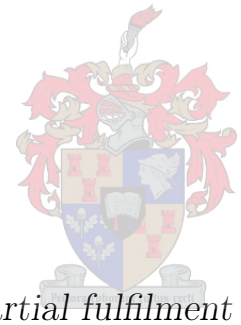
Improving Visual Speech Synthesis using Decision Tree Models

by

Christiaan Frans Rademan



Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Electronic Engineering in the Faculty of Engineering at Stellenbosch University

Supervisor: Prof. T.R. Niesler

March 2016

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2016

Abstract

Improving Visual Speech Synthesis using Decision Tree Models

C.F. Rademan

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MScEng (Elec)

March 2016

Visual speech synthesis is essential for believable virtual character interaction. Traditionally, animation artists recreate the oral motions expected from speech utterances.

In response, we present decision tree-based clustering techniques which are employed in automating visual speech animation. This is achieved using a small dataset of phonetically-annotated audiovisual speech.

Our work focuses on extending existing tree-based clustering algorithms by improving on the modelling of coarticulation effects. This is accomplished by capturing the motion of natural speech segments, referred to as dynamic visemes, and conserving their parameters during clustering and speech synthesis. Dynamic visemes are defined as the trajectories of oral features segmented by triphone boundaries. By applying simple search and concatenation criteria, our visual speech synthesis system uses decision trees to better predict which dynamic visemes to use.

Experimentation guided all design decisions, suggesting which oral features were of greatest importance, identifying an appropriate dynamic viseme length and finding an effective interpolation method for conserving coarticulation.

We evaluate the performance of our visual speech synthesis models by computing squared error differences between synthesised and measured feature trajectories. Perceptual tests also asked participants to compare virtual characters animated by model outputs. Both measured and perceptual tests show that our approaches lead to a clear improvement over a comparable baseline.

Through our research, we intended on making speech synthesis more accessible. Therefore, the conversational agents are based on the freely available MakeHuman and Blender software components. The customised oral feature motion capture system is also easily reproduced and requires only consumer grade recording equipment.

Uittreksel

Verbetering Visuele Spraaksintese Behulp Besluitnemingsboom Modelle

("Improving Visual Speech Synthesis using Decision Tree Models")

C.F. Rademan

Departement Elektriese en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MScIng (Elec)

Maart 2016

Visuele spraaksintese is noodsaaklik om geloofwaardige interaksie met virtuele karakters moontlik te maak. In die verlede het animasiekunstenaars mondbewegings vanaf werklike spraak nageboots. In hierdie studie bied ons tegnieke aan wat gebaseer is op saambondeling met behulp van besluitnemingsbome. Hierdie tegnieke word gebruik om die animasie van visuele spraak te outomatiseer, en maak gebruik van 'n klein datastel van foneties geannoteerde oudiovisuele spraak.

Ons werk fokus op die uitbrei van bestaande besluitnemingsboom-saambondelingsalgoritmes, deur die modellering van koartikulasie-effekte te verbeter. Dit word moontlik gemaak deur eers die beweging van natuurlike spraaksegmente (viseme) vas te vang, en dan hul parameters te bewaar tydens die saambondeling en spraaksintese.

Dinamiese viseme word gedefinieer as die trajekte van mondeienskappe, gesegmenteer deur trifoongrense.

Deur eenvoudige soek- en saamvoegingskriteria toe te pas, kan ons visuele spraaksintesestelsel van besluitnemingsbome gebruik maak om beter te voorspel watter viseme aangewend moet word.

Alle ontwerpsbesluite is deur eksperimintering gelei, om bv. die mondeienskappe van grootste belang te identifiseer, om 'n gepaste viseemlengte vas te stel, en om 'n effektiewe interpolasiemetode te vind wat koartikulasie bewaar.

Ons evalueer die werksverrigting van ons visuele spraaksintesemodel deur die kwadraatfout tussen die gesintetiseerde en gemete eienskaptrajekte te bereken. Tydens perseptuele toetse is deelnemers gevra om die geloofwaardigheid van virtuele karakters, aangedryf deur die modeluittrees, te beoordeel. Beide gemete en perseptuele toetse het aangedui dat die voorgestelde tegnieke 'n duidelike verbetering bo 'n geskikte basislynmeting toon.

Die doel van hierdie navorsing is om spraaksintese meer toeganklik te maak. Om hierdie rede is die gespreksagente gebou op die vrylik beskikbare MakeHuman- en Blendersagtewarekomponente. Die pasgemaakte mondeienskap-bewegingsaftaster is ook eenvoudig om te herproduseer, en benodig slegs verbruikersgraad-opneemtoerusting.

Publication

Part of this thesis' findings were presented and published in the proceedings of FAAVSP '15 in Vienna, Austria. The paper can be accessed in the ISCA archive available at: `http://www.isca-speech.org/archive/avsp15`.

Bibliographic reference: Rademan, C.F., Niesler, T.: Improved visual speech synthesis using dynamic viseme k-means clustering and decision trees. In: *FAAVSP'15-The 1st Joint Conference on Facial Analysis, Animation, and Auditory-Visual Speech Processing*, pp. 169-174. 2015.

Acknowledgements

The author would like to thank Prof. T.R. Niesler for his unwavering guidance and support throughout this work. This work was supported in part by the National Research Foundation of the Republic of South Africa (grant TP13081327740) and the MIH Media Lab. The author would also like to thank Alison Wileman for her phonetic annotations and the Blender and Python on-line communities for sharing their knowledge.

Dedications

This thesis is dedicated to Chris, Linda and Kathleen Rademan, Candice Townsend and all the friends I made along the way.
To the reader, I can only hope that my work inspires you to discover and enjoy the complexity of life's everyday.

Contents

List of Figures

List of Tables

Nomenclature

Glossary Of Abbreviations

3D	Three Dimensional
ANN	Artificial Neural Network
BAP	Body Animation Parameters
CV	Cross-Validation
FA	Facial Animation
FACS	Facial Action Coding System
FAP	Face Animation Parameters
FPS	Frames Per Second
IPA	International Phonetic Alphabet
HMI	Human Machine Interfacing
HMM	Hidden Markov Model
HCI	Human Computer Interaction
LPC	Linear Predictive Coding
PDF	Probability Density Function
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
TTS	Text To Speech Engine
TDNN	Time Delay Neural Network
VTTS	Visual Text To Speech
VSS	Visual Speech Synthesis
VQ	Vector Quantization
X-SAMPA	Extended Speech Assessment Methods Phonetic Alphabet

Chapter 1

Introduction

Advances in technology have increased the demand for natural, human-like communication interfaces. For example, virtual conversation agents can be found in computer games and in other human-like interface mediums, such as public or personal assistance avatars. Character-based visual speech synthesis (VSS) plays a critical role in these interactions. In response, this thesis presents models for improving virtual character-based VSS. This research falls within human-computer interaction (HCI) research and can be extended into the subject area of human-machine interfacing (HMI).

VSS deals with the reproduction of visual speech cues expressed by the human face during speech articulation. Our primary aim is to select and improve upon a suitable approach to VSS. Our secondary aim is to make our solution accessible by using open source software and readily available consumer equipment.

The project's primary aim is motivated by the importance of bimodal audiovisual communication for aiding conversational interpretation. Agelfor *et al.* [1] and Summerfield [2] demonstrate the importance of lip reading, concluding that a conversational agent's visual speech can compensate for deficiencies in the audio. Conversational agents also deepen the level of engagement with an interface, making it more life-like and engaging compared to audio alone [3].

The secondary aim will allow others to efficiently develop their own avatars for research purposes. There are currently no existing simple-to-use toolkits or flexible platforms for dynamic character generation and control. The pipeline developed in this study makes VSS, as well as other forms of data-driven or algorithmic control of virtual characters, more accessible.

Both aims are developed to allow VSS to exist in a resource limited environment. This means that, besides using 'off-the-shelf' equipment and open source software, the VSS algorithms must require relatively little training data and the chosen software and programming languages should have active online support communities.

1.1 Problem Statement

The mapping of audio-to-visual speech is not one-to-one. In fact, visual speech alone has no accepted pre-defined units, making its segmentation and its mapping to audio units an open-ended problem.

A simplistic, but often unconvincing approach to VSS can be achieved by relating a set of static visual speech cues, known as visemes, to audio speech [4; 5; 6]. When automating this approach, however, visual speech realism comes at the price of an ever

larger set of static visemes, which, in turn, require a more complex audio-to-visual speech mapping. Using a corpus of static visemes is thus resource-intensive and is inherently subject to limitations.

Therefore, the research objective of this work is to identify and improve data-sparse approaches to modelling many-to-many audio-to-visual speech mappings. This requires a number of sub-problems to be addressed. The VSS system must be able to naturally capture coarticulation effects. It will also need to synthesise new speech sequences based on pre-formulated audio-visual relationships. This requires an elegant visual speech clustering and mapping method. Once created, this mapping must be coupled with a search and concatenation algorithm to synthesise new visual speech. The VSS systems' output must then be used to control a virtual avatar. Consequently, a motion capture and control system for the virtual avatar must also be developed. Once completed, the VSS system should be assessed using subjective and perceptual evaluations.

Furthermore, the VSS system must be attractive to individuals in under-resourced work environments. This is critical for making HCI research accessible in new settings, particularly in those with unexplored languages, a category under which much of sub-Saharan Africa falls. Therefore, the implementation of a VSS solution will rely on a versatile and automated animation pipeline that is adaptable to many languages. To further comply with low-resource requirements, the VSS approach must employ accessible software resources and avoid advanced video capturing equipment and data-intensive training algorithms.

1.2 Limitations and Assumptions

Conversational agents touch upon many aspects of study in the field of motion in graphics. Therefore, it is important to define the limitations and assumptions relating to data collection, clustering algorithms, and animation techniques.

Data limitations are a result of the high expense and level of difficulty incurred from audio phonetic annotation. To compensate for these limitations, the recorded speech should be taken from a phonetically rich dataset. To determine the effects of this limitation, the VSS system must be tested using variously sized training data sets. One objective is to gauge if there is an optimal dataset size for the developed VSS system.

The method of data capture will be dictated by the animation control technique. The recording procedure must also be easily to reproduce and cost effective, using simple and locally sourced equipment. These specifications will impact the precision and diversity of the captured data. It is also assumed that only the monitoring of speech-related, external oral features is feasible for this work. Therefore, no tongue, greater facial, eye, neck or chest motion capture recordings are considered. Instead these will remain the domain of future work.

The data clustering algorithm must be suitable for implementation on a standard desktop computer. Real-time synthesis is beyond the scope of this work, though the final VSS system should preferably be suitable to this type of usage.

The VSS clustering algorithm will be limited to speech synthesis in English using the Extended Speech Assessment Method Phonetic Alphabet (X-SAMPA) for phonetic annotations. It is assumed that the VSS technique employed will serve as a proof of concept for synthesis in other languages since X-SAMPA covers a wide spread of other languages.

The avatar generation and manipulation systems should allow for maximum diversity in human-like character design and usage. That said, there will inevitably be limitations to the characters' graphic realism and VSS. The consequences of these limitations must be considered in terms of interaction psychology and intelligibility. Perceptual tests should assess these limitations and their effects on the user.

1.3 Significance

In the course of this work, decision tree-based, time-series clustering algorithms are developed. The performance of these algorithms as VSS models is evaluated through objective and subjective analyses, which demonstrate improvement over an adapted baseline algorithm. Testing also revealed the limitations of mean squared error evaluations for predicting visual speech perception. These findings have been published in a peer-reviewed international conference paper [7] which is included in Appendix D.

As mentioned in Section 1.1, the developed system is attractive in constrained situations where neither advanced motion capture equipment nor much phonetically transcribed audio data is available. By making use of freely-available software tools, the proposed system enables small research groups in poorly resourced environments to produce flexible virtual avatars capable of VSS.

There is also an array of applications for the VSS system which go beyond the research scope of this work. These include: HCI; conversational and interactional psychology research; usage as a tool for efficient, digital media content production, as well as assisting in studying the intelligibility of visual speech and language learning.

1.4 Scope

This thesis has identified and implemented an appropriate set of facial animation tools. This involved using the open source software MakeHuman [8], for character generation, and the Blender Game Engine [9] as the animation control environment. Subsequently, a unique control technique has been implemented, referred to as scripted bone-driven shape key animation, allowing visual speech reproduction to be mastered.

In the process of developing the facial animation system, a motion capture system was devised. This identified and captured a minimised set of oral facial features, which were used to animate that character's visual speech.

A database of audio and tracked visual speech was also created using the motion capture system. Testing revealed the best method of segmenting and joining the data so to conserve and preserve coarticulation effects during VSS. The database contained oral feature trajectories, which were segmented into triphones and joined for VSS using an interpolation algorithm.

The developed algorithms were evaluated with respect to each other, as well as the training set size, and decision tree meta parameters. Testing revealed a set of suitable parameters as well as improvements on the baseline algorithm.

Chapter 2

Introduction to Speech Communication

Communication is vital to human beings, with face-to-face conversation being our primary means of exchanging information. Speech is, however, a bimodal form of communication, being both audible and visual. Visual speech deals with the visible oral cues associated with audio speech, specifically the lips, tongue and jaw, in the context of this work. Humans are well versed in interpreting oral movements, which contribute to the intelligibility of face-to-face conversation. Reproducing or synthesising these subtle and intricately linked audiovisual gestures is a non-trivial task.

To fully comprehend synthesis of visual speech, one needs insight into a number of fields, including articulatory speech, visual speech animation techniques and theories of social interaction. This chapter will introduce the key concepts in each of these fields.

2.1 Anatomy of the Human Speech System

To produce speech, the human body tensions the diaphragm to compress the lungs and consequently push air through the larynx and out via the mouth and/or nasal cavity (Figure 2.1). The tension of the vocal cords in the larynx, as well as the mouth shape and tongue position, coupling with the nasal cavity, leads to the production of different sounds as air is expelled. For example, a 'h' sound is produced by forcing air through a tightened glottis and open mouth and an 'f' is generated by forcing air through the lower lip and top front teeth which are in contact with each other [10]. The left hand diagram in Figure 2.1 reveals that most of the anatomical mechanisms of sound production are hidden from view during visual speech. The anatomy of the oral features, visible during speech, are therefore of greater relevance.

The right side of Figure 2.1 illustrates the intricate arrangement of muscles around the mouth, labelled A to G. For example, the orbicularis oris (G) is a complex of muscles that is used to close the lips. Consequently, when reproducing the many visual speech poses that need to be simulated for realistic animation, it is important to note the direction and magnitude of skin region displacement effected by these muscles. This is, however, an extremely complex task, with many levels of abstraction discussed further in Section 3.2, with Section 3.2.3 specifically investigating approaches to oral muscle simulation.

2.2 Phonetic Description of Speech

Each speech sound in the English language can be classified based on the place and manner of its articulation. These units of speech are known as phonemes. Referring back

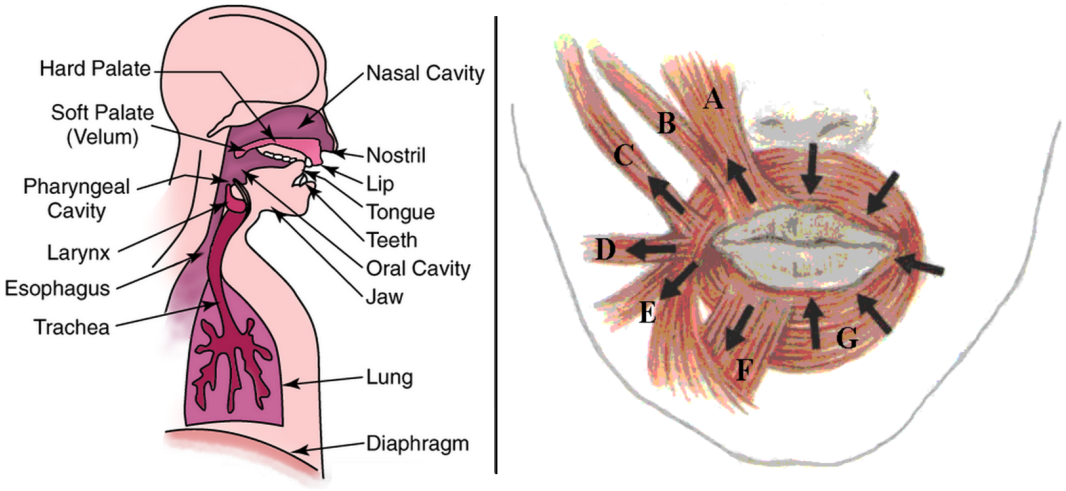


Figure 2.1: Left: Anatomy of the human speech system. Reproduced from [10]. Right: labelled oral muscles and their directions of contraction (A: levator labii superioris, B: zygomaticus minor, C: zygomaticus major, D: risorius, E: depressor anguli oris, F: labii inferioris, G: orbicularis oris). Reproduced from [11].

to the examples used in Section 2.1, a 'h' sound can be described as an unvoiced glottal fricative, while an 'f' is an unvoiced labiodental fricative, as defined by the International Phonetic Alphabet (IPA) [12]. The full IPA chart can be found in Appendix B. Table 2.1 classifies phonemes, into a variety of pronunciation categories. These classifications are important for relating oral cues to audio, as discussed in Section 2.3.

VOICING					
Voiced					Unvoiced
b,d,g,m,n,v,ð,z,ʒ,dʒ					p,t,k,f,θ,s,ʃ,tʃ
MANNER OF ARTICULATION					
Stop		Nasal	Fricative		Affricate
b,p,g,k,d,t		m,n	v,f,ð,θ,z,s,ʒ,ʃ		dʒ,tʃ
PLACE OF ARTICULATION					
Bilabial	Lingua-Velar	Lingua-Alveolar	Lingua-Dental	Lingua-Palatal	Labio-Dental
b,p,m	g,k	d,t,n,s,z	ð,θ	ʒ,ʃ,dʒ,tʃ	v,f

Table 2.1: Classification of audio speech components. Reproduced from [13].

Since its introduction, the IPA has been extended to include sounds from many languages as well as computer-readable encodings. These extensions have lead to the extended speech assessment methods phonetic alphabet (X-SAMPA), which is used in this work.

By using X-SAMPA, our proposed methods can, in principle, be extended to include the diversity of languages in sub-Saharan Africa and beyond.

2.3 Relating Sound to Mouth Shape

The key to good visual speech synthesis is the appropriate reproduction of the relationships between audio and visual speech. In the entertainment industry, VSS has traditionally been achieved by trained animators. Their work involves the painstaking generation of visual speech animation sequences that are aligned with spoken audio recordings. This process can be simplified and made more efficient by the definition of visual speech-phoneme pairings, known as visemes, such as those shown in Figure 2.2. The word viseme is a contraction of the words “visual" and “phoneme" [14].

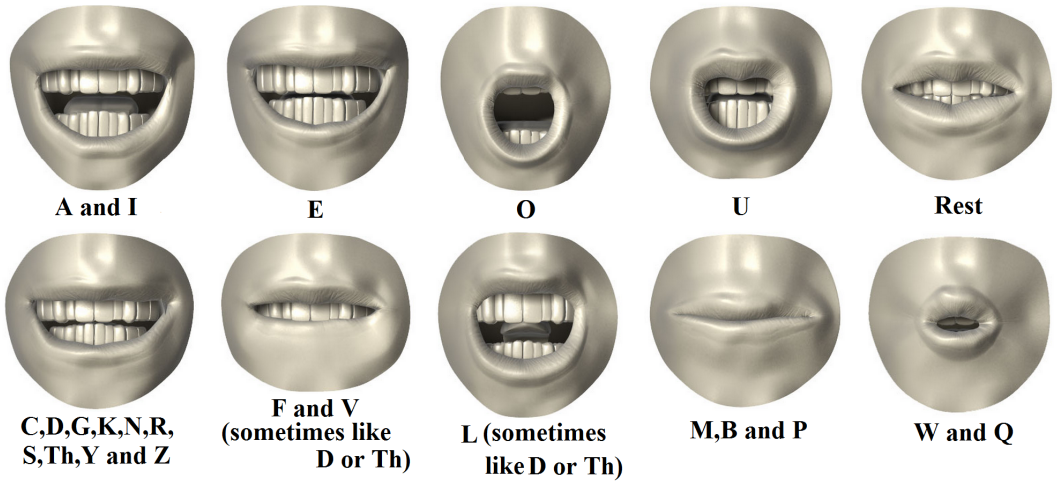


Figure 2.2: Illustrated example of the Preston Blair viseme-phoneme paring series. Reproduced from [15].

Although static viseme-phoneme pairing systems have been proposed, the compilation of a corpus of visemes diverse enough to fully capture the intermediate coarticulatory speech poses would be a difficult and laborious process. The animator’s solution to producing seamless visual speech sequences, known as ’speech cycles’ [4], is to use computer graphic software components. These software options provide tools to efficiently interpolate static visemes and tailor visual speech sequences to the level of realism and intelligibility required of the virtual character (also referred to as an embodied conversational agent or avatar).

The shape of the mouth is influenced by the articulation and inertia of the surrounding units of speech [16]. Therefore, the limitations of static visemes can be attributed to their inability to account for these coarticulatory effects. With this in mind, this work explores models for phoneme-to-viseme clustering for visual speech synthesis taking into account coarticulatory effects.

2.4 Conversational Agent Interaction Theories

Before discussing visual speech animation and synthesis, it is important to have a broad understanding of interaction theory, focusing on the cognitive and psychological underpinnings of virtual characters capable of audiovisual speech.

2.4.1 Confluence of Seeing and Hearing

Visual speech has been shown to increase the intelligibility of auditory information [17]. For example, the “cocktail party effect" has shown that a listener is able to pick up significantly more information during conversation, whilst surrounded by other active speakers, by focusing their attention on the speakers lips [18]. This is particularity useful for those in noisy environments or who are hard of hearing.

Brain imaging research suggests that humans anticipate audio speech based on visual speech [17]. These anticipatory gestures are perceived as natural when observed 100ms to 300ms before the audio [19; 20]. However, there is no known rule for determining by how much the mouth movements should precede the audible speech. The timing between audio and visual cues also varies between different parts of words and between vowels and consonants. Capturing and reproducing these subtleties is key to good a VSS system.

As part of the reproduction of good VSS, mismatched audio and visual speech must be avoided. In some cases, this can lead to an auditory illusion, known as the “McGurk effect"; caused by visual information (from the speaker’s lip movements) conflicting with the audio information, and may result in the perception of a sound matching neither the audio or the visual output [21; 22]. The most common example is the visual cue /ga/ with an auditory /ba/, which usually result in the perception of /da/. The McGurk effect is relevant to the understanding of atypical audio and visual speech on facial animation.

Other basic concepts to consider for clear audiovisual speech perception include clear visibility of the mouth and a natural speech rate. The latter can be difficult to synthesis as speech has many dynamic properties, including accent, stress and context related accentuations. The captured visual speech data must therefore be clearly pronounced/enunciated at a natural speech rate. This data must then be accurately reproduced by the virtual avatar from the captured recordings.

2.4.2 Realism and the Uncanny Valley

For our synthetic system to have human-like realism, it must perfectly imitate a human’s visual speech. The conversation agent’s appearance and motion play a critical role in the perceived quality and emotional response experienced by the observer. Therefore, the study and identification of attributes effecting realism, for example, human-likeness and the naturalness of motion, are important.

The study of humanoid robotics has revealed that the affinity felt towards objects of increasing human-likeness grows, until a point is reached where discerning what is real and what is not becomes difficult, and the onlooker experiences an eerie or uncomfortable sensation. This cognitive dissonance subsides when the observed entity reaches complete naturalness. Figure 2.3 illustrates this effect, referred to as the ’Uncanny Valley’, with the observable dip in affinity [23]. For moving or animated characters, the graph’s hypothetical peaks and valleys are amplified [24].

The Uncanny Valley effect is not as easily reproduced as might be expected from Figure 2.3 [25; 26; 27]. Avoiding it requires an understanding of the interplay between appearance, movement and realism of interaction. When these fall just short of human-likeness, users may find the interaction disturbing [28]. To avoid the phenomenon during virtual character development, the theories below, which reveal the intricate mix of psychological phenomena associated with the Uncanny Valley effect, must be considered.

- The theory of evolutionary aesthetics suggests the avoidance of the Uncanny Valley by making the face appealing and/or attractive [29]. This is, for example, achievable

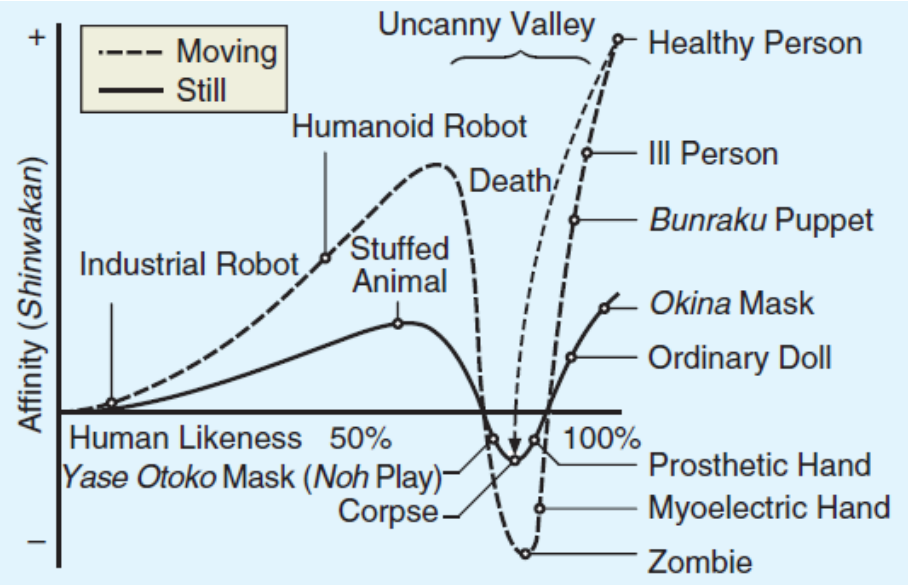


Figure 2.3: Graph illustrating the Uncanny Valley effect with examples (including comparisons to traditional Japanese theatre masks and puppets). Reproduced from [23].

with the use of a symmetrical face with good skin. In line with these requirements, the face's proportions must be within those of human norms, especially for the eyes [30].

- Rozin's theory of disgust is related to theories of disease avoidance [29]. It stresses the tendency for humans to avoid those with poor health indications, namely; bad colouring, jerky movements and dozy looking eyes. The face's skin should therefore be smooth to give the impression of the character being of good health [30].

- The Uncanny Valley effect can also be attributed to paradoxes involving personal and human identity. This occurs when a human's cognitive conceptualization of being, which is brought about when observing a human-like entity, causes the onlooker to grapple with the paradox of dealing with something that is neither human nor machine [31]. MacDorman *et al.* [29] adopt this idea suggesting that eeriness is experienced when the cognitive act of linking quantitative metrics between qualitatively dissimilar subjects (i.e. a human face against a computer generated face) calls into question the original differentiation between the two categories.

- Terror management theories postulate that machines with human characteristics can elicit a fear of death. Human-like machines can evoke thoughts of soullessness, mortality or the fear of being replaced [29]. A simple solution is to give the characters the appearance of a full body that appears natural, lively and preventing the face from having an overall scared or menacing appearance.

- Expectation violation theory suggests that irregular social behaviour can cause an interaction to feel eerie [29]. The solution to this problem, although beyond this work's scope, is to implement realistic, contextually relevant gestures and responses to fulfil the users normal social interaction expectations. Graf *et al.* [32] suggest the addition of head movements to avoid the Uncanny Valley effect, even if they are not related to the speech's context. The authors also recommended that the appearance

of having a full body is important as a 'floating head' is judged by most viewers as eerie.

MacDorman *et al.*'s [30] mention a few design choices to avoid the Uncanny Valley effect. First, is to refrain from using photo-realistic texturing on the character. Additionally, the authors suggest avoidance of inconsistencies in the level of human-likeness in character features. By avoiding near or mixed photo-realistic character components, the need for near-perfect visual speech is prevented. Other attributes to avoid, which are more common to VSS and graphics, include penetration of colliding surfaces or facial tissue and the prevention of unnatural mouth shapes and movements.

2.4.3 Realism in Visual Speech

There is a perceptual relationship between character realism and movement that is evident in visual speech. For example, the ability to correctly interpret poorly animated, non-human-like cartoon character speech demonstrates that an approximate representation of visual speech can be sufficient. As a result, it is justifiable to state that the level of graphic realism can govern the required level of visual speech realism and that the chosen level of character realism sets the minimum standard required for passable visual speech.

According to Osipa [4], the most important aspect of lip syncing is achieving the simple but well timed motions of opening and closing and widening and narrowing of the mouth. By morphing between these various shapes, the observer can get a basic impression that the character is mouthing the words it is saying, as demonstrated in Figure 2.4.

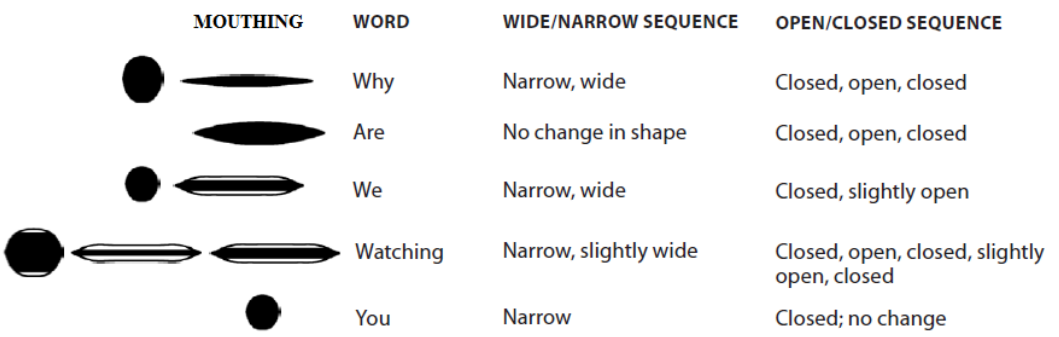


Figure 2.4: Example of a speech cycle's summation of mouth poses, which give the impression of articulated speech, adapted from [4].

Identifying visual speech errors that are acknowledged by animators will help in the VSS systems' assessment. Key attributes to avoid are overly active lips and over enunciation, which can result in disjointed or jumpy looking visual speech [4]. Visual speech is often relaxed, with the mouth only slightly opening and closing, and the tongue playing a minor role.

2.5 Conclusion

This introduction to audiovisual speech highlighted the anatomy of the mouth and the techniques used by artists, leading to a cognisance of speech coherency and interaction theory. It can be concluded that VSS should not rest in identifying a set of defined

visemes, but focus on capturing and incorporating the natural gestures and coarticulation effects of visual speech. A good impression of visual speech can be generated provided the captured mouth shapes are diverse, adaptable and that the VSS system's synthesises speech with the correct timing in relation to the phonemes' utterances, while accounting for coarticulation effects. The appearance of the conversational agent must also avoid the attributes which supposedly elicit the Uncanny Valley effect.

Chapter 3

Facial Animation Techniques

The previous chapter presented a broad background of visual speech and interaction with virtual characters. The focus will now shift toward identifying the most appropriate facial animation (FA) technique for this project.

3.1 Introduction to Computer Animation

From the perspective of a computer animation artist, a scene can be broken down into five procedural elements: forming, surfacing, lighting, motion and rendering. All are considered to be of equal importance for good animation. In the context of this work, however, the focus will be on motion. Forming, surfacing lighting and rendering are all of secondary importance as they are only a means towards creating the virtual character.

As motion is usually the last step addressed in computer animation, the secondary goals will be explained first. To start, forming involves the creation of three dimensional (3D) virtual objects. Forming a 3D object is accomplished with a computer generated mesh; a net-like polygonal geometric surface consisting of edges linked by vertices to create an intricate multi-faced object. A mesh's form represents the outer shape of an object. Surfacing is the process of colouring each of the mesh's multiple surface faces, giving the object the appearance associated with the materials it is made of. For a human-like character's facial mesh, a dynamic colouring scheme is required to represent the expected colour variations of healthy skin. Lighting and rendering are also important as they effect realism. The lighting must sufficiently illuminate the character. Rendering involves the generation of each pixel of the final 2D projection of the 3D scene from the camera's perspective. Section 3.4 addresses the selection and use of the animation software options capable of forming, surfacing, lighting and rendering the virtual character.

Returning to the primary focus of character motion, computer animation techniques used in the movie industry have many parallels with stop-motion animation, where an animator must create a character's poses for every scene. In computer animation, each of these unique poses, known as keyframes, can be interpolated between, thereby automating much of the animation process. Motion, in the context of this work, will, however, be achieved by animating the avatar using the output of the developed VSS system. There are many approaches toward using this output to generate motion in graphics, which will be discussed next in Section 3.2.

3.2 Facial Animation Techniques

The current approaches to FA are laid out in Figure 3.1.

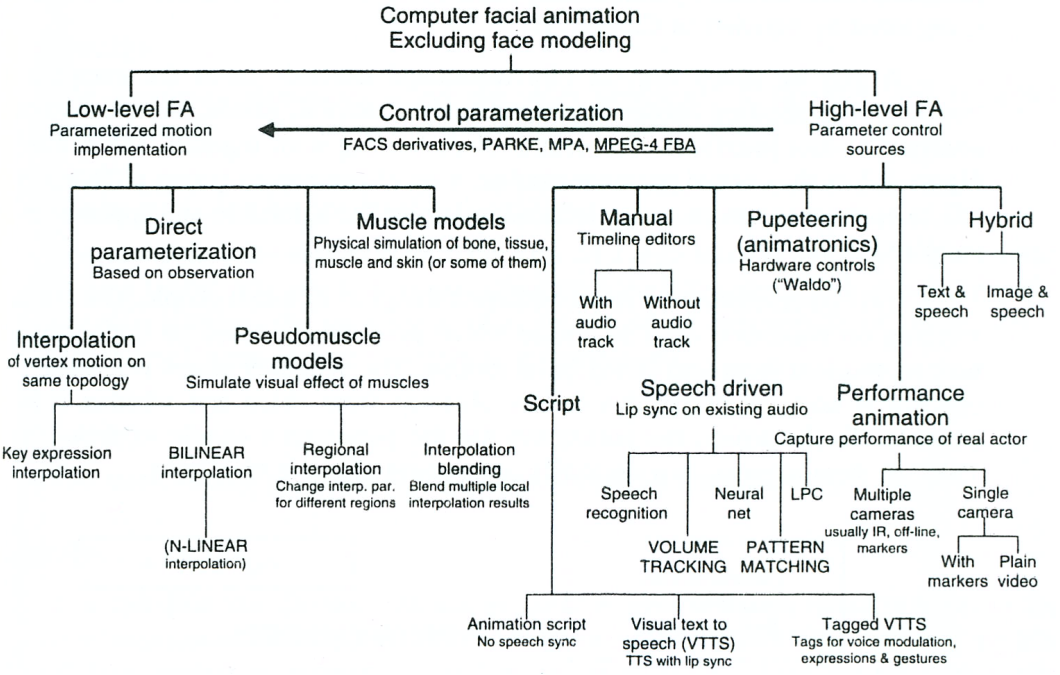


Figure 3.1: Taxonomy of approaches to control computer facial animation. Reproduced from [33].

Low-level FA (the left topmost branch of Figure 3.1) concerns the way in which basic motion is parametrised and achieved. This involves the morphological processes carried out on the 3D object's mesh.

High-level FA concerns the production of whole animation sequences, hence high-level FA can be considered a tier above low-level FA as it governs the large scale motion processes, which are translated into virtual object surface manipulations by low-level FA.

The boundaries between high and low-level FA depend on the control technique used and, in practice, are not necessarily split as clearly as suggested by the figure. The main sub-branches of FA will now be considered, starting from low-level FA on the left to high-level FA on the right of Figure 3.1.

3.2.1 Interpolation

Interpolation translates the geometric representation of an animated character's mesh from one configuration to another over time. For example, linear interpolation allows a key expression of a character's mesh to morph to a subsequent key expression along a straight path (usually using the vertices), within a specified time [9]. As mentioned, these key expressions are commonly referred to as **keyframes**; extreme positionings of a character at a particular frame number in an animated sequence [34]. Keyframe morphological tools commonly available in animation software include linear, polynomial and spline interpolation as well as Bézier curves. Bézier curves are popular for generating curvaceous regional interpolations, allowing for more flexible and thus more natural pose transition rates.

Interpolation styles are important because they must smoothly link the VSS system's outputs, which are going to be in the form of time-positional data segments. It is recommended that the influences of the interpolation style used joining the visual speech segments should be experimented with to consider their affects on speech realism.

3.2.2 Direct Parametrization

Direct parametrization attempts to surpass the limitations of interpolation methods [35]. In direct parametrization, a set of parameters are defined, for example the distance between two vertices of a mesh. By altering the parameter's value, the mesh is deformed using basic geometric transformations, including scaling, rotation, translation and interpolation. This technique allows a variety of dynamic shapes to be controlled by modulating the appropriate set of parameters.

Although direct parametrization allows for efficient computation, setting up these parameters is very time consuming for geometries composed of many vertices. It can also lead to unnatural results if, for example, an individual or mixture of parameter changes excessively exaggerate a facial mesh deformation, thereby causing the virtual model to assume an unnatural pose. If direct parametrization is implemented, it must be done so with great caution so to avoid unnatural facial poses or movements, which could cause the cognitive discomforts related to the Uncanny Valley effect.

3.2.3 Pseudomuscle and Muscle Simulation Models

Pseudomuscle models use parameters, which are related to mesh deformations, to emulate muscle movements [36]. Pseudomuscle visual speech models attempt to replicate the magnitude and direction of skin movements induced by bones, tissue and muscles, requiring far more detailed knowledge of human anatomy than that presented in Section 2.1.

Physics-based muscle models often simulate muscle and skin layers as interacting mass and spring structures, exhibiting near-anatomic correctness [37; 38]. Simpler muscle simulation models create facial actions by modelling more generalised mesh topology transformations which are not limited by a set number of parameters or by the facial topology [39].

The complex development of physics-based muscle simulations pushes it beyond the scope of this project. However, emulating facial movements using a set of pseudomuscle-like parameters is a viable option, depending on the mechanisms driving the FA. This will be revisited in Sections 5.1 and 3.4.

3.2.4 Script Based Animation

High-level script-based FA is driven by a program that relates text-based dialogues to animations. These algorithms work with text input or orthographic transcriptions of recorded speech to produce visual speech synchronised with audio. Tagged facial-animation-from-text systems, such as that described by Albrecht *et al.* [40], include expressive tags in their text input, as well as text-to-speech (TTS) engines, for generating synchronised visual and audio speech-related expressions during VSS.

Script based speech animation algorithms are advantageous over other high-level FA techniques because they allow for accurate audio to visual speech stochastic modelling. This is because the algorithms are based on accurate transcriptions of audio speech, resulting in better audio to visual speech predictive models. Chapter 4 explores script-based animation techniques and algorithms for VSS.

3.2.5 Manual Animation

Manual animation is the working realm of artists. As mentioned in Section 3.1, animators go through the time consuming process of creating a character's form, surfacing, lighting, animating motion and lastly rendering. These processes require an excellent technical knowledge of graphics and the graphics software. To reduce the need for developing this technical knowledge, Sections 3.4 and 5.1 identify processes for automating dynamic virtual character formation, surfacing and animation. This will allow the developed VSS pipeline to be implemented in under-resourced environments, where the technical skills and tools required for manual animation are not available.

3.2.6 Speech Driven Animation

Speech-driven animation attempts to match audio speech with lip motions by identifying patterns that can be used to map audio signals to their naturally occurring visual speech counterparts [33]. In contrast to script-based animation, no orthographic or phonetic transcript is required. Stochastic modelling approaches are generally used, which can include volume tracking, artificial neural networks and linear predictive coding (LPC).

3.2.7 Pupeteering

Puppeteering, animatronics and humanoid robotics refer to the use of electronically controlled hardware to represent a human-like form. Figure 3.2 demonstrates three implementations of back projected robot heads, created using a combination of both electromechanical and graphical systems. These mixed medium systems also claim to improve engagement and speech intelligibility [41; 42].

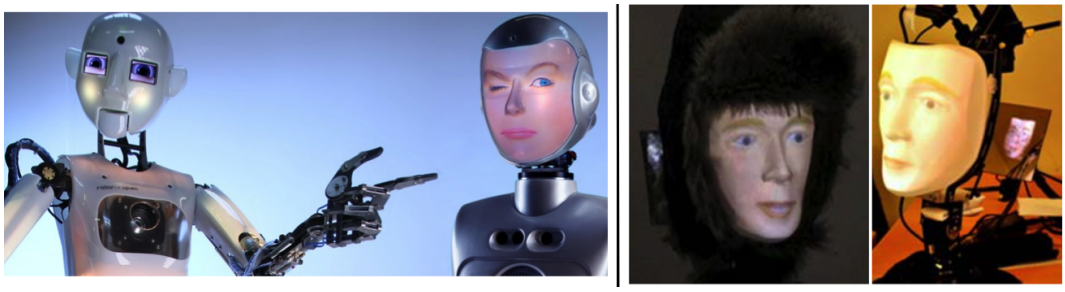


Figure 3.2: Examples of mixed machine and computer graphic interaction technologies. Left image: RoboThespian (left) and SociBot (right). Centre and right images: Furhat. Reproduced from [41] and [42] respectively.

Research using mixed electromechanical and graphical robots has shed light on a relevant hypothesis which relates forward facing two-dimensional head positioning to intelligibility. This hypothesis is based on the Mona Lisa effect, which describes the fixated gaze experienced by an observer of a forward looking facial image. Al Moubayed *et al.* [42] attribute the Mona Lisa effect to improving a user's intelligibility of forward facing screen-based conversational agents', specifically when observed from an angle. This bodes well for the proposed screen-based avatar, which should face directly toward the camera, thereby reproducing the supposed benefits associated with the Mona Lisa effect.

3.2.8 Performance Animation and Control Parametrization

Performance animation employs motion data captured from a human actor, through the use of motion tracking software, to drive virtual characters. The adopted motion capture technique will largely influence the digital representation of visual speech. In turn, this will influence the control parametrization technique appropriate for visual speech animation as well as the VSS's audio-to-visual speech mapping algorithms. The interplay of these components highlights the importance of correctly selecting animation software with a suitable set of control parametrization tools. Section 5.1 delves further into motion capture and performance animation techniques relevant to the selected FA system.

3.2.9 Hybrid Animation Techniques

Conversational agents using hybrid animation systems explore the combined synthesis of visual and audio speech synthesis. While audio synthesis is out of the project's scope, the processes used to map audio signals to visual speech images is useful, and will be considered in Chapter 4.

3.3 Formalised Facial Parametrisation

The primary advantages of formalising facial parametrisation is to allow more comparable systems and to allow easier collaboration in research.

Park [43] created the first virtual talking head. He did so by experimenting with polygons painted onto a face, which he used as a basis to guide virtual face mesh topologies. Park applied direct parametrization to allow the mesh to morph into expressions. Though successful, this technique was never formalised. Only later, in the pioneering work of Ekman *et al.* [44], were means identified to recognise the emotions surprise, fear, anger, disgust, sadness and happiness. This led to the development of the Facial Action Coding System (FACS), which separated observable components of facial muscle movements called Action Units (AUs) [45]. Eventually this influenced the development of the computer animation MPEG-4 parametrised control standard [33]. The MPEG-4 standard uses Face Animation Parameters (FAPs), marked in Figure 3.3, and Body Animation Parameters (BAPs). Both FAPs and BAPs are derived from geometric parametrization of the face and body.

There are a number of disadvantages of using pre-defined standards, like MPEG-4. Firstly, the motion capture technique needs to match a specific set of facial parameters, therefore requiring a specific animation model and control system to be used. This can be time consuming to reproduce and limit the flexibility of the animation and control techniques. Standardised parameters may also limit new insight into articulatory gestures. Therefore, this work has focused on meeting the accessibility and reproducibility issues mentioned in Section 1.1. This does not mean to say that selection of an optimal set of parameters for facial control was neglected. It is discussed further in Section 5.1.

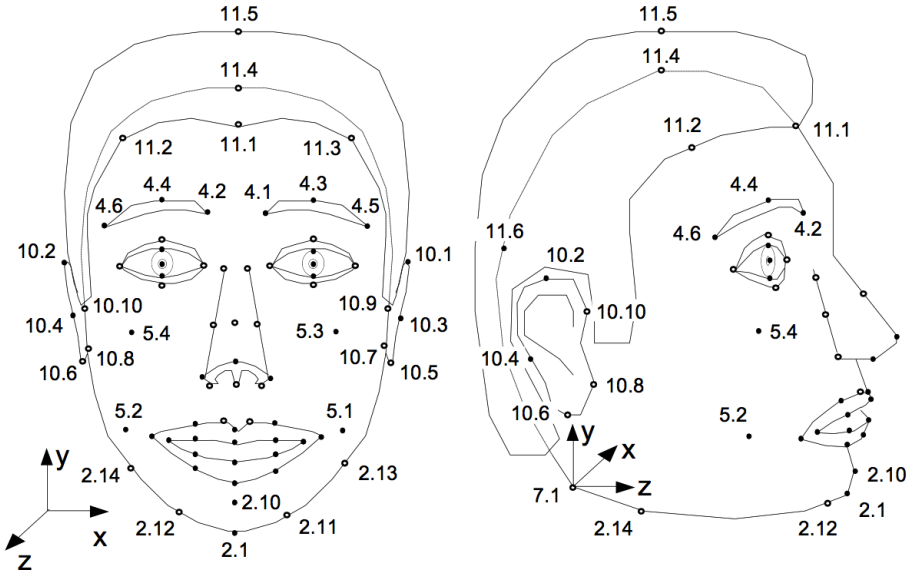


Figure 3.3: Face Animation Parameters (FAPs). Reproduced from [33].

3.4 Animation Software

3.4.1 Software Selection

For a preferable FA technique to be identified, it is necessary to asses the suitability of the development environments in which such a system will be created. Three such environments are considered here.

The most basic type of development environment is a graphic rendering application programming interface. This provides for efficient coding of every aspect of the virtual character and its control parameters. An example of such a development environment is OpenGL [46]. By implementing specially-written software, the animation control system would be optimised for usage with the virtual agent. The disadvantages of this approach would be the time consuming programming required to develop the control and rendering systems and the resultant complexity limitations imposed. Both these disadvantages may also negatively effect those trying to improve this system at a later stage. Character formation would also be required, a complex task ill-suited to those unfamiliar with 3D animation.

Environments with pre-determined character control systems are available. Freely available options include agents with control parameters, for example: Xface [47], RUTH: Rutgers University Talking Head [48], GRETA (a real-time 3D embodied conversational agent) [49] or USCICT's Virtual Human Toolkit [50]. These tools are freely available, save time generating a character and are already equipped with a control system. Editing the input and control systems is, however, either not possible or difficult to reverse engineer. Without the needed flexibility for developing the input and control systems, these tools limit their usability for VSS experimentation.

Animation and game engine software are an intermediate option between the first 'build-all' and second 'already-made' character development environments. These software options provide customisable high-level FA and pre-defined low-level FA control parameters. Examples include Maya [51], the Unity game engine [52] and Blender [9]. These packages are cost effective since they are often free for research purposes, and in Blender's case, open source. They are also highly accessible, have online software support

resources and are compatible with multiple operating systems. The high and low-level FA systems are appropriate for the types of graphic manipulations required for this work. This is because the means to control the complex low-level FA systems have already been established. the high-level FA can be controlled using script-based animation, greatly simplifying the requirements for a VSS control system.

In practice, the selection process was iterative. However, the advantages of game engines, specifically Blender's game engine (BGE) [9], made it the obvious choice for the character control and animation development environment. The use of graphic application programming interfaces would take too long to develop and software options with pre-built agents were found to be too limited in animation and control diversity.

Blender is a feature rich open source 3D animation software package. It has minimal computational requirements and is compatible with several major operating systems. Its integrated game engine control parametrization tools have the major advantage of combining easy to use game logic controllers (elaborated on in Section 3.4.2) with Python scripts to control events in a virtual scene. This is an accessible programming language, particularly when it comes to availability and diversity of its libraries. Although scripted control of features is not unique to the BGE, it is the most powerful freely-available option and is well documented with active online communities. Furthermore, Blender 2.70 includes integration with the NumPy Python library [53], which provides access to advanced numeric operations useful for character control.

A secondary advantage to using the BGE was its compatibility with MakeHuman [8], an open source 3D customisable character generation software. Forming, surfacing and rigging (defined in Section 3.4.2) are major obstacles to overcome in creating a virtual character. By using MakeHuman, our VSS system could avoid the need for a background understanding of computer game character development.

MakeHuman provides over 1000 customisable character topologies and surfacing features for the body and face and is compatible with major operating systems. The software also automates the creation of a consistent and extensive set of animation tools, which accompany the character when imported into Blender. These animation tools include a large selection of oral gestures with the potential to collectively represent any visual speech unit.

3.4.2 Animation Tool Selection

The characters created by the MakeHuman software are automatically paired with two forms of animation tool, namely bone-driven animation and shape key animation. First an introduction to these tools will be given, followed by a motivation and method for selecting the latter as the character's VSS animation control system.

Once a character is formed, and before animation can begin, a binding and control structure, commonly known as an **armature**, is created through a process called rigging [4]. For human-like characters, their armature often resembles a skeleton-like system that consists of kinematic chains called bones, with which the character can be animated. The function of the bones in a rigged character has been described as "digital orthopaedics", because bones manipulate the areas of the character's mesh to which they are bound, in a way that is reminiscent of how human bones manipulate skin [54]. Figure 3.4 illustrates bones, seen as linked grey octahedrons in the left image and straight line-segments in the central and right hand images. The colour variation, seen in the right hand image in Figure 3.4, reveals the extent to which the torso bone is associated (or weighted) to the chest area of the character's mesh. By this mechanism, the torso area

is moved by angling the chest bone relative to the hip bone, giving rise to a bend which is similar to a human's back movement. This is referred to as bone-driven animation.

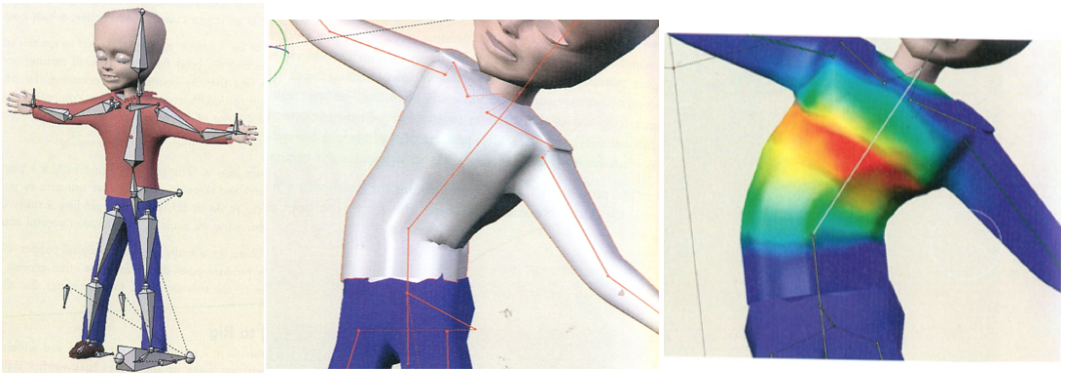


Figure 3.4: A rigged character with bones (left) with highlighted example of a bone weighted to the character's mesh (right). Reproduced from [54].

Shape keys (also known as morph targets or blend shapes) are another commonly available animation tool. They remove the need for individual bones to be repetitively moved to produce frequent gestures. Animators use shape keys to create a library of character mesh deformations with which to speed up the animation process [54].

A shape key is created by saving a deformation of the character's mesh, usually relative to its neutral state. It is then possible to use the software to interpolate between the neutral and fully-formed poses [54]. Figure 3.5 demonstrates a shape key, which interpolates between a neutral base position (left) and an extreme pose (right) in which the character's mouth is closed. Shape key animation has many parallels with a system which couples a low-level FA direct paramaterization system with a high-level FA pseudomuscle model (discussed in Sections 3.2.2 and 3.2.3 respectively).

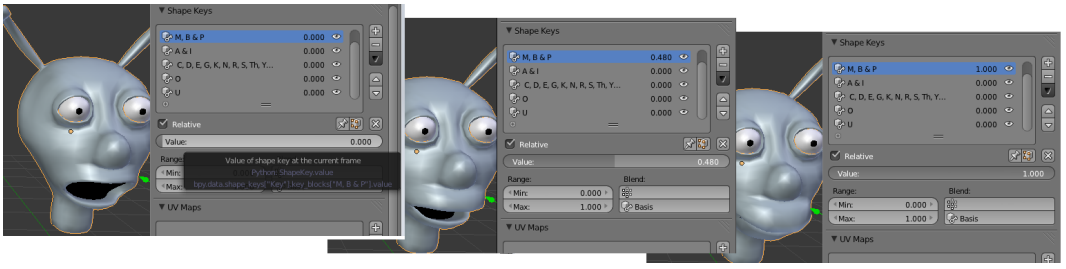


Figure 3.5: Example of a shape key that characterises the closing of the characters mouth to form a B,M,P viseme, adopted from [55].

Revisiting the character imported from MakeHuman, either bone-driven animation or shape key animation could be used to animate the character in the BGE. Implementing bone-driven animation is, however, unnecessary. This is because the imported MakeHuman character already has a large selection of oral shape keys, which have the potential to collectively represent any visual speech unit.

For game engine control, it was found that the interaction of multiple bone-driven animations becomes cumbersome when trying to achieve the mesh deformations necessary to mimic the subtle facial movements required for speech. This is because bones manipulate only the areas of the mesh they are associated with. Combining multiple bone movements

can result in unnatural mesh deformations which are difficult to detect and prevent. The use of shape keys can overcome this problem. This is because shape keys are limited to a specific manipulation of the mesh. The interaction of the extremes of these manipulations is also better coordinated during shape key formation.

The use of shape key animations as the character's VSS animation control system presented two problems. First, shape keys cannot be accessed directly through Python scripts in the BGE. Second, an oral feature motion capture system must be created, which must also consider the selection of facial markers and their related shape key animations.

Solving the first problem was crucial because, without the BGE Python scripts having access to the MakeHuman character's shape keys, the automation of the visual speech animations could not be realised. This problem was solved using a non-trivial and undocumented script-based animation process, referred to as **scripted bone-driven shape key animation**. This uses bones as a proxy to control shape keys, and is achieved with the use of a script. Note, that in the context of this work, a script is a short Python program is written to automate the task of bone-driven shape key animation.

Appendix A provided a detailed explanation of how scripted bone-driven shape key animation is be achieved. Figure 3.6 shows one of the MakeHuman character's bone-driven shape key animations, which controls mouth opening.

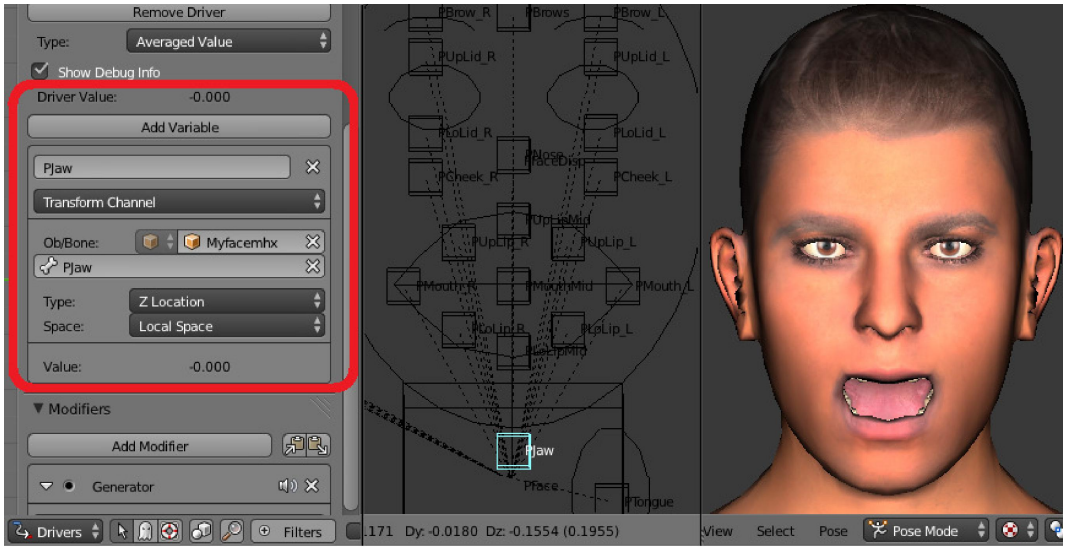


Figure 3.6: Example of bone-driven shape key animation controlling the MakeHuman character's 'open jaw' shape key animation.

The second problem, namely the development of the oral feature motion capture system, is discussed in Section 5.1. The key point to understand here is that a custom motion capture system was developed to track oral features. The MakeHuman character's oral bone-driven shape key gestures are then animated using a Python script, which is given access to the tracked facial features. These facial marker placements can be seen in Figure 5.2 and their relationship with the MakeHuman character's oral shape keys is discussed in Section 5.1.2.

3.5 Conclusion

This chapter has surveyed current techniques of FA. Drawing from these insights, a suitable open source tool for animation control was identified. The motion capture system compatible with the characters scripted bone-driven shape key animations is also mentioned. With the character's facial animation control system established, different approaches to VSS can be addressed.

Chapter 4

Visual Speech Synthesis Overview

Chapter 3 explored animation techniques with a special focus on the character's visual speech control system, which used scripted bone-driven shape key animations. Our attention will now turn to visual speech modelling and synthesis techniques.

According to Chen *et. al* [56], VSS has two common approaches. The first approach uses audio-to-visual mapping rules that are often derived from observations of utterances, which lead to gestural speech models. The second approach uses functional approximations, generating probabilistic visual speech models which emulate the relationship between the sounds produced and their corresponding oral features or mouth shapes. This chapter will identify which of these VSS models is suited to this work's objectives.

4.1 Gestural Speech Models

In the context of this work, a gestural speech model implies a VSS system which uses predefined deterministic models. These models commonly employ parametrised gestural dominances to static visemes in an attempt to reproduce natural coarticulation effects.

4.1.1 Rule-Based Visual Speech Models

Rule-based VSS attempts to create a theoretical representation of visual speech which mimics the process of coarticulation. Parke's [43] pioneering work, referred to in Section 3.3, is considered a rule-based VSS model because it uses a simple cosine interpolation scheme to compute the intermediate frames between predefined static visemes. These simple rules, in combination with a parametrically controlled polygon 3D face topology, allowed both visual speech and expressions to be interpolated [57]. Parke estimated the control parameter values by recording and studying his own visual speech. This simple interpolative method is ill suited to automated visual speech as it requires parameters to be manually adjusted to suit each speech sequence.

Cohen *et al.* [58] used Park's parametrized facial animation model to create a more complex rule-based VSS system, derived from Löfqvist's [59] theories on visual speech gestural patterns, coherency and aggregation, as well as observations of oral feature dominance over time. The model assigns a dominance to each vocal articulator, which, over time, increases and then decreases in priority of physical articulation over the surrounding speech segments. The top right of Figure 4.1 illustrates the dominance of two vowels in a VCV sequence. Articulatory dominance is modelled by decay functions, like Equation 4.1.1, which differ in time offset, duration, and magnitude.

The bottom right of Figure 4.1 illustrates the physical displacement of a vocal articulatory feature relative to its neutral/rest position. This continuous trajectory is synthesised based on the overlapping dominance speech parameter functions in the top right graph of Figure 4.1. In other words, the temporal weightings determine the visual importance of an articulatory feature's displacement at each instance in time [58]. On the left of Figure 4.1 is an example set of dominance functions for all vocal articulators for a single speech segment.

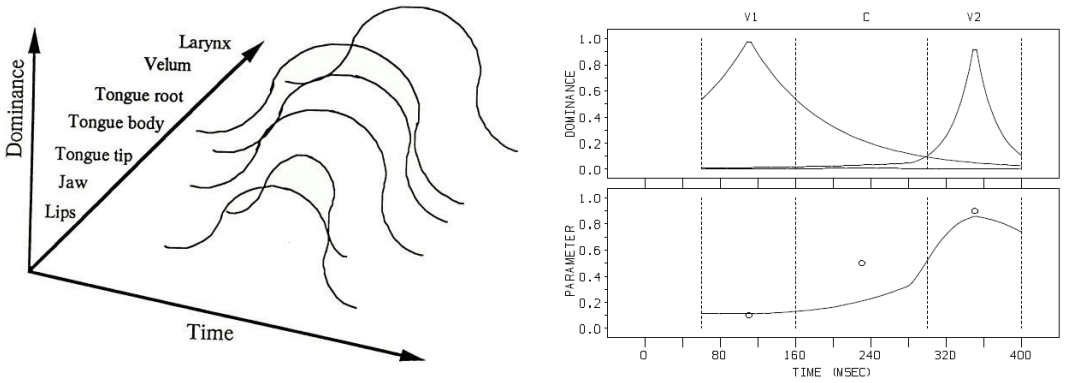


Figure 4.1: The left graph represents the dominance of articulatory features over time for a typical speech segment (as adopted by [58] from [59]). The right reveals synthesised plots representing time-dominance (top right) and time-resultant displacement (bottom right) values for a VCV word. The circles represent usual static vowel or consonant viseme displacement values, known as target control parameters [58].

The dominance speech parameter function, D_{sp}, developed by Cohen *et al.* [58] is expressed by Equation 4.1.1. Here, α_{sp} affects the magnitude of dominance, θ_{sp} represents the increment rate before or after the dominance peak point and the power, c, modifies the rate parameter θ. The magnitude of τ is the amount of time before or after the target control parameter.

$$D_{sp} = \alpha_{sp} e^{-\theta_{sp}|\tau|^c} \tag{4.1.1}$$

Cohen *et al.* do not provide a 'correct' set of values for the parameters in Equation 4.1.1. Instead, they subjectively discuss their use of various theories on speech patterns, like those related to Löfqvist's works, which influenced their chosen values.

Pelachaud [60] extended Cohen *et at.*'s dominance model by adapting the rule-based VSS technique to automatically tune the parameters of a dominance functions to synthesise VCV sequences. This used a neural network to minimise the difference between a feature's synthesised VCV parameters and its measured parameters. The neural network used radial basis functions to determine the λ_j and σ_j parameters of the dominance function, represented by f_i in Equation 4.1.2. This involves using quasi-Newton algorithms for unconstrained non-linear optimization. Equation 4.1.2 represents the dominance of a visual speech feature's parameter as a summation of nine target control parameters per VCV utterance.

$$f_i(t) = \sum_{j=1}^{9} \lambda_j e^{-\frac{|t-time(t_j)|^2}{\sigma_j^2}} \tag{4.1.2}$$

Figure 4.2 illustrates a synthesised VCV trajectory, "/*apa*/", whose dominance function values have been automatically determined.

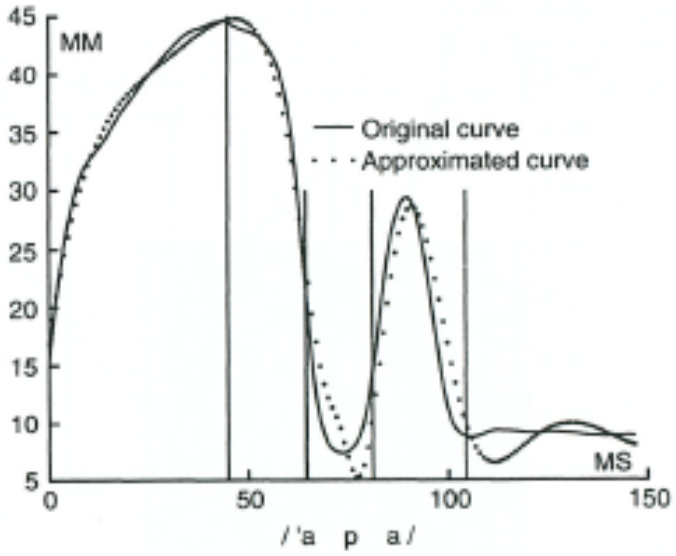


Figure 4.2: Lip trajectories for the measured */a p a/* non-sense word and the RBF's approximated trajectory. The vertical lines represent silence-VCV-silence phonetic segmentation. Reproduced from [33; 60].

Pelachaud's model is limited to VCV segments, and requires the dominance function's parameters to be tailored to the particular VCV sequence for accurate synthesis. These limitations make dominance functions unsuitable for continuous visual speech synthesis.

Caldognetto *et al.* [61] extended the dominance functions presented by Cohen *et al.* with the addition of temporal resistance functions and shape functions. The temporal resistance functions consider the rate at which each segment is uttered, altering the dominance of a segment accordingly. The benefit of this temporal awareness is its ability to vary the physical displacement of vocal articulators for fast or slow speech rates. The shape function synthesises distinctive feature trajectories, which may otherwise not a be achievable by considering dominance functions alone. In both functions, an automatic optimization algorithm is used to estimate the parameters. This is based on a least squared minimization of the error between measured data and modelled trajectories. This VSS approach is again limited to VCV sequences, and also requires manual correction if the error minimization algorithm produces undesired results.

Rule-based VSS models using phonetic tokens to define lip movements are a potentially viable method for synthesis, however, they have a number of limitations. For example, because rule-based model's relationships are based on pre-defined phoneme-to-viseme mappings, the visemes and model parameters must be defined separately. This is a counter-intuitive approach because the model's parameters and viseme corpus cannot effectively reproduce coarticulation if considered independently. This is because optimizing the synthesis algorithm to assign a multitude of dominance rankings to speech tokens in varying contextual situations becomes cumbersome as the viseme corpus expands. Using pre-defined visemes in VSS also ignores the difficult problems of clustering visemes and relating them to phonemes. Bailly *et al.* [62] and Kent *et al.* [63] conclude that the only way to improve models based on a finite set of pre-defined visemes is to increase the

complexity of the rule-based control system as well as the number of allophonic variations available to the synthesis model.

4.1.2 Anticipatory and Preservatory Models

Another notable class of rule-based VSS models uses numerical models to determine the onset of vowel or consonant visual speech units [59; 64; 65; 66]. Two examples of these models consider forward anticipatory and backward preservatory coarticulation effects. Forward or anticipatory coarticulation is a form of high-level articulatory planning that affects future speech segments on the basis of current articulatory gestures. Backward, carry-over or preservatory coarticulation accounts for the inertia in the biomechanical structures of the vocal tract, which cause current articulatory gestures to be affected by earlier speech segments [67].

Cohen *et al.* [58] identified numerical models for coarticulation, resulting in what they refers to as time-locked, look-ahead and hybrid-models. Time-locked, or co-production models, trigger the onset of the lip's protrusion at a fixed time before the vowels' articulation. Look-ahead models start lip movements as soon as possible, following an un-protruded vowel. Therefore, the second vowel's onset differs depending on the number of intervening consonants [64]. Systems that use a mixture of the look-ahead and time-locked models are termed hybrid models. Figure 4.3 illustrates the onset timing of these models, plotting lip protrusion in typical VCV and VCCV segments.

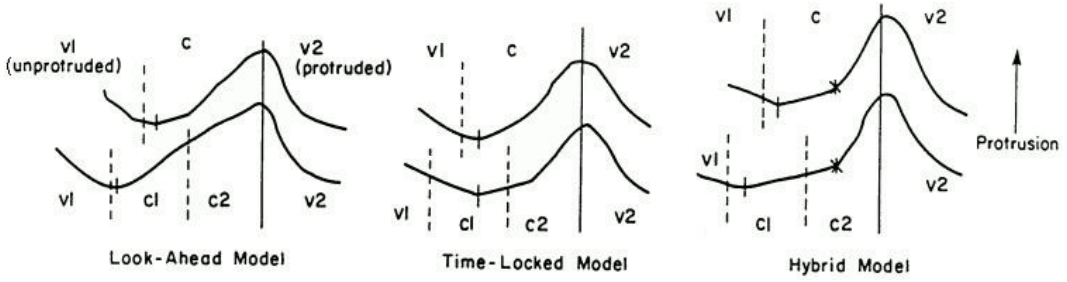


Figure 4.3: Graphs representing lip protrusion trajectories synthesised by the look-ahead, time-locked and hybrid models, for VCV (represented by the top lines) and VCCV (represented by the bottom lines) nonsense words. Reproduced from [58]. Note, that for the hybrid model, the rule phase transition is marked with an 'X'.

According to Cohen *et al.*, the look-ahead model is similar to that of Pelachaud *et al.*'s [60] dominance model, as described in Section 4.1.1. In the dominance model, vowel phonemes are assigned a deformation ranking, meaning that their associated facial gestures are influenced by their intermediate context.

These alternative rule-based models require either hand calibration or semi-automatic parameter optimization methods to tune their associated functions. These constraints are, however, similar to those of the dominance functions described in the previous section. These coarticulation rules are not dynamic enough to be trained to synthesise a diverse range of speech for realistic VSS. Combining these approaches is also not necessarily conducive to a better VSS model as the models will become less intuitive and more cumbersome to calibrate. A quantitative analysis of visual speech highlights its apparent asynchronous relationship with acoustic speech. This relationship can modelled in a more reliable manner using stochastic methods, discussed in Section 4.2.

4.2 Probabilistic Visual Speech Models

From a statistical point of view, a phoneme-to-viseme mapping can be expressed by a variety of probability models whose parameters are determined by observations of visemes under different phonetic contexts [68]. These include vector quantization models, hidden Markov models, neural networks, Gaussian mixture models and decision trees.

4.2.1 Vector Quantization Models

One of the earlier approaches to probabilistic VSS used vector quantization (VQ) to map feature vectors extracted from audio speech to lip parameters [69]. A VQ codebook was trained with the Linde–Buzo–Gray algorithm [70], providing a method of selecting visual speech parameters during VSS. VQ was demonstrated to improve on rule-based VSS. However, although VQ is easily realized and computationally efficient, it is prone to inaccuracies and discontinuities, due to the modelling limitations of the codebook [71; 72].

Chen *et al.* [73] used VQ to map audio speech to visual speech parameters. First, VQ was used to classify the acoustic signals data into classes. These were then mapped, using an averaged acoustic classes' visual code word, to a corresponding visual speech parameter's centroid. Centroids were found by averaging a the parameters of a visual speech unit. However, errors resulting for the use of visual speech centroids made this approach unsuitable for realistic VSS. Creating distinct visual speech parameters also led to discontinuity in the VSS. These artefacts were found to be particularly problematic if the audio contained noise or echoes.

4.2.2 Hidden Markov Models

When using hidden Markov models (HMMs), the Viterbi algorithm can be used to predict state sequences base on observed events [68]. An example of an early HMM-based VSS system, proposed by Yamamoto *et al.*, mapped acoustic speech signals to lip parameters, determined by forced Viterbi alignment. In this work, the audio and visual speech recordings were first parametrised to create sequences of phonemes and 3D lip parameters. Phonetic HMMs are then trained using a look-up table of lip parameters, thereby associating each acoustic HMM state with a corresponding visual speech parameter. This look-up table is trained using Viterbi alignment. A HMM-based system with context dependent lip parameters was shown to outperform a HMM with context independent lip parameters, as well as a baseline VQ VSS model [74].

As in the case of VQ-based VSS, Yamamoto *et al.* [74] also demonstrated that limiting the number of lip parameters in the look-up table dramatically effects the continuity of the VSS. Such discontinuities are compounded by incorrect audio-visual parameter alignments. This HMM VSS method can also be applied to synthesis from text, provided a text-to-phonetic transcription application is available [74; 75; 76].

Tamura *et al.* [77] proposed a HMM-based text-to-audio-visual speech synthesis system which modelled audio and visual features in Japanese. Their system simultaneously synthesised audio speech with synchronized lip movements. Speech Mel-Cepstral coefficients and the corresponding mouth lip parameters are used as the audio and visual parameters respectively, as seen in Figure 4.4. The training phase divided the observation sequences into auditory and visual parameter streams. For the triphone-based HMMs, decision tree based clustering was used to share the audio and visual states, as they are known to be influenced by different contextually factors. Decision trees also had the advantageous

capability of synthesising states that were not observed in the training data, merely by descending toward features that were closely associated. Training the HMM models on both single phone syllables and triphones, it was found that the latter provided better VSS results.

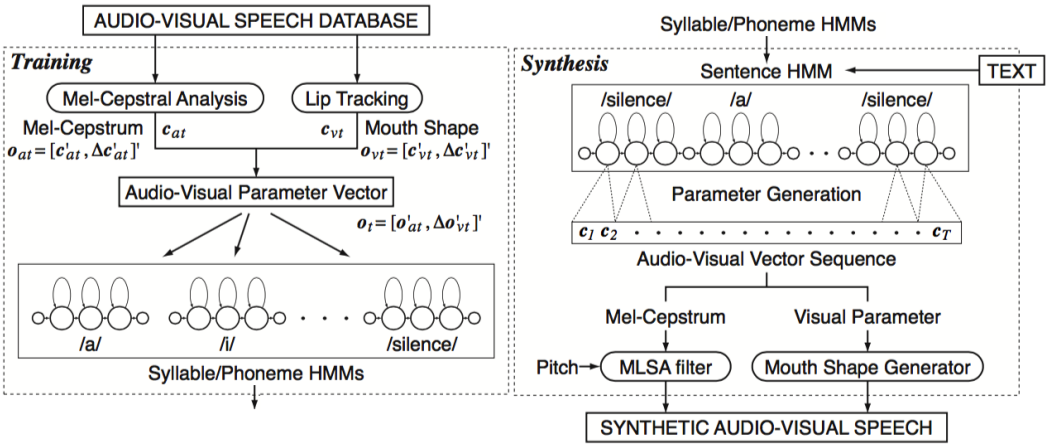


Figure 4.4: HMM-based text-to-audio-visual speech synthesis system. Reproduced from [77].

Hofer *et al.* [78] more recently developed a two-step parameter generation "trajectory HMM" to map audio signals to oral feature parameters. In the recognition step, the most likely viseme unit sequence is chosen given an audio speech signal. During the subsequent synthesis step, the viseme unit sequence is translated into motion trajectories. The parameters generated by the motion trajectory HMMs include first and second order derivatives in order to synthesise smoother trajectories. This technique is similar to Kalman smoothing, or regularisation, in that they also employ continuity constraints. Hofer *et al.* considered only mouth opening and width as lip parameters. A disadvantage of the method is that it required a relativity large dataset containing 500 annotated sentences selected for optimal phoneme balance.

In conclusion, audio-to-visual and text-to-visual mapping using HMMs is popular and has been well explored [56; 73; 79; 80; 81; 82]. However, HMMs are fundamentally limited by their discretization of visual speech parameters. The HMM systems quantise visual speech parameters into set values or levels, which makes the resulting VSS trajectories prone to discontinuity. Addressing this by means of trajectory smoothing algorithms has so far met with limited success. HMM-based VSS also requires large training sets, particularly if higher resolutions of quantisation are used. Therefore, HMM's requirement for large data sets and its limitations on audio and visual parameters suggest that other stochastic approaches should be considered.

4.2.3 Artificial Neural Networks

An artificial neural network (ANN) consists of multiple processing units, known as neurons, which are connected in a network. Each connection has a direction and a weighting. These weightings can be either excitatory or inhibitory and are 'learnt' by applying a propagation function to the ANN. In this way, ANNs automate the creation of a relationship between the input and output neurons [83].

Morishima *et al.* [84] proposed an ANN-based voice-to-image conversion system, trained using backpropagation. Their ANN consisted of three-layers, sixteen input units and eight output units. The inputs were based on LPC cepstrum parameters extracted from speech audio and the outputs were visual speech parameters. The authors also describe a text-to-image conversion model as well as a second voice-to-image conversion model base on a VQ codebook. These approaches were used as a baseline to asses the performance of their ANN-based voice-to-image conversion system. Figure 4.5 shows the various VSS models. Note that the text-to-image conversion model used simple rule-based VSS, with a pre-determined phoneme-to-mouth shape conversion system and seventeen mouth-shape categories. The VQ codebook, for voice-to-image conversion applied the methods described in Section 4.2.1 to relate audio parameters to image parameters.

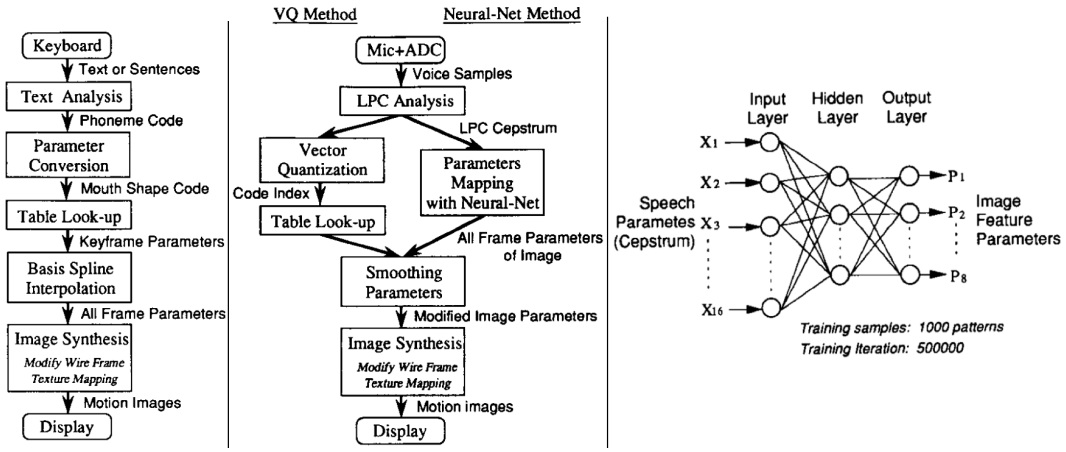


Figure 4.5: Left: rule-based text-to-image conversion model. Middle left: VQ-based speech-to-image conversion model. Middle right: ANN-based speech-to-image conversion model. Right: Illustration of the ANN used for audio-to-visual speech synthesis. Reproduced from [84].

Testing concluded that the ANNs' ability to synthesise continuous values made is superior to both the VQ and rule-based models [84].

Öhman *et al.* [85] also trained an ANN, similar to that of Morishima *et al.*, to directly map acoustic signal parameters to visual speech parameters. They used the NICO toolkit [86] to create a three-layer ANN with thirteen units in the input layer, fifty in the hidden layers and eight in the output layer.

Testing compared the ANN to the rule-based phoneme-to-viseme VSS system developed by Beskow [65], with the addition of a HMM to classify the audio signal to linguistic units. ANN-based VSS was found to be perceived correctly more often than rule-based VSS. Öhman *et al.* also state that an advantage of direct mapping is that it avoids intermediate classification errors resulting from errors in the HMMs audio to text or phonetic transcriptions. Rule-based VSS was also said to make less accurate visual speech predictions than ANN's, which used continuous visual speech parameters to its advantage. Although better than rule-based VSS, ANN-based direct audio-to-visual parameter VSS evaluations were commonly found to yield low intelligibility ratings [85].

Like Öhman *et al.* [85], Agelfors *et al.* [1] used the NICO toolkit [86] to train three ANNs with the same topology but for different speakers. In this work, it was found that a rule-based system with a HMM-based transcription model synthesised more intelligible speech than the ANN-based system.

Chen *et al.* [73] trained ANNs using the back-propagation algorithm, mapping input audio parameters to output visual speech parameters. They state that the difficulty with ANNs is determining an effective topology, which can only be determined experimentally. Optimizing requires experimentation with the number of hidden layers, the number of nodes per layer and the number of networks, i.e. a single ANN for all visual parameters or one ANN per visual parameter.

Hong *et al.* [71] notes the importance of incorporating contextual information to account for coarticulation when training an ANN. They suggest that contextual awareness can be accounted for by using time delay neural network (TDNNs) models, like those used by Lavagetto *et al.* [87] and Curinga *et al.* [88]. These TDNNs map LPC cepstral coefficients of speech to lip parameters. However, Hong *et al.* state that, for these networks to handle a larger vocabulary, they require a large number of hidden units, which results in high computational complexity during model training.

In conclusion, ANNs have the potential to capture complex audio-to-visual mapping, However, though they have two main problems. Firstly, more hidden layers appear to produce better pattern recognition but larger ANNs require more data for effective training. Secondly, ANNs are a'black box' approach to VSS, meaning that there is no way to learn from or understand the non-linear relationships formed between the ANN's inputs and outputs.

4.2.4 Gaussian Mixture Models

Rao and Chen [56; 89; 73] experimented with joint probability distributions of audio and visual parameters using Gaussian mixture models. Their direct estimation method was based on eleven-dimensional continuous audio-visual parameters. These consisted of eight LPC-derived cepstral coefficients (or filter-bank coefficients) from audio speech, and three oral parameters, extracted from visual speech recordings.

An optimal estimate of a visual parameter v given an acoustic feature a was derived using a joint probability density function (PDF) $f_{av}(a, v)$, given by Equation 4.2.1. Here, the mixed audio-visual parametric model uses K Gaussian distributions.

$$f_{av}(a, v) = E\langle v|a\rangle = \sum_{i=1}^{K} c_i \aleph(\mu_i, R_i) \tag{4.2.1}$$

In Equation 4.2.1, c represents a non-linear mixture weighting and $\aleph(\mu, R)$ represents the Gaussian distribution with mean μ_i and correlation matrix R_i. These variables can be partitioned into audiovisual sub-matrices as shown in Equation 4.2.2.

$$\mu = \begin{bmatrix} \mu_a \\ \mu_v \end{bmatrix}, R = \begin{bmatrix} R_a & R_{av} \\ R_{av}^T & \sigma_v^2 \end{bmatrix} \tag{4.2.2}$$

where μ_a and μ_v are the means of the acoustic and visual parameters, respectively. R_a is the auto correlation matrix of the acoustic features, R_{av} is the covariance matrix of the acoustic and visual parameters, and σ_v^2 is the variance of the visual parameters.

The optimal estimate of v given a is found using Equation 4.2.3, which can be expressed in closed form as Equation 4.2.4.

$$\hat{v} = \int v \frac{f_{av}(a, v)}{f_a(a)} dv \tag{4.2.3}$$

$$\hat{v} = \sum_{i=1}^{K} \frac{c_i \aleph(\mu_{i,j}, R_{a,i})|_a}{f_a(a)} b_i^T \begin{bmatrix} 1 \\ a \end{bmatrix} \tag{4.2.4}$$

where

$$b = \begin{bmatrix} 1 & \mu_a^T \\ \mu_a & R_a \end{bmatrix}^{-1} \begin{bmatrix} \mu_v \\ R_{av} \end{bmatrix} \tag{4.2.5}$$

The result is a Gaussian mixture model which estimates continuous VSS parameters [73].

The advantage of this approach is its use of continuous valued parameters, which can give more accurate estimates and smoother transitions for visual parameters [71; 73]. Figure 4.6 illustrates the VSS parameters estimated by the Gaussian mixture-based model and a VQ model. Here, the VQ's estimates, represented by the dashed line, are seen to be step-wise unlike the line deduced from the continuous Gaussian distributions, which is represented by the solid line.

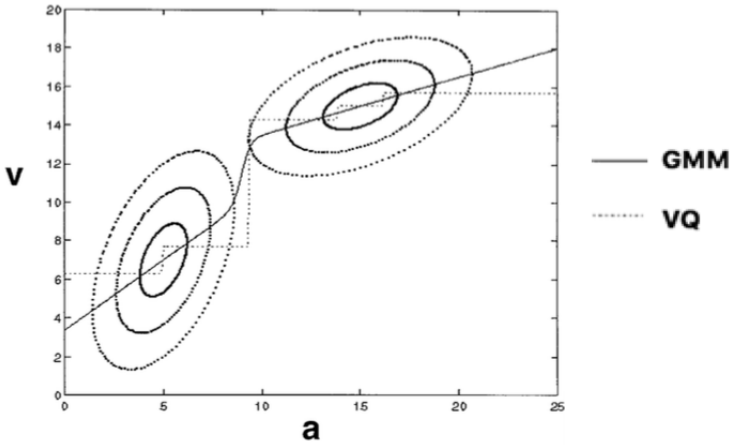


Figure 4.6: Example of a Gaussian mixture-based audio-to-visual parameter prediction. Reproduced from [73].

Rao and Chen's Gaussian mixture model was trained using only four vowel sounds. Therefore, the potential for their model to reliably synthesise each oral feature's full range of parameters needed for English is highly speculative. Testing also revealed that the Gaussian distribution model is sensitive to noise in speech recordings. Tao *et al.* [72] also note that, although the Gaussian mixture model smoothly synthesises visual speech, it is also subject to over smoothing the visual parameters, which can result in under-articulation. In conclusion, it appears unlikely that this approach will effectively relate a larger set of audio instances to their visual speech counterparts.

4.2.5 Decision Trees

Galanes *et al.* [90] presented a VSS method which used a set of regression trees to cluster oral feature parameters based on their phonetic attributes. In their database, each phoneme kept the before and after phoneme information.

Galanes *et al.* split tree nodes by first asking all possible phonetic attribute questions, creating multiple subsets. The question leading to the most homogeneous cluster of oral

feature parameters would then be chosen to split the node. This process was repeated at each node until a minimum occupancy stopping criteria was reached. Cluster homogeneity was based on the average distance of the feature parameters vectors in the cluster to their centroid. To ensure convergence during clustering, the homogeneity of the new clusters had to be greater than that of the parent node cluster. During VSS, the tree is searched for the desired viseme feature parameters which correspond to the phonemes in the target sequence. Interpolation of the centroid visemes then ensures a continuous feature trajectory is formed for the target phoneme sequence.

Galanes *et al.* did not perform subjective or perceptual evaluation of their method, and made no attempt to control a virtual avatar. The authors do mention, however, that the synthesised visual speech trajectories showed signs of under-articulation, when compared to the measured feature trajectories. The authors state that this under-articulation is likely due to the use of speech feature parameter centroids. A major advantage of their VSS system is the ability of the decision trees to take into account the previous and next phonemes associated with the current phone during synthesis. According to Galanes *et al.*, this contextual awareness greatly improved the likelihood of reproducing naturally occurring coarticulation effects.

Mattheyses *et al.* [6; 67] subsequently expanded on the work by Galanes *et al.*, developing their own classification and regression tree (CART) algorithm capable of video-realistic phoneme-to-viseme mapping. They measured homogeneity using an impurity metric, I, as described by Equation 4.2.8. Here, N is the number of phoneme instances in a cluster and Z denotes which phonetic attribute question was asked. The mean and variance values are denoted as μ and σ, respectively. Each subset's mean, μ_i, is calculated by summing the Euclidean distances between each pair of oral feature parameters p_i and p_j, as per Equation 4.2.7.

A visual speech feature parameter, p, is extracted from each phoneme in a video recording's utterance. This parameter is based on three weighted samples extracted from the 25%, 50% and 75% segments, c, of the phonemes video. To calculate impurity, the distance between a subset's oral feature parameters is required. This is given by weighting the sum of the phoneme segment's feature displacements, which is denoted by $d(pi, pj)$ in Equation 4.2.6.

$$d(p_i, p_j) = \frac{1}{4} \mid c_i^{25} - c_j^{25} \mid + \mid c_i^{50} - c_j^{50} \mid + \frac{1}{4} \mid c_i^{75} - c_j^{75} \mid \tag{4.2.6}$$

$$\mu_i = \frac{\sum_{j=1}^{N} d(p_i, p_j)}{N - 1} \tag{4.2.7}$$

$$I_Z = N \times (\mu + \lambda \times \sigma) \tag{4.2.8}$$

Once the phonetic question leading to the cluster with the lowest impurity, I_Z, is identified, it is used to split the parent node into two child nodes. Note that Mattheyses *et al.* use λ as a scaling factor in Equation 4.2.8, but omits its derivation. The authors' calculation of the variance, σ, is also omitted.

Though their technique required careful image manipulation to maintain photo-realism, Mattheyses *et al.*'s life-like results make decision trees appear highly attractive for good VSS. Their decision tree based VSS has two major advantages, the foremost being its ability to correctly predict visual speech parameters using a many-to-many phoneme-to-viseme mapping scheme, which does not require an exact phonetic match in a database.

The other advantage is the relatively limited data requirements needed to produce this VSS model.

An arguable disadvantage of the approach by Galanes *et al.* and Mattheyses *et al.* is their limited consideration of coarticulation. Both do not consider the actual oral feature parameters of the before and after phonemes, but only the phonetic context. Therefore these systems still ultimately use static visemes. However, the success of their approaches gives strong indications that decision trees are a robust means to account for coarticulation during the phoneme-to-viseme mapping stage.

4.3 Conclusion

Research indicates that VSS based on audio-to-visual mapping rules does not synthesise speech as well as probabilistic visual speech models. The latter approaches show more promising outcomes as they automatically determine more agreeable relationships between visual and audio cues. With the project's limitations in mind, decision trees present the greatest strengths among the considered functional approximation approaches.

Compared to VQ, ANN, HMM and Gaussian-based VSS, decision trees require little data, provide a many-to-many phoneme-to-viseme mapping, are more intuitive to interpret, do not require quantisation of continuous data (unlike VQ models), are quick to train, are computationally efficient to apply, and have already shown great potential in the most demanding VSS scenario, namely photo-realistic VSS. All these advantages support this project's objectives and constraints. Therefore, decision trees were chosen for our VSS system.

Equally important to choosing a visual speech model for viseme selection was the method of data capture and preservation of coarticulation information during unit concatenation. These problems are discussed next in Chapter 5.

Chapter 5

Data Compilation

This chapter begins by detailing the development of the data capture system and its coupling with the selected MakeHuman character and BGE animation control tools. We then establish the best units upon which to base our viseme boundaries. These findings then influence the chosen interpolation method and the database's final formatting.

5.1 Motion Capture

Blender's integrated motion capture tool can bind tracked marker movements to specific vertexes of a character's mesh. For example, Blender is able to track fiducial markers placed on a performers face and map them to individual mesh vertices on a virtual character. The marker mappings must be manually assigned to mesh vertices, requiring a carefully scaled calibration to reproduce the observed movements.

This approach was not used for VSS because it was found that assigning and moving individual mesh vertices and their related units required a lot of calibration and vertex re-targeting. Motion capture used in this manner traditionally goes through a post-editing process where artists remove the character mesh's vertex deformations or unnatural gestures. This is not a feasible method for a VSS system.

We chose to use an alternative data-driven approach to animation. To achieve this, performance data, in the form of tracked facial marker trajectories, was utilised to control a character generated by MakeHuman through scripted bone-driven shape key animation (discussed in Section 3.4.2). This approach was very similar to that of Chuang *et al.* [91]. They refer to this approach as motion re-targeting, an animation procedure which maps measured facial expressions directly to a virtual character's corresponding target animation. Chuang *et al.* state that this approach is less likely to cause distortions when a smaller number of shape keys is used.

5.1.1 Tracking Method

Although a basic open source tracking software is provided by Brown [92], a custom marker-based motion capture system was developed for this project. The advantages of this system include: time saved by including automated video file opening functionalities, the ability to tailor facial marker location and identification, a supervised error detection system specific to the marker layout used and a customised data output format.

The marker layout of was devised before shape key selection. However, layout was based on existing VSS research, including formalised facial parametrisation techniques,

discussed in Section 3.3, and data capture methods related to the work discussed in Chapter 4. A common trend in facial marker layouts was found in several papers [33; 69; 78; 82; 84; 93]. A typical example of the this layout can be seen in Figure 5.1. Accordingly, this layout was chosen for our motion capture system.

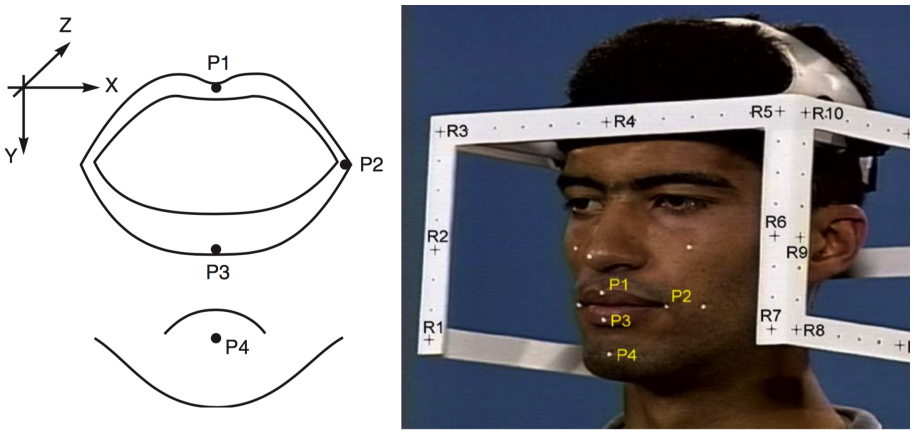


Figure 5.1: Popular fiduciary marker layout (left) demonstrated by De Martino *et al.*. A frame sample from the left side camera reveals their head apparatus (right). Reproduced from [93].

A Panasonic HDC-TM900 video camera, recording at 1920x1080 pixel resolution and 24 frames per second, was used to capture video. Since this device is consumer-grade, it is readily available. The video camera, along with the head of the recorded subject, was held in fixed positions by an adjustable desk mounted rig. A mirror, set at a 45° angle, was also included in the recording setup to allow for a side view of the face, as shown in Figure 5.2.

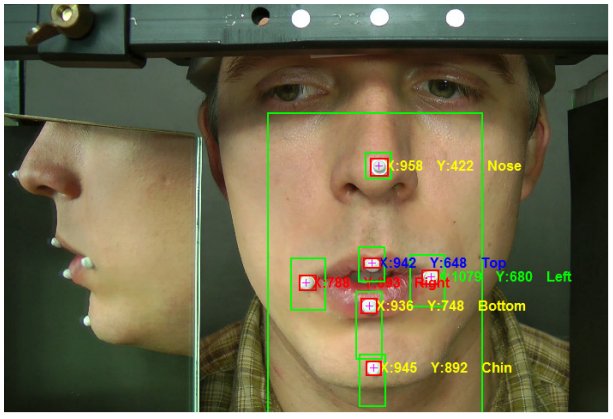


Figure 5.2: Facial feature marker trajectory tracking.

Our tracking algorithm identifies the horizontal and vertical positions of each white marker in each video frame, taking previous locations into account. In other words, the markers X and Y-axis coordinates were captured 24 times a second. If an identification error is detected, the user is prompted to manually locate the markers for that frame. All frames with labelled marker locations are made observable, allowing for semi-supervised automated tracker checking. Figure 5.2 shows an example of a fully labelled frame.

The advantages to this layout include effective use of a sparse set of markers as well as depth information for feature tracking. Depth information included the bottom lip's roll, deduced by considering the distance between the bottom lip and the chin marker and lip pursing (needed for "OO"-shaped poses). This was deduced from the distance between the left and right lip corner markers. Consequently, the side view of the head, provided by the mirror, was found unnecessary.

To reduce variations in recorded trajectories, the positions of the facial feature markers were normalised relative to the position of the nose. The nose was chosen because its movements correlated well with those of the performers head. The head and camera rig were used as further measures to stabilise the recordings. Note that is was also of great importance to ensure that the performer's mouth was closed at the start and end of each sentence to preserve rest-to-speech and speech-to-rest transitions.

5.1.2 Perfecting Digital Mimicry

The motion capture data can be used to animate the MakeHuman character as follows. Each of the six feature trajectories' X and Y-axis co-ordinates are read by the Python script, which controls BGE animations. The script reads a set of coordinates 24 times a second, equivalent to the recording frame rate discussed in Section 5.1.1. In this way, the virtual character's bone-driven shape key animations are updated with their new X and Y-axis co-ordinate values at the same rate that the feature markers were recorded. The sentence's audio starts playing when the first set of coordinates is read. This reproduces the captured audiovisual sentence on the virtual avatar in a time synchronous fashion.

Both Ekman's FACS [45] and the MPEG-4's FAPs [33] formalised facial parametrisation techniques were used as guidelines in discerning appropriate oral animations to select from MakeHuman's predefined shape keys. Consequently, a set of shape key animations was identified which could collectively represent common visual speech gestures. Each shape key was then assigned a scaling factor, which mapped the magnitude of the facial marker's displacements to the shape key animation's appearance. The process of calibrating the scaling factors was achieved by hand. Seven shape keys were identified. These are shown in Figure 5.3 and described in Table 5.1, which details their corresponding facial markers.

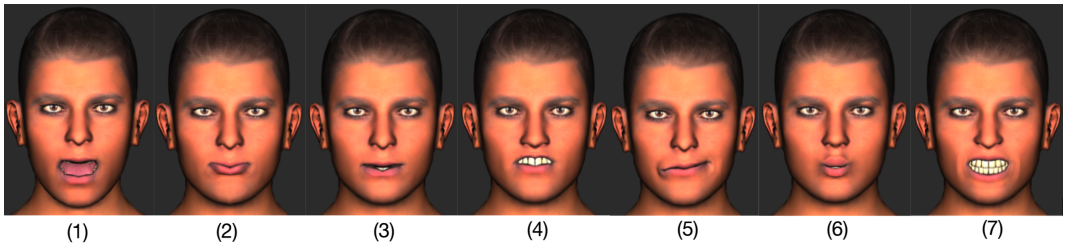


Figure 5.3: Selected MakeHuman oral shape key animations.

Although the MakeHuman character model does have the means to animate the tongue, we were unable to measure the motion trajectories that would be required to animate it. Therefore, animation of the character's tongue fell beyond the scope of this project.

Another shortcoming of the MakeHuman model was its inability to cater for more subtle effects. For example, the natural occurrence of moist lips pealing apart during

Table 5.1: Shape key animation and facial marker relationships.

Shape Key	Controlling Marker Feature
(1) Jaw	Chin marker
(2) Mouth eversion	Bottom lip roll (the distance from bottom lip to the chin marker)
(3) Mid lower lip	Bottom lip marker
(4) Mid upper lip	Combination of both the bottom and top lip markers
(5) Left and right mouth corners	Left and right lip corner markers
(6) Mouth pursing	Mouth width (the distance between the left and right lip corner markers)
(7) Even lip parting	Bottom lip marker

speech, which differs slightly depending on how dry the lips are. Other such effects include wrinkles and the relationships between muscle and skin movements, which can cause dimples for example.

5.2 Establishing Viseme Units

Good synthesis must be achieved by effectively utilising a small but visually diverse corpus. Therefore, VSS data compilation must first focus on the capturing and preservation of coarticulation effects. Here, the critical design choice is the selection of the boundaries which define the length of the viseme units. Once this design choice is selected, the most effective interpolation function can be evaluated, as discussed in Section 5.3.

Coarticulation considers the identification and influences of sequentially articulated sounds and their visual speech units [63]. These influences are particularly strong when the timing between consecutive phonemes is smaller than the contraction or relaxation time of the articulating muscles [94]. Coarticulation effects are dependant on the inertia of the articulatory organs in the context of their speech segments. In some cases, visual speech has been found to be influenced by segments up to five positions before or after the current pose [63].

By breaking recorded speech into segments, the naturally occurring transitions between the visual speech units can be preserved and re-used during synthesis. Understanding the coarticulation influences of these segments is vital to natural-looking VSS. Therefore, identification of an appropriate segmentation and interpolation style, which best captures and preserves coarticulation effects, is necessary.

5.2.1 Literature Review of Viseme Units

Lazalde *et al.* [95] evaluated three frequently used styles of viseme, which can be created using different visual speech segmentation approaches. The first, and notably most commonly encountered style, was static visemes, which are captured whilst a speaker holds a pose to produce select sounds. This is similar to the process used to identify the visemes in Figure 2.2 in Section 2.3. The second style uses coarticulated visemes, which are captured by calculating the mean pose of a central phone within a triphone sequence. The final style employs enhanced visemes, which use static viseme poses combined with additional pre and post phoneme oral feature information. Lazalde *et al.*'s results indicate that, the

more contextual information is re-used from the data recordings, the better the VSS. This is attributed to the reduction in the number of discontinuities resulting from the use of longer speech units [82].

VSS systems that incorporate coarticulation context require at least a carefully constructed database of **dynamic visemes**. In the context of this work, dynamic visemes do not represent a fixed-point pose, but rather a trajectory describing the evolution of an articulatory feature over time. But what is the most effective data segmentation approach for creating dynamic visemes for VSS?f

Cao *et al.* [96] captured dynamic visemes associated with single phoneme instances. These tracked motions, named *animes*, were selected from a number of frames for the duration of each phoneme. VSS involved the selection of the most plausible naturally occurring sequence of animes. If a sequence could be found, a "jump-match" cost function determined the best substitute sequence to use, based on phonetic labelling, feature trajectories, prosody and emotional labelling. This linked list of animes is then time-warped and blended in a piece-wise linear fashion to create the continuous synthetic visual speech sequence.

The disadvantage of the approach proposed by Cao *et al.* is that the single phone representation does not incorporate naturally occurring coarticulation. Instead, a transition cost model simply minimizes jerky movements from one anime to the next, disregarding any greater contextual settings.

An improvement is offered by divisemes or bivisemes, which are time-varying oral poses whose boundaries are defined by a sequence of two phones (diphones). The difference between biphone-based visemes, bivisemes, and diphone-based visemes, divisemes, is their segmentation coverage. Biviseme coverage includes the whole of both phonemes, were as divisemes cover the phonemes' centre-to-centre. Ma *et al.* [68] and Toutios *et al.* [97] both approached VSS using diphone-based visemes. According to Toutios *et al.*, diphones are advantageous because they capture the coarticulation phenomena which occur in the transition region between the centres of the two phones. The advantages of divisemes include their ability to capture and re-use natural phone transitions with concatenative techniques. Capturing every possible diphone is also achievable without requiring an excessively large database [97].

Bregler *et al.* [98] successfully demonstrated that trivisemes can lead to good VSS. Trivisemes are a style of dynamic viseme based on the boundaries of three sequential phonemes, known as triphones. Bregler *et al.*'s triphone approach was based on a photo-realistic conversational agent. This required a high level of VSS realism as unnatural speech motions are more apparent when using realistic avatars.

VSS using variable-length visemes has also proven to be effective. Taylor *et al.* [5] define visual speech boundaries as the zero-acceleration points of articulatory feature trajectories. These variable length units were clustered based on similarity to identify a concise set of dynamic visemes for VSS. In comparative tests to static viseme-based VSS, Taylor *et al.* found their dynamic visemes were strongly preferred.

Ma *et al.*'s [82] later work also uses dynamic visemes of variable-length. A target sentence is synthesised by searching for and concatenating the longest corresponding motion capture sequences, thereby maximizing natural coarticulation conservation. This was achieved using radial basis function networks based on phonetic transitions in motion capture data. Here, an advantage over divisme based VSS is suggested, however, the advantage over trivisemes or longer dynamic visemes is not addressed.

Another approach using variable-length visemes is the use of visyllables, a visual counterpart of the syllable [99]. For visyllable-based VSS, common vowel consonant arrange-

ments were segmented from motion capture information. These were then concatenated and time scaled. Capturing the over 10,000 syllables used in the English language was not practical, hence a syllabification algorithm was developed. This required knowledge of phonological rules for the segmentation of phoneme streams into valid syllable clusters. Due to the linguistic prior knowledge needed by this technique, it is not feasible in the context of our work. However, an interesting question raised by the researchers is whether having a greater range of audio units with which to associate visual speech units is advantageous in any way.

Comparison of the effects of viseme length in VSS is difficult to asses since published research jointly evaluates selection and concatenation procedures, the interplay of which makes analysis of the ideal viseme selection procedure difficult. When comparing the work by Taylor *et al.* [5] to the work by Bregler *et al.* [98], a question that arises is whether the increased complexity of variable length dynamic visemes relative to triphone based visemes leads to a significant improvement in VSS. Bregler *et al.* use photo-realistic agents and employed a significantly smaller database, in one case using only 2 minutes (1157 triphones) of film for VSS [1], as opposed to Taylor *et al.*'s dataset of 2542 sentences, which equates to approximately 8 hours of speech. Pelachaud [94] also suggests that the simplest method of accounting for coarticulation in VSS is to note the previous, present, and post-phoneme mouth positions throughout a speech segment. It is therefore arguable that triphone based segmentation can suffice for natural coarticulation in VSS.

5.2.2 Experimental Selection of Viseme Units

Previous work demonstrated the advantages of dynamic visemes over static visemes, however, direct comparisons of biphone and triphone-based dynamic visemes were limited. The data capture techniques and VSS models used by these authors also varied from those employed here. Consequently, we consider VSS performance using biviseme and triviseme-based visemes to select the best segmentation style.

Figure 5.4 illustrates biviseme (a and b) and triviseme (c and d) based VSS, synthesised using preliminary test motion capture data. In this Figure, all trajectories represent the chin feature's displacement relative to the nose. In both cases, the individual dynamic viseme trajectories as well as the continuous synthesised trajectory, which were created by interpolating the dynamic visemes, are shown relative to a natural speech trajectory. The visemes were generated by recording and capturing motion of oral features, which were carefully selected from a set of experimental test sentences. These sentences were segmented into biphone and triphone-based dynamic visemes and used to synthesise a target sentence. Both approaches synthesised feature trajectories by overlapping the dynamic visemes corresponding to a target sentence's phonetic sequence. The continuous synthesised trajectory was created using a linear transition function which merged each overlapping dynamic viseme. For each set of graphs, the Y-axis represents oral feature displacement and the X-axis represents time. Vertical lines were used to represent phoneme boundaries.

Figure 5.4 reveals that, although bivisemes have the potential to capture some transitional coarticulatory effects, trivisemes are better able to model the true trajectory. It is concluded that the additive contextual information captured by trivisemes is effective for trajectory modelling. Consequently, triphone-based dynamic visemes were chosen as

[1] Videos available at: `http://mrl.nyu.edu/~bregler/videorewrite/` (Accessed: November 12, 2015)

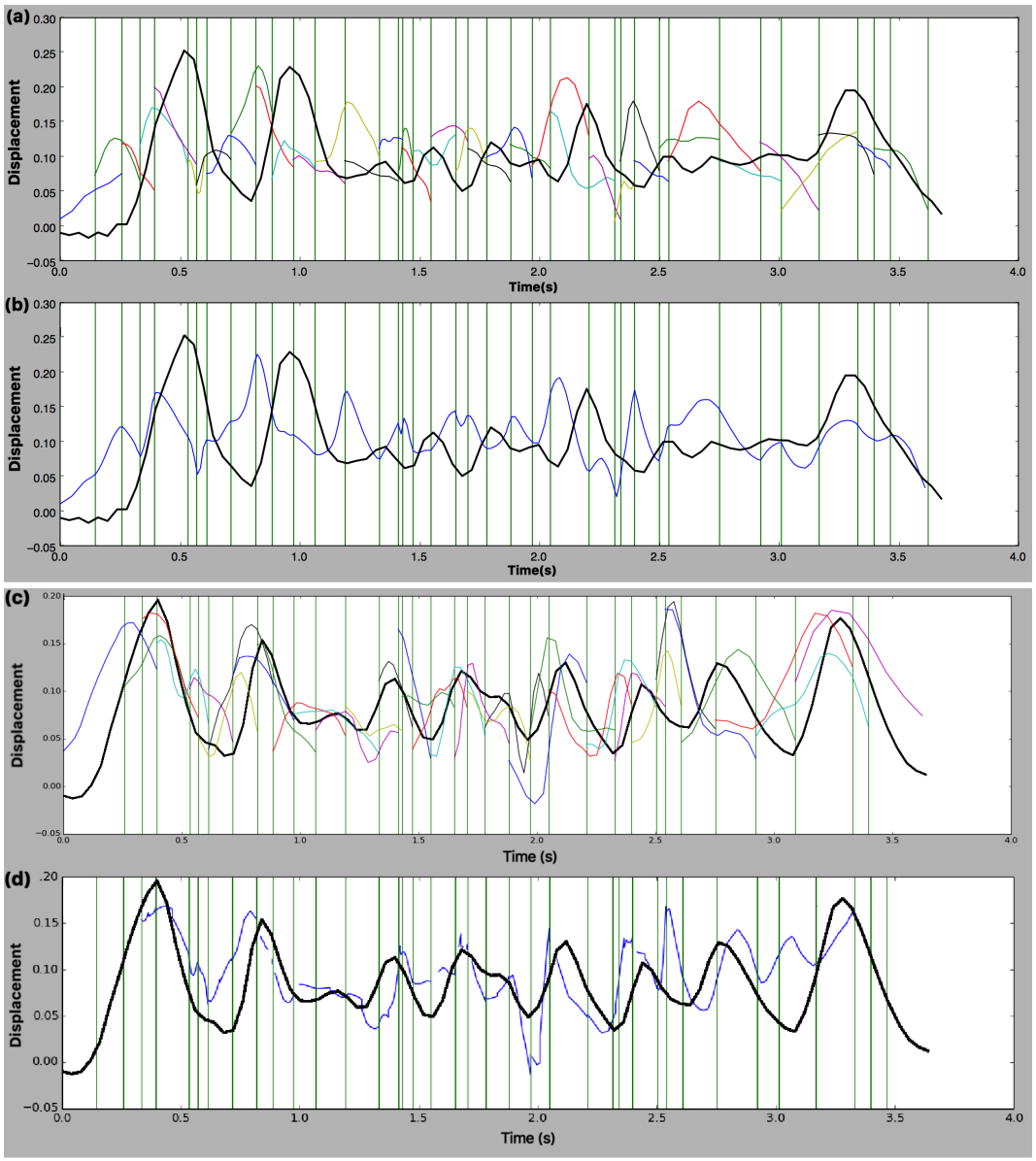


Figure 5.4: Two sets of comparative graphs illustrating biviseme (a and b) and triviseme (c and d) trajectories for a chin feature. Plots (a) and (c) show: (i) thin lines representing the individual biviseme and triviseme trajectories and (ii) a thick line representing the measured trajectory. Plots (b) and (c) show: (i) a thin line indicating the interpolated biviseme and triviseme trajectories and (ii) a thick line representing the measured trajectory. All plots use vertical lines to show phoneme boundaries.

the basic unit for our VSS system. Triviseme boundary locations are also easily extracted from phone alignments, making automated database formatting possible, which is discussed further in Section 5.4.1.

5.3 Establishing the Viseme Interpolation Function

An interpolation function compatible with the chosen dynamic viseme must now be identified.

5.3.1 Time Warping

Audio signal prosody is critical for believable VSS [100]. A key concern here is matching the timing of dynamic visemes to the triphone audio segments, in order to achieve natural VSS for the target utterance. It is therefore necessary to consider an appropriate formatting of the dynamic viseme corpus, which is both compatible with the chosen VSS algorithms and also preserves the visual speech information as best as possible.

It may be unreasonable to assume that the synthesis of visual speech at different rates can be achieved by merely warping the recordings of slow speech. At higher speaking rates, utterances are found to use fewer and/or less well articulated visemes [101]. However, most approaches assume that, provided the synthesised visual speech utterances have roughly the same tempo as the database recordings used to synthesise them, fairly simple contextual warping will yield credible results. Such systems are therefore limited in speech scenario types as their variations in speech rate must be small.

Data clustering for VSS requires comparative assessments of visemes. Ways in which trajectories can be compared include linear re-sampling of feature trajectories to obtain a set of trajectories of a fixed length [5], uniform time-scale warping [68; 98] and dynamic time warping [102].

Our approach is to re-sample the triphone-based visemes at ten uniformly spaced instances to obtain dynamic visemes of a fixed length. This allows the similarity of two trajectories to be computed by simple point-wise comparisons, and later, provides for employment of other comparative algorithms. Additional advantages to the chosen sample resolution include preservation of the trajectory's shape, which ensures continuity is maintained during viseme time-rescaling, predictable data storage requirements and simple time rescaling during VSS. For time scaling, each dynamic viseme would be stretched uniformly to match the length of the target triphone.

5.3.2 Interpolation Function Selection

Most VSS approaches apply viseme sequence selection followed by interpolation. Both steps aim to synthesise a cohesive trajectory for each visual speech feature, which is then used to animate a virtual character's speech. This section considers the latter problem, which tries to produce smooth transitions between the boundaries of selected visual speech segments.

There are no acknowledged rules to correctly shift, stretch and concatenate trajectories. Instead, the literature suggests that each VSS system employs a tailor-made concatenation and smoothing method, which is specialised to a viseme type. Therefore, it is difficult to develop a general approach.

Two common VSS boundary types are sequential and overlapping. Both boundary types commonly employ interpolation mechanisms to join sequences of static or dynamic visemes. Interpolation methods common to animation software (which use sequential units) include linear and Bézier interpolation [5; 9]. Other interpolation methods include B-Splines (and derivations thereof, such as duration control B-Splines [103]) and Hermite interpolation [104]. Interpolation methods that conserve coarticulation between trajectory boundaries are favoured. However, sequential boundaries are prone to discontinuity errors. For example, in cases were there is little temporal spacing between sequential feature trajectories, stretching algorithms have to be developed to adjust boundary values to better coincide the joining of the trajectories [99]. Some researchers have considered the

smoothing of sequentially synthesised trajectories using low pass filters, but with little improvement over the other trajectory-smoothing algorithms [82].

Transitions between overlapping dynamic visemes tend to depend on the structure of the visemes. Transition options range from simple linear transitions to complex weighted functions. Although Cohen *et al.* [58] (Section 4.1.1) use static visemes, they use a rule-based function which applies overlapping dominances assigned to sequential static viseme trajectories. Their approach gives greatest dominance to the current viseme, with the predecessor and successor visemes being effected exponentially less the further away they are in time. The major advantage to overlapping dynamic viseme functions is the possible blending of start and end trajectory segments. This prevents full bias being given to potentially discontinuous or incorrect segments of dynamic viseme trajectories.

Bregler *et al.* [98] synthesised continuous video sequences from natural speech utterances during VSS by interpolating visemes. They emphasised the middle of each triphone-based viseme, cross-fading the overlapping regions with the neighbouring visemes. This approach worked well, with the findings suggesting that such a transition method is effective in capturing and reproducing both forward and backward coarticulation effects. Consequently, a similar overlapping approach was chosen for the basis of our VSS. There is, however, room for experimentation with the transition approaches.

To begin the assessment of our interpolation function, the overlapping regions of the triphone-based dynamic visemes were segmented. In total, each viseme was divided into six segments, thereby allowing for appropriate consideration of each overlapping phonetic portion by the transition functions. A tapered piecewise linear weighting was then assigned to each phase of the overlapping viseme trajectories, as seen in Figure 5.5. To avoid under and over-articulation, the overlapping phases of the interpolation function always sum to one at any point in time. An example of two overlapping phases are highlighted in Figure 5.5. Note that the interpolation function is applied after each dynamic viseme has been scaled to match the duration of the triphone in the target utterance.

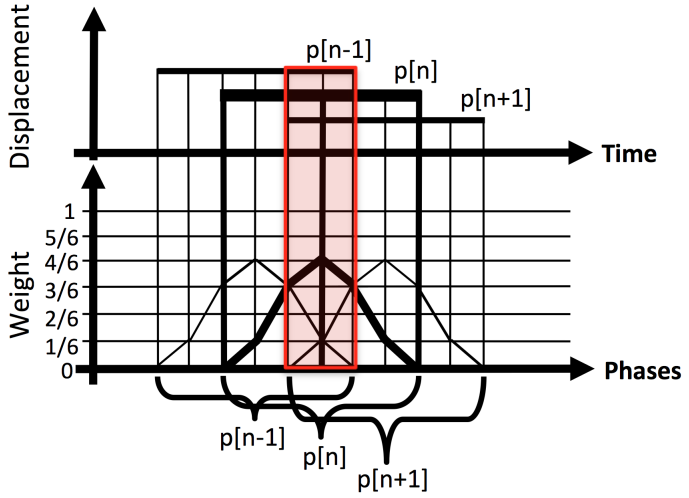


Figure 5.5: Illustration of how three successive overlapping dynamic visemes p[n-1], p[n] and p[n+1] are interpolated to create a continious feature trajectory. The overlapping portions of the visemes are combined using a piecewise linear weighting which accentuates the centres of each dynamic viseme. The highlighted portion indicates how the overlapping interpolation functions always sum to one.

Tests involved varying the weightings assigned to different phases. It was found during informal testing that too heavily weighting the centre of the trivisemes, as well as too high a rate of increase towards the viseme centre, results in a noisy and unnatural synthesised feature trajectory, as shown in the top graph of Figure 5.6. This leads to jerky motions when animating the virtual avatar.

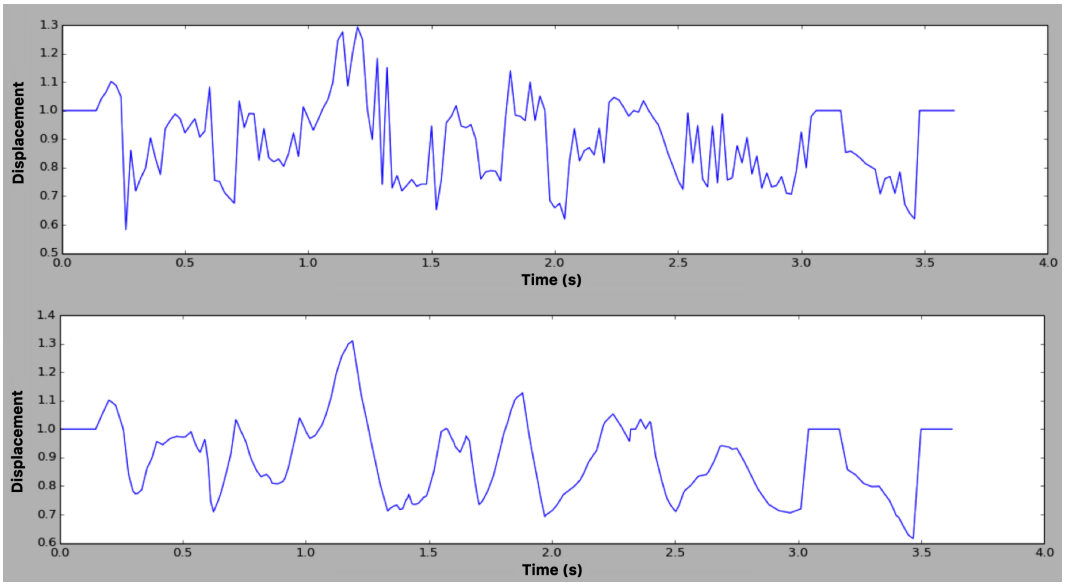


Figure 5.6: Triviseme VSS feature trajectories resulting from different weightings used by interpolation functions. Top: represents a fast increment with heavy centre weightings. Bottom: represents a slow increment with lesser central weighting.

Various design iterations of interpolation functions eventually led to an agreeable set of values, seen in Figure 5.5, which consistently synthesised natural-looking trajectories. Informal perceptual tests concluded that our chosen interpolation parameters avoided discontinuity by conserving pre- and post-viseme context information during VSS.

It is also important to be aware that natural utterances are variable, as Figure 5.7 demonstrates. Therefore, differences are to be expected, not only between measured and synthesised speech, but also between two measured repetitions of the same passage.

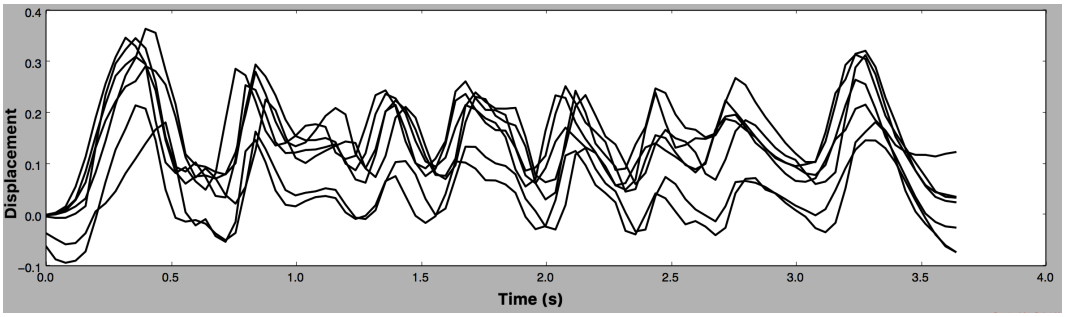


Figure 5.7: The measured trajectories of the chin feature for six repetitions of the same sentence.

This raises a question first asked by Theobald *et al.* [105]: "are the differences between the reference and synthesised parameters significant"? Answering this question is difficult.

However, it is assumed that by evaluating the VSS system's interpolation function to carefully conserve natural speech segments, that these differences were minimised.

5.4 Database Construction

With the data capture system, segmentation style and interpolation function concluded, the database's formatting could be finalised.

5.4.1 Database Processing

Hand labelled and time aligned X-SAMPA phonetic annotations defined the boundaries of our dynamic viseme trajectories. The segmented trajectories were sampled at ten uniformly spaced instances to obtain fixed-length dynamic visemes that will be used for VSS. Most of this process is automated. However, each sentence in the continuous speech recording must be manually edited to form precise sentence segments for both tracking and phonetic annotations.

Once each sentence has been phonetically annotated, and its corresponding trajectory triphone segments linearly sampled to create the dynamic visemes, each triphone was labelled with articulatory attributes. As mentioned in Section 2.2, each X-SAMPA phoneme can be classified using articulatory attributes. For example, an 'f' is a voiceless, labiodental fricative and an 'n' is a nasal alveolar. Therefore, a list of all possible articulatory attributes was made, and used as a binary labelling system for each dynamic viseme's phone instance in the database. These phonetic attribute labels are later used by the implemented decision tree training algorithms, described in Section 6.1.

5.4.2 Dabase Size

Previous work on virtual character-based VSS used databases ranging in size from around 300 to over 2000 sentences [5; 72; 78; 97]. For photo-realistic VSS, the employed corpus ranged from 422 natural speech sentences [96] to around 2000 audiovisual sentences (equivalent to 138 minutes) [67]. Bregler *et al.* [98] successfully demonstrated photo-realistic concatenative triphone-based VSS using 1300 female or 2900 male natural speech utterances. Such datasets are too large to be considered for our targeted low resource environment. The main limitation is the required manual phonetic annotations. This process is time consuming and requires expert phonetic knowledge, making it both difficult to perform and expensive.

For this thesis, 120 phonetically rich sentences (approximately 10 minutes of speech) were read aloud by a single non-professional voice actor. Continuous speech was used to capture naturally occurring coarticulation effects. The sentences used were selected from the TIMIT corpus [106], which includes utterances that are designed to be phonetically diverse. The assumption was that, by covering a broad variety of phoneme sequences or sounds, a wide range of visual speech gestures would be captured.

The limited size of the database is likely to effect the performance of the VSS system. This impact is assessed in Section 7.1.

5.5 Conclusion

Scripted bone-driven shape key animation was successfully employed to control visual speech using a custom motion capture system's data. Through experimentation, a viseme unit size and interpolation function was devised. These findings then influenced the format of our database.

The next challenge is to develop a decision tree training algorithm which is capable of clustering similar dynamic visemes. The dynamic visemes determined by the decision tree must then be interpolated to synthesise visual speech trajectories.

Chapter 6

Implementation

This chapter will detail the development of three decision tree-based algorithm's, which will be used for many-to-many phoneme-to-viseme mapping for VSS. First, a baseline decision tree algorithm, capable of using our dynamic visemes, will be described. Subsequently, we present two attempts to improve the performance of this baseline system. All three approaches will be assessed by means of objective and subjective evaluation.

6.1 Framework for All Decision Tree Algorithms

We will begin by adapting ideas developed by the authors, referred to in Section 4.2.5, to cluster our linearly-sampled dynamic viseme trajectories into groups of a similar shape. In other words, a decision tree-based algorithm will be developed to cluster homomorphic time-series data instances.

Section 4.2.5 described how decision tree-based algorithms have been used in the literature to cluster a corpus of static visemes in a manner that is useful for VSS. These algorithms split visemes clustered in a tree node by determining the phonetic attribute question that leads to the most homogeneous sub-clusters. The leaf nodes of the tree are then split using the respective optimal questions, and the process repeated for the new child nodes, until a stopping criteria is reached.

Note, that a separate decision tree must be trained to sort through the dynamic visemes for each facial feature. These features were mentioned in Table 5.1 and include the top, bottom, chin, left, right, mouth width and bottom lip roll features, which are abbreviated to the TBCLRWr features. Each TBCLRWr dynamic viseme feature is represented by its phonetic attribute labels and corresponding feature trajectory. It is the dynamic viseme's phonetic attributes that will be questioned to split the leaf nodes of the decision tree. Figure 6.1 depicts what such a tree may look like.

Mattheyses *et al.* [6; 67] suggest two ways of initiating decision trees, which they refer to as pre-clustering. The first approach creates multiple trees, with each tree's starting node containing only trivisemes with a specific central phonetic attribute. The second approach creates two larger trees, the first one only querying vowel attributes and the second querying only consonant attributes.

Comparing our final dataset, with just over 3000 triphone instances, to that used by Mattheyses *et al.*, which had over 120,000 instances, it is clear that our dataset is limited. Therefore, it was assumed that early partitioning of the data, or the use of separate decision trees for different phonetic question types, would not have any benefits.

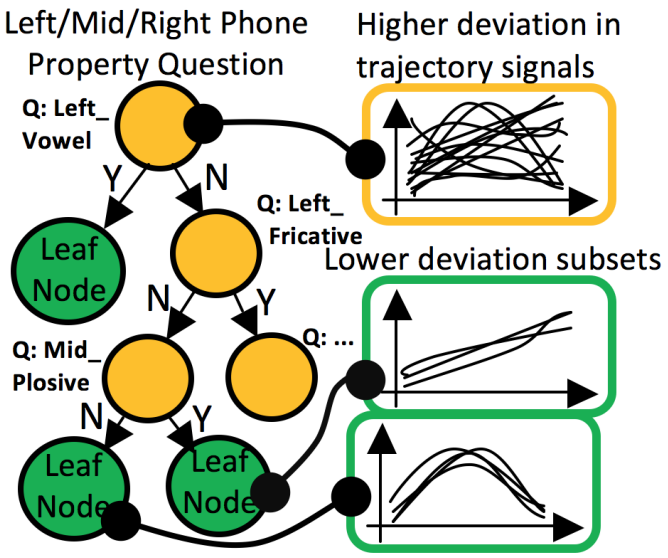


Figure 6.1: Illustration of the decision tree clustering dynamic viseme trajectories with similar morphologies based on their phonetic attributes.

We therefore chose to use a single decision tree per TBCLRWr feature, which worked as follows.

Each tree uses a hierarchical list of phonetic questions during training. The first three tiers of the tree always ask vowel/consonant questions, querying the central, start and end phonemes respectively, thereby splitting the first node into three tiers with eight possible vowel/consonant triphone combinations. Further splitting is based on identifying the phonetic articulatory attribute question whose application leads to subsets with the greatest homogeneity in feature trajectories.

In cases where two or more articulatory attribute questions result in subsets with equal homogeneity, a prioritised reference list is used. This list ranks articulatory attributes in order of importance, based on how common their occurrence is in X-SAMPA. This was done to prevent the decision tree from having excessively long or thin branches. For articulatory attributes which have the same frequency of occurrence, those with the greatest effect on visual speech were prioritised. Section 2.2 and Appendix B assisted in gauging the magnitudes and number of features involved in the sound's articulation, which were used to identify it ranking of visual importance. This was intended to help better the decision tree splits by grouping attributes which are more visually pronounced.

A holistic view of the decision tree's algorithm is shown in Figure 6.2. This framework is used for each of the algorithms discussed in this chapter. Figure 6.2 starts with the root node of the tree, which represents all the dynamic visemes for one of the seven TBCLRWr features. A recursive procedure is then entered, which runs one of the three node splitting algorithms, discussed later in this Chapter (the three algorithms are abbreviated to MD/KM/SS-CART in Figure 6.2). The framework for the decision tree also includes control of meta parameters related to stopping criteria. These include node occupancy and a minimum improvement in deviation between parent and child nodes. These parameters are important in preventing over fitting during training.

Once trained, the decision tree can be used to synthesise new TBCLRWr feature trajectories. This employs a synthesis algorithm, shown in Figure 6.3. To start, triphone sequences are extracted from the target sentence. Each TBCLRWr feature's decision tree is then traversed in search of the target sentence's triphones. The dynamic visemes with

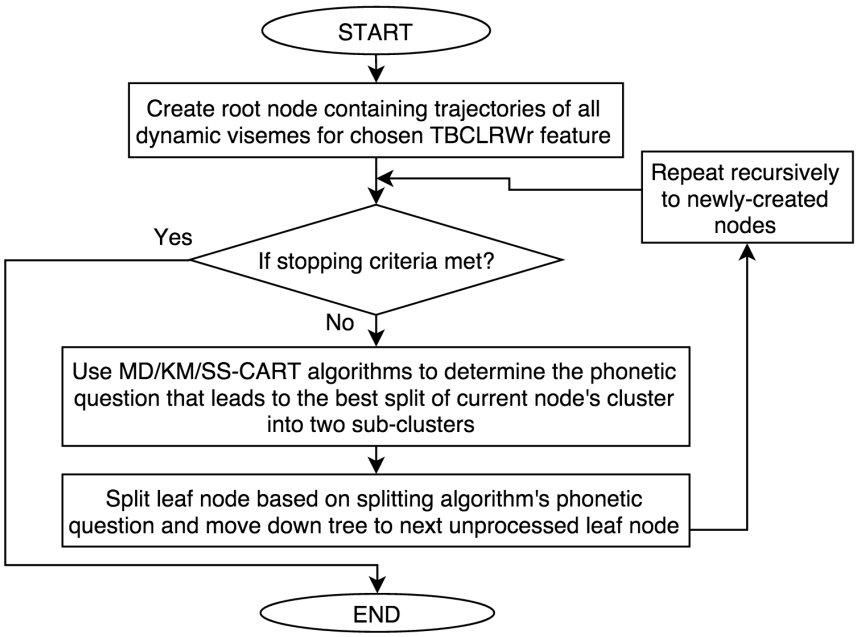


Figure 6.2: Flow diagram representing the framework of the decision tree's training algorithm.

the closest match to the target triphone's phonetic context are selected.

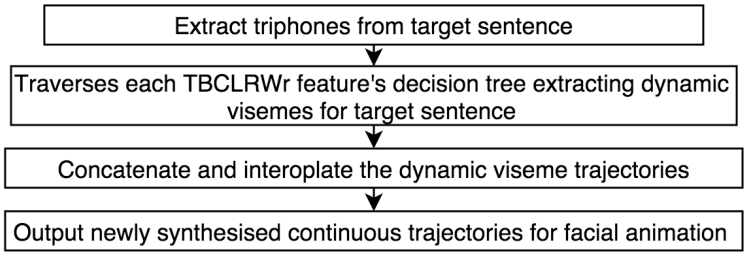


Figure 6.3: Flow diagram for employing the decision tree for VSS.

Once all dynamic visemes have been located, an interpolation function (discussed in Section 5.3.2) is employed to synthesise a continuous trajectory for each of the TBCLRWr features. The resulting trajectories are then ready to be passed to the BGE for character animation. As discussed in Section 5.1.2, scripted bone-driven shape key animation enables the new sentences to be spoken by the virtual character

6.2 Baseline Minimum Deviation Algorithm

This section describes the first variant of the dynamic viseme decision tree training algorithm, referred to as the baseline **minimum deviation CART (MD-CART)**. Here, the key design decision is the adoption of a previously proven metric for assessing the homogeneity of dynamic viseme subsets.

In Section 6.1, Mattheyses *et al.* [6; 67] described how a decision tree-based learning algorithm clusters visemes by applying an impurity metric (shown in Equation 4.2.8). This metric is based on the Euclidean distances between static viseme instances (calculated by Equation 4.2.6 and Equation 4.2.7). These distances are used to determine

the phonetic question leading to the greatest homogeneity in the new child nodes. Our baseline MD-CART is an adaptation of this work, capable of clustering dynamic visemes, and is computed as follows.

Firstly, all dynamic visemes are pooled into the root node of the decision tree. Each possible phonetic attribute question is then used in turn to split the root node into two subsets. The mean of each subset's dynamic viseme trajectory instances, μ_n, is calculated by applying Equation 6.2.1 to each of the ten trajectory instances P, therefore $n : 1 \rightarrow 10$. The average deviation of the constituent trajectories from the mean, D_{Avg}, is then calculated for each subset using Equation 6.2.2. Finally, the phonetic attribute question leading to the smallest is chosen.

$$\mu_n = \frac{1}{M} \sum_{i=1}^{M} P_{n,i} \tag{6.2.1}$$

$$D_{Avg} = \frac{1}{10} \sum_{n=1}^{10} \frac{1}{M} \sum_{i=1}^{M} \mid P_{n,i} - \mu_n \mid \tag{6.2.2}$$

This process is now applied recursively to each new leaf node until a stopping criteria is reached, as shown in Figure 6.2's flow diagram. The stopping criteria used for the MD-CART algorithm is dependent on both a minimum node occupancy count as well as a minimum improvement in deviation, ΔD_{min}, calculated using Equation 6.2.3. Here, improvement in deviation, ΔD, is measured between a set of parent node's trajectories S_p and its yes and no child node subsets, S_{cy} and S_{cn}. Experimentation with these thresholds is discussed in Section 7.2.

$$\Delta D = D(S_p) - (D(S_{cy}) + D(S_{cn})) \tag{6.2.3}$$

The next two sections describe the two approaches taken to improve the MD-CART's node splitting algorithm. The first approach uses k-means clustering to assign the dynamic visemes to classes. The second approach uses an information gain metric which searches for the phonetic question leading to the greatest improvement in log-likelihood. The former decision tree node splitting approach will be referred to as the ***k*-means clustering CART (KM-CART)** algorithm and the latter as the **split score CART (SS-CART)** algorithm.

6.3 *k*-means CART Algorithm

Popular decision tree algorithms, such as CART [107; 108], Iterative Dichotomiser 3 (ID3) and C4.5 [109; 110], require each data point to have a 'defining attribute', referred as a class or class type.

An example for defining a class attribute can be taken from Cancer research. A tumour in the human body may have many attributes, such as size, shape, place, age and so on. In this case, the class type is the attribute which the decision tree model uses to identify a good node split, that being if the tumour is benign (non-cancerous) or malignant (cancerous). Such a model can then be used to make predictions about whether a new tumour is cancerous or not based on its attributes alone. Classes can be binary, multivariate or real numbered.

As our dynamic viseme corpus has no class type, it is necessary to develop a method of assigning it one. This can be achieved by classifying the viseme's trajectories into

groups with similar morphologies. A CART algorithm could then be used to split nodes with respect to their class types. The classification method thus requires a measure of time-series similarity across data.

Aggarwal [111] suggests that the k-means algorithm is a reliable means of time-series data clustering. He states that

> a significant difference between time-series data clustering and clustering of objects in Euclidean space is that the time series to be clustered may not be of equal length. When this is not the case, so all time series are of equal length, standard clustering techniques can be applied by representing each time series as a vector [...] with such an approach, only similarity in time can be exploited.

As stated in Section 5.3.1, our visemes were all sampled at ten uniformly spaced instances. This means that each trajectory has a fixed length, thereby reducing the complexity of comparisons and allowing for standard application of k-means clustering.

The k-means algorithm partitions datasets into similar groupings using an iterative technique that minimises a sum of point-to-centroid distances. These distances are calculated as an error function, E, shown in Equation 6.3.1 [112]. Here, $d(\mathbf{x}, \mu(Ci))$ denotes the Euclidean distance between the data instances $\mathbf{x}$ and the centroids of their k clusters, $\mu(C_i)$. The objective here is to find clusters which minimise the error measure over all time-series instances.

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} d(\mathbf{x}, \mu(C_i)) \tag{6.3.1}$$

De Martino *et al.* [93] also considered clustering of phonetically different articulatory trajectories into geometrically similar groups. They successfully applied k-means clustering to CVCV non-sense words. They report identifying "distinguishable articulatory patterns from the trajectory of a phoneme in different phonetic contexts, using the Euclidian distance between them as the criterion of similarity". An average value of these clustered visemes was subsequently used to represent the articulatory targets composing the cluster.

Initial applications of k-means clustering to our corpus of dynamic visemes failed to produce a functional set of homogeneous clusterings. The problem was linked to the even spread of values at each of the dynamic viseme trajectory's ten instances, which made it impossible for the algorithm to clearly identify areas of local error minima. It was also difficult to determine the correct number of clusters to use.

The solution was to implement a step-wise approach which used a modified version of the DISMEA divisive hierarchical clustering algorithm [112]. The DISMEA algorithm continuously splits nodes by applying k-means clustering to a parent node, resulting in k new child nodes.

Our splitting algorithm uses k-means clustering [108] to find two groupings for each node's data (therefore k=2). It then differs from the DISMEA algorithm by assigning the groupings as a class attribute to each dynamic viseme, based on the outcome of the k-means clustering. A CART algorithm [108] (like that mentioned at the beginning of this Section) is then used to identify the phonetic attribute which best reproduces this clustering, using the new class attributes to assess goodness of split. This phonetic attribute is sent back to the decision tree training algorithm and used to split the parent

node. Figure 6.4 illustrates this KM-CART algorithm, which is repetitively called by the decision tree training algorithm, as seen in Figure 6.2.

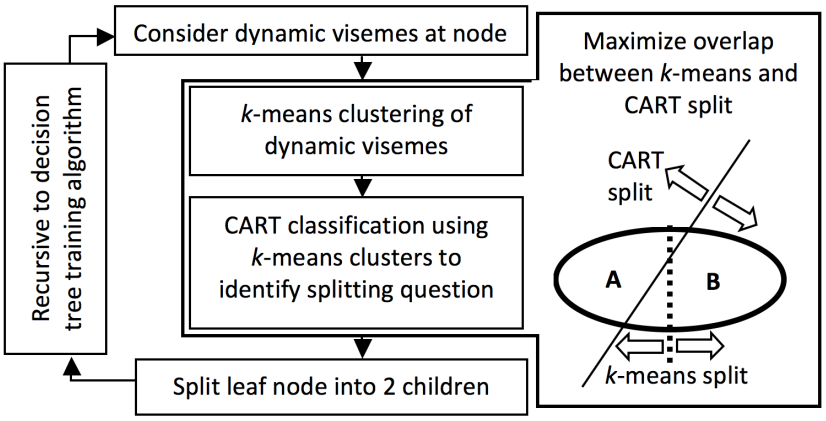


Figure 6.4: Illustration of the KM-CART algorithm, showing how the k-means (k=2) classifier assigns the dynamic visemes a class type. Subsequently, the CART classification algorithm finds the phonetic attribute question which splits the parent node's dynamic visemes into two child nodes which best match the clusters produced by the k-means algorithm.

The divisive nature of our KM-CART algorithm reduces the initial complexity of clustering all the data into a large number of groups. Therefore, the algorithm can gradually sort the data into an ordered hierarchical structure. Note that, because the KM-CART algorithm tests more than one attribute when attempting to split a node, it can still be classified as multivariate CART algorithm [107; 109; 110].

The KM-CART algorithm's stopping criteria is the same as that used by the MD-CART algorithm, namely a minimum node occupancy count as well as a minimum improvement in deviation threshold, ΔD_{min}. Using a consistent set of stopping criteria made it easy to evaluate each decision tree's performance.

It should be noted that the employed CART algorithm supported two metrics for measuring the quality of a node split [108]. These include Gini impurity and entropy-based information gain (originating from information theory). Research suggested that splitting criteria will not make much difference to a tree's performance [108; 113]. Consequently, preliminary tests were conducted during implementation. These found that the information gain metric performed marginally better during data clustering, thus it was selected for use.

Information gain IG is based on the uniformity of class attributes in a set S, formed by a phonetic attribute question Q. The objective is to find the question leading to the maximum gain in information, as shown in Equation 6.3.2. Here, IG is calculated by finding the entropy difference between a parent node's set of trajectories $H(S_p)$ and its potential child nodes, $H(S_c)$. $P(S_{pc})$ is the proportion of visemes in each child node to the number of visemes in the parent node. Entropy H, is dependant on the portion of elements of a class $P(x_i)$ appearing in a node's set, as outlined in Equation 6.3.3. In our case, the node's class domain $dom(X)$ has two states, which are assigned to its visemes by the k-means algorithm.

$$Q_{opt} = \underset{Q}{\operatorname{argmax}}\ IG(Q,S) = H(S_p) - \sum_{j\in(Q)} P(S_{pcj})H(S_{cj}) \tag{6.3.2}$$

where

$$H(S) = -\sum_{x_i \in dom(X)} P(x_i) \log_2 P(x_i) \tag{6.3.3}$$

6.4 Maximum Log-Likelihood Algorithm

Audio speech synthesis and recognition research has demonstrated that context dependent phone instances can be clustered using decision trees which assess their node splits using a likelihood criterion. For example, Young *et al.* [114] developed such a probabilistic tree, which maximised the change in log-likelihood in subsets resulting from phonetic attribute questions. Bahl *et al.* [115] give a partial proof for this goodness-of-split evaluation, which was also adopted by Willet *et al.* [116] for further improving tree-based state clustering.

The common form of this algorithm's expression can be seen in Equation 6.4.1. Here, the maximum log-likelihood gain is found by splitting a node's set using each possible phonetic attribute question, Q. Gain is calculated by comparing the change in log-likelihoods between a parent node's set of trajectories, $\log L(S_p)$, to each potential yes and no child node subsets, $\log L(S_{cy})$ and $\log L(S_{cn})$ respectively. As mentioned, this approach is referred to as the SS-CART algorithm.

$$Q_{opt} = \underset{Q}{\operatorname{argmax}}((\log L(c_y) + \log L(c_n)) - \log L(p)) \tag{6.4.1}$$

To cluster our dynamic visemes, multivariate Gaussian PDFs were used to model the log-likelihood of subsets resulting from each phonetic attribute question split. The question with the maximum log-likelihood improvement is identified, and then returned to the decision tree training algorithm. The phonetic attributes question is then used to split the parent node, and the operation is repeated on the child nodes.

Equation 6.4.2 expresses the likelihood PDF for multivariate Gaussian distributions [117]. In our case, x is a dynamic viseme's trajectory, therefore $N : 1 \rightarrow 10$. In Equation 6.4.2, Σ is the $(d \times d)$ covariance matrix, $|\Sigma|$ is its determinant, and μ it a d-dimensional mean vector for each trajectory.

$$L(x_1, x_2 \ldots x_N \mid \mu, \Sigma) = \prod_{n=1}^{N} \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right) \tag{6.4.2}$$

To be consistent with the previous two algorithms, the SS-CART's stopping criteria was dependent on a minimum node occupancy count as well as a minimum improvement in deviation, ΔD_{min}. The outcome of varying these thresholds is explored in Section 7.2.

6.5 Conclusion

The three decision tree-based algorithms were employed to sort the trajectories of dynamic visemes based on their phonetic attributes. This was achieved by developing a reusable framework, which could train a decision tree using all three variations of the node splitting algorithms. Each algorithm is then assessed, as discussed in Chapter 7.

Chapter 7

Evaluation

Quantitative assessments first evaluated the performance of each algorithm in terms of mean squared error as a function of training set size. This was followed by an evaluation of the two meta parameters which control the growth of the tree decision trees. These parameters are referred to as the minimum occupancy threshold and minimum improvement in deviation threshold (calculated using Equation 6.2.3). Finally, a perceptual test is carried out to evaluate how well the algorithm with the best quantitative results compares to the baseline MD-CART algorithm.

7.1 Training Set Size

Root mean squared error (RMSE) is commonly used for comparing the similarity of synthesised visual speech parameters. In this work, RMSE comparisons were used to gauge how closely the synthesised feature trajectories resembled the measured trajectories, which are taken directly from the motion capture recordings. The RMSE was calculated by averaging the difference between the synthesized and measured trajectories every 2.4 milliseconds.

The three decision tree-based algorithms discussed in Chapter 6 were trained using varying amounts of training data. Of the 120 sentences in our dataset, 12 were randomly chosen and reserved as an independent test set, while the remaining 108 were used for training. The training set was first divided into 11 subsets containing 9 sentences each. Each subset was then used to train a decision tree-based algorithm. These were then used to synthesise trajectories for the utterances in the independent test set, from which a RMSE was calculated. Once this procedure was repeated for each subset, an averaged RMSE was found. The process described was then repeated, grouping 2, 3, 6 and lastly all 11 of the subsets for training. The minimum node occupancy and minimum improvement in deviation threshold were kept constant at 5 and 0.005 respectively during these tests.

Figure 7.1 shows averaged RMSE results as a function of the number of training sentences. Table 7.1 reveals the RMSE and standard deviation when using all 108 sentences for training.

Testing revealed that all three algorithms continuously improve as the number of training sentences increases. In Table 7.1 it can be seen that MD-CART performed marginally better than the KM-CART algorithm, but that the latter had a slightly lower RMSE standard deviation, indicating more consistent dynamic viseme clustering. The SS-CART system showed the worst synthesis performance.

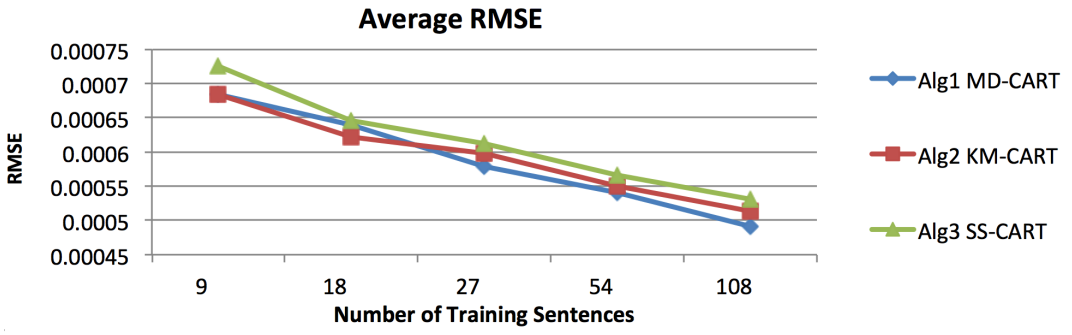


Figure 7.1: RMSE results using incremental training subsets for MD-CART, KM-CART and SS-CART algorithms.

Table 7.1: RMSE and RMSE standard deviation (STD) using all 108 training sentences to synthsise the independent test set's twelve sentences.

	MD-CART	KM-CART	SS-CART
RMSE	0.000491	0.000513	0.000531
RMSE STD	0.000238	0.000220	0.000262

In conclusion, although Figure 7.1's RMSE plot uses logarithmically increasing training set sizes, the tests do not yet show an indication of reaching a maximum synthesis accuracy. Therefore, the size of our dataset does not allow the optimal performance (in terms of RMSE) to be reached.

7.2 Meta Parameter Evaluations

This section assesses the effects of varying the decision tree's meta parameters, specifically, the minimum node occupancy threshold and minimum improvement in deviation threshold. RMSE was again used to compare the performance of the three decision tree-based algorithms relative to the measured feature trajectories.

Experiments were conducted using non-exhaustive 11-fold cross-validation (CV), with 12 sentences reserved for a final independent evaluation. For each CV fold, 10 of the 11 training subsets were used to train the decision trees. The 11^{th} subset's sentences were synthesised and used to determine a RMSE performance. This was repeated ten times, each time leaving out a different subset for synthesis. An average RMSE was then found using all 11 combinations.

The minimum node occupancy threshold was assessed first. In these tests, the minimum improvement in deviation threshold is set to zero. This value was selected because preliminary experimentation suggested that node occupancy had a far greater effect on VSS performance. Therefore, it was assumed that setting the improvement in deviation threshold to zero would be a good starting point in finding the optimal thresholds of each model.

Figure 7.2 reveals each decision tree algorithm's 11-fold CV RMSE results, which were obtained from increments in their node occupancy thresholds.

Figure 7.2 shows that the KM-CART reaches the lowest RMSE, followed by the SS-CART then MD-CART algorithms. It can also be seen that the KM-CART algorithm requires a larger node occupancy to reach an optimal performance, and that the SS-CART,

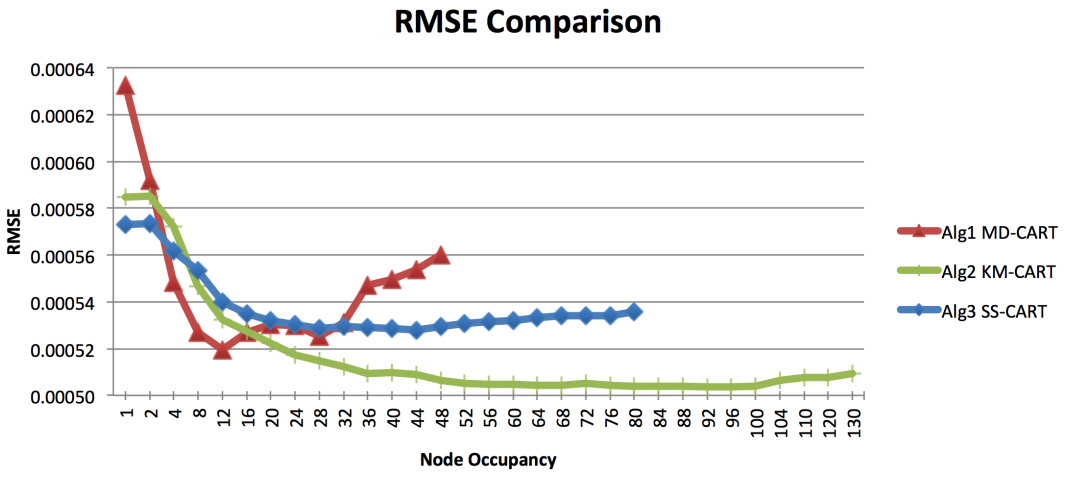


Figure 7.2: Graph showing 11-fold CV RMSE results using incremental minimum occupation criteria for MD-CART, KM-CART and SS-CART algorithms.

followed by the MD-CART algorithm, require smaller node occupancies.

Next, the minimum improvement in deviation threshold was assessed. These tests were conducted using the optimal node occupancies found by the node occupancy tests. These values were chosen so that a local RMSE minima, which considered both decision tree meta parameters, would be identified.

Figure 7.3 shows the RMSE performance results for varying the minimum improvement in deviation threshold. Again, 11-fold CV testing was employed. Note, that the threshold began at zero and uniformly increased until it prevented any vowel or consonant-based phonetic question splits (which take place in the first three tiers of each tree, as discussed in Section 6.1). The increments in deviation threshold intervals were chosen to cover a broad range of improvements, thereby quickly determining a local RMSE minima.

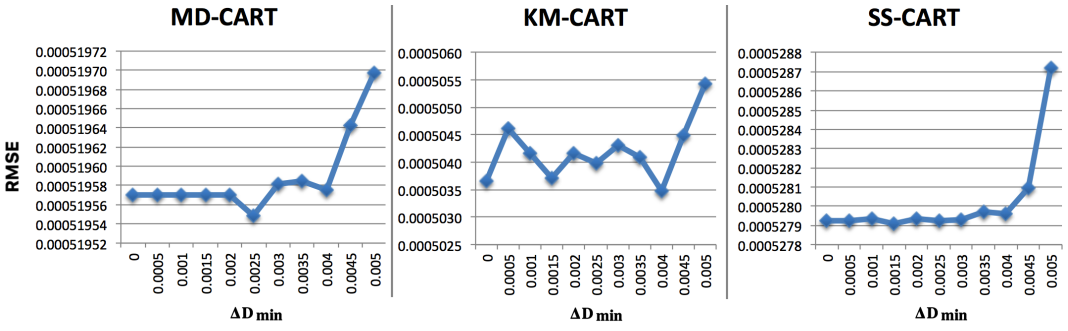


Figure 7.3: RMSE results using incremental improvement in deviation threshold criteria for MD-CART, KM-CART and SS-CART algorithms.

From Figure 7.3 it can be seen that the fluctuations in RMSE results for the MD-CART and SS-CART algorithms are smaller than those of the KM-CART algorithm. The non-deterministic nature of the KM-CART algorithm could be responsible for this behaviour. However, further informal testing revealed that there were many more areas of local RMSE minima for all algorithms, and that further testing is needed to substantiate our findings.

Finally, each decision tree-based algorithm's optimal thresholds were then used to synthesise the 12 sentences in the independent test set. Table 7.2 shows the averaged RMSE and RMSE standard deviation results for these tests.

Table 7.2: Averaged RMSEs and RMSE standard deviations (STD) for the three algorithms trained on all training sentences and synthsising only the independent test senteces. Each algorithim is trained using its stated optimal node occupancy and minimum improvement in deviation threshold (ΔD_{min}).

	MD-CART	KM-CART	SS-CART
Optimal occupancy	12	92	44
ΔD_{min}	0.0025	0.004	0.0015
RMSE	0.000479	0.000455	0.000464
RMSE STD	0.000237	0.000241	0.000236

Table 7.2's test results indicate that the KM-CART is the best performing VSS algorithm. It also indicates that the SS-CART outperforms the MD-CART algorithm. The latter test's outcome is in contrast to the 11-fold CV test results. It can therefore be argued that the MD-CART algorithm is more sensitive to changes in training data and meta parameters. This suggests that further testing is needed to assess the effects of changing meta parameters when using a larger set of data.

For the purposes of this thesis, it was concluded that finding a local RMSE minima was sufficient in assessing the working performance of the decision tree-based algorithms. Finding a global RMSE minima, for both node occupancy and the minimum improvement in deviation threshold, would be exhaustive, and would not lead to a better understanding of how the algorithms function. Following from this assumption, the results from the final independent test set concluded that the KM-CART showed the greatest improvement relative to the baseline MD-CART algorithm, and was followed closely by the SS-CART algorithm.

An important aspect overlooked by the quantitative tests above was the training time required for each algorithm. Multiple iterations of the CV tests made it clear that the KM-CART algorithm was the least time consuming to train. The SS-CART, shortly followed by the MD-CART algorithm, were found to be much slower to train.

This can be explained by the KM-CART's less involved computations. The SS-CART uses a computationally expensive log-likelihood calculation which has to be performed for each phonetic attribute question. The MD-CART algorithm requires an averaged deviation calculation for each possible phonetic attribute question. These approaches become exponentially more complex with larger training sets. The KM-CART employs the k-means algorithm, with $k = 2$, followed by a single iteration of the CART algorithm. This combination of algorithms exhibits a more linear increase in computational complexity with increased data. The greater efficiency of the KM-CART algorithm is an added benefit in under-resourced environments.

Once trained, minimal computational time is spent on traversing the decision trees and interpolating between the selected visemes. During VSS it was noticed that multiple sentences were synthesised in less than a second. It is therefore conceivable that the completed VSS system could operate in near real-time. However, further testing is needed to confirm these findings.

7.3 Perceptual Tests

Perceptual tests compared avatars animated using the KM-CART, MD-CART and motion capture recording's feature trajectories. The thresholds discovered in Section 7.2 were not found in time for them to be employed in perceptual tests. Instead, the avatars were animated using the feature trajectories synthesised by the decision trees trained in Section 7.1, utilising the full training dataset.

The avatar animated by the SS-CART algorithm was not included in the perceptual tests. This was because its RMSE results were consistently worse than those of the KM-CART algorithm, making it unlikely to be more visually compelling. To test all combinations of algorithms would require $12 \times 4 = 48$ perceptual tests to be conducted. This was considered to be too time consuming and would likely reduce participant focus.

The format of the perceptual tests was based on the methods employed by other VSS authors [5; 78]. Testing required participants to watch a sequence of videos, each showing 2 of the 3 possible avatars displayed in a random order. In each video, the first avatar speaks, followed by the second, and finally both speak together, as illustrated in Figure 7.4. Participants were then asked to indicate which avatar was perceived to best articulate the spoken sentence. This was repeated three times for each sentence, allowing all combinations of avatars to be compared for each test sentence. In total, each of the test participants evaluated $12 \times 3 = 36$ videos. Participants were permitted to re-view videos before making a decision. To prevent guessing, participants could also indicate when they could not differentiate between the two avatars. Participants were also given the option to leave comments discussing their comparative observations and general thoughts. This helped to gauge if participants found the avatar's speech to be natural.

Figure 7.4: Example frame taken from a video used in the perceptual test. In this case the left and right avatars were driven by KM-CART and MD-CART algorithms, respectively. The frames show the articulation of the sound "K".

Note, that each sentence's synthesised feature trajectory was hand calibrated in the BGE's bone-driven shape key animation control script. These calibrations used linear scaling for each of the seven features, as discussed in Section 5.1.2. In this way, the VSS system's outputs are scaled to make effective use of the synthesised trajectories, thereby producing the most visually compelling avatar animations. This calibration also

reduces under-articulation, a previously addressed shortcoming associated with the use of averaged trajectories [90].

Forty randomly selected participants took part in the perceptual tests. Most were male post-graduate students of 20 to 26 years of age. All participants were fluent in English however, none were considered experts in VSS.

Figure 7.5 presents the perceptual evaluation results. When compared with the baseline MD-CART approach, the KM-CART algorithm achieves an improvement of over 10%. KM-CART also achieved a slightly more favourable assessment than MD-CART when compared with the avatar animated by the motion capture recording's feature trajectories.

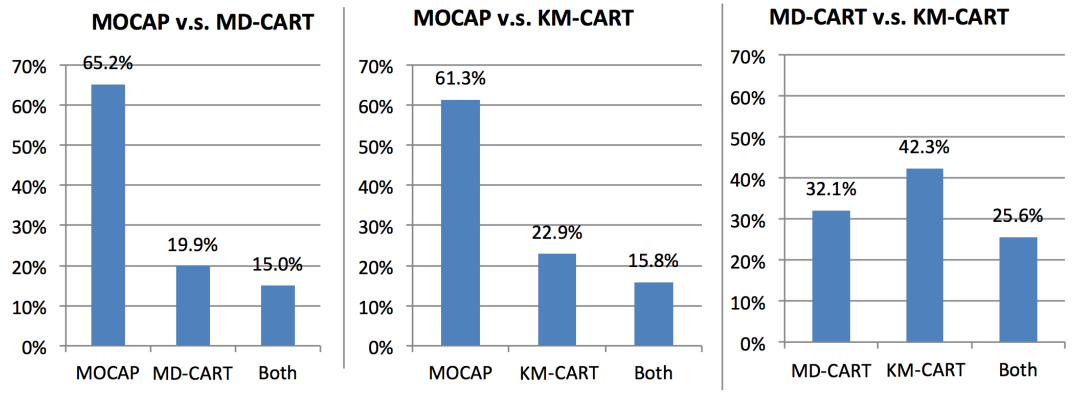


Figure 7.5: Perceptual test results showing participant preferences as a percentage. Testing compared avatars animated by motion capture data (MOCAP), MD-CART and KM-CART algorithms. The "Both" option indicates when participants could not differentiate between the two avatars.

As perceptual tests are the ultimate gauge for VSS performance, this section can conclude that the KM-CART-based VSS model is more compelling than the baseline algorithm.

The results of the perceptual experiments question the suitability of RMSE as a metric for gauging VSS performance. The RMSE results, seen in Table 7.1, indicate that the trajectories synthesized using the baseline MD-CART algorithm tend to be spatially slightly closer to the measured motion capture trajectories. However, the perceptual tests, which use the same twelve independent test sentences, indicate that the KM-CART is subjectively preferred by some margin.

These findings highlight the limitations of using RMSE to determine if a VSS model is synthesising suitably realistic speech. Developing better mathematical tools for assessing visual speech realism is clearly necessary. Consequently, the effectiveness of a similar metric to RMSE, known as the coefficient of determination (R^2), was also assessed. Appendix C details the findings of these tests. The conclusion drawn from these tests suggest little insight is given by coefficient of determination. This was attributed to the calculations still being based on mean squared errors.

Further work is also suggested to evaluate if our VSS model performs better than the stochastic modelling techniques mentioned in Chapter 4. These evaluations would also need to include comparative perceptual tests.

7.4 Discussion of Evaluation

When reading through participant comments and engaging them in post-perceptual test conversations, it was found that non-binary answers to the question, "is the avatar's speech realistic", were always provided. This made it clear that speech perception is a complex topic to convey. Consequently, participant evaluations are related to concepts such as intelligibility, realism and the Uncanny Valley effect, as discussed here.

Under-articulation was the most common fault identified by participants assessing VSS intelligibility. As discussed in Sections 4.2.5 and 5.3.2, the use of an averaged trajectory from a node can result in the more extreme measured excursions not being followed. This was occasionally noticed by participants, with comments such as "the lips look lazy" or "the pronunciation of the sound is not that obvious". Under-articulation is also identifiable by inspection of the synthesised trajectories. For example, the synthesised trajectories, seen as dashed lines in Figure 7.6, are consistently below the measured trajectories, seen as continuous solid lines. This indicates consistent under-articulation for all our decision tree-based algorithms.

In Figure 7.6, the shaded band identifies an area where a bilabial plosive 'P' phoneme is articulated. Here, the bottom lip's closure is most clearly achieved by the KM-CART algorithm, as seen in the central panel of the Figure. Correct clustering and selection of this critical visual speech cue bodes well for the KM-CART algorithm, and may suggest why it was often perfected in the perceptual tests.

Addressing under-articulation is a difficult problem. Our solution was to hand calibrate the avatar's scripted control system, as mentioned in Section 5.1.2. The process of hand calibrating the scripted bone-driven shape key animation was subject to the discretion of the author. As these tests were not supervised by someone familiar with visual speech, such as an animation artist, the style and consistency of the calibrations may have varied.

As an alternative approach, it may be possible to ensure that important visual cues are correctly articulated by employing rule-based visual speech models. These could apply a set of fixed rules to ensure full lip closure for more critical visual speech cues, such as P/M/B phoneme utterances.

Synchronicity of the visual speech was also considered to be of great importance. In practice, participant's feedback largely suggested that the audio and visual speech's synchronicity was good for all algorithms. Such asynchronicity may be due to the time-scaling of the sampled dynamic viseme trajectories, which was necessary to fit synthesis-time phonetic segmentation. Mistakes were, however, most often noticed during side-by-side viewings of the avatars. A possible reason for some of the differences notice here may be because of the software used to create the perceptual test recordings, which was limited it its ability to exactly align the audio clips of the two VSS recordings. This occasionally created a very slight echo during simultaneous viewings, which may have affected some comparative tests.

The smoothness of the visual speech trajectories was also noted during perceptual tests. Sentences were sometimes perceived as having unnatural jerky lip movements, with this reaction correlating with more phonetically rich sentences. The interpolation function, discussed in Section 5.3.2, was designed to avoid sharp inflection/turning points during trajectory interpolation. This problem may have been compounded by deviations from perfect monotonicity in the uttered speech. However, such ideal articulation is very hard to produce in practice. Another cause for jerkiness may have been the variations in the selected sequences of dynamic visemes. Further testing on the the selection and

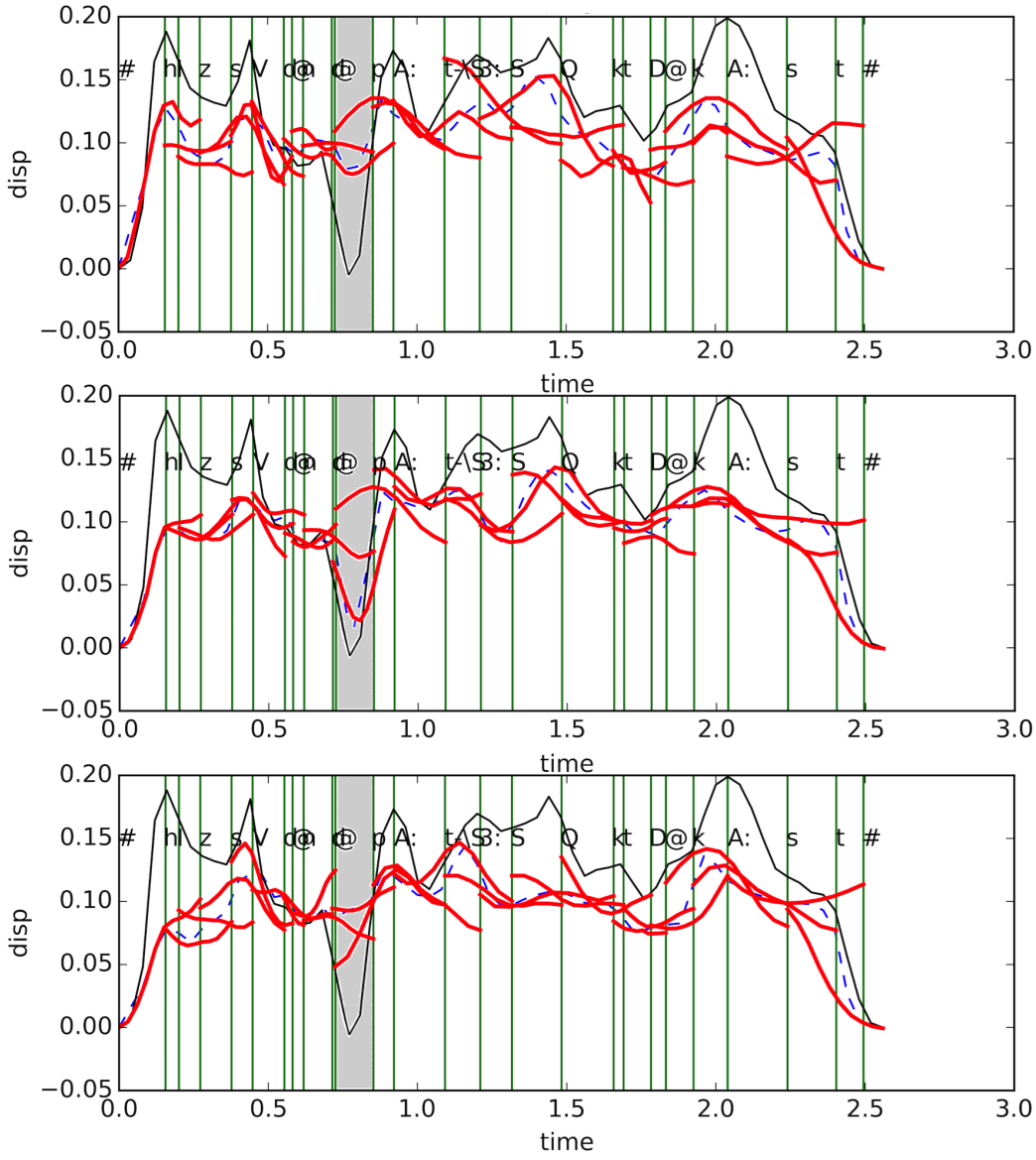


Figure 7.6: Synthesised bottom lip feature trajectories for the test sentence "His sudden departure shocked the cast". From top to bottom, the graphs are for the MD-CART, KM-CART and SS-CART algorithms. The continuous dashed lines (blue) are the final synthesised trajectories, the short lines (red) are the triphone-based dynamic visemes and the continuous solid lines (black) are the measured motion capture feature trajectories. The vertical green lines indicate phone boundaries.

interpolation functions is needed to better understand this. A related issue was discussed in Section 2.4.3, in particular, the importance of avoiding overly active lips.

The mixed reception of identical avatar speech segments by different people demonstrated that perceptions of visual speech vary greatly. This was also expressed by Bregler *et al.* [98], who concludes that a deeper insight into visual speech perception is necessary to design assessments which allow participants to give more informative feedback. However, user comments regarding speech realism were predominately positive, suggesting that although synthesised speech was not perfect, it was still intelligible, and at least demonstrated progress towards life-like VSS. These comments are reflected in the perceptual test results, seen in Figure 7.5, where at least 35% of people either could not

tell the difference or preferred the synthesised avatar's speech.

Few participants found the avatar disturbing, suggesting that the character generally avoided the Uncanny Valley effect. Those that found the interactions eerie attributed the discomfort to the avatar's lack of movement and dead-pan gaze. This reaction to the avatar's irregular social behaviour may be explained by expectation violation theory [29], discussed in Section 2.4.2. The character's control system does, however, allow for other facial features to be animated, the effects of which should be examined in future work.

It should also be noted that most of the test subjects had had exposure to conversational characters in computer games. What effect this exposure had on the participants is unknown. In addition, it was noticed that female subjects were more sensitive to unnatural visual speech cues, and were marginally more critical of the characters appearance.

Feedback from viewers observing the avatar driven by the motion captured speech feature trajectories gave insightful feedback with regards to some initial limitations of the avatar and its visual speech. For example, some subjects found that the talking style of the avatar to be unusual as its mouth movements were occasional asymmetric and it had an unusual accent. These problems were perhaps compounded by the tongue being stationary. Using a professional voice actor with more visually symmetric speech and a consist pronunciation style may minimise these discrepancies, and would likely improve the avatars perceived speech. It is possible that this may also lead to better data clustering as feature trajectories might be more consistent.

7.5 Conclusion

The evaluation of the decision tree-based VSS systems lead to two conclusions. Firstly, the adopted MD-CART baseline algorithm was demonstrated to successfully and effectively utilise dynamic visemes. Secondly, the KM-CART algorithm offered the greatest improvement over the baseline. This was shown in perceptual tests. The advantage of the KM-CART algorithm was its approach to clustering of the dynamic viseme trajectories. It achieved this by first finding trajectory clusters in an unsupervised way, and then maximised the reproduction of these clusters using a phonetic attribute question. The result was that tree leaf nodes were associated with more homogeneous datasets which in turn led to improved trajectory synthesis.

It was also found that effective evaluations can use mathematical metrics for performance assessments, for example RMSE or coefficient of determination. However, RMSE did not agree fully with the results of perceptual tests. This indicates that the factors important in human assessment of facial gestures are not well modelled by simple metrics like RMSE. Perceptual tests, however, require a lot of effort to conduct, therefore, a more informative computational VSS evaluation method remains a worthwhile research objective.

Chapter 8

Summary and Conclusion

This chapter discusses the contributions of this work, its importance and the future of VSS research and applications.

8.1 Summary and Conclusion

We presented three decision tree-based models which map the relationship between the motion of the mouth's features and their corresponding sounds produced during natural speech. This involved the development of an avatar, which can use time-alighted phonetic input to animate the mouth movements, thereby synthesising visual speech.

The final avatar's appearance and control system were a result of the chosen open source software packages Blender and MakeHuman. Scripted bone-driven shape key animation was identified as the best tool for controlling an avatar in the BGE because it can be calibrated and re-used to control any character produced by MakeHuman. Coupled with the customised motion capture system, the virtual character could either be re-animated using measured feature trajectories alone or it can articulate new speech using any one of our VSS models.

The tests conducted in Chapter 5 then examined the effects of viseme length and joining techniques. Here, triphone-based segmentation was found to be optimal for preserving coarticulation effects. Experimentation with joining the overlapping viseme trajectories also led to the design of an interpolation function which effectively conserved coarticulation.

Motivated by previous published research, we chose probabilistic modelling to be superior to gestural VSS models, because they automatically discover the relationship between audio and visual speech. The disadvantage to stochastic VSS models is their tendency to be data greedy. This made decision tree-based VSS a good choice, as it can effectively utilise sparse data sets, and can provide predictions for missing data points.

Our baseline MD-CART algorithm was an adaptation of an existing decision tree-based VSS model, made to accommodate dynamic visemes. The theory upon which the MD-CART algorithm is based (discussed in Section 6.2) was chosen because it showed the greatest potential for success by having been applied in believable video-realistic VSS. This algorithm was then developed further, resulting in the KM-CART and SS-CART algorithms.

The success of this work is detailed in Chapter 6, were both training set size and optimal meta parameters were investigated. All algorithms were proven to be capable

of VSS. The KM-CART provided the most convincing improvement over the baseline MD-CART algorithm.

Evaluations also suggested that all the VSS systems had one common issue, their synthesised trajectories were often not reaching the same magnitude of displacement as the measured trajectories. This resulted in the virtual avatar under-articulating its sentences. This finding was attributed to the use of averaged values for the dynamic visemes.

Another key finding to this work was the limitation of RMSE evaluations. RMSE comparisons of the three decision tree algorithms during varying training set size indicated that the MD-CART performed best, however, perceptual tests found the KM-CART to be preferable. A similar square error metric, coefficient of determination (R^2), was used in an attempt to improve on RMSE's limited performance indications, however, it was found to give no further insight. These finding suggest that RMSE and R^2 evaluations alone are not sufficient for evaluating the goodness of VSS. Perceptual tests are therefore the best and only analytical process which confidently infer how well VSS systems perform.

Our use of open source software and off the shelf equipment make the developed VSS models attractive to low-resourced work environments. The chosen software packages and the control system used to animate characters will also bode well for low-resource work environment. This is because they can generate and control multiple variations of human-like characters with relative ease using today's standard desktop computer. This satisfies the need for a cost effective and versatile VSS system. The use of X-SAMPA also opens these opportunities further by allowing for VSS research in different languages.

In conclusion, this work successfully demonstrated the mastery of a multi-disciplinary research topic. The solutions posed, while not perfect, diligently met the necessary objectives, resulting in the development of a unique approach to VSS, the significance of which is discussed below.

8.2 Contributions

Previous works suggested that the use of static visemes limit VSS because they do not preserve coarticulation effects. Our work supported these views as it successfully demonstrated that longer viseme units employed in VSS models are better received. In the process of achieving this, our work has contributed two unique approaches to working with visemes comprised of triphone-based time-varying oral poses.

The unique approaches of the baseline MD-CART, KM-CART and SS-CART algorithms resulted in them being considered as the most relevant contribution of this work. The KM-CART being of greatest significance based on its improved performance over the baseline MD-CART and SS-CART algorithms. By these findings, the KM-CART is considered to be a more informed system because it first determines the best possible clusterings and then tries to reproduce this 'best split' based on phonetic attributes questions.

The limitations of the RMSE and R^2 as performance metrics was also considered an important contribution to VSS evaluations. The impact of this finding is far reaching as it implies that perceptual test of avatars are the only reliable tests, and that there is a need for more advanced quantitative feedback algorithms, which could aid in system training prevent the need for many comparative tests.

Lastly, our contribution with the broadest impact, was the explanation of how open source software packages Blender and MakeHuman can be used to create a wide range of conversational agents. The corresponding control system used to animate the avatars

is also of significance as its versatility provides a platform for any person to puppeteer a avatar's movements.

8.3 Application of Work

It is necessary to consider the different environments in which VSS can be applied.

Automated visual speech has the greatest potential for use in computer games. Here, VSS can serve as an effective tool for achieving speech animation. VSS models can also be extended to virtual assistants, operating in both public and private settings. A virtual assistant can aid in speech dependant tasks, such as teaching people a new language, communicating information in a noisy environment or just being helpful with articulating text, such as e-mails. Research has indicated audiovisual speech improves speech intelligibility and also boost levels of engagement, both of which are desirable attributes for any HCI system [71].

Another application of VSS is in animating the visual speech of digital movie characters. It should be noted that the nature of animated movie characters heightens the need for diversity and realism of oral movements. For such a VSS system to be effective, it implies the need for a more divers training set, which may required a mix of over-exaggerated speech, speech with emotion and non-speech related oral gestures. As the current system is implemented in a game engine, development of an animation tool which can output visual speech sequences as keyframes in a film scene is required. But, if the current VSS tool were made compatible, it would make for a much more efficient means to animate character dialogue.

In cases where VSS of new sentences is not needed, sole usage of the motion capture technique developed in Section 5.1 has demonstrated the potential for efficiently reproducing visual speech in animated characters. Therefore, instead of using the VSS system, a voice actor's speech and feature trajectories can be captured and their character animated directly through the motion capture control system. This process is advantageous over regular motion capture systems as it is based on shape key animation, which allows easy control of any character created by the freely available MakeHuman software. Our motion capture system is also easy to set up as it requires few markers to be placed on the face.

A future use of the VSS system could include encoded visual speech transmissions for avatar based online communication. Internet video telephony (or voice over internet providers) allow people to talk face-to-face, but this requires a large amount of data to be transferred. With a voice to phonetic translation system coupled to a VSS system, only the voice transmission is required. This is because the VSS system can use the phonetic translations to animate an avatar at the user's end. Therefore, less data transmission is required for avatar-to-face based online communication. This could also be made more engaging by allowing users to customise their avatars. This tool would likely be popular for online gaming and social networks. Chen *et al.* [73] also discuss this idea, and its applicability to reducing face-to-face communication's data transmissions when using video-realistic avatars.

Lastly, the potential usage of the time-series data clustering techniques developed in this work may also be applicable to other areas of research. For example, the clustering of time-series data instances is applicable to domains such as traffic management and transportation optimisation, ecological studies of animals motions/migrations and many areas of stock market analysis [111].

8.4 Future work

To conclude this thesis, we will identify aspects of the presented work for which further research is applicable.

8.4.1 Data Capture

Potential developments for data capture falls into three categories: increased accuracy of data capture; increased diversity of data capture; and automation of data capture.

Accuracy can be improved by refining the data capture procedure. Some possible avenues include: using different markers, different marker arrangements, marker-less tracking systems and developing different tracking or marker detection sensors. The most beneficial improvements would be those which reduce audio and visual signal noise, especially for captured feature trajectories.

Combining accuracy with freedom of head movement is ideal for capturing natural face-to-face conversation. Therefore, creating a wearable capture system, or using a system that can track movement at a distance, are also worthwhile research topics. Note, that there are many different technologies capable of detailed and accurate face tracking which could be purchased for VSS-related data capture.

VSS work would also benefit from research which indicates an optimal set of training sentences, which cover the most important words or syllables. Ma *et al.* [82] applied an approach based on these principles by identifying the most frequently occurring syllables in multisyllabic words and the most frequent words, based on the TIMIT corpus, and recording them as part of their VSS system's training data. It was unclear, however, what the effect of this was on improving the system's synthesis performance.

Increasing the diversity of the recorded data provides new dimensions along which to develop our avatar. The use of X-SAMPA, for example, provides the potential to extend our approach to multi-lingual VSS. The suitability of different tree-based algorithms to cluster and synthesise different languages could also be investigated.

Research could also consider capturing and synthesising oral gestures that are not related to language, but to communication in general. Previous work has attempted to automate voice driven animation of non-speech articulation, such as laughing, crying, sneezing and yawning, but much work still remains [118]. Non-speech acts are not accounted for by X-SAMPA. To resolve this incompatibility, further research is needed to incorporate peripheral oral audio and visual gestures.

Of all the features that could be captured to extend this work, capturing information about the movement of the tongue would be the most beneficial. The MOCHA-TIMIT database [119] is one of the few freely available datasets of tongue movements. This database could be adapted to use X-SAMPA for compatibility with our decision tree algorithms. While the database is free, reproducing the tongue tracking procedure is not feasible for low-resourced environments as it makes use of electromagnetic articulography.

Lastly, research into tools that automate aspects of data capture would be highly beneficial to the VSS community. Whilst there are open source speech-to-text engines available, no system exists which automates speech to time-aligned X-SAMPA phonetic annotations. However, previous work has automated time-aligned phonemic labelling of speech [82; 96; 98], thus it is feasible that such a tool can be created for X-SAMPA transcriptions.

8.4.2 Clustering Algorithms

It is common knowledge that visual speech varies enormously depending on the speakers' unique anatomy and linguistic style, speech rate, facial expressions and the language spoken. Cohen *et al.* [58] concluded that visual speech has no single theory or model that could account for the variations in coarticulation for all situations. Therefore, further development of VSS algorithms is always possible.

To further develop the decision tree-based time-series clustering algorithms, Aggarwal [111] suggests they should account for more specific properties which are related to the nature of the time-series data. Such properties could include:

- Assessment of the data's dimensionality.
- Integrating or differentiating trajectories and performing related line gradient operations.
- Detecting and reducing the presence of noise, particularly in the tracked feature trajectories.
- Identification of systemic indifferences in time-series data, including discrepancies in the synchronicity of trajectories. This could include improved trajectory shape comparators, for example dynamic time warping.
- Decision tree pruning algorithms.
- Hybrid decision tree-based algorithms, which combine the best performing aspects of the MD,KM and SS-CART algorithms.
- The use cost functions to analyse and select more preferable trajectories represented by nodes.

Finally, future research could focus on extending the proposed algorithms to allow visual speech to be synthesised directly from audio. This has been shown to be a difficult but more efficient approach to VSS because it is no longer necessary to present time-aligned phonetic sequences to the system. The literature describes attempts to analyse audio signals and identify information, such as stress, pitch or energy, which are indicative of certain visual speech cues [77; 81; 98]. Much work still remains, however, before audio spectral analyses effectively relates sound to mouth movements.

8.4.3 Character Animation

Character animation covers a large body of research, including topics such as enhanced graphic realism and full body animation. However, only work related to conversational agents will be discussed here. These aspects include enhancing the face's control system and implementing full face animation.

Scripted bone-driven shape key animation has a number of control problems associated with it. These include a lack of feedback that can prevent unnatural facial gestures or severe asymmetry in the face, the need for an automated calibration system for scaling the feature trajectories controlling the shape key animations, and the need for additional shape key animations to allow for wrinkles, dimples or other facial features to be animated.

A feedback system is particularly useful for avoiding the many undesirable animation possibilities that our system does not prevent. For example, the penetration of colliding

surfaces/facial tissue and the production of unnatural mouth or face movements. Enhancing the VSS control system of our avatar to provide full face animation opens up many new opportunities for HCI-related research.

8.4.4 Other Future Research Options

Our project research found that Blender has the potential to control external motors based on the movement of its content. Therefore there is potential to create animatronic characters, like those seen in Section 3.2.7. By puppeteer robots using a character based in the BGE, human-robotic interfaces to be compared to human-computer based interactions. This would be of great significance as there is little research comparing HMI to HCI.

Another future development of the work presented here is to couple it with a TTS engine. If the output of the TTS system includes time-aligned phonetic labels, VSS could be performed for new target sentences. Mattheyses *et al.* [6; 67] achieve this using their own customised TTS engine coupled with a photo-realistic avatar.

8.4.5 Future Evaluations

We have considered RMSE and R^2 as metrics for assessing VSS performance. However, our investigation showed that more advanced evaluation techniques are needed. Such assessment techniques might be based on an analysis of common VSS errors, and thereby also determine the causality of the errors.

Further testing may also include the psychological effects of interacting with our character when using the different VSS models. For example, can they reproduce the McGurk effect or, do they have the ability to improve the intelligibility of speech in a noisy environment. Another possible evaluation might establish whether the avatar allows lip reading. This may also lead to improved techniques to teach lip reading.

Bibliography

[1] Agelfors, E., Beskow, J., Granström, B., Lundeberg, M., Salvi, G., Spens, K.-E. and Öhman, T.: Synthetic visual speech driven from auditory speech. In: *AVSP-International Conference on Auditory-Visual Speech Processing*, pp. 123–127. 1999.

[2] Summerfield, Q.: Lipreading and audio-visual speech perception. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 335, no. 1273, pp. 71–78, 1992.

[3] Liu, K. and Ostermann, J.: Realistic talking head for human-car-entertainment services. In: *Proceedings IMA 2008 Informationssysteme für mobile Anwendungen*, pp. 108–118. Braunschweig, Germany, 2008.

[4] Osipa, J.: *Stop staring: facial modeling and animation done right.* 3rd edn. Wiley Pub, Indianapolis, IN, 2010.

[5] Taylor, S.L., Mahler, M., Theobald, B.-J. and Matthews, I.: Dynamic units of visual speech. In: *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*, pp. 275–284. Eurographics Association, 2012.

[6] Mattheyses, W., Latacz, L. and Verhelst, W.: Comprehensive many-to-many phoneme-to-viseme mapping and its application for concatenative visual speech synthesis. *Speech Communication*, vol. 55, no. 7-8, pp. 857–876, 2013.

[7] Rademan, C.F. and Niesler, T.: Improved visual speech synthesis using dynamic viseme k-means clustering and decision trees. In: *In: FAAVSP'15-The 1st Joint Conference on Facial Analysis, Animation, and Auditory-Visual Speech Processing*, pp. 169–174. 2015.

[8] MakeHuman team: Makehuman | Open source tool for making 3d characters. `http://www.makehuman.org/`, 2005. [Online; accessed 11-Nov-2015].

[9] Blender.org: Shape keys. `http://wiki.blender.org/index.php/Doc:2.4/Manual/Animation/Techs/Shape/Shape_Keys`, 2011. [Online; accessed 3-Feb-2014].

[10] Bouman, C.A.: Lab 9a - Speech Processing (part 1). `http://cnx.org/contents/059da0cc-99cf-4701-a4cb-1c5a4764b3c8@3/Lab-9a---Speech-Processing-par`, 2009. [Online; accessed 11-Nov-2015].

[11] Simunek, M.: Visualization of talking human head. `http://www.cescg.org/CESCG-2001/MSimunek/paper.pdf`, 2001. [Online; accessed 11-Nov-2015].

[12] International Phonetic Association (ed.): *Handbook of the International Phonetic Association: a guide to the use of the International Phonetic Alphabet.* Cambridge University Press, Cambridge, U.K., 1999.

[13] Grant, K.W., Walden, B.E. and Seitz, P.F.: Auditory-visual speech recognition by hearing-impaired subjects: Consonant recognition, sentence recognition, and auditory-visual integration. *The Journal of the Acoustical Society of America*, vol. 103, no. 5, pp. 2677–2690, 1998.

[14] Fisher, C.G.: Confusions Among Visually Perceived Consonants. *Journal of Speech, Language, and Hearing Research*, vol. 11, no. 4, pp. 796–804, 1968.

[15] Martin, G.C.: Preston blair phoneme series. `http://www.garycmartin.com/mouth_shapes.html`, 2013. [Online; accessed 11-Nov-2015].

[16] Krňoul, Z., Železnỳ, M., Müller, L. and Kanis, J.: Training of coarticulation models using dominance functions and visual unit selection methods for audio-visual speech synthesis. In: *Proceedings INTERSPEECH 2006*, pp. 585–588. Pittsburgh, PA, 2006.

[17] Schwartz, J.-L., Berthommier, F. and Savariaux, C.: Seeing to hear better: evidence for early audio-visual interactions in speech identification. *Cognition*, vol. 93, no. 2, pp. B69–B78, 2004.

[18] Werda, S., Mahdi, W. and Hamadou, A.B.: Lip localization and viseme classification for visual speech recognition. *CoRR*, vol. abs/1301.4558, 2013.

[19] Chandrasekaran, C., Trubanova, A., Stillittano, S., Caplier, A. and Ghazanfar, A.A.: The Natural Statistics of Audiovisual Speech. *PLoS Computational Biology*, vol. 5, no. 7, 2009.

[20] Troille, E., Cathiard, M.-A. and Abry, C.: Speech face perception is locked to anticipation in speech production. *Speech Communication*, vol. 52, no. 6, pp. 513–524, 2010.

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

[22] MacDonald, J. and McGurk, H.: Visual influences on speech perception processes. *Perception & Psychophysics*, vol. 24, no. 3, pp. 253–257, 1978.

[23] Mori, M., MacDorman, K. and Kageki, N.: The Uncanny Valley [From the Field]. *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 98–100, 2012.

[24] Beane, N.: *3d animation essentials*, chap. Chapter 4 Exploring Animation, Story, and Pre-visualization. 1st edn. Wiley, Indianapolis, IN, 2012.

[25] Bartneck, C., Kanda, T., Ishiguro, H. and Hagita, N.: Is the uncanny valley an uncanny cliff? In: *The 16th IEEE International Symposium on Robot and Human interactive Communication*, pp. 368–373. IEEE, 2007.

[26] Tinwell, A. and Grimshaw, M.: Bridging the uncanny: an impossible traverse. In: *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era.*, pp. 66–73. ACM, 2009.

[27] Burleigh, T.J., Schoenherr, J.R. and Lacroix, G.L.: Does the uncanny valley exist? An empirical test of the relationship between eeriness and the human likeness of digitally created faces. *Computers in Human Behavior*, vol. 29, no. 3, pp. 759–771, 2013.

[28] Ho, C.-C., MacDorman, K.F. and Pramono, Z.D.: Human emotion and the uncanny valley: a GLM, MDS, and Isomap analysis of robot video ratings. In: *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pp. 169–176. ACM, 2008.

[29] MacDorman, K.F. and Ishiguro, H.: The uncanny advantage of using androids in cognitive and social science research. *Interaction Studies*, vol. 7, no. 3, pp. 297–337, 2006.

[30] MacDorman, K.F., Green, R.D., Ho, C.-C. and Koch, C.T.: Too real for comfort? Uncanny responses to computer generated faces. *Computers in Human Behavior*, vol. 25, no. 3, pp. 695–710, 2009.

[31] Ramey, C.H.: The uncanny valley of similarities concerning abortion, baldness, heaps of sand, and humanlike robots. In: *Proceedings of views of the uncanny valley workshop: IEEE-RAS international conference on humanoid robots*, pp. 8–13. 2005.

[32] Graf, H.P., Cosatto, E., Strom, V. and Huang, F.J.: Visual prosody: Facial movements accompanying speech. In: *Proceedings Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 396–401. IEEE, 2002.

[33] Pandzic, I.S. and Forchheimer, R.: *MPEG-4 Facial Animation: The Standard, Implementation and Applications*, chap. 9 Emotion Recognition and Synthesis Based on MPEG-4 FAPs, pp. 141–167. 1st edn. Wiley, Hoboken, NJ, 2002.

[34] Kuperberg, M., Peacock, A., Bowman, M. and Manton, R.: *A Guide to Computer Animation for TV, Games, Multimedia and Web*. Focal Press Visual Effects and Animation, 1st edn. Focal Press, Oxford, United Kingdom, 2002.

[35] Parke, F.I.: Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, vol. 2, no. 9, pp. 61–68, 1982.

[36] Kalra, P., Mangili, A., Thalmann, N.M. and Thalmann, D.: Simulation of facial muscle actions based on rational free form deformations. In: *Computer Graphics Forum*, vol. 11, pp. 59–69. Wiley, 1992.

[37] Lee, Y., Terzopoulos, D. and Waters, K.: Realistic modeling for facial animation. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 55–62. ACM, 1995.

[38] Sifakis, E., Neverov, I. and Fedkiw, R.: Automatic determination of facial muscle activations from sparse motion capture marker data. In: *ACM SIGGRAPH 2005 Papers*, vol. 24 of *SIGGRAPH '05*, pp. 417–425. ACM, 2005.

[39] Waters, K.: A muscle model for animation three-dimensional facial expression. In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, vol. 21, pp. 17–24. ACM, 1987.

[40] Albrecht, I., Haber, J., Kahler, K., Schroder, M. and Seidel, H.: " May i talk to you?:-)"-Facial Animation from Text. In: *Proceedings. 10th Pacific Conference on Computer Graphics and Applications*, pp. 77–86. IEEE, 2002.

[41] Engineered Arts limited: Engineered arts limited. `https://www.engineeredarts.co.uk/`, 2015. [Online; accessed 11-Nov-2015].

[42] Al Moubayed, S., Skantze, G. and Beskow, J.: Lip-reading: Furhat audio visual intelligibility of a back projected animated face. In: *Intelligent Virtual Agents*, vol. 7502 of *Lecture Notes in Computer Science*, pp. 196–203. Springer Berlin Heidelberg, 2012.

[43] Parke, F.I.: Computer generated animation of faces. In: *Proceedings of the ACM annual conference*, vol. 1, pp. 451–457. ACM, 1972.

[44] Ekman, P. and Friesen, W.V.: *Unmasking the face: A guide to recognizing emotions from facial clues*, chap. Chapter 3 Research on facial expressions of emotion, pp. 21–33. Malor Boooks, Los Altos, CA, 2003.

[45] Ekman, P. and Rosenberg, E.L.: *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*, chap. Introduction The Study of Spontaneous Facial Expressions in Psychology, pp. 3–18. Oxford University Press, Oxford, United Kingdom, 1997.

[46] Khronos Group: Opengl. `https://www.opengl.org/`, 2015. [Online; accessed 11-Nov-2015].

[47] Cognitive and Communication Technologies: Xface. `http://xface.fbk.eu/`, 2008. [Online; accessed 22-May-2014].

[48] De Carlo, D. and Stone, M.: Rutgers University Talking Head. `http://www.cs.rutgers.edu/~village/ruth/`, 2003. [Online; accessed 11-Nov-2015].

[49] Pelachaud, C.: GRETA: Embodied Conversational Agent. `"http://perso.telecom-paristech.fr/~pelachau/Greta/"`, 2001. "[Online; accessed 11-Nov-2015]".

[50] Institute for Creative Technologies: Virtual human toolkit. `"https://vhtoolkit.ict.usc.edu/"`, 2015. "[Online; accessed 11-Nov-2015]".

[51] Autodesk Inc.: 3d Animation And Modeling Software | Maya. `http://www.autodesk.co.za/products/maya/overview`, 2015. [Online; accessed 11-Nov-2015].

[52] Unity Technologies: Unity overview | Unity. `https://unity3d.com/unity`, 2015. [Online; accessed 11-Nov-2015].

[53] Numpy Developers: Numpy. `http://www.numpy.org/`, 2013. [Online; accessed 05-Jun-2014].

[54] Hess, R.: *Blender foundations: the essential guide to learning Blender 2.6*. Taylor & Francis, Burlington, MA, 2010.

[55] BornCG: Blender 2.6 Tutorial 38 - Shape Keys: Blinking - YouTube. `https://www.youtube.com/watch?v=gScAPFxTfv0`, 2012. [Online; accessed 21-Mar-2014].

[56] Rao, R.R. and Chen, T.: Cross-modal prediction in audio-visual communication. In: *Proceedings of the Acoustics, Speech, and Signal Processing, 1996. On Conference Proceedings., 1996 IEEE International Conference*, vol. 4 of *ICASSP '96*, pp. 2056–2059. IEEE, Washington, DC, 1996.

[57] Pearce, A., Wyvill, B., Wyvill, G. and Hill, D.: Speech and expression: A computer solution to face animation. In: *Graphics Interface*, vol. 86, pp. 136–140. 1986.

[58] Cohen, M. and Massaro, D.: Modeling coarticulation in synthetic visual speech. In: *Models and Techniques in Computer Animation*, Computer Animation Series, pp. 139–156. Springer Japan, 1993.

[59] Löfqvist, A.: Speech as audible gestures. In: *Speech Production and Speech Modelling*, vol. 55 of *NATO ASI Series*, pp. 289–322. Springer Netherlands, 1990.

[60] Pelachaud, C., Magno-Caldognetto, E., Zmarich, C. and Cosi, P.: Modelling an Italian talking head. In: *AVSP-International Conference on Auditory-Visual Speech Processing*, pp. 72–77. 2001.

[61] Caldognetto, E.M., Perin, G. and Zmarich, C.: Labial coarticulation modeling for realistic facial animation. In: *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, p. 505. IEEE, 2002.

[62] Bailly, G., Bérar, M., Elisei, F. and Odisio, M.: Audiovisual speech synthesis. *International Journal of Speech Technology*, vol. 6, no. 4, pp. 331–346, 2003.

[63] Kent, R.D. and Minifie, F.D.: Coarticulation in recent speech production models. *Journal of Phonetics*, vol. 5, no. 2, pp. 115–133, 1977.

[64] Öhman, S.E.: Numerical model of coarticulation. *The Journal of the Acoustical Society of America*, vol. 41, no. 2, pp. 310–320, 1967.

[65] Beskow, J.: Rule-based visual speech synthesis. In: *EUROSPEECH '95. 4th European Conference on Speech Communication and Technology*, pp. 299–302. 1995.

[66] Pelachaud, C., Badler, N.I. and Steedman, M.: Linguistic issues in facial animation. In: *Computer animation'91*, pp. 15–30. Springer, 1991.

[67] Mattheyses, W.: *A Multimodal Approach To Audiovisual Text-To-Speech Synthesis*. Ph.D. thesis, Vrije Universiteit Brussel, Brussels, 2013.

[68] Ma, J., Cole, R., Pellom, B., Ward, W. and Wise, B.: Accurate automatic visible speech synthesis of arbitrary 3D models based on concatenation of diviseme motion capture data. *Computer Animation and Virtual Worlds*, vol. 15, no. 5, pp. 485–500, 2004.

[69] Morishima, S., Aizawa, K. and Harashima, H.: An intelligent facial image coding driven by speech and phoneme. In: *ICASSP-International Conference on Acoustics, Speech, and Signal Processing*, pp. 1795–1798. IEEE, 1989.

[70] Linde, Y., Buzo, A. and Gray, R.M.: An algorithm for vector quantizer design. *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[71] Hong, P., Wen, Z. and Huang, T.S.: Real-time speech-driven face animation with expressions using neural networks. *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 916–927, 2002.

[72] Tao, J., Xin, L. and Yin, P.: Realistic Visual Speech Synthesis Based on Hybrid Concatenation Method. *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 3, pp. 469–477, 2009.

[73] Chen, T. and Rao, R.R.: Audio-visual integration in multimodal communication. In: *Proceedings of the IEEE*, vol. 86, pp. 837–852. 1998.

[74] Yamamoto, E., Nakamura, S. and Shikano, K.: Lip movement synthesis from speech based on hidden Markov models. *Speech Communication*, vol. 26, no. 1, pp. 105–115, 1998.

[75] Brooke, N.M.: Talking heads and speech recognisers that can see: The computer processing of visual speech signals. In: *Speechreading by Humans and Machines*, vol. 150 of *NATO ASI Series*, pp. 351–371. Springer Berlin Heidelberg, 1996.

[76] Bregler, C., Covell, M. and Slaney, M.: Video rewrite: Visual speech synthesis from video. In: *Proceedings of Audio-Visual Speech Processing: Computational & Cognitive Science Approaches*, pp. 153–156. 1997.

[77] Tamura, M., Kondo, S., Masuko, T. and Kobayashi, T.: Text-to-audio-visual speech synthesis based on parameter generation from HMM. In: *EUROSPEECH*, pp. 959–962. Citeseer, Budapest, Hungary, 1999.

[78] Hofer, G., Yamagishi, J. and Shimodaira, H.: Speech-driven lip motion generation with a trajectory HMM. `https://www.era.lib.ed.ac.uk/handle/1842/3883`, 2008. [Online; accessed 11-Nov-2015].

[79] Huang, F.J. and Chen, T.: Real-time lip-synch face animation driven by human voice. In: *Multimedia Signal Processing, IEEE Second Workshop on*, pp. 352–357. IEEE, Los Angeles, California, 1998.

[80] Brand, M.: Voice puppetry. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 21–28. ACM Press/Addison-Wesley Publishing Co., 1999.

[81] Li, Y. and Shum, H.-Y.: Learning dynamic audio-visual mapping with input-output Hidden Markov models. *IEEE Transactions on Multimedia*, vol. 8, no. 3, pp. 542–549, 2006.

[82] Ma, J., Cole, R., Pellom, B., Ward, W. and Wise, B.: Accurate visible speech synthesis based on concatenating variable length motion capture data. *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 2, pp. 266–276, 2006.

[83] Kriesel, D.: A Brief Introduction to Neural Networks. `http://www.dkriesel.com/en/science/neural_networks`, 2009. [Online; accessed 11-Nov-2015].

[84] Morishima, S. and Harashima, H.: A media conversion from speech to facial image for intelligent man-machine interface. *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 4, pp. 594–600, 1991.

[85] Öhman, T. and Salvi, G.: Using HMMs and ANNs for mapping acoustic to visual speech. *Speech, Music and Hearing - Quarterly Progress and Status Report*, vol. 40, no. 1-2, pp. 45–50, 1999.

[86] Ström, N.: Phoneme probability estimation with dynamic sparsely connected artificial neural networks. *The Free Speech Journal*, vol. 5, pp. 1–41, 1997.

[87] Lavagetto, F.: Converting speech into lip movements: A multimedia telephone for hard of hearing people. *IEEE Transactions on Rehabilitation Engineering*, vol. 3, no. 1, pp. 90–102, 1995.

[88] Curinga, S., Lavagetto, F. and Vignoli, F.: Lip movements synthesis using time delay neural networks. In: *Proceedings EUSIPCO*, vol. 96, pp. 999–1002. 1996.

[89] Rao, R.R. and Chen, T.: Cross-modal predictive coding for talking head sequences. In: *Multimedia Communications and Video Coding*, pp. 301–308. Springer, 1996.

[90] Galanes, F.M., Unverferth, J., Arslan, L.M. and Talkin, D.: Generation of lip-synched synthetic faces from phonetically clustered face movement data. In: *AVSP-International Conference on Auditory-Visual Speech Processing*, pp. 191–194. 1998.

[91] Chuang, E. and Bregler, C.: Performance driven facial animation using blendshape interpolation. Tech. Rep., Stanford University, 2002.

[92] Brown, D.: Tracker video analysis and modeling tool. `https://www.cabrillo.edu/~dbrown/tracker/`, 2014. [Online; accessed 03-Jun-2014].

[93] De Martino, J.M., Pini Magalhães, L. and Violaro, F.: Facial animation based on context-dependent visemes. *Computers & Graphics*, vol. 30, no. 6, pp. 971–980, 2006.

[94] Pelachaud, C.: *Communication and coarticulation in facial animation*. Ph.D. thesis, University of Pennsylvania, 1991.

[95] Lazalde, O.M. and Maddock, S.C.: Comparison of Different Types of Visemes using a Constraint-based Coarticulation Model. In: *Theory and Practice of Computer Graphics*, pp. 199–206. The Eurographics Association, 2010.

[96] Cao, Y., Tien, W.C., Faloutsos, P. and Pighin, F.: Expressive speech-driven facial animation. *ACM Transactions on Graphics*, vol. 24, no. 4, pp. 1283–1302, 2005.

[97] Toutios, A., Musti, U., Ouni, S., Colotte, V., Wrobel-Dautcourt, B. and Berger, M.-O.: Setup for acoustic-visual speech synthesis by concatenating bimodal units. In: *Interspeech*, pp. 486–489. ISCA, Makuhari, Chiba, Japan, 2010.

[98] Bregler, C., Covell, M. and Slaney, M.: Video rewrite: Driving visual speech with audio. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pp. 353–360. ACM Press, ACM Press/Addison-Wesley Publishing Co., New York, NY, 1997.

[99] Kshirsagar, S. and Magnenat-Thalmann, N.: Visyllable based speech animation. In: *Computer Graphics Forum*, vol. 22, pp. 631–639. 2003.

[100] Risberg, A. and Lubker, J.: Prosody and speechreading. *Speech Transmission Laboratory Quarterly Progress Report and Status Report*, vol. 4, pp. 1–16, 1978.

[101] Taylor, S., Theobald, B.-J. and Matthews, I.: The effect of speaking rate on audio and visual speech. In: *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 3037–3041. IEEE, 2014.

[102] Cao, Y., Faloutsos, P., Kohler, E. and Pighin, F.: Real-time speech motion synthesis from recorded motions. In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '04, pp. 345–353. Eurographics Association, 2004.

[103] Klompje, G.: *A parametric monophone speech synthesis system*. Ph.D. thesis, University of Stellenbosch, 2006.

[104] Bevacqua, E. and Pelachaud, C.: Expressive audio-visual speech. *Computer Animation and Virtual Worlds*, vol. 15, no. 3-4, pp. 297–304, 2004.

[105] Theobald, B.-J., Wilkinson, N. and Matthews, I.: On evaluating synthesised visual speech. In: *International Conference on Auditory-Visual Speech Processing*, pp. 7–12. 2008.

[106] Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G. and Pallett, D.S.: Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon Technical Report N*, vol. 93, 1993.

[107] Breiman, L., Friedman, J., Olshen, R. and Stone, C.: *Classification and Regression Trees*, chap. 2.4 Initial tree growing methodology, pp. 74–93. 1st edn. Chapman & Hall/CRC Press, Boca Raton, FL, 1984.

[108] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. *et al.*: Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[109] Witten, I.H. and Frank, E.: *Data Mining: Practical machine learning tools and techniques*, chap. 4.3 Divide-and-conquer: Constructing decision trees, pp. 97–119. 2nd edn. Morgan Kaufmann Publishers Inc., Burlington, MA, 2005.

[110] Quinlan, J.R.: *C4. 5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2014.

[111] Aggarwal, C.C.: *Data classification: algorithms and applications*, chap. 15 Time Series Data Clustering, pp. 356–377. CRC Press, Boca Raton, FL, 2014.

[112] Gan, G., Ma, C. and Wu, J.: *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*, chap. 9 Center-based Clustering Algorithms, pp. 161–182. Siam, Philadelphia, PA, 2007.

[113] Rokach, L. and Maimon, O.: *Data mining with decision trees: theory and applications*. 2nd edn. World Scientific Publishing, Hackensack, NJ, 2015.

[114] Young, S.J., Odell, J.J. and Woodland, P.C.: Tree-based state tying for high accuracy acoustic modelling. In: *Proceedings of the Workshop on Human Language Technology*, HLT '94, pp. 307–312. Association for Computational Linguistics, Stroudsburg, PA, 1994.

[115] Bahl, L.R., de Soutza, P.V., Gopalakrishnan, P.S., Nahamoo, D. and Picheny, M.A.: Context dependent modeling of phones in continuous speech using decision trees. In: *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, pp. 264–269. Association for Computational Linguistics, Stroudsburg, PA, 1991.

[116] Willet, D., Neukirchen, C., Rottland, J. and Rigoll, G.: Refining treebased state clustering by means of formal concept analysis, balanced decision treesand automatically generated model-sets. In: *Proceedings of the International Conference on Acoustics, Speech, Signal Processing*, pp. 565–568. 1999.

[117] Jones, E., Oliphant, T. and Peterson, P.: SciPy: Open source scientific tools for Python. `http://www.scipy.org/`, 2001. [Online; accessed 05-Oct-2015].

[118] Cosker, D. and Edge, J.: Laughing, crying, sneezing and yawning: Automatic voice driven animation of non-speech articulations. pp. 225–234. 2009.

[119] Wrench, A.: A new resource for production modelling in speech technology. *Proceedings of the Workshop on Innovations in Speech Processing*, vol. 23, no. 3, pp. 207–218, 2001.

[120] Gujarati, D.N. and Porter, D.C.: *Essentials of econometrics*. McGraw-Hill, New York, NY, 1999.

Appendices

Appendix A

Scripted Bone-Driven Shape Key Animation

Scripted bone-driven shape key animation uses bones as a proxy to control shape keys. This is achieved with the use of Blenders' animation drivers, which links the movement of a bone to a shape key animation.This process is implemented in the BGE as follows:

- To start, a fully rigged virtual character is needed in the 3D view whilst the BGE is in "Blender Render's" "object mode". The character must include a completed armature and set of shape key animations (terms introduced in Section 3.4.2).
- Any simple transparent mesh object, such as a small cube, must first be created. Eventually, this cube will act like a slider for adjusting the shape key animation. The cube must be "parented" to a new bone, which can be created inside it. The cube is created because bones acting as shape key drivers are not visible whilst in "Pose Mode". Once the new bone is associated with the cube, it must be "parented" to the rest of the rigged character's armature. For ease of control, the bone can be "constrained" to move only a set direction and distance.
- To complete the bone-driven shape key animation set up, the character's mesh must be selected, going back into "object mode". Then, selecting the desired shape key in the shape key's panel, right click on its "value" and select "add driver". The shape key should then turn purple, as seen in on the right side of Figure A.1. In the "graph editor" , in "drivers", the new driver for the shape key can be found. Selecting the driver allows the types of transformations effecting the shape key to be defined. Transformations in the vertical Z-axis in local space are often used. The animation drivers panel with these settings can be seen on the left of Figure A.1.

This process can be repeated to include all the character's shape keys, making them all accessible via bones. Next, Blender's "Game" mode (also known as the BGE) can be entered to set up script based bone-driven shape key animation as follows:

- In the BGE, the character's armature must be selected in the 3D view. The "Logic Editor" panel must then be opened to set up to control events in the BGE. Three logic blocks are needed, as seen in Figure A.2, starting with a "Sensor" block set to "Always", connected to a "Controllers" block set to "Module", connected to an "Actuators" block with the type set to"Armature" and the constrain type set to "Run Armature".

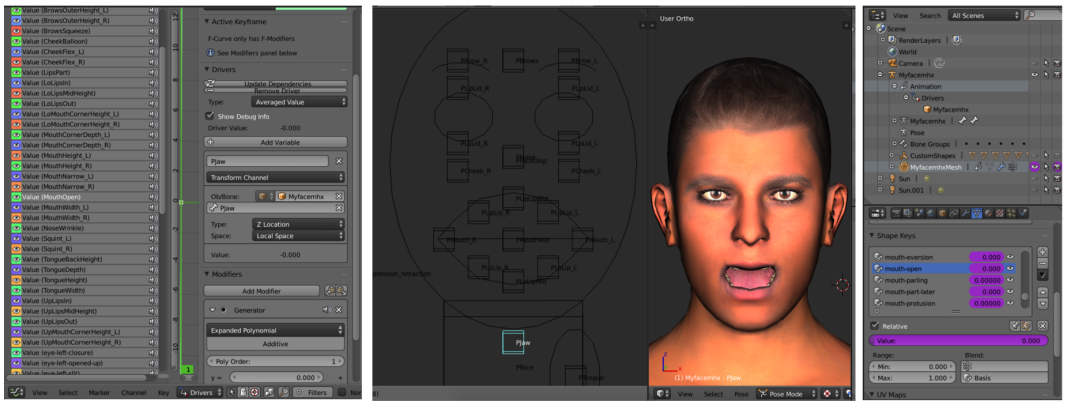


Figure A.1: Rigged character head with bone-driven shape key animation set up exhibiting jaw cube's bone control over the open mouth shape key.

- In the "Text Editor" panel, the following Python script must be written and saved:

```
#First import the BGE library
import bge

#Get access the current game logic controller running the
Python script,seen in Figure A.2
control=bge.logic.getCurrentController()

#Gets the game object which the controller is on,
Figure~\ref{fig:LogicBlocks} shows it is  linked to the
Armature
owner=control.owner

#If not already initiated, set game object and get the
controller attached to an actuator and activate the actuator
if not "init" in owner:
    owner["init"]=1
    control.activate(control.actuators["Armature"])

#Access the game object access to a particular bone and
then give that bone its positional values. As the bone is
associated with a shape key, the character will be
animated repetitively.
Pjaw=owner.channels["PJaw"]
Pjaw.location=[xAxisValue,yAxisValue,zAxisValue]
```

- Finally the "Controllers" block's "Module" must be given the Python script's file name.

Detailed instructions of bone-driven shape key animation for animators is available in [54]. However, insights into script based animation using the BGE were obtained only through Blender's on-line chat forums.

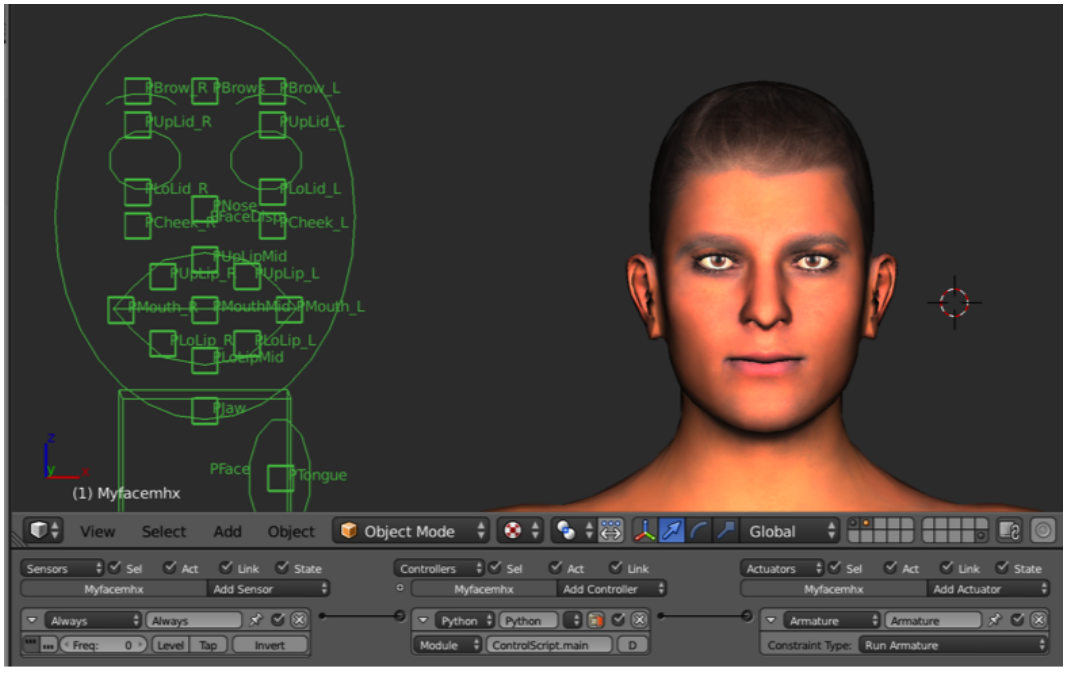


Figure A.2: Rigged character head with logic blocks set up to permit script based animation using bone-driven shape key animation in Blender's Game Engine.

Appendix B

International Phonetic Alphabet

The International Phonetic Association's IPA chart [12].

CONSONANTS (PULMONIC) © 2005 IPA

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Where symbols appear in pairs, the one to the right represents a voiced consonant. Shaded areas denote articulations judged impossible.

CONSONANTS (NON-PULMONIC)

Clicks		Voiced implosives		Ejectives	
ʘ	Bilabial	ɓ	Bilabial	’	Examples:
|	Dental	ɗ	Dental/alveolar	pʼ	Bilabial
!	(Post)alveolar	ʄ	Palatal	tʼ	Dental/alveolar
ǂ	Palatoalveolar	ɠ	Velar	kʼ	Velar
‖	Alveolar lateral	ʛ	Uvular	sʼ	Alveolar fricative

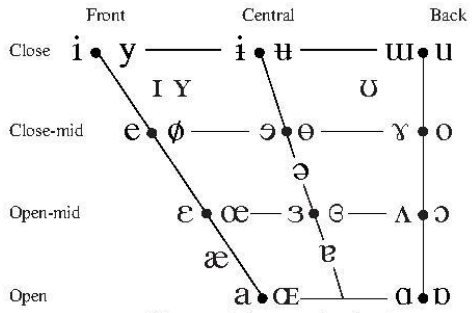
OTHER SYMBOLS

ʍ	Voiceless labial-velar fricative	ɕ ʑ	Alveolo-palatal fricatives
w	Voiced labial-velar approximant	ɺ	Voiced alveolar lateral flap
ɥ	Voiced labial-palatal approximant	ʧ	Simultaneous ʃ and x
ʜ	Voiceless epiglottal fricative		
ʢ	Voiced epiglottal fricative		
ʡ	Epiglottal plosive		

Affricates and double articulations can be represented by two symbols joined by a tie bar if necessary.

k͡p t͡s

VOWELS



Where symbols appear in pairs, the one to the right represents a rounded vowel.

SUPRASEGMENTALS

ˈ	Primary stress
ˌ	Secondary stress ˌfoʊnəˈtɪʃən
ː	Long eː
ˑ	Half-long eˑ
˘	Extra-short ĕ
|	Minor (foot) group
‖	Major (intonation) group
.	Syllable break ɹi.ækt
‿	Linking (absence of a break)

TONES AND WORD ACCENTS

LEVEL		CONTOUR	
e̋ or ˥	Extra high	ě or ˩˥	Rising
é ˦	High	ê ˥˩	Falling
ē ˧	Mid	e᷄ ˦˥	High rising
è ˨	Low	e᷅ ˩˧	Low rising
ȅ ˩	Extra low	e᷈ ˩˥˧	Rising-falling
↓	Downstep	↗	Global rise
↑	Upstep	↘	Global fall

DIACRITICS Diacritics may be placed above a symbol with a descender, e.g. ŋ̊

◌̥	Voiceless	n̥ d̥	◌̤	Breathy voiced	b̤ a̤	◌̪	Dental	t̪ d̪
◌̬	Voiced	s̬ t̬	◌̰	Creaky voiced	b̰ a̰	◌̺	Apical	t̺ d̺
◌ʰ	Aspirated	tʰ dʰ	◌̼	Linguolabial	t̼ d̼	◌̻	Laminal	t̻ d̻
◌̹	More rounded	ɔ̹	◌ʷ	Labialized	tʷ dʷ	◌̃	Nasalized	ẽ
◌̜	Less rounded	ɔ̜	◌ʲ	Palatalized	tʲ dʲ	◌ⁿ	Nasal release	dⁿ
◌̟	Advanced	u̟	◌ˠ	Velarized	tˠ dˠ	◌ˡ	Lateral release	dˡ
◌̠	Retracted	e̠	◌ˤ	Pharyngealized	tˤ dˤ	◌̚	No audible release	d̚
◌̈	Centralized	ë	◌̴	Velarized or pharyngealized	ɫ			
◌̽	Mid-centralized	e̽	◌̝	Raised	e̝	(ɹ̝ = voiced alveolar fricative)		
◌̩	Syllabic	n̩	◌̞	Lowered	e̞	(β̞ = voiced bilabial approximant)		
◌̯	Non-syllabic	e̯	◌̪	Advanced Tongue Root	e̪			
◌˞	Rhoticity	ɚ a˞	◌̺	Retracted Tongue Root	e̺			

Appendix C

Alternative Objective Tests

This section briefly discusses an attempt to improve on the limitation of RMSE's judgement by using a related metric, known as the coefficient of determination, R^2.

R^2 is a measure of "goodness of fit" for a regression line to a set of data [120]. Equation C.0.1 expresses R^2. Note that each dynamic viseme's trajectory is represented at a time-series and has n data point. Equation C.0.1 defines SS_{res}, the sum of squared of residuals between the actual and synthesised trajectory points, y_i and f_i respectively. Equation C.0.3 represents SS_{tot}, the sum of squared deviations of the dependent variables, y_i, from their sample mean, $\bar{y}$. R^2 thus has a range of zero to one, with one indicating perfect synthesis.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{C.0.1}$$

where

$$SS_{res} = \sum_{i=1}^{n} (y_i - f_i)^2 \tag{C.0.2}$$

and

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{C.0.3}$$

R^2 evaluations were applied during tests conducted in Section 7.1 and 7.2 with results graphed in Figure C.1 and Figure C.2 respectively.

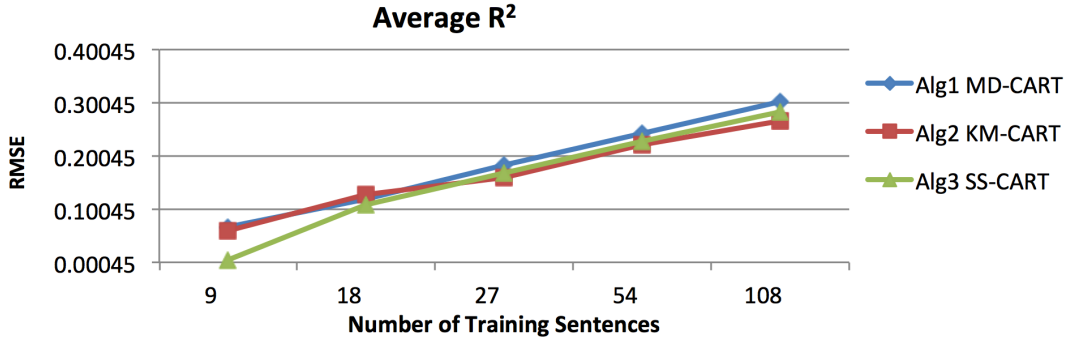


Figure C.1: R^2 results using incremental training subsets for MD-CART, KM-CART and SS-CART algorithms.

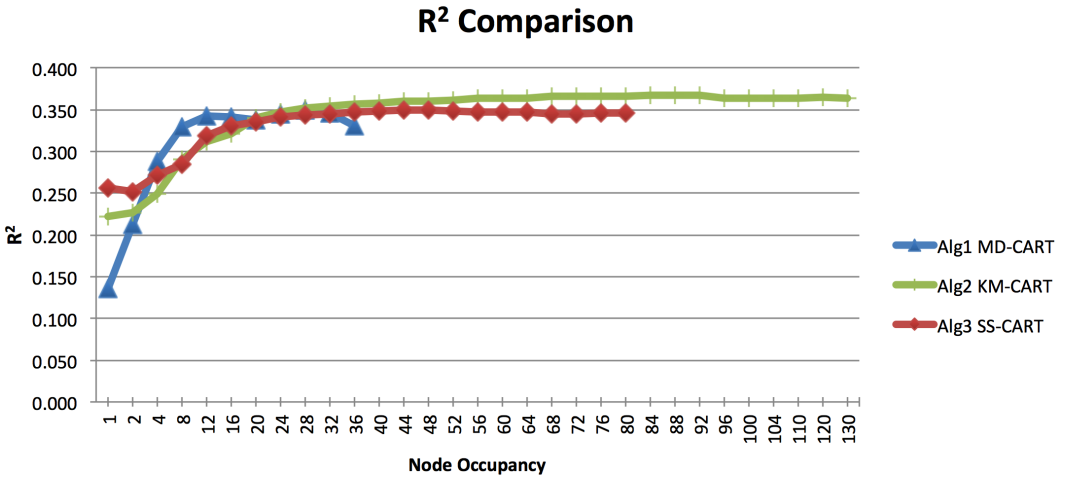


Figure C.2: R^2 results using incremental minimum occupation stopping criteria for MD-CART, KM-CART and SS-CART algorithms.

Comparing Figure C.2 R^2 results to Section 7.2's Figure 7.2 RMSE results, it can be seen that no additional insight is presented. The three CART-based algorithms showed a very similar style of performance with an even smaller variation in the evaluative metric. I conclusion, both RMSE and R^2 provide a usable performance metric, however, their aggregative properties render them less effective in identifying discrepancies relevant to the nature of VSS works.

Appendix D

Publication

The published findings of this thesis are presented here [7]. The work was presented and published in the proceedings of the 1st Joint Conference on Facial Analysis, Animation, and Auditory-Visual Speech Processing, in Vienna, Austria, on September 11-13, 2015. The paper can be accessed in the ISCA Archive, available at: `http://www.isca-speech.org/archive/avsp15`.

ISCA Archive
http://www.isca-speech.org/archive

FAAVSP - The 1st Joint Conference on Facial Analysis, Animation, and Auditory-Visual Speech Processing
Vienna, Austria,
September 11-13, 2015

Improved Visual Speech Synthesis using Dynamic Viseme *k*-means Clustering and Decision Trees

Christiaan F. Rademan, Thomas Niesler

Department of Electrical and Electronic Engineering,
University of Stellenbosch, South Africa
christo@ml.sun.ac.za, trn@sun.ac.za

Abstract

We present a decision tree-based viseme clustering technique that allows visual speech synthesis after training on a small dataset of phonetically-annotated audiovisual speech. The decision trees allow improved viseme grouping by incorporating k-means clustering into the training algorithm.
The use of overlapping dynamic visemes, defined by tri-phone time-varying oral pose boundaries, allows improved modelling of coarticulation effects. We show that our approach leads to a clear improvement over a comparable baseline in perceptual tests.
The avatar is based on the freely available MakeHuman and Blender software components.
Index Terms: conversational agent, talking head, visual speech synthesis, lip animation, coarticulation modelling, CART-based viseme clustering, audio-visual speech data corpus.

1. Introduction

Interaction with sophisticated technologies is increasing the demand for more natural and human-like communication in user interfaces. Part of this movement looks towards virtual avatars as a medium for intuitive, human-like conversational agents. Visual speech synthesis (VSS) plays a critical part in this drive. The work we present here focuses on advancing methods for VSS in computer animated characters.
A phoneme is a classification of a distinct speech sound unit, based on the place and method of articulation. Phonemes can themselves be grouped based on their visual articulation and their pronunciation attributes. When phonemes are grouped based on visually-similar attributes, they are referred to as visemes, a contraction of the words "visual" and "phoneme" [1]. Visemes do not have a one-to-one relation to phonemes [2]. This is due to the inertia of articulatory organs as the shape of the mouth is greatly affected by the articulation of the surrounding units of speech [3]. It is critical to consider this dominance of vocal articulators and their effects on visual speech segments in VSS [4]. It is for this reason that we have chosen to use dynamic visemes.
The dynamic visemes employed in this paper are an adaptation of those described in [5]. Dynamic visemes do not represent a fixed-point pose, but rather a trajectory describing the evolution of displacement with time. We introduce dynamic visemes that are bound to groups of three separate phones (tri-phones). During synthesis, the overlapping dynamic visemes are concatenated using a weighted transition function.
A recent and successful approach to phoneme-to-viseme mapping makes use of classification and regression trees (CART) [6, 7]. Here, clusters of visemes are recursively subdivided based on questions regarding their phonetic properties with the aim of maximising within cluster viseme similarity. We present, on extension to this, a decision tree approach which incorporates k-means clustering to improve viseme similarity.
Gathering all possible examples of naturally occurring tri-phone arrangements would require a huge database. Further more, phonetic annotation is time consuming and expensive. Therefore, our VSS system is based on freely available software components and has been developed and tested using a small dataset and consumer audiovisual equipment. It may therefore be of interest in situations where resources such as annotated data are scarce. For these reasons we have chosen to use a more simplistic model for VSS, whose configuration requires minimal input data.

2. Data

Audiovisual data was captured from a single speaker reading the first 120 sentences (approximately 10 minutes of speech) from the TIMIT corpus [8]. The motivation for using TIMIT is that the sentences it contains are designed to be phonetically diverse, and cover a broad variety of phoneme sequences. A Panasonic HDC-TM900 video camera, at 1920x1080 resolution and 25 frames per second, was used during recordings. The head was stabilised and a mirror, set at a 45° angle, was included for a side view of the face, as shown in Figure 1.

2.1. Phonetic annotation

The audio tracks were segmented and hand labelled using Praat, a speech analysis software [9]. The extended speech assessment methods phonetic alphabet (X-SAMPA) was used for phonetic annotation. Hand labelling was done by a phonetic and orthographic transcription specialist to ensure accuracy and consistency.

2.2. Feature tracking

Figure 1 shows a frame taken from a recorded sentence in which the facial markers have been identified. Markers on the nose, chin, centre of the top and bottom lip and the left and right corners of the lips were tracked. These six facial feature markers were chosen based on experimentation with animated character motion capture, discussed in Section 3.
The tracking algorithm detects and identifies the white markers' horizontal and vertical positions using relative and previous locations. All trajectories were subsequently manually checked for accuracy. The trajectories of the facial feature markers were normalised relative to the nose, because its movements correlated with those of the performers. We did not capture or model

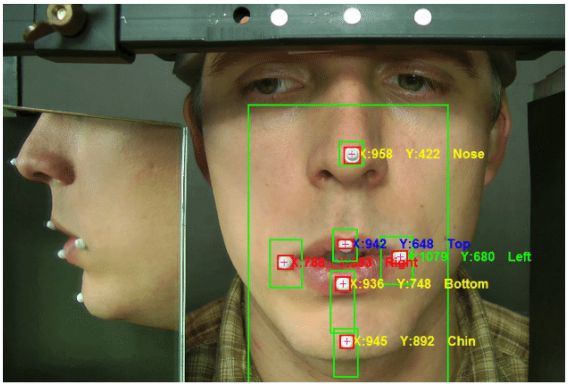


Figure 1: *Facial feature marker trajectory tracking.*

the articulatory movements of the tongue.

2.3. Data processing

The full corpus consists of the original video recordings, X-SAMPA phonetic annotations (including the timing of the phonetic segment boundaries), audio recordings and time-position trajectories for the five features (relative to the nose). The corpus was processed to create a training dataset containing, for each feature, a list of all tri-phone labels, tri-phone duration, true or false values for articulatory phonetic properties (e.g. vowel, schwa, alveolar, fricative, pause etc.) and time-position trajectories, referred to as dynamic visemes in the context of this work. The dynamic visemes were obtained by sampling the marker trajectories of each tri-phone at ten uniformly spaced instances.

3. Animating captured motion

The open source 3D computer graphics software MakeHuman 1.0.2 [10] and Blender 2.70 [11] were used to create and animate the avatars. MakeHuman provides realistic, customisable humanoid character models, which can be imported into the Blender Game Engine (BGE).

Anatomically correct physics-based muscle and skin simulation can produce good VSS, as demonstrated in [12]. This process is very complex. However, we have chosen a much simpler approach that, nevertheless, leads to good results. Our solution uses the rig provided by the MakeHuman model, applying bone driven shape key animation based on tracked feature trajectories.

Rigging is the process of creating a skeleton-like system that consists of bones with which the character can be animated. The function of the bones in a rigged character has been described as "digital orthopaedics", because bones manipulate the areas of the character's mesh to which they are bound, in a way that is reminiscent of how human bones manipulate skin [13]. However, the interaction of multiple bone driven animations becomes cumbersome when trying to achieve the mesh deformations necessary to mimic the subtle facial movements required for speech. Our solution was to use the shape keys (also known as morph targets or blend shapes) provided by MakeHuman.

Animation artists use shape keys to create a library of character mesh deformations with which to speed up animation processes [13]. A shape key is created by saving the deformation of the character's mesh relative to its neutral state. It is then possible to interpolate between the neutral and fully-formed poses of the characters [13]. In the BGE, bones can be used as a proxy to drive the shape keys using a process known as bone driven shape key animation, illustrated in Figure 2 [14]. This process of data-driven animation allows for the motion of the face, captured by the marker trajectories, to be mimicked on the avatar's face. The displacement of the facial markers is used to govern the shape key animation values that manipulate the avatar's mesh, thereby creating the equivalent pose on the avatar's face. The trajectories of the markers at the top, bottom, left and right corners of the mouth, and the chin marker, were scaled to drive shape keys that produced equivalent animations on the avatar. For example, the shape key animation for mouth pursing used the width between the mouth corner markers to animate "O"-shaped poses. To animate bottom lip roll (mouth eversion), the bottom lip to chin distance was used. When compared with traditional motion capture or bone driven animation, our resultant system has the advantage of allowing trajectories to be mimicked using any MakeHuman model.

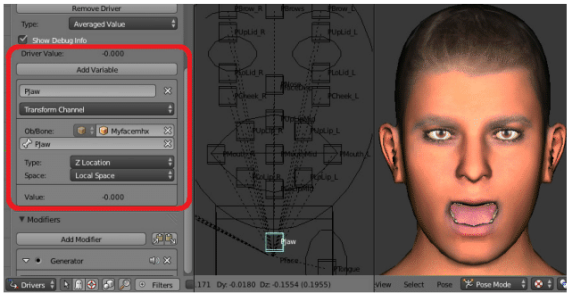


Figure 2: *Example of bone driven shape key animation interpolating between the neutral and open jaw pose.*

4. Minimum deviation decision trees

In this section we describe the decision tree-based viseme clustering methods first proposed in [6], and subsequently expanded to many-to-many phoneme-to-viseme mappings in [7]. Both contributions discuss the application of regression trees to the grouping of static visemes. Clusters of static visemes are split by querying their phonetic context or properties. Figure 3 illustrates the decision tree training algorithm extended to use our dynamic visemes. Since the decision tree algorithms test more than one attribute when attempting to split a group of visemes in a leaf node, they can be classified as multivariate CART algorithms [15, 16, 17].

The decision tree described in [7] applies all possible phonetic context questions to the static visemes grouped in a decision tree's leaf node. The algorithm then measures how homogeneous the resulting child nodes are. An active appearance model (AAM) is used for automatic markerless facial tracking, generating the parameters that numerically describe the static visemes.

Equation 1 is applied to each phoneme instance p_i in a leaf node, where $d(p_i, p_j)$ is the Euclidean distance between instances p_i and p_j and N is the number of phonemes in the node. The smallest value μ_{best} and variance σ_{best} are then selected. Equation 2 is then used to determine the subset impurity I_Z, in which λ is a scaling factor. This procedure is repeated to find the question whose subset best minimizes the impurity of the AAM parameters in the child nodes.

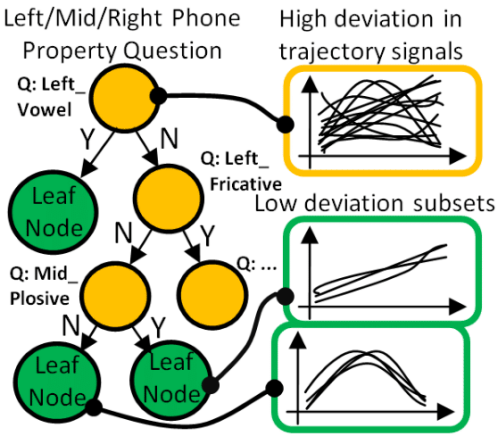


Figure 3: *Illustration of the intended resultant effect of phonetically grouped dynamic visemes. Static visemes, as used in [7], would be represented by points of similar displacement in leaf nodes.*

$$\mu_i = \frac{\sum_{j=1}^{N} d(p_i, p_j)}{N-1} \quad (1)$$

$$I_Z = N \times (\mu_{best} + \lambda \times \sigma_{best}) \quad (2)$$

Our baseline is an adaptation of the work described above. We consider time-dependent visemes, and measure all distances relative to a mean viseme calculated using Equation 3. We then minimise the average deviation D_{Avg} from this mean to split the leaf nodes, using Equation 4. This alleviates the need for an arbitrary scaling factor. This process is repeated for every possible phonetic question. The question resulting in subsets with minimum average deviation in their constituent dynamic viseme subset is chosen to form the child nodes. A minimum occupancy count, as well as a minimum reduction in deviation, are used as stopping criteria.

$$\mu_n = \frac{1}{N} \sum_{i=1}^{N} P_i \quad (3)$$

$$D_{Avg} = \frac{1}{M} \sum_{j=1}^{M} \frac{1}{N} \sum_{i=1}^{N} \mid P_{j,i} - \mu_i \mid \quad (4)$$

5. *k*-means decision trees

We now present an alternative way of choosing optimal decision tree questions. We use k-means clustering [18], with $k = 2$, to classify similar dynamic visemes in a parent node. A separate CART then tries to find the phonetic attribute question which would split the data into subsets that best match these two classes. For this, entropy was used as an impurity measure for discriminating information gain between phonetic attributes [19].

Entropy of a set $H(S)$, is dependant on the number of elements of each discrete class x_i in the class domain (x_1, x_2), as outlined in Equation 5. High entropy is defined by having a large portion p of elements of a class appearing in a set. Information gain IG is calculated by finding the entropy difference between subsets and the original set. Subsets are created by asking

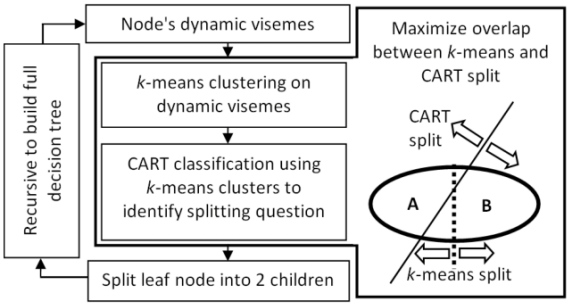


Figure 4: *Illustration of how the k-means (k=2) classifier is used to choose questions and grow a decision tree. The CART classification algorithm finds the question that splits the dynamic visemes into two nodes that best match the clusters produced by the k-means algorithm.*

all phonetic questions. The discrete attribute phonetic question PQ resulting in the set S with maximum information gain is found using Equation 6. In Equation 6, $p(c_j)$ is the proportion of elements in the child node C to the number of elements in the parent node P.

$$H(x_{1,2}, S) = -\sum_{x_i \in (x_1, x_2)} p(x_j) \log_2 p(x_i) \quad (5)$$

$$argmax IG(PQ, S) = H(P) - \sum_{c_j \in (PQ)} p(c_j) H(C) \quad (6)$$

Once the phonetic attribute leading to the greatest information gain is identified, it is used to split the parent node and populate the child nodes. This process is repeated at each node, as illustrated in Figure 4. For our k-means CART algorithm, only the minimum number of dynamic visemes in a parent node was specified as a stopping criterion.

This method was inspired by the work of DeMartino *et al.* [20], who applied k-means clustering to find static poses of fiduciary points in speech according to geometric similarity. By considering nonsense CVCV words, a set of context-dependent visemes could be found for VSS. Our algorithm extends this work by integrating k-means clustering into a phonetic-based decision tree.

6. Dynamic viseme concatenation

For synthesis, a sequence of tri-phone labels, as well as tri-phone start and end times, is provided. For each tri-phone, the decision tree is traversed and the corresponding mean dynamic viseme is retrieved from the leaf node. The dynamic visemes of successive tri-phones are then concatenated using a weighted concatenation function.

Our tri-phone based visemes overlap each other in thirds. A weighted concatenation function tapers each viseme in a piecewise linear fashion, assigning the greatest weight to the dynamic viseme's centre, as show in Figure 5. The weighting functions are applied after each dynamic viseme is scaled to match the duration of the tri-phone in the synthesized utterance. This weighted concatenation approach was used to prevent visemes from dominating whole segments of synthesized speech, while conserving contextual coarticulation effects. The methods used by [4, 5, 20] interpolate between individual static or dynamic visemes, which could be problematic if the chosen

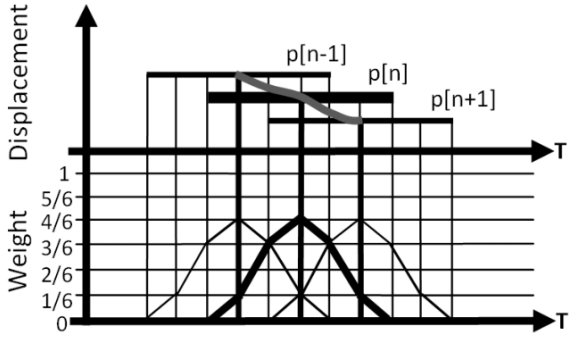


Figure 5: *Illustration of how three successive dynamic visemes p[n-1], p[n] and p[n+1] are concatenated to create a feature's trajectory. The overlapping portions of the visemes are combined using a linear weighting that accentuates their centres.*

viseme misrepresents the speech segment. Our system avoids this potential pitfall by including the effects of the pre and post dynamic viseme context during synthesis, therefore capturing and reproducing more natural coarticulation effects.
The concatenated dynamic visemes are scaled and used to drive the bone driven shape key animations, as discussed in Section 3.2, to produce synthesized visual speech.

7. Testing and evaluation

Of the 120 sentences in our dataset, twelve were randomly chosen and reserved as an independent test set, and the remaining 108 used for training. The twelve test sentences were synthesized using both minimum deviation and k-means decision tree algorithms for both objective and subjective evaluations.

7.1. Objective testing

For quantitative analyses, the synthesized feature trajectories were compared to the original trajectories by calculating an averaged root mean square error (RMSE).
The training set was divided into 12 subsets. Each subset was then used to synthesize the 12 test sentences. The RMSE was calculated by measuring the difference between the synthesized and original trajectories every 2.4 milliseconds. The RMSE was averaged over every feature for all 12 test sentences. This calculation was repeated for every subset. The subsets were then merged into groups of increasing size, and the procedure repeated. In this way 1, 2, 3, 6 and 12 (i.e. all) subsets were used as training material.
Figure 6 shows the RMSE as a function of training set. Table 1 reveals the RMSE and standard deviation when using all 108 sentences for training. From these results it can be seen that both algorithms show continuous improvement as the number of training sentences increases, with neither clearly outperforming the other. Table 1 shows that the minimum deviation algorithm performed marginally better than our k-means decision trees, but that the latter had a slightly lower RMSE standard deviation, indicating better dynamic viseme clustering.

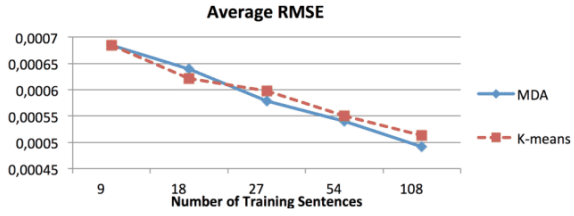


Figure 6: *RMSE using incremental training subsets for minimum deviation algorithm (MDA) and k-means algorithm.*

Table 1: *RMSE and RMSE standard deviation (STD) using the twelve random test sentences with 108 training sentences.*

	Minimum deviation decision tree	k-means decision tree
RMSE	0.000491	0.000513
RMSE STD	0.000238	0.000220

7.2. Subjective testing

The twelve test sentences were also used for subjective evaluation in the form of human perceptual tests. Avatars were animated using both minimum deviation and k-means decision trees. As a baseline, an avatar was also animated using the trajectories obtained from motion capture directly. Test participants were required to watch a sequence of videos, each showing 2 of the 3 possible avatars, randomly chosen. In each video, the first avatar speaks, followed the second, and finally both speak together, as illustrated in Figure 7. Participants were then asked to indicate which avatar was perceived to best articulate the spoken sentence. This was repeated three times for each sentence, allowing all combinations of avatars to be compared for each test sentence. In total, each of the 40 test participants therefore evaluated $12 \times 3 = 36$ videos. Participants were permitted to re-view videos before making a decision. To prevent guessing, participants could also indicate when they could not differentiate between the two avatars.
Figure 8 presents the perceptual evaluation results, showing an improvement of over 12% of our VSS method against the baseline algorithm. Our method also afforded a more favourable assessment than the minimum deviation algorithm, with approximately 15% improvement, when compared with the motion capture baseline.

Figure 7: *Example frame taken from a video used in the perceptual test. In this case the left and right avatars were driven by k-means and minimum deviation decision trees, respectively. The frames show the articulation of the sound "K".*

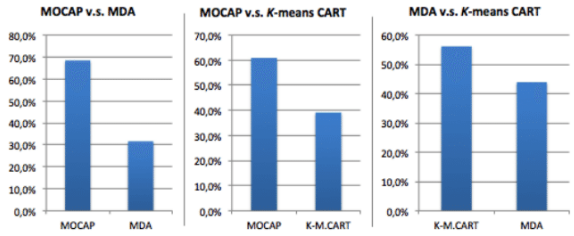


Figure 8: *Perceptual test results comparing direct motion capture (MOCAP), minimum deviation (MDA) and k-means decision tree (K-M.CART) avatar animations.*

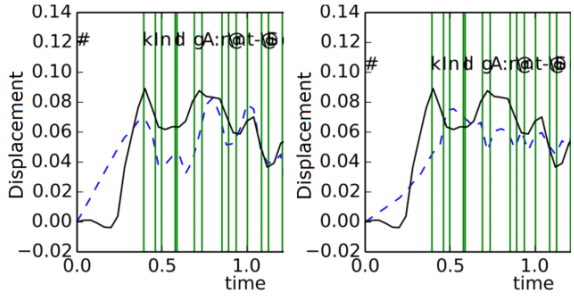


Figure 9: *Synthesised chin trajectories for sentence starting: "Kindergarten children...". The left and right graphs use the k-means and minimum deviation decision trees, respectively, to generate the synthesised blue dashed line. The original chin trajectory is marked by the solid black line. The vertical green lines indicate phone boundaries.*

7.3. Discussion

The baseline algorithm first splits the nodes based on phonetic properties, then tries to find the subset whose visemes are most alike. Our algorithm takes the opposite approach: it first finds a commonality between the visemes and then asks which phonetic property would best preserve this. Although the objective assessment, in terms of RMSE, shows that trajectories synthesized using the baseline algorithm tend to be spatially slightly closer to the original trajectories, RMSE may not be a good indicator for comparing the shape similarity of the synthesized and original trajectories.

As an example, Figure 9 shows the synthesized chin trajectories for the two algorithms. The trajectory synthesized by the minimum deviation algorithm is closer to the original trajectory in terms of RMSE. However, the shape of the trajectory synthesized by the k-means decision tree algorithm better matches that of the original trajectory. This greater qualitative shape similarity appears to be reflected in the perceptual tests.

It is possible that the use of a non-professional voice actor affected the intelligibility of the avatars' speech. With a professional, the recordings may have been more enunciated and better repeated, likely improving dynamic viseme clustering results and the avatars' visual speech. An investigation of this remains for future work.

8. Conclusion and future work

We successfully implemented two improved decision tree algorithms for VSS using dynamic visemes. The trees are trained using tracked and phonetically-annotated audiovisual data. The decision trees were subsequently used to synthesize oral feature trajectories for avatar animation using phonetically-annotated audio alone.

The proposed algorithm incorporates k-means clustering into the decision tree training process. Leaf nodes are split into two child nodes by a process that selects phonetically-based questions which best agree with the k-means clustering result. In this way, the selected decision tree questions best explain the groups seen in the data. The trajectories synthesized using our decision trees led to a slight increase in mean square error relative to a baseline. However, perceptual tests showed a clear improvement over the same baseline. Furthermore, informal qualitative assessment of the trajectories themselves showed that their character better corresponded to the ground truth, even through this was not captured by the mean squared error.

The VSS and animation techniques we present have been shown to work on a small dataset. The algorithms gave good synthesis results even when trained on just 108 sentences. This dataset is sparse when compared with the 1199 and 2542 utterances used in related work [5, 7]. Furthermore, the number of tracked features (6) is far smaller than those typically employed in alternative systems, such as AAMs. Hence, our system may be attractive in situations where neither advanced video capturing equipment nor a lot of data is available. This is typically the case in under-resourced language environments, a category in which much of sub-Saharan Africa falls. By making use of freely-available software tools, such as MakeHuman and Blender, it is possible for small research groups in poorly resourced environments to produce a flexible avatar for use in human-computer interaction.

Future work will include the addition of tongue data capture and animation, and the incorporation of multiple facial features to allow for expressions or gestures typically used in conversations.

9. Acknowledgements

This work was supported in part by the National Research Foundation of the Republic of South Africa (grant TP13081327740). The authors would like to thank Alison Wileman for her phonetic annotations and the Blender and Python online communities for sharing their knowledge.

10. References

[1] Fisher. C.G., "Confusion among visually perceived consonants", Journal for Speech and Hearing Research, 11: 796-804, 1968.

[2] Turkmani, A., "Visual analysis of viseme dynamics", Ph.D. dissertation, Dept. Eng. and Physical Sciences, University of Surrey, Surrey, 2008.

[3] Krňoul, Z., Železný, M., Müller, L., and Kanis, J., "Training of coarticulation models using dominance functions and visual unit selection methods for audio-visual speech synthesis", Proc. Interspeech, 585-588, 2006.

[4] Cohen, M.M., and Massaro, D.W., "Modeling coarticulation in synthetic visual speech", in *Models and Techniques in Computer Animation*, Magnenat-Thalmann, M., and Thalmann, D., Eds., Tokyo: Springer-Verlag, 1993, pp. 139-156.

[5] Taylor, S.L., Mahler, M., Theobald, B.J., Matthews, I. "Dynamic units of visual speech". Proc. 11th ACM SIGGRAPH/Eurographics Conf. Computer Animation. Eurographics Association, 275-284, 2012.

[6] Galanes, F., Unverferth, J., Arslan, L., Talkin D., "Generation of lip-synched synthetic faces from phonetically clustered face movement data" Proc. Int. Conf. Auditory-visual Speech Processing, 191-194, 1998.

[7] Mattheyses, W., Latacz, L., and Verhelst, W., "Comprehensive many-to-many phoneme-to-viseme mapping and its application for concatenative visual speech synthesis", Speech Communication, 55 (7-8): 857-876, 2013.

[8] Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., and Dahlgren, N., "The DARPA TIMIT acoustic-phonetic continuous speech corpus", CD-ROM, National Institute of Standards and Technology, 1986.

[9] Boersma, P., and Weenink, D., "Praat: doing phonetics by computer", Computer program, Version 5.4.08, Online: http://www.praat.org/, accessed on 10 Dec 2015.

[10] blender.org, Blender v2.70, Online: http://www.blender.org, accessed on 19 Feb 2014.

[11] makehuman.org, MakeHuman v1.0.2, Online: http://www.makehuman.org/, accessed 25 Feb 2014.

[12] Sifakis, E., Selle, A., Robinson-Mosher, A., Fedkiw, R. "Simulating speech with a physics-based facial muscle model", Proc. Symposium on Computer Animation (SCA), 261270, 2006.

[13] Hess, R. "Blender foundations: the essential guide to learning Blender 2.6", Amsterdam: Focal Press, 2010.

[14] Thames, C., "Tutorials for Blender 3D", Online: http://www.tutorialsforblender3d.com/, accessed on 19 Feb 2014.

[15] Breiman, L., Friedman, J. H., Olshen, R.A, and Stone, C. J., "Classification and regression trees", Monterey, CA: Wadsworth, 1984.

[16] Witten, I. H., and Frank E., "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", San Francisco, Morgan Kaufmann, 1999.

[17] Quinlan, J.R., "C4.5: Programs for Machine Learning", San Francisco, Morgan Kaufmann, 1993.

[18] Pedregosa, G., Varoquaux, A., Gramfort, V., Michel, B., Thirion, O., Grisel, M., Blondel, P., Prettenhofer, W.R., and Dubourg, V., "scikit-learn: machine learning in Python", in Journal of Machine Learning Research, 12:2825-2830, 2011.

[19] Rokach, L., and Maimon, O., "Data mining with decision trees: theory and applications", Second edition. Hackensack, New Jersey: World Scientific, 2015.

[20] DeMartino, J.M., Magalhaes, L.P., and Violaro, F., "Facial animation based on context-dependent visemes," Computers and Graphics, 30, 971-980, 2006.