Lipreading Using Eigensequences [*]

Nan Li, Shawn Dettmer, and Mubarak Shah
Computer Vision Lab
Computer Science Department, University of Central Florida
Orlando, FL 32816
Email: {*shah.linan.dettmer*}*@cs.ucf.edu*

Abstract

To enable computer systems to recognize meaningful lip movements is a potential means to let computers interact naturally with people. We consider the problem of recognizing spoken English alphabet, and present a method for lipreading which uses eigensequences. Since lip movements are essentially spatiotemporal in nature, to remove this statistical redundancy in an integral way, we use the spatiotemporal eigen decomposition, in which the set of eigenvectors span the space of all possible sequences. In order to recognize the successive utterances, a method for separating letters in a connected sequences is also proposed. The method is based on a function of the frame average difference for the sequences and the separation proceeds in a coarse to fine manner.

We have experimented with sequences of English letters "A" to "J", and obtained very encouraging results. The eigensequence based approach for lipreading is very simple and straightforward; the major computation during recognition is a simple dot product.

1 Introduction

Lipreading using computers is a challenging task. Many ideas and methods have been put forward. Yet the general problem of lipreading remains unsolved. In this paper, we present a method for lipreading which uses eigensequences. We consider the problem of recognizing the spoken English alphabet. In our approach, gray level values of all the pixels in all frames in a sequence representing a spoken letter are put in one large vector. Several such vectors corresponding to the training sequences are used to compute eigenvectors (*eigensequences*), for each spoken letter. The recognition of an unknown sequence representing a spoken letter is performed by computing the energy ratio when the sequence is projected on the model eigenspace. For a perfect match, this ratio tends to 1.

Our approach is based on the demonstrated success of the eigenvector approach using static images for face recognition [10, 7] and the similar approach for illumination planning [6]. Turk and Pentland [10] decompose face images into a small set of characteristic feature images called "eigenfaces", the principal components of the initial training set of face images. Recognition is performed by projecting a new image into the subspace spanned by the eigenfaces. As for the problem of lipreading, separate spatial and temporal eigen decompositions have been used [4]. Since lip movements are essentially spatiotemporal in nature. To exploit this statistical redundancy in an integral way, we use the spatiotemporal eigen decomposition, where a set of *eigensequences* is derived as the basis for each modular space corresponding to a spoken letter. Recognition depends on the selection of the modular space which best represents the input lip sequences.

In order to recognize successive utterances, we have developed a method for extracting letters from connected sequences. Our method measures the averaged frame difference of a sequence and extract subsequences corresponding to individual letters in a coarse to fine manner. The procedure begins with the valleys of the smoothed version of the difference function, then locates the beginning and the end of a letter in the neighborhood of the respective valley.

We have experimented with several sequences of English alphabet letters "A" to "J", and obtained very encouraging results. These sequences varied in length, since in real life we speak the same letter with differ-

[*] The research reported here is supported by the National Science Foundation Grants CDA-9200369

ent speeds at different occasions. We use dynamic time warping to align each sequence to a fixed length. Our eigensequence based approach for lipreading is very simple and straightforward; once the eigensequences of each letter are obtained, the major computation during recognition is a simple dot product.

2 Related Work

In Petajan et al. [8], the lipreading task is performed through vector quantization, which replaces each mouth opening image of a sequence by the index of the closest image in the codebook, thus creating a vector of indices representing the sequence. Recognition is done by computing the distance between vector quantized word samples and every vector quantized word model.

In Finn and Montgomery's algorithm [2], twelve dots were placed around the mouth of a speaker and tracked during the experiments; a total of fourteen distances were measured, and used as a feature vector for recognition.

A different scheme was developed by Mase and Pentland [5]. They used optical flow to express the two principal types of motions of the mouth as functions with respect to time: mouth opening $O(t)$ and elongation of the mouth $E(t)$. Templates were used for recognition after a resampling step that normalizes the time to speak one word (time warping).

Kirby et al. [4] used a linear combination of the fixed set of eigenvectors of the ensemble averaged covariance matrix to express mouth images. A spoken word made up of P images can then be expressed as a $Q \times P$ matrix of coefficients computed with respect to the set of Q eigen images. A template matching technique was then used for identification of particular words.

Goldschen [3] used optical information from the oral-cavity shadow of the speaker for continuous speech recognition. His system uses Hidden Markov Models to recognize a set of sentences using visemes, trisemes (triplets of visemes), and generalize trisemes (clustered trisemes).

Bregler and Konig [1] created a hybrid system that uses both acoustic and visual information. Either the principal components of the contours or the principal components of a gray level matrix centered around the lips are taken as the basis of recognition. In the presence of noise, their worked showed improvement for the combined architecture over just acoustic information alone.

3 Eigensequences

Consider a sequence of mouth images, $I_1, I_2, \ldots, I_P$, where each image has M rows and N columns, and P is the number of frames in the sequence. The gray level values of all pixels are then collected throughout the sequence in a column vector as follows:

$$\begin{aligned} u \; = \; & (I_1(1,1), \ldots, I_1(M,N), I_2(1,1), \ldots, I_2(M,N), \ldots, \\ & I_P(1,1), \ldots, I_P(M,N))^T, \end{aligned}$$

where $I_j(x,y)$ is the value of the pixel at location (x,y) in frame j. For n sequences with P frames, a matrix A is made as

$$A = \left[u^1, u^2, \ldots, u^n\right]. \qquad (1)$$

where u^j is the column vector from jth sequence. The eigenvectors of the correlation matrix $L = AA^T$ are defined as

$$L\phi_i = \lambda_i \phi_i \;\; 1 \leq i \leq n$$

where ϕ_i is the eigenvector and λ_i is the corresponding eigenvalue. Since the eigenvectors ϕ_i are of the same dimension as the lip sequences, we call them *eigensequences.*

The matrix L is a $MNP \times MNP$ matrix, which is exceedingly large, even for small M,N and P. However, the eigenvectors, ϕ_i, can be derived through computing the eigenvectors of the smaller matrix $C = A^T A$.

In equation 1 above, we have assumed that all u^j vectors, hence sequences, are of the same length. Since this can not be guaranteed in the real world, we apply a warping algorithm [9] to obtain sequences of equal length.

Any unknown sequence, u^x, can be represented as a linear combination of eigensequences as follows:

$$u^x = \sum_{i=1}^{n} a_i \phi_i. \qquad (2)$$

The linear coefficients, a_i, can be computed by finding the dot product of vector u^x with the eigensequences as:

$$a_i = u_x^T . \phi_i \quad 1 \leq i \leq n \qquad (3)$$

4 Model Generation and Matching

In our approach, several training sequences for each spoken letter are used to compute eigensequences, and the Q most significant eigensequences are selected and used as the model for that letter. Assume that we are given a novel sequence, representing an unknown spoken letter. In order to recognize this sequence, we project it on the individual eigenspace represented by the model (by computing the linear coefficients, a_i's), then compute the energy ratio (described below) for each projection. The model corresponding to the highest projection energy ratio indicates the possible match.

Suppose the set of the most significant eigensequences for letter ω is the set $\{\phi_1^\omega, \phi_2^\omega, \ldots, \phi_Q^\omega\}$, where the superscript denotes the letter, and the subscript denotes the eigensequence number. To generate the model, one training sequence for each letter is selected as a reference, and the remaining training sequences for that letter are warped to the reference sequence so that all of them are of the same length.

The projection of a novel sequence, u^x, on the eigenspace of letter ω is given by:

$$a_i^\omega = u_x^T . \phi_i^\omega, \quad 1 \leq i \leq Q; \quad \omega \in \{A, \ldots, Z\}. \qquad (4)$$

Note that before computing this projection, the novel sequence, u^x, is also warped to the reference sequence of letter ω in order to make it of equal length.

Due to slight head movement during the utterances, the lips in the novel sequence may be spatially misaligned with the reference sequence. We compensate for this through frame by frame registration between the two sequences. The frame-size of the original novel sequence is vertically greater (by a margin of 40 pixels) than that of the reference sequence. Through the criterion of maximum correlation, a part, which is of the same size of the reference frame, in each novel frame is selected to represent the novel sequence for actual projection.

Let the energy of projection of u^x on the eigenspace of letter ω be $\sum_{i=1}^{Q}(a_i^\omega)^2$, and the energy of u^x is $||u^x||$. We will use the ratio of these two energies, E^ω, defined below, as the measurement for matching.

$$E^\omega = \frac{\sum_{i=1}^{Q}(a_i^\omega)^2}{||u^x||}. \qquad (5)$$

For a perfect match, this ratio will be close to 1.

The definition of energy ratio in (5) is equivalent to the definition of normalized distance. Suppose we use all the eigensequences, then the novel sequence u^x can be expressed as:

$$u^x = \sum_{i=1}^{n} b_i \phi_i, \qquad 1 \leq i \leq n, \qquad (6)$$

where a_i's and b_i's are related as follows:

$$a_i = \begin{cases} b_i & \text{if } 1 \leq i \leq Q \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

Now, consider the normalized distance between u^x and its projection

$$D = \frac{\sum_{i=1}^{n}(b_i - a_i)^2}{\sum_{i=1}^{n} {b_i}^2}, \qquad (8)$$

which is equivalent to

$$D = 1 - \frac{\sum_{i=1}^{Q} {a_i}^2}{||u^x||} \qquad (9)$$

where $||u^x|| = \sum_{i=1}^{n} {b_i}^2$. Consequently, minimizing the normalized distance D is equivalent to maximizing the energy ratio, E^ω, defined in equation 5.

It is important to use the energy ratio or *normalized* distance in our case. As noted earlier, spatial registration has been applied to the novel sequences and the part of the (spatial) sequence used by the projection may vary with each mapping. This change needs to be compensated for by the normalization.

5 Modular Vs. Global Eigenspaces

In the space of all possible sequences, the lip sequences map to the clusters of individual letters. The task of lipreading then becomes determining which cluster an unknown sequence belongs to. We can use two methods of eigen representation. One method is to compute the eigenvectors of the entire space (*global eigenspace*) and discriminate the lip patterns by the distance to the respective cluster centers. The other is to use the *modular eigenspace*, in which the principal eigenvectors which give the most compact description of individual clusters can be constructed, and the distance from the input to the subspaces spanned by the principle eigensequences can be used.

We use modular eigenspaces in our approach, that is, separate eigensequences are computed for each spoken letter. While the global approach would use training sequences of all letters to compute global eigensequences. As noted earlier, before computing eigensequences, we must convert all the sequences to some fixed length. An important advantage of the modular eigenspace is that sequences for construction of each model are only warped among that group. Whereas in the global approach, it is difficult to select any reference letter to which all other sequences can be warped, because the sequences significantly differ from each other.

This issue of modular vs. global has also been addressed in the context of face recognition using static images. As pointed out by Pentland et al. [7], the relevant analogy is that of modeling a complex distribution by a single cluster model or by the union of several component clusters. Naturally, modular representation can yield a more accurate representation. Pentland et al's experimental results show a slightly superior performance obtained by the modular eigenspaces.

6 Extracting letters from connected sequences

The approach used in this paper treats each spoken letter as a basic unit for recognition. It is assumed that the lip movements for a given letter can be expected to follow similar spatiotemporal patterns. Consequently, a good method for automatically isolating and extracting letters from a continuous sequence is needed for successful recognition.

For simplicity, we assume that our task is to recognize independent letters from lip sequences. The speaker is required to begin each letter with the mouth closed, a constraint which was enforced with no difficulty during the experiments. The separation of the letters is based on the temporal variation of the sequence. This is determined by computing the *average absolute intensity difference function*, $f(n)$, as defined below:

$$f(n) = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} ||I_n(x,y) - I_{n-1}(x,y)|| \qquad (10)$$

From the plots of the average frame difference function, f, for a connected sequences, we found that the value of f during the articulation intervals is not necessarily greater than that during the non-articulation intervals, so separation of letters by using direct thresholding will not succeed. However, we note that the articulation intervals in this function correspond to clusters of big peaks and the non-articulation intervals correspond to the valleys between peaks, which may also have small local peaks.

Based on the waveform analysis, our approach begins with separating those clusters of peaks. First, the frame difference function, f, is smoothed to obtain function g. Then the global valleys are detected in g. These valleys occur between two consecutive letters. For each valley in g, starting from the frame number corresponding to the location of a valley in g, the hillside on the left and the hillside on the right in f, where f crosses a preset threshold, are identified. Next the first valley on left of right hillside, and the first valley on the right of left hillside in f are determined. The left valley is the ending of a previous letter, and the right valley is the beginning of the next letter. The threshold, T, used for determining hillsides in f should satisfy the following constraint:

$$\max_i(p_I(i)) < T \leq \min_j(p_L(j)),$$

where, p_I is the value of a local peak in the non-articulation interval and p_L is value of a (left-most and right-most) outer-most peak during the articulation. Since the outermost peaks usually are prominently high, a large margin can be allowed for the setting of T.

7 Warping

Warping is used twice in our method for lipreading. First, during the generation of model eigensequences, second during the matching of a novel sequence with the model eigensequence. In this section, we briefly describe warping. This temporal warping of two sequences uses the Dynamic Programming Algorithm of Sakoe and Chiba [9]. The columns of each frame of a sequence are concatenated to form one vector, and a sequence of vectors is created. For each pair of sequences we have:

$$A = [a_1, a_2, \dots, a_i, \dots, a_I]$$
$$B = [b_1, b_2, \dots, b_j, \dots, b_J]$$

where a_n is the n^{th} vector of sequence A, and b_n is the n^{th} vector of sequence B.

The algorithm employed uses the *DP-equation*in symmetric form with or without slope constraint. Without the constraint, $g(i,j)$ is computed as follows:

Initial Condition:

$$g(1,1) = 2d(1,1)$$

where $d(i,j) = ||a_i - b_j||$.

The *DP-equation*:

$$g(i,j) = min \begin{bmatrix} g(i,j-1) + d(i,j) \\ g(i-1,j-1) + 2d(i,j) \\ g(i-1,j) + d(i,j) \end{bmatrix}$$

The minimum equation used for the calculation of g at point (i,j) gives the path from the previous point to that point, thus creating a path from $(1,1)$ to (I,J). Each point on this path indicates which frames from the input sequence match to frames in the reference sequence, which creates a warped sequence that uses the frames of the input sequence and is the same length as the reference sequence. If several frames from the input sequence are matched to one frame in the reference sequence, the average of those frames is used as the result. Conversely, if one input frame is mapped to multiple reference frames, copies of that single frame are used.

8 Results

We have performed experiments with sequences of ten spoken letters (A–J). In one test, five sequences of each letter were digitized. Three sequences out of each five (seq-1, seq-2, and seq-3) were used as a training set to generate the model eigensequences, and the method was tested on recognizing the two remaining sequences (seq-4 and seq-5). Images were collected at a rate of 15 frames per second. One person supplied all the sequences. The sequences were taken with good lighting conditions. The resulting images were then cropped from 640 × 480 to 220 × 180 centered around the lips.

During model generation, three training sequences of each letter were first warped to a selected reference using the dynamic time warping method [9] without constraint. Next, the eigensequences were computed (which will be shown in the paper). During the recognition, seq-4 for a given unknown letter was warped to each of the ten model letters for possible match. Then, energies were computed using equation 5. This process was repeated for all ten unknown letters in seq-4. The recognition rate was 100% for sequence seq-4, and 90% for sequence seq-5.

We have also experimented with two connected sequences containing letters A,B,C,and D. First, the method discussed in section 6 was used to isolate spoken letters. Then, the extracted sequence corresponding to each letter was matched with the models as discussed above. The recognition rate was 100% for both sequences.

9 Conclusions

We presented a method for lipreading which uses eigensequences. In our approach, gray level values of all the pixels in all frames in a sequence representing a spoken letter are taken as a whole vector. Several such vectors corresponding to the training sequences are used to compute eigenvectors (eigensequence), for each spoken letter. The recognition of an unknown sequence representing a spoken letter is performed by measuring the ratio of energy when the sequence is projected to the model eigenspace over the energy of the sequence.

Future work will include the experimentation of the proposed method with more training and test sequences. Recognition of other letters “K” to “Z”, and digits “0” to “9” will also be performed. We expect the main idea can be extended to the general problem of ”motion based recognition” where spatiotemporal variation of the objects is involved. Since the proposed spatiotemporal eigen decomposition results in a very compact representation, it may also be useful for video signal compression.

References

[1] C. Bregler and Y. Konig. Eigenlips for Robust Speech Recognition. 1994.

[2] K. E. Finn and A. A. Montgomery. Automatic Optically-Based Recognition of Speech. *Pattern Recognition Letters*, 8:159–164, 1988.

[3] Alan Jeffrey Goldschen. Continuous Automatic Speech Recognition by Lipreading. Technical report, George Washington University, School of Engineering and Applied Science, 1993.

[4] M. Kirby, F. Weisser, and G. Dangelmayr. A Model Problem in the Representation of Digital Image Sequences. *Pattern Recognition*, 26(1):63–73, 1993.

[5] K. Mase and A. Pentland. Lip Reading: Automatic Visual Recognition of Spoken Words. Technical Report 117, M.I.T. Media Lab Vision Science, 1989.

[6] Murase, H. and Nayar, S. Illumination planning for object recognition in structured environment. In *IEEE CVPR-94*, pages 31–38, 1994.

[7] Pentland, A., Moghaddam, B., Starner, T. View-based and modular eigenspaces for face recognition. In *IEEE CVPR-94*, pages 84–91, 1994.

[8] E. D. Petajan, B. Bischoff, D. Bodoff, and N. M. Brooke. An Improved Automatic Lipreading System to Enhance Speech Recognition. In *SIGCHI '88: Human Factors in Computing Systems*, pages 19–25, October 1988.

[9] H. Sakoe and S. Chiba. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-26(1):43–49, February 1978.

[10] Turk, M., and Pentland, A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, pages 71–86, 1991.