFC 61272342, NHTRDP 863 2012AA011602, NBRPC 973 2011CB302202.

References

[1] M. Aharon and R. Kimmel. Representation analysis and synthesis of lip images using dimensionality reduction. *IJCV*, 67(3):297–312, 2006.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] J. Chen, H.-r. Fang, and Y. Saad. Fast approximate k nn graph construction for high dimensional data via recursive lanczos bisection. *The Journal of Machine Learning Research*, 10:1989–2012, 2009.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. on PAMI*, 23(6):681–685, 2001.

[5] 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.

[6] A. Criminisi. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2-3):81–227, 2011.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.

[8] W. Dong, C. Moses, and K. Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*, pages 577–586. ACM, 2011.

[9] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Trans. on PAMI*, 33(11):2188–2202, 2011.

[10] K. Gray, P. Aljabar, R. Heckemann, A. Hammers, and D. Rueckert. Random forest-based manifold learning for classification of imaging data in dementia. *Machine Learning in Medical Imaging*, pages 159–166, 2011.

[11] J. Ham, D. Lee, and L. Saul. Semisupervised Alignment of Manifolds. In *Proc. of the Tenth Int'l Workshop on Artificial Intelligence and Statistics*, pages 120–127, 2005.

[12] S. Lafon, Y. Keller, and R. Coifman. Data Fusion and Multicue Data Matching by Diffusion Maps. *IEEE Trans. on PAMI*, 28(11):1784–1797, 2006.

[13] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Trans. on PAMI*, 24(2):198–213, 2002.

[14] F. Moosmann, B. Triggs, F. Jurie, et al. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS'06*, pages 985–992, 2006.

[15] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011.

[16] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[17] Y. Pei, F. Huang, F. Shi, and H. Zha. Unsupervised image matching based on manifold alignment. *IEEE Trans. on PAMI*, 34(8):1658–1664, 2012.

[18] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proc. the IEEE*, 91(9):1306–1326, 2003.

[19] 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.

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conf. on CVPR*, volume 2, page 7, 2011.

[21] R. Sim and N. Roy. Global a-optimal robot exploration in slam. In *IEEE Conf. on ICRA*, pages 661–666, 2005.

[22] C. Wang and S. Mahadevan. Manifold Alignment without Correspondence. In *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI)*, pages 1273–1278, 2009.

[23] G. Yu, J. Yuan, and Z. Liu. Unsupervised random forest indexing for fast action search. In *IEEE Conf. on CVPR*, pages 865–872, 2011.

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

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

[26] Z. Zhou, G. Zhao, and M. Pietikainen. Towards a practical lipreading system. In *IEEE Conf. on CVPR*, pages 137–144, 2011.