LOW-DIMENSIONAL MOTION FEATURES FOR AUDIO-VISUAL SPEECH RECOGNITION

Andrés Vallés Carboneras[*], *Mihai Gurban*[+], *and Jean-Philippe Thiran*[+]

[+]Signal Processing Institute,
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
{mihai.gurban, jp.thiran}@epfl.ch

[*]E.T.S.I. de Telecomunicación
Universidad Politécnica de Madrid
28040 Madrid, Spain
andreu.vc@gmail.com

ABSTRACT

Audio-visual speech recognition promises to improve the performance of speech recognizers, especially when the audio is corrupted, by adding information from the visual modality, more specifically, from the video of the speaker. However, the number of visual features that are added is typically bigger than the number of audio features, for a small gain in accuracy. We present a method that shows gains in performance comparable to the commonly-used DCT features, while employing a much smaller number of visual features based on the motion of the speaker's mouth. Motion vector differences are used to compensate for errors in the mouth tracking. This leads to a good performance even with as few as 3 features. The advantage of low-dimensional features is that a good accuracy can be obtained with relatively little training data, while also increasing the speed of both training and testing.

1. INTRODUCTION

Humans use visual information subconsciously to understand speech, especially in noisy conditions, but also when the audio is clean. The same integration can be performed by computers to improve the performance of speech recognition systems, when dealing with difficult audio conditions. Audio-visual speech recognition (AVSR) improves recognition rates beyond what is possible with only audio. An overview of AVSR can be found in [1].

While for audio speech recognition the types of features that are used are more or less established, with mel-frequency spectral coefficients being used in the majority of approaches, the same is not true for visual features.

We propose a method for extracting visual features based on the motion of the lips and show how these low-dimensional features lead to an audio-visual speech recognition accuracy comparable to that of discrete cosine transform (DCT) features, which are used in many AVSR systems. We then show how adding only a one-dimensional variable describing the average brightness from the image center further improves the performance of the system. For experiments, we use two different types of multimodal integration, feature fusion and decision fusion.

Our low-dimensional features are differences between the optical flow vectors computed on different regions of the speaker's mouth. Other approaches based on the optical flow have been proposed in the literature, but the dimensionality of their features is much higher. For example Gray et al. [2] use a 140 dimensional input vector as a visual feature. To that, they add a 150 dimensional vector consisting of the pixels of the downsampled images of mouths. Our feature vector has only three dimensions: the vertical and horizontal relative movement, and the average luminance of the center of the mouth.

The advantage of having a low-dimensional feature vector is that less data is necessary to train the audio-visual recognizer and the performance is improved when little data is available, since estimation errors caused by a too high dimensionality are avoided. Moreover, both training and testing are much faster when the dimensionality of the features is smaller. Our method also includes an estimation of the position of the mouth's middle line, which should improve the accuracy of the motion features. Actually, the fact that we use differences of motion vectors means that any movement of the head is canceled, since it will be present in both components that we differentiate.

Our paper is structured as follows. We introduce the background on audio-visual speech recognition in Section 2. In Section 3 we describe the details of our method and the database that we use. Section 4 illustrates our results and discuses them in comparison with results obtained on the same database with the widely used DCT visual features. Section 5 concludes our paper and presents directions for future work.

2. AUDIO-VISUAL SPEECH RECOGNITION

In this section we briefly present the structure of an audio-visual speech recognition system. While all such systems share common traits, they can differ in three major respects. The first one is the visual front-end; i.e., the part of the system that tracks the region of the mouth and extracts the visual features. The second one is the audio-visual integration strategy, that is, the way audio and visual information are put together in order to reach a decision about the recognized word. Finally, the type of speech recognition system can differ depending on the particular task (isolated-word recognition, continuous speech or large-vocabulary speech recognition). Our system recognizes sequences of words separated by silence, from a small-vocabulary database.

The majority of speech recognition systems use hidden Markov models [3] (HMMs) as the underlying classifiers used to represent and recognize the spoken words. Our audio-visual system also used HMMs, with two types of modality integration.

2.1 Visual front-end

All audio-visual speech recognition systems require the identification and tracking of the region of interest (ROI), which can be either only the mouth, or a larger region, like the entire face. This typically begins with locating the face of the speaker, using a face detection algorithm. The second step is

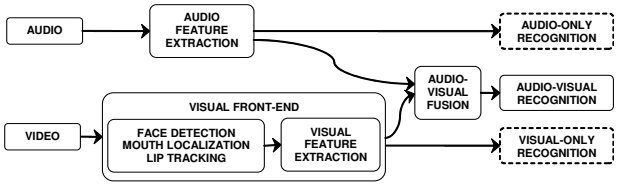

Figure 1: The general structure of an audio-visual speech recognition system.

locating the mouth of the speaker and extracting the region of interest. This region can be scaled and rotated such that the mouth is centered and aligned.

2.2 Visual feature types

Once the ROI has been extracted, the useful information that it contains needs to be expressed using as few features as possible. This is because the high dimensionality of the ROI impairs its accurate statistical modeling. Three main types of features are used for visual speech recognition [1]:

- Appearance based features, extracted directly from the pixels of the ROI.
- Shape based features, extracted from the contour of the speaker's lips.
- Joint appearance and shape features, the result of combining both previous types.

In general, the use of shape features requires a good lip tracking algorithm and makes the limiting assumption that speech information is concentrated in the contour of the lips alone. Several articles report that DCT features outperform shape based ones [4, 5]. Both DCT features and motion-based features like the ones we use fall into the first category, as no lip contour is extracted.

2.3 Audio-visual integration

The integration of audio and visual information [1] can be performed in several ways. The simplest one is feature concatenation [6], where the audio and video feature vectors are simply concatenated before being presented to the classifier. Here, a single classifier is trained with combined data from the two modalities.

Although the feature concatenation method of integration does lead to an improved performance, it is impossible to model the reliability of each modality, depending on the changing conditions in the audio-visual environment.

Using decision fusion, separate audio and video classifiers are trained, and their output log-likelihoods are linearly combined with appropriate weights. There are three possible levels for combining individual modality likelihoods [1]:

- Early integration, in the case when likelihoods are combined at the state level, forcing the synchrony of the two streams.
- Late integration, which requires two separate HMMs. The final recognized word is selected based on the n-best hypothesis of the audio and visual HMMs.
- Intermediate integration, which uses models that force synchrony at the phone or word boundaries.

Aside from feature fusion, we also tested our recognition system with early decision fusion, in this case using a multi-stream HMM classifier [7].

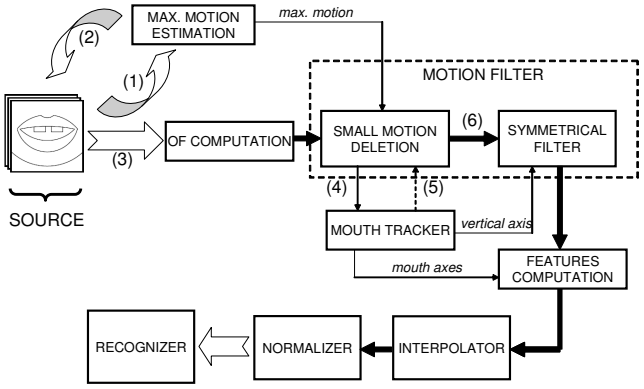

Figure 2: The feature extraction steps.

3. OUR PROPOSED METHOD

3.1 The data

For our experiments, we use sequences from the CUAVE audio-visual database [8]. They consist of 36 speakers repeating the 10 digits. We use only the static part of the database, that is, the first 5 repetitions.

The video sequences are filmed at 30 fps interlaced, so we can effectively double this framerate through deinterlacing. The average length of one video sequence is around 50 seconds (3000 deinterlaced frames).

Out of the 36 sequences, 30 are used for training, and 6 for testing. We use a six-fold crossvalidation procedure, that is, we repeat training and testing 6 times, each time changing the respective sets using a circular permutation. The performance reported is the average on the 6 runs.

To extract the ROI, the region of the mouth is located, scaled and rotated, so that all the mouths have more or less the same size and position. The mouth tracking procedure is semi-automatic, that is, correlation-based tracking is used until correlation falls under a threshold, at which point user input is required to refresh the search mask. However, since this method is not perfect, there is a need to estimate the position of the center line of the mouth while extracting the features.

On the audio side, different levels of white gaussian noise are added in order to show the gains obtained by combining our visual features with audio at different SNRs.

3.2 The low-dimensional visual features

When reducing the dimension of the visual features, the aim should be to represent the basic mouth motion, rather than to capture details of the articulation. The features built in such a way do not lead to a high performance in automatic lipreading (video-only), but they contain enough information to complement the audio vectors. Our goal here is to obtain visual features with a very low dimensionality.

Our proposed method uses two motion values and the brightness of the center of the mouth as visual features. The Lucas-Kanade optical flow [9] algorithm is used for motion analysis. Appending the brightness further improves system performance.

Our two motion features are extracted from the vertical and horizontal relative motion of the mouth and contain information about the opening and closing of it, leaving out

Figure 3: Movement filter processing.

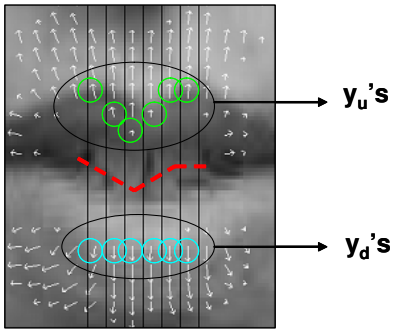

Figure 4: Details of mouth tracking

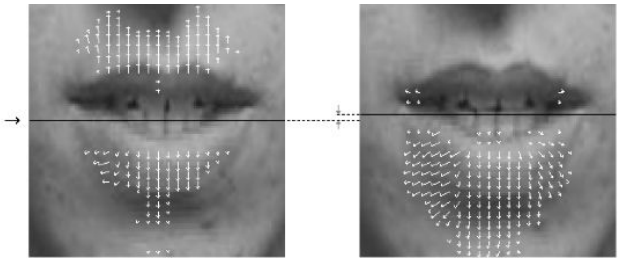

Figure 5: Mouth tracking example.

complex movements near the contour. On its side, the intensity of the center provides information about the use of tongue and teeth when articulating. These three features lead to a representation that is just as informative as the higher-dimensional DCT features.

An important issue for the visual feature processing is its dependence on the accuracy of the optical flow algorithm. In our case, the ROI has a very low contrast, leading to "false" responses in the optical flow, which we need to identify and eliminate.

The block diagram in Fig. 2 illustrates the main steps that we take to extract features from a video sequence. (notice that the numbers point out the sequence of actions).

The main processes are the following: at first, in the *Max. Motion Estimation* step, we estimate the maximum amplitude movement in the video sequence. This maximum will be used in the *Motion filter* block to remove the motion vectors whose amplitude is smaller than a certain percentage of the maximum one. Thus, we eliminate the smallest amplitude optical flow vectors, which are usually due to the aforementioned errors in the optical flow computation.

Once the maximum movement is estimated, the system is restarted: images are handled one by one and optical flow is computed using two consecutive frames. The flow resulted is filtered in order to deal with the "false" motion the optical flow algorithm output. Once motion is "clean", features are computed. Afterwards an *Interpolation* step is needed to increase the video rate to 100Hz, as audio-visual synchrony is required by our integration methods. All features are then normalized to mean zero and variance one in the *Normalization* step.

As it can be noticed in the figure, once we have a thresholded version (*Small Motion Deletion*) of the optical flow, the mouth is tracked. This is further useful for deleting "false" movements as well as for computing the features.

Specifically, the track lies in estimating the height of the mouth location in both vertical and horizontal components, namely, *horizontal* and *vertical* axes. The latter one is used in *Symmetrical Filter*, where we take advantage of the symmetry of mouth movements around a central vertical line. Thus, both the left and right sides are analyzed and non-symmetrical motion is eliminated. This filtering, in addition to *Small Motion Deletion*, allows only true motion to pass to the *Features Computation* step. However, some noisy movement still remains in practice. From left to right, fig. 3 shows the same image with the original optical flow, after erasing low amplitude motion, and finally after the whole *Motion Filter*.

Besides in filtering, the tracked axes are also useful for feature computation. For the case of the central brightness, we take advantage of that central point location is known. Thus, such visual component is computed as a 9-terms average using the intensity of the nearest pixels around the central, including this one.

On the other side, motion features are computed as follows. For the vertical case, the horizontal axis splits the image into two parts. Overall motions are computed for each of the two parts. Afterwards their difference is computed, resulting in a positive feature value when the upper and lower motion vectors point outwards (i.e. the mouth is opening), or a negative value when the upper and lower motion vectors point towards the center (i.e. the mouth is closing). Analogous manipulation is carried out for the horizontal component.

Here are the exact feature definitions:

$$F_v = \frac{1}{N} \cdot (OF_{v^+} \cdot N_{v^+} - OF_{v^-} \cdot N_{v^-}) \quad (1)$$

$$F_h = \frac{1}{N} \cdot (OF_{h^+} \cdot N_{h^+} - OF_{h^-} \cdot N_{h^-}) \quad (2)$$

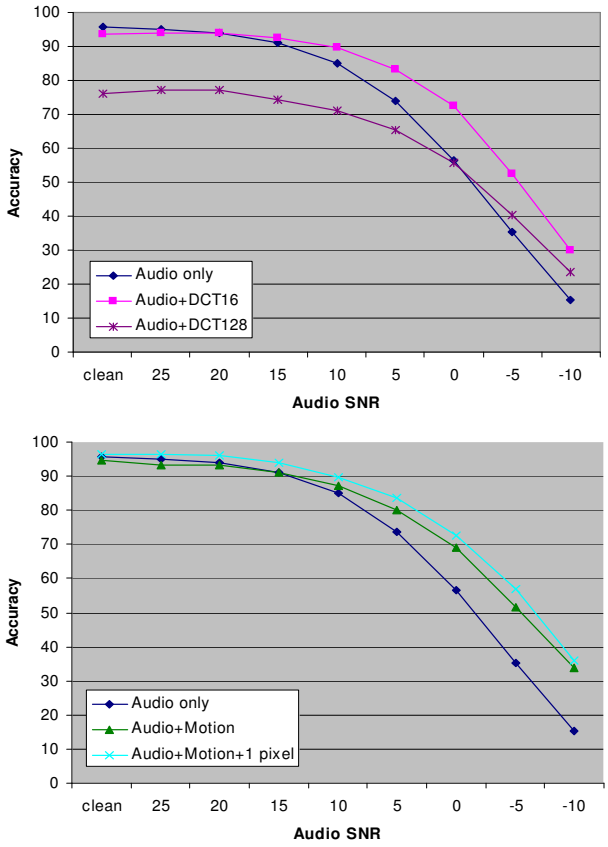

Figure 6: Results with feature fusion.

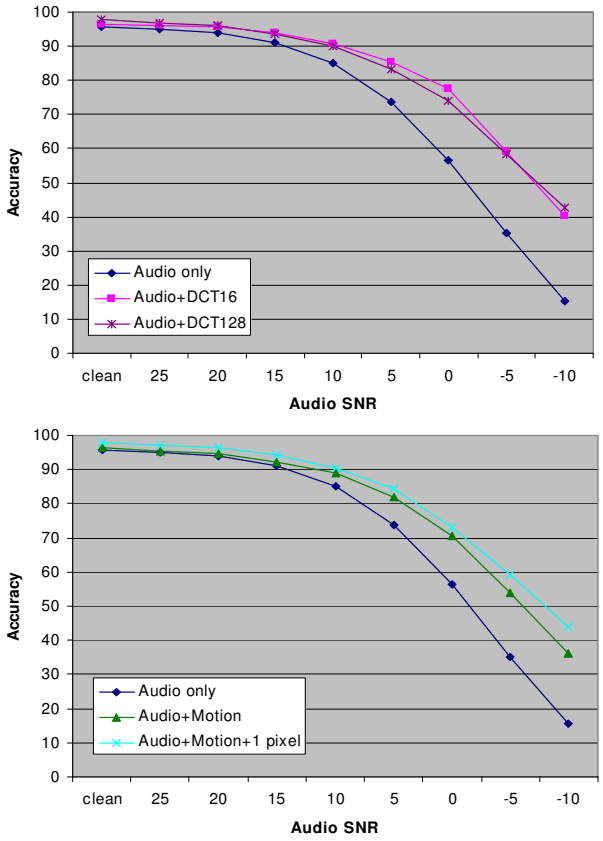

Figure 7: Results with decision fusion.

Here OF_{v+} and OF_{v-} represent the overall vertical motion of the mouth, computed as the maximum vertical motion components for the upper and respectively lower parts of the mouth. The horizontal components are obtained analogously. We expect these maxima to be more robust to optical flow errors than other ways of computing overall motion, such as taking the mean. Also, a weight is applied, allowing us to deal again with noisy movements due to optical flow computation. Specifically, the optical flow components OF_{v+} and OF_{v-} are weighted by $\frac{N_{v+}}{N}$ and $\frac{N_{v-}}{N}$, where N is the total number of pixels and N_{v+} and N_{v-} refer to the number of pixels of the upper and lower part respectively that contribute to the movement.

As seen, our feature extraction method is highly dependent on mouth location, so accurate mouth center tracking is needed. This is done in the *Mouth Tracker* step. Images in which the lips are "clearly" moving in opposite directions are *automatically selected* by analyzing the central columns in the case of the horizontal axis tracking and central rows in the estimation of the vertical axis. These selected frames provides information of the location of the mouth, as we will see.

Specifically, for the vertical case, first an overall motion estimation is done, deciding if mouth is either closing or opening. Then, the optical flow vectors distribution must be such that for the opening case, the downwards vector (y_d) located the highest must be under the upwards (y_u) vector located the lowest. For the closing case, the upwards vector located the highest must be under the downwards vector located the lowest must be complied. Analogous analysis is done for the horizontal component

Fig. 4 shows an example of an "opening vertical case" where optical flow distribution is appropriate to perform the tracking, i.e., where the frame is *selected*. The dashed line of the figure passes through every "column estimation", computed as the middle point between y_u and y_d. The average of such points is assumed to be the estimation of the track for frame. That is way these *selected* frames provide information of mouth location.

The arithmetic means of the last 15 *selected* images are computed by mobile average, resulting the final track. Figure 5 shows the results of the *Mouth Tracker*.

3.3 Our speech recognition system

We used the HTK library [10] to build the HMMs for speech recognition. The word models have 8 states per word, and one gaussian for each state. Our audio features are the mel-frequency cepstral coefficients (MFCCs), with delta and delta-delta values, adding up to a total of 39 coefficients.

We use either feature fusion, simply concatenating the audio and visual feature vectors, or early decision fusion to obtain our results. In the case of the latter, the audio-visual HMM's state-level emission probabilities are estimated separately for the audio and visual streams. The emission probability $b_j(o_t)$ for state j and observation o_t is computed as

follows [10]:

$$b_j(o_t) = \prod_{s=1}^{S} N(o_{st}; \mu_{js}, \Sigma_{js})^{\lambda_s} \quad (3)$$

where $N(o; \mu, \Sigma)$ is a multivariate gaussian with mean μ and covariance matrix Σ. The stream weight λ_s for stream s is chosen manually for the moment.

The product rule is one of the most widely used probability combination rules, along with the sum rule, the min rule or the max rule [11]. These rules are compared in [12], with the purpose of combining the outputs of classifiers trained on different types of audio-only features. The product rule was found to be the best performer. The same weighted product rule can be found in [6], integrating word-level probabilities.

4. RESULTS

We performed two types of experiments on the CUAVE database. First, we used the simpler type of audio-visual integration, feature fusion, to obtain the results presented in Fig. 6. Then we used decision fusion, leading to improved results, as can be seen from Fig. 7. In both cases, we compared our motion features with DCT features, either with 16 or with 128 coefficients. The accuracy presented is the word recognition accuracy. Since we are recognizing sequences of words, substitutions, deletions and insertions all count as errors.

The feature fusion experiments show that, for a high dimensionality, there is not enough data for proper training. This can be seen from the fact that all tests with 128 DCT coefficients have a very bad recognition rate, while the 16 DCT coefficients perform quite well. Our low-dimensional features have a good performance, outperforming the DCT features by 2-3%.

With decision fusion, both DCT feature types perform similarly well. Again, our low-dimensional features outperform the DCT by a few percents when the central pixel is added. Overall, decision fusion outperforms feature fusion in all cases. However, the combination weight for the two streams is chosen manually such that the best accuracy is obtained. As future work, we intend to implement a way to find this weight automatically, either by estimating the SNR of the audio, or by using reliability estimates based on the output probabilities for the two streams.

Finally, it can be seen that adding a single pixel to the two motion features always leads to an improved performance. These three-dimensional features perform just as well as the 128-dimensional DCT features.

5. CONCLUSION

We have presented a visual feature extraction method that creates very low-dimensional features derived from optical-flow vectors. The center line of the mouth is tracked so that the accuracy of the features is improved. Our low-dimensional visual features outperform the commonly-used DCT features, both with feature fusion and with decision fusion.

As future work, we would like to improve the quality of the estimation of the optical flow, and possibly replace it with a block matching motion estimation algorithm. We would also like to improve the decision fusion method, by finding a way to compute the stream weights dynamically.

Acknowledgement

This work is supported by the Swiss National Science Foundation through the IM2 NCCR.

REFERENCES

[1] G. Potamianos, C. Neti, J. Luettin, and I. Matthews, "Audio-visual automatic speech recognition: an overview," in *Issues in audio-visual speech processing* (G. Bailly, E. Vatikiotis-Bateson, and P. Perrier, eds.), MIT Press, 2004.

[2] M. S. Gray, J. R. Movellan, and T. J. Sejnowski, "Dynamic features for visual speech-reading: A systematic comparison," in *Advances in Neural Information Processing Systems* (M. C. Mozer, M. I. Jordan, and T. Petsche, eds.), vol. 9, MIT Press, 1997.

[3] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77(2), 1989.

[4] G. Potamianos, H. P. Graf, and E. Cosatto, "An image transform approach for HMM based automatic lipreading," in *Proceedings of the International Conference on Image Processing*, vol. 3, pp. 173–177, 1998.

[5] R. Reilly and P. Scanlon, "Feature analysis for automatic speechreading," *Proc. Workshop on Multimedia Signal Processing*, pp. 625–630, 2001.

[6] A. Adjoudani and C. Benoît, "On the integration of auditory and visual parameters in an HMM-based ASR," in *Speechreading by humans and machines* (D. G. Stork and M. E. Hennecke, eds.), pp. 461–471, Springer, 1996.

[7] H. Bourlard and S. Dupont, "A new asr approach based on independent processing and recombination of partial frequency bands," *Proc. International Conference on Spoken Language Processing*, pp. 426–429, 1996.

[8] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy, "Moving-talker, speaker-independent feature study and baseline results using the CUAVE multimodal speech corpus," *EURASIP JASP*, vol. 2002(11), pp. 1189–1201, 2002.

[9] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

[10] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge, Entropic Ltd., 1999.

[11] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[12] K. Kirchhoff and J. Bilmes, "Dynamic classifier combination in hybrid speech recognition systems using utterance-level confidence values," *Proceedings ICASSP-99*, pp. 693–696, 1999.