REAL-TIME AUDIOVISUAL SPEECH CAPTURE AND MOTION TRACKING FOR SPEECH-DRIVEN FACIAL ANIMATION

A Senior Honors Thesis

by

KARL ADAM JABLONSKI

Submitted to the Office of Honors Programs
& Academic Scholarships
Texas A&M University
in partial fulfillment of the requirements of the

UNIVERSITY UNDERGRADUATE
RESEARCH FELLOWS

April 2003

Group: Computer Engineering

REAL-TIME AUDIOVISUAL SPEECH CAPTURE AND MOTION TRACKING FOR SPEECH-DRIVEN FACIAL ANIMATION

A Senior Honors Thesis

by

KARL ADAM JABLONSKI

Submitted to the Office of Honors Programs
& Academic Scholarships
Texas A&M University
in partial fulfillment of the requirements of the

UNIVERSITY UNDERGRADUATE
RESEARCH FELLOWS

Approved as to style and content by:

Ricardo Gutierrez-Osuna
(Fellows Advisor)

Edward A. Funkhouser
(Executive Director)

April 2003

Group: Computer Engineering

ABSTRACT

Real-Time Audiovisual Speech Capture and

Motion Tracking for Speech-Driven

Facial Animation. (April 2003)

Karl Adam Jablonski
Department of Electrical Engineering
Texas A&M University

Fellows Advisor: Dr. Ricardo Gutierrez-Osuna
Department of Computer Science

Currently, some methods for implementing facial animation systems are based on a direct subphonemic mapping of speech acoustics onto orofacial motion. Although these systems provide all of the necessary components for the detection of facial movements through speech patterns, they are limited by their storage requirements and high-priced equipment. Furthermore, the audiovisual processing can be delayed, taking place after the initial capture and transfer of all required audio and video. The method described in this work focuses on developing a system aimed at achieving realistic facial animations in real-time, incorporating synchronized capture of both audio and facial motions. With data processing taking place at run-time, the system saves only crucial information to file, providing additional storage for larger amounts of data. Results confirm the efficiency of this approach, which provides a fast, accurate and inexpensive tool for speech-driven facial animation.

ACKNOWLEDGEMENTS

Dr. Ricardo Gutierrez-Osuna was the driving force behind this research, and is greatly acknowledged for his time, knowledge and patience. Praveen Kakumanu and Nishant Balan are also distinguished for their support in several stages of the project. Roberto Pockaj is acknowledged for supplying the Facial Animation Engine.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Human-computer interaction becomes significantly more important as more and more computational devices come into our daily life. One aspect limiting such interaction is the inability of computers to match the human perceptual capabilities. The importance of perceptual ability is astounding. Just as animal survival depends on highly developed sensory abilities, human cognition depends on highly developed abilities to perceive, integrate, and interpret visual, auditory, and touch information [3]. Clearly, computers and other electronic devices would be considerably more powerful given their ability to mimic the abovementioned human perceptual abilities [2]. This in turn would allow for an enhanced interaction between humans and computers. Speech and facial motion recognition serves as a key element in enhancing the interaction between humans and computers. A variety of published papers along with a set of perceptual experiments have proven the significance of this technology in today's world. The research gathered in these tests demonstrated that facial animation can convincingly provide useful and subjective benefits in communication between humans and computers [4]. Some of these applications include enhancing the communication with hearing-impaired individuals, and improving the intelligibility of speech in noisy environments. Recognition of humans' emotional states during waiting times, and improving the performance of current speech recognition systems are also potential contributions. One other possible application of this system lies with the computer animation industry. Providing faster and more convenient techniques for

---

This thesis follows the style and format of ACM Transactions on Human-Computer Interaction.

animating realistic facial expressions will further enhance animated media in the future. An efficient implementation of such a system calls for the development of a technique utilizing synchronized capture of audio and video with real-time processing. Earlier attempts at implementing facial animation focused on capturing data in synchrony, with processing taking place after the initial capture [4]. Due to the fact that audio and video files grow rapidly, the system's disk storage was limited in the amount of data it could capture. These shortcomings eventually proved the system to be impractical. With the new technique the system has the ability to perform all of the required processing on the spot, saving to file only essential data. This in turn allows the user to collect vast amounts of data, ready to be used in further analysis and facial animation. In addition, the proposed system is remarkably inexpensive, a concern previous efforts in implementing facial animation seemingly did not address. The whole system consists of mainly off-the-shelf PC technology, totaling less than $1000.

To develop a system capable of producing three-dimensional animations of a human face driven by a speech signal, several additional goals had to be accomplished. First, following a thorough review of all background information, testing of all hardware and software components was completed. This consisted of testing the available PC system, microphone array, and the camera. The next step involved synchronizing the capture of video frames with speech from a sound card. Subsequently, speech and video processing was achieved without compromising the efficiency of the system. This consisted of implementing specific algorithms in order to extract particular speech and video parameters in real-time. Finally, in order to verify the accuracy of the system, facial animation was achieved by utilizing the Facial Animation Engine (FAE) with MPEG-4 specifications. The resulting system provides an accurate and high-speed solution for generating speech-driven facial animations. With the ability to collect enormous amounts of audio-visual data, the system serves as an important tool in estimating facial movements from speech.

II. SYSTEM ANALYSIS

A. Overview of System Components

As mentioned before, the proposed system consists primarily of off-the-shelf PC technology. All of the components are inexpensive and easily obtainable. The system is based on a personal computer (Pentium IV 2.0 GHZ, 512 MB RAM.) This also includes an audiovisual capture card (Winnov Videum 1000 Plus) capable of acquiring 640x480 video at 30 frames per second in synchrony with an audio stream at 16 kHz. IBM's PupilCAM, a system used for detecting the pupils of a human, was employed as the main video input device. The PupilCam is based on two sets of time multiplexed infrared (IR) light emitting diodes (LED) and an analog, black and white, National Television Standards Committee (NTSC), Charge Couple Device (CCD) camera [5]. One set of LEDs is positioned near the camera's optical axis so that the pupils appear bright, and the second set is positioned off-axis, so that the pupils appear dark. The PupilCam is programmed to be synchronized with the video grabber, so that even frames contain the on-axis LED image and the odd frames contain the off-axis LED image. Subtracting the two images will result in two objects representing pupils that are easy to segment. The PupilCam system was then modified to track a number of retro-reflective markers, which reflect only infrared light. Since the markers reflect the most infrared light with only the inner axis LEDs turned on, the outer LEDs were turned off permanently. Furthermore, to preserve the reflectivity of the markers within the capture environment, the intensity of the LEDs was reduced significantly. Finally, to limit the amount of extraneous (light) as well as excessive infrared light, several wratten filters were also installed. Consequently, this enabled the system to focus specifically on the infrared light produced by the retro-reflective markers used in the capture process. Audio input was accomplished with an array microphone (Acoustic Magic's Voice Tracker), chosen because of its noise filtering capabilities and superior range of capture.

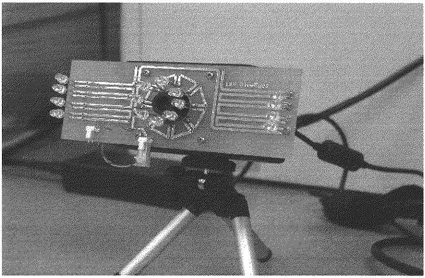

Fig. 1. IBM's PupilCAM.

The voice tracker, utilizing eight microphones geared in the direction of the speaker, provided numerous advanced features including noise reduction, increased sensitivity and echo elimination. The proposed system was developed on Microsoft's Visual C++ platform, incorporating the use of Winnov's Software Development Kit (SDK) that was provided with the hardware. Overall, the price tag for the system falls near the $1000 range, an inexpensive alternative to other high-priced facial animation technologies.

B. Video Processing

The video stream is captured at 30 frames per second, in the Red Green Blue (RGB) video format. To increase the performance of the system, the video capture resolution was adjusted to 320x240. In order to accurately track specific facial features, 17 markers were placed on the face of a subject at several key positions, as defined by the MPEG-4 standard. This resulted in the frame-by-frame tracking of 22 individual feature points, consequently used in the animation process. These points included the left and right eyebrow region (4.4, 4.2(x), 4.2(y), 4.1(x), 4.1(y), 4.3), the cheeks (5.4, 5.3), the mouth (8.4(x), 8.4(y), 8.6, 8.1, 8.5, 8.3(x), 8.3(y), 8.8, 8.2, 8.7), the nose (9.3(x), 9.3(y)), and the chin area (2.10, 2.1). Figure 3 shows the locations of MPEG-4 facial feature points.

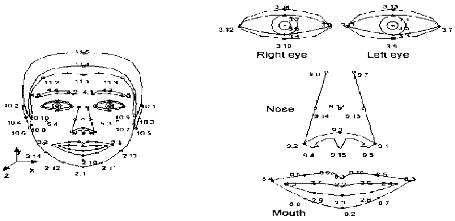

Fig. 2. MPEG-4 Feature Points.

The tracking process begins with the user manually entering the initial position of each marker by means of a graphical user interface. The selections are made on a 640x480 snapshot of the live video image to be recorded, as shown in Figure 3.

Fig. 3. Video tracking before the start of capture.

The algorithm automatically converts the image to black and white, thereby segmenting the markers, which become distinctly bright as a result of IR imaging. Tracking is performed by centering the user-defined marker positions in an 11x11 pixel template at the first frame of capture. Starting with the second frame, a 21x21 pixel search area is defined and centered at the marker position from the previous frame. The search is then performed by finding the closest match of the initial 11x11 pixel matrix within the ensuing search area. This is achieved by sliding the initial pixel template within the search area, and computing the resulting correlation coefficient at each change of a row or column. The correlation coefficient is found by first multiplying each element of the search matrix by the corresponding element of the initial template, and summing up the resulting numbers. This number is then stored in a vector, and compared with all preceding coefficients. If the new number is greater than or equal to the largest coefficient stored in the array, it then becomes the closest match, and its (X,Y) position is recorded. At the final change of a column, the greatest correlation coefficient is chosen. The corresponding (X,Y) position is printed to file and stored as the new center point for the ensuing search area. The detailed computations needed to determine the correlation coefficient can be found in Table 1.

Table 1. Description of the Correlation Ceofficient procedure.

For each feature point to be tracked, a square template (a) of size *NxN* is initially created and centered at the (X,Y) position of the previously selected marker. Next, a search window (b) of size *KxK* is defined in the subsequent frame, where $K=(2*N)-1$. Thus, the equation to find the correlation coefficient becomes,

$$CC = max\{\Sigma((a)i,j * (b_n)i+r,j+c)\},$$

where $(a)i,j$ represents the rows and columns of the initial square template, incrementing from (0,0) to (N,N), and $(b)i,j$ represents the rows and columns of the search area incrementing from (0,0) to (K,K). Variables r and c represent the amount of displacement in the search window, with r being the row displacement incrementing every time the initial template moves down one row, and c being the column displacement incrementing each time the template is moved one column over. Variable *n* represents the corresponding frame of video currently being processed.

This process is repeated every time a frame is captured, tracking all markers individually. Thus, for any given marker, in one second of video data, 30 distinct (X,Y) coordinates are determined and saved to disk. Figures 3 and 4 illustrate the subject's face before and during the tracking process.

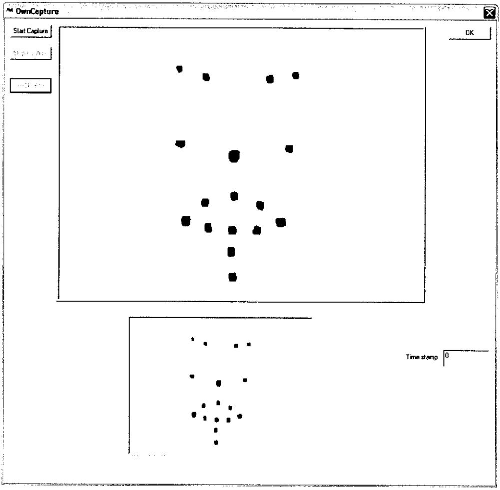

Fig. 4. Video tracking during capture.

Next, a vector of relative displacements is computed by subtracting the coordinates of the first frame from each subsequent frame of video data. This is achieved by defining the first video frame as a reference frame that signifies the neutral face of the subject. The resulting vector is then converted to a special file to be loaded into the Facial Animation Engine (FAE) for verification purposes. All of these computations are done on the spot, at every occurrence of a captured frame.

C. Speech Processing

The speech signal is sampled at 16 kHz, with each sample occupying 2 bytes of audio data. This signal is formatted in the Standard Windows PCM WAVE format, which contains only Pulse Code Modulation data without compression. PCM is the only format that saves the entire wave completely, with no data loss. By utilizing the advanced features of the array microphone, the audio signal is filtered with proprietary noise reduction algorithms to cut out background noise and other reverberations. In addition, the microphone also employs Location Dependent Squelch (LDS), which de-sensitizes the device to sounds coming from pre-selected directions. Combined with its automatic electronic steering toward the speaker, the microphone produces an audio signal comparable to utilizing a spectral subtraction algorithm for noise reduction. This procedure can be seen in figures 5 and 6 below, which show a generic waveform depicting a speech sample supplied with the software, and one sampled through the Voice Tracker.

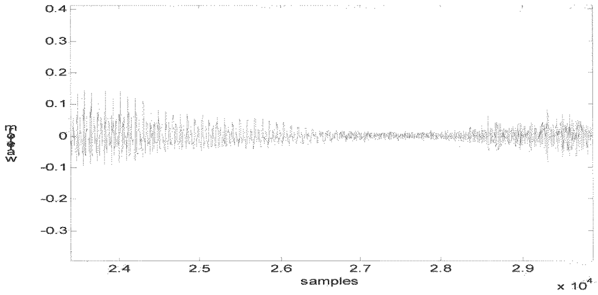

Fig. 5. Waveform before Voice Tracker filtering.

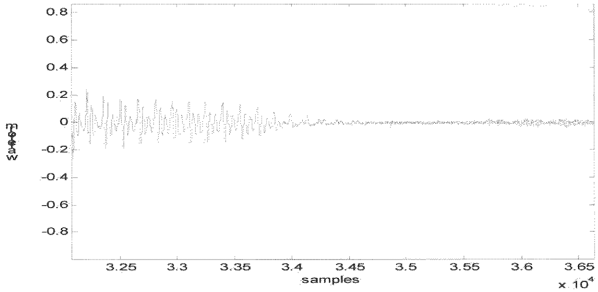

Fig. 6. Waveform after Voice Tracker filtering.

To facilitate proper synchronization, each audio frame is divided into 30 equal-size blocks and processed with the corresponding video frames. Two different speech processing algorithms are employed to extract acoustic features from each block of audio. First, Mel Frequency Cepstral Coefficients (MFCC) are utilized to model perceptual features of speech production. In the procedure, a DFT spectrum of a signal is frequency-warped through a Mel-scale transformation, and amplitude-warped using a logarithmic transformation [2]. Twelve cepstral coefficients are extracted for a given audio window. The system implements this algorithm through a shared library scheme, utilizing Matlab's C Math Library, which is available as a stand-alone application. This is the main source of the proposed system's audio processing approach, as it has been previously proven to be the best technique in predicting facial movements from speech [2]. Fig. 7 and 8 illustrate the effect of this algorithm on the TIMIT sentence, "A huge tapestry hung in her hallway". The figures show a generic waveform of the speech pattern along with the MFCC representation depicted as a color map. In the MFCC figure, each column corresponds to a specific time step in frames, and each row represents the trajectories of a coefficient over time.

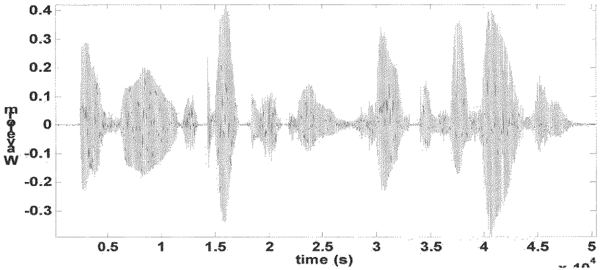

Fig. 7. Sample speech waveform for the TIMIT sentence "A huge tapestry hung in her hallway".

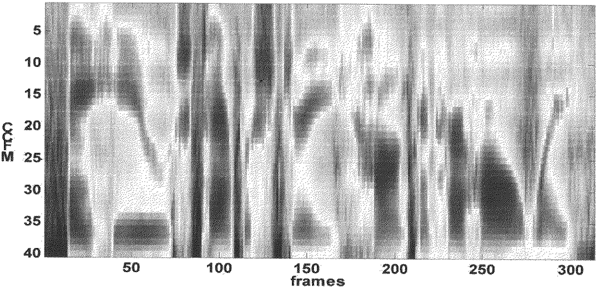

Fig. 8. Resulting MFCC analysis for the TIMIT sentence "A huge tapestry hung in her hallway".

In addition, Linear Prediction Coefficients (LPC) are used as an optional algorithm to model the articulatory parameters exploited by the auditory system. This technique models speech as a source-filter system excited at the glottis by noise, or by voiced sounds, and articulated through a transformation dependent on the vocal tract configuration for a given speech sound[2]. Given that facial positions affect the filter, and LPC coefficients are designed to encode articulatory movements, this approach also provides reasonable approximations for lip-syncing models generated by speech [2]. The abovementioned algorithm, although fully implemented within the system, is optional and can be utilized to further enhance the audio-visual analysis.

D. Audio-Video Synchronization

The proposed system achieves audio/video synchronization through both hardware and software. Winnov's Videum 1000 Plus frame-grabbing card simplifies the capture process with the use of timed event handler functions. The event functions control access to the software-supplied ActiveX controls, which provide a standard interfaces for embedding, user interface, methods, properties, events, and persistence. The functions give the user an opportunity to implement capture to a memory buffer, inspect the audio/video data, modify it, and pass the data to an API function for further analysis. Each event function gives the user access to the raw data, along with several other crucial pieces of information including frame timestamps, total number of bytes recorded in the specified audio/video frame, the buffer length, and flags indicating the validity of each audio/video frame.

For video, events fire every time a frame is captured, just then allowing the user access to the data in a specified frame. As previously mentioned, video capture is performed at 30 frames per second. Thus, video events give the user data access at 33ms intervals during the capture process. Audio events are structured differently, giving the user data access once an audio buffer has been filled, consequently once per second. To facilitate continuous capture of audio data, the hardware specifies two distinct sets of buffers. Initially, a large driver-owned audio buffer gets filled as video data comes in on a frame-by-frame basis. Concurrently, audio data is transferred over to a user-owned buffer where it is filled until it reaches its predefined capacity. When the event function procedures are running and the user buffer is held up, data is fed into the driver buffer and is continuously captured until the user buffer is released. The system defines the size of the user-owned buffer to be (# of channels) * (Bytes Per Sample) * (Sampling Rate). With the sampling rate set to 16kHz and 16 bits per sample through one channel, each user-owned buffer is filled with 32KB of data before granting the user access. Once the user-owned buffer is full, it fires an event, thereby giving the user access to the data. At that point the user can inspect the data, modify it and send it back to its final destination. This process provides continuous protection from data loss in spite of various intricate calculations within the system. Fig 9 depicts this process.

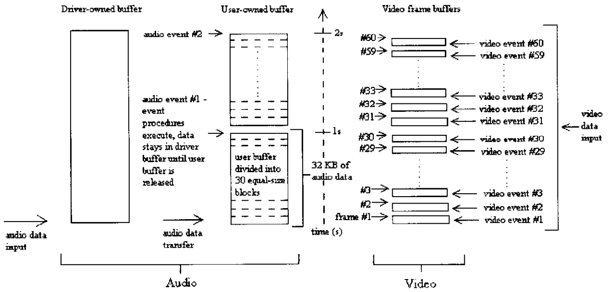

Fig. 9. Audio/Video Synchronization Scheme.

Software synchronization is achieved by effectively partitioning each audio window into 1/30s blocks of data. Thus, 30 equal-size audio blocks are created, according to the video frame rate. Each block is then fed into both speech processing functions (MFCC and LPC), which extract the designated audio features sequentially. Once the audio parameters are calculated, the system interleaves the corresponding marker positions with each set of audio features. Optionally, to further prevent loss of audio data, each audio window can also be broken into additional audio blocks. In this scheme, the audio frame can be broken into $n+1$ blocks, with n being the number of video frames. This way each audio block corresponds to a little less than one video frame, thereby spanning the subsequent blocks across the video frame gaps. This approach provides some overlap between audio and video frames, thus eliminating the uncertainties associated with data loss.

D. Data Collection

In order to effectively test the system accuracy in producing facial movements from speech, substantial amount of data was collected utilizing a predefined list of sentences. This small database was extracted from the TIMIT compact set, which contains phonetically balanced sentences specifically designed to provide speech data for acoustic-phonetic studies, and for the development and evaluation of automatic speech recognition systems. The sentences are spoken by a female American English speaker. During each recording session, the subject is asked to utter several sentences from the database. To minimize head motion, the speaker holds her head on each side with her arms, resting her elbows on the table and staring straight into the camera. The speaker's initial facial pose mirrors the neutral face position, as defined by the MPEG-4 standard. MPEG-4 defines this expression through several detailed instructions, including the subject's gaze, lip positions and pupil diameter. By employing this technique the system reduces any variations that might come about as a result of altering the initial facial expression. The neutral face functions as a reference frame in the tracking of retro-reflective markers. After the processing is completed, data is saved to disk, renamed, and fed into the animation engine along with the corresponding wave file. One limitation to the data collection process is the existence of a matching video file. For debugging purposes, the system also allows an AVI file to be collected. This uncompressed file depicts the marker tracking process. This feature, however, should be used with caution, as the size of the AVI video file can grow substantially for long acquisition times. This limits the overall performance. Thus, since capture of the AVI file is not essential in the facial animation process utilized in this study, it can be omitted in the case of extensive data collection.

E. Animation

In order to verify the precision of the marker-tracking algorithm employed in the proposed system, as well as the accuracy of its facial feature estimations, the resulting data is integrated with the Facial Animation Engine (FAE). The engine is designed to provide a hi-level interface capable of animating MPEG-4 compliant faces at high frame rates in synchrony with an audio track[6].

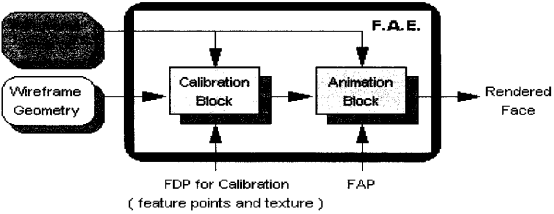

Fig. 10. Facial Animation Engine block diagram from [6].

The FAE uses the Calibration block to reshape the proprietary face model according to the received set of feature points. It then utilizes the Animation block to generate, in real-time, the *"animation rules"* which animate the face in response to the FAP stream [6]. The FAE inputs consist of the face model, the FAP file that describes the displacements of all relative feature points, as well as a corresponding WAVE file. As mentioned before, the FAP file is generated by tracking the 22 designated facial features, and calculating their relative displacements on a frame-by-frame basis. Each feature displacement is transformed according to its assigned Facial Animation Parameter Units (FAPU), which represent fractions of key facial distances.

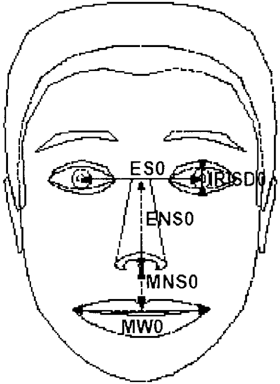

Fig. 11. MPEG-4 Facial Animation Parameter Units from [6].

These special distances vary from person to person and must be measured to facilitate an accurate conversion to FAPUs. The measurements were made using a simple tape measure and a conversion chart. A simplified model of the complete procedure utilized in the proposed system is depicted in Fig. 12.

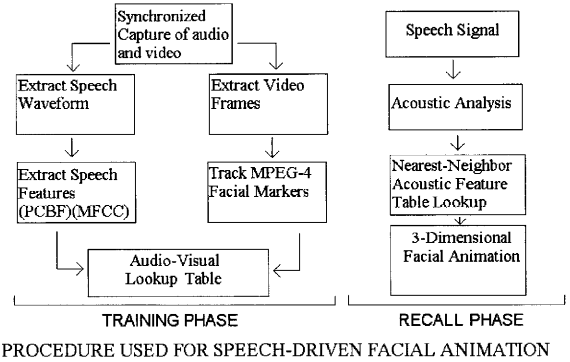

Fig. 12. Speech-driven facial animation procedure (adapted from [2]).

III. CONCLUSIONS

The system described in this work performs exceptionally well. However, as a proof-of-concept system, it has limitations that should be addressed in further studies. First, even though head motion is reduced during the capture process, it cannot be totally avoided. A moderate alteration of the subject's initial head position might result in the loss of accuracy, consequently providing misleading data. Furthermore, human errors can occur in measuring facial distances and marker placement positions according to MPEG-4 specifications. Finally, the speaker's facial features might be proportioned differently from the MPEG-4 FAPU definition. Facial features vary from person to person. This might cause some specific markers (like those placed on the outside of the eyebrow) to temporarily disappear during the capture process, again supplying the animation engine with erroneous data. Further inquiries involving the proposed system should take under consideration the possibility of the abovementioned concerns.

Overall, the technique employed in this study performs well, exhibiting exceptional functionality and reliability. Results obtained in this work are expected to provide a significant contribution to speech-driven facial animation systems. The collected data is verified through the Facial Animation Engine and has proven to be an accurate representation of human facial motion. This can be seen in figure 13, which depicts the tracking process and the resulting animation using the FAE.

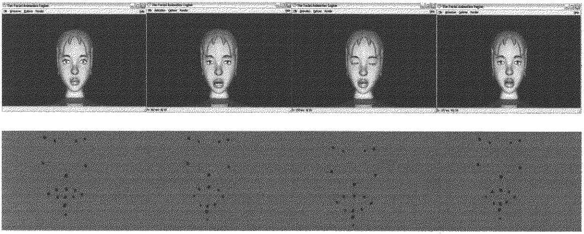

Fig. 13. Animation resulting from the tracking process.

The system was built with a very moderate budget, satisfying one of the main objectives of the project. It is crucial that the final product be constructed with affordable, off-the-shelf PC technology. Altogether, the equipment utilized totaled near $1,000. More importantly, speech and video processing is performed in synchrony, at 30 frames per second, while the data is being collected. Although previously built systems have primarily been able to produce similar capture and animation results, they do not deal with the data in real-time. This presents a major problem, as it is extremely time-consuming and inefficient to process considerable amounts of unwanted data. The achievement of this stage of the project allowed for the collection of substantial amounts of data, a vital step in producing efficient future results. Consequently, all that is saved to disk is a small text file with the (X,Y) positions of the markers, a file describing the audio parameters, along with the corresponding wave file. Utilizing a relatively simple computational procedure, the proposed technique produces realistic facial animations in real-time, providing extensive data capture and increased performance for any speech-driven facial animation system.

REFERENCES

1. DAVIS, J.W., AND VAKS, S. 2001. *A Perceptual User Interface for Recognizing Head Gesture Acknowledgements.* Ohio State University, Columbus, OH.
2. ESPOSITO, A., AND GUTIERREZ-OSUNA, R., AND KAKUMANU, P. , AND GARCIA, O. N. 2000. *Optimal Data-Encoding for Speech-Driven Facial Animation.* Submitted to IEEE Transactions on Speech and Audio Processing. Also WSU Report, WSU-CS-02-04.
3. IBM Almaden Computer Science Research Center. www.almaden.ibm.com.
4. KAKUMANU, P., AND GUTIERREZ-OSUNA, R., AND ESPOSITO, A., AND BRYLL, R., AND GOSHTASBY, A., AND GARCIA, O. N. 2001. *Speech Driven Facial Animation.* Proceedings of the ACM conference on Perceptual User Interfaces. Department of Computer Science and Engineering, Wright State University, Dayton, OH.
5. KOONS, D. 2001. *PupilCam Construction Instructions.* IBM Almaden Research Center, CA.
6. LAVAGETTO, F. AND POCKAJ, R. 1999. The Facial Animation Engine: towards a high-level interface for the design of MPEG-4 compliant animated faces. *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 9, n.2, March 1999, 277-289.

7. MORIMOTO, C., AND KOONS, D., AND AMIR, A., AND FLICKNER, M.; *Pupil Detection and Tracking Using Multiple Light Sources*. RJ10117, 1998.

8. SLANEY, M. 1998. Auditory Toolbox: A Matlab Toolbox for Auditory Modeling Work, version 2. Interval Research Corporation, CA.

VITA

Karl Adam Jablonski

HOME ADDRESS

8210 Brixton
San Antonio, TX 78254
Tel: (979) 777-4449
Email: karljab@hotmail.com

EDUCATION

1999-present **Texas A&M University** **College Station, TX**

Major: Computer Engineering ***Minor: Math*** ***Graduation: May 2003***

- Overall GPA: 3.53 Major GPA: 3.8
- Texas A&M University Honors Program

EMPLOYMENT

2001-2001 **IBM** **Houston, TX**

Intern

Created a Network Management System using Tivoli Netview Management software and various other components. Worked on actively communicating with an assortment of devices, services and applications using the Simple Network Management Protocol.

1998-1999 **West Telemarketing** **San Antonio, TX**

Production

Processed phone orders for Fortune 500 companies and provided customer service to callers and clients. Assisted in a company project to motivate employees and increase productivity.

RESEARCH EXPERIENCE

2002-Present **Texas A&M University** **College Station, TX**

Undergraduate Research Fellow

A year long research project administered by the Honors Program. Specifically, the project deals with building a system aimed at achieving realistic speech-driven facial animations using commercial, off-the-shelf PC technology.

HONORS AND ACHIEVEMENTS

- Texas A&M Academic Incentive Award Scholarship
- Schlumberger Foundation Scholarship
- ESP Engineering School Program Scholarship
- Dr. R. D. Chenoweth Engineering Scholarship
- Polish National Alliance Scholarship

PROFESSIONAL MEMBERSHIPS AND OTHER ACTIVITIES

- Sigma Xi Scientific Research Society - Member
- Polish National Alliance – Recording Secretary (1998)/Member
- Golden Key National Honor Society , Engineering Scholars Program (ESP) - Member
- IEEE, Texas A&M Computing Society - Member
- Project Sunshine Volunteer Organization - Member (1999/2000)
- Familiar with AutoCAD, MS Word/Excel/PowerPoint/Outlook, Unix, Sun OS, Java, C++, C, Assembly,Verilog,Windows 98/ME/2000/XP, Pspice, MaxII+Plus, Rational, Matlab and others
- Athletics such as Basketball, Karate, Weightlifting, Strength Training and Running