Audio-Visual Automatic Speech Recognition using Dynamic Bayesian Networks

Helge Reikeras
MIH Media Lab
Stellenbosch University
helge@ml.sun.ac.za

Ben Herbst
Applied Mathematics
Stellenbosch University
herbst@sun.ac.za

Johan du Preez
and Herman Engelbrecht
Electrical & Electronic Engineering
Stellenbosh University
{dupreez, hebrecht}@sun.ac.za

***Abstract*—In audio-visual automatic speech recognition (AVASR) both acoustic and visual modalities of speech are used to determine what a speaker is saying.**

In this paper we propose a basic AVASR system that uses mel-frequency cepstrum coefficients (MFCCs) as acoustic features, active appearance model (AAM) parameters as visual features, and dynamic Bayesian Networks (DBNs) as probabilistic models of audio-visual speech.

The performance of the AVASR system is tested using the Clemson University audio-visual experiments (CUAVE) database. As expected, we find that visual-only speech recognition (automatic lip-reading) performs worse than audio-only speech recognition. However, by integrating visual and acoustic speech information we are able to significantly increase performance, in particular in noisy acoustic environments.

I. Introduction

Motivated by the multi-modal manner humans perceive their environment, research in audio-visual automatic speech recognition (AVASR) focuses on the integration of acoustic and visual speech information with the purpose of improving accuracy and robustness of automatic automatic speech recognition systems. AVASR is in particular expected to perform better than audio-only automatic speech recognition (ASR) in noisy acoustic environments, as the visual channel is not affected by acoustic noise.

Functional requirements for an AVASR system include acoustic and visual feature extraction, learning, and classification.

In this paper we propose a basic AVASR system using mel-frequency cepstrum coefficients (MFCCs) as acoustic features, active appearance model (AAM) parameters as visual features, and dynamic Bayesian Networks (DBNs) as probabilistic models of audio-visual speech.

The performance of the AVASR system is tested using the Clemson University audio-visual experiments (CUAVE) database. As expected, we find that visual-only speech recognition (automatic lip-reading) in general performs worse than audio-only speech recognition. However, by integrating visual and acoustic speech information we are, in particular in noisy acoustic environments, able to obtain significantly better performance than what is possible with audio-only ASR.

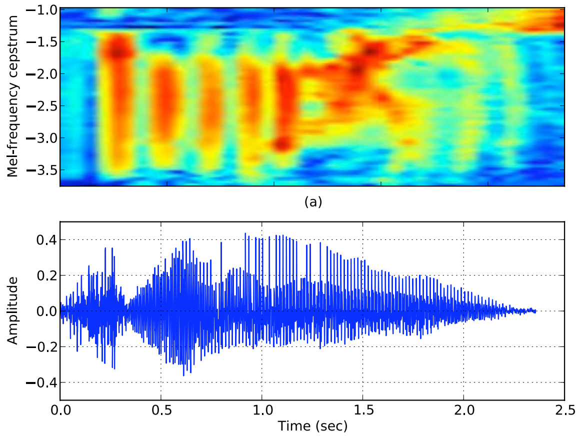

Fig. 1. Acoustic feature extraction from an audio sample of the spoken word ‘zero’. Mel-cepstrum (top) and original audio sample (bottom).

II. Feature Extraction

A. Acoustic Speech

MFCCs are the standard acoustic features used in most modern speech recognition systems. In [1] MFCCs are shown experimentally to give better recognition accuracy than alternative parametric representations.

MFCCs are calculated as the cosine transform of the logarithm of the short-term energy spectrum of the signal, expressed on the mel-frequency scale. The result is a set of coefficients that approximates the way the human auditory system perceives sound.

The total number of MFCC feature vectors obtained from a single audio sample depends on the duration of the original sample, the sample rate, the chosen window size, and the amount of overlap between adjacent windows.

Figure 1 shows an audio sample of the word ‘zero’ together with the corresponding mel-frequency cepstrum.

B. Visual Speech

While acoustic speech features can be extracted through a sequence of transformations applied to the original input signal, extracting visual speech features is in general more

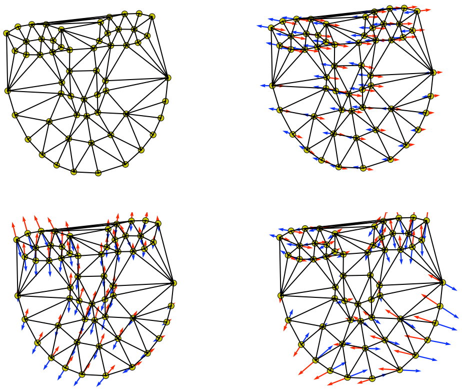

Fig. 2. Triangulated base shape $\mathbf{s}_0$ (top left), and first three shape vectors $\mathbf{p}_1$ (top right), $\mathbf{p}_2$ (bottom left) and $\mathbf{p}_3$ (bottom right) represented by arrows superimposed onto the triangulated base shape.

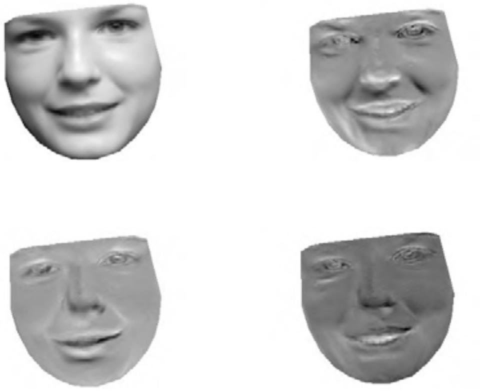

Fig. 3. Mean appearance A_0 (top left) and first three appearance images A_1 (top right), A_2 (bottom left) and A_3 (bottom right).

complicated. The visual information relevant to speech is mostly contained in the motion of visible articulators such as lips, tongue and jaw. In order to extract this information from a video sequence it is advantageous to track the complete motion of the speaker's face and selected facial features.

AAM fitting [2] is an efficient and robust method of tracking the motion of deformable objects in a video sequence. AAMs model variations in shape and texture of an object of interest. In contrast to MFCCs, AAMs require prior training before being used for feature extraction. In order to build an AAM it is necessary to provide sample images with the shape of the object annotated.

The shape of an appearance model is given by a set of (x, y) coordinates represented in the form of a column vector

$$\mathbf{s} = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)^{\mathrm{T}}. \tag{1}$$

The coordinates are relative to the coordinate frame of the image.

Shape variations are restricted to a base shape $\mathbf{s}_0$ plus a linear combination of N shape vectors

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{N} p_i \mathbf{s}_i \tag{2}$$

where p_i are called the shape parameters of the AAM.

The base shape and shape vectors are normally generated by applying principal component analysis (PCA) to a set of manually annotated training images. The base shape $\mathbf{s}_0$ is the mean of the object annotations in the training set, and the shape vectors are the N singular vectors corresponding to the N largest singular values of the training shape data matrix. Figure 2 shows an example of a base mesh and the first three shape vectors corresponding to the three largest singular values.

The appearance of an AAM is defined with respect to the base shape $\mathbf{s}_0$. Appearance variation is restricted to a base appearance plus a linear combination of M appearance vectors

$$A(\mathbf{x}) = A_0 + \sum_{i=1}^{M} \lambda_i A_i(\mathbf{x}) \qquad \forall \mathbf{x} \in \mathbf{s}_0. \tag{3}$$

To generate an appearance model, the training images are first shape-normalized by warping each image onto the base mesh using a piecewise affine transformation. Noting that two corresponding sets of three points are sufficient for determining an affine transformation, the shape mesh vertices are first triangulated. The collection of corresponding triangles in two shape meshes then defines a piecewise affine transformation between the two shapes. Next, the pixel values within each triangle in the training shape $\mathbf{s}$ are warped onto the corresponding triangle in the base shape $\mathbf{s}_0$ using the affine transformation defined by the two triangles.

The appearance model is generated from the shape-normalized images using PCA. Figure 3 shows an example of a base appearance and the first three appearance images.

Tracking of an appearance in a sequence of images is performed by minimizing the difference between the base model appearance, and the input image warped onto the coordinate frame of the AAM. For a given image I we minimize

$$\underset{\boldsymbol{\lambda},\mathbf{p}}{\operatorname{argmin}} \sum_{\mathbf{x}} \left[A_0(\mathbf{x}) + \sum_{i=1}^{M} \lambda_i A_i(\mathbf{X}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \right]^2 \tag{4}$$

where $\mathbf{p} = \{p_1, \ldots, p_N\}$, $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_M\}$ and $\mathbf{x} \in \mathbf{s}_0$. Note that, for the remaining discussion of AAMs, we assume that the domain of $\mathbf{x}$ is the image coordinates contained within the base mesh $\mathbf{s}_0$ (as in (4)).

In (4) we are looking for the optimal alignment of the input image, warped backwards onto the frame of the base appearance $A_0(\mathbf{x})$. A motivation for the backwards warp can be found in [3].

For simplicity of the presentation, we shall only consider variation in shape and ignore texture variation. The derivation for the case including texture variation is available in [3].

Consequently (4) reduces to

$$\underset{\mathbf{p}}{\operatorname{argmin}} \sum_{\mathbf{x}} [A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]^2. \tag{5}$$

Solving (5) for $\mathbf{p}$ is a non-linear optimization problem. This is the case even if $\mathbf{W}(\mathbf{x};\mathbf{p})$ is linear in $\mathbf{p}$ as the pixel values $I(\mathbf{x})$ are in general nonlinear in $\mathbf{x}$.

The quantity that is minimized in (5) is the same quantity that is minimized in the classic Lucas-Kanade image alignment algorithm [4]. In the Lukas-Kanade algorithm the problem is first reformulated as

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \sum_{\mathbf{x}} [A_0(\mathbf{X}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p}))]^2. \tag{6}$$

This equation differs from (5) in that we are now optimizing with respect to $\Delta\mathbf{p}$ while assuming $\mathbf{p}$ is known. Given an initial estimate of $\mathbf{p}$ we update with the value of $\Delta\mathbf{p}$ that minimizes (6) to give

$$\mathbf{p}^* = \mathbf{p} + \Delta\mathbf{p}. \tag{7}$$

This will necessarily decrease the value of (5) for the new value of $\mathbf{p}$. Replacing $\mathbf{p}$ with the updated value for $\mathbf{p}^*$, this procedure is iterated until convergence at which point $\mathbf{p}$ yields the (local) optimal shape parameters for the input image I.

To solve (6) Taylor expansion is used, which gives

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \sum_{\mathbf{x}} \left[A_0(\mathbf{W}(\mathbf{x};\mathbf{p})) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) - \nabla I \frac{\partial\mathbf{W}}{\partial\mathbf{p}} \Delta\mathbf{p}\right]^2 \tag{8}$$

where ∇I is the gradient of the input image and $\partial\mathbf{W}/\partial\mathbf{p}$ is the Jacobian of the warp evaluated at $\mathbf{p}$.

The optimal solution to (8) is found by setting the partial derivative with respect to $\Delta\mathbf{p}$ equal to zero which gives

$$2\sum_{\mathbf{x}} \left[\nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^{\mathrm{T}} \left[A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x})) - \nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p}\right] = 0. \tag{9}$$

Solving for $\Delta\mathbf{p}$ we get

$$\Delta\mathbf{p} = \mathbf{H}^{-1}\sum_{\mathbf{x}} \left[\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^{\mathrm{T}} [A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))] \tag{10}$$

where $\mathbf{H}$ is the Gauss-Newton approximation to the Hessian matrix given by

$$\mathbf{H} = \sum_{\mathbf{x}} \left[\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^{\mathrm{T}} \left[\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]. \tag{11}$$

For further details on how to compute the piecewise linear affine warp and the Jacobian see [3]. An extension of this method that includes a global shape normalizing transform and appearance variation is described in [3].

The resulting AAM variation parameters $\boldsymbol{\lambda}$ are used together with the shape parameters $\mathbf{p}$ as visual features in the AVASR system.

Figure 4 shows an AAM fitted to an input image. When tracking motion in a video sequence, the previous optimal fit is typically used as a starting point for the search in the following frame.

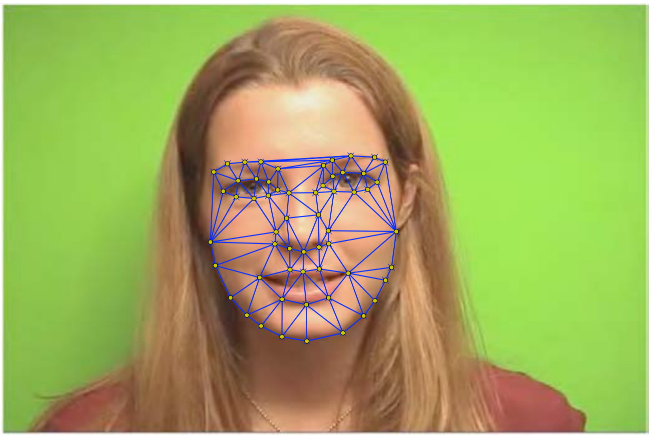

Fig. 4. AAM fitted to an image

The AAM fitting method described above is referred to as the *forwards-additive* method [5]. In AVASR applications with real-time performance constraints we are often willing to sacrifice some accuracy for increased efficiency. In [3] several variations of the Lucas-Kanade method are evaluated and it is concluded that the *inverse-compositional* method gives the best trade-off between performance and accuracy. We have used the inverse compositional method for the research presented in this paper.

III. Modeling Audio-Visual Speech using Dynamic Bayesian Networks

A. *Definition*

A dynamic Bayesian network (DBN) is an extension of Bayesian networks that allows for modeling variable-length (and potentially semi-infinite) sequences of hidden and observed random variables and their dependencies. Variables are represented by nodes in a graph, and dependencies are represented by directed arcs connecting the nodes. As with Bayesian networks, a DBN must constitute a directed acyclic graph (DAG). The term *dynamic* is used as DBNs are typically used to model dynamic systems.

A DBN graph constitutes a set identically structured time slices. The semantics of a DBN is defined by a prior distribution over the nodes in the initial slice

$$p(\mathbf{V}_1) = \prod_{i=1}^{N} p(\mathbf{v}_1^i|\mathrm{pa}(\mathbf{v}_1^i)), \tag{12}$$

and a distribution over a two-slice temporal Bayesian network defining the transition from a slice to the next

$$p(\mathbf{V}_t|\mathbf{V}_{t-1}) = \prod_{t=1}^{T}\prod_{i=1}^{N} p(\mathbf{v}_t^i|\mathrm{pa}(\mathbf{v}_t^i)). \tag{13}$$

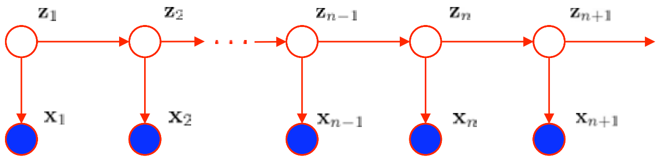

Fig. 5. HMM modeled as a DBN

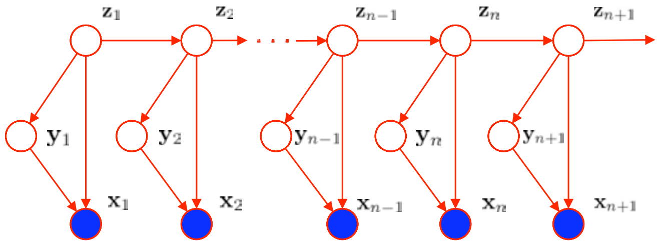

Fig. 6. An HMM with GMM observation model

In (12) and (13) $\mathbf{v}_t^i$ is the random variable represented by the ith node in time slice t of the DBN and $\text{pa}(\mathbf{v}_t^i)$ is the set of variables representing the parents of the ith node in the graph. We restrict parent nodes to lie in the same time slice as node i, or in the previous time slice. The set of random variables is typically partitioned into hidden and observed nodes $\mathbf{V} = (\mathbf{Z}, \mathbf{X})$ where $\mathbf{Z}$ and $\mathbf{X}$ are hidden and observed variables, respectively.

A well-known example of a DBN is the Hidden Markov Model (HMM). An HMM is probabilistic model defined by

$$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = p(\mathbf{x}_1|\boldsymbol{\pi}) \left[\prod_{t=1}^{T} p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{A}) \right] \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_t, \boldsymbol{\phi}) \tag{14}$$

where $\boldsymbol{\pi}$ is the prior over HMM states, $\mathbf{A}$ is the transition matrix and $\boldsymbol{\phi}$ is the observation model parameters. From (12), (13) and (14) we see that an HMM has the graphical representation shown in Figure 5.

The observation model can in principle have any distribution. In speech recognition applications it is common to model observations as a mixture of Gaussians

$$b_t(\mathbf{x}_t|\mathbf{z}_t) = \sum_{j=1}^{M} w_{ij} \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_{ij}, \boldsymbol{\Sigma}_{ij}), \tag{15}$$

where w_{ij}, $\boldsymbol{\mu}_{ij}$ and $\boldsymbol{\Sigma} ij$ are the weight, mean and covariance of the jth mixture component and ith HMM state, respectively.

In fact, by introducing an additional multinomial hidden variable $\mathbf{y}_t$ indicating which mixture component is selected, we can model an HMM with a Gaussian mixture observation model as a DBN where each node in the graph has a distribution belonging to the exponential family [6] (we can show that GMMs are not in the exponential family). This property has important consequences for inference and learning, as it allows us to use general DBN inference and learning algorithms. The graphical representation of the HMM with GMM observations is shown in Figure 6.

B. *Modeling asynchrony*

In the DBN framework it becomes possible to model additional properties of audio-visual speech as extensions to the basic HMM model. An interesting property to model is the asynchrony between acoustic and visual speech (when speaking the motion of articulators comes prior to the sound being uttered).

One possibility is to simply concatenate the acoustic and visual features into a single feature vector and use the standard HMM as shown in Figure 5. We call this the *audio-visual* HMM (AV-HMM). A minor variation on the AV-HMM is the audio-visual *product* HMM (AV-PHMM). In the AV-PHMM the audio and visual observation models have separate nodes, allowing us to weigh each stream differently. The AV-HMM and AV-PHMM models assume perfect synchrony of the audio and visual observation streams, and hence might not adequately capture the natural asynchrony between the two streams.

Another possibility is to use two separate HMMs for the audio and visual observation streams. Each stream will then have a separate state space independent from the other. The resulting DBN is the audio-visual *independent* HMM (AV-IHMM). Although this model will allow state asynchrony between the acoustic and visual observation streams, it may fail to capture the natural correlation between the two streams.

Ideally, we would like an asynchrony model that constrains the level of asynchrony to somewhere in between that of the AV-PHMM and AV-IHMM.

In [7], several asynchrony DBN models for AVASR are proposed. From the seven models considered, it was found that the *coupled* HMM gives the lowest word error rate on the AVASR task. In the audio-visual coupled HMM (AV-CHMM) the observation and state nodes of the audio and visual streams are separate, but coupled at the state level. The AV-CHMM model is shown in Figure 7. Note that, to avoid clutter we have omitted the node labels and the details of the observations model. The level of asynchrony is constrained by limiting the number of states that the two streams are allowed to de-synchronize. In our experiments we have allowed the stream to de-synchronize by one state only. In practice, we apply this constraint by setting transitions that lead to 'illegal' levels of de-synchrony to be 0 in the transition matrix.

In the experiments we use the AV-CHMM as the audio-visual speech model.

C. *Learning*

Learning DBNs is typically done using the *expectation maximization* (EM) algorithm. EM iteratively calculates the expected sufficient statistics and re-estimates the parameters of each node in the DBN. This can be done in general for models that are in the exponential family and with nodes that have multinomial or Gaussian distributed random variables.

The objective of the EM algorithm is to maximize the log likelihood function of the model parameters given the observed data. In general, the algorithm will only find a local maximum. It is therefore important to have good initial values for the

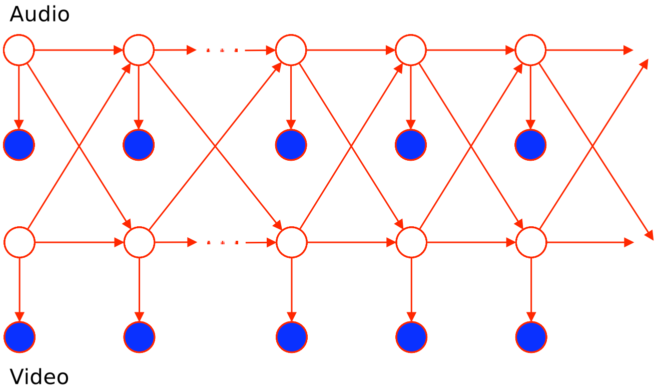

Fig. 7. Audio-visual coupled HMM

model parameters. The k-means algorithm is commonly used for this purpose.

In the common case that we have multiple observation sequences, e.g. multiple recorded samples of the same word or phoneme, the expected sufficient statistics for all sequences are pooled before the parameters are re-estimated.

In an isolated word recognition task, a separate model is typically learned for each word.

D. Classification

Classification is performed using the max-sum algorithm. The max-sum algorithm is analogous to the Viterbi algorithm for HMMs. The max-sum algorithm will calculate the most likely state sequence given a novel observation sequence. In an isolated word recognition task, a novel sample is classified as belonging to the model with the most likely state sequence, i.e. the model that is most likely to have generated the observation sequence.

Further details on interference and learning in DBNs can be found in [6].

IV. Experimental results

An AVASR system using MFCCs as audio features, AAM coefficients as visual features, and DBNs for audio-visual speech modeling, was implemented as described in the previous sections. The audio and visual feature extraction (MFCCs and AAMs) is implemented in Python as described in [8]. The DBNs are implemented using the Bayesian Network Toolkit [9] written by Kevin Murphy.

In order to test the system we use the Clemson University audio-visual experiments (CUAVE) database [10]. The CUAVE database consists of 36 speakers, 19 male and 17 female, uttering isolated and continuous digits. Video of the speakers is recorded in frontal, profile and while moving. We only use the portion of the database where the speakers are stationary and facing the camera while uttering isolated digits (referred to as the ‘normal’ part in the CUAVE documentation). We use $2/3$ of the data from all speakers for training and the remaining $1/3$ for testing. The speakers in the training set are the same as in the test set, resulting in a closed-set speech recognition experiment. As such, the results reported here are applicable for *speaker-dependent* automatic speech recognition systems. A sample frame from each individual in the data corpus is shown in Figure 8.

Fig. 8. Sample frames from CUAVE

Visual feature extraction is done by building individual AAMs for each speaker in the data corpus. The AAMs are learned from manually annotated training data. We found it sufficient to manually annotate every 50th frame of each video. The visual features (AAM coefficients) are subsequently extracted by fitting the AAM to each frame of the video.

Training the audio-visual speech recognition system consists of learning audio-visual DBNs for each digit in the data corpus from the training data. Learning is performed using the EM algorithm. Testing is performed by classifying each of the isolated digit samples in the test data using the max-sum algorithm. To evaluate the performance of the system we use the misclassification rate, i.e. the number of wrongly classified test samples divided by the total number of test samples.

A core feature of AVASR is the robustness to acoustic noise. We therefore wish to test the effects of varying the level of noise in the audio channel. Acoustic white Gaussian noise, ranging from -5dB to 15dB in steps of 5 dB signal-to-noise ratio (SNR), is added to the test data.

In order to compensate for the varying levels of acoustic noise in the model, we make use of stream weights. The audio and visual observation probabilities are weighted exponentially by stream weights λ_A and λ_V, where A and V are the audio and video streams, respectively. From (15) we get

$$\tilde{b}_t^s = b_t^s(\mathbf{x}_t^s|\mathbf{z}_t^s)^{\lambda_s}, \qquad (16)$$

where $s \in \{A, V\}$.

We constrain λ_A and λ_V to lie between zero and one and sum to one, i.e.

$$\lambda_A + \lambda_V = 1 \qquad \text{for} \qquad 0 < \lambda_A, \lambda_V < 1. \qquad (17)$$

For each of the SNR levels, we estimate an optimal stream

AV-CHMM					
SNR	-5	0	5	10	15
λ_A	0.0	0.3	0.6	0.9	1.0
λ_V	1.0	0.7	0.4	0.1	0.0

TABLE I
OPTIMAL STREAM WEIGHTS FOR THE AV-CHMM

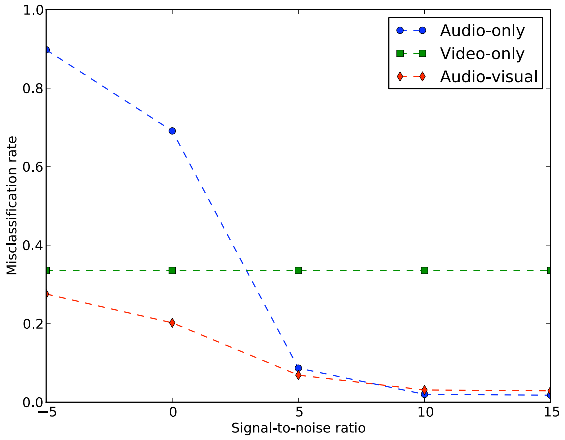

Fig. 9. Misclassification rate for varying SNR levels

exponent. The optimal stream exponent is the exponent that results in the lowest misclassification rate on the test data. As the number of classes in our experiment is relatively small we perform this optimization using a brute-force grid search, directly minimizing the misclassification rate. We vary λ_A from 0 to 1 in steps of 0.1. The corresponding λ_V will then be given by $1 - \lambda_A$. The set of parameters λ_A and λ_V that results in the lowest misclassification rate on the test data are chosen as optimal parameters. The optimal stream weights for the AV-CHMM for each SNR level is shown in Table I.

In order to evaluate performance, we perform classification for each of the SNR levels using the estimated optimal stream weights for the respective SNR levels. This is not an unrealistic model, as several methods exist for estimating the noise level in both the audio and video input signals [11]. We evaluate the system performance by calculating the average misclassification rate for the digits using each of the SNR levels. We compare audio-only, visual-only, and audio-visual classifiers. Figure 9 shows average misclassification rate for the different models and noise levels.

From the results we see that the visual channel indeed does contain information relevant to speech, but that the performance of visual-only speech recognition is, as expected, lower than audio-only speech recognition. However, as SNR in the audio channel decreases, AVASR performs significantly better than audio-only speech recognition. This is because the combination of acoustic and visual speech information is superior to any of the two modalities separately.

V. Conclusion

In this paper we propose a basic AVASR system that uses MFCCs as acoustic features, AAM parameters as visual features, and audio-visual DBNs for modeling audio-visual speech.

The AVASR system is tested using the CUAVE database. Based on the experimental results, we conclude that the visual channel does contain information that is relevant to speech, but that the performance of visual-only speech recognition (automatic lip-reading) is, as expected, lower than audio-only speech recognition. However, in the presence of increasing acoustic noise we are able to increase recognition performance beyond that of audio-only speech recognition by combining acoustic and visual speech information. This conforms to our daily experience, as it is typically easier to understand someone speaking if we can indeed see them.

Acknowledgment

The authors wish to thank MIH Holdings for providing funding for the research presented in this paper.

References

[1] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[2] T. F. Cootes, G. J. Edwards, and C. Taylor, “Active appearance models,” 1998.

[3] I. Matthews and S. Baker, “Active appearance models revisited,” *International Journal of Computer Vision*, vol. 60, pp. 135–164, 2003.

[4] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *IJCAI’81: Proceedings of the 7th international joint conference on Artificial intelligence.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[5] S. Baker and I. Matthews, “Equivalence and efficiency of image alignment algorithms,” in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 1090–1097.

[6] K. P. Murphy, “Dynamic bayesian networks: Representation, inference and learning,” Ph.D. dissertation, UC Berkeley, Computer Science Division, 2002. [Online]. Available: file:///home/helge/resources/theses/2002-dbnril.pdf

[7] A. V. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy, “Dynamic bayesian networks for audio-visual speech recognition,” *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 1, pp. 1274–1288, 2002.

[8] H. Reikeras, B.M. Herbst, J. du Preez, and H.A. Engelbrech, “Audio-Visual Automatic Speech Recognition in Python,” *Proceedings of the 9th Python in Science conference (SciPy 2010)*, 2010.

[9] K. P. Murphy, “The Bayes net toolbox for Matlab,” 2001. [Online]. Available: http://code.google.com/p/bnt/

[10] E.K. Patterson, S. Gurbuz, Z. Tufekci, and J.N. Gowdy, “Cuave: A new audio-visual database for multimodal human-computer interface research,” 2002.

[11] G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos, “Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition,” *Trans. Audio, Speech and Lang. Proc.*, vol. 17, no. 3, pp. 423–435, 2009.