Comparing Visual Features for Lipreading

Yuxuan Lan[1], *Richard Harvey*[1], *Barry-John Theobald*[1], *Eng-Jon Ong*[2] *and Richard Bowden*[2]

[1]School of Computing Sciences, University of East Anglia, UK
[2]School of Electronics and Physical Sciences, University of Surrey, UK
{y.lan,r.w.harvey,b.theobald}@uea.ac.uk,{e.ong,r.bowden}@surrey.ac.uk

Abstract

For automatic lipreading, there are many competing methods for feature extraction. Often, because of the complexity of the task these methods are tested on only quite restricted datasets, such as the letters of the alphabet or digits, and from only a few speakers. In this paper we compare some of the leading methods for lip feature extraction and compare them on the GRID dataset which uses a constrained vocabulary over, in this case, 15 speakers. Previously the GRID data has had restricted attention because of the requirements to track the face and lips accurately. We overcome this via the use of a novel linear predictor (LP) tracker which we use to control an Active Appearance Model (AAM).

By ignoring shape and/or appearance parameters from the AAM we can quantify the effect of appearance and/or shape when lip-reading. We find that shape alone is a useful cue for lipreading (which is consistent with human experiments). However, the incremental effect of shape on appearance appears to be not significant which implies that the inner appearance of the mouth contains more information than the shape.

Index Terms: lip-reading, feature extraction, feature comparison, tracking

1 Introduction

The use of lip-reading has been documented since the 16th century and hearing-impaired people often use lip-reading as an adjunct to understanding fluent speech. When it comes to automating the process, there are many challenges compared to conventional audio recognition. Firstly audio speech has well-defined units known as a phonemes and there are pronunciation dictionaries that give the mapping between words and phonemes. Secondly, the data rate for uncompressed audio rarely exceeds 100 kbits s^{-1} whereas compressed video can easily have a rate 50 times higher. Thirdly there is a consensus opinion that audio features should be based on the mel-frequency cepstal coefficients whereas for visual speech the choice of features is rather wide. This paper is about this latter problem: the choice of visual features.

The literature has generated several types of visual features[1], which have been broadly categorised as those depending on pixels (in [2] this is referred to as the "bottom-up" approach because it uses few models) or those based on models (the "top-down" approach in [2]). Although this definition is rather general (there are a great number of possible models), interest has tended to focus on methods which operate just on intensities in regions of interest (such as the method used in [1]) or those which model the shape of the mouth (as in [2] for example). The question as to which was superior, appearance- or shape-based features, was first examined in [3] via Active Shape Model (ASM) features augmented with intensity profile information on the Tulips 1 database (four words each spoken twice by 12 subjects). They show a 8.33% difference between intensity-related features and shape but there are no confidence intervals or error bars and the technology of the time meant that that the database was rather restricted. In [2] this question was revisited using greyscale features called sieves compared to Active Shape Models on a small database known as AVletters (three repetitions by ten talkers of the letters 'A' to 'Z'). The conclusion was that the greyscale and shape-based methods performed with similar error rates but, via a McNemar's test, they were able to show that they failed in different ways. This led to the use of Active Appearance Models (AAMs) but there was no separate analysis of the shape and appearance components so the role of shape and appearance was not fully resolved.

In this paper we re-implement the Active Appearance Model on a much more challenging task known as the GRID dataset [4] which consists of sentences spoken at high speed in a variety of accents[2]. These data are tricky to track so we introduce a new form of tracker, known as the Linear Predictor, or LP, tracker that allows AAMs to be fitted to these data. Hence we hope to resolve the interplay between shape and appearance for lip-reading.

2 Features

2.1 AAM

The *shape*, $\mathbf{s}$, of an AAM is defined by the concatenation of the x and y-coordinates of n vertices that form a two-dimensional triangulated mesh: $\mathbf{s} = (x_1, y_1, \ldots, x_n, y_n)^T$. A compact model that allows a linear variation in the shape is given by,

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

where $\mathbf{s}_0$ is the base shape or mean of all the shapes and $\mathbf{s}_i$ are the shapes that are the eigenvectors corresponding to the m largest eigenvectors. The coefficients p_i are the shape parameters. Such a model is usually computed by applying Principal Component Analysis (PCA) to a set of shapes hand-labelled in a corresponding set of images.

The *appearance*, $A(\mathbf{x})$, of an AAM is defined by the pixels $\mathbf{x}$ that lie inside the base mesh $\mathbf{s}_0$. AAMs allow linear appearance

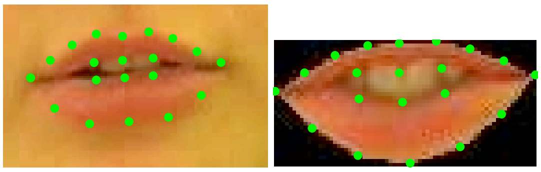
[1] The most recent review of methods is [1] which is focussed on audio-visual recognition in which the purpose of the video feature is to provide complementary information to the audio.

[2] GRID dataset is available to download via http://www.dcs.shef.ac.uk/spandh/gridcorpus/.

variation, so $A(\mathbf{x})$ can be expressed as a base appearance $A_0(\mathbf{x})$ plus a linear combination of l appearance images $A_i(\mathbf{x})$:

$$A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{l} \lambda_i A_i(\mathbf{x}) \tag{2}$$

where λ_i are the appearance parameters. As with shape, the base appearance A_0 and appearance images A_i are usually computed by applying PCA to the (shape normalised) training images [5]. A_0 is the mean shape normalised image and the vectors A_i are the (reshaped) eigenvectors corresponding to the l largest eigenvalues. An example of $\mathbf{s}$, $\mathbf{s}_0$, and A is given in Figure 1.



(a) $\mathbf{s}$

(b) $\mathbf{s}_0$ and $A(\mathbf{x})$

Figure 1: (a): landmarks $\mathbf{s}$ plotted on top of the original image. (b): mean shape landmarks $\mathbf{s}_0$, and the appearance image $A(\mathbf{x})$, where $\mathbf{x} = (x, y)^{\mathrm{T}} \in \mathbf{s}_0$. There is a guided warp from (a) to (b) using the correspondence between the mesh defined by $\mathbf{s}$ and that of $\mathbf{s}_0$, while the values of $A(\mathbf{x})$ are computed using a bilinear interpolation.

Although the shape and the appearance of an AAM can be used separately as features for lipreading, a combination of the two is likely to be a more discriminative feature. A primitive AAM feature is formed by concatenating the appearance parameters with the shape parameters:$(p_1, ..., p_m, \lambda_1, ..., \lambda_l)^T$, which we denote as *aam_cat*. A statistical approach is adopted in [6], where a PCA is applied to both the shape and the appearance, which creates a more compact, and most importantly, de-correlated feature, denoted here as *aam_pca*.

2.2 Tracking AAM landmarks using an LP tracker

The first stage of our tracking algorithm uses a set of Linear Predictors (LPs). The basis of an LP is that a point with coordinates $\mathbf{c} = [c_x, c_y]^T$ in an image taken from a video sequence, frame n, moves an amount $\mathbf{t} = [t_x, t_y]^T$ to frame $n + 1$. The assumption is that $\mathbf{t}$ is related to the measured change in intensity via

$$\mathbf{t} = \mathbf{H}\delta\mathbf{p} \tag{3}$$

where $\mathbf{H}$ is some learnt mapping between intensity differences and position, and

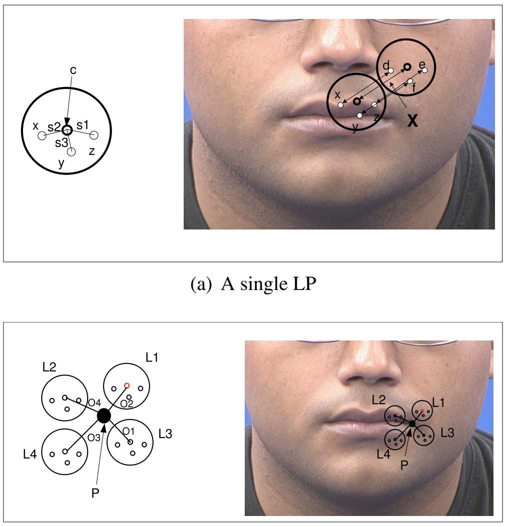
$$[\delta\mathbf{p}]_i = V_i^{(n+1)} - V_i^{(n)} \tag{4}$$

where $V_i^{(n)}$ is the i^{th} support pixel grey-value in frame n.

Each point, $\mathbf{c}$, has an associated pixel support region which is defined via a set of (x, y) offsets, $\mathbf{S}$. Each point that we wish to track is therefore represented by a four-tuple vector

$$L = \{\mathbf{c}, \mathbf{H}, \mathbf{V}, \mathbf{S}\} \tag{5}$$

where $\mathbf{c}$ is the location of the point to be tracked, $\mathbf{H}$ is the learnt mapping for that point, $\mathbf{S}$ are offsets giving the support region, and $\mathbf{V}$ are the values of the support pixels.



(a) A single LP

(b) A flock of LPs

Figure 2: LP tracker.

Here, the offset positions, $\mathbf{S}$, are chosen as 80 points randomly positioned within a 30-pixel radius. To improve tracking each LP is grouped into a rigid flock. Each flock has 200 LPs. To track the lips and eyes we use 30 landmarks: each was associated with a rigid flock.

The training algorithm is quite subtle and is described in [7]. It allows components of a flock to be accepted or regected on the basis of the effectiveness at predicting $\mathbf{t}$ during training. The final displacement of a flock is the mean of the predicted displacements of its member LPs. [3]

Each person-specific LP is trained using between 9 to 31 training images. And for each image, a set of 30 landmarks are manually positioned on the contour of eyes and lips. See Figure 3 for examples of some of the landmarks. Note that landmarks around the eyes are tracked purely for the benefit of AAM tracking later.

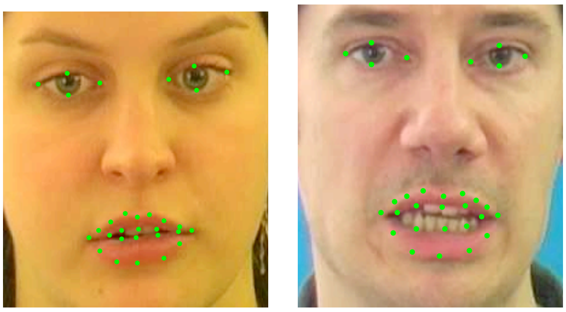


Figure 3: Examples of LP tracked landmarks.

In the second parse of the tracking, a person-specific AAM face-model is trained using the same training images as by the LP tracker. The LP tracked landmarks are then tracked again by the

[3] For examples of LP tracking results, see `http://www.ee.surrey.ac.uk/Projects/LILiR/update/ej/tracking_web.html`.

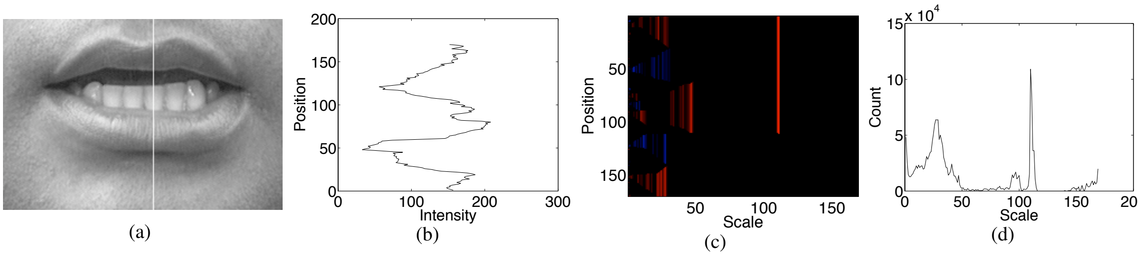


Figure 4: A vertical scan-line from a greyscale version of the mouth sub-image (a) is shown as an intensity plot (b). The granularity spectrum from an m-sieve with positive/negative granules shown in red/blue (c). These granules are then counted, or summed, over all scan-lines to produce the scale-histogram (d).

AAM tracker that is seeded on the LP landmarks frame by frame. It is worth pointing out that the AAM tracking is optional, and the result of this slight re-adjustment is a set of landmarks that are more consistent and "AAM-like". We find the use of LPs to be highly necessary since, common practice, in which an AAM is initialised on tracked landmarks of previous frame [2], fails on the GRID data.

To extract lip-only AAM features, a lip AAM model is trained using only landmarks on the lips. Enhanced LP landmarks are then projected onto the model from which the feature is computed. Figure 1 shows an example of the shape and the appearance of AAM from an image created using this method.

2.3 Sieve

The second type of feature derives from *sieves*, [8], which are a class of scale-space filters. The one-dimensional variants can be described as a cascade of filters such that the signal at scale s is $x_s = f_s(x_{s-1})$ where x_0 is the original signal and $f_s(\cdot)$ is a scale-dependent operator and is one of the greyscale opening $\mathcal{O}_s$, closing $\mathcal{C}_s$, $\mathcal{M}_s$, or $\mathcal{N}_s$ operators where $\mathcal{M}_s = \mathcal{O}_s\mathcal{C}_s$, $\mathcal{N}_s = \mathcal{C}_s\mathcal{O}_s$, $\mathcal{O}_s = \psi_s\gamma_s$ and $\mathcal{C}_s = \gamma_s\psi_s$. ψ_s is defined as:

$$\psi_s(x_{s-1}(n)) = \min_{p\in[-s,s]} z_{s-1}(n+p) \qquad (6)$$

$$z_s(n) = \max_{p\in[-s,s]} x_{s-1}(n+p) \qquad (7)$$

with γ_s *mutatis mutandis* with max and min swapped. An important property of sieves, and one which gives them their order-N complexity [9], is that the small scales are processed before the larger ones – they are a cascade with the output from the small scale feeding into the larger scale. In the original literature the morphological operator was replaced with a recursive median filter (the so called m-sieve) but nowadays the variants given above are more common.

When applied to lip-reading outputs at successive scales can be differenced to obtain *granule functions* which identify regional extrema in the signal by scale. These difference signals form a scale signature which should change as the mouth opens. The feature extraction system follows that used in [2] and is illustrated in Figure 4.

3 Database

For the experiments described in this paper, we use an audio-visual speech database called GRID[4] which consists of recordings of 1000 utterances per speaker, and a collection of 34 speakers. Each sentence is created using a fixed grammar model with 6 components: command, colour, preposition, letter, digit, and adverb, with a vocabulary size of 51 word. An example of such a sentence is "bin blue at f two soon". Visual speech was captured at a frame rate of 25 frame/second and was converted to MPEG-1 format with datarate of 6Mbits s^{-1}. The resolution of the MPEG movies is 720×576 pixels. The database also includes a word level audio alignment using flat-start force alignment, and marks the beginning and the end of each word during speech. The lip region has been semi-automatically detected [10], and is specified by a bounding box, from which a lip sub-image can be extracted for computing features, including sieve1d, 2D DCT and eigen-lips [11]. Some examples of lip sub-images are shown in Figure 5. Two types of bounding boxes are included in the dataset. One is the tracked bounding box which is centralised on the center of lip region, the other is the static bounding box that is positioned on the mean location of tracked bounding boxes of the whole sequence. To be comparable with experiments done in [10], the experiments described in this paper use the 2D DCT featues that are supplied with the GRID dataset, which were computed from static bounding boxes. Sieve1D feature and eigen-lip feature are computed on lip sub-images within the tracked bounding box, so that they are not affected by the movement of head, although this does introduce some tracking noise..

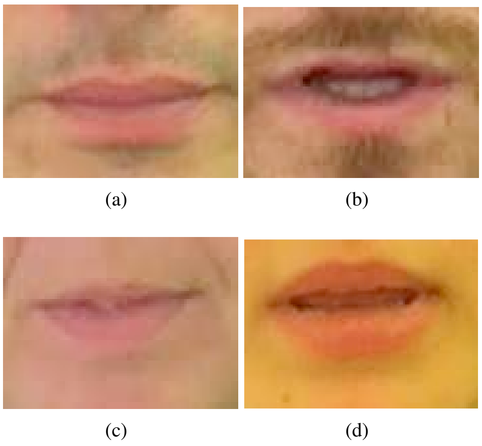


Figure 5: Example lip sub-images from GRID database.

4 Experiments and results

This experiment uses all utterances from 15 speakers (speakers 1–12, 20, 23, and 24), nine of whom are males. There is a check to remove any sequences that are incomplete or damaged during compression. For classification we use Hidden Markov Models (HMMs) which are the method of choice for speech recognition and have been shown to be successful for lip-reading [2, 12]. The standard HMM toolkit, HTK [13], is applied here for building and manipulating HMMs.

A total of 51 HMMs, one for each word, are trained. In addition, an extra HMM is dedicated to model non-speech movements, the 'silence' model . Left-right HMMs with a diagonal covariance Gaussian Mixture Model (GMM) associated with each state are used. The number of states in each HMM is decided based on a principle of one state per phoneme, and the number of components in each Gaussian mixture is four. HMMs are initialised using the Viterbi algorithm, via HTK module `HInit`. Baum-Welch re-estimation is then used (via `HRest`) to refine each individual HMM, followed by a series of embedded training via `HERest`, which updates all HMMs simultaneously.

A total of eight different visual features are tested, four of which are AAM-derived features, which including, *app* and *shape*, the appearance parameters and the shape parameters of an AAM, and *aam_pca* and *aam_cat*, denoting two different approaches of combining the shape and appearance parameters defined earlier. We refer to sieve features computed on lip subimages as $sieve1d$, and those computed on shape normalised appearance images $A(\mathbf{x})$ (Figure 1(b)), as *app_sieve*. Lip subimages are also used to compute *eigen_lip* features and 2D DCT features, the latter of which is actually supplied with the GRID dataset. Eigenlips are computed via a PCA of the intensities in the lip-subimage and retaining the eigenvalues that account for 95% of the variation. In all cases the features are augmented with Δ and $\Delta\Delta$ coefficients (velocity and acceleration)

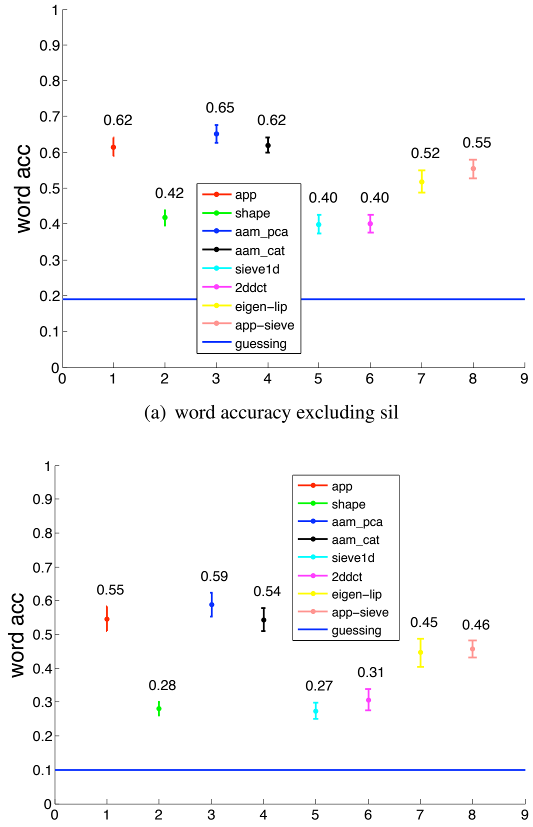
To test the robustness of the features across speakers, we designed a set of speaker independent experiments using a strategy of 15-fold cross-validation: for each fold, a different speaker is held-out for testing and the classifier is trained on the data of the remaining speakers. [4] Performance of a classifier is measured using the word accuracy rate Acc, where

$$Acc = \frac{H - I}{N} \tag{8}$$

in which, N is the total number of word instances to be regonised, H is the number of correctly recognised word instances, and I is the number of insertion errors: for example, if the reference sentence is 'a c' and the recognised sentence is 'a b c', then 'b' is an insertion error.

Results from the HMM classifier using the eight different visual features are plotted in Figure 6. It is worth pointing out that although Figure 6(b) evaluates only one component of the whole vocabulary, the ranking order of the features in terms of their word accuracy stays the same. This is a likely indication that the performance display in Figure 6 is representative across all classes.

Looking at both graphs in Figure 6, one can see a clear trend that the set of AAM features with appearance parameters, i.e., *app*, *aam_pca*, *aam_cat*, outperform other type of features. However, when using only shape parameters of an AAM (*shape* feature), classifier performance decreases significantly. If the shape and the appearance components are combined properly, here by using a PCA in the case of aam_pca, a slight improvement can be gained. In other words, given the choice between shape and appearance, one would always choose appearance.



(a) word accuracy excluding sil

(b) word accuracy on letter

Figure 6: 15-fold cross-validation results of HMM classifiers on various visual features, evaluated on the GRID database. The mean word accuracy rate is plotted with errorbar, showing ±1 standard error to the estimated mean accuracy. In (a), the word accuracy rate is calculated across all words excluding silence, and the chance by guessing is 0.19. (b) shows the accuracy rate on only the digits, when the chance by guessing is 0.10.

One also notices the effect of image shape normalisation on features computed using pixel intensity values. For example,the $sieve1d$ features are computed on a lip sub-image that contains affine variation (see Figure 5 for examples). These variations do not exist in the shape-normalised appearance image $A(\mathbf{x})$, from which *app_sieve* feature is computed, and a 15% improvement on recognition is gained by applying the normalisation. Similarly, for the pair of *app* feature and *eigen_lip* feature, there is a 10% improvement on performance due to the shape normalisation.

It is also desirable to determine if the error pattern is similar for the classifiers trained using different features. McNemar's [14] test is used to determine whether the difference in the accuracies

[4] Subsequently, in each iteration, any features extracted with application of a PCA need to be recomputed. In the example of AAM, a new AAM model is trained on only the the training speakers during each iteration.

of a pair of classifiers A and B is statistically significant. Firstly, we construct the joint performance of two classifiers A and B as shown in Table 1 The disagreement between two classifiers when parsing the same dataset are used in McNemar's test. N_{10} denotes the number of patterns that are identified correctly by A but incorrectly by B, and N_{01} denotes the number of patterns identified incorrectly by A but correctly by B. Assuming that A and B are not significantly different, if only one of them misclassifies on a pattern, it is equally likely to be A and B. Therefore, for the null hypothesis H_0 that A and B are not significantly different, N_{01} and N_{10} obey the binomial distribution $\mathcal{B}(k, q)$ in which $k = N_{01} + N_{10}$ and $q = 0.5$. H_0 will be rejected if p-value is smaller than a given significant level α.

		B	
		Correct	Incorrect
A	Correct	N_{00}	N_{01}
	incorrect	N_{10}	N_{11}

Table 1: Joint performance of classifier A and B on two-class problem

We compute the McNemar's test on all combinations of pairs of features. The only pair of features whose p-value is not zero is the *app* and *aam_cat* features, with a p-value of 0.141. This implies that classifiers using these two features behave similarly during recognition. Knowing that *aam_cat* is formed by concatenating *shape* and *app*, it is likely that, unlike a PCA, the concatenation fails to incorporate the two components effectively, therefore the classifier trained on *aam_cat* doesn't gain extra information from $shape$ feature.

5 Conclusions

In this paper, we compared various features for lipreading, including four types of AAM features, sieve features, 2D DCT features and eigen-lip features. A subset from GRID dataset, containing 15 speakers and and total of 14620 utterances was applied to measure the performance of features in terms of word accuracy rate. To obtain the AAM features, a novel LP tracker was utilised to track a set of target points on the lips. Each feature was 15 fold cross-validated on a speaker-independent manner, where each fold held out a different speaker's data for testing. It was observed that, in general, AAM features with appearance parameters outperform other types of feature, implying that the appearance is more informative than the shape. Results also showed that pixel based methods can benefit from an image normalisation that removes the shape and affine variation from the region of interest, and a significant improvement on recognition result was observed with the normalisation on sieve feature and eigen-lip feature.

There are some ways we can further advance the work in this paper. Instead of recognising words, classifiers can be trained to recognise visemes, which are the smallest visual units distinguishable in lipreading and is equivalent to phonemes in audio speech. On the other hand, since we know that the shape normalised appearance performs almost as well as a full AAM feature, it is very likely that it is the inner appearance of the mouth that contains most information, although further experiments are needed to examine this on features designed on only the inner appearance of mouth.

References

[1] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, "Recent advances in the automatic recognition of audio-visual speech," in *Proceedings of the IEEE*, vol. 91, no. 9, Sept 2003, pp. 1306–1326.

[2] I. Matthews, T. F. Cootes, J. Bangham, S. Cox, and R. Harvey, "Extraction of visual features for lipreading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 198–213, February 2002.

[3] J. Luettin, N. A. Thacker, and S. A. Beet, "Speechreading using shape and intensity information," in *Fourth International Conference on Spoken Language Processing (ICSLP)*, vol. 1. IEEE, 1996, pp. 58–61.

[4] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, November 2006.

[5] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, June 2001.

[6] T. Cootes and C. Taylor, "Statistical models of appearance for computer vision," Imaging Science and Biomedical Engineering, Univeristy of Manchester, Tech. Rep., October 2001. [Online]. Available: http://www.isbe.man.ac.uk/~bim/

[7] E.-J. Ong and R. Bowden, "Robust lip-tracking using rigid flocks of selected linear predictors," in *Proceedings of Eighth IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[8] J. A. Bangham, N. Bragg, and R. W. Young, "Data processing method and apparatus," GB Patent 9512459, June 1995.

[9] J. A. Bangham, S. Impey, and F. Woodhams, "A fast 1D sieve transform for multiscale signal decomposition," in *Signal Processing VII, 'Theories and applications*, G. Holt, Cowan and Sandham, Eds., vol. 7E.9, 1994, pp. 1621–1624.

[10] X. Shao and J. Barker, "Audio-visual speech recognition in the presence of a competing speaker," in *Proceedings of Interspeech*, 2006, pp. 1292–1295.

[11] C. Bregler and Y. Konig, "“eigenlips" for robust speech recognition," in *International Conference on Acoustics Speech and Signal Processing, ICASSP-94*, vol. ii. IEEE, 1994, pp. ii/669–ii/672.

[12] J. Luettin and N. Thacker, "Speechreading using probabilistic models," *Computer Vision and Image Understanding*, vol. 65, no. 2, pp. 163–178, 1997.

[13] S. Young, G. Evenmann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book (version 3.2.1)*, 2002.

[14] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 532–535.