A Nonlinear Framework for Facial Animation

by

Hanieh Bastani

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Copyright © 2008 by Hanieh Bastani
Abstract

A Nonlinear Framework for Facial Animation

Hanieh Bastani

Master of Science

Graduate Department of Computer Science

University of Toronto

2008

This thesis researches techniques for modelling static facial expressions, as well as the dynamics of continuous facial motion. We demonstrate how static and dynamic properties of facial expressions can be represented within a linear and nonlinear context, respectively. These two representations do not act in isolation, but are mutually reinforcing in conceding a cohesive framework for the analysis, animation, and manipulation of expressive faces. We derive a basis for the linear space of expressions through Principal Components Analysis (PCA). We introduce and formalize the notion of expression manifolds, manifolds residing in PCA space that model motion dynamics for semantically similar expressions. We then integrate these manifolds into an animation workflow by performing Nonlinear Dimensionality Reduction (NLDR) on the expression manifolds. This operation yields expression maps that encode a wealth of information relating to complex facial dynamics, in a low dimensional space that is intuitive to navigate and efficient to manage.
Acknowledgments

My sincere gratitude goes to my advisor, Dr. Eugene Fiume. Without his thought provoking discussions, patience, and understanding in affairs academic as well as non-academic, this work would not have been possible.

I would like to thank Dr. Karan Singh for his time, and for being my second reader.

Finally, I would not be where I am today if it wasn’t for the love and support of my family, and for the faith they have in me. With all my heart...Thank you!
Contents

1 Introduction ................................................. 1
   1.1 Motivation ........................................... 1
   1.2 Statement of Thesis ................................. 4
   1.3 Contributions ....................................... 6
   1.4 Thesis Overview ................................... 6

2 Anatomy and Psychology ................................. 8
   2.1 Anatomy of the Face ............................... 9
   2.2 Psychological Principles of Expression .......... 9

3 Survey of Facial Animation .............................. 13
   3.1 Foundations ....................................... 14
   3.2 Parameterizations and Standardizations ......... 18
      3.2.1 FACS ........................................ 18
      3.2.2 MPEG-4 ..................................... 20
   3.3 Synthesizing and Animating Expressions ......... 20
      3.3.1 Animating with respect to facial basis ... 20
      3.3.2 Synthesizing and animating from data ... 23
      3.3.3 Facial transfer ................................ 25
3.4 Summary ................................................................. 29

4 Manifold Learning Techniques ........................................... 30
4.1 The General NLDR Problem ........................................... 31
  4.1.1 A unified framework for NLDR embedding ................. 32
4.2 Isomap ................................................................. 34
  4.2.1 Geodesic distances ............................................. 34
  4.2.2 The Isomap algorithm ......................................... 34
4.3 Locally Linear Embedding ........................................... 36
4.4 Laplacian Eigenmaps .................................................. 38
4.5 Mapping vs. Embedding .............................................. 39
4.6 Manifold Alignment ................................................... 41
4.7 Probabilistic Manifold Modelling .................................. 44
  4.7.1 A general framework ........................................... 44
  4.7.2 Algorithms ..................................................... 45
4.8 The Facial Manifold .................................................. 47
  4.8.1 The Facial manifold in research and applications ........ 49
4.9 Summary ............................................................... 52

5 Spatio-Temporal Facial Dynamics Learning .......................... 54
  5.1 The Linear and Nonlinear Spaces .................................. 55
  5.2 Creating the Action Unit Basis ..................................... 56
    5.2.1 The PCA algorithm ........................................... 56
    5.2.2 The Data ..................................................... 57
    5.2.3 Analysis of PCA behaviour .................................. 61
    5.2.4 Multiple PCA spaces ........................................ 63
6 NLDR Analysis on Expression Manifolds

6.1 Assumptions and Requirements on the Manifold

6.2 Spline Approximation of Expression Manifolds

6.2.1 Background: interpolation and resampling

6.2.2 Resampling of the manifold

6.3 Performing Local and Global NLDR

6.4 Inverse Projection

6.4.1 Approximating $R^m \rightarrow R^M$

6.4.2 From points to sequence of frames

6.5 Summary

7 From representation to animation

7.1 Working in the PCA Space

7.2 Working in the NLDR Space

7.3 Integration of Various Expression Manifolds

7.4 Workflow of the System

7.5 Summary

8 Conclusion

8.1 Future Directions
Chapter 1

Introduction

“When people laugh at Mickey Mouse, it’s because he’s so human; and that is the secret of his popularity.” - Walt Disney

1.1 Motivation

Art requires the active involvement of an audience. Such involvement is not necessarily of a physical nature, but can take the form of mental engagement or emotional identification. The motion picture relies on the notion of viewer engagement and identification in a principled manner. As an art form, the motion picture becomes a discourse between a storyteller and an audience that incites curiosity and emotional response. This discourse requires its audience to understand, identify with, and internalize the journey of the characters as the plot unravels. Animation follows these same principles; however, character identification is faced with the delicate challenge to anthropomorphize non-human characters. Perhaps the most significant portal to the world of the character is his face. Face and body gestures are what breathe life into an animated character. It is the believability, fluidity, and aesthetics
of such gestures that allows a viewer’s imagination to be masked in the character within an animated world. However, our intimate familiarity with the human face as a primary channel of communication makes the viewer highly sensitive even to minutely perceptible elements of facial motion. This phenomenon poses both an artistic and technical challenge to facial animation.

Contrary to what one may think, the resolution to this challenge does not lie in creating faces with higher complexity or fidelity to a real human face, neither in the structure of the face nor its motion. In fact, simplicity has been a virtue in traditional animation, including many acclaimed Disney films where a face is brought to life merely with a few lines. Scott McCloud [28] offers a thoughtful view why simplicity has allowed audiences to cherish and internalize the personalities of many animated characters. McCloud suggests that abstraction is the phenomenon that allows an entity to become a concept, universal in its domain. When a character is rich in detail, it represents a single instance of a particular personality and appearance. On the other hand, when a character is abstracted it becomes universal, taking the form of what our imagination attributes to it. In particular, abstraction allows us to attribute our own personalities to that of the character’s. The reason we are able to do so is the mechanism by which we perceive the self and the other. The realm of the self is one of abstraction. When we mentally visualize ourselves, we do not picture details of our appearance or actions. The realm of the other is one perceived through our senses, and in an attempt to understand and analyze the other, we engage in detailed examination. Hence, by abstracting characters, we can mask ourselves in them, and actively engage in perceiving their world. As Picasso said, “Art is the elimination of the unnecessary.”

Another compelling phenomenon that we must consider in a discussion about facial animation (FA) is the Uncanny Valley conjecture [30], introduced by Masahiro Mori in 1970. The Uncanny Valley relates the degree of similarity between a non-human entity and a hu-
man to the emotional response the entity incites in an audience. It is important to note that
The Uncanny Valley has neither been proved nor disproved. It is not a theory of science,
rather a suggestive metaphor to elucidate human reaction to non-human entities. While
this conjecture was originally stated for robotics, it has been extended to the context of
3D characters. The Uncanny Valley hypothesizes that as the behaviour and appearance of
a non-human entity increases in similarity to a human, our emotional response towards it
also increases. However, when this entity reaches an “almost human” stage, our perceptive
system emphasizes on its non-human characteristics, masked within the mold of a human.
This results in a cognitive dissonance and incites a negative response. As the leap is taken
from an “almost human” to a “fully human,” our emotional response reaches its positive
peak. This relationship is visualized in Figure 1.1. In the context of facial animation, our
high expectations of the behaviour of the face can indeed lead to a cognitive dissonance when
a character’s face is intended to fully resemble a human face, but has unnatural nuances.
Such a response may be detrimental to engaging the audience in character identification.
With technology advancing in mesh modelling, and motion capture and transfer capabilities, animators are more often making the leap to the “fully human” facial appearance and behaviour. Intriguingly, the art of telling a story through the moving image is not revolutionized as much as the artistic techniques are. However, with the increasing availability of advanced 3D techniques, complex data, and computational power, we must devise and continuously update a unifying framework to represent, manipulate, and reason about the data. Furthermore, we must be able to integrate our perpetually advancing canvas with the traditional and emerging principles of art direction. Responding to these challenges takes one on a journey through an exciting interdisciplinary area of research drawing from various specializations of art and science alike, computer animation!

1.2 Statement of Thesis

We will motivate the discussion of our work through a challenge. The challenge is to visualize a continuous sequence of human facial expressions through line drawings that artistically depict the fluidity of facial motion. Several approaches may come to mind immediately. A very simple approach may associate the slope of the line with the perceived emotion of the facial expression. When the facial expression communicates a positive emotion, we draw a positive slope. As the perceived emotion becomes negative, we transition to a negative slope accordingly. A similar approach may use this slope-to-emotion representation but at a local level. For example, we map the emotion slope of a local facial feature to the emotion slope of another feature. The question this challenge poses is what are the ways in which one can represent continuous facial dynamics intuitively, accurately, and aesthetically? More intriguingly, are there any representations that are natural to the space of faces, if such a space could be constructed?
This thesis attempts to explore precisely these questions, and to provide evidence that a representation natural to the space of faces does in fact exist. We seek a representation for the static and dynamic expressive face that lends itself to the demands of 3D facial animation. Such demands include the intuitive *modelling* and *manipulating* of an expression or sequence of expressions. In doing so, we employ the characterization of the static expression as a composition of one or more activated muscles. If we interpret these muscles as degrees of freedom acting on the face, a static expression can be expressed as a weighted combination of these DOFs. Such a phenomenon can naturally be expressed in terms of a linear basis, spanning a space parameterized by the facial DOFs, such as the muscles.

How about the continuous sequence of expressions? If we were to plot each static expression in a continuous sequence onto the space described above, we would observe that the static expressions are not strictly colinear. Rather they follow a continuous trajectory that exhibits some level of curvature as a result of the fluidity and smoothness of facial dynamics. This phenomenon suggests that a space spanned by independent facial degrees of freedom not only can represent static facial expressions, but is one that can also successfully embed certain properties of facial dynamics. The question now becomes how one can use these curved trajectories to systematically represent facial dynamics, and gain a manipulation handle to such potentially complex dynamics for purposes of animation.

In answering these questions, we introduce and formalize the concept of *expression manifolds*, and present a nonlinear framework for representing and managing facial dynamics embedded in the expression manifolds. We hope that our ideas and techniques will take the animator one step closer to interacting with facial expressions within a space that respects their intrinsic properties.
1.3 Contributions

The main contributions of this thesis are outlined as follows:

- A survey and constructive criticism of existing techniques in facial animation.
- A proposal in light of this survey for a new approach to modelling facial animation that combines linear and nonlinear analysis.
- The introduction and formalization of *expression manifolds*, Section 5.3.

1.4 Thesis Overview

In chapter 2, we review some important non computational matters relating to human facial expressions, including anatomical and psychological issues.

Chapter 3 provides a survey of computer facial animation, dating back to the groundbreaking experiments of Parke. We introduce the reader with the dominating “schools of thought” in FA, and present the current and previous approaches to parameterizing, animating, and standardizing the face mesh. We present these approaches in the context of the prevailing challenges of computer facial animation, and hope to create a sense for the directions in which FA research has evolved in responding to these challenges.

Chapter 4 investigates techniques for modelling data sets with inherently nonlinear patterns, residing on high dimensional and potentially complex manifolds. We present the general problem of analyzing such data sets, and examine how this analysis can benefit from dimensionality reduction. We examine three general approaches to dimensionality reduction on curved manifolds, a local, global, and probabilistic approach, each with its own benefits and shortcomings. Finally, we explore the concept of “The Facial Manifold” in recent literature. Modelling facial data on a manifold is a relatively new direction of research that
has produced highly promising results. In light of this new area of research, it may seem irrational to analyze facial behaviour with traditional linear models. We survey recent experiments and findings in this area, as a prelude to presenting our work in the subsequent chapters.

Chapter 5 examines the suitability of Principal Component Analysis (PCA) in learning an efficient basis for the linear space of static facial expressions. We state the desirable properties of such a basis and analyze the advantages and disadvantages of deriving this basis through a statistical approach. We evaluate the potency of the resulting PCA space in embedding temporal facial dynamics, which are often projected as nonlinear curves onto a space that respects the underlying facial motion degrees of freedom. We then formally introduce expression manifolds, nonlinear manifolds residing in PCA space that model the temporal dynamics of semantically similar facial expressions.

Chapter 6 applies the local and global Nonlinear Dimensionality Reduction (NLDR) techniques discussed in Chapter 4 to the expression manifolds defined in Chapter 5. We elucidate some of the fundamental differences between local and global NLDR techniques. Unfortunately, data sets derived from natural phenomenon, including facial motion, must undergo some processing before they can be used as suitable input to various algorithms. For our data set to meet density and uniformity requirements imposed by NLDR techniques, we present a resampling method that derives a well behaved data set approximating the underlying smooth expression manifold. We also present an inverse mapping scheme for NLDR points.

Chapter 7 brings the disparate components discussed throughout this thesis together into a cohesive animation system. We discuss certain issues relating to the user interface with our proposed system and also provide the blueprints for a coherent workflow, integrating expression manifolds and NLDR maps into the facial animation pipeline.
Chapter 2

Anatomy and Psychology

“Animation’s language is the language of caricature. Our most difficult job was to develop the cartoon’s unnatural but seemingly natural anatomy for humans and animals.” -Walt Disney

In order to portray the human body, one must know the human body. Of course this is trivial if realism is the objective of portrayal, but what if caricature is the intention of the artist? In order to exaggerate the expression of human motion, an artist must study the human form, know how the muscles, bones, and joints function, and just as importantly, know their limitations. Beyond the anatomy, the artist must know how different forms of motion convey different emotions. Only then will the artist know what motion to exaggerate, and in what form. This is true even if the character being portrayed does not possess a human figure. A strong understanding of the anatomy and psychology of human motion is a key to attributing human emotions to nonhuman characters. For example, the style of swift and vivacious motions that characterize an exuberant individual can be applied to a fish or a teapot to communicate the same set of emotions. To that extent, we start our discussion by reviewing important properties of the anatomy and psychology of facial expressions.
2.1 Anatomy of the Face

Facial muscles are similar to layers of elastic sheet, extending over the skeletal structure and through the subcutaneous facial tissue. The skeletal structure is composed of two bone categories: the cranium, and the facial bones, such as the mandible. Unlike other skeletal muscles, facial muscles are fixed to the bone of the skull only at their origin, and extend into the skin dermis. For this reason, a contraction of the muscle causes deformation to the facial skin. The muscles are also attached to each other in the connective tissue, and for this reason a small contraction of one muscle often pulls another muscle. Out of about twenty-five muscles in the face, only about eleven of them are responsible for facial expressions. These are termed the *muscles of expression*, and are visualized in Figure 2.1 on the forefront of the face.

2.2 Psychological Principles of Expression

One of the most acclaimed scientific documents on expressions is Darwins *The expression of emotions in man and animals*, [9] in which he identifies three principles governing involuntary expressions in response to various emotions and sensations. The principle of *serviceable associated habits* states that complex actions are carried out in response to certain states of the mind. When one finds himself in familiar state of mind, through the force of habit and power of association, the same movements are performed. The principle of antithesis states that when a state of mind is opposite in nature to a specific state of mind, there is an involuntary tendency to perform movements which are also opposite in nature. The principle of *the actions of nervous system* states that certain movements are a result of an excited nerve force. These movements are independent of will, and to a certain extent, independent of habit. The motions induced by these three principles are what we perceive as expressive
Figure 2.1: Muscular Anatomy of the Face
[23]
movements. As an example of the first two principles, a disappointed state of mind almost
certainly results in shutting the eyes for a brief amount of time, shaking the head, or turning
the head away, as if refusing to see the subject of disappointment. On the other hand, an
accepting or pleased state of mind results in dilating the eyes, and nodding ones head, as
if to see or grasp the subject of pleasure. Examples of the third principle include vascular
expressions, such as blushing, or reflex movements, such as suddenly shutting the eyes with
a shake of the head when in close proximity of an unexpected event.

A great deal of modern facial expression analysis and research has been credited to Paul
Ekman, a psychologist whose primary interests are related to human expressions, gestures,
and emotions. Ekman is a vocal supporter of the notion of expression families. In his early
research, he postulated six universal classes of expression, anger, disgust, fear, happiness,
sadness and surprise. Each of these classes represents the particular emotion through a wide
range of expressions with visual variation, but common core configuration properties. The
variation across expressions in each family reflects information regarding the intensity of the
emotion, whether the emotion is controlled, whether it is simulated or spontaneous, and to
some extent information about the events which provoked the emotion [12].

There are several conflicting schools of thought on how expressions are interpreted. Ek-
man, a supporter of the categorical school of thought, insists that emotions belong to discrete
categories. The non-categorical school of thought (Schlosberg, Russell) claims that emotions
are points in a low dimensional space. This space can be described by two axes, relating
to the degree of arousal (i.e. bashful, surprised) and the degree of pleasantness (i.e. guilty,
delighted). Etcoff and Magee (1992) raise an integral question about whether categorization
is a function of the human perception system, or whether we perceive emotions continuously
but categorize them with our linguistic system.

Computational models of facial structure and expressions have been significantly influ-
enced by anatomical models of the face, and psychological models and interpretations of expressions, respectively. A survey of these computational models is presented in the next chapter.
Chapter 3

Survey of Facial Animation

“Are we to paint what’s on the face, what’s inside the face, or what’s behind it?” - Pablo Picasso

The fascination of artists with the human face is not an artifact of modern art. In 2006, archaeologists discovered what they believe to be the oldest manifestation of a human face, a cave drawing in Angouleme, France, dated an approximate 27,000 years old. However, our review of the fascination with the human face only dates back to 1972, with Frederick Parke’s pioneering work in computer facial animation. This chapter provides a survey of popular FA techniques introduced since then. The human face has always grasped the attention of performance and visual artists alike. Animation is the point where these two worlds converge.
3.1 Foundations

In this section, we examine the early approaches to facial animation, and use them as a context to present some of the general and ongoing problems in FA. The foundations of modern computer facial animation were predominantly established by Parke [34] [35], Waters [49] [50] [51] [48] [26], Platt and Badler [36], who developed models for skin, muscle dynamics, and parameterized deformations. Traditionally, expressive faces were brought to life by specifying key configurations of the evolving face for every few frames. Indeed, this is how the first 3D facial model was animated. In 1972, Parke [35] presented a set of guidelines for creating a polygonal facial mesh. The process began with drawing polygons onto a real assistant’s face, as in Figure 3.1, and analyzing whether the topology of the polygon was effective in deforming with the expressive face! For example, polygon edges should coincide with creases of the face. The polygon was then animated by specifying the positions of vertices for key frames, and interpolating intermediary frames with a cosine interpolant. The keyframe vertex positions were measured directly from the photographs of the assistant. Applying the keyframe technique directly to the face itself soon proved to be inadequate in dealing with complex expressions, and even more so with complex 3D meshes. This directed research in FA towards formulating parametric and muscle models, and applying the keyframe technique to the facial parameters, rather than the face itself. From the early days of computer FA, two questions have prevailed:

• How does one best represent the facial structure?

• How does one best represent the facial motion?

The field of FA immediately became bipartite, branching into approaches that intimately tied facial structure and motion representation, and those that focused on motion representation without regard to a particular structural representation. Approaches in the former group
were concerned with employing a skin and muscle model with some level of anatomical fidelity to drive facial animation. Often such facial models were composed of a layered network, accounting for skin, muscle, tissue, bone, and the dynamics within. Approaches in the latter group were more generalized, offering FA techniques that work directly on a given input facial mesh.

In 1981, Platt and Badler [36] introduced an ambitious system which models a network of springs to represent the bone, muscle, and skin hierarchy of the face. Their system captured true facial expressions from an actor, determined facial activity, and encoded the activity using the *Facial Action Coding System* (FACS) [14] (Section 3.2.1). Subsequently, the FACS encoded expression is simulated by applying the necessary force, as looked up by a parser, to muscle fibers in parallel or sequence to drive the animation of a 3D facial mesh.

Platt and Badler’s work embarked on a fundamental challenge of FA:

- Given a true human expression, how can one best derive the independent facial actions involved?
The commonly used term for independent facial actions is *action unit*, or AU. The term originated from the FACS literature, and is used to communicate the notion of an atomic communicable facial motion which is not susceptible to a further break down of constituents. An action unit is the result of one or multiple muscle contractions. Representing an expression through action units is generally preferable to a strictly muscle based representation, as action units provide a high level handle and express facial motion in terms of its visible behaviour, rather than the unseen underlying anatomy. However, as is mentioned in [36], actions units are not exclusionary in their interactions. The combination of multiple AUs may give rise to the belief that an inactive AU is actually active. For example, the activation of multiple mouth AUs may lead one to believe that the cheek is raised, when in fact there is no AU activity initiated in the cheek region. Indeed, such ambiguities become inevitable as the complexity of the facial expressions considered increase. The above mentioned challenge can be extended and phrased as such: *Given a training sample of the space of actual human expressions, how can we best derive the action units that characterize this space?* In this thesis, we take a statistical approach to solving this problem using local Principal Components Analysis (PCA).

In [34], Parke presented a simple parametric model for facial animation which applies directly to the facial mesh and does not require a model of the underlying facial structure. Localized parameters were derived by interactively adjusting regional vertices of a character mesh to artist drawn sketches of target extreme expressions. Combinations of these local parameters encode full expressions, and are interpolated to yield animation. Despite the simplicity of this model, it illustrates two salient principles of FA, that of *localization* and *target expression definition*. As will become apparent in the coming sections, these are not two independent principles; rather they interact with each other in a mutually reinforcing manner, and account for much of the foundation on which a substantial body of FA research
Another ongoing problem in FA is devising animation techniques which are topology invariant, such that facial motion can be easily transferred and applied to various meshes representing different faces or the same face at different levels of detail. Parke’s parameterization is defined with respect to a particular facial mesh and may yield undesirable results when transferred to another mesh, in particular if there is large discrepancy between the structure of the two meshes. Waters’ [50] early work in FA made important contributions to topology independent facial animation. He extends the principles in [34] [36], and presents a parameterization of a muscle model that accounts for linear muscles that pull along a line of contraction (e.g. frontalis), and sphincter muscles that contract radially. The model parameters are based on simplified physical properties consisting of the muscle zone of influence, muscle force falloff, and material properties relating to muscle spring and tension, and skin elasticity. Waters uses the concept of muscle vectors to describe the direction and magnitude of linear/parallel muscles. Each muscle vector affects a specified zone of influence on the skin. The mesh is animated by displacing skin mesh nodes within the zone of influence by a nonlinear interpolant, as a function of the node’s radial and angular distance to the muscle vector’s line of contraction. Sphincter muscle action is simulated by contracting an elliptical zone around a point on the mesh.

While Waters’ muscle model can be inserted beneath any facial mesh, defining the positions of muscle vectors and the precise parametric values for obtaining a desired expressions requires extensive knowledge and experience. Such usability factors have in fact played an integral role in determining the applicability of a particular technique in industry, and as a result have been a driving force in FA research.
3.2 Parameterizations and Standardizations

It is important that artists and researchers in the FA industry have a common language for encoding and expressing facial motion. Several parameterizations have been established as accepted standards in an attempt to taxonomize facial communications. As briefly mentioned in Section 3.1, parameterizations of the face can be performed with respect to the underlying anatomical activity, or with respect to the perceptible facial behaviour. Parameterizations of the latter category can be best understood as providing a high level behavioural handle to the low level muscle activity. Instead of interfacing and manipulating each muscle directly, multiple muscles are grouped together to express a single perceptible and communicable gesture. We examine two behavioural standardizations that have become ubiquitous in industry and research, Facial Action Coding System (FACS), and MPEG-4.

3.2.1 FACS

The Facial Action Coding System [14], developed in 1976 by Paul Eckman and Wallace Friesen, is a widely accepted guide to analyzing facial expressions in terms of primitive behaviours, called action units (AU). An AU is a description of a localized facial gesture, defined by a single or group of muscles and their state (e.g. contracting, relaxing). FACS identifies forty-six AUs which are related to muscle behaviour, and an additional fourteen gestures that cover eye movement and rigid motions of the head.

It is important to note that FACS is a specification of how facial behaviour is related to musculature activity, not a description of an expression space. Hence there are no well defined parameters to drive animation, nor is there a systematic way of combining AUs to guarantee plausible results.

There are two primary applications of FACS in FA. In muscle based animation, FACS may
be used as a guide to drive the required muscle activations to create a particular expression. Where a particular muscle model is not employed, FACS AUs are often used as a reference for creating key target shapes. Localized target shapes are then blended together to animate complete expressions, as in the blendshape technique 3.3.1.

While FACS has been successfully used in FA, it is not a comprehensive doctrine for facial expressions. Two significant shortcomings of FACS are related to its inability to encode various amplifications of an expression, as well as temporal patterns. The binary specification of muscle activation in FACS limits muscles to either a fully contracted, or fully relaxed state. This specification can be especially problematic when there is a need to represent smooth and continuous facial motion. As one possible solution to smooth intensity labelling, Fidaleo [19] models muscle actuation with continuous quadratic polynomials, parameterized by amplification of the expression. The polynomial is bounded by the neutral to the apex of the gesture it models, and the intensity of an expression is labelled by its position along the curve. The idea is that by navigating these gesture curves, one can represent the space of smooth expression change.

The representation of temporal characteristics, in particular, the dynamic relationship between the activation of muscles as an expression develops and releases, is crucial in synthesizing communicable and beautiful expressions. Such dynamics often act as a signature to an individual’s facial communication. Essa [16] developed FACS+, a system which models both the spatial and temporal patterns of the face. Temporal patterns were modelled by fitting exponential curves to muscle contraction and release over time for various expressions. In this thesis, we take a different approach to capturing temporal characteristics. Instead of fitting curves to individual gestures and actuations, we learn through statistical techniques the facial dynamics from a comprehensive data set of facial motion data that includes the range of expressions we wish to represent and simulate. This allows us to more efficiently
capture the relationships among various gestures and expressions. Our techniques will be elaborated upon in Chapters 5 to 7.

3.2.2 MPEG-4

The MPEG-4 standard for facial animation is an encoding method for the definition, animation, and compressed transmission of facial features. The encoding method is comprised of two sets of parameters for producing expressions and visemes, the Facial Animation Parameters (FAP) and the Facial Definition Parameters (FDP). FDPs define the structure of the spatial face through the location of prominent facial feature points. A complete facial appearance model (shape and texture) can be specified using the FDPs. FAPs are low level descriptions of the motion of facial expressions, and are based on minimal yet meaningful muscle actions. The parameters responsible for motion are expressed in Facial Animation Parameter Units (FAPU). FAPUs specify the proportions and fractions of distances between the facial features. FAPUs allow the FAP parameterization to be adapted to various facial structures, essentially making MPEG-4 a topology invariant parameterization. Note that the FDPs, FAPs, and FAPUs actually parameterize a space of expressions, in contrast to FACS which simply provides a specification of expressions.

3.3 Synthesizing and Animating Expressions

3.3.1 Animating with respect to facial basis

Of particular interest to FA research and application is the problem of constructing an expression space with the following properties:

1. The space is spanned by axes that parameterize local and independent motion.
2. The axes are as orthogonal as possible, such that the deformations along any two dimensions have the least amount of conflict and overlap.

3. Each dimension correlates closely to what one would perceive as a primary facial action.

4. The space is comprehensive, i.e. any required static holistic expression can be expressed as a linear combination of local deformations.

The most common approach to building such a facial basis is the shape interpolation technique known as blendshape animation \cite{33}. The idea behind blendshape animation is that the range of deformations of a neutral source mesh is distributed among and parameterized by several target meshes, each of which exhibit the final shape of the mesh under a local deformation. Essentially, the set of target faces act as the basis for the face space, and are analogous to the concept of action units, described above. At any given frame, the shape of the source mesh can be related to the set of target meshes as follows:

$$ S = \sum_{i=1}^{n} w_i T_i, $$

(3.1)

where the scalars $w_i, w_i \geq 0$ for all $i$ and $\sum_{i=1}^{n} w_i = 1$, are the blending weights. The source mesh is then animated by interpolating the weighted combination of the target set for every three to five frames. It must be noted that within this framework, displacement vectors of target vertices are applied to their corresponding vertices on the source mesh. Therefore, Equation 3.1 assumes that every member of the target set has an identical topology with the source mesh in order to build this correspondence.

The linear face space afforded to the user by Equation 3.1 imposes no constraints on the relationships between the targets. The true space of expressions is constrained by both the communicative plausibility and anatomical flexibility of unison muscle actuation. These
constraints are taken into account by bounding the blending weights to construct a plausible subset of the face space expressed in Equation 3.1, formally referred to as a character rig.

**Constructing the basis: Where do the target shapes come from?**

The modelling of a target expression basis is equally a science and an art, and has traditionally been done manually. In software packages such as Maya, the animator sculpts target expressions using popular deformation tools such as clusters, wires, and lattice deformers, as well as character reference images and principles relating muscle activation to facial appearance, such as FACS or Faigin’s [17] guide for artists.

Perhaps one of the most important principles in constructing an expression basis is that of *localization*. It may seem that spanning a comprehensive range of facial expressions requires an extensive library of blendshapes. In fact, studies [13] have demonstrated that permutations of a succinct collection of local deformations can span a wide range of complex expressive faces. Furthermore, within the framework of additive blendshapes, localization of target expressions is critical. To illustrate why, consider two blendshapes $T_1$ and $T_2$ which deform a mutual subset, $C$, of the source mesh. Assuming $C$ is in neutral configuration, application of $T_1$ leads to

$$C_1^{\text{Source}} = C^{\text{Neutral}} + w_1 T_1(C).$$  \hspace{1cm} (3.2)

A subsequent application of $T_2$ leads to

$$C_2^{\text{Source}} = C_1^{\text{Source}} + w_2 T_2(C).$$  \hspace{1cm} (3.3)

The displacement described by $T_2$ is presumed to operate on the neutral face when in fact it is being applied to an already deformed region, resulting in an extrapolated or simply uncommunicable expression. An animator can exercise control over such situations by lo-
-calizing and hence minimizing conflicting deformations. While orthogonality between the target shapes is desirable, the natural overlap of influence zones, and propagation of muscle contraction to adjacent regions of the face renders strict orthogonality infeasible.

In [25], Joshi et al. use motion data to create a segmentation of the face which respects the expressive idiosyncrasies. Specifically, the laplacian vector of a displacement field is used to create a deformation map across the face for each expression. The deformation maps are combined to express local deformation patterns on the face, which are in turn used to segment the face. Local target blendshapes are then automatically modelled based on the segmentation.

In [7], an artist models the initial expression basis based on guidelines that relate muscle activation to facial appearance. Using the artist sculpted basis as a blueprint, a fixed point iteration algorithm between a modification stage and an analysis stage iteratively improves the basis. In the analysis stage, performance data is projected onto the face basis. A least squares procedure provides improved basis parameters to accommodate the motion data. The basis parameters are then optimized with respect to the least squares solution. This process is repeated until the error falls below a certain threshold.

### 3.3.2 Synthesizing and animating from data

Creating believable animation requires specifying motion weights and parameters for every few frames. A manual specification of weights is an overwhelming process, not to mention that fidelity to human facial motion can rarely be achieved. As a consequence, many effective techniques have been devised to drive animation weights directly from data. Performance driven animation, pioneered by Williams [53] in 1990, is the branch of FA that deals with techniques to allow a human actor’s face to puppeteer a character mesh. Williams tracked a set of markers placed on feature positions of an actor’s face in the $x,y$ plane, and used these
to deform the texture coordinates of a digitized face. The problem of performance driven animation can be formalized as one of creating a mapping between the source parameter space, (i.e. motion capture space), and the target parameter space. Quite often, the target space is parameterized by a set of blendshapes, and motion mapping requires deriving the optimized set of blendshape weights that minimize discrepancy between the animation and the actor’s facial motion. What makes this mapping challenging is that it is one-to-many, as multiple blendshape weight combinations can result in the same expression. As a result, attention must be paid to the smoothness and consistency of solutions across frames.

Kouadio et al. [27] devised a simple system in which the marker configuration from performance capture is used to weight a bank of artist sculpted 3D key shapes. These weights are extracted by a least squares solver to minimize the error between the actor’s face and the reconstructed expression. Continuous animation is created by repeating the weight extraction and interpolation process for every frame of the source data.

Deng et al. [10] present a technique in which a user manually selects salient expression frames from a mocap stream, such that the frames span the space of visemes and emotions. These expressions are analyzed and expressed in PCA space. The user then manually adjusts the blendshape weights to create a perceptual correspondence between the character and the selected PCA expressions. Given this supervised correspondence between blendshape weights and PCA coefficients for a training set, new mocap data can be used to drive continuous animation by inferring the new blendshape weights with Radial Basis Functions (RBF), and smoothing the weights over time to avoid abrupt motion. This technique allows the adaption of mocap data to character meshes which are quite different than the actor, but comes with the obvious disadvantage of required manual specification.

Choe and Ko [8] map performance data to a muscle actuated basis. They employ a muscle model for the target mesh that accounts for parallel and sphincter muscle deformation.
tions, simulated through the Finite Element Method (FEM). The performance of an actor is captured through markers, and the motion of these markers are transferred to “virtual markers” on a 3D model of the actor, augmented with their muscle model. The actuation parameters with respect to the underlying muscle model are then extracted. These are the optimal muscle parameters that minimize the discrepancy between the trajectories of the virtual and actual source markers. These muscle parameters are then used as input to the target mesh FEM simulator to generate the corresponding expressions.

Chuang and Bregler [4] present a performance driven approach which does not make assumptions about the existence of a target (or source) basis. The blendshapes parameterizing the target space are derived directly from the source motion. By projecting every frame of the source motion onto the PCA axes of the input space, salient key shapes are identified as those residing on the minimum and maximum ends of the $k$ principal eigenvectors. The reasoning behind this is that selecting the extreme expressions in principal directions of variation is in fact constructing a spanning set whose interpolation can express intermediary expressions. The animator sculpts key shapes for the target mesh corresponding to the PCA derived source key shapes. It is then straightforward to transfer the weights of the source basis to the target basis.

### 3.3.3 Facial transfer

Facial animation transfer refers to the process of driving the facial animation of a target model with deformations created for a source model, with the aim of not only transferring animation parameters, but emphasizing the transfer of the character and communicative qualities of the source model, while respecting the biomechanics of the target model. Facial animation transfer is motivated by several factors. FA transfer encourages the development of a common repository of animated expression segments within the larger community. By
facilitating the reuse of high quality motion data, heavy costs incurred by motion capture as well as animator labour are significantly reduced.

At the heart of the expression retargetting problem lie two subproblems, that of building a correspondence between the source and target models, and that of mapping the source deformation space onto the target deformation space. These challenges must overcome the fact that facial features, proportions, and mesh topology may vary significantly across the source and target models.

At the most basic level, correspondence implies a spatial mapping from one mesh to another, at times with the requirement of topology embedding between the meshes of interest. Often, markers or feature vertices are used in identifying locations of semantic similarity. Given this set of explicit correspondence points, scattered data interpolation using RBF have been used successfully to embed the source mesh into the target mesh.

Noh and Neumann [54] use RBF in conjunction with cylindrical projection to achieve a dense surface correspondence. Given a dense surface correspondence, facial transfer reduces to the problem of transferring the motion parameters. The characterizations of this transfer is contingent on the type of parameterization spanning the deformation space. The deformation space may be characterized by per vertex displacement vectors, by action unit activations within a blendshape context, in a hierarchical displacement framework, etc. Noh and Neumann perform per-vertex motion vector transfer, computing the target vertex displacements from the embedding source vertices using linear interpolation with barycentric coordinates. To compensate for variations in facial structure, the displacement of each source vertex is defined in its own local coordinate system. This allows an adjusted displacement of the corresponding target vertex $x$ to be obtained by a transformation of the coordinate system from $local_{source}^{(x)}$ to $world$ to $local_{target}^{(x)}$. Subsequently, the magnitude of the displacement vector is scaled to properly fit the target mesh proportions.
While Noh and Neumann laid the groundwork for future research in FA transfer, their technique has some notable disadvantages. Their method of transfer interfaces the displacement vectors directly, rather than transferring an expression \textit{space}. Hence, there is no high level manipulation of the transferred expressions, such as varying the intensity or blending independent expressions, etc. In addition, it is important to note that the dynamics of musculature-skin interactions, behavioural preferences for certain movements (e.g. half smile vs. symmetric smile), and anatomical nuances (e.g. a dimple) contribute to the uniqueness of an individual’s facial expressions. A vertex based displacement transfer does not preserve such individuality, as the correspondence is built strictly between the topology of the mesh.

Example based approaches have been used in motion retargetting to build an explicit correspondence between the source and target sets of key model expressions. An example based approach has a higher level of stability when the target model has a quite different structure than the source. The artist decides upon a set of reference key shapes which can span the entire expression space of the source model. Corresponding shapes perceptually resembling the source reference set must be built for the target model, although the target key shapes emphasize the biomechanical characteristic features of the target character. The correspondence between these key models, in effect, aligns the bases of the source and target expression space. The problem of transferring motion becomes that of transferring the source parameters with respect to this space.

Pyun et al. \cite{37} build a correspondence between eighty-four key models spanning the range of Ekman’s six primary expressions and the thirteen primary visemes. The target key models are parameterized based on the PCA representation of displacement vectors differentiating their corresponding source shape from the neutral source shape. Given an input source animation, the output target animation is computed at each frame by projecting the source parameter vector (displacement from neutral) $p_{in}$ onto the target space. The target
expression $T_{\text{new}}$ representing this projection is described as a weighted combination of the target key models, $T_i, 1 \leq i \leq M$:

$$T_{\text{new}}(p_{in}) = T_B + \sum_{i=1}^{M} w_i(p_{in})(T_i - T_B). \quad (3.4)$$

The weights are derived through precomputed linear and radial basis weight functions, whose coefficients are computed analytically by optimizing the residuals of the target key models.

Several issues must be discussed with respect to Pyun’s expression space and blending process. The complexity of the transferred expression space is a function of the number of example key faces used. In Pyun’s experiments, eighty-four key models where used, twenty of them sculpted manually, while the rest were created in a semi-automatic fashion. This limits the use of an example based approach for transferring highly detailed facial motion. Since the key shapes are decided by a user, there is no guarantee of orthogonality, nor is there a guarantee that the selected key shapes truly span the space of expressions. Furthermore, the blending process takes a holistic approach, as expressions are represented as a linear combination of global deformation, rather than local deformations, which is preferred.

Na and Jung [31] augment an example based framework with a hierarchical method of transfer to surmount some of the disadvantages mentioned above. A face mesh is represented as a normal mesh, constituting a base mesh and a series of $n$ normal offsets defining an $n$-level multi-resolution hierarchy. Facial deformation is represented as the collective motion of the base mesh and the gradient of the normal offsets at each level of the hierarchy. Based on a set of user specified feature correspondences, an RBF interpolator provides a retargetting map for coarse motion transfer between the base mesh representations of the source and target example models. Emphasis is placed on retargetting the displacement for each feature point independently, allowing combinations of local motion from a smaller set of example key
models. The weights $\vec{w}$ of the RBF interpolation must satisfy the linear system expression in Equation 3.5.

$$T_k[i] - T_{neut}[i] = \sum_{k=1}^{M} \vec{w}_k[i] h(S_k[i] - S_{neut}[i]).$$

(3.5)

To transfer the nuances of facial motion, a subsequent iterative application of source normal offset differences is applied to the target normal mesh.

### 3.4 Summary

In this chapter we have surveyed existing techniques in facial animation, and analyzed these techniques within the context of the FA challenges they attempt to resolve. We started our discussion by reviewing the foundations of FA, and how they have influenced the direction of research in the area. We then reviewed issues relating to the parameterization and standardization of facial motion, and finally presented different approaches to facial animation, including animating through a blendshape basis, animating from data, and facial motion transfer.
Chapter 4

Manifold Learning Techniques

“Therefore, either the reality on which our space is based must form a discrete manifold or else the reason for the metric relationships must be sought for, externally, in the binding forces acting on it.” - Bernard Riemann

Data that is captured from natural phenomena quite often exhibit nonlinear patterns. Facial motion is no exception. The progression of localized and independent actions of the face are related to one another in a nonlinear fashion. Such data sets articulate the need for computational models that can represent and reason about nonlinear patterns. The study of unsupervised Nonlinear Dimensionality Reduction (NLDR) is a relatively new focus of research in computer science, with a stronghold in the machine learning community. In this chapter, we discuss the general problem of NLDR, and survey the associated techniques and issues. Finally, we examine how NLDR has been used in the context of facial analysis.
4.1 The General NLDR Problem

Highly coherent and correlated data representing a continuous process in high dimensional observation space are embedded on manifolds. A manifold is a mathematical space characterized by local neighbourhoods that are Euclidean at some scale. The relationships among adjacent neighbourhoods, however, yield a global topology that cannot be analyzed within a Euclidean framework. A classic example of a manifold space is our visual perception of the Earth. Within the boundaries of our line of sight, the Earth appears to be a flat, though a panoptic view reveals a more complex structure that nevertheless can be mapped to the plane.

We are concerned here with data residing in observation spaces in which the correlation among local neighbourhoods is of a nonlinear nature. As correlation between two points increases in input space, the distance between the two points decreases on the manifold. To reason about this data, we need to model the data with respect to the intrinsic geometry and connectivity of this manifold. The geometry of the manifold is induced by the correlation among parameters of the data. While the existence of the manifold attests that the data follows certain correlation patterns, the high dimensionality of the data prohibits an intrinsic global parameterization. This leads to the problem of Nonlinear Dimensionality Reduction [44][39]. NLDR can be regarded as an unsupervised learning algorithm that discovers a meaningful low dimensional space whose coordinates parameterize the latent structure of a high dimensional observation space. These coordinates offer an intuitive handle to analyzing and manipulating the data, as they have experimentally been shown to extract a data set’s intrinsic degrees of freedom closer to how the human mind might reason about the data. As an example, NLDR applied to a data set of face images captured from different viewing angles and under different lighting conditions resulted in a parameterization of the face space
Previous dimensionality reduction techniques, such as PCA, ICA, and MDS, guarantee the extraction of the latent structure embedded on a linear subspace. However, such techniques fail to “unfold” multivariate data embedded on a manifold. The classic “Swiss roll”, illustrated in Figure 4.1, illustrates the problem of obtaining a single linear subspace projection from such datasets. The dotted line represents the Euclidean distance between the encircled points. This distance metric is ignorant to the underlying structure connecting the two points. A metric that follows the solid curved line is a more semantically accurate approximation to the distance between the points, as it respects the nonlinear pattern of the data set.

4.1.1 A unified framework for NLDR embedding

The general problem of NLDR can be stated as follows.
Given an $M$-dimensional data set $D = \{x_1, \cdots, x_n\}$ whose points $x_i \in \mathbb{R}^M$ are sampled from an underlying continuous manifold, derive an $m$-dimensional embedding of $D$, such that $m \leq M$, and the embedding coordinates parameterize the data with respect to the manifold’s intrinsic degrees of freedom. Here, $M$ specifies the input dimensionality of the data, and $m$ specifies the dimensionality required to represent the underlying manifold’s intrinsic degrees of freedom.

Various techniques have been introduced over the recent years that tackle this problem. These techniques can be better understood in light of a common framework [2]. It is important to mention that what is referred to as the global structure of a manifold can be explained through the relationships among the manifold’s neighbourhoods. Recall that by definition, a manifold is a structure that is locally Euclidean. Hence, within a local manifold neighbourhood defined by some scale, the mapping from high dimensional input space to low dimensional embedding space is effectively linear. As we shall see, different algorithms capitalize on this property in various ways. The common NLDR framework is outlined as follows.

1. The determination of what constitutes a neighbourhood on the manifold. Some neighbourhood definitions for a data point $x_i$ include its $k$-nearest neighbours, or points that fall within a pre-specified radius $\varepsilon$ of $x_i$.

2. The computation of an $n \times n$ neighbourhood matrix $R$, in which $R_{i,j}$ denotes some proximity metric (e.g. distance) between data points $x_i$ and $x_j$. The computation of $R$ is a function of the type of relationship the NLDR algorithm is preserving. This is expressed as a relationship function $K$ for the set $D$, such that

$$R_{i,j} = K_D(x_i, x_j).$$ (4.1)
3. An eigenvector decomposition on $R$. This allows reasoning about the neighbourhood relations in terms of their principal components.

4. Mapping $D$ onto the embedding coordinates $Y$, expressed in terms of the eigenvectors found above.

With this common framework in mind, we will survey the prominent NLDR techniques. The steps outlined above will be made more explicit when specific techniques are presented.

## 4.2 Isomap

### 4.2.1 Geodesic distances

*Geodesic distance* is a metric in graph theory measuring the closeness of two nodes through their shortest path in the graph. In many recent graphics applications, use of a geodesic metric has gained precedence over the Euclidean metric as Euclidean distances are oblivious to the underlying structure imposed by the connectivity of the mesh topology. In [55], the geodesic metric is used successfully in modelling facial topology, while the deformations of the facial mesh were inferred within a Bayesian regression framework. Figure 4.2 illustrates how geodesic distances can offer a more semantically informed metric. The Euclidean distance demonstrated by the yellow line does not properly capture the connectivity and thus the structure of the Orbicularis Oculi region. The geodesic distance coloured in red acknowledges the underlying structure.

### 4.2.2 The Isomap algorithm

Isomap [44] is a technique for mapping the global structure of high dimensional input data onto a low dimensional Euclidean convex subspace $Y$. An optimal $Y$ is one that best pre-
serves the geodesic distances and connectivity of the original input data, embedded within a nonlinear manifold. This mapping is accomplished by minimizing the following cost function:

$$E = \|D_G - D_Y\|_{L^2}. \quad (4.2)$$

$D_G$ is analogous to the neighbourhood matrix described in Section 4.1.1, and is defined over the graph $G$ of the input data. $G$ is constructed by connecting adjacent neighbours in the data set with an edge, with edge lengths being initialized to the Euclidean distances between neighbours in input space. $D_G$ preserves information pertaining to inter-point distances of the graph. The distance metric used in local neighbourhoods is defined by the input space Euclidean distance. The distance metric used between all other points is defined as the shortest path through common neighbours. The entries of the matrix $D_G$ are found as follows.

$$d_G(i,j) = \min(d_G(i,j), d_G(i,k) + d_G(k,j)). \quad (4.3)$$
If \( x_i \) and \( x_j \) are connected by an edge in \( G \), \( d_G(i,j) \) is initialized to their edge length. In the case that \( x_i \) and \( x_j \) are not connected in their neighbourhood topology, \( d_G(i,j) \) is set to \( \infty \), and the shortest path through the common neighbours \( k \) is found. \( D_Y \) in Equation 4.2 denotes the matrix of inter-point Euclidean distances in an embedding space \( Y \). Hence,

\[
d_Y(i,j) = \|y_i - y_j\|,
\]

(4.4)

where \( y_i \) denotes the optimal set of coordinate vectors for \( Y \).

The intuition behind this cost function is this: the length of the shortest path between two points in high dimensional input space, which may induce a curved trajectory, is approximated by a similar Euclidean length in low dimensional embedding space. Figure 4.3 illustrates this; the red line is the geodesic length, the blue line is the approximating Euclidean distance. As a result, the isometric mapping preserves the global structure of the data. To find an optimal \( Y \), the global minimum of the cost function in Equation 4.2 is acquired by applying multidimensional scaling [46] to the neighbourhood matrix \( D_G \). The coordinate vectors \( y_i \) are set to the largest \( m \) eigenvectors of \( D_G \). The \( m \)-dimensional Euclidean space spanned by \( y_i \) is the space that best preserves the topology of the original data.

### 4.3 Locally Linear Embedding

Locally Linear Embedding (LLE) [39] takes a completely different approach to manifold structure embedding that can be described as “think globally, fit locally.” LLE is based on the principle that if enough data points are provided to represent a smooth manifold surface, collectively analyzing local dense neighbourhoods of the surface can yield information about
the intrinsic global geometry. Based on the assumption that a data point and its immediate
neighbours approximate a locally linear patch of the manifold, LLE attempts to induce an
optimal low dimensional coordinate set $Y$ that best approximates the intrinsic geometry of
the manifold at its various local neighbourhoods. This is in contrast to the Isomap approach
where the global structure is represented directly through an inclusive matrix of geodesic
distances.

LLE solves two linear cost functions. The first characterizes the geometry of the local
patches by finding an optimal set of linear weights to reconstruct each data point $i$ in terms
of its $k$-nearest neighbours $j$. It is important to note that the weights are invariant to affine
transformations on their respective neighbourhood data points, and thus do not depend on
a particular frame of reference. The reconstruction error of each data point is measured and
aggregated pointwise as follows,

$$
\varepsilon(W) = \sum_i |\vec{X}_i - \sum_j W_{ij} \vec{X}_j|^2,
$$

subject to $\sum_j W_{ij} = 1$.

The optimized set of weights, holding information relating to local structure, are then

$$
\varepsilon(W) = \sum_i |\vec{X}_i - \sum_j W_{ij} \vec{X}_j|^2,
$$

subject to $\sum_j W_{ij} = 1$.
used in the second cost function to globally embed the manifold in Euclidean space. The Euclidean coordinates optimally preserving the collective local geometry are defined as

$$\Phi(Y) = \sum_i |\vec{Y}_i - \sum_j W_{ij} \vec{Y}_j|^2.$$  \hspace{1cm} (4.6)

One drawback of the LLE algorithm is its sensitivity to its only parameter, the specification of the $k$ nearest neighbours. Chang et al. [5] explore the case where the input data consists of two types of variation, varying intensity across a specific expression (i.e. from the neutral face to the apex of the expression), and various expressions. LLE performs well when there is only one sequence of intensity varying frames for each expression. However, when the input data consists of several sequences for each expression, a small $k$ renders the algorithm incapable of properly capturing intensity variation, while increasing $k$ results in a higher vulnerability for the algorithm to mix images of various expressions.

### 4.4 Laplacian Eigenmaps

Similar to LLE, Laplacian eigenmaps (LEM) [1] is another manifold learning technique with locality preserving properties. The locality preserving property guarantees that if two points are connected and their distance is weighted by $W_{ij}$, the mapping to a lower dimensional space will respect this weighted distance. The Laplace Beltrami operator, defined on Riemannian manifolds, is a differential operator whose eigenfunctions corresponding to its $k$ smallest eigenvalues provide an optimal mapping to $k$-dimensional Euclidean space. Let $Y = (f_1, f_2, \cdots, f_m)$ be the optimal low dimensional embedding, and $W$ be the matrix of weights between points in their manifold neighbourhood subspace, as determined by the
adjacency graph. The optimal choice of $f$ would minimize the following objective function

$$\sum_{ij} \| f(x_i) - f(x_j) \|^2 W_{ij}$$  \hspace{1cm} (4.7)

as it penalizes a mapping which does not respect the closeness of points determined by the weights $W_{ij}$. The objective function can be written as

$$\sum_i f_i^2 D_{ii} + \sum_j f_j^2 D_{jj} - 2 \sum_{i,j} f_i f_j W_{ij} = \text{tr}(F^T L F),$$  \hspace{1cm} (4.8)

where $D_{ii} = \sum_j W_{ji}$ and $L = D - W$ is the laplacian matrix. It follows that the $F$ which satisfies $\text{argmin} \text{tr}(F^T L F)$ is given by the minimum eigenvalue functions of $L f = \lambda D f$. An optimal $m$-dimensional embedding is then expressed as

$$x \rightarrow (f_1(x), f_2(x), ..., f_k(x)).$$  \hspace{1cm} (4.9)

### 4.5 Mapping vs. Embedding

The techniques discussed above discover an embedding $f : D \rightarrow \mathbb{R}^m$ where $D \subset \mathbb{R}^M, m \leq M$, rather than a continuous mapping $f : \mathbb{R}^M \rightarrow \mathbb{R}^m$ which is preferable. This has several consequences.

- Because the embedding is calculated with respect to the neighbourhood properties of the training set $D$, a new data point ($x_{\text{new}} \in \mathbb{R}^M$) $\notin D$ will not yield a well defined projection onto the embedding space without performing eigenvector recalculations of the neighbourhood matrix.

- Because the embedding is not defined over a space, rather over a data set, the problem
of finding a continuous inverse mapping from the low dimensional Euclidean space to the input manifold space is ill posed.

Several approaches have been proposed to overcome these disadvantages. Techniques such as Radial Basis Function (RBF) interpolation and kernel eigenfunction decomposition generalize the common NLDR algorithms to extend to new data points, while manifold charting and global alignment offer altogether new manifold learning algorithms that build a continuous low dimensional probabilistic space. We dedicate Section 4.7 to the latter, and present techniques of the former here.

In [2], existing NLDR algorithms are augmented with a kernel eigenfunction decomposition. Within the unified NLDR framework described in 4.1.1, a continuous kernel function $K(a,b)$ is chosen for each method, which yields the neighbourhood matrix $R$ according to Equation 4.1. $K(a,b)$ is constructed with respect to the similarity metric of the method in question. For the initial data set $D = x_1, \cdots, x_n$, the embedding coordinates are derived in the usual manner through the generalized eigenvalue problem,

$$Rv_k = \lambda_k v_k,$$

(4.10)

where $R$ is the matrix preserving neighbourhood relations. However, the embedding coordinates of a new point outside of the training dataset is derived by solving the kernel eigenfunction problem

$$K_pf_k = \lambda'_k f_k,$$

(4.11)

where

$$\lambda'_k = \frac{1}{n} \lambda_k$$

(4.12)

$$f_k(x) = \frac{\sqrt{n}}{\lambda_k} \sum_{i=1}^{n} v_{ik} K(x, x_i)$$

(4.13)
and \( p \) describes the empirical distribution over \( D \). Intuitively, given a new data point \( x_{new} \), its embedding with respect to the space created over \( D \) is given by the weighted sum of the existing coordinates \( v_i \), where the weights describe the similarity between \( x_{new} \) and each \( x_i \), as returned by the kernel function. This method avoids a computationally heavy recalculation of eigenvectors.

For the Isomap technique, the kernel function uses the notion of squared geodesic distances, and the embedding of a new point is given by

\[
e_k(x_{new}) = \frac{1}{2\sqrt{\lambda_k}} \sum_i v_{ik} \left( E_{x'} [D^2(x', x_i)] - D^2(x_i, x) \right)
\]

where \( E_{x'} \) is an average over the data set.

In addressing the problem of deriving an inverse mapping \( f^{-1} : D_Y \rightarrow \mathbb{R}^M \) where \( D_Y \subset \mathbb{R}^m, m \leq M \), RBF networks have been employed to produce successful results [43]. Given an explicit set of correspondences between the high dimensional data and their respective low dimensional embeddings, a generalized RBF network learns the interpolation weights and thus approximates an inverse mapping. Elgammal has also reported successful results using RBF interpolation for mapping, and inverse mapping, between manifolds and low dimensional spaces [15].

### 4.6 Manifold Alignment

Often NLDR analysis is performed on a subset of the data that share semantic or perceptual similarity, and are sampled from the same underlying manifold. The choice of this subset depends on the nature of the variation that is to be classified. For example, the emphasis may be on the modelling of individual faces. In this case, the manifold will embed the range
of expressions across a single subject. On the other hand, if the emphasis is on modelling classes of expressions, the manifold may embed the cluster of similar expressions across various faces. Still, the emphasis may be on modelling the local behaviour of distinct regions of the face. For many applications it is desirable to determine an explicit relationship across the set of local manifolds, and encompass them within a unified space for global analysis, search and manipulation. Manifold alignment then becomes a requisite for such a global space.

*Manifold alignment* refers to mapping a correspondence among low dimensional embeddings of multiple data sets. Often these data sets are sampled from the same underlying manifold. Consider for example, two spaces that embed various facial expressions for two individuals. Given a specific point (e.g. expression) $p_A$ in space $A$, manifold alignment is concerned with the problem of finding a point $p_B$ in space $B$ with the highest semantic congruence to $p_A$. In [5], this problem is resolved by applying a nonlinear transformation to reference points (e.g. key expressions) of disparate embeddings such that the expression vectors align to a common vector. In their experiments, this common vector was a vector of ones. The mapping for the remaining data points are interpolated based on the mapping of the selected reference points. The reference points are chosen such that they convey the structure and axes of the embedding. In [47], non-rigid transformations are used to warp disparate manifolds to yield a unified *mean* manifold, while correspondences are established between manifolds by re-sampling splines fitted to data residing on each manifold. However, this approach assumes a strong correlation between how two individuals carry out an expression, and ignores the possible significant variation in the *styles* with which individuals perform expressions.

Ham et al. [22] propose two semisupervised approaches to manifold alignment, both relying upon user specified information for a subset of the data points, to create a semantic
correspondence between the spaces of disparate embeddings. In the first approach, a labelled subset of corresponding points from both manifolds are mapped onto known coordinates of a common space, illustrated in Figure 4.4. In the second approach, an explicit pairwise correspondence is specified between the two manifolds for a subset of the data, illustrated in Figure 4.5. The mapping for the remaining data points are determined by a cost function that minimizes the fitting error and enforces smoothness.
4.7 Probabilistic Manifold Modelling

Dimensionality reduction has its roots in modelling a continuous mapping between a high dimensional and low dimensional space. However, the recent NLDR approaches, described above, have instead emphasized finding a low dimensional embedding of a subset $\mathcal{D} \subset \mathbb{R}^M$. Parallel to the development of embedding algorithms, several techniques have been proposed to represent a low dimensional continuous space of a nonlinear manifold within a probabilistic framework. As a natural byproduct of this continuous modelling, problems associated with inclusion of new points and inverse mapping are alleviated. There also exists a computational advantage to such models, as large eigenvalue problems are bypassed. We now turn our focus to such techniques.

4.7.1 A general framework

Probabilistic NLDR is structured as a local model, that is, it upholds the assumption that at a certain locality scale, the manifold approximates a linear subspace, and hence the local mapping from $\mathbb{R}^M$ to $\mathbb{R}^m$ is linear. Accordingly, local neighbourhoods are explained by models which maximize their likelihood. This yields a mixture of models, each optimally representing a local subspace of the manifold. The problem of global probabilistic NLDR is now stated as follows: the intrinsic global coordinates of the manifold are formulated as the hidden variables that maximize the likelihood of a given set of observed variables, often relating to parameters of the local models, and respects a consistent projection of a given point by independent local components that have non-negligible responsibility for that point. In other words, a point may have different local coordinates with respect to multiple disparate local components, but the mapping from each local subspace to the global space must yield consistent global coordinates for the point.
As [38] notes, probabilistic NLDR simultaneously unifies the goals of manifold learning and density estimation. However, there is a trade off between the two. To illustrate, imagine a subset of the data that is perpendicular to the smooth underlying manifold, as in Figure 4.6. The model fitted to this subset must also be perpendicular to the manifold in order to maximize the local likelihood of the data. However, such a model with perpendicular dominant axes will induce an inconsistency of projection of nearby data points with adjacent local components. Hence, local optimality must be balanced with global consistency.

4.7.2 Algorithms

Within the general framework stated above, [3] fits a Gaussian Mixture Model (GMM) to the locally linear neighbourhoods, with the eigenvectors of the covariance matrix parameterizing each local component, or chart. A consistency prior based on minimizing the cross entropy between GMM components is imposed to yield maximal alignment of their dominant axes, and as a result, a consistent projection of nearby points between adjacent components. While employing this prior trades off local optimality for global consistency, it provides powerful means for suppressing noise, as GMM components are discouraged from fitting to outliers and degenerate data patterns. The manifold is now approximated by a series of
aligned local subspaces, but it yet remains to acquire a single unified representation. To accomplish this, a mixture of $k$ affine transforms maps each chart onto a unified global space. The set of transforms is solved for in a least squares manner, by minimizing the distance between projections of a given point by adjacent charts, within the global space. The global coordinates of a point $x_i$ are then given by a weighted sum of the transforms of each chart that holds a non-zero responsibility for $x_i$. Although an exact inverse does not exist for this mapping, a pseudo inverse can be obtained by integrating out uncertainty over which chart(s) generated a given point, mapping a point back to the weighted mean $\mu_i$ of its GMM components.

Roweis and colleagues [38] model local neighbourhoods with Mixture of Factor analyzers (MFA), and derive a global model by completely aligning the mixtures. Let $s$ represent the various manifold neighbourhoods, and $z_s$ represent the local coordinates of each MFA component. The global coordinates $g$ are then expressed

$$P(g|s, z_s) = \delta(g - A_s z_s - \kappa_s),$$

with $A$ being the coordination weights between adjacent neighbourhoods, and $\kappa$ being a translation factor. They arrive at an objective function which trades off local maximum likelihood with component conformity simultaneously. Based on the principle that disparate components should project the same point to the same global coordinates, a regularizer term composed of the Kullback-Leibler divergence in the objective function penalizes the multimodality of MFA posterior distributions. The distributions of the K-L term includes the true multimodal posteriors $P(g, s|x_n)$ and a family of Gaussian unimodal posteriors $Q(g, s|x_n)$. By minimizing the divergence of the two, global alignment between local models is achieved. It is through this complete alignment that one can traverse the global manifold.
structure continuously. The exact parameters of the transformations required to align the models are acquired through an iterative expectation-maximization (EM) optimization of the objective function. Note that this is in contrast to [3] in which absolute alignment between local models is not required.

4.8 The Facial Manifold

Images of the face as well as motion vectors of 3D marker data lie in unnecessarily high dimensions which are infeasible for the animator to control. It becomes critical to obtain an abstraction of facial data into parameters that control expressions, phonemes, muscle activations, pose, lighting, just to name a few. The face has successfully been modelled as a manifold in 2D and 3D applications. Nonlinear analysis on the face manifold has yielded powerful results in extracting structure with respect to facial activity/expressions, and has thus served as a framework for clustering similar expressions and visualization of the expression space. A discussion of these results will follow shortly in Section 4.8.1. Such analysis is further motivated by the observation that temporal expression processes trace out a 1D curve in low dimensional space, parameterized by the amplitude of the expression. The surfacing of such structural patterns is theoretically intriguing. However, in order to gain a deeper understanding and manipulation handle of the facial dynamics for practical applications, manifold embedding and dimensionality reduction can be employed in conjunction with other techniques. Once in this low dimensional embedded space, a number of operations can be performed on the face points, including learning shape models within the clusters, deriving temporal dynamics from transition probabilities, as well as synthesis and animation through inference.

Decisions to be made in analyzing the facial manifold relate to locality and choice of
A universal facial embedded space is one that models the spectrum of facial degrees of freedom in its entirety. Often, a low dimensional embedding is constructed from a subset of the universal facial manifold, resulting in a space that embeds structure with respect to a certain perceptual class of parameters, depending on the application at hand. We will now discuss issues pertaining to structure and locality.

**Structure**

It is important to determine a priori what perceptual class of structure, statistics, and parameters we wish to model. Is the low dimensional embedded space to be parameterized based on phonemes, or expressions? Are we concerned with embedding a temporal process in the physical world, such as the contraction, rest, release lifespan of muscle activity, or a static series, such as modelling the happy face space across several individuals. Does a single manifold capture the transitions among a wide range of expressions, or are temporal dynamics within each category of expressions embedded in an expression specific manifold? These illustrate the important considerations relating to structure that must be taken into account before constructing and analyzing a facial manifold.

**Locality**

Questions pertaining to locality are concerned with whether a single manifold represents the global face, or is each constituent region embedded and processed with an independent space? If a localized analysis is adopted, based on what set of criteria are facial regions identified to avoid correlation between neighbours?

An argument that is in favour of operating on individual regions of the face is that, depending on the structure of interest, dimensionality may vary across separate regions.
For example, if the structure of interest is AU activation, the eyelid has only one degree of freedom, whereas the corners of the lips can simultaneously be under the influence of several muscle contractions. In cases where decisions of locality lead to several independent embedded spaces, coordination algorithms may be employed to align these subspaces onto a generalized manifold, such as that suggested by [42].

4.8.1 The Facial manifold in research and applications

Facial manifold embedding has been applied to applications of facial recognition, tracking, expression synthesis, and animation interfaces. Fidaleo was the first to employ manifold analysis to model non-rigid facial deformations as a biometric for facial analysis [19] [20]. Although he uses linear PCA to analyze facial data, the principles of his research related to gesture manifolds are quite relevant and can be extended to the non-linear analysis case. Gesture manifolds, or G-folds embed all possible gestures and their intensity levels, from neutral to apex, for a particular co-articulation region [21] of the face. PCA reduces the high dimensionality of the local facial dynamics, giving rise to a region space that is parameterized by the various gestures. Each gesture is then further projected onto $\text{PCA}_2$ space, along its two greatest eigenvectors. The progression of the gesture can be seen as a curve in this space, parameterized by muscle activation, or intensity, of the gesture. This curve is approximated by a parabola. As a result, a set of continuous polynomials were derived, each modelling the appearance properties of a given gesture as it progresses from the neutral face to the gesture apex. For applications of mapping motion capture data to a character, the intensity of a new gesture can be easily identified by computing its orthogonal projection onto the gesture polynomial.

Chang et al. [5] assess the efficacy of a manifold embedding as a unified framework for analytical representation of all possible facial expressions. They perform both LLE and Lip-
shitz embedding on a manifold of faces, and report that while LLE returns favourable results for visualization purposes, Lipshitz embedding was more suitable for representational purposes. Within a \textit{Lipschitz embedding}, a coordinate vector is associated with each independent subset of the input data, which hold points with semantic similarity. In the context of faces, each subset is associated with images from the apex of one of \( k \) primary expressions. In their experiments, \( k = 6 \). A series of frames relating to a single expression with time varying intensity will project a curve on the manifold, between the central neutral expression to the respective apex. For visualizations purposes, the Lipschitz embedding for \( k = 3 \) is illustrated in Figure 4.7, and can be seen to approximate a sphere. It is noted that for higher values of \( k \), the Lipschitz embedding approximates a super-spherical manifold in \( k \)-dimensions. In this space, expressions with high perceptual correlation map to local neighbourhoods, and new expression blends lie on the interpolated surface between the expression curves.

This analysis provided in Chang’s work\cite{5} is extended in \cite{6} within the domain of video data, where a probabilistic model is learned over a unified Lipschitz embedding of various
facial expressions. Once the transitions between facial expressions are learned, target expressions on the manifold can be searched for by their likelihood to satisfy user requirements and constraints. The deformation vector of the embedded expression can then be transferred to a new subject in a manner similar to the deformation transfer techniques presented by Sumner and Popovic in [41].

In both [5] [19], it is observed that the manifold projection of perceptually identical facial activity for different individuals leads to curves which share a high level of structural similarity. This demonstrates the success of manifold embedding in extracting the global structure related to each facial expression. Although these curves are quite similar in nature, they are not identical and each one has subtle distinct characteristics, leading to the reasoning that aside from universal structure, manifold embedding also extracts the unique details of each data set. In an animation context, these properties of manifold embedding can be used, for example, to model and transfer the unique facial dynamics of an acclaimed actor.

Deng and Neumann [11] have developed a data driven facial animation system that applies Isomap to a cluster of motion capture frames for a particular phoneme, articulated with different emotions. These clusters, termed phoneme-clusters, are initially projected from motion capture space onto PCA space. The Isomap framework is then applied independently on the PCA representation of each phoneme-cluster, embedding the cluster in a two dimensional phoneme specific manifold which distributes the points according to their articulated emotion. A Gaussian kernel point rendering offers animators an interface to the Isomap space. It is reported that Isomap coordinates were often correlated with perceptual variations of the facial state, such as increasingly opening the mouth. This phenomenon is testament to the numerous advantages of NLDR embedding, relating to a higher level modelling of the underlying structure, as well as the grounds for a more intuitive user interface.

In [24], Hu et al. take a more global approach and embed 2D video feature marker motion
across different individuals and expressions in a single Isomap space. Their results show that points were distributed in Isomap space predominantly based on the expression variation, rather than individual actor variation. However, to reason the distribution further, a Gaussian Mixture Model (GMM) with Expectation Maximization (EM) is applied to the Isomap embedded points to distinguish the expression classes. Local structure is then extracted by training an Active Shape Model for each GMM. GMM clusters also allow temporal facial dynamics to be learned at two levels, one for dynamics within a cluster and one for transition probabilities between clusters. Their application is related to facial tracking and recognition.

An interesting application of Isomap is presented in [40], in an approach to facial synthesis termed facial expression hallucination. The objective is to synthesize an expressive face (e.g. the happy face in their experiments) for an individual, given just her neutral face, in the domain of static images. This is accomplished by learning an Isomap embedding over a manifold of neutral faces, and one over a manifold of expressive faces. This unfolds both manifolds onto Euclidean subspaces. Multiple linear regression is then employed to learn a matrix of parameters $\gamma$ defining the relationship between faces in the neutral space, and their corresponding faces in the expressive space. Given a new input neutral face, its low dimensional coordinates in neutral space are expressed as a weighted combination of its nearest neighbours in face space, and subsequently these coordinates are transformed by $\gamma$ to synthesize, or “hallucinate” its corresponding face in the expressive space.

4.9 Summary

In this chapter, we have discussed the importance of nonlinear analysis in cases where a linear subspace of Euclidean space does not accurately model the data at hand. We presented the general problem of Nonlinear Dimensionality Reduction (NLDR), and surveyed two formal
approaches to NLDR, manifold embedding and probabilistic mapping. We discussed specific techniques from each approach in the context of a unifying framework. Finally, we surveyed recent applications and evaluations of NLDR for representing face data. In the next chapter, we make a case for extending the application of NLDR to represent the dynamics of temporal facial expressions.
Chapter 5

Spatio-Temporal Facial Dynamics

Learning

"Space then is a necessary representation a priori, which serves for the foundation of all external intuitions." -Immanuel Kant

In this thesis, we support the idea that the complexity of the face comes from nonlinear interactions between a small number of AUs, rather than the conventional method of creating a large library of AUs. We now turn our attention to a framework for representing facial expressions in a linear space spanned by independent AU components. We show how this AU space lends itself to a nonlinear representation of the dynamics of facial motion.
5.1 The Linear and Nonlinear Spaces

In the following chapters, we describe how the continuous expression of the human face can be viewed in light of a linear component and a nonlinear component. In this section, we provide a brief description and definition of these components and their characteristics. We develop upon these definitions more elaborately in later sections, and show how facial data resides in these spaces.

We define the linear space of facial expressions, $L$, as one with a near-orthogonal basis that spans the possible facial AUs. Then, any static facial expression $E$ can be expressed as a combination of $m$ basis functions $b(x)$ where $x \in R^n$, weighted by the weight vector $w$;

$$E = \sum_{i=1}^{m} w_i b_i.$$  \hspace{1cm} (5.1)

Facial communication, however, is more than the collective static expressions at each frame in time. Understanding the evolving human face requires building a temporal coherence across these frames, such that a finite group of static expression frames can be perceived to communicate a common statement of emotion. Analyzing the temporal patterns between consecutive expression frames would be more manageable if we were to analyze them with regard to the independent components of the expressions, their AUs. To visualize the dynamics of facial motion for a particular temporal expression, for example neutral to surprise, we project each static expression frame in the neutral to surprise sequence onto the linear AU space. Each static expression frame then becomes a point in this space, described by the activation and amplification of its component AUs. The activation and progression of the component AUs in the expression sequence may occur at different times and with varying speed. Hence, mapping one expression frame to the next traces along a curved trajectory. In Section 5.3, we explain how trajectories of similar expression sequences map to a curved
We define the nonlinear subspace of facial motion, $N$, as the curved manifold residing within linear AU space, $L$, whose geometry and curvature models the AU dynamics of temporal expression sequences. We seek a model that inherently adapts to the geometry of this manifold using the best possible spanning representation. In this pursuit, we investigate NLDR techniques.

5.2 Creating the Action Unit Basis

5.2.1 The PCA algorithm

PCA is a data representation technique whose potency lies in deriving a basis that respects the patterns of the data lying on a linear subspace. Formally, these patterns can be thought of as the range of variance caused by a hidden variable along one degree of freedom. Hence, each axis of the PCA basis is aligned with the data such that it optimally captures this variance. The optimal axes are in fact, the eigenvectors of the input data’s covariance matrix. The general PCA algorithm is now summarized. Given a row specific data matrix $M$ that has undergone any specific pre-processing requirements,

1. Calculate the mean along all dimensions in input space, produce mean vector $\Psi$. 

2. Subtract $\Psi$ from each data point and create mean-subtracted data matrix, $M_{Inp}$. 

3. Find the covariance matrix, $C$, of $M_{Inp}$. 

4. Conduct an eigen decomposition to find the eigenvalues and eigenvectors of $C$. 

5. Retain and order the eigenvectors corresponding to the top $k$ eigenvalues, according to the desired reconstruction error tolerance. The basis $B$ for the new PCA space is then
constructed as the set of $k$ eigenvectors $E$ as such $B = [Ev_1, Ev_2, \cdots, Ev_k]$.

Upon deriving the basis spanning the PCA space, mean subtracted data can be represented with respect to this basis through the following projection:

$$M_{pca} = M_{inp} B,$$  \hspace{1cm} (5.2)

while the inverse projection

$$M_{inp} = (M_{pca} B^{-1}) + \Psi$$  \hspace{1cm} (5.3)

yields the input space coordinates of PCA data. Because the basis vectors in $B$ are orthogonal, $B^{-1}$ can conveniently be replaced by $B^T$.

Let us assume there are $n$ intrinsic hidden variables controlling the variance in the data, where $n$ is considerably less than the input space dimension. Because the basis is constructed with respect to the variance generated by these $n$ hidden variables, we can represent the data within a threshold error $\epsilon$ with the eigenvectors associated with the $k$ most significant eigenvalues. Our intent here is twofold; obtaining a compact representation of a high dimensional input space, and finding a more meaningful basis which parameterizes facial motion.

### 5.2.2 The Data

We begin with the data set of facial motion $D = \{s_1, s_2, \cdots, s_n\}$ where each sequence $s_i = \langle f_1, f_2, \cdots, f_n \rangle$ is a coherent progression of $n$ frames, from the neutral face to the apex of a particular expression. Each frame $f_i$ constitutes the absolute positions of landmarks, which can be vertices of a mesh, or markers used in motion capture. In our experiments, the source of our data is a facial mesh from the Institute of Animation [32], animated with a blendshape basis constructed with the aid of motion capture data, as part of their Facial
Animation Toolset, provided as an extension to Maya. Our input data at each frame consists of the positions of the mesh vertices. The data set $D$ should contain sufficient and diverse sequences in order to span the desired facial configurations. PCA yields stronger results when $D$ contains sequences representing primary facial DOFs in isolation, in addition to more complex combinations of them. In other words, suppose our hidden variables correspond to musculature action which produces one basic unit of perceptible facial motion, such as a raised eyebrow. The axis to AU correspondence is stronger when there is one sequence of “raised eyebrow” in isolation along with sequences of “raised eyebrow” in conjunction with other AUs in other regions, rather than when every instance of “raised eyebrow” coincides with another AU in the data set.

If human perception were able to resolve to minuscule details of facial displacement, given the exponential measure of muscle contraction combinations, our biological and psychological facial communication system would be much different than our current architecture. There is strong established evidence from neurology and cognitive science that the human perceptual system employs an intrinsic clustering functionality in processing information, that allows us to group similar signals, and process them as a semantically unified block. In fact, much of the science of machine learning is based on this understanding. The human perceptive system has been observed to have a similar mechanism for processing facial expressions. We analyze facial action through its high level behaviour, rather than low level displacement measurements, such that, for example, two slight variations in the angle between the lower and upper lip in a “jawOpen” action will be clustered together semantically to communicate the same expression. It is in fact this type of high level perceptual clustering that public standards and parameterizations (e.g. FACS) as well as expression recognition systems aim to emulate. However, as interpreted within a Euclidean-metric PCA framework, even minute variations such as the lip angle mentioned above may be strongly distinguished. As a result,
different eigenvectors could emerge to account for semantically similar expressions having different Euclidean representations. In our experiments, providing PCA with a stream of facial expressions exhibiting natural mouth movements resulted in perceptual redundancy corresponding to the first few significant eigenvectors.

To alleviate this problem, we limited our motion data to animation constructed from consistent blendshape actuations, such that a facial action is always performed in the same manner. Unfortunately, this is not possible when working with raw mocap data from an actual actor. Two instances of performing an AU most likely will not map to identical displacement vectors in Euclidean space. In such cases, we suggest clustering techniques, such as Mixture of Gaussians (MoG), to be applied to the data stream as a preprocessing step. By applying a clustering technique, displacements which are within a certain tolerance of each other will be categorized under the same expression. The average of each cluster can then be taken to represent the expression. This eliminates redundancy in representing expressions which are perceived as the same to the human visual system, but exhibit minor displacement discrepancies when projected onto Euclidean space.

The PCA algorithm, as outlined earlier, is then applied to the data set with a slightly modified step 1, that we motivate by means of an example. If the data set contains a preponderance of a particular type of deformation, such as a wide open mouth, the average of the data, Ψ, will be biased toward that deformation. As a result, the biased Ψ will be mapped to the origin of the PCA space, and furthermore, all variation will be tabulated with respect to the bias. In our experiments, it proved inevitable to obtain an “open mouth” bias due to several factors. The lower region of the face has more degrees of freedom than the upper region. A larger number of muscles around the mouth region, the jaw, and an array of various phonemes account for the higher count of DOFs which must be spanned in the training set. Additionally, lower face deformations generally have a greater magnitude than
upper face deformations, hence weighting the neutral face toward the “open mouth.” Figure 5.1 illustrates the “open mouth” bias on the activation of the “half smile” on the left face, and the “half smile” without a bias on the right face. A semantically sensitive basis should not combine the mapping of two independent AUs onto the same axis; however, when the neutral face has an open mouth bias, the bias naturally propagates to all facial deformations. Similarly, figure 5.2 illustrates a “smile” bias on the “jaw open” action in a data set in which faces were heavily biased toward a smile expression. To account for this, we explicitly provide the neutral face to the algorithm as $\Psi$. This ensures that the actual neutral face, not the average of the data set, is mapped to the origin, and serves as the foundation to which all additive deformations are applied.
5.2.3 Analysis of PCA behaviour

In order to analyze and properly utilize the resulting PCA space, we must remind ourselves that PCA is a statistical technique. Our aim in employing PCA is to obtain a space that has the following properties:

1. The axes represent the DOFs parameterizing facial motion, and furthermore extracted DOFs correlate closely to what one would perceive as a primary facial AU.

2. The space spans the complete range of expressions in the convex bounding volume of the axes.

The latter part of the first property is subjective and is clearly difficult if not impossible to characterize precisely. Its purpose is to ensure that the animator can intuitively construct faces as a weighted combination of AUs and easily navigate the face space. However, how does one quantify the success of an axis correlating closely to what one would perceive as
a primary facial AU? What is certain is that humans do not perceive facial deformation as a vector field, with certain magnitudes and directions of motion. Consider the FACS AU “lip corner puller” which is an activation of the *Zygomatic Major* muscle, and the FACS AU “lip puckerer” which is an activation of the *Incisivii labii superioris* and *Incisivii labii inferioris* muscles. These AUs neither communicate the same emotion, independent of intensity, nor do they activate the same muscles. However, if we start with the “lip corner puller” configuration and apply a vector field that continuously transforms the lip and mouth vertices inward, we will in fact arrive at the “lip puckerer” configuration. Within a statistical purview, the variance caused by this deformation can be navigated along a single axis.

Clearly, there is a disadvantage of perceptual contradiction associated with PCA’s behaviour in deriving statistical axes. This requires understanding and adaptation on the part of the user. In return, however, mapping mathematically opposite facial behaviour to different regions along the same axis has several strong advantages.

First, this behaviour acts as natural constraint to the space of faces by disallowing combinations of two deformations in opposite directions. The potency of this behaviour can be appreciated more in light of examining face spaces constructed through other methods, in particular manual blendshape building. Often additive combinations of deformations in opposite directions yield unnatural, incommunicable, or vague expressions. In addition, PCA enforces efficiency in constructing an expression with as few active AUs as possible. *Frown* and *Brow* are examples of deformations in opposite directions that do not necessarily produce unpleasant results if they are both active, but in most implementations of blendshape spaces, activating one simply mitigates the intensity of the other. Mapping both AUs to the same axis discourages activating two parameters when the same result can be achieved by a single parameter. In complex animations where numerous AUs may be active simultaneously, this constraint yields a more succinct representation and avoids unpredictable results.
due to a high number of AU interactions.

Second, there is an implicit dimensionality reduction of spanning two perceptual AUs with one axis. We say *implicit* to differentiate this form of dimensionality reduction from the explicit form of retaining the $k$ significant eigenvectors.

Understanding the discussion above allows one to see how *left smile, right smile* and frown, raise brow may be mapped to opposite polarities of the same axis. However, not all axes derived through the PCA algorithm will have positive and negative polarity within the constrained space of feasible faces. Such axes represent deformations that are unidirectional with respect the neutral face. An example is the open jaw, as there is no vector field opposite to the deformation of the open jaw that can be applied to the neutral face to produce plausible expressions.

### 5.2.4 Multiple PCA spaces

Under many circumstances, it is beneficial to make use of multiple PCA spaces. In our experiments, we have observed that PCA exhibits greater stability when the effective parameter space controlling the input data is constrained. Specifically, we have noticed an undesirable tendency of the PCA algorithm as the number of intrinsic degrees of freedom in the data set increase. Because PCA ranks the significance of a DOF with respect to the value of its corresponding eigenvalue, facial AUs are rendered less significant as the magnitude of their deformation in Euclidean space decreases. This is quite problematic as eigenvalues do not necessarily indicate the significance of an AU’s contribution to human expression. In our experiments, the most significant principal component usually correlates to the deformation of the *open jaw*. On the other hand, an action unit, such as an *eye squint*, whose semantic value is arguably just as significant as the *open jaw* in many contexts, yet whose displacement has a minor magnitude, may be assigned a considerably smaller eigenvalue. In addition, it
may be intuitively beneficial from a user interface point of view to have dedicated spaces for regions of the face, such as a Mouth Shop or an Eye Shop. To realize these ideas, we adopt a local PCA approach, leading to a space representing upper face “eye” motion, and one representing lower face “mouth” region. As a result of this parameter space reduction, our results show that smaller deformations are better preserved, and PCA returns axes that more accurately represent AUs. The result of performing PCA on the Eye and Mouth region can be seen in Appendix A. The left images represent the face mapped to the minimum of each principal component, while the right image represents the maximum.

Another realization of the multiple PCA space approach is hierarchical PCA, although we did not implement this. Different PCA spaces can represent levels of expression details. For example, a base PCA space defining generic/common AUs can be used to model expressions for a cast of characters. Then each character in turn may have his own specialized PCA space representing nuances such as a dimple, lower eyelid squint, or even a range of expressions that are a result of particular skin-musculature interactions, including wrinkles and asymmetry.

5.2.5 Augmented “X-space”

From a user perspective, a localized approach provides the animator with an organized and dedicated space for various regions of the face. However, the integration of local spaces within the animator’s toolbox would be of negligible value if a holistic animation handle were compromised. To compensate, we augment the local PCA spaces to form an expression space, or X-space, that provides the user with a unified handle to create and animate global expressions. The animation procedures discussed in the future sections are carried out with respect to X-space.

An X-space is constructed by conjoining local subspaces from each regional space, as
expressed in Equation 5.4:

$$X = (E = PC_1, \ldots, PC_n \subseteq PCA^{Eye}) \cup (M = PC_1, \ldots, PC_g \subseteq PCA^{Mouth}). \tag{5.4}$$

By restricting $E$ and $M$ to proper subsets, the X space can be customized to the AU requirements of a particular expression, and its dimensionality reduced to the minimum essential dimensionality needed to represent the expression of interest. In order to have a consistency of scale across local spaces, the active subspace of each principal component in $PCA^{local}$, representing the range of variance of the training data along each dimension, is mapped to $[0, 1]$, or $[-1, 1]$ for bipolar axes (see Section 5.2.3), in $PCA^X$.

To invert data from the augmented X space to input space, the subspaces $E$ and $M$ are inverted through their respective parent PCA basis, to obtain $Input^{Eye}$ and $Input^{Mouth}$. The local deformations specified in each of the input space faces are then applied to a common base face. Since there are few if any regions of the face covered by both spaces, there is minimal risk of undesired extrapolation during the combining phase.

## 5.3 Expression Manifolds

By projecting facial motion onto the PCA AU space, we conjecture that continuous facial expression is a temporal process of nonlinear interactions among primary action units, and that a continuous sequence of expressions can best be approximated by piecewise gain curves in AU space. This nonlinearity is illustrated the expression trajectories in Figure 5.3. The space, shown from two different angles, is spanned by the AUs $RaiseBrow$, $OpenMouth$, and $Smile$, whose frame sequences from neutral to apex, represented by the red, green, and blue points, are mapped to the first three principal coordinates. The black points
represent expression frames from five other sequences, each made with a unique and natural combination of the three principal components.

The fact that continuous facial expressions incur a sinuous trajectory onto the AU space is a testament to the potency of a PCA representation for analyzing, organizing, navigating, and manipulating facial animation. Given a set of handles to the expression curves, an animator may easily alter or fine tune the interplay among multiple AUs, or scale the influence of a particular AU.

We now extend the analogy of “a single sequence of continuous expression frames to a curved trajectory” to “a family of continuous expression sequences to a manifold”. Depictions of human emotion that are semantically related exhibit trajectories that not only have highly overlapping AU activations, but also have similar interactions among their AU components. Then the collective trajectories of these emotions occupy a manifold in the convex bounding volume of their respective AUs.

Among the simpler types of facial behaviour are those in which the activation of one
AU is accompanied by the activation and release of another AU. Common examples include raising and lowering the eyebrows during a laugh, or opening and closing the jaw as in a yawn, while closing the eyes.

Variations of AUs correspond to slightly different emotions being communicated or different styles of expression. By allowing an expression trajectory to span one or more “style parameters”, we obtain a sampled manifold that models such variation.

Figure 5.4 illustrates a surprise expression sequence in the Smile, OpenMouth, RaiseBrow space considered above. The expression includes raising and lowering the eyebrows (represented by the red axis), while smiling (represented by the blue axis). Here we have added an “open mouth” style parameter (represented by the green axis) to model the surprise factor of the expression class. As the expression trajectory spans the green axis, the amplification of the “open mouth” increases, and gives a more surprised look. The points in magenta are the static frames of the surprise sequences, approximating an underlying parabolic manifold. We formally define an expression class as follows: An expression class is a constrained subset of nonlinear interactions between multiple AUs that define semantically related expression processes. Every point on the generated manifold is anatomically plausible or expressionistically communicable, depending on whether the goals of the animator are realism or caricature.

5.3.1 Expression manifold types

While the range of possible expression manifolds are as multifarious as facial communication itself, by examining more complex examples we hope to develop an intuition about the shape and properties of such manifolds as they relate to dynamic facial behaviour. We consider manifolds in three dimensions for the purpose of visualization, although many natural ex-
Figure 5.4: Parabolic Manifold
pression manifolds may reside in higher dimensions.

**Gaussian Dome**

In the following example, we model various modes of kissing, stylized with smile and wink. As can be seen in Figures 5.5 and 5.6, the smile and wink AUs have similar relationships with the kiss AU. Based on this similarity, the manifold revolves around the kiss principal component, and takes the shape of a slightly irregular half dome, shown in Figure 5.7. The foundation of the manifold can be seen through the following three trajectories. Figure 5.5 maps the left wink and right wink against kiss. The left wink and right wink are represented through the negative and positive halves of the same Eye space principal component. The expression is modelled such that the character first initiates the wink, then follows naturally with the kiss. Figure 5.6 maps the smile against the kiss. The slope of the initial progression of the kiss in Figure 5.6 is sharper than that of Figure 5.5. This is because the smile and kiss AUs act upon the same region, and care must be taken that the smile does not overwhelm the kiss AU. The sampled manifold can be seen in Figure 5.7, where the outer trajectories represent the left wink against kiss and the right wink against kiss. The middle trajectory represents the smile against kiss, and the remaining intermediary trajectories are weighted combinations of the kiss and wink stylizations.

**Freeform**

The next example models a character’s expressions from pleasure to disgust. In previous examples, the relationships defined between AUs across various trajectories could be related through a scale and/or translation factor. This type of consistency does not hold in the following example, and results in a freeform manifold shape. The initial trajectory representing pleasure, shown in Figure 5.8, includes a smile which gradually pushes the brows upwards (note that the progression of the expression flows from right to left, as the right most point represents the neutral face at the origin). As the smile reaches its apex, the brows are slightly...
Figure 5.5: Gaussian Half Dome Manifold, “Stylized Kiss, Wink trajectories”

Figure 5.6: Gaussian Half Dome Manifold, “Stylized Kiss, Smile trajectory”
Figure 5.7: Gaussian Half Dome Manifold, “Kiss stylized with Smile and Wink”
lowered. Both AUs lie in the negative quadrant of the principal components involved. Figure 5.9 illustrates the final trajectory, representing *disgust*. Here, the *upperLipRaise* and *frown* AUs proceed along more or less a parabola in the positive quadrant of their PCs. Figure 5.10 shows the intermediary trajectories. Note that *browRaise* and *frown* are represented by different polars of the same principal component.
Figure 5.10: Freeform Manifold, “Pleasure to Disgust”
5.4 Summary

In this chapter, we have analyzed the efficacy of PCA in extracting primary facial action units that drive an input stream of continuous facial motion, and have localized the application of PCA to the upper and lower regions of the face. We have introduced expression manifolds, nonlinear manifolds that reside in the linear PCA AU space, and model semantically similar expressions in terms of their constituent action units. Having such manifolds of expressions, the animator has the resources he needs to begin the facial animation of the character through the desired expression manifolds. However, what is required now is a technique for efficiently managing and navigating this resource. This task becomes more challenging as the structure and dimensionality of the manifold increases. In the next chapter, we address the problem of managing and navigating nonlinear expression manifolds through nonlinear dimensionality reduction.
Chapter 6

NLDR Analysis on Expression Manifolds

“It requires a very unusual mind to undertake the analysis of the obvious.” - Alfred North Whitehead

The challenges posed by data representation - the need for accuracy, efficiency, and intuitiveness - are central to the problem of modelling a natural phenomenon, in our case, temporal facial dynamics. In lieu of a workspace composed of disparate facial action units, such as a library of blendshapes, we proposed the use of smooth and continuous manifold structures that associate semantically similar expressions together. Although these manifolds are accurate in modelling temporal expressions, improvements can be made with respect to their representation. We present in this chapter an accompanying framework for analysis that respects the manifold’s curved geometry, and delivers a representation that renders the manifold more efficient and intuitive to manage and to navigate.
6.1 Assumptions and Requirements on the Manifold

NLDR operates successfully under certain assumptions relating to the uniformity and density of the manifold. The stringency with which these assumptions are met and the nature of how the particular NLDR algorithm functions if they are not, are slightly different for each method. Most local and global methods, however, require the manifold to be uniformly sampled (at some local level) and densely sampled to produce satisfactory results.

Consider a manifold which is well sampled along only one dimension. This situation can arise naturally in practice, such as when each independent trajectory is well sampled, but the manifold as a whole is not sampled at sufficient intermediary trajectories. Retrieving the $k$-nearest neighbours for a given point $x$ will most likely return points that are along the densely sampled dimension and (locally) collinear with $x$, with adjacent points indexing an almost identical neighbour set with one another. This results in negative consequences for both LLE and Isomap. LLE, in optimizing the embedding so as to emphasize the local structure whose scale is determined by the $k$ parameter, will collapse the structure of a collinear “neighbourhood” onto a single point. On the other hand, Isomap preserves global structure by identifying the connected components of its input graph. In a densely sampled manifold, global connectivity is provided through local adjacent neighbourhoods, and through this “connected whole” Isomap is able to compute the required geodesic distances. However, in the under sampled or unequally sampled cases described above, the global structure is distributed across numerous disparate components. Consequently, Isomap returns erratic and unexpected results in extracting global and continuous intrinsic coordinates in the absence of continuously connected local neighbourhoods. Undesired or deviant NLDR behaviour can be avoided if certain requirements on the distribution of sampled points are met. The stability and success of both local and global methods are based on the assumption that the manifold
is dense, and at some local scale, data points are distributed uniformly.

An additional assumption imposed by local methods including LLE and Laplacian is on the uniqueness of sampled points. Consider a manifold that is sampled at $n > 1$ instances at a particular point. The $k \leq n$ nearest neighbours for each redundant point will be itself, resulting in NaN values in the calculation of neighbourhood reconstruction weights. These NaN values result in a failure of the algorithm when the eigenvalue decomposition is performed. Unfortunately, this situation arises frequently in practice if numerous trajectories pass through a particular point, for example the ubiquitous neutral face. Expression classes are at high risk for this behaviour since in principle their trajectories follow similar patterns and AUs. To obviate this problem, we scanned our data stream and removed all data redundancies.

6.2 Spline Approximation of Expression Manifolds

Situations in which assumptions on manifold properties are violated arise frequently in natural facial motion data sets. To avoid the resulting undesirable artifacts, we make use of spline interpolation and resampling to derive a well behaved data set. Our aim in resampling the manifold is twofold: data amplification and uniformity. Spline modelling techniques for representation and smoothing operations on data sets are ubiquitous in computing. We begin our discussion on the matter by first providing some background and definitions.

6.2.1 Background: interpolation and resampling

In its broadest context, interpolation is defined as an informed estimate of the unknown [52]. Yet, a more elegant definition [45] views interpolation as a model based recovery of continuous data from discrete data within a known range of abscissa, given the following
The four postulations:

1. The underlying data is continuously defined;

2. Given data samples, it is possible to compute a data value of the underlying continuous function at any abscissa;

3. The deterministically recovered continuous data is entirely described by the discrete sample data;

4. The evaluation of the underlying continuous function at the sampling points yields the same value as the data themselves.

In other words, given an observed set of data points $z_i$ respecting a hypothetical continuous pattern, the objective of interpolation is to fit the function $f(x)$ to this pattern such that unobserved points in the continuous domain $(z_i, z_{i+1})$ of the underlying data set can be approximated. The general form of interpolation can be expressed as follows:

$$f(x) = \sum_{i=1}^{n} z_i R_i(x), \quad (6.1)$$

where the interpolated value $f(x)$ of a given point $x$ is expressed as a weighted superposition of the discrete set of sampled points $z_i$. The points $z_i$ are weighted by the functions $R_i(x)$, often referred to as **blending functions**. Within the structure of Equation 6.1, the freedom in interpolation lies in the choice of the blending functions. Popular and established choices of weighting functions include B-splines, Gauss, Hermite, Lagrange, Rom-Catmull, and NURBS. We employ the family of curves known as **nonuniform rational B-splines**, or NURBS, as the choice for $R_i(x)$. For an in depth discussion on NURBS, we refer the reader to [18].
In most practical cases, the rate at which the input space data is sampled from an underlying continuous function does not adhere to imposed requirements, such as that of frequency or uniformity. **Resampling** refers to the process by which the underlying continuous data is represented by the discrete set \( Y = \{y_1, y_2, \cdots, y_n\} \), derived by evaluating the interpolating function \( f(x) \) in Equation 6.1 according to requirements on the distribution of the data samples, such as a local or global sampling rate. The beneficial consequence of this operation is a well behaved distribution and frequency demonstrated in \( Y \), relative to the original data set \( Z = \{z_1, z_2, \cdots, z_n\} \).

### 6.2.2 Resampling of the manifold

Our approach makes use of a two-tier interpolation scheme. Consider the facial data points shown in Figure 6.1. These points describe the surprise manifold as explained in Section 5.3, and are composed of several trajectories, each modelling a certain variation or style of *surprise*. The colour of adjacent trajectories are alternated between red and blue, and the points are shown from two angles for ease of visualization. Initially, we approximate each trajectory \( n_j \) by fitting piecewise smooth interpolating NURBS to its data points, as illustrated in Figure 6.2. We will refer to the direction of the trajectories as the \( u \) direction of the manifold. The splines fit to the trajectories are then evaluated at \( u_j^m \) locations, where \( m \) is the number of evaluation points, typically specified with respect to the magnitude of the trajectory. In most of our experiments, \( m \) was experimentally specified by the user. For the data points in Figure 5.3, successful values of \( m \) ranged between 50 to 75. Figure 6.3 depicts the evaluation of all trajectories at 75 points. We have now obtained a well behaved distribution of points that meet intra-trajectory density and uniformity requirements in the \( u \) direction. However, the distribution of points in the \( v \) direction of the manifold is far from satisfactory. Let us define the \( v \) direction of the underlying manifold as the fields of
Figure 6.1: Raw Data Points
Figure 6.2: Fitting Splines to Trajectories

Figure 6.3: Evaluation of Splines in the $u$ Direction of Manifold
Figure 6.4: Densely Sampled Regions of Nearby Trajectories

Figure 6.5: Fitting Splines to the $v$ Direction

$m$ correspondences between the trajectories. By examining Figure 6.4, it becomes apparent that the manifold is heavily sampled in regions where trajectories are close to one another or near overlapping. Compare the density of the region emphasized by the rectangular frame with the density of the regions on the left of the image. To alleviate this problem, we fit another set of interpolating NURBS in the $v$ direction of the manifold, between corresponding $m$ points on the $u_j$ splines. The result is shown in Figure 6.5.

Finally, we evaluate each $v$-spline at $v_m^p$ points, where the number of evaluation points $p$ is determined as a function of the density in the $u$ direction. In other words, to obtain a uniformly sampled manifold, we evaluate the $v$-splines with the same ratio of sampling frequency to distance as the spline evaluation in the $u$ direction. This achieves our goal of
Figure 6.6: Evaluation of Splines in the $v$ Direction of the Manifold

global uniformity, as depicted in Figure 6.6. A top is shown in Figure 6.7 for comparison with Figure 6.4. The matrix of all points evaluated in the $v$ direction satisfies the general density and uniformity assumptions required by NLDR techniques. We now turn our attention to applying local and global NLDR to this matrix.

6.3 Performing Local and Global NLDR

We now analyze the NLDR maps of the expression manifold examples in Section 5.3.1 and make some important observations regarding the differences in performance of local and global NLDR. Each approach has certain benefits and disadvantages. Depending on factors such as curvature of manifold and sampling distribution, one technique may be found more preferable than the other.

We first consider the Gaussian half dome, modelling *Stylized kiss*. The shape and tra-
Figure 6.7: Evaluation of Splines in the $v$ Direction of the Manifold, Top View

Figure 6.8: Isomap on half dome manifold

jectories of this expression manifold in PCA AU space was examined in Chapter 5, and illustrated in Figure 5.7. After performing the sampling procedures discussed in Section 6.2.2, we applied the Isomap and LLE algorithms to the sampled manifold and obtained the maps in Figure 6.8 and 6.9, respectively.

A fundamental difference between Isomap and LLE can be appreciated by examining Figures 6.8 and 6.9. Both techniques extract the local neighbourhood properties. This guarantees that if two points are close together in manifold space, their proximity will be preserved in NLDR space. However, because LLE is not concerned with global manifold properties, there is no penalty for mapping distant points in manifold space to nearby points in NLDR space. This is well illustrated in Figure 6.9, where the progression of trajectories
emanating from the neutral face are not under global constraints and thus result in a higher arc degree than in the Isomap embedding and the actual manifold.

We next consider the *Pleasure to Disgust* manifold from Chapter 5. Figure 6.10 is a sampling of the manifold spanned by the trajectories in Figure 5.10, and resides in PCA AU space. The magnitude of the trajectories in this manifold are not equal, yet a global $u$ parameter is used to parameterize all trajectories. This has resulted in slight condensation in regions where the trajectories have a shorter length. Figure 6.11 and 6.12 illustrate the Isomap and LLE embeddings, respectively.

In this case, Isomap’s stringent distance preservation has transferred the manifold condensation onto the NLDR map. LLE, on the other hand, has returned a more uniform and smooth embedding that is easier to work with. The reason, in fact, is not simply that LLE is a local method, but is related to how LLE extracts local properties. Recall that the metric LLE attempts to preserve is not related to distance, but to the neighbourhood reconstruction weights of each points. There weights are invariant to scale. Because of this, LLE is not as sensitive to varying sampling rates across local neighbourhoods, and hence, the sampling
Figure 6.10: “Pleasure to Disgust” Freeform Manifold Sampled

Figure 6.11: Isomap on Freeform Manifold

Figure 6.12: LLE on Freeform Manifold
condensation is not carried over to NLDR space.

6.4 Inverse Projection

Current NLDR techniques that provide an embedding of data do not provide an explicit inverse function. Hence, the projection back to PCA space can at best be approximated. In NLDR literature, there are numerous references to scattered data interpolation techniques, such as the use of Radial Basis Functions, to solve this type of approximation problem. The general issues pertaining to NLDR mapping vs. embedding has been elaborated upon in Section 4.5. Here we offer an approach to inverse approximation that capitalizes on both global and local NLDR’s ability to preserve local properties. We state the inverse NLDR problem formally as follows. Starting with a high dimensional finite and discrete data set \( D = \{d_1, d_2, \cdots, d_n\} \), residing in \( R^M \), we perform NLDR on \( D \) and obtain a low dimensional embedding \( L = l_1, l_2, \cdots, l_n \) with dimensionality \( m \leq M \). Clearly every point in the embedded data set \( L \) has an explicit correspondence in \( D \). Suppose now, that we have a point \( y \) which is not in the data set \( L \), but resides in the convex hull of all points in \( L \). How can we determine the inverse projection of \( L_y \) to \( R^M \)? To summarize, we wish to construct a continuous space \( R^m \) within the convex hull of the embedded data set \( L \), and derive a mapping function between this low dimensional continuous space and its higher dimensional continuous space \( R^M \).

6.4.1 Approximating \( R^m \rightarrow R^M \)

Given any point \( y \) in \( R^m \) that is not a member of the discrete data set \( L \), but resides in some neighbourhood in \( L \), we reconstruct \( y \) from its \( k \)-nearest neighbours in \( L \). We extract the reconstruction weights in a manner similar to [39], to obtain
where
\[ y_{\text{rfn}}^{\text{NLDR}} = w \cdot L', \] (6.2)

with \( w \) being the derived weights and \( L' \) being the subset of \( L \) that are in the \( k \)-nearest neighbour set of \( y \). The subscript \( \text{rfn} \) stands for “reconstructed from neighbours.” Under both local and global NLDR mapping, neighbourhood properties in \( R^M \) are preserved in \( R^m \).

Preservation of neighbourhood properties implies preservation of the \( k \)-neighbour set and reconstruction weights for a point in both spaces. Following this implication, approximating \( y^{\text{NLDR}} \rightarrow y^{\text{PCA}} \) relies on applying the reconstruction weight vector derived in Equation 6.2 to the points in the data set \( D \) corresponding to the \( k \) neighbours of \( y \) in \( L \). Equation 6.5 expresses this. Recall that every point in \( L \) has an explicit correspondence in \( D \), and so indexing these neighbours is trivial.

\[ y^{\text{PCA}} = w \cdot D', \] (6.5)

The weights \( w_i \) are the same as in Equation 6.2, and \( D' \) is the nearest neighbour indices of \( y \) in PCA space, corresponding to \( L' \).

We evaluated the validity of our inverse approach, and in particular the implications of neighbourhood preservation that act as the foundation of our inverse technique by applying
Equations 6.2 and 6.5 to the points in \( L \). Every point in \( L \) was reconstructed from its neighbours in NLDR space, and projected onto PCA space by applying its weight vector to its corresponding neighbour set in PCA space. The PCA approximation of every point in \( L \) returned precisely its corresponding point in \( D \), showing that the local preserving properties of NLDR were properly adapted to our derivation of a successful inverse mapping. The metric we used to test the accuracy of the inverse \( L \rightarrow D \) is expressed in in Equation 6.7.

\[
e = \|d_{approx}^{PCA} - d^{PCA}\|_{L2}.
\]  

(6.7)

where \( d_{approx}^{PCA} \) is the approximated PCA inverse of a point in \( L \), and \( d^{PCA} \) is the point’s explicit corresponding point in the PCA set \( D \).

The only free parameter in Equations 6.2 and 6.5 is the \( k \) parameter, and its value does indeed effect the accuracy of the approximation. On the numerous manifolds that we used as a test bed for our inverse mapping, we found that optimal approximation results with respect to Equation 6.7 were achieved by setting the neighbourhood size \( k \) in the range of three to five, depending on the sampling density and other properties relating to shape and curvature of the manifold. The errors for particularly these values of \( k \) were absolutely negligible. To place the scale of the error in perspective, the distance \( \|d_{approx}^{PCA} - d^{PCA}\|_{L2} \) is normally a minuscule fraction of the distance between \( d^{PCA} \) and its \( k \) nearest neighbours. Considering that the manifold is sampled at a higher frequency than the input which presumably was already visually satisfactory, the approximation errors can be considered negligible in PCA space, and even more so when projected back to input space.
6.4.2 From points to sequence of frames

In most practical cases, however, animation through expression manifolds means that the user has specified a series of frames in NLDR space, and the inverse mapping must now be applied to a sequence of points from NLDR to PCA. This sequence is expected to be of a temporal and continuous nature, rather than at discrete, nonuniformly spaced points. The user can specify this sequence efficiently by a set of landmark points $l_1, l_2, \cdots, l_g$ that will act as handles for constructing an interpolating NURBS structure for the desired animation sequence. Let us call this an animation spline. The affine invariance properties of the inverse transformation becomes important, as we would like the path of two animation splines, in PCA and NLDR space, to be constructed through the same set of corresponding handles, to project onto each other. Because the inverse projection of each point is concerned only with the weights of its neighbourhood, and because these weights are invariant to transformations, evaluated points on the animation spline between $y_{i}^{NLDR}$ : $y_{j}^{NLDR}$ will successfully map to the path of the spline between $y_{i}^{PCA}$ : $y_{j}^{PCA}$.

6.5 Summary

In this chapter, we examined how NLDR techniques can be applied to expression manifolds as a means to render such potentially complex structures intuitive and efficient to navigate on a low dimensional space. We discussed the requirements on the distribution of points mandated by NLDR techniques, and proposed a two-tier resampling approach that allows the sampled manifold to successfully meet these requirements. We demonstrated multiple results of NLDR maps, and used these as a framework to analyze and compare the performance of local and global NLDR. Finally, we closed the loop by proposing an inverse projection from NLDR space to PCA space. Having the ability to construct an efficient workspace composed
of NLDR expression maps, the next question is one of \textit{workflow}. How expression manifold maps integrate into the animator’s workflow is the subject of the next chapter.
Chapter 7

From representation to animation

"In theory, there is no difference between theory and practice. But, in practice, there is."

-Jan L. A. van de Snepscheut

Although the techniques described in this thesis take a different approach to representing and navigating the face space than what is integrated into most conventional workflows, the conventional user interfaces and interactions which have been upheld for their usability and efficiency will transfer seamlessly into the PCA-NLDR workflow. Hence, the user need not familiarize himself with a new set of tools, but only adapt to the concept of constructing a facial animation library within a manifold framework. In this chapter we explore the practicalities of integrating expression manifold maps within the animation workflow. Furthermore, we attempt to bridge together the independently discussed components throughout this thesis into a coherent animation system.
7.1 Working in the PCA Space

Many of the interface components of animating faces in PCA space are analogous to blendshape facial animation in Maya. However, beneath the common user interfaces, PCA’s continuous Euclidean space lends itself to a multitude of useful operations that are not natural to a user defined blendshape space. In contrast to blendshape space, animating in continuous space provides the user with a notion of distance, and as a result an intuitive interpretation of motion as a natural navigation of distance between two points.

To create an animation sequence, the user makes use of the common notion of keyframes. If the expressions in the desired sequence are not of a high AU dimensionality, e.g. (≤ 4), the user can navigate and plot the sequence of static expression frames in a continuous space formed by its respective principal components. Because we have characterized eye motion and mouth motion in distinct spaces, a wide array of expressions can be derived merely through permutations afforded by two continuous spaces. For more detailed animation composed of a higher local AU dimensionality than that which can be visualized, the animator can scroll through different values of the axes of interest. Whether the keyframes are designated through continuous space navigation, a scroll interface, or any other method, as with conventional workflow they can be interpolated in several ways, including linear and higher-order spline interpolation. The animation curves can also be manipulated accordingly.

In addition to the specification and manipulation of individual keyframes, one can apply a range of operations in PCA space, such as scaling or translation of a temporal expression curve in its entirety or through piecewise segments. The effect of a scaling transformation, for example, would be an exaggeration of deformation in the direction of scaling.

The design of an expression class is slightly more involved, although the process makes use of typical curve and surface tools. Central to this process is defining the relationship between
pairwise AUs, which are often nonlinear with respect to time, as explained in Chapters 5 and 6. These relationships can be extracted from motion capture data by projecting the mocap stream onto PCA space and manipulating the resulting curve as needed. Alternatively, an animator can define AU relationships for characteristic trajectories of the expression class within the application, with the aid of an interactive plotter and renderer. We use the term characteristic trajectories to refer to the PCA trajectories of those expression sequences that form the backbone of the manifold. Examples of characteristic trajectories are depicted in Figures 5.7 and 5.10. Many expression classes inhabit a simple manifold. We use the term simple to refer to a manifold in which AU relationships can be explained through simple mathematical expressions or operations. As an example, a single trajectory can be extruded along a style parameter or AU. Style parameters were discussed in Section 5.3.

For more complicated expression classes, the user can build the manifold on a trajectory by trajectory basis. By plotting out several characteristic trajectories, intermediate ones can be interpolated. This was the approach we used to create the Freeform manifold in Section 5.3.1. By making use of curve modelling tools in an AU oriented space, the foundation for an entire class of FA sequences can be designed in a matter of minutes, and fine tuned conveniently in an intuitive workflow.

7.2 Working in the NLDR Space

Perhaps the most significant contribution of NLDR analysis is the ability to navigate complex facial behaviour in high AU dimension space through various “planar” interfaces. With the continuous rise of interactive tangible media and interface gadgets, traditional workflow practices such as navigating through countless scroll bars for high dimensional data is infeasible, out of context, or obsolete in many applications. Once projected onto NLDR
space, avatar faces in games and tangible computing modules can be navigated gracefully through finger puppetry, projecting motion curves onto a life size wall, or walking across a sensored mat, to list a few examples. The professional animator can navigate libraries of FA represented through expression manifold maps with a pen and tablet interface, and several animators can discuss sketches of facial expression flow within storyboards by the simple tracing of a curve.

Recall from Section 6.4 that our NLDR embedding is defined only within the bounds of the manifold of interest, as we described a continuous space within the convex hull of the embedded points $L$. Furthermore, the inverse scheme requires that a point lie completely within the convex hull of some local neighbourhood. Then, a legitimate operation defined on this space must provide closure within the manifold bounds. This renders operations such as scaling and translating points challenging, due to the ambiguity raised as to where the newly created point projects back to in PCA space.

Another characteristic of NLDR space is that the embedding does not respect the preferred parameterization or temporal dynamics of the high dimensional manifold. In other words, the necessary local and global relationships that must be extracted do not change as a function of how points on the manifold are sampled. For example, consider the structure in Figure 6.6 which has been resampled and processed for NLDR analysis. This structure does not retain any information pertaining to the direction of the expression trajectories, and the ordering of the sampled data points are insignificant. As a result, the NLDR axes most likely will not coincide with the trajectories of motion. We regard the representation of motion trajectory information in NLDR space as a design issue. For example, points and subspaces in NLDR space can colour be coded as a function of the trajectory parameterization variable $u$. The direction of motion can then be communicated to the animator by a simple glance.

The issues discussed in the previous paragraphs suggests that NLDR space does not pro-
vide a cohesive environment for creation or manipulation of facial expressions, but is ideal for navigation operations. More specifically, in the planar NLDR space, an animator can select keyframes, explore and re-structure the path of expression trajectory curves, design new motion curves, interpolate between points and curves, all within a reduced dimensionality. The results of the animator’s work are then projected back to PCA space, and subsequently to input space.

7.3 Integration of Various Expression Manifolds

Animating a face mesh will rely on a library of expression classes, and it is important to address how these expression classes will interoperate. Within the user’s toolbox, an expression class can simply be regarded as an encapsulation of semantically similar expressions, with the necessary tools to organize, visualize, and manipulate this dedicated and specialized manifold space. The animator need not be concerned with the details of the underlying PCA and NLDR algorithms. Because all expression classes defined over a local region (e.g. eye or mouth) of the face mesh are a subspace of a common euclidean PCA space, there is no need for transforming between expression spaces. A point defined on the surface of one expression class space can be succeeded and interpolated with a point from a different expression class.

7.4 Workflow of the System

In this section we attempt to bring together the components discussed thus far, and present a panoptic view integrating the underlying representation, implementation, and the workflow. In our experiments, we used Maya [29] for the animation of faces, and Matlab for PCA and NLDR computations. However, the steps we will review shortly are application independent,
and serve as a blueprint for our proposed animation system, rather than a manual of our implementation details.

**Step 1:** Data acquisition and reformatting.

**Description:** Acquire two streams of input training data, in the form of mocap markers or mesh objects/vertices.

- $D^{Eye}$: Only the eye region is animated.
- $D^{Mouth}$: Only the mouth region is animated.

Each input training stream is structured as follows:

$D = \{s_1, s_2, \ldots, s_j\}$ where each sequence $s_i = \langle f_1, f_2, \ldots, f_k \rangle$ is a coherent progression of $k$ frames, from the neutral face to the apex of a particular expression. In addition, provide the landmark points of the neutral face, the face void of any muscular deformations.

**Input:** $D^{Eye}$, $D^{Mouth}$, neutral face.

**Function:** Organize the raw data stream into the matrix $M_{dxn}$. Each of the $n$ columns represents the face at one frame, by concatenating the set of $(x, y, z)^T$ values of all vertices in the frame. Then $d$ is the number of vertices $\times$ 3. The same concatenation is performed on the neutral face object, to acquire $\psi$.

**Output:** $M^{Eye}$, $M^{Mouth}$, $\psi$.

**Step 2:** AU Basis creation.

**Description:** Construct the local PCA spaces. Provide the neutral face $\psi$ explicitly to the PCA algorithm to use in place of the average of the data set (see Section 5.2.2 for ex-
Input: $M^{\text{Eye}}, M^{\text{Mouth}}, \psi$.

Function: $\text{PCA}^{\text{Eye}} = \text{PCA}(M^{\text{Eye}}, \psi)$, $\text{PCA}^{\text{Mouth}} = \text{PCA}(M^{\text{Mouth}}, \psi)$.

Output: Local space structures $\text{Space}^{\text{Eye}}, \text{Space}^{\text{Mouth}}$, containing the following members:

- **eigenValue**: Eigenvalues associated with each eigenvector; reveals the scale of the deformation.
- **pcaVectors**: Eigenvectors with non negligible eigenvalues; forms the space basis.
- **pcaCoords**: PCA coordinates of the training data trajectories; used to determine the active subspace of each eigenvector, representing legitimate points in face space as determined by the variance of the training data set.

Step 3: Preparing the animator’s toolbox.

Description: Upon construction of the required spaces, the animator can start building her toolbox. As explained in Section 5.2.4, PCA’s eigenvector ranking scheme does not fare well with the nature of facial deformation data. In this phase, the animator retains the “perceptually significant” principal components. It is important to mention that this action does not actually alter the underlying PCA representation of the space. Rather, it hides any redundant or otherwise inefficient principal components from the user’s view, while congregateing the useful principal components to the animator’s AU workspace.

Output: A library of usable $\text{Eye}$ and $\text{Mouth}$ region AUs.

Step 4: Constructing X spaces.
To gain a global animation handle, the animator combines the local spaces leading to either a general global space encompassing all Eye and Mouth AUs, or a set of X-spaces (expression spaces) constructed from selected local principal components. The advantage of the latter approach is that the representation of each expression can be reduced to the minimum essential dimensionality required. Assuming the latter approach, the animator develops a library of dedicated X-spaces whose union spans the range of desirable expressions for the character. How this library is partitioned is a matter of design and personal choice. As an example, the animator may partition the global space into an X-space with negative components (e.g. nose wrinkle, sadBrow, frown) and an X-spaces with positive components (e.g. smile, browRaise). Alternatively, an X-space may contain AUs to model very specific expressions (e.g. surprise, shyness). It is natural, even expected, that the set of X-spaces not be mutually exclusive.

**Input:** \( E = PC_1, \cdots, PC_n \subseteq PCA^{Eye} \), \( M = PC_1, \cdots, PC_g \subseteq PCA^{Mouth} \).

**Function:** Based on the polarity of each principal component in \( E \) and \( M \), map its negative and positive active subspaces to \([-1,0]\) and \([0,1]\), respectively, to derive the set of unit principal components \( E^u, M^u \). The active subspace of a principal component corresponds to the range of its values that map to desirable deformations in the face space of the character. These unit PCs form the basis of the X-space.

**Output:** Library of customized expression spaces \( X = X_1, X_2, \cdots, X_n, X_i = E^u_i \cup M^u_i \).

**Step 5:** Designing expression manifolds.

The animator encapsulates trajectories of semantically similar expressions in an expression manifold structure. The trajectories are then resampled according to Sec-
tion 6.2.2, to approximate the manifold to meet the uniformity and density requirements explained in Section 6.2. Designing an expression manifold is an art in addition to a science. Like any other art form, it demands a design layout or a storyboard. The animator must familiarize herself with the behaviour of the character, and give thought as to the best way to organize the character’s behaviour into expression classes to optimize usability and efficiency.

**Design methods:** Expression class trajectories are organized in the following structure:

\[
\{\text{ExpClassName}\}
\]

\[
.trj_1 = key_1, key_2, \ldots, key_i
\]

\[
.trj_2 = key_1, key_2, \ldots, key_j
\]

\[\vdots\]

\[
.trj_n = key_1, key_2, \ldots, key_k
\]

Derivation of the trajectories may be through any method the animator feels comfortable with. The following are a few suggestions:
**X-space navigation:** Key frames of trajectories can be visually specified by navigating through X-spaces. By adjusting the value along principal components in continuous space, the animator can “key” the current location. However, the use of a space-visualization technique becomes cumbersome as the number of DOFs increase.

**Scroll entry:** Keys of trajectories requiring higher DOFs can be specified through the ubiquitous scroll interface.

**Interactive plotting:** Relationships between pairwise AUs can be specified by interactive plotting. Figures 5.5, 5.6, 5.8, and 5.9 illustrate plotting pairwise relationships among AUs.

**From data:** The manifold trajectories can be acquired through an external source such as mocap. The eye and mouth region deformations are separated and projected onto their respective local PCA spaces to obtain their AU representation.

**Manifold Resampling:** The manifold is then resampled according to Section 6.2.2. We will not repeat the details here.

**Output:** The matrix of points, $M_{dxn}^{exClass}$, resulting from the two-tier resampling in the $u$ and $v$ directions of the underlying manifold, providing a dense and uniform approximation to the expression manifold.

**Step 6:** Obtaining NLDR map.

**Description** Local and global NLDR analysis is applied to each expression manifold, sampled by the matrix $M_{dxn}^{exClass}$. This operation effectively “unfolds” the expression manifold to
obtain a means of planar navigation.

**Input:** The set of expression manifolds $M_j$, $j = 1, \cdots, n$.

**Function:** $Isomap(M_j, k)$, $LLE(M_j, k)$, where $k$ is the $k$-nearest neighbour parameter.

**Output:** The set of Isomap and LLE NLDR spaces, $I_j$ and $L_j$, respectively, across all expression manifolds $j$.

**Step 7: Inverting Animation.**

**Description:** Animation creation, manipulation, and representation now has a new space. Unfortunately, however, the rendering process requires data to be expressed in input (e.g. vertex) space. This incites the need for an inverse mapping; Input $\leftarrow X$-space $\leftarrow$ NLDR.

**Input:** A point, or sequence of points, $P$, expressed in Isomap or LLE space, $I_j$ and $L_j$, respectively.

**Function:** For all points in $P$, the mapping from NLDR to local PCA space can be expressed as

$$P^{X\text{space}} = \text{invNldr}(P),$$

(7.1)

where $\text{invNldr}()$ returns the inverse projection according to Section 6.4. The mapping from local PCA space to input space can be expressed as

$$P^{\text{Input}} = \text{invX}(P^{X\text{space}:\text{Eye}}) + \text{invX}(P^{X\text{space}:\text{Mouth}}),$$

(7.2)

where $\text{invX}$ inverts the $\text{Eye}$ and $\text{Mouth}$ components to input space, through the inverse of their respective parent PCA basis, as expressed in Equation 5.3.

The deformations embedded within the two regions are then applied to a common neutral
face. This yields the vertices of the resulting mesh, which combined with the face and material descriptions, normals, and other standard specification, form a complete mesh structure.

Output: The corresponding faces or sequence of faces in input space $P^{\text{Input}}$.

7.5 Summary

In this chapter we reviewed the possible operations in PCA and NLDR space from the user’s vantage point. Of equal significance of knowing what is possible within any computational framework, is knowing the limitations imposed by the framework. We presented some of the limitations of NLDR spaces in conceptualizing new facial expressions, but discussed how NLDR can act as a solid means of navigating and manipulating existing temporal expression curves. Finally, we presented a cohesive blueprint for integrating expression manifolds and their NLDR maps within a facial animation workflow.
Chapter 8

Conclusion

Our motivation for the work presented in this thesis stemmed from the challenge of modelling the dynamics of continuous facial motion. Modelling facial motion is by no means a novel endeavor. However, we took upon this challenge with the ambition of deriving a representation that naturally emphasizes the biomechanics of continuous facial expressions. In other words, when one “reads” the representation of an expression, they gain high level insight into the biomechanics of the facial motion. We employed local Principal Components Analysis (PCA) to automatically construct a linear basis whose coordinates correspond to the near-orthogonal degrees of freedom, or action units, responsible for deforming independent regions of the expressive face. We demonstrated how the projection of continuous facial expressions onto PCA space incur a curved trajectory, and used this phenomenon to support the postulation that the relationship between various action units are nonlinear in time. As a consequence of this phenomenon, we introduced and formalized the notion of expression manifolds, nonlinear manifolds residing in linear PCA space that embed the motion dynamics of semantically similar facial expressions. To integrate these potentially complex and high dimensional manifolds within the facial animation toolset, we employed Nonlinear Di-
mensionality Reduction (NLDR) techniques to map the manifold to a low dimensional space that can be more efficiently managed. We then presented the framework for a new workflow which allows the animator to model and manipulate animated expressions in the PCA space, and organize and navigate repositories of semantically similar expressions in NLDR space.

8.1 Future Directions

We conclude our discussion by outlining several key directions in which this work can be extended upon.

Representation

Given an accurate parameterization of facial motion, it is a straightforward process to create an expression in mind. However, this claim can not be made regarding the inverse problem. Given a data bank of facial motion data, deriving an accurate parameterization of facial motion remains a problem to be optimized. We employed local PCA for this task. However, it is well worth exploring representations that are more suitable to the nature of facial motion, for example, representations that extract perceptually independent degrees of freedom, but respect the fact that facial action units are not strictly orthogonal in Euclidean space. In addition, as standards in animating character realism are rapidly increasing, more powerful techniques for extracting and parameterizing nuances in facial behaviour become of significant importance.

Software development

The research and ideas discussed in this thesis were realized through various prototype implementations. These various prototypes must be consolidated into a cohesive production
environment with a user interface. As with any system, scalability is an integral factor. To fully realize the potential of a nonlinear framework for facial animation, we must scale to large repositories of high fidelity motion capture data and more complex manifold geometry in higher dimensions.

**Extension to other applications**

There are many other facial animation procedure that can benefit from the framework we have presented here. Our discussion of expression manifolds and the PCA-NLDR space was centered around the modelling, manipulation, and navigation of temporal expressions. However, since our method of representation is geometry-independent, we believe there is strong potential in extending this approach to applications of facial motion transfer between characters, and the general problem of motion search and reuse.
APPENDIX

Global Space, Eye Space, Mouth Space
Figure 1: Global Face Space: Min and Max of Principal Component 1

Figure 2: Global Face Space: Min and Max of Principal Component 2
Figure 3: Global Face Space: Min and Max of Principal Component 3

Figure 4: Global Face Space: Min and Max of Principal Component 4
Figure 5: Global Face Space: Min and Max of Principal Component 5

Figure 6: Global Face Space: Min and Max of Principal Component 6
Figure 7: Global Face Space: Min and Max of Principal Component 7

Figure 8: Global Face Space: Min and Max of Principal Component 8
Figure 9: Global Face Space: Min and Max of Principal Component 9

Figure 10: Global Face Space: Min and Max of Principal Component 10
Figure 11: Global Face Space: Min and Max of Principal Component 11

Figure 12: Global Face Space: Min and Max of Principal Component 12
Figure 13: Global Face Space: Min and Max of Principal Component 13
Figure 14: Eye Space: Min and Max of Principal Component 1

Figure 15: Eye Space: Min and Max of Principal Component 2

115
Figure 16: Eye Space: Min and Max of Principal Component 3

Figure 17: Eye Space: Min and Max of Principal Component 4
Figure 18: Eye Space: Min and Max of Principal Component 5

Figure 19: Eye Space: Min and Max of Principal Component 6
Figure 20: Eye Space: Min and Max of Principal Component 7

Figure 21: Eye Space: Min and Max of Principal Component 8
Figure 22: Mouth Space: Min and Max of Principal Component 1

Figure 23: Mouth Space: Min and Max of Principal Component 2
Figure 24: Mouth Space: Min and Max of Principal Component 3

Figure 25: Mouth Space: Min and Max of Principal Component 4
Figure 26: Mouth Space: Min and Max of Principal Component 5

Figure 27: Mouth Space: Min and Max of Principal Component 6
Figure 28: Mouth Space: Min and Max of Principal Component 7

Figure 29: Mouth Space: Min and Max of Principal Component 8
Figure 30: Mouth Space: Min and Max of Principal Component 9
Bibliography


[27] C. Kouadio, P. Poulin, and P. Lachapelle. Real-time facial animation based upon a 


[34] Parke. Parameterized models for facial animation. _Computer Graphics and Applications, 

the ACM annual conference_, pages 451–457, New York, NY, USA, 1972. ACM.


