International Journal of Computer Vision 67(3), 297–312, 2006

DOI: 10.1007/s11263-006-5166-3

Representation Analysis and Synthesis of Lip Images Using Dimensionality Reduction

MICHAL AHARON AND RON KIMMEL
Department of Computer Science, Technion—Israel Institute of Technology, Technion City, Haifa 32000, Israel
michalo@cs.technion.ac.il

Received June 11, 2004; Revised February 11, 2005; Accepted August 17, 2005

First online version published in March, 2006

Abstract. Understanding facial expressions in image sequences is an easy task for humans. Some of us are capable of lipreading by interpreting the motion of the mouth. Automatic lipreading by a computer is a challenging task, with so far limited success. The inverse problem of synthesizing real looking lip movements is also highly non-trivial. Today, the technology to automatically generate an image series that imitates natural postures is far from perfect.

We introduce a new framework for facial image representation, analysis and synthesis, in which we focus just on the lower half of the face, specifically the mouth. It includes interpretation and classification of facial expressions and visual speech recognition, as well as a synthesis procedure of facial expressions that yields natural looking mouth movements.

Our image analysis and synthesis processes are based on a parametrization of the mouth configuration set of images. These images are represented as points on a two-dimensional flat manifold that enables us to efficiently define the pronunciation of each word and thereby analyze or synthesize the motion of the lips. We present some examples of automatic lips motion synthesis and lipreading, and propose a generalization of our solution to the problem of lipreading different subjects.

Keywords: automatic lipreading, image sequence processing, speech synthesis, multidimensional scaling, dimension reduction, locally linear embedding

1. Introduction

Automatic understanding and synthesizing of facial movements during speech is a complex task that has been intensively investigated (Bregler et al., 1993; Vanroose et al., 2002; Li et al., 1997; Bregler et al., 1998; Bregler and Omohundro, 1994; Bregler et al., 1997; Kalberer and Van Gool, 2001; Luettin, 1997). Improving the technology in this area may be useful for various applications such as better voice and speech recognition, as well as comprehension of speech in the absence of sound, also known as lipreading. At the other end, generating smooth movements may enhance the animation abilities in, for example, low bit-rate communication devices such as video conference transmission over cellular networks.

In this paper we introduce a framework that handles frontal view facial images, and is capable of representing, synthesizing, and analyzing sequences of facial movements. Our input is a set of frontal facial images. These images are extracted from training sequences of a single person (the model), that pronounces known syllables. The ascription of the images to their specific syllable is important, and is used during the synthesis process.

The images are first automatically aligned with respect to the location of the nose. Every two images are compared and a symmetric dissimilarity matrix is computed. Next, the images are mapped onto a plane, so that each image is represented as a point, while trying to maintain the dissimilarities between images. That is, the Euclidean distance between each two points on the plane should be as close as possible to the dissimilarity between the two corresponding images. We justify this flat embedding operation by measuring the relatively small error introduced by this process.

Next, the faces representation plane is uniformly sampled and 'representative key images' are chosen. Synthesis can now be performed by concatenating the different sequences of images that are responsible for creating the sound, while smoothing the connection between each two sequential sequences.

Using the 'representative key images', the coordinates of new mouth images can be located on the map. Each word, which is actually a sequence of mouth images, can now be considered as a contour, given by an ordered list of its coordinates. Analysis of a new word is done by comparison of its contour to those of already known words, and selecting the closest as the best match.

Again, in our experiments, all training sequences and their corresponding mapping process were done with a single subject facial images. Nevertheless, we show that the same concept can be generalized with some success to lipreading of different subjects, by exploiting the fact that the sequence of pronounced phonemes in the same word is similar for all people. This process requires first correlating between the new person images and the model, and then embedding of the new person's pronounced word on the model's lip configuration surface and calculating a new contour. Next, comparison between the new contour and contours of known words, previously calculated for the model, is computed, and the closest word is chosen as the analysis result.

2. Previous Work

Automatic understanding (analysis) and generation (synthesis) of lip movements may be helpful in various applications, and these areas are under intense study. We first review some of the recent results in this field.

2.1. Analysis

The problem of analyzing lip movements, and automatic translation of such movements into meaningful words was addressed in several papers. Some researchers treat lipreading as a stand-alone process (Bregler et al., 1998; Li et al., 1997), while others use it to improve voice recognition systems (Bregler and Omohundro, 1994; Bregler et al., 1993; Luettin, 1997).

Li et al. (1997) investigated the problem of identification of letter's pronunciation. They handled the first ten English letters, and considered each of them as a short word. For training, they used images of a person saying the letters a few times. All images were aligned using maximum correlation, and the sequence of images of each letter were squeezed or stretched to the same length. Each such sequence of images was converted into a $M \times N \times P$ column vector, where $M \times N$ is the size of each image, and P is the number of images in the sequence (simple concatenate of the sequence). Several such vectors representing the same letter created a new matrix, A, of size $MNP \times S$, where S is the number of sequences. The first eigenvectors of the squared matrix AA^T were considered as the principle components of the specific letter's space. Those principle components were called eigen-sequences. When a new sequence is analyzed, it is aligned as before and matched with each of the possible letters. First, the new sequence is squeezed or stretched to the same length of a possible letter's sequence. Then, the new sequence is projected onto this letter's basis, and the amount of preserved energy is tested. The letter which basis preserves most of the new letter energy is chosen as the pronounced letter in the new sequence. In that paper, an accuracy of about 90% was reported.

An interesting trial for lipreading was introduced by Bregler et al. (1998) under the name of 'the bartender problem'. The speaker, as a customer in a bar, is asked to choose between four different drinks, and due to background noise, the bartender's decision of the customer's request is based only on lipreading. Data was collected on the segmented lips' contour, and the area inside the contour. Then, a Hidden Markov Model (HMM) system was trained for each of the four options. With a test set of 22 utterances, the system was reported to make only one error (4.5%).

A different approach was used in Mase and Pentland (1991), where the lips are tracked using optical flow techniques, and features concerning their movements and motion are extracted. They found that the vertical lip separation and the mouth elongation capture most of the information about the pronounced word. In the recognition stage, this information is compared with previously known templates, and a decision is taken. Another interesting use in optical flow techniques for human facial expressions detections was done by Yacoob and Davis (1996). There, the tracking algorithm integrates spatial and temporal information at each frame, and those motion characteristics are used to interpret human expressions.

The latter techniques extract specific information about the lips motion and formation, while assuming these features determine most the underlying pronounced word (or expression). Here, we preferred to work with images of the mouth area, and allow the application decide which are the most dominant features that identify the pronunciation.

Acoustics-based automatic speech recognition (ASR) is still not completely speaker independent, its vocabulary is limited, and it is sensitive to noise. Bregler et al. (1998, 1993) showed, using a neural network architecture, that visual information of the lip area during speech can significantly improve (up to 50%) the error rate, especially in a noisy environment. In their experiments, they use a neural network architecture in order to learn the pronunciation of letters (each letter is considered as a short word). Apart from acoustic information, their systems made use of images of the lips area (grey level values, first FFT coefficients of the region around the lips, or data about the segmented lip). The results demonstrated that such hybrid systems can significantly decrease the error rate. More improvement was achieved, as expected, when the amount of noise was high, or for speakers with more emphasized lips movements, i.e., speakers that move their lips more while talking.

Duchnowski et al. (1995) developed a similar framework for an easy interaction between human and machine. A person, sitting in front of a computer, was recorded and videotaped while pronouncing letters. The subject's head and mouth were tracked using a neural network based system. Several types of visual features were extracted, such as gray level values, band-pass Fourier magnitude coefficients, principal components of the down sampled image, or linear discriminant analysis coefficients of the down sampled image. The acoustic and visual data was processed by a multi-state time delay neural network system with three layers, and 15 units in the hidden layer. By combining the audio and visual information, they achieved a 20-50% error rate reduction over the acoustic processing alone, for various signal/noise conditions.

2.2. Synthesis

Bregler et al. (1997) introduced 'video-rewrite' as an automatic technique for dubbing, i.e. changing a person's mouth deformations according to a given audio track. They preferred handling triples of phones, and so achieved natural connection between each two. Using segmentation of a training audio track, they labelled the facial images, and each sequential three phonemes were handled separately. Next, they segmented the phonemes in the new audio track, and combined triples of phonemes that resembled the segmentation results. The choice of handling triples of phonemes enabled natural connection between all parts of the sentence. They used a 'stitching' process to achieve correspondence between the synthesized mouth movements and the existing face and background in the video.

A different synthesis procedure by Bregler et al. (1998) was based on their concept of 'constrained lip configuration space'. They extracted information on the lip contour, and embedded this information in a five-dimensional manifold. Interpolation between different images of the mouth was done by forcing the interpolated images to lie on this constrained configuration space.

Kalberer and Van Gool (2001) and Vanroose et al. (2002) chose to handle 3D faces. They worked with a system called "ShapeSnatcher", that uses a structured light technique, in order to acquire 3D facial data. The 3D structure has an advantage over flat images in both analysis and synthesis. It better captures the facial deformations, it is independent of the head pose, and when synthesizing, the geometric information enables animation of a virtual speaker from several viewing directions.

Kalberer and Van Gool (2001) introduced the concept of 'eigenfacemasks'. A 124 vertices in 3D define a facial mask, where 38 vertices are located around the lip area. They acquired face geometry of a single person pronouncing various phonemes. Each frame was analyzed separately, and represented as a mask. The mask's vertices are matched to facial points by marking black dots on the face of the speaking subject. After acquiring several such sequential masks, the first 10 eigenvectors were extracted. The space that these eigenvectors span was considered as the space of intra-subject facial deformations during speech. For animation of a certain word, its viseme[1] face masks were displayed, and spline interpolation between the coefficients of the eigenvectors was used to smooth the transitions. The interpolation is between the coefficients of the projection of the different visemes masks on the chosen eigenvectors. It means that each intermediate mask was embedded in the eigenmask space. The eigenfacemasks' compact space requirements enabled an easy generation of intermediate masks, that look realistic.

In the latter two papers the use of a small lip configuration space allows transitions between two

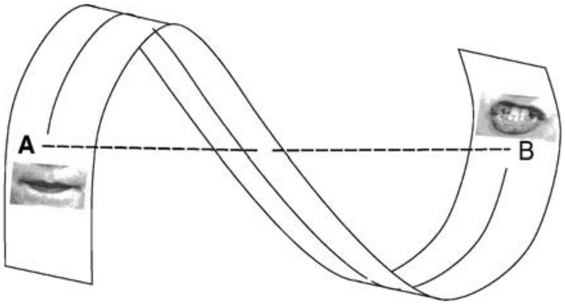

Figure 1. Smoothing the transition between different lips configurations.

configurations that is restricted to that space. Indeed, interpolating on a simple space that captures the lips configurations enables efficient natural transitions, and will be used also in our framework. In Fig. 1, the surface illustrates a limited 3D lips configuration space, and points 'A' and 'B' are two specific lips configurations on that manifold. These two configurations are different, so sequential presentation of them might cause a 'jerky' effect. Linear interpolation between the two configurations creates images off the restricted space (the dashed line), and would look un-natural. A much better synthesis of a smooth and natural transition between the two configurations, is restricted to the lips configuration space (described as a solid line on the manifold).

3. Visual Speech Synthesis and Lipreading by Flat Embedding

Different people pronounce the same vowels differently. Even the pronunciation of the same person in different scenarios may change. We chose to explore the case of a single subject speaking to the camera and slightly accentuating the words.

Each vowel is pronounced differently when said in different parts of a word. For example, the vowel 'A' in 'America' looks different from the vowel 'A' in 'Los Angeles'. This difference occurs (among other subjective reasons) due to the location of the syllable 'A' in the word, and the syllables that appear before and after it. One may realize that the main reason for different pronunciation of the same vowel is the formation of the mouth just before and after this syllable is said.

In our framework, we divide each word into isolated parts, each containing a consonant and a vowel, or a consonant alone, e.g. 'ba', 'ku', 'shi', 'r' etc. Each of these sounds is considered as a syllable. We assume that each syllable has its own '*visual articulation signature*' (VAS in short), i.e. the sequence of mouth motions that must occur in order for the sound to be vocalized. These mouth motions may differ from one person to another. Other parts of the full visual pronunciation of a syllable can be neglected. Figure 2 shows a series of images of a mouth pronouncing the syllable 'sha'. The VAS is defined by images 11–19. Here, identification of the VAS images was done manually.

3.1. The Input Data

Our subject (the first author) was videotaped while pronouncing 20 syllables, each pronounced six times, each time as a different vowel (A, E, I, O, U, and 'sheva', a consonant that carries an ultra-short vowel or no vowel sound). Each of the 120 sequences started and ended with a closed mouth. An example of such a sequence is shown in Fig. 2. For each sequence, the indices of the VAS were registered and recorded. The total number of images was about 3450.

3.2. Comparing Images

Alignment: The images were taken using a stationary camera, while the subject was sitting. Nevertheless, slight movements of the head are unavoidable, and the images were first aligned. As the nose is stable while talking, it was chosen as the alignment object. Each image was translated, using an Affine Motion detector algorithm (Lucas and Kanade, 1981; Bergen et al., 1992; Aharon and Kimmel, 2004), so that the nose is completely stable. After alignment, only the mouth-area (as seen in Fig. 2) was considered.

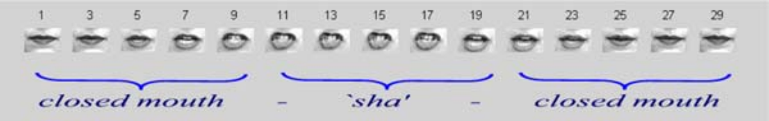

Figure 2. One syllable image sequence.

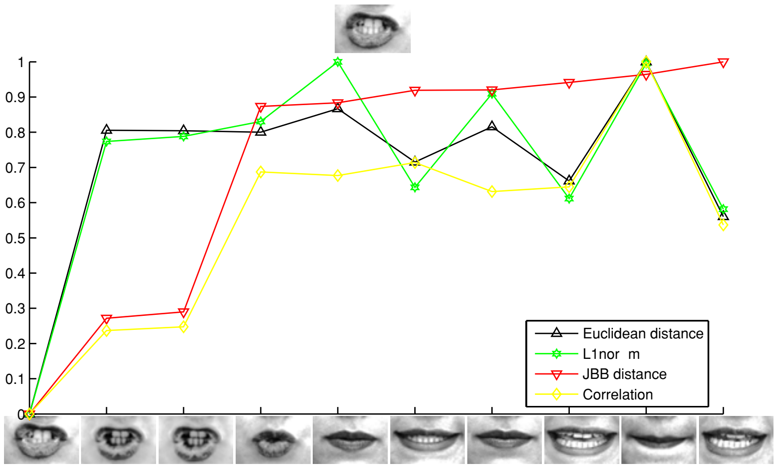

Figure 3. Comparison between various measures for distance between images.

Comparison Measure: As a distance measure between images we chose a variation on the Jacobs, Belhumeur, and Basri (JBB) measure (Jacobs et al., 1998), given by

$$E(I, J) = \int\int I \cdot J \left|\nabla\left(\frac{I}{J}\right)\right| \cdot \left|\nabla\left(\frac{J}{I}\right)\right| dxdy, \quad (1)$$

where $I(x, y)$ and $J(x, y)$ are two images and $E(I, J)$ is the relative distance between them.

Let us briefly motivate the JBB measure. Assume that an object $\{x, y, f(x, y)\}$ is viewed from direction $(0, 0, -1)$, its surface normals are $\frac{(f_x, f_y, 1)}{\sqrt{f_x^2+f_y^2+1}}$. When this object, assumed to be Lambertian, is illuminated by one light source from direction (s^x, s^y, s^z), the intensity image is given by

$$I(x, y) = \alpha(x, y)\frac{-(s^x, s^y, s^z) \cdot (f_x, f_y, 1)}{\sqrt{f_x^2 + f_y^2 + 1}}, \quad (2)$$

where $\alpha(x, y)$ is the albedo function of the object.

Dividing two images of the same object, taken under different illumination conditions, the albedos and the normalization components cancel out one another. Roughly speaking, the resulting ratio is ‘simpler’ than the ratio of two images of different objects. A simple measure of the complexity of the ratio image is the integral over its squared gradients $|\nabla(\frac{I}{J})|^2$. Symmetry consideration, and compensating for singularities in shadowed areas lead to the above measure.

In order to validate the JBB measure, we compared it to the L_1 and L_2 norms and to the correlation measure, all calculated on a slightly smoothed (with a 5 × 5 Gaussian kernel with standard deviation 0.8) version of the images. We chose a specific mouth image and compared it to 10 randomly selected mouth images, taken from various pronunciations, at different times, and under slightly different illumination conditions. The comparisons results were normalized between 0 (most similar) and 1 (most different), and are shown in Fig. 3. The random images are ordered according to their JBB distances from the image at the top. The first three images describe the same syllable as the top image (although taken under slightly different illumination conditions). Those images were considered closest to the original image by both the JBB and the correlation measure. However, the JBB was able to better enhance the difference from images that describe other syllables.

Next, using the JBB measure, we calculated the differences between each two images in the input set. We thereby obtained an $N \times N$ symmetric matrix of relative distances (dissimilarity measures), where N is the total number of images.

3.3. Flattening

Our next goal is to embed all the images as points in a finite dimensional Euclidean space, such that the Euclidean distance between each two images is as close as possible to the dissimilarity between the images. This flat embedding offers a compact representation that simplifies the recognition process. For our application, small distances are more significant than larger ones. The reason is that we are interested in representing one

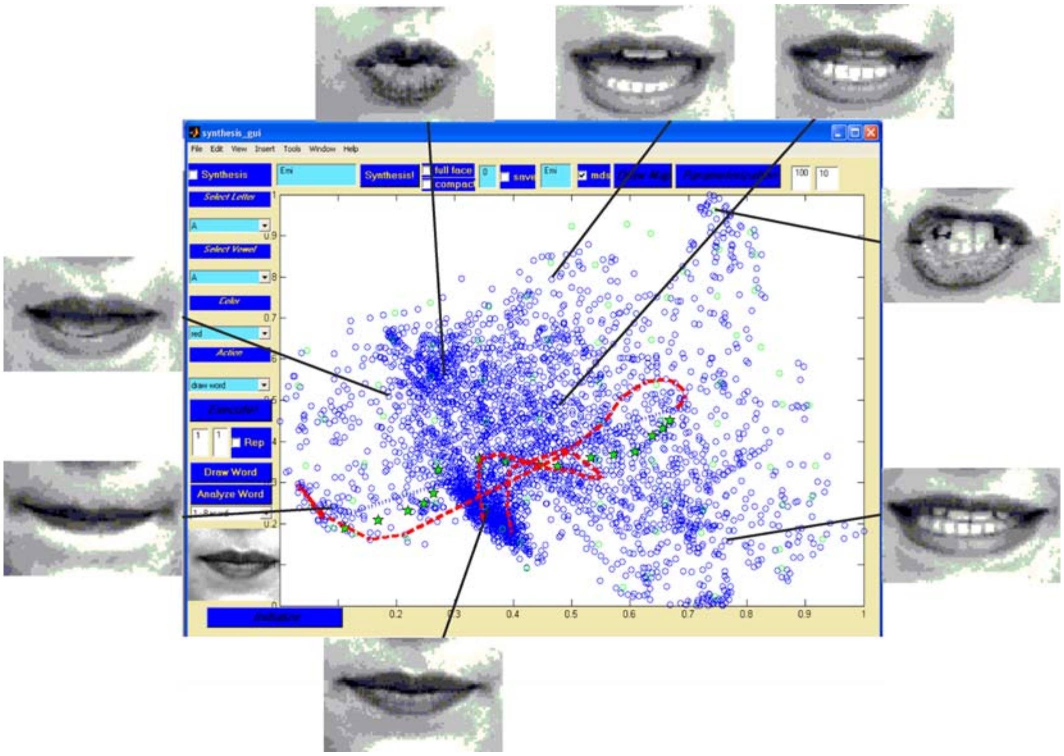

Figure 4. The flat embedding onto a plane.

image using another that is close to it on the flat surface. The accurate distance between two different images is less important, as long as they are far from each other on the representation plane. A related flattening procedure was explored by Roweis and Saul (2000) with full face images, using locally linear embedding (LLE).

Figure 4 shows the screen of a tool we built in order to explore the properties of the flattening process. The embedding flat surface, on which the blue circles are located, is seen in the middle. Each blue circle represents an image, where similar looking images are close to each other. The red contour represents a sequence of mouth images saying the word ‘Emi’. We see that this path is divided into two parts, one for each of the two different syllables that define this word. The green stars represent the images that are shown when synthesizing the word, in order to create a smooth transition between the two syllables. The stars lie almost along a straight line, which connects the two parts of the word. More about this synthesis procedure a head.

It is interesting to note that the flattening procedure we use maps the open mouth images to the upper right part of the screen, while closed mouth images are found at bottom left. Images that contain teeth are mapped to the right, while images without teeth are found at the left part.

We next investigate three flattening methods - locally linear embedding, classical scaling and least squares multidimensional scaling. Each of these methods was tested on our data base of images, and their results were compared. The chosen embedding space is the planar mapping shown in Fig. 4. It was found using least squares MDS with classical scaling initialization.

3.3.1. Locally Linear Embedding.

Locally linear embedding (Saul and Roweis, 2003) is a flattening method designed for preserving the local structure of the data, and addressing the problem of nonlinear dimensionality reduction. The mapping is optimized to preserve the local configurations of nearest neighbors, while assuming a local linear dependence between them. The ‘neighborhood’ definitions of each point is set by the user, and may include all points which distances from a given point is smaller than a certain value, a fixed number of closest points, or any other reasonable neighborhood definition. Given an input set of N data points $X_1, X_2, \dots X_N$, the embedding procedure is divided into three parts:

- Identify the neighbors of each data point, X_i.
- Compute the weights W_{ij} that best reconstruct each data points X_i from its neighbors, by minimizing the cost function

$$E(W) = \sum_i \left| X_i - \sum_j W_{ij} X_j \right|^2. \qquad (3)$$

  A least squares problem.
- Compute the embedded points Y_i in the lower dimensional space. These coordinates are best reconstructed (given the weights W_{ij}) by minimizing the equation

$$\Phi(Y) = \sum_i \left| Y_i - \sum_j W_{ij} Y_j \right|^2. \qquad (4)$$

  An eigenvalue problem.

The output of the algorithm is a set of points $\{Y_i\}_{i=1}^N$ in a low dimensional space, that preserves the local structure of the data. A more detailed description of this method is given in Aharon and Kimmel (2004).

In our case, the input to the LLE algorithm was the matrix of pairwise distances between each two points, and not the initial coordinates of each point (which would have been the image gray-level values). We therefore derived from this matrix the neighborhood relations and the weights calculations, as described in (Saul and Roweis, 2003; Aharon and Kimmel, 2004).

An improvement to the LLE algorithm and the related Isomap (Tenenbaum et al., 2000) was proposed by Donoho and Grimes (2003) by the name of 'Hessian Eigenmaps'. That method can handle the case of a connected non-convex parametrization space. We did not experiment with this method.

3.3.2. Classsical Scaling. *Multidimensional scaling (MDS)* (Borg and Groenen, 1997), is a family of methods that try to represent similarity measurements between pairs of objects, as distances between points in a low-dimensional space. This allows us to visually capture the geometric structure of the data, and perform dimensionality reduction.

First we tested *classical Scaling* (Borg and Groenen, 1997; Aharon and Kimmel, 2004). This method assumes that the dissimilarities between the images are Euclidean distances in some d-dimensional space. Based on this assumption it reveals a centered configuration of points in a d-dimensional world, that best preserves, under Frobenius norm, those distances. Classical scaling's solution in a d-dimensions minimizes the following function,

$$\left\| B - \tau_1(\Delta^2) \right\|_F^2, \qquad \text{subject to} \qquad B \in \Omega_n(d), \quad (5)$$

where $\Omega_n(d)$ is the set of symmetric $n \times n$ positive semi-definite matrices that have rank no greater than d, $\Delta^2 = [\delta_{ij}^2]$ is the matrix of squared dissimilarities, $\tau_1(D) = -\frac{1}{2}(I - \mathbf{1}\mathbf{1}')D(I - \mathbf{1}\mathbf{1}')$ is the double centering operator, and $\mathbf{1} = [1, \dots, 1]' \in R^n$.

This method includes four simple steps. Let Δ^2 be the matrix of squared dissimilarities.

- Apply double centering: $B_\Delta = \tau_1(\Delta^2)$.
- Compute eigendecomposition of $B_\Delta = Q \Lambda Q'$.
- Sort the eigenvalues, and denote

$$\Lambda_{ii}^+ = \begin{cases} \Lambda_{ii} & \text{if } \Lambda_{ii} > 0, i < d \\ 0 & \text{otherwise} \end{cases}$$

- Extract the centered coordinates by $X = Q\Lambda^{+^{1/2}}$.

If the distances would have been indeed between points in a d-dimensional Euclidean space, then classical scaling provides the exact solution. Otherwise, it provides only an approximation, and not necessarily the one we would have liked.

In our application, we tested the classical scaling solution in two-dimensional space. The coordinates in the representation planar space are given by the first two eigenvectors of the double centered distances matrix, scaled by their corresponding (largest) eigenvalues. This method also provides the accuracy of the representation captured by the first two eigenvalues, which can be measured by the following 'energy' term, (a variation of the Frobenius norm)

$$E = \sqrt{\frac{\sum_{i=1}^{2} \lambda_i^2}{\sum_{i=1}^{N} \lambda_i^2}}, \qquad (6)$$

where λ_i is the ith largest eigenvalue of the distances matrix, after double centering. In our case, the first two eigenvalues capture approximately 95% of the energy. This number validates the fact that our images can be embedded in a plane with insignificant distortion, which is somewhat surprising.

3.3.3. Stress Definitions. Classical scaling prefers the order by which the axes are selected, and thus minimize the Frobenius norm. Next, we use an unbiased measure, that takes into consideration the dimension of the target space, in order to evaluate the quality of the flattening procedure (Borg and Groenen, 1997). We first define the representation error as

$$e_{ij}^2 = (\delta_{ij} - d_{ij})^2, \tag{7}$$

where δ_{ij} and d_{ij} are the dissimilarity and the Euclidean distance in the new flat space between points i and j, respectively. The total configuration's representation error is measured as the sum of e_{ij}^2 over all i and j, that defines the *stress*

$$\sigma(X) = \sum_{i<j} (\delta_{ij} - d_{ij})^2. \tag{8}$$

Here d_{ij} is the Euclidean distance between points i and j in the configuration X. In order to weigh differently smaller and larger distances, we consider a weighted sum

$$\sigma_W(X) = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij})^2. \tag{9}$$

Finally, we normalize the stress to obtain a comparable measure for various configurations with some insensitivity to the number of samples,

$$\hat{\sigma}_W(X) = \frac{\sum_{i<j} w_{ij}(\delta_{ij} - d_{ij})^2}{\sum_{i<j} w_{ij} \cdot \delta_{ij}^2}. \tag{10}$$

Using this measure, we can compare between various flattening methods. The stress results for classical scaling and LLE, calculated without weights, and with weights $w_{ij} = 1/\delta_{ij}^2$, are given in Tables 1 and 2.

3.3.4. Least Squares Multidimensional Scaling. Least-Square MDS (Borg and Groenen, 1997) is a flattening method that directly minimizes the stress value in Eq. (10). The optimization method we used is called *iterative majorization* (Borg and Groenen, 1997; Aharon and Kimmel, 2004). The initial configuration of the least squares MDS is crucial, due to the existence of many local minima. In our experiments, when initialized with a random configuration, the resulting stress values were worse than the one achieved by classical scaling. We thus initialized the algorithm with a configuration that was found by classical scaling. That yielded a significant improvement (see Table 1). We also tried to multiply the classical scaling configuration by a scalar according to the suggestion of Malone et al. (2000) for a better initial configuration for the least squares procedure. In our experiments this initialization did not improve the final results.

Table 1. Stress values for different variations of MDS. The weighted stress is calculated with $w_{ij} = 1/\delta_{ij}^2$.

Method	Un-weighted stress	Weighted stress
Classical MDS	0.095	0.1530
Least Squares MDS with random initialization	0.159	0.0513
Least Squares MDS with LLE initialization	0.022	0.0550
Least Squares MDS with Classical Scaling initialization	0.022	0.0361

Table 2. Stress values for different versions of LLE. The weighted stress is calculated with $w_{ij} = 1/\delta_{ij}^2$.

Method	Un-weighted stress	Weighted stress
Fixed number of neighbors (5)	0.951	0.948
Fixed number of neighbors (20)	0.933	0.948
Fixed Threshold (0.019)	0.927	0.930

We search for a configuration that better preserves small distances, and gives higher accuracy. For that end, we defined a weight matrix, that is derived from the dissimilarities matrix by $w_{ij} = 1/\delta_{ij}$, In this case, errors are defined by the relative deformation. By this normalization, larger distances can suffer larger deformations.

3.3.5. Flattening Methods—Comparison. All the three methods were applied to our data base. The stress values (weighted and un-weighted) of classical scaling and least squares MDS (with different initial configurations) can be seen in Table 1. Stress values are computed with the same weights used in the minimization.

We also tested LLE using three different neighborhood definitions: 5 nearest neighbors for each point, 20 nearest neighbors for each point and all neighbors which distances to the point is less than 0.019 (between

1 and 1102 neighbors for each point). The results were tested by calculating the un-weighted stress value, and the weighted stress value with the same weights as before ($w_{ij} = 1/\delta_{ij}^2$). The results are presented in Table 2.

Another recent method for dimensionality reduction, which we did not investigate, is the *'Isometric feature mapping'* or *ISOMAP* (Tenenbaum et al., 2000), see Schwartz et al. (1989) for an earlier version of the same procedure. This method assumes that the small measured distances approximate well the geodesic distances of the configuration manifold. Next, using those values, geodesic distances between faraway points are calculated by a graph search procedure. Finally, classical scaling is used to flatten the points to a space of the required dimension. Isomap introduces a free parameter that sets the neighborhood size, and prevents us from comparing reliably between the various methods. In our application, using Least-Squares MDS enabled us to decrease the influence of large distances. Weighting the importance of the flattened distances can replace the need to approximate large distances, as is done to in Isomap.[2] Moreover, we demonstrate empirically, that the small stress values computed by the flat embedding via Least-Squares MDS validates the numerical correctness of the method we used for the lips images.

3.4. Space Parametrization

Thousands of images were flattened to a plane, and generated regions with varying density, as can be seen in Fig. 4. In order to locate the coordinates of a new image in the plane, we first select 'representative key images' by uniformly sampling the plane. We use only this sub-set of images to estimate the coordinates of a new image. In our experiments we selected 81 images (out of 3450) to sample the representation plane. This was done by dividing the plane into 100 squares (10 squares in each row and column). For each square that contained images, the image which is closest to the median coordinates was selected as a 'representative key image' (the median coordinate in both x and y were calculated, and then the image which is closest to this point was selected). Next, in order to locate the coordinates of a new image in the representation plane the following steps were followed.

1. The nose in the new image is aligned, by comparing to one of the previously taken images, using an affine motion tracker algorithm.
2. The JBB distances between the new image, and each of the 'representative key images' were calculated.
3. The coordinates of the new image are set as a weighted sum of the representatives' coordinates, according to the distance from each representative.

$$x_{\text{new}} = \frac{\sum_{i=1}^{N} w_i \cdot x_i}{\sum_{i=1}^{N} w_i}, \tag{11}$$

where N is the number of 'representative key images', x_i is the x coordinate of the ith representative key image and the weight w_i is set to be $1/\delta_i^3$, where δ_i is the JBB distance between the new image, and the ith representative key image. The y coordinate was calculated in a similar way.

3.5. Sentence Synthesis

A simple way to synthesize sequences using the facial images is by concatenating the VAS of the syllables that integrate into the sentence, so that the 'crucial' part of each syllable is seen. The first and last part of the sequence of pronunciation of each syllable appears only if this syllable is at the beginning or the end of the synthesized word, respectively.

This concatenating procedure results in unnatural speaking image sequences because the connection between the different partial sequences is not smooth enough. An example can be seen in Fig. 5. There, a simple synthesis of the name "Emi" is performed as described above, and the transition between the two syllables (images 14 and 15) can be easily detected.

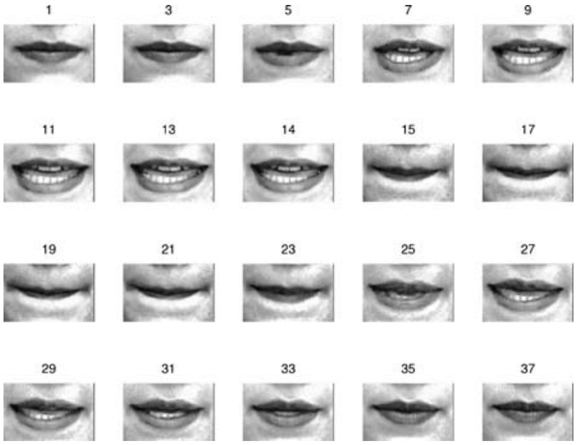

Figure 5. Simple synthesis sequence of the name "Emi."

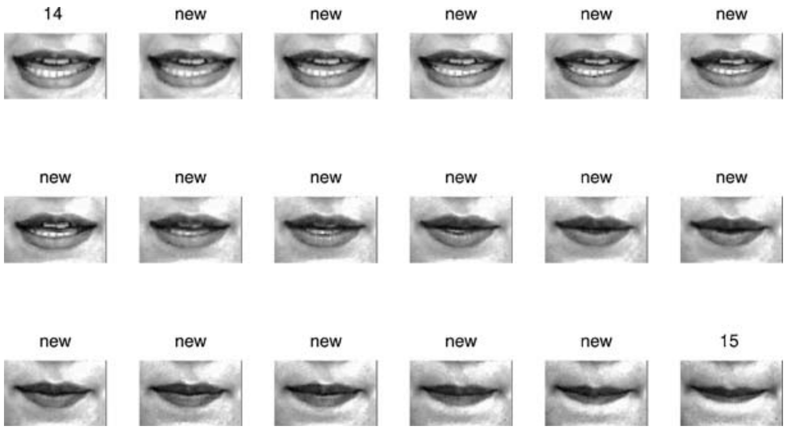

Figure 6. Smooth transition between images 14 and 15 in Fig. 5.

A possible solution to this problem is by defining a weighted graph clique; a graph in which there is a weighted edge between each two vertices. The vertices represent the input images and the weight of the edge between vertex i and j is the dissimilarity measure between the two corresponding images. A smooth transition between images A and B can be performed by presenting the images along the shortest path between A and B. This path is easily found using Dijkstra's algorithm. The shortest path between an image at the end of the VAS of the first syllable, and an image at the beginning of the VAS of the next syllable is used to synthesis smooth transactions, as shown in Fig. 6. There, 16 images, marked as 'new', were found by the Dijkstra algorithm as the shortest weighted path between the last image of the viseme signature of the phoneme 'E' (number 14) and the first image of the viseme signature of the phoneme 'Mi' (number 15). This smooth connection between two different lips configurations is obviously embedded in the constrained lips configuration space.

In this solution, a problem may occur if the path that is found includes too many images. Merging those images may slow down the pronunciation, whereas the duration of the pronunciation and synchronization with the sound is crucial when synthesizing speech. We, therefore, control the number of images that are displayed by re-sampling the sequence. An example of a shorter smooth transition is shown in Fig. 7.

Another solution, that exploits the embedding surface and the chosen representative key images is to define a clique weighted graph which nodes are the representative key images, and the two images between which the smoothing should be done. The weight of the edge that connects images i and j is the distance measure between the two images. The smooth transition contains the images along the shortest path between the two images. Computing this solution is much faster, as the Dijkstra algorithm runs on a much smaller graph. The paths that are found rarely need re-sampling, as they are much shorter than those in the full graph. An example of the synthesis of the name 'Emi' appears in Fig. 8.

The synthesis procedure is completely automatic. The input is defined by the text to be synthesized and possibly the time interval of each syllable

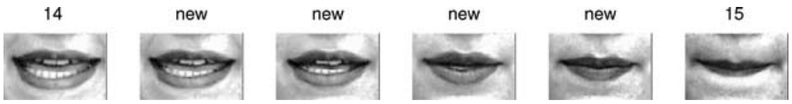

Figure 7. Sampled transition between images 14 and 15 in Fig. 5.

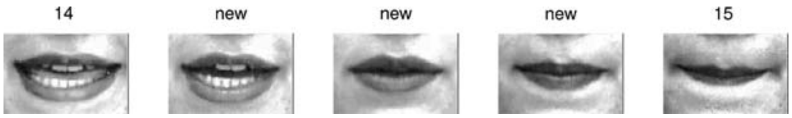

Figure 8. Smooth transition between images 14 and 15 in Fig. 5, using the (sampled) embedding-based synthesis method.

pronunciation, as well as the pauses between the words. The results look natural as they all consist of realistic, aligned images, smoothly connected to each other.

3.6. *Lipreading*

Here we extend the 'bartender problem' proposed by Bregler et al. (1998). We chose sixteen different names of common drinks,[3] and videotaped a single subject (the same person that pronounced the syllables in the training phase) saying each word six times. The first utterance of each word pronunciation was chosen as reference, and the other utterances were analyzed, and compared to all the other reference sequences. After the surface's coordinates of each image in each word sequence (training and test cases) are found, each word can be represented as a contour. Analyzing a new word reduces to comparing between two such contours on the flattened representation plane.

Comparing Contours: The words' contours, as an ordered list of coordinates, usually include a different number of images. In order to compare two sequences we first fit their lengths. We do so by using a version of the Dynamic Time Warping Algorithm (DTW) of Sakoe and Chiba (1978) with a slope constraint of one. This algorithm is commonly used in the field of speech recognition (Li et al., 1997). The main idea behind the DTW algorithm is that different utterances of the same word are rarely performed at the same rate across the entire utterance. Therefore, when comparing different utterances of the same word, the speaking rate and the duration of the utterance should not contribute to the dissimilarity measurement.

Let us denote the two sequences images as: $A = [a_1, a_2, \dots a_m]$, and $B = [b_1, b_2, \dots b_n]$, where $a_i = \{x_i, y_i\}$ are the x and y coordinates of the $i-th$ image in the sequence. We first set the difference between images a_1 and b_1 as $g(1, 1) = 2d(1, 1)$, where $d(i, j)$ is the Euclidean distance $\|a_i - b_j\|_2$. Then, recursively define

$$g(i,j) = \min\left\{\begin{array}{l} g(i-1, j-2) + 2d(i, j-1) + d(i, j), \\ g(i-1, j-1) + 2d(i, j), \\ g(i-2, j-1) + 2d(i-1, j) + d(i, j) \end{array}\right\}. \tag{12}$$

Where $g(i, j) = \infty$, if i or j is smaller than 1. The distance between sequences A and B is $g(m, n)$. The indices of the minimum chosen values (each index can vary from 1 to 3, for the 3 possible values of $g(i, j)$) indicates the new parametrization of the sequence A, in order to align it with the parametrization of the sequence B. Using dynamic programming, the maximum number of Euclidean distance computations is $m \cdot n$, and therefore, the computation is efficient.

When a new parametrization s is available, the first derivative of sequence A is calculated using backward approximation $x'^A_s = x^A_s - x^A_{s-1}$, and second derivatives using a central scheme $x''^A_s = x^A_{s+1} - 2x^A_s + x^A_{s-1}$. In this new parametrization the number of elements in each sequence is the same, as well as the number of elements of the first and second derivatives, that can now be easily compared. Next, three different distance measures between the two contours are computed

$$\begin{aligned} G(A, B) &= g(m, n) \\ P(A, B) &= \sum_{s=1}^{n}\left(\left(x'^A_s - x'^B_s\right)^2 + \left(y'^A_s - y'^B_s\right)^2\right) \\ Q(A, B) &= \sum_{s=1}^{n}\left(\left(x''^A_s - x''^B_s\right)^2 + \left(y''^A_s - y''^B_s\right)^2\right). \end{aligned} \tag{13}$$

Those measures are used to identify the closest reference word to a new pronounced word.

Let us summarize the whole analysis process. When receiving a new image sequence N,

1. find the path that corresponds to the sequence by locating the representation plane coordinates of each image in the sequence as described in Section 3.4.
2. For each reference sequence R_j, for $j = 1$ to k, where k is the number of reference sequences (16 in our experiments) do:

   (a) Compute the DTW between the sequence N and R_j.

   (b) Use these results to compute the distances $G(N, R_j)$, $P(N, R_j)$, and $Q(N, R_j)$.

3. Normalize each distance by

$$\tilde{G}(N, R_j) = G(N, R_j) \bigg/ \sum_{i=1}^{k} G(N, R_i)$$

$$\tilde{P}(N, R_j) = P(N, R_j) \bigg/ \sum_{i=1}^{k} P(N, R_i)$$

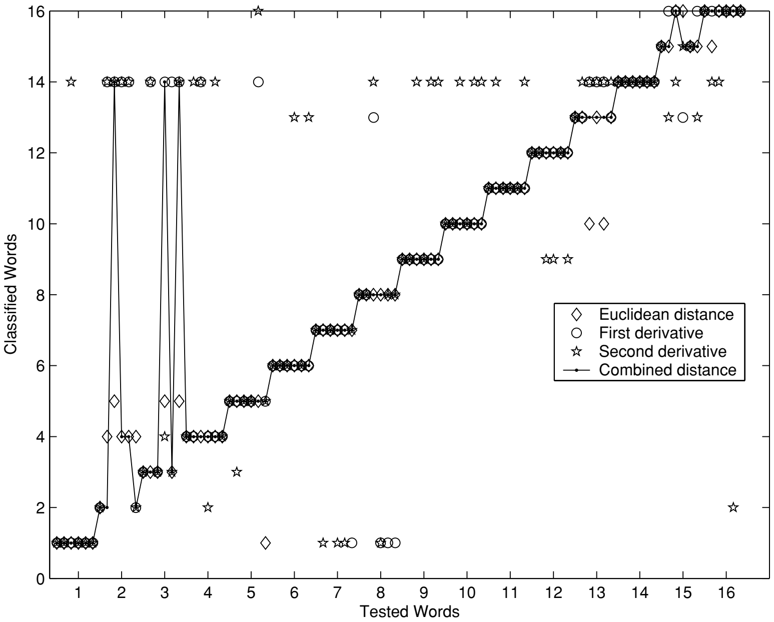

Figure 9. Analysis results of the different distance measures.

$$\tilde{Q}(N, R_j) = Q(N, R_j) \Bigg/ \sum_{i=1}^{k} Q(N, R_i). \tag{14}$$

4. For each reference sequence, compute the distances

$$D_j(N) = \tilde{G}(N, R_j) + \alpha \cdot \tilde{P}(N, R_j) + \beta \cdot \tilde{Q}(N, R_j). \tag{15}$$

   In our experiments, we empirically found that $\alpha = \beta = \frac{1}{2}$ give the best classification results.
5. Select the closest reference sequence, the one with the smallest distance $D_j(N)$, as the analysis result.

The combination of the integral Euclidean distance with the first and second derivatives is an approximation of the Sobolev Spaces norm, defined as

$$\|f\|_{H^2}^2 = \sum_{j=0}^{k} \left\|f^{(j)}\right\|_{L^2}^2 = \|f\|^2 + \left\|f'\right\|^2 + \left\|f''\right\|^2. \tag{16}$$

We next show that this hybrid norm gives better classification results than each of its components alone.

Results: We tested 96 sequences (16 words, 6 utterances of each word, one of which was selected as the reference sequence). The accuracy rate is 94% (only 6 errors). A careful inspection of the misclassified words, we noticed that those were pronounced differently because of a smile of other spasm in the face. When analyzing single words, those unpredictable small changes are hard to ignore. The results of the different measures (Euclidean, first, second derivatives, and their combination) can be viewed in Fig. 9. The 96 utterances are divided into 16 groups along the x axis. The diamond, circle, and star icons indicate the analysis results computed with the Euclidean, first derivative, and second derivative distance, respectively. The line indicates the result of the approximated Sobolev norm that combines the three measures. The six miss-classifications are easily detected as the deviations from the staircase structure. We see that the Euclidean distance is more accurate than the noisy first and second derivative distances. That was the reason for its relative high weight in the hybrid Sobolev norm. The combination of the three measures yield the best results.

We believe that an increase of the number of different identified words will be difficult using the above framework, mainly due to the current small differences between each two words. Which is an indication that lip-reading is intrinsically difficult. However, supporting an ASR system, differing between 2–3 possible words or syllables is often needed in order to achieve higher identification rates. In this case, our framework would be useful.

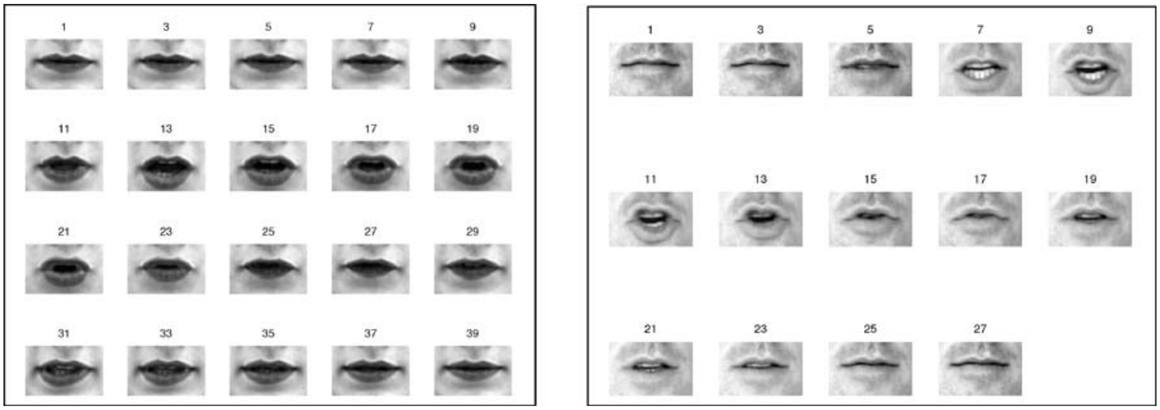

Figure 10. Pronunciation of two different people.

3.7. Generalization: Lipreading Other People

Up until now, we handled facial images of a single person (female). Here, we present a generalization in which we lip read other people. Instead of performing the whole learning process, we exploit the fact that different people say the different words in a similar way. That is, the sequence of pronounced phonemes is equal, when saying the same word. Therefore, after a proper matching between the lips configuration images of the model and the new person, we expect the representing contours of the same word to look similar.

For that end, we took pictures of second person (male), pronouncing the various drinks' names, three times each word. In Fig. 10 we can see the two people pronouncing the word 'Coffee'.

Comparing mouth area images of two different people might be deceptive because of different facial features such as the lips thickness, skin texture or teeth structure. Moreover, different people pronounce the same phonemes differently, and gray level or mouth's contour comparison between the images might not reveal the true similarity between phonemes. For that end, we aligned the new person's nose to the nose of the model using Euclidean version of Kanade-Lucas. An affine transformation here may cause distortion of the face due to different nose structures. Next, the rest of the images are aligned to the first image (of the same person) using affine Kanade-Lucas algorithm, so that all the mouth area images can be extracted easily.

Then, we relate between images of the new person and our model by defining a set of phonemes, and assigning each phoneme a representing image (also known as viseme). The visemes of the model are located on the representation plane using the method described in Section 3.4. The new person's visemes are assigned **exactly the same coordinates**. In our experiments, the process of assigning an image for each phoneme was done manually. Figure 11 shows part of the visemes we assigned for the model and the new person.

Next, the location of the new person's images on the surface is found using the following procedure.

- The image is compared to all the assigned visemes of the same person, resulting the similarity measures $\{\delta_i\}_{i=1}^{N}$, where N is the number of visemes.
- The new coordinates of the image is set by

$$x_{new} = \frac{\sum_{i=1}^{N} w_i \cdot x_i}{\sum_{i=1}^{N} w_i}, \tag{17}$$

  where x_i is the x coordinate of the ith viseme and the weight w_i is set to be $w_i = 1/\delta_i^2$. The y coordinate is set in a similar way.

In the above procedure, only images of the same person are compared. This way, each new person's image can be located on the representation plane, and each new pronounced word is described as a contour which can be compared with all the other contours. In Fig. 12 four such contours are shown, representing the pronunciation of the words 'Cappuccino' and 'Sprite' by two different people – the model on the left, and the second person on the right.

For comparison between pronunciation contours of two different people we added two additional measures, which we found helpful for this task,

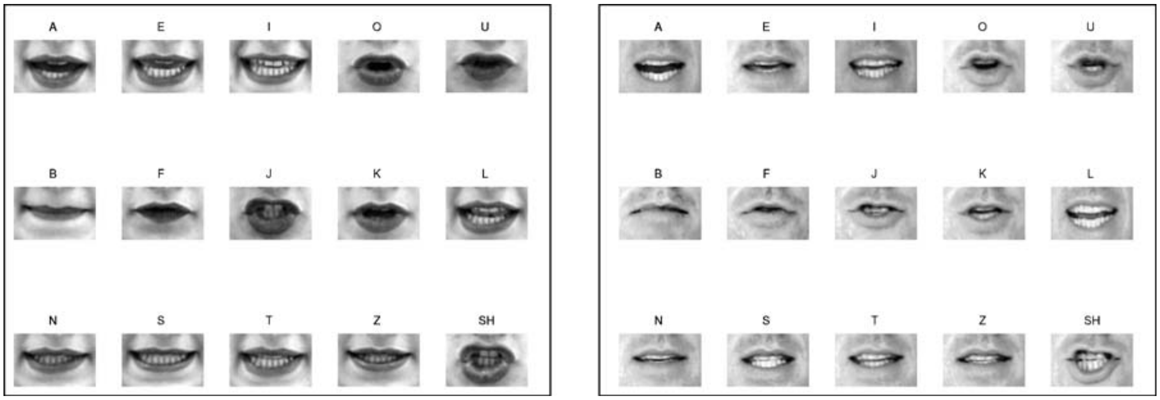

Figure 11. Visemes assignment. Partial group of the visemes we assigned for the model (left) and the new person.

- maximum distance, which is defined by

$$\min_{1\leq s\leq n}\{d(X_s^A - X_s^B)\} \tag{18}$$

  where $X_s^A = [x_s^A, y_x^A]$ and $X_s^B = [x_s^B, y_x^B]$ are the parametrization of the two contours, as seen in Section 3.6, after executing DWT, and $d(X, Y)$ is the Euclidean distance between points X and Y.
- Integrated distance, defined by

$$\sum_{1\leq s\leq n} d(X_s^A - X_s^B). \tag{19}$$

The above two measures refer only to the parametrization of the contour after processing DWT. The maximum distance measures the maximum distance between two correlated points on the two contours, and the integrated distance accumulates the Euclidean distances between the correlated points.

We discovered that the derivative distances that were defined in 3.6 and helped comparing between two contours of the same person, were too noisy in this case. The inner structure (first and second derivatives) of the contour was less important than its coordinates. An example can be seen in Fig. 12 where contours of the same pronounced word are shown. The point locations of the two contours is similar, but their inner structure is different.

The identification rate was 44% in the first trial, and reached 60% when allowing the first two answers (out of 16) to be considered. We relate this relatively low success rate to the assumption that different people pronounce the transitions between phonemes differently, and therefore, correlating between the phoneme's images of two different people is not enough for perfect identification. This conclusion is based on the fact that the lowest identification rate were for names composed of two words ('Bloody Mary','Milk Shake','Orange Juice' and 'French Vanilla'). There especially, although pronouncing all the phonemes in the same order, different people connect differently between the words. Considering only the single word-drinks a success rate of 69% is achieved, and considering the first two answers, we reach 80% success rate.

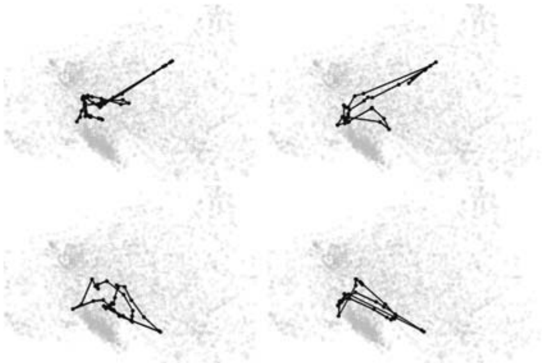

Figure 12. Contours representation of words pronunciation.

3.8. *Choosing Drink Names in Noisy Bars*

Next, we explore the following question, 'What kind of drink names should be chosen in a noisy bar, so that a lipreading bartender could easily recognize between them?'. To answer this question, we measured the distances between each two contours from the set of 96 calculated drink sequences. We received a distances matrix, and performed Classical Scaling. The first two

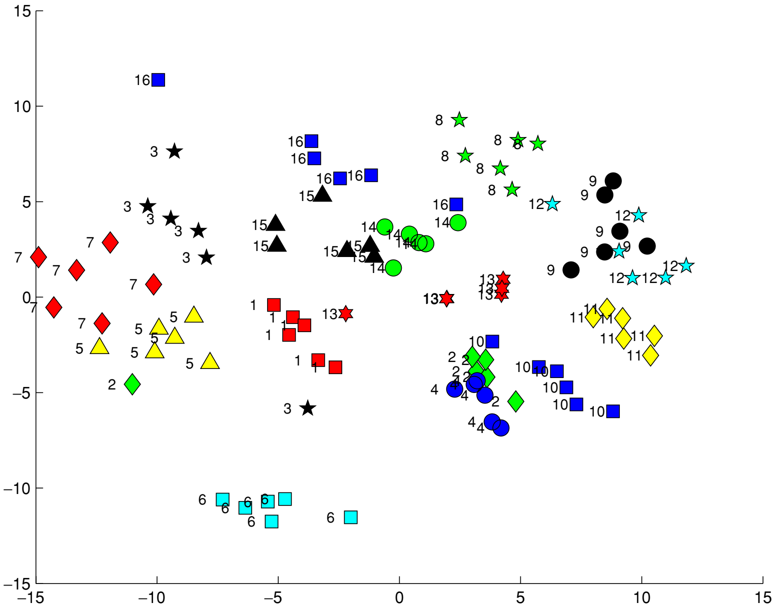

Figure 13. Choosing drink names in noisy bars.

eigenvectors captured 88% of the energy, and the first three 92%. The map presented in Fig. 13, shows that the drinks: 'Bacardi' (1), 'Martini' (5), 'Champagne' (6), 'Milk Shape' (8),'Vodka' (10), 'Cappuccino' (11) and 'liqueur' (14) have more distinct names. Putting them on the menu, possibly with 'Cola' (4) (but without 'Coffee' (2)), or 'Orange Juice' (9) (but without 'French Vanilla' (12)), would ease lipreading of the customers requests.

4. Summary

We introduced a lipreading and lips motion synthesis framework. We qualitatively justified and used the JBB measure for distance evaluation between different images, a measure that is robust to slight pose changes and varying illumination conditions.

We then flattened the visual data on a representation plane. A process we referred to as flattening. This map, which captures the geometric structure of the data, enabled us to sample the space of lips configurations by uniformly selecting points from the embedding surface (the representation plane). Using those selected representatives and the Dijkstra algorithm, we managed to smoothly tile between two different images, and synthesize words.

The embedding surface is then used to represent each pronounced word as a planar contour. That is, a word becomes a planar contour tracing the points on the plane for which each point represents an image. The lip reading (analysis) process was thereby reduced to comparing between planar contours. Comparison between words was then done using an efficient dynamic programming algorithm, based on Sobolev spaces norms.

Finely, generalization of the lipreading process was performed with promising results by exploiting the fact that the sequence of pronounced phonemes is similar to all people pronouncing the same word. This was done by first correlating between a given model and and new subject lips configurations, and then comparing images of the same person only. This way, we find a warp between the representation planes of two unrelated subjects. We then recognize words said by the new subject by matching their contours to known word contours of our model.

Our experiments suggest that exploring the geometric structure of the space of mouth images, and the contours plotted by words on this structure may provide a powerful tool for lip-reading. More generally, we show that dimensionality reduction for images can provide an efficient tool for representation of a single image or a sequence of images from the same family. It can therefor offer a way to perform synthesis and analysis for such sequences.

Acknowledgments

This research was partly supported by European FP 6 NoE grant No. 507752 (MUSCLE).

Notes

1. The term *'viseme'* (Fisher, 1968) is a compound of the words *'visual'* and *'phoneme'*, and here represents the series of visual face deformations that occur during pronunciation of a certain phoneme.
2. Note that evaluating distance by graph search introduces metrication errors and the distances would never converge to the true geodesic distances. This argument is true especially when the data is sampled in a regular way, which is often the case.
3. The tested drink names: Bacardi, Coffee, Tequila, Cola, Martini, Champagne, Bloody Mary, Milk Shake, Orange Juice, Vodka, Cappuccino, French Vanilla, Lemonade, Liqueur, Sprite, Sunrise.

References

Aharon, M. and Kimmel, R. 2004. Representation analysis and synthesis of lip images using dimensionality reduction. Technical Report CIS-2004-01, Technion—Israel Institute of Technology.

Bergen, J.R., Burt, P.J., Hingorani, R., and Peleg, S. 1992. A three-frame algorithm for estimating two component image motion. *IEEE Trans on PAMI*, 14(9).

Bregler, C., Covell, M., and Slaney, M. 1997. Video rewrite: Driving visual speech with audio. *Computer Graphics*, 31:353–360.

Borg, I. and Groenen, P. 1997. *Modern Multidimensional Scaling—Theory and Applications*. Springer-Verlag New York, Inc.

Bregler, C., Hild, H., Manke, S., and Waibel, A. 1993. Improving connected letter recognition by lipreading. In *Proc. IEEE Int. Conf. on ASSP*, pp. 557–560.

Bregler, C. and Omohundro, S.M. 1994. Surface learning with applications to lipreading. In *NIPS*, vol. 6, pp. 43–50.

Bregler, C., Omohundro, S.M., Covell, M., Slaney, M., Ahmad, S., Forsyth, D.A., and Feldman, J.A. 1998. Probabilistic models of verbal and body gestures. In *Computer Vision in Man-Machine Interfaces*, R. Cipolla and A. Pentland (eds), Cambridge University Press.

Donoho, D.L. and Grimes, C.E. 2003. Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Arts and Sciences*, 100(10):5591–5596.

Duchnowski, P., Hunke, M., Bsching, D., Meier, U., and Waibel, A. 1995. Toward movement-invariant automatic lipreading and speech recognition. In *Proc. ICASSP'95*, pp. 109–112.

Fisher, C.G. 1968. Confusions among visually perceived consonants. *Journal of Speech and Hearing Research*, 11(4):796–804.

Jacobs, D.W., Belhumeur, P.N., and Basri, R. 1998. Comparing images under variable illumination. In *Proc. of CVPR*, pp. 610–617.

Kalberer, G.A. and Van Gool, L. 2001. Lip animation based on observed 3d speech dynamics. In *Proc. of SPIE*, S. El-Hakim and A. Gruen, editors, vol. 4309, pp. 16–25.

Li, N., Dettmer, S., and Shah, M. 1997. Visually recognizing speech using eigensequences. In *Motion-Based Recognition*. Klwer Academic Publishing, pp. 345–371.

Lucas, B.D. and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pp. 674–679.

Luettin, J. 1997. *Visual Speech And Speaker Recognition*. PhD thesis, University of Sheffield.

Mase, A. and Pentland, K. 1991. Automatic lipreading by optical flow analysis. Technical Report Technical Report 117, MIT—Media Lab.

Malone, S.W., Tarazaga, P., and Trosset, M.W. 2000. Optimal dilations for metric multidimensional scaling. In *Proceedings of the Statistical Computing Section*.

Roweis, S.T. and Saul, L.K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326.

Sakoe, H. and Chiba, S. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. ASSP*, ASSP-26:43–49.

Saul, L.K. and Roweis, S.T. 2003. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, pp. 119–155.

Schwartz, E.L., Shaw, A., and Wolfson, E. 1989. A numerical solution to the generalized mapmaker's problem: Flattening non convex polyhedral surfaces. 11(9):1005–1008.

Tenenbaum, J.B., de Silva, V., and Langford, J.C. 2000. A global geometric frame-work for nonlinear dimensionality reduction. *Science*, 290:2319–2323.

Vanroose, P., Kalberer, G.A., Wambacq, P., and Van Gool, L. 2002. From speech to 3D face animation. *Procs. of BSIT*.

Yacoob, Y. and Davis, L.S. 1996. Recognizing human facial expressions from long image sequences using optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6).