Techniques for the Synthesis of Visual Speech

James D. Edge

Doctor of Philosophy
Department of Computer Science
University of Sheffield
September 2004

Abstract

Face-to-face dialogue is the natural mode of communication between humans. We see changes in expression and hear changes in intonation, and the combination of these provides semantic information that communicates ideas, feelings, and concepts. This is exhibited not only in the changes in speech, which confers the majority of the meaning, and the properties of vocalisation (e.g. tone, tempo and loudness), but also in changes of facial expression. This thesis investigates techniques for the synthesis of visual speech movements from the initial data capture process through to the final animation of a talking head.

Synthesis can be split into three general processes: modelling, capture, and animation. Modelling requires techniques to represent and parameterise changes in facial expression during speech production. Capture is the retrieval of information about speech articulation from real speakers, which can be either *static* poses (visual-phonemes/visemes) or *dynamic* speech movements. Finally, animation techniques take captured information about speech articulation and use it to generate trajectories through the parametric space of a facial model.

This thesis presents novel methods in each of these categories, in the framework of several systems for text-to-visual speech synthesis. Modelling is performed using geometric free-form deformation techniques to manipulate two- (image) and three-dimensional (mesh) representations of faces. Statistical techniques are used to parameterise the manipulation of facial expression. A novel technique for the retargetting of captured motions to meshes, which vary in both shape and scale from the original actor, is introduced. Animation is performed using target-based models of coarticulation, and by concatenating captured motion fragments. A novel technique for the target-based modelling of coarticulation, based upon constrained-optimization techniques, is reported.

”Language is a process of free creation; its laws and principles are fixed, but the manner in which the principles of generation are used is free and infinitely varied. Even the interpretation and use of words involves a process of free creation.”

- Noam Chomsky

Acknowledgements

The creation of this thesis and the research that it describes could not be accomplished were it not for the help and support of my supervisors and colleagues in the graphics group at Sheffield University. Their help, support and encouragement have been invaluable in turning hours of head scratching, late nights and stress into a (hopefully) coherent body of work. I would like to pick out the following in particular: *Mike Meredith*, for the answers to all those dumb questions which I should have known myself; *Mark Eastlick*, for listening to my complaints about the world; and *Ahmed Bin Subaih*, for being the terrorist in our midst.

I would like to thank *Steve Maddock* for his support and supervision over the past five years (way back into undergraduate territory.) His encouragement has probably prevented this thesis from taking another year to finish. Also, many thanks to my original supervisor, *Alan Watt*, who lightens the mood in the lab when all the rest of us can see is work. And thanks to Scott King who provided the motion-capture data used in this thesis.

I cannot imagine finishing this thesis without the help of *Manuel Sánchez*, with whom I collaborated on the retargetting of facial movement. I hope that we have both benefited from the sharing of ideas. Also, thanks to *Harini Kulatunga* for letting me sleep on the floor when I had nowhere else to go.

Finally, all my love to my parents. I know they wanted me to get a proper job when I finished my degree, but I also know that they will support me in anything I do. I may not accept all their offers of help, but that doesn't mean I don't appreciate them.

Meshes used in figures 3.3, 3.10, 3.11, 4.8, 4.9 were generated using FaceGen (`www.facegen.com`.) Meshes in fig. 3.2 were provided by Curious Labs (`www.curiouslabs.com`.)

This research was funded by the Engineering and Physical Sciences Research Council (EPSRC.)

Table of Contents

List of Figures

List of Tables

Supporting Publications

Journal Papers

M.A. Sánchez Lorenzo, J.D. Edge, and S. Maddock, Realistic performance-driven facial animation using hardware acceleration. Transactions on Graphics (*submitted*).

Conference Papers

J.D. Edge, M.A. Sánchez Lorenzo, and S. Maddock. Reusing motion data to animate visual speech. Proceedings of AISB'04, Leeds, March $30^{th} - 31^{st}$ 2004.

M.A. Sánchez Lorenzo, J.D. Edge, S.A. King, and S. Maddock. Use and re-use of facial motion capture data. Proceedings of Vision Video and Graphics, Bath, July $10^{th} - 11^{th}$ 2003.

J.D. Edge and S. Maddock. Image-based talking heads using radial basis functions. Proceedings of EGUK'03, Birmingham, June $3^{rd} - 5^{th}$ 2003.

J.D. Edge and S. Maddock. Expressive visual speech using geometric muscle functions. Proceedings of EGUK'01, London, April $3^{rd} - 5^{th}$ 2001.

Presentations

J.D. Edge and S. Maddock. Constraint-based synthesis of visual speech. SIGGRAPH'04 Sketches Programme, Los Angeles, August $8^{th} - 12^{th}$ 2004[a]

Technical Reports

M.A. Sánchez Lorenzo, J.D. Edge, and S. Maddock, Realistic Performance-driven facial animation using hardware acceleration. Department of Computer Science Technical Report CS-04-10, 2004.

J.D. Edge and S. Maddock. Spacetime constraints for viseme-based synthesis of visual speech. Department of Computer Science Technical Report CS-04-03, 2004.

J.D. Edge, M.A. Sánchez Lorenzo, and S. Maddock. Animating speech from motion fragments. Department of Computer Science Technical Report CS-04-02, 2004.

[a] SIGGRAPH sketches are refereed but are not considered as formal publications.

Chapter 1

Introduction

Face-to-face dialogue is the natural mode of communication between humans. We see changes in expression and hear changes in intonation, and the combination of these provides semantic information that communicates ideas, feelings, and concepts. This is exhibited not only in the changes in speech, which confers the majority of the meaning, and the properties of vocalisation (e.g. tone, tempo and loudness), but also in changes of facial expression. Expressions change not only with the physical process of creating speech audio (the movement of the lips and tongue), but also with emotion (happy, sad, etc.) and discourse punctuation and regulation (turn-taking, emphasis, etc., see [Pelachaud, 1991].) These contribute to the reasons why personal communication is often preferred over remote alternatives such as telephony and e-mail. In fact, humans often use technology in a way which mimics this form of personal communication, as is the case with emoticons in e-mail.

Computer-generated facial animation has long been used as a means of reproducing human-to-human interaction in widely varying settings from desktop assistants and computer games to animation and even synthetic characters in non-animated film. Traditional animation techniques have long been used to reproduce the various actions of facial communication [Lasseter, 1987], but whilst these are adequate for non-real characters (e.g. cartoon characters such as *Goofy* or *Mickey Mouse*) the same techniques do not work well as the animation gets more visually realistic. Statically it is currently possible to recreate virtual characters to a high degree of realism, and yet dynamic realism lags far behind and most production involves a large degree of manual artistic involvement.

The difficulty in modelling realistic facial movement is to be expected, given the complexity of the system we are attempting to simulate. Traditional cell animation works by in-betweening (interpolating) key poses, and thus cannot hope to adequately model the physical system of muscles, bone, skin and fatty tissue involved in facial motion. Inevitably, all animation techniques are going to be a compromise between the detail in which the model is constructed, and the complexity of the simulation. Ideally, a physical simulation would be used to model all the complexities of facial action, and yet for on-line purposes these models are far too computationally intensive to be useful. Thus, much work in facial animation is directed at approximating facial movement, whilst avoiding any such physical simulation [Cao et al., 2004, Joshi et al., 2003, Noh and Neumann, 2001, Brand, 1999, Guenter et al., 1998, Pighin et al., 1998, Bregler et al., 1997, Williams, 1990, Waters, 1987, Parke, 1974].

The animation of faces can be split into several sub-problems: representation (e.g. photographic images, triangulated point geometry), modelling, and motion generation. Any model of facial morphology

must be capable of representing both the fine and large scale geometric structure of expression, from wrinkles and skin texture up to the deformation of the skin in smiles, frowns and pouts. There must also be a way of manipulating this geometric representation to produce novel, recognisably human, expressions. And finally the animation of expression implies the generation of trajectories through whichever parametric space we use to represent our static expressions.

Representing facial morphology and modelling expression are intrinsically tied together. Modelling techniques are usually specific to a representation, whether that be a cloud of points, a set of connected geometric patches, or simply a photographic image. Such representations are often abstract and non-specific to faces. In order to aid an animator, and to conceptually link the representationto the notion of facial expression a parameterisation is often imposed as an intermediate layer (e.g. MPEG-4 facial animation parameters[Koenen, 1999].) It is this intermediate parameterisation that enables the building of compound expressions from lower level building blocks, and it is the varying of these parameters that enables the specification of animation.

The animation of faces can be performed at different levels of complexity. Simply in-betweening targets can be effective for global changes in expression, given appropriate blending functions which taper the movement between extrema. However, speech animation is a particular example where such gross simplification is inadequate. The lips, tongue and jaw do not move ina linear fashion between extrema. Animations where this approach is taken typically appear sped up (over-articulated) and unrealistic. The reason for this is that there is a causal relationship between the speech audio, which we hear, and the articulatory movements, which we see. The audio is produced, in part, by the movements of the lips and tongue, and there is a direct perceptual link between the two. In fact, experiments show both that seeing someone speak improves the recognition rate of the audio [Sumby and Pollack, 1954] and that incorrect visual movements can change its perception[McGurk and MacDonald, 1976]. This necessitates a more thorough handling of speech movements in facial animation.

The major contributory factor to the difficulties in animating speech movements is the physical phenomenon of coarticulation. Speech is often segmented into atomic units known as phonemes, representing constituent elementary sounds and their related vocal tract state. Given that each of these sounds is related to a shape or transitional movement of the vocal tract, coarticulation describes the motion of the articulators as they transition between states. In fact coarticulation is difficult to simulate because some phonemes are less*important* than others and disappear in the final transitional movement. Numerous models have been reported to describe specific effects of coarticulation [Löfqvist, 1990, Kent and Minifie, 1977, MacNeilage, 1970, Wickelgren, 1969, Öhman, 1967].

Systems for generating speech trajectories can typically be split into three categories: target-based synthesis, concatenative synthesis, and model-based synthesis. Target-based synthesis uses combinations of static poses to structure a trajectory, usually with some form of approximating curve [Cosi et al., 2003, Cohen and Massaro, 1993, Waters and Levergood, 1993]. Concatenative synthesis, similarly to concatenative audio synthesis (e.g. Festival [Black et al., 1999]), uses combinations of captured units (speech movements) to generate trajectories [Bregler et al., 1997]. Model-based synthesis attempts to find a relationship between the audio speech signal and the movements of the vocal articulators, usually using some kind of finite-state machine [Ezzat et al., 2002, Brand, 1999].

This thesis presents several novel methods for the animation of visual speech. These span the process of constructing virtual talking heads, from capture and representation through to animation and synthe-

sis of speech movements and coarticulation. A method is reported for the use of motion-captured data from one individual on several different virtual characters which vary widely in facial shape and scale. A novel method for generating speech trajectories is reported which uses constrained-optimization techniques to resolve transitions between targets according to coarticulation. A second synthesis technique is reported which generates trajectories by blending short segments of pre-captured speech movements. These techniques can be used to animate talking heads in varying domains (general vs. limited-domain synthesis) and with varying computational and memory requirements.

1.1 Main Thesis Contributions

The novel contributions of this thesis to the area of facial animation and visual speech synthesis are:

- A retargetting solution for using sparsely-sampled motion data on meshes of varying structure and geometry (shape and scale.) The method uses *radial basis functions* (RBFs) to warp the space inwhich the motions are embedded to coincide with that of a given target mesh. The relative motion of the markers before and after retargetting will remain the same. Retargetting is based upon *only* the placement of a small number of markers at fiducial points on the surface of the target mesh. This method has been published in [Sánchez et al., 2003].

- A novel method for the synthesis of speech trajectories based upon constrained-optimization is presented. Instead of using dominance (basis) functions to model transitions between targets, any form of spline can be used to represent speech transitions. The optimization minimizes the distance from the trajectory to a number of speech targets, whilst a global constraint upon acceleration models the rôle of coarticulation in speech. Positional, derivative and range constraints can be used to define the properties of the speech trajectory. An implementation of this method has been published in [Edge and Maddock, 2004].

- Methods for the adaptation and selection of motion units for concatenative synthesis. A fast greedy unit selection algorithm for variable length synthesis units is presented. Also methods for phonetic alignment and resampling based upon the use of RBFs. These techniques are published in [Edge et al., 2004], which describes the implementation of a limited-domain visual speech synthesizer for the time-domain.

- A method for applying coarticulation rules to image morphing. Such models usually work by linear interpolation of targets. Because image morphs are parameterised by point geometry (e.g. sampled splines) coarticulation cannot be modelled because the parameters of the model are not covariant with muscular action. The geometry of the morph is reparameterised using *principal components analysis* (PCA) and dominance functions used to control the variation of these parameters over time. The principal components correspond to important features, such as jaw opening, and thus allow coarticulation to be modelled. This method has been published in [Edge and Maddock, 2003].

1.2 Thesis Structure

Chapter 2, *Background: The Production and Perception of Speech*, contains an overview of background information pertaining to speech and visual aspects of speech in particular. Mostly this concerns the production of speech, phonetic categorisation of speech sounds and the relationship between phonemes and articulatory movement. Also, this chapter includes a discussion of coarticulation and numerical models to represent its effects. The perception of speech, both audible and visible, is also discussed.

Chapter 3, *Parameterisation and Modelling of Facial Expression*, contains an overview and classification of techniques for representing and manipulating facial expression. This covers specific facial parameterisations, such as FACS [Ekman and Friesen, 1978] and MPEG-4 [Koenen, 1999], along with general geometric modelling techniques and physical models.

Chapter 4, *Capturing and Retargetting Facial Motion*,introduces techniques developed for the capture, processing and retargetting of facial motion data. The retargetting method from [Sánchez et al., 2003] is discussed in detail in this chapter. Also discussed are methods for capturing motion data from actors, the nature of captured data, and processing techniques for dealing with noise and the removal of rigid motion.

Chapter 5, *Animating Speech*, discusses several methods for the generation of speech trajectories. These include work on dominance function, constrained-optimization, and concatenative methods for synthesis. The constrained-optimization method from [Edge and Maddock, 2004] is discussed in detail here, as well as methods developed for concatenative synthesis from [Edge et al., 2004]. A method for fitting dominance functions to real data is also discussed.

Chapter 6, *Implemented Systems*, describes the systems used to demonstrate the techniques from Chapters 3, 4, and 5. Chapter 7, *Conclusions*, overviews the conclusions and directions of future work. Appendix A, *Mathematical Techniques*, contains background and detail on the mathematical techniques used in the thesis. Appendix B, *Audio Speech Synthesis*, briefly discusses the common methods used in audio synthesis, as well as the Festival system [Black et al., 1999] used in the developed systems.

Chapter 2

Background: The Production and Perception of Speech

Speech synthesis requires not only the modelling of a complex physical system (the tissue and organs of the vocal tract), but also an understanding of the mechanisms behind speech production and perception. The production of speech consists of several levels from the communicative intent (the message to be passed onto a listener), through organisational groupings (phonetic, syllabic, word and sentence level units), down to the physical level of nerve impulses and coordinated muscular movement. There is a direct relationship here between the organisation of real speech production and the questions which must be tackled in synthesis: what do we want to say, how do we represent this internally, and how do we control a model of the vocal tract to produce the correct movements? Similarly, the effectiveness of any synthesis technique will be compared to the audience's experience of speech in real life, and so the perceptual mechanisms involved in speech communication can influence the design of such systems. For these reasons this chapter summarises speech production and perception background relevant to the synthesis of visual speech movements.

Speech communication is a coordinated process between speaker and listener. The speaker formulates an idea to communicate, which is subsequently encoded into a linguistic form consisting of a sequence of words from a particular language and according to its grammatical rules. These words are transformed into motor signals controlling the muscular activity of the vocal aparatus (respiration and movement of the organs involved in speech articulation.) As air is pushed through the vocal tract the articulatory movements cause pressure changes which result in speech sounds. The sounds of speech are carried acoustically as pressure waves in the intermediate atmosphere separating speaker and listener. Finally, the listener's sense organs, primarily the ears, although vision is also involved (see Section 2.2), recognize the sounds which are then decoded into the corresponding linguistic structures and interpreted. Thus, the speech communication process consists of the following steps: linguistic encoding → motor control → acoustic transmission → auditory/visual retrieval → linguistic decoding. This is, of course, a two-way process with conversation occuring back and forth with speaker and listener rôles being interchanged. This view of the speech communication process is shown in fig. 2.1.

The speech communication process, as described, is a complex interaction of physical and cognitive linguistic systems to produce and receive speech signals. The system can be considered as mainly

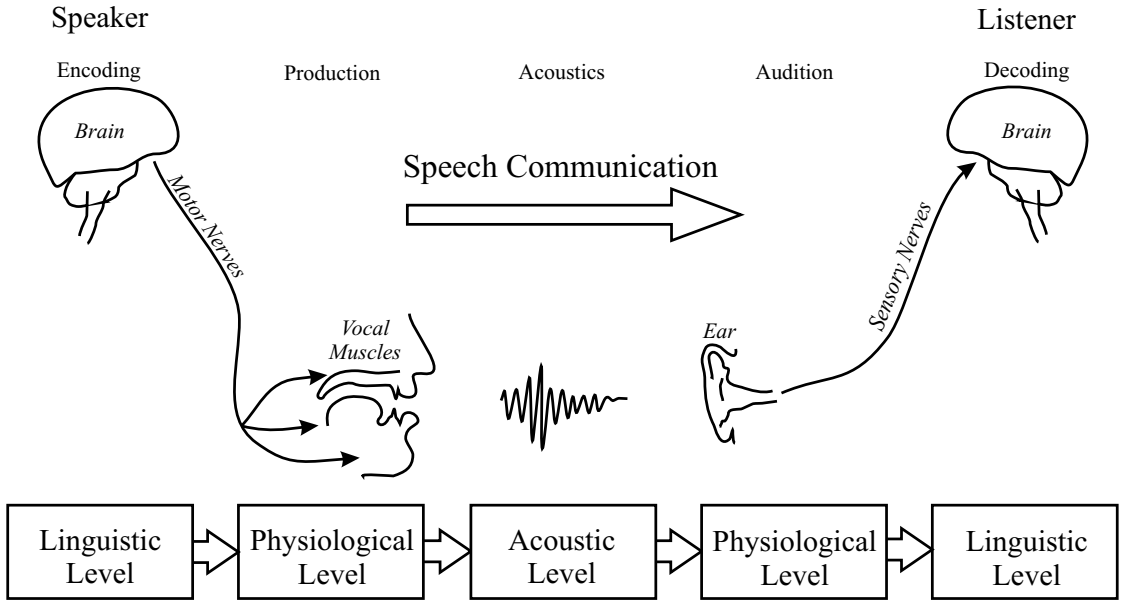


Figure 2.1: A classical view of the speech communication process (after [Denes and Pinson, 1973].)

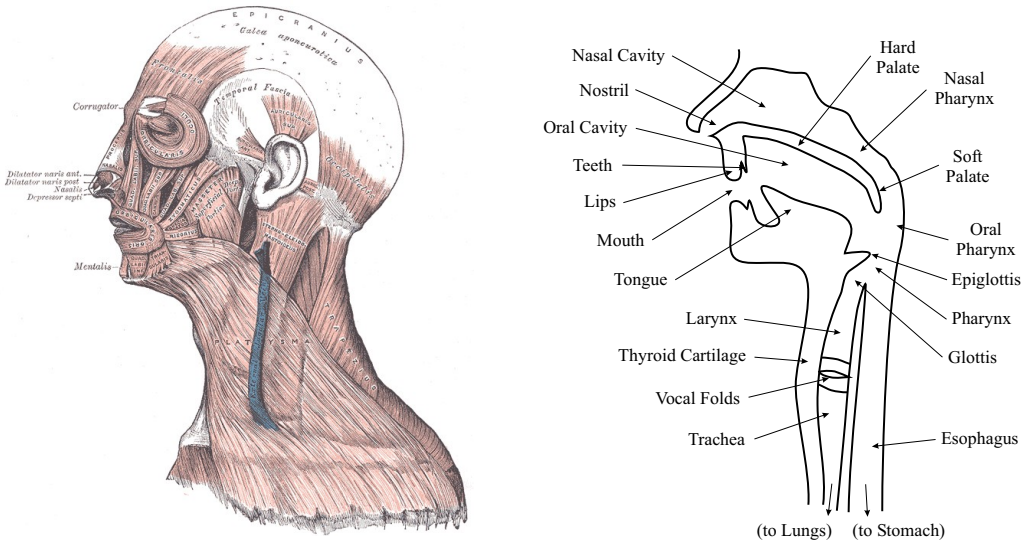
constituting mechanisms of production (anatomy, motor control etc.) and perception (audio-visual fusion.) Therefore the following sections shall deal with speech using this distinction. First, the process of speech production between linguistic encoding and acoustic transmission will be described. Second, the perception of speech focussing mainly upon audio-visual fusion given the importance of this to the success of audio-visual speech synthesis will be covered.

2.1 Production of Speech

Speech production is a coordinated physical process of the vocal articulators (lips, jaw, velum etc.) to shape the vocal tract such that intelligible speech sounds are produced. The important matters with regards this process include: vocal tract anatomy, the physical structure of the main articulatory structures; phonetics, the underlying structural categorisation of speech utterances; and speech motor control, transforming speech utterances into low level muscular control - particularly with regards to the context sensitivity of speech movement and the resulting audio (coarticulation.)

2.1.1 Anatomy of the Vocal Tract

The vocal tract is a complex physical structure which, for the purposes of speech production, regulates the passage of air from its source (the lungs) and out towards a listener. As the speech organs are moved the passage of air is constrained, or made turbulent to modify the properties of the resulting speech sounds. The main structures used in the production of speech are labelled in fig. 2.2 (b). The most important of these structures in speech production are the lungs, the trachea, the nasal cavity, the jaw and the mouth (including the hard and soft palates, the teeth, the tongue and the lips.) The structures above the pharynx are what is commonly referred to as the vocal tract, and the organs within the vocal tract used for speech production are the *articulators*.



(a) (b)

Figure 2.2: Anatomy of the vocal tract: (a) facial muscles (from [Williams et al., 1995]), (b) articulatory structures.

Table 2.1: Muscles of the lips.

NAME	ACTION
Buccinator	Compresses the cheek against the teeth, and retracts the corner of the lip down.
Depressor anguli oris	Draws the corner of the lip downward.
Depressor labii inferious	Depresses the lower lip.
Incisive inferior	Pulls the lower lip toward the teeth.
Incisive superior	Pulls the upper lip toward the teeth.
Levator anguli oris	Moves the corner of the mouth up.
Levator labii superioris	Raises the upper lip.
Levator labii superioris alaeque nasi	Raises the upper lip and nostril.
Mentalis	Raises and protrudes the upper lip.
Obicularis oris	Closes the lips, compresses the lips, and protrudes the lips.
Platysma	Pulls the corner of the mouth down and back.
Risorius	Pulls the corner of the mouth back.
Zygomaticus major	Draws the corner of the mouth laterally and upward.
Zygomaticus minor	Draws the outer part of the upper lip upward, laterally and outward.

Articulatory movement is produced by the action of a number of muscles upon the skin, bone and other subcutaneous tissues of the mouth. Muscles consist of bundles of fibres and are suspended between other structures such as bones, skin and other muscles. The muscle fibres cause motion by contracting (shortening in length) and thus applying force to the structures between which they are suspended. These muscular contractions are controlled by electrical impulses transmitted along associated nerve fibres. Linguistic units (e.g. words, phonemes etc.) must be transformed into nerve impulses to control articulatory movement, and therefore speech production. The lips and tongue are both highly muscular structures which allow a great variety in movement. The muscles of the lips and jaw are shown in fig. 2.2 (a) and table 2.1. More in depth discussion of muscle structure and function during speech can be found in [Tatham, 1969].

The oral and nasal cavities affect the resonation of the air after it passes out of the pharynx (the throat.) These cavities are separated by the hard and soft palates. The velum (soft palate) actively regulates the passage of air into the nasal cavities by raising to block the entrance, and conversely lying passive to allow air to flow out through the nose.

The mouth contains the most important articulatory structures for speech production. Most speech sounds are majorly influenced by a combination of tongue and lip movement. The tongue restricts the passage of air through the oral cavity, and the lips extend and shape the exit to the vocal tract. Lip shapes are usually categorised as: spread, as in ch*ee*se; rounded, as in *w*itch; unrounded/relaxed, as in l*o*ck. Unfortunately, such categories are rather coarse and do not reflect the full range of lip shapes that are used in natural fluid speech.

2.1.2 Phases of Speech Production

The anatomical structures briefly described in the previous section provide the means of producing speech sounds. The process of speech production can be summarised into three general phases:

- *Respiration* provides the impetus behind speech production. The lungs force air through the vocal tract and out through the oral/nasal cavities. This is a *pulmonic egressive airstream*; i.e. the lungs, via the action of the diaphragm, cause air to be forced out of the body. Respiration provides an overall structure to speech production as a continuous driving pressure must be maintained. This structure can be observed in the grammatical constructs of a language which structure sentences to allow the speaker to breathe (e.g. the length of sentences, and the presence of sentence breaks, such as commas, to allow a reader to repeat the written word.)

- *Phonation* describes the action of the vocal chords within the Larynx. The vocal folds are muscular bands of tissue which either allow expired air from the lungs to pass through passively (unvoiced, e.g. *s*ap), or rapidly vibrate to create a pulsating air stream (voiced, e.g. *z*ap.) The vibration of the vocal folds occurs in the range 80-500 Hz, with varying frequencies giving rise to the auditory sensation of *pitch*[1].

- *Articulation* is the shaping of the vocal tract above the vocal chords (the *supralaryngeal* vocal tract) to generate distinct speech sounds. This involves the movement of the major articulators (the lips, jaw, tongue etc.) to constrain the passage of air through either the oral or nasal cavities.

[1] Pitch also varies according to other factors such as age and sex.

Some articulators are passive, such as the hard palate and the alveolar ridge, and some active such as the tongue and the velum. Articulatory movement typically constrains air passage by regulating contact between active and passive articulators (e.g. tongue contact with the teeth, hard and soft palates), or by regulating airflow (e.g. the velum raises/lowers to regulate airflow through either the oral or nasal cavities.) This shaping of the supralaryngeal vocal tract changes its resonance characteristics which determines the acoustic properties of the resulting speech sounds.

The process of speech production is often considered as a combination of source and filter: the source is a combination of respiration and phonation to create a source of sound energy; the filter is the shaping of the supralaryngeal vocal tract. It is the flexibility of the articulators (lips, tongue, jaw etc.) which allows the great range of speech sounds to be created.

2.1.3 Phonetics and the Vocal Tract

Phonetics is concerned with the classification of speech sounds with regards to how they are produced (articulatory), the physical properties of the sound (acoustic), and its perception (auditory.) For the purposes of synthesis we are most interested in articulatory phonetics and how it allows us to describe speech movements and gestures according to a number of low level atomic units, i.e. the *phones* or the *phonemes* of a particular language. The International Phonetic Alphabet (IPA) provides a means of phonetically transcribing speech, which requires an extended character set. For computer-readable transcriptions the Speech Assessment Methods Phonetic Alphabet (SAMPA) has been designed which only relies upon the standard ASCII character set. Table 2.2 shows a comparison of these phonetic transcriptions with English. It is important to note that the English spelling does not have a direct symbol-to-sound relationship (particularly true of British English.)

Table 2.2: Comparison of phonetic transcription systems.

ENGLISH	the	quick	brown	fox	jumps	over	the	lazy	dog
IPA	ðʌ	kwɪk	braʊn	fɒks	dʒʌmps	əʊvɜː	ðʌ	leɪzi	dɔːg
SAMPA	DV	kwIk	braUn	fQks	dZVmps	@Uv3:	DV	leIzi	dO:g

Phonemes[2] are commonly considered to be atomic structures in speech production, however, there are a number of difficulties with this view of speech (most notably the variation of realised phonemes in natural speech, see Section 2.1.5) and several different units have also been proposed (e.g. syllables.) Even so phonemes are interesting because they are small enough in number to be a useful discription and they denote the articulatory targets in speech production. Phonetic transcription is almost invariably required at some stage of speech synthesis (both audio and visual.)

The most important categorisation in phonetics is between the vowel sounds and the consonants.

Consonants

Consonants are defined by a place of constriction within the vocal tract. There are three main features which can be used to describe consonant speech sounds: place of articulation, i.e. where the constriction of the vocal tract occurs; manner of articulation, i.e. the method by which the sound is produced; and

[2]From here on where the word *phonemes* is used, equally phones could be substituted to refer to the cross-language case.

Table 2.3: Consonant common manners of articulation

NAME	DESCRIPTION	EXAMPLE
STOP/PLOSIVE	Airflow is entirely cut off, followed by a rapid release.	*p*at, *b*at, *m*at
FRICATIVE/SPIRANT	Airflow is severely constrained, but not cut off.	*th*ick, *th*is
APPROXIMANT	Airflow is partially constrained.	*w*ell
AFFRICATE	Begins like a stop but ends with a fricative release.	*ch*eek
NASAL	Airflow through the oral cavity is entirely blocked, instead the air flows through the nose.	*m*oo*n*
LATERAL	Approximants where the airflow passes along the sides of the tongue.	*l*ady
TAP	Taps instantaneously block and release the airflow through the oral cavity.	u*tt*er[a]
TRILL	A rapid succession of taps.	pe*rr*o[b]

Table 2.4: Consonant common places of articulation.

NAME	DESCRIPTION	EXAMPLE
BILABIAL	Constriction of airflow between the lips.	*p*at
LABIODENTAL	Constriction of airflow between lip and teeth.	*f*ast
DENTAL	Constriction of airflow between the top teeth and tongue tip.	*th*at
ALVEOLAR	Constriction of airflow between the gum ridge and tongue.	*d*ebt
POSTVEOLAR	Constriction of airflow between the palatal ridge, behind the alveolar position, and tongue.	ru*sh*
PALATAL	Constrciction of airflow between the tongue and the hard palate.	*y*es
RETROFLEX	Constriction of airflow between the tongue and the palate, with the tongue curled back to face the palate.	*n*ord[c]
VELAR	Constriction of airflow between the tongue and the soft palate.	ru*ng*
UVULAR	Constriction of airflow between the tongue and the uvular.	maî*tr*e[d]
NASAL	Constriction of airflow any of the above, with the velum lowered.	*m*oo*n*

[a] from U.S. English
[b] from Spanish
[c] from Swedish
[d] from French

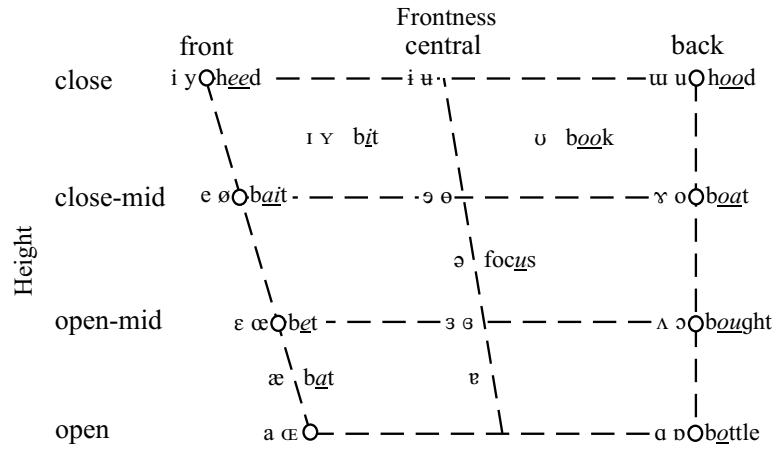


Figure 2.3: IPA vowel space with English examples (cardinal vowels are marked with circles.)

Table 2.5: Vowel qualities

QUALITY	DESCRIPTION
HEIGHT	Refers to the height of the tongue in the oral cavity.
BACKNESS	Refers to the position of the tongues heighest point in the oral cavity.
ROUNDEDNESS	Refers to whether the lips are rounded or not.
NASALISATION	Refers to whether the velum is lowered allowing air through the nose.
VOICING	Refers to whether the vocal chords are active during production of the vowel sound.

voicing, i.e. whether the vocal chords are active or passive. The manners and places of articulation for consonant speech sounds are summarised in tables 2.3 and 2.4 respectively.

Vowels

Vowels are open-mouth sounds in speech. They are best defined by the twin features of tongue position (height and location), and the roundedness of the lips. Most vowels are monophthongs, meaning that they are stable and do not include a transitional movement (e.g. h*i*t), however there also exist diphthongs which include transitions between two or more target articulatory positions (e.g. b*oy*.) Some of the qualities used to describe vowel articulation are summarised in table 2.5. Voicing is not used as a distinguishing factor for vowels in most Western languages: vowels are almost invariably voiced, except in the case of whispered speech.

Vowels are defined in relation to known *cardinal vowels* which occur at the extrema of tongue positioning. The cardinals are the extreme front and back vowels, and are used to define a space for all

Figure 2.4: A selection of visual phonemes showing spread, rounding and constriction of the vocal tract (bilabial/labiodental/dental.)

vowels (see fig. 2.3.) Note that for some positions in this vowel space there are two distinct sounds, corresponding to the rounded and unrounded variants.

2.1.4 Visual Phonetics

Articulatory phonetics, as described in the previous sections, relates the production of speech sounds to the configuration of the main articulatory structures (lips, jaw, tongue etc.) It is also known that the visual extent of these articulatory structures can be used as a cue in the perception of speech, so called lipreading or, more generally, speechreading[3].

Given that the perception of speech involves visual as well as audio cues, it is sensible to create a visual phonetic classification of speech. This classification describes speech in terms of visual-phonemes (often shortened to viseme.) Viseme classifications are non-standard, and no visual-IPA exists as such. However, visemes are usually reclassifications of their equivalent audio-phonemes[4] according to a particular parameterisation of articulatory gestures. This effectively means that if we have a parameterisation of articulatory gesture (for example using the features in tables 2.3, 2.4, and 2.5), then a viseme set could be formed by removing parameters which are not directly visible (e.g. nasality.) Mainly this concerns the place of articulation and the voicing of a speech sound. As a consequence viseme sets are significantly smaller than phonetic alphabets (like the IPA.) Some of the variation in lip shapes during speech production is demonstrated in fig. 2.4.

[3]Lipreading and speechreading differ in that the first assumes that only lip movements are important speech cues, whereas the latter assumes all visible aspects of speech production are cues in perception.

[4]From now on the word phoneme shall refer to audio-phonemes and viseme to its visual equivalent, even though phoneme could (and probably should) refer to a cross-modal speech unit.

2.1.5 Coarticulation

Coarticulation is the physical phenomenon which describes the blurring of boundaries between, what are often assumed to be, atomic units of speech, both visibly *and* audibly. Transitions between articulatory gestures are brought about by a physical system of muscles stretching skin over a boney/cartilegous substructure. The constraints of that physical system prevent instantaneous transitions between gestures, and thus coarticulation is the result of a goal-oriented task performed with a physically-constrained system.

The result of coarticulation is that the articulatory gesture formed for a certain speech unit (and the resulting sound itself) will vary during the production of natural speech. Some aspects of the gesture will vary less (e.g. lip contact in bilabial stops), and some more (e.g. jaw rotation in vowels.) In this regard phones/phonemes are variably dominant (i.e. have varying influence) over a speech utterance. This varying dominance has been described in [Recasens et al., 1997] using a scale ranking phonemes according to degree of articulatory constraint (DAC scale); such a scale can be used in conjunction with coarticulation rules to determine final trajectories through a space (parameterisation) representing vocal articulatory states.

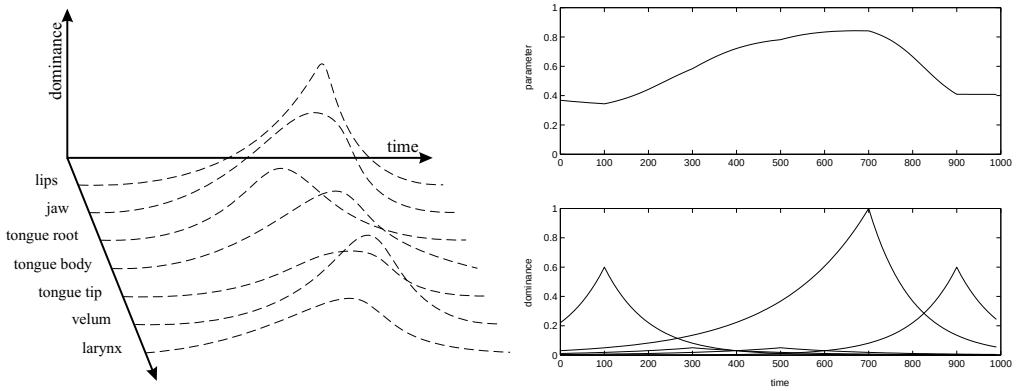
By its nature coarticulation does not only regard the extent to which a gesture is realised, but also the influence of that gesture over a period of a speech act. Coarticulation can be anticipatory, i.e. the vocal tract is preparing for an upcoming important gesture (forward coarticulation, e.g. lip rounding in $\overrightarrow{two}$), and also can reflect the effect carried over from a previous gesture (backwards coarticulation, e.g. lip protrusion in $\overleftarrow{boots}$.) Contextual effects of coarticulation have been observed up to seven segments preceeding a gesture in the French vowel /*y*/ from $ist\overrightarrow{rstr}y$ in the phrase '*une sinistre structure*' [Benguerel and Cowan, 1974].

In order to account for the nature of coarticulation, several theories have been proposed. Kent and Minifie [Kent and Minifie, 1977] categorise these into the following: learnt allophonic models; target-based models; and hierarchical models. Allophonic models, such as [Wickelgren, 1969], contest that the lowest level units for speech production are allophones (context-allophones) of some form. These units are context dependant and invariant, or at least exhibit far less variation than phones/phonemes. Target-based models [MacNeilage, 1970] assert that speech production is a goal-oriented task, where neuromotor commands are generated in a lookahead manner to attain contextually-invariant targets. Finally, hierarchical models place coarticulation as a part of an overall speech production strategy, for example Kent and Minifie themselves propose a hierarchy which covers the broad range of speech tasks from neuromotor control up to syllabic grouping. Whilst there are many proposals, with matching supporting arguments and evidence, few are concrete enough to be put to practical use (e.g. in a synthesis system.) In the following section the most common are discussed in more detail.

2.1.5.1 Modelling Coarticulation

In order to both understand and reproduce the effects of coarticulation on natural speech, numerical models have been applied. Such models must reliably reproduce the variation seen in speech, which means accounting for the physical constraints of articulatorymovement.

Öhman [Öhman, 1967] describes a numerical model which accounts for the effects of coarticulation on non-symetric vowel-consonant-vowel syllables (V_1CV_2.) In this model themovement of the tongue



(a) (b)

Figure 2.5: Modelling coarticulation: (a) dominance functions (after [Löfqvist, 1990]) representing the temporal influence of a segment over an utterance for different articulators, (b) dominance functions (after [Cohen and Massaro, 1993]) - above is the final trajectory generated by a combination of the below dominance functions.

body in x-ray sequences is predicted by (2.1).

$$s(x;t) = v(x;t) + k(t)[c(x) - v(x;t)]w_c(x) \tag{2.1}$$

In this equation, $s(x;t)$ represents the shape of the vocal tract at a point x on the tongue body at a time $t_{V1} \leq t \leq t_{V2}$ (i.e. between the centre of the initial and final vowels V_1 and V_2.) $v(x;t)$ and $c(x)$ represent the surrounding vowel and consonant vocal tract shapes respectively. The vowel shape is related to the current time because it is a transitional function between the initial and final vowel shapes. Between the initial and final vowels the influence of the central consonant is represented by a combination of $k(t) \in [0,1]$, which represents the location of the central consonant, and $w_c(x)$ which scales the influence of the consonant according to its dominance. $k(t)$ varies from 0 at time t_{V1} to 1 at t_C and back to 0 at t_{V2}, and is a smoothly varying function of time. As a result the consonant has maximum influence at time t_C which occurs at some point between t_{V1} and t_{V2}.

This model is a simple extension of interpolation to the modelling of complex coarticulation phenomena. However, the model, as Öhman himself points out, is overly simplified for the purposes of general modelling or, indeed, the application to synthesis. For example there is no way to model consonant-consonant coarticulation, and scaling the solution to non-VCV syllables provides significant challenges. Regardless of these shortcomings this model has been applied to general coarticulation in the *Mother* visual-speech synthesis system [Revéret et al., 2000].

Löfqvist extends the ideas from Öhman's simplified coarticulation model to general speech [Löfqvist, 1990]. In this model each articulator (lips, tongue, jaw etc.) has a number of related dominance functions which determines the influence a segment (phoneme) exerts over its trajectory (see fig. 2.5.) The dominance a segment exerts will vary with each articulator; for example bilabial plosives, such as *p*at, will exert a greater influence over the motion of the lips than that of the tongue.

In Löfqvist's model the shape of the dominance functions will directly determine the trajectory of a speech utterance. Although only loosely defined these functions are maximal at the centre of a seg-

ment and decrease with temporal distance. Naturally, the width of a dominance function will determine the section of an utterance over which the segment will have an influence, and thus must be no more than seven segments wide to maintain consistency with previously reported results [Benguerel and Cowan, 1974]. Dominance functions of this form easily compare with Öhman's equivalent $w_c(x)k(t)$ term in (2.1); both describe time-varying influence of one segment over its neighbours.

Cohen and Massaro [Cohen and Massaro, 1993] describe a model which implements Löfqvist's model of coarticulation. In this model negative exponential functions are used to represent the time-varying dominance functions, D_{sp} in (2.2).

$$D_{sp}(\tau) = \begin{cases} \alpha_{sp} e^{-\Theta_{\leftarrow sp}|\tau|^c} & \tau \geq 0 \\ \alpha_{sp} e^{-\Theta_{\rightarrow sp}|\tau|^c} & \tau < 0 \end{cases} \tag{2.2}$$

These functions, as dictated by Löfqvist's ideas, are maximal (as scaled by α_{sp}) at the centre of the segment ($\tau = 0$) and decrease to 0 with increasing temporal distance ($|\tau| \rightarrow \infty$). The shape of these functions depends upon the power coefficient c. Also the influence of a segment is directional, hence the difference between $-\Theta_{\leftarrow sp}$ and $-\Theta_{\rightarrow sp}$, which deals with the differences in forward and backward coarticulation. Simple additive combination of dominance functions would lead to over articulation where the dominance of several segments closely overlap. For this reason a normalized contribution of dominance functions is used to resolve coarticulation across an utterance (2.3).

$$F_p(t) = \frac{\sum_{i=1}^{n}(D_{sp}(\tau_i)T_{sp})}{\sum_{i=1}^{n} D_{sp}(\tau_i)} \tag{2.3}$$

In (2.3) T_{sp} is the target parameter (i.e. the viseme described by the parameter space of the model) for a segment outside of context. The final speech trajectory formed by this method can be fitted to real speech motions, demonstrating a relationship between this technique and speech coarticulation (although this does not imply the use of dominance functions in speech production.) Several limitations of this method for generating trajectories with specific types of speech targets (e.g. bilabial stops) have been reported [Le Goff and Benoît, 1996]. Despite this the Cohen and Massaro/Löfqvist model is the most commonly used by the visual synthesis community [Cosi et al., 2003, King, 2001, Le Goff and Benoît, 1996, Cohen and Massaro, 1993], probably because it is a fairly simple technique to implement. A more complete overview of speech synthesis and coarticulation modelling, along with contributions to the area, is presented in Chapter 5.

2.1.6 Prosody

Prosodic, also known as suprasegmental, features are those aspects of spoken communication that are evident over several phonemes of an utterance. The most commonly referenced examples are related to the pitch (fundamental frequency) of the speech audio. Stress and intonation are the speech features directly related to pitch.

Stress is a means of accentuating a syllable or word, and often emphasises the meaning of a sentence. The cues to stress are increased volume, duration and pitch. Intonation is the pattern of pitch variation across a sentence. This pattern varies according to whether the sentence carries a statement (a sharp decrease in pitch at the end), a what/where/who/when question (same as a statement), or a yes/no

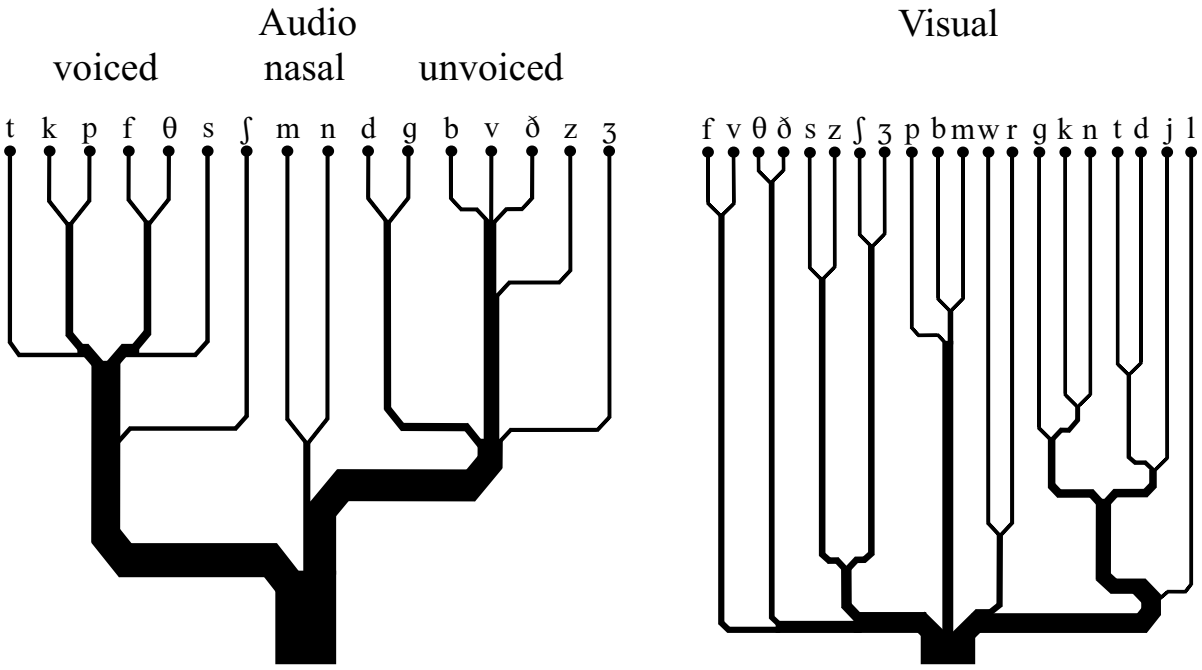


Figure 2.6: Confusion trees for audio and visual speech stimuli (after [Summerfield, 1987]), vertical axis shows decreasing noise.

question (a rising tone at the end.)[5]

Prosodic variation of this sort is accompanied by changes in the manner of speech. For example modifying the pitch of an utterance implies a change in the frequency at which the vocal folds vibrate. Similarly volume and durational changes have corresponding links with the movement of the vocal aparatus (e.g. exageration of lip shapes.) Often prosodic variation also implies the mood of the speaker (high pitched is happy, low pitched is sad), and emotional facial expressions, such as smiles, result in a variation in the prosodic features of an utterance.

2.2 Perception of Speech

The results of speech production are audio waves to be interpreted by a listener or a group of listeners. The meaning of speech is entirely contained within the properties of the audio signal, which enables remote voice communication without the immediate presence of the speaker (e.g. telephone communication.) Primarily speech is perceived via the ear and the workings of the auditory system. However, further to the information communicated audibly the visual aspects of speech act in a complementary manner, helping with the disambiguation of speech audio. This is demonstrated in fig. 2.6 which shows the confusion of audio and visual signals in the presence of increasing noise. The figure shows that those distinguishing factors which are invisible, such as voicing or nasalisation, are more important in the perception of audible speech. In contrast visual cues are more easily confused between lip shapes, such as rounding or spread lip shapes, or visible tongue movement, such as dentals. Thus easily distinguished audible speech is often difficult to separate visibly. A strong example of this is the dentals (θ as

[5]To some extent these prosodic variations are also culturally specific (e.g. statements in Australian English are similar to British English yes/no questions.)

Table 2.6: Audio-visual contradictions due to the McGurk effect (adapted from [Summerfield, 1987].)

	VIDEO	AUDIO	PERCEIVED
EXAMPLE #1	*g*oes	*b*ows	*th*ose
PLACE	Velar	Bilabial	Linguodental
MANNER	Consonantal,	Consonantal,	Consonantal,
	Voiced,	Voiced,	Voiced,
	Non-nasal	Non-nasal	Non-nasal,
	Interrupted	Interrupted	Fricative
EXAMPLE #2	*t*ap	*m*ap	*n*ap
PLACE	Alveolar	Bilabial	Alveolar
MANNER	Consonantal,	Consonantal	Consonantal
	Voiceless,	Voiced,	Voiced,
	Non-nasal,	Nasal,	Nasal,
	Interrupted	Interrupted	Interrupted
EXAMPLE #3	*m*ap	*t*ap	*p*ap
PLACE	Bilabial	Alveolar	Bilabial
MANNER	Consonantal,	Consonantal,	Consonantal,
	Voiced,	Voiceless,	Voiceless,
	Nasal,	Non-nasal,	Non-nasal,
	Interrupted	Interrupted	Interrupted

in *th*ing and ð as in *th*ese) which are never confused audibly because the voiced/unvoiced distinction is strong, but visibly are virtually indistinguishable. In contrast, fricatives are difficult to distinguish using audio cues (f as in *f*in and θ as in *th*in), yet visibly the labio-dental/dental distinction is strong.

The complementarity of the audible and visible aspects of speech is also demonstrated by the improvement in recognition rates of speech when accompanied by visual cues. In [Sumby and Pollack, 1954] as much as a +15dB improvement in signal-to-noise ratio is reported for speech audio with visual cues; this leads to a corresponding improvement in the recognition rates and thus intelligibility of speech in these circumstances. For these reasons speech research has been focussing upon audio-visual speech synthesis as an aid to communication in noisy environments [Berthommier, 2003, Le Goff et al., 1994].

2.2.1 Conflicting Audio-Visual Signals: The McGurk Effect

One phenomenon which characterises the fusion of audio and visual speech modalities is described by McGurk [McGurk and MacDonald, 1976]. The so-called McGurk effect occurs with the perception of directly conflicting audio and visual speech signals. It is found that when conflicting audio is dubbed onto a video that a subject will perceive a third distinct speech sound. An example of this is the perception of the audio */ba/* dubbed onto video of the lip movements for the syllable */da/* which leads to the perceived syllable */va/*. Several more examples can be found in table 2.6.

The McGurk effect, whilst by its nature an entirely synthetic phenomenon, demonstrates the fusion of both audio and visual modalities in the perception of speech. This is a form of synesthesia[6] with

[6]The influence of one sensory experience uponanother, i.e. vision influences the perception of audio.

the visual information augmenting the audio. Extreme examples such as those used to demonstrate McGurk would not be expected to *naturally* occur using any competant synthesis technique. However, this does demonstrate the strong link between visual speech movements and the perception of speech. The possibility that poor synchronisation could create, at the very least, ambiguity is of concern and is a demonstration of why competant models to generate speech movements are important in animation.

2.3 Summary

This chapter has introduced some of the concepts in speech production and perception necessary for speech animation. In order to generate visual speech it must be possible to both model static vocal tract articulations, and to generate movements of the articulators consistent with the nature of speech production. It is these areas that the main body of this thesis tackles.

The visible extent of the vocal tract (e.g. lips, tongue, etc.) must be modelled for synthesis. The modelling technique used must be capable of producing the same expressions[7] as are created by the muscles of the face (see table 2.1.) These expressions fit into the phonetic categories in tables 2.3 and 2.4, and are the viseme targets of synthesis. Such modelling techniques do not necessarily need to model the structure and function of the muscles and skin themselves, only the observed result. Modelling and parameterisation of facial expression are discussed in detail in Chapter 3.

Given a set of static viseme targets transitional movements must be generated according to the nature of coarticulation. This means that targets are unlikely to be met during natural speech articulation. Models relating to coarticulation are often too specific (i.e. related to individual contextual effects of coarticulation), or too abstract for implementation. The method in most common usage for visual speech synthesis is the dominance function method (see Section 2.1.5.1), which is an implementation of [Löfqvist, 1990]. An evaluation of this method along with an alternative formulation of target-based synthesis is discussed in Chapter 5. These target-based models lie in stark contrast to motion-based synthesis, which attempts to avoid modelling coarticulation by concatenating fragments of real speech movements. A comparison of these contrasting techniques is also made in Chapter 5.

The necessity of modelling speech movements correctly, and maintaining audio-visual coherence, is demonstrated by the McGurk effect. Poor coherence can adversely affect the quality of animation, as in badly dubbed film, and may change the perception of the speech itself. Whilst it is unlikely that the McGurk effect will occur accidentally, the implication is that poor speech animation will produce at least ambiguous visual signals. This is of course undesirable, and provides evidence that a thorough treatment of animation is required.

[7] The visual extent of vocal articulation can be considered to be changes in facial expression.

Chapter 3

Parameterisation and Modelling of Facial Expression

Facial animation relies upon techniques to model and encode expressions in a compact and efficient manner. Such techniques can relate to the raw storage of facial appearance and geometry (e.g. photographic images and triangle meshes), and to the modification of these structures to generate novel expressions (e.g. geometric deformation and physical simulation techniques.) These need to be able to accurately recreate the soft body deformations caused by muscular action upon the skin of the facial mask. Given the complexity of facial structure and motor function, this is a complex task especially for real-time interactive applications.

These topics are obviously important for the animation of visual speech, given that it requires intermediary mouth shapes to be modelled in synchronisation with speech audio. In many systems this is simply a matter of interpolating visemes [King et al., 2000, Ezzat and Poggio, 1999], however, as implied by coarticulation (see Section 2.1.5) such a trivial technique may not be appropriate. Furthermore, most motion capture techniques [Williams, 1990] retrieve only the motion of a sparse sampling of points on the surface of the face; to generate animation from such data (as discussed in Chapter 4) it is necessary to have techniques to interpolate the motion of these feature points across the surface of a target mesh.

This chapter contains an overview and comparison of modelling techniques which may be used for facial animation. These can be generally split into two categories:

- *Geometric techniques* - these deform a facial surface according to the manipulation of a geometric control structure (see Section 3.2.)
- *Physical techniques* - these attempt to approximately model the elastic tissue structure of the skin which is deformed by the application of muscle forces (see Section 3.3.)

Above the technique used to model facial expression there is often a need to provide a parameterisation. This simplifies the modelling of facial expression by allowing the face to be modified by intuitive quantities (e.g. *jaw opening*.) Parameterisation is also important for the generation of speech movements as coarticulation effects different aspects of the vocal tract in different ways (see Chapter 5.) The parameterisation of facial expression is discussed in Section 3.1.

3.1 Parameterising Facial Expression

The intent of facial modelling techniques is to parameterise expression in such a manner that it can be consisely described by a small number of variables. Parke's early work [Parke, 1974] introduced the concept of an intuitive facial parameterisation. In this work parameters are applied to both change the basic structural form of the face (*conformation* parameters), and to directly modify facial expression (*expression* parameters.) Conformation parameters are used to individualize a facial model, in particular so that it may resemble a particular character. Expression parameters are used to produce particular gestural (e.g. a nod), emotional (e.g. a smile or scowl), and physical (e.g. blinks) facial expressions.

The completeness of a particular parameterization refers to the ability of the model to recreate observed changes in facial shape and expression. The ideal, often referred to as *universal*, parameterization is capable of recreating observed changes in facial expression with the following properties:

- *Complete* - it should be possible to create all possible facial expressions using the parameterisation. At the very least it should be possible to model all expressions in an identified subset (e.g. speech lip shapes.)
- *Parameter Independence* - parameters should not reproduce work, and thus should be independent (orthogonal.) Truly independent parameter sets prevent additive combinations of parameters from creating unrealistic expressions. Furthermore, this requirement ensures a one-to-one mapping between expressions and parameters.
- *Minimal* - a parameterisation should consist of as few parameters as possible to accurately represent facial expression. Concise parameter sets for modelling are more usable, and generally more easily interpreted. This is partially coupled with parameter independence, since if there is no covariance between parameters the set should also be minimal.
- *Intuitive* - each parameter should have an easily-recognisable intuitive label (e.g. *jaw rotation* or *blink*.) This aids in the interpretation of parameters and the modelling of specific expressions.
- *Physically Plausible* - all expressions created with the parameter set should be observable, i.e. unrealistic facial expressions cannot be created by any combination of parameters.

Obviously from the above requirements direct vertex/control point manipulation is a poor parameterization of facial expression: covariance of neighbouring vertices is not taken into account (non-minimal), and individual vertex displacements do not correspond to recognisable sub-expressions (non-intuitive.) Numerous parameterisations for facial expression have been defined [Kähler et al., 2001, King et al., 2000, DeCarlo et al., 1998, Koch et al., 1998, Lee et al., 1995, Kalra et al., 1992, Waters, 1987, Parke, 1974] in an attempt to fulfill these requirements. However, the only existing *standard* methods are the Facial Action Coding Scheme and MPEG-4.

3.1.1 Facial Action Coding Scheme (FACS)

FACS [Ekman and Friesen, 1978] is a facial expression parameterization based upon the activation of individual muscles and muscle groups. The parameters have been attained by experimentation into which independent sub-expressions, known as Action Units (AUs, see table 3.1), can be physically

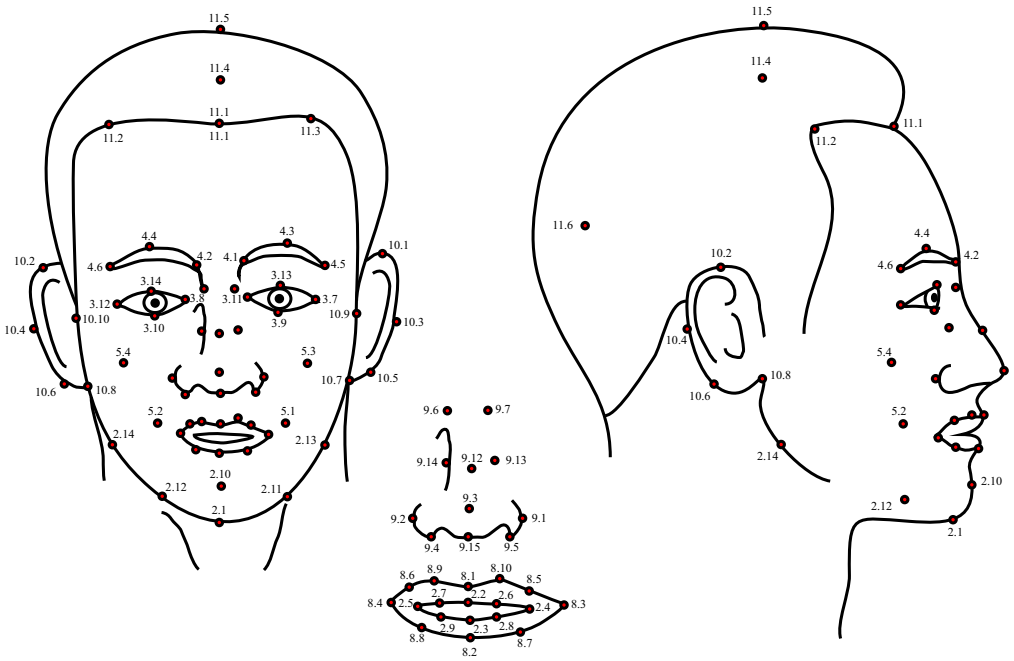


Figure 3.1: MPEG-4 feature points.

Table 3.1: Examples of FACS AUs.

AU	NAME	MUSCULAR BASIS
1	Inner Brow Raiser	Frontalis; Pars Medialis
4	Brow Raiser	Depressor Glabellae; Depressor Supercilli; Corrugator
8	Lips Together	Obicularis Oris
15	Lip Corner Depressor	Triangularis
20	Lip Stretcher	Risorius
26	Jaw Drop	Masseter; relaxed Temporal and Internal Pterygoid

Table 3.2: Examples of MPEG-4 FAPs.

FAP	NAME	DESCRIPTION
3	open_jaw	vertical jaw displacement (does not affect mouth opening)
4	lower_t_midlip	vertical top middle inner lip displacement
5	raise_b_midlip	vertical bottom middle inner lip displacement
6	stretch_l_cornerlip	horizontal displacement of left lip corner
7	stretch_r_cornerlip	horizontal displacement of right lip corner
8	lower_t_lip_lm	vertical displacement of midpoint between left corner and middle of top inner lip

produced. The assumption is that combinations of these sub-expressions can be used to recreate more complex compound expressions.

The scheme, as initially defined by Ekman and Friesen, is not intended for modelling facial expression, but to be used as a descriptive/evaluative tool. For modelling FACS has a number of disadvantages. Primarily action units may be independently produced. However, this does not mean that the effect upon the skin of the facial mask is also independent. Little is known as to how the AUs combine together to create compound expressions, which limits its use as a generative model. Despite this problem the presence of a recognized standard descriptive tool for facial expression has led to the use of FACS to directly model expression [Frydrych et al., 2003, Kalra et al., 1992], and as an influence upon the design of modelling tools [Waters, 1987]. FACS is also the basis behind the more recent MPEG-4 facial parameterisation.

3.1.2 MPEG-4 Facial Coding (FDPs/FAPs)

MPEG-4 is an ISO/IEC standard for the production and distribution of digital television, interactive graphics applications (synthetic content) and interactive multimedia [Koenen, 1999]. The standard defines specifications for synthetic face and body animation which includes both conformation (Facial Definition Parameters) and expression parameters (Facial Animation Parameters.)

Facial Definition Parameters (FDPs) transform a generic facial model stored at a terminal node such that it takes on a particular apppearance. Shape and texture can both be controlled using FDPs which also allow an entire facial model to be downloaded over a network (e.g. the Internet.) FDPs perform the same rôle as Parke's conformation parameters.

Facial Animation Parameters (FAPs) define changes in expression as offsets from a neutral facial pose. Feature points located at important locations (see fig. 3.1) are used at the lowest level to define expression. FAPs relate sub-expressions, similar to FACS action units, with the movement of the feature points on the surface of the face (see table 3.2.)

To allow FAPs to be applied across faces which vary both in shape and scale the displacement of feature-points is parameterised according to FAPUs (Facial Animation Parameter Units.) Each FAPU represents a standard measurement across the face (e.g. eye separation), and thus FAPs can be applied to models with widely varying FDPs.

FAPs have many of the same problems as AUs in parameterizing facial expression, which is unsurprising given that FAPs are based upon FACS action units. The FAPs in table 3.2 demonstrate that the units are not inpedendent, e.g. jaw rotation has been separated from lower lip movement even though the two are intrinsically linked. MPEG-4 is becoming increasingly popular in the development of facial models [Sánchez and Maddock, 2003], which is probably due to the benefits of using a standardized parameterization.

3.1.3 Statistical Parameterisation of Facial Expression

Whereas most techniques described in this chapter focus uponeither directly manipulating the geometry of the face or modelling the complex physical process of creating facial expression, generative statistical techniques provide a data-driven approach to facial modelling. These techniques determine a basis for facial expression by accounting for the observed variation in a discretely sampled subset of all possible

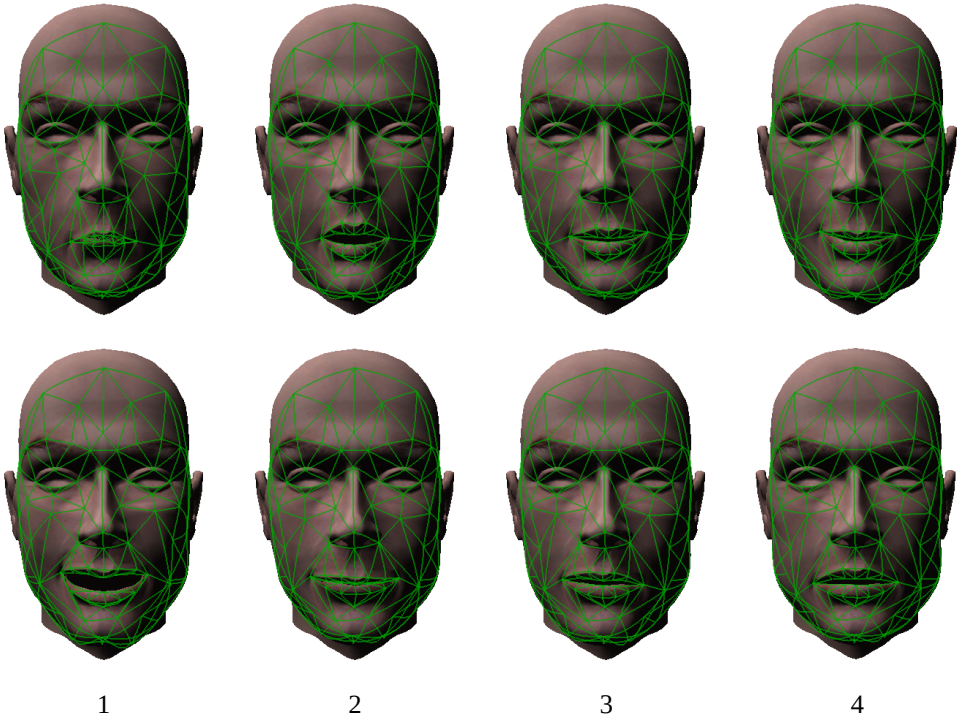


Figure 3.2: Principal component model of lip and jaw movement created from a database of 16 captured poses. Upper row shows $\mu + 3\sqrt{\sigma_i}$, lower row shows $\mu - 3\sqrt{\sigma_i}$. These principal components capture mouth opening (1), lip rounding (2), asymmetric lip-stretching (3), and smile/scowl (4).

expressions.

The fundamental assumption in the usage of statistical techniques for this purpose relies upon the completeness of the available data. Because the parameters of a statistical model are directly derived from an initial dataset, any required variation must be wholly evident in that data.

A benefit of statistical techniques lies in the ability to rank the parameters of these models by the variation they account for in the observed dataset. This allows for some degree of data compression and noise removal by culling parameters which account for only a small percentage of the variation in the data. Examples of models using statistical methods to parameterise facial expression include [Revéret et al., 2000, Blanz and Vetter, 1999, Cootes et al., 1998, Guiard-Marigny et al., 1996].

Principal Components Analysis

The most commonly used technique for computational multivariate statistics is Principal Components Analysis (PCA.) This method computes an orthogonal basis for an observed dataset directly from its covariance matrix.

Any element, v, in the original dataset can be represented using (3.1), where μ is the mean vector, e_i is the i^{th} prinicpal component, and the b_i are weights uniquely defining v.

$$v = \mu + \sum_{i=1}^{s} e_i b_i \tag{3.1}$$

The e_i principal components can be directly calculated by finding the eigenvectors of the covariance

matrix for the observed dataset, with the corresponding eigenvalues, λ_i, representing the variance, σ_i, accounted for by each component. This means that components with low eigenvalues represent only small variations in the dataset, and thus may be culled with little loss of accuracy in the model. There are a number of methods for determining which components to remove, a discussion of which can be found in [Jolliffe, 1986].

PCA has been used by several authors to create data-driven models of facial expression. In [Cootes et al., 1998] the principal components representing the change of shape and texture in a database of facial images are used in combination to model facial variation and to recognise faces in images. The same method extended to three-dimensional geometry is applied in [Blanz and Vetter, 1999] to provide a generative model of facial shape, texture and expression.

In fig. 3.2 PCA is applied to several morph targets gathered using a three camera experimental set-up. The morph targets consist of the a sparse sampling of points which deform an underlying mesh using techniques described in Section 3.2.2. Applying PCA to the gathered points allows the definition of a space of expressions, the basis of which are the principal components. This experiment is described in more detail in [Sánchez et al., 2004].

The first few parameters for PCA Models representing facial expression often intuitively correspond to parameters in hand-derived models, e.g. FACS. However, parameters representing less of the variance in the initial dataset can be less intuitive and difficult to use for modelling specific facial expressions by hand. For this reason PCA models are mostly used for data compression or as an intermediate layer in facial representation for problems such as computer vision or animation [Blanz and Vetter, 1999, Cootes et al., 1998].

3.2 Geometric Modelling of Facial Expression

Most methods used for the modelling of expressions rely upon geometric techniques to directly manipulate the surface representation of the facial mask. These techniques coincide with those Massaro [Massaro, 1998] refers to as *terminal analogue*, i.e. methods which have no direct relationship to the structure and function of facial tissue (muscles, skin, bone, etc.) The advantage here lies in the efficiency of geometric operations in contrast with the computational complexity of physical models of facial structure.

Each of the methods in the following sections manipulates the low-level representation (e.g. vertices or control points) of the face using a small number of parameters relating to facial geometry.

3.2.1 Interpolation Techniques

The most basic form of geometric modelling is to form a shape-space for facial expression as a linear combination of extremes (often referred to as morph targets.) This is simply defined in (3.2).

$$V' = \alpha V_{exp0} + (1-\alpha)V_{exp1} \tag{3.2}$$

Given two expressions, V_{exp0} and V_{exp1}, (3.2) can form a continuous transition between the two for $\alpha \in [0,1]$, or extreme caricatures for $\alpha \leq 0$ or $\alpha \geq 1$ (i.e. extrapolation.) This form of expression

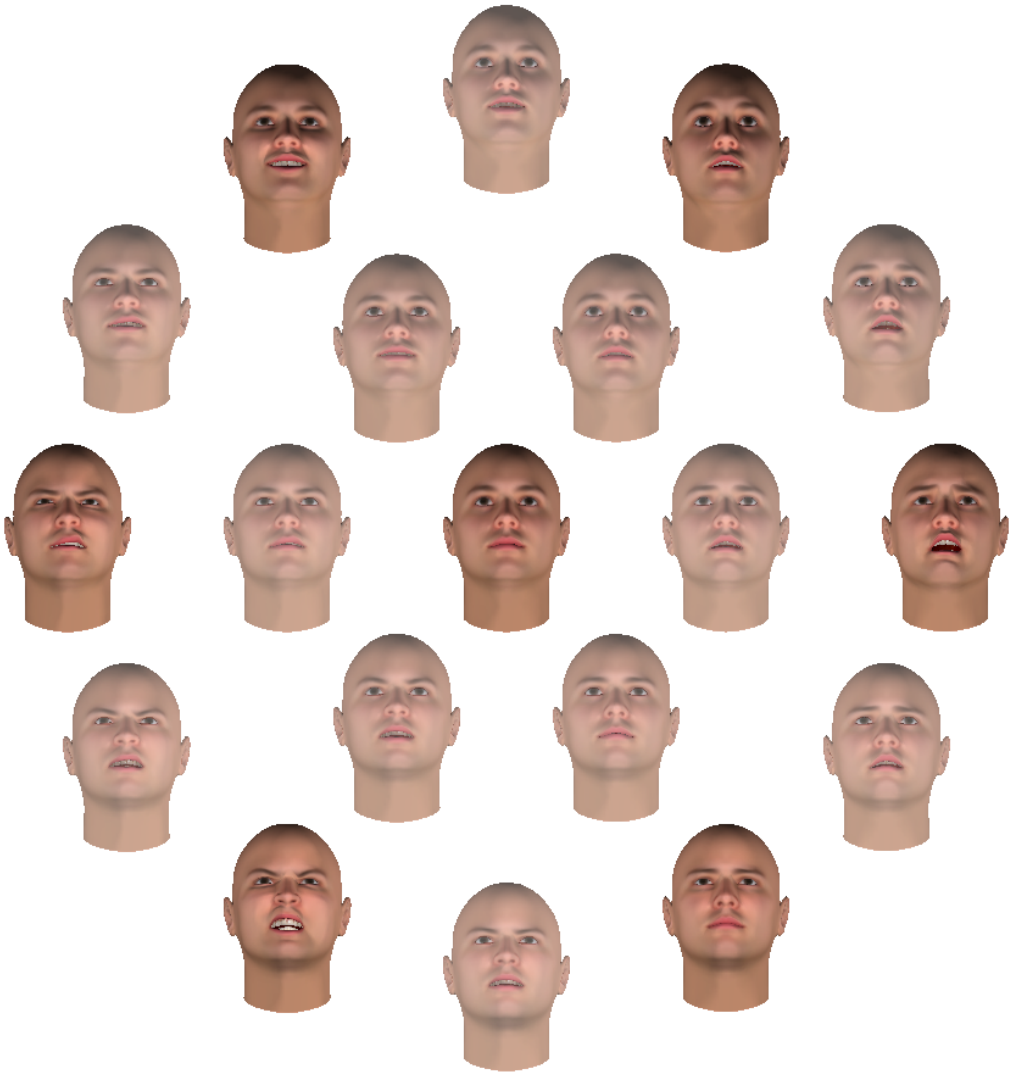


Figure 3.3: Interpolation-based model using the six universal emotional expressions (in bold clockwise from top left: Joy, Surprise, Fear, Sadness, Anger, Disgust; central pose is neutral) as key poses.

modelling is similar to the statistical techniques previously mentioned, albeit with a non-orthogonal basis.

Unfortunately global morph targets are inadequate, or at least highly inefficient for the modelling of complex facial expressions. It is necessary to localize the morph targets such that, for example, a blink morph target *only* affects the region surrounding the eye. This is the case in [Joshi et al., 2003, Pighin et al., 1998] where masks are used to explicitly define the spatial region over which the morph is active.

A further extension to the interpolation technique is to define expression not only between two extreme expressions, but between a range of expressions (3.3).

$$V' = \frac{\sum_{i=1}^{n} \alpha_i V_{expi}}{\sum_{i=1}^{n} \alpha_i} \tag{3.3}$$

This implies that changes in facial expression can be defined as a manifold in a high-dimensional

parametric space. This closely relates to the idea of expression from [Russel, 1980] which has been implemented in [Ruttkay et al., 2003]. Obviously, using a greater number of morph targets to define the manifold, and/or changing the method of interpolation (e.g. linear vs. spline interpolation), will directly impact upon the accuracy with which the manifold is approximated. Furthermore, the parameterisation of each target will determine how compact and intuitive interpolation is as a modelling technique. An example of the use of morph-targets for facial animation can be seen in fig. 3.3.

3.2.2 Free-form Deformation

Free-Form Deformation (FFD) techniques are applied to general modeling problems, and provide an interface for the sculpting of surfaces (e.g. a piecewise linear triangulation of vertices.) This sculpting process implies the definition of a function, $ffd : \mathbb{R}^d \rightarrow \mathbb{R}^d$, which applies the deformation to each point (vertex or control point) defining the target surface.

FFD tools provide an interface to manipulate a surface using a small, in comparison to the number of vertices in the target, number of controlling primitives. Typically the controlling structure determines what type of deformation will be produced. Usually, the shape of the controlling structure, and therefore the shape of the underlying target surface, is manipulated by a small number of control points. FFD techniques are best categorised by the form of controlling structure:

Point Deformers

The simplest form of FFD is to use a weighted combination of displacements from defined control points surrounding, or embedded within, the target surface (3.4).

$$V' = V + \sum_{i=1}^{l} \alpha_i (P_i - P_i') \tag{3.4}$$

As the control points, P_i, are displaced to P_i', a weighted combination of the displacement vectors $(P_i' - P_i)$ are added to the vertices, V_i, within the control point's region of influence toproduce the deformed vertices, V_i'.

Necessarily, the important factor with point-based deformers is acquiring appropriate weights, the α_i. Often a simple drop-off function can be used with points closest to a control point given greater weighting than points lying further away, e.g. a cosine drop-off within a given radius, r, (3.5).

$$\alpha_i = \begin{cases} \cos\left(\frac{\pi}{2}\left(\frac{r_i - \|V - P_i\|}{r_i}\right)\right) & \|V - P_i\| <= r_i \\ 0 & otherwise \end{cases} \tag{3.5}$$

Such a parameterization of a surface is by its nature non-continuous, and has the disadvantage that the radii must be specifically selected in order to fully enclose the target surface. An alternative, as proposed in [Kshirsagar et al., 2000], is to use a walk across the target surface to determine appropriate weights. This has the advantage that discontinuities in the target surface will affect the weighting and thus the resulting deformation should be capable of expanding/compressing gaps (e.g. for facial expression, parting the lips.) Unfortunately, a walk across the mesh is implicitly dependent upon the topology of the target surface, and applying the same method to different meshes may produce inconsistent results.

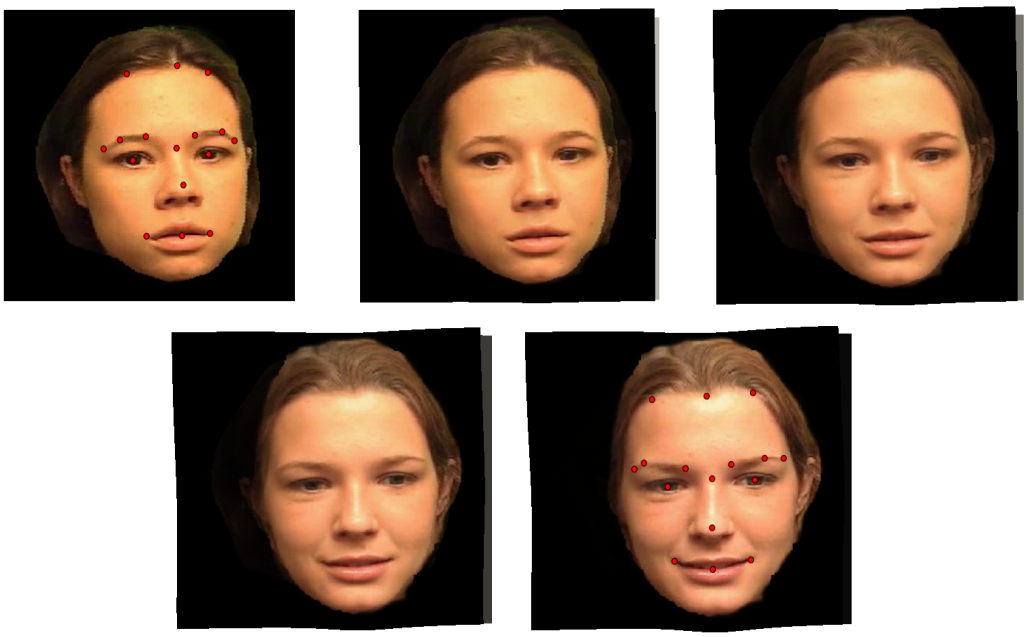


Figure 3.4: Image morphing using RBFs. Red points show corresponding features in initial and final images.

Radial Basis Functions (RBFs), a commonly used technique in scattered data interpolation, can be used for point-based deformation [Noh and Neumann, 2000, Noh et al., 2000, Ulgen, 1997] [Ruprecht and Muller, 1995, Arad et al., 1994]. RBFs are defined as functions which vary uniquely with distance from a defined basis centre. By placing an RBF centre at each of a number of control points, P_i, an interpolation can be defined as (3.6).

$$V' = p_m(V) + \sum_{i=1}^{l} \alpha_i \phi_i(\|V - P_i\|) \tag{3.6}$$

A linear combination of radial basis functions, ϕ, centered upon each of the control points P_i can thus be used to define the location of all points in $\mathbb{R}^3$ (or $\mathbb{R}^2$ if working with images.) The interpolation is defined by the α_i weights, the polynomial term p_m, and the choice of basis function ϕ (multiquadric, gaussian, thin-plate spline, etc.) The weights and polynomial term can be determined by solving a linear system which places the centres back into (3.6) and mapping onto the deformed control points, i.e. $V' = P'_i$ (see Appendix A.1.1.) The choice of basis function depends upon the required properties of the interpolation, such as locality and continuity. Radial basis function interpolation is thoroughly discussed in Appendix A.1.1.

RBFs in this case interpolate displacements across the target surface. RBFs are both global and can be selected to provide the necessary continuity in deformation. For these reasons they provide a more mathematically acceptable formulation for point-based deformation. Unfortunately, it is difficult to account for discontinuities in the target mesh using RBFs. This is because a continuous surface distance metric is required, otherwise the displacement of control points will be interpolated across mesh boundaries.

An example of point-based deformation using RBFs is image morphing. Given two images, I_0 and I_1, two warping functions are specified, $d_{0\to1} : \mathbb{R}^2 \to \mathbb{R}^2$ and $d_{1\to0} : \mathbb{R}^2 \to \mathbb{R}^2$, which respectively

forward warp I_0 to coincide with I_1 and backward warp I_1 to coincide with I_0. Once the geometric mapping between images is determined the pixels can simply be interpolated between the two images, both in colour and location. These warping functions can be implemented using RBFs. Figure 3.4 demonstrates image morphing using RBF warping functions.

Planar Deformers

Planar deformers provide a mapping between a collection of linear surface elements (e.g. triangles or quads) and each point in thetarget surface. This is a surface-to-surface mapping which allows the deformer structure to be tailored to the specific target surface which is being sculpted. Each planar deformer will consist of n control points, and the deformed surface point will be a weighted combination of only these. The major difference between point and planar-element deformers lies in recovering the weights for each point in the target surface. For planar elements, spanning a set of control points, P_i, the parameterisation consists of a mapping between the planar element and the target surface, e.g. the barycentric coordinates, β_i, of the point projected onto the planar element and an offset vector, of length d_V along the deformed surface normal, n', (3.7). This means that planar deformers are fully parameterising the target surface, whereas point deformers can only interpolate displacements. Figure 3.5 demonstrates a triangular planar-deformer.

$$V' = n' d_V + \sum_{i=1}^{n} \beta_i P_i \tag{3.7}$$

In this formulation, each point in the target surface is only bound to a single element in the control surface. The appropriate element may be determined by projecting (e.g. a cylindrical projection, projection along the surface normal etc.) the target point onto the control surface or finding the closest element in the controller surface. This deformation technique requires that the entire domain of the target surface is encased in a controlling structure, otherwise a decision must be made on how to deal with points lying outside of all control elements. Optionally a weighting function may be imposed on top of the basic deformation technique, such as (3.5), which allows the strength of deformation to be tapered off with distance. Variations upon this method are described in [Sánchez and Maddock, 2003, Singh and Kokkevis, 2000].

Piecewise-polynomial Deformers

The previously described FFD tools parameterise a target mesh with regards to discrete primitives, with no regard to higher order continuity in the deformation (with the exception of RBF-based point deformation.) To provide more complex deformations the number of primitives required could reduce the effectiveness of these tools. To allow greater flexibility, without too great an increase in the complexity of the controlling structure, piecewise-polynomial primitives can be used to control deformation. The continuity of the deformation in the target surface now becomes a factor of the continuity of the underlying polynomial basis. Furthermore, continuity across deformer boundaries can be ensured in the same way that it would be if the deformer primitives were to be used to define the surface itself. FFD tools for univariate (splines) [Singh and Fiume, 1998, Lazarus et al., 1994], bivariate (patches) [Sánchez et al., 2004], and trivariate (volumes) [Hsu et al., 1992, Coquillart, 1990,

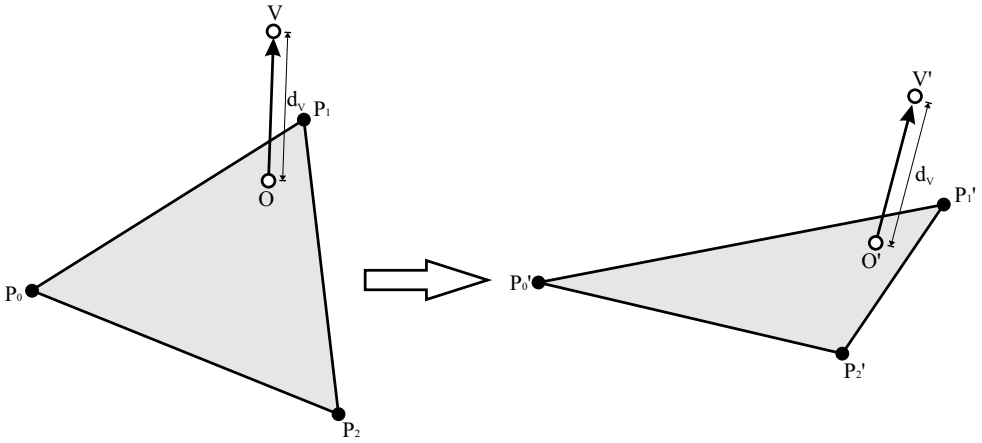


Figure 3.5: Planar free-form deformation.

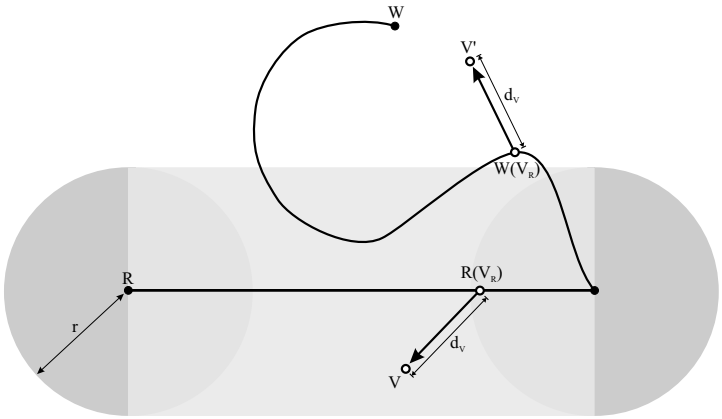


Figure 3.6: Spline-based free-form deformation.

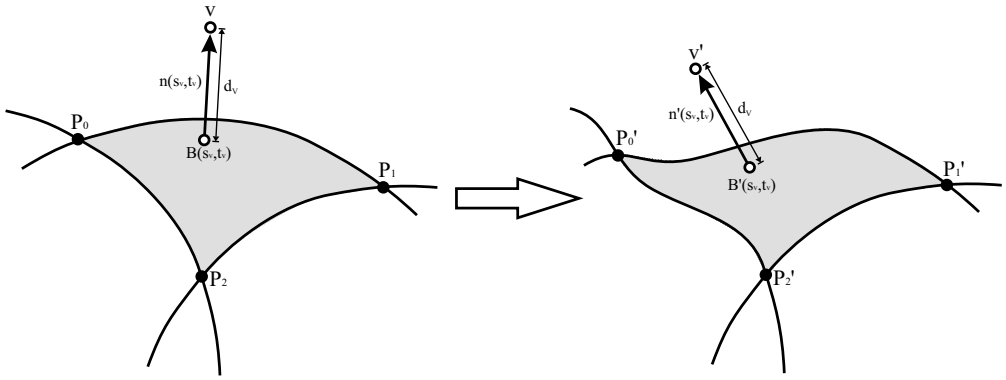


Figure 3.7: Patch-based free-form deformation.

Sederberg and Parry, 1986] piecewise-polynomial primitives have been developed. The polynomial basis for these tools can be changed depending upon the required deformation, e.g. a Bézier or B-spline (quadratic, cubic, etc.) basis could equally be used.

Splines

Spline FFDs [Lazarus et al., 1994], also known as Wires [Singh and Fiume, 1998], act like a sculptor's armature in manipulating a target surface. A combination of splines lying on or close to the target surface are manipulated to perform a deformation of that surface. As the deformation technique is based upon the manipulation of a spline, the properties (e.g. derivative continuity) of the spline used will be transferred to the deformation technique. The properties of the spline basis functions will determine the form of deformation.

Each vertex in the target surface is parameterized according to its distance to the closest point on the spline deformer. In this manner the deformation technique is alike linear axial deformations, commonly known as bone deformers, with the extra degrees of freedom imparted by the spline formulation. Determining the closest point on the spline deformer requires an optimization technique, and a method to resolve ambiguities, e.g. where more than one point on the spline deformer is equidistant to a vertex in the target surface. Figure 3.7 demonstrates spline FFD deformation.

Spline Axial deformers [Lazarus et al., 1994] manipulate a surface by forming a frame, such as a Frenet frame, with its origin at the closest point on the controlling spline. This frame will transform according to the displacement of control points, and this transformation is applied to the attached vertices to deform the target surface.

In the Wires [Singh and Fiume, 1998] formulation the FFD is defined by a tuple, $\langle W, R, s, r, f \rangle$. W and R are splines representing wire and its undeformed reference curve respectively, s is a scaling factor, whilst r is the radius of influence surrounding the wire. The definition is completed by an implicit function, $f : \mathbb{R} \rightarrow [0,1]$, which controls the decrease in influence with distance perpendicular to the wire. According to this definition the wire deforms a vertex, V, with closest point on the reference curve R_V where $f\left(\frac{\|V-R_V\|}{r}\right) > 0$, in the target surface according to the following sequence:

1. Scale V uniformly about R to create V_{scaled}, i.e. $V_{scaled} = V + (V - R_V)(1 + (s-1) f\left(\frac{\|V-R_V\|}{r}\right))$
2. Take the angle, θ, between the tangent of the closest point on the wire, W'_V and its reference curve, R'_V, and rotate V_{scaled} by the modulated angle $\theta f\left(\frac{\|V-R_V\|}{r}\right)$ to create V_{twist}. This creates a twist deformation along the wire.
3. Add the translation to the result of scaling and twisting, i.e. $V' = V_{twist} + f\left(\frac{\|V-R_V\|}{r}\right)(W_V - R_V)$.

In the wires formulation, because attachments to the target surface are weighted according to f, combinations of several wires can be used to deform the target. To provide higher-derivative continuity between wires RBFs could be used to provide the weighting function f.

Patches

Patches, e.g. Bézier or NURBS, are commonly used to represent free-form surfaces. They can be used to produce surfaces of an ascertained degree of continuity by applying constraints to the placement of the control points which define the surface [Clough and Tocher, 1965]. Similarly networks of patches can

be used so that they accurately represent a sampled surface [Krishnamurthy and Levoy, 1996]. These properties of patch surfaces make them particulatly appropriate to the free-form deformation of meshes.

In [Sánchez et al., 2004] a triangulation of feature points on a mesh, using Bézier triangle patches, is used to parameterise and deform that mesh. The Bézier surface is an approximation to the surface of the model with only the placement of the control points exposed for user interaction. Each of the vertices, V, in the target mesh is parameterised according to three parameters: the parametric coordinates (s_V, t_V) of the vertex projected onto the closest patch in the deformer surface, and the normal offset d_V (i.e. $d_V = \|V - B(s_V, t_V)\|$, where B is the parametric definition of the controller patch.) Given this parameterisation vertices in the target surface may be reconstructed by evaluating the Bézier patches and their respective normal maps (3.8).

$$V' = d_V n_i(s_V, t_V) + B_i(s_V, t_V) \tag{3.8}$$

By evaluating (3.8) with displaced control points the target surface will deform accordingly (see fig. 3.7.) The deformation of the target surface is both local and continuous as determined by the deformer surface and the constraints upon its shape.

Identifying the closest point on a surface constructed as a network of Bézier patches is a non-trivial problem. To solve this a Newton-Raphson steepest descent method (see [William H. Press and Flannery, 1992] for a description of steepest descent) is used to minimize the function in (3.9).

$$\left(\frac{\partial \|V - B_i(s,t)\|^2}{\partial s}\right)^2 + \left(\frac{\partial \|V - B_i(s,t)\|^2}{\partial t}\right)^2 = 0 \tag{3.9}$$

In practice this is expensive for large numbers of patch deformers, so an initial sampling of the deformer surface is generated. By determining the closest point in the sampled pointset, the closest patch can be determined and the parametric coordinates of the closest point can be used to initialize the Newton-Raphson search. The continuity of this deformation technique relies solely upon the continuity of the deformer surface. In [Sánchez et al., 2004] it is demonstrated that for a surface of continuity C^n the continuity in the deformation will be C^{n-1}.

Volumes

The original FFD formulation, as described in [Sederberg and Parry, 1986], employs a trivariate Bézier lattice to control an embedded object. The method requires only that the target surface be locally parameterised within the lattice structure and that (3.10) be evaluated with the displaced control points, P'_{ijk}.

$$V' = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} B_i(s) B_j(t) B_k(u) P'_{ijk} \tag{3.10}$$

Where $B_{\{i,j,k\}}$ are the basis functions of the spline representation. The parametric coordinates, $\{s,t,u\}$, of vertices in the target surface are defined by (3.11), where $\{S,T,U\}$ are the unit axis vectors and X_{orig} is the origin of the local frame containing the FFD lattice (see fig. 3.8.)

$$s = \frac{T \times U(V - X_{orig})}{T \times US}, t = \frac{S \times U(V - X_{orig})}{S \times UT}, u = \frac{S \times T(V - X_{orig})}{S \times TU} \tag{3.11}$$

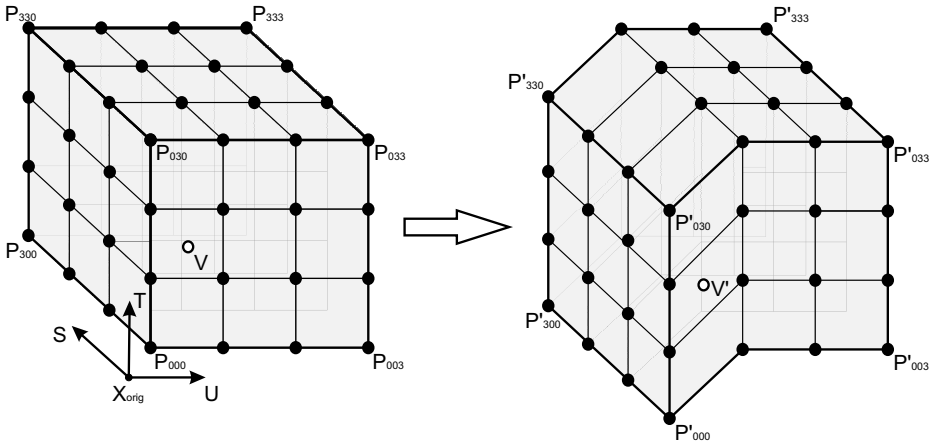


Figure 3.8: Volume-based free-form deformation.

This form of FFD is limited by its cubic lattice structure. In [Kalra et al., 1992] a rational basis, i.e. weighted basis functions, is used to apply cubic lattice deformers to facial modelling. This problem domain requires the manipulation of surfaces which are dissimilar from a cubic lattice, and thus some modification of the standard technique is necessary.

In [Coquillart, 1990] and later [MacCracken and Joy, 1996], extend the basic method to allow for arbitrary topology lattices. To allow for arbitrary shaped lattices the parametric coordinates must be for an embedded object must be obtained, which for the case of non-cubic lattices may not have an analytic solution. Coquillart uses numerical methods (Newton-Raphson steepest descent) to determine the parameterisation, with the disadvantages of computational expense and the numerical sensitivity of the problem. MacCracken uses a Catmull-Clark subdivision approach to parameterise a target surface.

Muscle functions

Geometric muscle functions are a form of univariate FFD which approximates the action of muscles upon the surface of the skin. These were introduced in [Waters, 1987] where two different types of muscle are modelled: linear muscles, which pull from an attachment to the skin towards an insertion into the skull; and sphincter muscles which pull the skin towards a central point.

The action of a linear muscle is modelled by contracting vertices within a conic section towards its apex (3.12), see fig. 3.9.

$$\begin{aligned} V' &= V + cos(\gamma) kr \frac{(X_{ins} - V)}{\|X_{ins} - V\|} \\ r &= \begin{cases} cos\left(\frac{1 - \|X_{ins} - V\|}{R_s}\right) & V \in \langle X_{ins}, P_n, P_m \rangle \\ cos\left(\frac{\|X_{ins} - V\| - R_s}{R_f - R_s}\right) & V \in \langle P_n, P_r, P_s, P_m \rangle \end{cases} \end{aligned} \tag{3.12}$$

In (3.12) the displacement of a vertex from V to V' is determined by the distance from the muscles apex, X_{ins}, and the angle from its central axis, γ. The strength of deformation is controlled by the factor k. The muscle is split into two regions, with maximum deformation occuring at the meeting of the central axis at a distance R_f from X_{ins}. At the border of the muscle no deformation occurs, ensuring continuity across the boundary of the muscle. Sphincter muscles act by displacing vertices towards the centre of an ellipse (3.13), see fig. 3.9.

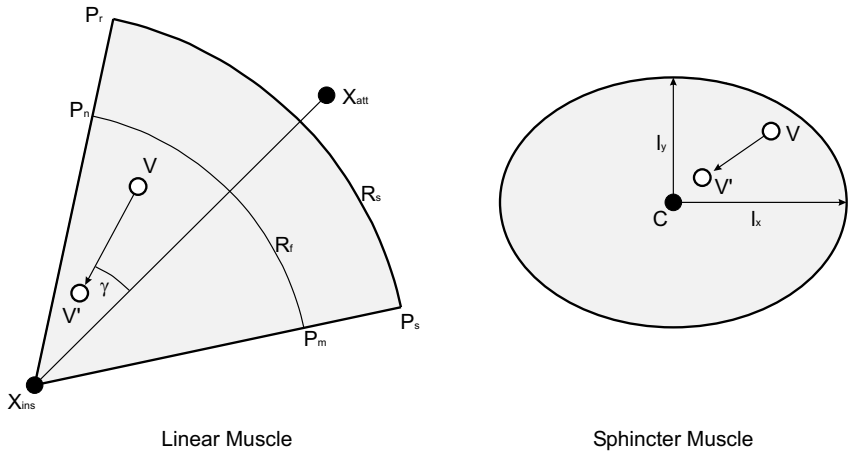


Figure 3.9: Geometric muscle function free-form deformation.

$$\begin{aligned} V' &= V + k f g \tfrac{C-V}{\|C-V\|} \\ f &= \tfrac{\sqrt{l_y^2 V_x^2 + l_x^2 V_y^2}}{l_x l_y} \\ g &= \tfrac{\|C-V\|}{l_x} \end{aligned} \tag{3.13}$$

In (3.13) the displacement of a vertex V to V' is determined by the distance between the vertex and the centre of the ellipse, C. Again the strength of deformation is determined by k, and there is not deformation at the boundary of the muscle. The geometry of the sphincter muscle is shown in fig. 3.9.

Both linear and sphincter muscle functions are 2D in nature. To apply the deformation to 3D meshes the functions can be extended to conic (linear muscles) and ellipsoid (sphincter muscles) volumes respectively, or alternatively the vertices of the mesh can be projected into the plane of the muscle. A weighted combination of muscle functions can be used to produce compound facial expressions. Further examples of muscle function models can be found in [Breton et al., 2001, Pasquariello and Pelachaud, 2001]. The use of muscle functions to animate speech is further discussed in Section 6.1.

3.2.3 Free-form Deformations and Discontinuities

Free-form modelling techniques typically interpolate the displacement of a set of markers/control points across the surface of an object or across the space within which the target object is embedded. However, this is inconsistent with the physical nature of soft body deformations, such as facial skin under strain. Because topological structure is not taken into account, discontinuities are poorly modelled by these techniques. This is particularly important for modelling expression because the facial mask has important functional discontinuities (e.g. the openings between the lips and the eyelids.)

The techniques described can be split two ways depending upon how control elements are bound to the geometry of an object: techniques in which a weighted combination of control elements are used to displace a vertex (e.g. bones, wires, etc.), and techniques in which each vertex is bound to only a single control element (patch- and some planar-element deformations.) To account for discontinuities either the influence of control elements must be culled, in the case where multiple control elements deform a single vertex, or some geometric test must be applied to ensure the correct binding between control element and vertex.

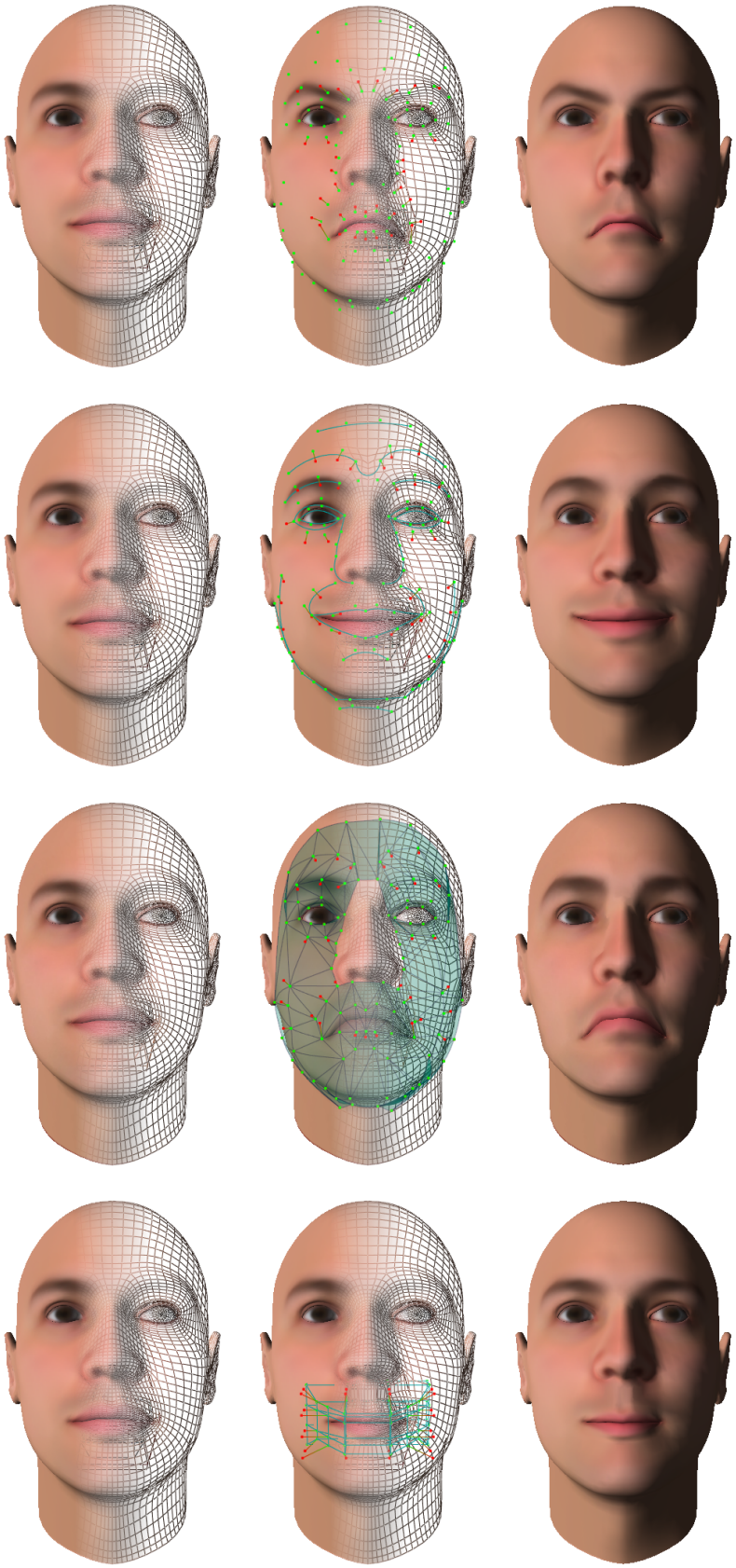


Figure 3.10: Free-form deformation for modelling facial expressions, from top to bottom: point (RBF), spline (Wires), surface (BIDS), and volume (Bézier-volume) deformers.

Directly culling the influence of each control element requires that a mask be defined. This is a manual, laborious and error-prone process which is also necessarily topology specific, and so must be repeated when applying the deformation technique to a new mesh. In contrast, for surface-based FFD techniques there is a similarity relationship between the controlling structure and the target surface, which can be used to enhance the attachment of vertices to the control elements. Primarily the orientation of the faces in the target mesh and the control elements themselves (or the attachment points on the control element for patch-based deformations) can be used as a disambiguating factor to rule out the attachment of vertices. The correlation between surface normals on the target and controller surfaces can be used to correctly bind surfaces whilst taking into account discontinuities. This can be done by using a threshold to define the maximum disparity between the vertex normals of the target and the surface normal at a proposed attachment point on the controller surface, preventing obviously inappropriate attachments. Where a discontinuity is present in the facial mask, vertices on either side will have surface normals facing in opposite directions, in these situations this method works well. A geometric test is obviously beneficial because it makes the process of attachment automatic, but also has the added benefit that in geometrically delicate situations (e.g. where the upper and lower lips overlap) it can be difficult to define masks.

The exception to this is the case of radial basis functions used to interpolate marker displacements. Because RBFs are global and require a linear system to be solved to calculate the deformation, it is not possible to directly mask the influences. Instead, the only way to incorporate discontinuities into RBF interpolation is to use a surface-based distance metric, and so the interpolation is performed across the surface itself and not in Euclidean space. Unfortunately, it is cumbersome and computationally expensive to define a general purpose surface distance metric and cheap alternatives such as edge-based distance metrics [Noh et al., 2000] do not work well.

3.3 Physical Modelling of Facial Expression

Contrasting with geometric models of facial expression, physical models attempt to model both structure and function of the human face. The facial mask is a complex structure consisting of skin, muscle, bone and fatty tissue. Expressions are created by muscles applying forces to the facial mask causing the skin stretch, crease, and wrinkle according to a combination of factors such as age and weight. Physical models of facial expression require the elastic nature of the facial mask to be simulated. These techniques can be split into two different areas: tension networks, which model the skin as a network of masses interconnected by springs; and finite-element models, which attempts to model the skin as an elastic continuum.

Tension networks [Lee et al., 1995, Platt and Badler, 1981] treat the facial mask as a set of interconnected masses and springs modelling the elastic response of skin to muscular forces. As a force is applied to a node in the network it will be applied to all the interconnected nodes and thus propogated across the skin until the system reaches equilibrium. The restitution force caused by springs in a tension network which are displaced from their rest length is simply calculated according to (3.14).

$$F_{i\rightarrow j} = k_{spring}\left(\frac{X_j - X_i}{\|X_j - X_i\|}\right)(\|X_j - X_i\| - \|X'_j - X'_i\|) \tag{3.14}$$

In (3.14) the force, $F_{i\rightarrow j}$, on a node at X_j due to a spring connected to node at X_i is directly related

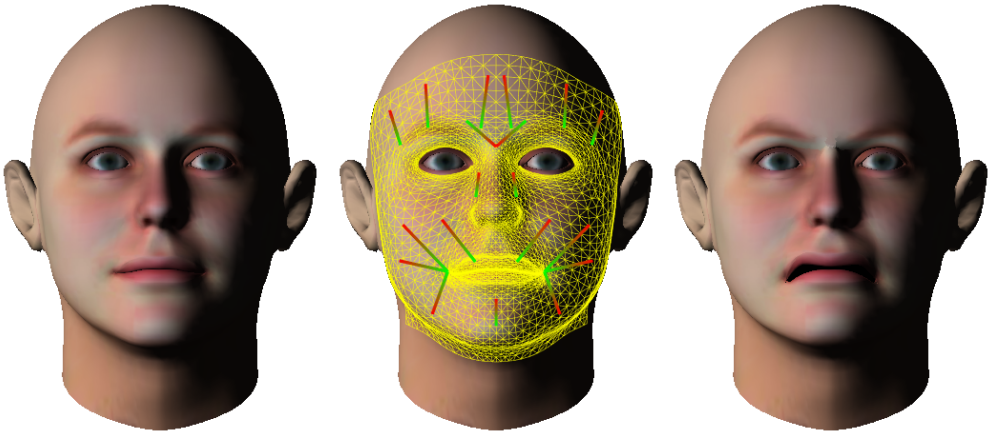


Figure 3.11: Tension-net model of facial expression (from left-to-right): original mesh, superimposed tension-net and muscle vectors, deformed mesh.

to the change from the nodes' rest positions X'_j and X'_i (i.e. due to a change from the rest length of the spring.) The spring constant k_{spring} determines the magnitude of force caused by a displacement of the nodes. For more accurate models of elastic behaviour the spring constant becomes a function of spring length (or equivalently displacement from the spring rest length), i.e. k_{spring} becomes $k_{spring}(\|X_j - X_i\|)$.

Choice of network structure is key to the success of tension-networks. Uniformity in the length of springs is required to maintain consistency in the elasticity of the skin. However, most models for convenience use the triangulation of the target mesh to define the structure and connectivity of the springs [Lee et al., 1995]. This leads to inconsistency in the resistance of the skin to muscle forces.

Forces are applied to the nodes of a tension network to create facial expressions. These forces represent the action of muscles as well as factors like volume preservation and prevention of skull penetration. In order to resolve the action of forces on a tension network, a system of second order differential equations (ODEs) must be solved. A variety of methods are available to integrate the equations of motion, which each trade off accuracy against speed and complexity; e.g. in [Lee et al., 1995] simple explicit-Euler integration is used, whilst in [Kähler et al., 2001] Verlet-leapfrog is used to improve stability in the solution.

In [Choe et al., 2001, Koch et al., 1998, Koch et al., 1996] finite-element models (FEM) are used to model the elastic properties of the skin. These models segment the skin into a number of simple geometric elements which are used to approximate the solution to a number of differential equations. The model proceeds by minimizing the overall energy of the surface, which is the combination of internal (resistance to bending and stretching) and external (muscle) forces, whilst enforcing a number of boundary constraints. FEM models converge to global solutions given a set of external forces applied to the surface.

Both tension-network and FEM models require models of facial muscles to produce expressions. These in general are of a similar form to the functions described in Section 3.2.2. A geometric function is used to determine the variation of muscle force across some defined volume. This, as in the geometric case, is a gross simplification of the action of facial muscles. Even so, in [Pitermann and Munhall, 2001] the model from [Lee et al., 1995] has been demonstrated to reasonably approximate measured human facial movement. One significant advantage of physics models over geometric deformations is that

discontinuities do not explicitly need to be modelled. This is due to the fact that the physical system models the structure of the facial mask, and thus forces will only be propogated across the surface of the skin (i.e. deformations do not occur in the space surrounding the facial model.) An example of physically modelling facial expression is demonstrated in fig. 3.11, where muscle functions similar to [Waters, 1987] are used to deform a tension-net model.

3.4 Summary

This chapter has discussed the wide variety of methods for parameterising and modelling facial expression. Modelling techniques provide low-level control of facial geometry, whilst parameterisation is used to provide a key intermediary layer between the manipulation of facial geometry and the animator or animation technique.

Modelling techniques can be split into two main categories: geometrically-, and physically-based. Geometric modelling of facial expression either forms novel expressions from combinations of captured expressions (morphing), or by directly deforming the surface geometry (free-form deformation.) Physical models attempt to model the structure and function of the skin and muscles in the creation of expression. Geometric modelling is efficient, but requires explicit modelling of discontinuities in the skin (e.g. between the lips.) In contrast modelling the elastic properties of the skin is currently not feasible for real-time applications, except where significant compromises are made in regard to the accuracy of the simulation. It is likely that in the medium-long term physically-based techniques will become more popular, if only because geometric techniques are an even coarser approximation to the action of facial muscles upon the skin. However, currently FFD deformers are the best way of modelling facial expression in real-time.

Free-form deformation techniques deform a mesh by interpolating the displacement of a few (i.e. much less than the number of vertices in the mesh) control points. The form of deformation depends upon the type of the controlling structure. Various geometric primitives have been used as FFD control structures, including: points, lines (bones), splines, triangles, patches, and volumes. The facial mask is a surface, and the motion of points on the skin can be measured, thus surface deformer primitives are intuitively appropriate for modelling facial expression. In Chapter 4, surface-based FFD primitives are used to interpolate the motion of captured markers across the surface of a mesh representing the face. In Chapter 6, point-, surface-, and muscle-based FFDs are demonstrated for the purposes of modelling static visemes, and animating visual speech.

The parameterisation of facial expression is necessary to mediate between modelling (e.g. FFDs) and animation techniques. This is particularly important for speech animation because coarticulation effects different aspects of the articulators (e.g. lip width/height) in different ways. The effect of coarticulation does not occur parallel with the axes of 2D/3D Euclidean space. In Sections 6.2 and 6.3 two systems are described which use *principal components analysis* to parameterise the geometry of speech articulation. This allows sampled geometry to be decomposed using parameters similar to the action of individual muscles, or groups of muscles. Such a technique is necessary, in particular for target-based models of synthesis, where raw geometry is used to define the changes in facial expression during speech production.

Chapter 4

Capturing and Retargetting Facial Motion

One of the most significant challenges in facial animation is the generalisation of techniques such that they are applicable across the entire population of faces. The shape, scale and structure of facial features will vary with sex, age, and ethnicity. Obviously, with the variety in facial morphology there is a corresponding diversity in the motions produced by individuals. For these reasons it is necessary to derive techniques which will not only be applicable to all types of face representation, such as those discussed in the previous section, but also to retarget captured motions such that they can be used to animate a whole range of different individuals.

Most animation is conducted using simple blends of acquired static expressions; these systems can be accused of not portraying the subtle motions inherrent in face-to-face communication. The linearity of the transitions between morph targets betray the synthesis even when the rendering of individual frames is highly realistic. In contrast physical models are computationally intensive and require detailed design. It is interesting to note that the most successful, as yet, computer-generated character in film is probably *Gollum* from the *Lord of the Rings* trilogy; this character was animated using a combination of artistic effort as well as motions captured from the actor Andy Serkis. Ideally, to streamline the process of animation, it would be beneficial to derive techniques which are almost exclusively automatic, although this may be some way off.

The facial retargetting problem is the analog of the similar group of techniques in full-body motion capture (e.g. [Gleicher, 1998].) Unfortunately, it would be impossible to apply the same techniques to faces as have been used in the case of articulated motion (e.g. [Bruderlin and Williams, 1995, Witkin and Popovic, 1995]), simply because of the differences between the underlying data. Whereas in the case of articulated motions the variation is held solely in limb length and joint angles, facial motion is often represented only as the motion of a cloud of points (possibly, but not necessarily, with associated topology.) Furthermore, full-body motion capture has been extensively reviewed and used in industry, and a large body of research has been carried out in the area, yet facial motion capture suites are only just becoming widely available and relatively few academic publications have demonstrated their use [Noh and Neumann, 2001, Williams, 1990].

This chapter presents novel techniques for the retargetting of captured motions to models with vary-

ing shape, scale and topology[1]. Section 4.1 is a discussion of techniques for the capture of human facial motion, further sections provide detail into the algorithms used for processing (Section 4.3), retargetting (Section 4.4) and animating meshes from a cloud of surface points (Section 4.5.)

4.1 Capturing Facial Motion

Most methods to capture facial motion rely upon Computer Vision algorithms to measure the motion of a surface projected onto the image plane of an optical camera [Williams, 1990, Cootes et al., 1998]. At the most trivial level this requires the extraction of markers placed upon the skin within each image in a sequence, and then inferring the motion of each point over time projected back into the real scene. The markers are often designed to be easily extracted in post-processing, either by using a colour which stands out against the skin (chroma-key methods), or by using materials which reflect a certain wavelength of light. Similarly, entire regions of the face can be coloured to enable them to be easily segmented. The seminal work in facial motion-capture [Williams, 1990] takes this approach. Also, in [Guenter et al., 1998] a large number of markers placed upon the surface of the face are used to capture soft deformation of the skin which along with a video-texture accurately capture an actor's performance.

These methods are consistent with the need to accurately capture the movement of the surface of the skin, whilst requiring least effort in post-processing. However, the highly error-prone process of preparing a subject for data capture is less than ideal. Considerable effort has been placed into the processing of image sequences to infer motion *without* the use of markings of any kind. These methods typically fall into the following categories: Optical Flow methods; Active Contours/Snakes; Active Appearance Models/Eigenfaces.

Optical Flow methods [Barron et al., 1992, Quénot, 1992, Horn and Shunk, 1981] use an optimal pixel alignment between successive frames in a sequence to infer the motion of the underlying object, in this case the motion of the face. These methods rely upon the following base assumptions:

1. The colour of each pixel remains constant across the entire sequence.
2. The illumination of the scene itself remains constant.
3. The inter-frame motion of pixels in the scene will be smooth, i.e. non-random in nature.
4. The motion of points across the surface of an object will be smooth.

Thus, given inter-frame flow by aligning successive frames, the motion of points within the scene will simply be the concatenation of these transitions. Optical flow also implies a dense field of motions at the same resolution as the images in the sequence, which is a far higher resolution than any marker-based method. Unfortunately, the above assumptions are extremely difficult to maintain in real scenes. Also, optical flow implies no structure in the motions which are captured, and thus driving animation from pixel flow is a difficult task requiring manual placement of the model within the scene. Nevertheless, Essa [Essa, 1995] has implemented a system for the animation of faces directly from the optical flow captured from a single camera. An example of detected optical flow using the method from [Quénot, 1992] is shown in fig. 4.1.

[1]The results of this chapter have been published in [Edge et al., 2004, Sánchez et al., 2003].

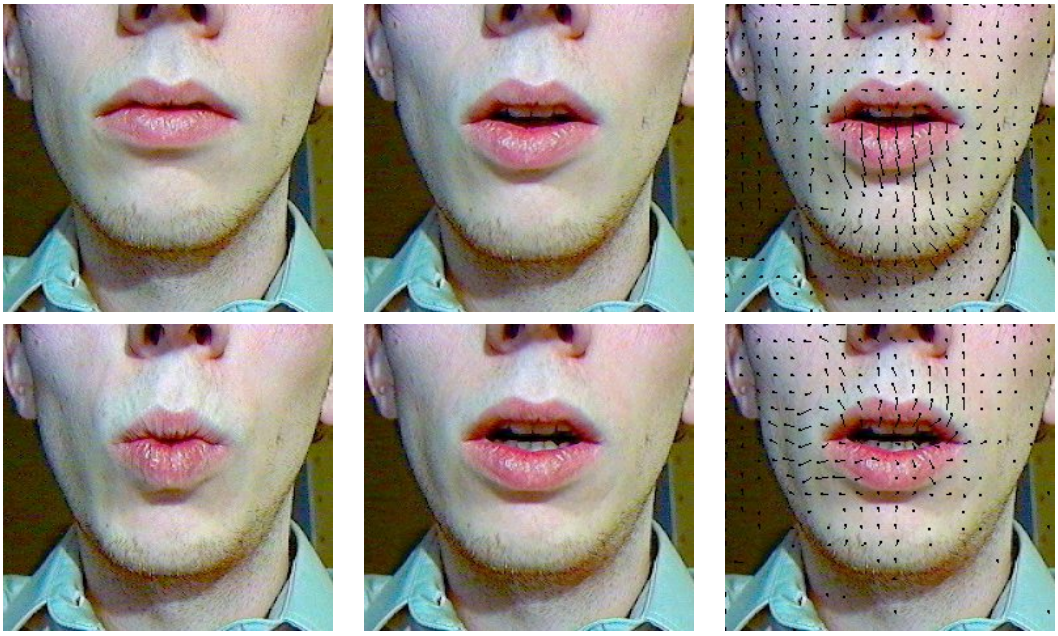


Figure 4.1: Optical flow captured between static images.

Figure 4.2: Active contour used to capture the outer lip contour. The bottom-right image shows an example of incorrect convergence that does not find the lip-contour.

Table 4.1: Comparison of vision-based tracking techniques.

OPTICAL FLOW	Advantages: • Markerless tracking. • Motion information can be measured to the same resolution as the image data.	Disadvantages: • Fails to track large scale motion. • Requires constant lighting conditions and textured surfaces. • Generated flow does not give structural information.
ACTIVE CONTOURS	Advantages: • Captures structural information from an image.	Disadvantages: • Reliant upon strong image gradients to isolate features. • Requires manual initialization close to desired features. • Can be attracted to groups of *weak* image features.
ACTIVE APPEARANCE MODELS	Advantages: • Specific to capturing facial expression. • Takes advantage of both texture and shape information • Generative model.	Disadvantages: • Requires large initial data capture. • Limited to capturing expressions which can be generated by the AAM.

Active Contours [Kass et al., 1988], also called Snakes because of the way in which they work, are a means of finding structural information within an image. Snakes are splines which either contract or expand to locate features within an image. Commonly, in the case of faces, they are used to track the motion of the lip contours and other stand-out features such as the eyes, nose and eyebrows. The behaviour of these models are defined by internal and external forces. Internal forces define the direction that the snake would naturally move in, should it find itself in a location where there are no features, i.e. the spline will either shrink to a point or expand to infinity. External forces define the features the snake should adhere to, e.g. strong image gradients as defined by a Sobel/Canny edge detector or regions of a particular colour.

Advantageously, snakes themselves imply structure within the image themselves. This is because each snake is located upon a salient image feature. By reinitializing snakes in subsequent frames the motion of the feature can be tracked over time. Unfortunately, the nature of snakes requires that they

are initialized close to, and in correct relation to the desired feature. For example, if an expanding snake is initialized outside of the desired contour it will expand to infinity. As a tracking method this can be problematic as large inter-frame differences can lead the snake to entirely miss the desired contour, or at worst locate a completely different feature. Snakes are only good at tracking clean image features and so either processing of the image to make features clearly distinguishable [Lievin and Luthon, 1999], or some form of marking of the actor's face, is usually required. An example of using snakes to determine the outer lip-contour can be seen in fig. 4.2.

Active Appearance Models (AAMs) [Cootes et al., 1998], like snakes, produce a model of the desired features and find an optimal match for that by traversing the image. AAMs produce this model from a database of samples of the desired features, e.g. images of faces. A statistical model is constructed using Principal Components Analysis[2] (PCA) for both shape and texture variation. The space of this model is then traversed as the AAM locates the most optimal location, orientation and internal parameters to describe the input image.

In order to track facial motion the AAM can be initialized from a sampling of the expected expressions, e.g. speech lip movements. AAMs are perhaps the most developed, state-of-the-art, method for markerless tracking currently available. Yet these models can still be unstable in the presence of noise, dropped frames and high frequency movement.

Capturing dynamic changes in facial expression can be a delicate process requiring precise and consistent experimental setup. The advantages and disadvantages of the described techniques are shown in table 4.1. No vision-based technique is perfect, and the requirements of the algorithms, along with the problem of accumulated error, usually prevent their practical use in production. For these reasons most commercial systems are usually based upon the placement of markers on the surface of the face.

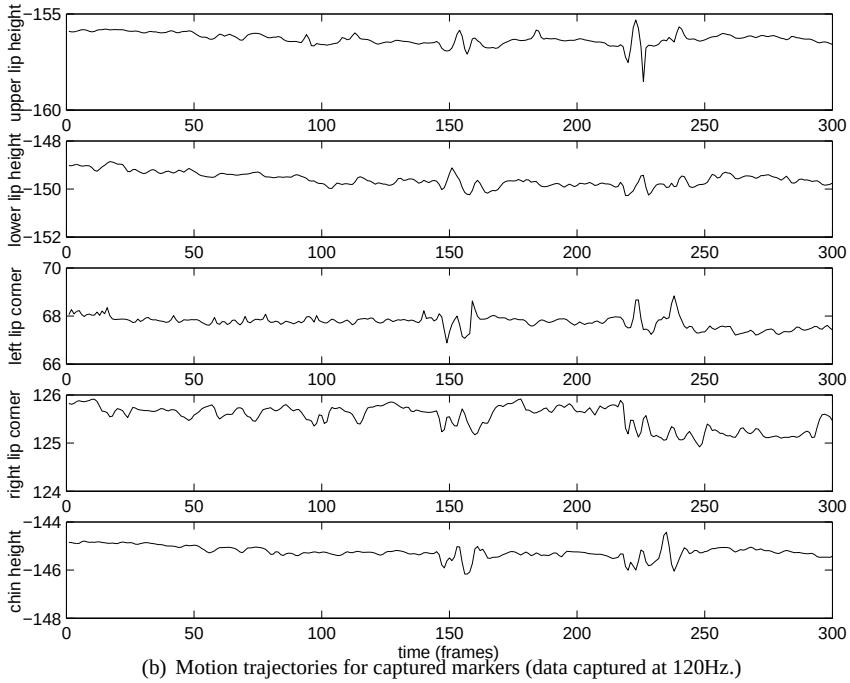
4.2 Facial Motion Data

The nature of facial motion data itself is intrinsically tied to the method by which it is captured. Most commonly this is a discretised sampling of the surface of an actor's face over time, often with no structure. The motion of each sampled point is a composite of both the articulated motion of the neck and the stretching of the skin by facial muscles over the boney substructure. Most methods can only retrieve the motion of surface points, and not the motion of the eyes, jaw or the tongue within the oral cavity. These features are either partially/fully occluded during capture, or it may be difficult to place markers at those locations (e.g. the inner lip contour.) The occlusion of markers or mis-registration may lead to significant parts of the motion being unavailable, and require that this is reconstructed using some data interpolation method. Furthermore, the motions may be noisy either due to error in the tracking or sensor noise. The final use of the data must take into account all of these factors.

Even given perfect noiseless tracking, animation requires that the course sampling of data points be interpolated across the surface of the target mesh. Any target mesh is likely to be far higher resolution than there are points in the motion-tracked data. The fact that the motion points define the surface of the subject's face implies that the deformation paradigm should be point or surface-based, and not volumetric (precluding, for example, FFDs.) Necessarily, the modelling technique should be able to fully

[2]PCA is generally used, although any similar statistical decomposition can be used. See Appendix A.2 for a description of PCA.

(a) Frames from captured motions.



(b) Motion trajectories for captured markers (data captured at 120Hz.)

Figure 4.3: Captured facial motion data (data courtesy of Scott King.)

define all possible expressions within the motion, which may preclude the use of physical or pseudo-physical techniques. In [Choe et al., 2001, Pitermann and Munhall, 2001] methods are described for driving physical models direct from motion-captured data. However, these rely either upon high detailed models of the subject's face, or simplistic projections onto the target model to allow the physical parameters to be determined.

Some example facial motion trajectories are shown in fig 4.3. From such data it is evident that facial motion is complex and highly non-linear. The motion of some points on the skin are highly rhythmical, particularly the movement of the lips during speech, whereas the motion of other points exhibit high frequency components. The use of such captured motions reduces the difficulty in simulating these complex trajectories (discussion of the synthesis of speech movements is described in detail in Chapter 5.)

4.3 Pre-processing Motion Data

Motion data, whether it is gathered from a subject using markers or marker-less vision algorithms, must be processed in order to make the data readily usable. Importantly any noise in the gathered signal must be removed, and the data must be rendered into a usable format. Here usable data is defined as containing only the facial motion of the subject, with no sections of missing data nor movement due to rigid motion of the head and neck[3]. The following sections detail the removal of noise and reconstruction of missing data using Kalman filtering techniques, and the estimation and removal of rigid head motion. The result of the pre-processing stage is data containing *only* the changes in facial expression captured from a given subject.

4.3.1 Removing Sensor Noise

The nature of motion capture technology is such that the resulting motions are frequently noisy and incomplete due to self occlusion amongst other factors. Given high frequency sampling of the point trajectories (e.g. the data used in this thesis is captured using high frequency cameras at 120 Hz), standard filtering techniques can be used both to smooth the motions and to recover missing data.

One common technique is to apply a Discrete Cosine Transform (DCT) to the data, and remove high frequencies that can be assumed to be the result of sensor noise. Using this method, missing data can be reconstructed by extending the sampling of the transform from neighbouring segments. Unfortunately, this method is highly sensitive to spurious spikes in the data. Spikes will induce significant distortion in the low frequency components of the DCT. The high frequencies compensate for this, and thus a low-pass filter can cause severe oscillations in the resulting trajectory.

The unsatisfactory results of applying low-pass filters to removing data noise requires the use of more sophisticated techniques. By conceptualizing the marker tracking as a stochastic process built around a linear model (approximating the motion equations of the markers), a Kalman filter can be applied to both smooth out noise and recover missing data.

Kalman filtering requires that a second order approximation of the position and velocity of a marker is constructed (4.1). In this equation x_i and $\dot{x}_i$ are the x components of the marker's position and velocity at time t_i, and Δt is the time interval between neighbouring samples.

$$\begin{bmatrix} x_{i+1} \\ \dot{x}_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \ddot{x}_i \tag{4.1}$$

The tuple $[x_i, \dot{x}_i]^T$ is the state vector used to estimate the actual position of a given marker. The second order term in (4.1), $[\frac{\Delta t^2}{2}, \Delta t]^T \ddot{x}_i$, is interpreted as the process noise (i.e. due to the difference between the model and the system being approximated.) This is assumed to follow a bidimensional Gaussian distribution with zero mean, and covariance matrix given in (4.2). In this covariance matrix, σ_a^2 is an estimate of the variance in the acceleration of markers, defined globally for the data.

$$Q = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t \end{bmatrix} \sigma_a^2 \tag{4.2}$$

[3]Rigid-head motion can be added back into the animation at a later stage, but for most purposes complicates the use of facial motion-capture data.

A separate noise term which must be taken into account is that involved in the actual measurement of the motion (e.g. due to tracking error.) As with the process noise, it is assumed to have zero mean and be Gaussian-distributed. However, the measurement noise is due to the nature of the capture devices, and thus the variance can be estimated empirically.

The Kalman filter defined in this way makes use of the linear equations from (4.1) to derive an estimate of the position of the marker, $\hat{x}$, so that the variance of an error function is minimized. That is, the difference between the estimate and the actual measurement is reduced for the defined specification of the system. The resulting trajectories satisfactorily discard the high frequency noise once the filter parameters are properly attuned. At the same time the estimate is driven by a dynamics model, thus preventing spurious spikes from being considered as part of the estimate; neither will they destabilize the estimate as seen with low-pass filtering of the DCT.

4.3.2 Estimation and Removal of Rigid Transformation

Facial motion gathered by motion capture systems may contain both the soft movement of the tissue under muscular influence as well as the rigid body motions of the head and neck [1]. For convenience it is useful to separate these motions thus allowing the animator more freedom to edit the motion of the head. In this work the rigid-body motions are estimated and separated from facial movement *before* any advanced processing of the mocap data (e.g. retargetting.) For the retargetting method described in Section 4.4.2 this is necessary because only small displacements from the surface of the skin can be accounted for. The estimated rigid movements can be re-applied to the model at a later stage.

Estimation of the translation and rotation transformations of the head can be fraught with difficulties when only the motion of points on the surface of the face are known. For this reason we use several points placed on a head-mounted jig to determine rigid movements. In order to determine the rigid transformation of the head from the, possibly noisy, location of these points a least-squares method is employed.

Consider the function $f_i(\Theta, T)$ as the estimate of the head motion for the i^{th} marker, given a rotation Θ and a translation T in three-dimensional euclidean space. For this marker we define the error in the estimated location in terms of the tracked point p_i (4.3).

$$error_i(\Theta, T) = \frac{1}{2}(f_i(\Theta, T) - p_i)^T (f_i(\Theta, T) - p_i) \tag{4.3}$$

Solutions for Θ and T can be found by minimizing (4.3) for the set of rigid markers, R, i.e. (4.4).

$$\textbf{minimize} \sum_{j \in R} error_j(\Theta, T) \tag{4.4}$$

It is important to define an adequate parameterisation in order to solve the least squares problem in (4.4). Rotation matrices could be used, and would allow the expression of the minimization problem in a linear form. However, the resulting transformation matrices would not necessarily represent the true transformation of the head. In fact it may not represent an affine transformation at all. This constraint can be properly enforced by using representations with less degrees of freedom; such as Euler angles or quaternions. Due to the singularities of Euler angles, quaternions are more appropriate and are used here.

[1] This, of course, may not be the case for head-mounted systems.

Solving the minimisation problem using quaternions requires only the constraint that they must be unitary. This can be enforced by penalizing factors in the minimization, or by explicitly imposing the constraint by means of the *exponential map* between $\mathbb{R}^3$ and the unit sphere S^3 in $\mathbb{R}^4$ [Grassia, 1998]. For each vector $r \in \mathbb{R}^3$ its mapping onto the unit sphere is defined in (4.5).

$$exp(r) = \begin{cases} [\sin(\|r\|)\frac{r}{\|r\|}, \cos(\|r\|)] & \textbf{where } \|r\| > 0 \\ [0,0,0,1] & \textbf{where } \|r\| = 0 \end{cases} \tag{4.5}$$

This reduces the constrained optimization problem to an exploration over $\mathbb{R}^6$ (for both rotation and translation.) Since the minima of the function is close to 0, a gradient descent method is used. The step length is computed by (4.6).

$$[r_{k+1}, T_{k+1}] = [r_k, T_k] + \delta_k \nabla error|_{\Theta = exp(r_k), T = T_k}$$

$$\nabla error = \left(\frac{1}{n}\sum_{i=1}^{n} J_{f_i}\right) J_{exp} \tag{4.6}$$

$$\delta_k = \frac{-error}{\nabla error (\nabla error)^T}$$

The initial estimate for the rotation, r_0, is computed as the average sum of the rotation observed in every pair of vectors defined by three non-colinear points in R. Given r_0, the initial translation, T_0, is trivial to compute.

The Jacobian of the exponential map, J_{exp}, when $\|r\| = 0$ has no analytical derivative. This can be remedied by using a Taylor expansion of $\frac{sin(\|r\|)}{\|r\|}$ for r with negligable value. Also, we find singularities as $\|r\| \to 2\pi$, and at multiples of 2π. In practice this is not a problem because there will not be extreme pose variations for the head.

The numerical method described in this section uses the properties of the minimization problem to compute the estimates without needing to evaluate second order derivatives. At the same time the method benefits from the filtering described in section 4.3.1.

4.4 The Retargetting Problem

The retargetting problem for non-articulated motions, such as the movement of the face, can be defined as follows:

Input: A set of source points at time $t \in [0,1]$, $X_t = \{x_0, x_1, \ldots, x_n\}$, where $x_i \in \mathbb{R}^3$.

Output: A mapping $\mathbb{R}^3 \to \mathbb{R}^3$ which projects the points in X_t onto a surface S, maintaining the relative positioning of the data points over time.

Obviously there are a number of conditions on S, the surface to which the motion is being retargetted. Most importantly, S must be *similar* to the object from which the original motion was gathered. Similarity, in this case, means an example from the same population; that is if retargetting facial motion the target surface should also represent a face, with the same structural idiosyncracies such as discontinuities and the general variation of curvature across the surface. Similarity is a condition which ensures that the motion *after* retargetting will be recognisable from the initial data.

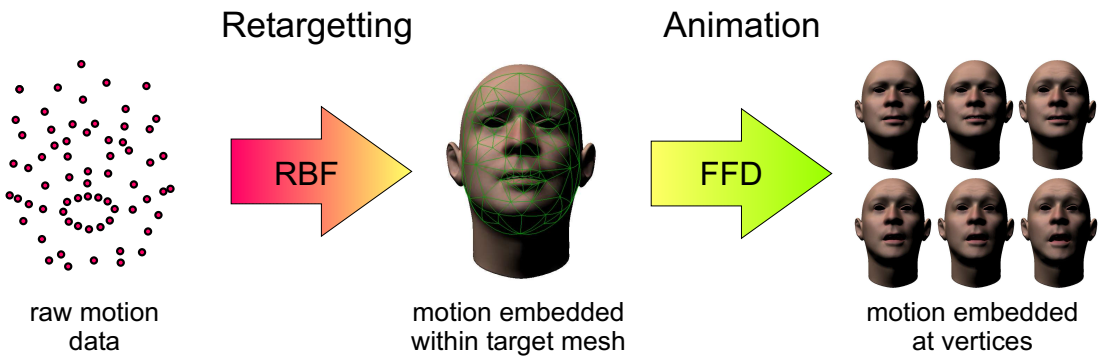


Figure 4.4: The retargetting process.

As the above problem only requires the definition of a mapping, non-similar meshes are not precluded. However, it is impossible to know how a motion would appear if retargetted onto an entirely alien object.

The required mapping should handle the composite of scale, rigid and non-rigid disparities between the source motion and the target surface. The rigid component, which represents the rotation and translation which would align the motion with the target surface, is to be removed. The scale and non-rigid components represent the differences in shape and expression over time and should be captured by the retargeting technique. As a result of mapping the motion each of the data points should be embedded within the target surface, and the motion of the points over time should maintain the relative motion in the original mesh. This condition ensures that the resultant motion reflects that of the captured data points, as though it was being created by the target face.

An overview of the method described here is shown in fig. 4.4. This consists roughly of two phases: firstly, *radial basis functions* are used to transform the motion data to the space of the target mesh; secondly, the markers are triangulated and a free-form deformation algorithm is used to interpolate the motion to each of the target vertices. This is an entirely geometric approach, requiring the placement of only a few points on the surface of the target mesh.

4.4.1 Previous Work

Overall, very little work has been conducted into the retargetting problem for facial motion, certainly when compared to the research into articulated motion capture (e.g. [Gleicher, 1998].) As with all models of facial activity, these methods can be distinguished into two areas: physically-based and non-physical/terminal analog methods. The seminal work in facial motion capture was conducted by Williams [Williams, 1990] into the use of marked points on an actor's face to drive expression animation. The animation was produced using a non-continous point-based deformation technique, where each marker deformed a local region using a so-called 'warping kernel'. This is similar to modelling techniques described in the previous chapter which define surface geometry directly in relation to the displacement of a few control points. Williams's technique relies upon the surface of the target mesh being identical to that of the actor's face (i.e. the surface was gathered using scanning technology), and performs no retargetting *per se*. However, the research demonstrated the feasibility of capturing a sparse sampling of the motion of an actor's face and directly using that data to drive an animation. In [Guenter et al., 1998] this approach is extended by taking several views of the scene and thus infering

three-dimensional geometric deformations, rather than the two-dimensional image plane deformations used by Williams.

The problems with Williams's approach lie in the non-continuous nature of the deformation paradigm, which is extremely evident in the results, and the lack of any retargetting strategy which would allow the data to animate models which do not directly conform to the surface of the original actor. In [Noh and Neumann, 2001] a technique is demonstrated which allows both problems to be solved. Using this technique, a motion embedded within a mesh is retargetted by determining a dense surface correspondance with a target mesh. Once a surface correspondence has been defined using a small set of user-defined surface correspondences, the source motion is embedded within the target mesh. Subsequently a number of rules are used to modify the motion vectors such that they correctly deform the target mesh:

- *Motion Vector Direction Adjustment* - The motion vectors are rotated such that they lie in the tangent plane defined by the surface normal at each vertex. This ensures that the motion occurs across the surface of the target mesh.
- *Motion Vector Magnitude Adjustment* - The motion vectors are scaled by the relative location of marked features between the source and target meshes. This prevents disparities between the scale of the captured motion and the scale of the target mesh from adversely affecting the resultant animation.
- *Lip Contact Alignment* - The contact line between the lips on the source and target meshes are aligned to preserve the discontinuous nature of movement in this area of the face. Without this step the upper and lower lips may not be able to move independently.

Furthermore, Noh describes a heuristic-driven algorithm to make this method fully automated. The disadvantage of this technique for the general purpose use of face motions lies in the base assumption that dense motions are embedded in a source mesh. It is a non-trivial step to apply the motion of a few control points across the surface of a dense mesh, and as Noh assumes this step has already been performed there is a large step missing from the technique.

In [Na and Jung, 2004] a similar approach is taken to that described within this chapter. A morphing approach is used to retarget the coarse motion, whilst high frequency details are retargetted by using normal disparities between the neutral expression and the frames of motion data which are subsequently imposed upon the target model. One of the major differences between Na's approach and the one described here is that Na requires that several expressions be modelled to define the correlation, whereas we use only one correspondence. A similar technique to that described by Na & Jung is found in [Pyun and Shin, 2003].

In [Joshi et al., 2003, Pighin et al., 1999] facial motion is tracked using combinations of morph-targets. However, this form of technique relies upon a significant capture effort to define the space of possible facial deformations before any tracking can take place. Tao and Huang [Tao and Huang, 1998] use FFDs to define a deformable model which likewise tracks movement in video. However, all of these techniques rely upon expensive optimization procedures to match live video or tracked markers to three-dimensional model states.

The methods described above exploit the geometric relationships between the motion and the target mesh to create animation. A significantly different group of techniques use physical models of the

skin to animate the mesh from the motion of the control points. Physics models of the physiological structure of the human face, based upon finite-elements or mass-spring networks, are used with muscle deformation parameters unrelated to the geometric structure of its surface. In order to correctly derive these deformation parameters for a given motion sequence, optimization procedures are required. In [Pitermann and Munhall, 2001] the Euclidean distance between nodes in the facial model and the motion-captured points is optimized over short temporal periods given constraints upon legal changes of system state. This method demonstrates that a physics-based model can be directly driven from geometric data, however the physical model is far from real-time and thus inappropriate for the majority of practical applications. A similar, albeit more simplistic, model based upon finite-elements is demonstrated in [Choe et al., 2001].

4.4.2 Retargetting Motion Data with Radial Basis Functions

The solution to the retargetting problem is a matter of defining a mapping from one set of points which vary over time, X_t, to their respective counterparts embedded within a target mesh. This problem can be seen as morphing the space of source points such that they lie upon the target surface; a volume transformation. One method for morphing volumes relies upon the use of *Radial Basis Functions* (RBFs, see Appendix A.1.1.) RBFs can provide a continuous mapping between two coordinate systems, in this case $\Psi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. The mapping function, Ψ, is defined as a linear combination of the basis functions ϕ (4.7).

$$\Psi(x) = p_m(x) + \sum_{i=1}^{n} \alpha_i \phi_i(\|x - c_i\|) \tag{4.7}$$

Each x is a member of the source data points, X_t. The mapped result should be embedded within the target mesh, reliant upon the correct choice of both the α_i weights and the basis centres c_i. These values are calculated by solving a system of linear equations (the details of constructing and solving the interpolant can be found in Appendix A.1.1.) This system requires the basis centres, and their mapped transformation onto the target mesh. A single frame of the motion captured data is used to provide the basis centres, and the transformed coordinates for the centres are identified on the target mesh. The chosen motion-captured frame must represent the same expression as the target surface, e.g. the neutral expression.

The RBF formulation allows for both rigid and non-rigid components, with p_m specifying the affine transformation from source to target. This allows the mapping to retarget motion data onto a grossly mis-aligned target surface, without any further user intervention. The polynomial term is the minimal requirement to at least align and scale the motion data to a surface with no further retargetting. The α_i weights carry the final non-linear component in matching the motion data to the target surface.

The described mapping function, Ψ, provides a continuous spatial transformation from the source data points (the motion data) to the labelled target surface. Thus, for relatively small[4] deviations from the basis centres, the mapping will retain the relative location of a transformed point. Applying Ψ to all points in X_t, for $t \in [0,1]$, the motion will be retargetted into the space of the target surface.

[4]The method is correct for small deviations, orthogonal to the surface constructed by the RBFs through the data points, on the scale of the source face. This is an adequate assumption for natural facial movement which does not to a great degree bulge outwards. The method will also be more accurate at regions of more concentrated sampling of the facial motion.

The Inverse Multiquadric (IMQ) RBF (4.8) is used here due to its global nature, particularly in comparison with the Gaussian. The global nature of the IMQ is reflected in the fact that it is C_∞ continuous, i.e. continuous in all derivatives. This is favourable because undulations in the spatial mapping will cause visually disturbing artefacts in the retargetted motion.

$$(x^2+\delta)^{-\mu} \text{ with } \mu > 0, \delta > 0 \tag{4.8}$$

The radius, δ, of each of the basis functions is defined as the minimum distance to a surrounding basis centre, i.e. $\delta_i = min(\|c_j - c_i\|)$ **where** $i \neq j$. A fixed value of $\mu = 2$ is used for the locality parameter, which defines the general shape of the IMQ.

The basic method described here for retargetting facial motion requires only the solution of an $(n+4) \times (n+4)$ linear system, where n is the number of data points in the motion data. Any common linear solver, such as Gaussian elimination, can be used to calculate both the α_i weights as well as the polynomial term at the same time. The scaling and rotation of motion vectors, as proposed in [Noh and Neumann, 2001], implicitly occurs as the three dimensional coordinates at each frame are retargetted. At this base level the implementation and use of the algorithm is straightforward. However, it is time consuming and error prone to manually label the transformed motion points on the target mesh. Incorrect relative placement of the markers can cause scaling and shearing of the motion. Automation in the labelling of the target surface removes a source of variability in the retargetting of motions.

4.4.3 Preparing the Target Surface

The retargetting technique described here relies upon the placement of markers on the target surface at equivalent relative positions to those in the original data. This is undoubtedly an error-prone procedure when performed manually. Ideally the markers should be placed fully or at least semi-automatically on the target surface. Not only does this reduce the effort required in retargetting the motion data, removing likely sources of error, but also aids in the repeatability of the method (i.e. retargetting the same motion onto the same surface should always produce at least a largely similar result.) To this end a semi-automatic method for labelling a target surface has been devised, requiring only the labelling of a few key points. The method described here lowers the user workload in retargetting motion data from tens of points[5] to around ten points at easily identifiable locations on the facial surface.

In order to correctly locate the position of data points on the surface of a target mesh several steps are performed:

- *Locate Key Feature Points* - A number of fiducial points are manually placed by a user, located at key features on the face. The tip of the nose, eye corners, and the apex of the chin make good locations for fiducials.

- *Simple Mapping and Projection* - Using the fiducial points and their counterparts in the original data a simple mapping is performed to align and scale the motion with the target surface. The mapped motion points are projected such that they are embedded within the target surface.

- *Triangulation of Data Points* - The data points are triangulated to facilitate the energy minimization phase and later the deformation of the target mesh.

[5] Much of the data these techniques have been tested upon hold in the region of 80-90 data points.

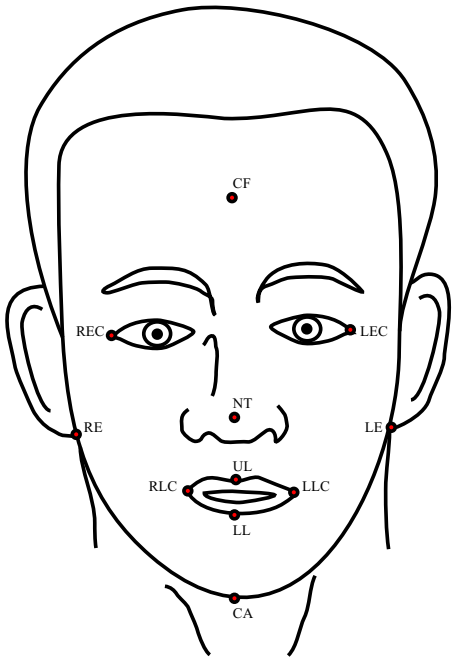


Figure 4.5: Fiducial points used in automatically labelling the target surface.

- *Energy Minimization* - Given the starting point from the previous stages of the registration process, a final optimization of the marker locations is performed. This relies upon the deformation of the mapped points in accordance with an energy function which maintains both their relative structure and the similarity with the original data.

The manually-placed fiducial points represent key features on the face. It is important that these are easily and accurately identifiable for the following stages of the mesh registration process to produce good results, and also to enable the process to be repeatable. The following fiducials are used for these reasons as well as the consistent coverage across the surface of the face that they provide: Centre Forehead (CF); Chin Apex (CA); Nose Tip (NT); Upper Lip Centre (UL); Lower Lip Centre (LL); Right/Left Lip Corner (RLC/LLC); Right/Left Ear (RE/LE); Right/Left Outer Eye Corner (REC/LEC). These fiducial points are shown in fig. 4.5. Should the technique not produce the desired results with this set of fiducials, there is the option of supplying more fiducials and thus further constraining the later stages of the registration process. However, for most purposes the above set are adequate, and furthermore correspond to a subset of important anthropometric features [Farkas, 1994].

Next a simple mapping between the labelled fiducials and their corresponding data points in the original motion is created, again using RBFs. Here we are essentially performing the same calculation as with the final retargetting to move all the motion data points into the space of the target surface. This is *not* sufficient to label the target surface because the interpolation formed by only a few (approximately 12 fiducials) will not accurately reflect the structure of the target surface. The mapped points will not necessarily lie embedded within the surface, apart from those corresponding to the fiducials themselves, and thus further steps must be taken to ensure that the motion points have been correctly placed in relation to the target surface.

The projection is a transformation from euclidean coordinates into cylindrical coordinates (4.9) followed by a projection onto the target surface in the same coordinate system. The projection takes the

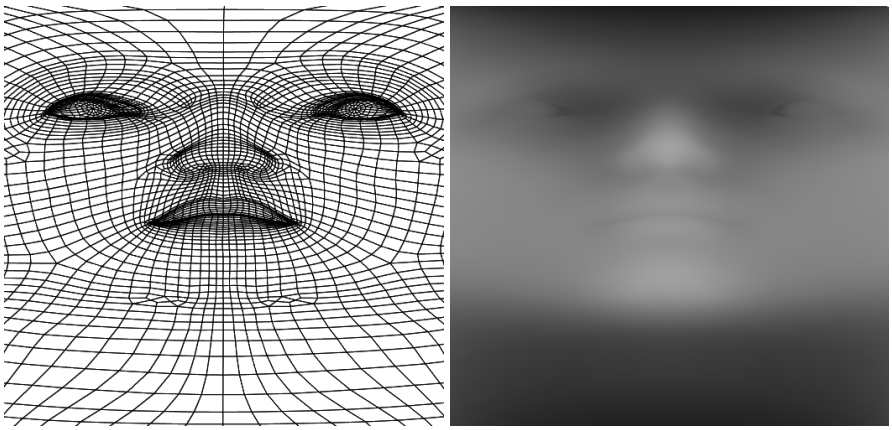


Figure 4.6: Cylindrical projection of a target mesh and interpolated depth coordinates.

form $cylind : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, and simply swaps the depth coordinate, r, of the mapped point with that of the target surface at the same elevation/angle coordinate, $\{\theta, y\}$. Interpolated depth coordinates are used where no target depth coordinate is coincident, which can be optimized by rendering depth coordinates into a texture map and performing lookup queries directly on the texture (see fig. 4.6.)

$$cylind : \{x,y,z\} \rightarrow \{\theta,y,r\} \quad \begin{array}{ll} \textbf{where} & r = \sqrt{x^2+z^2} \\ \textbf{and} & \theta = \tan^{-1}(\frac{z}{x}) \end{array} \tag{4.9}$$

Once the mapped points are cylindrically projected onto the target surface, they will remain embedded within that surface throughout subsequent stages. Unfortunately a cylindrical projection can lead to the mis-placement of data points, i.e. it will lead to a degradation in the similarity between the structure of the initial motion data and the labelled points on the target surface. To rectify this an optimization procedure is used to deform the points so that they move to the correct relative positions whilst remaining embedded within the target. The previous stages are present to ensure that the points are close enough to their 'correct' placement that the optimization finds the global minima.

The global energy term, E_{mesh}, is a sum of three terms (4.10): E_{dist} which pulls the data points so that they are embedded within the mesh, E_{strain} which maintains the relative location of each data point in respect to its surrounding neighbours, and E_{bend} which pulls the data points to a solution with similar curvature to the original motion data.

$$\textbf{minimize} \quad E_{mesh} = \alpha E_{dist} + \beta E_{strain} + \gamma E_{bend} \tag{4.10}$$

This can be seen as an analogous approach to the use of three-dimensional snakes (i.e. active surfaces [Xu and Prince, 1997]) to locate the correct structure of data points embedded within the target surface. The internal energy term, E_{int}, consists of the combination of bend and strain, i.e. $E_{int} = \beta E_{strain} + \gamma E_{bend}$. The external energy term, E_{ext}, consists only of the distance term ensuring that the data points lie embedded in the surface, i.e. $E_{ext} = \alpha E_{dist}$. The weights, $\{\alpha, \beta, \gamma\}$, are tuned to transform the range of each of the terms such that no one energy term dominates. The three energy terms are defined in (4.11). The terms require the data points to be triangulated, which is also necessary for the deformation algorithm and is discussed in Section 4.5.

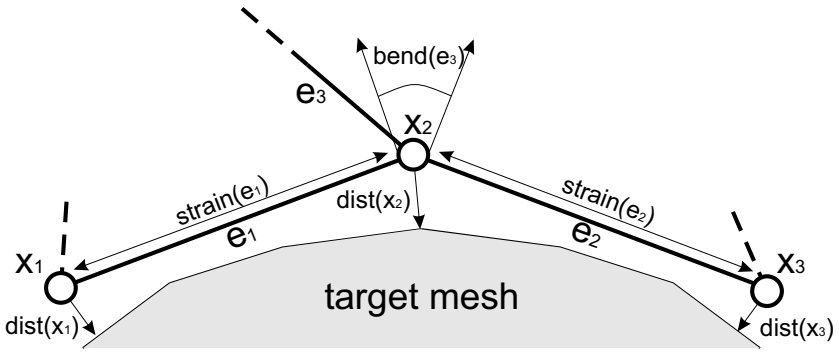


Figure 4.7: Fitting a control surface to a target mesh.

$$E_{dist} = \frac{\sum_{i=1}^{n} dist(x_i)}{n}$$

$$E_{strain} = \frac{\sum_{i=1}^{n} \|len(e_i') - len(e_i)\|}{n} \tag{4.11}$$

$$E_{bend} = \frac{\sum_{i=1}^{n} \|bend(e_i') - bend(e_i)\|}{n}$$

The equations in (4.11) rely upon the definitions of the functions *dist*, *len*, and *bend*, which all return a scalar value. These functions take either an edge, e_i, defined as a pair of mapped data points and related faces, or an individual data point, x_i. The edges e_i represent the state after mapping and projection, whilst the e_i' represent the current state in the optimization. The function *dist* is defined as the shortest distance from a data point, x_i', to the surface of the mesh, *len* is the length of a given edge, and bend is computed as the angle between the normals of the faces adjacent to an edge. Figure 4.7 shows the features related to the optimization process.

The optimization procedure used is the Downhill Simplex (DS, see appendix A.3.1) method. DS is used because it does not require explicit derivatives to find the minima of a function. As the optimization landscape is a complex one, for which we have no analytic definition, blind methods such as DS are the only ones available. However, DS is workable in this situation as the user-placed fiducials will have located the mapped points close enough to the minima that DS is likely to be successful. The $3n+1$ dimensional simplex, where n is the number of data points in the motion to be retargetted, is initialized with vertices corresponding to the mapped data points translated by a constant value, λ, along each coordinate axes, i.e. along each of the axes of each of the n data points. The value of the initial offset, λ, is calculated as a fraction of the largest dimension of the target surface. This prevents DS steps which pull the data points far away from the target surface and away from the desired minima.

The result of the three stages: manual location of fiducials; mapping and projection; and finally energy minimisation is the optimal placement of data points from the original motion onto the surface of the target mesh. These located points are now used as the target points in providing the mapping used to retarget the complete motion. The method requires only the labelling of a few points and thus is at least as time efficient as the basic method described in [Noh and Neumann, 2001]. The fully automatic retargetting which Noh mentions relies upon heuristics to identify the fiducials on the target surface. A

similar approach could be used here. However, heuristics are not very consistent in identifying fiducials accurately and for this reason the method retains an element of user interaction.

4.5 Animation from a Cloud of Points

The result of the described retargetting method is the motion of points on the surface of the target mesh. These points are not attached to the target mesh, and thus a technique must be applied to transfer the motion from the sparsely sampled data points to the densely-sampled vertices of the target mesh. This implies an interpolation of the motion-captured points across the surface of the target mesh to create the final animation.

Techniques to interpolate the displacement of a few control points across an object have already been described in Chapter 3. These are free-form deformation techniques that either deform the space in which an object is embedded (e.g. [Sederberg and Parry, 1986]), or provide a mapping between the object and a number of deformer primitives (e.g. [Lazarus et al., 1994, Singh and Kokkevis, 2000].)

In the case of facial motion-capture (i.e. as described in Section 4.2), the free-form deformation primitives represent a sparse sampling of the facial surface. Thus, intuitively, the deformation technique should use a surface as the deformer primitive. This surface should span the control points, yet should be capable of maintaining discontinuities (particularly the lip contact line) in the target mesh. The techniques which best match these criteria are planar-element FFDs [Singh and Kokkevis, 2000, Sánchez and Maddock, 2003], and patch-based FFDs [Sánchez et al., 2004] (for details of both techniques see Section 3.2.2.) The major difference between the two is that the former technique requires discontinuity masks to be generated for the target mesh, whilst the latter technique implicitly maintains discontinuities.

To derive an FFD structure to control the target mesh a triangulation procedure must be defined. Delaunay triangulation can be used, however it must be constrained to maintain a close fit and topological similarity with the target mesh. In the absence of constraints no discontinuities will be present in the control surface. Also, the Delaunay method leads to convex hull-like triangulation of control points on the nose and cheek, and thus constraints must be applied to prevent this. Details of the application of constrained Delaunay approach for deriving an FFD control surface can be found in [Sánchez et al., 2004].

Whilst the majority of the motion will be evident in the data itself, there are usually points which were not, or could not be, captured initially. This is particularly the case for the lip contour line, where the sorts of markers used in motion capture systems cannot be placed. In these case relationships with surrounding markers can be used to reconstruct the missing data. The movement of the lips involves a complex physical deformation, yet the motion of the inner contour can be adequately modelled with offset vectors from the outer contour. A more complex model could be constructed, but would require the modelling of lip contact deformation, which would impact upon the real-time nature of the animation.

4.6 Results

Frames from an animation[6] are shown in figures 4.8 and 4.9. Retargetting produces realistic motions when the target mesh resembles a human face, i.e. the target has the same general features in the same general structure. The process from capturing a new motion to animating a mesh is short requiring only the labelling of a few points on the target mesh. Also, as the deformation is performed using a geometric FFD algorithm, animation is real-time and non-specific to any target mesh (as would be the case with a physics-based model.)

Figures 4.8 and 4.9 also demonstrate the use of bump-mapping to add fine detail into the model. By fading in wrinkles according to the compression of the deformer structure high frequency details can be added to the animation. This technique is discussed in detail in [Sánchez et al., 2004].

4.7 Summary

This Chapter has discussed the process of capturing and processing motion data, and also introduced a novel method for the retargetting of motions to animate meshes. Numerous methods have been proposed for capturing human facial motion, but most extract a sparse sampling of the motion of points on the surface of the skin[7]. These motions are often noisy, and contain rigid head motion which complicates its use. Processing is required to extract the motion of markers in a form that can be used for animation. In Section 4.3 commonly used techniques are described for the processing of this data.

Given the raw motion of a set of markers, retargetting is required to transform them such that they can be used to drive a mesh which may vary in both shape and scale. RBFs can be used to warp the space of the original motion to coincide with that of the target mesh. This relies upon correspondences between a frame in the original motion and the target mesh. These can either be manually labelled or semi-automatically positioned according to the placement of a small number of fiducial points. The retargetting is performed by simply evaluating the spatial warp for each frame of the source motion, no further processing is required. The retargetted motion will exhibit the same relative motion of markers as the original captured data.

To animate the target mesh the motion of markers must be interpolated to displace individual vertices. A surface-based FFD technique is used for this (both planar- [Sánchez et al., 2003], and patch-based [Edge et al., 2004, Sánchez et al., 2004] deformers have been used.) Deformer primitives span the markers in the retargetted motion data, and thus as the markers/control points move the attached vertices are displaced. The surface-to-surface mapping provided by these techniques is an intuitive way to map the sampled motion of points on an actors face onto the target geometry.

In Section 6.4 the retargetting technique from this chapter is used as part of a limited-domain concatenative visual-speech synthesis system. The advantage of using a retargetting technique, such as the one described here, is that motions captured once can be used to animate many virtual characters. Given the difficulty and expense involved in capturing high quality facial motion, maximizing its use post-capture is important.

[6] Animations demonstrating the retargetting technique can be found in the folder ‘animations/section_4.6/‘ on the accompanying CD.

[7] Notable exceptions to this attempt to capture the facial surface itself [Zhang et al., 2004, Tibbalds, 1998]. Although the results are excellent, storing facial geometry for each frame is highly inefficient in storage, and the results are specific to the original actor.

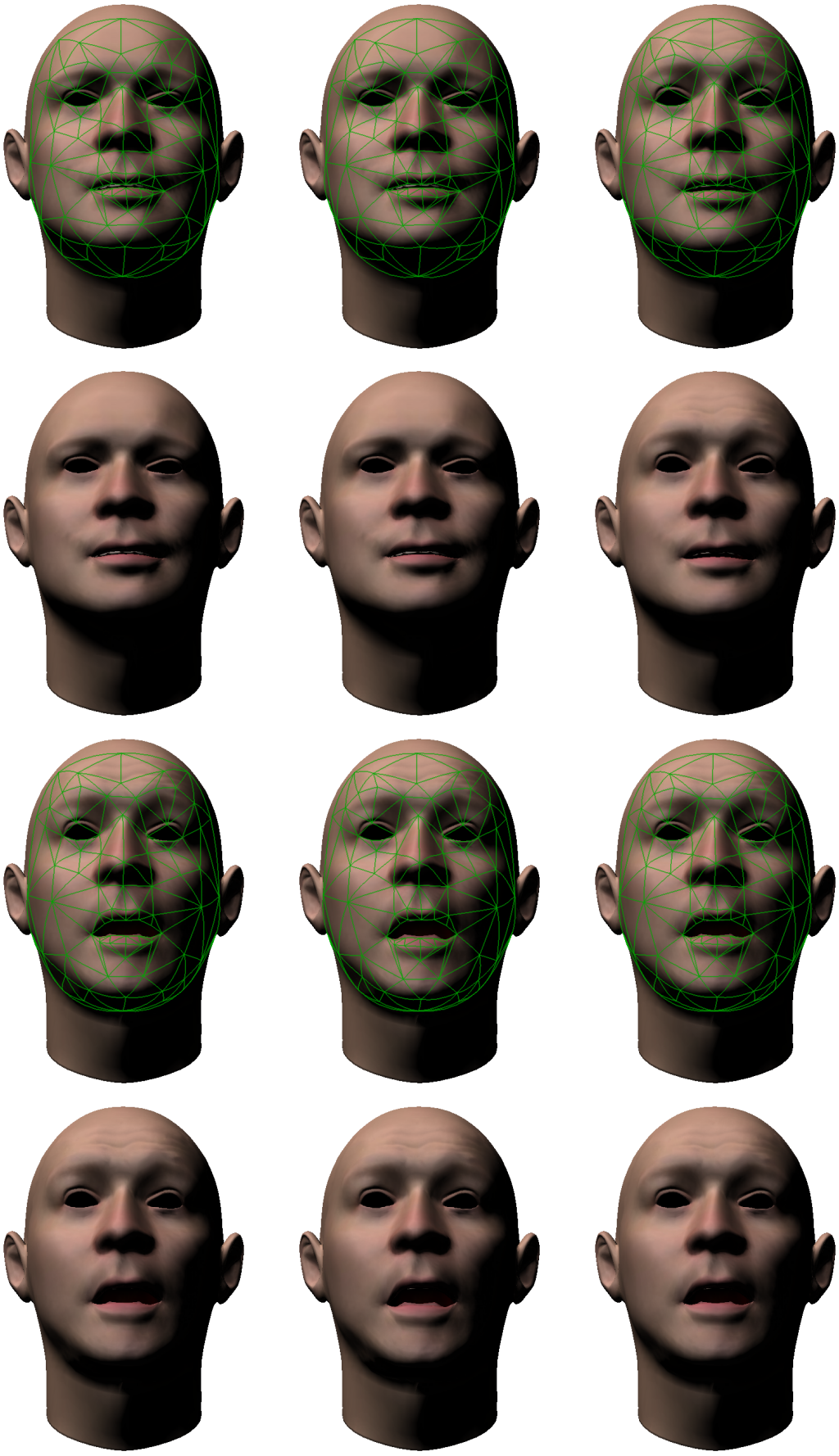


Figure 4.8: Frames from an animation showing control mesh (in green) and rendered mesh.

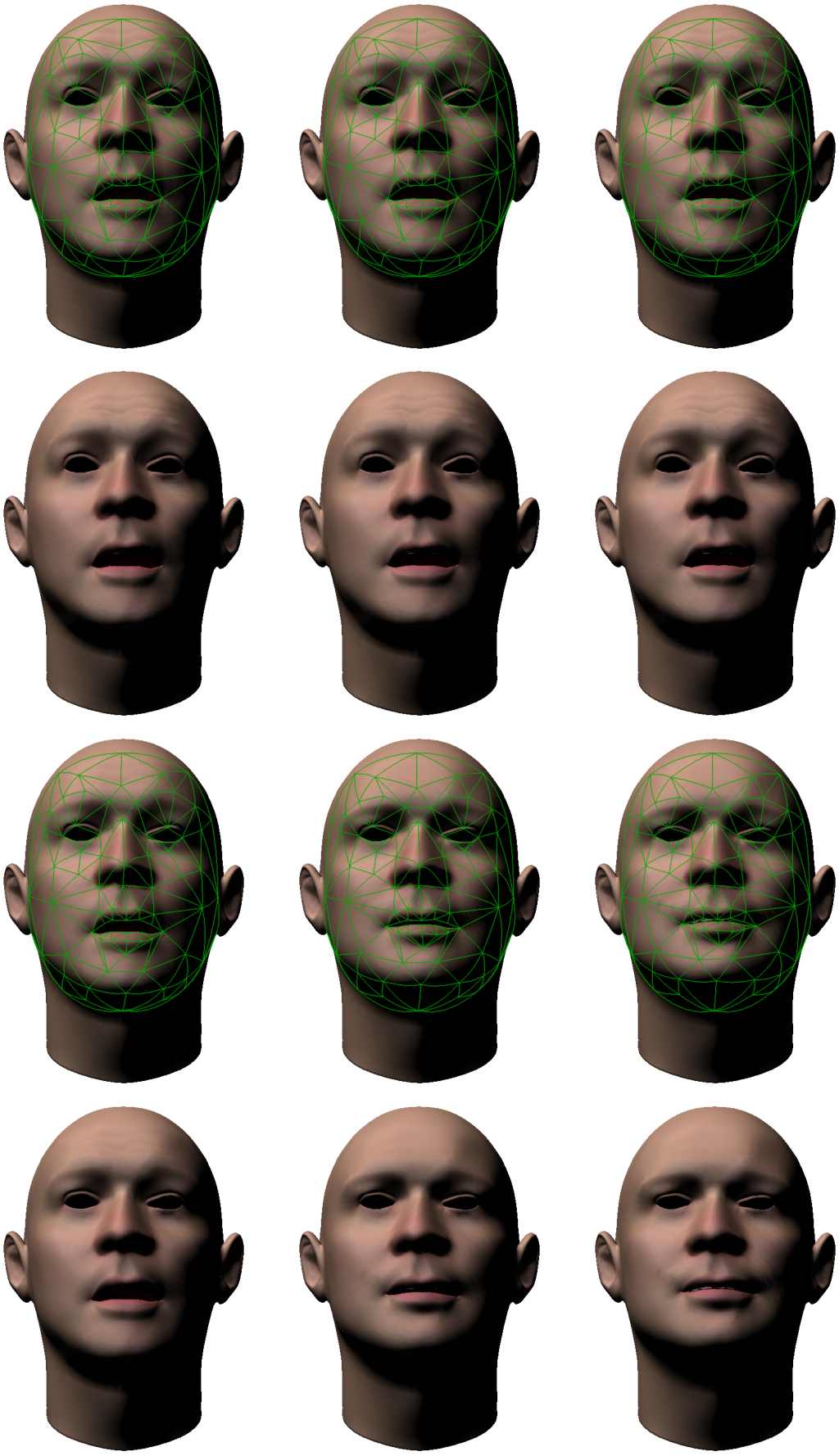


Figure 4.9: Frames from an animation showing control mesh (in green) and rendered mesh.

Chapter 5

Animating Speech

Speech production can be considered as a process balancing the physical properties of the articulators (lips, jaw, tongue etc.) and a set of ideal objectives. These objectives may be phoneme/visemes, context-dependant allophones, or larger units such as syllables. The size and nature of these targets is a matter of debate and a number of theories have been put forward [Löfqvist, 1990, Kent and Minifie, 1977, MacNeilage, 1970, Wickelgren, 1969, Öhman, 1967] (see Section 2.1.5.)

In order to synthesize the visual extent of speech articulatory movements it is necessary to take these theories and produce generative models[1] which given a target utterance, usually defined by its phonetic timing, produces trajectories for the vocal articulators. These trajectories are time-varying parameters defining properties such as: lip width, tongue protrusion, jaw rotation etc. The parameterisation of speech articulators is dealt with in detail in Chapter 3. The following sections shall deal with the generation of speech trajectories generically, that is with no reference to particular articulatory parameters. Thus the methods described here could be used with anatomically inspired parameter sets (e.g. FACS), physical parameters (e.g. muscle forces), or parameters relating to facial geometry (e.g. surface control points.) The methods described here are implemented in several systems described in detail in Chapter 6.

The main problem which must be accounted for in speech synthesis is the resolution of coarticulation, the effect of context upon speech movements. Generative models of coarticulation can be split three ways: target-based models, motion-based models, and finite-state models. The first two represent the main thrust of work into speech production, i.e. static phonetic units vs. dynamic units (e.g. syllables.) In comparison HMM/neural-net models generate a visual signal directly from an audio signal. Section 5.1 discusses prior work in the field, and the classification of visual-speech synthesis systems. In Sections 5.2 and 5.3 the use of dominance functions and optimization approaches to generating speech trajectories are discussed in detail. Finally, Section 5.4 discusses a contrasting approach to synthesis, that of concatenating pre-captured speech movements. The approaches taken to generating speech trajectories using optimization techniques, and concatenating motions are significant novel contributions of this thesis.

[1] This is the case for Text-To-Visual-Speech-Synthesis (TTVS), however, Audio-To-Visual-Speech-Synthesis (ATVS) systems typically do not rely upon speech production theory [Ezzat et al., 2002, Brand, 1999].

5.1 Previous Work

Methods for generating speech trajectories in TTVS systems can be split into several broad categories:

- *Target-based* - where a speech trajectory is generated between several distinct static targets (usually visemes.) The synthesis technique models the rôle of coarticulation in transitions between speech targets. The most simple of these methods directly interpolate targets, and so do not model coarticulation at all with an accompanying loss in naturalness [King et al., 2000] [Ezzat and Poggio, 1999, Kulju et al., 1998].
- *Motion-based* - where a selection of motion units are concatenated to generate trajectories. These are analogous to the concatenative methods in audio synthesis (see Appendix B.3.)
- *Model-based* - where a model is generated from captured speech motions relating speech audio to generated trajectories.

Target-based models are the most common in the animation/synthesis community. This is most evident with the approach in [Cohen and Massaro, 1993] where a number of dominance (basis) functions are used to generate a trajectory between viseme targets. Essentially the dominance functions act like basis functions for a spline. This method has been implemented in [Cosi et al., 2003, Albrecht et al., 2002, Breton et al., 2001, King, 2001, Le Goff and Benoît, 1996], and is discussed in depth in Section 5.2. It is important to note that there are other models which relate a time-varying dominance to the generation of a trajectory from static targets [Bui et al., 2004, Fagel and Clemens, 2003, Ezzat et al., 2002, Revéret et al., 2000] and these in effect are all implementations of the ideas in [Löfqvist, 1990] (and by extension [Öhman, 1967].) In [Waters and Levergood, 1993] a similar target-based approximation is defined using a physical system of nodes and springs which are used to find the motion of points on the face over time.

Motion-based models take real-life data relating to the articulation of natural speech, decompose the data into units (e.g. syllables, words etc.), and use combinations of these units to generate natural speech. An example of this is the Video-Rewrite model [Bregler et al., 1997], where segments of video representing triphones are concatenated together. In [Kshirsagar and Magnenat-Thalmann, 2003] segments of motion-captured data representing visual-syllables (i.e. the visual component of a syllable in the same way that a viseme is the visual component of a phoneme) are concatenated to perform global-domain synthesis. Similar techniques can be found in [Cao et al., 2004, Huang et al., 2002, Bulut et al., 2002]. The disadvantage of these techniques lies in the size of database required to perform synthesis, as the data must capture all variations in the target domain. Possible units for synthesis ordered in increasing size include: phones, diphones (phoneme-to-phoneme transitions), triphones, demisyllables (half-syllables split at the central vowel), syllables, words, and phrases. As the unit size increases so does the quality of the synthesis, as there are less *synthetic* transitions, but the size of the database increases exponentially. For comparison consider the number of diphone transitions in British English, numbering in the low thousands, versus the number of syllables, numbering in the tens of thousands - thus, the use of syllables requires significantly greater time in data-capture, labelling and other preparation before any synthesis can occur. Visually there will be less perceivable units in natural language, yet the greater difficulty in accurately capturing the movement of the articulators more than makes up for this (see Chapter 4.) In Section 5.4 techniques required to implement motion-based synthesis are discussed

in detail. Other models in this group include [Arslan and Talkin, 1998, Hällgren and Lyberg, 1998, Henton and Litwinowicz, 1994].

Model-based synthesis builds an inverse[2] relationship between speech audio and articulatory motion, and thus given a novel source of speech audio (either natural or synthetic speech) a trajectory can be generated. In order to capture this relationship Hidden Markov Models (HMMs), finite-state machines with probabilistic transitions, are trained upon databases of recorded speech audio and movements. A number of systems based upon this method have been reported [Williams and Katsaggelos, 2002, Angelfors et al., 1999, Brand, 1999, Brooke and Scott, 1998, Tamura et al., 1998], which mainly vary in the structure and training of the HMM. Neural networks have been used to similar effect in [Massaro et al., 1999, Eisert et al., 1997, Frank et al., 1997, Lagana et al., 1996]. Other models which can be attributed to this group include [Kshirsagar and Magnenat-Thalmann, 2000, Lewis and Parke, 1987]

5.2 Target-based Synthesis using Dominance Functions

The most common technique for the synthesis of speech movements is analogous to target/feature-based models of coarticulation [Löfqvist, 1990, MacNeilage, 1970, Öhman, 1967]. In these methods static target feature sets, representing individual visemes, are approximated using various methods. Here the word *approximated* is used to represent the fact that coarticulation cannot be implemented using an interpolating scheme. The targets will most likely not be met, and thus a synthesis technique is alike an approximating spline (albeit a complex and highly parameterized one) where the control points are the relevant target features. Several schemes have been proposed [Cosi et al., 2003, King, 2001, Revéret et al., 2000, Le Goff and Benoît, 1996, Cohen and Massaro, 1993], however, all can be traced back to Löfqvist's general model [Löfqvist, 1990] (and further back to [Öhman, 1967].) The basic model has already been described in Section 2.1.5.1. Below is a more thorough description of the method, its properties, advantages and disadvantages.

The basic equation, defined by Cohen and Massaro [Cohen and Massaro, 1993], implementing Löfqvist's model of speech production is found in (2.2) and (2.3) (reproduced below for convenience in (5.1) and (5.2).)

$$D_{sp}(\tau) = \begin{cases} \alpha_{sp} e^{-\Theta_{\leftarrow sp}|\tau|^c} & \tau \geq 0 \\ \alpha_{sp} e^{-\Theta_{\rightarrow sp}|\tau|^c} & \tau < 0 \end{cases} \tag{5.1}$$

$$F_p(t) = \frac{\sum_{i=1}^{n} (D_{sp}(\tau_i) T_{sp})}{\sum_{i=1}^{n} D_{sp}(\tau_i)} \tag{5.2}$$

In (5.1) a negative exponential[3] dominance function, D_{sp}, is defined which controls the temporal extent (influence) of a segment s (more specifically viseme target) over a particular parameter trajectory, p. The coefficients $\{\alpha_{sp}, \Theta_{\leftarrow sp}, \Theta_{\rightarrow sp}, c\}$ determine the shape of D_{sp}. A combination of these is used to weight the contribution of each viseme over the final parameter trajectory F_p (5.2). Thus, the final trajectory can be thought of as a type of approximating non-uniform rational C^0 spline: *approximating*

[2]The relationship is inverse because the audio drives the speech movements. This is the opposite of the causal relationship between the physical movement of the articulators and the resulting speech waveform.

[3]Most authors use negative exponential functions to model the temporal influence of a segment. However, Cohen and Massaro [Cohen and Massaro, 1993] propose that different dominance functions could be used to model specific properties of speech trajectories (although there is no mention of how to do this.) Löfqvist's original proposal [Löfqvist, 1990] does not make any particular claims as to the shape of these functions.

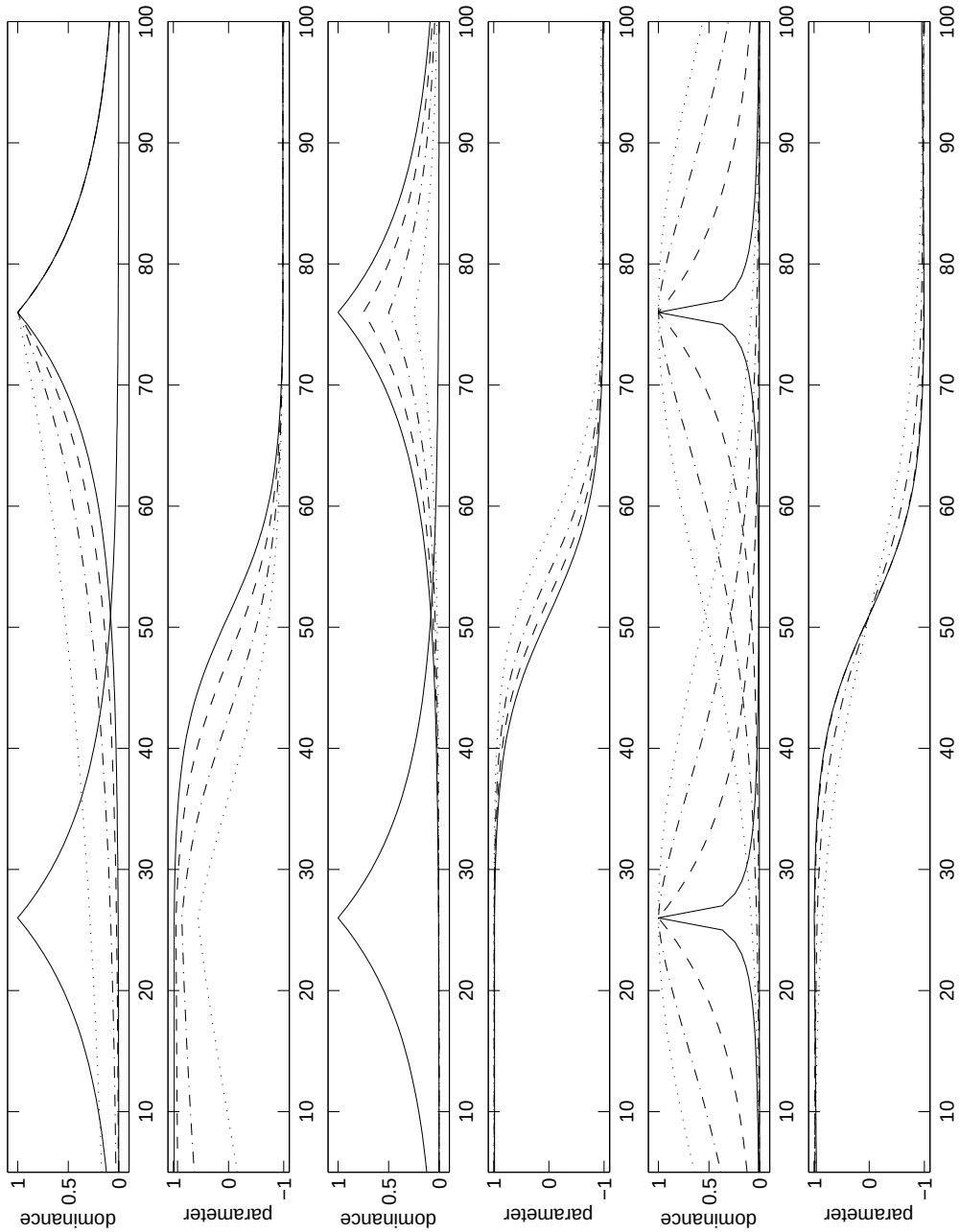


Figure 5.1: Dominance functions (after [Cohen and Massaro, 1993]) and the effect of coefficients upon parameter trajectories. Each trajectory is generated from a combination of two targets: $\{T_{sp1} = 1, t_{sp1} = 25\}$ and $\{T_{sp2} = -1, t_{sp2} = 75\}$. From top to bottom (each a dominance/parameter trajectory pair): varying $\Theta_{\rightarrow sp}$, varying α_{sp}, varying c.

because, as implied by coarticulation, the trajectory does not pass through all of the targets; *non-uniform* because the targets may occur at arbitrary intervals, as specified by the phonetic timing of the utterance; and *rational* because the coefficient α_{sp} defines the degree to which the target is approximated (or, at extreme values, whether it is interpolated.) This formulation leads to a number of observations:

- By the described formulation *only* C^0 continuity can be asserted. This leads to problems in realising physical properties of articulatory movements, for example the onset/offset characteristics of muscular contractions [Fung, 1993]. Furthermore, as there is no control over higher derivatives it is impossible to assert, for example, directional control over lip movements (e.g. forcing the lips to be moving apart.)

- The degree to which a target is realised in the dominance function approach is entirely a function of context. For example, in no context a target will be met entirely[4], and by adding more targets with overlapping influences the original target will be less well met. This can lead to problems where the context in which a segment finds itself will prevent that target from being met sufficiently for audio-visual fusion (visually the articulatory movements contradict the audio.) This has been found to be the case in particular for easily recognised visemes (i.e. those which are strongly dominant), e.g. bilabial plosives [Le Goff and Benoît, 1996]. Infinite dominance, that is an absolute guarantee that a target will be interpolated, does not exist in this model.

- In the dominance function, model parameters which affect the resulting speech trajectory are bound with the visemes themselves. This implies that the physical properties of speech are not due to the physical system itself (muscles, skin etc.) but by the placement of the targets in an utterance. If two contradicting, and equally dominant targets are moved increasingly closer together, until they virtually coincide, they will cancel each other out. Thus, a higher level planning process must exist, and must have knowledge of which targets can exist in which context and also the allowable proximity of those targets.

- This model does not define speech as a displacement from a neutral state. For this reason silence itself is considered a *target* and has its own dominance function. That silence has an influence over speech movement is a strange concept, given that silence is what occurs when there is no speech movement. This is due to the weighted combination (5.2) used to find the final parameter trajectory, F_p, which implies that before the beginning of the utterance and after the end the trajectory will tend to the first and last targets respectively (i.e. not to 0 or the neutral expression - which for many parameterisations will be the same.)

- There are no global parameters to control articulation. This prevents the modelling of speaker-independent characteristics (e.g. degree of articulation and speech rate.) In order to model the degree of variation in speech movements would require the parameters of the model to be modified.

- It has been reported that coarticulation only occurs over periods of up to seven segments [Benguerel and Cowan, 1974], and usually far less. However, (5.2) is a summation for all segments in an utterance, and thus contributions may be occuring over longer durations than are

[4] In fact in no context the trajectory will be static because there are no contradicting targets.

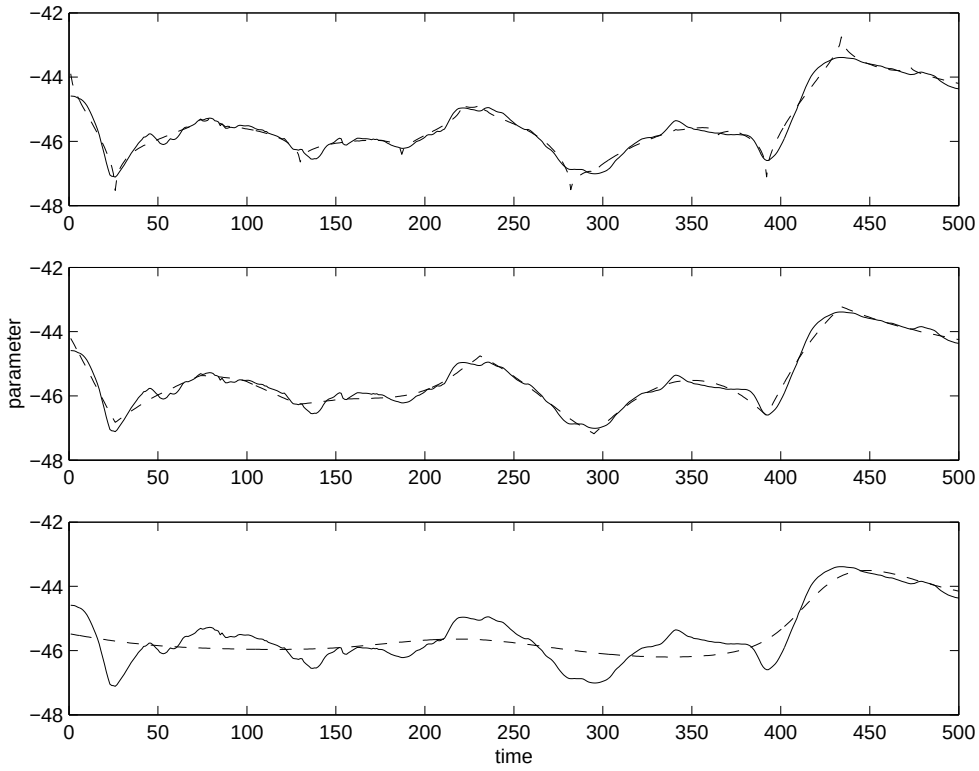


Figure 5.2: Fitting dominance functions (after [Cohen and Massaro, 1993]) to real articulatory motion using Simulated Annealing (solid - real trajectory, dashed - fitted trajectory), from top to bottom: $c = 0.5$, $c = 1$, $c = 2$.

observed in real-life. This can be remedied in the described model only by correct choice of dominance function parameters[5]

These observations indicate possible directions in which to modify the Cohen & Massaro approach. Examples of modifications include limiting the number of contributions at each point in an utterance [King, 2001], and changing the dominance functions themselves [Cosi et al., 2003]. Despite any possible limitations of the described technique it is accepted as the *de facto* standard for modelling coarticulation in visual speech synthesis. A novel alternative based upon constrained-optimization is described in Section 5.3.

5.2.1 Fitting Dominance Functions to Speech Trajectories

The quality of a speech synthesis technique is directly related to its ability to reproduce observed articulatory movements. Fitting synthesized trajectories to captured data is an optimization problem in several unknowns: the viseme targets, T_{sp}; the parameters for each of the related dominance functions, D_{sp} (i.e. $\{\alpha_{sp}, \Theta_{\leftarrow sp}, \Theta_{\rightarrow sp}\}$); and the shape parameter, c, which controls the properties of the approximation. The optimization process minimizes (5.3), the square distance between the parameter trajectory in the initial data, G_p, and the synthesized trajectory, F_p, from (5.2).

[5] In [King, 2001] this is remedied by only taking into account the closest dominance functions.

$$\textbf{minimize} \quad \sum_t \|G_p(t) - F_p(t)\|^2 \tag{5.3}$$

The space of this minimization problem is non-trivial with many local minima which prevent simplex methods from being used. The derivatives of the objective function are also unknown which further precludes many optimization techniques, e.g. steepest descent methods. For these reasons Simulated Annealing (SA, see Appendix A.3.3) provides an appropriate alternative, which can find a global minima without the requirement for exact derivatives. SA takes random steps (mutations) in the parametric space of the model, always accepting improvements in the objective (i.e. minimizing (5.3)), but also accepting *some* steps leading to a worse state. This allows SA to perform a semi-global search of the optimization landscape, and thus find the minima.

Figure 5.2 demonstrates the results of fitting a dominance function model directly to a speech trajectory (in this case the motion-captured trajectory of the upper lip) whilst varying the global shape parameter, c, of the dominance functions. It can be seen that with high values of c, and therefore more continuous dominance functions, that the SA algorithm has more difficulty in matching the captured trajectory. As c increases higher frequency characteristics cannot be reproduced. However, with lower c the generated trajectory may overshoot and does not closely match the continuity criteria of the original trajectory. It is possible that c should vary with different targets, although this would make the fitting procedure far more complex.

An interesting result of the fitting process is that because the T_{sp} are also being determined directly from the trajectory they are not ideal targets as would be visualised. In fact the T_{sp} are extreme exagerations. Whether this is true of real speech production, i.e. that the aim is to meet exagerated viseme targets, is a matter for debate. However, dominance functions can be reasonably fit to real speech trajectories and so are at least approximately functionally equivalent to the mechanisms behind speech production[6]. In the next section an alternative method for generating speech trajectories is proposed.

5.3 Target-based Synthesis using Constrained-Optimization

The limitations of the Cohen and Massaro approach (and its derivatives) do not necessarily preclude the use of target-based models for speech synthesis. In fact, even given the limitations such models can produce good results, and have been shown to reasonably approximate observed speech dynamics [Cohen et al., 2002] (also see Section 5.2.1.) However, it may be appropriate to reformulate the problem in order to overcome these problems whilst still conforming to the idea of speech production as a target-based process.

In [Witkin and Kass, 1988] physics-based articulated body motion is formulated as a global optimization problem. An objective function, $Obj(X)$, specifies the goodness of the system state X for each step in an iterative optimization procedure, whilst a set of bounded constraints, C_j, maintain the physicality of the motion, i.e. solving (5.4). For most spacetime[7] constraints problems the objective function ensures energy conservation (i.e. perform a task with minimum effort), and the constraints define some physical system within which the task must be solved.

[6] Of course many forms of spline can be fit to trajectories in much the same manner.

[7] Spacetime implies problems with objectives in space and time (i.e. animation), which are solved using global optimization methods.

$$\begin{aligned} &\textbf{minimize} && Obj(X) \\ &\textbf{subject to} && \forall j : \underline{b_j} \leq C_j(X) \leq \overline{b_j} \end{aligned} \tag{5.4}$$

Global optimization of this form fits well with the notion of speechproduction; i.e. a task-oriented system constrained by the physical nature of the articulatory structures used to produce speech. In order to use constrained-optimization techniques to generate speech trajectories it must be determined what function is being optimized, and how this is constrained during natural speech production.

5.3.1 Objective Function

The essential objective of speech production, as maintained by target-based models, is to attain a number of serially ordered vocal tract targets. By observation there is a high degree of variability in each of these targets with respect to the immediate context (i.e. due to coarticulation.) Thus it is sensible to include the variation in targets as a constituent of the system itself; that is if a viseme can include a certain degree of variability and still produce the appropriate speech sound, that variation should be encoded prior to the generation of speech trajectories. To this end visemes rather than being static targets (or morph targets in animation terminology), as is the case with Cohen and Massaro's model, are distributions within a spatial coordinate system (parameterisation) representing the vocal tract.

Consider each viseme, V_i, to be represented as a distribution in our parameterisation (be that any of the methods discussed in Chapter 3), with *ideal* target V_i and lower, $\underline{V_i}$, and upper bounds, $\overline{V_i}$. In this notation vocal tract shapes offset in each dimension of our model may still be considered to be members of the viseme distribution where they lie in the range $V_i \in [\underline{V_i}, \overline{V_i}]$. In this manner the variation of speech poses can be captured *a priori* in the model, and reasonable limits placed upon generated speech trajectories.

Given a definition which emphasises the distributed nature of speech targets, the objective function can be defined (5.5).

$$Obj(X) = \sum_i \omega_i (S(t_i) - V_i)^2 \tag{5.5}$$

This objective function optimizes the difference between the speech trajectory, S, defined by the system state, X, and the ideal targets, V_i, at the appropriate times t_i. The square difference between the speech trajectory and the ideal targets is however insufficient as some targets will be met more closely than others. For this reason the difference is weighted by a factor, ω_i, which defines the extent of the dominance that target exerts over the speech trajectory. In this manner ω_i performs a similar function to α_{sp} from (5.1). However, in the presence of no constraints[8] ω_i will have no effect upon the final trajectory and each of the V_i will be interpolated.

This objective function contrasts with *most* spacetime methods in that it does not contain an energy conservation term. This is due to the fact that in natural speech targets are not met, and thus the solution will use all the available energy to get as close as possible to the targets. Essentially there is no slack in speech trajectories to remove.

[8] A set of constraints which are not violated is equivalent to no constraints at all.

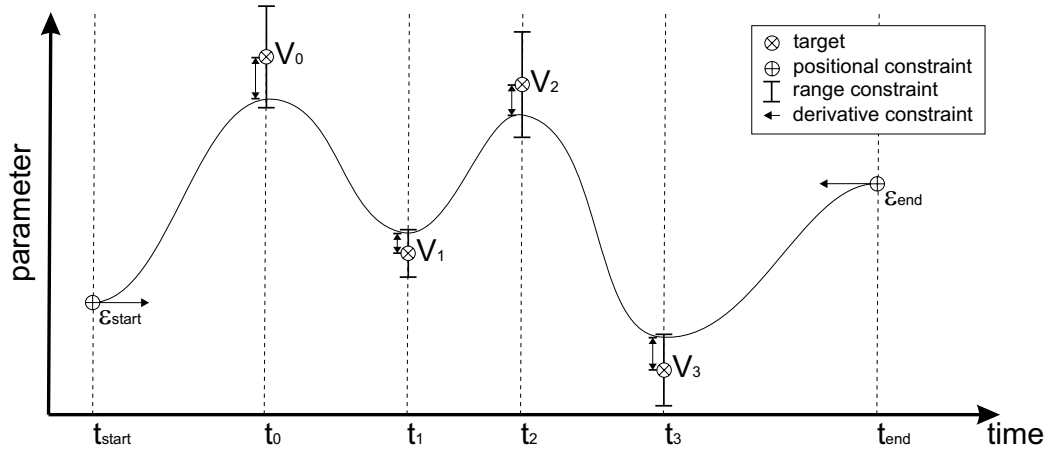


Figure 5.3: Conceptual view of optimization-based generation of speech trajectories.

5.3.2 Constraints

The previous section discussed the final objective of speech production, i.e. to get as close as possible to several ideal targets. Now it is important to define the constraints (physical and otherwise) upon a speech trajectory.

A speech trajectory is a curve passing through a spatial coordinate system which represents vocal tract gestures. This curve will begin and end at ε_{start} and ε_{end} respectively (possibly the same position, e.g. the neutral expression.) In-between it will pass close to the relevant V_i according to (5.5) but not interpolate them because the parameters may only change with regard to certain restrictions. These constraints upon the speech trajectory can be classified in two ways: global constraints, which determine the physical rules of the system; and local constraints which ensure speech-like behaviour is generated.

Constraints can be used to ensure positional and derivative values at specified times (equality constraints) and parameter ranges (inequality constraints) across the speech trajectory. Boundary constraints at the beginning and end of the trajectory are used to ensure that the motion starts and ends with the correct vocal tract gesture and in a rest configuration (i.e. with no residual forces); several boundary constraints are listed in table 5.1. Similarly, such constraints can be used to append trajectories together by matching position and derivatives at the adjoining boundary.

Table 5.1: Boundary constraints.

CONSTRAINT	DESCRIPTION
$S(t_{start}) = \varepsilon_{start}$	Ensures trajectory starts at ε_{start}
$S(t_{end}) = \varepsilon_{end}$	Ensures trajectory ends at ε_{end}
$S(t_{start})' = S(t_{end})' = 0$	Ensures the articulators are stationary at the beginning and end of the trajectory.
$S(t_{start})'' = S(t_{end})'' = 0$	Ensures the articulators are in a rest state at the beginning and end of the trajectory.

For each of the visemes in an utterance there will be a range of shapes that the vocal tract can take according to coarticulation. Outside of these ranges the vocal tract shape cannot create the matching audio. Thus, the extent to which the target is met at the appropriate time, t_i, is constrained to lie between

maximum, $\underline{V_i}$, and minimum, $\overline{V_i}$, values (5.6).

$$S(t_i) \in [\underline{V_i}, \overline{V_i}] \tag{5.6}$$

Without the presence of a global constraint the combination of objective function, (5.5), and local constraints, table 5.1 and (5.6), will simply lead to an interpolation of the viseme targets. The global constraint is required to prevent the targets from being met, i.e. damping the trajectory. In order to do this the parametric acceleration is limited across the trajectory, implicitly constraining the parametric forces and thus a physical constraint on motion, (5.7).

$$|S(t)''| \leq \gamma \quad \textbf{where } t \in [t_{start}, t_{end}] \tag{5.7}$$

In (5.7) γ is the maximum allowable magnitude of acceleration across the entire trajectory. As this constraint becomes more strict, i.e. $\gamma \to 0$, the trajectory is not capable of meeting all the targets and thus in combination with the objective function (5.5) targets will be realised according to their dominance, ω_i. A conceptual view of the optimization and related constraints can be seen in fig. 5.3.

5.3.3 Representing the Speech Trajectory

In order to apply the objective function and constraints defined in the previous sections a concrete representation for $S(t)$ must be defined. The curve representation must have enough degrees-of-freedom to represent any particular speech trajectory, and idealy should exhibit at least C^2 continuity to make the application of (5.7) feasible.

The curve representation used here is a cubic non-uniform B-spline. C^2 continuity, as previously mentioned, allows the global constraint to be applied at a sampling of the spline. Otherwise, there is no natural way to apply the constraint. Also, because turning points in the spline will only occur at the t_i (i.e. the viseme targets will be extrema in the trajectory), the spline is non-uniform requiring only $n+2$ control points to define a trajectory between n visemes, and two end conditions (ε_{start} and ε_{end}). The control points of the spline are the members of X, and there will to be $n+6$ knots to define a spline between n control points (5.8).

$$\begin{aligned} X &= \{X_1, X_2, X_3, \ldots, X_{n-2}, X_{n-1}, X_n\} \\ T &= \{t_1, t_1, t_1, t_1, t_2, t_3 \ldots, t_{n-2}, t_{n-1}, t_n, t_n, t_n, t_n\} \quad \textit{where } t_{i-1} \leq t_i \leq t_{i+1} \end{aligned} \tag{5.8}$$

The beginning and end knots are repeated to ensure that the first and last control points are interpolated, although this is not necessary for the method to work. The curve is defined between the fourth and fourth-from-last knots (i.e. between X_1 at t_1 and X_n at t_n.) The knot vector, T, is required to define the basis functions, B_i, which define the curve (5.9). The Cox-deBoor recursion is used for this - see [Farin, 1997, Bartels et al., 1987] for a discussion of B-splines.

$$S(t) = \sum_i X_i B_i(t) \tag{5.9}$$

This is the general formula for a B-spline. At any point on a cubic spline only four basis functions will be non-zero. For this reason (5.9) can become (5.10) within the curve segment $t \in [t_i, t_{i+1})$.

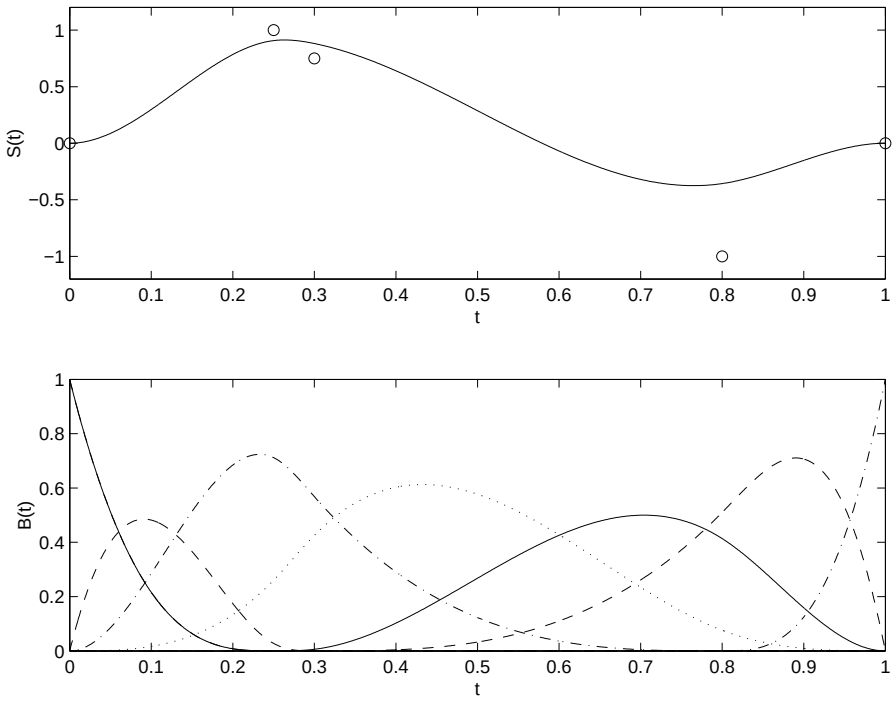


Figure 5.4: Non-uniform Cubic B-spline and its basis functions

$$\begin{aligned} S(t) &= \sum_{j=-2}^{1} X_{i+j} B_{i+j}(t) \\ &= X_{i-2}B_{i-2}(t) + X_{i-1}B_{i-1}(t) + X_i B_i(t) + X_{i+1}B_{i+1}(t) \end{aligned} \tag{5.10}$$

Similarly, the first, $S'(t)$, and second, $S''(t)$, derivatives ofthe spline can be defined using the derivatives of the basis functions (5.11).

$$\begin{aligned} S'(t) &= \sum_{j=-2}^{1} X_{i+j} B'_{i+j}(t) \\ &= X_{i-2}B'_{i-2}(t) + X_{i-1}B'_{i-1}(t) + X_i B'_i(t) + X_{i+1}B'_{i+1}(t) \\ S''(t) &= \sum_{j=-2}^{1} X_{i+j} B''_{i+j}(t) \\ &= X_{i-2}B''_{i-2}(t) + X_{i-1}B''_{i-1}(t) + X_i B''_i(t) + X_{i+1}B''_{i+1}(t) \end{aligned} \tag{5.11}$$

Figure 5.4 demonstrates a Cubic B-spline curve with non-uniform knot spacing, and its basis functions, B_i.

5.3.4 Solving The Constrained Optimization Problem

The constrained-optimization problem described in the Sections 5.3, 5.3.1 and 5.3.2 can be solved by any of the conventional means (see Appendix A.3.) In the case where the derivatives of the objective function and constraints are available, the Sequential Quadratic Programming (SQP) method is used (see Appendix A.3.2.) SQP at each step takes a second order step optimizing the objective function and a first order step in the constraints to project up to the constraint boundary. The derivatives are available when the trajectory is being represented by a cubic B-spline, and thus a projection method of this form can be used. In the case where derivatives are not available finite-differences or some other numerical method can be used to approximate them with some associated loss of accuracy.

The SQP method requires the Hessian of the objective function, H_{obj}, and the Jacobian of the constraint functions, J_C, to be calculated (5.12).

$$H_{obj} = \begin{pmatrix} \frac{\partial^2 Obj}{\partial X_1 \partial X_1} & \frac{\partial^2 Obj}{\partial X_1 \partial X_2} & \cdots & \frac{\partial^2 Obj}{\partial X_1 \partial X_n} \\ \frac{\partial^2 Obj}{\partial X_2 \partial X_j} & \ddots & & \frac{\partial^2 Obj}{\partial X_2 \partial X_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 Obj}{\partial X_n \partial X_1} & \frac{\partial^2 Obj}{\partial X_n \partial X_2} & \cdots & \frac{\partial^2 Obj}{\partial X_n \partial X_n} \end{pmatrix} \quad J_C = \begin{pmatrix} \frac{\partial C_1}{\partial X_1} & \frac{\partial C_1}{\partial X_2} & \cdots & \frac{\partial C_1}{\partial X_n} \\ \frac{\partial C_2}{\partial X_1} & \ddots & & \frac{\partial C_2}{\partial X_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial C_m}{\partial X_1} & \frac{\partial C_m}{\partial X_2} & \cdots & \frac{\partial C_m}{\partial X_n} \end{pmatrix} \tag{5.12}$$

As discussed in the previous section, the trajectory is represented with a non-uniform cubic B-spline, according to (5.9). Given this definition, the objective function becomes (5.13).

$$\begin{aligned} Obj(X) &= \textstyle\sum_i \omega_i (S(t_i) - V_i)^2 \\ &= \textstyle\sum_i \omega_i ((\sum_j X_j B_j(t_i)) - V_i)^2 \end{aligned} \tag{5.13}$$

The matrix elements of H_{obj} can be generalised to the form in (5.14).

$$\frac{\partial^2 Obj}{\partial X_j \partial X_k} = \sum_i (2\omega_i B_j(t_i) B_k(t_i)) \tag{5.14}$$

This is a summation for all i. However, where the spline is cubic, basis functions, B_l, where $(l < i-2) \vee (l > i+1)$ will be zero at t_i. This means that $B_j(t_i)B_k(t_i) = 0$ if $(j < i-2) \vee (j > i+1) \vee (k < i-2) \vee (k > i+1)$ and thus these terms do not contribute. H_{obj} will be a symmetric matrix with non-zero elements lying across the diagonal in the range $(k-2) \leq j \leq (k+1)$.

The elements of the jacobian, J_C, depend upon the individual constraints. These fall into the following categories:

- *Global Constraint* - at a sampling along the trajectory restrict the magnitude of $S''(t)$ to prevent targets from being met (5.7).
- *Positional Constraints* - at the beginning and end clamp the trajectory to pass through ε_{start} and ε_{end} respectively (see table 5.1.)
- *Derivative Constraints* - at the beginning and end of the trajectory constrain $S'(t_{start}) = S''(t_{end}) = 0$ (see table 5.1.)
- *Range Constraints* - at each t_i constrain the trajectory to lie in the range $S(t_i) \in [\underline{V_i}, \overline{V_i}]$ (5.6).

The global constraint, C_{global}, by substitution from (5.11), will become (5.15) where S is a cubic B-spline and $t \in [t_i, t_{i+1})$.

$$C_{global}(t) = \begin{cases} (\gamma - S''(t))^2 = (\gamma - (\sum_{j=-2}^{1} X_{i+j} B''_{i+j}(t)))^2 & \textbf{where } S''(t) > \gamma \\ (-\gamma - S''(t))^2 = (-\gamma - (\sum_{j=-2}^{1} X_{i+j} B''_{i+j}(t)))^2 & \textbf{where } S''(t) < -\gamma \\ 0 & \textbf{otherwise} \end{cases} \tag{5.15}$$

The elements of J_C corresponding to (5.15) will become (5.16).

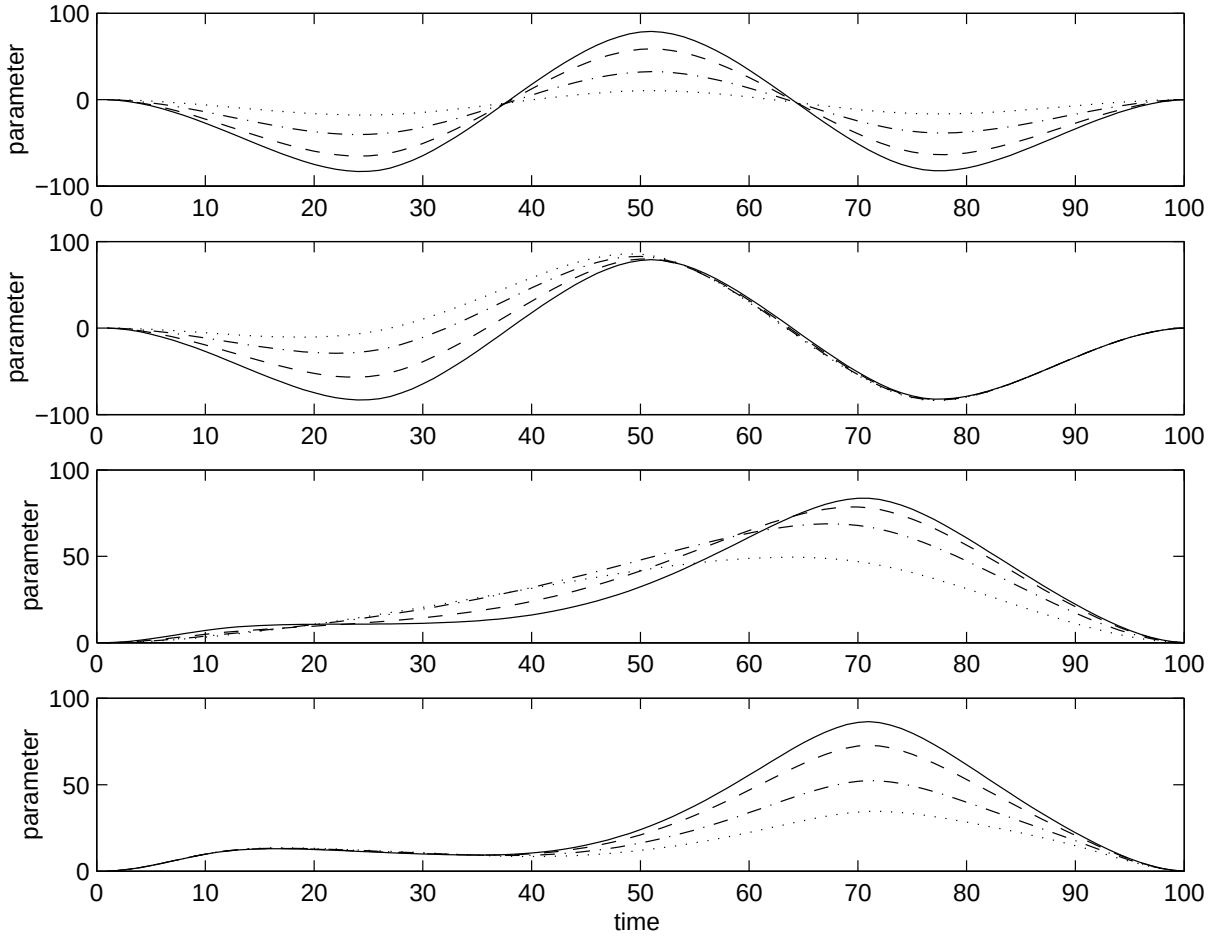


Figure 5.5: Effect of varying dominance and global constraint upon speech trajectories generated using the constrained-optimization target-based approach. Top and second-top trajectories have three targets: $\{\{t_1 = 25, \omega_1 = 1.0, \mu_1 = -100\}; \{t_2 = 50, \omega_2 = 1.0, \mu_2 = 100\}; \{t_3 = 75, \omega_3 = 1.0, \mu_3 = -100\}\}$. Top shows increasing global acceleration constraint, from least constrained (solid) to most constrained (dotted.) Second top demonstrates decreasing the dominance of the first target from equal dominance ($\omega_1 = 1.0$, solid) halving the dominance at each step. Bottom and second-bottom trajectories have five targets: $\{\{t_1 = 10, \omega_1 = 0.6, \mu_1 = 20\}; \{t_2 = 30, \omega_2 = 0.1, \mu_2 = 20\}; \{t_3 = 50, \omega_3 = 0.05, \mu_3 = 20\}; \{t_4 = 70, \omega_4 = 1.0, \mu_4 = 90\}; \{t_5 = 90, \omega_5 = 0.6, \mu_5 = 20\}\}$. Second-bottom shows increasing global acceleration constraint, from least constrained (solid) to most constrained (dotted.) Bottom demonstrates decreasing the dominance of the forth (most dominant) target from $\omega_4 = 1.0$ (solid) halving the dominance at each step.

$$\frac{\partial C_{global}(t)}{\partial X_k} = \begin{cases} -2\left(\gamma - (\sum_{j=-2}^{1} X_{i+j} B''_{i+j}(t))\right) B''_k(t) & \textbf{where } S''(t) > \gamma \\ 2\left(\gamma + (\sum_{j=-2}^{1} X_{i+j} B''_{i+j}(t))\right) B''_k(t) & \textbf{where } S''(t) < -\gamma \\ 0 & \textbf{otherwise} \end{cases} \tag{5.16}$$

This constraint is applied at a sampling along the trajectory, due to its global nature. This is adequate as the spline is cubic the second derivative, $S''(t)$ varies continuously along the spline.

To constrain the trajectory to exactly pass through a point, ε, at time $t_\varepsilon \in [t_i, t_{i+1})$ the constraint equation C_{pos} is used (5.17).

$$\begin{aligned} C_{pos} &= (\varepsilon - S(t_\varepsilon))^2 \\ &= (\varepsilon - (\sum_{j=-2}^{1} X_{i+j} B_{i+j}(t_\varepsilon)))^2 \end{aligned} \tag{5.17}$$

The derivatives of J_C corresponding to (5.17) will become (5.18).

$$\frac{\partial C_{pos}}{\partial X_k} = -2\left(\varepsilon - \left(\sum_{j=-2}^{1} X_{i+j} B_{i+j}(t_\varepsilon)\right)\right) B_k(t_\varepsilon) \tag{5.18}$$

Derivative constraints can be applied in exactly the same way as (5.17), by replacing $B(t)$ with $B'(t)$ or $B''(t)$.

Finally to constrain the trajectory at time t_i to lie in the range $S(t_i) \in [\underline{V_i}, \overline{V_i}]$, C_{rng} is used (5.19).

$$C_{rng} = \begin{cases} (\overline{V_i} - S(t_i))^2 = (\overline{V_i} - (\sum_{j=-2}^{1} X_{i+j} B_{i+j}(t_i)))^2 & \textbf{where } S(t_i) > \overline{V_i} \\ (\underline{V_i} - S(t_i))^2 = (\underline{V_i} - (\sum_{j=-2}^{1} X_{i+j} B_{i+j}(t_i)))^2 & \textbf{where } S(t_i) < \underline{V_i} \\ 0 & \textbf{otherwise} \end{cases} \tag{5.19}$$

The derivatives of J_C corresponding to (5.19) become (5.20).

$$\frac{\partial C_{rng}}{\partial X_k} = \begin{cases} -2\left(\overline{V_i} - (\sum_{j=-2}^{1} X_{i+j} B_{i+j}(t_i))\right) B_k(t_i) & \textbf{where } S(t_i) > \overline{V_i} \\ -2\left(\underline{V_i} - (\sum_{j=-2}^{1} X_{i+j} B_{i+j}(t_i))\right) B_k(t_i) & \textbf{where } S(t_i) < \underline{V_i} \\ 0 & \textbf{otherwise} \end{cases} \tag{5.20}$$

The complexity of the system is directly related to the number of viseme targets. Increasing the number of targets will lead to a linear growth in size of both H_{obj} and J_C. Also, the number of parameters required to represent the vocal tract will increase the time to convergence of the system. This may possibly be improved by using a windowing approach such as that from [Cohen, 1992].

Several example trajectories demonstrating the application of the global constraint and dominance are shown in fig. 5.5. In fig. 5.6 trajectories for a complex speech trajectory are shown. As can be seen the global constraint has the desired effect of dampening the entire trajectory and preventing all targets (the V_i) from being met. As the dominance, ω_i, of an individual target is increased the trajectory will gravitate towards it, at the expense of the surrounding targets. Conversely, as ω_i is reduced the surrounding targets are better met at the expense of V_i. The described method is a very powerful approach, allowing arbitrary extensions to target-based synthesis by changing the objective function, the global constraints or adding local constraints to get desired changes in generated trajectories.

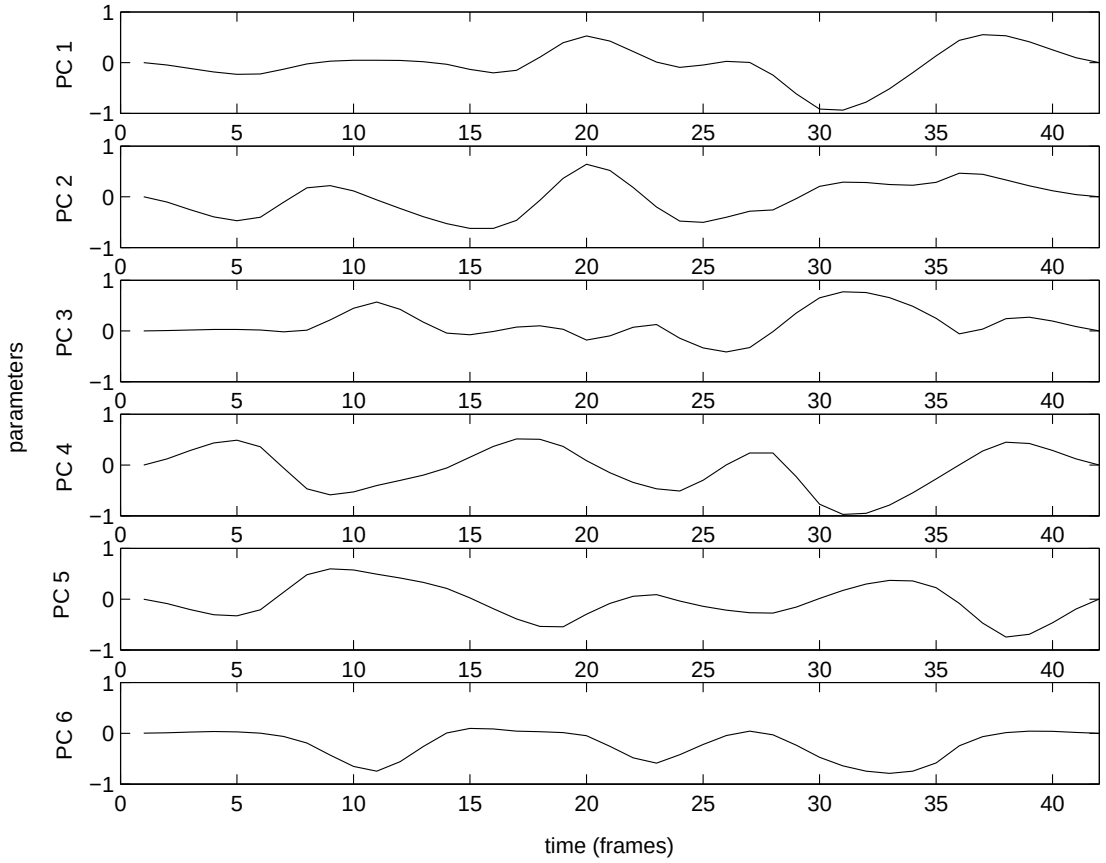


Figure 5.6: Speech trajectories generated using the constrained-optimization method for the sentence '*my name is not baldy*'. Trajectories sampled at 25Hz.

5.3.5 Comparison with Dominance Functions

The observations discussed in Section 5.2 can be taken as a set of criteria for comparing the capabilities of dominance functions (after [Cohen and Massaro, 1993]) and the constrained-optimization technique introduced in this thesis.

In terms of extensibility the shape of dominance functions can be controlled by changing their parameters or by allowing different forms of function for different *types* of speech target (e.g. after tables 2.3 or 2.4.) This requires the manipulation of abstract parameters indirectly related to the appearance of a speech trajectory. In contrast, the optimization approach allows arbitrary constraints to be applied, either clamping its position or derivatives, or constraining the range of values the trajectory can take at a particular time. Thus, *speech-like* properties can be applied directly, e.g. the time at which the lips part during a bilabial plosive can be directly asserted. This property allows iterative enhancement of speech trajectories according to known properties of vocal articulators.

The fact that positional constraints can be directly applied prevents the problem of explicitly modelling silence. In the constrained-optimization approach silence is modelled by enforcing the trajectory to pass through start, ε_{start}, and end, ε_{end}, targets. Further positional constraints can be applied across the trajectory, e.g. to apply emotional expressions between sentences. Thus, the start and end points of the motion can be asserted and will affect the surrounding speech transitions. Using dominance functions

the interpolation cannot be asserted except by extremely high $\alpha_{silence}$ which will adversely affect the coarticulation modelling. It could be considered inconsistent that silence, the absence of vocalisation, has a dominance over a speech trajectory.

By applying a range constraint on the trajectory at each of the viseme targets we can assert that the centre of each viseme is met well enough for audio-visual coherence. Implicitly this is also a constraint upon how closely targets can be placed, as the global constraint and the range constraints may compete (i.e. γ could be too strict to allow some of the range constraints to be met.) This is a condition on audio-visual coherence which cannot be ensured using dominance functions. The only way to ensure this with dominance functions would be significant tuning of the parameters and applying some constraints upon the duration and context of each target.

The global constraint (5.7) also provides some control over *manner* of articulation. It is obvious that with no constraint the trajectory will be a simple interpolation of the targets (over-articulation), and that highly constrained trajectories will exhibit little motion (under-articulation.) This is a continuous range of articulatory styles, with the extreme values at either end being unrealistic. However, there are a range of values for the global constraint where articulation is realistic. This variability in generated trajectories can be considered to provide stylistic control, for which there is no mechanism using dominance functions.

One disadvantage of the optimization technique is that the coarticulation of adjacent targets will be a symmetric effect. This is different to dominance functions where each basis, D_{sp}, is skewed to provide for the asymmetric properties of coarticulation (i.e. the differences between forward and backwards coarticulation.) In order to model this the global constraint must be non-linear in nature, whereas currently a linear constraint on the acceleration is used. Onset/offset characteristics surrounding each viseme target would need to be modelled. To apply such a constraint using the optimization method may require a higher order spline.

It should be pointed out that the optimization-based method described is a member of a class of techniques. Different methods could be considered for applying the global constraint, and different constraint and objective functions could be used to improve the approximation to observed speech characteristics. Also, the structure of speech trajectories are manipulated directly in the parametric space of the model. This is beneficial over the manipulation of abstract dominance parameters which cannot be directly measured from real speakers, only estimated using methods such as described in Section 5.2.1.

5.4 Motion-based Synthesis

Motion-based synthesis contrasts with target-based methods as, instead of approximating (or interpolating) a set of discrete positions in parameter space, fragments of captured motions are concatenated to form the final trajectory. Instead of modelling coarticulation explicitly motion-based synthesis assumes that the *majority* of the effects of coarticulation will be captured within the units to be concatenated. It is at the joins, or concatenation points, between units that the synthetic nature of the motion will be most apparent. This method is analagous to concatenative audio synthesis; considered to be the most natural means of synthesizing speech audio (see Appendix B.3.)

This form of synthesis requires the following problems to be tackled:

- *Database Design and Capture* - From a target domain design a database which will cover all

possible variations. Capturing motion data is discussed in Chapter 4.

- *Unit Selection* - Given a new utterance, which is distinct from samples already present in the database, select fragments which can be combined to produce an appropriate trajectory.
- *Alignment of Speech Fragments* - Stretch and squash the selected motion fragments so that they are aligned with the phonetic transcription of the target utterance.
- *Resampling of Speech Fragments* - Process the fragments to a consistent sample rate.
- *Motion Blending* - With the results of the previous stages blend the motion fragments to generate a trajectory for the target utterance.

Database design for concatenative synthesis has been covered in detail for audio synthesis [Black and Lenzo, 2001]. The same techniques can be directly applied for visual speech synthesis. Obviously, for large- or general-domain synthesis smaller units will be required to make data capture feasible. In the following sections are methods for the synthesis of speech trajectories given a database of speech motion samples consisting of variable-length units (word and phrase.) The methods are equally applicable for different size units.

As a preparatory stage in motion-based synthesis the fragments will already have been filtered, and rigid-body motion removed. Without rigid-alignment units cannot be blended coherently. For the purposes of the following sections the data will have been processed according to the methods in Section 4.3.

5.4.1 Unit Selection

The method used for unit selection is dependent upon the underlying speech units. In the case where units of varying duration are available, a method must be defined to select the most appropriate units to synthesize a target utterance[9]. As input to the process the phonetic labels and timing of the target utterance are required, which can be directly recovered from the audio synthesis procedure. Ultimately the aim of unit selection is to find the smallest number of fragments that account for the phonemes in the target utterance (see fig. 5.7.) Pseudocode for the basic algorithm is shown in table 5.2.

In this code FIND-UNIT is a subprocedure which searches for a speech fragment which spans several phones in the target utterance, e.g. the closed sequence ['c','a','t']. APPEND-UNIT appends the found unit to the output list of fragments. Primarily this algorithm chooses fragments of longer duration, which is beneficial to the naturalness of the output speech. However, disambiguation is required where more than one speech fragment is available within the database for a given sequence. In this case, the factors which are taken into account when selecting units are: similarity in the phonetic timing to the target utterance (using the sum of square differences as an indicator), and similarity of context. Where two units are highly similar context is taken into account by selecting the unit with the closest immediate context (preceding and following phonemes), if this still does not separate the units the algorithm compares the next surrounding context until the best unit is found. Each of these conditions biases towards using fragments as similar as possible to the target utterance, and thus the synthesized trajectories

[9]The case of variable length units is the most complex. If only, for example, diphones are available the selection of units is simpler.

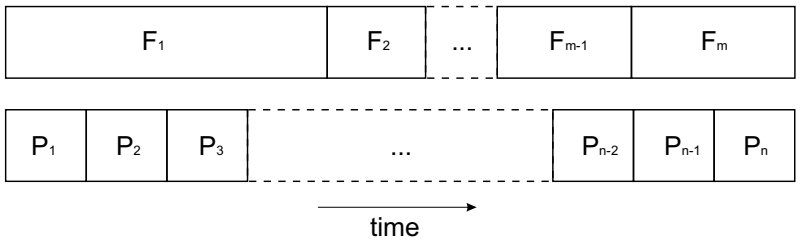


Figure 5.7: Unit selection consists of finding the minimal number of fragments, F_i, which account for the phonemes, P_j, in a target utterance.

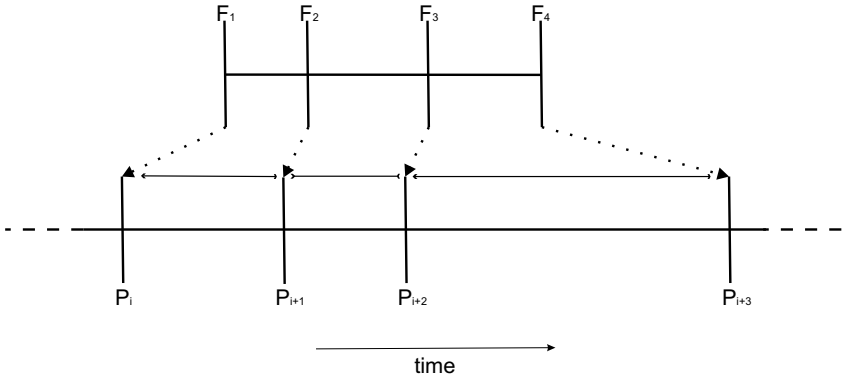


Figure 5.8: Alignment of fragment, F, with phoneme timings F_j to utterance segment with phoneme timings P_i. The fragment is stretched and squashed during the alignment.

Table 5.2: Fragment Selection Algorithm.

Input: List of *phones*
Output: List of $fragments$

$frags \leftarrow []$
$i \leftarrow 1$
$j \leftarrow numPhones$
while $i <$ *numPhones* **do**
while not FIND-UNIT$(phones, i, j)$ **do**
$j \leftarrow j - 1$
end while
APPEND-UNIT$(frags, phones, i, j)$
$i \leftarrow j$
$j \leftarrow numPhones$
end while

should maintain the naturalness in movement of the captured data. A similar unit selection method has also been reported in [Cao et al., 2004].

5.4.2 Alignment and Resampling of Speech Fragments

Given an appropriate selection of units, the next stage is to adapt these fragments so that in combination they can be used to synthesize the target utterance. Essentially, this requires that the units are temporally aligned with the target utterance. Each speech fragment, whether it be a diphone or a sentence, has a phonetic labelling, and must be variously stretched/squashed so that the labels are correctly aligned with the phonetic structure of the synthesized audio. This is visually depicted in fig. 5.8.

This can be achieved by evenly distributing motion samples between repositioned phonetic labels. However, that will lead to an uneven distribution in the sampling of the speech fragments, which will give an inconsistent frame-rate for animation. For this reason, having adapted the fragments so that they are aligned with the target utterance, the fragments must be further resampled to achieve a consistent frame-rate before blending.

This is the scattered-data interpolation problem, i.e. given a scattered sampling of data form a continuous curve/surface passing through the points. Many methods, such as B-spline interpolation, could be used to resample the data, here radial-basis functions (RBFs) are used (see Appendix A.1.1.) To use RBFs for the purposes of resampling motion fragments, a basis centre is placed at each sampled point, ensuring that the interpolating curve will exactly fit the known data. The interpolated motions are in fact a mapping from the time-domain onto the spatial domain, and thus to finally resample the data requires only querying the interpolated motion at uniform temporal intervals. This is manageable because any of the motion fragments will only ever be short in duration (up to a couple of hundred frames, depending upon the sampling rate of the initial data.)

5.4.3 Blending Motions

The final stage of synthesis, given appropriate aligned speech fragments from the previous stages, is to blend the fragments such that visibly continuous motion is exhibited in the resulting trajectory. This involves only the overlapping regions of motions at the joints. A small degree of context is required in the fragments to facilitate this. Within the overlapping section, $t \in [t_0, t_1]$, a weighted blend of the two motions is used (5.21).

$$\begin{aligned} F_{blend}(t) \quad &= g(u)F_0(t) + (1-g(u))F_1(t) \\ \textbf{where} \quad &u = \left(\tfrac{t-t_0}{t_1-t_0}\right) \end{aligned} \tag{5.21}$$

In (5.21), $g(u)$ is a weighting function (see fig. 5.9) which returns a value in the interval $[0,1]$. The weighting function facilitates the blend and ensures a smooth transition between the fragments, which are represented here as functions of time ($F_x(t)$.) The speed of decay in g will determine how fast the second fragment is faded in.

The size of the overlapping regions depends upon the frame-rate of the fragments themselves. However, they should always be a fraction of the smallest phone-to-phone interval to prevent large fragments dominating over the target utterance. In practice, for animation frame-rates of ~ 30 fps, there will

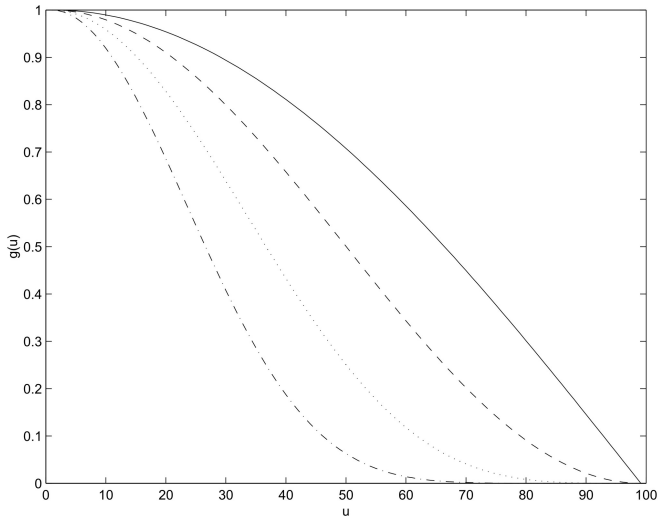


Figure 5.9: Example weighting functions $g(u)$.

never be more than a couple of frames overlap at each join, and for this reason high speed capture is advantageous as it allows larger blend intervals.

The blending of fragments only has an impact upon the generated trajectory over small periods of the trajectory. This coupled with the fact that the motions are aligned prior to synthesis means that we do not need to perform fragment alignment during synthesis, as described in [Kshirsagar and Magnenat-Thalmann, 2003]. Examples of speech trajectories generated using this technique compared with captured trajectories can be seen in fig. 5.10.

5.5 Summary

This chapter has discussed methods for the generation of speech trajectories. These can be split into three broad categories: target-based, motion-based, and model-based synthesis. Target- and motion-based synthesis are more appropriate to synthesis from text (or equivalently phoneme timing information), whilst model-based approaches typically attempt to relate articulatory movements directly to speech audio.

Target-based models are mainly derivatives of the dominance function approach [Cohen and Massaro, 1993]. These form a speech trajectory using an approximating spline, with the control points being the viseme-targets of the utterance. In fact, a NURBS curve provides a similar level of control, (except for the skewing of the basis functions according to the directional nature of coarticulation using the $\Theta_{\leftarrow sp}$ and $\Theta_{\rightarrow sp}$ parameters in (5.1).) The properties of the trajectory are manipulated by changing the basis (dominance) functions of the spline, which model the temporal influence of the viseme over the trajectory. The problem with this formulation is that no assertions can be made as to the properties of the trajectory. The degree to which a target is realised is determined both by the context and the parameters defining the dominance functions. Determining these parameters for all possible speech combinations is a challenging task, and determining the parameters directly from captured speech trajectories leads to unexpectedly exaggerated viseme-targets.

An alternative, proposed here, is to use a constrained-optimization approach. This technique opti-

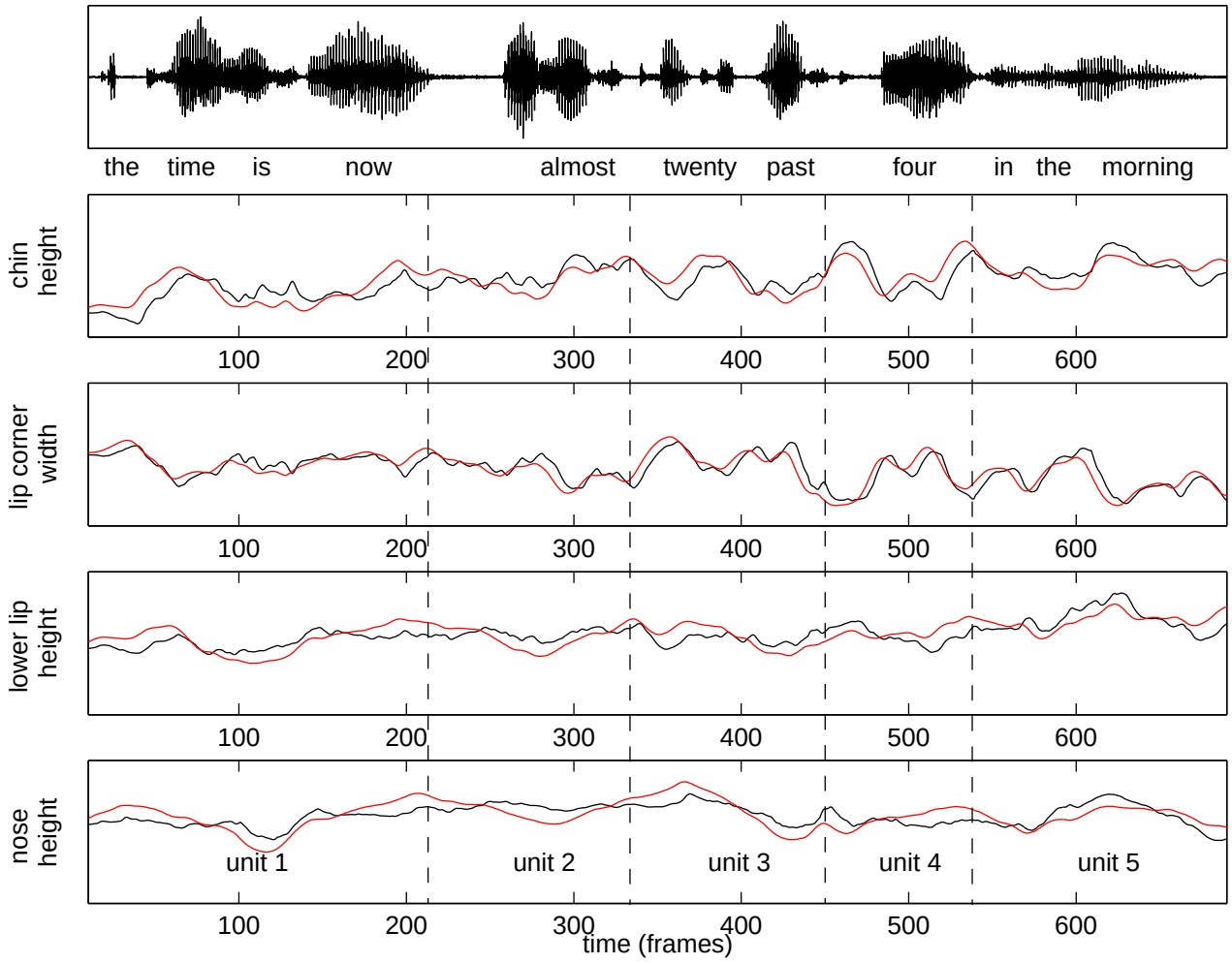


Figure 5.10: Speech trajectories generated by concatenating word and phrase length units (red) compared with natural speech (black.) Trajectories are sampled at 120Hz.

mizes the distance between the trajectory, represented by a spline, and the *ideal* targets for the visemes in an utterance. The trajectory is represented by a spline, yet the control points are not the visemes. Instead visemes are defined as parameter ranges through which the trajectory must pass at appropriate times. By constraining the trajectory so that the targets cannot be met, and relatively weighting the importance of the targets, the solution approximates the rôle of coarticulation. This method is flexible because arbitrary constraints can be placed upon the position and derivatives of the trajectory. Another key difference between this method and dominance functions is that all manipulation of the trajectory is in the parametric domain, not an abstract dominance domain, and thus observed properties of speech articulation can be directly applied as constraints upon a synthesised trajectory. The flexibility of this method lends itself to iterative refinement by applying further constraints, possibly retrieved directly from real speech movements.

In contrast, motion-based models take fragments of real speech movements and blend them together to generate novel speech trajectories. The process of concatenative visual synthesis can be split into several stages: data capture, unit preparation, unit selection, unit alignment, and blending. The technique

relies upon collecting an appropriate corpus containing all the variations in the target domain. Once captured the data must be processed to remove noise, rigid head motion, and to recover missing data (see Chapter 4.) A phonetic transcription is used to segment the data into arbitrary-length fragments, according to the target domain (e.g. diphones, syllables, words, phrases, etc.) Synthesis consists of selecting the best units (using the algorithm in table 5.2) to represent the target utterance and then blending these units to provide a continuous trajectory. The relative difficulty in capturing speech movements, as opposed to speech audio,acts as a limiting factor in the use of concatenative synthesis forvisual speech. The physical plausibility of synthesis produced byconcatenating motions is high. This is because the basis units forsynthesis are real speech movements, yet it is possible thattarget-based models used with physical modelling of facial expressioncould produce similar results. The techniques from Chapter 4 canbe used to use motions generated in this way on any target mesh, andthus maximize the use of any captured data.

Chapter 6

Implemented Systems

In order to demonstrate the techniques from Chapters 3, 4, and 5, several full text-to-visual-speech systems have been implemented. Each uses contrasting methods for modelling, parameterisation, and the generation of speech trajectories. Table 6.1 overviews the implemented systems. In particular these systems demonstrate contrasting representations of visual speech units (e.g. dynamic vs. static, geometric vs. image-based), and methods for generating trajectories through whichever parameterisation has been chosen (e.g. interpolation vs. dominance functions.)

In the systems which perform complete TTVS, the Festival speech synthesis system is used to generate audio. Festival [Black et al., 1999] is a concatenative audio synthesis system (see Section B.3) which, for general synthesis, uses diphones as base units. The system can also provide limited domain synthesis. Festival is used to generate phoneme timing information as well as other subsidiary information (e.g. pitch variation) which it passes to the visual synthesis module as input. Whilst Festival has been used here, each of the systems could equally be used with any other audio synthesis module (e.g. MBROLA [Dutoit et al., 1996]) given appropriate means of extracting information from the audio engine. Audio-To-Visual-Speech (ATVS) could also be provided given appropriate transcriptions (e.g.

Table 6.1: Implemented TTVS systems.

SYSTEM	MODELLING	SYNTHESIS	UNITS
[Edge and Maddock, 2001][a]	muscle functions	linear interpolation	visemes
[Edge and Maddock, 2003]	image morphing	dominance functions	visemes
[Edge and Maddock, 2004]	principal components model	constraint-based	viseme groups
[Edge et al., 2004]	FFD patches	unit concatenation (limited domain)	varying dynamic units

[a] This system does not perform TTVS, but generates a trajectory from a set of phoneme timings - which is essentially the same problem.

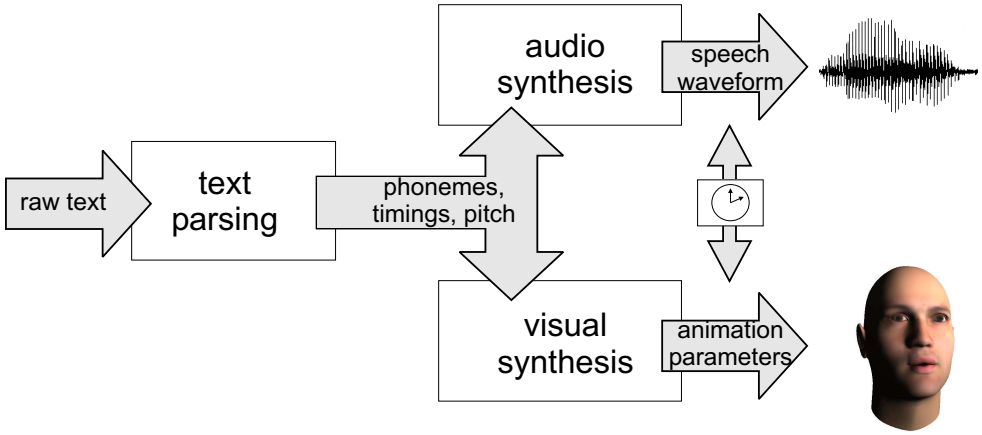


Figure 6.1: General structure of the synthesis systems.

by using a speech recognition module, such as SPHINX [Ravishankar, 2004].) A general structural overview of the implemented systems is shown in fig. 6.1.

It is important to note that the synthesis techniques described require *a priori* information with regards the phonetic structure of an utterance. This can be seen from the techniques in Chapter 5. The visual synthesis is not tied into the technique used to generate the audio, i.e. there is no interaction between the audio and visual synthesis modules. Ideally, a synthesis technique would use the same parameters for *both* audio and visual synthesis. An example would be to use some form of articulatory synthesis with parameters analogous to physical states (see Section B.1.) However, in practice articulatory synthesis is not necessarily the best quality, and most visual speech synthesis systems take the approach used here [Cosi et al., 2003, Albrecht et al., 2002, King et al., 2000, Ezzat and Poggio, 1999, Le Goff and Benoît, 1996, Cohen and Massaro, 1993, Lewis and Parke, 1987].

6.1 Synthesis using Geometric Muscle Functions

This system represents a baseline standard for visual speech synthesis. The units used are simply visemes parameterised usinggeometric-muscle functions (see Section 3.2.2.) Each muscle function geometrically warps a region of the mesh according to a muscle actuation value (i.e. the degree to which the muscle should be contracted.) Thus each viseme, V_i, consists of a set of these actuation values, i.e. $V_i = \{k_1, k_2, \ldots, k_n\}$ for the n muscles used to model expression. In the implemented model there are twenty-five muscles modelled, of which twenty-four are linear muscles (twelve left-right pairs) with one sphincter muscle surrounding the mouth.

The muscle functions used are derivatives of those described in [Waters, 1987]. These are extended to provide the ability to seperately move the upper and lower lips, with a discontinuity plane used to cull the influence of muscle functions in the region of the mouth. Also, the sphincter muscle functions are extended to allow puckering of the lips. These functions are intended to approximate the action of facial muscles without any complex physical simulation. Effectively these are free-form deformations with only one degree of freedom (the actuation of the muscle.) By rotating vertices in the jaw about a pivot simple mouth opening is achieved.

To synthesise speech, trajectories through the muscle actuation space are generated. The simplest,

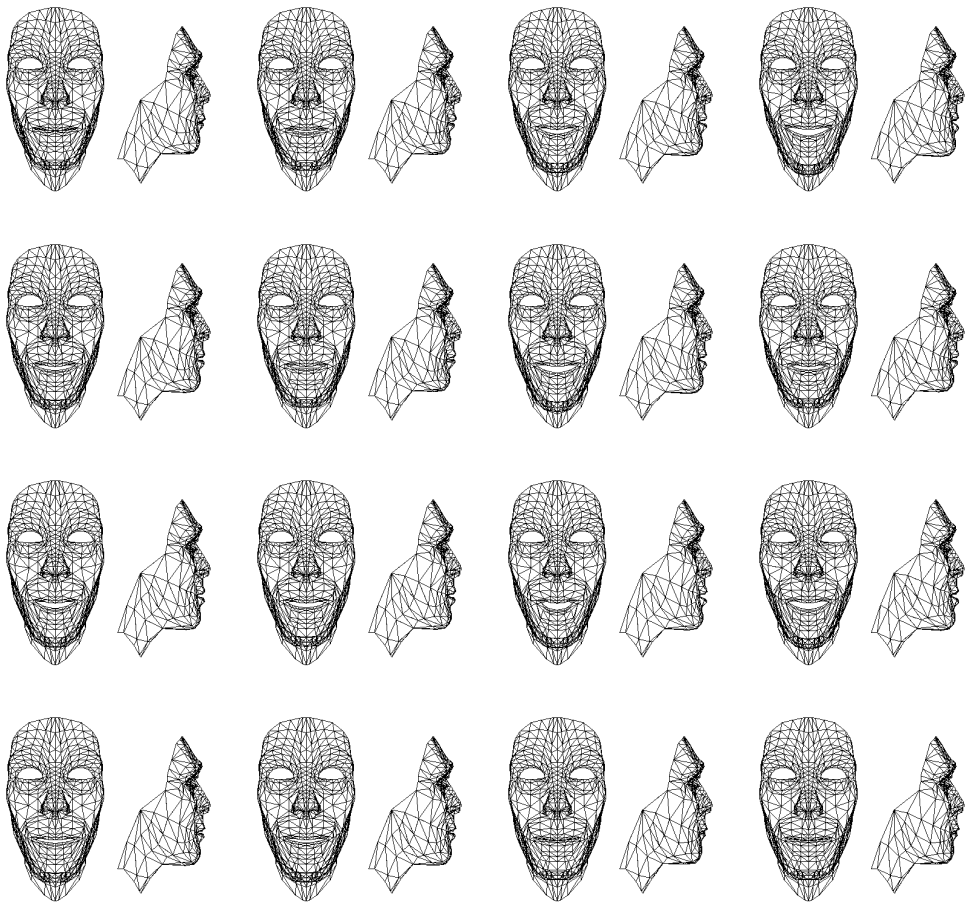


Figure 6.2: Frames from the animation 'one-five-zero-zero-six', generated using the muscle-based method.

and most coarse, method is used: interpolation (6.1).

$$\begin{aligned} T_{ij}(t) &= (1-\phi_{ij}(t))V_i + \phi_{ij}(t)V_j \\ \phi_{ij}(t) &= \sin\left(\tfrac{\pi(t-t_i)}{2(t_j-t_i)}\right) \end{aligned} \tag{6.1}$$

In (6.1) T_{ij} is the section of the speech trajectory between visemes V_i and V_j at times t_i and t_j respectively. The transition function, $\phi_{ij} : \mathbb{R} \rightarrow [0,1]$, smoothly interpolates between the muscle actuation values representing V_i and V_j. In this case a sine function is used, but any smooth function which takes on the values $\phi_{ij}(t_i) = 0$ and $\phi_{ij}(t_j) = 1$ will be sufficient (e.g. linear interpolation.)

Clearly this does not acurately model coarticulation (see Section 2.1.5 and Chapter 5) as each of the visemes will be perfectly interpolated. Context plays no rôle in (6.1) and thus the resulting animations appear over-articulated and thus unrealistic. In fact, the short periods between visemes mean that the interpolation can appear discontinuous which further impedes realism. Frames from an animation[1] generated using this technique can be seen in fig. 6.2. This is a baseline standard because, whilst coarticulation is not modelled, the phonetic structure of the utterance is used to generate the animation (i.e. to place the viseme targets.)

[1] Animations generated using the muscle-based system can be found in the folder 'animations/section_6.1/' on the accompanying CD.

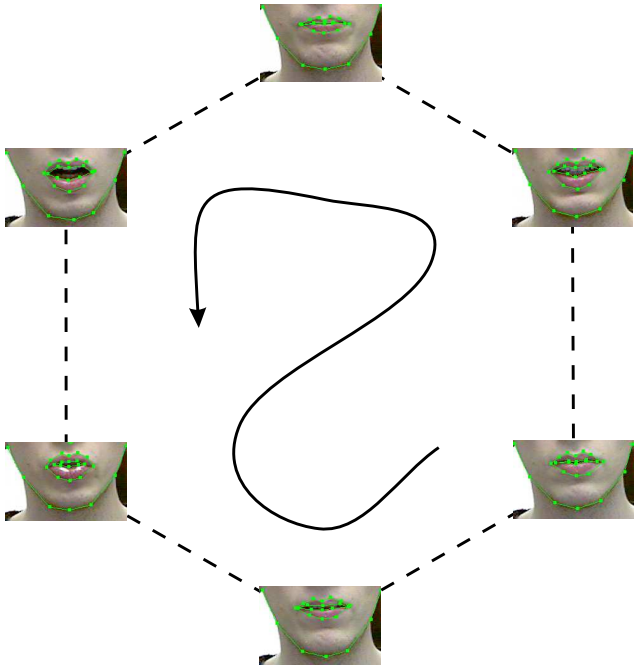


Figure 6.3: Trajectories generated using the image-based model are bounded by the sampled viseme targets.

6.2 Image-based Synthesis with Dominance Functions

An alternative to representing facial expression with three-dimensional meshes is to use images. An image has an advantage over other representations in terms of static realism - given that the image represents a real face (i.e. photographic images.) Such a model bounds the space of visible speech movements by a captured sampling of real speech articulation (see fig. 6.3.) Trajectories are generated relative to these samples to create animation. The method described here differs from [Ezzat and Poggio, 1999] because coarticulation is explicitly modelled, and so non-linear trajectories between sampled images are generated.

As with the model in Section 6.1, this model uses visemes to represent the extremes of facial expression during speech. Each trajectory consists of a sequence of visemes, V_i, at time t_i and consisting of both image, I_i, and geometry, G_i. However, this model includes a model of coarticulation. Dominance functions from [Cohen and Massaro, 1993] are used to generate speech trajectories. In order to apply dominance functions a parametric space of facial expressions must be created. This parametric space must represent the variation of speech articulation in a manner coincident to the effects of coarticulation. Geometric primitives, essentially a labelling of 2D points on each of the viseme-images, are used to facilitate the morphing of images. In this model each G_i consists of 44 points on the face; points surrounding the mouth are shown in fig. 6.4. However, 2D points in the image-plane are an inadequate parameterisation of the articulators for use with dominance functions. This is because the effects of coarticulation do not happen parallel to the axes of the image plane. An improved parameterisation can be achieved by processing the labelled points to determine a set of mutually orthogonal parameters. Such a parameterisation can be created using *principal components analysis* (PCA, see Section 3.1.3 and Appendix A.2.1.) Applying PCA to the labelled visemes leads to the components shown in fig.

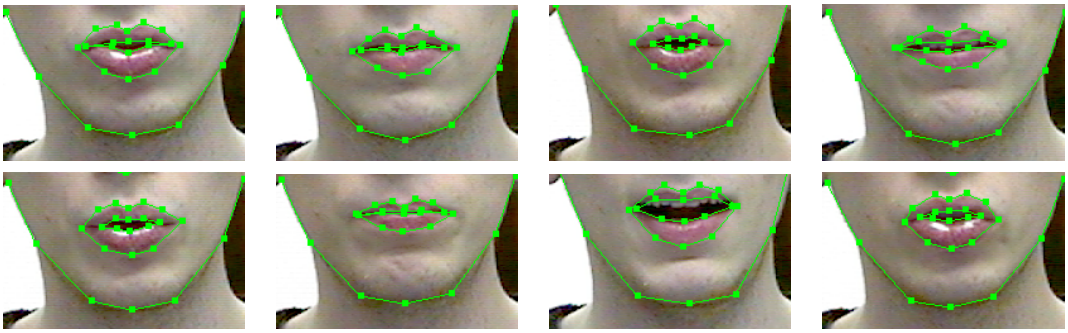


Figure 6.4: Labelled visemes for the image-based model.

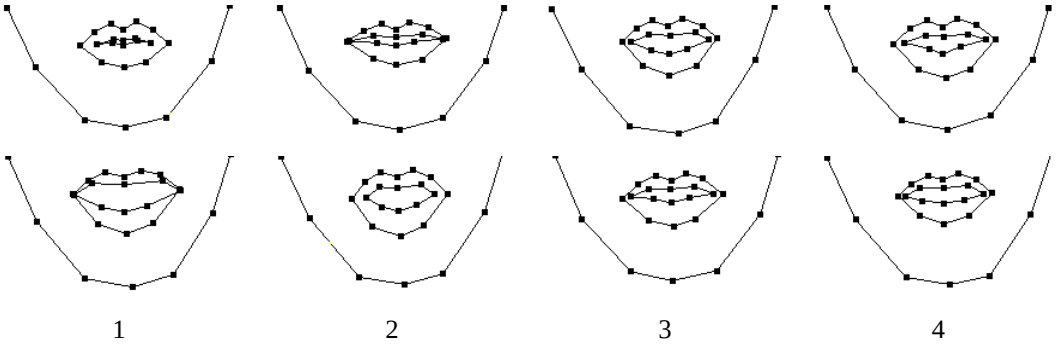


Figure 6.5: The first four geometric principal components: top-row shows $\mu + 3\sqrt{\sigma}$, bottom-row shows $\mu - 3\sqrt{\sigma}$.

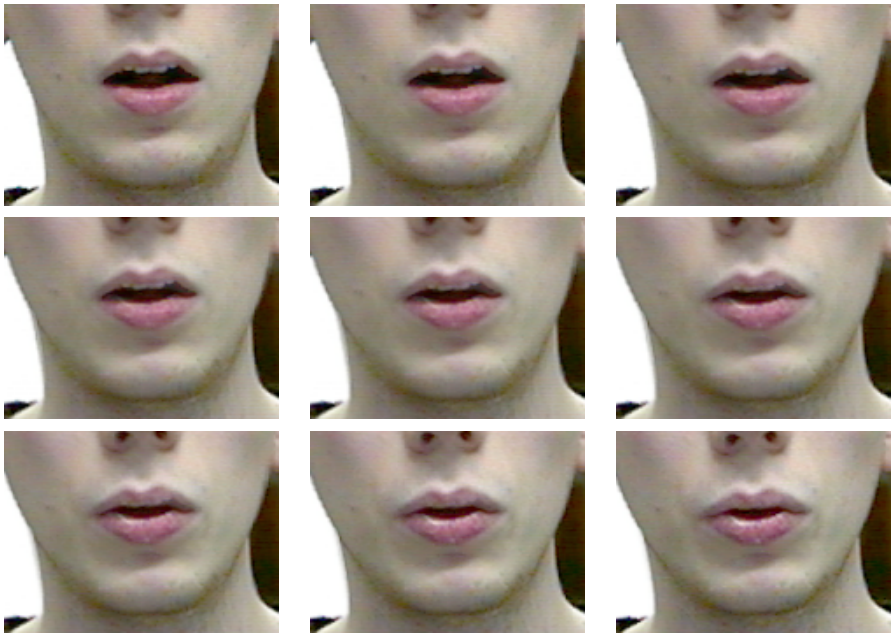


Figure 6.6: Transition between visemes /*aw*/ and /*uw*/.

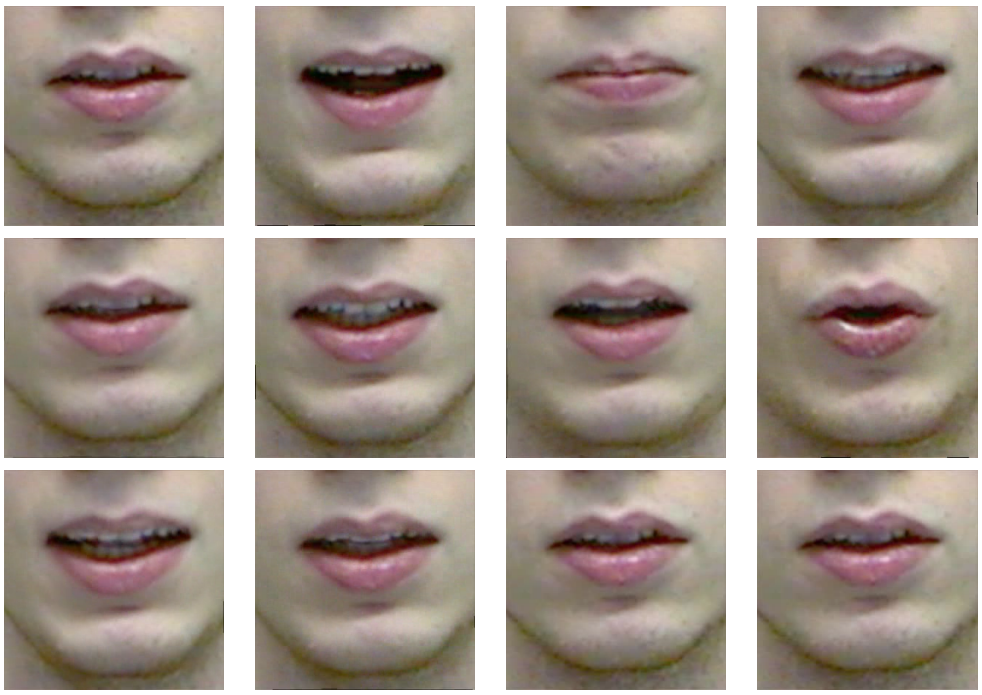


Figure 6.7: Frames from the animation 'lack of money is the root of all evil', generated using the image-based method.

6.5. The first several components (3 or 4, depending upon the labelled data) typically account for over 99% of the variance in the data. The rest of the components are culled providing an efficient representation of viseme geometry. Each component-viseme pair has a related dominance function to control the temporal influence of the viseme over time.

Trajectories through the parametric space of the model are generated using the dominance functions (i.e. according to equations (5.1) and (5.2).) However, this only recovers the geometry of the face at intermediate frames in the animation, $G_{coart}(t)$. Surrounding viseme images in the animation are used to generate the frame itself. Only the two surrounding images (I_i with geometry G_i and I_{i+1} with geometry G_{i+1}, where $t \in [t_i, t_{i+1}]$) are used, with a morph algorithm used to create the blend. This requires two warping functions: $\delta_{\rightarrow} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, which warps I_i such that $G_i = G_{coart}(t)$; and $\delta_{\leftarrow} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, which warps I_{i+1} such that $G_{i+1} = G_{coart}(t)$. These warping functions are created using RBFs (see Section 3.2.2 and Appendix A.1.1.) Once the images are aligned, using the warping functions, an alpha blend is used to combine the warped images, I'_i and I'_{i+1} (6.2).

$$\begin{aligned} I_{coart} &= (1-\alpha)I'_i + \alpha I_{i+1} \\ \alpha &= \tfrac{t-t_i}{t_{i+1}-t_i} \end{aligned} \tag{6.2}$$

A transition between two visemes is shown in fig. 6.6 and synthetic frames from a generated animation[2] in fig. 6.7. RBFs are used here, but any image morphing algorithm could be used to provide these transitions (e.g. [Wolberg, 1998, Beier and Neely, 1992].) The model that is created is similar in nature to *Active Appearance Models* (AAM, [Cootes et al., 1998].) However, PCA is only used to represent

[2] Animations generated using the image-based system can be found in the folder 'animations/section_6.2/' on the accompanying CD.

the geometry, and not the image. A full AAM could equivalently be used, yet other than perhaps data compression there is no particular reason to do so.

Image-based models, such as the described system, have a key advantage in static realism over models which use 3D geometry to model the face. However, the problem with image-based models lie in the modelling of rigid-body transformations. In the image plane these are complex non-linear transforms, which require some method to recover texture not present in the original images. View morphing [Seitz and Dyer, 1996] can be used to improve this, with extra views of the face taken from different angles providing extra necessary degrees-of-freedom. Also, projecting the animation onto a simple 3D object can help [Brooke and Scott, 1998], for small variations in pose. The restriction of pose and expression manipulation to the image plane are important when expressive realism is the goal. For these reasons the remaining systems use 3D meshes to represent the face.

6.3 Constraint-based Synthesis

Instead of using dominance functions to generate trajectories between targets representing visemes, the constrained-optimization technique described in Section 5.3 can be used. This requires that a solution to (6.3) is found given a set of constraints, C_j, on the trajectory.

$$\begin{array}{ll} \textbf{minimize} & Obj(X) = \sum_i \omega_i (S(t_i) - V_i)^2 \\ \textbf{subject to} & \forall j : \underline{b_j} \leq C_j(X) \leq \overline{b_j} \end{array} \tag{6.3}$$

The constraints prevent targets from being met whilst asserting that certain conditions are met (e.g. the start and end expressions.) Details of constraints applied to the speech trajectory can be found in Section 5.3.2. The optimal trajectory matching the requirements of this constrained-optimization problem is found using the *Sequential Quadratic Programming* (SQP) approach described in Appendix A.3.2. SQP can be used as the derivatives of the constraints and objective function are available. Of course, there are combinations of constraints which are unsatisfiable, i.e. they contradict one another. These situations can be prevented by detecting cyclical steps in the optimization. However, relaxation of the global constraint will typically remedy these situations.

The size of system required to generate a speech trajectory will be directly related to two factors: the number of parameters used to model facial expression, and the number of phonemes in the target utterance. The coarticulation of parameters can be assumed to be independent[3] and so (6.3) can be solved for each parameter separately. It is more efficient to break the solution up into sub-problems in this way than to solve one large system. It may be advantageous in the future to add constraints between parameters. This is the case in the original spacetime system [Witkin and Kass, 1988] where joint angles for an articulated figure are interdependent.

The parameterisation used to control facial expression is based upon morph targets. However, to create a more efficient representation, PCA is applied to the data to retrieve the components in fig. 6.10. These components are produced by applying PCA to the geometry (vertices) of the morph targets. Only the region surrounding the mouth is processed, with the tongue separated for the purposes of parameterisation. As with the image-based system, this leads to a reduction in the data, and a parameterisation more closely related to the action of muscle groups on the face. After culling, ~ 20 morph targets can

[3]Dominance functions, after [Cohen and Massaro, 1993], implicitly assume that coarticulation of parameters is independent.

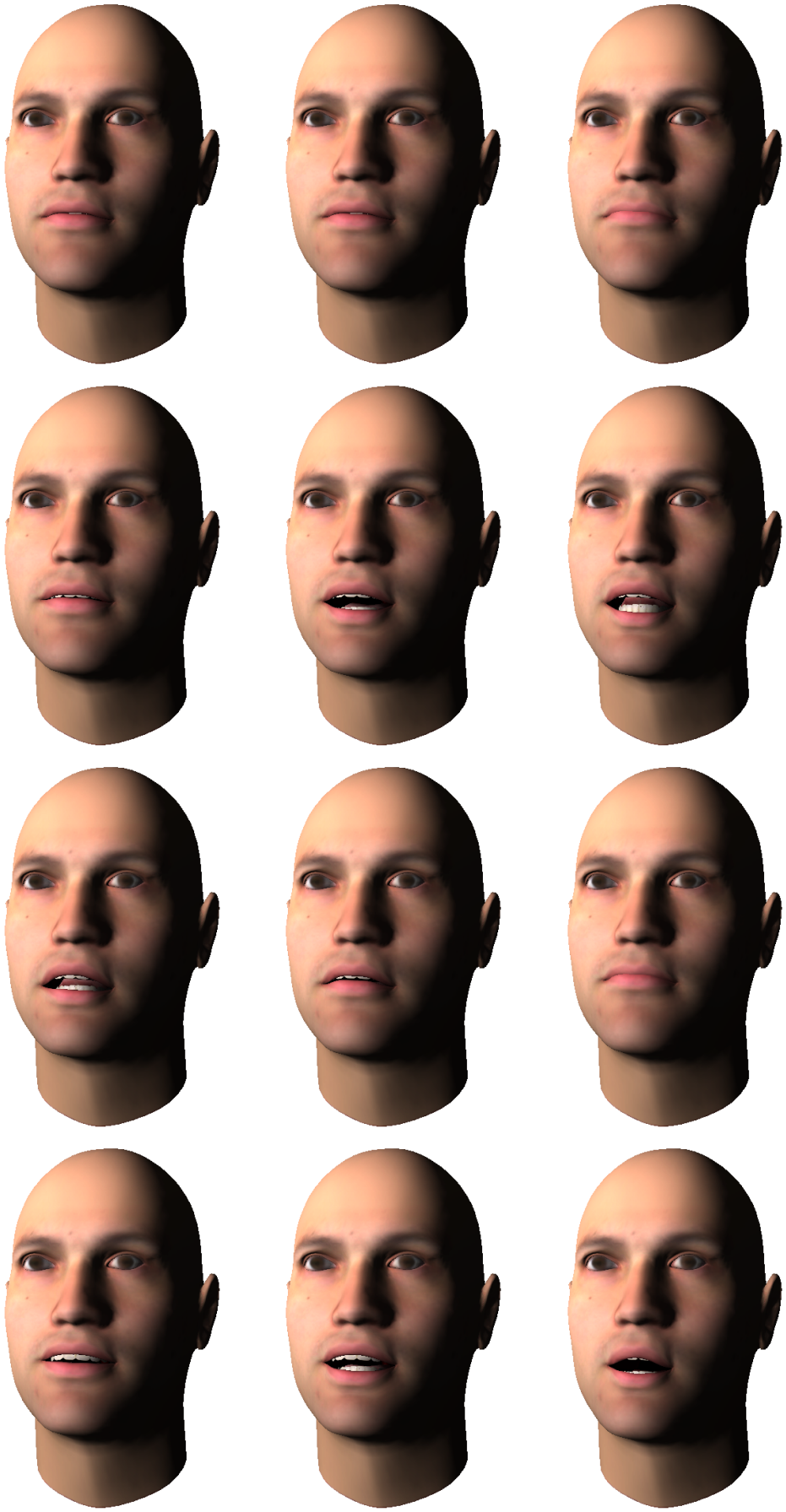


Figure 6.8: Frames from the animation 'I am at two with nature',generated using the constrained-optimization method (i).

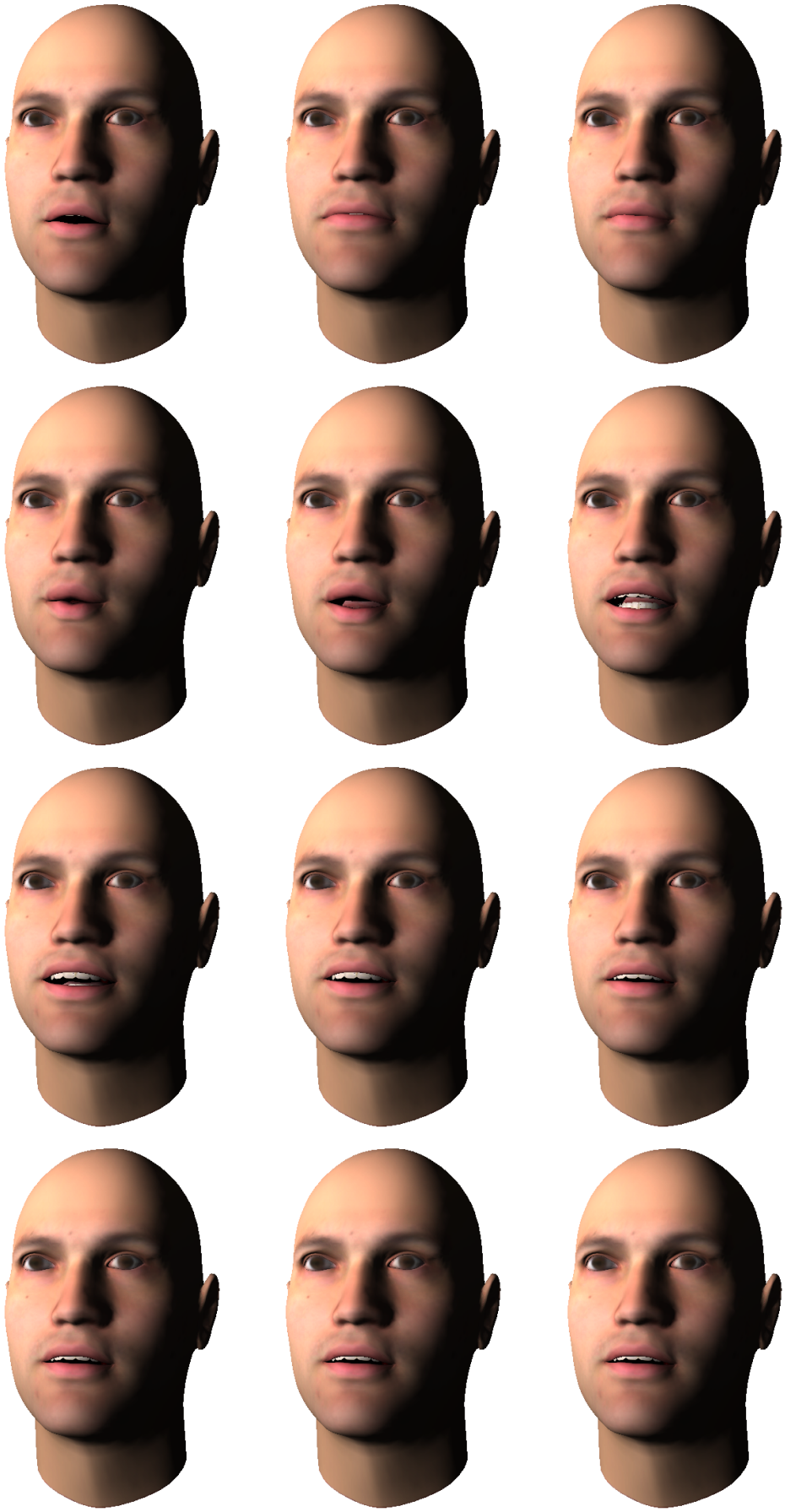


Figure 6.9: Frames from the animation 'I am at two with nature',generated using the constrained-optimization method (ii).

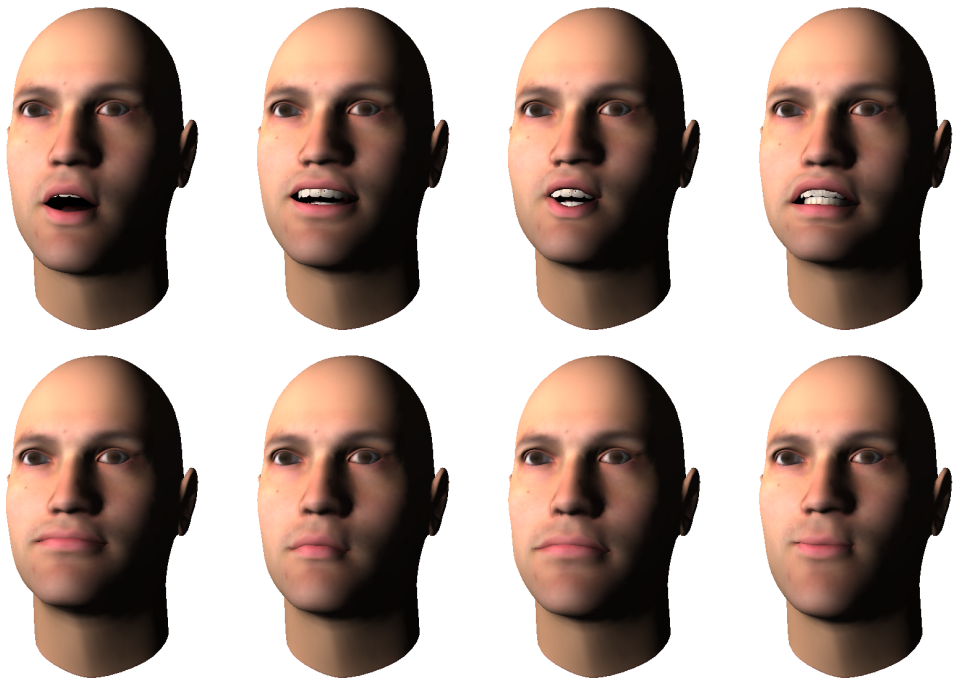


Figure 6.10: First four principal components of the constrained-optimization model: top-row shows $\mu + 3\sqrt{\sigma}$, bottom-row shows $\mu - 3\sqrt{\sigma}$.

be represented accurately with ~ 10 principal components. This is more than the previously described image-based system, but can be accounted for by the three-dimensional nature of the model and the fact that emotional expressions are also included. Thus, to generate an animation 10 separate optimization problems must be solved.

The phonetic structure of a target utterance is also an important factor for this technique. The size of the Hessian (H_{obj} from Section 5.3.4) will grow with the number of viseme targets in the trajectory. Smaller systems are obviously beneficial to solving the problem at an interactive rate. However, splitting up the animation into pieces which are too small will lead to a culling of the effect of coarticulation. Natural phrase boundaries are used to chunk the animation (commas, full stops, etc.), with the assumption that the effect of coarticulation will be least evident at these points. This is a coarse, but necessary, assumption where the synthesis of long utterances is required. A better solution would be to use a windowing approach, by using combinations of sub-problems to generate the entire trajectory. Such a windowing approach has been described in [Cohen, 1992].

Frames from an animation[4] generated using the constrained-optimization technique can be seen in fig. 6.8 and 6.9.

[4] Animations generated using the constraint-based system can be found in the folder ‘animations/section_6.3/‘ on the accompanying CD.

6.4 Limited-domain Synthesis by Unit Concatenation

The most natural audio synthesis techniques concatenate fragments of speech waveforms to generate novel utterances. In an analagous manner small fragments of visual speech movements can be concatenated for visual synthesis (see Section 5.4.) In this system limited-domain synthesis is achieved by concatenating small fragments of motion-captured data. This type of system requires a significant data capture and processing phase before any synthesis can be done.

The data used in this system consists of motion data from a commercial Vicon capture system. High speed cameras, operating at 120Hz, capture the movement of 66 markers on the surface of an actors face plus 7 more on a head mounted jig to capture rigid motion. Audio data is captured simultaneously and synchronized with the motion data. Fifty-five sentences were captured from a limited domain time corpus. The sentences take the form in table 6.2.

Table 6.2: Time-domain corpus.

prompt	:=	{*prolog*} / {*time-info*} / {*day-info*}.
time-info	:=	{*exactness*} {*minutes*} {*hours*}
prolog	:=	'the time is now'
exactness	:=	'exactly' **or** 'just after' **or**
		'a little after' **or** 'almost'
minutes	:=	'five past' **or** 'ten past' **or**
		'quarter past' **or** 'twenty past' **or**
		'twenty-five past' **or** 'half past' **or**
		'twenty-five to' **or** 'twenty to' **or**
		'quarter to' **or** 'ten to' **or** 'five to'
hours	:=	'one' **or** 'two' **or** ... **or** 'twelve'
day-info	:=	'in the morning' **or** 'afternoon' **or**
		'am' **or** 'pm'

This corpus can be used to generate simple time sentences such as:

'*the time is now / exactly one / in the afternoon.*' or
'*the time is now / quarter to ten / in the morning.*'

The data is specific to the time domain, and thus the implemented system is limited in generality. However, as already mentioned in Section 5.4, the same techniques are equally applicable to the general-domain, using smaller units (e.g. diphones/triphones) as the building blocks for synthesis. The simple corpus described is adequate to demonstrate the technique. It is also important to note that increasing the scope of synthesis, and therefore the size of the dataset, hugely increases the time required to capture, label and process the data for use. Consistency in the captured data is of key importance, and the greater the time spent in data capture, the greater the likelihood that there will be inconsistencies in the labelling of the face which would adversely affect the quality of synthesis.

The captured motions require some processing in order to both remove noise and reconstruct missing data. Kalman filtering is used to remove noise from the data. The rigid head motion is also extracted at

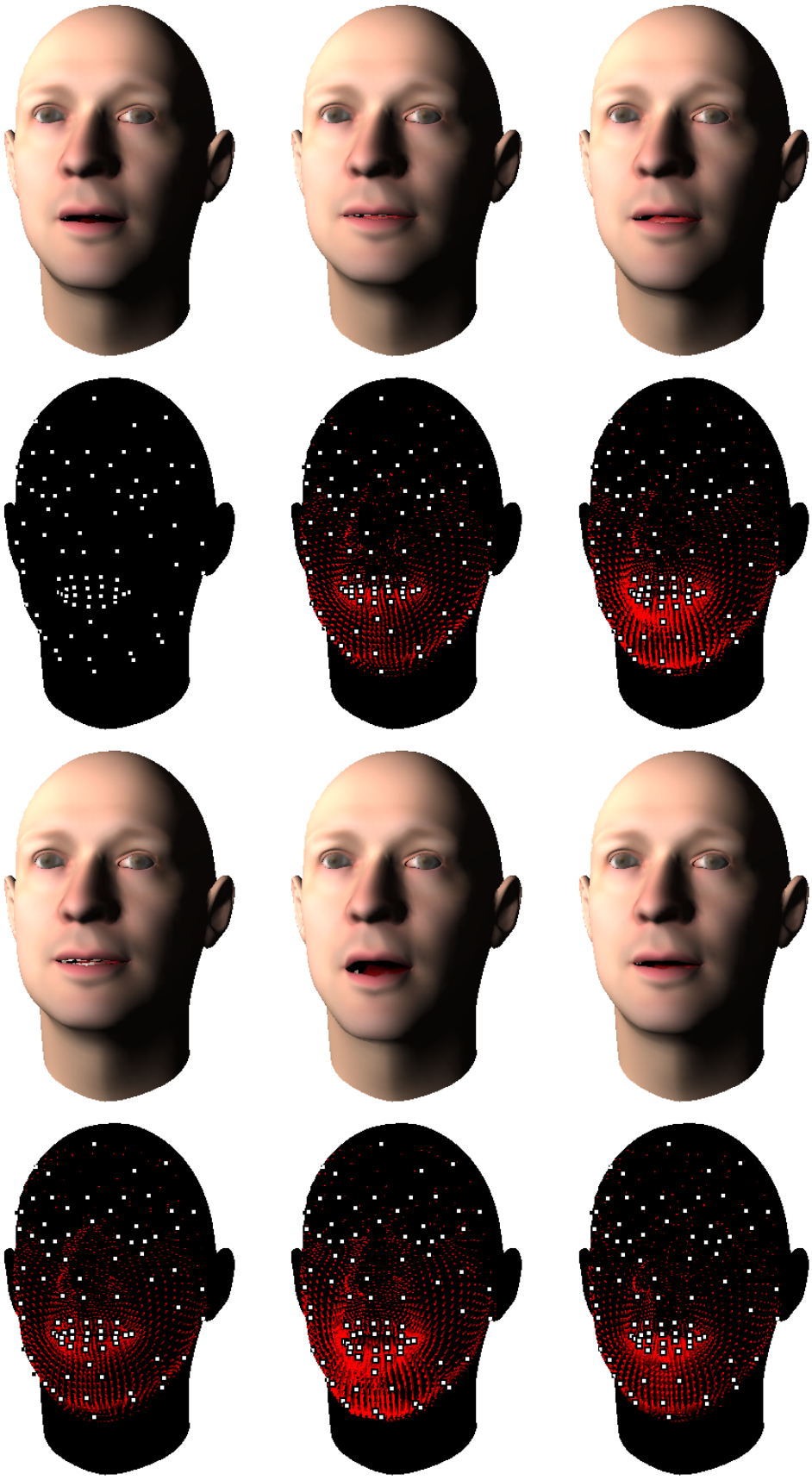


Figure 6.11: Frames from the animation ’the time is now, just after twenty-five to six, in the morning’, generated using motion concatenation (i).

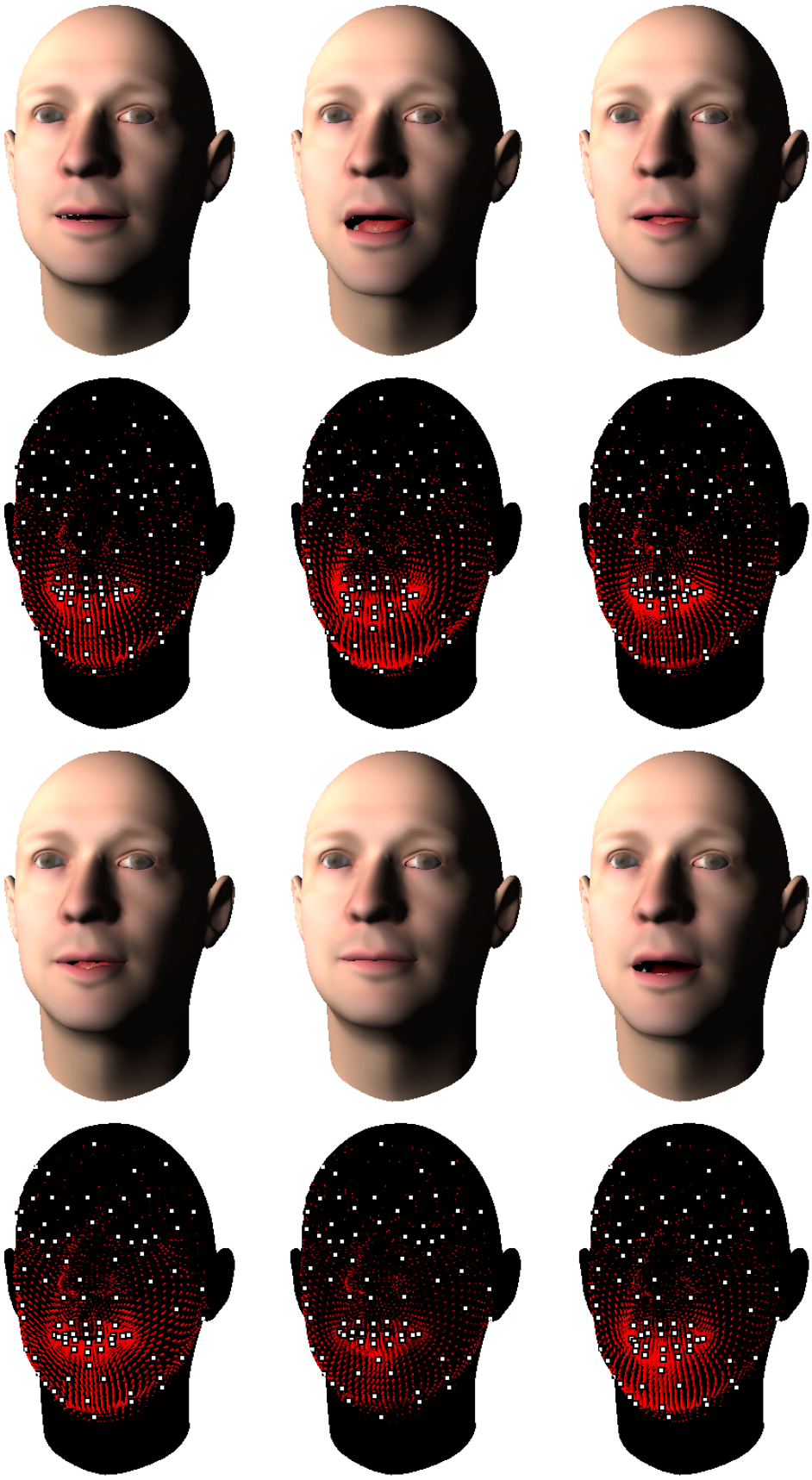


Figure 6.12: Frames from the animation ’the time is now, just after twenty-five to six, in the morning’, generated using motion-concatenation (ii).

this stage using a combination of the estimate from the head mounted jig and a least-squares approach. This last step has the added benefit that the motion samples are initially spatially aligned enabling simpler concatenation during synthesis. Detail on the processing applied to motion data can be found in Section 4.3.

Each of the audio sentences is labelled phonetically. The same labels can be used to provide a transcription of the aligned motion data. This motion data can then be segmented into variable length units consisting of phrase (e.g. '*the time is now*', '*in the afternoon*', etc.) and word length units (e.g. '*one*' → '*twelve*'.) Smaller units can also be used. However, smaller units lead to more concatenation points, and thus lower quality synthesis. For a limited domain system, such as this, phrases and words can be used without unmanagable data-capture and labelling phases. Each unit consists of the frames from the centre of the first phoneme to the centre of the last, with an additional blend period either side of the unit to facilitate the concatenation process.

The synthesis process consists of several steps:

- *Unit Selection* - Appropriate units must be selected from the database to generate the utterance (see Section 5.4.1.)
- *Phonetic Alignment and Resampling* - Each of the selected units must be phonetically aligned such that the movements appear insynchrony with the speech. As a consequence of alignment speech fragments must be resampled to a consistent frame-rate for animation (see Section 5.4.2.)
- *Blending* - Having aligned and resampled the motions, overlapping sections are blended to achieve a consistent trajectory over the synthesized utterance (see Section 5.4.3.)
- *Retargetting and Animation* - A target face model is animated from the synthesized speech movements (see Chapter 4.)

The synthesis is performed in the space of the original actor's face, i.e. before retargetting, and at the original framerate (120Hz) whichtakes advantage of all the data available before it is scaled and manipulated for final animation. Blending the motions at the original framerate is also advantageous because at the final framerate (25-30Hz) there will only ever be a couple of frames in the blend intervals to facilitate the transition between motion fragments. Frames from an animation[5] generated using this method are shown in figures 6.11 and 6.12.

One of the main disadvantages of using motion-capture data for concatenative synthesis is that only the surface of the skin is present in the data. Thus, the movement of the tongue and teeth must either be modelled separately or their motion inferred from the markers on the skin. For the teeth this can be accomplished by recovering the rotation of the chin apex about the jaw axis. The surface of the skin does not directly move with the jaw, but the approximation is accurate enough for the purposes of animation. The same can be done for the rigid motion of the tongue. However, when producing sounds such as *th*ick the tongue must be visibly constraining the flow of air. Thus, a model of tongue deformation is required to make up for this lack in the initial data. In the implemented system a simple morph-based model is used to deform the tongue appropriately. This is linear, and could be better modelled using some form of coarticulation model (see Chapter 5.) However, in most cases the tongue is not very

[5] Animations generated using the motion-based system can be found in the folder 'animations/section_6.4/' on the accompanying CD.

visible, and it is only required that the tongue be in the right position for a small subset of phonemes (e.g. dental fricatives.)

One of the major advantages of motion concatenation is that the generated animations achieve a high degree of dynamic realism. This is due to the non-linear relationship between the motion of markers on the surface of the skin. A similar effect can be achieved using physical models of the skin, but such systems are computationally intensive. The described system can animate speech in real-time. Furthermore, the techniques introduced in Chapter 4 allow the use of motions on any given model, and thus motion concatenation can be used for generating animations with meshes largely different from the actor from whom the original motions were captured.

Chapter 7

Conclusions

In this thesis methods for the synthesis of visual speech, from initial data capture through to final animation, have been introduced. The process of creating a talking head can be split into several areas (discussed in Chapters 3, 4, and 5): modelling, capture, and animation. Most systems for visual-speech synthesis include aspects of all three, but concentrate upon one particular area above all else. Here, several systems have been used to demonstrate a variety of methods for text-to-visual-speech synthesis (see Chapter 6.)

The first system [Edge and Maddock, 2001] acts as a baseline definition for synthesis. Geometric muscle functions are used to deform a 3D polygon mesh to create facial expressions. The animation of speech involves the modelling of visemes representing the extrema of visible articulation. These are interpolated to generate speech movements. Such a system entirely ignores the effect of coarticulation on speech movement. For this reason the animations are over-articulated and unnatural. The short temporal periods between targets do not give the physical system of muscles enough time to reach each of the distinct targets in real speech. This is especially apparent when using interpolation for animation. As some transitions between visemes may only be one or two frames long this leads to non-continuous motion. For this reason, interpolation for generating speech animation is of the lowest quality to be expected from a synthesis system. In fact, worse quality could only be achieved by entirely ignoring the phonetic structure of an utterance. Even though this system represents the lowest level in terms of quality, this is as far as many commercial systems, particularly computer games, ever achieve. However, it should be born in mind that even low quality synthesis can be relatively convincing when there are other visual aspects to draw the attention of a viewer.

Secondly, an image-based talking head has been developed [Edge and Maddock, 2003]. Instead of modelling facial expression in three dimensions, photographic images of a speaker are captured. Animation is produced by generating trajectories through a space bounded by these images. Direct interpolation between targets would produce animations of the same dynamic quality as the muscle-based system. Instead, dominance functions, after [Cohen and Massaro, 1993], are used to generate smooth transitions through the space bounded by the viseme images. In order to use dominance functions with an image-based model of facial expression, *principal components analysis* is used to provide an intermediate parameterisation. The parameterisation of facial expression is an important aspect of creating a talking head. The effect of coarticulation is not easily related to a generic spatial parameterisation (such as vertices or feature-points in an image), and PCA provides a means by which to transform a sam-

pling of speech motion into components related to muscular action (both individual muscles and muscle groups.) Advantageously, this can be recovered directly from the data, and does not require further manual classification of facial expression (in contrast to a scheme such as FACS [Ekman and Friesen, 1978].) Unfortunately, for manual control only the first few principal components, dependent upon the dataset, provide intuitive control of facial expression. Thus, statistical techniques, like PCA, can be useful in providing an intermediary layer of control for animation, but are not necessarily useful for direct control of facial expression.

The main disadvantage of using image-based models of facial expression is that rigid-head motion, and occluded facial features are difficult to model. This is because the only available geometry lies in the image plane. Using three-dimensional techniques, rotation and transformation of the head is simple to model. The only way to do this using images is to capture a sampling of expressions across pose variations and use view interpolation [Seitz and Dyer, 1996]. Another possibility is to project the resulting animation onto simple geometry, as in [Brooke and Scott, 1998]. However, in profile such a model is obviously image-based, and lighting such a model is problematic. An improvement over current facial models would vary both the texture and the geometry of a mesh over time to generate high quality animation. This has already been attempted, but only for playing pre-captured animations [Guenter et al., 1998] and not for synthesis.

As an alternative to the use of dominance functions, a constrained-optimization method for target-based modelling of coarticulation has been introduced [Edge and Maddock, 2004]. The main problems with using dominance functions lie in the use of an abstract dominance domain to control the influence of visemes over neighbouring segments. This is like directly manipulating the basis functions of a spline to control the spline itself. Instead, an optimization technique can be used to attract a trajectory towards a number of targets (the visemes), whilst constraining the trajectory to prevent the targets from being exactly interpolated. This is analogous to the idea of speech production being a target-based system constrained by the physical properties of the vocal tract. Constraints are applied to the system to assert properties upon a generated trajectory. This provides a stronger form of control over the final trajectory than is possible with dominance functions. Also, the global constraint upon the trajectory (i.e. the constraint which prevents the targets from being met) can be used to provide stylistic control over the final utterance. The constraint used here limits the parametric acceleration, and provides a continuous range of possible trajectories between exact interpolation of the targets (over-articulation), and no motion at all (under-articulation.) The extensibility of this method is implicit in the constrained-optimization formulation. Arbitrary constraints can be added, allowing iterative refinement of the method. Also, the objective function and global constraint can be modified to change the properties of convergence. This is in contrast to dominance function methods which can only be modified by substituting different basis functions [Cosi et al., 2003].

The main disadvantage to using an optimization approach is the non-linear nature of the solution. Dominance functions are analytic and thus do not require expensive numerical solutions. However, the described algorithm could be improved by using a moving window approach and only taking into account local context when solving the system (e.g. using a method such as [Cohen, 1992].) This is a natural approach to take given that coarticulation has only been observed over relatively short periods of an utterance [Benguerel and Cowan, 1974]. A windowing approach could also enable larger utterances to be generated without the associated problems of solving large numbers of simultaneous equations.

This would also enable spontaneous speech, whereas currently all articulatory motion must be calculated prior to animation.

The constrained-optimization method also allows a number of possibilities for future research, particularly into the individualisation of speech animation. Global parameters can be used to extend the variation in output speech trajectories (e.g. speech rate, adding emotional expressions, etc.) Another application would be to use this method to fill gaps in a motion-based synthesis system. Boundary constraints could be used to append generated trajectories to captured motions thereby extending the practical use of a set of motion data. It would be difficult to solve similar problems using the standard dominance function approach, and it is this extensibility/flexibility which is the strength of the described technique.

In contrast to target-based models for synthesis, motion-based models attempt to avoid directly modelling coarticulation by appending short fragments of real speech movements. A motion-based model for limited-domain synthesis has been developed [Edge et al., 2004]. This system constructs time-domain utterances from word and phrase length units. Motion-based systems require solutions to unit selection, alignment, and blending to produce continuous speech trajectories for novel utterances. A shifting window approach to unit selection (see algorithm in table 5.2) has been introduced to quickly find appropriate units for synthesis. Alignment is performed by fitting a continuous curve through the sampled motion data, warping the curve to fit the phonetic timings of the target utterance and resampling. Finally, overlapping motion sections are used to blend between fragments and determine the final trajectory. In order to use these trajectories to animate face meshes, which vary significantly in shape and scale from the original actor, a novel facial motion retargetting technique has been used [Sánchez et al., 2003]. *Radial basis functions* are used to warp the space of the original motion to coincide with that of the target mesh. This is a semi-automatic process, requiring only the labelling of a few points on the target mesh. By the application of this retargetting technique the use of motion data can be maximised. This is important because the capture of facial motion data is time-consuming and expensive, especially in comparison with capturing speech audio.

From the described systems we can draw some general conclusions about motion- and target-based synthesis of visual speech. Motion-based synthesis exhibits high dynamic quality in comparison with target-based models. This is to be expected as the initial fragments have been captured from real speakers. However, target-based models are significantly easier to construct, requiring only a few ($\sim 10-20$) visemes to be defined. The size of motion database required to generate general speech is large in comparison. This, along with the expense and difficulties involved in capturing consistent motion data mean that, for the time-being at least, target-based models will remain. Also, target-based models better conform to traditional animation techniques used during the majority of the last century. These are well understood, and by manipulating the targets an animator can directly control the expressive nature of a generated motion. When using motion data there is no direct way to impose stylistic or expressive control over the output motion. It may be the case that future research will lead to a merging of these techniques, e.g. using target-based systems to efficiently encode motion units. Unfortunately, current target-based models cannot represent features of speech not directly related to visemes. Captured motions often include high frequency components which cannot be directly related to the phonetic structure of an utterance, the representation of which is impossible if all parameterisation of speech is directly related to visemes.

One important aspect which has not been discussed in this thesis is the formal evaluation of talking heads. The evaluation of audio speech synthesis has been thoroughly tackled in recent times, and similar work can be conducted into audio-visual speech synthesis. Speech evaluation is usually conducted with regards to naturalness and intelligibility. Naturalness is necessarily an subjective measure of the quality of synthesis, whereas intelligibility can provide an objective measure of how the synthesis technique impacts upon recognition rates. We know, following from [Sumby and Pollack, 1954], that visual speech movements can provide as much as a $+15dB$ improvement in signal-to-noise ratio, which indicates that the intelligibility improvement provided by the visual component of a talking head is measurable. Word recognition rates in increasing audio noise can indicate the intelligibility of a talking head when compared with synthetic audio, natural audio, and natural video. Such an experiment, conducted with a reasonable number of subjects ($\sim 20-30$), would allow the perceptual benefits of the described systems to be directly measured. Naturalness, being subjective, is more difficult to measure. Usually a five-point scale (e.g. Mean Opinion Score: 5 - Excellent, 1 - Bad) is used to measure naturalness, and a comparison with and without the synthetic visual component could be used to evaluate the relative improvement which can be accounted for by the talking head. A thorough evaluation of the systems desribed in Chapter 6 is an important direction of future research. An overview of speech synthesis evaluation techniques can be found in [Lemmetty, 1999].

We can speculate that the main application of talking heads will be in entertainment (films and computer games) and human-computer interaction (HCI.) The needs of these two areas vary significantly. Within HCI a talking head is performing a communicative task where clarity and intelligibility are the most important factors. Within entertainment expressiveness is far more highly regarded, and is required to communicate feelings/emotions and to engage a viewer. It is clear that whilst the ultimate goal of visual speech synthesis is the same, i.e. to accurately render the appropriate articulatory movements for an utterance, yet the techniques used to do this will vary for different applications. Off-line animated film can afford to use computationally intensive physical models of the face and manual animation. In contrast, an interface for a cashpoint, where the number of responses is limited, could use geometric modelling techniques to drive a motion-based synthesis system. The static realism of a modelling technique, the dynamic and expressive qualities of an animation technique, and the expense in capturing the initial data will be the factors which determine how visual speech synthesis is performed for a particular task. There is no single technique which is ideal for all applications. However, none of the current methods deals adequately with the dynamic action of muscles on the skin, or the emotional expressivity of speech. It is the action of physical constraints upon the production of speech and the nature of visual prosody that must be tackled to improve visual speech synthesis in the future.

Appendices

Appendix A

Mathematical Techniques

The following appendices detail several of the methods used in the main body of this thesis for data interpolation, multivariate statistics and the optimization of functions.

A.1 Scattered Data Interpolation

Scattered data interpolation refers to the class of problems where smooth plots, such as splines or surfaces are fitted such that they pass through a set of sampled data points. The data points for these problems are often sparse and unevenly spaced, and the desired properties of the resulting interpolation favour smoothness and affine invariance. From now on we shall refer to the two-dimensional case of curves for simplicities sake, all methods are equally applicable in the three-dimensional (or indeed n-dimensional) case.

The general formulation of the scattered data interpolation problem, given n pairs of data points $p_i, q_i \in \mathbb{R}^d$, is the definition of a continuous function $f : \mathbb{R}^d \to \mathbb{R}^d$ with $f(p_i) = q_i$. The most fundamental concept to this problem is polynomial interpolation. These methods fit a curve to the input data points given that the interpolation constraint must be satisfied, examples being *Aitken's algorithm* and *Cubic Hermite interpolation*. In the following sections we describe methods for data point interpolation based upon the use of *radial basis functions*.

A.1.1 Radial Basis Functions

All forms of data interpolation use a set of basis functions to represent the influence of each segment over the length of the curve, in the linear case this is a simple linear dropoff. Basis functions are *radial* if the value of the function depend only upon the distance from its centre. Thus, *Radial Basis Function* (RBF) interpolation constructs a curve from a linear combination of radial bases (A.1).

$$f(x) = p_m(x) + \sum_{i=1}^{n} \alpha_i \phi_i(d_i(x)) \tag{A.1}$$

In this equation a linear combination of the ϕ_i basis functions weighted by the α_i's are used to determine the resulting interpolation. The value of each ϕ_i basis function is determined solely by the euclidean distance from its centre to the point x, i.e. $d_i(x) = \|x - c_i\|$. Finally, $p_m(x) : \mathbb{R}^d \to \mathbb{R}^d$ is a term which ensures a degree m of polynomial precision; i.e. as $\sum \phi_i \to 0$, $f(x)$ will tend solely to

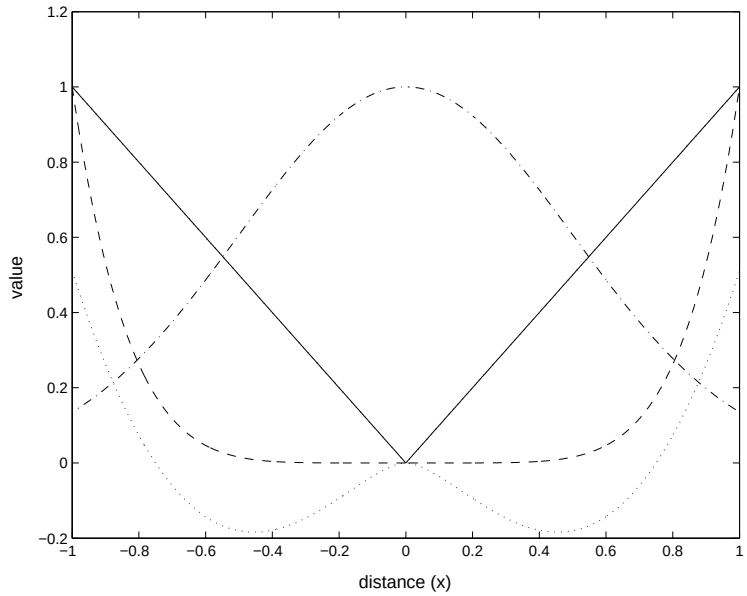


Figure A.1: Radial basis functions: Linear (solid); Cubic Spline (dash); Gaussian (dot-dash), $\sigma = 0.5$; Thin-plate Spline (dotted), $\sigma = 0.75$.

the result of the polynomial term. The polynomial term is simply an affine transformation of the data points. Where there is no polynomial term the RBF is referred to as a *pure radial sum*; however, such models may yield a poor approximation to a curve away from the influence of the basis centres (the result of the interpolation will tend to zero.) Several common basis functions are shown in table A.1 and demonstrated in figures A.1 and A.2.

Table A.1: Radial basis functions.

FUNCTION	$\phi(x)$	CONSTRAINTS
LINEAR	x	-
THIN-PLATE SPLINE	$(x/\sigma)^2 log(x/\sigma)$	$\sigma > 0$
CUBIC SPLINE	$\|x^2\|^3$	-
MULTIQUADRIC	$(x^2+\delta)^{+\mu}$	$\mu > 0, \delta > 0$
INVERSE MULTIQUADRIC	$(x^2+\delta)^{-\mu}$	$\mu > 0, \delta > 0$
GAUSSIAN	$e^{-(x^2/\sigma)}$	$\sigma > 0$

Several of the RBFs in table A.1 include locality parameters which allow fine control of the shape of each basis function. Hardy Multiquadrics are one such example, where exponent μ and locality δ parameters control the spatial influence of each basis centre. Figure A.2 (c) and (d) demonstrate the effect of changing the exponent and locality parameters for the inverse multiquadric; it can be seen that increasing δ leads to a more global function, whilst increasing μ localizes the function but maintains its extent.

The RBFs introduced here are, of course, global in nature. This means that for each interpolated point all basis functions must be taken into account, as in equation (A.1). For examples of ϕ which tend to 0 at some finite distance r from the centre evaluation can be culled. This is obviously more efficient,

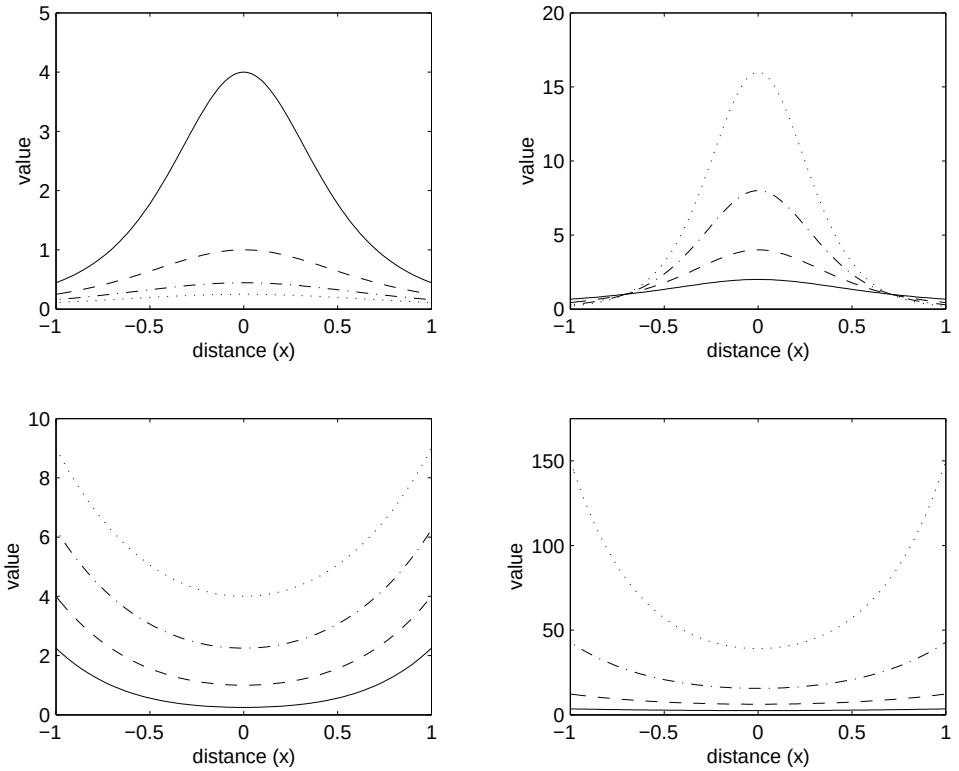


Figure A.2: Hardy Multiquadrics. Top left: Inverse Multiquadric with $\mu = 2$ and $\delta = 0.5$ (solid), 1 (dash), 1.5 (dot-dash), 2 (dotted.) Top right: Inverse Multiquadric with $\delta = 1$ $\mu = 1$ (solid), 2 (dash), 3 (dot-dash), 4 (dotted.) Bottom left: Multiquadric with $\mu = 1$ and $\delta = 0.5$ (solid), 1 (dash), 1.5 (dot-dash), 2 (dotted.) Bottom right: Multiquadric with $\delta = 2.5$ and $\mu = 1$ (solid), 2 (dash), 3 (dot-dash), 4 (dotted.)

and where this strategy is used the basis functions are referred to as *compactly supported* or *locally bounded*. Applying this to the Hardy Multiquadric would yield the function in (A.2).

$$\phi(x) = \begin{cases} (x^2 + \delta)^{\pm\mu} & \text{if } x < r \\ 0 & \textbf{otherwise} \end{cases} \tag{A.2}$$

Examples of interpolated surfaces using Hardy Multiquadrics are demonstrated in fig. A.3.

Constructing the Interpolant

In order to acquire the weights to interpolate a set of input data points a linear system is constructed. This system is formed by placing the points back into equation (A.1), the resulting weights are guaranteed to interpolate the centres of the basis functions (A.3).

$$A = \Phi^{-1} X \tag{A.3}$$

$$A = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \Phi = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \dots & \phi_{1,n} \\ \phi_{2,1} & \phi_{2,2} & & \phi_{2,n} \\ \vdots & & \ddots & \vdots \\ \phi_{n,1} & \phi_{n,2} & \dots & \phi_{n,n} \end{bmatrix} X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{A.4}$$

In this equation $\phi_{i,j}$ is the evaluation of the basis function centered upon the i'th data point given the j'th data point, i.e. $\phi_{i,j} = \phi_i(\|x_i - x_j\|)$. In the d dimensional case there will be $dn + (d+1)$ coefficients, including both the α weights and the coefficients of the polynomial term. The simplest option for the polynomial coefficients would be the identity transform, however this assumes that there is no underlying rigid transformation of the data points. The polynomial coefficients p_i can be determined at the same time as the basis weights given compatibility constraints which ensure that the result of the interpolation reduces to the affine component wherever possible, i.e. it is affine reducible (A.5).

$$\sum_{i=1}^{n} \alpha_i^k = \sum_{i=1}^{n} \alpha_i^k x_i^k = 0 \tag{A.5}$$

In (A.5) x_i^k refers to the k^{th} component of the i^{th} RBF centre, e.g. in two-dimensions $k \in \{1,2\}$ representing the x and y components of each centre. Given these compatibility constraints the α_i weights and components of the polynomial term p_i can be calculated at the same time by solving the system of equations in (A.6).

$$\begin{bmatrix} A \\ p_m \end{bmatrix} = \begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} X \\ 0 \end{bmatrix} \tag{A.6}$$

$$\begin{bmatrix} A \\ p_m \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \\ p_0 \\ p_1 \end{bmatrix} \begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix} = \begin{bmatrix} \phi_{1,1} & \dots & \phi_{1,n} & x_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \phi_{n,1} & \dots & \phi_{n,n} & x_n & 1 \\ x_1^T & \dots & x_n^T & 0 & 0 \\ 1 & \dots & 1 & 0 & 0 \end{bmatrix} \tag{A.7}$$

The linear systems in (A.3) and (A.6) can be solved using any standard technique, such as Gaussian elimination. The α weights and polynomial coefficients can then be placed back into (A.1) and used to interpolate the initial set of data points x.

A.2 Multivariate Statistics

Multivariate statistical techniques are intended to provide the ability to analyze high-dimensional datasets where many variables are correlated with one another. The intent is usually to describe the data using a few highly descriptive variables which are easier to interpret and demonstrate important relationships. Several techniques are commonly used, these include: *factor analysis*, *principal components analysis*, *independent components analysis*, and *singular value decomposition*. These all provide a basis for an input dataset, and are useful in data compression and parameterisation. In the following sections *principal components analysis* and *singular value decomposition* are described in more detail.

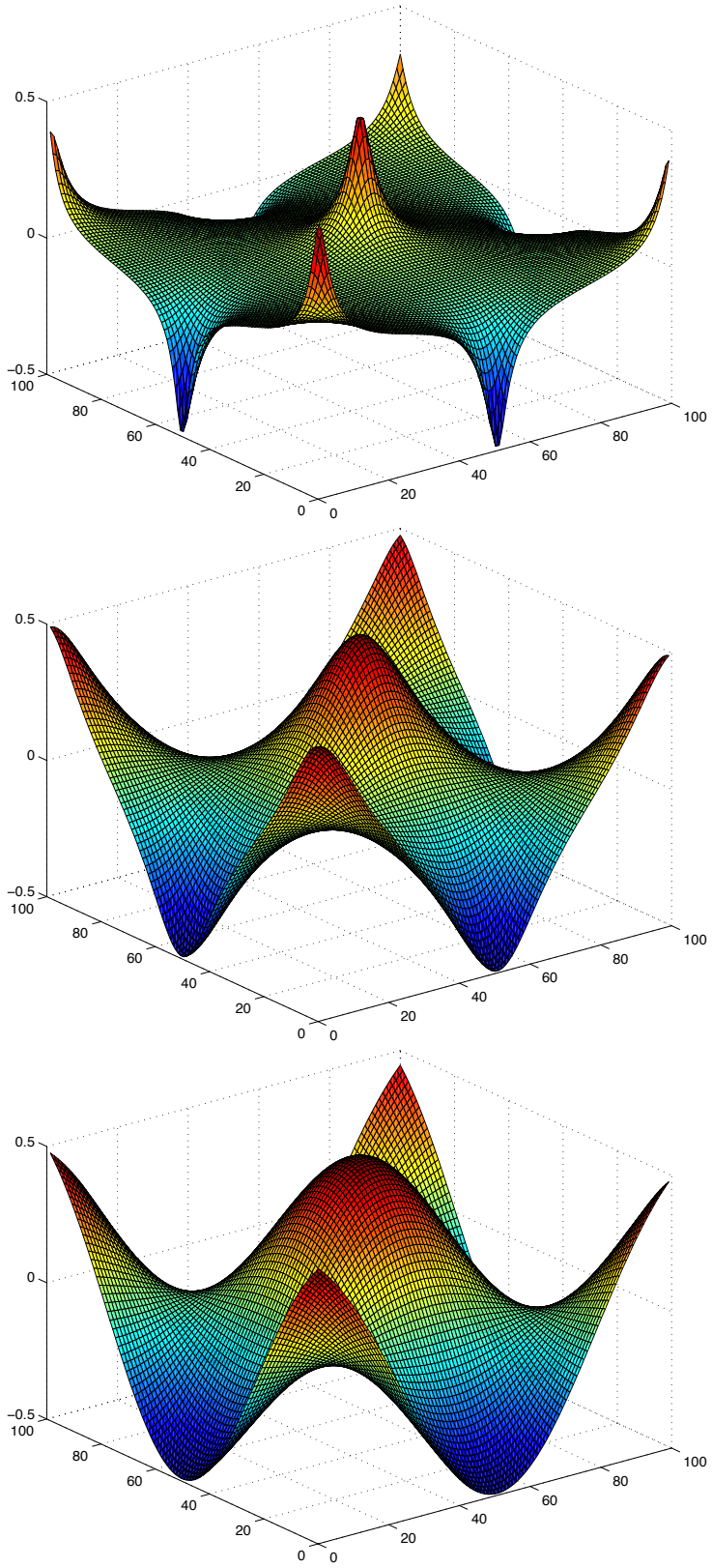


Figure A.3: Interpolated surfaces using Hardy Multiquadrics (top to bottom): $\delta = 0.05, 0.25, 1$.

A.2.1 Principal Components Analysis

Principal Component Analysis (PCA) (also known as the Karhunen-Loeve or Hotelling Transform) is a statistical technique often used in the analysis and compression of datasets consisting of large numbers of interdependent variables. PCA assumes that the data being represented can be expressed as a hyperellipse; the axes of which point in the principal directions of variance and are mutually orthogonal. Describing the initial dataset in terms of its principal components implies rotating the data such that there is no correlation between variables. Using PCA, a randomly sampled vector population $v = \{v_1, \ldots, v_s\}^T$ can be defined in terms of its mean, μ_v, and its principle directions of variation e_i (A.8).

$$v = \mu_v + \sum_{i=1}^{s} e_i b_i \tag{A.8}$$

Where each vector in a population is of length n, there are potentially $n_{pc} = min(n,s)$ principal components, in the case where all variables are mutually independent (and are themselves the principal components.) However, in most cases the number of components required to accurately represent the data will be much less than the number of samples in the initial population, i.e. $n_{pc} \ll s$. This has important consequences for data compression, as truncating the number of principal components can lead to a significant reduction in storage requirements. An example of PCA applied to 2D point data is demonstrated in fig. A.4, notice that the principal components better describe the input data than the initial x and y axes.

PCA makes several assumptions about the data to which it is being applied:

- Gaussian distributed - PCA assumes that the underlying datset is Gaussian distributed, and only under this condition will it yield statistically independent variables.
- Linearity - PCA describes a dataset as a linear combination of components, and thus is not good at representing non-linear relationships.
- Completeness of sample data - As with all multivariate techniques PCA can only represent the relationships in the provided data, and thus a rich input dataset is required to produce a good model.

Usually the first few principal components of a dataset will be highly descriptive, with lower components unintuitive and describing relationships apparent in the input sample data but not in the entire population. Despite this PCA is a commonly used method for data compression and reparameterisation of large sets of variables. It is important to note that principal components are not *statistically independent*, this is a stronger qualification and can be accommodated using *Independent Components Analysis* (ICA, see [Hyvärinen et al., 2001].) A more complete overview of PCA and methods for determining which components to select can be found in [Jolliffe, 1986].

Calculating the Principal Components

The principal components for a dataset can be calculated directly from the covariance matrix, C_v, defined in (A.9).

$$\begin{aligned} \mu_v &= E\{v\} \\ C_v &= E\{(v-\mu_v)(v-\mu_v)^T\} \end{aligned} \tag{A.9}$$

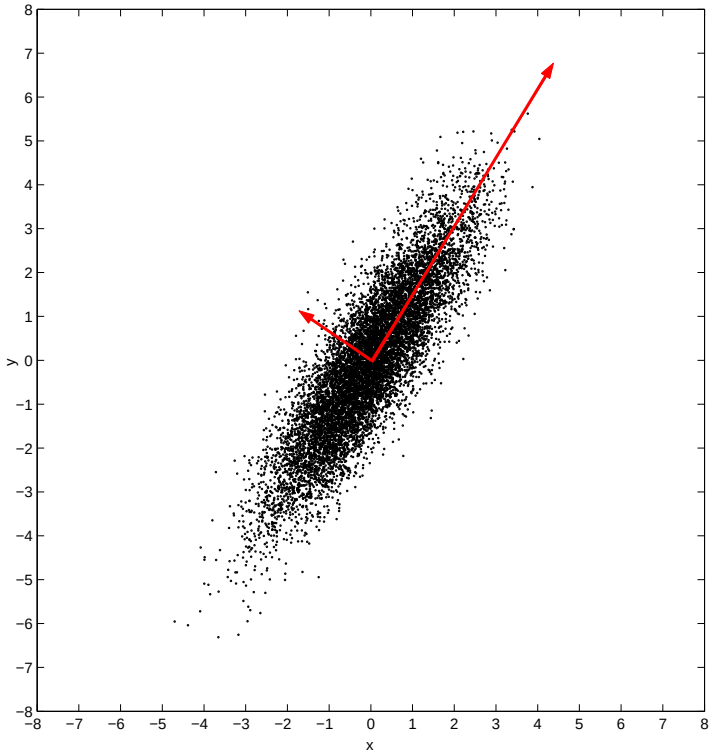


Figure A.4: Principal components (red arrows) of a gaussian distributed point dataset.

The components c_{ij} of C_v contain the covariance between the variable components v_i and v_j. If the components are uncorrelated then $c_{ij} = 0$. As C_v for a given sample population is symmetric it can be used to calculate an orthogonal basis, i.e. the principal components of the dataset. In order to calculate a basis the eigenvalues, λ_i, and eigenvectors, e_i, can be calculated, these are the solutions to the characteristic equation (A.10).

$$\begin{aligned} C_v e_i &= \lambda_i e_i \\ |C_v - \lambda I| &= 0 \end{aligned} \tag{A.10}$$

In (A.10) I is the identity matrix, and $|.|$ denotes the determinant of a matrix. Each of the λ_i contains the variance of the i^{th} principal component, which is the eigenvector e_i. There are many ways of calculating the eigenvalues and associated eigenvectors for a matrix, for a thorough discussion of eigen decomposition routines and the advantages/disadvantages of each technique see [Golub and Van Loan, 1996, Jolliffe, 1986].

A.2.2 Singular Value Decomposition

Another way to calculate a basis for a number of observations is to calculate the *Singular Value Decomposition* (SVD.) The SVD takes a matrix of n samples, X, and decomposes it into three sub-matrices (A.11).

$$X = U\Sigma V^T \tag{A.11}$$

U and V are $n \times r$ and $p \times r$ matrices respectively, where p is the size of each sample vector and r is the rank of X (i.e. the number of non-zero principal components.) Both these matrices are column orthonormal (i.e. $UU^T = I$ and $VV^T = I$.) Σ is a diagonal matrix of length r. In fact this is an efficient way to calculate the principal components of the sample matrix. The principal components (i.e. the eigenvectors e_i from (A.10)) are the columns of V, whilst the diagonal components of Σ are the $\sqrt{\lambda}_i$. An efficient algorithm for applying this decomposition can be found in [William H. Press and Flannery, 1992].

A.3 Optimization

Optimization problems are those where three factors can be identified: firstly, a set of independent variables, X, the values of which must be refined; secondly, a measure of the *goodness* for a particular state of the independent variables, $Obj : \mathbb{R}^d \to \mathbb{R}$, the objective function; and finally, a set of restrictions upon valid system states, $C_i : \mathbb{R}^d \to \mathbb{R}$, the constraints. Given these factors, optimization involves either minimizing or maximizing the result of the objective function by varying the state of the independant variables. Of course there can be problems when maximizing a non-finite function, for example maximizing $Obj(x) = x^2$ would never converge, thus we usually formulate optimization problems as the minimization of an objective function (A.12).

$$\begin{array}{lll} \textbf{minimize} & Obj(X) & X \in \mathbb{R}^d \\ \textbf{subject to} & C_i(X) = 0, & i = 1, 2, \ldots, m'; \\ & C_i(X) \geq 0, & i = m'+1, \ldots, m. \end{array} \tag{A.12}$$

The objective function, $Obj(X)$, and the constraints, $C_i(X)$, are all scalar valued functions. Two different forms of constraints are applicable: equality and inequality constraints. Inequality constraints set bounds upon the set of possible solutions, and can be transformed into equality constraints by introducing a slack variable (A.13).

$$\begin{array}{rl} & C_i(X) \geq 0 \\ \textbf{becomes} & C_i(X) - y_i^2 = 0 \end{array} \tag{A.13}$$

This implicitly constrains $C_i(X)$ to be greater than 0, given that y_i^2 can only be positive. Thus all constraints may be considered uniformly as equalities and the same methods can be applied to both. Alternative methods for enacting inequality constraints can be found in [Gill et al., 1995]. A further distinction to make is between active and inactive constraints. Constraints are only active when for a given state of the system $\tilde{X}$ the constraint $C_i(\tilde{X})$ is violated. Inactive constraints can be disregarded in the solution of the optimization problem until they become active, i.e. are violated by a step in the optimization procedure.

The final solution to an optimization problem will occur once any further improvement would lead to violation of active constraints. Thus, to halt iterative optimization routines requires that the objective step, ΔX_{obj}, be compared to the constraint step, ΔX_{cstr}; i.e. the routine should halt when $\Delta X_{obj} \to -\Delta X_{cstr}$. It is also useful to maintain a count of the number of iterations to detect nonconvergent behaviour. The methods in the following sections can be used to solve general optimization or constrained optimization problems of the form defined in (A.13).

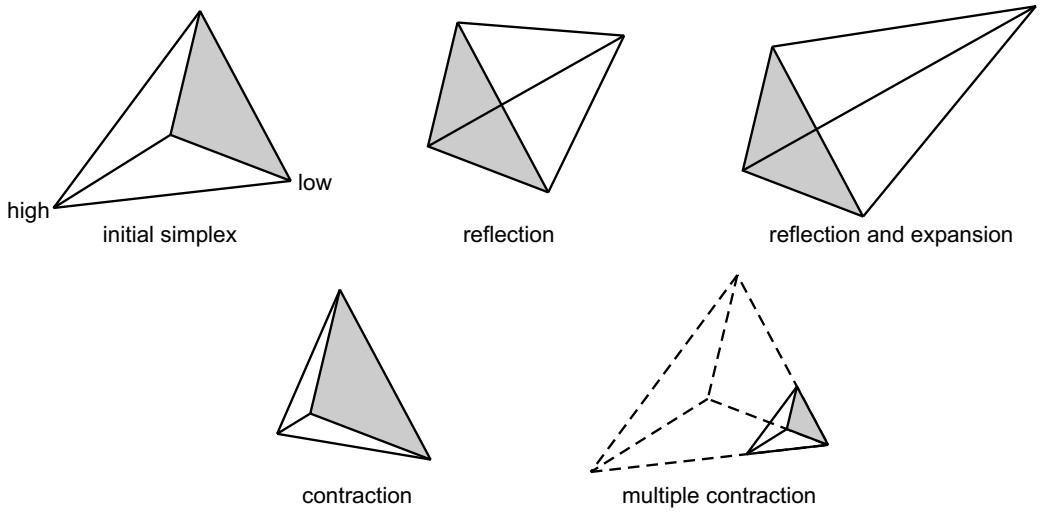


Figure A.5: Steps in the downhill simplex method (after [William H. Press and Flannery, 1992].)

A.3.1 Downhill Simplex

The downhill simplex method is a special form of optimization method as it does not require derivatives of the objective or constraint functions. One important feature is that the optimization requires only function evaluations, and not derivatives of the objective function. The method relies upon the definition of a simplex, a geometric figure consisting of $n+1$ vertices in n dimensions, i.e. a triangle in two-dimensions or a tetrahedron in three-dimensions. In constructing a simplex from an initial point $p_0 \in \mathbb{R}^d$ the remaining vertices, p_i, are defined as offsets by a scalar λ along unit vectors, e_i, usually aligned with the axes of the parameter space (A.14).

$$p_i = p_0 + \lambda e_i \tag{A.14}$$

At each of the vertices the objective function will take on a particular value, thus the simplex is a discrete representation of the objective over a small area of the optimization landscape. In order to perform the optimization the simplex must traverse the optimization landscape from its highest vertices towards its lowest, and therefore more optimal vertices. In this manner the simplex can be visualised as sliding down the slopes of an optimization landscape towards the minima. The simplex proceeds by taking a number of steps depending upon the objective value at each of the vertices, these steps are (see also fig. A.5 for a graphical depiction of the steps) :

- *reflection* - a reflection pushes the worst point through the opposite face towards the minima of the function.
- *reflection and expansion* - if a reflection yields a worse vertex, the vertex is projected back through the face and expanded.
- *contraction* - contraction along one edge from the worst point towards the best.
- *multiple contraction* - contraction of all vertices toward the best vertex.

The downhill simplex method will continue with the above steps until it converges upon a minima. The action of the simplex implicitly calculates the derivatives in a manner similar to the method of finite differences, and is thus sensitive to the same problems with the choice of λ particularly problematic. Simplex methods are inappropriate for non-smooth optimization landscapes, however this can be ameliorated in the case where the procedure is initialized close to the minima and locally the landscape is relatively smooth.

A.3.2 Sequential Quadratic Programming

Sequential Quadratic Programming (SQP) is a generalization upon Newton's method for unconstrained optimization. The variant described here computes a second order Newton step in the objective function, Obj, and a first order step in the C_i constraints. These two steps are combined by projecting the result of the first onto the null space of the second. Contrasting with the downhill simplex method, SQP requires derivatives of both the objective and constraint functions; the *Hessian* of the objective function, H_{obj}, and *Jacobian* of the constraints, J_{cstr}, are defined in (A.15).

$$H_{obj} = \begin{pmatrix} \frac{\partial^2 Obj}{\partial X_1 \partial X_1} & \frac{\partial^2 Obj}{\partial X_1 \partial X_2} & \cdots & \frac{\partial^2 Obj}{\partial X_1 \partial X_n} \\ \frac{\partial^2 Obj}{\partial X_2 \partial X_j} & \ddots & & \frac{\partial^2 Obj}{\partial X_2 \partial X_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 Obj}{\partial X_n \partial X_1} & \frac{\partial^2 Obj}{\partial X_n \partial X_2} & \cdots & \frac{\partial^2 Obj}{\partial X_n \partial X_n} \end{pmatrix} \quad J_{cstr} = \begin{pmatrix} \frac{\partial C_1}{\partial X_1} & \frac{\partial C_1}{\partial X_2} & \cdots & \frac{\partial C_1}{\partial X_n} \\ \frac{\partial C_2}{\partial X_1} & \ddots & & \frac{\partial C_2}{\partial X_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial C_m}{\partial X_1} & \frac{\partial C_m}{\partial X_2} & \cdots & \frac{\partial C_m}{\partial X_n} \end{pmatrix} \tag{A.15}$$

The first step, ΔX_{obj}, optimizes a second order approximation of the objective function (A.16).

$$\Delta X_{obj} = -H_{obj}^{-1} \begin{pmatrix} \frac{\partial Obj}{\partial X_1} \\ \vdots \\ \frac{\partial Obj}{\partial X_n} \end{pmatrix} \tag{A.16}$$

The second step, ΔX_{cstr}, drives the constraints to 0 whilst simultaneously projecting ΔX_{obj} onto the null space of J_{cstr} (A.17).

$$\Delta X_{cstr} = J_{cstr}^{+}(J_{cstr}\Delta X_{obj} - C) \tag{A.17}$$

The matrix J_{cstr}^{+} is the pseudoinverse of J_{cstr}. For a non-square matrix the inverse, J_{cstr}^{-1} cannot be calculated using the standard techniques, this is because the equation no longer has a unique solution. Thus the pseudoinverse is chosen to find the optimal solution. The pseudoinverse can be calculated directly from the SVD (see Appendix A.2.2) (A.18).

$$\begin{aligned} &\textbf{if} \quad && A = U\Sigma V^T \\ &\textbf{then} \quad && A^{+} = V\Sigma^{-1}U^T \end{aligned} \tag{A.18}$$

As Σ is a diagonal matrix of the singular values, the inverse, Σ^{-1}, is the matrix with the reciprocal diagonal elements (A.19).

$$\textbf{if} \quad \Sigma = \begin{pmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \end{pmatrix} \quad \textbf{then} \quad \Sigma^{-1} = \begin{pmatrix} \frac{1}{\sigma_1} & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_3} & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_4} \end{pmatrix} \tag{A.19}$$

Where the inverse, A^{-1}, does exist the inverse and pseudoinverse are the same, i.e. $A^{-1} = A^{+}$. This follows as the inverse will drive the error in $Ax = b$ to zero.

The next step, X_{j+1}, in the SQP method is a linear combination of the optimization and constraint steps (A.20).

$$X_{j+1} = X_j + (\Delta X_{obj} + \Delta X_{cstr}) \tag{A.20}$$

The iteration continues until the optimization criteria are met. This is the variant of SQP described in [Witkin and Kass, 1988]. For a more thorough overview of quadratic programming methods and optimization procedures in general see [Gill et al., 1995].

A.3.3 Simulated Annealing

Simulated Annealing (SA) methods are techniques which apply ideas from the Bolzman probability distribution to global optimization (A.21).

$$prob(E) \approx e^{-E/kT} \tag{A.21}$$

This equation describes the probability of a system being in thermal equilibrium with energy E at a temperature T (k is the Boltzman constant and not significant with regards the SA algorithm.) This allows for a system to be in a high energy state whilst still being at a low temperature, albeit with a relatively low probability. In order to minimize a function it can be advantageous to *not* only travel down the steepest slope, but to sometimes take uphill steps to avoid local minima, and to therefore find a global solution to the problem; this is the same notion as expressed in (A.21).

At each step of the SA algorithm the current state of the system, X, is mutated to explore the space of the objective function, i.e. $X_{mutated} \leftarrow X + \delta_X$. If $X_{mutated}$ is *better* than X (i.e. $E_X > E_{Xmutated}$) then

Table A.2: Simulated annealing algorithm.

Input: List of *init*	**if** $mutantEval < bestEval$ **do**
Output: List of *vars*	$bestEval \leftarrow mutantEval$
	$best \leftarrow mutant$
$vars \leftarrow init$	$vars \leftarrow mutant$
$i \leftarrow 0$	**else if** $BOLZMAN(mutant) = true$ **do**
$best \leftarrow vars$	$vars \leftarrow mutant$
$bestEval \leftarrow EVAL(best)$	**end if**
while $i < maxDepth$ **do**	$i \leftarrow i+1$
$mutant \leftarrow MUTATE(vars)$	**end while**
$mutantEval \leftarrow EVAL(mutant)$	$vars \leftarrow best$

$X \leftarrow X_{mutated}$ and the algorithm continues. This means that in general the algorithm will proceed in a downwards direction towards the minima of the objective function, in a similar manner to the previous methods. If $E_X < E_{Xmutated}$ then a random decision is taken according to the probability distribution in (A.21) to determine whether the mutated state should be kept or not. This ensures that uphill steps may be taken and thus globally optimal solutions can be found. The convergence properties of the SA algorithm are determined by the temperature, T, and the algorithm used to mutate the system state.

A pseudo-algorithm for the SA optimization procedure is shown in table A.2. In this code: *EVAL* returns the result of the objective function; $MUTATE$ mutates the current state according to some procedure, e.g. random variable perturbation; and *BOLZMAN* returns *true* if the mutant, which is less optimal than the best encountered solution, should be kept.

Appendix B

Audio Speech Synthesis

Audio speech synthesis is the generation of audio waveforms that mimic the patterns and vocal properties of natural human speech. Usually synthesis systems provide text-to-speech conversion with the input being unparsed text, possibly with some form of markup language, and the output being a digitally encoded waveform. Two main stages are involved: firstly, text is transformed into some linguistic representation (usually phonemes) which unambiguously represents the sounds in the utterance and processed to determine its prosodic features (duration, intonation, stress etc.); secondly, the linguistic representation and prosodic features are used by the low level synthesis procedure to generate the final waveform.

Linguistic transcription of text requires procedures which are highly language specific. In some languages, such as Spanish, there is a high degree of correlation between the written word and its phonetic representation. However, other languages, in particular British English, require complex transcription procedures. This can be split into several stages (see also fig. B.1):

- Preprocessing - Usually text is preprocessed to expand numbers (e.g. 1750 becomes *one thousand seven hundred and fifty*), abbreviations (e.g. Dr. becomes *doctor*), and special characters (& becomes *and*) into the appropriate words. In some cases this is context dependent (e.g. time vs. currency) or particular to the symbol/abbreviation (e.g. N.A.T.O. is pronounced as written, whereas S.A.S. is pronounced letter-by-letter.)

- Pronunciation - For many words the pronunciation will be fixed for a particular regionalisation of a language. However, some words are so-called *homographs*, that is they have different pronunciation according to context (e.g. 'three *lives* were lost' vs. 'one *lives* to eat'.) Thus context-dependent rules must be produced to deal with these situations. This is particularly the case for proper names, such as the French town *Nice*.

- Prosody - Prosodic features, such as duration and stress, are products of the individual speaking (e.g. because of gender and sex), the emotional content of the utterance (e.g. anger, happiness etc.), and the meaning of the utterance itself (e.g. statement, question etc.) Some of these can be directly determined from or are implied by the input text. Unfortunately, phrase breaks are sometimes not textually indicated and accentuation is rarely indicated. This is important because incorrect prosodic features can entirely change the meaning of a sentence (e.g. 'John says: Peter is a liar' vs. 'John, says Peter, is a liar'.)

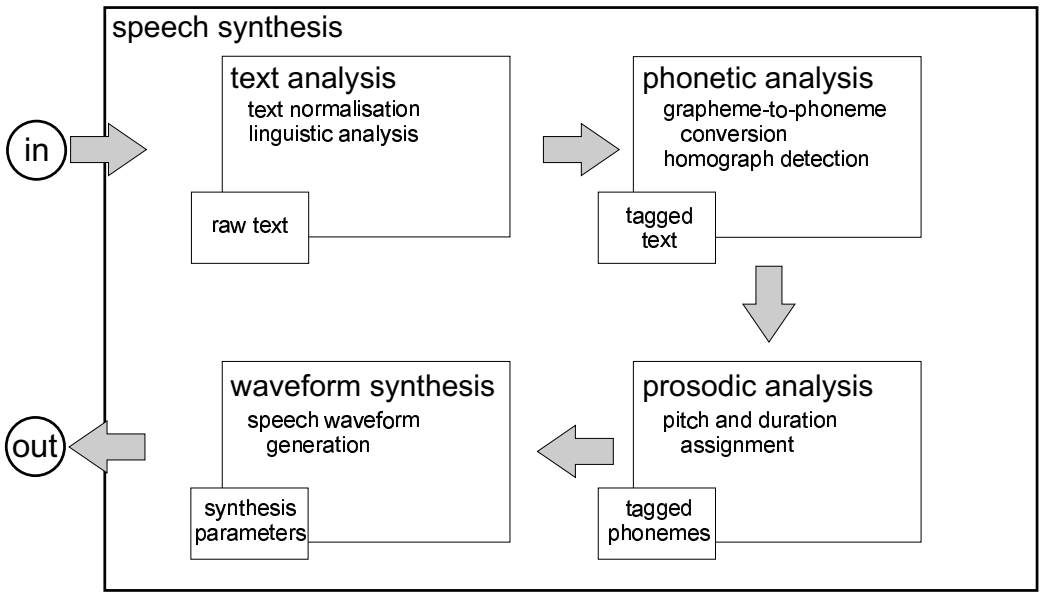


Figure B.1: Sequence of audio speech synthesis processes.

A number of different methods have been described to take the output of the described stages and transform this phonetic/prosodic encoding of speech into a waveform. The methods for low level speech synthesis can be summarised into the following categories: articulatory synthesis, source-filter synthesis, and concatenative synthesis. A thorough discussion of audio speech synthesis technologies can be found in [Lemmetty, 1999].

B.1 Articulatory Synthesis

Articulatory synthesis attempts to produce speech by modelling the vocal tract and the process of speech production itself. Conceptually this is the best means by which to produce natural-sounding audio. However, the complexity of the underlying processes make this a highly difficult challenge and synthesisers of this form this form tend to be computationally intensive.

In order to fulfill the challenge of articulatory synthesis each of the organs in the vocal tract are functionally modelled. One of the first articulatory models used a table of vocal tract area functions between the larynx and lips for each phoneme as the basis for synthesis. Other example parameters for rule-based synthesis include: lip aperture, lip protrusion, tongue tip height, tongue tip position, tongue height, tongue position, and velic aperture. These are clearly related to the phonetic structure of an utterance (as shown by the consonant and vowel classifications in tables 2.3, 2.4 and 2.5.)

By its nature articulatory synthesis is attempting to model the complex three-dimensional system of the vocal tract and its dynamic changes. Necessarily, all current systems simplify the nature of real articulation considerably, and yet are still more complicated than other systems described in this appendix. Few articulatory systems are in development in comparison with those which use formant and concatenative methods. However, it is the case that the nature of the parameters for articulatory models fit best with visual synthesis and this may be a source of research in the future.

B.2 Source-filter Synthesis (Formant Synthesis)

Speech production can be considered as a combination of acoustic source and vocal tract filter (i.e. the source-filter model), and it is this notion on which formant synthesis is based. Parameters and rules which determine the frequency and amplitude of the first several formants[1] and the characteristics of the excitation source are used to control the output speech waveform. Because the formants are related to the filtering caused by the vocal tract by imposing formants at appropriate frequencies synthetic speech can be produced (synthetic music can be produced in a similar manner.)

Formant synthesisers can be either structured as a cascade of resonators, with formants added one after the other to an excitation signal, or in a parallel structure. The cascade structure is considered better for non-nasal voiced sounds, whilst parallel structures are better for nasals, fricatives, and stop-consonants - due to this many formant synthesisers now consist of both parallel and cascade resonator arrays. Newer formant synthesisers may also have parameters to control the source, glottal waveform, and radiation characteristics of the mouth.

B.3 Concatenative Synthesis

Concatenative synthesis works by taking parts of real speech waveforms and combining them to create novel utterances. The parts used depend upon the domain of the synthesis (e.g. unrestricted domain vs. limited domain), and the quantity of available data. For general text-to-speech synthesis the units used are typically much smaller than in limited domain systems, this is because with increased unit size the number of units required will exponentially increase, and thus using syllables for synthesis will require many thousands more units than, for example, diphones.

The size of unit used will impact directly upon the quality of synthesis. This is because less concatenation is being performed, and if these joints coincide with natural word, phrase, or sentence boundaries then the synthetic transitions will be less obvious. The quality of limited domain concatenative synthesis is very high, and is the only method to be commonly used in a commercial situation (e.g. railway anouncements.) General synthesis is of a lower quality, yet still produces the most natural audio of all the methods for synthesis. The reason for this is that the non-stationary aspects of speech are captured well in waveform fragments and prosodic qualities can be included in the data capture.

To generate an utterance from a sequence of waveform fragments a synthetic transition must be produced at a blend region where the fragments overlap. The most common method for this is Pitch-Synchronous-Overlap-and-Add (PSOLA.) In this method short-term signals created by multiplying the original signal with a pitch-synchronous Hanning window are overlap-added to produce the resulting waveform. By separating or compressing the short-term signals in recombination the pitch of the output signal can also be modified. This is Time-Domain PSOLA (TD-PSOLA); other PSOLA methods exist including Linear-Predictive PSOLA (LP-PSOLA) and Frequency-Domain PSOLA (FD-PSOLA.)

The main problem with concatenative synthesis lies in the amount of data that must be captured and labelled *before* any utterances can be generated. Also, by the nature of the waveform fragments, the synthesised utterance will always sound like the individual from whom the data is captured.

[1] Formants are pole frequencies, and antiformants are zero frequencies.

ɒ	*o*dd	u	t*wo*	b	*b*e	k	*k*ey	θ	*th*eta
æ	*a*t	ɛ	*E*d	m	*m*e	d	*d*ee	f	*f*ee
ʌ	h*u*t	ɜ	h*ur*t	p	*p*ee	t	*t*ea	v	*v*ee
ɔ	*ough*t	e	*a*te	tʃ	*ch*eese	h	*h*e	g	*g*reen
ɑʷ	c*ow*	ɪ	*i*t	dʒ	*g*ee	l	*l*ee	s	*s*ea
aɪ	h*i*de	i	*ea*t	ʃ	*sh*e	n	*kn*ee	w	*w*e
ʊ	h*oo*d	o	*oa*t	ʒ	seizure	ŋ	pi*ng*	j	*y*ield
ɔʸ	t*oy*			ð	*th*ee	r	*r*ead	z	*z*ee

Table B.1: English phoneme classification used in Festival.

B.3.1 Festival

Festival is a general multi-lingual concatenative text-to-speech synthesiser developed at Edinburgh University [Black et al., 1999]. The low-level units for general synthesis are diphones. New voices can be created using the FestVox building tools, requiring a database of $\tilde{1}500$ mono-syllabic utterances to create an English voice. The system also provides the ability to define limited domain voices (e.g. time or rail announcements) using a cluster unit selection algorithm (CLUnits.) The phonetic categorisation used by Festival is shown in table B.1.

Bibliography

[Albrecht et al., 2002] Albrecht, I., Haber, J., and Seidel, H. (2002). Speech synchronization for physics-based facial animation. In *Proceedings of WSCG'02*, pages 9–16.

[Angelfors et al., 1999] Angelfors, E., Beskow, J., Dahlquist, M., Granström, B., Lundeberg, M., Salvi, G., Spens, K.-E., and Öhman, T. (1999). Synthetic visual speech driven from auditory speech. In *Proceedings of AVSP'99*.

[Arad et al., 1994] Arad, N., Dyn, N., Reisfeld, D., and Yeshurun, Y. (1994). Image warping by radial basis functions: application to facial expressions. *Computer Vision, Graphics, and Image Processing. Graphical Models and Image Processing*, 56(2):161–172.

[Arslan and Talkin, 1998] Arslan, L. and Talkin, D. (1998). 3-d face point trajectory synthesis using an automatically derived visual phoneme similarity matrix. In *Proceedings of AVSP'98*, pages 175–180.

[Barron et al., 1992] Barron, J., Fleet, D., Beauchemin, S., and Burkitt, T. (1992). Performance of optical flow techniques. *CVPR*, 92:236–242.

[Bartels et al., 1987] Bartels, R., Beatty, J., and Barsky, B. (1987). *An introduction to splines for use in computer graphics & geometric modelling*. Morgan Kauffmann Publishers Inc.

[Beier and Neely, 1992] Beier, T. and Neely, S. (1992). Feature-based image metamorphosis. In *Proceedings of SIGGRAPH'92*, pages 35–42.

[Benguerel and Cowan, 1974] Benguerel, A. and Cowan, H. (1974). Coarticulation of upper lip protrusion in french. *Phonetica*, 30:41–55.

[Berthommier, 2003] Berthommier, F. (2003). Direct synthesis of video from speech sounds for new telecommunication applications. In *Proceedings of SOC'03*.

[Black and Lenzo, 2001] Black, A. and Lenzo, K. (2001). Optimal data selection for unit selection synthesis. In *Proceedings of 4th ISCA Tutorial and Research Workshop on Speech Synthesis*.

[Black et al., 1999] Black, A., Taylor, P., and Caley, R. (1999). The festival speech synthesis system. (`http://www.cstr.ed.ac.uk/projects/festival/manual/festival_toc.html`).

[Blanz and Vetter, 1999] Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *Proceedings of SIGGRAPH'99*, pages 187–194.

[Brand, 1999] Brand, M. (1999). Voice puppetry. In *Proceedings of SIGGRAPH'99*, pages 21–28.

[Bregler et al., 1997] Bregler, C., Covell, M., and Slaney, M. (1997). Video rewrite: driving visual speech with audio. In *Proceedings of SIGGRAPH'97*, pages 353–360.

[Breton et al., 2001] Breton, G., Bouville, C., and Pelé, D. (2001). Faceengine a 3d facial animation engine for real time applications. In *Proceedings of the 6th international conference on 3D web technology*, pages 15–22.

[Brooke and Scott, 1998] Brooke, N. and Scott, S. (1998). Two and three-dimensional audio visual speech synthesis. *Proc. AVSP'98*, pages 213–220.

[Bruderlin and Williams, 1995] Bruderlin, A. and Williams, L. (1995). Motion signal processing. In *Proceedings of SIGGRAPH'95*, pages 97–104.

[Bui et al., 2004] Bui, D., Heylen, D., and Niyholt, A. (2004). Combination of facial movements on a 3d talking head. In *Proceedings CGI'04*, pages 284–291.

[Bulut et al., 2002] Bulut, M., Narayanan, S., and Syrdal, A. (2002). Expressive speech synthesis using a concatenative synthesizer. In *Proceedings of ICSLP'02*, pages 1265–1268.

[Cao et al., 2004] Cao, Y., Faloutsos, P., Kohler, E., and Pighin, F. (2004). Real-time speech motion synthesis from recorded motions. In *Proceedings SCA'04*, pages 225–231.

[Choe et al., 2001] Choe, B., Lee, H., and Ko, H.-S. (2001). Performance-driven muscle-based facial animation. *Journal of Visualization and Computer Animation*, pages 67–79.

[Clough and Tocher, 1965] Clough, R. and Tocher, J. (1965). Finite element stiffness matrices for analysis of plates in bending. In *Proceedings of Conference on Matrix Methods in Structural Analysis*.

[Cohen, 1992] Cohen, M. (1992). Interactive spacetime control for animation. In *Proceedings of SIGGRAPH'92*, pages 293–302.

[Cohen and Massaro, 1993] Cohen, M. and Massaro, D. (1993). Modeling coarticulation in synthetic visual speech. In *Proceedings Computer Animation '93*, pages 139–156.

[Cohen et al., 2002] Cohen, M., Massaro, D., and Clark, R. (2002). Training a talking head. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, pages 499–510.

[Cootes et al., 1998] Cootes, T., Edwards, G., and Taylor, C. (1998). Active appearance models. In *European Conference on Computer Vision*, pages 484–498.

[Coquillart, 1990] Coquillart, S. (1990). Extended free-form deformation: a sculpturing tool for 3d geometric modeling. In *Proceedings of SIGGRAPH'90*, pages 187–196.

[Cosi et al., 2003] Cosi, P., Fuysaro, A., and Tisato, G. (2003). Lucia a new italian talking head based on a modified cohen-massaro labial coarticulation model. In *Proceedings of Eurospeech'03*, pages 2269–2272.

[DeCarlo et al., 1998] DeCarlo, D., Metraxas, D., and Stone, M. (1998). An anthropometric face model using variational techniques. In *Proceedings of SIGGRAPH'98*, pages 67–74.

[Denes and Pinson, 1973] Denes, P. and Pinson, E. (1973). *The speech chain: the physics and biology of spoken language*. Anchor (New York).

[Dutoit et al., 1996] Dutoit, T., Bataille, F., Pagel, V., Pierret, N., and Van Der Vreken, O. (1996). The mbrola project: towards a set of high-quality speech synthesizers free of use for non-commercial purposes. In *Proceedings of ICSLP'96*, pages 1393–1396.

[Edge et al., 2004] Edge, J., Lorenzo, M. S., and Maddock, S. (2004). Reusing motion data to animate visual speech. In *Proceedings of AISB'04*.

[Edge and Maddock, 2001] Edge, J. and Maddock, S. (2001). Expressive visual speech using geometric muscle functions. In *Proceedings of EGUK'01*, pages 11–18.

[Edge and Maddock, 2003] Edge, J. and Maddock, S. (2003). Image-based talking heads using radial basis functions. In *Proceedings of EGUK'03*, pages 74–80.

[Edge and Maddock, 2004] Edge, J. and Maddock, S. (2004). Constraint-based synthesis of visual speech. In *Proceedings of SIGGRPAH'04 Sketches Programme*.

[Eisert et al., 1997] Eisert, P., Chaudhuri, S., and Girod, B. (1997). Speech driven synthesis of talking head sequences. In *Proceedings of 3D Image Analysis and Synthesis*, pages 51–56.

[Ekman and Friesen, 1978] Ekman, P. and Friesen, W. (1978). *Facial action coding system*. Consulting Psychologists Press inc. (Palo Alto).

[Essa, 1995] Essa, I. A. (1995). *Analysis, interpretation, and synthesis of facial expressions*. PhD thesis, Massachusetts Institute of Technology.

[Ezzat et al., 2002] Ezzat, T., Geiger, G., and Poggio, T. (2002). Trainable videorealistic speech animation. In *Proceedings of SIGGRAPH'02*, pages 388–398.

[Ezzat and Poggio, 1999] Ezzat, T. and Poggio, T. (1999). Visual speech synthesis by morphing visemes. Technical Report AIM-1658, Massachusetts Institute of Technology.

[Fagel and Clemens, 2003] Fagel, S. and Clemens, C. (2003). Two articulation models for audio-visual speech synthesis - description and determination. In *Proceedings of AVSP'03*, pages 215–220.

[Farin, 1997] Farin, G. (1997). *Curves and surfaces for computer aided geometric design*. Academic Press.

[Farkas, 1994] Farkas, L. (1994). *Anthropometry of the head and Face*. Raven Press.

[Frank et al., 1997] Frank, T., Hoch, M., and Trogemann, G. (1997). Automated lip-sync for 3d-character animation. In *Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*.

[Frydrych et al., 2003] Frydrych, M., Kätsyri, J., Dobsík, M., and Sams, M. (2003). Toolkit for animation of finnish talking head. In *Proceedings of AVSP'03*, pages 199–204.

[Fung, 1993] Fung, Y. (1993). *Biomechanics - mechanical properties of living tissues*. Springer-Verlag.

[Gill et al., 1995] Gill, P., Murray, W., and Wright, M. (1995). *Practical optimization*. Academic Press Inc.

[Gleicher, 1998] Gleicher, M. (1998). Retargetting motion to new characters. In *Proceedings of SIGGRAPH'98*, pages 33–42.

[Golub and Van Loan, 1996] Golub, G. and Van Loan, C. (1996). *Matrix computations (3rd ed.)*. Johns Hopkins University Press.

[Grassia, 1998] Grassia, F. (1998). Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48.

[Guenter et al., 1998] Guenter, B., Grimm, C., Wood, D., Malvar, H., and Pighin, F. (1998). Making faces. In *Proceedings of SIGGRAPH'98*, pages 55–66.

[Guiard-Marigny et al., 1996] Guiard-Marigny, T., Tsingos, N., Adjoudani, A., Benoît, C., and Gascuel, M. (1996). 3d models of the lips for realistic speech animation. In *Proceedings of Computer Animation'96*, pages 80–92.

[Hällgren and Lyberg, 1998] Hällgren, A. and Lyberg, B. (1998). Visual speech synthesis with concatenative speech. In *Proceedings of AVSP'98*, pages 181–184.

[Henton and Litwinowicz, 1994] Henton, C. and Litwinowicz, P. (1994). Saying it and seeing it with feeling: techniques for synthesizing visible, emotional speech. In *Proceedings of ESCA'94*, pages 73–76.

[Horn and Shunk, 1981] Horn, B. and Shunk, B. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

[Hsu et al., 1992] Hsu, W., Hughes, J., and Kaufman, H. (1992). Direct manipulation of free-form deformations. In *Proceedings of SIGGRAPH'92*, pages 177–184.

[Huang et al., 2002] Huang, F., Graf, H., and Cosatto, E. (2002). Triphone-based unit selection for concatenative visual speech synthesis. In *Proceedings of ICASSP'02*, pages 2037–2040.

[Hyvärinen et al., 2001] Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independant component analysis*. John Wiley and Sons Inc.

[Jolliffe, 1986] Jolliffe, I. (1986). *Principal component analysis*. Springer-Verlag.

[Joshi et al., 2003] Joshi, P., Tien, W. C., Desbrun, M., and Pighin, F. (2003). Learning controls for blend shape based realistic facial animation. In *Proceedings of SCA'03*, pages 187–192.

[Kähler et al., 2001] Kähler, K., Haber, J., and Siedel, H.-P. (2001). Geometry-based muscle modeling for facial animation. In *Proceedings Graphics Interface'01*, pages 37–46.

[Kalra et al., 1992] Kalra, P., Mangili, A., Magnenat-Thalmann, N., and Thalmann, D. (1992). Simulation of facial muscle actions based on rational free form deformations. In *Proceedings Eurographics'92*, pages 59–69.

[Kass et al., 1988] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331.

[Kent and Minifie, 1977] Kent, R. and Minifie, F. (1977). Coarticulation in recent speech production models. *Journal of Phonetics*, 5:115–135.

[King, 2001] King, S. (2001). *A facial model and animation techniques for animated speech*. PhD thesis, Ohio State University.

[King et al., 2000] King, S., Parent, R., and Olafsky, B. (2000). An anatomically-based 3d parametric lip model to support facial animation and synchronized speech. *Proceedings of Deform 2000*, pages 7–9.

[Koch et al., 1998] Koch, R., Gross, M., and Bosshard, A. (1998). Emotion editing using finite elements. In *Proceedings of Eurographics'98*, pages 295–302.

[Koch et al., 1996] Koch, R., Gross, M., Carls, F., Büren, D., Frankhauser, G., and Parish, Y. (1996). Simulating facial surgery using finite element models. In *Proceedings of SIGGRAPH'96*, pages 421–428.

[Koenen, 1999] Koenen, R. (1999). Overview of the mpeg-4 standard. Technical Report ISO/IEC JTC1/SC29/WG11 N2725, Moving Picture Experts Group.

[Krishnamurthy and Levoy, 1996] Krishnamurthy, V. and Levoy, M. (1996). Fitting smooth surfaces to dense polygon meshes. In *Proceedings of SIGGRAPH'96*, pages 313–324.

[Kshirsagar et al., 2000] Kshirsagar, S., Garchery, S., and Magnenat-Thalmann, N. (2000). Feature point based mesh deformation applied to mpeg-4 facial animation. In *Proceedings of Deform 2000*, pages 24–34.

[Kshirsagar and Magnenat-Thalmann, 2000] Kshirsagar, S. and Magnenat-Thalmann, N. (2000). Lip synchronization using linear predictive analysis. In *Proceedings of IEEE International Conference on Multimedia*, pages 1077–1080.

[Kshirsagar and Magnenat-Thalmann, 2003] Kshirsagar, S. and Magnenat-Thalmann, N. (2003). Visyllable based speech animation. In *Proceedings of Eurographics'03*, pages 631–639.

[Kulju et al., 1998] Kulju, J., Sams, M., and Kaski, K. (1998). A finnish-talking head. *Linguistica Uralica*, 3:329–333.

[Lagana et al., 1996] Lagana, A., Lavagetto, F., and Storace, A. (1996). Visual synthesis of source acoustic speech through kohonen neural networks. In *Proceedings of ICSLP'96*, pages 2183–2186.

[Lasseter, 1987] Lasseter, J. (1987). Principles of traditional animation applied to 3d computer animation. In *Proceedings of SIGGRAPH'87*, pages 35–44.

[Lazarus et al., 1994] Lazarus, F., Coquillart, S., and Jancène, P. (1994). Axial deformation: an intuitive technique. *Computer Aided Geometric Design*, pages 607–613.

[Lee et al., 1995] Lee, Y., Terzopoulos, D., and Waters, K. (1995). Realistic modeling for facial animation. In *Proceedings of SIGGRAPH'95*, pages 55–62.

[Le Goff and Benoît, 1996] Le Goff, B. and Benoît, C. (1996). A text-to-audiovisual-speech synthesizer for french. In *Proc. ICSLP'96*, pages 2163–2166.

[Le Goff et al., 1994] Le Goff, B., Guiard-Marigny, T., Cohen, M., and Benoît, C. (1994). Real-time analysis-synthesis and intelligibility of talking faces. In *Proceedings of 2nd ESCA/IEEE Workshop on Speech Synthesis*, pages 53–56.

[Lemmetty, 1999] Lemmetty, S. (1999). Review of speech synthesis technology. Master's thesis, Helsinki University of Technology.

[Lewis and Parke, 1987] Lewis, J. and Parke, F. (1987). Automated lip-synch and speech synthesis for character animation. In *Proceedings of Graphics Interface'87*, pages 143–147.

[Lievin and Luthon, 1999] Lievin, M. and Luthon, F. (1999). Unsupervised lip segmentation under natural lighting conditions. In *Proceedings of ICASSP'99*, pages 3065–3068.

[Löfqvist, 1990] Löfqvist, A. (1990). Speech as audible gestures. *Speech Production and Speech Modelling*, pages 289–322.

[MacCracken and Joy, 1996] MacCracken, R. and Joy, K. (1996). Free-form deformations with lattices of arbitrary topology. In *Proceedings of SIGGRAPH'96*, pages 181–188.

[MacNeilage, 1970] MacNeilage, P. (1970). Motor control of serial ordering of speech. *Psychological review*, 77:182–196.

[Massaro, 1998] Massaro, D. (1998). *Perceiving talking faces: from speech perception to a behavioral principle*. Bradford Books Series in Cognitive Psychology. MIT Press.

[Massaro et al., 1999] Massaro, D., Beskow, J., Cohen, M., Fry, C., and Rodriguez, T. (1999). Picture my voice: audio to visual speech synthesis using artificial neural networks. In *Proceedings of AVSP'99*, pages 133–138.

[McGurk and MacDonald, 1976] McGurk, H. and MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, 264:746–748.

[Na and Jung, 2004] Na, K. and Jung, M. (2004). Hierarchical retargetting of fine facial motions. In *Proceedings pf Eurographics'04*, pages 687–695.

[Noh et al., 2000] Noh, J., Fidaleo, D., and Neumann, U. (2000). Animated deformations with radial basis functions. In *Proceedings of the ACM Symposium on Virtual Reality, Software and Technology*, pages 166–174.

[Noh and Neumann, 2000] Noh, J. and Neumann, U. (2000). Talking faces. In *IEEE International Conference on Multimedia*, pages 627–630.

[Noh and Neumann, 2001] Noh, J. and Neumann, U. (2001). Expression cloning. In *Proceedings of SIGGRAPH'01*, pages 277–288.

[Öhman, 1967] Öhman, S. (1967). Numerical model of coarticulation. *Journal of the Acoustical Society of America*, 41:310–320.

[Parke, 1974] Parke, F. (1974). *A parametric model for human faces*. PhD thesis, University of Utah.

[Pasquariello and Pelachaud, 2001] Pasquariello, S. and Pelachaud, C. (2001). Greta: a simple facial animation engine. In *6th Online Conference on Soft Computing in Industrial Applications*.

[Pelachaud, 1991] Pelachaud, C. (1991). *Communication and coarticulation in facial animation*. PhD thesis, University of Pennsylvania.

[Pighin et al., 1998] Pighin, F., Hecker, J., Lischinski, D., Szeleski, R., and Salesin, D. (1998). Synthesizing realistic facial expressions from photographs. In *Proceedings of SIGGRAPH'98*, pages 75–84.

[Pighin et al., 1999] Pighin, F., Szeliski, R., and Salesin, D. (1999). Resynthesizing facial animation through 3d model-based tracking. In *International Conference on Computer Vision*, pages 143–150.

[Pitermann and Munhall, 2001] Pitermann, M. and Munhall, K. (2001). An inverse dynamics approach to face animation. *Journal of the Acoustical Society of America*, 110:1570–1580.

[Platt and Badler, 1981] Platt, S. and Badler, N. (1981). Animating facial expressions. In *Proceedings of SIGGRAPH'81*, pages 245–252.

[Pyun and Shin, 2003] Pyun, H. and Shin, S. (2003). An example-based approach for facial expression cloning. In *Proceedings of SCA'03*, pages 167–176.

[Quénot, 1992] Quénot, G. (1992). The orthogonal algorithm for optical flow detection using dynamic programming. In *Proceedings of IEEE conference on Acoustics, Speech and Signal Processing*, pages 249–252.

[Ravishankar, 2004] Ravishankar, M. (2004). Sphinx-3 s3.x decoder. (`http://cmusphinx.sourceforge.net/sphinx3`).

[Recasens et al., 1997] Recasens, D., Pallàres, M., and Fontdevila, J. (1997). A model of lingual coarticulation based on articulatory constraints. *Journal of the Acoustical Society of America*, 102:544–561.

[Revéret et al., 2000] Revéret, L., Bailly, G., and Badin, P. (2000). Mother: a new generation of talking heads providing a flexible articulatory control for video-realistic speech animation. *Proceedings of ICSLP'2000*, pages 755–758.

[Ruprecht and Muller, 1995] Ruprecht, D. and Muller, H. (1995). Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications*, 3:37–43.

[Russel, 1980] Russel, J. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178.

[Ruttkay et al., 2003] Ruttkay, Z., Noot, H., and ten Hagen, P. (2003). Emotion disc and emotion squares: tools to explore the facial expression space. *Computer Graphics Forum*, 22(1):49–53.

[Sánchez et al., 2003] Sánchez, M., Edge, J., King, S., and Maddock, S. (2003). Use and re-use of facial motion capture data. In *Proceedings of Vision Video and Graphics'03*, pages 135–142.

[Sánchez et al., 2004] Sánchez, M., Edge, J., and Maddock, S. (2004). Realistic performance-driven facial animation using hardware acceleration. Technical report, University of Sheffield.

[Sánchez and Maddock, 2003] Sánchez, M. and Maddock, S. (2003). Planar bones for mpeg-4 facial animation. In *Proceedings of EGUK'03*, pages 81–88.

[Sederberg and Parry, 1986] Sederberg, T. and Parry, S. R. (1986). Free-form deformation of solid geometric models. In *Proceedings of SIGGRAPH'86*, pages 151–160.

[Seitz and Dyer, 1996] Seitz, S. and Dyer, C. (1996). View morphing. In *Proceedings of SIGGRAPH'96*, pages 21–30.

[Singh and Fiume, 1998] Singh, K. and Fiume, E. (1998). Wires: a geometric deformation technique. In *Proceedings of SIGGRAPH'98*, pages 405–414.

[Singh and Kokkevis, 2000] Singh, K. and Kokkevis, E. (2000). Skinning characters using surface oriented free-form deformations. In *Proceedings of Graphics Interface'00*, pages 35–42.

[Sumby and Pollack, 1954] Sumby, W. and Pollack, I. (1954). Visual contribution to speech intelligibility in noise. *The Journal of the Acoustical Society of America*, 26(2):212–215.

[Summerfield, 1987] Summerfield, Q. (1987). *Hearing by Eye: The Psychology of Lipreading*, chapter Some preliminaries to a comprehensive account of audio-visual speech perception, pages 3–51. Lawrence Earlbaum Associates Ltd.

[Tamura et al., 1998] Tamura, M., Masuko, T., Kobayashi, T., and Tokuda, K. (1998). Visual speech synthesis based on parameter generation from hmm: speech-driven and text-and-speech-driven approaches. In *Proceedings of AVSP'98*, pages 221–226.

[Tao and Huang, 1998] Tao, H. and Huang, T. (1998). Bezier volume deformation model for facial animation and video tracking. In *Proceedings of CAPTECH'98*, pages 242–253.

[Tatham, 1969] Tatham, M. (1969). The control of muscles in speech. Technical report, University of Essex.

[Tibbalds, 1998] Tibbalds, A. (1998). *Three dimensional human face acquisition for recognition*. PhD thesis, University of Cambridge.

[Ulgen, 1997] Ulgen, F. (1997). A step toward universal facial animation via volume morphing. In *Proceedings 6th IEEE International Workshop on Robot and Human communication*, pages 358–363.

[Waters, 1987] Waters, K. (1987). A muscle model for animation three-dimensional facial expression. In *Proceedings of SIGGRAPH'87*, pages 17–24.

[Waters and Levergood, 1993] Waters, K. and Levergood, T. (1993). Decface: an automated lip-synchronization algorithm for synthetic faces. Technical report, DEC Cambridge Research Labs.

[Wickelgren, 1969] Wickelgren, W. (1969). Context-sensitive coding, associative memory, and serial order in (speech) behaviour. *Psychological review*, 76:1–15.

[William H. Press and Flannery, 1992] William H. Press, Saul A. Teulkolsky, W. T. V. and Flannery, B. P. (1992). *Numerical recipes in C: the art of scientific computing*. Cambridge University Press.

[Williams and Katsaggelos, 2002] Williams, J. and Katsaggelos, A. (2002). An hmm-based speech-to-video synthesizer. *IEE Transactions on Neural Networks*, 13(4):900–915.

[Williams, 1990] Williams, L. (1990). Performance-driven facial animation. In *Proceedings of SIGGRAPH'90*, pages 235–242.

[Williams et al., 1995] Williams, P., Bannister, L., Berry, M., Collins, P., Dyson, M., Dussec, J., and Ferguson, M. (1995). *Gray's anatomy*. Churchill Livingston (NY).

[Witkin and Kass, 1988] Witkin, A. and Kass, M. (1988). Spacetime constraints. In *Proceedings of SIGGRAPH'88*, pages 159–168.

[Witkin and Popovic, 1995] Witkin, A. and Popovic, Z. (1995). Motion warping. In *Proceedings of SIGGRAPH'95*, pages 105–108.

[Wolberg, 1998] Wolberg, G. (1998). Image morphing: a survey. *The Visual Computer*, 14:360–372.

[Xu and Prince, 1997] Xu, C. and Prince, J. (1997). Gradient vector flow: a new external force for snakes. In *Proceedings of CVPR'97*, pages 66–71.

[Zhang et al., 2004] Zhang, L., Snavely, N., Curless, B., and Seitz, S. (2004). Spacetime faces: high resolution capture for modelling and animation. In *Proceedings of SIGGRAPH'04*, pages 548–558.