

A REAL-TIME PROTOTYPE FOR SMALL-VOCABULARY AUDIO-VISUAL ASR

J.H. Connell, N. Haas, E. Marcheret, C. Netti, G. Potamianos, S. Velipasalar*

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

E-mails: {jconnell, nhaas, etiennem, cneti, gpotam, —*}@us.ibm.com

ABSTRACT

We present a prototype for the automatic recognition of audio-visual speech, developed to augment the IBM ViaVoice™ speech recognition system. Frontal face, full frame video is captured through a USB 2.0 interface by means of an inexpensive PC camera, and processed to obtain appearance-based visual features. Subsequently, these are combined with audio features, synchronously extracted from the acoustic signal, using a simple discriminant feature fusion technique. On the average, the required computations utilize approximately 67% of a Pentium™ 4, 1.8 GHz processor, leaving the remaining resources available to hidden Markov model based speech recognition. Real-time performance is therefore achieved for small-vocabulary tasks, such as connected-digit recognition. In the paper, we discuss the prototype architecture based on the ViaVoice™ engine, the basic algorithms employed, and their necessary modifications to ensure real-time performance and causality of the visual front end processing. We benchmark the resulting system performance on stored videos against prior research experiments, and we report a close match between the two.

1. INTRODUCTION

Visual speech information, extracted from the video of a speaker's face, has been demonstrated to be of benefit to *automatic speech recognition* (ASR) [1]. Indeed, over the past 25 years, research in *audio-visual ASR* (AV-ASR) has clearly shown that the visual modality can consistently enhance accuracy on both small- and large-vocabulary recognition tasks, representing a dramatic improvement in ASR robustness to noise [2–6]. However, vision has yet to become a component of main-stream ASR, due mostly to the data rate intensity and computational expense associated with the high quality capture and robust processing of video.

In this paper, we describe a real-time implementation of AV-ASR, thus demonstrating that the above issues, long viewed as hindrances to real-time AV-ASR, can be successfully remedied. The prototype is architected within the existing IBM ViaVoice™ ASR platform, it utilizes popular and relatively inexpensive hardware for video capture, and, on higher-end desktop or portable PCs, achieves faster than real-time visual feature extraction and audio-visual feature integration. Thus, the remaining resources are available to the ViaVoice™ engine for speech recognition on basis of the extracted features, resulting in real-time AV-ASR of low-complexity tasks, such as connected-digit recognition.

Clearly, a number of challenges are to be overcome, in order for the prototype to meet two basic requirements: Real-time performance and improved accuracy over audio-only ASR. The first

challenge is the design of appropriate algorithms for visual front end processing (i.e., extraction of visual features) and audio-visual fusion. The techniques adopted for use in this prototype have been developed in earlier work, where they were demonstrated to dramatically improve ASR in acoustically degraded environments [5]. In particular, the visual front end uses pixel-based statistical methods for detecting the face and its features, as well as to subsequently extract speech informative visual features [6]. It is therefore significantly more efficient than lip-contour based feature extraction [3, 4]. Section 2 is devoted to the description of these algorithms. The second group of challenges is related to their implementation in the prototype: Proper modifications are necessary to speed-up the visual front end, and to ensure causal processing on the input video sequences, for example. Integration of the algorithm into the ViaVoice™ engine is also required, as well as proper communication with the engine, which retains control of all processing. These issues are discussed in Section 3. A third challenge concerns the video data, input to the prototype. Data quality is poorer compared to the corpora used in our prior research work [6]. As a result, AV-ASR accuracy somewhat suffers. The issue is discussed in Section 4, where performance of the prototype is benchmarked against research results. It is important to note, that this work is our first take on the implementation of real-time AV-ASR. A number of possible improvements can be readily identified, and are discussed in Section 5.

2. THE CORE AV-ASR SYSTEM ALGORITHMS

There are three main areas that differentiate AV-ASR systems [3]: The visual front end design, the audio-visual integration strategy, and the speech recognition method used. With respect to the first area, given video data, there exist three possibilities for visual speech representation [3]: Appearance-based features that typically seek a suitable transform of the pixel values within a visual *region of interest* (ROI) [2, 6], shape-based features that consist of a geometric or statistical representation of the lip contours [4], and combination of the two strategies [4]. Concerning audio-visual integration, most methods fall within the feature or decision fusion framework. The former approach combines the speech information at the feature level and utilizes a single classifier for recognition [5], whereas the latter combines the two single-modality classification decisions, typically at the likelihood level [4]. Finally, *hidden Markov models* (HMM) [7] with Gaussian mixture emission probabilities [4, 6], or alternatively, artificial neural network classifiers [2] can be used for AV-ASR. Among these techniques, our prototype system employs appearance-based features based on the ROI *discrete cosine transform* (DCT), as in [2, 6], discriminant feature fusion as in [5], and HMMs for ASR [7] (see also Fig. 1).

In more detail, given the video of a spoken utterance, audio- and visual-only *static* features are first extracted. The former ones

* Currently at the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544; E-mail: svelipas@princeton.edu

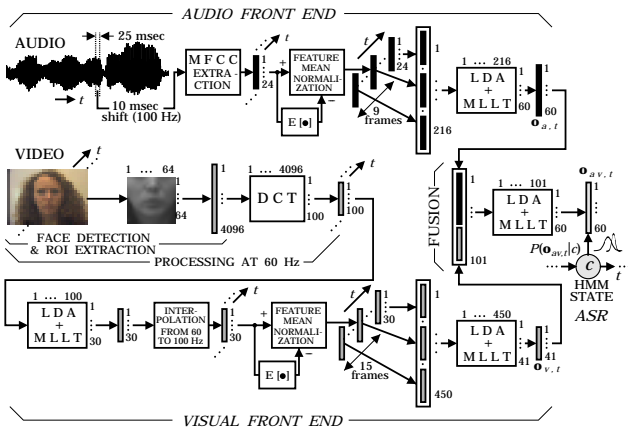


Fig. 1. Block diagram of the AV-ASR system, used as the basis of the real-time prototype. Time-synchronous, 60-dimensional audio feature vectors, $\mathbf{o}_{a,t}$, and 41-dimensional visual observations, $\mathbf{o}_{v,t}$, are extracted, both at a 100 Hz rate. Subsequently, these are combined using discriminant feature fusion, and classified by means of hidden Markov models.

consist of 24 mel-frequency cepstral coefficients, computed over a sliding window of 25 msec, at a rate of 100 Hz, followed by the application of *feature mean normalization* (FMN) [7].

To extract static visual features, a statistical face tracking algorithm is first used to detect the speaker's face and estimate the mouth location, size, and orientation [8]. To improve robustness, these quantities are smoothed over a temporal window. Based on the resulting estimates, a 64×64 pixel *region-of-interest* (ROI) is obtained for every video frame. The ROI contains the lower face around the speaker's mouth, including the jaw and cheeks, and is properly normalized to compensate for rotation, size, and lighting variations (see also Fig.2). Subsequently, a two-dimensional, separable DCT is applied to the ROI, and the 100 highest-energy DCT coefficients are retained. To reduce dimensionality and improve discrimination among the speech classes, an *intra-frame linear discriminant analysis* (LDA) projection is applied, resulting in a 30-dimensional feature vector. This is followed by a *maximum likelihood linear transformation* (MLLT) [6], that improves maximum likelihood based statistical data modeling. To facilitate audio-visual fusion, linear interpolation is employed that synchronizes the features to the audio rate of 100 Hz. FMN is used to further compensate for lighting variations, providing the final visual-only static features (see also Fig.1).

At each modality, *dynamic* features are obtained by forming a concatenation of a small number of consecutive static feature vectors (9 for audio, 15 for video), which is subsequently projected/rotated by means of an *inter-frame* LDA/MLLT combina-



Fig. 2. Region-of-interest extraction for an example video frame in the real-time prototype: (a) Face and facial part detection, with 11 estimated facial features depicted. (b) Face-region lighting compensation. (c) Final extracted ROI, appropriately normalized.

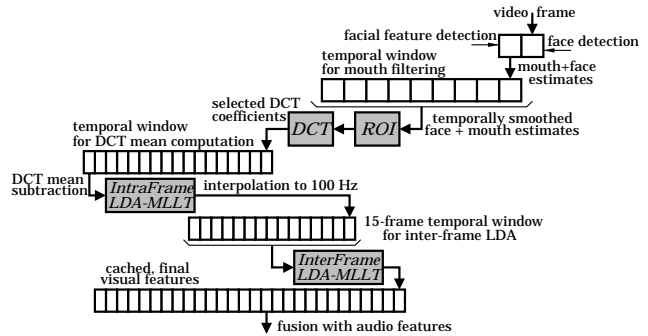


Fig. 3. Real-time processing employs several circular buffers for implementing the required visual front end steps of Fig.1.

tion. The final audio- and visual-only features are of dimensions 60 and 41, respectively [6]. A concatenated audio-visual feature vector can then easily be obtained. To reduce its high dimensionality (101), an additional LDA/MLLT projection is employed, giving rise to 60-dimensional audio-visual features. This is also known as discriminant feature fusion [5].

Following feature extraction, AV-ASR can be performed. For decoding, a two-stage stack decoding algorithm is employed, based on HMMs trained using the traditional maximum likelihood approach [7]. In particular, for connected digit recognition, unknown digit-string length is assumed, and HMMs with 159 context dependent states and approximately 3.2k Gaussian mixture components are used. To speed up decoding, these components are properly clustered and quantized.

3. THE REAL-TIME SYSTEM

Although the above algorithms for AV-ASR are well understood, additional challenges must be met to allow their integration into a real-time system within the ViaVoice™ platform. They are discussed in detail in this section.

3.1. The video capture hardware

To allow acceptable recognition of visual speech, full frame rate video of a sufficiently large frame size needs to be captured. In addition, in order to meet real-time processing requirements, such video should preferably be available in an uncompressed format. As a result, the prototype should be able to handle a high data rate. For example, for the 30 frame/sec rate, 320×240 pixel frame size, and uncompressed RGB color format of the video chosen to be used as input to our system, the resulting data are of a 55.3 Mb/sec rate. Two recently introduced standards/interfaces, namely the Firewire and USB 2.0, can handle these rates using relatively inexpensive hardware and software. A number of cameras can then be used to connect to the respective devices. In our system, an inexpensive iBOT™2 web-camera is employed to capture and transfer uncompressed color video to the PC via a USB 2.0 card.

3.2. The real-time visual front end

In prior work [6], a number of visual processing steps assumed the availability of the entire video sequence. For example, the computation of the mouth size and orientation necessary to ROI extraction, as well as of the DCT coefficient means needed in FMN, was performed over the entire utterance, thus introducing unacceptable

Computation type	Time (ms)	Fraction of total
Frame grab SIF	3.3	15 %
Face Finding*	10.5	24 %
Feature Localization*	20.9	48 %
Image Buffer and ROI	2.0	9 %
DCT and LDAs	0.9	4 %
Total	21.9	100 %

Table 1. Average processing load of the visual front end and audio-visual feature fusion for the AV-ASR prototype. In summary, 21.9 ms of processing are required for each incoming video frame, available at 33.3 ms. That corresponds to a 67% processor utilization, allowing the remaining to be used for acoustic signal processing and ASR decoding. All numbers are reported on a Pentium™ 4, 1.8 GHz desktop. * Face finding and feature localization are performed on alternate frames at a 15 Hz rate.

latency, equal to the utterance length. In addition, the computationally intensive steps of face and facial feature detection were performed at each frame, in order to improve ROI robustness to face tracker failures. Clearly, and in order to achieve real-time processing, certain modifications in the implementation of the visual front end are required, at the expense of ASR accuracy.

The first such modification is that both face and facial feature detection alternate at every second frame, hence introducing a 2-frame latency in processing. Subsequently, the computation of the smoothed mouth geometry estimates is altered: Size, rotation, and face boundaries, all used in ROI extraction (see Fig.2), are averaged over an appropriate temporal window, requiring a limited look-ahead of 10 frames. A further latency is introduced due to the DCT coefficient mean computation: In order to obtain reliable estimates of such means, some temporal look-ahead is required, especially at the beginning of the utterance, where few data are available. In our implementation, a 10-frame look-ahead is used. In addition to the above, the intra-frame LDA/MLLT requires the availability of 7 future frames at the 100 Hz rate (see Fig.1). All these steps add up to a latency of approximately 0.8 secs, for a 30 frame/sec input video rate. Implementation of the modified visual front end requires the use of circular buffers for holding sufficient number of video frames and visual features at various stages of processing, as shown in Fig.3. In addition, a buffer is required to hold the final visual features due to possible lag in the ViaVoice™ engine request for audio-visual fusion (see Section 3.3). Notice that, on the average, the entire visual front end, including fusion, utilizes 67% of the processor in a Pentium™ 4, 1.8 GHz desktop, thus achieving better than real-time performance. An exact break-down per visual front end stage is depicted in Table 1.

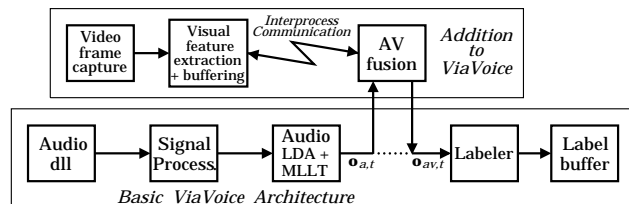


Fig. 4. Integration of the visual modality processing within the IBM ViaVoice™ architectural pipeline (see also Fig.1).



Fig. 5. The prototype graphical user interface. Two central windows depict the captured video and uttered vs. recognized text, while two meters show the resulting audio-only vs. audio-visual word error rates.

3.3. Integration into the ViaVoice™ engine

A main objective in designing the AV-ASR prototype is to provide back-compatibility to the audio-only IBM ViaVoice™ ASR system. Such an approach has multiple advantages, for example re-using all basic infrastructure for the acoustic front end processing and HMM based recognition, easier benchmarking of the benefits of the visual modality to ASR, as well as allowing (in the future) to switch between audio-only and audio-visual ASR.

Therefore, the whole prototype implementation is practically an addition to the basic ViaVoice™ architecture, as depicted in Fig.4. The basic ViaVoice™ pipeline is interrupted right after the extraction of dynamic audio features (i.e., after the inter-frame audio LDA/MLLT, following the audio waveform capture and the acoustic front end signal processing). At this stage, the ViaVoice™ engine sends a request for visual features to be used in conjunction with the available audio ones for audio-visual discriminant fusion. Such request activates the visual front end processing, implemented as discussed in the previous subsection. Upon successful completion of fusion, audio-visual ASR commences as shown in Fig.4 (labeling and label buffering). Interprocess communication between the audio pipeline and the visual front end is implemented using TCP/IP socket technology. Although this approach guarantees minimal architectural interruption to the ViaVoice™ platform, it has the disadvantage that all visual processing is driven by the engine, which can cause drops in the visual feature sequence, whenever the engine recognition load becomes high. In a sense, the lower data-rate sequence (audio) is driving the higher rate one (visual). For small-vocabulary tasks, however, our experience with the prototype shows that this is not a problem.

In order to demonstrate the benefit of the visual modality to recognition, audio-only ASR is also performed in the prototype. This is achieved by storing all audio features, and invoking the traditional ViaVoice™ engine to recognize the sequence of such features. Both resulting audio-only and AV-ASR performances are shown on the system GUI, as also depicted in Fig.5. A subset of this GUI is also employed for data collection (see Section 4.1), necessary to improve system performance by means of HMM adaptation [9].

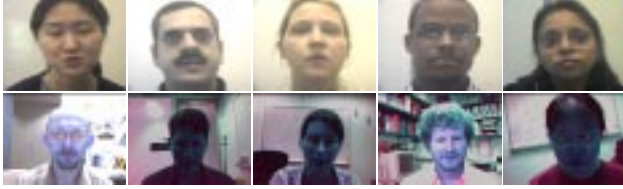


Fig. 6. Example frames of five subjects from the two audio-visual datasets, considered here for connected digit recognition (see Section 4.1).

4. PROTOTYPE PERFORMANCE

In this section, we describe the prototype ASR performance, relative to results obtained off-line using our research system [5, 6]. We first briefly describe the audio-visual datasets used for this purpose, followed by the recognition experiments.

4.1. Data

We use two databases, suitable for connected-digit recognition, for testing the performance of the AV-ASR prototype. The first corpus has been used in the development of our research system [6], and contains high-quality (“visually clean”) video data captured in a studio environment with rather uniform background, lighting, and subject head-pose, due to the use of a teleprompter. The video is in color, interlaced at 30 Hz, and of 704×480 pixel frame size (i.e., half frames are available at 60 Hz), whereas wideband audio is recorded using a good quality desktop microphone at a relatively clean office environment. The second dataset is captured using the real-time configuration, with the inexpensive video camera and a poor-quality microphone, built in a portable PC. The database subjects are recorded in their own offices without the use of a teleprompter, and thus, lighting, background, and head-pose vary significantly. The video format is also of lesser quality, as discussed in Section 3.1. The significantly more “visually challenging” nature of this second set becomes clear from Fig.6, where example frames from the two corpora are shown. Characteristics of the two databases are depicted in Table 2.

4.2. Recognition results

A summary of recognition results on the two datasets is depicted in Table 3. We first report off-line ASR using the modified visual front end of Section 3.2 and the audio front end of our research code. The audio channel is artificially corrupted by additive speech babble noise resulting in a 8 dB SNR. Noticeably, the performance in all modalities worsens for the “challenging” data domain, and the visual modality benefit to ASR decreases. Subsequently, we perform audio-only and AV-ASR using the prototype. Due to a discrepancy in the audio front end, the results degrade with respect to the off-line system, however the visual modality still significantly improves performance by 43% relative, compared to audio-only ASR. Notice, that the real-time visual front end implementation of Section 3.2 degrades visual recognition, compared to the 26.65% visual-only word error rate, achieved by our research system [6].

5. SUMMARY AND FUTURE WORK

We presented a prototype for real-time AV-ASR, built on top of the IBM ViaVoice™ speech recognizer. With proper modifications of

Visual Environ.	Set	Utter.	Dur.	Sub.
Clean	Train	5490	8:01	50
	Test	52	0:05	50
Challenging	Adapt	1007	1:15	10
	Test	200	0:15	10

Table 2. Two multi-speaker connected-digit databases used in benchmarking the prototype performance. Their partitioning into training (or, adaptation) and test sets is depicted (number of utterances, duration (in hours), and number of subjects are shown for each set).

Corpus	System	VI	AU	AV
Clean	Off-line	35.24	14.10	7.49
	Prototype	n/a	23.13	13.22
Challenging	Off-line	48.59	19.47	15.53

Table 3. Visual-, audio-only, and audio-visual word error rate, %, on both clean and challenging data, using off-line processing and, on the former, the prototype AV-ASR system. The audio is at 8 dB SNR.

the visual front end, we were able to achieve real-time performance on small-vocabulary tasks, such as connected-digit recognition.

A number of improvements to the prototype can be readily envisioned. On the visual front end side, a reduction to the overall latency can be achieved, since, for example, DCT mean estimation becomes stable after the first couple of seconds of video. On the audio-visual fusion side, implementation of a multi-stream HMM strategy for combining the likelihoods of the audio and visual modalities using stream reliability-dependent exponents will make the system more robust to local degradations in either modality. Such improvements will be incorporated into future versions of the prototype.

6. REFERENCES

- [1] T. Chen, “Audiovisual speech processing. Lip reading and lip synchronization,” *IEEE Signal Process. Mag.*, 10(1):9–21, 2001.
- [2] P. Duchnowski, U. Meier, and A. Waibel, “See me, hear me: Integrating automatic speech recognition and lip-reading,” *Proc. Int. Conf. Spoken Lang. Process.*, pp. 547–550, 1994.
- [3] M.E. Hennecke, D.G. Stork, and K.V. Prasad, “Visionary speech: Looking ahead to practical speechreading systems,” in *Speechreading by Humans and Machines*, D.G. Stork and M.E. Hennecke, Eds. Berlin: Springer, pp. 331–349, 1996.
- [4] S. Dupont and J. Luetttin, “Audio-visual speech modeling for continuous speech recognition,” *IEEE Trans. Multimedia*, 2:141–151, 2000.
- [5] G. Potamianos, J. Luetttin, and C. Neti, “Hierarchical discriminant features for audio-visual LVCSR,” *Proc. Int. Conf. Acoust., Speech, Signal Process.*, pp. 165–168, 2001.
- [6] G. Potamianos and C. Neti, “Improved ROI and within frame discriminant features for lipreading,” *Proc. Int. Conf. Image Processing*, vol. III, pp. 250–253, 2001.
- [7] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs: Prentice Hall, 1993.
- [8] A. W. Senior, “Face and feature finding for a face recognition system,” *Proc. Int. Conf. Audio Video-based Biometric Person Authent.*, pp. 154–159, 1999.
- [9] L. Neumeyer, A. Sankar, and V. Digalakis, “A comparative study of speaker adaptation techniques,” *Proc. Europ. Conf. Speech Commun. Technol.*, pp. 1127–1130, 1995.