Talking Head Detection by Likelihood-Ratio Test†

Carl Quillen, Kara Greenfield, and William Campbell

MIT Lincoln Laboratory,
Lexington MA 02420, USA
wcampbell@ll.mit.edu

Abstract

Detecting accurately when a person whose face is visible in an audio-visual medium is the audible speaker is an enabling technology with a number of useful applications. These include fused audio/visual speaker recognition, AV (audio/visual) segmentation and diarization as well as AV synchronization. The likelihood-ratio test formulation and feature signal processing employed here allow the use of high-dimensional feature sets in the audio and visual domain, and the approach appears to have good detection performance for AV segments as short as a few seconds. Computation costs for the resulting algorithm are modest, typically much less than the front-end face-detection system. While the resulting system requires model training, only true condition training (i.e. video where the talking speaker is audible) is required.

1. Introduction

The audio track of a video recording may or may not be related to a face visible on the screen. Determining when this is the case has a surprising number of interesting applications. For example automatic meeting transcription systems could use such a system in addition to audio localization to properly attribute transcripts to individual speakers. Visual verification systems can employ such an algorithm as part of a liveness test to prevent replay attacks by asking a subject to respond to a variable prompt. Systems that automatically diarize video and the related audio streams, segmenting them by speaker and face change can also make use of this information. Finally biometrics that are available in one domain, e.g. voice-prints, can be used to locate video samples of the same subject present in an AV stream, allowing multi-modal user searches.

A number of different approaches have been tried to this problem. A detailed review may be found in [1], and more recent work has extended and improved the approaches portrayed there, e.g. [2]. This work approaches the problem in a way related to one of the earlier ideas tried, that is, using mutual information as a decision statistic[3]. The results presented here are conceptually similar, but permit the use of much higher-dimensional feature sets.

Many previous approaches to this problem have relied on least-squared modeling and correlation analysis, e.g. the CANCOR approach of FaceSync [4]. [2] presents a method that combines using feature selection derived by CANCOR with joint least-squared prediction of the resulting low-dimensional features used to compute a decision statistic. Other approaches use direct parametric modeling of joint AV features, e.g. via Gaussian Mixture models [5] or HMM models [6]. The likelihood score from these models can then be used to detect synchronicity, however these methods are not typically founded on classical Bayesian decision theory.

2. Likelihood Ratio Correlation Detection

The approach used here is parametric, involves trained models, and is very similar conceptually to a mutual information statistic. Following [2], we can denote by A a multi-dimensional random variable derived from an audio information stream, and by V a similar variable putatively derived from a related video channel. The mutual information $I(A,V)$ of A and V may be defined as

$$I(A,V) = \int p(a,v) \log \frac{p(a,v)}{p(a)p(v)} \, da \, dv.$$

If we are given an audio-visual segment, we can assume single-Gaussian densities for $p(a,v)$, $p(a)$ and $p(v)$, estimate them from the same segment and then use this to estimate $I(A,V)$:

$$I(A,V) \approx \frac{1}{2} \log \frac{|\Sigma_A| \, |\Sigma_V|}{|\Sigma_{AV}|}, \tag{1}$$

where $\Sigma_A, \Sigma_V, \Sigma_{AV}$ are the sample covariances of the segment. This method requires no pre-trained models, but requires at a minimum enough data to reliably estimate the covariances, especially for high-dimensional A and V. Using trained parametric models we can avoid this difficulty. If we view the problem as performing a likelihood ratio test comparing two hypotheses, (A) that the audio-visual features are statistically dependent, or (B) that they are independent, and we have T samples of data $\{a_{t,}v_t | \, t \in \{1 \ldots T\}\}$, at different times t, then the log likelihood-ratio decision statistic becomes

$$\log \prod_{t=1}^{N} \frac{p(a_t, v_t)}{p(a_t)p(v_t)} = \sum_{t=1}^{N} \log \frac{p(a_t, v_t)}{p(a_t)p(v_t)}. \tag{2}$$

Here we are implicitly assuming the independence of data samples at different times. If single-Gaussian models are used for the densities, the result is conceptually very similar to the mutual information approach. However a single-Gaussian likelihood-ratio test uses statistics that may be estimated on large amounts of held-out training data, and so we would expect to be able to use much higher-dimensional feature vectors.

In practice the desired feature dimension for video features v_t is likely to be much higher than for the audio. This framework accommodates this situation. Writing the likelihood ratio in (2) as

$$\log \frac{p(a,v)}{p(a)p(v)} = \log \frac{p(a|v)}{p(a)}. \tag{3}$$

† This work was sponsored by the United States Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

we see that just estimates of $p(a|v)$ and $p(a)$ are required. The approach we take is equivalent to carrying out a maximum likelihood estimate of $p(a,v)$ and then computing the needed marginals. If maximum-likelihood estimation is used to estimate $p(a,v) = p(a,v|\lambda_1,\lambda_2)$, for appropriate model parameters λ_1, λ_2, then we can write w.l.o.g. $p(a,v|\lambda_1,\lambda_2) = p(a|v,\lambda_1)p(v|\lambda_2)$. The maximum log-likelihood then becomes

$$\max_{\lambda_1,\lambda_2}\sum_t \log p(a_t,v_t) = \max_{\lambda_1}\sum_t p(a_t|v_t) + \max_{\lambda_2}\sum_t p(v_t).$$

Thus finding maximum likelihood estimates of $p(a|v)$ and $p(v)$ is mathematically equivalent to a finding a maximum-likelihood estimate of $p(a,v)$ and then computing the needed marginals. A similar argument also implies that the maximum likelihood estimate of $p(a)$ is equivalent to the marginal computed from the estimate of $p(a,v)$. Therefore we may directly estimate maximum likelihood estimates of $p(a|v)$ and $p(a)$ use them in (3) with the added advantage that they may have non-singular covariance even when the estimated $p(a,v)$ and $p(v)$ would be problematic.

2.1. Single-Gaussian Parameter Estimation

In the single Gaussian case, formulas can be derived for estimation of the parameters of both $p(a|v)$ and $p(a)$ in Equation 3. We use the convention that the last element of the video feature vector v always has a constant value 1 and may be written $v = \binom{v'}{1}$. This allows us to write affine transformations of v' as a linear transformation of v, e.g.

$$Mv = \begin{pmatrix} M' \\ \mu \end{pmatrix} v = M'v' + \mu.$$

Without loss of generality, we may then write a single Gaussian model for $p(a|v)$ and $p(a)$ in the following form:

$$p(a|v) = N(a, Mv, \Sigma_{av}) = \frac{1}{\sqrt{|2\pi\Sigma_{av}|}} \exp\left(-\frac{1}{2}(a - Mv)^t \Sigma_{av}^{-1}(a - Mv)\right).$$

and

$$p(a) = N(a, \mu_a, \Sigma_a)$$

If we define sample covariances

$$\begin{aligned} \overline{vv^T} &= \frac{1}{N}\sum_{t=1}^{N} v_t v_t^T \\ \overline{av^T} &= \frac{1}{N}\sum_{t=1}^{N} a_t v_t^T \\ \overline{aa^T} &= \frac{1}{N}\sum_{t=1}^{N} a_t a_t^T \end{aligned}$$

where t ranges over all the frames in the training set, then a straight-forward calculation derives maximum likelihood estimates for M and Σ_{av} :

$$M = \overline{av^T}\left(\overline{vv^T}^{-1}\right) \qquad (4)$$

$$\Sigma_{av} = \overline{aa^T} - \overline{av^T}M^T. \qquad (5)$$

In practice, $\overline{vv^T}$ may well be singular and a pseudo-inverse is employed instead of $\overline{vv^T}^{-1}$.

3. Gaussian Mixture Modeling

Gaussian mixture modeling can be used to extend the modeling power of the single-Gaussian framework above. A straightforward approach would be to use maximum-likelihood estimation via the EM algorithm to estimate a mixture model for $p(a,v)$, and then derive marginals from this. The likelihood ratio scoring would then use equation (2), rather than in (3). This again presents the problem of dealing directly with very high combined AV feature dimensions. Here we tried two different approaches based on the single-Gaussian modeling used in section 2.1. The first approach is to use a mixture model for

$$p(a|v) = \sum_i w_i N(a, M_i v, \Sigma_{av}^i) \qquad (6)$$

and to directly estimate the parameters $w_{i,}M_i$ and Σ_{av}^i via the EM algorithm. Posterior probabilities from this model are then used to compute counts for a mixture model

$$p(a) = \sum_i w_i N(a, \mu_a^i, \Sigma_a^i) \qquad (7)$$

where w_i are perforce the same mixture weights as in equation (6). This approach appears to work better than one where $p(a)$ is estimated as an independent mixture model, and has independently estimated weights.

4. Experiments

Two corpora were used to evaluate the algorithms described here. The first was a 1000-video subset of the XM2VTS database [7] which is a high-quality database of isolated individuals. It was partitioned into a 640 video training set amounting to 182434 frames of video and the rest used for test. In order to construct false examples for testing, for each test video 10 false examples were created by randomly picking the audio from another element of the test set and creating a new video clip using the incorrect audio.

The second corpora was derived from the Youtube Faces database[8]. This is a database of found video in flash format that was extracted from Youtube. We worked with a small 304 video manually annotated subset, where we marked short segments where isolated individuals were visible speaking and used only the video data from these subsets for training and test. Total video data amounted to 303149 frames of video with detected faces. Audio was substituted in the same way as for XM2VTS to create false trials. Facial poses were variable as was the video quality, which generally was rather poor. We used 30-fold cross validation with randomly chosen 250-video subsets of this data used as training and the remainder as test.

We used the OpenCV[9] face detector to detect faces in both datasets. The OpenCV detector performed extremely well on XM2VTS. It was generally less successful on the Youtube data, nevertheless it permitted an initial evaluation of the algorithm.

The XM2VTS database consists of relatively short video segments with a median duration of 12 seconds, which provides only limited information for the the matching system. The median duration of annotated single-speaker data in the Youtube dataset was 27 seconds, and we only evaluated videos where a minimum of 100 frames containing faces were detected.

4.1. Features

Some experimentation with possible audio and video features were carried out. Audio features were composed overlapping

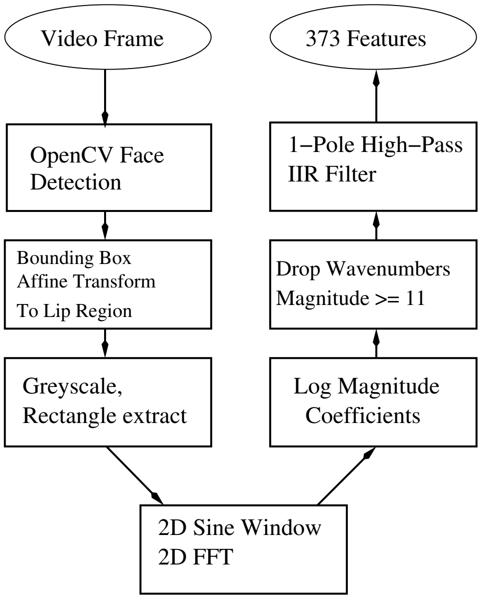
Video Frame
OpenCV Face Detection
Bounding Box Affine Transform To Lip Region
Greyscale, Rectangle extract
2D Sine Window 2D FFT
Log Magnitude Coefficients
Drop Wavenumbers Magnitude >= 11
1−Pole High−Pass IIR Filter
373 Features

Figure 1: The video feature processing pipeline

blocks of 20 audio frames, extracted 100 frames a second. Log energy, and three causal mean-removed mel-cepstral coefficients were extracted from each frame of audio data. The video feature set was extracted from a rather involved processing pipeline depicted in Figure 1. A number of alternatives were tried in terms of stacking windows of video features and using PCA derived features, but the best performing alternative to date on XM2VTS is a single vector of 373 log-magnitude spectral features derived from a single frame. For the much noiser Youtube dataset, the best performance was achieved with a smaller 145-dimensional feature vector. The processing pipeline in both cases can be described in detail below as follows:

1. The OpenCV face detector is used to locate the face within the video. In XM2VTS and the annotated segments of Youtube, only one face is visible at a time in the video, so in the case of multiple detections, picking the maximum size bounding box is an adequate strategy for selecting between them.

2. The OpenCV detection rectangle is a square that includes the eyes and mouth and includes regions with generally low correlation with the audio signal. Reducing the bounding box to an area more focused on the lips results in higher performance. Ideally a separate lip tracker would be used. On XM2VTS it seemed likely that adequate performance would be achieved by carrying out a deterministic affine transformation of the bounding box. We verified that this would likely be the case by running the OpenCV face detector on 2503 mostly vertically-posed faces from the FERET face database [10] which is annotated for lip position. We found that in coordinates normalized so that OpenCV bounding box is $[0, 1] \times [0, 1]$, the mouth position was at coordinates (0.5, 0.83), with standard deviations (0.025, 0.039). This suggested that a bounding box $[0.1, 0.8] \times [0.5, 1.1]$ would likely always contain the lip region.

3. Once a bounding box is found, a greyscale version of the contained image is resized to a fixed 42×32 size and multiplied by a 2D sine-window. A 2D-FFT is then calculated, and the log magnitude of the resulting coefficients computed. Finally all coefficients with a wavenumber magnitude greater than a fixed threshold were dropped. For XM2TVS, we dropped wavenumbers ≥ 11. For Youtube we dropped those ≥ 7, in both cases leaving just the spectral modes in the corners.

4. These features are also causal-mean removed, i.e. subject to a single-pole high-pass IR filter to approximately remove any mean offset. This mean removal operation performs a form of automatic gain control and was essential for good performance. Given inputs x_t, the outputs y_t of this filter can be computed by

$$y_t = (1 - \alpha)(y_{t-1} - x_{t-1} + x_t)$$

where $\alpha = 0.1$.

The details of this processing pipeline were the result of some optimization over the course of a number of experiments. It is likely that further refinement would provide better performance. Certainly much more variable face poses are problematic in Youtube and a new face-detector would be desirable. One aspect of the current face detection system that is worrisome is that there is a high-degree of interframe jitter in terms of bounding box size and position. Smoothing via a Kalman filter and 1-pole IIR filtering were tried, but they did not lead to improved performance. In any event, this inter-frame jitter may explain why the log-power spectrum features (which ignore phase information) were observed to perform better than PCA-based features.

A number of things were not tried and might improve performance further. For example, a 3D FFT approach across windows including the time axis would also be worth trying. Approaches to feature selection in the literature like CANCOR or latent semantic indexing might also prove useful.

4.2. Results

The likelihood ratio statistic in equation (2) assumes frame-wise independence of features. In practice one would expect considerable inter-frame and session dependence. In order to compensate for this effect, the actual statistic used in scoring was a frame-average likelihood ratio

$$\frac{1}{N}\sum_{t=1}^{N} \log \frac{p(a_t|v_t)}{p(a_t)},$$

which amounts to taking the N-th root of the statistic in (2). The detection performance for a single Gaussian system on XM2VTS may be seen in the DET curve [11] in Figure 2. The equal error rate (EER) is about 5%. The best results to date for a Gaussian mixture system do perform slightly better than this, yielding a 4.6% EER for a two Gaussian system, but this is unfortunately not a statistically significant improvement.

As might be expected, detection performance is quite sensitive to video duration. We were able to explore this effect on the XM2VTS test set used by limiting the duration of the video scored. In figure 3 the results can be seen. EER performance is already fairly good by 4 seconds.

The results of 30-fold cross validation on the Youtube database are presented below in Figure 4. The Youtube database is obviously much more challenging that XM2VTS. One particular area of concern was the level of audio-visual synchronization present. Most Youtube video has be re-encoded, and many

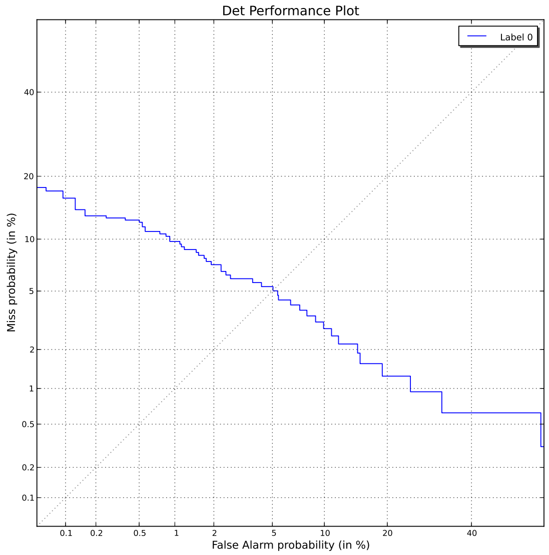

Figure 2: The XM2VTS single-Gaussian DET curve

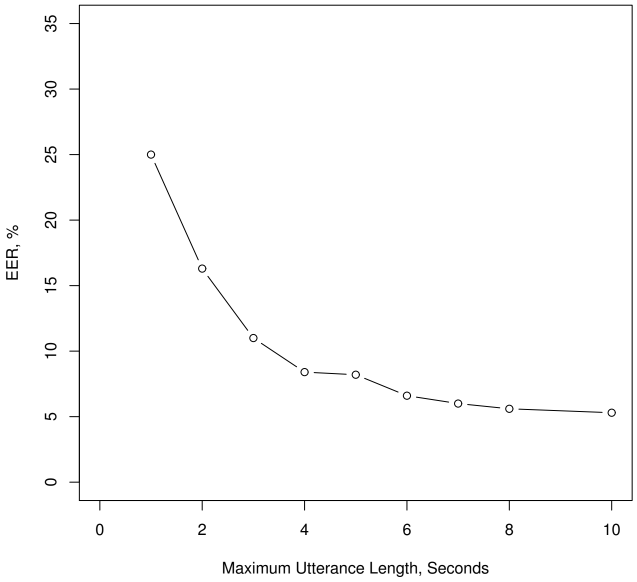

Figure 3: XM2VTS EER as maximum scored duration increases

of the toolsets used do not maintain accurate synchronization. For test data we computed likelihood of the match statistic at a number of different fixed temporal offsets ranging from ± 0.2 second and picked the maximum score. This reduced the test equal error rate from 28% to the 18% depicted here.

5. Conclusions

A single-Gaussian likelihood-ratio test appears to provide adequate performance for a talking face detection task, at least on clean controlled video databases like XM2VTS. Gaussian mixture extensions to this approach may provide a modest boost in performance.

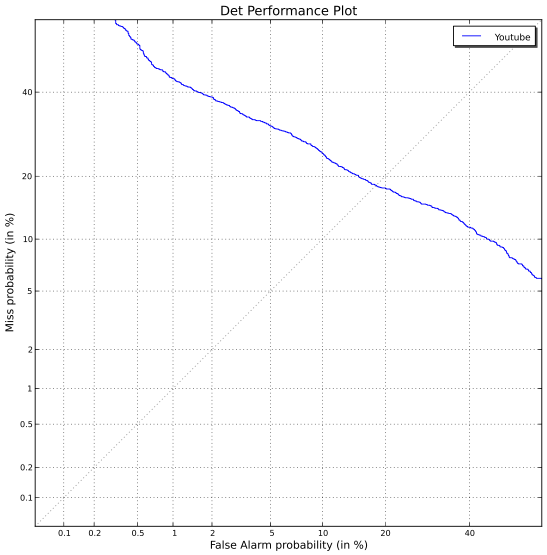

Figure 4: DET results for the proposed approach using Gaussian distributions on the Youtube corpus.

Results on more realistic databases derived from web video data are considerably worse, but are still promising. The existing OpenCV face detector may need work for best performance on the task. Additionally the highly variable face poses that are present in realistic video may be providing challenges to the current modeling framework. Various approaches could be used to address this problem, including scoring alternate rotations of the video.

Finally the pre-trained model approach taken here is one end of a spectrum of possibilities. MAP adaptation could be used to create a compromise between it and the MLLR approach of Equation (1).

6. References

[1] H. Bredin and G. Chollet, “Audiovisual speech synchrony measure: application to biometrics,” *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 179–179, 2007.

[2] K. Kumar, J. Navratil, E. Marcheret, V. Libal, G. Ramaswamy, and G. Potamianos, “Audio-visual speech synchronization detection using a bimodal linear prediction model,” in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 53–59, IEEE, 2009.

[3] J. Hershey and J. Movellan, “Audio-vision: Using audio-visual synchrony to locate sounds,” in *Advances in Neural Information Processing Systems 12*, 2000.

[4] M. Slaney and M. Covell, “Facesync: A linear operator for measuring synchronization of video facial images and audio tracks,” in *NIPS*, pp. 814–820, 2000.

[5] D. Sodoyer, J.-L. Schwartz, L. Girin, J. Klinkisch, and C. Jutten, “Separation of audio-visual speech sources: a new approach exploiting the audio-visual coherence of speech stimuli,” *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 1, pp. 1165–1173, 2002.

[6] S. Bengio, “An asynchronous hidden markov model for audio-visual speech recognition,” in *Advances in Neural Information Processing Systems*, pp. 1213–1220, 2002.

[7] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, “XM2VTSDB: The extended M2VTS database,” in *Second international conference on audio and video-based biometric person authentication*, vol. 964, pp. 965–966, 1999.

[8] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 529–534, IEEE, 2011.

[9] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O’reilly, 2008.

[10] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, “The FERET database and evaluation procedure for face-recognition algorithms,” *Image and vision computing*, vol. 16, no. 5, pp. 295–306, 1998.

[11] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, “The DET curve in assessment of detection task performance,” tech. rep., DTIC Document, 1997.