VISUALLY RECOGNIZING SPEECH USING EIGENSEQUENCES

NAN LI, SHAWN DETTMER AND MUBARAK SHAH
Computer Vision Lab
Computer Science Department
University of Central Florida
Orlando, FL 32816

1. Introduction

Humans have the very complex ability to interpret facial expressions, gestures, even the so called "body language". Hearing-impaired people further develop this ability since most of them can perform some lipreading and/or understand sign language. It is well known that the visual information about speech through lipreading is very useful for human speech recognition. Humans use visual information to enhance our speech recognition, even when the available visual signal is noisy, distant or incomplete.

It is essential for computer systems to possess the ability to recognize meaningful gestures and lip movement if computers are to interact naturally with people. Currently, human-computer interface is mainly through a keyboard and/or mouse. Physically challenged people may have difficulties with such input devices and may require a new means of entering commands or data into the computer. Gesture, speech, and touch inputs are a few possible means of addressing such users' needs to solve this problem. For example, using Computer Vision, a computer can recognize and perform the user's gesture and vocal commands thus alleviating the need for a keyboard. Applications for such a vision system are the remote control of a robotic arm, guiding a remote computer presentation system, and executing computer operational commands such as opening a window or executing a program.

Lipreading is a very difficult task, especially since certain phonemes can appear visually identical (phonemes are minimal meaningful units of sound from which two words can be distinguished). For instance, the phonemes "b", "p", and "m" sound different but look the same when spoken [2].

Acoustically-based automatic speech recognition (ASR) is still not completely speaker independent, is limited in vocabulary and is sensitive to noise [2]. Combining acoustic and visual speech recognition is one possibility to better achieve lipreading capability.

The task of lipreading using computers is a complicated one. Many ideas and methods have been put forward. The general problem of lipreading remains unsolved. 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 (eigensequence), for each spoken letter. The recognition of an unknown sequence representing a spoken letter is performed by computing the ratio of the energy of projection of the sequence on the model eigenspace and the energy of the sequence. For a perfect match, this ratio tends to 1.

The success of any pattern recognition method usually depends on two stages: i) feature selection and extraction which results in easily separable clusters of patterns with minimal influence from class independent factors. ii) correct and efficient partitioning of the pattern clusters. In the eigenvector based work of face recognition, image intensity is just taken as the feature and eigenvectors are derived for a compact representation of the face patterns. Classification is performed with ease in a much lower dimensional space.

In the case of lips sequences, the samples may contain global head movements and the change in the duration of the articulations. To reduce those class independent variations spatial registration of the mouth position and temporal warping of the sequence are introduced as the the first step during training and recognition. In our method, we choose for each letter a sequence sample as the reference. The training sequences of individual letters are registered and warped against the corresponding reference sample. This creates well matched groups of sequences for each letter and those in each group have the same length as the reference sample. For an efficient recognition, instead of representing all the sequences of all possible lengths by a common basis, we use the principal eigenvectors derived from each group of training sequences as the model for the corresponding letter. This can be equivalent to characterizing the letter by a number of major components common to those sequences. Recognition is based on to what extent those components are contained in a novel sequence. The use of major components for modeling is a non-linear procedure in which noise and minor difference between the samples of the same group can be discarded as minor components so that they can not influence the decision.

Our approach is based on the demonstrated success of the eigenvector

approach using static images for face recognition problem [10, 7] and similar approach for illumination planning [6]. Turk and Pentland [10] decompose face images into a small set of characteristic feature images called "eigenfaces", which are essentially 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, then classifying face by comparing its position in face space with positions of known individuals. Murase and Nayar [6] address the problem of illumination planning for object recognition. For each object, they obtain a large number of images by varying pose and illumination. Images of all objects, together, constitute the planning set. The planning set is compressed using the K-L transform to obtain a low-dimensional subspace.

Instead of using the eigenvectors of a set of still images, we use *eigensequences* of a spatio-temporal sequence of images for the lipreading problem. We believe that lip movements for the same letter are expected to follow the similar spatio-temporal patterns. Therefore, eigensequences are suitable for the lipreading problem. Previously, separate spatial and temporal eigen decompositions have been used [4]. Since lip movement is essentially spatiotemporal in nature, to exploit this statistical redundancy in an integral way, in this paper we use the spatiotemporal eigen decomposition, in which the set of eigenvectors span the space of all possible sequences.

In order to recognize continuous utterances, we have developed a method for extracting letters from connected sequences. Our method uses the average frame difference function of a sequences and extract subsequences corresponding to individual letters by detecting the beginnings and endings of letters. This detection, in turn, is based on the peaks and valleys in the smoothed version of average frame difference function.

We have experimented with several sequences of English alphabet letters "A" to "J", and obtained very encouraging results. Since, in real life we speak the same letter with different speeds at different occasions, these sequences were variable in length. We use dynamic time warping to align sequences to a fixed length. Our eigensequence based approach for lipreading is very simple and straightforward; the major computation during recognition is simple dot product.

2. Related Work

In Petajan et al. [8], the lipreading task is performed by using the mouth opening area of a speaker to create a codebook. The mouth window is located by tracking the nostrils. The mouth image is then binarized, then a threshold is applied, so that only the mouth opening created a dark area. This large set of mouth images was reduced by clustering to about 255

clusters. A representative of each cluster is stored in a codebook of mouth images, which are ordered by increasing size of dark area, and identified with an index value. Once the mouth images codebook has been created, an inter-cluster distance table is computed, for faster computation during the matching process. The models of spoken words (the spoken letters and digits from zero to nine) are stored and vector quantized. Vector quantization 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. The model with smallest distance represents the sample.

In Finn and Montgomery's approach [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. The data were normalized relative to time and overall amplitude of distance measurements. The recognition consisted of computing a total root mean square value between two utterances: the model with smallest difference was considered the correct model.

A different scheme was developed by Mase and Pentland [5]. They observed that the most important features that affect mouth shape relate to the elongation of the mouth, and to the mouth opening, affecting upper and lower lips. Using optical flow, the authors expressed the two principal types of motions of the mouth as functions with respect to time: mouth opening $O(t)$ and elongation of the mouth $E(t)$. $O(t)$ and $E(t)$ are computed in each frame, then smoothed and normalized to a fixed variance. Word boundaries were taken to be times when $O(t) = 0$, i.e. when the mouth is closed, and can easily be located on the $O(t)$ plots. Templates were used for recognition, and matching was performed, 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 visual information from the oral-cavity shadow of the speaker for continuous speech recognition. His system uses Hidden Markov Models for distinguishing optical information. The HMM's were trained to recognize a set of sentences using visemes, trisemes (triplets of visemes), and generalized trisemes (clustered trisemes).

Bregler and Konig [1] created a hybrid system that uses both acoustic and visual information. They use a procedure similar to "snakes" and "de-

formable templates" to locate and track the lip contours. Then either the principal components of the contours are used, or the principal components of a gray level matrix centered around the lips are used. The latter uses matrix coding in an approach they termed "Eigenlips". Their work showed improvement for the combined architecture over just acoustic information alone in the presence of noise.

3. Eigensequences

Eigen decomposition represents the signal by a linear combination of a set of statistically independent orthogonal bases. This representation is most compact in the sense that, taking few number of the bases, the proportion of the signal energy projected on them is statistically maximum among all possible linear decompositions.

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 value of all pixels are then collected throughout the sequence in a long vector (of size MNP) as follows:

$$u^j = (I_1(1,1), \ldots, I_1(M,N), I_2(1,1), \ldots, I_2(M,N), \ldots, I_P(1,1), \ldots, I_P(M,N)),$$

where $I_n(x,y)$ is the value of the pixel at location (x,y) in frame n. Matrix A is then made from these vectors, u^j as follows:

$$A = \left[u^1, u^2, \ldots, u^s\right]. \tag{1}$$

The eigenvectors of a matrix $L = AA^T$ are defined as

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

where ϕ_i is the eigenvector and λ_i is the corresponding eigenvalue. The eigenvectors ϕ_i are called the *eigensequences.*

The matrix L is a $MNP \times MNP$ matrix, which is exceedingly large even for small M and N. However, eigenvectors, ϕ_i, can be computed from matrix $C = A^T A$, which is a smaller matrix, $s \times s$, where s is the number of sequences. Let α_i and λ_i respectively be the eigenvector and eigenvalue of matrix C, then

$$C\alpha_i = \lambda_i \alpha_i, \tag{2}$$

$$A^T A \alpha_i = \lambda_i \alpha_i. \tag{3}$$

Premultiplying the above by A, we get

$$AA^T(A\alpha_i) = \lambda_i(A\alpha_i). \tag{4}$$

Since $L = AA^T$, $A\alpha_i$ are the eigenvectors of L.

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:

$$u^x = \sum_{i=1}^{n} a_i \phi_i. \tag{5}$$

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

$$a_i = u_x^T . \phi_i \quad 1 \leq i \leq n \tag{6}$$

4. Model Generation and Matching

We use eigensequences to solve the lipreading problem. First, several training sequences for each spoken letter are used to compute eigensequences, and the Q (typically 3 to 5) most significant eigensequences are selected and used as a model in recognition. Assume that we are given a novel sequence, representing an unknown spoken letter. In order to recognize this sequence, we first determine its projection on the eigenspace of model letters (by computing the linear coefficients, a_i's), then compute the energy (described below) for each possible match. The letter with the highest energy is selected as a possible match.

In our approach, each model is a set of eigensequences, e.g., the model for letter ω is a set $\left\{\phi_1^\omega, \phi_2^\omega, \ldots, \phi_Q^\omega\right\}$, where the superscript denotes the letter, and the subscript denotes the eigensequence number. To generate a model, one training sequences for each letter is selected as a reference sequence, and the remaining training sequences for that letter are warped to the reference sequence to obtain fixed length sequences.

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

$$a_i^\omega = u_x^T . \phi_i^\omega, \qquad 1 \leq i \leq Q; \quad \omega \in \{A, \ldots, Z\}. \tag{7}$$

Note that before computing this projection, the novel sequence, u^x, is warped to the reference sequence of letter ω in order to make it of the same length. Due to slight head movement during the utterances, the frames in

the novel sequence may be spatially misaligned with respect to the reference model sequence. We compensate for this by registering the frames using maximum correlation. Correlation of a window in the novel frame is computed in the reference frame for possible displacement of ± 20 pixels in the y direction, and the best displacement is selected to register the frames.

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 for matching.

$$E^\omega = \frac{\sum_{i=1}^{Q} (a_i^\omega)^2}{||u^x||} \tag{8}$$

For a perfect match, this ratio will be equal to 1. E^ω is computed for all model letters, and the letter with the highest energy is selected as a match.

As stated earlier, we are using the Q most significant eigensequences in our method. If we use all the eigensequences, then the novel sequence u^x can be expressed as:

$$u^x = \sum_{i=1}^{n} b_i \phi_i \quad 1 \leq i \leq n. \tag{9}$$

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

$$a_i = \begin{cases} b_i & \text{if } 1 \leq i \leq Q \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

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

$$D = \frac{\sum_{i=1}^{n} (b_i - a_i)^2}{\sum_{i=1}^{n} {b_i}^2}, \tag{11}$$

which is equivalent to

$$D = 1 - \frac{\sum_{i=1}^{Q} {a_i}^2}{||u^x||} \tag{12}$$

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 8.

It is important to use the *normalized* distance in matching. As noted earlier, before a novel sequence is projected onto each model eigenspace, it is spatially registered to the reference sequence of that model. Therefore,

the part of the (spatial) sequence taking part in the projection is not fixed and may vary with each mapping. Therefore, the absolute distance is not suitable.

The recognition process, in summary, requires that, first, the models be generated. For model generation, we

- Warp all the training sequences for a spoken letter with respect to a reference sequence.
- Perform spatial registration using correlation.
- Represent each letter by its Q most significant eigensequences: $\left\{\phi_1^\omega, \phi_2^\omega, \ldots, \phi_Q^\omega\right\}$.

Then, for matching, the following steps are taken:

- Warp the unknown sequence with respect to the reference sequence of each model.
- Perform spatial registration.
- For each model, compute the ratio of the energy of the projection of the unknown sequence into the model's eigenspace, and the energy of the unknown sequence.
- Determine best match by the maximum of all ratios computed.

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 that of 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 are constructed, and the distance from the input to the subspaces spanned by the principal eigensequences is used.

We use modular eigenspaces in our approach, that is, separate eigensequences are computed for each spoken letter. 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.

6. Extracting letters from connected sequences

The approach used in this chapter 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 in the closed position, a constraint which was enforced with no difficulty during the experiments. The separation of the letters is based on the temporal variation, between successive frames. This is determined by computing the *average absolute intensity difference function*, $f(n)$, as defined below:

$$f(n) = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} ||I_n(x,y) - I_{n-1}(x,y)|| \tag{13}$$

Figure 1 shows the plot of the average frame difference function, f, for a connected sequence seq-a. From this plot, it is easy to see 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.

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 end 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 large, a large margin can be allowed for the setting of T.

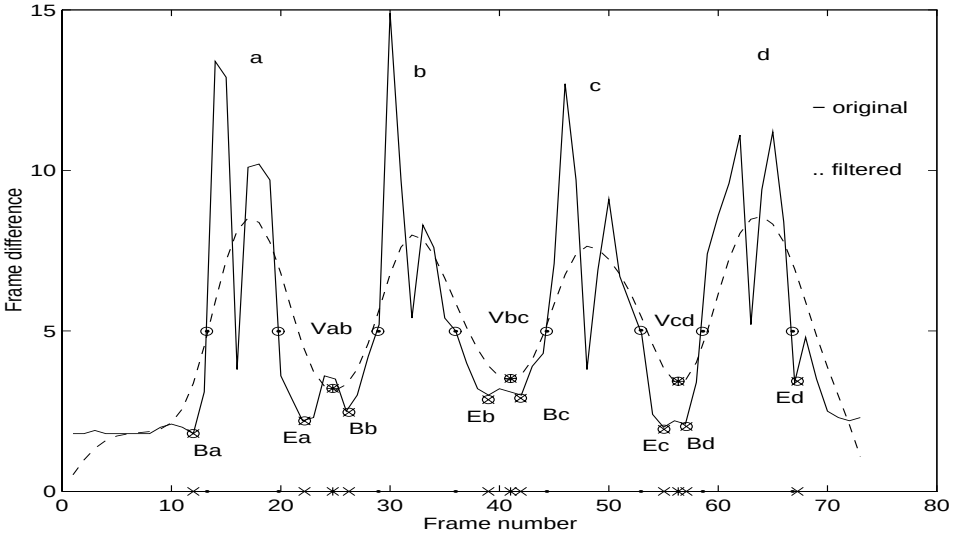

Figure 1. Plots of the frame difference function, f, and smoothed (filtered) version of f, g, for connected sequence seq-a. The valleys in g are shown as Vab, Vbc, Vcd. The beginning and endings of letters "A", "B", "C" and "D" respectively are shown as Ba, Ea, Bb, Eb, Bc, Ec, Bd and Ed.

The plots for f and g and detected beginning and end of letters in connected sequences a and b are shown in Figures 1 and 2.

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. 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. Thus for each pair of sequences we have:

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

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 a slope constraint of 1. Therefore, g(i,j) is computed as follows:

Initial Condition:

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

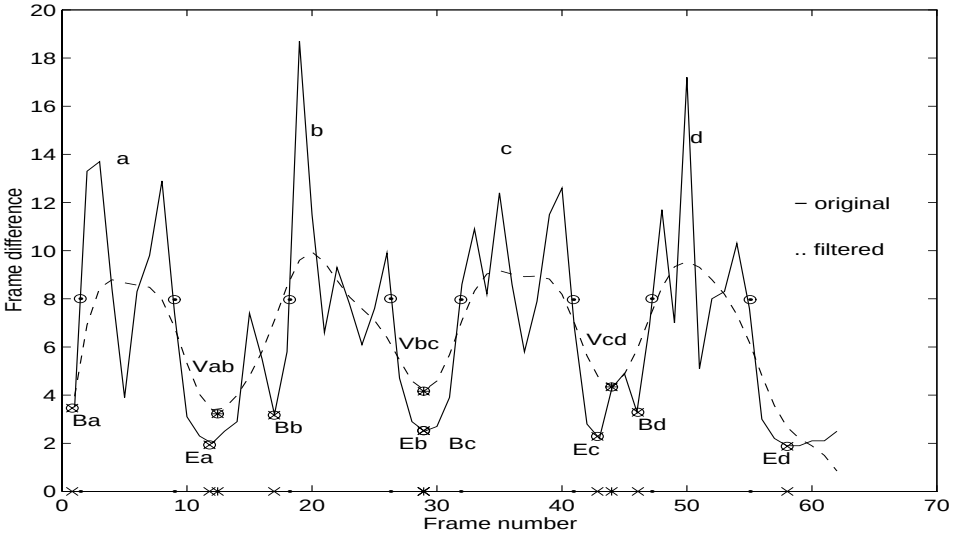

Figure 2. Plots of the frame difference function, f, and smoothed (filtered) version of f, g, for connected sequence seq-b. The valleys in g are shown as Vab, Vbc, Vcd. The beginning and ending of letters "A", "B", "C" and "D" respectively are shown as Ba, Ea, Bb, Eb, Bc, Ec, Bd and Ed.

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

The *DP-equation* we used:

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

The minimum equation used for the calculation of g at point (i,j) gives the path from the previous point to the current 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. Having a slope constraint of 1 allows for three possible cases for matching the lengths of the sequences. Two frames from the input sequence match to one frame in the reference sequence, in which case the two input frames are averaged to create one. There is a one-to-one correspondence, in which case the input frame is unchanged. Or, one frame from the input sequence matches to two frames in the reference sequence, in which case the input frame is just repeated.

8. Results

Our first experiments used sequences of ten spoken letters. (A–J). For each letter, five sequences were digitized. Three sequences (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 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 dynamic time warping method [9]. Then the eigensequences were computed. The eigensequences for letters "A" to "J" are shown in Figures 3-12. During the recognition, seq-4 (shown in Figure 13) for a given unknown letter was warped to each of the ten model letters for possible match. Then, energies were computed using equation 8. This process was repeated for all ten unknown letters in seq-4. The results for matching are summarized in Figure 17. The matching of seq-5 (shown in Figure 14) was performed as for seq-4, and the results are summarized in Figure 18. The recognition rate is 90% for both sequences.

We also experimented with two connected sequences shown in Figures 15, and 16. First, the method discussed in section 6 was used to isolate spoken letters. Then extracted sequence corresponding to each letter was matched with the models as discussed above. The results are summarized in the tables shown in Figures 19-20. The recognition rate is 100% for both connected sequences.

Note that the values of energy ratios shown in the tables in Figures 17–20 are densely centered around 1, therefore high precision is needed to distinguish between them. This can be easily improved by subtracting the average image from each frame of the sequences, including frames in the models and the unknown sequences. The subtraction will not alter the relative distance between the sequences but reduce their energy on the whole, so the dynamic range of the representative energy ratio will be increased.

In our next set of extensive experiments, we studied the effect of reduced resolution on our method. In this case, we also used sequences of ten spoken letters (A–J). For these tests, we digitized twenty connected sequences, where each sequence was of the letters A through J. The average frame difference function was used to isolate the individual letters from these sequences. The images were cropped as stated above, but during the warping, every image was reduced in size to 29×19. Ten sequences for each letter were used for model generation, and the remaining ten were used as

unknown sequences.

The results are shown in the table in Figure 21. Using only the highest ratios for matches, we had 10 matches out of a possible 10 for the letters "A", "B", "F", and "H". There were 8 of out 10 matches for letters "E", "G", and "J", while "I" matched 7 times, and "C" and "D" matched 6. Considering the second highest ratios, the letters "C", "D", and "I" had 3 additional matches, and "E", "G", and "J" had 1. And going to the third highest ratios, the letters "C" and "E" had 1 additional match. Thus, considering only the highest ratios as matches, we achieved an 83% correct recognition rate. Considering the two top ratios for matching, that rate jumps to 95%, and allowing for the three top ratios gives a recognition rate of 97%.

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 put in one large 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 computing the ratio of energy of projection of the sequence on the model eigenspace and the energy of the sequence.

Future work will include the experimentation of the proposed method with sequences of other letters "K" to "Z", and digits "0" to "10". Since the proposed spatiotemporal eigen decomposition results in a more compact representation, it can also be used to solve the image compression problem.

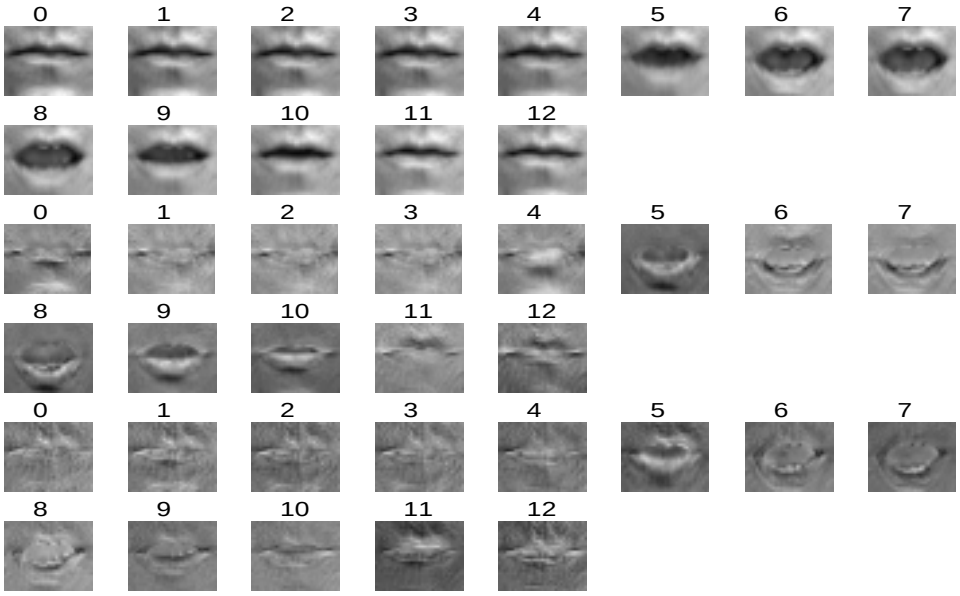

Figure 3. Three eigensequences of letter "A".

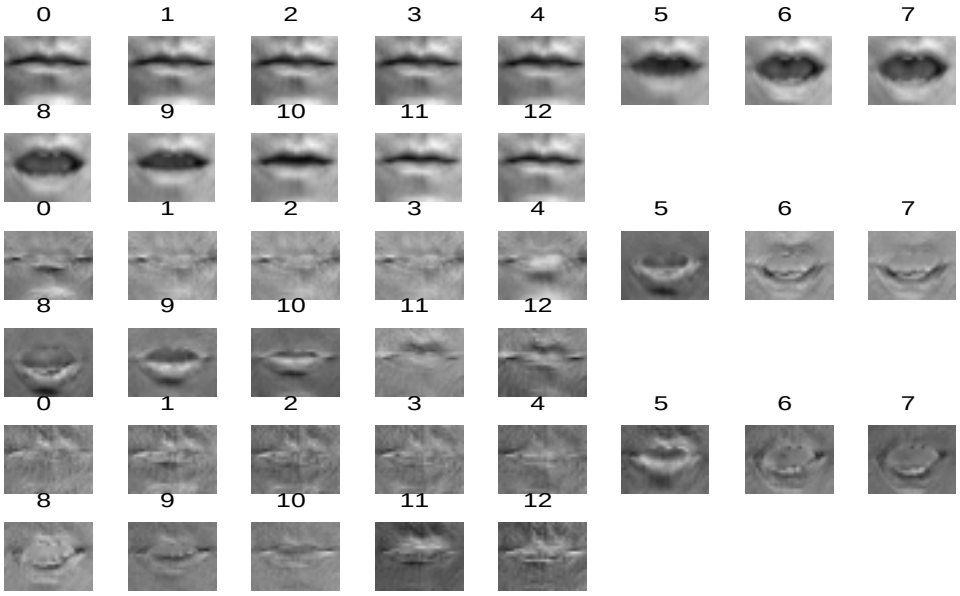

Figure 4. Three eigensequences of letter "B".

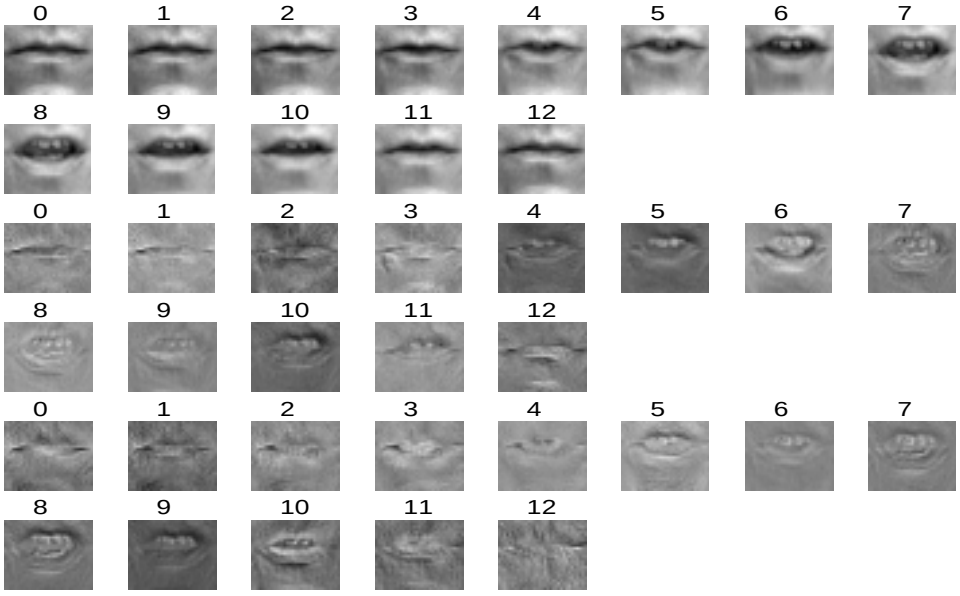

Figure 5. Three eigensequences of letter "C".

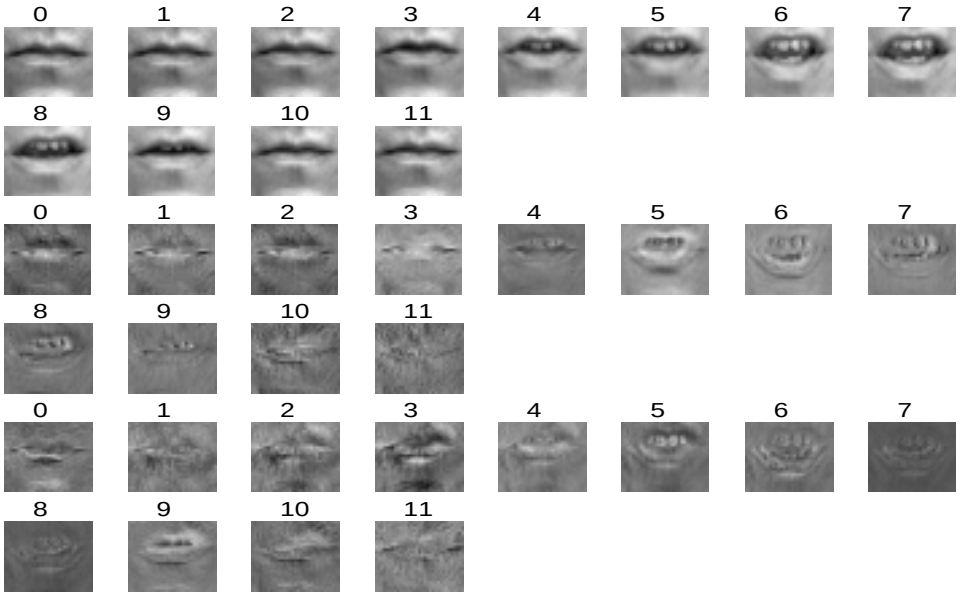

Figure 6. Three eigensequences of letter "D".

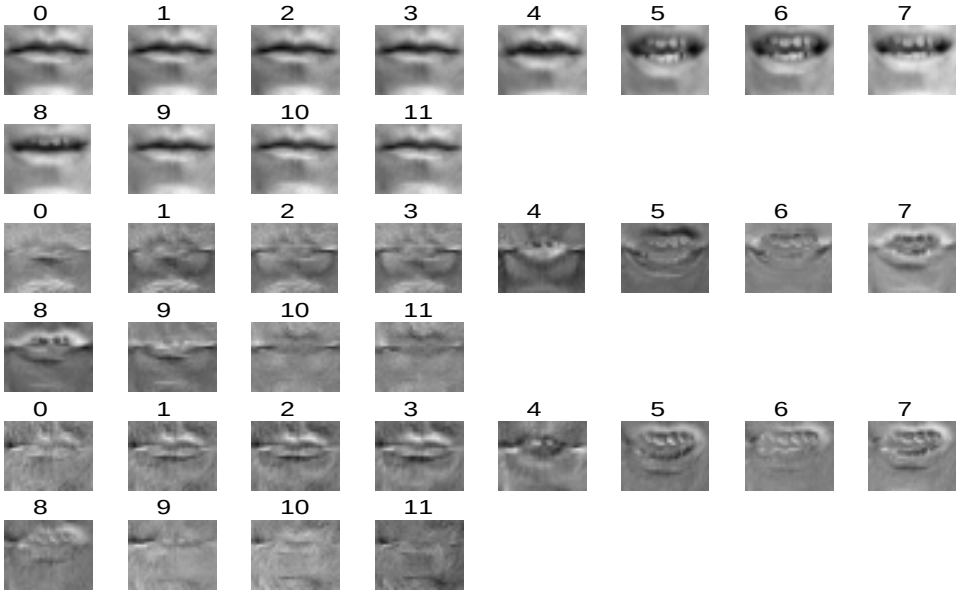

Figure 7. Three eigensequences of letter "E".

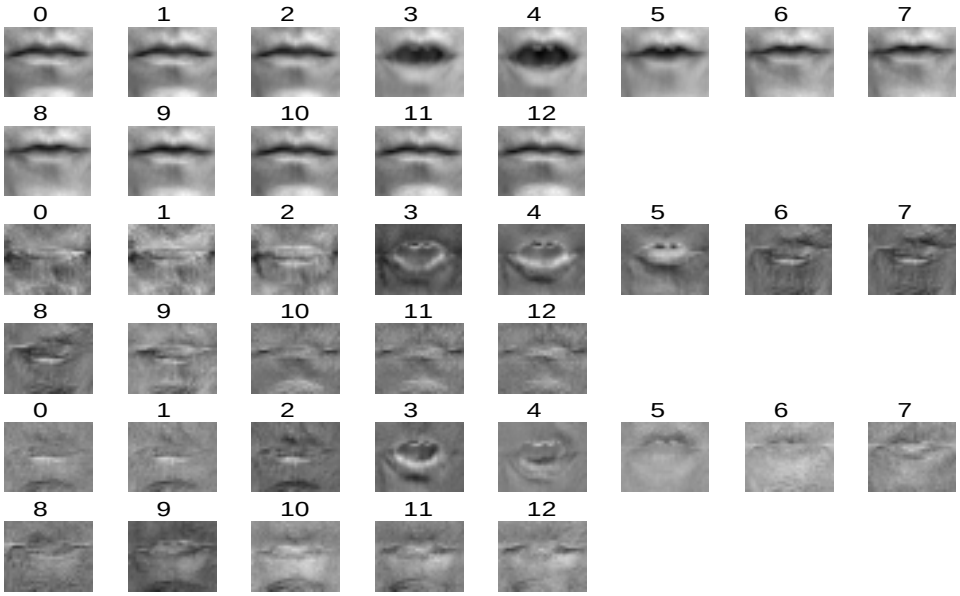

Figure 8. Three eigensequences of letter "F".

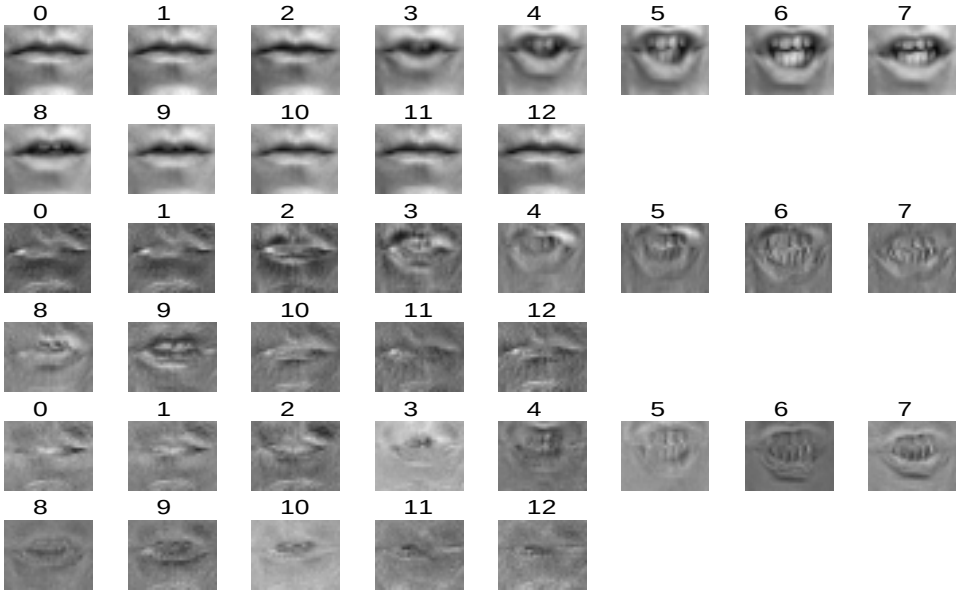

Figure 9. Three eigensequences of letter "G".

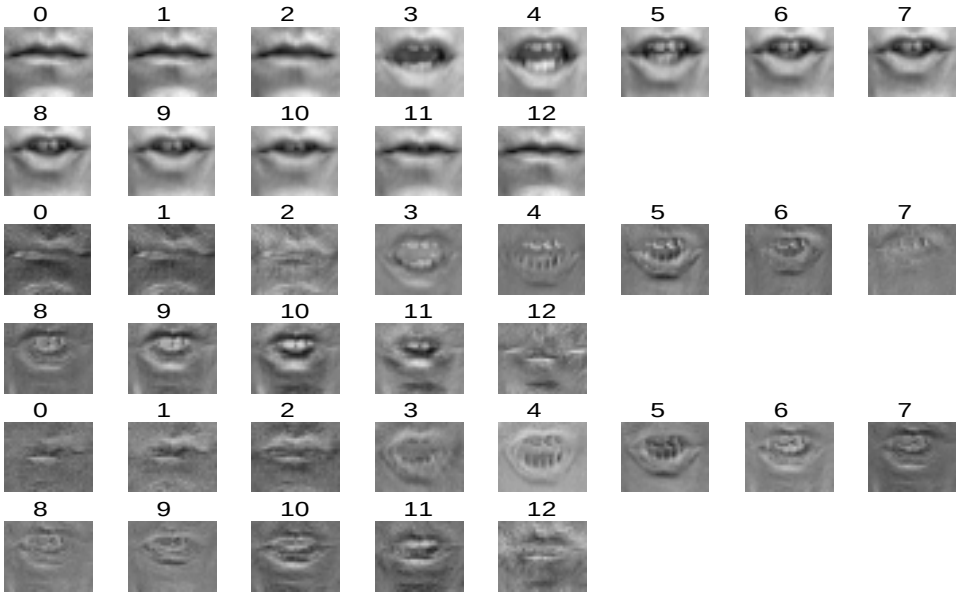

Figure 10. Three eigensequences of letter "H".

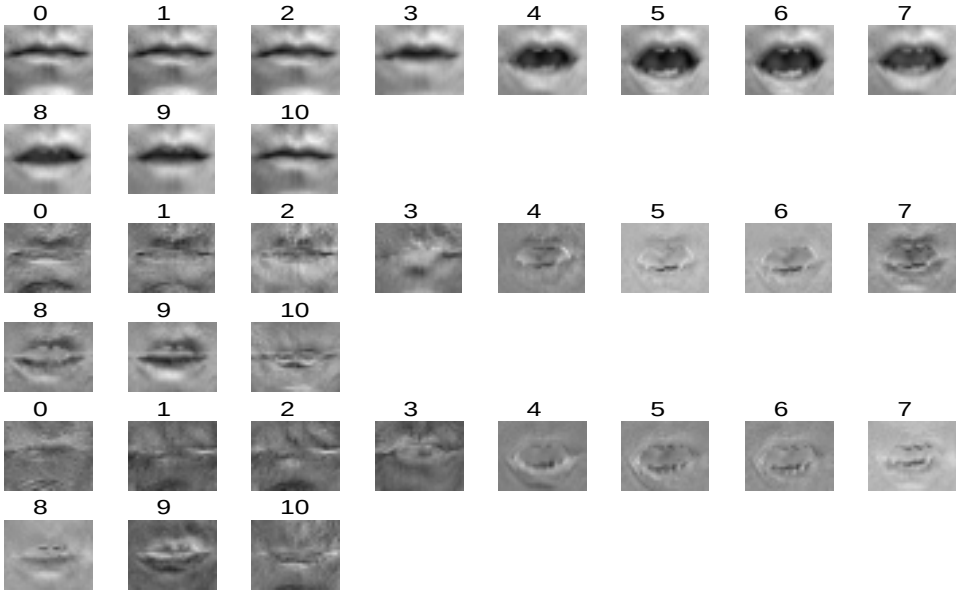

Figure 11. Three eigensequences of letter "I".

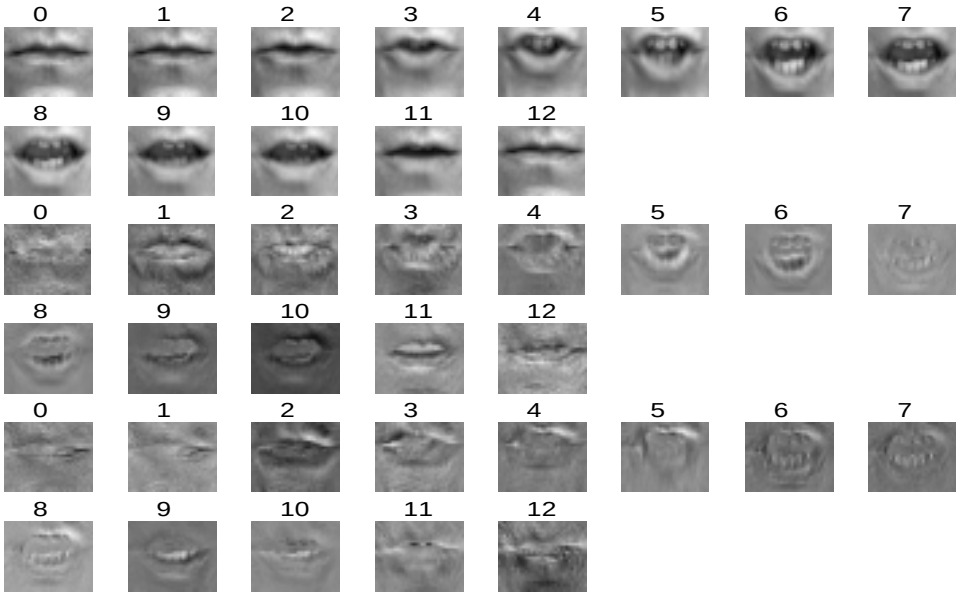

Figure 12. Three eigensequences of letter "J".

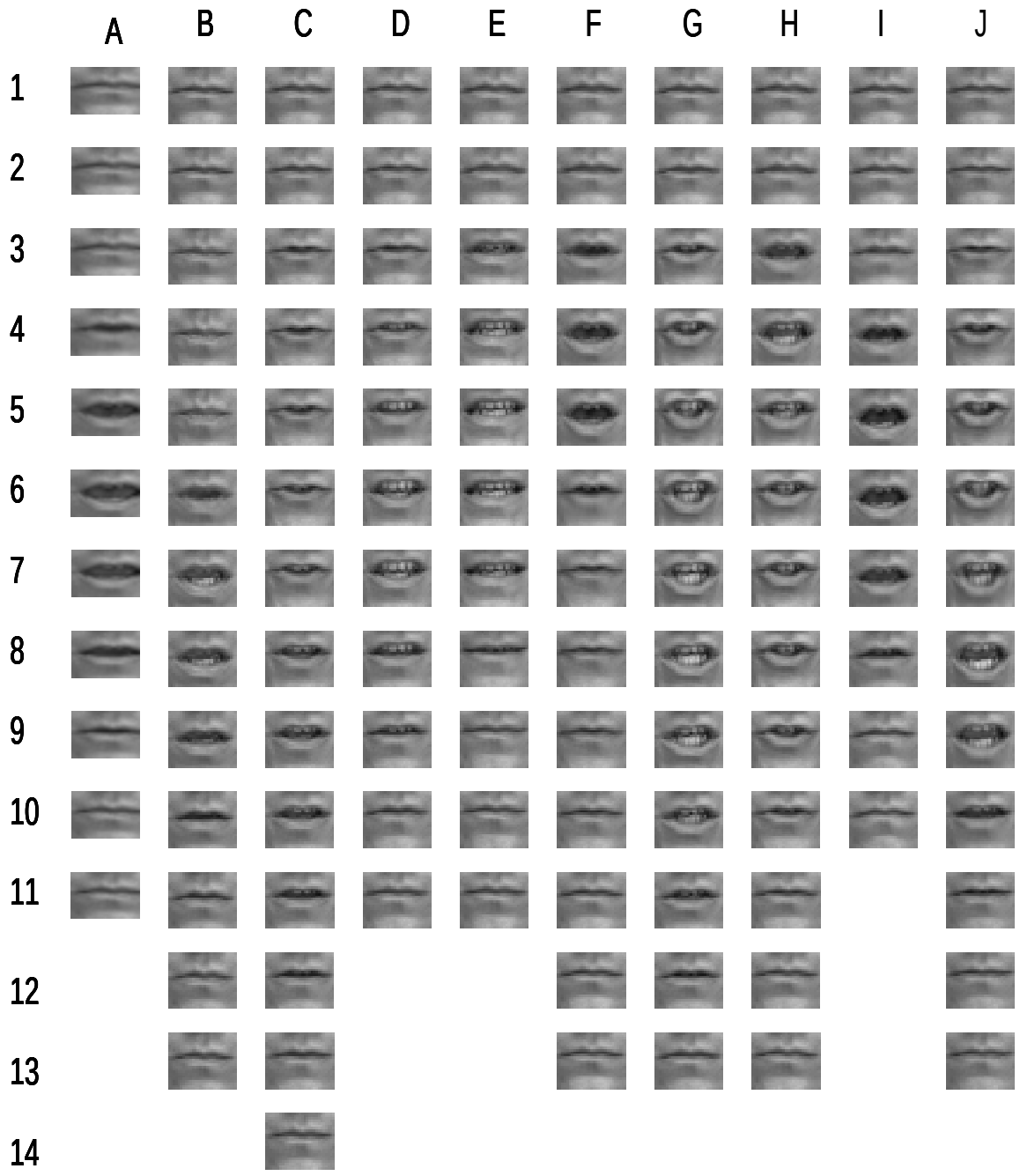

Figure 13. Sequence seq-4.

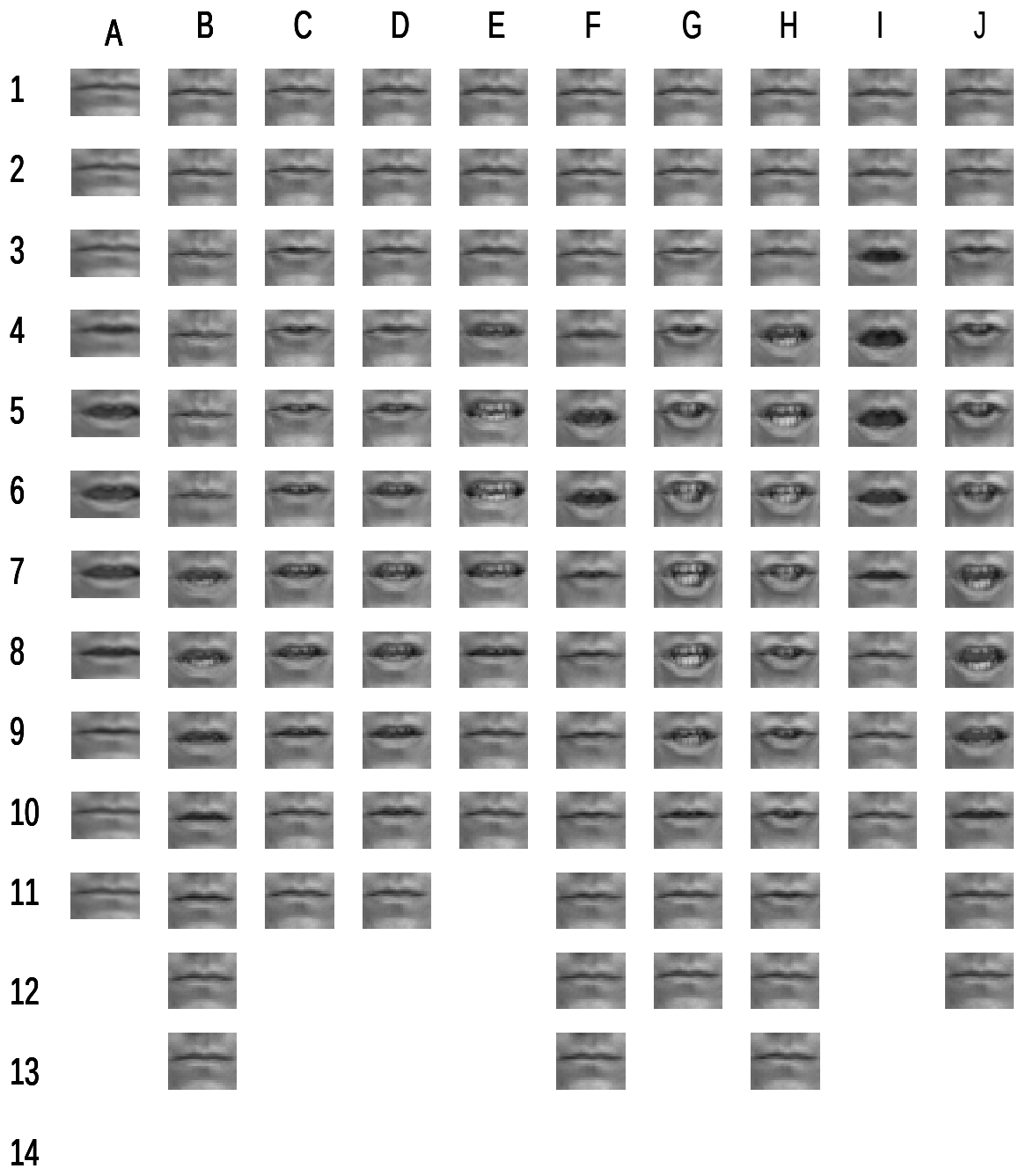

Figure 14. Unknown Sequence seq-5.

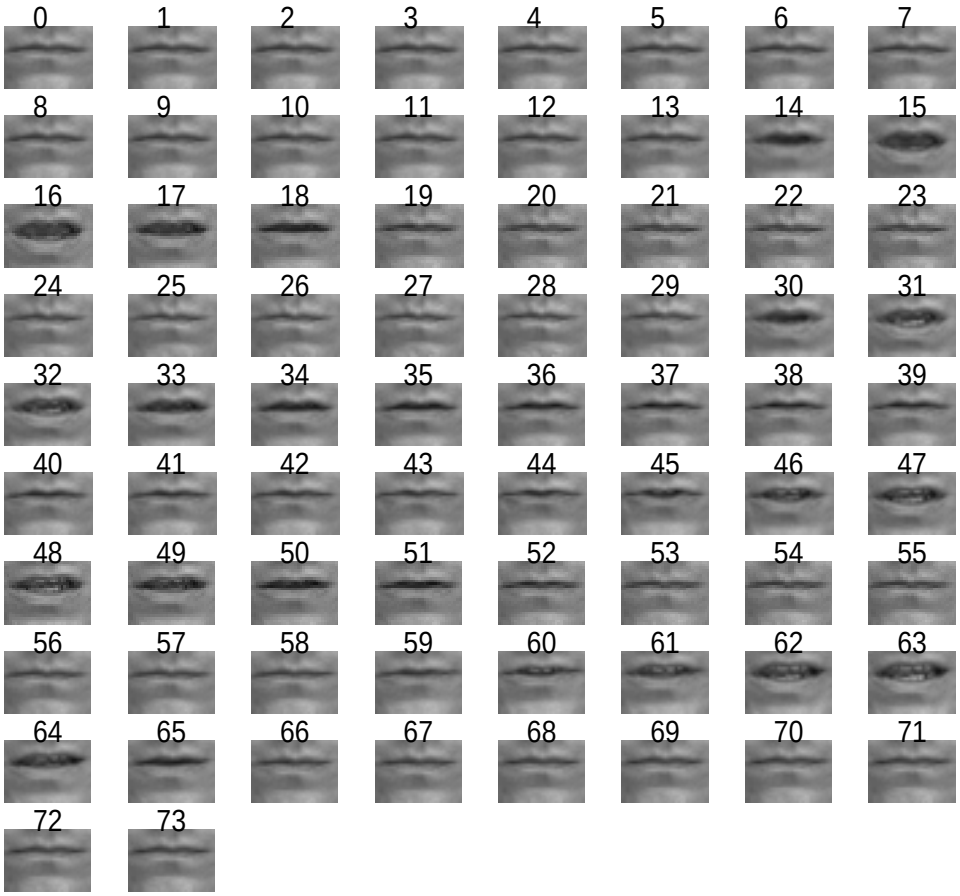

Figure 15. Connected sequence seq-a. This contains letters "A", "B", "C", and "D". The method discussed in this paper extracted four subsequences: frame 12–frame 22 ("A"), frame 26–frame 39 ("B"), frame 42–frame 55 ("C"), and frame 57–frame 67 ("D").

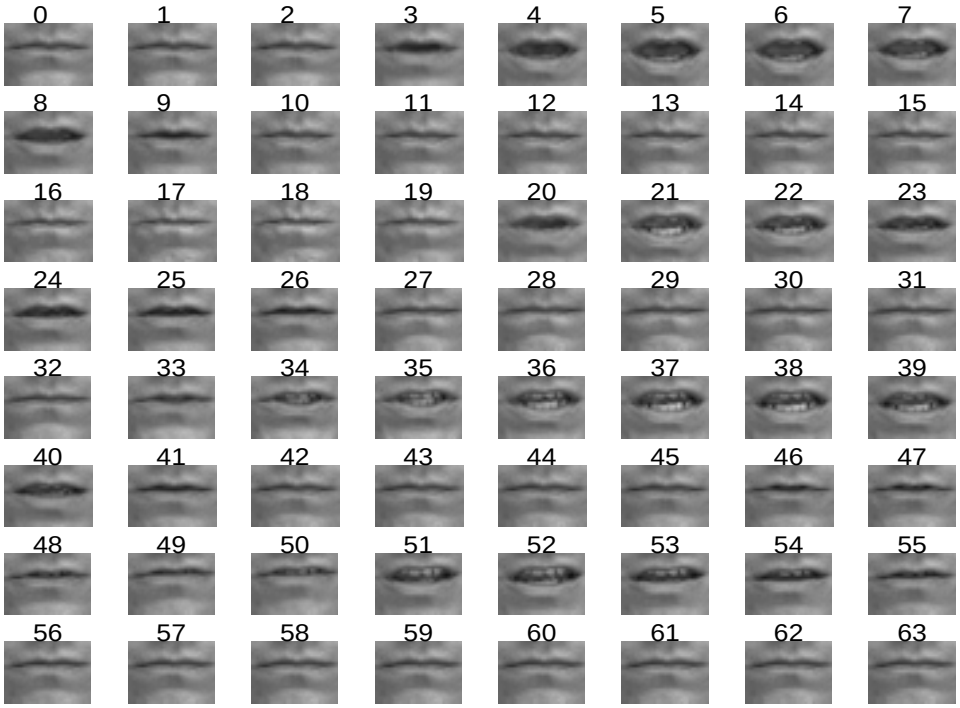

Figure 16. Connected sequence seq-b. This contains letters "A", "B", "C", and "D". The method discussed in this chapter extracted four subsequences: frame 1–frame11 ("A"), frame 17–frame 29 ("B"), frame 30–frame 43 ("C"), and frame 46–frame 58 ("D").

References

1. C. Bregler and Y. Konig. Eigenlips for Robust Speech Recognition. In *Proc. IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing*, 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.* PhD thesis, 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.

Model	Unknown A	Unknown B	Unknown C	Unknown D	Unknown E
A	0.9975	0.9905	0.9859	0.9495	0.9748
B	0.9894	0.9926	0.9890	0.9579	0.9789
C	0.9887	0.9889	0.9971	0.9540	0.9815
D	0.9851	0.9817	0.9902	0.9966	0.9900
E	0.9840	0.9843	0.9858	0.9914	0.9903
F	0.9931	0.9927	0.9929	0.9873	0.9865
G	0.9813	0.9786	0.9884	0.9836	0.9833
H	0.9792	0.9801	0.9829	0.9843	0.9831
I	0.9948	0.9862	0.9799	0.9673	0.9582
J	0.9831	0.9825	0.9879	0.9827	0.9753
Best match	A	F	C	D	E

Model	Unknown F	Unknown G	Unknown H	Unknown I	Unknown J
A	0.9933	0.9827	0.9884	0.9936	0.9846
B	0.9892	0.9889	0.9880	0.9845	0.9878
C	0.9904	0.9912	0.9906	0.9893	0.9921
D	0.9880	0.9902	0.9906	0.9858	0.9887
E	0.9820	0.9876	0.9874	0.9838	0.9860
F	0.9981	0.9914	0.9936	0.9945	0.9895
G	0.9827	0.9968	0.9816	0.9797	0.9902
H	0.9875	0.9869	0.9976	0.9819	0.9861
I	0.9911	0.9759	0.9881	0.9952	0.9797
J	0.9825	0.9885	0.9819	0.9848	0.9937
Best match	F	G	H	I	J

Figure 17. Results for sequence seq-4. The entries in the table are the energy ratios, the perfect match has ratio equal to 1. Recognition is 90%. Every input letter, except for letter "B", was recognized correctly.

Model	Unknown A	Unknown B	Unknown C	Unknown D	Unknown E
A	0.9965	0.9910	0.9809	0.9882	0.9773
B	0.9893	0.9948	0.9882	0.9889	0.9882
C	0.9889	0.9864	0.9965	0.9942	0.9913
D	0.9852	0.9804	0.9941	0.9967	0.9927
E	0.9857	0.9863	0.9875	0.9893	0.9944
F	0.9939	0.9925	0.9932	0.9811	0.9919
G	0.9821	0.9761	0.9779	0.9883	0.9801
H	0.9778	0.9796	0.9895	0.9901	0.9859
I	0.9953	0.9873	0.9659	0.9774	0.9649
J	0.9842	0.9836	0.9879	0.9879	0.9846
Best match	A	B	C	D	E

Model	Unknown F	Unknown G	Unknown H	Unknown I	Unknown J
A	0.9914	0.9841	0.9870	0.9948	0.9877
B	0.9867	0.9859	0.9882	0.9814	0.9854
C	0.9872	0.9906	0.9885	0.9886	0.9921
D	0.9781	0.9875	0.9884	0.9788	0.9878
E	0.9839	0.9836	0.9821	0.9825	0.9847
F	0.9962	0.9914	0.9917	0.9920	0.9915
G	0.9772	0.9745	0.9751	0.9771	0.9904
H	0.9809	0.9873	0.9949	0.9804	0.9825
I	0.9920	0.9773	0.9868	0.9949	0.9806
J	0.9774	0.9879	0.9813	0.9794	0.9933
Best match	F	F	H	I	J

Figure 18. Results for sequence seq-5. The entries in the table are the energy ratios, the perfect match has ratio equal to 1. Recognition is 90%. Every input letter except letter "G", which was matched to letter "F", was recognized correctly.

Model	Unknown A	Unknown B	Unknown C	Unknown D
A	0.9949	0.9846	0.9770	0.9770
B	0.9920	0.9928	0.9822	0.9818
C	0.9921	0.9873	0.9930	0.9907
D	0.9904	0.9878	0.9910	0.9926
E	0.9846	0.9800	0.9846	0.9875
F	0.9925	0.9901	0.9904	0.9815
G	0.9826	0.9819	0.9836	0.9838
H	0.9860	0.9839	0.9860	0.9831
I	0.9903	0.9710	0.9715	0.9650
J	0.9856	0.9821	0.9848	0.9817
Best match	A	B	C	D

Figure 19. Results for connected sequence seq-a. This sequence contained letters "A", "B", "C" and "D". First the subsequence corresponding to these letters were extracted using the method described in the chapter, then the extracted subsequence were matched with ten model letters. The recognition is 100%.

Model	Unknown A	Unknown B	Unknown C	Unknown D
A	0.9953	0.9856	0.9871	0.9833
B	0.9895	0.9927	0.9900	0.9847
C	0.9914	0.9844	0.9911	0.9901
D	0.9864	0.9825	0.9894	0.9921
E	0.9824	0.9779	0.9876	0.9884
F	0.9911	0.9884	0.9879	0.9895
G	0.9797	0.9760	0.9869	0.9856
H	0.9843	0.9790	0.9826	0.9853
I	0.9935	0.9797	0.9819	0.9670
J	0.9872	0.9840	0.9874	0.9840
Best match	A	B	C	D

Figure 20. Results for connected sequence seq-b. This sequence contained letters "A", "B", "C" and "D". First the subsequence corresponding to these letters were extracted using the method described in the chapter, then the extracted subsequence were matched with ten model letters. The recognition rate is 100%

	A	B	C	D	E	F	G	H	I	J
1st	10	10	6	6	8	10	8	10	7	8
2nd			3	3	1		1		3	1
3rd			1		1					

Figure 21. Results for reduced resolution series of tests using 10 unknown sequences. The numbers indicate the number of matches out of 10 that were correct. The first row shows results for the highest ratios. The second row shows the number of additional matches from the second highest ratios. And the last row shows the same for the third highest ratios. The success rate is 83% from the highest ratios, 95% from top two highest ratios, and 97% from top three highest ratios.