Comparison of Localised and Feature-Based Variants of the Mixelgram Algorithm to Perform Audio-Visual Speaker Association

Trent W. Lewis and Patrick Klaosen

The Centre for Knowledge and Interaction Technologies
School of Computer Science, Engineering, and Mathematics, Flinders University, Adelaide, Australia
trent.lewis@flinders.edu.au, patrick.klaosen@gmail.com

Abstract

Audio-Visual Speaker Association (AVSA) is the task of determining who is speaking at each point in time from a given video and audio signal. The Mixelgram algorithm calculates mutual information between audio and video features. This work explores AVSA with a Localised Mixelgram using face-detection and a Feature Based Mixelgram using an Active Appearance Model. The Localised (mouth) Mixelgram achieved 80% accuracy and scaling the image to 0.1x increased speed by 170% without loss of accuracy. The feature based Mixelgram estimated the speaker 85% of the time but only achieved a speed of 15fps.

Index Terms: audio-visual speaker association, mixelgram, face tracking, active appearance model

1. Introduction

Audio-Visual Speaker Association (AVSA) is the task of determining who, if anyone, is speaking at each point in time. Specifically, it is matching a given audio stream to one of multiple video streams [1]. The task of correctly matching the audio and visual signals is becoming increasingly important as speech, speaker and emotion recognition are becoming utilising multiple modalities for robustness [2].

The research in this paper furthers work which has shown that the Mixelgram (described below) is useful for AVSA in realistic settings [3]. The main aim of this work is to implement a functional system that can perform AVSA in real time using two variations of the Mixelgram algorithm. The first is a localised Mixelgram using face-detection to locate the faces and then estimating the speaker using the Mixelgram on these regions. The second is a feature based Mixelgram using an active appearance model to track feature points on the face, using the distance between the upper and lower lips as the visual source to the Mixelgram. The experiments are performed on the the Clemson University Audio-Visual Experiments (CUAVE) database.

2. Audio-Visual Corpus

The CUAVE database was utilised [4] to assess the extension of the Mixelgram algorithm. It is a speaker-independent corpus of both connected and continuous digit strings totalling over 7000 utterances; however, for this work only the multi-speaker partition of the database was used. Each of the 22 multi-speaker clips involves two speakers arranged as in Figure 1, taking turns in reading a series of digits. At the end of the clips, both subjects speak simultaneously reading different sequences of digits. The ground truth for when a speaker is talking was taken from the work of [5].

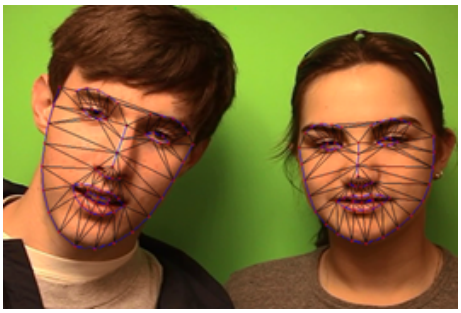

Figure 1: *Example of the multi-speaker partition of the CUAVE corpus. In the example, facial features are tracked using an Active Appearance Model[6].*

3. The Mixelgram

As mentioned, the Mixelgram represents the synchrony between each pixel in an image and the audio signal [7]. Specifically, the Mixelgram between the audio features A and video feature V at time t_k is given by

$$MI(x,y,t_k) = \frac{1}{2} log_2 \frac{|\Sigma_A(t_k)||\Sigma_V(x,y,t_k)|}{|\Sigma_{A,V}(x,y,t_k)|} \quad (1)$$

where $MI(x,y,t_k)$ is the mixel computed from a "sequence" of pixels located at position (x,y) over the window of $[t_{k-S},t_k]$ frames and the corresponding audio data. x ranges over the height of the image, y ranges over the width, and $|a|$ is the matrix determinant of a, and Σ_a is the covariance of a over the period S frames. In the experiments described here, S was set to 15 frames and equates to a window of half a second of data for each Mixelgram calculation.

The audio feature used in this series of experiments is the root mean square (RMS) of the amplitude over a period of audio samples centred around one video frame (captured at 30 frames per second) such that it puts the different modalities at the same sampling rate. The difference in edge information between sequential video frames is used as the video feature, as it has been shown by previous research (e.g. [3]) using the CUAVE corpus and the Mixelgram algorithm to achieve higher accuracies than only the pixel intensity level.

The final result of a Mixelgram calculation is an x by y matrix that covers the same time period as the original video sequence. An individual mixel value (the MI calculation at a particular (x,y) location for a particular frame) alone is relatively uninformative as to which speaker is speaking. One must calculate both the main mass of mixel activity spatially and also track

this temporally to compensate for noise in the data and create a more stable estimate of the association between the audio and video data. For this set of experiments the main mass was calculated by first applying a threshold of 0.1 to the raw mixel values and then calculating weighted centroid, (x_c, y_c), over all (x_p, y_p) pairs weighted by the corresponding mixel value $MI(x_p, y_p$ at particular $t_k)$, that is,

$$x_c = \frac{\sum_{p=0}^{P}(MI(x_p, y_p)x_p)}{\sum_{p=0}^{P} MI(x_p, y_p)}, y_c = \frac{\sum_{p=0}^{P}(MI(x_p, y_p)y_i)}{\sum_{p=0}^{P} MI(x_p, y_p)} \quad (2)$$

This center of mass is then smoothed over time using an exponential moving average (EMA) of the last ten frames ($n = 10$), that is

$$x_{ct_k} = \frac{(n-1)x_{ct_k-1} + x_{ct_k}}{n}, y_{ct_k} = \frac{(n-1)y_{ct_k-1} + y_{ct_k}}{n} \quad (3)$$

The threshold of 0.1 was derived empirically to give a balance between removing spurious, low mixel values and maintaining the actual areas of interest.

3.1. Localised Mixelgram

Traditionally, the Mixelgram is calculated over the entire input image. As the intended use of the output of this work is "speaker" association, a logical step is to restrict the analysis to areas of the image that relate to speaking, that is the face and more focally the mouth region.

To locate the face regions in the image sequence the Viola-Jones face detection algorithm from the OpenCV toolbox was used [8, 9]. The mouth region was extracted as the lower third of the face region. The Mixelgram was calculated on the entire image and then the mutual information in the face/mouth regions was summed and divided by their respective areas to get an average mutual information per pixel. The current speaker was estimated by comparing the average mutual information of the two regions and selecting the speaker in the region with the higher value.

An issue that arose was that the identified regions did not entirely encapsulate the faces, but missed parts of the upper and lower face. This was solved by increasing the height of the identified rectangles by 50% and repositioning as necessary. Face detection also occasionally suffered from not being able to detect both face. This was due to faces being tilted which is not handled well by the detection algorithm, but mostly due to some videos containing portions of the faces being absent from the screen. If at any time a face could no longer be detected, it was assumed that it had not moved and the previous location and position was used until the face could be detected again. This assumption is reasonable for the CUAVE dataset where the speakers' movement was restricted; for real-life problems, a certain time period would need to be set such that once the period of time has elapsed the assumption can no longer be assumed to be true.

3.1.1. Optimised Mixelgram

One weakness of the current implementation was that the mutual information was being calculated for every pixel in the image and only those pixels contained in the face/mouth regions were being considered. This meant that processing time was spent calculating the mutual information for pixels that were not being considered in the speaker estimate. A better solution was to first run the face-tracking and then run the Mixelgram only on these regions.

To implement this, separate Mixelgram objects were created for each face and the calculations were performed on each object. One difficulty implementing this was that the Mixelgram requires the size of the vectors containing the audio and visual data to be constant over the entire length of the video, that is the dimensions of the rectangular window needs to remain fixed. However, the window returned by face-tracking varies in size throughout the video, particularly noticeable when speakers move towards and away from the camera. To solve this problem, the face dimensions are stored in the first detection and then subsequent faces are resized to equal these dimensions, ensuring that the Mixelgram is always working on a fixed number of pixels.

3.1.2. Image Scaling

Further efforts were made to improve the speed by reducing the number of pixels used in the Mixelgram calculation by reducing the size of the input images in code. Scales of 1, 0.5, 0.25, 0.1, 0.05 and 0.025 were used and the effects on the accuracy and speed were analysed.

3.2. Feature Based Mixelgram

As a further refinement to the localised Mixelgram, specific visual features were extracted rather than using the edge information. Face-based Active Appearance Modelling (AAM) was applied to the video sequence [6]. The model (as shown in Fig. 1) was used to extract the height of the mouth opening which was then used as the visual input into the Mixelgram. The best results were achieved when the visual feature is taken to be the difference between the upper and lower lips from the current frame minus the distance from four frames prior.

4. Results

The accuracy of the algorithms was evaluated based on the Rand Accuracy, that is (number of true positives + true negatives) / total number of instances). This indicates when a speaker correctly identified as the current speaker. All accuracies in this are presented as $\mu \pm \sigma/2$, representing the range which 68% of the data lies assuming Gaussian distribution. The speed of the algorithms was determined using OpenCV functions getTickCount and getTickFrequency functions. The speed was calculated twice to increase reliability of results. They were run on a standard consumer laptop. Analysis of the results was conducted using ANOVA with Bonferroni corrected post-hoc comparisons made as necessary.

4.1. Localised Mixelgram

The Localised Mixelgram experiments performed included face-detection, mouth-detection and different levels of scaling. Figure 2 shows the different comparisons where a split-screen (no face-detection) was also used as a comparison. Five video sequences (05, 06, 18, 19 and 20) were excluded from the analysis due to the unreliability of face tracking, resulting from poor recordings where faces were partially off the screen.

The mean accuracy for the baseline was 74.66% $\pm$ 6.79%, the face 74.93% $\pm$ 5.49% and the mouth 79.34% $\pm$ 5.48%. Although not statistically significant, the mouth region performed more accurately than the face ($p < 0.18$) throughout the CUAVE dataset indicating that the mouth region may provide a better es-

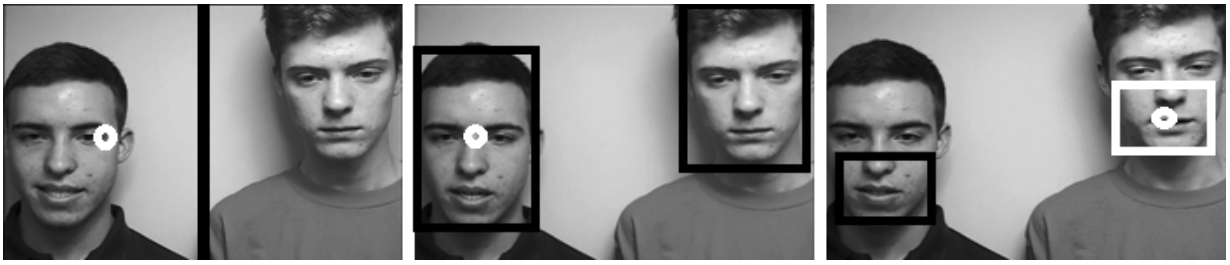

Figure 2: *Samples of identified speakers (white circle) using a split-screen (left), face-detection (middle) and mouth-detection (right).*

timate than the face. It is surprising to see virtually no change in accuracy between the split-screen and the face-detected region. This is likely due to the CUAVE dataset being recorded from a close distance where the faces occupy close to half the screen. If the CUAVE dataset was recorded from a further distance, an increase in the accuracy of the face region compared to the baseline may have been seen.

The Baseline achieved a speed of 11.94fps $\pm$ 0.34fps, the face achieved 10.52fps $\pm$ 0.37fps and the mouth achieved 10.68fps $\pm$ 0.31fps. The speed of the Mixelgram on the split-screen was significantly faster ($p < 0.05$) than the other two methods, this was because no face-detection was done when using the split-screen and the Mixelgram was calculated on the entire image for all three scenarios.

4.1.1. Optimised Mixelgram

There is very little change in accuracy when the Mixelgram is optimised to only run on the faces and mouth regions, the mouth performing marginally better than the face ($p < 0.175$) achieving an accuracy of 80.14% $\pm$ 5.75% compared to 74.27% $\pm$ 6.14%. The speed significantly improved when using the face region, achieving 17.08fps $\pm$ 0.96fps up from initially 10.52fps $\pm$ 0.37fps and the mouth achieving 23.54fps $\pm$ 1.01fps up from initially 10.68fps $\pm$ 0.31fps. This shows that speed of the algorithm has improved significantly ($p < 0.05$) for both cases, with the mouth achieving more than twice the speed. The larger improvement seen using the mouth-tracking is a result of the mouth occupying fewer pixels than the face and therefore when optimised to only run on these pixels a bigger improvement is seen compared to the face.

4.1.2. Image Scaling

The change in speed of the optimised version on the face and mouth regions were compared when the input image is reduced in scale from 1 down to 0.025. The face region showed a continual improvement in speed increasing from 17.08fps $\pm$ 0.96fps at a scale of 1 to 28.42fps $\pm$ 3.34fps at a scale of 0.025. The mouth region follows a similar trend achieving 28.57fps $\pm$ 2.23fps at a scale of 0.025 but less of an improvement is seen as the initial speed at a scale of 1 was 23.54fps $\pm$ 1.01fps. Both regions saw less of an improvement made as the scale continues to drop with both variations achieving over 27fps at a scale of 0.25. These speeds are more than adequate for real-time applications and show that the Mixelgram is fast enough to be performed on live data in real time on consumer grade hardware. Very little improvement is seen past 28fps, suggesting that the main speed bottleneck of this approach is the face-detection step.

Figure 3 shows the accuracy of the Mixelgram as the scale

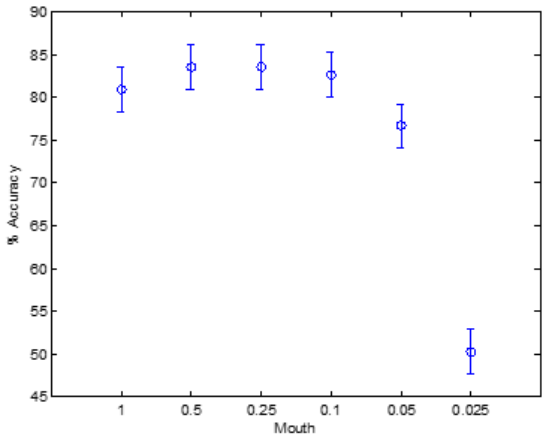

Figure 3: *Accuracy of using the Mouth Localised Mixelgram calculation using different image scales.*

is reduced for the mouth region. The graphs follow a similar trend with initially no decrease in accuracy, in fact a slight increase and then eventually rapidly decreasing to 50% accuracy, effectively a guess. The trend is similar when using the face region. The decline in accuracy for both regions begins at a scale of 0.1 with the face region dropping to an accuracy of 60.39% $\pm$ 5.08% at 0.025 and the mouth dropping to an accuracy of 50.18% $\pm$ 1.49% at the same scale.

These results show that the Mixelgram does not require high resolution images and can perform just as accurately on images at a scale of 0.1, that is 1/10th of the original size. From these figures, the optimal scale to be used to maximise speed without sacrificing accuracy on the CUAVE database is a scale of 0.1 which achieved an accuracy of 82.61% $\pm$ 5.81% and 27.65fps $\pm$ 0.93fps on the mouth region and 79.29% $\pm$ 6.59% and 27.67fps $\pm$ 0.65fps on the face region.

4.2. Feature Based Mixelgram

The feature based Mixelgram was only run on 7 out of the 22 videos, as both faces could not be reliably tracked on the other 15 videos. Although 26 out of 44 speakers (59%) could be accurately tracked, only 7 out of the 22 videos (32%) were included in the analysis as the Mixelgram requires both faces to be tracked. The mean accuracy of the feature based Mixelgram was 86.63%, the lowest accuracy was 79.25% and the highest accuracy was 90.02%. Figure 4 compares the accuracy of the feature based algorithm to the localised Mixelgram using face-

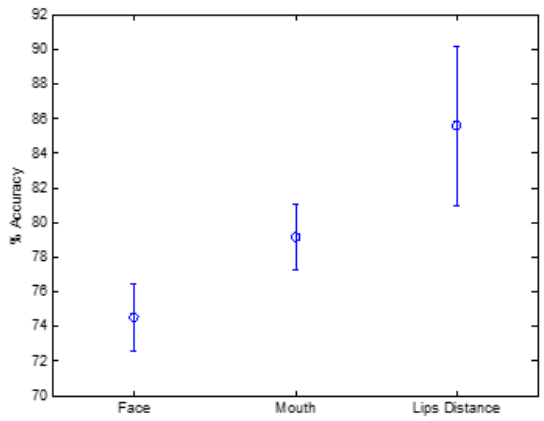

Figure 4: *The accuracy of speaker detection between using the face region, mouth region and the feature based lip distance.*

tracking and mouth-tracking. As can be seen, the feature based Mixelgram performs more accurately than the localised Mixelgram using the mouth-tracking (although not statistically significant, $p < 0.19$) achieving an accuracy of 85.61% 4.39% compared to 80.14% 5.75%. The mean speed of the feature based Mixelgram across the 7 videos was 15.68fps $\pm$ 0.80fps, much less than the localised Mixelgram which achieved an average 27fps using scaling.

5. Conclusions

In this study, the use of the Mixelgram to perform Audio-Visual Speaker Association was explored. Two variations of the algorithm have been implemented, a localised Mixelgram and a feature based Mixelgram. It was found that the localised Mixelgram performed better when the mouth region was considered compared to the face region achieving an accuracy of 80% compared to 74%. Speeds were increased from 10fps to 17fps using the face-region and to 23fps using the mouth-region by optimising the Mixelgram to only run on the regions of interest. This was further improved to 27fps (a 170% increase) for both face and mouth without loss of accuracy, by using a scale of 0.1x. This demonstrated that the system could be run in real-time.

These statistics are generated on the CUAVE dataset which are recorded from a close distance where the speakers occupy most of the screen. Had the video been recorded from a greater distance, the faces would occupy less pixels and the same level of scaling would result in loss of accuracy earlier. Therefore, for real-time applications, the scaling cannot be fixed at a value but needs to consider how far away users are from the camera. For the mouth-tracking, the CUAVE dataset showed no loss of accuracy at a scale of 0.1x which corresponded to dimensions of approximately 18x12 pixels. To allow for users at different distances, the Mixelgram could be modified to locate the faces, then dynamically scale each mouth to occupy no less than 18x12 pixels which should result in maximum speed with no loss of accuracy.

The feature based Mixelgram was implemented using the distance between the upper and lower lips as the visual source to the Mixelgram. This method was much less robust than the localised Mixelgram only tracking faces 59% of the time allowing analysis to be done on only seven video sequences. However, results showed that when the faces could be tracked, it performed more accurately than the localised Mixelgram using mouth tracking successfully estimating the speaker 85% of the time. The speed was not as quick as the localised Mixelgram only achieving around 15fps.

5.1. Future Work

The speeds achieved using this algorithm were very good and more than adequate for real-time execution, however there is still work that can be done to improve the accuracy and reliability of the system. The audio features used in the Mixelgram calculations was limited to the root mean square. There are many other, more useful features which still need to be explored such as Mel-frequency cepstral coefficients which indicates the perceived pitch of an audio signal. These which might be able to improve the accuracy.

The work has also limited the testing to the CUAVE database. This database is very good for initial testing purposes. However, the limited sample size resulted in differences in performance not reaching statistical significance and, more importantly does not resemble real life conditions. Further testing and future work needs be done on larger and more realistic datasets such as the AMI corpus which is a more real life dataset containing movement and background noise [10]. Almost all real world applications need to be able to handle harsh environments and therefore there is a need to use more challenging datasets.

6. References

[1] M. Siracusa and J. Fisher, "Dynamic dependency tests for audio-visual speaker association," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2, April 2007, pp. II–457–II–460.

[2] R. Goecke, "Current trends in joint audio-video signal processing: A review," in *Proceedings of the IEEE 8th International Symposium on Signal Processing and Its Applications ISSPA 2005*. Sydney, Australia: IEEE, Aug 2005, pp. 70–73.

[3] T. Lewis, M. Luerssen, S. Fitzgibbon, and D. Powers, "Audio visual speaker association for embodied conversational agents," in *Proceedings of the 14th Australasian International Conference on Speech Science and Technology*. ASSTA, Dec 2012, pp. 121–124.

[4] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. Gowdy, "Moving-talker speaker-independent feature study and baseline results using the cuave multimodal speech corpus," *EURASIP Journal on Applied Signal Processing,*, vol. 11, pp. 1189–1201, 2002.

[5] P. Besson, G. Monaci, P. Vandergheynst, and M. Kunt, "Experimental framework for speaker detection on the CUAVE database," EPFL, Lausanne, Switzerland, Tech. Rep. EPFL-REPORT-87331, 2006.

[6] J. Saragih and R. Goecke, "Learning aam fitting through simulation," *Pattern Recognition*, vol. 42, no. 11, pp. 2628–2636, Nov 2009.

[7] J. Hershey and J. Movellan, "Audio-vision: Using audio-visual synchrony to locate sounds," *Advances in Neural Information Processing Systems*, vol. 12, pp. 813–819, 2000.

[8] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O'Reilly, 2008.

[9] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[10] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005.