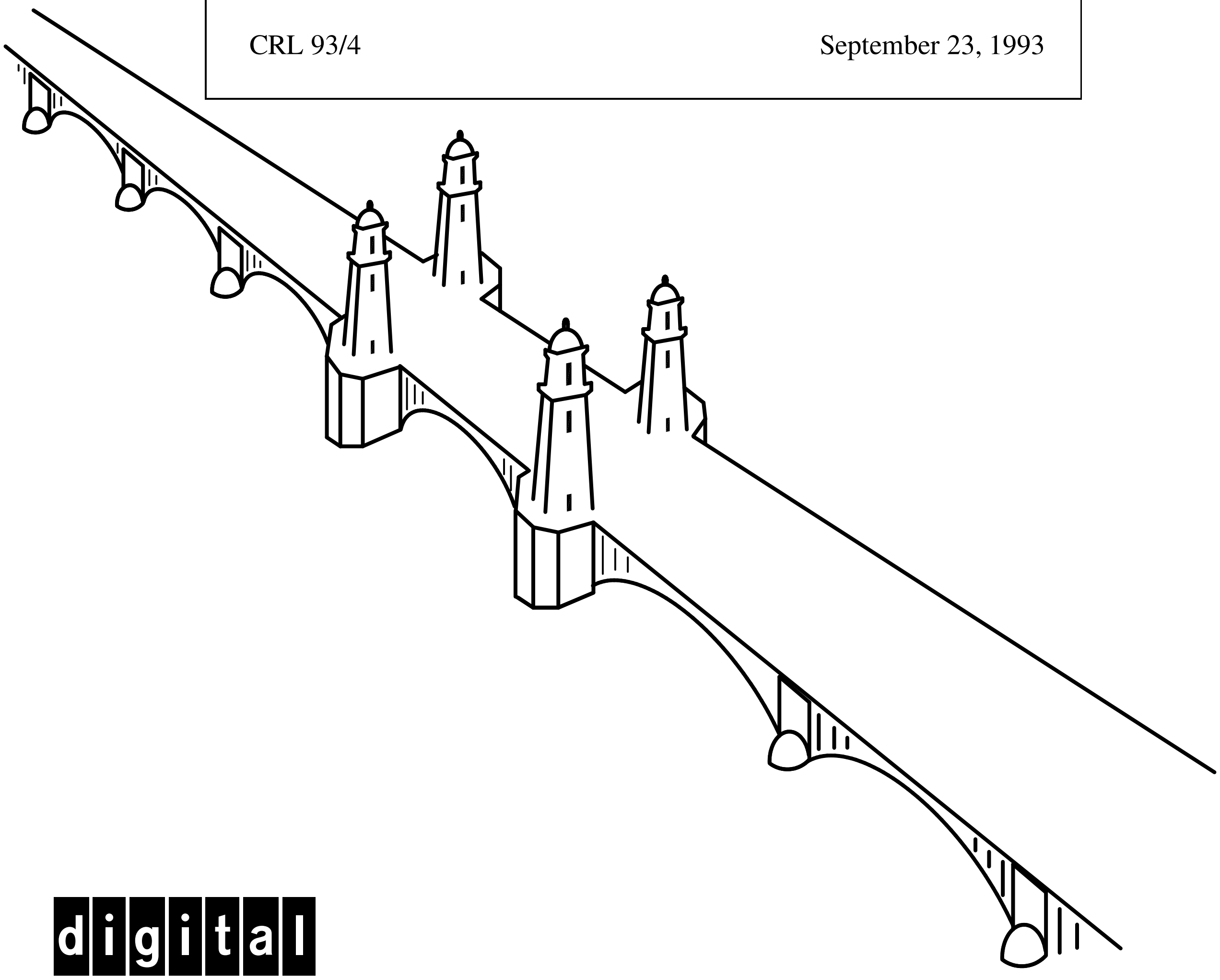
DECface: An Automatic Lip-Synchronization Algorithm for Synthetic Faces

Keith Waters and Thomas M. Levergood

Digital Equipment Corporation
Cambridge Research Lab

CRL 93/4 September 23, 1993



digital

CAMBRIDGE RESEARCH LABORATORY
Technical Report Series

Digital Equipment Corporation has four research facilities: the Systems Research Center and the Western Research Laboratory, both in Palo Alto, California; the Paris Research Laboratory, in Paris; and the Cambridge Research Laboratory, in Cambridge, Massachusetts.

The Cambridge laboratory became operational in 1988 and is located at One Kendall Square, near MIT. CRL engages in computing research to extend the state of the computing art in areas likely to be important to Digital and its customers in future years. CRL's main focus is applications technology; that is, the creation of knowledge and tools useful for the preparation of important classes of applications.

CRL Technical Reports can be ordered by electronic mail. To receive instructions, send a message to one of the following addresses, with the word **help** in the Subject line:

On Digital's EASYnet:	CRL::TECHREPORTS
On the Internet:	techreports@crl.dec.com

This work may not be copied or reproduced for any commercial purpose. Permission to copy without payment is granted for non-profit educational and research purposes provided all such copies include a notice that such copying is by permission of the Cambridge Research Lab of Digital Equipment Corporation, an acknowledgment of the authors to the work, and all applicable portions of the copyright notice.

The Digital logo is a trademark of Digital Equipment Corporation.

digital™

Cambridge Research Laboratory
One Kendall Square
Cambridge, Massachusetts 02139

DECface: An Automatic Lip-Synchronization Algorithm for Synthetic Faces

Keith Waters and Thomas M. Levergood

Digital Equipment Corporation
Cambridge Research Lab

CRL 93/4 September 23, 1993

Abstract

This report addresses the problem of automatically synchronizing computer generated faces with synthetic speech. The complete process is called DECface which 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 DECface is the run-time facility that adaptively synchronizes the graphical display of the face to the audio.

Keywords: face-to-face, key-framing, anatomically-based, text-to-speech, viseme, phonetics

©Digital Equipment Corporation 1993. All rights reserved.

UNIX is a trademark of Unix Systems Laboratories.
Apple is a registered tradmark of Apple Computer, Inc.
The following are trademarks of Digital Equipment Corporation: Alpha AXP, DEC, DECaudio, DECstation, DECtalk, TURBOchannel, ULTRIX, XMedia, and the DIGITAL logo.

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 [McG85]. In fact, people with normal hearing use all available visual information that accompanies speech, especially if there is a degradation in the acoustical signal [MM86]. 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. In the near future we are going to see man-machine interfaces that mimic the way we interact face-to-face. A few years ago Apple Computer, Inc. produced a futuristic video called the *The Knowledge Navigator*, popularizing and advertising the notion of active agents. In the video, Phil, an active agent, performs a variety of tasks at the request of the user. In reality, no such system or environment exists. However it is worth noting that Phil, a head and shoulders image of a real actor, was the primary interface to the computer. 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 analyzing the user's speech, it would be possible to transform the computer from an inert box into a *personable computer* [CHK$^+$92].

2 Background and Previous Work

Some of the first images of animated speech were created by G. Demeny in 1892 with a device he called the Phonoscope [Des66]. 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 [TJ81]. 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.

The first attempts in computer-based facial animation involved key-framing, where two or more complete facial expressions are captured and in between frames computed by interpolation [Par72]. The immense variety of facial expressions makes this approach extremely data intensive, and prompted the development of parametric models for facial animation. Parametric facial models [Par74, Par82] 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 [Pla80, PB81, Wat87, Wai89, MPT88]. Such models operate with parameters based on facial muscle structures. When anatomically based models incorporate facial action coding schemes as control procedures[EF77], it becomes relatively straightforward to synthesize a range of recognizable expressions.

The geometric surface of the face is typically described as a collection of polygons and displayed using standard lighting models [Gou71, Pho76]. Texture mapping of reflectance data acquired from photographs of faces or from laser digitizers provide another valuable technique for further enhancing the realism of facial modeling and animation [NHS88, Wil90, CT91]. Even when the geometry is coarse, striking images can be generated [OTO$^+$87].

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. Key-framing requires a complete face posture for each key position [BL85]. Parametric models harness fudical nodes that are moved in synchronization with timings in a script [Par74]. Likewise, anatomical models coordinate muscle contractions to synchronize with the timing information in the script file [Wat87]. 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. This latter task is difficult, and the former acoustic pre-processing is reasonably effective for driving phonetic scripts [Wei82, LP87].

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 [HPW88]. 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.

2.1 Lip-reading and the Phonetics of Speech

In English lip-reading is based on the observation of forty-five phonemes and associated visemes [Wal82]. 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.

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. 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 [McG85]. 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. We use phonemes and visemes as the basic units of visible articulatory mouth shapes.

2.2 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 [Kla87].

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 [Kla87]. 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 [Kla87] and DECtalk [BMT83] 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. Olive's *Utter* system is an example of diphone synthesis in a concatenative system [Oli90].

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 concatenative 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 [Kla87]. 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 DECface.

2.3 DECface

The previous work described in Section 2 produces animation in a single frame mode that is subsequently recorded to video tape. In contrast to this previous work this report describes a fundamentally different approach to automatically synchronizing computer generated faces with synthetic speech. The complete process is called DECface which provides a novel form of *real-time* face-to-face communication.

The unique feature of DECface 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. 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 an Alpha AXP workstation.

3 The Algorithm

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. DECface executes the following sequence of operations:

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 synchronize the graphical display

Step 1, 2, and 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, 5, and 6 are concerned with adaptive synchronization and are repeated until no more speech samples are left to be played.

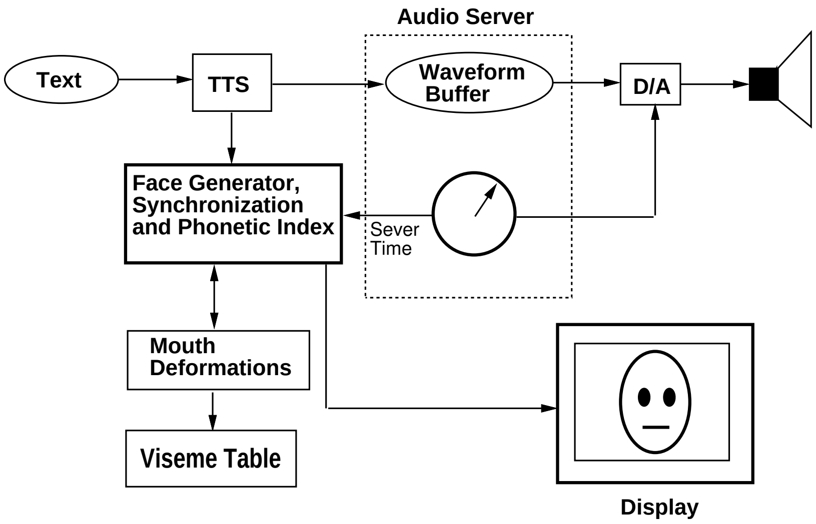


Figure 1: The synchronization model TTS = text-to-speech.

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.

3.1 Text-to-Speech

The DECface algorithm uses DECtalk. DECtalk is an algorithm that has many implementations; in our case, it is a software implementation. With sufficient computing power available, there is no special hardware or coprocessor needed by DECtalk to synthesize real-time speech. DECtalk comprises three major algorithms: (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. As part of this process, the input text may be reformatted to convert numbers and abbreviations into words and punctuation.

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 is also provided to the DECface 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 [Kla80]. 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.

3.2 Time Base

Timing is the most critical component of the DECface algorithm since the audio/graphics synchronization can only be achieved when a common global time base exists. The initial time q_0 is recorded from the audio server as the first sample of speech is output. By re-sampling the audio server at time q_1, an exact correspondence to a phoneme or phoneme pair can be determined. The relative time t since the start of speech output is $q_1 - q_0$ and is used to calculate the current viseme mouth shape.

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

3.3 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 $\mathrm{x}_i(t) = [x(t), y(t), z(t)]'$, 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 $\mathrm{x}(s)$ can be calculated between viseme nodes x^0 and x^1 by:

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

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:

$$s' = s * (1 - \cos(\pi * (s_0 - s)))/2 \tag{2}$$

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 [Far90]. The addition of parameters (such as tension, continuity, and bias) to splines begins to emulate physical motion trajectories [KB82]. 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 configuration of the nodes $\mathrm{x}_i(t)$ are specified with positions, masses, and velocities $\mathrm{x}_i(t) = [m_i, \mathrm{v}_i(t); i = 1, \ldots, n]$. Once the geometric configuration has been specified, Newtonian physics can be applied:

$$\frac{d\mathrm{x}_i}{dt} = \mathrm{v}_i \tag{3}$$

$$m\frac{d\mathrm{v}_i}{dt} = \mathbf{f}_i - \gamma \mathrm{v}_i \tag{4}$$

Because we know the state positions x^0 and x^1, we are in fact calculating the trajectory along the vector $\overrightarrow{\mathrm{x}^0\mathrm{x}^1}$. To resolve the forces that are applied to nodes, it is assumed that $\mathbf{f}_i = 0$ when $\mathrm{x} = \mathrm{x}^1$. 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 *t*, 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 positions $\mathbf{x}_i^o$ to update the new nodal positions. More rigorous numerical integration techniques such as Runge-Kutta [PFTV86] could be used to improve numerical stability and convergence, but this would be more complex to implement and increase the computation time.

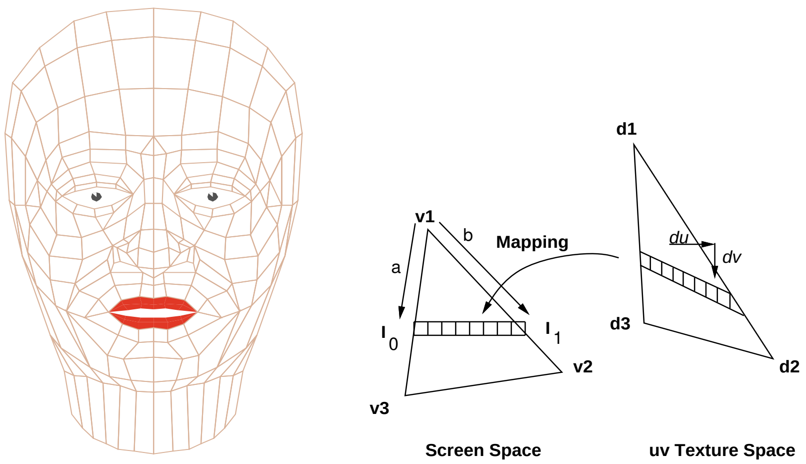
The dynamic equations of motion have the desirable attribute of approximating the node positions rather than peaking at the viseme mouth shape. In addition, the dynamic system adapts itself as the rate of speech increases, thus reducing the lip displacements as it tries to accommodate the new position.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

3.4 Synchronization

The synchronization of the audio and graphics is achieved as follows. The audio server is initialized, returning the start time for the sequence. A small number of samples of the sequence are then played, returning the duration in milliseconds and the current server time. The relative animation time is computed from the current server time and is used to calculate the current mouth deformation. Once the mouth deformation has been calculated, the other manipulations, such as eye blinking, take place with reference to the relative animation time. The 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 [Par90]. For simplicity we construct a simple 2D wire frame representation of the frontal view (Figure 2(a)). This model consists of 200 polygons of which 50 represent the mouth and an additional 20 represent the teeth. The jaw nodes are moved vertically as a function of displacement of the corners of the mouth [Fro64]. The lower teeth are displaced along with the lower jaw. To add a level of dynamic realism, the eyelids are animated.



(a) (b)

Figure 2: (a) Polygonal representation of the face. (b) Texture mapping.

4.1 Face Rendering

While the wire frame provides a suitable model to observe the motion characteristics of the face, the visual representation is clearly unrealistic. The face polygons can be shaded using Gouraud or Phong shading models, but this often results in faces that look plastic. However, texture mapping is a powerful technique that adds visual detail to synthetic images [Hec86]. In particular, texture mapping greatly enhances the realism of synthetic faces. We use an incremental scanline texture mapping technique to achieve realistic faces.

Incremental texture mapping is based on the scanline Gouraud shading algorithm [Gou71]. 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 (Figure 2(b)). For each step along the current scanline in screen space, du and dv are incremented by a constant amount in texture space. Effectively the increment has a slope of dv/du and can be used to rapidly index through texture space. For efficiency the (u, v) coordinate samples texture space and returns a (r, g, b) value for the current scanline (x, y).

_	**f**	**rr**	**s**	**t**	**ax**	**f**	**yu**	**p**	**r**	**ae**	**k**
58	109	147	90	51	77	109	186	77	70	141	70
t	**ix**	**s**	**k**	**w**	**eh**	**s**	**ch**	**ix**	**n**	**z**	**_**
58	77	90	58	64	128	77	115	102	83	90	563

Table 1: Phonemes and durations (milliseconds) for the sample sentence.

5 Example

The text *"First, a few practice questions."* is used to demonstrate the DECface algorithm in this section. The words are spoken at 165 words-per-minute in a female voice. These words will be referred to as the sample sentence. The synthesizer produces a sequence of phoneme/duration pairs (Table 1) for the sample sentence. The phonemes belong to the two-character arpabet used by DECtalk and have a direct correlation to visemes depicted in Table 3. To create Table 3, we used shapshots of a persons mouth while uttering CvC and VcV strings (Table 4).

Figure 3 is a time displacement graph illustrating three different computed trajectories, while Figure 4 illustrates every third frame from each of the three trajectories. 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 $\gamma = 0.850$. Figure 5 illustrates a sequence of frames of the complete texture mapped facial model speaking the sample sentence.

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

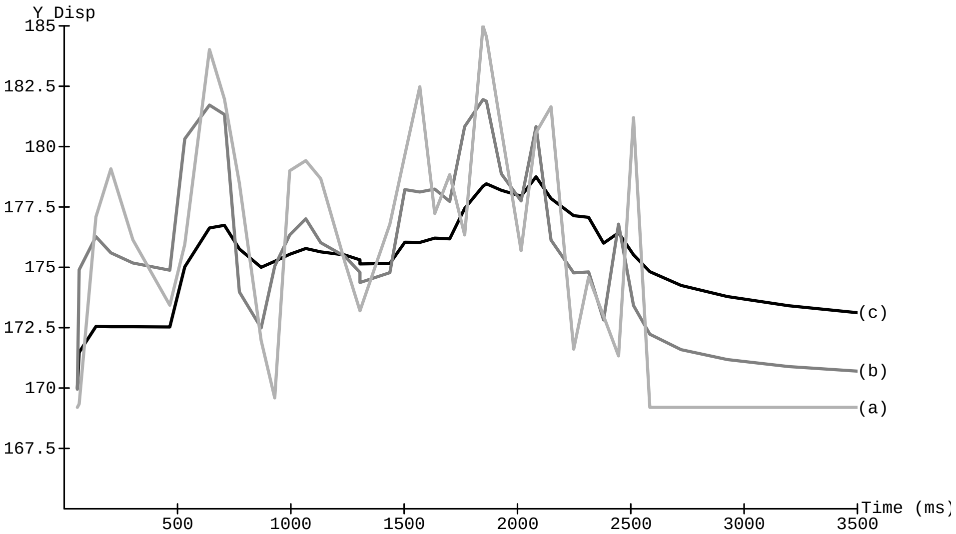


Figure 3: The mid top lip vertical displacement trajectories of the sample sentence. Trajectory (**a**) is the cosine activity peaking at each phonetic mouth shape. Trajectory (**b**) is the physical model with a small damping coefficient and a large spring constant, while trajectory (**c**) is the physical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

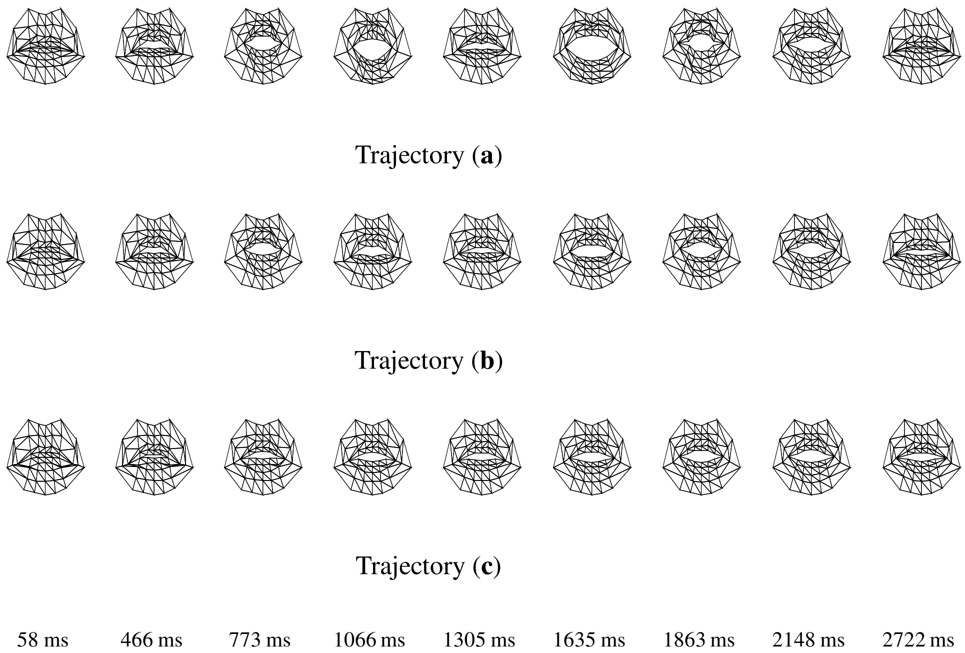


Figure 4: Every third viseme produced by the sample sentence for each trajectory.

6 Implementation

To implement a real-time version of DECface, we incorporated several existing hardware and software components on an Alpha AXP workstation running the DEC OSF/1 Alpha V1.2 operating system.

The synthesized 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. For example, the baseboard CODEC [1] on Alpha AXP workstations or the DECaudio module [Lev93] may be used. DECaudio is a TURBOchannel I/O peripheral containing highly flexible audio A/D and D/A converter capabilities.

The AudioFile audio server and client library [LPG$^+$93] were used to interface to the audio hardware and provide an applications programming interface for the audio portion of DECface. The client library provides the application programming interface for accessing the audio server and audio hardware.

DECface uses the X Window System to display the rendered facial images.

A Tk [Ous90, Ous91] widget-based user interface facilitates the interactive use of DECface (Figure 6). A variety of commands can be piped to DECface. For the speech synthesizer, arbitary 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 [Wat87] allowing simple facial expressions to be created. Finally several graphical display characteristics can be selected, including texture or wireframe, a physical simulation, muscles, and an SMP (Software Motion Pictures) clip generator.

7 Performance

DECface performance data for the DEC Alpha AXP 3000/500 workstation (150MHz clock)were collected to provide an overall performance indicator. Both the wireframe and the texture mapped versions were timed on images of 512x320 pixels. Table 7 illustrates frames rates for the wireframe and texture modes. The timing data includes the cost to generate the synthetic speech. The performances of the wireframe and the texture mapped models are both about 15 frames per second. At these frame rates, convincing lip-synchonization can be created. This is because the texture mapping code had been highly optimized, whereas little effort was spent in improving the performance of the wireframe model.

[1] Contraction of Coder and Decoder.

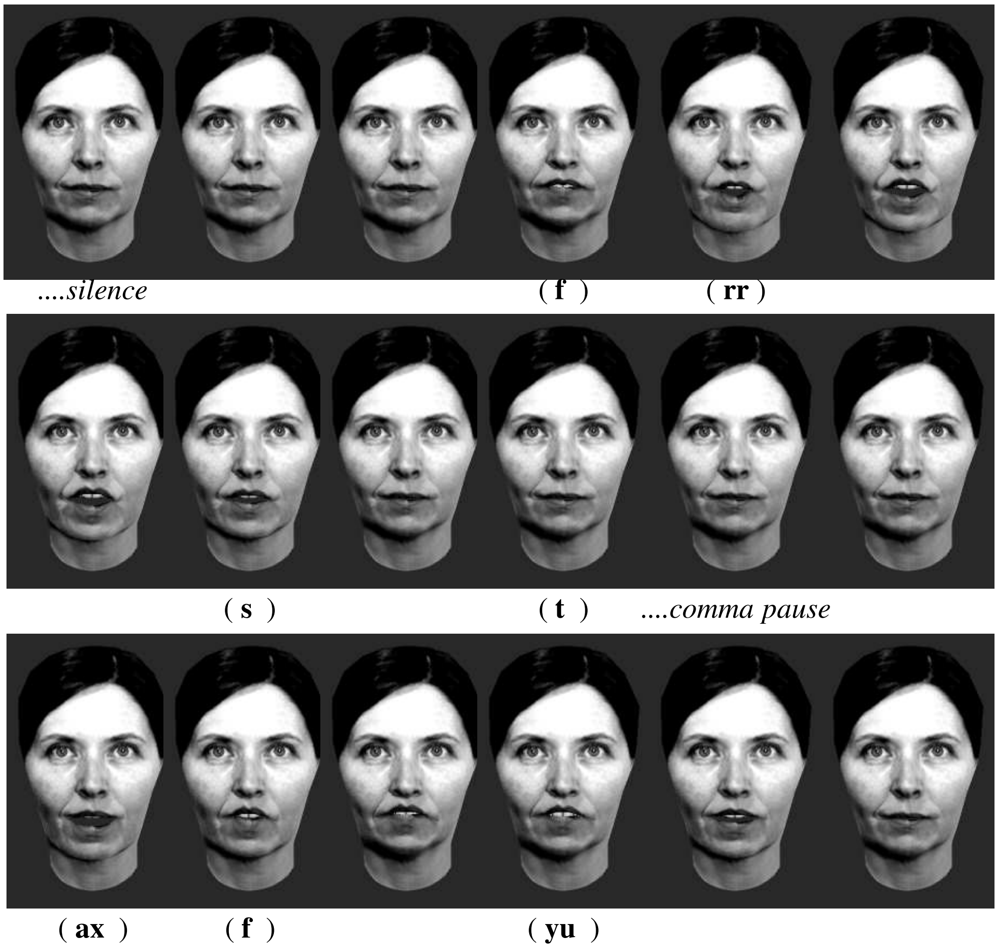


Figure 5: Frames 1 through 18 of the test sequence: *"First, a few.."* The phonemic characters are indicated below key visemes. To emphasize the mouth articulation the cosine trajectory was computed.

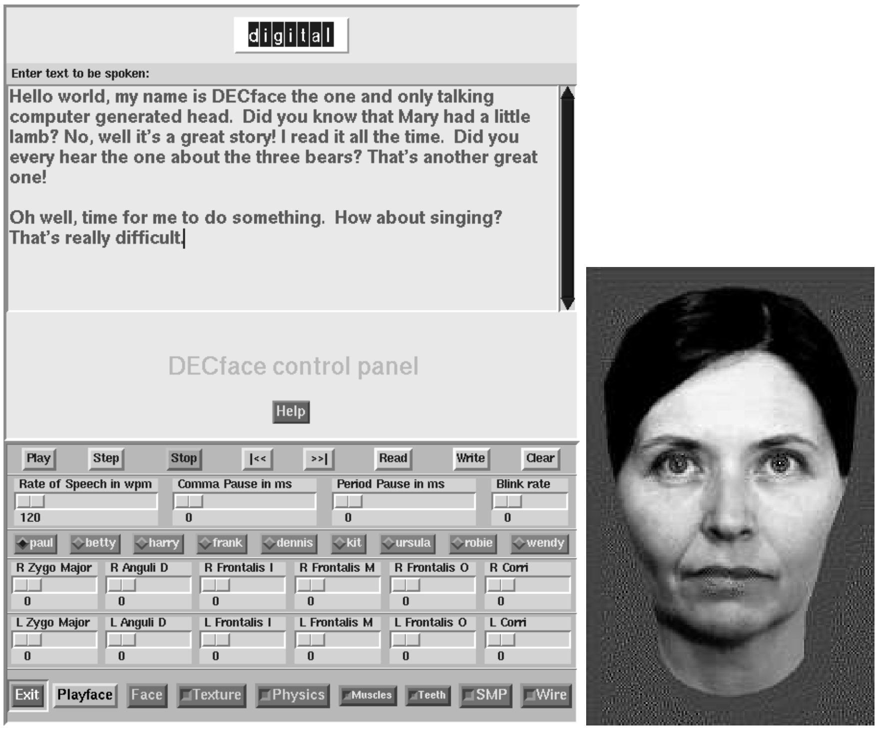


Figure 6: The Interface

Geometry	**Pixels**/*sec*	**Total frames**/*sec*
Wireframe	*(not computed)*	15.52
Wireframe + Physics	*(not computed)*	15.67
Texture mapped	3053792	14.82
Texture mapped + Physics	4213741	15.91

Table 2: Performance table for DECface on the DEC 3000/500 workstation.

8 Discussion

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. One could easily imagine a character with an expressive voice and face closely interacting with the user. While this is easy to envision, significant technical issues have to be addressed both from the graphics and audio domains. For example, how do you create a character capable of telling a joke, or of expressing anger, distress, or other emotional states?

Perhaps the most perplexing challenge from a graphics perspective 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.

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.

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 DECface algorithm can be extended to co-articulated words, but the viseme synchronization can ultimately be only as good as the text-to-speech synthesizer.

While DECface has been designed to operate on 2D images, extensions to 3D are straightforward. In fact, physically based face systems can be simply modified to incorporate DECface. We plan to incorporate DECface into a 3D facial model.

9 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 DECface truly unique. Arbitary text, derived from sources such as database queries, expert systems, mail files, and editors can be presented via DECface; the humanized face provides a personable character capable of engaging the user in simple verbal interactions.

The dynamic model presented in this paper provides trajectories that mimic the motion of real lips. The viseme table can be constructed with little effort and correlated to a specific speech synthesizer. In conjunction, the table of visemes and the dynamic model provide the necessary vehicle to develop various mouth shape trajectories. This is highly desirable because no two people speak exactly the same.

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

10 Acknowledgements

We would like to thank Richard Szeliski at CRL who re-wrote and optimized the original texture mapping code; without his assistance, we would not have achieved a real-time graphical performance. We would also like to thank Andy Payne for his tutorial on Tk on which the interface is based. Thanks also to Jan Walker for her constant suggestions for improvements. To Lee Sproull and Anne Dix for the use of their face and lip images. To Dick Beane who provided his editorial skills. Finally, we would like to acknowledge Dave Wecker and the Visualization Group at CRL for valuable suggestions and comments.

A Appendix

Table 3 illustrates the mouth shapes with associated phonemic characters and was derived from an observation of real lips. Table 4 illustrates the mouth shapes with associated phonemic characters and examples.

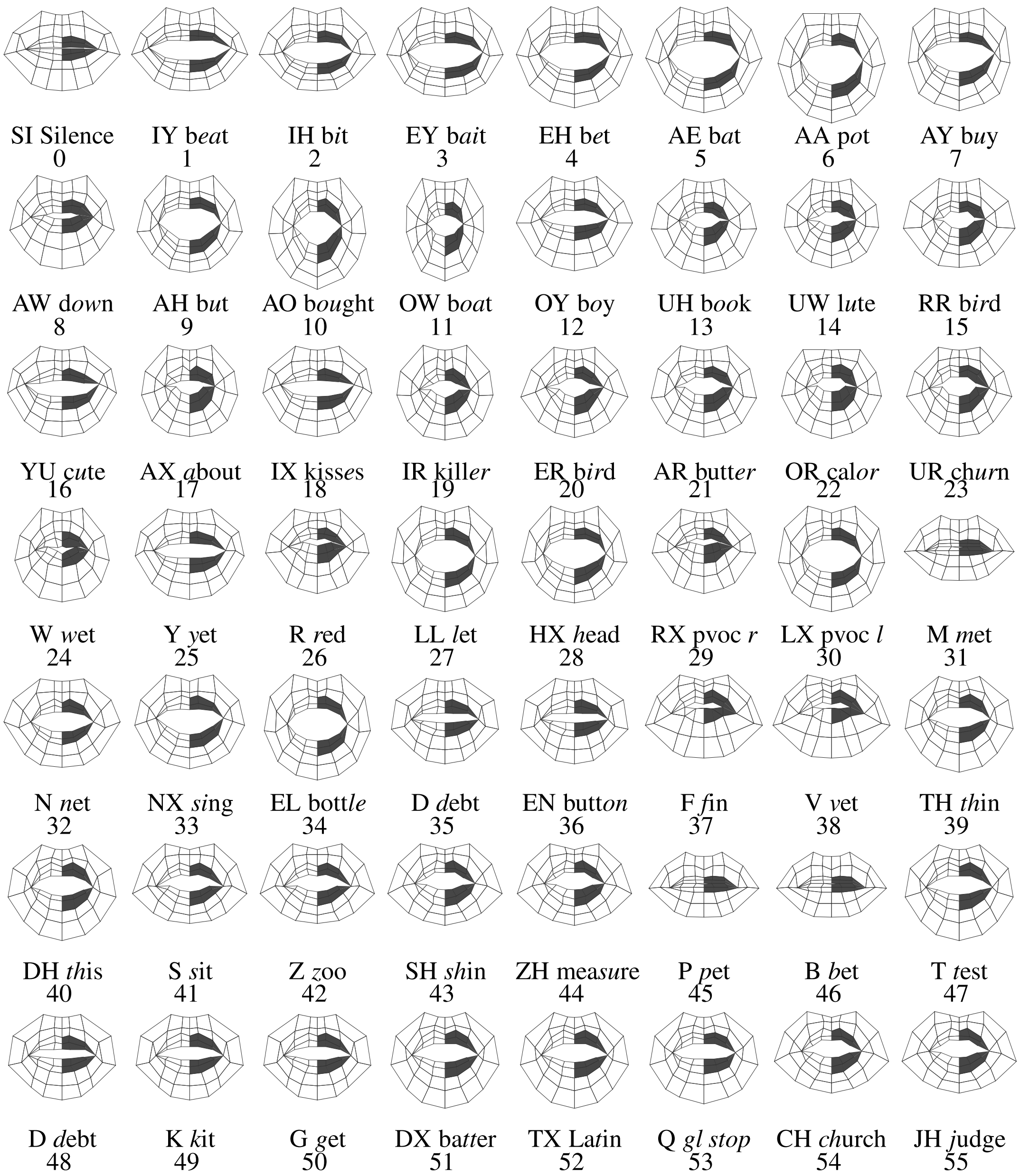


Table 3: A viseme table of mouth shapes with associated phonemic characters and examples. This table was derived from an observation of real lips

Table 4: A viseme table of mouth shapes with associated phonemic characters and examples.

Author Information

Keith Waters joined the Cambridge Research Lab (CRL) in 1991. His research interests include computer graphics, physically-based modeling, volume visualization, medical facial applications, and facial synthesis. Before joining Digital, Keith was a member of staff at the Schlumberger Lab for Computer Science Austin Texas. He recieved a Ph.D in Computer Graphics in 1988 and a BA Hons in Graphic Design in 1985 both from Middlesex Polytechnic.

Thomas M. Levergood joined CRL in 1990. Tom's focus is on speech and audio-related research. He is also involved in Alpha AXP system and software projects, including an experimental evaluation of split user/supervisor cache memories. He received his B.S. in Electrical Engineering in 1984 and M.S. in Electrical Engineering in 1993, both from Worcester Polytechnic Institute.

Both the authors can be reached at: Digital Equipment Corporation, Cambridge Research Lab, One Kendall Square, Bldg. 700, Cambridge, MA 02139, or by e-mail as {waters, tml}@crl.dec.com.

References

[BL85] P. Bergeron and P. Lachapelle. Techniques for animating characters. In *Advanced Computer Animation*, volume 2 of *SIGGRAPH '85 Tutorials*, pages 61–79. ACM, 1985.

[BMT83] E. Bruckert, M. Minow, and W. Tetschner. Three-tiered software and VLSI aid developmental system to read text aloud. *Electronics*, 1983.

[CHK+92] I. Carlbom, W.M. Hsu, G. Klinker, R. Szeliski, K. Waters, M. Doyle, J. Gettys, K.M. Harris, T.M. Levergood, R. Palmer, L. Palmer, M. Picart, D. Terzopoulos, D. Tonnesen, M. Vannier, and G. Wallace. Modeling and analysis of empirical data in collaborative environments. *Communications of the ACM (CACM)*, 35(6):74–84, June 1992.

[CT91] H. Choi, S.C. Harashima and Takebe T. Analysis and synthesis of facial expressions in knowledge-based coding of facial image sequences. In *Internationl Conference on Acoustics Speech and Signal Processing*, pages 2737–2740, 1991.

[Des66] J. Destandes. Histoire comparee du cinema. 1, 1966.

[EF77] P. Ekman and W.V. Friesen. *Manual for the Facial Action Coding System*. Consulting Psychologists Press, Palo Alto CA, 1977.

[Far90] G Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press Inc., New York, 1990.

[Fro64] V. Fromkin. Lip positions in american english vowels. *Language and Speech*, 7(3):215–225, 1964.

[Gou71] H. Gouraud. Continuous shading of curved surfaces. 20(6), 1971.

[Hec86] P. Heckbert. Survery of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, 1986.

[HPW88] 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.

[KB82] H.D. Kochanek and R.H. Bartels. Interpolating splines with local tension, continuity and bias control. *Computer Graphics*, 3(18):33–41, 1982.

[Kla80] Dennis H. Klatt. Software for a cascade/parallel formant synthesizer. *J. Acoust. Soc. Am.*, 67(3):971–995, 1980.

[Kla87] Dennis H. Klatt. Review of text-to-speech conversion for English. *J. Acoust. Soc. Am.*, 82(3):737–793, 1987.

[Lev93] Thomas M. Levergood. LoFi: A TURBOchannel audio module. CRL Technical Report 93/9, Digital Equipment Corporation, Cambridge Research Lab, 1993.

[LP87] 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.

[LPG+93] Thomas M. Levergood, Andrew C. Payne, James Gettys, G. Winfield Treese, and Lawrence C. Stewart. AudioFile: A network-transparent system for distributed audio applications. Technical Report 93/8, Digital Equipment Corporation, Cambridge Research Lab, 1993.

[McG85] 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.

[MM86] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:126–130, 1986.

[MPT88] N. Magnenat-Thalmann, N.E. Primeau, and D. Thalmann. Abstract muscle actions procedures for human face animation. *Visual Computer*, 3(5):290–297, 1988.

[NHS88] M. Nahas, H. Huitric, and M. Sanintourens. Animation of a b-spline figure. *The Visual Computer*, 3:272–276, 1988.

[Oli90] J. P. Olive. A new algorithm for a concatenative speech synthesis system using an augmented acoustic inventory of speech sounds. In *Proceedings of the European Speech Communications Association Workshop on Speech Synthesis*, pages 25–29, Autrans, France, September 1990.

[OTO+87] 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.

[Ous90] John K. Ousterhout. Tcl: An embeddable command language. In *Proceedings of the USENIX Winter Conference*, January 1990.

[Ous91] John K. Ousterhout. An X11 toolkit based on the Tcl language. In *Proceedings of the USENIX Winter Conference*, January 1991.

[Par72] F.I. Parke. Computer generated animation of faces. Master's thesis, University of Utah, Salt Lake City, June 1972. UTEC-CSc-72-120.

[Par74] F.I. Parke. *A Parameteric Model for Human Faces*. PhD thesis, University of Utah, Salt Lake City, Utah, December 1974. UTEC-CSc-75-047.

[Par82] F.I. Parke. Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, 1982.

[Par90] F.I. Parke. State of the art in facial animation. *ACM SIGGRAPH Course Notes*, 26, 1990.

[PB81] S.M. Platt and N.I. Badler. Animating facial expressions. *Computer Graphics*, 15(3):245–252, 1981.

[PFTV86] W. Press, B. Flanney, S. Teukolsky, and W. Verttering. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1986.

[Pho76] B. T. Phong. Illumination for computer generated pictures. *CACM*, 18(6):311–317, 1976.

[Pla80] S.M. Platt. A system for computer simulation of the human face. Master's thesis, The Moore School, Pennsylvania, 1980 1980.

[TJ81] F. Thomas and O. Johnson. *Disney the Illusion of Life*. Abbeville Press, New York, 1981.

[Wai89] C.T. Waite. The facial action control editor, face: A parametric facial expression editor for computer generated animation. Master's thesis, Massachusetts Institute of Technology, Media Arts and Sciences, Cambridge, Febuary 1989.

[Wal82] E.F. Walther. *Lipreading*. Nelson-Hall Inc, Chicago, 1982.

[Wat87] K. Waters. A muscle model for animating three-dimensional facial expressions. *Computer Graphics (SIGGRAPH'87)*, 21(4):17–24, July 1987.

[Wei82] P. Weil. About face. Master's thesis, Massachusetts Institute of Technology, Architecture Group Cambridge, August 1982.

[Wil90] L. Williams. Performace driven facial animation. *Computer Graphics*, 24(4):235–242, 1990.