An Automatic Lip-Synchronization Algorithm for Synthetic Faces

Keith Waters and Tom Levergood

Digital Equipment Corporation, Cambridge Research Lab,
One Kendall Square, Cambridge, MA 02139, USA

Abstract

This paper addresses the problem of automatically synchronizing computer-generated faces with synthetic speech. The complete process provides a novel form of face-to-face communication and the ability to create a new range of talking *personable* synthetic characters. Based on plain ASCII text input, a synthetic speech segment is generated and synchronized in real-time to a graphical display of an articulating mouth and face. The key component of the algorithm is the run-time facility that adaptively synchronizes the graphical display of the face to the audio.

1 Introduction

From an early age we are sensitive to the bimodal nature of speech, using cues from both visual and auditory modalities for comprehension. The visual stimuli are so tightly integrated into our perception of speech that we commonly believe that only the hearing-impaired lip-read [17]. In fact, people with normal hearing use all available visual information that accompanies speech, especially if there is a degradation in the acoustical signal [18]. Fluent speech is also emphasized and punctuated by facial expressions, thereby increasing our desire to observe the face of the speaker.

Our goal is to use the expressive bandwidth of the human face in real-time synthetic facial models capable of interacting with and eliciting responses from the user. Although an ideal interface might eventually be a natural dialogue between humans and computers, we consider a subset of this larger goal: a technique to automatically synchronize mouth shapes to a real-time speech synthesizer.

A computer-generated face has distinct advantages over images of real people, primarily because it is possible to create and control precise, repeatable facial actions. These faces suggest some unique and novel scenarios for presenting information, particularly where two-way interaction can be enhanced by listening rather than reading information. Examples of this type of interaction can be found in walk-by kiosks, ATM tellers, office environments, and videophones of tomorrow.

Synthetic facial images also have potential in situations where information has to be presented in a controlled, unbiased fashion, such as interviewing. Furthermore, if the synthetic speech/face generator were combined with systems that perform basic facial analysis by tracking the focus of the user and analysing the user's speech, it would be possible to transform the computer from an inert box into a *personable computer* [5]. These man-machine interfaces will mimic the way we interact face-to-face.

2 Background

Some of the first images of animated speech were created by G. Demeny in 1892 with a device he called the Phonoscope [3]. The device mounted images of the lower face on a disk that rotated fast enough to exploit persistence of vision. Although the Phonoscope is nearly a hundred years old, it is the underlying process employed in animation today. Rather than using photographs, traditional animation relies on hand-drawn images of the lips [27]. A sound track is first recorded, then an exposure sheet is marked with timing and phonetic information. The animator then draws a tailor-made mouth shape corresponding to a precise frame time. As one might expect, the whole process is extremely labor intensive.

2.1 Facial Animation

The first attempts in computer-based facial animation involved key-framing, where two or more complete facial expressions are captured and the in between frames computed by interpolation [22]. The immense variety of facial expressions makes this approach extremely

Multimedia 94- 10/94 San Francisco, CA, USA

data intensive, and prompted the development of parametric models for facial animation. Parametric facial models [23] create expressions by specifying sets of parameter value sequences; for instance, by interpolating the parameters rather than direct key-framing. The parameters control facial features, such as the mouth opening, height, width, and protrusion. The limitations of *ad hoc* parameterized models prompted a movement towards models whose parameters are based on the anatomy of the face [25, 32, 29, 16]. Such models operate with parameters based on facial muscle structures. When anatomically based models incorporate facial action coding schemes as control procedures[4], it becomes relatively straightforward to synthesize a range of recognizable expressions.

2.2 Lip-synchronized Animation

To date, non-automated techniques are most commonly used to achieve lip-synchronization. This process involves recording the sound track from which a series of control files are manually created. These files specify jaw and lip positions with key timing information, so that when the graphics and audio are recorded at *exactly* the same time, the face appears to speak [1]. Parametric models harness fudical nodes that are moved in synchronization with timings in a script [23]. Likewise, anatomical models coordinate muscle contractions to synchronize with the timing information in the script file [32]. In essence these techniques are obvious extensions to traditional hand animation. Unfortunately, they have the same inherent problem: their lack of flexibility. If the audio is modified, even slightly, ambiguities in the synchronization are created, and the whole process has to be repeated.

Automatic lip synchronization can be tackled from two directions: synchronization to synthetic speech and synchronization to real speech. In the latter case, the objective is to use speech analysis to automatically identify lip shapes for a given speech sequence. In brief, the speech sequence is transferred into a representation in which the formant frequencies are emphasized and the pitch information largely removed. This representation is then parsed into words. While the latter task is difficult, the former acoustic pre-processing can be reasonably effective at driving phonetic scripts [34, 15].

Lip synchronization to a synthesizer is the converse problem, where all the governing speech parameters are known. Hill, Pearce, and Wyvill extended a rule-based synthesizer to incorporate parameters for a 3D facial model [10]. When the synthesizer scripts were created, facial parameters could also be modified. Once a speech sequence had been generated, it was recorded to the audio channel of a video tape. The facial model was then generated frame-by-frame and recorded in non-real-time to the video section of the tape. Consequently, when the sequence was played back, the face appeared to speak.

2.3 Speech Reading

Traditionally, lip-reading has been considered to be a completely visual process developed by the small number of people who are completely deaf. There are, however, three mechanisms employed in visual speech perception: auditory, visual, and audio-visual. Those with hearing impairment concern themselves with the audio-visual, placing emphasis on observing the context in which words are spoken, such as posture and facial expression. It is this wealth of information on audio-visual speech mechanisms that provides valuable pointers for the construction of a synthetic talking face [31].

Speech comprises a mixture of audio frequencies, and every speech sound belongs to one of the two main classes known as vowels and consonants. Vowels and consonants belong to basic linguistic units known as phonemes which can be mapped into visible mouth shapes known as visemes. In English, lip-reading is based on the observation of forty-five phonemes and associated visemes [31]. Each vowel has a distinctive mouth shape, and viseme groups such as {p,m,b} and {f,v} can be reliably observed like the vowels, although confusion among individual consonants within each viseme group is more common [17]. Despite the low threshold between understanding and misunderstanding, the discrete phonetics provide a useful abstraction because they group together speech sounds that have common acoustic or articulatory features. In this research we use phonemes and visemes as the basic units of visible articulatory mouth shapes.

2.4 Text-to-Speech Synthesis

Automatic text-to-speech synthesis describes the process of generating synthetic speech from arbitrary text input. In most cases this is achieved with a letter-to-sound component and a synthesizer component. For a complete review of automatic speech synthesis, see [12].

In general, a letter-to-sound system (LTS) accepts arbitrary ASCII text as input and produces a phonemic transcription as output. The LTS component uses a pronunciation lexicon and possibly a collection of letter-to-sound rules to convert text to phonemes with lexical stress markings. These letter-to-sound rule sets are used to predict the correct pronunciation when a dictionary match is not found.

Synthesizers typically accept the input phonemes from the letter-to-sound component to produce synthetic audible speech. Three classes of segmental synthesis-by-rule techniques have been identified by Klatt [12]. They are (1) formant-based rules programs, (2) articulation-based rule programs, and (3) concatenation systems. Each technique attempts to create natural

and intelligible synthetic speech from phonemes.

A formant synthesizer recreates the speech spectrum using a collection of rules and heuristics to control a digital filter model of the vocal tract. Klattalk and DECtalk [12] are examples of formant-based synthesizers. Concatenation systems, as the name suggests, synthesize speech by splicing together short segments of parameterized stored speech. For example, speech segments might be stored as sequences of LPC (linear predictive coding) parameters which are used to resynthesize speech.

Formant-based rule programs and concatenation systems are both capable of producing intelligible synthetic speech. However, concatenation systems tend to introduce artifacts into the speech as a result of discontinuities at the boundaries between the stored acoustic units. Furthermore, concatenation systems have to store an inventory of sound units that may be on the order of 1M bytes per voice for a diphone system. Formant-based synthesizers require far less storage than concatenation systems, but they are restricted in the number and quality of the voices (speakers) produced. In particular, synthesizing a voice with a truly feminine quality has been found to be difficult [12]. For the purpose of synchronizing speech to synthetic faces, either the formant-based or concatenative approach could have been used. We use a formant synthesizer for the algorithm.

3 The Algorithm

The previous work described in 2.2 produces animation in a single frame mode that is subsequently recorded to video tape. In contrast, this research describes a fundamentally different approach to automatically synchronizing computer-generated faces with synthetic speech. The complete process provides a novel form of *real-time* face-to-face communication.

The unique feature of the algorithm is the ability to generate speech and graphics at real-time rates, where the audio and the graphics are tightly coupled to generate expressive synthetic facial characters (Figure 1). This demands a fundamentally different approach to traditional techniques. Furthermore, to compute synthetic faces and synchronize the audio in real-time requires a powerful computational resource, such as a Digital Alpha AXP workstation [33].

3.1 Overview

This section presents an algorithm to automate the process of synchronizing lip motion to a formant-based speech synthesizer. The key component of the algorithm is the run-time facility that adaptively synchronizes the graphical display of the face to the audio. The algorithm executes the following sequence of operations:

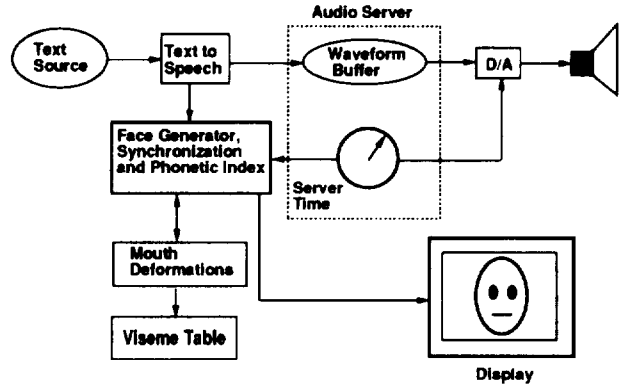

Figure 1: The synchronization model.

1. Input ASCII text
2. Create phonetic transcription from the text
3. Generate synthesized speech samples from the text
4. Query the audio server and determine the current phoneme from the speech playback
5. Compute the current mouth shape from nodal trajectories
6. Play synthesized speech samples and adaptively synchronize the graphical display

Steps 1-3 are an integral component of the algorithm and can be viewed as a pre-processing stage. Therefore both the phonetic transcription and the audio samples can be generated in advance and stored to disk if desired. Steps 4-6 are concerned with adaptive synchronization and are repeated until no more speech samples are left to be played.

The system requires a digital-to-analog converter (DAC) to produce the analog speech waveform. Associated with the audio playback device is a sample clock maintaining a representation of time that can be queried by an application program. We use the term *audio server* to describe the system software supporting the audio device. Since loss of synchronization is more noticeable in the audio domain, we used the audio server's clock for the global time base. The audio server's device clock is sampled during initialization, and thereafter the speech samples are played relative to this initial device time.

3.2 Text-to-Speech

The lip-synchronization algorithm uses the DECtalk algorithm that has many implementations; in our case, it is implemented as a software library. With sufficient

Alpha AXP, AXP, and DECtalk are trademarks of Digital Equipment Corporation.

computing power available, there is no special hardware or coprocessor needed to synthesize real-time speech. The DECtalk algorithm comprises three major components: (1) the letter-to-sound system, (2) the phonemic synthesizer, and the (3) vocal tract model.

The letter-to-sound system accepts arbitrary ASCII text as input and produces a phonemic transcription as output by using a pronunciation lexicon and a collection of letter-to-sound rules for English.

The phonemic synthesizer accepts the phonemic transcription output from the letter-to-sound system and produces parameter control records for the vocal tract model. This component applies intonation, duration, and stress rules to modify the phonemic representation based on phrase-level context. Once these rules have been applied, the phonemic synthesizer calculates parameters for the vocal tract model. The resulting phonetic sequence, including duration information, is also provided to the synchronization component.

The vocal tract model accepts the control records from the phonemic synthesizer and updates its internal state in order to produce the next frame of synthesized samples. The vocal tract model is a formant synthesizer based on the model described by Klatt [11]. The vocal tract model consists of voiced and unvoiced waveform generators, cascade and parallel resonators, and a summing stage. Frequency, bandwidth, and gain parameters in the control record are used to compute the filter coefficients for the resonators.

3.3 Time Base

Timing is the most critical component of the algorithm, since the audio/graphics synchronization can only be achieved when a common global time base exists. The initial time t_0 is recorded from the audio server as the first sample of speech is output. By re-sampling the audio server at time t_1, an exact correspondence to a phoneme or phoneme pair can be determined. The relative time t since the start of speech output is $t_1 - t_0$ and is used to index the phoneme/viseme Table 1, which is then used to calculate the current viseme mouth shape.

The audio server's sample rate is typically $8000Hz$ and ideally, the graphical display frame rate is $30Hz$. Therefore to avoid aliasing artifacts in the interpolation, all values are computed in audio server time.

3.4 Mouth Deformations

Once t, the current time relative to t_0, has been determined from the audio device, the displacement of the mouth nodes can be calculated. Each viseme mouth node is defined with position $\mathbf{x}_i(t) = [x(t), y(t), z(t)]'$, where $i = 1, \ldots, n$ are sequences of nodes defining the geometry and topology of the mouth. To permit a complete mouth shape interpolation, the topology must remain fixed and the nodes in each prototype mouth shape must be in correspondence. An intermediate interpolation position $\mathbf{x}(s)$ can be calculated between viseme nodes $\mathbf{x}^0$ and $\mathbf{x}^1$ by:

$$\mathbf{x}(s) = [u\mathbf{x}_0^0 + s\mathbf{x}_0^1, u\mathbf{x}_1^0 + s\mathbf{x}_1^1, \ldots, u\mathbf{x}_n^0 + s\mathbf{x}_n^1] \quad (1)$$

where $u = 1 - s$.

The parameter s is usually described by a linear or non-linear transformation of t where $0 \leq s \leq 1$. However, motions based on linear interpolation fail to exhibit the inertial motion characteristics of acceleration and deceleration. A closer interpolated approximation to acceleration and deceleration uses a cosine function to ease in and out of the motion:

$$s' = s * (1 - \cos(\pi * (s_0 - s)))/2 \quad (2)$$

The cosine interpolant is an efficient solution and provides acceptable results. However, during fluent speech, mouth shape rarely converges to discrete viseme targets due to the short interval between positions and the physical properties of the mouth. To emulate fluent speech we need to calculate co-articulated visemes.

Piecewise linear interpolation can be used to smooth through node trajectories, and various splines can provide approximations to node positions [6]. The addition of parameters (such as tension, continuity, and bias) to splines begins to emulate physical motion trajectories [13]. However, the most significant flaw of splines for animation is that they were originally developed as a means for defining static geometry and not motion trajectories.

A dynamic system of nodal displacements, based on the physical behavior of the mouth, can be developed if the initial configurations of the nodes $\mathbf{x}_i(t)$ are specified with positions, masses, and velocities $\mathbf{x}_i(t) = [m_i, \mathbf{v}_i(t); i = 1, \ldots, n]$. Once the geometric configuration has been specified, Newtonian physics can be applied:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i \quad (3)$$

$$m\frac{d\mathbf{v}_i}{dt} = \mathbf{f}_i - \gamma\mathbf{v}_i \quad (4)$$

Because we know the state positions $\mathbf{x}^0$ and $\mathbf{x}^1$, we are in fact calculating the trajectory along the vector $\vec{\mathbf{x}^0\mathbf{x}^1}$. To resolve the forces that are applied to nodes, it is assumed that $\mathbf{f}_i = 0$ when $\mathbf{x} = \mathbf{x}^1$. Forces can then be applied to the nodes, where γ is a velocity-dependent damping coefficient. It should be noted that forces are not generated from node neighbors as in a mesh, but rather from target node to target node. The mouth shape deformations use a Hookean elastic force based

on the separation of the nodes, thereby providing an approximation to elastic behavior of facial tissue:

$$\mathbf{f}_i = s_p * \mathbf{r}_k \tag{5}$$

where the vector separation of the nodes is $\mathbf{r}_k = \mathbf{x}^1 - \mathbf{x}$.

The equations of motion can then be integrated using the projected time Δt, derived from the audio server time, to calculate the new velocities and node positions:

$$\mathbf{v}_i = \mathbf{v}_i^o + \frac{\mathbf{f}_i}{m_i}\Delta t \tag{6}$$

$$\mathbf{x}_i = \mathbf{x}_i^o + \mathbf{v}_i \Delta t \tag{7}$$

Equation(7) uses the previous velocity $\mathbf{v}_i^o$ and position $\mathbf{x}_i^o$ to update the new nodal positions. More rigorous numerical integration techniques such as Runge-Kutta [26] could be used to improve numerical stability and convergence, but this would be more complex to implement and increase the computation time.

The dynamic equations of motion have the desirable attribute of approximating the node positions rather than peaking at the viseme mouth shape. In addition, the dynamic system adapts itself as the rate of speech increases, thus reducing the lip displacements as it tries to accommodate the new position. This behavior is characteristic of real lip motion.

3.5 Synchronization

Once a time base has been established, the synchronization of the audio and graphics is achieved as follows. A small number of samples of the sequence are played, returning the current server time. The relative animation time is then computed from the current server time and used to calculate the current mouth deformation. Once the mouth deformation has been calculated, the other manipulations, such as eye blinking, can take place with reference to the relative animation time. The whole face is then updated, rendered, and displayed on the screen.

4 The Face Model

Topologies for facial synthesis are typically created from explicit 3D polygons [24]. For simplicity we construct a simple 2D representation of the full frontal view. This model consists of 200 polygons of which 50 represent the mouth and an additional 20 represent the teeth (Figure 2(a)). The jaw nodes are moved vertically as a function of displacement of the corners of the mouth [7]. The lower teeth are displaced along with the lower jaw. To add a level of dynamic realism, the eyelids are animated.

While the wire frame provides a suitable model to observe the motion characteristics of the face, the visual representation is clearly unrealistic. The polygons of the face can be shaded using Gouraud or Phong shading,

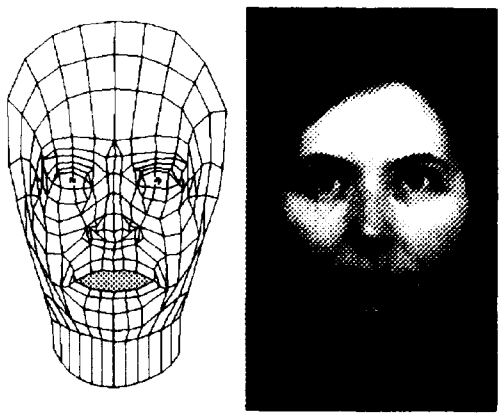

Figure 2: (a) Polygonal representation and (b) the resulting texture mapped image of the face.

but this often results in faces that look plastic. Texture-mapping of reflectance data acquired from photographs of faces or from laser digitizers provides a valuable technique for further enhancing the realism of facial models [9, 35, 2]. Even when the geometry is coarse, striking images can be generated [19]. Therefore we use a software incremental scanline texture mapping technique to achieve realistic faces (Figure 2(b)).

Incremental texture mapping is based on the scanline Gouraud shading algorithm [8]. Instead of interpolating intensity values at polygon vertices, it interpolates texture coordinates. Computing (u, v) texture coordinates for a polygon provides fast mapping into texture space. A color value can then be extracted and applied to the current scanline in screen space [33].

5 Example

Figure 3 is a time displacement graph illustrating three different computed trajectories for the sentence: *"First, a few practice questions."* spoken at 165 words-per-minute in a female voice. Trajectory (a) is the cosine displacement that peaks at the viseme mouth shapes. Trajectories (b) and (c) are two dynamic trajectories computed from equation(4). The two trajectories (b) and (c) are controlled by two variables γ and s_p representing the velocity damping coefficient and the spring constant respectively. The mass m remains constant between the two examples at $m = 0.25$. For (b) $sp = 0.650$ and $\gamma = 0.500$, and for (c) $sp = 0.150$ and $\gamma = 0.850$.

The physical model of facial tissue motion provides acceleration and deceleration, emulating inertial characteristics as the mouth changes shape. This is most evident during rapid speech, where the mouth does not

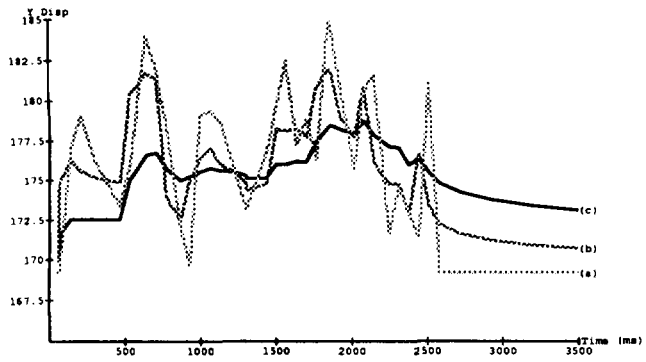

Figure 3: The mid top lip vertical displacement trajectories of the sample sentence. (a) is the cosine activity peaking at each phonetic mouth shape. (b) is the physical model with a small damping coefficient and a large spring constant, while (c) is the physical model with a larger damping coefficient and lower spring constant.

make complete mouth shapes, but instead produces a blend between shapes under muscular forces. The final result is a more natural looking mouth motion.

6 Implementation

To implement a real-time version of the algorithm, we incorporated several existing hardware and software components on an Alpha AXP workstation running the DEC OSF/1 Alpha V2.0 operating system.

The synthesised speech output by the DECtalk vocal tract model can be played on any D/A converter hardware supporting 16 bit linear PCM (pulse code modulation) or 8 bit μ-law (log-PCM) encoded samples operating at an 8 KHz sampling rate.

The AF audio server and client library [14] were used to interface to the audio hardware and provide an applications programming interface for the audio portion of the algorithm. The client library provides the application programming interface for accessing the audio server and audio hardware.

The algorithm uses the X Window System to display the rendered facial images. A Tk [20, 21] widget-based user interface facilitates the interactive use of the algorithm [33]. For the speech synthesizer, arbitrary text can be created in the text widget and spoken using one of eight internal voices. In addition, the user can specify the number of words per minute, as well as the comma and period pause durations. For the face synthesizer, sliders are associated with six linear muscles [32], allowing simple facial expressions to be created. Finally several graphical display characteristics can be selected, including texture or wireframe, a physical simulation, and muscles.

An earlier implementation of this algorithm on a DEC Alpha AXP 3000/500 running OSF/1 V1.2 using a texture mapped image of 320x512 pixels displays in excess of 15 frames per second [33]. At these frame rates, convincing lip-synchronization can be created.

7 Discussion

Imagining a character with an expressive voice and face that interacts with humans is not difficult. While this is easy to envision, significant technical issues have to be addressed both from the audio and graphics domains. For example, how do you create a character capable of expressing anger or distress, telling a joke, or providing other emotional states? In the context of speech synthesis we expect the naturalness of the synthetic speech, rather than the intelligibility measure, will be the most compelling factor in the presentation of a synthetic character's voice.

From a graphics perspective the most perplexing challenge is our critical examination of the face model as it approaches reality. Shortcomings in bland, exaggerated plastic faces are readily dismissed; perhaps our expectations are too low for such caricatures. Whatever the reason, we become much less tolerant when faces of real people are manipulated, since we know precisely what *they* look like and how *they* articulate. Therefore, if images of real people are to be used, the articulations have to mimic reality extremely closely. The alternative is to exploit the nuances of caricatures and create a new synthetic character, in much the same way that cartoon animators do today. In this context we believe that getting the behavior *correct* has a more significant impact on the perception of the synthetic character than accuracy in the facial animation and display. Some of these attributes are being borne out in experimental social interface agents [28] and using human faces in interfaces [30].

While lip synchronization is crucial to building articulate personable characters, there are other important characteristics, such as body posture and facial expression, that convey information. Combining facial expression with speech is beyond the scope of this paper and has been deliberately omitted. However, our research has demonstrated that basic expressions can dramatically enhance the realism of the talking face; even simple eye blinks can bring the face to life.

Finally, it is important to remember that a phonetic transcription is an *interpretation* imposed on a continuously varying acoustic signal. Therefore, visemes are extensions to phonemes. The algorithm presented in this paper can be extended to diphones and co-articulated words, but the synchronization can ultimately be only as good as the text-to-speech synthesizer.

8 Conclusion

We have demonstrated an algorithm for automatically synchronizing lip motion to a speech synthesizer in real-time. The flexibility of the algorithm is derived

from the ability to completely synthesize the face and speech explicitly from a stream of ASCII text. It is this ability to interpret unstructured text and generate real-time facial articulations that makes the algorithm truly unique. Arbitrary text, derived from sources such as database queries, expert systems, mail files, and editors can be presented via the face interface; the humanized face provides a personable character capable of engaging the user in simple verbal interactions.

Finally, we believe that completely synthetic facial models, coupled to synthetic speech generators that operate in *real-time*, will provide a unique form of interaction with the computer.

References

[1] P. Bergeron and P. Lachapelle. Techniques for animating characters. In *Advanced Computer Animation*, volume 2 of *SIGGRAPH '85 Tutorials*, pages 61–79. ACM, 1985.

[2] H. Choi, S.C. Harashima and T. Takebe. Analysis and synthesis of facial expressions in knowledge-based coding of facial image sequences. In *International Conference on Acoustics Speech and Signal Processing*, pages 2737–2740, 1991.

[3] J. Destandes. *Histoire Comparee du Cinema*, volume 1. Castarman, Paris, 1966.

[4] P. Ekman and W.V. Friesen. *Manual for the Facial Action Coding System*. Consulting Psychologists Press, Palo Alto CA, 1977.

[5] I. Carlbom et al. Modeling and analysis of empirical data in collaborative environments. *Communications of the ACM (CACM)*, 35(6):74–84, June 1992.

[6] G Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press Inc., New York, 1990.

[7] V. Fromkin. Lip positions in American English vowels. *Language and Speech*, 7(3):215–225, 1964.

[8] H. Gouraud. Continuous shading of curved surfaces. *IEEE Trans on Computers*, 20(6), 1971.

[9] P. Heckbert. Survery of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, 1986.

[10] D.R. Hill, A. Pearce, and B. Wyvill. Animating speech: An automated aproach using speech synthesis by rules. *The Visual Computer*, 3:277–289, 1988.

[11] Dennis H. Klatt. Software for a cascade/parallel formant synthesizer. *J. Acoust. Soc. Am.*, 67(3):971–995, 1980.

[12] Dennis H. Klatt. Review of text-to-speech conversion for English. *J. Acoust. Soc. Am.*, 82(3):737–793, 1987.

[13] H.D. Kochanek and R.H. Bartels. Interpolating splines with local tension, continuity and bias control. *Computer Graphics*, 3(18):33–41, 1982.

[14] Thomas M. Levergood, Andrew C. Payne, James Gettys, G. Winfield Treese, and Lawrence C. Stewart. AudioFile: A network-transparent system for distributed audio applications. In *Proceedings of the USENIX Summer Conference*, June 1993.

[15] J.P. Lewis and F.I. Parke. Automatic lip-synch and speech synthesis for character animation. In *CHI+CG '87*, pages 143–147, Toronto, 1987.

[16] N. Magnenat-Thalmann, N.E. Primeau, and D. Thalmann. Abstract muscle actions procedures for human face animation. *Visual Computer*, 3(5):290–297, 1988.

[17] M. McGrath. *An Examination of Cues for Visual and Audio-Visual Speech Perception using Natural and Computer Generated Faces*. PhD thesis, University of Nottingham, England, November 1985.

[18] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:126–130, 1986.

[19] M. Oka, K. Tsutsui, A. Ohba, Y. Kurauchi, and T. Tago. Real-time manipulation of texture-mapped surfaces. *Computer Graphics*, 21(4):181–188, 1987.

[20] John K. Ousterhout. Tcl: An embeddable command language. In *Proceedings of the USENIX Winter Conference*, January 1990.

[21] John K. Ousterhout. An X11 toolkit based on the Tcl language. In *Proceedings of the USENIX Winter Conference*, January 1991.

[22] F.I. Parke. Computer generated animation of faces. Master's thesis, University of Utah, Salt Lake City, June 1972. UTEC-CSc-72-120.

[23] F.I. Parke. Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, 1982.

[24] F.I. Parke. State of the art in facial animation. *ACM SIGGRAPH Course Notes*, 26, 1990.

[25] S.M. Platt and N.I. Badler. Animating facial expressions. *Computer Graphics*, 15(3):245–252, 1981.

[26] W. Press, B. Flanney, S. Teukolsky, and W. Verttering. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1986.

[27] F. Thomas and O. Johnson. *Disney Animation: the Illusion of Life*. Abbeville Press, New York, 1981.

[28] K. Thorisson. Dialogue control in social interface agents. *INTERCHI (Ajunct Proceedings)*, pages 139–140, April 1993.

[29] C.T. Waite. The facial action control editor, face: A parametric facial expression editor for computer generated animation. Master's thesis, MIT, Feb 1989.

[30] L. Walker, J. Sproull and R. Subramani. Using a human face in an interface. *ACM CHI*, pages 85–91, April 1994.

[31] E.F. Walther. *Lipreading*. Nelson-Hall Inc, Chicago, 1982.

[32] K. Waters. A muscle model for animating three-dimensional facial expressions. *Computer Graphics (SIGGRAPH'87)*, 21(4):17–24, July 1987.

[33] K. Waters and T. Levergood. DECface: An automatic lip-synchronization algorithm for synthetic faces. Technical Report 93/4, Digital Equipment Corp, Cambridge Research Laboratory, Sept 1993.

[34] P. Weil. About face. Master's thesis, MIT, Aug 1982.

[35] L. Williams. Performace driven facial animation. *Computer Graphics*, 24(4):235–242, 1990.

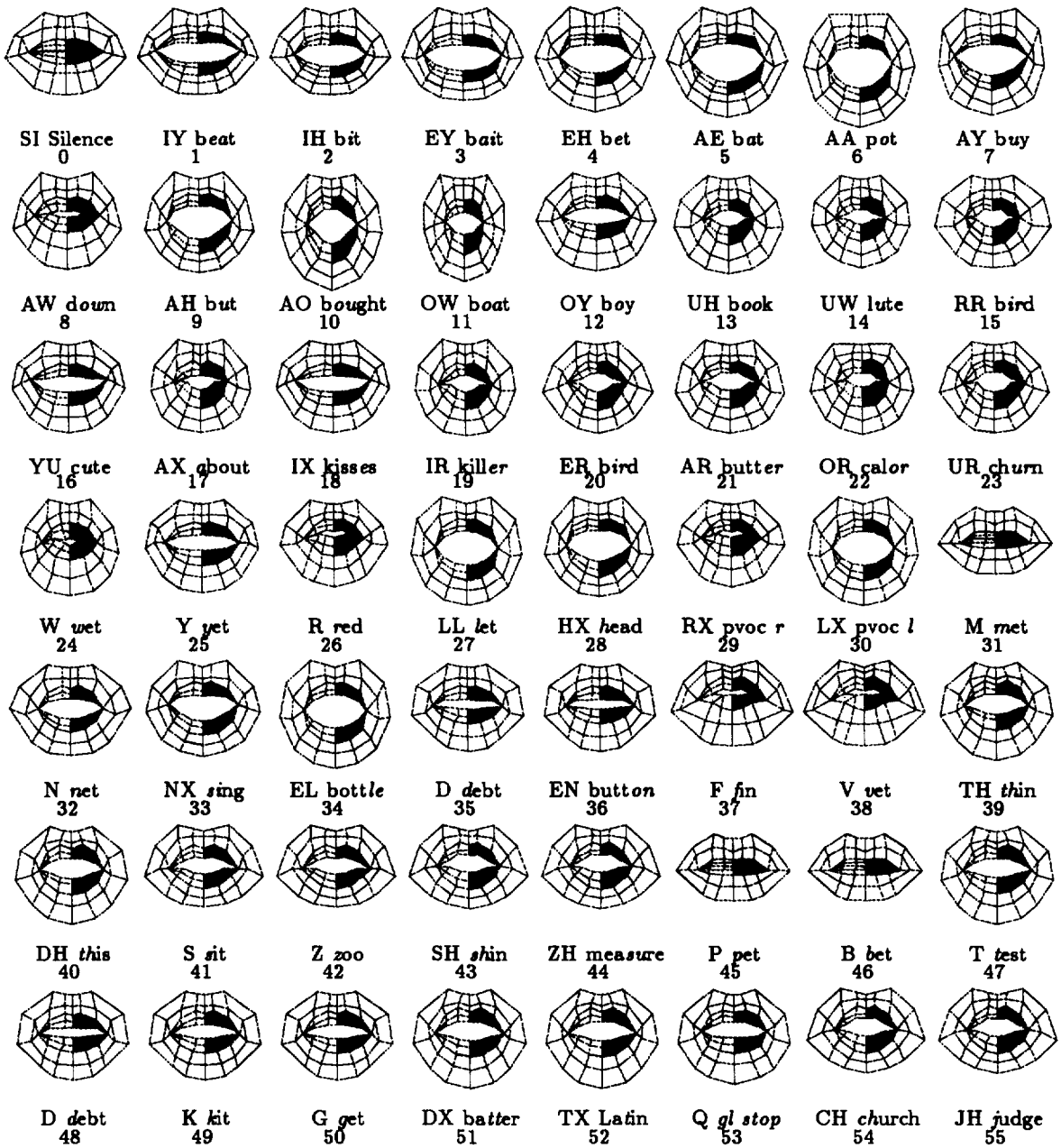

Table 1: The table of viseme mouth shapes with associated phonemic characters and examples. This table was derived from an observation of a persons lips from a live NTSC video source. Each mouth posture was subsequently digitized and mapped to the phonemic characters provided by DECtalk.