The Use of 3D Viseme Transition Units for Improved Speech Animation

by

Meagan Leflar

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements of the degree of

Master of Applied Science

in

Human Computer Interaction

Carleton University
Ottawa, Ontario

ABSTRACT

Since the 1970s, researchers have endeavoured to recreate speech algorithmically on a digital avatar. Sparked by applications in human-robot interaction, virtual secretaries, web navigation assistance and e-learning, the ever increasing appearance of virtual characters in video games and film has made speech synthesis an area of growth. Due to major challenges such as the sensitivity of viewers to subtle nuance in speech and the complexity of mouth anatomy, realistic speech synthesis has yet to be realized.

A realistic speech synthesis tool could be used in dynamic therapeutic applications as well as revolutionize the animation pipeline in the entertainment industry. In this thesis, we propose a data-driven speech synthesis method that uses Viseme Transition Units (3D animation data describing the transition between mouth shapes) in the stead of the static visemes used in classic data-driven speech synthesis methods. To test this method, viseme transitions were recorded using optical flow and blob tracking algorithms, analyzed, and imported into Autodesk Maya to dynamically animate a custom mouth rig based on user input.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor, Professor Chris Joslin, for his guidance and endless patience, for access to and assistance with the Vicon motion capture lab, and for providing an array of cameras as well as a high-end machine to work on.

I would like to thank all of the Professors from Human Computer Interaction and Information Technology for all the knowledge and support they've gifted me over the years. I also must thank Erenia Oliver for her excellent administrative support, helping me deal with all the paperwork involved in this endeavour, and Chris Clarke for his great tech support.

I am grateful to the twelve volunteers over the course of this thesis who participated as I tried different capture methods. In particular, I thank the five volunteers who gave up an hour of their time to read a very long script and enable me to test the final capture method.

Finally, I thank my close friends and my family for understanding my busy schedule and for supporting me through the challenges and bumps on the road.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

3D Three-Dimensional

API Application Programming Interface

AMA Abstract Muscle Action

ASL American Sign Language

AU Action Units

CVCCVC Consonant-Vowel-Consonant-Consonant-Vowel-Consonant

CVCVC Consonant-Vowel-Consonant-Vowel-Consonant

EFFD Extended Free Form Deformation

FACS Facial Action Coding System

FFD Free Form Deformation

FIX Feature-based Image Transformation

HCI Human Computer Interaction

HMM Hidden Markov Model

MMM Multidimensional Morphable Model

RFFD Rational Free Form Deformation

TTAVS Text to Audio-Visual Speech Synthesis

VFX Visual Effects

VTU Viseme Transition Unit

XML Extensible Markup Language

CHAPTER ONE

Introduction

1.1 Motivation

The human face. We spend most of our lives interacting with it. Speech is an audio visual form of communication that dominates human society and we are very sensitive to its nuances. Attempts to recreate speech algorithmically on a digital avatar started in the 1970s, fueled by dreams of human-robot interaction, virtual secretaries, web navigation assistance and e-learning applications [1, 59]. The ever increasing appearance of virtual characters in video and film, in conjunction with the decreasing expenses of computer processing power, opens up the potential for innovative digital communication metaphors for Human-Computer Interaction (HCI) [24]. For instance; an intelligent and artificial 3D talking avatar called Head X [60].

Digital speech synthesis has attracted a lot of attention in the computer graphics and image processing communities. Accurate speech synthesis in facial animation is quickly evolving into a central feature in computer animation, from software agents to movie production [12]. Prolific visual effects companies like Weta Digital, Industrial Light and Magic (ILM) and Digital Domain use facial animation techniques in their blockbuster releases. This is useful for replacing real actors in dangerous action sequences, making an actor look younger (like Jeff Bridges in *Tron: Legacy* or Arnold Schwarzenegger in *Terminator Salvation*), or to make an actor into something not quite human (such as the

creature Gollum in *The Lord of the Rings* and *The Hobbit* trilogies, or the apes in *Dawn of the Planet of the Apes*).

Figure 1.1 3D Facial Animation from *The Hobbit* and *Dawn of the Planet of the Apes* [68].

Figure 1.2 3D Facial Animation used to replace actors with a younger version of themselves in *Tron: Legacy* and *Terminator Salvation* [69, 70].

There is a great deal of interest in the application of simulated speech animation for dialogue systems in commercial applications [2]. One very popular commercial

application of dialogue systems in the entertainment industry is video games. The videogames industry uses facial animation to animate their characters. Many games now strive for a more video realistic quality to create a sense of greater immersion and investment in their players. *Heavy Rain* by Quantic Dream is a famous example of a game that invested heavily into generating realistic facial animation for their virtual gaming experience. For *Heavy Rain*, Quantic Dream hired actors and used motion capture systems to capture performances, which they then implemented in their highly successful game [71].

Professionals have been animating faces by hand since even before Disney came out with their first cartoon classic in 1937, *Snow White and the Seven Dwarfs*. Now, half a century later, a lot of research has been done into systems that can automate the speech animation process. A perfect system does not yet exist, to achieve realistic results. Any algorithmically generated facial animations must still be fine-tuned by professional animators [12], but this kind of potential automation is an extremely powerful economic motivator. While it may take a team of professional animators months to produce a short animation segment by hand, costing the company hundreds of thousands of dollars, an automated system for such a complex task could cut down time and manpower.

Outside of the entertainment industry, leading professionals believe that digital speech synthesis can change lives in its applications in medical science and forensic analysis [27]. Visual speech synthesis can be used to contribute to the quality of life of hearing impaired individuals. It can be used as a training method for lip-reading, as well as an additional layer of communication, building on American Sign Language (ASL)

systems [31]. The hope is to alleviate some of the challenges that the deaf and hearing impaired must face in their day to day lives, but this research can also be applied to the broader medical field with potential applications including the development of better rehabilitation programs and providing speech therapy. Visual speech synthesis can offer insight into psychophysical and psychological questions pertaining to the study of speech perception, as well as function as a device for the evaluation of theories of human speech production [22].

Automated speech animation could also be used to develop communication systems outside of the medical and disabled communities. Video chat, such as a virtual meeting room, and customer care services are popular application concepts that are examples of communication technology that could be improved by visual speech synthesis [1, 27, 62].

Many of these applications are promising, but this area of research is challenging because humans are attuned to the nuances in the facial dynamics of other humans. No perfect solution has yet been presented, leaving space for new research in this important field.

1.2 Problem Statement

When native English speakers speak English, they automatically formulate sounds correctly without considering the complex array of subtle aural movements they are executing. Speech perception is a form of audio-visual pattern recognition that we have been developing since the day they were born [22]. When they watch a video with poor speech animation, they know intuitively that it does not look quite right although

they may not understand the reason [10]. Some have labeled this phenomenon the believability flip or the uncanny valley [3, 26]. On the quest for realism, researchers develop increasingly realistic animation outputs. The uncanny valley occurs when the product is almost realistic but something subtle is inconsistent, and this leads to a sense of the scene being creepy or not quite right [3, 26].

One particularly well known example of the uncanny valley is in the 2007 animated film rendition of *Beowulf* starring a digital version of prominent actors such as Ray Winstone and Angelina Jolie, among others [3, 72]. More recently, in 2010, *Tron: Legacy* received similar criticism for the digital reconstruction of Jeff Bridges as a younger version of himself, portraying the synthetic character Clu [70]. All of these instances of the uncanny valley describe character animations. Viewers are particularly sensitive to the nuance of human faces, and speech animation is a significant contributing factor to the interpreted realism of a digital character.

Although some recent work has produced fairly realistic results, the uncanny valley still has not been bridged in a significant way. Some recent work also offers fairly fast performance, opening the door for real time applications, but the process for generating realistic facial animation still requires extensive human intervention and tedious fine tuning [24].

Some of the greatest challenges faced when developing a speech animation system include realistically modeling a 3D human head, and effectively capturing and algorithmically reconstructing speech animation. Realistically modeling a 3D human head includes creating a 3D representation that realistically reflects the human face and its capacity to be manipulated to create different facial movements. It can also include

detailed textures, subsurface scattering models (digital materials that can be used to simulate the semi-transparent nature of human skin and underlying tissue), and hair simulation. A good model with good textures does make a good photograph, but realistic movement is what will (or will fail to) breathe life into an animated digital avatar [61]. Therefore textures and modeling will not the focus of this thesis. Instead, we will explore the challenge of collecting data that describes how a person speaks, and how this data can be used to reproduce realistic animation.

Many existing systems approach this deconstruction and reconstruction of facial movement based on the concatenation of different units or states. Units that have been explored primarily fall into two categories: parameters and visemes. Parameters are often face poses described by things like jaw rotation, mouth width, or tapered lower lip “f” tuck for example [56]. To understand visemes, one must first define phonemes. Phonemes are the generally accepted unit of the phonetic system of a language that corresponds to a set of similar sounds [11]. Visemes, then, are the visual representations - the shapes that the mouth forms - of phonemes [10].

The problem that many of these systems face is the transition between one unit, or state, and the next. In speech theory, the way that a phoneme influences phonemes before and after itself in an utterance is called coarticulation. Early systems used a linear relationship to extrapolate transitions from one state to the next. Newer methods use complex morphing algorithms to accomplish this. However, these systems have not managed to effectively recreate realistic coarticulatory behaviour in speech synthesis. This is because transitioning from one viseme to another cannot always be described with a single curve. This transition is often more complex, involving rises and dips that

describe the softness and elasticity of the mouth – stiffness is a common criticism of speech animation – and the unique ways that we form words during speech. An example of this is the bounce that occurs on the inner lip after bilabial consonants. This is the challenge that this thesis addresses. Further animation graphs of different Viseme Transition Units are included in the results section of this thesis.

1.3 Proposed Solution

The system that we propose puts forward the use of the transition between visemes as the synthesized unit instead of the viseme itself. In early speech theory, it was debated whether a viseme was a mouth shape (at the peak of a phoneme for example) or a movement, the whole phoneme or a combination of phonemes [15]. Most speech synthesis systems use visemes as a static mouth shape. The proposed method treats a viseme transition as a series of translations describing mouth movement.

In our prototype of the proposed method, viseme transitions were recorded using optical flow and blob tracking algorithms. Three high-fps cameras, in conjunction with small colour-coded semi-spherical markers that were placed on the subject's lips, captured speech animation data which was recorded as XML files. This procedure was done using a script for participants to read that contains every relevant viseme transition. This animation data is then imported by a python script to dynamically drive a custom mouth rig in Autodesk Maya.

This prototype was specifically designed to support urban Ontario Canadian English. This language classification was created for this study to describe individuals with the same geographical backgrounds, urban central Canada, in order to minimize

the impact of the cultural influence of speech on our results. While this method involves a much larger set of possible nodes than in traditional speech synthesis methods, it takes into account the way that visemes affect each other much more comprehensively. The end goal of the theories tested with our prototype is to create software which accepts text input from the end user and produces realistic speech animation.

1.4 Thesis Contribution

The main contribution of this thesis is the proposed conceptual shift between using visemes as the node in speech synthesis to instead using viseme transitions as this node. We call this new node a Viseme Transition Unit (VTU). The goal of this new unit is to generate more realistic coarticulation in speech synthesis systems. This would produce greater realism and more efficient tools for animation professionals to expedite the speech animation process.

We have also proposed and evaluated a modified phoneme-to-viseme map. This map organizes and groups phonemes based on the mouth shape that they produce until we are left with a list of possible visemes and the associated phonemes that can produce them. We review many existing phoneme-to-viseme maps and have selected and modified a mapping ideal for Canadian English.

Additionally, we have contributed a new marker layout for speech capture for the lips. This marker layout uses seventeen markers that focus on defining the visual aspect of the five lobes of the lips. This system could be used in conjunction with a full facial capture marker system, though for the purpose of this thesis we track only the lips. Also,

these markers are directly reflected in our custom mouth rig which drives a digital avatar.

1.5 Organization of Thesis

Chapter 1 introduces the research project presented in this thesis, giving context to potential applications within the medical, communication, and entertainment industries, as well as giving an overview of the challenges present in the field of speech synthesis and automated facial animation.

Chapter 2 gives an overview of speech theory, as it pertains to Canadian English. We define and explore terms such as viseme, phoneme and coarticulation. Additionally, we review the anatomy of speech, discussing facial anatomy and how it influences the sounds that humans produce. Finally, chapter 2 surveys phoneme-to-viseme mapping techniques that are used in speech synthesis.

Chapter 3 deliberates on the facial animation techniques in literature and in the industry. Although there is a great deal of crossover between methods of speech synthesis, this thesis divides them by delving into the early years of parameterization methods and Free Form Deformation models, muscle and physics-based speech simulation techniques, image-based facial animation techniques, performance driven systems, as well as data-driven systems.

Chapter 4 explains the development methodology for the prototype of our proposed method, and chapter 5 contains the results and evaluation of our system. Finally, chapter 6 discusses our results further in order to draw conclusions and explore future work possibilities.

CHAPTER TWO

Phoneme-to-Viseme Mapping and the Anatomy of Speech

2.1 Phonemes

Phonemes are an abstract unit of speech that represents similar sounds in a phonetic system of a language [11]. Phonemes are the standard unit in many speech recognition systems [14]. Here we will specifically be examining the phonemes of American and Canadian English.

Although there is some small discrepancy in the reported number of phonemes in American English, in recent research it is generally agreed upon that there are 44 phonemes in American English [6, 20, 63]. These phonemes are listed in table 2.1.

Every sound that we make, each phoneme that we utter, is produced as a result of our muscles contracting and interacting with bone and tissue. For this reason, in literature, phonemes are often described based on the physical movements and places of articulation that lead to each sound. The primary places of articulation that we deal with in speech synthesis in facial animation are the lips, teeth, and tongue.

The two primary categories of phonemes are consonants and vowels. Consonants are phonemes that occur when the air flow is fully or partially obstructed [5]. Vowels, conversely, are phonemes that are created when the air is allowed to pass unobstructed through the mouth.

1 *Symbols for phonemes*

ɪ	as in 'pit' pɪt	iː	as in 'key' kiː
e	as in 'pet' pet	ɑː	as in 'car' kɑː
æ	as in 'pat' pæt	ɔː	as in 'core' kɔː
ʌ	as in 'putt' pʌt	uː	as in 'coo' kuː
ɒ	as in 'pot' pɒt	ɜː	as in 'cur' kɜː
ʊ	as in 'put' pʊt		
ə	as in 'about', upper' əbaʊt, ʌpə		
eɪ	as in 'bay' beɪ	əʊ	as in 'go' gəʊ
aɪ	as in 'buy' baɪ	aʊ	as in 'cow' kaʊ
ɔɪ	as in 'boy' bɔɪ		
ɪə	as in 'peer' pɪə		
eə	as in 'pear' peə		
ʊə	as in 'poor' pʊə		
p	as in 'pea' piː	b	as in 'bee' biː
t	as in 'toe' təʊ	d	as in 'doe' dəʊ
k	as in 'cap' kæp	g	as in 'gap' gæp
f	as in 'fat' fæt	v	as in 'vat' væt
θ	as in 'thing' θɪŋ	ð	as in 'this' ðɪs
s	as in 'sip' sɪp	z	as in 'zip' zɪp
ʃ	as in 'ship' ʃɪp	ʒ	as in 'measure' meʒə
h	as in 'hat' hæt		
m	as in 'map' mæp	l	as in 'led' led
n	as in 'nap' næp	r	as in 'red' red
ŋ	as in 'hang' hæŋ	j	as in 'yet' jet
		w	as in 'wet' wet
tʃ	as in 'chin' tʃɪn	dʒ	as in 'gin' dʒɪn

Table 2.1 List of Phonemes and Symbols [6].

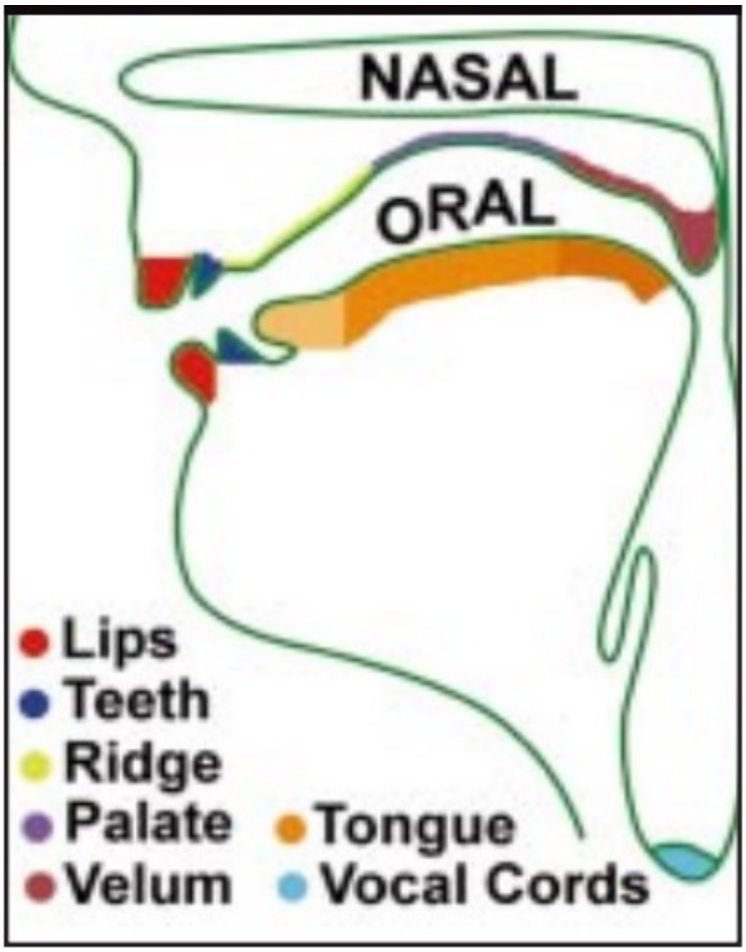


Figure 2.1 Side-cut view of places of articulation [10].

2.1.1 Consonants

Consonants are one of the two main categories of phonemes. Consonants can be broken down and described further by classifying them based on the points of articulation involved and the flow of air through the mouth during their production [5, 6, 10].

Consonants can be described as oral or nasal. Oral consonants include the majority of consonants, and occur when the stopped or partial airflow would occur through the mouth. Sounds like “b” are oral consonants. Nasal consonants involve no

airflow through the mouth, instead the air flows through the speaker's nasal cavity. These are sounds like "m".

Oral consonants can be further broken down into stops, fricatives, affricates, and approximates. A consonant is a stop if the airflow is completely closed off, such as the "b" in "box". A consonant is a fricative if airflow is turbulent or partially obstructed, but not fully closed off, such as the "th" in "the". Affricates are consonants that are a combination of stops and fricatives; for example "j" in "join". Finally, approximates are consonants that are created when the points of articulation create a narrowing of the airflow. This occurs in sounds like "r" in "read" or "w" in "wait".

More specific to their points of articulation, consonants can be described as bilabial, labiovelar, labiodental, interdental, dental, alveolar, palato-alveolar, palatal, velar, and glottal [10]:

- Bilabial consonants are produced when both lips come into contact when producing the sound. "P" and "b" are bilabial consonants.
- Labiovelar consonants occur when the lips are pursed and the tongue lowers at the back of the throat, such as the "w" in "well".
- Labiodental consonants are described by contact between a lip and the teeth; for example "f" or "v" are labiodental consonants.
- Interdental consonants are when the tongue tip is between the teeth during the production of the sound, this covers phonemes like "th" in "this".
- Dental consonants are produced by the tongue tip touching the teeth, such as "l" in "letter".

- Alveolar consonants occur when the tongue tip touches the ridge behind the teeth, creating sounds such as “d” in “deal”.
- Palato-alveolar consonants are produced when the tongue blade (instead of the tip, as described for alveolar consonants) makes contact with the ridge behind the teeth. “Ch” in “cherish” is a palato-alveolar consonant.
- Palatal consonants are characterized by the tongue blade coming into contact with the top of the mouth palate, such as the first “y” in “yesterday”.
- Velar consonants occur when the back of the tongue makes contact with the palate. These are sounds like “g” in “gate”.
- Glottal consonants are produced by a closing off in the throat, such as “h” in “hello”.

Vocal chords, though not a major consideration for visual speech synthesis, do influence the classification of a consonant. A consonant can be voiced or voiceless [6, 10]. A voiced bilabial consonant would be “b” in “boat”, and a voiceless bilabial consonant would be “p” in “park”.

2.1.2 Vowels

The partners to consonants, vowels are the other half of the phoneme family. Unlike consonants, which can be voiceless and voiced, all vowels are voiced. Vowels can be categorized based on the location of the prominent place of articulation [10]. Vowels whose place of articulation is centralized closer to the front of the mouth (the front being characterized as closer to the teeth and lips) include sounds like “i” in “wit”

and “e” in “vet”. Vowels whose place of articulation is closest to the center of the mouth are like the “u” in “rut”. Vowels whose place of articulation is the back of the mouth are the deeper vowels like “oo” in “fool”.

Further vowel classification includes long and short vowels. There are 5 long vowels, and they last longer and are sometimes emphasized more strongly than the 6 short vowels [6]. The long vowels in American English include the “ea” sound in “beat”, the “ir” sound in “bird”, the “a” sound in “pass”, the “oar” sound in “board” and the “oo” sound in “food” [6].

In addition to long vowels, there are vowels which are called diphthongs. These are sounds which include a shift or glide from one vowel to another, creating a unique phoneme that is neither the first vowel nor the second [6]. Diphthongs in American in English are exemplified in words such as “beard”, and “gown”.

Slightly more complex than diphthong vowels are triphthongs. Similar in concept, triphthongs are the combination of numerous vowels that glide or shift into each other, in this case three phonemes, creating a whole new sound [6]. Examples of triphthongs in American English include “layer”, “liar”, and “loyal”.

2.2 Visemes

The term viseme was first coined in 1968 by Fisher as a combination of the words “visual” and “phoneme” [9]. Phonemes are the sound units of speech and visemes are their visual representations; the articulatory gesture (mouth shape) that accompanies the production of the sound.

2.2.1 Phoneme-to-Viseme Mapping

Interestingly, not all phonemes have different visual representations. For example, voiced and voiceless consonants look exactly the same, and are formed by the same lip, tongue and teeth movements [10]. Similarly, all nasal phonemes look alike [10]. For this reason, visemes are often defined as visual references not to individual phonemes, but to groups of phonemes [10].

The use of visemes as an atomic speech unit in visual speech animation is well-established [21], however the range and specification of visemes is not yet standard. Some systems use as few as 5 visemes, and others choose to use one for each phoneme [14, 61].

Typically phoneme-to-viseme mapping is a many-to-one relationship, meaning that one viseme is often used to visualize many phonemes, however some many-to-many and tree-based clustering has also been proposed [21]. Arguably the most famous of the early viseme maps is the Disney 12 [10]. These visemes are the references that Disney used when creating their classic 2D animation work, for which they are famed. 12 visemes may be enough for a friendly 2D cartoon, but the International Lip-reading Association lip-reading samples base their speech recognition on 18 visemes [13].

Figure 2.2 The Disney 12 [10].

Different methods are used to map visemes to phonemes. Cappelletta and Harte did a series of studies in 2011 and 2012 which showed that overall linguistically motivated approaches to phoneme-to-viseme mapping were most intuitive and caused the least confusion to viewers [14, 15]. Jeff Landers, in debating how many phonemes to feature in his system, designed a system optimized for whichever number of visemes the client decided they wished to support [10]. Bozkurt et al. used only 16 visemes, and concluded that a viseme-based unit with contextual information outperforms other models [16]. SoftImage goes all out with their Face Robot system, developing a robust collection of 38 visemes [19]. Face Robot provides a basic library of viseme poses and recommends that users modify, create, and save their own visemes to be used for facial

animation. However, it must be noted that this is a much less automated system. Microsoft seems to find a happy medium with their Microsoft Speech SDK, showcasing a system of 22 visemes which developers and researchers alike can then use to develop their own projects [8, 59].

For the specific phonemes of these maps, please refer to Appendix A.

2.2 Coarticulation

When people speak, they transition from one phoneme to another, one viseme to another. Phonemes and visemes are not speech entities that exist in a vacuum. They are not always perfectly enunciated. Sometimes the phonemes that follow or precede a phoneme will influence how it is formed, and how the transition unfolds. This slight slurring of phonemes is called coarticulation [20]. The length of English vowels in particular is affected according to their context, such as the type of sound that comes before or after them, as well as the presence of stress [6].

When tackling the challenges of automated speech synthesis, coarticulation is central to the apparent realism of the speech animation system [20]. In some early research, it was not addressed at all [61]. In others it was common for the transitions between visemes, the place where coarticulation is most likely to be noticeable, was solved with a linear, cosine or bilinear interpolation function, or morphing methods [29, 50, 45]. It is fairly common for the viseme data to be recorded and then have the transition data be extrapolated from that. In these scenarios, a single image is used to describe each viseme and the system becomes a concatenation these visemes using extrapolated transitions [61]. Another approach is to develop phoneme chaining,

creating a tree structure to define phoneme relationships instead of simply concatenating them to create syllables [5]. This is similar to using viseme labels that are determined using a many-to-many phoneme-to-viseme mapping techniques to compensate for visual coarticulation effects [21] .

The Disney guidelines, used for manual speech animation, use a slightly different viseme for “b”, “p” and “m” if they precede the ea sound (as in “beat”) [10]. Their solution is to add more phonemes based on transitory relationships. This seems to have served them well, but many speech synthesis applications require a great deal of manual tuning to compensate for lack of coarticulation in the model, often relying on “artistic judgement” [10].

Montgomery developed a 2D model for lip shape which allowed computation coarticulation effects for CVCVC segments (Consonant Vowel Consonant Vowel Consonant) based on a system using 130 vectors in real time [23]. This is promising research, but 3 dimensional cues are very important in the perception of realism in speech visualization. The “Look Forward” method is a method that shortens the second “open mouth” viseme in order to simulate coarticulatory effects on speech, and have the speech synthesis look more realistic [18].

CHAPTER THREE

Facial Animation Techniques

3.1 Animation Technique Overview

Lip synchronization is the determination of the motion of the mouth during speech [77]. In this chapter, we explore the different approaches that researchers have taken to facial animation, specifically as it pertains to speech. We will first revisit early techniques that more recent researchers have built upon and included in their own methods. Primarily, these include parameterization and deformation models. Following that, facial animation techniques have been organized into facial animation techniques that are physically and anatomically based, animation techniques that are image-based, performance-driven animation techniques, and finally data-driven animation techniques. While these are the primary approaches to facial animation in literature, the boundaries between these techniques are rather blurred as they are often combined [24].

3.2 Early Research: Parameterization and Deformation

3D speech animation research kicked into high gear in the early 1970s, inspired by the prolific publications of Frederic Parkes proposing a parameterized model, which many researchers have built on since [50]. Slightly later, but also very central to further work on the animation of 3D faces specifically, deformation models explored the ways in which control points can deform a 3D model of a face.

Although Parke's research on parameterization sparked the most follow up research, his theory was not the only one in early animation. In 1978, Erber and

DeFlippo developed a method of simulating lip movement with Lissajous figures (the combination of Lissajous curves) that are displayed on an oscilloscope [48]. The controlled the height and the width of the simulated lips with analog voltage controls.

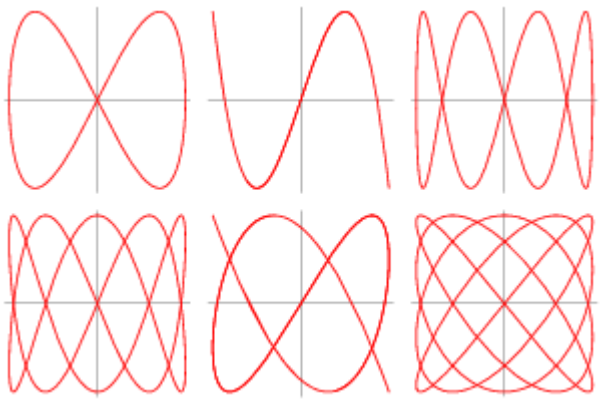


Figure 3.1 Diagram of Lissajous Figures [73].

3.2.1 Parameterized Models

Parameterized models of facial animation are models that parametrically control polygon typology. Examples of potential parameters include features like jaw rotation, mouth width, mouth protrusion (also considered z or depth), tapered lower lip “f” tuck, tapered upper lip raise relative to lower lip, and teeth positioning data [56].

Parke’s paper from 1972 describes the previous generation of facial animation, half-toned animation sequences of a human changing facial expressions, and proposes a 3D polygonal model of the human face, equating polygons with skin [50]. His model contain only 250 polygons, which with today’s computational power is very low but was deemed at the time to be sufficient to achieve a realistic face [50]. The data used to describe the expressions of the face (or “key expression poses”), were acquired by

processing pairs of photographs. Animation between these poses was accomplished using a cosine interpolation scheme [50]. At the time, this facial animation was evaluated as being quite realistic.

Two years later, in 1975, Parke developed a parametric model for the human face [54]. With this model, speech animation is reduced to varying parameters based on a temporal speech sequence. Similar to his previous project, this model uses a 3D polygonal representation of a human face which is manipulated through the use of these parameters which control the translation, rotation, and scaling of various facial features. A major advantage of this parameterization technique is that less information is needed to generate a specific facial expression; one must simply adjust the parameters. With this model, Parke concluded that 10 parameters can be used to produce a reasonably realistic speech facial animation [50].

Nearly a decade later, in 1982, Parke embellished upon his previous parameterization method by developing a system which supports specific facial configuration based on the structure of an individual's face [51]. This method combines parameter values which are related to the conformation of the face, and expression content to drive poses. This gives the model a great deal of flexibility, as it can be modified for any face.

In 1987, Parke teamed up with John Lewis to create an automated method of synchronizing facial animation. This method uses linear prediction, a common speech synthesis method, to first perform phoneme recognition and then to associate the recognized phonemes with mouth positions, driven by a parametric model of the human

face [53]. Parke summarizes his work in 1991 with the development of universal control parameters and interfaces, and specifically how they apply to facial animation [55].

Other researchers have also developed parameterized models. In 1978, Ekman and Friesen published such a model in *Consulting Psychologist* 2. Ekman and Friesen's model is known as a Facial Action Coding System or FACS, and is commonly referenced in newer facial animation research [52]. Their system represents facial expressions in terms of Action Units (AU). An Action Unit can be comprised of one or more muscles and is described by their levels of contraction or relaxation. This method was introduced to widen the range of accurate facial expressions that could be simulated.

Commonly, the sets of parameters in parametric methods describe the lips predominantly. For example, Cohen et al use the following list of parameters for their facial animation method [58]:

- Jaw rotation.
- Mouth x (horizontal) scale
- Mouth z (depth) offset
- Lip corner x (horizontal) width
- Mouth corner z (depth) offset
- Mouth corner x (horizontal) offset
- Mouth corner y (height) offset
- Lower lip "f" tuck
- Upper lip raise

With their research in 1983, Brooke and Summerfield argue that important visual properties of vowel phonemes may not be properly captured by the traditional parameters in parameterized models, specifically referring to descriptions of vertical jaw movements, horizontal and vertical mouth opening, as well as lip shape [49]. They propose that other visual cues, such as the animation of the tongue and the teeth, would produce greater realism in lip synchronization methods.

In 1997, Ali et al use slightly different terminology. The mouth shapes that they describe using parameters are called moments, which they use in their lip synchronization technology [77].

Another one of the prominent criticisms of parameterized facial animation models is that conflict in parameters can cause unrealistic expressions. This often occurs with parameters such as blend shapes, which are used heavily in manual 3D facial animation [25]. Blend shapes are created by manually creating the extreme pose for each parameter, by moving vertices of the 3D mesh for example, and then linearly interpolating between these poses using a controller.

3.2.2 Deformation Methods

Comparably to the parameterized facial animation methods, deformation methods are often a fundamental part of 3D digital facial animation methods. The deformation methods in this section are so central because they allow for deformations of 3D objects using control points, instead of having to manually determine the shift of each individual vertex of the model.

Free-form deformations (FFD) are performed using a cubic lattice of control points, also called a Bezier volume, which envelops the 3D model to be deformed [25, 28]. As the control points are manipulated (meaning their positions are shifted, scaling individual control points does not usually influence the deformation) the 3D model is deformed accordingly.

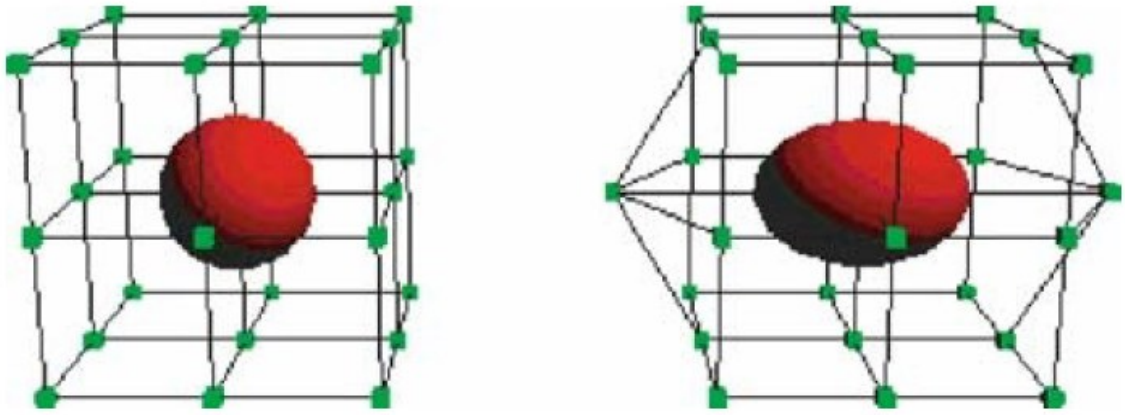


Figure 3.2 Free-form deformation, illustrating the cubic lattice of control points and their influence on the model [25].

Building on free-form deformations are extended free-form deformations (EFFD), which uses a cylindrical lattice for the control points, offering greater flexibility and control than the cubic lattices [25, 28]. Rational free-form deformations (RFFD) provides further control by assigning weights to each control vertex. The weight of the control vertex determines how much control it has over which vertices. If all weights were equal to one, then the RFFD would essentially be an FFD [25]. RFFDs can be used in facial animation to simulate abstract muscle actions [41].

3.3 Muscle and Physics-Based Animation Techniques

To find realism in facial animation, some researchers have endeavoured to recreate the human face in a 3D model as physically accurate as possible. Digital human faces are made by constructing a computational model of the bone structure, the musculature, and the skin layers. To this end, researchers have explored relationships between flexible masses (such as muscles) and rigid (such as bone) ones [31]. There has also been a great deal of research exploring muscle simulation methods.

Abstract muscle action procedure (AMA) is a facial muscle simulation system developed by Magnenat-Thalmann et al [37]. AMAs are described as a similar to an action unit (AU) from the Facial Action Coding System (FACS), and are often used for modeling muscles [41, 51].This system simulates the specific action of a facial muscle, and uses a 3 level process: the first level is the AMA, the second is the facial desired facial expression, and the third is the script driving the expressions. Using this system, they made a synthetic movie.

Waters presented his first muscle model for 3D facial animation at SIGGRAPH in 1987 [38]. He proposes a parameterized facial muscle process that aims to overcome the hard-wired and restrictive nature of existing parameterization methods. The restrictions of facial animation motions should be determined by physically based muscle models, as that limits what would realistically be possible, not based or arbitrary, abstract parameters.

In 1990, Terzopoulos and Waters developed a three layered system of their own, except their three layer system referred to the composition of the facial model itself [35]. The layers of interaction included in their system represent the skin, the fatty tissue, and

the facial muscles, which are combined together to faithfully reflect the anatomical structure of the human face. A year later, Terzopoulos and Waters brought radial laser scanners into the mix to acquire accurate geometry and texture data for modeling faces to be used with their animation techniques, feeling that this lends an extra level of physical accuracy to their systems [33]. Yuencheng Lee joined the team in 1993 to develop an automated system for building physics-based facial models. These models were anatomically correct, and created algorithmically [36].

To study synthetic facial muscle actuators and facial tissue, based on biomechanical theories, Terzopoulos and Waters analyzed video footage of subjects performing expressive articulations [32]. From this analysis they estimated facial muscle contractions, and used this data as their control parameters are part of a parameterized facial animation method.

In 1992 Kalra et al. combined a number of previous theories by proposing the use of rational free-form deformations (RFFD) to simulate abstract muscle actions (AMA) [41]. This system manages the displacement of control points on a defined region of interest, certain muscle groups. Deformation is controlled by changing the weight factors which are assigned to each control point. This approach is often used for modeling muscles [41].

Elastic and visco-elastic models are also used to describe and simulate the properties of skin in facial models that aspire to anatomically accurate [39, 40]. In some systems, a elastic or spring system is extended to muscles as well, particularly when dealing with intertwining muscles [40]. In Kahler's model, muscle contraction and how it relays to the skin is simulated by using a mass-spring system which links the skin,

muscle, and bone layers of the 3D model [40]. Kahler uses this model to create reconstructions from a skull, as well as simulate animation in real time [34]. The animation in this system is driven by deformation methods based on parameters defined with reference to anthropometric standards of facial landmarks [34].

Muscle and physics-based facial animation techniques may lead to a very accurate representation of human speech, but these systems are based on complex models which require heavy processing power. Generally these systems are designed for applications in the medical and therapeutic industry.

3.4 Image-Based Animation Techniques

Borshukov et al. stated in their SIGGRAPH paper that they do not feel that muscle deformers or blend shapes ever work due to the viewer's extreme sensitivity to facial nuances [42]. Their solution, and those of other researchers who have worked on image-based animation techniques, is to use existing video footage or photographic data as part of their visual output.

In their work on the Matrix Reloaded, Borshukov et al. worked with familiar actors to tackle the challenge of creating realistic digital human faces [42]. To accomplish this, they used a 3D recording of the real actor's performance, which meant they could play back the video footage from different angles. Using their photo-realistic rendering method, they could also play the footage back under different lighting conditions. By extracting lighting, geometry, texture, and movement data from the 3D footage, they were able to reconstruct new footage using existing data.

Image-based animation can also be used in conjunction with parameterized facial animation techniques by gathering a collection of images and associating each image with a set of expression parameters [43]. Then, in conjunction with a morphing approach for novel parameters, you can use this parameterized labeling of images to compute the correspondence from one viseme to another [43].

Taking this a step further, Ezzat et al. developed a trainable image-based animation method which uses machine learning techniques to generate speech animation [44]. First they recorded the subject speaking a script that they selected because it contained many of the lip shapes in American English, and then they put this video through an automatic analysis process which “learns” from the data in order to create new phonemes to fill gaps in the recording data [44]. This learned data was then applied to produce new sentences using a Multidimensional Morphable Model (MMM).

Commonly, image-based facial animation techniques require a large database of 2D images [28, 45]. All possible visemes are recorded and the situationally necessary viseme is retrieved from the large database of visemes, in order to be used as part of the synthesis of required facial animation [45]. The visemes from the bank of visemes are then concatenated into the final video sequence by employing image morphing techniques. The challenge is, this does not compensate for 3D perspective changes in viewing the virtual subject as it speaks, or the database becomes much larger.

The ShowFace system, developed by Arya and Hamidzadeh in 2003, is an image-based facial animation system that does not depend on a large database of images [46]. The lack of complex 3D models and heavy database of images makes their system efficient computationally. To accomplish this, they use Feature-based

Image Transformations (FIX), which contain translations for each facial activity. ShowFace supports speech, facial expressions, as well as head movement.

The greatest challenge inherent in image-based facial animation techniques, aside from the large databases of 2D images, are the flexibility constraints. Without an extreme quantity of data sets, differences in lighting and differences in perspective are difficult to account for [17].

3.5 Performance-Driven Animation Techniques

Performance-driven animation techniques are techniques that involve tracking an actor's performance and applying it directly to a digital character. A number of means of performance capture, often referred to as optical flow and motion capture, exist in the industry right now.

Lance Williams described performance driven animation techniques as "CG puppetry" [47]. He advocated that the physical models of muscle and skin that had been devised do not directly address the issue of performance. His work from 1990 describes the process of acquiring the expressions of real faces, and applying them to computer generated faces [47]. He compares this system to an "electronic mask" which offers a mode for actors to be flexibly incorporated into digital scenes, in hopes that this method would more accurately preserve and incorporate human nuance.

Modern motion capture systems, which use multiple depth cameras to track white, semi-spherical markers that are attached to actors, are fairly popular for use in films and story-heavy games. Films praised for their use of motion capture include *The Hobbit* and *The Lord of the Rings* trilogies for their renditions of Gollum and Smaug the

dragon [68]. When animating Smaug, Weta Digital used a combination of a head-mounted camera to capture details of facial movement, and larger body markers within a room of wall-mounted cameras to capture body movement.

Figure 3.3 The animation of Smaug by Weta Digital using motion capture.

Despite that quality motion capture systems cost tens of thousands of dollars, they have begun to cross over from use in Hollywood films into AAA games as well. Games such as *Heavy Rain* and *Beyond: Two Souls* use motion captured footage of actors to create a compelling and interactive digital storytelling experience [71].

Markerless motion capture systems do exist, with varying degrees of accuracy. The Microsoft Kinect supports a rudimentary motion capture system that, while not very accurate or consistent, can be used in real-time for amusing gameplay experiences. The FaceShift system uses a creative parameterized approach to real-time motion

capture [75]. In order to use FaceShift, the system must first be calibrated to each actor. The calibration process takes approximately twenty minutes and involves the actor making extreme facial expressions for each blend shape in the FaceShift system to define their range of motion. Recorded facial expressions are identified as a scaled combination of blend shapes, which are then triggered in the mesh. For example, the expression in figure 3.4 could be described as 100% eyebrow lift and 80% lip pucker. Markerless motion capture systems are not as loyal to the initial actor's nuanced performance as motion capture systems that use markers, however they can be more efficient and used more commonly in real-time.

Figure 3.4 FaceShift markerless real-time motion capture [75].

Performance-driven animation techniques can be an excellent way of capturing human nuance, however they are often very expensive and acquiring skilled actors can be its own hurdle. For each piece of dialogue, new footage must be captured, and more

time must be spent, making performance-driven motion capture one of the more time consuming methods to use as an end product.

3.6 Data-Driven Facial Animation Techniques

Data-driven facial animation techniques can be described as speech synthesis. Often, this is accomplished by the concatenation of segments of recorded speech [20]. This synthesis is usually driven by text input and is referred to as Text to Audio-Visual Speech Synthesis (TTAVS).

Early TTAVS systems, such as the work of Andrew et al from 1986, built on Parke's parameterization models. In this system, one could enter a string of phonemes which would produce the desired animation sequence by converting these phonemes into control parameters [56]. Each phoneme was defined by values for segment duration, segment type (possible types include stop or vowel), and eleven control parameters. These control parameters described items such as jaw rotation, mouth width, mouth depth, tapered lower lip "f" tuck, tapered upper lip raise relative to lower lip, and teeth z and x offsets. In order to transition between phonemes, Andrew et al defined a nonlinear relationship based on transitions speeds specified by the segment type of the phoneme.

Cohen and Massaro also inherited Parke's parameterized model in their publication from 1990 [58]. Their facial animation system manipulated Parke's control parameters to synthesize a sequence of speech articulations. Additionally, they were able to synthesize novel articulations, such as one that is halfways between standard

phonemes (for example, an articulation that is halfway between “ba” and “da”). This speech synthesis was applied to a low-polygon facial mesh that included teeth.

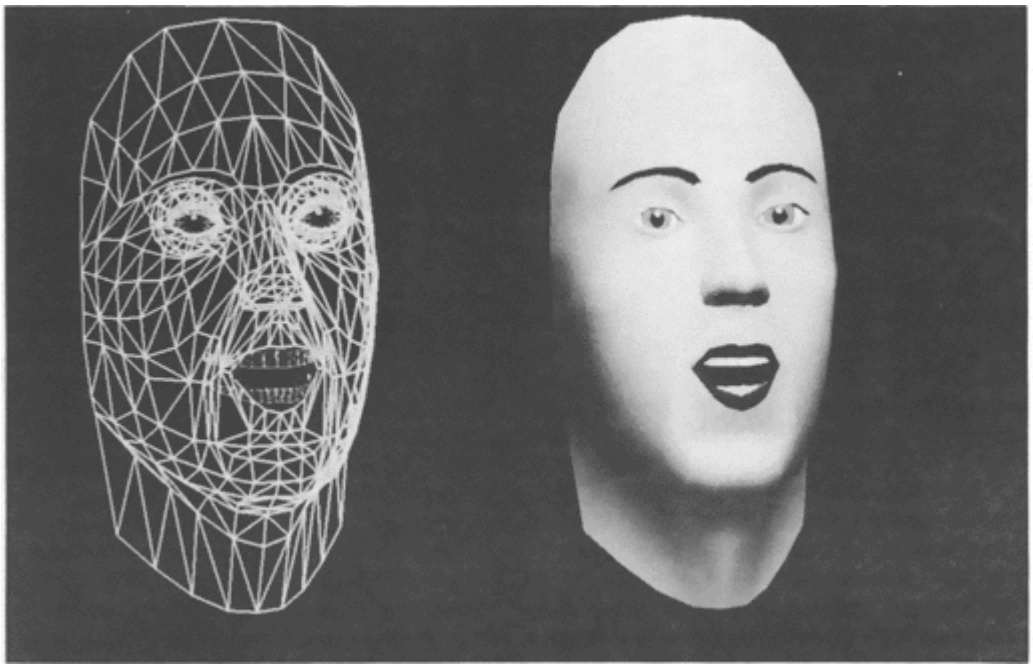


Figure 3.5 Framework and shaded renderings of Cohen and Massaro’s talking head [58].

In 1988, Nahas et al took this work a step further by developing their own parameters based on control points derived by using a scanning device to obtain 3D surface slices of the lips and mouth [57]. Inheriting concepts also from image-based animation techniques, Nahas et al concatenated the acquired images according to the sequence of phonemes desired.

Advancing from the classic parameterization models, some researchers explored the use of muscle-based systems for real-time lip synthesis methods. Provine and Bruton implemented a muscle-based talking head to be used in real-time video conferencing [79]. Aggarwal and Jindal propose a method using recorded audio input

to animate a facial system described using muscle models to drive layers of facial tissue [80]. However, this system also includes manual tweaks in order to accomplish specific visemes.

Ezzat and Poggio continue away from the parameterized and muscle-based models with their speech synthesis method from 1997 [61]. Instead, they recorded participants reading a script which was carefully selected to contain most of the American English viseme images. The team identified and extracted the viseme data by hand from these recordings. Ezzat and Poggio computed the transitions between visemes using a morphing method based to optical flow methods. They utilized the timing information in order to determine which transitions to use, and the rate at which the morphing process should occur.

Goranka Zorić and Igor Pandžić also developed a lip synchronization method using visemes as their unit [78]. Their goal specifically was to develop a real-time method that was not-language specific, meaning it had a collection of visemes that could be used across languages instead of limiting to those used in only one language.

More recently, Rizvic et al worked to a script for Autodesk 3D Studio Max which creates key frames based to a pre-recorded soundtrack, taking the process a step further than text-to-speech and deriving phonemes from audio cues [12]. Ali et al focused their research more heavily on where people drop phonemes when talking in order to create realistic speech. Their system was implemented using a small selection of 10 visemes that were hand animated and then concatenated using a MEL script. Their work involved a great deal of artistic license and hand animating [76].

Many data-driven facial animation methods require that facial animations be manually fine-tuned by adding expressions and emotions in order to achieve realistic results. Another drawback to these methods is that often a single image, or set of parameters, is used to describe each viseme. Speech synthesis then becomes the concatenation of calculated viseme transitions which do not realistically represent the influences of coarticulation.

CHAPTER FOUR

4. Proposed Method

The method we propose is a data-driven speech synthesis system which uses the transition from one viseme to another (or Viseme Transition Unit) as the node to concatenate. As described in chapter 3, many data-driven models generate speech synthesis through the concatenation of static visemes and morphed transitions. Unfortunately, those systems do not always take into consideration coarticulation, the way that neighbouring visemes influence each other. Our system is designed specifically to make coarticulation central to the structure of the speech animation.

The proposed method inherits advantages from other animation techniques as well, while endeavouring to avoid their disadvantages. In our method, we save data similarly to the image-based animation techniques described in chapter 3 in order to capture a high level of detail. However, we save our animation data in XML files to avoid heavy and memory-intensive image databases [46]. The proposed method inherits the efficiency of data-driven speech synthesis models, and the nuance of performance-based animation techniques without an actor or exorbitant costs.

In this chapter, we describe the process taken to develop a prototype of a data-driven speech synthesis system using Viseme Transition Units.

4.1 Data Acquisition

The proposed data acquisition method uses three high frame rate cameras to capture the three dimensional transformations of facial markers. These cameras are set

up around a seated participant who has seventeen strategically placed facial markers attached to their lips. The participant is then recorded as they read a script containing all of the selected viseme transitions.

This video data is recorded as a series of images, which are then processed using blob detection and optical flow tracking to extract the marker data. This data includes the frame as well as the relative translation of each marker based on a starting neutral template.

4.1.1 Capture System Layout

In order to acquire three dimensional data about the translation of facial markers, three high frame rate cameras were used. One camera, henceforth referred to as the front camera, was placed squarely in front of the seated participant, shooting head on. For consistency, each camera's position is described by the portion of the face that it captures. The front camera captures the vertical (y) and horizontal (x) movement of the participant's lips. The other two cameras were positioned to the left and right of the individual, on a 90° angle from front camera and the direction the participant is facing. Each camera must be level and its position carefully measured. The side cameras capture the depth (z) movement of the left and right sides of the participant's lips, respectively.

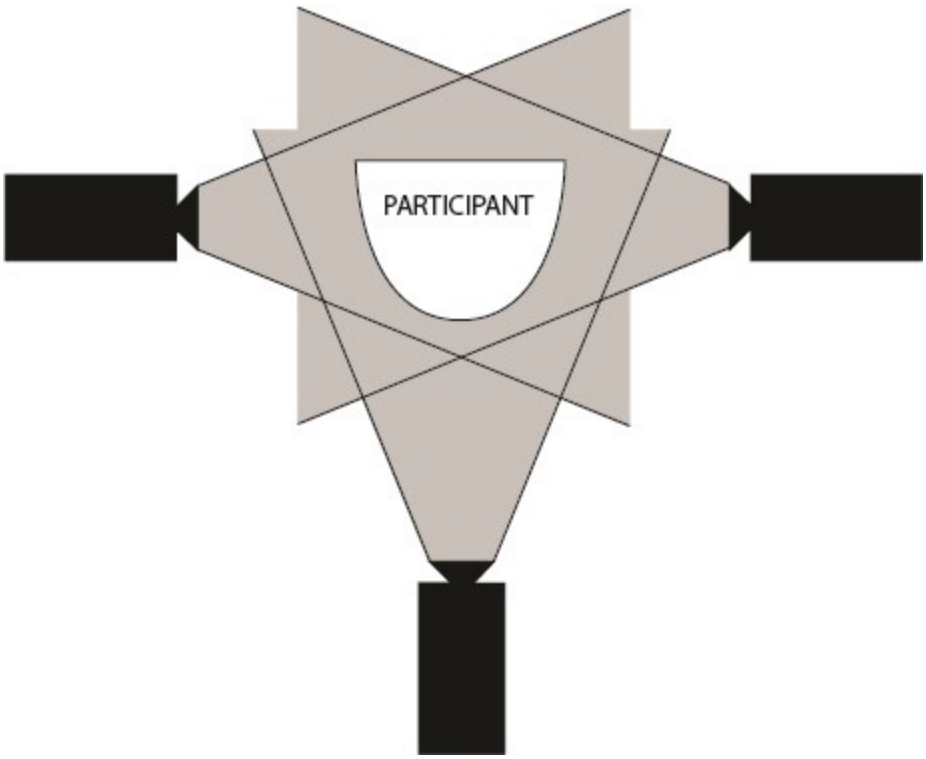


Figure 4.1 Camera Layout.

4.1.2 Camera Details and Settings

High frame rate cameras are ideal for speech data capture because of the agility with which the lips can move while speaking. It is possible that data accuracy will be ruined by motion blur when using lower shutter speed cameras, even high end webcams. High frame rate cameras support high shutter speeds, which mitigate motion blur. A prototype of this capture system was built using Logitech HD 1080p webcams, which are fairly high resolution, however the shutter speed was too slow and thus data accuracy was compromised due to motion blur. Therefore, while it is possible to use high end webcams to create an affordable system with the proposed method, for more precise data specialized high frame rate cameras are recommended.

The high frame rate cameras used for the final data acquisition method were the Grasshopper 3 (model identification GS3-U3-41C6C) by Point Grey Research [66]. The Grasshopper 3 supports resolutions up to 2048 by 2048, uses a USB 3.0 interface, and has a bus speed of S5000. These cameras support the high resolution, colour quality, and frame rate that is required to gather the relevant information.

With the exception of the frame rate, which was always set to 30fps, and the shutter speed, which was always set to 10ms, the specific settings are dependent on the light of the environment during capture. In moderate, indoor lighting situations, the settings should approximate those in table 4.1.

Camera Settings	**Value**
Brightness	12.5 %
Exposure	0.415 EV
Sharpness	1024
Hue	0 deg
Saturation	100 %
Gamma	1.2
Shutter	10 ms
Gain	0 dB

FrameRate	30 fps
W.B. (Red)	482
W.B. (Blue)	762
Temperature	318K / 44.85°C / 112.73°F

Table 4.1 Camera Settings.

The ideal lighting situation for this data capture is characterized by high saturation and even brightness. High saturation is important because colour contrast is pivotal to using hue as an optical tracking method. The algorithm must be able to pick out specific hue values. Additionally, the algorithm must be able to filter the recorded images using value data describing the brightness or darkness of each colour pixel. If there is a harsh lighting situation with drastic lights and darks, the hue data may be lost or less accurate and therefore harder to track.

4.1.3 Marker Placement

In order to accurately track the lip movement of participants, a set of seventeen markers is used. There are advantages and disadvantages to using markers in speech recording. The greatest disadvantage is the possibility that having things attached to one's lips may influence the way that one enunciates while speaking. Individuals may over-emphasize because they feel self-conscious, or they may move their lips less than normal out of concern of dislodging one of the markers. On the other hand, an

advantage to using markers is that the location tracking is more precise and more reliable than markerless tracking. For this system we are very concerned with the accuracy of results, and have therefore decided to use markers for or tracking.

To mitigate some of these disadvantages, we have made an effort to minimize the impact of the markers. They are light and quite small - blue semi-spheres with a 4mm radius - and as few as possible while still tracking all essential lip anatomy. A spherical shape was ideal for tracking, and a flat section was necessary for successful adhesion to the skin. The small, semi-spheres were then painted Cobalt Blue, a very high saturation blue of middling value (not dark, nor pale).

The placement of these markers was selected based on visible characteristics of the human mouth. The marker set placement was selected to give representation to each of the five lobes of the lips. This is done by tracking the vermilion cupid's bow apex zones, vermillion lateral zones, the oral commissures (or the corners of the mouth) as well as the place where the peristomal medial and lateral zones meet. Distinctive locations that describe the shape of an individual's lips, such as the edges of the cupid's bow, and the deformation of the vermillion border, are essential to communicating mouth movement. Viewers derive speech cues from these distinguishing locations.

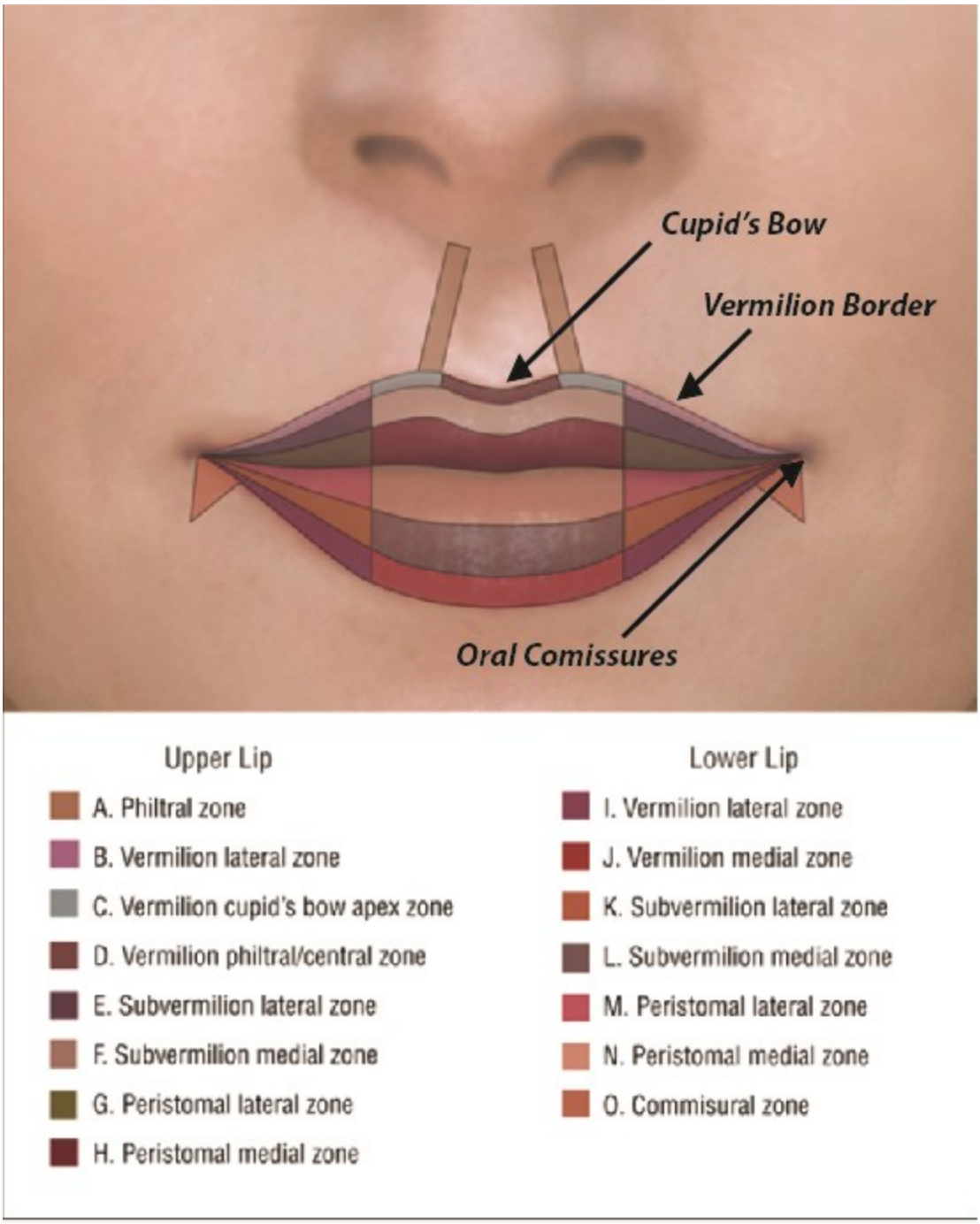


Figure 4.2 Anatomy of the lips [74].

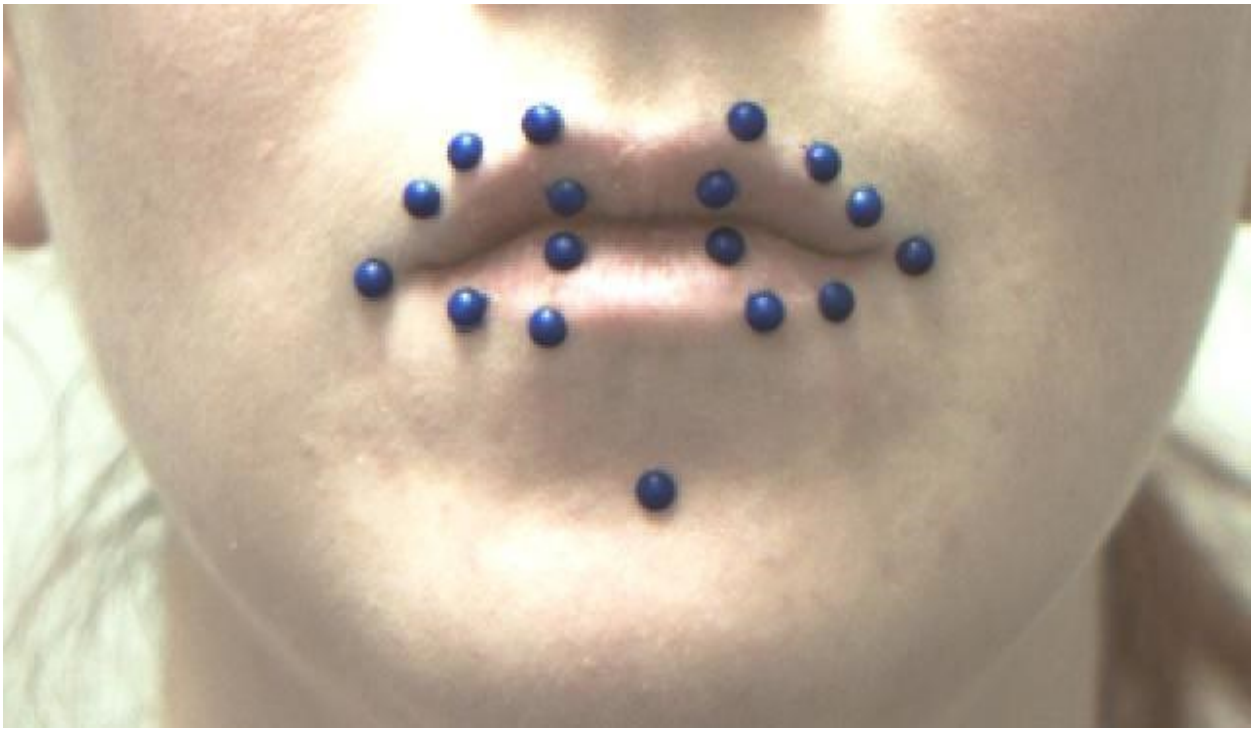


Figure 4.3 The seventeen marker system.

The decision to map only the lips was made because this system is designed to be complementary to a facial animator, purely providing emotionless speech synthesis. When skinned properly, these marker points can effectively drive a 3D mesh as an excellent speech base layer for a character animator, saving days of work. The chin marker, conversely, is used to track the jaw movement of the subject. This can be used to drive the teeth and jaw line of the 3D mesh. Only one marker is necessary because the jaw is one solid unit and its movement is uniform and mechanical. More information on skinning will be discussed later in this chapter.

4.1.4 Viseme Mapping and Capture Script

The participants, sitting amidst the cameras, with the markers adhered to their lips, were then prompted to recite a script. It can loosely be called a script, but many of the words spoken do not necessarily exist in the English dictionary. This script was designed specifically to incite the participants to utter phoneme transitions that would produce each possible combination of the selected visemes.

Many phoneme-to-viseme maps were reviewed in order to develop our own. Arguably the most accurate is a one-to-one mapping, where each phoneme is assigned its own viseme identity, however this is unnecessary due to the overlap between voiced and unvoiced phonemes. We decided to begin with Microsoft's phoneme-to-viseme map that features 22 visemes, including silence, and cull it based on a cross reference with the International Lip-reading Association's 18 visemes as well as by doing some preliminary captures [8, 13]. To determine which visemes to cull, the relevant comparison phonemes were recorded and then compared, frame by frame, for sufficient differentiation. Specifically, it was found that the viseme "ao" (the vowel from "bought") was not sufficiently different from "aa" (the vowel from "bob") to warrant being processed separately. Additionally, in central Canadian English, the consonant "h" is largely silent and takes on the viseme of whatever vowel follows it, so it was also removed from the mapping. See the final list of 20 visemes is found in table 4.2.

Viseme	**Phoneme**	**Examples**
0	silence / neutral	
1	ae ax ah	Bat / about / but

2	aa ao	Bob
3	ey eh	Bait / bet
4	er	Bird
5	y iy ih ix	You / beat / bit /
6	w uw	Won / boot
7	ow, uh	Boat / book
8	aw	Down
9	oy	Boy
10	ay	Bite
11	r	Rat
12	l	Lot
13	s z	Sit
14	sh ch jh zh	Shut / church / jump
15	th dh	Thick / that
16	f v	Fog / vat
17	d t n	Fig / top / nod
18	k g ng	Cat / got / sing
19	p b m	Pot / bet / mom

Table 4.2 Phoneme-to-Viseme map.

Scripts used in speech synthesis research are commonly sentences or paragraphs with keywords that collectively contain most visemes [61]. In our case, we created words to ensure that every possible combination of the visemes in our mapping is recorded. This system supports synthesizing speech that has the common CVC structure (C = consonant, V = vowel), as well as CVCCVC structure, meaning that

transitions can be effectively simulated between two consonants, and between consonants and vowels. Vowel-to-vowel viseme transitions were not recorded because combined vowels create their own unique visemes.

The words in the capture script are as listed in table 4.3. The participants are instructed to return to a neutral facial position after each table entry. Table entries with more than one word are spoken as a continuous sentence without returning to a neutral lip position until the end of the table entry.

Viseme	**Word**	**Viseme**	**Word**
Viseme 1: ae in “bat”	back	Viseme 2: aa as in “bob”	bock
	cab		cob
	rat		rot
	tar		tor
	chaz		choz
	sash		sosh
	wath		woth
	tha-oo		tho-oo
	fa-ee		fo-ee
	yav		yoff
	hal		hol
	lah		loh
Viseme 3: ey as in “bait”	bake	Viseme 4: er as in “bird”	birk
	kabe		kirb
	rate		rirt

	tare		tir
	chaze		chirz
	sashe		search
	wathe		wirth
	thay-oo		thir
	fay-ee		firry
	yave		yirf
	hale		hirl
	lay		lirh
Viseme 5: y as in “beat” or as the “y” consonant	beak	Viseme 6: w as in “boot” or as the consonant “w”	mook
	keeb		goom
	reet		root
	teer		toor
	cheese		choose
	seesh		zooch
	weath		wooth
	thee-oo		thoo
	yeave		fooey
	vee		you've
	heal		hool
	leah		loo
			wow why yes
Viseme 7: ow as in “boat”	boak	Viseme 8: aw as in “down”	bowk
	koab		kowb
	wrote		rowt

	tore		towr
	chose		chowz
	soshe		sowch
	wothe		wowth
	thoe-oo		thow
	foe-ee		fowy
	yove		yowv
	hoal		howl
	low		l-ow
Viseme 9: oy as in “boy”	boy	Viseme 10: ay as in “bite”	bike
	coy		kibe
	roy		rite
	toy		tire
	choy		chize
	soy		siche
	wothe		withe
	thoy		thiwe
	foy		fie-ee
	yoy		yive
	hoyl		hile
	loy		lie
Viseme 11: r as in “rate”	hall roar la	Viseme 12: l as in “lot”	has loll zeal
	has roar zeal		ash loll sha
	ash roar sha		hath loll though
	hath roar though		of loll vo

	of roar vo		dot loll dot
	dot roar dot		oak loll got
	oak roar got		um loll boo
	um roar boo		see loll yes
	see roar yes		ew loll we
	ew roar we		
Viseme 13: s/z as in “sit”	ash size sha	Viseme 14: sh / ch / jh / zh as in “shut / church / jump / measure”	hath shush though
	hath size though		of shush vo
	of size vo		dot shush dot
	dot size dot		oak shush got
	oak size got		um shush boo
	um size boo		see shush yes
	see size yes		ew shush we
	ew size we		
Viseme 15: th as in “thick”	of thath vo	Viseme 16: f / v as in “frog / vat”	dot fave dot
	dot thath dot		oak fave got
	oak thath got		um fave boo
	um thath boo		see fave yes
	see thath yes		ew fave we
	ew thath we		
Viseme 17: d / t / n as in “top / nod”	oak dot got	Viseme 18: k / g / ng as in “cat / got / sing”	um kong we
	um dot boo		see kong yes

	see dot yes		ew kong we
	ew dot we		
Viseme 19: p / b / m as in “pot / bet / mom”	see bomb yes		
	ew bomb we		

Table 4.3 Capture script for viseme transitions.

4.1.5 Exploration of Alternate Capture Methods

Alternate optical flow capture methods that were explored for this system include depth cameras and a Vicon motion capture system.

Similar hue-based optical flow tracking prototypes were developed using Microsoft Kinect and Soft Kinetic depth cameras. While the Microsoft Kinect had fairly good image colour and resolution, the Soft Kinetic depth camera could not support a high resolution depth and colour feed simultaneously and the data quality was too low for the high level of subtle detail required to analyze speech. The depth data also seems to be influenced by value. Dark objects occasionally can appear to be farther away, making depth data (particularly that of facial markers which are often a different value than skin tone) slightly inconsistent.

The motion capture system that was used as a preliminary method of data acquisition was a Vicon MX/T-Series System with eight cameras. In addition to the expense of this system, the greatest drawback for using this system was the extensive computation and data extrapolation that is performed by the motion capture software. Skipped frames are extrapolated from previous and past frames, and occasionally markers are misinterpreted as other markers due to automatic computation. Motion

capture systems are excellent means of performance driven animation. However, for the analysis of speech animation it is not a perfect solution. This is because speech is very quick and very nuanced, making a lost or inaccurate frame a critical issue. In designing a system that would allow one to proceed frame by frame to ensure that the data is processed correctly - such as the proposed data acquisition method ultimately used - one ensures the accuracy of the data is preserved.

4.2 Image Processing

The images that are produced using the three-camera capture method are then processed in order to derive the relative translation of each maker. This is done using OpenCV, an image processing library for C/C++, and a custom blob detection algorithm [65].

The images are first imported, based to the selected frames, and any lens distortion is corrected. To identify the location of each marker, the image pixels are then filtered by specific hue values that describe the correct colour of blue. The markers are then detected and sorted based on size and relative location of clumps of pixels that meet the hue value requirements. Once the pixel location of each marker is determined, the translation relative to the size of the lips and the starting neutral position is calculated and saved into an XML file. An XML file is saved for each viseme transition.

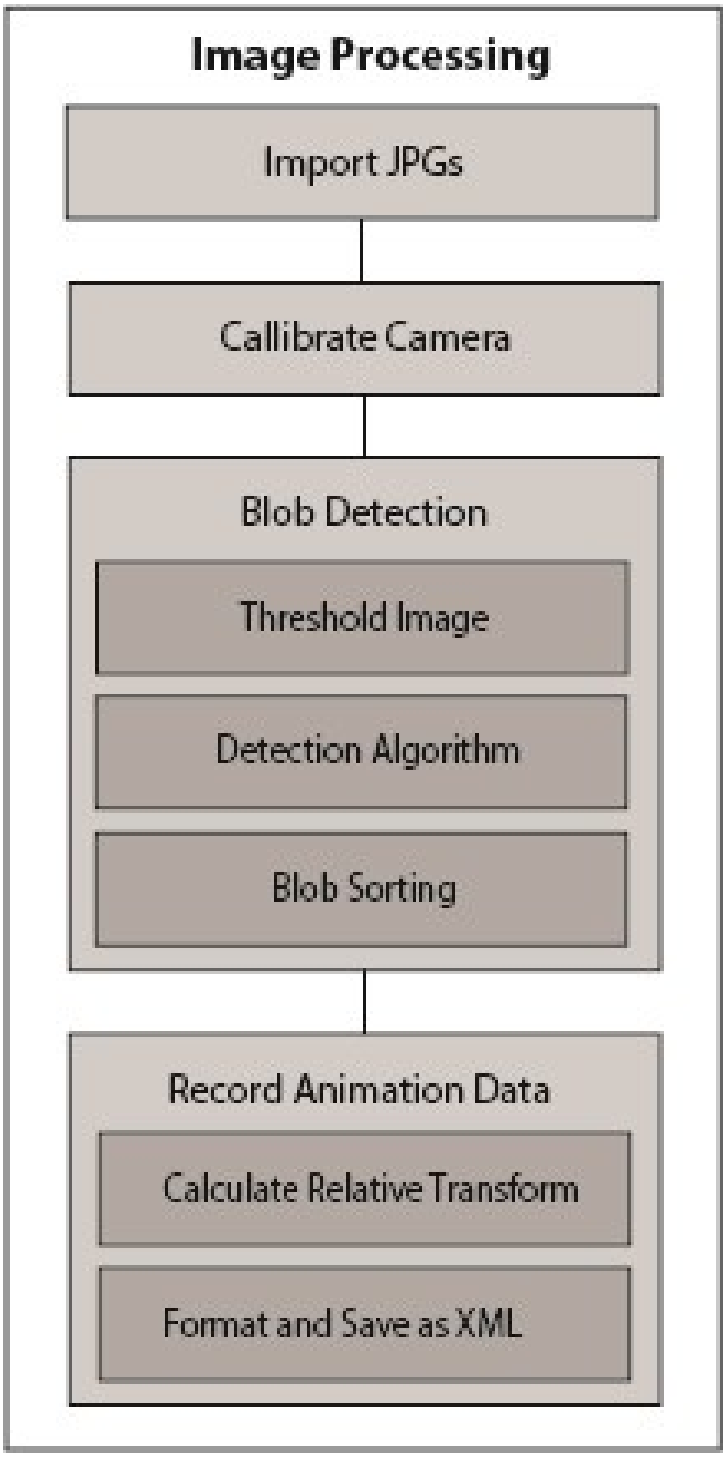


Figure 4.4 Image processing flow chart.

4.2.1 Import and Calibrate Images

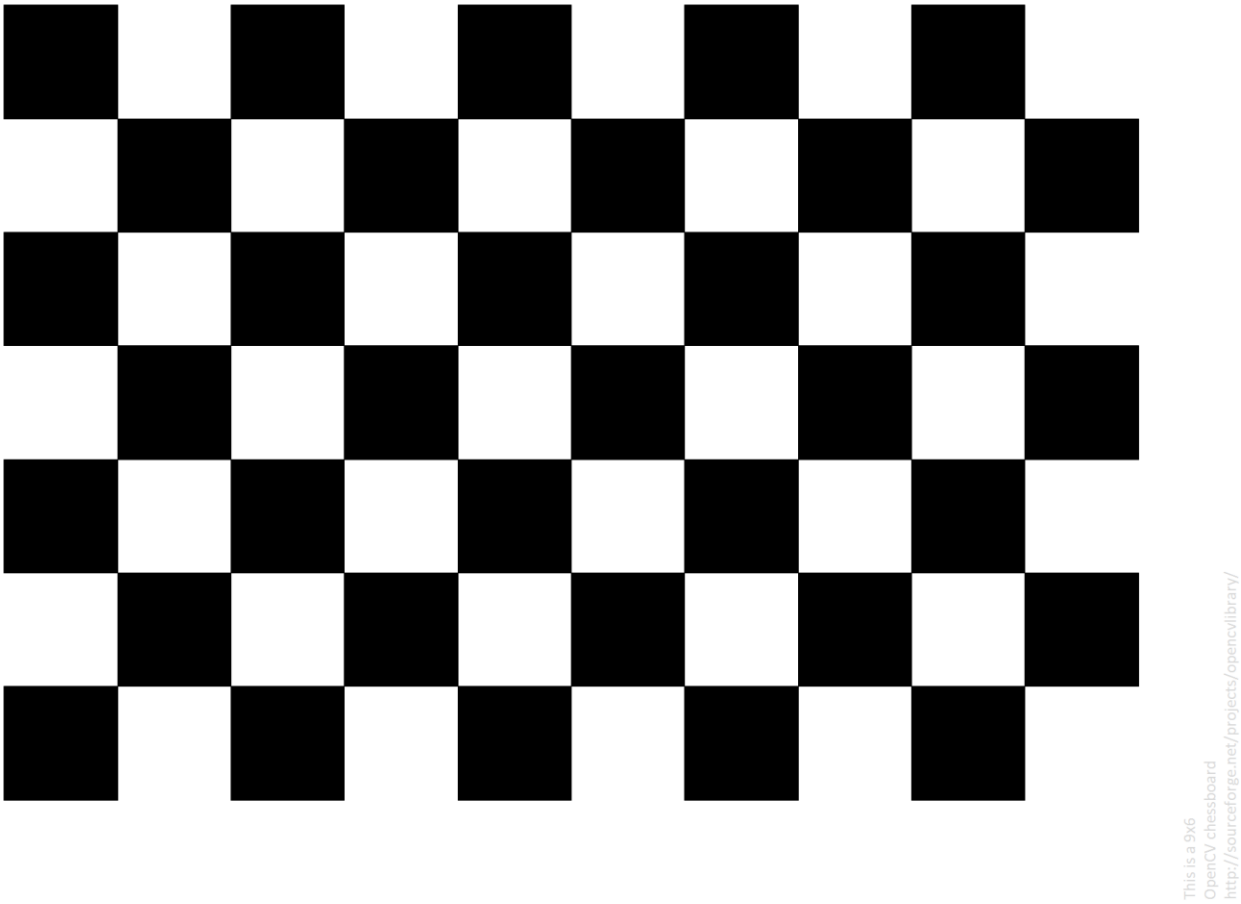


Figure 4.5 The checkerboard used in OpenCV for camera calibration [65].

The first step of the image processing software written as part of this method is to import the images within the specified range of frames. Full colour images were imported from all three cameras and assigned a direction variable (left, right, and center). The image resolution that was found to be the most efficient while upholding a high quality of accuracy was 640 x 480.

Another step taken to ensure the highest level of data accuracy was to correct for any possible lens distortion. Differently shaped lenses can alter an image captured by a camera, which could have skewed our results. To mitigate this, we used OpenCV's

camera calibration method, a streamlined application of Zhang's checkerboard method [64, 65]. This method uses images of a checkerboard, such as the one below, to calculate the camera matrix and and the distortion matrix. The camera matrix is a 3 by 3 matrix that describes the camera focal lengths and optical centers in pixel coordinates. The distortion matrix is a one row matrix with 5 columns that describes the five distortion parameters (a combination of radial and tangential distortion factors).

4.2.2 Threshold Image

In order to derive the marker locations from the imported and corrected images, we created a threshold image using OpenCV. Firstly, the images are converted from RGB (red, green, blue) space in HSC (hue, saturation, value) space. Once the images are in HSV space it is possible to filter pixels by their individual hue, saturation, and value. During this filtering process, pixels with HSV values that fall within the specified range would be white in the threshold image. Conversely, pixels with HSV values that did not fall within this range would appear black.

For this method, we were interested in thresholding based on the hue of each pixel so that we could identify the blue facial markers. For this reason, it was important to identify the correct HSV range for each set of images. We endeavoured to recreate the same lighting situation for each of the five participants, however we also built in HSV threshold controls so it is quite easy to adjust HSV limits as necessary for different sets of images. This enabled us to tweak values on a case by case basis for the best level of accuracy. The low end of the acceptable hue range was consistently 89, while the upper range varied between 155 and 170.

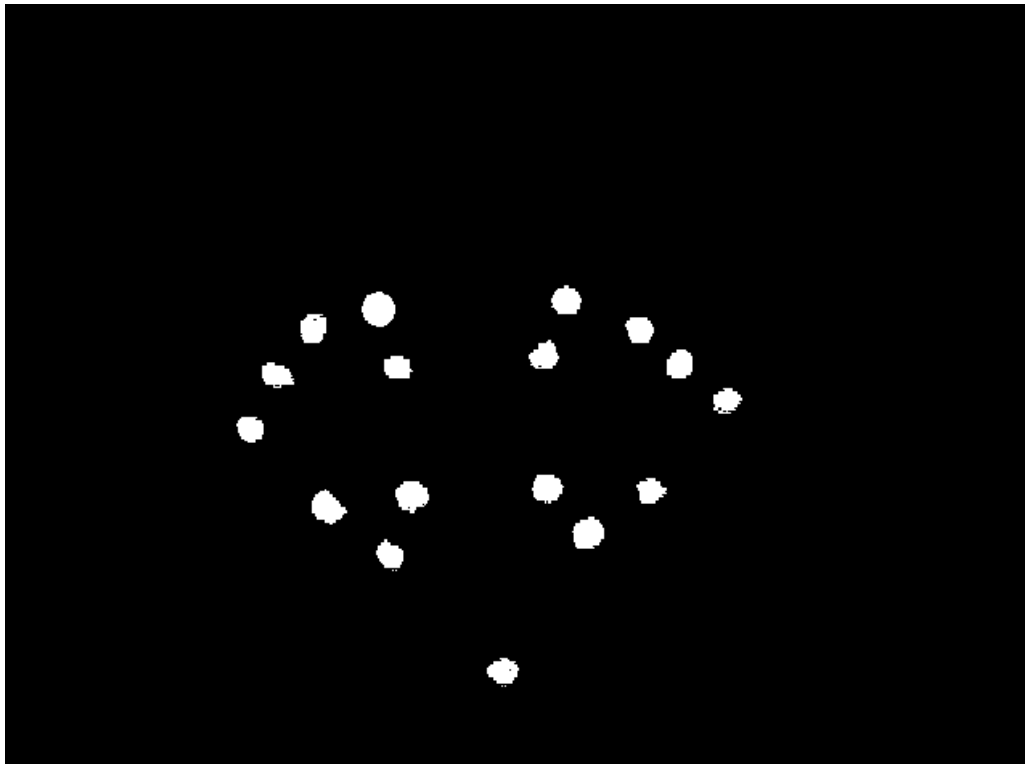


Figure 4.6 An example threshold image from the center camera.

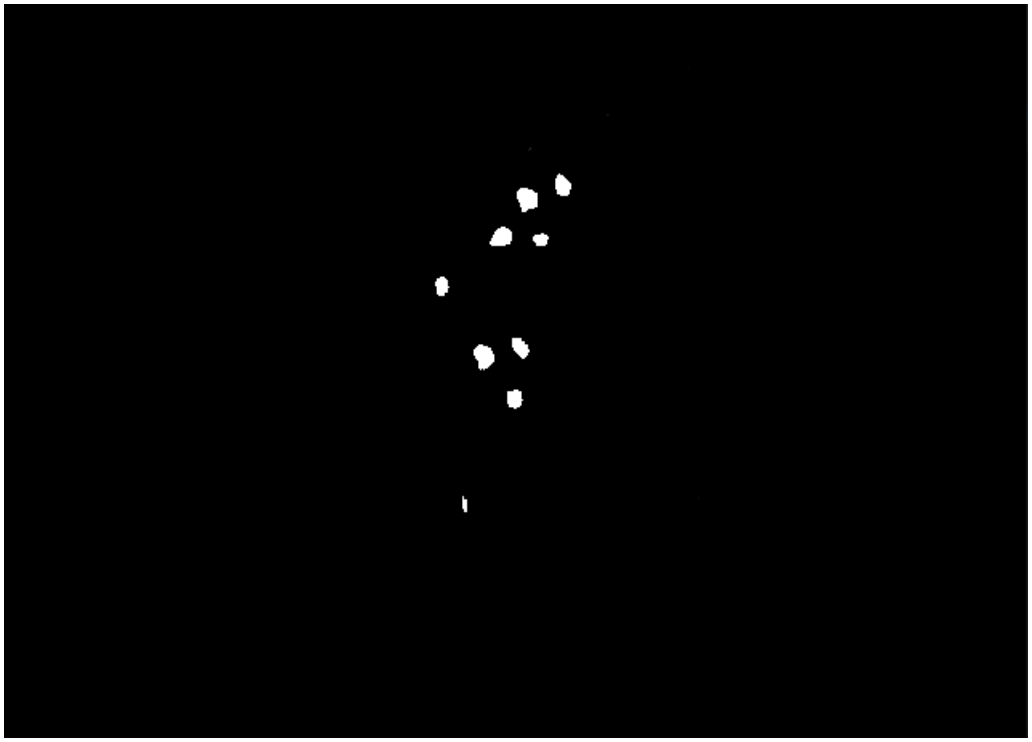


Figure 4.7 An example threshold image from the right camera.

Figure 4.8 An example threshold image from the left camera.

4.2.3 Blob Detection

The blob detection algorithm used for this method is based on classic blob detection algorithms, but modified to suit the specific needs of optical flow tracking. Originally, the OpenCV blob detection method was explored. Unfortunately, it did not yield reliable results in this situation and seldom identified the markers from the threshold images.

The first step of the blob detection algorithm is to label all potential blobs. This is done by looping through each pixel of the black and white threshold images from the top left to the bottom right. If the pixel is white, its neighbours (the pixel above and the pixel to the left) are checked. If they are also white, then the current pixel inherits the label (a

positive integer number) of its neighbours. Otherwise, a new label is assigned to that pixel. However, if both the left and above pixels are white but bear different labels, then the lowest label value is taken. If the pixel is black, meaning it did not pass the filtering process, then its label is simply 0. These labels are saved in an OpenCV matrix data type.

The labeled blobs are then individually evaluated based on a set of adjustable parameters. These parameters include the minimum blob size and the minimum distance between blobs. Blobs must be large enough to be a marker, a small cluster of pixels is not sufficient. Specifying a minimum distance between blobs minimizes cases of double detection or overlapping blobs. Blobs that meet these criteria are approved and the x and y coordinates of their center points are calculated and recorded.

In optical flow tracking, it is important that moving blobs retain the same label from frame to frame despite any movement they may make. Trajectory crossover can occur and one blob may be mistaken for another. For example, the marker at the right corner of the lips must always be identified as marker one. In order to ensure this, the array of approved blobs are sorted based to their positions relative to each other, instead of simply their positions relative to the top left corner of the image. The numbered blobs are also assigned colours in order to facilitate a visual confirmation that markers are being detected and identified correctly. For every frame that is processed, an image likes those in figures 4.9 through 4.11 are saved for review to ensure data accuracy. The assigned colours follow the colour spectrum from red to violet, followed by white and then black, where red is the first marker label and black is the last.

The blob sorting algorithm functions as follows. The lowest marker (the one on the chin) is identified by position and is always set as the last marker in the array. The blob array for the center camera is then sorted from left to right, making the left oral commissure the first marker and the right oral commissure the second to last marker. The central markers are then prioritized based to their height. The same system is implemented for sorting the right camera, however the left camera sorts from right to left instead of from left to right, making the visible oral commissure the first marker in every array.

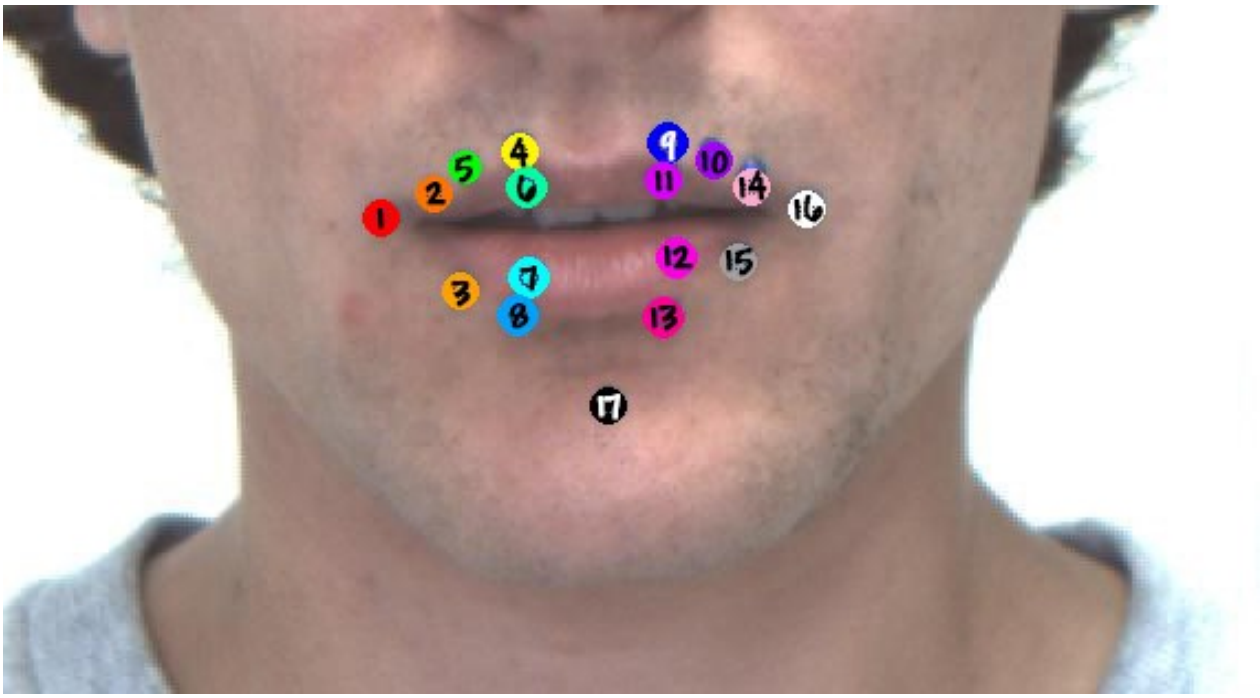


Figure 4.9 Marker identification from the center camera view.

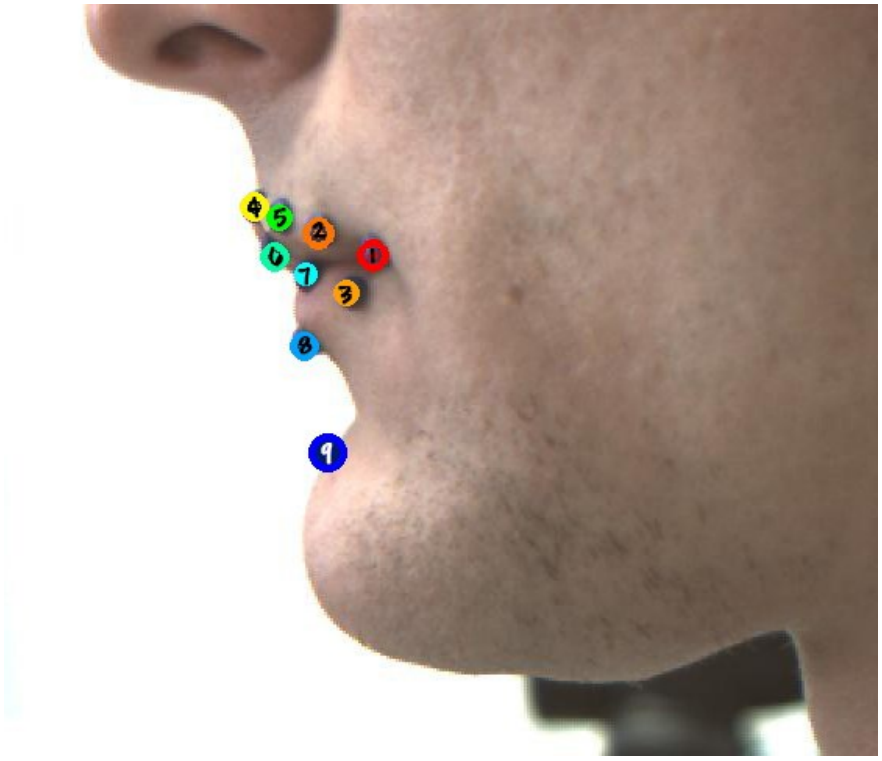


Figure 4.10 Marker identification from the left camera view.

Figure 4.11 Marker identification from the right camera view.

4.2.5 Recording Animation Data

The data that has been acquired by the process described so far is simply the x and y pixel coordinates of each marker from each image, independent of each other. The data that is required, both for analysis and to drive a 3D facial rig, must be relative to the scale of the mouth and contain three-dimensional data for each marker. Whichever models that this data may someday be applied to will not consistently be to the same scale, nor will the neutral position be precisely the same. For this reason, it is important that the recorded animation data must be relative to the scale of the mouth and to a neutral facial position, not an absolute value.

In order to provide this data, we first solve for the relative translation of each marker for each axis. This is accomplished by taking the first frame of each sequence (center, left, and right), processing it as described above, and saving it as a template. This first frame is not recorded as part of the image sequence output, merely used as a reference point, and should be of a neutral facial position. This neutral reference location will be used as the origin (x = 0, y = 0, z = 0) position for each individual marker, from which relative position will be derived by comparing the positional difference in subsequent frames. This template also provides scaling information such as the width or depth of the lips, depending to the direction of capture, and the height of the lips.

The equations for acquiring the relative translation are:

```
relativePositionX = distanceFromNeutralX / widthOfLips;
relativePositionY = distanceFromNeutralY / heightOfLips;
```

```
relativePositionZ = distanceFromNeutralZ / depthOfLips;
```

It is important to note that the depth of the lips is measured and applied separately for the left and right cameras to ensure the depth data captured is in the correct scale. Another important consideration is the direction of the x, y and z axis used in OpenCV image processing versus within 3D animation software such as Autodesk Maya or 3DS Max. Because Autodesk Maya is the software used in the prototype of this method, it is Maya's right-handed Cartesian coordinate system that must be considered. In OpenCV, the greatest y is at the bottom of the image, while in a right-handed Cartesian coordinate system the greatest y is considered to be that which is highest in the digital scene. Therefore, it was necessary to reverse the y value of each marker whilst exporting it to XML for Autodesk Maya.

For each viseme transition recorded, an XML file was created [67]. These files contain a list of all of the frames for the length of the animation. Each frame tag has an identification number and the relative translation data calculated for each marker. The x and y translations of each marker were derived from the center, or front, camera. The z translations of markers one through nine in figure 4.9 were determined using the x translations of the markers from the left camera. Similarly, the z translations of markers ten through sixteen were derived from the x translations of markers from the right camera.

4.3 Driving the Animation

4.3.1 The Facial Rig

The gathered and processed data was then used to drive a high-polygon mesh of a human head in Autodesk Maya. This mesh, purchased from Turbo Squid, was modeled for facial animation and contains 10630 polygons.

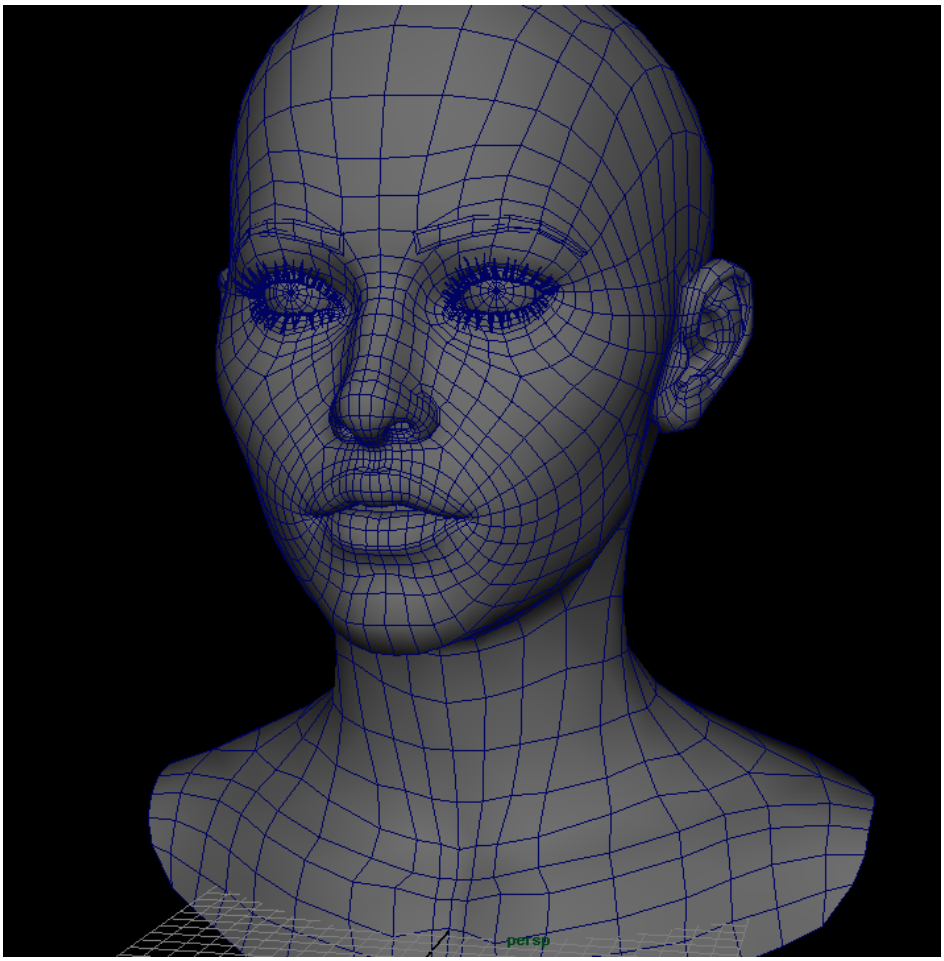


Figure 4.12 The high-polygon facial mesh.

A rig is used to animate a 3D mesh. Rigs are generally composed of joints, which make up the skeleton, and controls, which are used to animate joints. Controls can be bound to joints in different ways. This binding is called a constraint. The types of constraints used in the facial mesh for this prototype include parent constraints and aim constraints. Parent constraints relate the position, scale and rotation of the controller to the constrained joint. In this case, we were specifically interested in the position. Most of the controls in figure 4.13 drive the lip joints directly with a parent constraint. The aim constraint is used to control the rotation of an additional joint that is at the back of the model's moth. This rotation controls the lower mouth and teeth of the mesh. An aim constraint functions such that the joint will always be rotated to face the control.

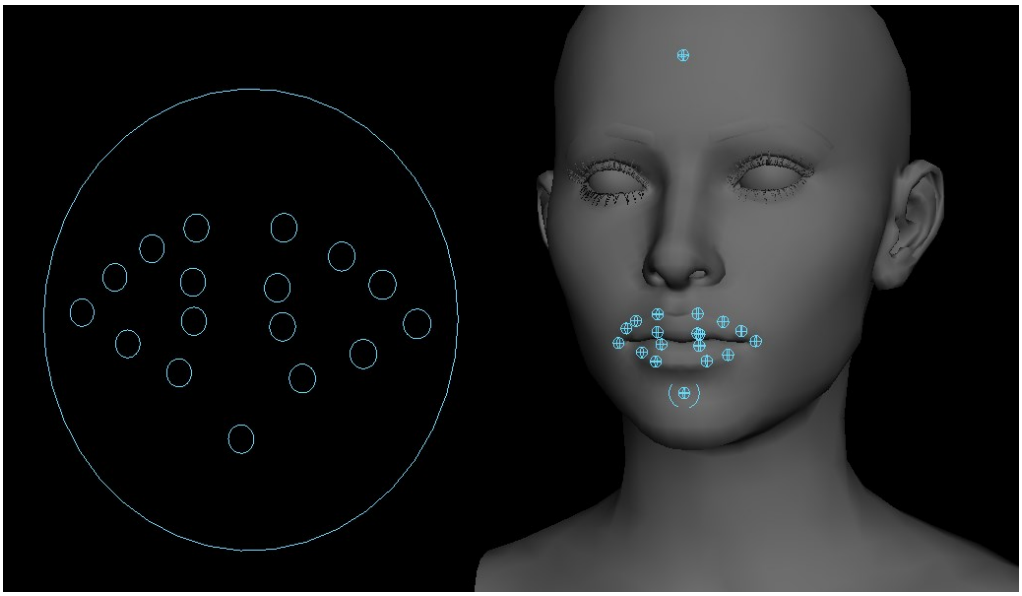


Figure 4.13 The facial rig used for the prototype; the controls are the 2D circles to the left and the bones are the spheres on the lips.

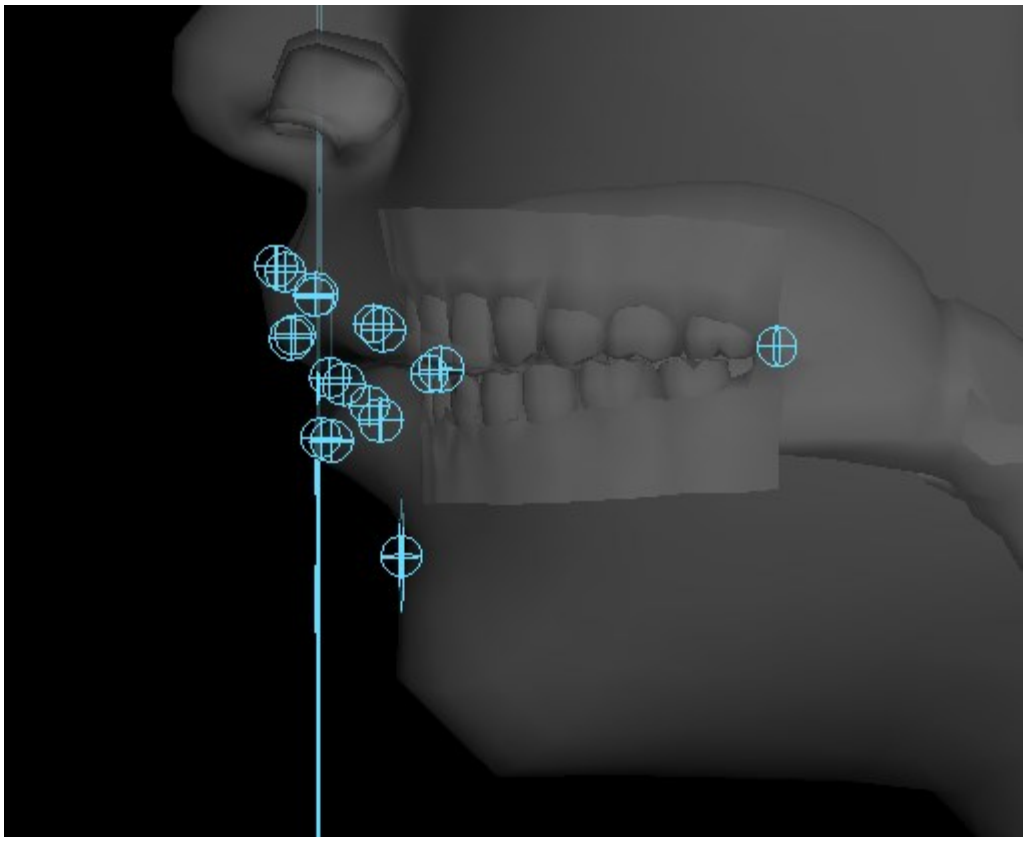


Figure 4.14 A side view of the facial rig used for the prototype, with a semi-transparent material.

With the exception of the joint to the forehead (which is meant to animate the rest of the head and stabilize the animation) and the joint in the back of the mouth (which is used to animate the teeth in accordance with the movement of the jaw), all of the joints are positioned to mirror the markers used in the data capture described earlier in this chapter.

The controls drive the skeleton; the skeleton in turn must drive the mesh. Attaching the polygonal mesh to the skeleton (joints) is a process called skinning. During the process of skinning, a weight is assigned to each vertex in the mesh. This

weight designates how much that part of the mesh follows the transformations of each bone. Weights can be manually adjusted in Autodesk Maya through a manual process called weight painting. For this prototype, the weights were set as accurately as possible using the captured image frames as reference.

4.3.2 The Python Script

In our prototype, a python script is run in Autodesk Maya to procedurally generate speech animation based on user input. The user is able to input the starting frame of the animation, as well as the desired string of visemes. This script accesses the XML files generated in the data acquisition stage, translates it into the correct space and scale for the receiving set of speech controllers, and creates the corresponding key frames. A key frame is a point on the timeline that describes the translation of an object at that time.

On a high level, the script functions as follows. Using the Document Object Model API, the script loops through the XML files that are associated with each transition in the string of visemes provided by the user. The current frame in the timeline is selected based to the frame identification in the animation data, the starting frame, and a modifier that tracks how many frames have been covered in past visemes in the same string. At each frame in the timeline, a key frame is set for each control in the facial animation rig. The transformation values of each control at each frame are derived using the following equations:

```
newPositionX = defaultCtrlPosX + (relativePositionX * widthOfLips);
```

```
newPositionY = defaultCtrlPosY + (relativePositionY * heightOfLips);
newPositionZ = defaultCtrlPosZ + (relativePositionZ * depthOfLips);
```

This process dynamically scales the animation data to the size of the lips used in any application, as long as the relative positioning of the markers to each other within the template is consistent. These equations also take into consideration any existing transformation that may have been applied to the controllers, but this can be toggled. In a good rig, all controllers should be at a zero transformation when they are in their neutral starting pose.

CHAPTER FIVE

5. Results and Evaluation

In chapter 4, we described our approach for speech animation using viseme transitions as animation nodes. In this chapter we will discuss the prototype developed using this approach, any challenges we have encountered, and evaluate the resulting data. Our results will be evaluated qualitatively using average Euler distances to compare the animation curves of a directly recorded phrase (which would therefore have accurate-to-life coarticulation) and the same phrase using our method. This comparison is modeled after standard qualitative lip synchronization analysis which endeavour to match audio with video [80]. Instead, we aim to match video with video. This data will tell us numerically how much of a variation exists between these two animations, and therefore how accurately our system produces realistic coarticulation in speech animation synthesis.

5.1 Data Acquisition Study

Many cultural factors influence the way that an individual forms words. For this reason, we have chosen to seek participants for our study with similar cultural and geological backgrounds as well as level of education in order to minimize any major discrepancies in data that may be caused due to differences in speech that can be produced by these influences. For example, cultural and geological backgrounds influence bone structure and accent of the speaker, and level of education can also influence elocution and emphasis. Therefore, five Caucasian individuals, who were

raised in urban Ontario, Canada, and had at least three years of post-secondary education, were selected. For the purpose of this study, we describe the dialect described by this cultural background as urban central Canadian English. Three participants were male and two participants were female.

Each participant gave up approximately an hour of their time. This study was approved by Carleton Ethics, and each participant signed a waiver and was educated about the study previous to being instructed to say the words in the script described in table 4.3. In order to minimize mispronunciations of the (sometimes unusual) words created for this process, participants were asked to practice the word until it was pronounced correctly, and then recorded.

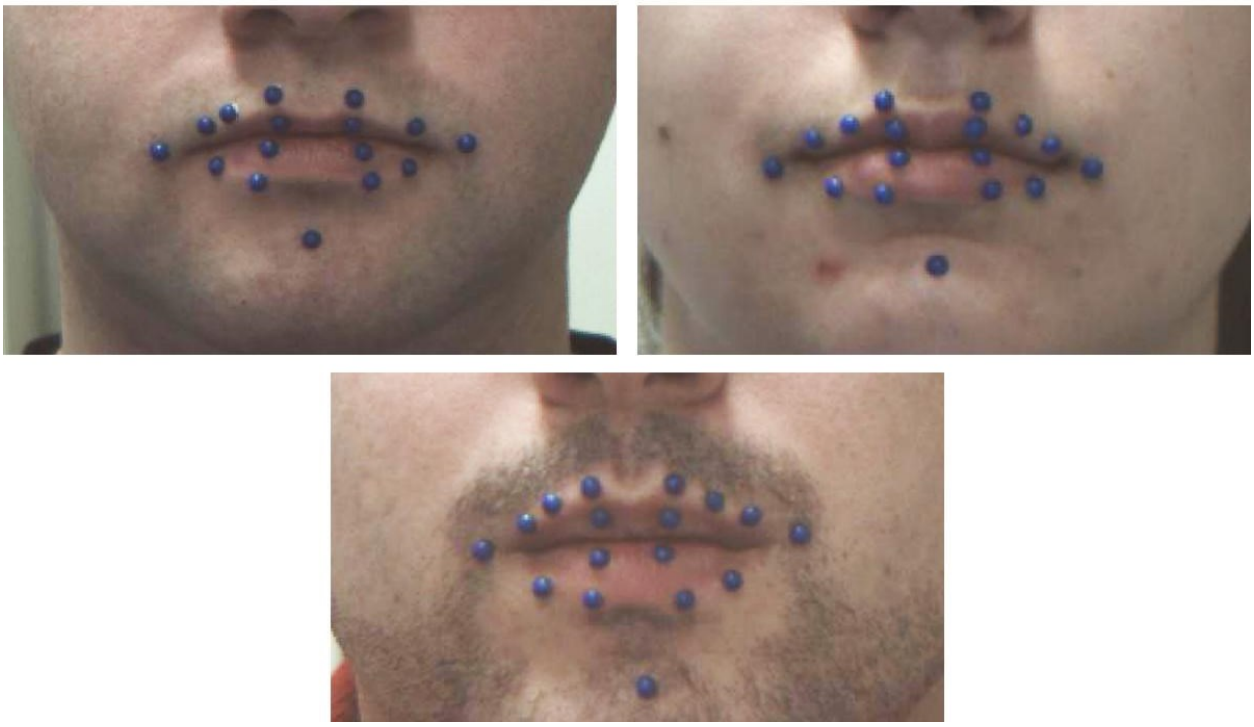


Figure 5.1 Neutral frames from each study participant.

5.2 Results

Each viseme transition node was imported and tested individually in Autodesk Maya using the methods described in chapter 4. The animation graphs of selected transitions are illustrated in figures 5.2 through 5.19. These animation graphs illustrate the change in the position of each controller (or marker) over time. In these graphs, red represents the x translation (movement of a controller from left to right), blue represents the z translation (movement forward and back), and green represents the y translation (movement up and down). The positional data illustrated is the distance of the controller from its initial neutral position. When the value of a point is zero, this means that the lips are in a neutral position. The unit of these animation graphs is a standard Maya Unit,

however the unit itself is not relevant to the results due to the fact that the positional values of each controller are scaled based on the size of the digital lips to which the animation is being applied. These nodes were selected as a demonstration because they are specifically relevant to our evaluation in the following section.

The animation data in the graphs appears quite smooth, and visually the animation accurately reflected the speech movements of participants. For example, as the distance between the lowest and highest green curves becomes greater, this represents that the mouth is being opened. Pursing of the lips is described by the narrowing of the distance between the upper and lower red lines on the graph over time.

With only minor refinement to ensure the transitions between nodes were consistent, one does not always form the same vowel the exact same way even in the same word, it became quite simple to produce words and phrases using the recorded Viseme Transition Units.

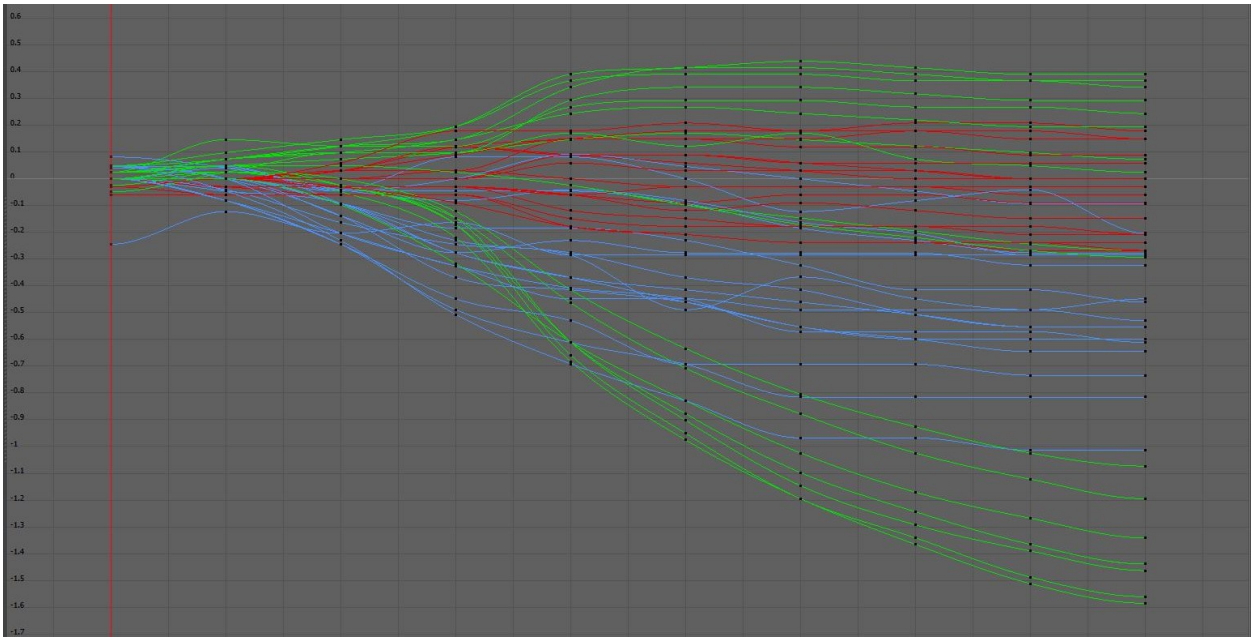


Figure 5.2 Animation graph representing the transition between the visemes 0 and 1.

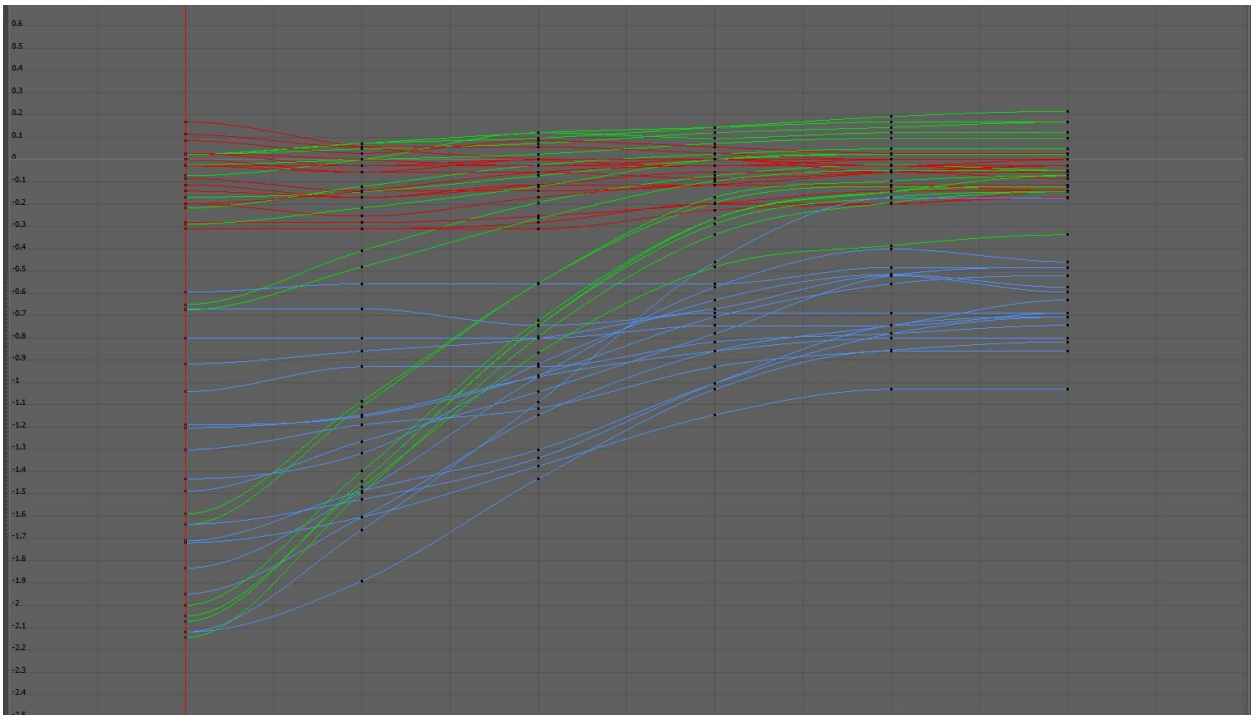


Figure 5.3 Animation graph representing the transition between the visemes 1 and 18.

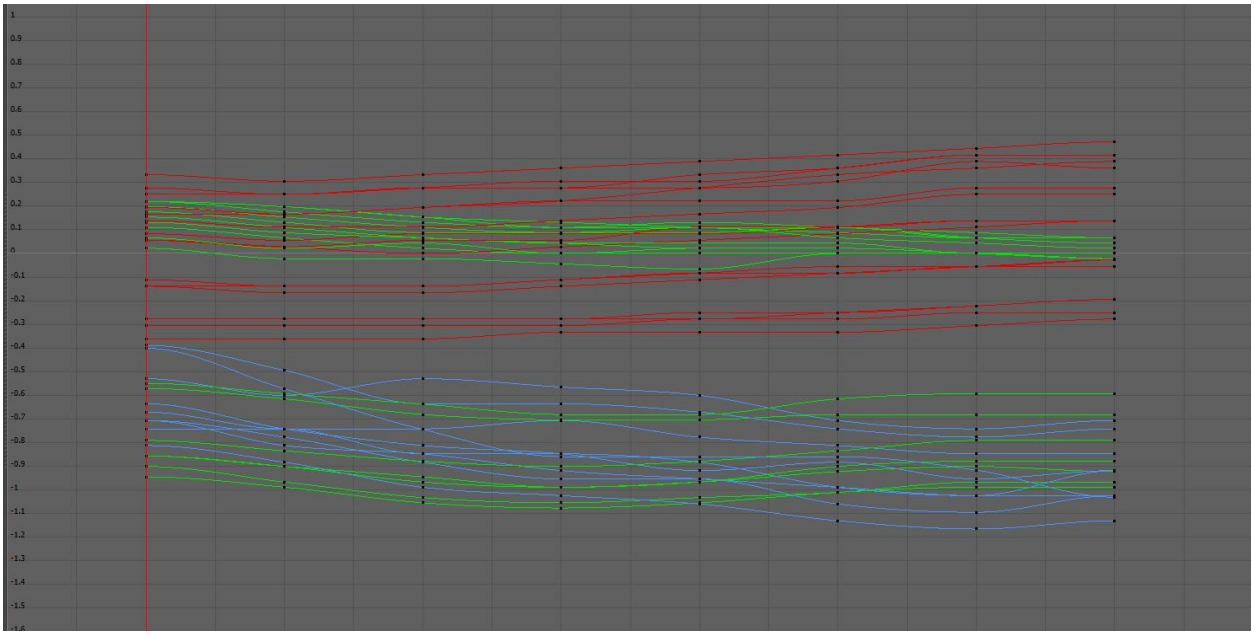


Figure 5.4 Animation graph representing the transition between the visemes 18 and 5.

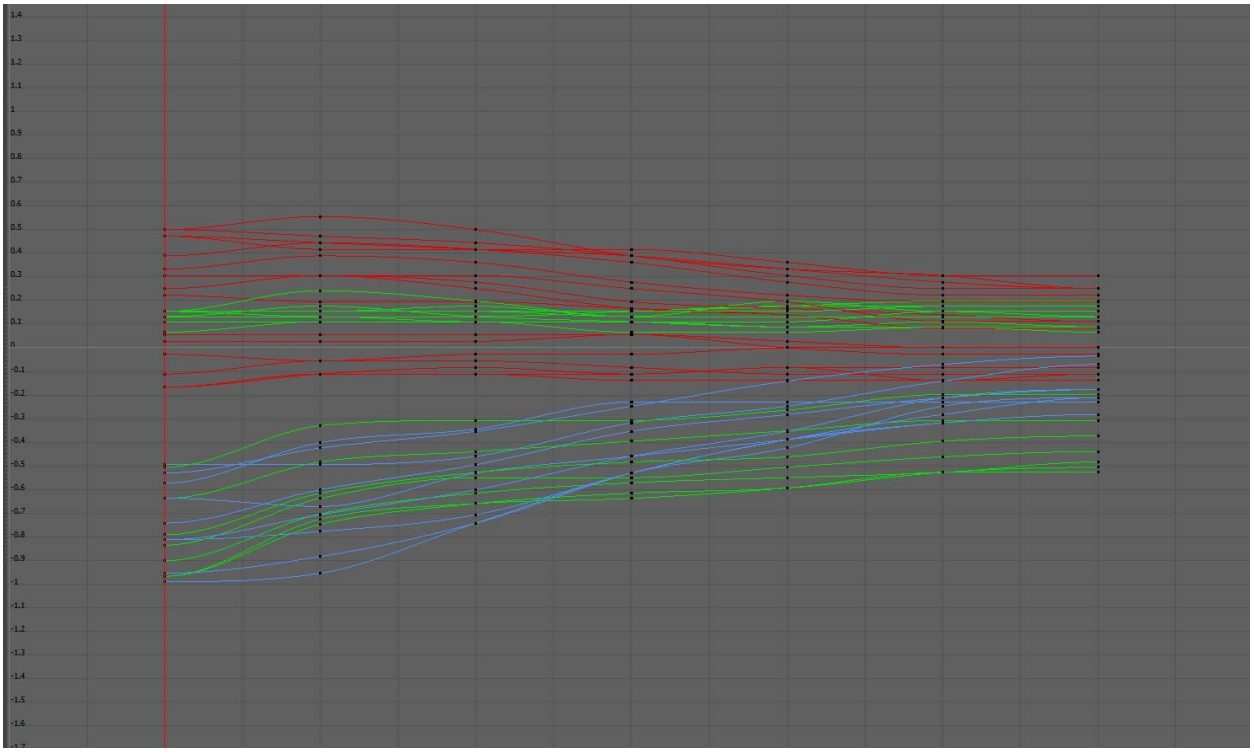


Figure 5.5 Animation graph representing the transition between the visemes 5 and 18.

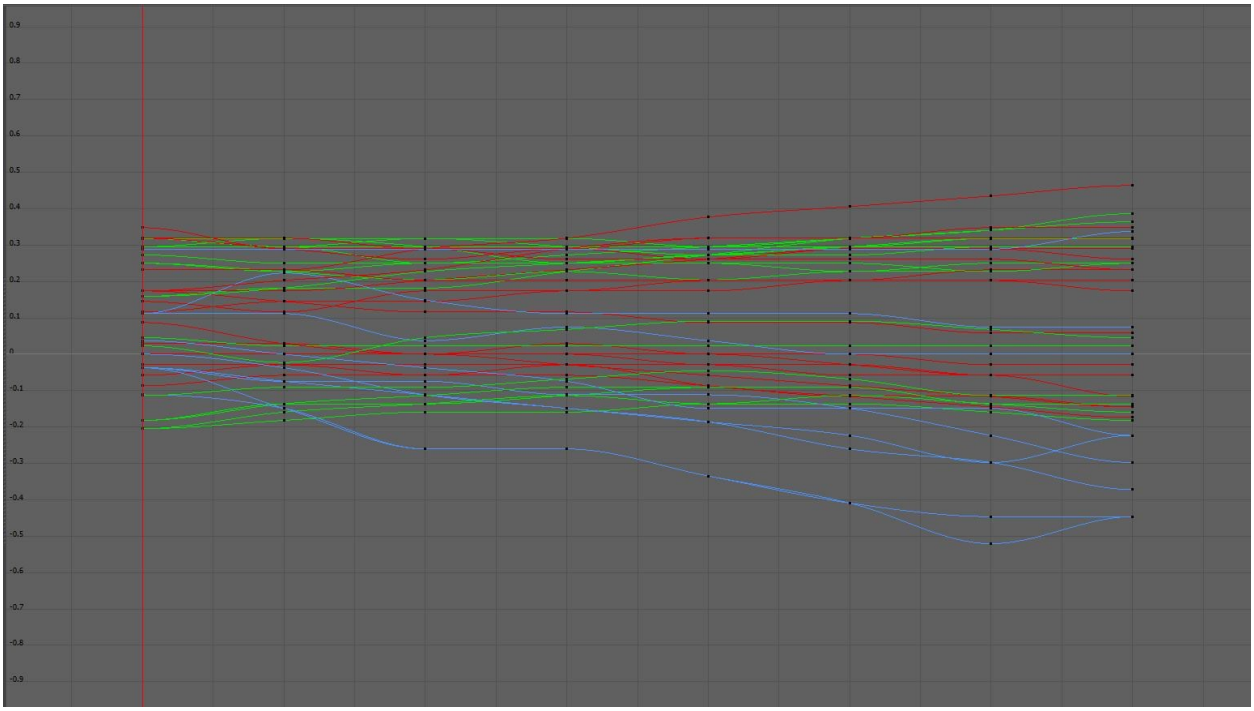


Figure 5.6 Animation graph representing the transition between the visemes 18 and 14.

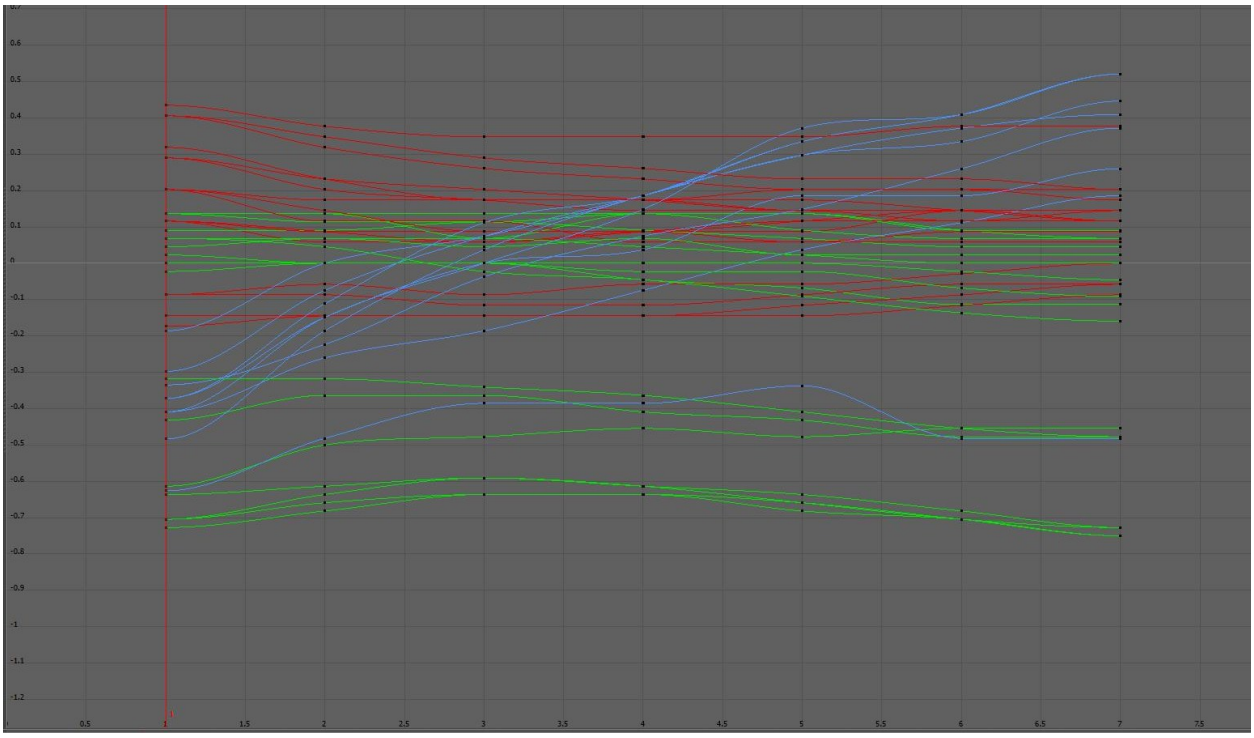


Figure 5.7 Animation graph representing the transition between the visemes 14 and 18.

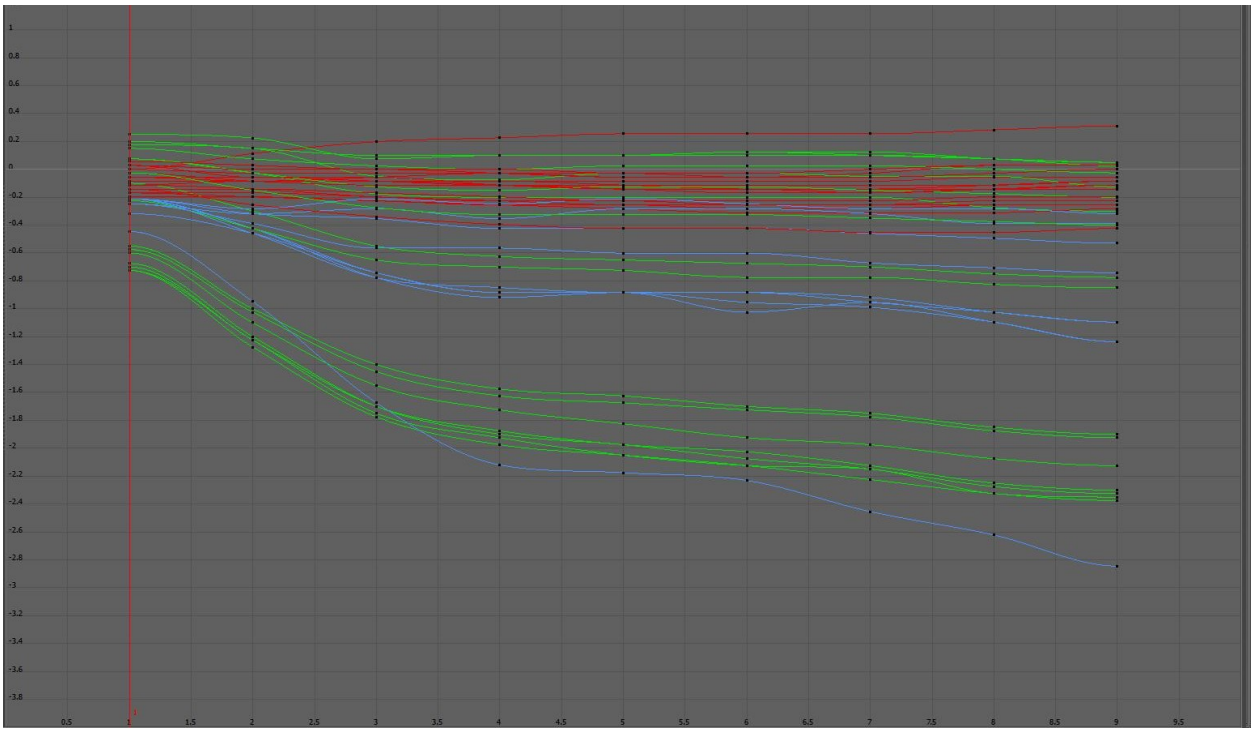


Figure 5.8 Animation graph representing the transition between the visemes 18 and 1.

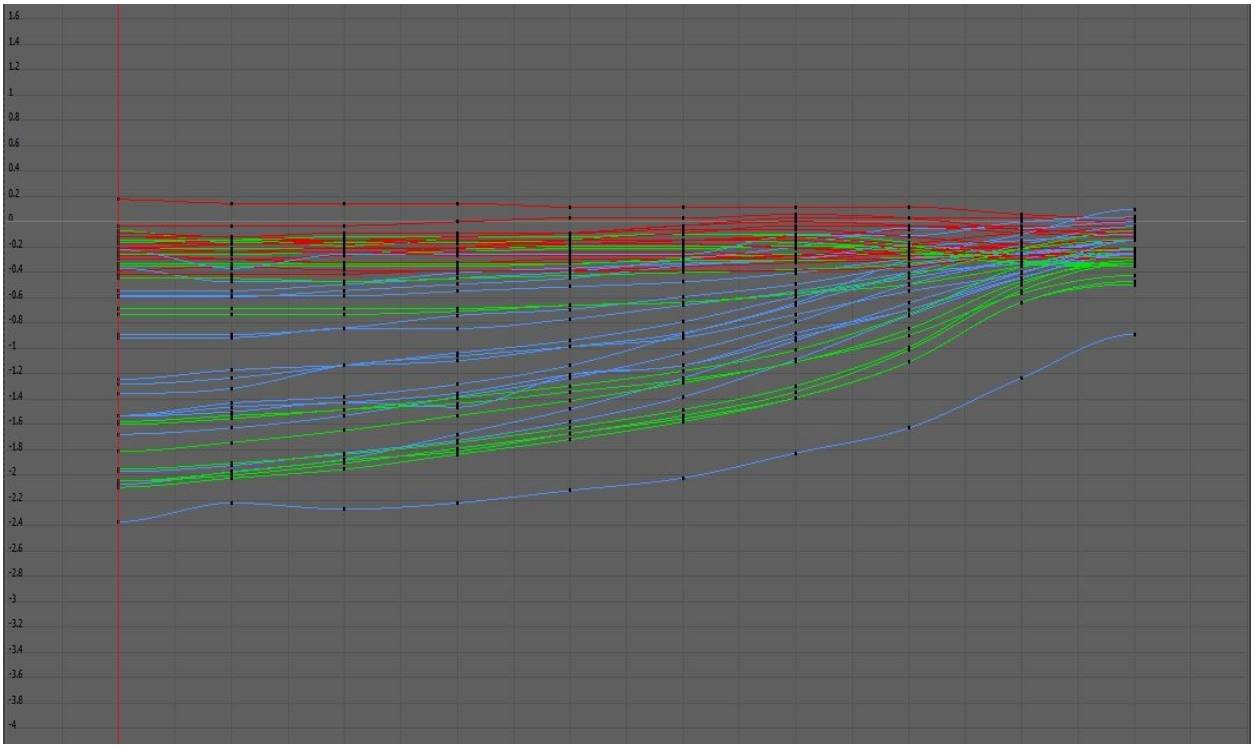


Figure 5.9 Animation graph representing the transition between the visemes 1 and 20.

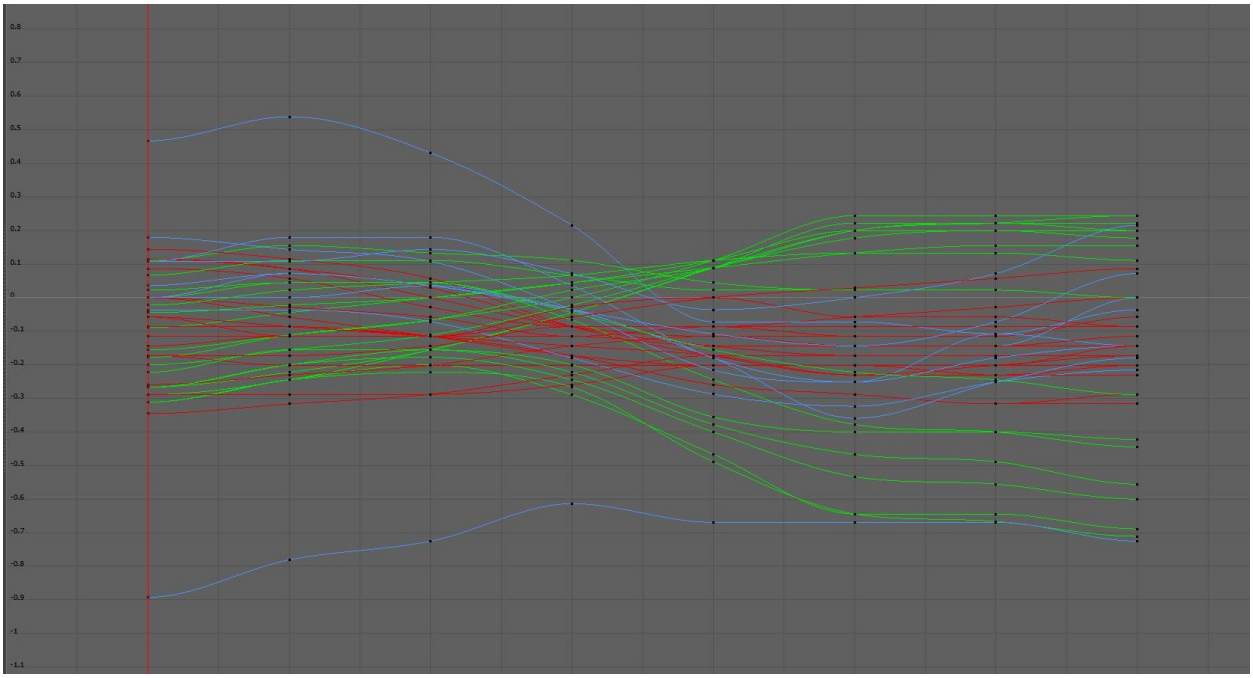


Figure 5.10 Animation graph representing the transition between visemes 20 and 13.

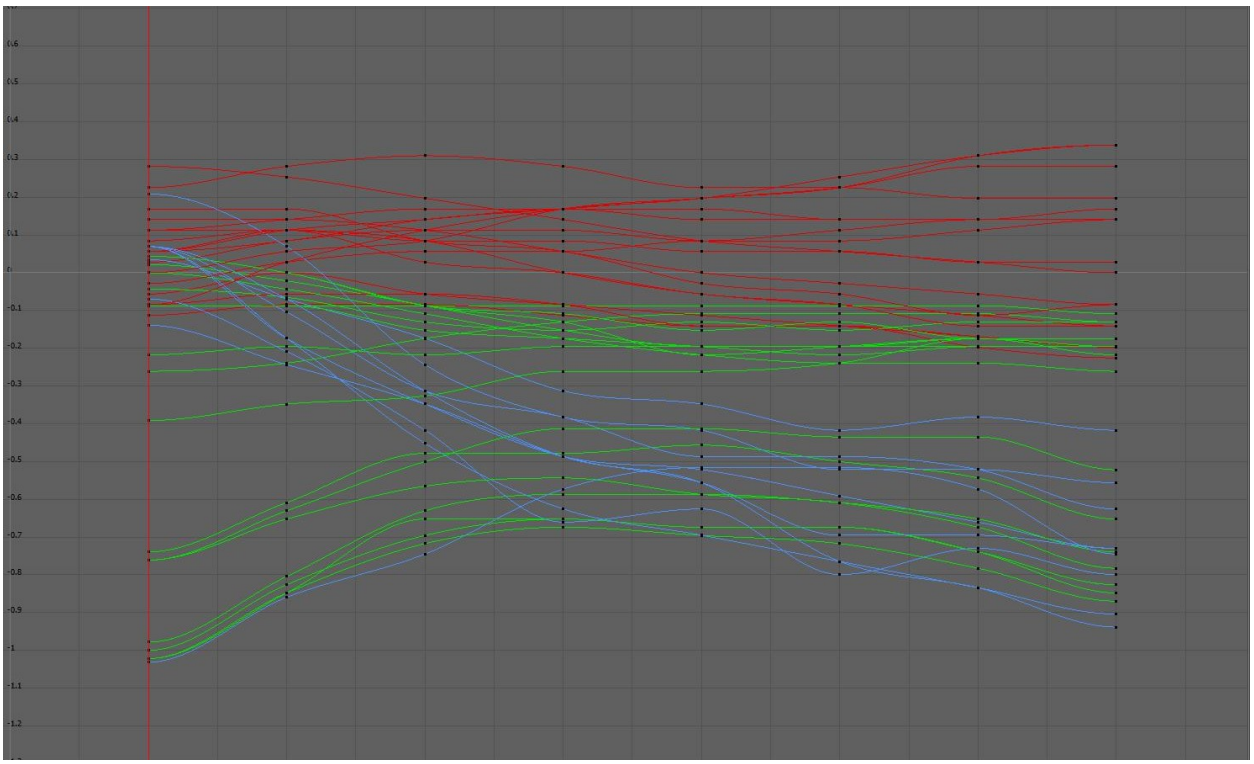


Figure 5.11 Animation graph representing the transition between visemes 13 and 5.

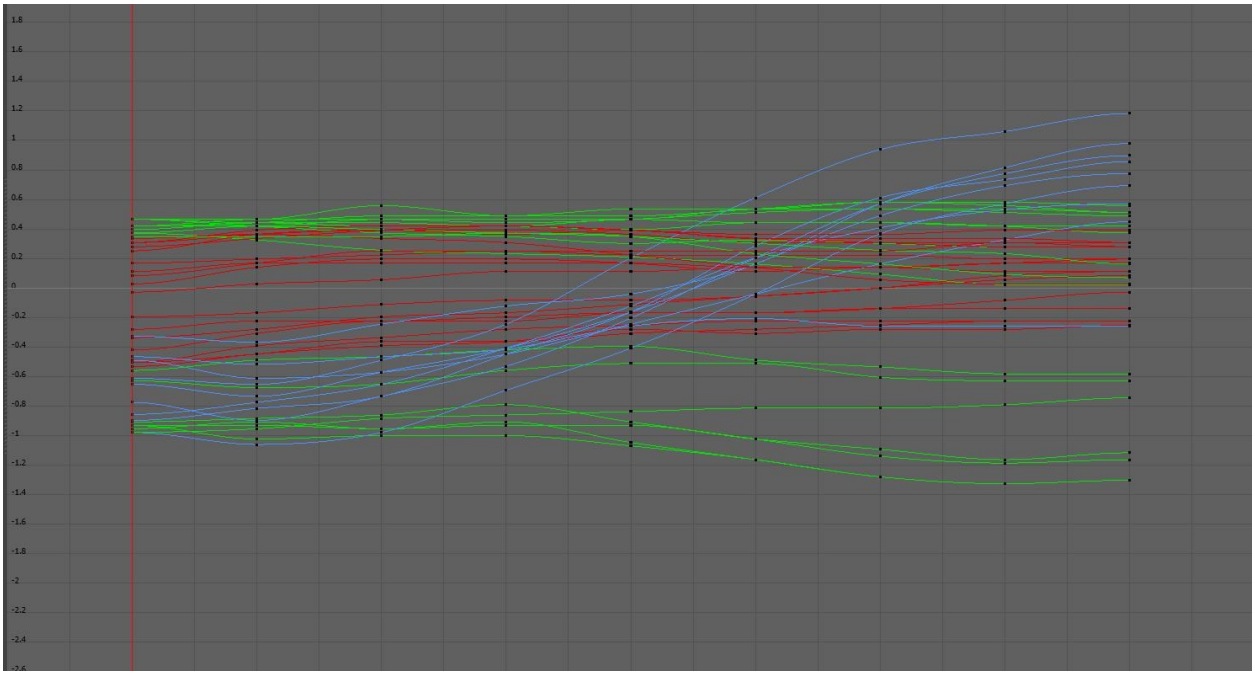


Figure 5.12 Animation graph representing the transition between the visemes 5 and 15.

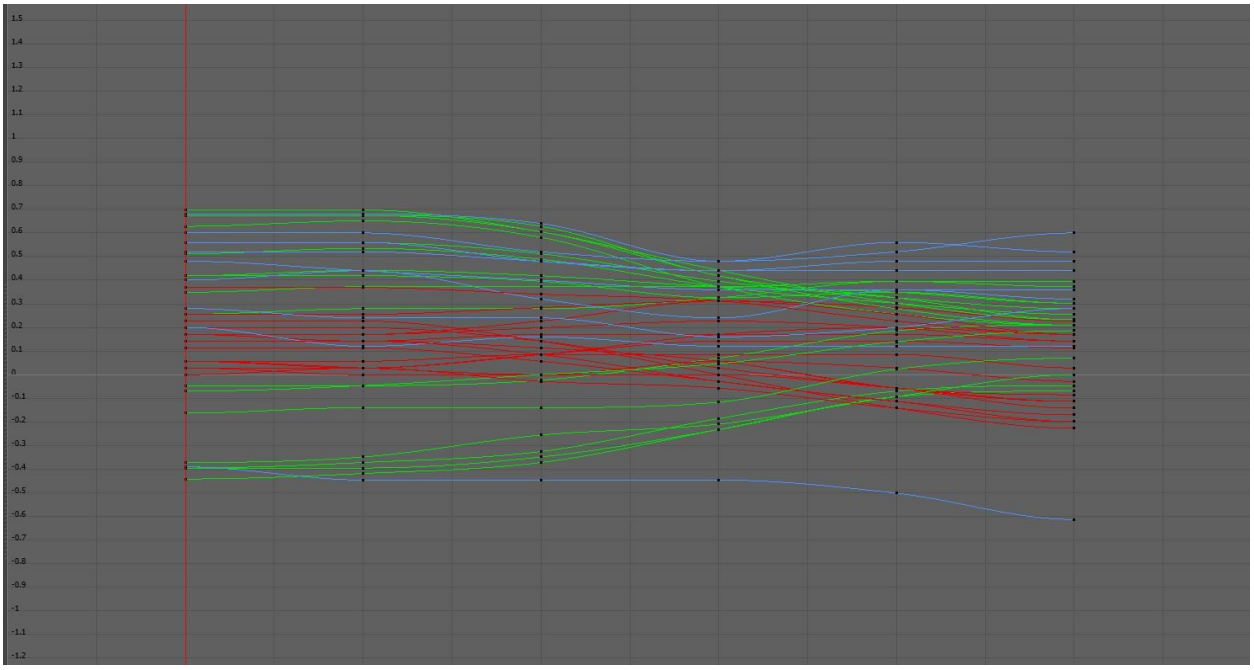


Figure 5.13 Animation graph representing the transition between visemes 15 and 20.

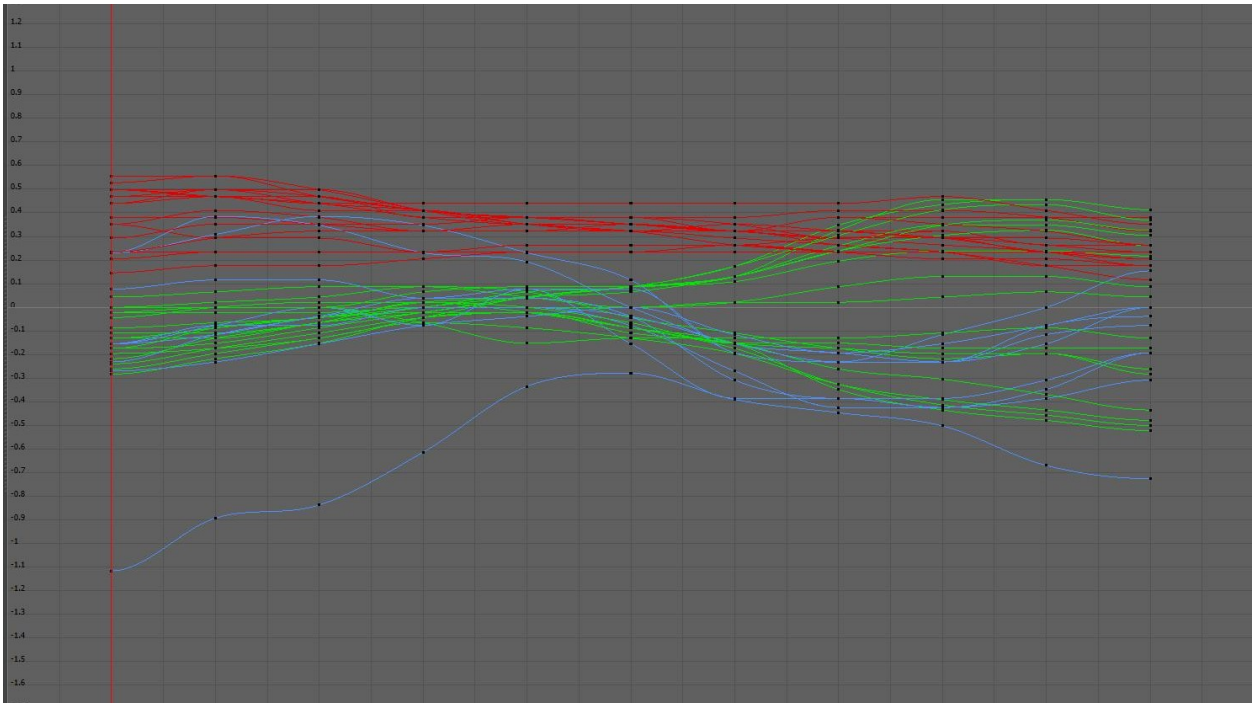


Figure 5.14 Animation graph representing the transition between visemes 20 and 18.

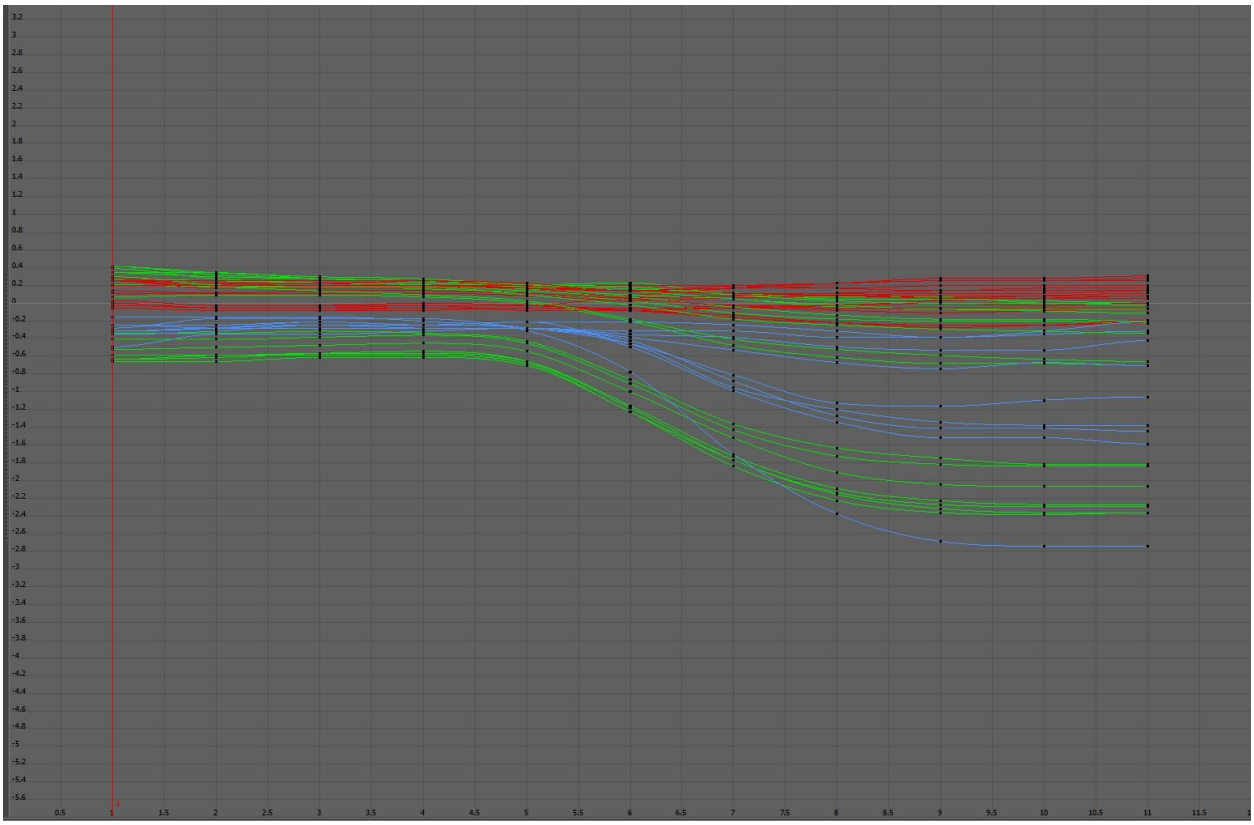


Figure 5.15 Animation graph representing the transition between the visemes 18 and 3.

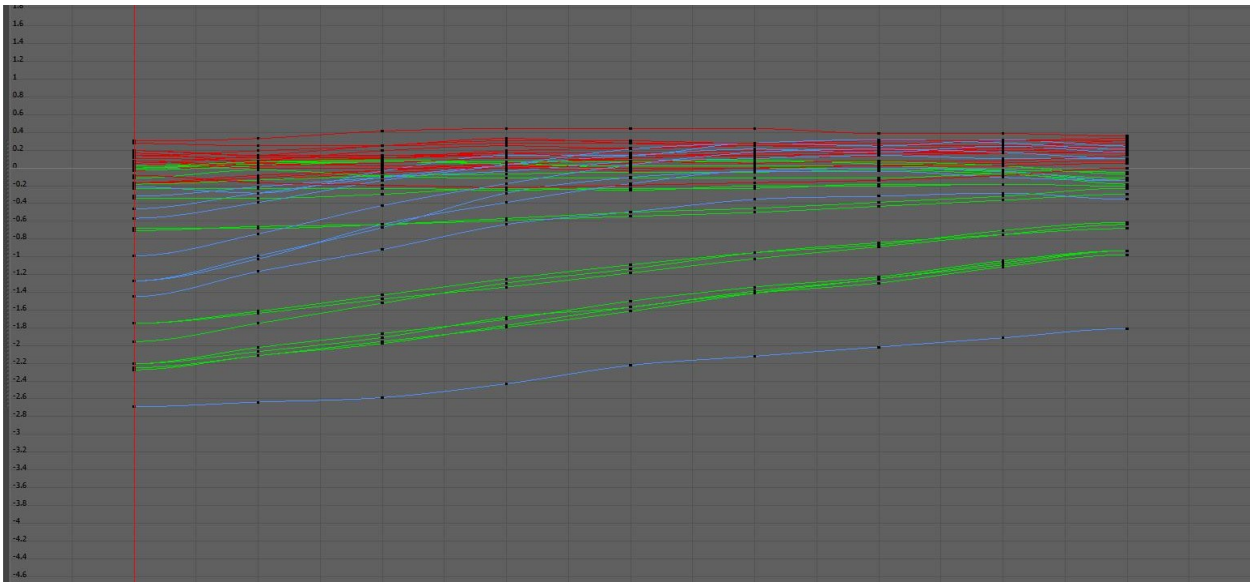


Figure 5.16 Animation graph representing the transition between the visemes 3 and 12.

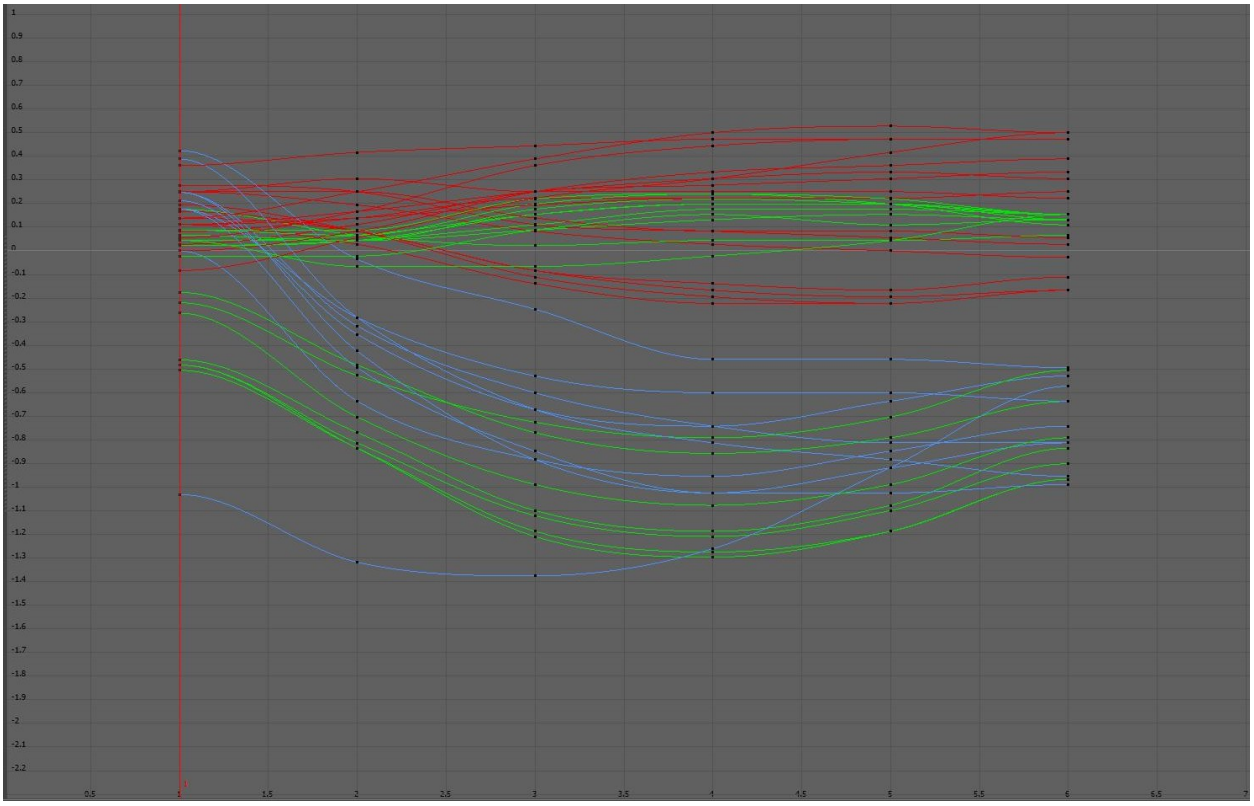


Figure 5.17 Animation graph representing the transition between the visemes 12 and 5.

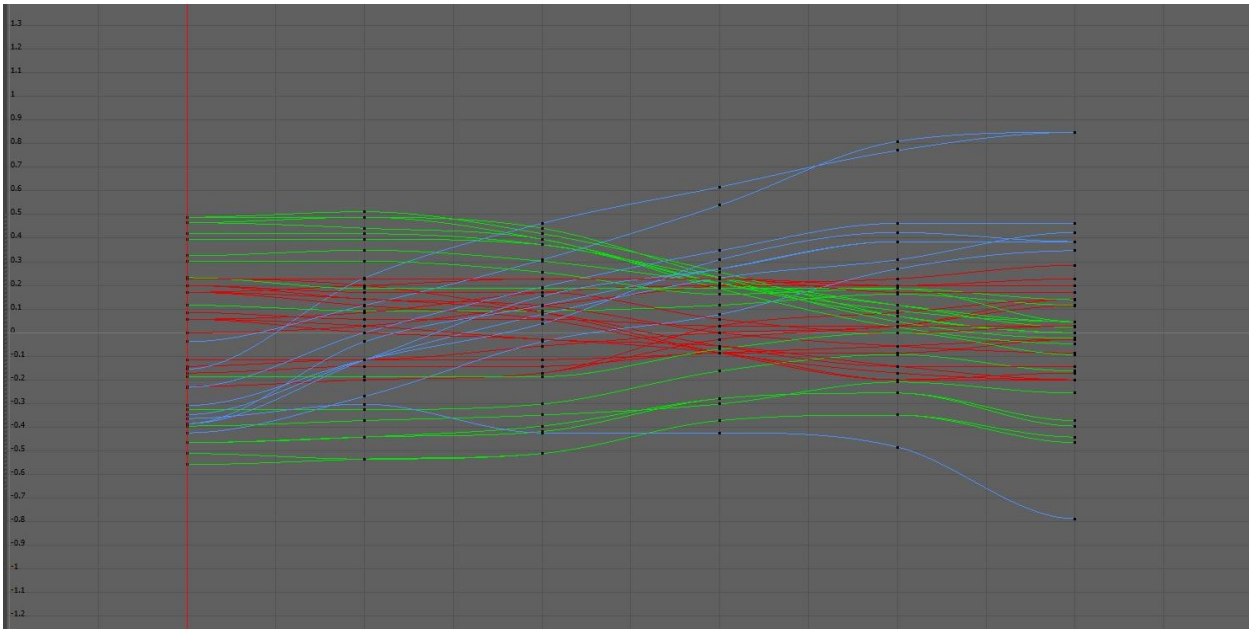


Figure 5.18 Animation graph representing the transition between visemes 14 and 20.

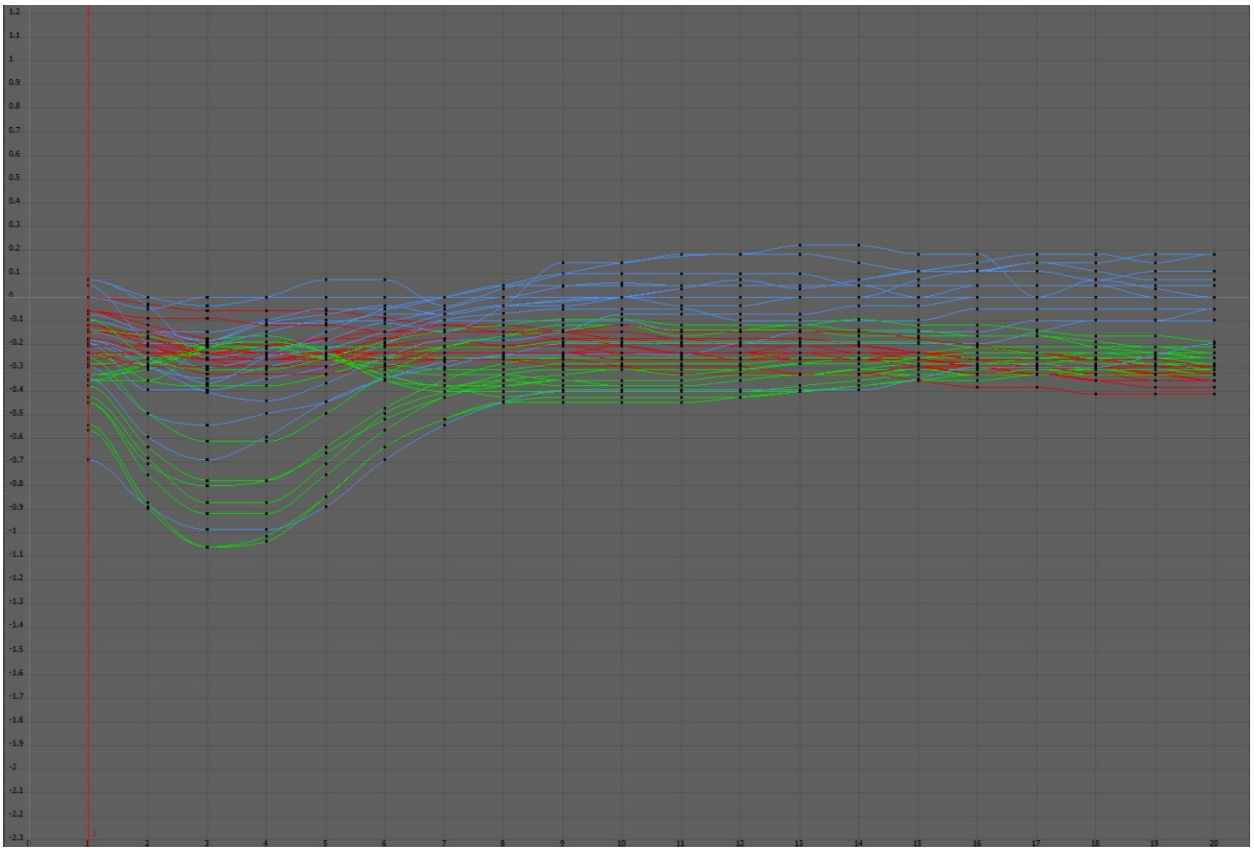


Figure 5.19 Animation graph representing the transition between visemes 20 and 0.

The software produced by this prototype is a script in Autodesk Maya which takes a user-inputted list of visemes and outputs facial animation on our custom facial rig. This system could be combined with a text-to-phoneme or audio-to-phoneme library to create a tool which converts user-inputted text or audio to lip animation matching the provided script.

5.3 Evaluation

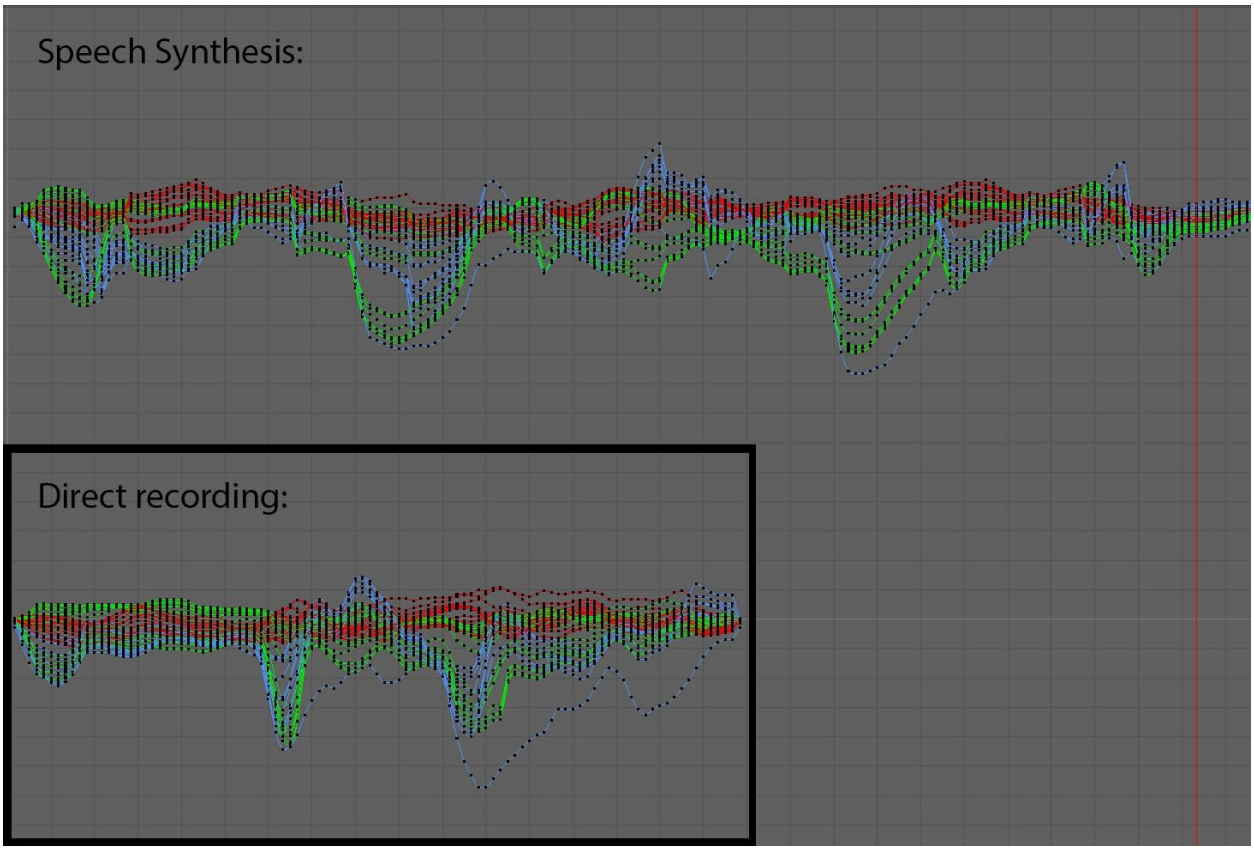


Figure 5.20 The animation graphs of “antidisestablishmentarianism” using speech synthesis compared to a direct recording.

The results produced by our prototype was evaluated both subjectively and objectively by comparing a direct capture of the word “antidisestablishmentarianism”

and a version of this word generated using our speech synthesis prototype. The word “antidisestablishmentarianism” was selected as our evaluation criteria because it is the longest word in the English language. Additionally, this word is one with a great deal of coarticulatory challenge. This is because “antidisestablishmentarianism” is all the same word instead of a sentence with as many syllables. Therefore, people are more likely to slur syllables together, causing one viseme to more heavily influence it+-s neighbours.

Figure 5.18 compares the animation graphs created using these two methods. The lower, smaller, graph illustrates the animation data of our evaluation word as it was recorded directly using our three camera system. The entire word was processed together in one XML file and the data was imported into Autodesk Maya. This is an example of purely performance-based speech animation.

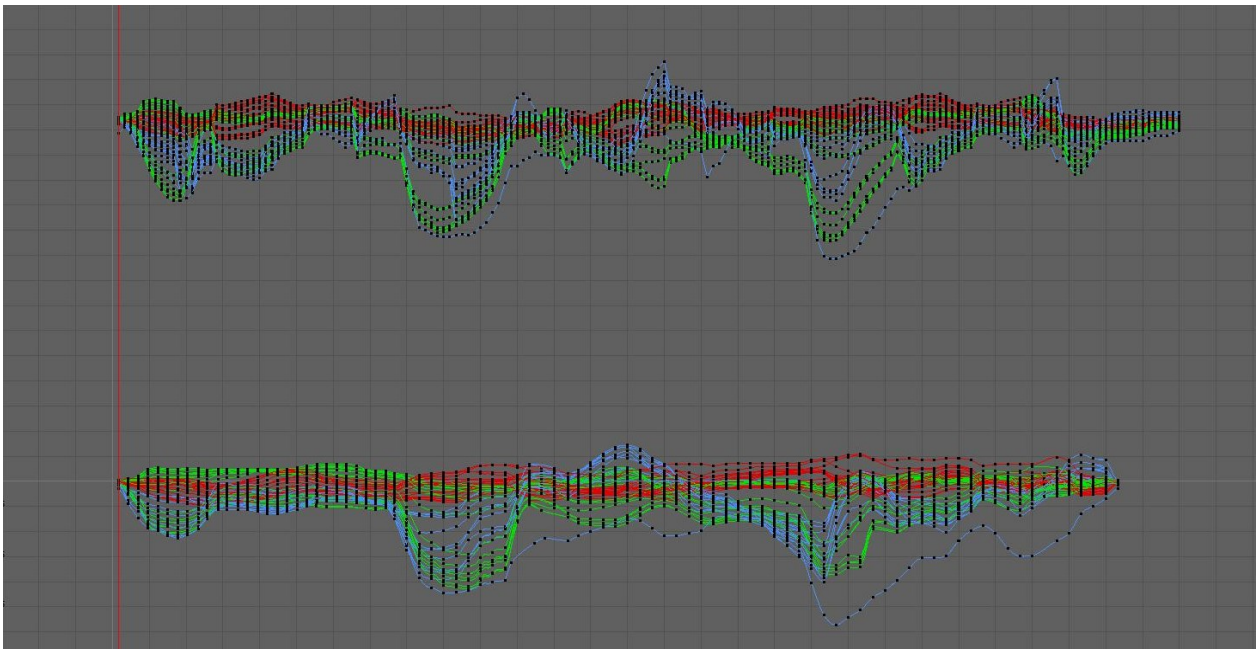
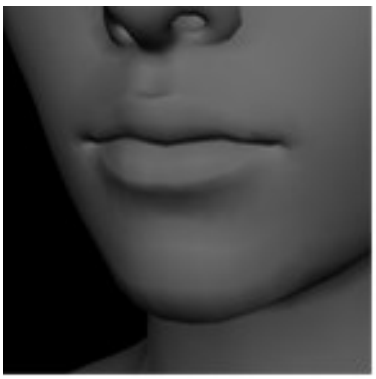


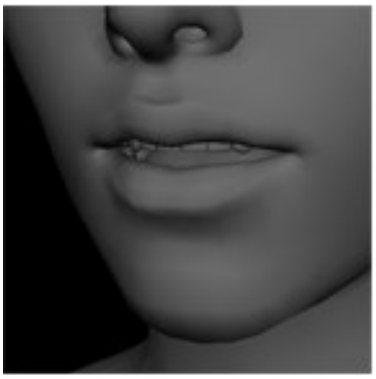
Figure 5.21 Animation graphs from 5.20 after scaling timeline to the same speed.



frame 0

frame 5

frame 10

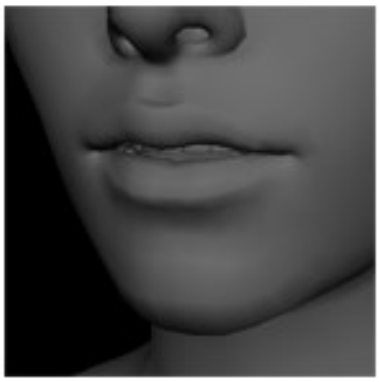


frame 15

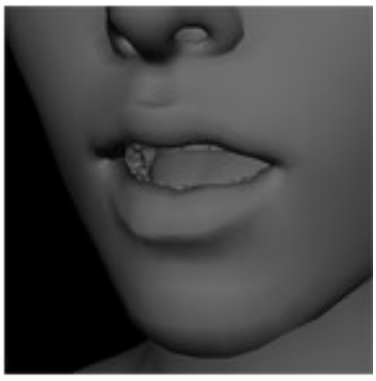
frame 20

frame 25

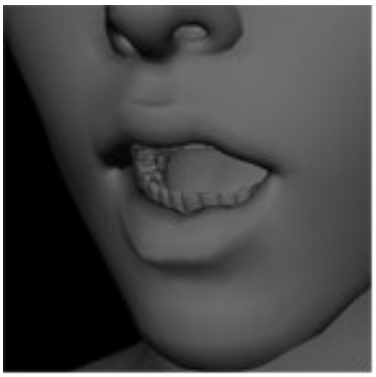
frame 30



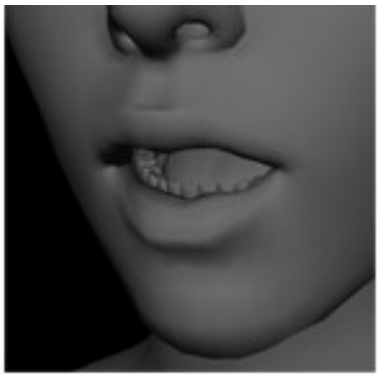
frame 35



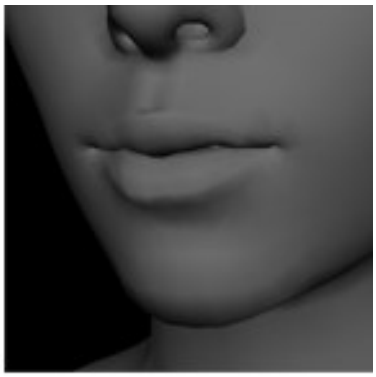
frame 40



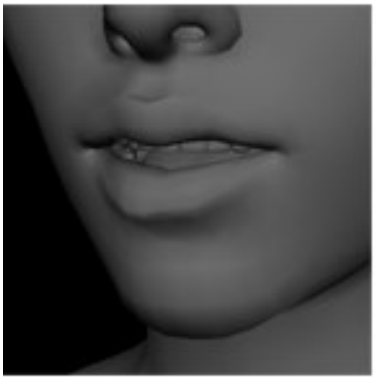
frame 45



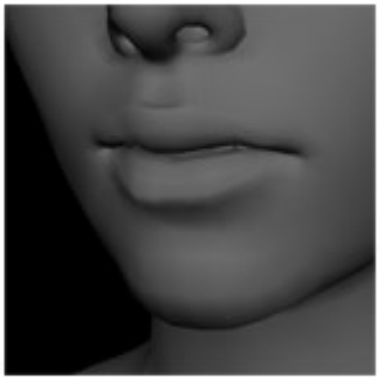
frame 50



frame 55

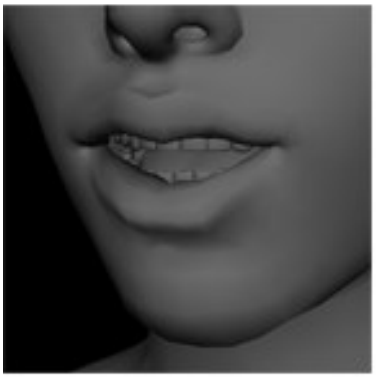


frame 60

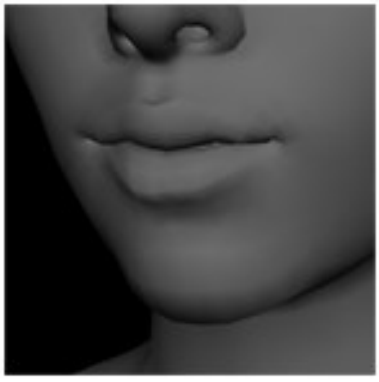


frame 65

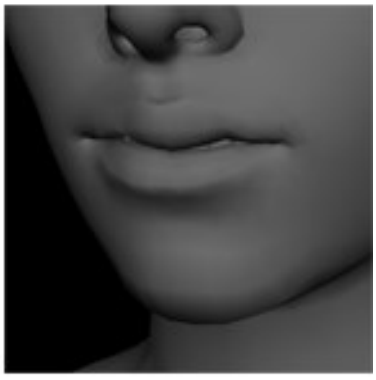
frame 70



frame 75



frame 80

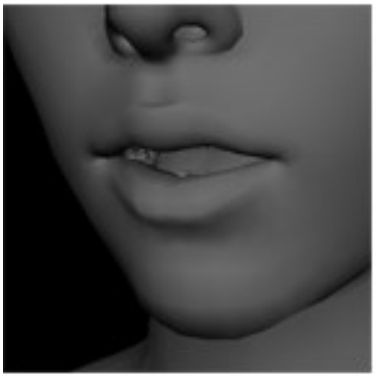


frame 85

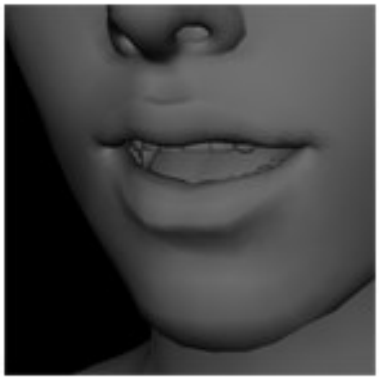
frame 90

frame 95

frame 100



frame 105

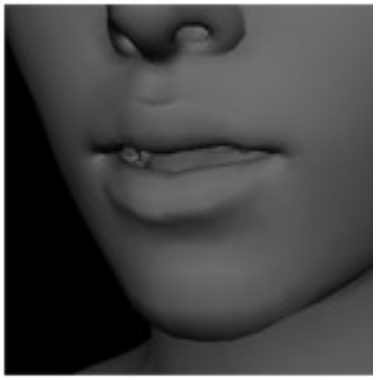


frame 110

frame 115

frame 120

frame 125



frame 130

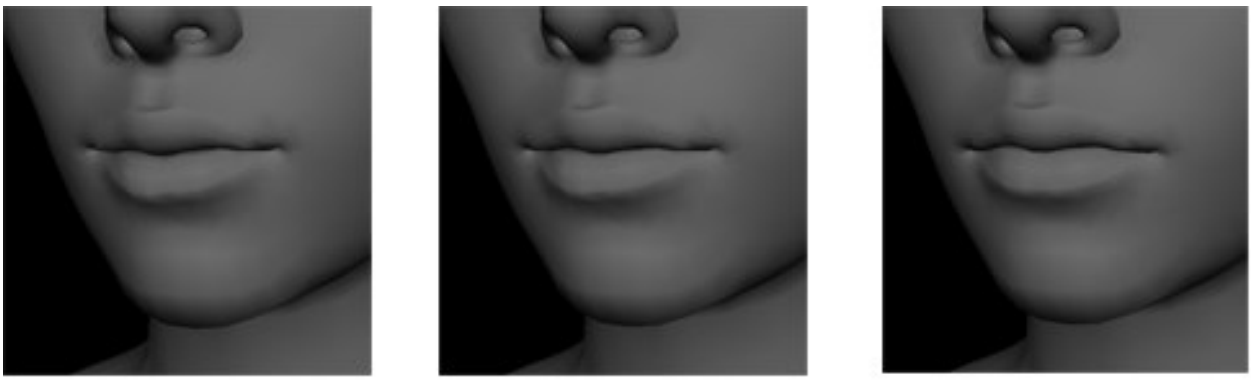


Figure 5.22 Images of the animation sequence.

Conversely, the larger upper graph shows the animation data produced of the evaluation phrase using our speech synthesis method. Again, in these graphs red represents the x translation, blue represents the z translation, and green represents the y translation.

Initially the direct recording of “antidisestablishmentarianism” is shorter than the synthesized version. The direct recorded version was 100 frames long, while the synthesized version was just over 150 frames. This is because participants spoke slowly and deliberately during the survey while speaking short words, and syllables became quicker in such a long word. It is also possible that participants were anxious about pronouncing such a challenging word, and therefore spoke more quickly. Regardless of the difference in length, we can see that the shape of the curves in both graphs share very similar paths.

In order to mathematically compare these two groups of animation curves, we first stretched them horizontally to make them match in length, essentially slowing down

the direct recording. Once the two graphs share the same timing information, see figure 5.19, they qualitatively appear very similar.

To quantitatively describe their similarity, we calculated the average Euclidian distance. This was done using a python script to compare the distance between the position of each pair of controllers (from the direct capture and the synthetic speech) on each frame, and taking the average. The value calculated for this match was 0.5 units, which is approximately 10% of the range of motion exhibited in this animation. This equates to a 90% level of accuracy.

CHAPTER SIX

Conclusions and Future Work

6.1 Conclusions

Realistic speech animation is challenging due to how attuned people are to the subtle nuances of each other's faces. Speech is one of our primary means of communication, and can be influenced by a wealth of factors from age and race to emotion and emphasis, to how dry the speaker's lips may be.

Many data-driven facial animation methods endeavour to solve the problem of speech synthesis by selecting visemes (the different mouth shapes made in a language) and concatenating them. Transitions between these viseme nodes are calculated algorithmically using morphing algorithms that do not always reflect the nuances of speech, feeling less organic. Unfortunately, these methods do not always properly reflected the way that visemes can affect the visemes that follow and precede them. This effect is called coarticulation.

Our method proposes that we should use the transition between visemes as the unit that is concatenated in speech synthesis, instead of the traditional static viseme. In developing a prototype of this method for this study, we performed the following work:

- The selection and refinement of a phoneme-to-viseme map which determined which viseme transitions would be recorded for our prototype.
- The development of an optical flow motion capture system featuring three high-fps cameras, seventeen blue semi-spherical 4mm adhesive markers, a custom

blob detection algorithm, and the composition of a script for the participants to read containing each transition of the selected visemes.

- The creation of a mouth rig in Autodesk Maya that deforms based on the translations of controllers representing the markers being tracked in our capture system.
- The implementation of the marker data acquired using the capture method onto the mouth rig using a python script.

This prototype was quantitatively evaluated to be accurate up to 0.5 Maya units, or 10% of the range of motion, meaning that the synthesized lip synchronization was 90% accurate. Qualitatively, the procedural animation of the word "antidisestablishmentarianism" and the directly recorded version both appear to be fairly accurate representations of speech, merely with subtle differences. The synthesized version was slower and more deliberately enunciated. In reality, any two recordings of the same word would not be precisely the same, so this comparison is quite good. In fact, it is this nuance that differentiates a more organic system from a mechanical one. The similarity in shape of the animation curves of each method also suggest that our phoneme-to-viseme map has effectively captured the relevant mouth shapes.

A system based off of Viseme Transition Units does involve much less interpolation of data than traditional morphing or linear transition methods. However, it is important to note that some interpolation remains necessary in our system. Autodesk Maya automatically interpolates between each key frame according to the animation curve produced by all key frames in a sequence. When people speak they do not always form visemes in exactly the same way, therefore some interpolation was

necessary to create smooth transitions between the Viseme Transition Units. Fortunately, static visemes (which are the transitions between Viseme Transition Units) are fairly easy to produce.

To conclude, we have developed a prototype able to match recorded speech with a high percentage of accuracy, using viseme transitions as nodes in the stead of traditional static visemes. This conceptual shift in mapping human speech offers the potential for more realistic and nuanced speech synthesis that could save time for animation professionals and money for industry leaders.

6.2 Future work

The end goal of this research is to achieve new levels of videorealism in digital characters and to facilitate the animation pipeline. While the procedure of capturing footage and converting it into useable data can be time consuming using our method, the product that could be developed would save a great deal of time for animation teams. With the acquisition and exploration of more robust data, one could develop software using this method that enables animators to toggle parameters such as age, accent, or gender of a character. Researchers could continue to support and expand the database of transitions and clients would subscribe to access.

In addition to a more robust data-set, future work could include an exploration of scaling emphasis of viseme transitions based on their placement within a phrase and the length of a phrase. This would enable a more organic solution to the timing discrepancies we encountered when comparing a synthesized phrase to a recorded phrase. Scale parameters to be explored include amplitude of the animation curves and length of the animation curve.

While a simple quantitative evaluation of our prototype was promising, further qualitative evaluation could be pursued in the form of a user study. In this study, participants could be invited to view animations produced using Viseme Transition Units and another lip synchronization method, and asked which one felt more intuitive or realistic.

Two other areas of research that would another level of quality to a product of this kind would be a tongue animation system - currently our rig supports teeth, mouth, and jaw animation only - and audio input. Audio input would enable users to input an audio file or recorded speech, and the software could then derive the timing and emphasis data and apply it to the speech synthesis system.

References

[1] Cosatto, Eric, Jörn Ostermann, Hans Peter Graf, and Juergen Schroeter. "Lifelike talking faces for interactive services." *Proceedings of the IEEE* 91, no. 9 (2003): 1406-1429.

[2] Ostermann, Joern, and Axel Weissenfeld. "Talking faces-technologies and applications." In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3, pp. 826-833. IEEE, 2004.

[3] Trutoiu, Laura C., Elizabeth J. Carter, Iain Matthews, and Jessica K. Hodgins. "Modeling and animating eye blinks." *ACM Transactions on Applied Perception (TAP)* 8, no. 3 (2011): 17.

[4] Mines, M. Ardussi, Barbara F. Hanson, and June E. Shoup. "Frequency of occurrence of phonemes in conversational English." *Language and speech* 21, no. 3 (1978): 221-241.

[5] Giegerich, Heinz J. *English Phonology: An Introduction.* Cambridge, England: Cambridge University Press, 1992.

[6] Roach, Peter. *English Phonetics and Phonology: A Practical Course*. 3rd ed. Cambridge, U.K.: Cambridge University Press, 2000.

[7] Microsoft. “Tellme Services Documentation.” Last modified 2015.

https://msdn.microsoft.com/en-us/library/ff929020.aspx.

[8] Microsoft. “Microsoft Speech API (SAPI) 5.3.” Last modified 2015. https://msdn.microsoft.com/en-us/library/ms720881(v=vs.85).aspx

[9] Fisher, Cletus G. "Confusions among visually perceived consonants." *Journal of Speech, Language, and Hearing Research* 11, no. 4 (1968): 796-804.

[10] Lander, Jeff. "Read my lips: facial animation techniques." *Game Developer Magazine, CMP Media Group* (1999): 17-21.

[11] Rodgers, Jake. "Animating Facial Expressions." *Game Developer Magazine*(1998).

[12] Rizvic, Selma, and Zikrija Avdagic. "Phoneme reduction in automated speech for computer animation." In *Proceedings of the 20th spring conference on Computer graphics*, pp. 89-96. ACM, 2004.

[13] Kmett, Edward A. "Real-Time Viseme Extraction." (2005).

[14] Cappelletta, Luca, and Naomi Harte. "Phoneme-to-viseme Mapping for Visual Speech Recognition." In *ICPRAM (2)*, pp. 322-329. 2012.

[15] Cappelletta, Luca, and Naomi Harte. "Viseme definitions comparison for visual-only speech recognition." In *Signal Processing Conference, 2011 19th European*, pp. 2109-2113. IEEE, 2011.

[16] Bozkurt, Elif, Cigdem Eroglu Erdem, Engin Erzin, Tanju Erdem, and Mehmet Ozkan. "Comparison of phoneme and viseme based acoustic units for speech driven realistic lip animation." *Proc. of Signal Proc. and Communications Applications* (2007): 1-4.

[17] Visser, Michiel, Mannes Poel, and Anton Nijholt. "Classifying visemes for automatic lipreading." In *Text, Speech and Dialogue*, pp. 349-352. Springer Berlin Heidelberg, 1999.

[18] Mason. “Getting your animatronic head to speak: From Phonemes to Visemes.” Last modified February 9, 2009. http://profmason.com/?p=743

[19] Autodesk Softimage 2011 Subscription Advantage Pack. “Working with Visemes.” Last modified 2011. http://softimage.wiki.softimage.com/xsidocs/face_lipsync_WorkingwithVisemes.htm

[20] Turkmani, Aseel, Adrian Hilton, Philip JB Jackson, and James Edge. "Visual analysis of lip coarticulation in VCV utterances." In *Interspeech 2007: 8TH Annual Conference of the International Speech Communication Association,* vols 1-4, pp. 1281-1284. 2007.

[21] Mattheyses, Wesley, Lukas Latacz, and Werner Verhelst. "Comprehensive many-to-many phoneme-to-viseme mapping and its application for concatenative visual speech synthesis." *Speech Communication* 55, no. 7 (2013): 857-876.

[22] Cohen, Michael M., and Dominic W. Massaro. "Modeling coarticulation in synthetic visual speech." In *Models and techniques in computer animation*, pp. 139-156. Springer Japan, 1993.

[23] Montgomery, Allen A. "Development of a model for generating synthetic animated lip shapes." *The Journal of the Acoustical Society of America* 68, no. S1 (1980): S58-S59.

[24] Deng, Zhigang, and Ulrich Neumann. "Computer Facial Animation: A Survey," in *Data-driven 3D facial animation*. London, England: Springer-Vertag, 2008, pp. 1-29.

[25] Deng, Zhigang, and Ulrich Neumann. "A User Interface Technique for Controlling Blendshape Interference," in *Data-driven 3D facial animation*. London, England: Springer-Vertag, 2008, pp. 132-145.

[26] Parke, Frederic I., and Keith Waters. *Computer facial animation*. CRC Press, 2008.

[27] Radovan, Mauricio, and Laurette Pretorius. "Facial animation in a nutshell: past, present and future." In *Proceedings of the 2006 annual research conference of the*

South African institute of computer scientists and information technologists on IT research in developing countries, pp. 71-79. South African Institute for Computer Scientists and Information Technologists, 2006.

[28] Iyaniwura, Anuoluwa. "Optical Motion Capture for Performance-based Facial Animation." PhD diss., Carleton University Ottawa, 2008.

[29] Parke, Frederic I. "Techniques for facial animation." In *New trends in animation and visualization*, pp. 229-241. Wiley-Interscience, 1991.

[30] Noh, Jun-yong, and Ulrich Neumann. *A survey of facial modeling and animation techniques*. USC Technical Report, 99–705, 1998.

[31] Platt, Stephen M., and Norman I. Badler. "Animating facial expressions." In *ACM SIGGRAPH computer graphics*, vol. 15, no. 3, pp. 245-252. ACM, 1981.

[32] Terzopoulos, Demetri, and Keith Waters. "Techniques for realistic facial modeling and animation." In *Computer Animation'91*, pp. 59-74. Springer Japan, 1991.

[33] Waters, Keith, and Demetri Terzopoulos. "Modelling and animating faces using scanned data." *The Journal of Visualization and Computer Animation* 2, no. 4 (1991): 123-128.

[34] Kähler, Kolja. "A head model with anatomical structure for facial modelling and animation." PhD diss., Universitätsbibliothek, 2003.

[35] Terzopoulos, Demetri, and Keith Waters. "Physically- based facial modelling, analysis, and animation." *The journal of visualization and computer animation* 1, no. 2 (1990): 73-80.

[36] Lee, Yuencheng, Demetri Terzopoulos, and Keith Waters. "Constructing physics-based facial models of individuals." In *Graphics Interface*, pp. 1-1. Canadian Information Processing Society, 1993.

[37] Magnenat-Thalmann, Nadia, E. Primeau, and Daniel Thalmann. "Abstract muscle action procedures for human face animation." *The Visual Computer* 3, no. 5 (1988): 290-297.

[38] Waters, Keith. "A muscle model for animation three-dimensional facial expression." In *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 17-24. ACM, 1987.

[39] Koch, Rolf M., Markus H. Gross, and Albert A. Bosshard. "Emotion editing using finite elements." In *Computer Graphics Forum*, vol. 17, no. 3, pp. 295-302. Blackwell Publishers Ltd, 1998.

[40] Kähler, Kolja, Jörg Haber, and Hans-Peter Seidel. "Geometry-based muscle modeling for facial animation." In *Graphics Interface*, vol. 2001, pp. 37-46. 2001.

[41] Kalra, Prem, Angelo Mangili, Nadia Magnenat Thalmann, and Daniel Thalmann. "Simulation of facial muscle actions based on rational free form deformations." In *Computer Graphics Forum*, vol. 11, no. 3, pp. 59-69. Blackwell Science Ltd, 1992.

[42] Borshukov, George, Dan Piponi, Oystein Larsen, John P. Lewis, and Christina Tempelaar-Lietz. "Universal capture-image-based facial animation for The Matrix Reloaded." In *ACM Siggraph 2005 Courses*, p. 16. ACM, 2005.

[43] Ezzat, Tony, and Tomaso Poggio. "Facial analysis and synthesis using image-based models." In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pp. 116-121. IEEE, 1996.

[44] Ezzat, Tony, Gadi Geiger, and Tomaso Poggio. *Trainable video realistic speech animation*. Vol. 21, no. 3. ACM, 2002.

[45] Bregler, Christoph, Michele Covell, and Malcolm Slaney. "Video rewrite: Driving visual speech with audio." In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 353-360. ACM Press/Addison-Wesley Publishing Co., 1997.

[46] Arya, Ali, and Babak Hamidzadeh. "FIX: Feature-based image transformations for face animation." In *Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on*, pp. 554-558. IEEE, 2003.

[47] Williams, Lance. "Performance-driven facial animation." In *ACM SIGGRAPH Computer Graphics*, vol. 24, no. 4, pp. 235-242. ACM, 1990.

[48] Erber, Norman P., and Carol Lee De Filippo. "Voice/mouth synthesis and tactual/visual perception of/pa, ba, ma." *The Journal of the Acoustical Society of America* 64, no. 4 (1978): 1015-1019.

[49] Brooke, N. M., and Quentin Summerfield. "Analysis, synthesis, and perception of visible articulatory movements." *Journal of phonetics* (1983).

[50] Parke, Frederick I. "Computer generated animation of faces." In *Proceedings of the ACM annual conference-Volume 1*, pp. 451-457. ACM, 1972.

[51] Parke, Frederic I. "Parameterized models for facial animation." *IEEE computer graphics and applications* 9, no. 2 (1982): 61-68.

[52] Ekman, Paul, and Wallace Friesen. "Facial Action Coding System: A technique for the measurement of facial movements." *Consulting Psychologist* 2 (1978).

[53] Lewis, John P., and Frederic I. Parke. "Automated lip-synch and speech synthesis for character animation." In *ACM SIGCHI Bulletin*, vol. 17, no. SI, pp. 143-147. ACM, 1987.

[54] Parke, Frederic I. "A model for human faces that allows speech synchronized animation." *Computers & Graphics* 1, no. 1 (1975): 3-4.

[55] Parke, Frederic I. "Control parameterization for facial animation." In *Computer Animation '91*, pp. 3-14. Springer Japan, 1991.

[56] Pearce, Andrew, Brian Wyvill, Geoff Wyvill, and David Hill. "Speech and expression: A computer solution to face animation." In *Graphics Interface*, vol. 86, pp. 136-140. 1986.

[57] Nahas, Monique, Herve Huitric, and Michel Saintourens. "Animation of a b-spline figure." *The Visual Computer* 3, no. 5 (1988): 272-276.

[58] Cohen, Michael M., and Dominic W. Massaro. "Synthesis of visible speech." *Behavior Research Methods, Instruments, & Computers* 22, no. 2 (1990): 260-263.

[59] Schlesinger, Brian, Michael Mensch, Christopher Rindosh, Joe Votta, and Yunfeng Wang. "A Semi-Autonomous Interactive Robot." In *AAAI*, pp. 1974-1975. 2006.

[60] Luerssen, Martin, Trent Lewis, and David Powers. "Head X: Customizable Audiovisual Synthesis for a Multi-purpose Virtual Head." In *AI 2010: Advances in Artificial Intelligence*, pp. 486-495. Springer Berlin Heidelberg, 2011.

[61] Ezzat, Tony, and Tomaso Poggio. "Videorealistic talking faces: A morphing approach." In *Audio-Visual Speech Processing: Computational & Cognitive Science Approaches*. 1997.

[62] Schreer, Oliver, Roman Englert, Peter Eisert, and Ralf Tanger. "Real-time vision and speech driven avatars for multimedia applications." *Multimedia, IEEE Transactions on* 10, no. 3 (2008): 352-360.

[63] Olive, Joseph P., Alice Greenwood, and John Coleman. *Acoustics of American English speech: a dynamic approach*. Springer Science & Business Media, 1993.

[64] Z. Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.

[65] Intel OpenCV Computer Vision Library (C++), http://www.intel.com/research/mrl/research/opencv/

[66] Point Grey Research, Inc. “FlyCapture SDK.” Last modified 2015. http://www.ptgrey.com/flycapture-sdk

[67] Ferrucio. “An XML Writer for C++.” Last modified June 7, 2010. https://tlzprgmr.wordpress.com/2010/06/07/an-xml-writer-for-c/

[68] Weta Digital, https://www.wetafx.co.nz/

[69] Industrial Light & Magic, http://www.ilm.com/

[70] Digital Domain, http://www.digitaldomain.com/

[71] Quantic Dream, http://www.quanticdream.com/fr/

[72] Sony Pictures Imageworks, http://www.imageworks.com/

[73] Weisstein, Eric W. "Lissajous Curve."
http://mathworld.wolfram.com/LissajousCurve.html

[74] Jacono, Andrew A. "A new classification of lip zones to customize injectable lip augmentation." *Archives of facial plastic surgery* 10, no. 1 (2008): 25-29.

[75] FaceShift, http://www.faceshift.com/

[76] Ali, Itimad Raheem, Ghazali Sulong, and Hoshang Kolivand. "Realistic Lip Syncing for Virtual Character Using Common Viseme Set." *Computer and Information Science* 8, no. 3 (2015): 71.

[77] McAllister, David F., Robert D. Rodman, Donald L. Bitzer, and Andrew S. Freeman. "Lip synchronization of speech." In *Audio-Visual Speech Processing: Computational & Cognitive Science Approaches*. 1997.

[78] Zorić, Goranka, and Igor S. Pandžić. "Real-time language independent lip synchronization method using a genetic algorithm." *Signal processing* 86, no. 12 (2006): 3644-3656.

[79] Provine, J. A., and Leonard T. Bruton. "Lip synchronization in 3-d model based coding for video-conferencing." In *Circuits and Systems, 1995. ISCAS'95., 1995 IEEE International Symposium on*, vol. 1, pp. 453-456. IEEE, 1995.

[80] Aggarwal, Sucharu, and Alka Jindal. "Comprehensive overview of various lip synchronization techniques." In *Biometrics and Security Technologies, 2008. ISBAST 2008. International Symposium on*, pp. 1-6. IEEE, 2008.

APPENDIX A

Phoneme-to-Viseme Maps

This appendix is a compilation of the more important phoneme-to-viseme maps discussed in chapter 2. These maps identify the different mouth shapes (visemes) in the American English language, and all the possible sounds (phonemes) that can be made with each shape.

A.1 Lander, 1999

When asked to develop a speech synthesis system, Jeff Lander carefully evaluated different phonemes based on their sonic and physical characteristics. While the system he eventually designed supported a flexible amount of visemes, meaning the system could be modified to support any number of visemes, he did decide on a set of 13 visemes that he felt best represented the selection of phonemes in American English [10].

The visemes that Jeff Lander selected are described as follows:

1. Closed lips.
2. Pursed lips.
3. Rounded open lips with corner of lips slightly puckered.
4. Lower lip drawn up to upper teeth.
5. Tongue between teeth, no gaps on sides.
6. Tip of tongue behind open teeth, gaps on sides.

7. Relaxed mouth with mostly closed teeth with pinkness of tongue behind teeth (tip of tongue on ridge behind upper teeth).
8. Slightly open mouth with mostly closed teeth and corners of lips slightly tightened.
9. Slightly open mouth with mostly closed teeth.
10. Wide, slightly open mouth.
11. Neutral mouth with slightly parted teeth and slightly dropped jaw.
12. Very round lips, slight dropped jaw.
13. Open mouth with very dropped jaw.

A.2 Bozkurt et al. 2007

A larger selection of visemes was put forth by Bozkurt et al in 2007. This mapping contains 16 visemes to describe the 44 phonemes in the English language. Their mapping is illustrated in table A-1. In addition to their work on viseme mapping, Bozkurt et al's research concluded that data-driven speech synthesis models which use a viseme-based unit with contextual information, such as emphasis, outperforms other models of speech synthesis [16].

Viseme	**Phonemes**	
1	pau	
2	ay, ah,	but
3	ey, eh, ae	bait, bet, bat

4	er	bird
5	ix, iy, ih, ax, axr,y	debit, beet, bit, about, butter, yacht
6	uw, uh, w	boot, book, way
7	ao, aa, oy, ow	bought, bott, boy, boat
8	aw	bout
9	g, hh, k, ng	gay, hay, key, sing
10	r	ray
11	l, d, n, en, el, t	lay, day, noon, button, bottle, tea
12	s, z	sea, zone
13	ch, sh, jh, zh	choke, she, joke, azure
14	th, dh	thin, then
15	f, v	fin, van
16	m, em, b, p	mom, bottom, bee, pea

Table A-1 Phoneme-to-viseme map by Bozkurt et al [16].

A.3 Microsoft Speech API

The Microsoft Speech API is an Application Programming Interface (API) developed by Microsoft that allows the use of speech synthesis and speech recognition within Windows applications. The Microsoft Speech API is used by central Microsoft software such as Microsoft Office programs, and is shipped as part of the Windows operating system.

In American English, the Microsoft Speech API supports 23 visemes, referred to within the API as Speech Viseme Types [8]. Microsoft's viseme mapping compares closely with the mapping described by Bozkurt et al in 2007, but includes additional visemes.

Microsoft's phoneme-to-viseme mapping is one of the longer selections of visemes, and was the basis for the phoneme-to-viseme mapping used in our study. However, it was found through capture tests and analysis that in Canadian English, the "h" is usually silent and the "ao" phoneme is visually very similar to the "aa" phoneme and both of these visemes were culled for the purpose of our research.

Viseme	**Phoneme**	**Examples**
0	silence	
1	ae ax ah/ a	Bat, about, but
2	aa / o u	Bob
3	ao	Bought
4	ey eh / e /ā/	Bait, Bet,
5	er /ə/	Bird
6	y iy ih ix / i /ē/	You / beat / bit /
7	w uw /ü/	Won / boot
8	ow uh / oo /ō/	Boat, book
9	aw /ow/	down
10	oy /oi/	boy
11	ay /ī/	bite
12	hh	hot

13	r	rat
14	l	lot
15	s z	sit
16	sh ch jh zh	Shut / church / jump / measure
17	th dh	Thick / that
18	f v	Fog / vat
19	d t n	Fig / top / nod
20	k g ng	Cat / got / sing
21	p b m	Pot / bet / mom

Table A-2 Phoneme-to-viseme map by Microsoft Research [8].