Driving Synthetic Mouth Gestures: Phonetic Recognition for FaceMe!

William Goldenthal, Keith Waters, Jean-Manuel Van Thong, and Oren Glickman
email: {thal, waters, jmvt, oren}@crl.dec.com

Digital Equipment Corporation
Cambridge Research Laboratory
One Kendall Sq., Building 700
Cambridge, Massachusetts 02139 USA

ABSTRACT

The goal of this work is to use phonetic recognition to drive a synthetic image with speech. Phonetic units are identified by the phonetic recognition engine and mapped to mouth gestures, known as *visemes*, the visual counterpart of phonemes. The acoustic waveform and visemes are then sent to a synthetic image player, called FaceMe! where they are rendered synchronously. This paper provides background for the core technologies involved in this process and describes asynchronous and synchronous prototypes of a combined phonetic recognition/FaceMe! system which we use to render mouth gestures on an animated face.

1. Introduction

This paper addresses the problem of driving an animated face using audio data. We present a phonetic recognition system as a front-end process to generate visemes, the visual analog of phonemes. The paper provides background for the core phonetic recognition technology, based on Statistical Trajectory Modeling [1] and the core FaceMe! animation technology. It then describes asynchronous and synchronous web-based prototypes of combined phonetic recognition/FaceMe! systems.

We believe that facial animation is important because the emergence of the web has provided new compelling opportunities for human-computer interaction. The systems we describe are useful for applications such as Internet chat, very-low bandwidth virtual video-conferencing, and an enhanced animated audio player. Section 2. gives background on the history of facial gesture synthesis and describes the FaceMe! technology. Section 3. describes the STM approach to phonetic recognition and discusses the details of our implementation for facial animation. Finally, Section 4. discusses existing and planned prototypes and summarizes the state of this work.

2. Facial Gesture Synthesis

A sequence of phonemes, created by a speech synthesizer and aligned to their visual compliments called visemes (the visual analog of phonemes), provides sufficient context to create a real-time image of a talking human face. In a previous implementation [2] the phoneme stream is generated from the DECtalk formant-based text-to-speech synthesizer. However, binding the technology to text-to-speech results in the following limitations:

1. Applications can be driven from text only and
2. The audio synthesis sounds far from natural, resulting in an audio visual dichotomy between the fidelity of the image and the naturalness of the synthesized speech.

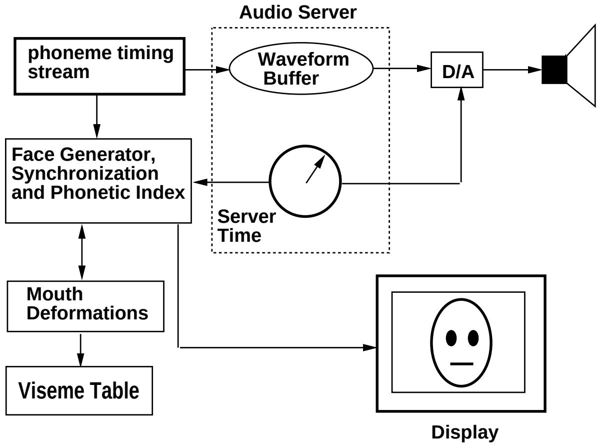

Figure 1: Speech synchronization for FaceMe!

To overcome these two limitations it is necessary to drive facial gestures from real speech. This technique is called lip-synchronization and has been the subject of prior investigation [3, 4, 5, 6]. To date the best approaches execute a limited phonetic recognition system where broad phonetic categorizations are computed from linear-prediction speech models or neural networks [3, 4]. While the results are superior to that produced from simple volume metrics, the underlying recognition remains primitive. In our system we use a complete high-performance phonetic recognizer.

By replacing the text-to-speech component of DECface and allowing a stream of phoneme timing symbols to be passed to the core engine of DECface, we can generate

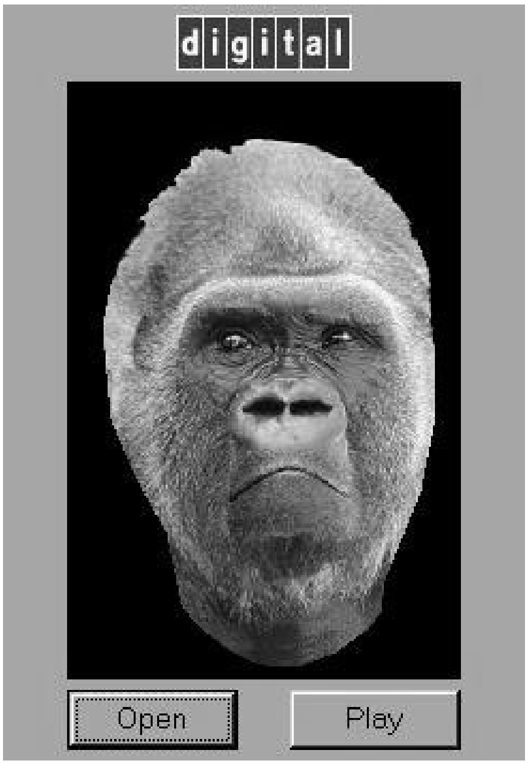

Figure 2: The plugin version of FaceMe! with a mapped image of a gorilla.

a real-time talking face with real speech. First, the phonetic recognition engine analyses the acoustic waveform to produce a phonetic stream. Then, the phonetic units are mapped to their corresponding visemes.

At each update time the current mouth posture is computed from two visemes and a timing marker. The timing marker computes an interpolated position using a non-linear cosine that provides the appearance of acceleration and deceleration that is consistent with a physical system. This implementation is depicted in Figure 1. A plugin version of the FaceMe! player is shown in Figure 2.

3. Phonetic Recognition

3.1. Statistical Trajectory Models

Digital's core phonetic and continuous speech recognition engines utilize segment-based technology. The front-end includes segmentation technology which works directly on the time waveform [7]. Acoustic-phonetic model construction and scoring is accomplished by Statistical Trajectory Modeling (STM) technology [1]. STM utilizes a segment–based framework to capture the dynamical behavior and statistical dependencies of the acoustic attributes used to represent the speech waveform. The approach is based on the creation of a *track* for each phonetic unit. The track serves as a model of the dynamic trajectories of the acoustic attributes over the segment. The statistical framework for scoring incorporates the auto- and cross-correlation properties of the track error over time, within a segment.

The acoustic–phonetic models used in this work are constructed for individual phonetic units and their *transitions*. Transition models enhance system performance by utilizing the acoustic information that spans adjacent segments. Transition models are well-suited for the STM approach because the transition regions are highly dynamic since the articulators are generally in motion during this interval.

To create the transition models, tracks were generated for each of the phonetic transitions found in the training corpus. Since the number of phonetic transitions is large, tracks were then clustered bottom–up in an unsupervised manner (see [1] for details on this clustering procedure). Clustering ceased when there was enough data to robustly estimate each track. After the transition tracks were estimated, clustering was resumed to determine the final clusters for estimating Gaussian p.d.f.'s, again using robust parameter estimation as a stopping criterion. This process results in most of the Gaussians being shared by multiple tracks. During the recognition process, the transition scores and (internal) segment scores are combined with a-priori and durational probabilities to determine the score for each hypothesized segment.

3.2. Experimental Results

The viseme recognition experiments were based on the TIMIT acoustic–phonetic speech corpus [9]. The primary data sets we used were the NIST designated training set consisting of 462 speakers, and a 50 speaker development set selected from a subset of the remaining speakers who are not in the NIST "core" test set. The eight "sx" and "si" utterances for each speaker were used for both training and test. The acoustic-phonetic models were constructed using the ten MFCC-based features which optimized performance on a prior classification task. Due to the incorporation of temporal correlation information, this resulted in a 30 dimensional statistical representation. A single full covariance Gaussian was used to represent the error p.d.f. for each acoustic-phonetic model. The models were trained on the 462 speaker TIMIT training set.

For this task, the purpose of the phonetic models was to maximize the viseme recognition accuracy for a given set of viseme classes. Using a larger number of viseme classes increases the animation fidelity which is ultimately achievable during rendering. The process for computing the viseme accuracy consists of performing phonetic recognition and then mapping the output of the recognizer and the TIMIT (hand-annotated) labels to a consistent set of visemes for comparison.

The number of phonetic models is an additional variable which effects the fidelity of the rendering. Several sets of phonetic models were constructed, ranging from a maxi-

Viseme Recognition Accuracy on TIMIT				
# of Phone	# of Viseme Classes			
Models	42	25	15	9
42	62.9%	66.7%	72.3%	74.0%
25	N/A	63.7%	69.3%	72.3%
13	N/A	N/A	69.3%	71.0%

Table 1: Viseme recognition accuracy as a function of the number of phonetic models created. The models were created using a maximum of 42 acoustic-phonetic units and a minimum of fifteen (broad class) units. Phonetic recognition was performed for each set of models, and the resulting phones were then mapped to a set of visemes. The greater the number of visemes used, the greater the number of facial gestures available.

mum of 42 models to a minimum of fifteen. The accuracy for each set of phonetic models and viseme classes is shown in Table 1.

3.3. Discussion

The system we currently deploy uses 42 phonetic models. The recognized unit labels are mapped to a set of 25 visemes. Table 1 indicates that such a system has an error rate of 33.3% which means that the resulting rendering should contain an incorrect mouth posture for one of every three sounds. In practice however, we find that even in noisy conditions the visual rendering appears highly accurate and compelling. This would seem to indicate that misrecognition of visemes which are visually "close" does not have a significant impact on the visual effect. As can be seen in the table, the accuracy does in fact go up as the number of viseme classes (and hence confusions) are reduced. However, this accuracy increase does not appear to tell the entire story.

Examination of the data appears to indicate that the majority of insertions and deletions do not have a significant adverse impact on the visual output. This appears to be because many deletions occur when two acoustically similar phonetic units are hypothesized as a single unit by the recognizer. If the two acoustically similar units also have similar mouth postures than the deletion may not be noticeable. Likewise, if a single acoustic unit is hypothesized to be two units which have similar mouth postures, the impact on the animated rendering would again be minimal. The implication is that only substitution errors, between visually distinct classes, significantly degrade the quality of the visual output. The substitution rate using 25 visemes is approximately 12% for both the 42 and 25 phonetic model sets. We feel that this error rate more closely reflects the apparent error rate of the system.

An additional point is that since only the viseme content is critical, the application works well in non-English languages. We have found that in practice, the quality of the rendering remains high for languages such as German, French, and Japanese. It appears that even for acoustic-phonetic units, which were not available in the training corpus (which was American English), the correct viseme is "close" to the viseme produced by mapping the phonetic output of the recognizer.

4. Prototypes and Future Work

Several prototypes based on these technologies have been developed to explore the use of synchronized facial animation in different domains including: Internet chat, animation, education, and entertainment. Each implementation is based on combining elements of three technologies:

1. Audio data acquisition and transmission: The DIGITAL Voice Plugin [10] is used to acquire audio and transmit it in real time, or to bundle it as a file.
2. Phonetic recognition: This is performed either on a server or within a plugin on the client.
3. Facial rendering: This is accomplished with the FaceMe! player. Executable, plugin, and ActiveX versions of the player have been created.

This separation of technical components permits us to combine any face with any voice.

An asynchronous version of the combined system has been implemented which permits audio voice notes with facial gesture annotations to be sent via email to a recipient. Voice messages are recorded using the DIGITAL Voice Plugin [10] embedded in a Web page. Instead of sending the message directly to the recipient, the message is posted to a Web server. The role of this intermediate step is to process the audio file to generate an annotated audio file. A CGI bin script gets the message, uses the phonetic recognizer module to generate visemes and their time alignments, and forwards them to the original recipient of the message. The email messages are encoded as multi-part MIME messages. Upon reception, the message is played using the FaceMe player. This approach allows for the separation of the audio part from its annotation. Future implementations will include additional information such as winking, frowning, and other facial gestures.

The synchronous version (see Figure 3) of the system allows us to perform audio chat with facial animation. In this configuration, audio is processed on the sender side in a streaming fashion. The phonetic recognition produces the annotations "on the fly" in real time. Both the audio and the annotations are then sent through a socket connection. Note that if desired, it is possible to compress the audio at this point without impacting recognition accuracy. The recipient decompresses the audio (if necessary) and plays it along with the facial gestures. The total latency of the system is driven by buffering requirements and averages under two seconds.

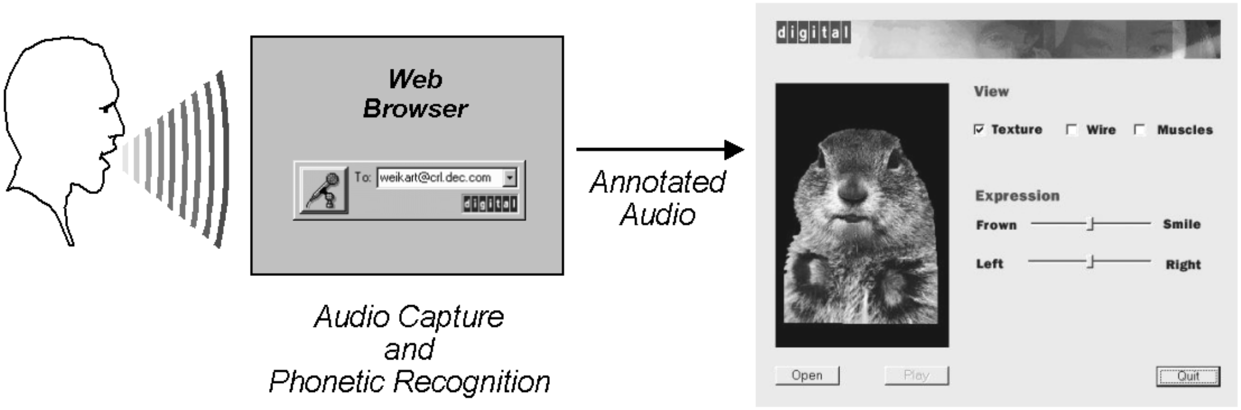

Figure 3: A synchronous architecture for Face2Face

All prototypes have been developed under Win32; the phonetic recognition engine itself runs on either Win32 or Alpha/Unix platforms. Phonetic recognition is performed real time in streaming mode on an Intel PentiumPro based machine. Average latency of the recognizer without adversely impacting performance is typically 0.3 seconds. The FaceMe! player operates in real-time on a 486/75 machine.

More synchronous applications include transforming FaceMe! into a regular audio file player, and full duplex Internet chat. We are working to improve overall performance of the system and implement new solutions to fully take advantage of the phonetic recognition/FaceMe! system capabilities.

5. REFERENCES

1. Goldenthal, W., "Statistical Trajectory Models for Phonetic Recognition," M.I.T. Ph.D. Thesis, September, 1994.
2. Waters, K. and Levergood, T., "DECface:A System for Synthetic Face Applications," *Journal of Multimedia Tools and Applications*, Kluwer Academic Publishers, Vol. 1, pp. 349–366, 1995.
3. Lewis, J., "Automated Lip-sync: Background and Techniques," *The Journal of Visualization and Computer Animation*, Vol. 2, pp. 118–122, 1991.
4. Morishima, S. and Harashima, H., "A Media Conversion from Speech to Facial Image for Intelligent Man-Machine Interface," *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 4, May 1991
5. Lavagetto, F. and Lavagetto, P., "A New Algorithm for Visual Synthesis of Speech," *Proceedings of Eurospeech '95*, pp. 303–306, Madrid, September, 1995.
6. Parke, F. and Waters, K., A.K. Peters, "Computer Facial Animation," pp. 285-295, 1996.
7. Eberman, B., and Goldenthal, W., "Time-Based Clustering for Phonetic Segmentation," *Proceedings of ICSLP '96*, pp. 1225–1228, Phil., PA, October, 1996.
8. Lee, Kai-Fu, and Hon, H. W., "Speaker-independent Phone Recognition using Hidden Markov Models," *IEEE Trans. ASSP*, Vol. 37, No. 11, pp. 1641-1648, November, 1989.
9. Lamel, L., Kassel, R., and Seneff, S. "Speech database development: design and analysis of the acoustic-phonetic corpus," *Proceedings DARPA Speech Recognition Workshop*, Report No. SAIC-86/1546, pp. 100–109, February, 1986.
10. Goldenthal, W., Goddeau, D., and Weikart, C., "Deploying Speech Applications over the Web," *Proceedings Eurospeech '97*, Rhodes, Greece, 1997.