AN ONLINE SYSTEM FOR SYNCHRONIZED PROCESSING OF VIDEO AND AUDIO SIGNALS

Mary Mikhail Giovanni Palumbo Jinane Mohammad Mohamed El-Helaly Aishy Amer

Department of Electrical and Computer Engineering, Concordia University
Montréal, Québec, Canada
Correspondence: amer@ece.concordia.ca

Abstract

For many audio-visual applications, the integration and synchronization of audio and video signals is essential. The objective of this paper is to develop a system that displays the active objects in the captured video signal, integrated with their respective audio signals in the form of text. The video and audio signals are captured and processed separately. The signals are buffered and integrated and synchronized using a time-stamping technique. Time-stamps provide the timing information for each of the audio and video processes, the speech recognition and the object detection, respectively. This information is necessary to correlate the audio packets to the video frames. Hence, integration is achieved without the use of video information, such as lip movements.

The results obtained are based on a specific implementation of the speech recognition module, which is determined to be the bottleneck process in the proposed system.

Keywords *— Video processing, audio processing, signals integration, synchronized audio and video signals; time-stamping.*

1. Introduction

Integration and synchronization between audio and video signals is very much needed in multimedia applications because they have very strict timing constraints. Multimedia applications with audio and video need a synchronization scheme if they are transmitted and processed independently. The synchronization scheme is responsible for ensuring that the audio and video streams are synchronized after processing.

The system proposed in this paper is an online synchronized system for audio and video processing. The video and audio signals are processed separately and so there is a need to synchronize them such that they can be related to one another. The main processing system is composed of a video processing module, that produces a segmented object image, and a speech recognizer that produces recognized text. The methods presented here describe how it is possible to relate the audio and the video outputs using their relative timing information.

2. Related Work

Kuo et al. [1] present a scheme that guarantees the synchronization playback of audio and video streams according to Real-time Transport Protocol (RTP) based on analysis of the jitter resistance, the end-to-end processing delay, and the buffer size required in order to improve the synchronization between audio and video streams. Kuo et al. [1] use time-stamps on the audio and video streams to determine the actual playback time to achieve synchronization. They also use a variable buffer system, where the buffer size depends on the network parameters since the method is based on an implementation using RTP. Their approach provides a synchronization method for systems that involve RTP and not online and processed signals as our system.

Robin [2] presents an outline of the audio synchronization requirement. The major purpose is to find a phase relationship between audio and video signals based on the number of audio samples per video frame while maintaining it to an integer to keep the audio packet synchronized to its relative video frame. Robin [2] provides a time-stamp approach for synchronization of audio and video signals. However, the audio and video signals are not processed as in our system.

Lienhart et al. [3] present a synchronization scheme for audio and video streams to be used in wireless applications. Time-stamping information is obtained at A/D conversion time and embedded in the transmitted stream. This information is then used at the destination to convert the sampling rates of the audio and video streams and therefore synchronize the streams accordingly. The synchronization in [3] is for a wireless communication system and is not focused on online applications as in the proposed system.

MPEG-1 Systems white paper [4] presents the time-stamping method for MPEG-1: PTS (Presentation Time-stamp), which is a reference clock value inserted at encoding time into the headers of the data stream at various intervals. In general, the system time clock used in the encoding process produces clock values or time-stamps associated with the access units of the audio and video streams. The white paper is focused on encoding techniques and not on multimedia applications.

3. Overall System

The goal of the proposed system is to synchronize separately-processed digital audio and video signals. The

1-4244-0038-4 2006 IEEE
CCECE/CCGEI, Ottawa, May 2006

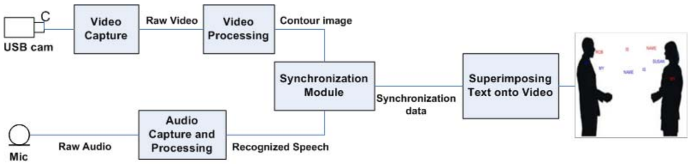

Fig. 1: Overall Modular System Design.

setup of the overall audio-video integration and synchronization system consists of five modules shown in Fig. 1: video capture, video processing, audio capture and speech recognition, superimposing graphical text onto video, and signal integration. The video and audio capture are performed using two different sensors, a camera and a microphone. These sensors are not connected, and so the signals captured are totally independent of one another. In traditional video cameras, the microphone used to record the audio is connected to the camera and is recorded at the same instance as the video.

3.1. Video Capture

The video capture is achieved through streaming (e.g., using a USB Webcam). The captured video is displayed using functions in the Open Source Computer Vision Library (OpenCV) [6]. The format of the video is Image Processing Library (IPL), which consist of RGB components, and a video resolution of 320x240.

3.2. Video Processing

The objective of the video processing module is to take as an input the images captured, i.e., the raw video, and perform edge detection and contour tracing to obtain the object silhouette, referred to as the contour image of video.

The nature of the video processing module [7] requires the conversion from RGB to YUV prior to processing, which is is then performed on the Y component of the video only, to obtain the black and white contour image.

Face detection and localization is also incorporated into the video processing module. It is achieved through the OpenCV [6] library. Face aspects are stored within predefined classifiers, which in turn are used to check for structures including frontal face and face profiles. These classifiers store the basic structure of a face and are used to detect faces on the captured images. The corresponding coordinates of each detected face are passed to the superimposing-text-onto-video module.

3.3. Audio Capture and Processing

The audio capture is built into the speech recognition module. Audio is captured through the Hidden Markov Model Toolkit (HTK) [5] open source library. The audio processing module takes as an input the raw audio and runs it through a speech recognition block which is built using the Hidden Markov Model Toolkit (HTK)[5], a set of functions and libraries for speech recognition. The user utters a sentence and the recognizer waits to detect a silence before recognizing it. The recognizer then outputs the recognized speech in the form of phonemes (syllables indicating the pronunciation of a word) in a text file. This text file, containing timing parameters as well as the phonemes, is then read by the audio processing module.

3.4. Superimposing Text Onto Video

After audio processing, the recognized speech (phonemes) is superimposed in the form of graphical text onto the video. Once again, the OpenCV [6] library is used. The text parameters, including text width, height, emphasis and font, are first initialized. Using these presets, the text is written to a captured video frame. The position of the text is determined by the coordinates of the previously detected faces. The text then flows from one face coordinate to the other on subsequent frames.

3.5. Integration and Synchronization

Since the audio and video signals are captured independently, there is a need to synchronize them. The objective of the synchronization module is to realign both the audio and video signals, which have been processed separately. Therefore, the final output of the system consists of the text, the speaker‘s recognized speech, displayed on the frame at which the sentence was spoken, moving from one speaker to the other. This process does not involve object-speech synchronization. Object-speech synchronization is the technique of relating the audio to a specific speaker and involves developing approaches based on speaker identification.

The proposed system setup is such that the video processing and audio processing are performed separately. In the true nature of things, however, the audio of a speaking person is synchronized with the individual’s lip movements and body interactions. Thus, a technique for the synchronization of the processed audio output and processed video output is proposed. The main idea is to present a synchronization system that does not use any video information (e.g., lip movements), but rather, uses timing information obtained from the audio and video processing. Each captured frame is processed through various filters [7], in order for the end result to become an object image.

There are two scenarios where is synchronization is required. The first case is when the audio delay is smaller than the video delay. In this case a frame-dropping technique can be implemented to compensate for the difference in delay times between the audio and video signals. The second case, as in our system, is when the audio delay is larger than the video delay. The video must therefore be delayed. This is implemented by buffering the video frames and relating the audio packets to the buffered frames.

The processing of each frame takes an amount of time, a delay which is incurred from the moment of capture of the raw frame to the moment when the processed frame is available. This delay can be masked by decreasing the output frame rate such that the delay is not noticeable. The other delay that is incurred is in the audio processing. The delay incurred from the moment of audio input to the moment of the recognized output is significant. Through experimentation, it is evident that in our system the audio delay is larger than the video delay. Therefore, there is a need to relate the audio output to the corresponding video output, by considering the delay of each process. If it were not for the significant audio delay, which is in the order of seconds, a simple solution for synchronization would be to drop frames and decrease the frame rate such that the delay of each of the audio and video processes is accounted for in the output. This situation is not the case here, since the audio delay is directly related to the length of the utterance spoken into the recognizer. Delaying the video according to the audio delay would result in an incomprehensible, jittery video stream.

The proposed solution is to develop a system that uses time-stamping and buffering to synchronize the processed audio and video signals. If the situation arises that the video delay is larger than the audio delay, the system is easily adapted since both the audio and video information are buffered.

Time-stamping is a method of obtaining the actual time at certain points in a process [1]. For the video processing, time-stamps are taken before the raw frame is passed on to the processing filters, and another time-stamp is taken after the processing. The difference between the time-stamps provides the processing time of each frame, and consequently the frame rate at the output of the system can be obtained.

For the audio processing, time-stamping is not as trivial. The output of the speech recognizer for each utterance is stored in a text file, with the corresponding recognized phoneme times. The time-stamps in this case are taken every time an output file is ready. Therefore, the delay of the audio is obtained using the difference in the time-stamps of the output files. This time represents the time of the real audio utterance plus the time of the delay of the recognizer.

Because the audio time-stamps include the delay, the synchronization concept must compensate for the delay, i.e., obtain the times without the delay incurred. The audio output file includes timing information for the phonemes. It provides the times of each spoken phoneme relative to the whole utterance. The audio output file also provides the time of the silence detection between the utterance and the recognized output. Therefore, from the measured time-stamps of the audio output files and the timing information read from the actual audio output files, the time of the spoken utterance is obtained relative to every other utterance, and does not include the delay time. This time is converted to its equivalent in frame numbers using the frame rate of the processed video. Therefore, the exact frame number of at which the utterance was spoken is known.

Kuo et al. [1] present a synchronization system that is also implemented using the time-stamp information of the audio and video signals. However, the time-stamp information is used to re-order the queued audio and video signal packets that are out of sequence due to the nature of the RTP protocol. The system proposed in this paper uses the concept of time-stamps and the queuing of the signals as in [1] to relate the audio packets to the video. However, re-ordering the audio and video packets is not required since there is no transport protocol involved in our system.

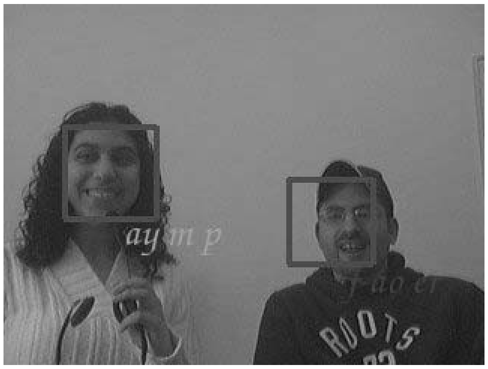

Fig. 2: Real Image: Two Faces.

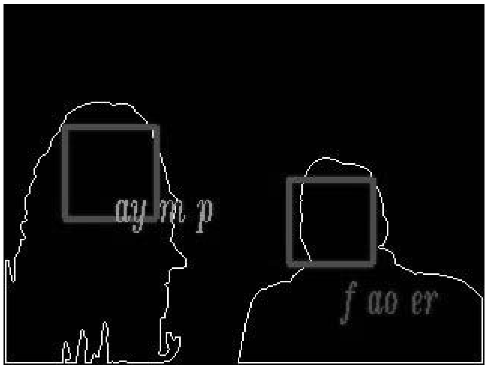

Fig. 3: Contour Image: Two Faces.

In the system presented the output of the audio is not ready in time for the video because of the incurred delays. To solve this problem, the processed video frames are buffered. The buffer size is set according to the average delay of the audio, obtained through experimentation. With the video buffer, the audio processing gets a buffer time in order for the synchronization process to determine the correct frame number at which the text needs to be superimposed.

4. Results

The following results of the system are obtained by subjective and objective evaluation. A sample output of the real and segmented object image (contour image) is displayed in Fig. 2 and Fig. 3, respectively. Similarly, Fig. 4 and Fig. 5 show an output taken with three faces. As can be seen, objects are correctly detected and speech is recognized and synchronized.

Objective evaluation is performed to show accurate syn-

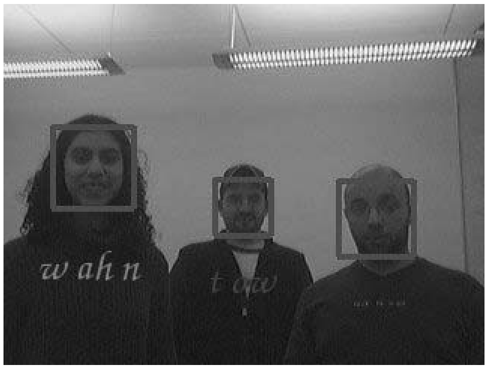

Fig. 4: Real Image: Three Faces.

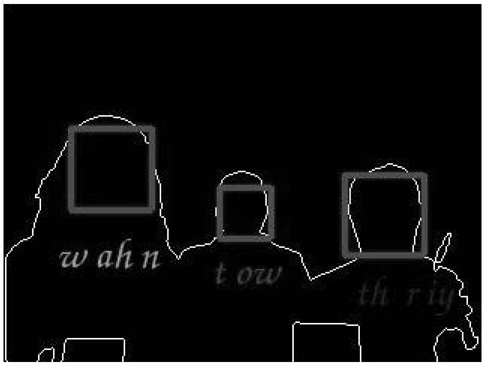

Fig. 5: Contour Image: Three Faces.

chronization results. Timing readings are obtained for both the video and audio processing blocks. From these delay times, we note that the bottleneck of the overall system is the audio processing block. The objective of synchronization is to match the audio with the "delayed" real-time video. The matching is achieved by inserting the audio at predetermined video frames. The buffered video is then output with the corresponding recognized audio.

To match the audio with the delayed video, the first variable calculated is the frame rate F_R. The F_R essentially comprises the total video delay δ_V, i.e., it is the rate of processed frames. The higher the δ_V, the lower the F_R and vice versa. This variable is constantly recalculated in real-time for increased accuracy. The second variable needed is the audio delay time δ_A. The F_R (in frames per second) multiplied by the δ_A (in seconds) is equal to the number of frames the audio should be delayed by. The system keeps track of the number of times it has synchronized the audio to the video. Note that this synchronization system cannot always produce perfect results due to the fact that frame numbers are rounded off to the closest integer.

5. Conclusion

A system for the integration and synchronization of audio and video signals is proposed in this paper . Synchronization is based on time-stamps. Due to the significant audio delay compared to the video delay, a complete real-time system could not be achieved. Therefore, a virtual real-time or online system is implemented through the use of a video buffer of constant size. Synchronization is accomplished by relating the audio delay of the speech recognizer to the corresponding video frame. The recognized text is superimposed onto the video frame at which the sentence is spoken.

Acknowledgments

The authors would like to thank Dennis Pavia and Elena Petrova for their contribution to the audio capture portion of this project.

References

[1] Chia-Chen Kuo, Ming-Syan Chen, and Jeng-Chun Chen, "An adaptive transmission scheme for audio and video synchronization based on real-time transport protocol," International Conference on Multimedia (ICME), pp. 403 - 406 , August 2001.

[2] Michael Robin, "The audio synchronization concept, Tech. Rep., Miranda, 1999.

[3] Rainer Lienhart, Igor Kozintsev, and StefanWehr Lienhar, "Universal synchronization scheme for distributed audio-video capture on heterogeneous computing platforms," Proceedings of the ACM International Conference on Multimedia (ICME), November 2003.

[4] Peter Schirling "MPEG-1 Systems white paper Multiplexing and Synchronization," International Organization for Standardization, October 2005.

[5] Machine Intelligence Laboratory, "Hidden Markov Model Tool Kit (HTK)," University of Cambridge, http://htk.eng.cam.ac.uk/, February 2006.

[6] SourceForge.net, OSTG (Open Source Technology Group), Open Source Computer Vision Library, http://sourceforge.net/projects/opencvlibrary/, August 2005.

[7] A. Amer, Memory-based spatio-temporal real-time object segmentation, in Proc. SPIE Int. Symposium on Electronic Imaging, Conf. on Real-Time Imaging (RTI), Santa Clara, USA, vol. 5012, pp. 10-21, Jan. 2003.