

EV/MA: An Architecture for Audio-Visual Speech Recognition

Bradley A. Swerdfeger

Supervised by: Purang Abolmaesumi

COGS499 Thesis, School of Computing, Queen's University
Kingston, Ontario, Canada

Abstract— This paper presents an architecture of audio-visual speech recognition. It is a hidden Markov model based system that uses completely separate models for the auditory and visual domains, with an integration that integrates the classifications made by both modalities. The visual domain uses the parameters of an ellipse approximating the shape of a speaker's lips as a feature for recognition (EV) and the auditory domain uses classical MEL-cepstrum based features (MA). Techniques for extracting these elliptical features are also discussed. The architecture is shown to provide a vast improvement over classical auditory-only recognition when there is noise in the auditory signal.

Index Terms— Multimodal Perception, Hidden Markov Models, Automatic Speech Recognition.

I. INTRODUCTION

IN their seminal 1976 paper, McGurk and MacDonald [12] describe a powerful phenomenon that is convincing evidence that speech perception is a process which uses information from both the audio and visual modalities.

In their demonstration, they present an audio stimulus of a speaker saying 'BA' repeatedly. At the same time, they present a visual stimulus of a speaker saying 'GA' repeatedly such that the two signals are synchronized. The study showed that often, the syllable 'DA' was perceived; which is a fusion between the visual and auditory stimuli. This effect is surprisingly robust. It occurs when the perceiver has already seen the effect before, on perceivers with all language backgrounds [11], on young infants [18], when the visual and auditory components are from speakers of different genders [7], with highly reduced facial images [17], and using haptics rather than vision [6].

The pervasiveness of the McGurk Effect demonstrates that human speech perception is a multi-modal process: our perceptual system makes the most out of the information available. For instance, the visual information becomes extremely important in noisy environments, such as a party or a busy street. Also, when conflicting information is presented (which is usually not experienced in situations other than laboratory experiments), the best match that satisfies the constraints of the incoming stimuli is perceived.

Given that this effect is so widely-known and vigorously studied, it is surprising that common automatic speech

recognition systems do not use any visual information; especially since webcams have become so inexpensive.

Classical automatic speech recognition suffers from a drastic reduction in accuracy when there is noise in the audio signal [16]. We propose an architecture that integrates the information from both the auditory and visual modalities for speech recognition. This is done by using an ellipse that approximates the shape of the lips as a feature for recognition in the visual domain (EV), and classical, MEL-cepstrum based features for the auditory domain (MA). Recognition and training using hidden Markov model back-ends occurs separately for the visual and auditory modalities, and the decisions are mixed and disambiguated when the final probabilities for classification have been computed.

This paper is organized as follows. Section II describes classical, audition-based, automatic speech recognition and briefly discusses its problems. Section III describes the architecture and recognition process behind EV/MA. Section IV describes in detail, two methods for elliptical feature extraction in the visual domain. Section V describes model parameters for an implementation of EV/MA. Section VI describes the performance of EV/MA by giving an example of how it has been trained and tested under different conditions. Section VII discusses strengths and weaknesses of the architecture and techniques outlined and directions for future work at improving the system. Finally, Section VIII provides a summary and concluding remarks.

II. CLASSICAL AUTOMATIC SPEECH RECOGNITION

Even though automatic speech recognition (ASR) is currently commonplace as a useful technology, in the 1970s and early 1990s, its study was considered esoteric by most. This was due to the fact that most techniques involved the use of neural networks or dynamic programming which did not perform very well since speech recognition is a sequentially dependant task. It was not until the widespread adoption of hidden Markov models (HMM) by engineers in the 1990s that ASR systems became accurate and reliable enough to be used in public systems; even though HMMs have been used in statistics and mathematics in the 1960s.

In 1989, Lawrence Rabiner published a paper [15] that helped propagate the adoption of hidden Markov models by ASR researchers. He explains their concept in a manner that is familiar to computer scientists and provides a detailed description of their use in speech recognition. This section

briefly explains the concept behind HMMs and how their training and deployment can be realized as an implementation for automatic speech recognition.

A. Hidden Markov Models

There are many signals in our world where we can only observe the signal itself. We do not necessarily know the underlying process that created these signals, or any model that describes these signals.

Hidden Markov models attempt to describe the statistical properties of these signals. They can be described as state machines where the transitions between states are probabilistic, and the states represent states of the system being described. The actual number of states generally has something to do with the system being described. For instance, if we were trying to describe a five phoneme word, our system would have five states.

The use of HMMs can be described by three problems:

- 1) Given an observation sequence O , and a model λ , how do we compute $P(O|\lambda)$: the likelihood of this observation, if we already have a model?
- 2) Given an observation sequence O , and a model λ , how do we come up with a sequence of states that best explains the sequence?
- 3) How do we adjust the parameters of a model λ in order to maximize $P(O|\lambda)$?

Problem 3 is of most importance when training a hidden Markov model for speech recognition, and problem 2 is used in deployment for recognition of input stimuli. This paper will discuss the use of hidden Markov models as a single word recognizer, where a separate HMM is trained to recognize each word in the chosen dictionary.

B. Inputs for Automatic Speech Recognition

Since an ASR system is classifying speech, it usually uses .wav files as input stimuli. Wav files contain a zero-sum normalized power spectrum, with 16-bit values, sampled at 44100 Hz (magnitude consistency is maintained with a gain parameter contained within the header) [19]. Obviously, this signal is extremely large and the information contained within must be compressed in order for recognition and training to be a tractable task. Also, data classifiers generally use feature vectors as inputs. A .wav file only presents one value at every sample. This power spectrum must be converted into something represented by a vector of features.

The most commonly used compression and feature extraction method used is called the MEL-cepstrum [21]. The MEL-cepstrum is calculated by taking a single, overlapping window of the power spectrum and then taking its Fourier transform as if it were a signal. Then, the log amplitudes of the power spectrum are converted into the MEL scale using a set of positive spectral windowing functions with bandwidths increasing at higher frequencies. This filtering process is similar to how the human auditory system filters incoming audio signals using a continuous set of band pass frequency filters that are logarithmically spaced and whose bandwidth increases as the frequency increases [22]. Instead of a

continuous number of filters, the MEL-cepstrum uses discrete sets (usually around 12 sets) of these filter banks. Then, the Discrete Cosine Transform (DCT) is calculated for the log amplitudes of the resulting filtered spectrum. The C most significant coefficients of the DCT, where C is the desired length of the feature vector, are used to represent the entire window. The rates of change of the resulting coefficients can also be appended to the feature vector.

The above process results in a data compression where most of the noise has been removed and the important, biologically plausible data has been retained. A one second wav file can be compressed into a 249x24 matrix with a window size of 8 ms and an overlap of 4ms.

C. Training a Hidden Markov Model for ASR

First, one must decide on the number of states in their model for a given word. Rabiner advises using a number of states that has some physical meaning within the domain. For instance, one might choose a number of states equivalent to the number of syllables or phonemes in the word being recognized.

Then, one must decide on the number of Gaussian mixtures that will be fit to each state. With continuous data, as is used in ASR, training occurs by fitting a mixture of Gaussians to the sequence of cepstral vectors for each state, such that the probabilities of making the training observations are maximized for the current model. This is done by using the well known iterative K-Means procedure on the parameters of the Gaussians, and then a further refinement is conducted using Expectation Maximization [3]. Generally, a good rule is to use slightly less than half of the size of your input vector in order to reduce the chances of overfitting but also allow for an accurate fit around the cepstral data. It is also a good idea to constrain the covariance matrices of the Gaussians to diagonal matrices in order to reduce the likelihood of it becoming a point function. Naturally, since these algorithms are optimization techniques, good initialization for the covariance matrices of the Gaussians is important so that the fitting procedure does not find a local minimum that will perform poorly; although, if a better initialization procedure is not known, uniform random values can suffice.

D. Word Recognition in ASR

Once the hidden Markov models have been trained for each word, it is ready for deployment. Recognition occurs by first, computing the MEL-cepstrum of the speech signal, and then calculating the most likely path through each model for the cepstral sequence. The 'best sequence' is the one that maximizes $Prob\{R|O,\lambda\}$, which is the same as $Prob\{R,O|\lambda\}$, where R is a sequence of states, O is the input sequence, and λ is the hidden Markov model for the current word. This measure is found using the Viterbi algorithm [20], which keeps track of the sequence of states that maximizes $\delta_k = \max_{r_1, r_2, \dots, r_{k-1}} Prob\{r_1 r_2 \dots r_k = i, O_1 O_2 \dots O_k | \lambda\}$ where δ_k is the score along a single path at time k , which accounts for the first k observations and ends at state S_i . To find the

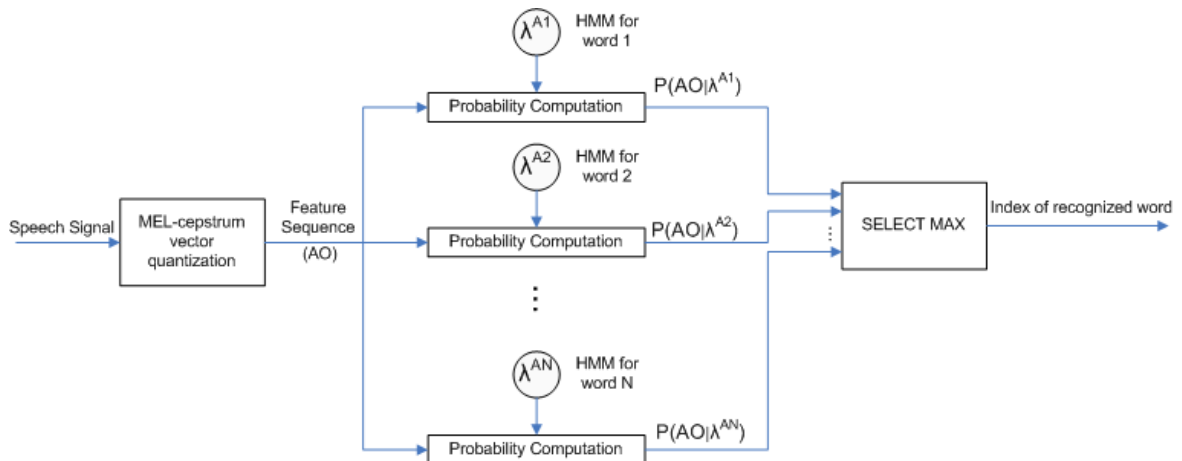


Fig. 1. Architecture of a classical speech recognition system.

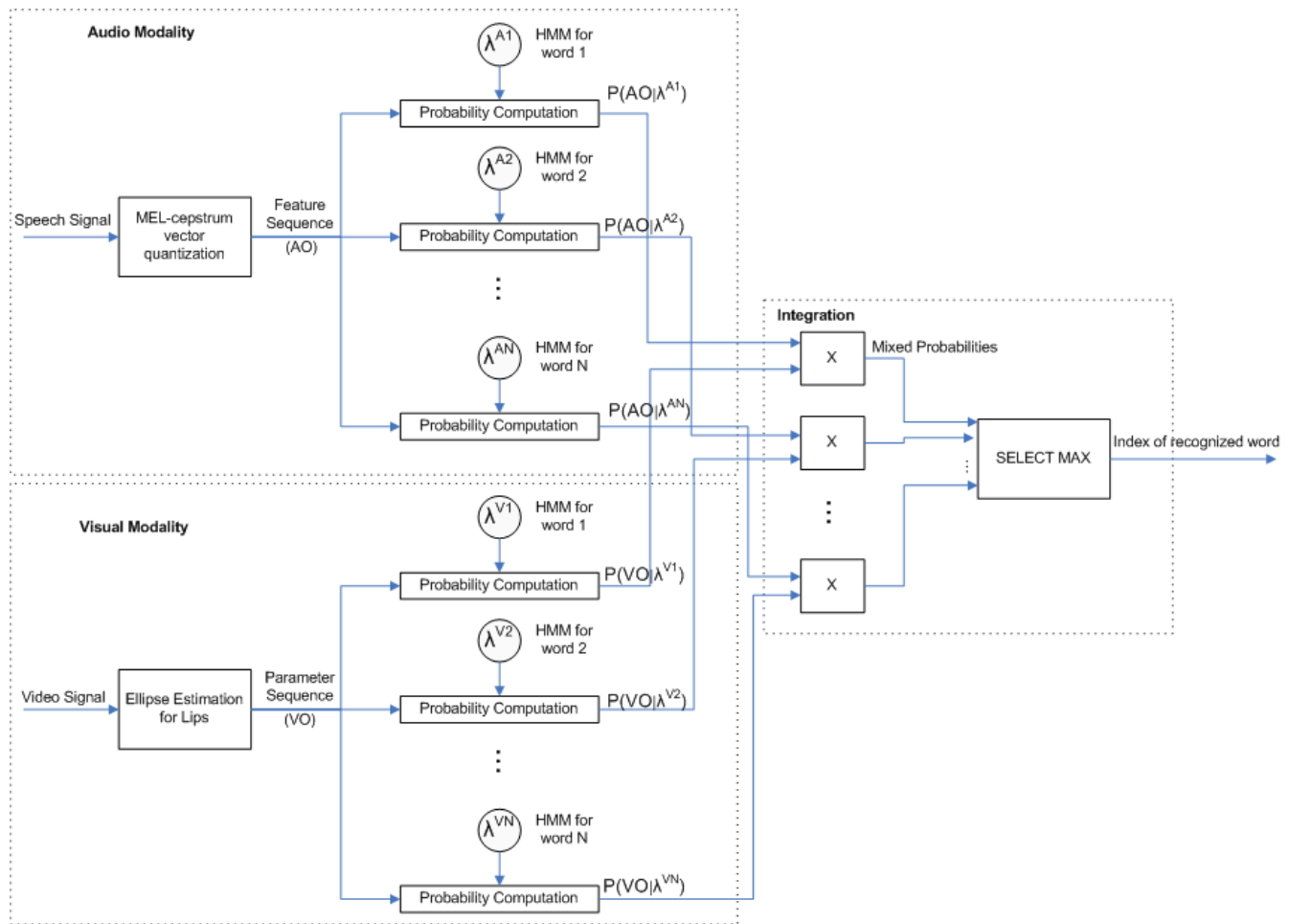


Fig. 2. EV/MA architecture.

'best' sequence, one must keep track of the state S_i for every k that maximizes the score. A word is classified by choosing the model that had the maximum final likelihood score. The recognition process and system architecture can be viewed schematically in Figure 1.

E. Accuracy and Problems

In ideal conditions, where the training conditions are nearly the same as the testing conditions, automatic speech

recognition systems work with 98-99% accuracy. However, when the testing conditions differ from the training conditions, which occurs when there is noise in the environment or a different microphone is used, performance degrades drastically; even below 20% accuracy using different microphones [16]. This degradation is unnecessary. Current computer-based speech recognition discards a very large amount of information: the entire visual modality. Since webcams are inexpensive, and they are even being shipped

with laptops, why not use what we have learned from the McGurk effect and perform recognition using a fusion of information from the auditory and visual modalities? This extra information is especially important when there is noise in the auditory signal.

III: EV/MA: AN ARCHITECTURE FOR AUDIO-VISUAL SPEECH RECOGNITION

This section presents an architecture for an automatic speech recognition system that uses information from both the visual and audio modalities in order to provide robust recognition even when there is noise in the audio signal.

The EV/MA architecture works by assuming that sufficient information for speech recognition in the visual modality can be extracted by representing the shape of the lips by an ellipse. An ellipse that closely fits the lips of the speaker is extracted from an image, and then the parameters for the ellipse are used as a feature vector to drive the input of a hidden Markov model that performs recognition solely in the visual domain. A separate, audio domain hidden Markov model performs recognition for the audio domain, and the two models are integrated. This architecture is shown schematically in Figure 2, and is computed as follows:

1. Normalizing the final, log Viterbi scores for the audio and visual models:

$$p^{Aw}(AO) = \text{Prob}\{AO|\lambda^{Aw}\} = \frac{\log(\delta_N^{Aw}(AO))}{\sum_w \log(\delta_N^{Aw}(AO))} \quad (1)$$

$$p^{Vw}(VO) = \text{Prob}\{VO|\lambda^{Vw}\} = \frac{\log(\delta_N^{Vw}(VO))}{\sum_w \log(\delta_N^{Vw}(VO))} \quad (2)$$

where AO is the auditory input for the observation, VO is the visual input for the observation, $p^{A/Vw}(AO/VO)$ is the probability of observing the input in the hidden Markov model, λ , in its respective modality for word w in the dictionary, and $\delta_N^{A/Vw}(\cdot)$ is the Viterbi likelihood score at state N , where N is the final state of the current model.

2. Multiplying the probabilities for the audio models by the probabilities for the visual models point-wise and then selecting the maximum resulting value:

$$w^* = \text{argmax}_{1 \leq w \leq W} (p^{Aw}(AO) \cdot p^{Vw}(VO)) \quad (3)$$

where w^* is the index of the word with the maximum mixed probability within the dictionary of size W .

The most difficult problem for this architecture is fitting the ellipse around the lips of the observer. The techniques used are outlined in the following section.



Fig. 3. Elliptical Approximation of the shape of a speaker's lips

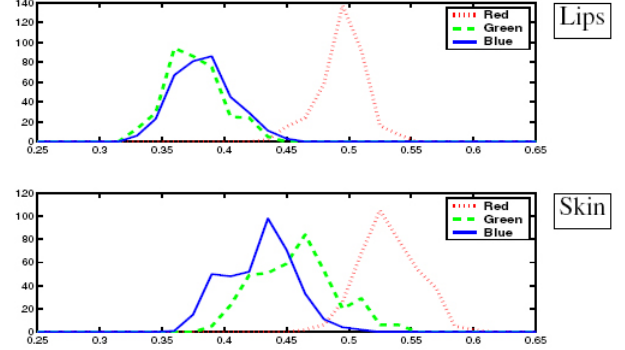


Fig. 4. Taken from [5], typical histograms of lips and skin.

IV. ELLIPSE EXTRACTION TECHNIQUES

The goal in this section is to develop a technique that fits an ellipse around the lips of a speaker for each frame in a video (Figure 3). The technique must work automatically for an entire video and must be accurate in order to build a model that is capable of recognizing speech based on visual input only.

Two techniques were implemented and evaluated. The first is a technique that uses an algorithm originally designed to extract cavity boundaries in noisy ultrasound images. The technique is called IMM/PDAF [2] and it uses a Kalman filter [10] with multiple, interacting models to estimate the boundary of the object. This algorithm works with intensity images, so a colour transformation is applied to the region of interest before running the segmentation. The second technique uses simple sum of squared differences tracker on four key templates for features on the lips.

A. IMM/PDAF Technique

This section will outline the process of estimating the parameters of the ellipse using the Kalman Filtering technique. First, the colour transformation will be explained; secondly, the model descriptions for the segmentation algorithm are revealed, followed by the process of estimating ellipse parameters inside a single frame and then the technique used to propagate the segmentation from frame to frame. Finally, a brief discussion of the performance of the technique will be presented.

i. The Colour Transformation

The goal of this process is to transform the region bounding the lips into a probability map where the intensity of a pixel corresponds to a probability that the pixel is part of the speaker's lips.

The transformation used is explained in detail in [5] and it takes advantage of the fact that skin and lip pixels have quite

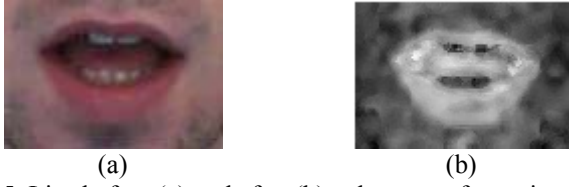


Fig. 5. Lips before (a) and after (b) colour transformation.

different RGB components. For both, red is quite prevalent. However, the difference between the red and green components is larger for lips than for skin. There is also more green than blue in skin pixels, and the two components are almost equal for lip pixels. Figure 4, taken from [5], explains this concept pictorially.

The technique works by taking three points for each pixel which are functions of the RGB components and fitting a parabola to these points. The idea is to have parabolas with high curvature for lip pixels and low curvature for skin pixels. This is done by fitting a parabola to the following three points for each pixel with coordinates (x, y) :

$$\begin{aligned} P_1(x, y) &: \begin{cases} -\alpha k(x, y) \\ B_{cor}(x, y) + \beta k(x, y) \end{cases} \\ P_2(x, y) &: \begin{cases} 0 \\ G_{cor}(x, y) \end{cases} \\ P_3(x, y) &: \begin{cases} 1 \\ \gamma k(x, y) \end{cases} \end{aligned} \quad (4)$$

where (α, β, γ) are parameters, $(R_{cor}, G_{cor}, B_{cor})$ are the respective colour channels that are corrected to reduce luminance dependency. This is computed with:

$$R_{cor}(x, y) = \frac{R(x, y)}{R(x, y) + 0.4L(x, y) + 0.4} \quad (5)$$

and similarly for G_{cor} and B_{cor} , where $L(x, y)$ is the luminance of pixel (x, y) . $k(x, y)$ is a normalized pseudo hue for the image, where the pseudo hue is calculated such that pixels with a greater difference between the red and green components are given lower values:

$$k(x, y) = \frac{h(x, y) - \operatorname{argmin}_{x, y}(h(x, y))}{\operatorname{argmax}_{x, y}(h(x, y)) - \operatorname{argmin}_{x, y}(h(x, y))} \quad (6)$$

where

$$h(x, y) = \frac{R_{cor}(x, y)}{G_{cor}(x, y) + R_{cor}(x, y)} \quad (7)$$

For lip pixels, $k(x, y)$ is high and G_{cor} and B_{cor} are close together, yielding a parabola with high curvature. For skin pixels, $k(x, y)$ is low and $G_{cor} > B_{cor}$. All three points have roughly the same elevation, so they define a parabola with low curvature.

Parabolas are fit to the three points using a quadratic interpolation technique, such that the parabola interpolates all three points. In other words, we are looking for a quadratic, $Q(u) = au^2 + bu + c$ that passes through the points P_1 , P_2 and P_3 :

$$Q(u) = TA \quad (8)$$

$$Q(u) = [u^2 \ u \ 1] \cdot [a \ b \ c]^T$$

$$A = MV$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} & & \\ & M & \\ & & \end{bmatrix} \begin{bmatrix} P_1(y) \\ P_2(y) \\ P_3(y) \end{bmatrix}$$

$$M^{-1}A = V$$

$$\begin{bmatrix} P_1(x)^2 & P_1(x) & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} P_1(y) \\ P_2(y) \\ P_3(y) \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} P_1(x)^2 & P_1(x) & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} P_1(y) \\ P_2(y) \\ P_3(y) \end{bmatrix}$$

Finally, the intensity of the resulting pixel is set to the curvature (second derivative) of the fitted parabola, $c = 2a$. This technique yields a probability map typical to that shown in Figure 5 with parameters (α, β, γ) chosen as $(0.4, 0.4, 2)$.

ii. IMM/PDAF for Elliptical Models

IMM/PDAF [2] is a spatial domain Kalman filtering technique that uses multiple models to segment the boundary of an object in a noisy image, such as one taken by ultrasound. It is an extension of the original PDAF algorithm [1] except it has the ability to switch between process models that have multiple trajectories or noise levels according to a set of transition probabilities. This means that the filter gain can be adjusted depending on the curvature of the boundary segmentation, which gives more accurate segmentations for irregular shapes.

The PDAF algorithm works by taking a seed point within an image, which represents the centre of the object, and then using a Kalman filter to estimate the radius of the boundary by treating ‘time-steps’ as increments in the angle of the boundary from the seed in polar coordinates, between 0 and 2π . It uses a constant velocity (circular) process model, and the observation model is defined by a number of candidate edge points (those with high one-dimensional first derivative values along the ray pointing to the current radius) and their sum after being multiplied by a normalized weighting factor that is proportional to the likelihood that the candidate edge point lies on the actual boundary and the magnitude of its edge value. This likelihood depends on the radius estimated for the previous angle-step, and it assumes a Gaussian distribution around the predicted radius of the current angle-step. The complete theoretical explanation and mathematical implementation can be found in [2].

Our implementation of IMM/PDAF is identical to the original, except it uses two elliptical process models with different levels of noise.

The radius of an ellipse at any angle $r(\theta)$, assuming that the centre is at the origin, can be defined by:

$$r(\theta) = ab / \sqrt{b^2 \cos^2(\theta - \phi) + a^2 \sin^2(\theta - \phi)} \quad (10)$$

where a is the length of the semi-major axis, b is the length of the semi-minor axis, and ϕ is the angle of the semi-major axis with respect to the image plane. The standard process model for a Kalman filter for IMM/PDAF is:

$$\mathbf{x}_j(k+1) = A_j \mathbf{x}_j(k) + \mathbf{v}_j(k) \quad (11)$$

where $\mathbf{x}_j(k)$ is the state vector at time k for $j = 1, \dots, M$, M is the number of trajectory models, A_j is the state transition matrix, and $\mathbf{v}_j(k)$ is the zero-mean Gaussian process noise vector defined by covariance matrix $Q_j(k)$. Therefore, since Equation 10 is not linear, an extended Kalman filter [9] must be used; which means that we must define the matrix A such that the process model is locally linear. This is done by using the state vector:

$$\mathbf{x}_j(k+1) = f_j(\mathbf{x}_j(k), \mathbf{v}_j(k))$$

$$\begin{bmatrix} r_j(k+1) \\ r_{\theta_j(k+1)} \end{bmatrix} = \begin{bmatrix} ab \\ \sqrt{b^2 \cos^2(\theta(k) - \phi) + a^2 \sin^2(\theta(k) - \phi)} \end{bmatrix} \quad (12)$$

***Please see Appendix for this equation, it is too large to fit in a column.

$$A_j(k+1) = \begin{bmatrix} 1 & \Delta\theta r_{\theta_j(k)} \\ 0 & \frac{r_{\theta_j(k+1)+\epsilon}}{r_{\theta_j(k)+\epsilon}} \end{bmatrix} \quad (13)$$

where $r_j(k)$ is the radius of the boundary from the seed point and $r_{\theta_j(k)}$ is its derivative with respect to angle θ , respectively, $\Delta\theta$ is the sampling angle and ϵ is a sufficiently small constant. For our implementation, the covariance for the process noise vector, $\mathbf{v}_j(k)$, was chosen as $Q_j(k) =$

$$\begin{bmatrix} \frac{\Delta\theta^4}{4} & \frac{\Delta\theta^3}{3} \\ \frac{\Delta\theta^3}{3} & \Delta\theta^2 \end{bmatrix} \sigma_{v_j}^2 \quad \text{where } (\sigma_{v_1}^2, \sigma_{v_2}^2) = (10^4, 10^2).$$

In [2], the search distance of the ray for a given angle was limited by two parameters, L_{min} and L_{max} . These represent the beginning and end points of the ray from the seed point in pixels, respectively. For our implementation, we used a band around the current elliptical model, where L_{min} is defined by an ellipse with its semi-major and semi-minor axes some distance inside the elliptical process model and L_{max} is defined by an ellipse with its semi-major and semi-minor axes some distance outside the process model. The actual bandwidths of these limits are dependent on the expected movement of the lips.

The parameters in Equation 10 and the process model of the Kalman filter can be viewed schematically in Figure 6.

iii. Fitting an Ellipse to the Segmentation

Once the IMM/PDAF segmentation was completed, it is represented as a vector of radius values that were obtained at each angle-step. This vector is used in a DD1 filter [14] to estimate the parameters for an ellipse that best fits the segmentation. A DD1 filter works in the same way as an extended Kalman filter [9], except that the Jacobian matrices

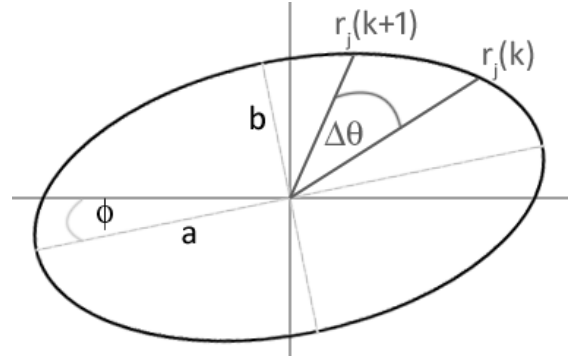


Fig. 6. Parameterization and process of elliptical model.

for the state transition model are replaced by divided differences. This technique for estimating ellipse parameters is similar to that described in [8].

Assuming that the ellipse is centered on a seed point, the state vector for the system is $\mathbf{x}_k = [r_k \ a_k \ b_k \ \phi_k]^T$ for each $\theta = 2\pi k/N$ from $k = 1$ to N , and our system is:

$$\mathbf{x}_{k+1} = \begin{bmatrix} a_k b_k / \sqrt{b_k^2 \cos^2(\theta_k - \phi_k) + a_k^2 \sin^2(\theta_k - \phi_k)} \\ a_k \\ b_k \\ \phi_k \end{bmatrix} + \xi_k \quad (14)$$

$$\mathbf{z}_k = \mathbf{r}_k + \boldsymbol{\eta}_k \quad (15)$$

where (a_k, b_k, ϕ_k) represent the parameters of the ellipse, as in Equation 10, r_k is the value obtained from the segmentation vector at angle step k , \mathbf{z}_k is the measurement vector, and ξ_k and $\boldsymbol{\eta}_k$ are the process and measurement noise sequences, respectively, with known covariance.

The estimated ellipse values are the means of the parameter values for $k = 2$ to N angle-steps. The first step is omitted since the process tends to converge after one step.

iv. Frame Propagation

Since the segmentation runs on a video, the parameterization of the models above must be automatically determined from the previous frame.

The new seed point is determined with a temporal Kalman filter on its position. It is exactly equivalent to the method explained in [1]: where a constant velocity kinematic process model is used and the observation model uses the centre of gravity of the segmentation as the observed position. The only difference is the fact that the seed is tracked at 15 Hz instead of 30 Hz, which affects the covariance matrix for the process.

The process model (Equations 12, 13) for the current frame conformed to the ellipse parameters estimated in the previous frame.

The initialization of the seed point and ellipse parameters for the first frame was conducted manually.

v. Performance and Discussion

With an implementation in MATLAB, ellipse estimation runs at 1Hz; this is unsuitable for real-time applications. Since most of the running time was spent fitting the parabolas to the points calculated for the colour transformation, there is still



Fig. 7. A typical lip segmentation using the IMM/PDAF technique.



Fig. 8. A segmentation that was attracted to false edges due to facial hair.



(a)



(b)

Fig. 9. Closed mouth IMM/PDAF technique segmentation (a), followed immediately by an open mouthed segmentation (b).

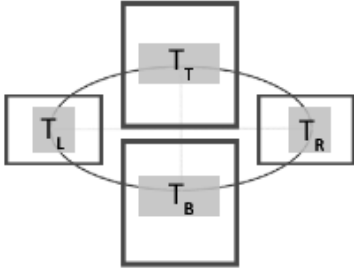


Fig. 10. Schematic representation of search area for SSD technique. $T_X, X \in \{L, T, R, B\}$ are the templates for the respective feature. The best match is searched for within its bounding box.

the possibility to use the IMM/PDAF technique (since it has been shown to run in real-time in [2]), but a faster colour transformation technique must be used.

Figure 7 shows a typical segmentation of the lips. As one can see, it is quite accurate in most cases, but there are frames in which the segmentation is not as accurate. One of these is shown in Figure 8, where the segmentation is attracted to the facial hair of the speaker. This occurs because the colour transformation creates a very dominant edge where there is a transition to pixels that are grey in colour. More weight is given to these edges and consequently, the boundary is pushed out. Histogram equalization techniques were attempted to increase the contrast of the colours inside the region of interest, but they did not improve accuracy.

Another issue occurs when there is a fast transition between a frame where there is a closed-mouth and then one where the mouth is open. This can take place when there is a bilabial consonant, followed by a vowel; such as in the word “fox”. Figure 9 shows two frames where this occurs, and it can be observed that the boundary does not extend far enough to reach the bottom of the lips. The likely reason for this is the fact that the parameters of the models are driven by the previous frame, therefore the elliptical process models constrain the growth of the boundary. This problem may be fixed by increasing the frame rate of the segmentation, but unfortunately, this is not possible for the moment since the algorithm only runs at 1 Hz.

This technique is currently unable to be used in practical applications unless a faster and more accurate colour transformation technique is discovered.

B. Sum of Squared Differences Technique

A sum of squared differences (SSD) tracker is simple: the position of an object in an image is the location that minimizes the point-wise sum of squared difference with a template image. That is:

$$(l, t)^* = \underset{l, t}{\operatorname{argmax}} \left(\sum_{x=l}^{l+W-1} \sum_{y=t}^{t+H-1} (R(x, y) - R^*(x-l, y-t))^2 + \sum_{x=l}^{l+W-1} \sum_{y=t}^{t+H-1} (G(x, y) - G^*(x-l, y-t))^2 + \sum_{x=l}^{l+W-1} \sum_{y=t}^{t+H-1} (B(x, y) - B^*(x-l, y-t))^2 \right) \quad (16)$$

where (l, t) are the coordinates of the top-left point of the current region being examined, W and H are the width and height of the template, respectively, $R|G|B(x, y)$ is the intensity in of the image being searched and $R|G|B^*(x, y)$ is the intensity in the template at coordinate (x, y) .

This technique is applied to search for templates representing the top, bottom, and left and right corners of the mouth in local areas represented by Figure 10.

Once the best locations of the templates are found, the parameters for an ellipse (a, b, ϕ) are calculated using the four points (after centering them within the template) as constraints such that the extremes of the ellipse interpolate the chosen points:

$$E_{\Gamma \in \{L, R, T, B\}}(x, y) = \begin{cases} l_{\Gamma}^* + \left\lfloor \frac{W_{\Gamma}}{2} \right\rfloor \\ t_{\Gamma}^* + \left\lfloor \frac{H_{\Gamma}}{2} \right\rfloor \end{cases} \quad (17)$$

$$\begin{aligned} a &= E_R(x) - c(x) \\ b &= E_B(y) - c(y) \\ \phi &= 0 \end{aligned} \quad (18)$$

$$c(x, y) = \begin{cases} \left\lfloor \frac{E_R(x) - E_L(x)}{2} \right\rfloor + E_L(x) \\ \left\lfloor \frac{E_B(y) - E_T(y)}{2} \right\rfloor + E_T(y) \end{cases} \quad (19)$$

where E_L, E_R, E_T, E_B are the tracked locations for the left corner, right corner, top and bottom of the lips, respectively and c is the centre point of the ellipse. The parameter ϕ is set to 0 since accuracy is usually not affected by its value due to the fact that the speaker does not often tilt their head very much from side-to-side while speaking to a computer.

The locations were tracked across frames by repositioning the regions of search around the best matches found for the previous frame.

Figure 11 shows the results of a typical ellipse approximation. As one can see, the approximation is



Fig. 11. Closed mouth SSD technique segmentation (a), followed immediately by an open mouthed segmentation (b).

extremely accurate, even for sequences of drastic change. Recall that the IMM/PDAF technique had problems with this exact sequence. With a MATLAB implementation on a 2.0 GHz processor, the technique runs at 13 Hz, which can easily translate to real-time with a C++ implementation.

Naturally, this technique is not likely to succeed with the same accuracy in situations with different lighting conditions, skin colours and orientation since it uses static, lighting independent templates. This technique was implemented in order to provide accurate training and testing parameters for the EV/MA architecture, assuming that accurate elliptical parameters could be estimated.

V. AN EV/MA IMPLEMENTATION

This implementation of EV/MA is concerned with the typical task of recognizing the words ‘one’ through ‘ten’. This section will explain the hidden Markov model parameters and vector structures for the audio modality and then for the video modality. All hidden Markov models were implemented in MATLAB using Kevin Murphy’s HMM Toolbox [13].

A. The Audio Modality

Feature vectors were sampled in windows of 8 ms in duration with 4 ms of overlap. This creates 249 samples for a one second audio input. Twelve cepstral coefficients were used for each vector, plus their rates of change, which yields a 24 feature vector. These vectors were computed using the VOICEBOX Toolbox for MATLAB [4].

Each hidden Markov model trained uses 5 states and a mixture of 9 Gaussians for each state, each with diagonal covariance matrices initialized with uniform random numbers between 0 and 1. These parameters were chosen using the guidelines explained by Rabiner in [15], although performance did not differ if slightly different parameters were used.

B. The Visual Modality

Feature vectors in this domain were sampled at 15 Hz offline, due to the fact that the MATLAB implementation did not run in real-time. This creates 15 samples for a one second long video input. The feature vectors consisted of the values for the radius of the semi-minor and semi-major axes of the fitted ellipse in pixels, and their rates of change; yielding a 4 feature vector. One could argue that a ratio between the semi-minor and semi-major axes should have been used since the values depend on each other, yielding a two feature vector. In this implementation, differences in scale were controlled for by keeping the head of the speaker stable,

therefore the ratio was not necessary. Future implementations should work to avoid this assumption.

Each hidden Markov model uses 3 states and a mixture of 3 Gaussians for each state, also with diagonal covariance matrices initialized in the same way. Three states were chosen for the hidden Markov model by examining the movement of the lips for each of the spoken words. On average, it was observed that the lips made three larger, separate movements for each word. Take for instance, the word ‘one’: The articulation begins with pursing of the lips to begin the /w/ sound, then the lips are opened to create the /ah/ sound, then the lips are closed again to create the nasal /n/ sound.

VI. PERFORMANCE OF EV/MA

Performance of the EV/MA architecture was tested by training the hidden Markov models under optimal conditions, and then testing their recognition performance on input stimuli where the signal from the audio modality was perturbed by differing amounts of additive Gaussian noise.

A. Training

An HMM was trained for each word using 20, one second long samples for each word. The samples were obtained by recording with an average quality webcam with an embedded microphone (Logitech Communicate STX webcam, ~\$60 CDN). The audio and visual streams were separated into one second long, synchronized chunks, manually, using video editing software. The video files were encoded as 640x480 resolution .avi files with Cinepak compression set to 100% quality and the audio files were encoded with 16-bit .wav files sampled at 32000 Hz.

The words were spoken by the same speaker, who attempted to keep their head as steady as possible, but different intonations were used for each sample. Samples were recorded over two different recording sessions.

B. Testing

Testing was performed on another set of 20 samples, also processed in the same way above, and selected from the two different recording sessions. However, before computing the MEL-cepstrum during recognition, a controlled amount of Gaussian noise was added to the 16-bit power spectrum of the audio signal.

The amount of Gaussian noise added was determined by taking the variance of the data section in the .wav file (which naturally has zero mean) and generating pseudo-random, zero-mean, Gaussian distributed noise with a variance equal to a constant multiplied by the square of the standard deviation of the audio signal:

$$\mathbf{S}^* = \mathbf{S} + \mathbf{v} \quad (20)$$

$$\mathbf{v} \sim N(0, \tau \cdot sd(\mathbf{S})^2) \quad (21)$$

where \mathbf{S} is the vector of length 32000 containing the original 16-bit, zero-mean signal, \mathbf{S}^* is the perturbed signal, and \mathbf{v} is a vector of size 32000 sampled from a zero-mean Gaussian

distribution with variance equal to a constant, τ , multiplied by the variance of the original signal. A new noise vector was sampled for each signal that the system was to recognize. Performance was tested in conditions with varying levels of the τ parameter, with values of 0, 0.25, 0.5, 0.75 and 1.0 variances.

The video signal was not perturbed in any way from the conditions used during training

C. Results

Table 1 summarizes the results obtained for each noise condition. As one can see, the EV/MA architecture provides a vast improvement over the audio-only recognition system when noise is added to the signal. Performance also degrades much more slowly. It should also be noted that the visual-only recognizer achieved 96% accuracy, although this performance was obtained using the same conditions in testing and training.

Most of the errors made by the system for noisy conditions were misclassifications of more sonorant words (one, five, nine) for words containing fricatives (eight, two, ten), which sound like Gaussian noise.

TABLE 1
PERFORMANCE OF EV/MA VERSUS A CLASSICAL ASR SYSTEM WITH
DIFFERING LEVELS OF NOISE IN THE AUDITORY SIGNAL (N=20).

τ	Audio Only (performance)	EV/MA (performance)
0	100.0%	100.0%
0.25	87.0%	97.5%
0.5	67.0%	86.5%
0.75	57.5%	81.0%
1.0	49.5%	79.0%

VII. DISCUSSION

As expected, the EV/MA architecture provides a vast improvement over classical, auditory-only automatic speech recognition. This is due to the fact that the visual domain provides extra information to disambiguate cause for uncertainty when there is noise in the auditory signal. These results are very consistent with human speech recognition, where the visual modality becomes extremely important when environments are noisy.

However, this paper assumed near-optimal conditions in the visual modality, where the same speaker was used, in mostly constant lighting conditions and where there was minimal movement of the head. The ellipse approximation is the most difficult problem for this speech recognition technique.

Future work will have to fully test the robustness of the SSD tracking technique in environments with different lightings conditions, noise, skin colours, facial hair, orientations, etc. This technique was employed in order to test the performance of EV/MA assuming that we could obtain accurate approximations. It will have to be improved or replaced altogether.

The IMM/PDAF technique has greater promise since it has performed very well in extremely noisy conditions, and it also provides more information. A better colour transformation or an implementation of IMM/PDAF that makes appropriate use of colour information needs to be discovered. One idea would be to adjust the observation model such that the weighting factors of the edges use colour information optimally.

Future work would also examine how to find more features in the visual domain. In order to provide scale invariance, half of the features in the feature vector used would have to be discarded since a ratio would be used. There is much more information in the visual domain that is not being used: for instance, segmentation of the tongue and inner boundary of the lips may glean more useful information. It would also be a good idea to find out how the entire segmentation vector found by IMM/PDAF could be used as features since a great deal of information is lost in the conversion to an ellipse.

The human brain is full of feedback mechanisms. It is likely that information from the audio domain affects perception in the visual domain, and vice-versa. It would be interesting to look at how the audio information could help drive the segmentation since it is sampled more quickly. One might be able to analyze the spectra of the audio signal and disambiguate the values for the parameters in the video segmentation. It would also be interesting to look for correlations between the audio and visual signals and reproduce speech by using the visual domain only.

Multi-modal perception is a new and exciting field. There is a great deal left to be discovered. This paper only provides evidence of its usefulness. Further work may explore the biological plausibility of hidden Markov models to see if an architecture such as this has any value for actually understanding how humans understand speech. This is the direction that the author would like to take for subsequent research.

VIII. CONCLUSION

This paper presented an overview of classical automatic speech recognition and then an architecture for audio-visual speech recognition, including the rationale for implementing one.

The architecture uses separate, hidden Markov model recognizers for the auditory and visual domains. Features for the visual domain are found by finding the best parameters for an ellipse that approximate the shape of the lips and features for the auditory domain use a classical MEL-cepstrum vector quantization. The approximation of the ellipse was implemented using a Kalman filtering technique, and then with a simple sum of squared differences technique. The Kalman filtering technique was shown to be limited by its colour transformation and the sum of squared differences technique was implemented to show the robustness of the EV/MA architecture for speech recognition in noisy environments. The EV/MA architecture provided a vast improvement over auditory-only recognition.

Future work will explore how to improve feature extraction in the visual domain in order for the techniques to be robust in all conditions and its usefulness in understanding human multi-modal perception.

ACKNOWLEDGMENTS

The author would like to thank Dr. Purang Abolmaesumi for helpful brainstorming meetings, his guidance during the process of developing this idea and agreeing to supervise the project. The author would also like to thank Dr. Roger Browse and Dr. Brian Butler for their encouragement to pursue the idea further. Finally, the author would like to thank Dr. Kevin Murphy at UBC for his easy-to-use and well-documented HMM Toolbox for MATLAB. Without it, this project would not have been completed in time.

REFERENCES

- [1] P. Abolmaesumi, S.E. Salcudean, W.H. Zhu, M.R. Sirouspour, and S.P. DiMaio, "Image-Guided Control of a Robot for Medical Ultrasound," *IEEE Transactions on Robotics and Automation*, Vol. 18(1): pp. 11-23, 2002.
- [2] P. Abolmaesumi, and M.R. Sirouspour, "An Interacting Multiple Model Probabilistic Data Association Filter for Cavity Boundary Extraction from Ultrasound Images," *IEEE Transactions on Medical Imaging*, Vol. 23(6): pp. 772-784, 2004.
- [3] L.E. Baum, "An Inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, Vol. 3: pp. 1-8, 1972.
- [4] M. Brookes, "VOICEBOX: Speech Processing Toolbox for MATLAB," <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html>.
- [5] N. Eveno, A. Caplier, and P.Y. Coulon, "A New Color Transformation for Lips Segmentation," *IEEE Fourth Workshop on Multimedia Signal Processing*, pp. 3-8, 2001.
- [6] C.A. Fowler, and D.J. Dekle, "Listening with eye and hand: cross-modal contributions to speech perception," *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 17(3): pp. 816-828, 1991.
- [7] K.P. Green, P.K. Kuhl, A.M. Meltzoff, and E.B. Stevens, "Integrating speech information across talkers, gender, and sensory modality: Female faces and male voices in the McGurk effect," *Perception & Psychophysics*, Vol. 50: pp. 524-536, 1991.
- [8] J. Guerrero, S.E. Salcudean, J.A. McEwen, B.A. Masri, and S. Nicolaou, "Deep Venous Thrombosis Screening System Using Numerical Measures," *Proceedings of the 25th Annual International Conference of the IEEE EMBS*, Vol. 1: pp. 894-897, 2003.
- [9] S.J. Julier, and J.K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II*, SPIE: 1997.
- [10] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME - Journal of Basic Engineering*, Vol. 82: pp. 35-45, 1960.
- [11] D.W. Massaro, M. Tsuzaki, M.M. Cohen, A. Gesi, and R. Heredia, "Bimodal Speech Perception: An examination across languages," *Journal of Phonetics*, Vol. 21: pp. 445, 1993.
- [12] H. McGurk, and J.W. MacDonald, "Hearing lips and seeing voices," *Nature*, 264: pp. 746-748, 1976.
- [13] K. Murphy, "Hidden Markov Model Toolbox for MATLAB," <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>, 1998.
- [14] M. Nørgaard, N.K. Poulsen, and O. Ravn, "Advances in Derivative-Free State Estimation for Nonlinear Systems," *Technical report: IMM-REP-1998-15, Department of Mathematical Modeling, Informatics and Mathematical Modeling, Technical University of Denmark (DTU)*, 1998.
- [15] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77(2): pp. 257-286, 1989.
- [16] M.G. Rahim, and B.H. Juang, "Signal Bias Removal by Maximum Likelihood Estimation for Robust Telephone Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, Vol. 4(1): pp. 19-30, 1996.
- [17] L.D. Rosenblum, and H.M. Saldana, "An audiovisual test of kinematic primitives for visual speech perception," *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 22(2): pp. 318-331, 1996.
- [18] L.D. Rosenblum, M.A. Schmuckler, and J.A. Johnson, "The McGurk effect in infants," *Perception & Psychophysics*, Vol. 59(3): pp. 347-357, 1997.
- [19] M. Seadle, "Digital Audio Best Practices," *CDP Digital Audio Working Group*, Ver. 2.0: 2005.
- [20] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions on Information Theory*, Vol. IT-13: pp. 260-269, 1967.
- [21] F. Zheng, G. Zhang, and Z. Song, "Comparisons of Different Implementations of MFCC," *Computer Science & Technology*, Vol. 16(6): pp. 582-589, 2001.
- [22] E. Zwicker, and E. Terhardt, "Analytical expressions for critical band rate and critical bandwidth as a function of frequency," *Journal of the Acoustical Society of America*, Vol. 68: pp. 1523-1525, 1980.

APPENDIX

Rest of Equation 12:

$$\frac{-ab(2bc\cos^2(\theta(k) - \phi) - 2b^2\cos(\theta(k) - \phi)\sin(\theta(k) - \phi) + 2asin^2(\theta(k) - \phi) + 2a^2\sin(\theta(k) - \phi)\cos(\theta(k) - \phi))}{2(b^2\cos^2(\theta(k) - \phi) + a^2\sin^2(\theta(k) - \phi))}$$