WAPUSK20 - A database for robust audiovisual speech recognition

Alexander Vorwerk, Xiaohui Wang, Dorothea Kolossa, Steffen Zeiler, Reinhold Orglmeister

Chair of Electronics and Medical Signal Processing (EMSP)
Technische Universitaet Berlin, Einsteinufer 17, 10587 Berlin
alexander.vorwerk@tu-berlin.de, wangxh321@hotmail.com, d.kolossa@ee.tu-berlin.de,
s.zeiler@ee.tu-berlin.de, reinhold.orglmeister@tu-berlin.de

Abstract

Audiovisual speech recognition (AVSR) systems have been proven superior over audio-only speech recognizers in noisy environments by incorporating features of the visual modality. In order to develop reliable AVSR systems, appropriate simultaneously recorded speech and video data is needed. In this paper, we will introduce a corpus (WAPUSK20) that consists of audiovisual data of 20 speakers uttering 100 sentences each with four channels of audio and a stereoscopic video. The latter is intended to support more accurate lip tracking and the development of stereo data based normalization techniques for greater robustness of the recognition results. The sentence design has been adopted from the GRID corpus that has been widely used for AVSR experiments. Recordings have been made under acoustically realistic conditions in a usual office room. Affordable hardware equipment has been used, such as a pre-calibrated stereo camera and standard PC components. The software written to create this corpus was designed in MATLAB with help of hardware specific software provided by the hardware manufacturers and freely available open source software.

1. Introduction

Speech recognition is utilized in many human-machine communication systems. However, noise that often will apperar in real world application environments is still highly challenging for automatic speech recognition in terms of extracting reliable audio features. One possibility to overcome this problem is using the visual modality, i. e. significant features of the speaker's face or mouth, to improve the recognition results. Many efforts have been made during recent years to incorporate visual information into the process of recognizing speech (Potamianos et al., 2004). For that purpose, useful speech material, containing both audio and video data that have been recorded simultaneously, is needed to design, develop and test robust algorithms for AVSR. Some corpora that provide respective audiovisual data are described in (Goecke, 2005). As can be seen, those corpora often are designed for different tasks and recorded under non-standard conditions. Furthermore, only a few of them are applicable for investigating the usefulness of stereo vision for speech recognition, since commonly there is only a single video stream available.

This paper describes a database that includes stereoscopic recordings of the speakers' faces. Another major goal of this work is the utilization of affordable hardware and software that is freely available (if possible) and can be used in recognition systems under low cost aspects regarding both implementation and performance.

In the following sections, we will describe in detail the corpus structure of WAPUSK20, the group of the speakers and the setup as well as the equipment used for recording.

2. Corpus parameters

Extracting visual features of speakers under varying conditions is a highly complex task. Besides differing illumination, rotations or other movements of the region of interest (e.g. the mouth) may compromise the recognition results considerably. Stereo vision promises to be helpful to overcome those challenges by simply choosing the better of two existing views for optimal 2-D processing or by realizing techniques to model 3-D information of the speaker's face or lips. For that reason, a stereo camera was used to capture the speaker from different angles (see section 4.2. for details). Because usually it is possible to store two channels of audio in each video file, four channels of audio have been provided. This can also be valuable for multitalker experiments.

In order to assure comparability, the sentence design of the GRID corpus (Cooke et al., 2006), which has been widely used in the past, e.g. (Lan et al., 2009), (Almajai and Milner, 2008), (Kolossa et al., 2009), (Gan et al., 2007) has been adopted. The sentences consist of six english words that are combined using the following choices:

1. command: {'Bin' 'Lay' 'Place' 'Set'}
2. color: {'blue' 'green' 'red' 'white' }
3. preposition: {'at' 'by' 'in' 'with'}
4. letter: {'A' ... 'Z'}
5. digit: {'1' '2' '3' '4' '5' '6' '7' '8' '9' '0'}
6. adverb: {'again' 'now' 'please' soon'}

In contrast to the GRID corpus, the letter 'W' is also included. Each speaker has recorded 100 sentences. The recording time was limited to three seconds per utterance while no restriction was given for the reading speed. All but two speakers recorded all sentences within a single session of about one hour including introduction to the setup, repetitions etc. Every sentence is unique over the entire database.

3. Group of Speakers

The corpus contains spoken utterances of 20 speakers, 9 female and 11 male thereof. Two of them are native English speakers. One speaker grew up in Greece, one in Kazakhstan and one in Spain. All other participants are native German speakers. The mean age of all speakers is 29.

All but two speakers lived in Germany during the recording period. All speakers were told to read the sentences in their own style with no obligations regarding pronunciation. This led to the fact that the letter 'Z' is pronounced as *zed* by some speakers and as *zee* by others.

4. Recording hard- and software

A standard personal computer running Windows XP with a Core2Duo Pentium processor 6600 @ 2.40 GHz, 3 GB of memory and a standard S-ATA 250 GB hard disk served as the recording device.

4.1. Audio hardware

Audio data was gathered using two pairs of omnidirectional small diaphragm microphones, one matched pair of OKTAVA MK 012 and one pair of Behringer ECM8000. As the microphone preamplifier, a TASCAM MA-8 was used. For A/D conversion, a soundcard LynxTWO-C (LST, 2009) was employed. It consists of six input channels with 24-bit multi-level $\Sigma\Delta$-analog-to-digital converters. Furthermore, this soundcard provides the possibility of muting every sound channel separately via software, which is useful for the synchronization of the audio and the video stream.

4.2. Video hardware

For capturing the video data, a color stereo vision camera, Bumblebee2 (type BB2-03S2), from PointGrey Research (PTG, 2009) was utilized. This camera is composed of two paraxial SONY 1/3" CCD sensors with a resolution of 640x480 pixels each and a baseline of 12 cm. Precalibration data for rectification of the images (see section 6.2.) is already stored inside the camera. It is connected to the PC via a FireWire interface. Internally, a 12-bit analog-to-digital converter is used. Gain and shutter of the camera have been chosen manually to ensure well-centered distributions in each color channel using the provided lighting (see next section). A reference image of the speaker's head position was provided by a Logitech web camera (QuickCam Pro 9000), which is mounted on top of the stereo camera and coupled to the PC via USB.

4.3. Lighting

The scene light was provided by a mixture of different lamp types. Background light came from a set of ordinary bulbs in different positions, the direct light was delivered from three pairs of base 2G11 daylight fluorescent lamps of Type 954 (two pairs) and type 865 (one pair). The light coming from these lamps was filtered by multiple diffusion films in order to reach a nearly uniform light distribution over the entire scene. All fluorescent lamps were controlled by electronic ballasts to avoid undesired flickering in the video data. For variable positioning (distance to the speaker, distance in between the lamps and their height), every lamp was installed on a plate connected with a standard microphone stand.

4.4. Software

The manufacturer of both the audio and video hardware are offering a Software Development Kit (SDK) for their products. These SDKs have been used to control the function of the hardware as capturing and saving the raw video and audio data. Saving raw data has become necessary in order to reach video frame rates that allow for a sufficient resolution in time. A MATLAB program has been implemented that can manage the recording process, particularly the synchronization of audio and video data. It is achieved by triggering the audio storing via the Lynx SDK after the reception of a first video frame from the camera. This first frame will be discarded afterwards. A simple graphical user interface (GUI, see Figure 1) has been programmed for managing the records by the speaker itself. Furthermore, a function has been designed that reads the raw stream data from the hard disk to convert every sentence into two video files according to the left and right camera sensor, respectively.

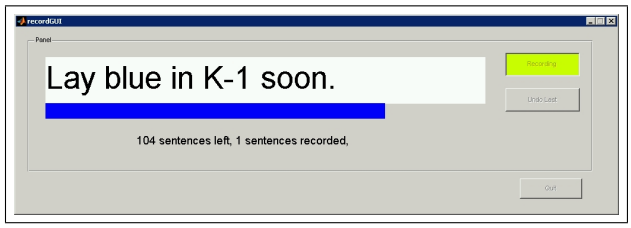

Figure 1: Graphical user interface for managing the records

5. Recording setup

5.1. Hardware setup

The recordings were conducted in a typical office room, which was not acoustically tuned, i.e. no measures for damping have been taken. In order to ensure comparable lighting conditions, the room was shaded and illuminated only by the artificial light described above. The fluorescent lamps were located symmetrically in a hemicycle in front of the speaker. The camera and both microphone pairs have been arranged on a table near the speaker, in order to achieve a close-up view of high resolution of the speaker's face and a sufficient signal-to-noise ratio of the audio signal, respectively. The distance between the inner microphones, the matched OKTAVA pair, was 12 cm and between the outer pair (BEHRINGER) 15 cm. A standard PC display was placed behind the camera showing the GUI including the sentence to speak and a time scale that indicated the recording process of every sentence (see Figure 2). The table was adjustable in height in order to avoid changes in between the entire setup while being able to capture data from speakers of different body height in comparable geometric conditions.

5.2. Speaker specific preparations

Before starting the recording, every speaker had to find a central position relative to the camera that would ensure similar angles of the camera views. That comprised finding the optimum position of the chair, the right height of the table and an unrotated head position of the speaker. Because of the small distance between the camera and the speaker's face, finding and keeping the ideal position was a crucial task. Since the signal from the stereo camera cannot be displayed while recording, when raw data is saved continually,

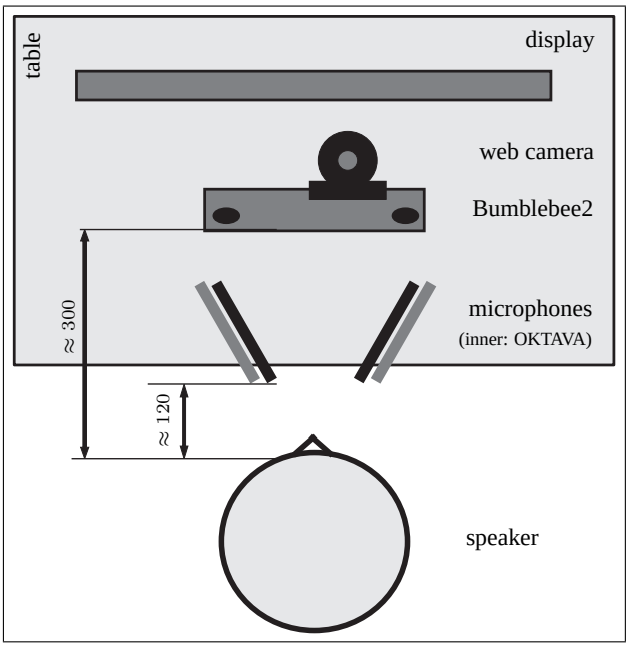

Figure 2: Setup structure in a topview

Figure 3: One speaker during recording

a control image provided by the web camera was shown to help the speaker in keeping the optimum position. As reference, a static image previously taken was additionally displayed during the entire recording session, showing the speaker in the desired position (Figure 3).

6. Data collection and postprocessing

6.1. Recording process

The recording process was controlled by the speakers themselves, who had to start a sentence record via the computer mouse. In the case of a failed recording (e.g. pronunciation problems), no data was stored and a repetition could be triggerd via the GUI. After every sentence, the raw audio and video data residing in memory buffers were saved on the hard disk. The frame rate of the video data is 32 fps, which is the maximum rate achievable at the highest color image resolution. The four audio channels are saved as 32-bit signals at a sampling rate of 16 kSamples/s.

6.2. Video postprocessing

The video data had to be postprocessed using the Point-Grey SDK. Firstly, RGB images had to be created from the raw data, which is saved as the popular Bayer filter mosaic, that is used with many digital photosensors. Secondly, the resulting images have to be rectified. Since the camera is already paraxial, rectification only compensates for misalignment of the sensors and lens distortions. For that purpose, camera specific calibration data provided by the manufacturer was used.

6.3. Audio postprocessing

A lot of noise sources have been present during the recordings, such as ventilation, elevators and sounds coming from outside the building (street noise). In order to cope with low frequency noise coming from this room environment, a high pass filter has been applied to the audio channels. Since the preamplifier used has no built-in low cut functionality, as many others do, this had to be implemented in software. For that purpose, a second order audio highpass IIR-filter with a cutoff frequency at 80 Hz and a slope of 12 dB per octave was used. This filter was reimplemented in MATLAB using the algorithm that is used in the open source audio editor software AUDACITY. For testing purposes, this filter has been applied to the audio channels of the GRID corpus with no measurable effects on the speech recognition results of an HMM-based recognizer. Additionally, the audio signals were scaled in order to provide a similar loudness of the audio channels over the entire database. The scaling is done pairwise using an RMS normalization algorithm similar to the one used in the open source software REPLAY GAIN resulting in an average level of 20 dB below the possible peak level. Nevertheless, all wave files containing the original audio data are stored, separately.

6.4. Merging and compression

After the postprocessing steps, audio and video data have been merged into two video files per sentence. For that step, a user contributed freely available MATLAB function has been used (Richert, 2008). The resulting files have a size of approximately 115 MB each, leading to a total storage size of more than 20 GB of all 200 files per speaker. For better everyday use handling, those files have been compressed using the free H.264 video encoder of the *ffdshow tryouts* project (ffd, 2009), resulting in sizes of approximately 1.7 MB per file. Similar to the audio, all original video data have also been stored, so that further examinations on uncompressed data will still be possible.

In addition to the speaker recordings, ten samples without a speaker have been recorded, providing extra information about the background color, illumination and the background noise present in the room.

7. Summary

A database for audiovisual speech recognition has been presented. It consists of a total of 2000 sentences containing six words each available as stereoscopic video files recorded by 20 speakers. Figure 4 shows samples of two speakers out of the database. The effects of incorporating stereoscopic video data on the robustness of AVSR will be one aspect in future experiments. Especially, using 3D-features gathered from both video channels or features calculated on disparity maps of the speaker will be of high interest. As a first experiment, such disparity maps calculated from database images extracted directly from the compressed video files without any adjustments using the Birchfield algorithm (Birchfield and Tomasi, 1998) are shown in Figure 5. After completion of time-stamped word-level transcriptions of the recordings, a release of the database for free research use is planned on `http://www.emsp.tu-berlin.de/forschung/AVSR`.

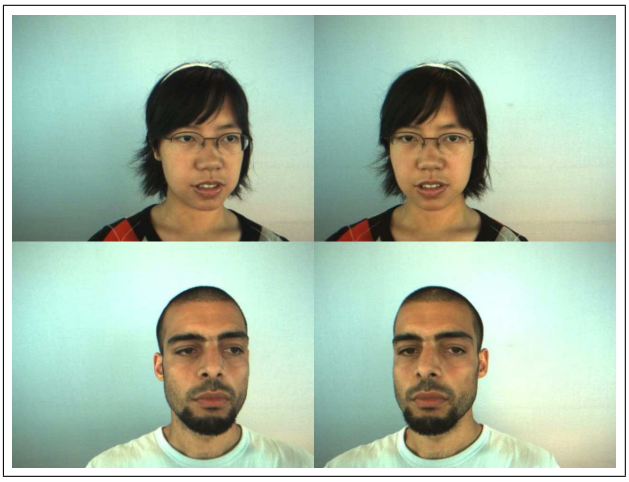

Figure 4: Samples of two speakers of the WAPUSK20 database

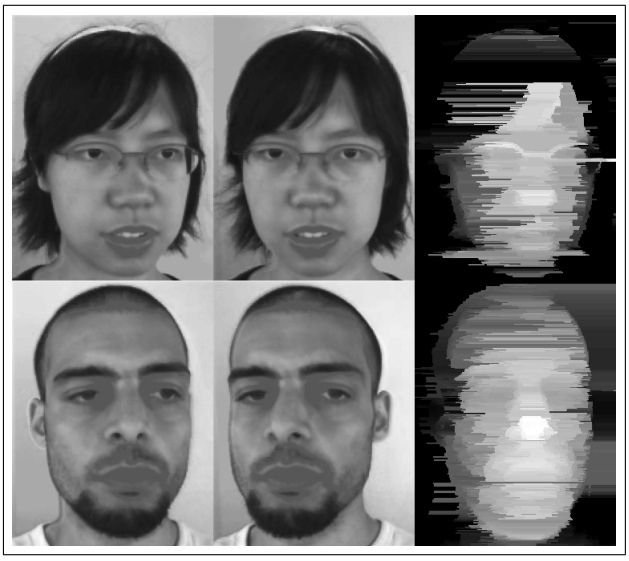

Figure 5: Disparity maps using images of Figure 4

8. Acknowledgements

The authors would like to thank all students and staff of the Technische Universitaet Berlin (TU Berlin) who volunteered as speakers for this corpus. Special thanks go to Dipl.-Ing. Stefan Thiel from the Chair of Lighting Technology and to Wilm Thoben from the Audio Communication Group of TU Berlin for their kind support.

9. References

I. Almajai and B. Milner. 2008. Using audio-visual features for robust voice activity detection in clean and noisy speech. In *Proc. EUSIPCO*.

S. Birchfield and C. Tomasi. 1998. Depth discontinuities by pixel-to-pixel stereo. In *IEEE International Conference on Computer Vision*, Bombay, India.

M. Cooke, J. Barker, S. Cunningham, and X. Shao. 2006. An audio-visual corpus for speech perception and automatic speech recognition. *Acoustical Society of America Journal*, 120:2421–2424.

ffd. 2009. the all-in-one codec solution. ffdshow tryouts. (WWW), `http://ffdshow-tryout.sourceforge.net/index.php`, last checked: 16 Mar 2010.

T. Gan, W. Menzel, and S. Yang. 2007. An audio-visual speech recognition framework based on articulatory features. In *Int. Conf. on Auditory-Visual Speech Processing (AVSP2007)*, Antwerp, Belgium.

R. Goecke. 2005. Current trends in joint audio-video signal processing: a review. In *Proceedings of the Eighth International Symposium on Signal Processing and Its Applications*, volume 1, pages 70–73.

D. Kolossa, S. Zeiler, A. Vorwerk, and R. Orglmeister. 2009. Audiovisual speech recognition with missing or unreliable data. In *Int. Conf. on Auditory-Visual Speech Processing (AVSP2009)*, Norwich, UK.

Y. Lan, B.-J. Theobald, E.-J. Ong, and R. Bowden. 2009. Comparing visual features for lipreading. In *Int. Conf. on Auditory-Visual Speech Processing (AVSP2009)*, Norwich, UK.

LST. 2009. Lynxtwo. Lynx Studio Technology (WWW), `http://www.lynxstudio.de/en/products_lynxtwo_info_2.html`, last checked: 16 Mar 2010.

G. Potamianos, C. Neti, J. Luettin, and I. Matthews, 2004. *Audio-Visual Automatic Speech Recognition: An Overview*. MIT Press.

PTG. 2009. Stereo vision products: Bumblebee2. PointGrey Research, Inc. (WWW), `http://www.ptgrey.com/products/bumblebee2/index.asp`, last checked: 16 Mar 2010.

M. Richert. 2008. mmwrite. MATLAB Central (WWW), `http://www.mathworks.com/matlabcentral/fileexchange/15881-mmwrite`, last checked: 16 Mar 2010.