

An Embedded System for In-Vehicle Visual Speech Activity Detection

Vit Libal, Jonathan Connell, Gerasimos Potamianos, Etienne Marcheret
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA
{libalvit, jconnell, gpotam, etiennem}@us.ibm.com

Abstract—We present a system for automatically detecting driver’s speech in the automobile domain using visual-only information extracted from the driver’s mouth region. The work is motivated by the desire to eliminate manual push-to-talk activation of the speech recognition engine in newly designed voice interfaces in the typically noisy car environment, aiming at reducing driver cognitive load and increasing naturalness of the interaction. The proposed system uses a camera mounted on the rearview mirror to monitor the driver, detect face boundaries and facial features, and finally employ lip motion clues to recognize visual speech activity. In particular, the designed algorithm has very low computational cost, which allows real-time implementation on currently available inexpensive embedded platforms, as described in the paper. Experiments are also reported on a small multi-speaker database collected in moving automobiles, that demonstrate promising accuracy.

I. INTRODUCTION

State-of-the-art automatic speech recognition (ASR), together with a suite of human language processing technologies, enables successful implementation of voice user interfaces (VUI) in various domains. One such domain of interest is the automotive environment, where VUI can allow hands-free/eyes-free control and access to information without compromising safety [1], for example for navigation, name dialing, and in-car device control [2].

Despite successful commercial deployment of in-car ASR based applications, challenging problems remain that limit utility and degrade user satisfaction. By far, the most important issue constitutes the lack of ASR robustness to the noisy automobile acoustic environment, for example due to road, wind, and engine noise, background conversations, radio music, and other audio sources. The problem is exacerbated by the fact that the overall car noise environment is non-stationary and difficult to model a-priori, which in turn reduces the effectiveness of traditional noise-robust ASR techniques, for example Wiener filtering [3], echo cancellation [4], beamforming [5], spectral subtraction [6], the Algonquin framework [7], or adaptation techniques [8], to name a few. As a result, in practice, ASR systems in automotive environments are typically used in the so-called “push-to-talk” or “push-to-activate” mode. In this case, the driver is required to push a button to indicate intent-to-speak. This is necessary to avoid the spurious recognition results of “always-listening” operation in noise, i.e. when the ASR system remains “on” continuously. This solution is of course not a panacea, since the driver may press the button too soon, too late, or not at all [9]. In addition, it constitutes a significant step backwards from the desirable natural, voice-only, hands-free, and eyes-free interface.

In this paper, we investigate an alternative approach to the problem of automatically detecting when the driver speaks. The proposed solution exploits visual information from the driver’s mouth region, which of course is not affected by the noisy acoustic car environment. The approach is motivated by ongoing work on audio-visual automatic speech recognition (AVASR), which has been repeatedly demonstrated to significantly improve ASR accuracy especially in noisy environments by automatic lipreading [10]. The system proposed here uses a similar principle. It captures visual information by an appropriately designed and placed camera and processes it by employing a sequence of simple algorithmic steps in order to drive voice activity detection (VAD). The particular algorithms are chosen to allow real-time implementation on limited-resource platforms, typically used in automobiles to allow low-cost solution integration. Details of the proposed system are presented in Section II, with embedded platform implementation described in Section IV. The algorithm is tested on a small multi-subject database recorded in moving automobiles, as reported in Section III. Finally, Section V closes the paper with a short summary.

II. APPROACH

A. Requirements

The driving force behind our VAD implementation were the restrictive requirements dictated by the business needs of the automotive industry: ruggedness and low cost. These requirements influenced the camera hardware selection and imposed complexity restrictions on the developed algorithms. In particular, the system had to be capable of real-time operation on an embedded platform with a 200MHz processor and only 32KB of cache memory.

B. Overall configuration

Our visual VAD method uses a single camera mounted inside the vehicle to view the driver’s face. Among all possible camera placements (side pillar, steering wheel, center console, and rearview mirror) a rearview mirror placement was selected. This provides the most stable and unoccluded image of the driver’s face. It also provides a reasonable field of view with standard lens and has the added advantage that it is somewhat self-adjusting (i.e., the driver aims the mirror at his face so he can look out the back window).

A low-cost monochrome CMOS camera sensor providing quarter VGA resolution (320x240 pixels) was used as an input device. The camera is equipped with near infra-red (IR) light emitting diodes (LED) and a visible-cut light filter. Since the



Fig. 1. Sample frame captured by the mirror-mounted camera. Notice the extreme side lighting and the presence of facial hair and glasses.



Fig. 2. An example “motion history” image used for delineating windows and finding the boundaries of the driver’s head.

filter blocks the visible light from being perceived by the camera, the IR LEDs are necessary to provide fill lighting (most standard monochrome cameras are naturally sensitive to near-IR). This arrangement is necessary for night driving situations, and the near-IR will not disturb the driver. It also provides some lighting stabilization during daytime, although sunlight also has a strong near-IR component. The camera’s field of view was chosen to be 70 degrees. This is wide enough to capture the driver’s head and shoulders in all driving positions while still providing enough resolution for finding the driver’s face and observing his lips (figure 1).

Once a suitable image has been acquired from this setup, a suite of algorithms determines whether the driver’s lips are moving. Note that moving lips do not always imply that the speaker is talking. He might, for instance, be smoking, yawning, or chewing gum. Yet, if we first detect every occurrence of lip motion, it may be possible to filter out the non-speech lip motions later. Then again, many of these other activities are essentially silent and would not lead to any ASR output. Thus, for now we have chosen to ignore this issue.

C. Head detection

The overall vision processing system works in a “telescoping” manner: first finding the head, then the eyes, then finally the mouth. Each of these steps is made significantly simpler by the constraints imposed by the previous step. The first step in this cascade, face detection, is based on motion analysis. It exploits the simple fact that the driver’s head is constantly in motion (mainly due to vehicle vibrations) around approximately the same position.

To do this, first an intensity difference image is calculated from every pair of consecutive frames of the input image sequence. Next a “motion history” image is calculated by accumulating intensity differences over a certain time constant. This image contains high values at locations where consistent motion is occurring, such as near the head region boundaries (figure 2). Yet the camera’s field of view includes the vehicle’s windows. This means the retrograde motion of the external objects (such as trees bordering the road) can also show up as high intensity regions. To keep this from confusing the head boundary detector we explicitly detect and exclude such regions when searching for the head. The actual method to find both windows and the head boundaries is the same: we

project the motion history pixels along the horizontal axis and look for peaks.

This head detection algorithm is very computationally undemanding compared to many other approaches available (e.g., stereo or template matching). It merely takes the absolute difference at each pixel and maintains a decaying sum. After performing a column-oriented projection, the rest of the processing is done in 1D instead of 2D.

D. Eyes and mouth detection

Once the driver’s head is located, the possible position of the eyes are highly constrained (although they do move around based on head orientation). We search for the eyes before looking for the mouth because the eyes are generally more distinctive. We model the driver’s eyes as two dark spots on a lighter background, and find candidates by taking the minimum of a narrow horizontal bar filter and a narrow vertical bar filter (figure 3). These two filters can be efficiently implemented as the difference of two rectangular regions.

We then search for the strongest response within the face region as one eye, and then find a similar response with an appropriate geometric displacement as the second eye. Additional checks are performed for each eye when there are several vertically displaced candidates as these can indicate eyebrows or glasses frames. However this whole selection and verification phase is very fast because it is constrained to only the face portion of the image and most of the computation consists of scalar coordinate comparisons.

Finally, the localization of the eyes gives us a good idea where the mouth should be. A prediction based on the current eye positions is combined with the previous tracked mouth position to give a new estimated region. The horizontal bar mask image (reused from eye finding) is then projected horizontally within this search box to find the sides of the mouth, and a similar vertical projection is used to find the top and bottom. The result is a axis-parallel rectangular box tightly bounding the presumed mouth (figure 4). Again, only a very small portion of the image is being examined which makes this processing step extremely fast.

E. Lip motion detection

Once the black bar of the mouth has been found, some means must be devised to determine whether it is opening or



Fig. 3. Output of the matched filter for eyes.

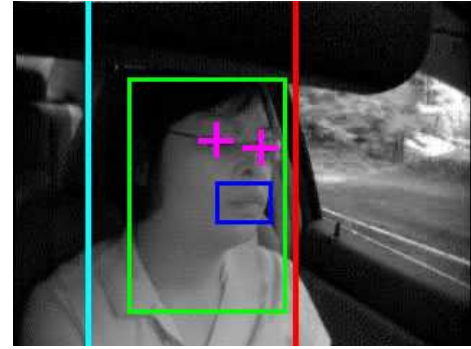


Fig. 4. Window limits, face region, eyes, and mouth are marked on the actual image.

closing. What we found to work best was simply examining the intensity of the mouth pixels in the bar mask (after suitable adjustments for ambient illumination and contrast). We histogram all the pixels and record the value for the 90th percentile and then compare this to a running average. If there has been a significant change we signal lip activity for the next few video frame times. Since the mouth opens and closes semi-regularly during speech, this indicator keeps getting renewed to generate a longer marked time interval of speech. It is this envelope which is used to enable the underlying ASR system.

III. EXPERIMENTS

A. Experimental setup

Our visual VAD system was aimed at actual in-car use so a strong emphasis was put on crafting a test setup that would reflect the real-world environment conditions. Our test set consists 4.8 hours of video recordings of 11 subjects made in a moving vehicle under daylight conditions (including cloudy and sunny days). The subject is always the driver, the windows were closed, and the maximum vehicle speed was about 30 mph (50 kph).

The sentences in the test set were a mix of command and control utterances, name dialer requests, and navigation phrases randomly generated by a realistic grammar. To prevent distractions to the driver each phrase was pre-recorded and played aloud to the driver. The driver then had to repeat the phrase when his workload allowed. A large number of non-speech intervals were included in the recordings – the non-speech vs. speech ratio is approx. 9:1 (i.e., only 10% speech). This was done to simulate a realistic usage model, where speech events are few and far between so most ASR errors are insertions due to noise being recognized as speech.

B. Head, Eyes and Mouth Detection Results

The detection error rate was measured by examining the program-determined bounding boxes for head and mouth and the point locations for the eyes. This was compared to the

ground truth locations of the head and facial features which were labeled manually. We did this for video frames at equally spaced intervals (every 10 seconds) providing a total of 1760 labeled frames.

The scoring algorithm marked a head detection error when the detected head box did not contain the true eyes or mouth. It also marked an error when the detected head box exceeded the extended head bounding box defined as true, minimum box that embraces the head with additional tolerance. An eye detection error was marked when either of the detected eyes were outside the bounding box formed by the ground truth positions of the hairline, nose, right ear/right cheek, and left ear/left cheek. A mouth detection error was marked when the detected mouth box did not contain the entire true mouth or when it exceeded an extended mouth bounding box.

The rationale behind this scoring scheme follows from the cascading detection algorithm: a feature searched in the next detection step must not be missed, in the same time, the search area for next detection step must be reasonably delimited. Note that if the face is missed it is very likely that the eyes and mouth will also be missed (as seen in table I). On the other hand, if eyes are missed, mouth may still be found with help of the tracking part of the mouth search algorithm. Finally, it should be noted that failures in the feature detectors do not necessarily imply failures in the overall speech detection system. Much of each video is composed of silent segments and the driver often looks around during these intervals. We have observed that this causes the face finder to “fall off” the face, or fail to detect it in profile (only one eye is visible). Yet when the driver is actually speaking his head is usually more-or-less stable and looking forward (perhaps a social convention?).

C. Speech/non-speech results

The lip motion error rate (see table II) was measured in terms of false negative rate (FNR), false positive rate (FPR),

TABLE I
FACE, EYES AND MOUTH DETECTION ERROR RATE

Face Detection	22.8%
Eyes Detection	28.3% (+5.5% wrt face)
Mouth Detection	27.4% (+4.6% wrt face)

TABLE II
VOICE ACTIVITY DETECTION ERROR RATE

False Negative Rate	17.1%
False Positive Rate	18.7%
Precision	31.8%
Recall	82.9%

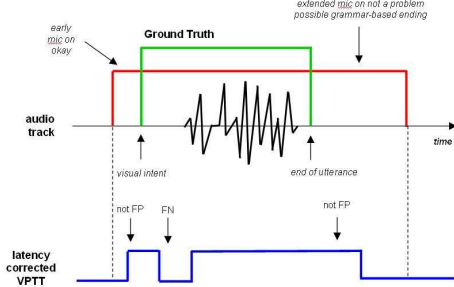


Fig. 5. For scoring the ground truth is extended slightly around the visible speech event.

precision (P) and recall (R), being defined as

$$FNR = \frac{N_{\hat{n}|s}}{N_s}, \quad FPR = \frac{N_{\hat{s}|n}}{N_n}, \quad P = \frac{N_{\hat{s}|s}}{N_{\hat{s}}}, \quad R = \frac{N_{\hat{s}|s}}{N_s}$$

where N_s is total number of true speech frames, $N_{\hat{n}|s}$ is number of true speech frames marked as silence, N_n is total number of true non-speech frames, $N_{\hat{s}|n}$ is number of true non-speech frames marked as speech, $N_{\hat{s}|s}$ is number of true speech frames marked as speech and $N_{\hat{s}}$ is total number of frames marked as speech.

The ground truth for speech/non-speech detection was labeled manually based on observed *visual* speech. That is, we marked those frames of the video recording in which a speech-producing lip motion occurred (as opposed to a yawn, etc.). For the scoring we allowed the system to mark a speech interval slightly early (1 sec) without penalty. We also allowed the system detection to be prolonged beyond the end of the end truth by a small amount (2.5 sec) without marking this as an error. The main intent was to key the ASR system at the start of a speech event in the same way that a driver would use a physical push-to-talk button. The extension of the scoring region is illustrated in figure 5.

IV. REAL-TIME IMPLEMENTATION ON EMBEDDED PLATFORMS

Typical image processing and computer vision algorithms are computation and memory intensive tasks and a powerful/expensive CPU is needed for their real-time implementation. The hardware costs, in fact, often impose a barrier that prohibits their widespread use in commercial applications. An important objective of the visual VAD algorithm design was to obtain a real-time implementation on an embedded platform that would serve as an example of a low-cost implementation of a complex vision task for commercial purposes.

While the original code development was done on a PC, the final visual VAD algorithm was also ported to two embedded CPUs: a SH7760 (200MHz, 32KB data cache, fast RAM) and a PXA250 (400MHz, 32kB data cache, slow RAM). Table III lists several steps of the visual VAD algorithm along with the processing times breakdown for each of the two CPUs. The total time per frame needed by our visual VAD algorithm was close to 20ms for a 160x120 @ 30 fps video stream on both CPUs. This shows that the algorithm can comfortably run in real-time on these embedded CPUs since only 10-15 fps is actually needed.

TABLE III

VISUAL VAD ALGORITHM COMPUTATION TIME ON EMBEDDED CPUs (MS PER FRAME AT 160x120 PIXEL VIDEO STREAM RESOLUTION)

	PXA250	SH7760250
face detection	5.5	5.2
eyes+mouth detection	9.6	8.0
lip motion detection	5.5	4.9
TOTAL	20.6	18.1

V. CONCLUSION

A visual-only speech activity detection system for in-car detection of driver's speech has been developed with the goal to eliminate the manual push-to-talk activation of ASR, and hence improve ASR robustness to noise. The proposed approach is simple and capable of running on embedded platforms using video from an inexpensive camera. The method was tested on a realistic multi-subject test set collected in moving automobiles. The method exhibits a 17% false negative rate while marking 19% of the frames as false positives. The errors can largely be ascribed to the extreme visual environment in the moving vehicle and to problems with face/eyes/mouth tracking when the driver moves his head fast around. These problems might be solved by adding features to allow the algorithms adapt to the particular driver, and by more aggressively using prior knowledge (the current algorithm is speaker independent and works with minimal prior knowledge). Despite a few problems, a matched detection error rate below 20% is a very promising result and indicates that visual voice activity detection does indeed hold potential to improve ASR performance in automotive environments, even with constrained hardware.

REFERENCES

- [1] *Usage of In-Vehicle Information and Communication Devices Survey*, <http://www.nuance.com/ads/automotive/survey>, April 2006.
- [2] IBM press release, *IBM and VoiceBox Technologies Extend Hands-Free Technology to Allow Drivers to Talk to Their XM Satellite Radios and Mobile Phones*, <http://www-03.ibm.com/press/us/en/pressrelease/19152.wss>, January 2006.
- [3] Agarwal A., Cheng Y.M., *Two-stage Mel-warped Wiener Filter for Robust Speech Recognition*. Proc. of Intl. Workshop on Automatic Speech Recognition and Understanding (ASRU'99), pp. 67-70, 1999.
- [4] Ichikawa O., Nishimura M., *Simultaneous Adaptation of Echo Cancellation and Spectral Subtraction for In-Car Speech Recognition*, IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E88-A, Issue 7 (July 2005), p.1732-1738, 2005.
- [5] Dahl M., Claesson I., Nordebo S., *Simultaneous echo cancellation and car noise suppression employing a microphone array*, Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 1, p.239-242, 1997.
- [6] Boll S.F., *Suppression of acoustic noise in speech using spectral subtraction*, IEEE Trans. Acoust., Speech, Signal Process., vol.27, pp. 113-120, Apr. 1979.
- [7] Frey J.F., Deng L., Acero A., Kristjansson T., *ALGONQUIN: Iterating Laplace Method to Remove Multiple Types of Acoustic Distortion for Robust Speech Recognition*, Eurospeech 2001.
- [8] Gales M.J.F., Young S.J., *Robust Continuous Speech Recognition Using Parallel Model Combination*, IEEE Trans. on Speech and Audio Processing, vol.5, no.5, pp.352-359, September 1996.
- [9] Kun A.L., Miller W.T. III, Pelhe A., Lynch R.L., *A Software Architecture Supporting In-Car Speech Interaction*, Proc of IEEE Intelligent Vehicles Symposium, pp. 471- 476, June 2004.
- [10] Potamianos G., Neti C., Luettin J., Matthews I., *Audio-Visual Automatic Speech Recognition: An Overview*, In: Audio-Visual Speech Processing, E. Vatikiotis-Bateson, G. Bailly, and P. Perrier (Eds.), MIT Press, ISBN: 0-26-222078-4, 2006.