

Visual Speech Recognition Using Hidden Markov Models: CS280 Course Project

Kofi Boakye

May 18, 2005

1 Introduction

In many ways computer vision is concerned with replicating (and perhaps even enhancing) the performance of visual tasks humans are known to do. Some such tasks are object recognition and tracking, image segmentation, and gesture recognition. From these examples it is clear that computer vision requires the processing and modeling of both static and dynamic phenomena, and one dynamic phenomenon that is of interest is visual speech articulation. Lip-reading humans have demonstrated that it is possible (with a substantial amount of training) to recognize speech solely based on visual observation of the speech articulation. In addition, though, laboratory studies have shown that the phenomenon is an important source of information in face-to-face speech perception, particularly in the presence of noise. Both of these observations suggest that there are many possible applications of visual speech recognition. Audio speech recognition, which has received a lot of attention in the research community but still generally attains modest performance levels, could be augmented by visual speech recognition in many cases. This could make automatic annotation of speech in various domains more accurate and robust.

As a dynamic process, a model which takes into account sequential information is most appropriate for visual articulation. A popular choice for such a model is a Hidden Markov Model (HMM). An HMM seeks to provide a model for the underlying process generating a sequence of observations. The process is modeled as a sequence of states and their transitions, where the transition probability from the current state to the next state depends only on the current state (the Markovian assumption). HMMs have shown

good performance in audio speech recognition, making them an especially natural choice for its visual counterpart.

This paper describes a project which sought to analyze the use of HMMs for visual speech recognition. The specific task was speaker-independent, single-digit recognition using data from the Tulips1 corpus for audiovisual speech recognition compiled by Javier R. Movellan and described in [1]. For the project, a cross-validation approach was taken whereby the model used to test a digit for a speaker was trained using all examples of the digit not originating from the speaker. Different features (hand-generated contour features, raw pixels with PCA, and preprocessed image features) were analyzed and compared. In addition, the performance for varying HMM structures—specifically the number of states and the number of Gaussians per state—was explored.

The paper is organized as follows: Section 2 describes the corpus used; section 3 details the procedure, specifically the feature selection, model training, and model testing; section 4 gives the results for the various experiments; and section 5 presents final conclusions.

2 The Tulips1 Corpus

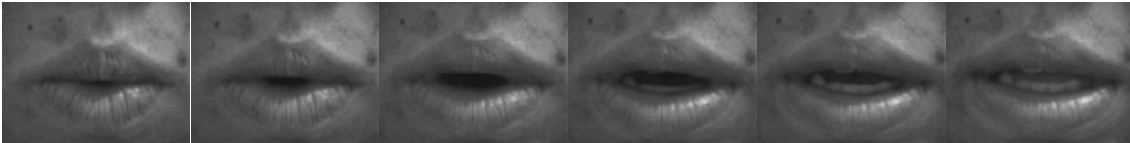


Figure 1: *Example sequence for the word “one” using images from the Tulips1 corpus.*

The Tulips1 corpus is a small audiovisual database consisting of 12 speakers (9 male and 3 female) saying the first four digits in English. Each speaker says each of these words twice, positioned in front of a video camera such that their lips are roughly centered on a 100x75 pixel image area. The images were digitized at 30 frames per second to 8-bit grayscale .pgm files. For the database, the video tracks were hand segmented by selecting a few relevant frames prior to the start and after the end of speech activity in the audio channel. The database contains a total of 934 images and consists of speakers with different ethnic origin, some with make-up or facial hair, and different illumination. It is easily obtainable via the Internet (<http://mplab.ucsd.edu/databases/databases.html>) and as such serves as a

very useful research resource. Both audio and video data is included, but for the purposes of the project only the video data was utilized. An example of the word “one” as a sequence of images from the database is shown in figure 1. In addition, statistics for the data set are given in table 1.

Digit	Mean	Std. Dev.
“One”	8.9	2.1
“Two”	9.6	2.1
“Three”	9.7	2.3
“Four”	10.6	2.2

Table 1: *Frame number statistics for Tulips1 utterances.*

3 Procedure

3.1 Feature Selection

To perform the recognition, a set of features derived from the data (i.e., the images) must first be selected. These features will then serve as observations for the training and testing of the HMM. For the project, three sets of features were chosen.

The first set consisted of contour features related to mouth and lip geometry. They are:

1. The width of the outer corners of the mouth
2. The height of the outer corners of the mouth
3. The width of the inner corners of the opening of the mouth
4. The height of the inner corners of the opening of the mouth
5. The height of the upper lip
6. The height of the lower lip

The above measurements are expressed in pixels. These representations were used by Anwar *et al.* in [2], and were provided in the Tulips1 database. Note that in order to get the measurements for the features, the images in the database were marked by hand, making the feature extraction non-automatic. It is, however, possible to automate this using additional search algorithms.

The second set of features were simply derived from the pixels in each frame. Each image matrix was “linearized” (i.e., made into a vector by concatenating successive rows) and Principal Component Analysis (PCA) was performed for dimensionality reduction. This is a common and quite straightforward method of obtaining features from the images. PCA was performed using different numbers of components to generate different feature sets and the relative performances are shown in 4.2).

The third and final set of features were obtained by preprocessing the images for dimensionality reduction. The steps were as follows:

1. Symmetry enforcement: Each raw image was symmetrized by averaging the left and right sides of the image pixel by pixel, using the vertical midline as the axis of symmetry.
2. Lowpass filtering and subsampling: The symmetrized image was filtered using a 9x9 Gaussian kernel with $\sigma = 1.5$. After filtering, the additional border pixels resulting from the edge effects of the convolution were removed. Subsampling by a factor of 5 was then performed.
3. Compression and linearization: Because the images are now symmetric, only half of each image is now needed for representation. This being the case, half of the image is discarded and the matrix containing the remaining pixels is linearized.

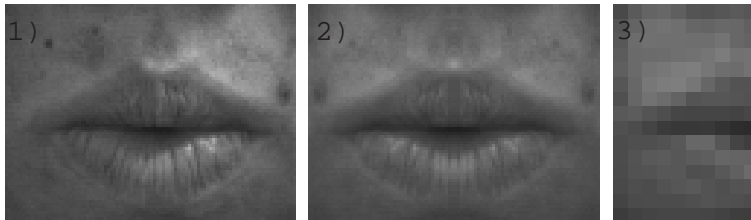


Figure 2: *Image preprocessing. 1) Raw image. 2) Symmetrized image. 3) Downsampled and compressed image.*

This preprocessing technique is discussed by Movellan in [1] and the steps are shown for an example image in figure 2. The resulting feature vectors contain 150 components (compared to 7500 in the original raw image). For some contrastive experiments, further dimensionality reduction using PCA was additionally applied to these features to examine the performance change.

In order to be used in the HMM utilities, the above features needed to be written to a special binary feature format. This, along with the processing to create the features (filtering, PCA, etc.), was done using Matlab. Additionally, all feature vectors were augmented by including their difference (delta) parameters as well. These represent a weighted difference of feature vector components within a symmetric window centered about the feature vector of interest. Delta features have been shown to yield significant improvements in performance because they help capture more dynamic information and, in the case of images, they are robust to changes in illumination.

3.2 Model Training and Testing

For model training and testing, a 12-way cross-validation approach was taken. That is, to perform recognition on a given digit for a given speaker, a model was trained using the examples of the digit provided by the 11 other speakers (two examples per speaker). This procedure was applied for all digits and all speakers.

All training and testing was performed using the HMM Toolkit (HTK) [3], a publicly available toolkit common to the audio speech recognition community. Prior to training the model, the prototype HMM structure must be defined. This consists of the number of HMM states, the types of permissible transitions (skips, self-loops, etc.), the number of Gaussian mixtures per state, the type of covariance matrices used (full or diagonal), along with a number of more subtle aspects. Some of these choices in design can significantly affect results and great care is taken to come up with good parameters. Often prior experience along with some limited exploration can result in good choices. For the project, some of the parameter space was explored by varying the number of Gaussians and HMM states in the first experiment. The fixed structure for the HMM was as follows: Left-to-right sequences with self-loops and no skips for the HMM state transitions, and diagonal covariances for the Gaussians. These settings are typical for audio speech recognition and it was hoped they would work well for the visual speech recognition case as well.

The training of each HMM occurred in two stages. In the first stage, the HMM was initialized through an iterative technique involving Viterbi segmentation and parameter updates. With this technique, state means and variances are computed by averaging all the feature vectors associated with each state. Owing to problems observed in training some data, the number of iterations was limited to three for this stage. The state transition matrix is estimated by time counts of state occupation. For the Gaussian mixtures

of a given state, each feature vector of the state is associated with its highest likelihood Gaussian and the mixture weights are computed according to the ratio of feature vectors per Gaussian. To start the process, a uniform segmentation of the digit to the HMM states is presumed and parameters are initially estimated. For initial estimation of the Gaussians, a modified K-means clustering algorithm is used. In the second stage, Expectation Maximization (EM) re-estimation of the HMM parameters is performed using the same training data.

Having performed the training, a model now exists for each digit for each speaker using data from all other speakers. To perform the digit recognition, the test feature vector sequence was scored against the HMM for each of the four possible digits uttered and the highest scoring HMM was selected as the hypothesized digit. This scoring consisted of generating the log-probability of the observation (i.e. feature vector) sequence being generated by the HMM. The recognized digit was compared to the actual digit and the results were tallied over all utterances. The overall accuracy (number of correctly-identified digit trials divided by the total number of trials) was the performance metric used.

4 Results

4.1 Contour Features

For the first set of features—the contour features—the training/testing evaluation procedure was run for different numbers of HMM states and Gaussians per state to explore the parameter space and determine good settings for later experiments. The results are given in table 2. The missing entry (five states and four Gaussians) results from failed training using those parameter choices. From the table the best choice appears to be five states and a single Gaussian per state, which yields an accuracy of 91.67%. The small number of Gaussians is not surprising because the small number of feature vectors per digit (on the order of 10) limits the number of Gaussians whose parameters can be adequately estimated. The larger number of states is surprising, though, because it, too, should be governed the average number of frames.

Also of interest is the fact that high accuracy can be obtained using a single state. In this case, the dynamic information is contained only in the delta parameters. This suggests the delta parameters may be very good at capturing the dynamic aspects of the process and are quite useful. Regardless of the HMM structure, the performance for these features is very

	G1	G2	G4
S1	84.38%	84.38%	80.21%
S3	89.58%	87.5%	85.42%
S5	91.67%	90.62%	-

Table 2: Accuracy results for different HMM Gaussian (*G*) and State (*S*) combinations for contour features.

good. A confusion matrix for the best performing system (i.e., choice of parameters) is given in table 3 and the settings were used for all subsequent experiments.

Recognized \ True	One	Two	Three	Four
One	23	0	3	2
Two	0	23	0	0
Three	1	0	21	1
Four	0	1	0	21

Table 3: Confusion matrix for best system using contour features

4.2 Raw Image Features

For the features derived from the raw image pixels, different levels of dimensionality reduction were done and the results compared. The accuracies obtained are shown in table 4. From these results, use of the first ten principle components seems to yield the best performance. Note the similarity in the best performance of these automatically obtained features (89.58%) to the previous ones (91.67%) which required human assistance. This demonstrates the power of such general-purpose statistical techniques. Again, a confusion matrix for the best performing set of features is given (see table 4).

4.3 Preprocessed Image Features

Table 6 shows the results using features obtained from the image preprocessing techniques described in 3.1 along with applications of PCA dimensionality reduction to those features. The preprocessing produces well-performing features which result in an accuracy of 79.17%. By applying PCA, too, the accuracy approaches that of the raw image PCA, but does not quite

# Comps.	Accuracy (%)
5	76.04
10	89.58
25	84.38
50	78.12
100	70.83

Table 4: *Results for PCA on raw image features for 5-state, 1-Gaussian HMMs. The first column gives the number of components used and the second the accuracy.*

True \ Recognized	One	Two	Three	Four
One	23	0	2	0
Two	0	20	0	1
Three	1	2	21	1
Four	0	2	1	22

Table 5: *Confusion matrix for best system using raw image features*

match it. Similarly, the accuracy is less than that obtained with the contour features. It appears that, within the limited exploration performed, the technique using special properties of the data (i.e., that it is a collection of images) is inferior to the more “blind” general-purpose one. A final confusion matrix is given for the best performing set of these features in table 7.

Features	Accuracy (%)
Preprocessing	79.17
Preprocessing+PCA5	77.08
Preprocessing+PCA10	83.33
Preprocessing+PCA25	87.5
Preprocessing+PCA50	77.08

Table 6: *Results for image preprocessing for features and application of PCA to those features. The first column gives the features used and the second the accuracy.*

True \ Recognized	One	Two	Three	Four
One	23	2	1	1
Two	0	19	0	1
Three	1	1	20	0
Four	0	2	3	22

Table 7: *Confusion matrix for best system using preprocessed image features*

5 Conclusions

In this paper, an analysis of the use of HMMs for visual speech recognition on the digit recognition task of the Tulips1 corpus was presented. Central to this analysis was the choice of HMM structure and the selection of features to be modeled. With regard to structure, it was determined that single-Gaussian HMMs with five states produced the best results for the task. This result conformed somewhat to expectations, given the small number of feature vectors per utterance, but not fully as the number of states did not seem to exhibit the trend of few parameters for few observations. Additionally, the importance of delta features was suggested by looking at the performance of the single-state case.

With regard to features, the hand-marked contour features worked very well, with a best accuracy of 91.67%. Though requiring significant human involvement, the procedure can alternatively be automated, and these good results provide motivation to do so. As for the two other feature sets, PCA dimensionality reduction yielded performance similar to the contour features and outperformed the basic described image-based approach. This being the case, it would be interesting to look at other image-based approaches such as vector quantization and optical flow [4] and compare them with this statistical approach.

References

- [1] J.R. Movellan, "Visual Speech Recognition with Stochastic Networks," in *Advances in Neural Information Processing Systems*, vol. 7, pp. 851-858, 1995.
- [2] M.A. Anwar, J.F. Baldwin, and T.P. Martin, "Learning Fuzzy Rules for Visual Speech Recognition," in *Proc. Adaptive Multimedia Retrieval*, pp. 164-175, 2003.
- [3] HTK Book is available at
<http://htk.eng.cam.ac.uk/docs/docs.html>
- [4] J. Luettin, and N. Thacker, "Speechreading Using Probabilistic Models," in *Computer Vision and Image Understanding*, vol. 65, pp. 163-178, 1997.