Evolving Credible Facial Expressions with Interactive GAs

by

Nancy Smith

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Computer Science

Graduate School of Computer and Information Sciences
Nova Southeastern University
2012

We hereby certify that this dissertation, submitted by Nancy T. Smith, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

_____ _____

Michael J. Laszlo, Ph.D. Date
Chairperson of Dissertation Committee

_____ _____

Wei Li, Ph.D. Date
Dissertation Committee Member

_____ _____

Sumitra Mukherjee, Ph.D. Date
Dissertation Committee Member

Approved:

_____ _____

Eric Ackerman, Ph.D. Date
Interim Dean

Graduate School of Computer and Information Sciences
Nova Southeastern University

2012

An Abstract of a Dissertation Proposal Submitted to Nova Southeastern University
in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Evolving Credible Facial Expressions with Interactive GAs

by
Nancy Smith
January 2012

A major focus of research in computer graphics is the modeling and animation of realistic human faces. Modeling and animation of facial expressions is a very difficult task, requiring extensive manual manipulation by computer artists. Our primary hypothesis was that the use of machine learning techniques could reduce the manual labor by providing some automation to the process.

The goal of this dissertation was to determine the effectiveness of using an interactive genetic algorithm (IGA) to generate realistic variations in facial expressions. An IGA's effectiveness is measured by satisfaction with the end results, including acceptable levels of user fatigue. User fatigue was measured by the rate of successful convergence, defined as achieving a sufficient fitness level as determined by the user. Upon convergence, the solution with the highest fitness value was saved for later evaluation by participants with questionnaires. The participants also rated animations that were manually created by the user for comparison.

The animation of our IGA is performed by interpolating between successive face models, also known as blendshapes. The position of each blendshape's vertices is determined by a set of blendshape controls. Chromosomes map to animation sequences, where genes correspond to blendshapes. The manually created animations were also produced by manipulating the blendshape control values of successive blendshapes.

Due to user fatigue, IGAs typically use a small population with the user evaluating each individual. This is a serious limitation since there must be a sufficient number of building blocks in the initial population to converge to a good solution. One method that has been used to address this problem in the music domain is a surrogate fitness function, which serves as a filter to present a small subpopulation to the user for subjective

evaluation. Our secondary hypothesis was that an IGA for the high-dimensional problem of facial animation would benefit from a large population made possible by using a neural network (NN) as a surrogate fitness function. The NN assigns a fitness value to every individual in the population, and the phenotypes of the highest rated individuals are presented to receive subjective fitness values from the user. This is a unique approach to the problem of automatic generation of facial animation.

Experiments were conducted for each of the six emotions, using the optimal parameters that had been discovered. The average convergence rate was 85%. The quality of the NNs showed evidence of a correlation to convergence rates as measured by the true positive and false positive rates. The animations with the highest subjective fitness from the final set of experiments were saved for participant evaluation. The participants gave the IGA animations an average credibility rating of 69% and the manual animations an average credibility rating of 65%. The participants preferred the IGA animations an average of 54% of the time to the manual animations. The results of these experiments indicated that an IGA is effective at generating realistic variations in facial expressions that are comparable to manually created ones. Moreover, experiments that varied population size indicated that a larger population results in a higher convergence rate.

Acknowledgements

I would like to thank my husband, Bruce, for his incredible support and belief in me. There were a tremendous number of weekends in which I was locked away finishing assignments and projects, and I never could have done it without his unwavering support. Also, his feedback on the participant experience of animation evaluations was invaluable. But then he always has been the most insightful person I've ever known.

I would like to thank my dance friends for joyously embracing the tedious evaluation of dozens of animations. I especially want to thank Sambra for organizing the get-together focusing on the evaluations. I had to give up most of my hobbies for the duration of my doctorate studies, but dancing and my dear dance friends provided a much-needed diversion and helped me maintain my sanity along the way.

Next, I would like to thank a man I've never met, but who had a tremendous impact on my life nonetheless. Ray Kurzweil wrote a book called "The Singularity is Near" which discussed the achievements and rate of progress in the fields of nanotechnology, genetics, and computing. It is this book that set my imagination on fire and motivated me to go back to school to gain the skills I need to be part of the incredible changes that artificial intelligence is going to bring to our world.

Also, I would like to thank my committee members, Dr. Sumitra Mukherjee and Dr. Wei Li, for their insightful comments and support. I can honestly say that Dr. Li's class on AI was by far my favorite class. The mix of interesting assignments and research provided a very thorough coverage of this fascinating field, and I am very glad that he was my professor for it.

Finally, this dissertation would not have been possible without the expert guidance of my esteemed advisor, Dr. Michael Laszlo, whose own work in computer graphics and genetic algorithms started me on this path. Dr. Laszlo always read and responded to my questions and the drafts of my papers more quickly than I ever could have hoped. His comments and suggestions were always extremely perceptive and helpful, as they helped to focus and guide my research efforts. Of all the decisions that I made along the way, the choice of Dr. Laszlo as my advisor is the one that I'm most grateful for.

Table of Contents

List of Tables

Tables

List of Figures

Figures

Chapter 1

Introduction

Background

A major focus of research in computer graphics is the modeling and animation of realistic human faces. There has recently been a dramatic increase of interest in facial animation (Parke and Waters, 2008). The film and game industries push the boundaries of established animation techniques, and the quest for believable embodied conversational agents (ECAs) is the subject of much research in the field of human-computer interaction. Modeling and animation of facial expressions is a very difficult task, requiring extensive manual manipulation by computer artists. Despite the advances in computer hardware and improvements in software algorithms, there is still no computational system that approximates the human face (Griesser, Cunningham, Wallraven, and Bilthoff, 2007). Nor is there any real-time system that generates subtle facial expressions and emotions realistically (Deng and Neumann, 2007). Facial modeling and animation are still being defined, with no widely accepted solutions (Parke and Waters, 2008). Animation techniques are ad-hoc and not easily extendible (Parke and Waters, 2008). One of the primary research goals for facial animation is a system that creates realistic animation while reducing the amount of manual manipulation.

Facial modeling and animation can be roughly classified in the following categories: blendshape or shape interpolation, parameterizations, facial action coding system (FACS) approaches, deformation-based approaches, physics-based muscle modeling, 3D facial modeling, performance-driven facial animation, MPEG-4 facial animation, visual speech animation, facial animation editing, facial animation transferring, and facial gesture generation (Deng & Neumann, 2008). The boundaries of these classifications overlap since many share the same techniques or integrate multiple methods. The research described in this paper focuses primarily on the blendshape technique.

A *blendshape* is the convex combination of a number of topologically conforming shape primitives

$$V_j = \sum_k w_k b_{kj}$$

where V_j is the *jth* vertex of the resulting animated model, w_k is the blending weight, and b_{kj} is the *jth* vertex of the *kth* blendshape. The weighted sum is applied to the vertices of polygonal models. The weights w_k are manipulated by the animator by sliders, with one slider for each weight, or automatically by algorithms.

Linear interpolation between successive blendshapes is commonly used for simplicity, but a cosine interpolation function can provide acceleration and deceleration effects at the beginning and end of an animation. When at least four keyframes are involved, bilinear interpolation can generate a wider range of facial expression changes than linear interpolation (Deng & Neumann, 2008).

Modifying the parameters of the interpolation functions generates interpolated images. Geometric interpolation directly updates the positions of the vertices, while parameter interpolation controls functions that indirectly move the vertices. Although interpolations are fast, they are typically restricted to a small range of facial configurations since a blendshape model must be created for each desired facial expression. This research project addressed this limitation by creating blendshapes not for specific expressions, but for the components of expressions, such as raising the brows or squinting the eyes.

Genetic algorithms are an effective way to search extremely large or complex solution spaces. Due to the complexity and large variations of facial images, many GA approaches have been applied to human face applications, including face detection, feature extraction, posture estimation, face recognition, and facial modeling. GAs have also been used to generate creative temporal sequences, such as simple animation and music compositions. When the goal of a genetic algorithm is subjective, that is, subject to human opinion, an *interactive genetic algorithm* (IGA) may be used. An IGA replaces the typical mathematical fitness function with an interactive human-machine interface so that a user can evaluate the individuals subjectively.

The major problem of IGAs is human fatigue. Psychological fatigue is especially problematic with temporal data such as movies since the user has to compare the current movie with previous ones in the user's memory (Takagi, 2001). This is typically dealt with by using small populations. Unfortunately, small populations suffer from the lack of genetic diversity, resulting in poor performance and a tendency to converge to a non-

optimal solution. To find solutions of high quality, the population size must be increased as much as possible (Harik, Cantu-Paz, Goldberg, & Miller, 1999). One method that has been used to address this problem is to use a fitness prediction function, also called a *surrogate function*. This algorithm uses a large population, applies a predictive fitness function to all the individuals, and then shows a small subset of the most likely candidates to the user for evaluation (Takagi, 2001; Jin, 2005).

The most popular models used as surrogate functions are polynomials, kriging, and neural networks, including multi-layer perceptrons, radial basis function networks, and support vector machines (Jin, 2005). It is desirable to use the simplest method possible. If the given samples can fit a lower order polynomial model, it is the best choice. However, in the case of a high-dimensional input space and a limited number of samples, a neural network is preferred. There is a significant body of research supporting the use of neural networks for classifying facial expressions.

Problem Statement

Computerized 3D facial animation is a difficult task that currently requires extensive manual intervention. Variations in a given facial expression are rarely pursued outside of high-budget feature films due to the time and effort involved in manually generating realistic variations. The research described in this paper used an interactive genetic algorithm (IGA) for generating realistic variations in 3D facial expressions.

In one of the more popular animation techniques, a number of face models are created from the basic shape. Then distinct model variations are selected and assigned (key framed) to points throughout the scene, interpolating from one model to the next.

This method of transitioning among predefined facial models, or blendshapes, has been used in commercial movie productions. Due to its efficiency and simplicity, the blendshape approach is widely used for key framing facial animation (Li and Deng, 2008). The mechanisms for creating and controlling blendshapes are built into many of the current commercial graphic programs. The increase in power and affordability of graphic systems such as Maya and 3D Studio have made modeling, animation, and rendering more easily available for research purposes (Parke and Waters, 2008).

One of the reasons that facial animation is difficult is that there is both a random component and a predictable component to expressions. Certain facial movements are expected due to the important role they play in communication (Griesser, Cunningham, Wallraven, and Bilthoff, 2007). Facial expressions are used to modify the meaning of what is being said, to control the flow of conversation, and to provide feedback to the speaker on how to proceed (Cunningham, Kleiner, Bilthoff, & Wallraven, 2004). For example, if the listener nods, the speaker knows to continue. If the listener appears confused, the speaker may try to explain in more basic terms. Such observations are the basis of rule-based systems that have been developed to drive embodied conversational agents (ECAs). Once the rule-based system has determined a new target expression, the face model is transitioned to the corresponding blendshape. Rule-based systems are useful for interactive roles. They focus on the roles of speaker and listener, but tend to suffer from static emotional expressions (Mana and Pianesi, 2006). A sad expression always generates the same sad expression.

Another important technique for driving the animation of facial expressions is to map motion-capture sequences to blendshape face models (Li and Deng, 2008). This technique is better suited for non-interactive applications such as making movies. Although it is a simple and fast animation technique, the modeling phase requires many tedious hours of manual labor to create enough target blendshapes. For instance, in the feature film The Two Towers, the facial animations of Gollum required 675 blendshapes.

Dissertation Goal

The goal of the research was to determine the effectiveness of using an interactive genetic algorithm to generate realistic variations in facial expressions. Given a target emotion, the IGA evolves variations of the face expressing that emotion. The emotions considered in this project are limited to the six basic emotions of happiness, sadness, disgust, anger, fear, and surprise, as identified by Ekman and Friesen (1978).

The input target emotion can be considered a predictable component of the typical facial animation system since variations in emotional expressions are usually created manually. The IGA evolved face models expressing the target emotion. The face models were assigned to the key frames in an animation sequence, which added a random component to the animation sequence. The graphics software did the interpolation between the key frames automatically. The human user evaluated the animation sequences.

To address human fatigue, a neural network was used as a surrogate fitness function. The initial neural network was trained with samples of the basic emotions. These samples represented the most predictable elements of the basic emotion.

IGAs have been used to generate animated art, animated arms and legs, 2-D facial expressions (Takagi, 2001), and 3-D facial models (Ho, 2001). However, it is believed that the use of an IGA to generate 3-D facial animation is a unique contribution to the research literature.

Neural networks have been widely used as a surrogate function for IGAs evolving music. These IGAs are similar to the research described in this paper in that they generate a creative temporal sequence that follows predictable rules. NNs have often been used in the field of facial expression classification using a variety of feature extraction techniques to represent the face (Fasel & Luettin, 2003). It is believed that the use of NNs as a surrogate fitness function for evaluating 3-D facial models is a unique contribution.

Research Questions

Chromosome representation

One of the key issues for any genetic algorithm is the genomic representation of the problem. Our genome represents a sequence of face models, which are comprised of FACS-based components.

Integration with rule-based requirements

Another question is how to integrate the random variations produced by the IGA with the rule-based requirements of facial animation. Our research does not attempt to determine why the face should express an emotion. It only produces variations in the emotion that would be predetermined by a rule-based or motion-capture system.

Convergence

One of the key design issues is how to ensure enough diversity in the population to converge to a satisfactory solution. This dissertation investigates the effectiveness of using a large population to ensure an adequate number of building blocks.

Fitness prediction function

Another research question is how best to incorporate a fitness prediction function that can quickly be applied to the total population so that only the most promising candidates have to be evaluated by the user. There are challenges to integrating the two types of fitness values in the selection criteria.

The details of the criteria used to select the mating population are not well established. One model evolves two separate populations, migrating individuals between them depending on fitness. The selection algorithm examines the subjective fitness value for one population, and the surrogate fitness value for the other population. In this model, the complication moves from selection to migration.

An alternative model is to evolve a single population, replacing the surrogate fitness value with the subjective fitness value. This model loses the subjective fitness value unless a mechanism is implemented so that it is not re-evaluated by the NN on the next run. One research paper associated a confidence rating to the fitness value. Another issue is being able to normalize the subjective value to have the appropriate value relative to the surrogate values.

A third option is to base the mating selection on the surrogate fitness value, and to use the user-supplied fitness values only to bias the surrogate function. This method of

evolution control is the one most often described in the literature. Surrogate fitness functions are used not only with subjective fitness functions, but also with computationally costly fitness functions. In one case where the original fitness function was computationally expensive, this option required 40% of the population to be evaluated with the original fitness function to ensure optimal convergence (Jin, Olhofer, and Sendhoff, 2001). However, when the original fitness function is subjective, the global optimum is a region of search space rather than a point since humans find it difficult to rate individuals that have small differences between them.

Relevance and Significance

There has recently been a dramatic increase of interest in facial animation (Parke and Waters, 2008). Most of the current animation techniques are based on research conducted over 20 years ago (Parke and Waters, 2008). The strongest interest in facial animation arises from the big animation studios producing feature films. The game industry also has a big influence on the demand for improved facial animation. Facial animation is also an important research goal in human-computer interaction, as in the quest to build a believable Embodied Conversational Agent (ECA). These agents would be able to communicate complex information with human-like expressiveness. ECAs are becoming popular as front ends to web sites, and as part of many computer applications such as virtual training environments, tutoring systems, storytelling systems, portable personal guides, and entertainment systems (Mana and Pianesi, 2006). The use of virtual humans in electronic commerce sites could enhance the user experience and boost sales. Conversely, users may be less likely to purchase items from a virtual agent who appears

dishonest or insincere. Sympathetic virtual assistants could help lessen a user's anger and frustration. These types of applications will require realistic and believable graphical renderings of facial expressions.

IGAs have been used to create animated graphic art by evolving mathematical equations that apply to the pixel attributes (Takagi, 2001). They have also been used to create animations by evolving the combination of joint angles for arms and legs, and for evolving deformations of a 2D body for comical movements (Takagi, 2001). IGAs have been used with 2D photos of partial images to compose a facial image for identifying a criminal suspect. IGAs have often been used to create line drawings of faces as research tasks. IGAs have also been used to change facial expressions in photo images by changing pixel positions (Takagi, 2001). Since IGAs have been used for animation and for 2D facial generation, it was reasonable to expect that IGAs could be extended to the animation of 3D facial expressions.

Modeling and Animation

Modeling and animation are two distinct components to facial animation research. In the last few years, research in both rule-based and statistical models have obtained important results in modeling facial expressions to be used in synthetic talking heads. But the generation of models for realistic animation remains critical and is still an open problem. Most of the rule-based systems suffer the disadvantage of a static generation, with no variation in the given facial expression. Stochastic models can be more flexible but most of the work in this area focuses on speech and lip movements (Mana and Pianesi, 2006). Our research explores the use of an IGA to introduce the

flexibility that allows for credible variations in facial expressions. The idea is to define the genome as the basic building blocks that comprise facial expressions. The building blocks are combined using the IGA until an acceptable variation of the target expression is evolved.

The building blocks of facial expressions have been studied extensively. There is a significant history of research into automatically extracting facial features for recognizing facial expressions, and also in computer graphics for representational models of facial expressions. One of the most widely used resources for describing facial expressions is the *Facial Action Coding System* (FACS) developed by Ekman and Friesen (1978). FACS describes 46 facial action units, segmenting the facial muscles into the smallest visible changes. Combinations of these action units can be used to describe different facial expressions. FACS provides an intuitive, semantic basis for facial animation (Griesser, Cunningham, Wallraven, and Bilthoff, 2007).

The face model must be created in such a way that it can be easily modified to reflect facial movements. This is often done with a polygonal mesh based on the pioneering work of Parke (Parke, 1972). The vertices of the mesh are manipulated to create changes in the basic face, such as raising the brows or squinting the eyes. Approximating a face with polygons has several performance advantages. Issues of determining visible surfaces and their shading have been solved with fast algorithms that are implemented in hardware. Moreover, Gouraud and Phong shading algorithms can make a polygonal surface appear continuously curved. Face models may also be created using parametric curves, but the rendering algorithms are less efficient, and deformations

such as wrinkles are difficult to simulate. Our research was conducted using a polygonal mesh. More specifically, there is a polygonal facial model for each of the facial action units to create a blendshape animation rig.

To illustrate how a blendshape animation rig works, consider an example with a sphere and a cube, with a one-to-one correspondence between the vertices of the two meshes. By subtracting a sphere point from the corresponding cube point, a delta vector is created. This is done for each pair of vertices. If, for each point of the sphere, the corresponding delta vector is added, the cube is created. The delta vector is multiplied by a weighting factor to move from the sphere (the weight is zero) to the cube (the weight is one). In terms of facial animation, for each of the FACS models, the default pose is subtracted from the FACS pose to generate delta vectors for each point in the geometry. Each set of delta vectors is assigned a weight, and by changing the weights, various FACS action units can be combined to create the desired expression. The weights of the delta vectors are modified using a blendshape control. The blendshape controls comprise the blendshape animation rig that is used to generate the phenotype. This FACS blendshape representation was used in the feature film, Benjamin Button, where the animation was driven by mapping motion capture data from an actor (Roble, 2009). Our research uses an IGA to drive the animation of a FACS-based blendshape animation rig. The FACS-based genome maps directly to the blendshape control values. This is a unique approach to the important problem of automatic generation of facial models for animation.

Interactive Genetic Algorithms

Genetic algorithms are search algorithms based on natural selection and genetics (Goldberg, 1989). They combine survival of the fittest with a structured yet randomized information exchange to form a search with human like characteristics (Goldberg, 1989). The search is guided by a fitness function which determines which of the individuals (solutions) in the population are fittest and should be chosen to reproduce. Typically, convergence of the GA is based on minimizing error criteria. Systems that create graphics or music must be subjectively evaluated for the best results since it is difficult to quantify the human opinion and understanding that is needed to rate the fitness of an individual (Takagi, 2001).

GAs are complex non-linear algorithms (Harik, et al., 1999). They work by discovering, emphasizing, and recombining good building blocks of solutions in a highly parallel manner (Mitchell, 1998). This is known as the schema theorem and is fundamental to the analysis of genetic algorithms. There must be a sufficient number of building blocks in the initial population to arrive at an optimal solution. Otherwise, the chances of the GA converging to a good solution are small (Harik et. al., 1999).

Large populations generally converge toward the best solutions, but they require more computational cost and memory requirements. Harik et al. (1999) showed that two variables that affect the needed population size are the length of the schemata, and the variability of the problem. Problems with long schemata are more difficult than those with short schemata since the long schemata occur less frequently in a random

population. Problems with high variability are hard because it is difficult to distinguish good from less-good solutions. In general, the GA population size grows with the square root of the size of the problem (Harik et al., 1999).

Since facial expressions are highly variable, it is desirable to have a large population in a GA. However, due to user fatigue, it is impractical to subjectively evaluate each chromosome of a large population. The use of a predictive fitness function enables an IGA to approximate the fitness of all individuals, so that only the most promising candidates are shown to the user for a subjective evaluation. One of the common surrogate functions used is a neural network (Jin, 2005).

Neural networks are well suited for complex pattern classifications, and have been used to classify facial expressions in a number of research projects. By using NNs as a fitness approximation function, it is possible to use larger populations required for high quality solutions. It also speeds up the convergence, thus avoiding user fatigue. Lastly, it is more likely to present satisfying solutions to the user, which lessens user frustration (Llora, Sastry, Goldberg, Gupta, & Lakshmi, 2005).

Barriers and Issues

Facial animation is still a very challenging research area. The deformation of facial movements is complex, and humans are extraordinarily sensitive to the subtleties of facial motion. Even the most sophisticated systems still require significant manual intervention since it is difficult to quantify acceptable images from those that appear impossible or "uncanny." It is one of the primary reasons why a human is required to be

in the loop for the final evaluation of a new expression sequence, even after a large database of examples has been gathered for training a classifier.

Most of the research on facial animation focuses on developing rule-based systems, where the rules are applied to the roles of speaker and listener. When an emotion is required, the same emotion elicits the same expression. The only variations are duration and intensity since these are easy to quantify in a rule-based system.

Human emotion is a difficult interdisciplinary research topic across the fields of computer graphics, artificial intelligence, communication, and psychology (Deng and Neumann, 2007). As powerful graphics programs become available, it is more feasible for researchers to focus on 3D animation. Moreover, the integration of separate fields of study is made possible through increasing internet access.

Even with these advantages, research that involves 3D animation is time-consuming. The graphics software packages have a large learning curve, and much effort is required to create the preliminary environment. There are many variations in the design of the genetic algorithm that can affect its success, which must be examined carefully in each step of the process to ensure they have the intended result.

Limitations and Delimitations

Ideally, the genotypes in this experiment would be mapped to a variety of faces to validate that the genome could be generalized across phenotypes. However, due to the time involved in generating facial models, our research project used a single face as the basis for the FACS models that comprise the phenotype expressed by the GA.

Facial animation research often focuses on the role of speech. While the GA in this project may evolve random visemes (visible component of a phoneme), there is no framework for specifying a phrase to be spoken while evolving the expression to accompany the phrase.

Since FACS does not incorporate any temporal data, transition rates between two expressions, and the duration of any particular expression, are not part of the genome. While the timing variations contribute to realistic expressions, they are not a part of this dissertation.

Definition of Terms

Allele – value or setting for a trait encoded in a gene.

Blendshape – a face model used as the beginning or ending frame for an animation sequence in which linear interpolation is used to generate the intermediate frames.

Chromosome – a collection of genes. In this project, a chromosome encodes a sequence of face models.

Facial Action Coding System – a system to categorize the physical components of facial expressions.

Face Model – a static 3D representation of the face. In this project, the face model is always a polygonal mesh.

Gene – components of a chromosome that can be thought of as encoding a trait. In this project, a gene corresponds to a face model.

Genome – the complete collection of genetic material (all chromosomes).

Genotype – the particular set of genes contained in a genome.

Interactive Genetic Algorithm – a GA that uses human opinion as the fitness function.

Phenotype – the physical expression of the traits encoded in the genotype. In this project, the phenotype is an animated sequence of 3D polygonal mesh face models.

Predictive or Surrogate Fitness Function – an heuristic for the actual fitness function. In a GA, it is used to select a subset of most fit chromosomes to be evaluated by the more accurate original fitness function.

Summary

Facial animation is still a difficult, manually intensive task that is the subject of ongoing research. One of the goals of facial animation is a system that creates realistic variations in human expressions and is automated as much as possible.

The blendshape approach of interpolating between face models is very fast, but requires face models to be created for each target expression. By creating a set of blendshapes based on FACS, it is possible to generate any possible facial configuration. This is a very large problem space, which may be suitable for a genetic algorithm approach.

Since the desired result of the GA is subjective, it is appropriate to use human opinion as the fitness function used to guide the evolution. To ensure a sufficient number of building blocks in the initial population, a predictive fitness function can be used with a large population to find a subset of the candidates who are the most fit. This subset of candidates can then be shown to the human for evaluation. A good choice for the predictive fitness function is neural networks, as they are very fast and very good at classifying human expressions.

The goal of the proposed research is to determine the effectiveness of using an IGA to generate realistic variations in facial expressions. One challenge is to represent this complex high-dimensional problem such that it is manageable, but still capable of

generating meaningful results. Another significant challenge is to incorporate the fitness prediction function in a way that optimizes the convergence of the algorithm.

Chapter 2

Review of the Literature

Background

There is a rich stream of research focused on enhancing computerized facial animation. There is also a large body of literature investigating the use of interactive genetic algorithms in generating creative works. Our research combines these two lines of research. This section presents an overview of the important research that forms the foundation of modeling and animation of human faces, as well as some examples of rule-based systems for facial animation. Then a brief description of several research projects using IGAs to generate creative works is provided, including faces, music, and fashion. The music and fashion IGAs were included as good examples of IGAs using NNs as surrogate fitness functions.

Modeling

The first computer 3D animation of faces was accomplished by the pioneering research by Parke at the University of Utah in 1971 using a small number of polygons to model the face. At the same time and at the same university, Gouraud was also developing the now widely-accepted Gouraud polygon shading algorithm, which Parke applied to achieve a level of realism to his face models (Parke and Waters, 2008). Parke developed the first parameterized face model in 1974, with the goal of producing facial

animations quickly. It consisted of 900 polygonal surfaces and was animated by changing the location of points in the grid under the control of 50 parameters, 10 of which were used for speech. Parke selected the control parameters of sentences by studying his own articulation and estimating the parameter values. Parke's method of polygonal modeling of 3D faces laid the foundation for facial animation that is still primarily in use today. In fact, the use of polygonal interpolation of blendshapes is common enough that it is built into the most common graphic programs, such as Autodesk Maya.

The second most promising alternative to polygonal representation is the muscle-based model. In 1987, Waters developed a muscle-based facial model, using an approximation of the skull and jaw pivot covered with muscles. This computation model required complex elastic models for compressible tissues. The surface layer changed according to the underlying structure. The model defined a dynamically changing set of contraction-relaxation muscle commands. Although this system achieved a high level of realism, the calculations needed for tissue simulations take much longer than calculating changing polygonal surface shapes (Parke & Waters, 2008).

Animation

The concept of interpolating between face geometries using a polygonal face model was first introduced by Parke (Parke, 1972). The terminology of blendshapes is commonly applied to the different face models, and the interpolation is often referred to as blending. Animation is typically driven by some combination of manual selection of each change in expression, speech-driven animation, and motion-capture mapping.

Pearce, Wyvill, Wyvill, & Hill (1986) introduced speech-driven animation for a facial mesh. A typed set of phonemes was input and mapped to control parameters to produce the animation sequence. Phonemes were defined with values for segment duration, segment type, jaw rotation, mouth width, mouth forward-back, lip width at corners, mouth corner coordinates, lower lip tuck, upper lip raise, and teeth offsets. They used non-linear interpolation between the phonemes, relying on the type of segments to regulate transition speed.

Cohen and Massaro (1993) further improved the speech-driven technique by incorporating dominance and blending functions into the control parameters. These functions captured differences in offset, duration, and magnitude based on coarticulation, or the changes that are dependent on the preceding and following phonemes. For example, the lips round at the beginning of the word "stew" in anticipation of the following vowel segment. The dominance function for the word "stew" would have low dominance for the "s" and "t" as compared to "u" for the lip protrusion control parameter. The "u" has a low rate value causing its dominance to continue forward in time from the vowel. The tongue angle control parameter has equal dominance for "t" and "u."

Cassell et al. (1994) developed a model for synchronizing speech, intonation, facial expressions, and hand gestures in a rule-based system. Their facial expressions focused on conversational signals, such as emphasizing a point being said, or regulating the flow of speech. Variables related to facial expressions included speed of head movement, gaze direction, duration of eye contact, and speed of eye blinks. An example of such a rule is that the speaker turns to look at the listener to relinquish control of the

conversation. The results look good but the system generates the same expression in any situation.

A significant amount of research followed the above works to derive rule-based emotional aspects to a dialogue. These rule-based systems are based on psychological research into emotions, such as Ortony et al.'s theory of appraisal.

Rosis, Pelachaud, Poggi, Carofiglio, and Carolis (2003) provided research toward automating the emotional component of speech based on a model that emotions are driven by beliefs and goals, and modified by personality and culture. Their dynamic belief network combined a belief network that represented the agent's mental state, and a network that monitored emotional triggers to another state. The belief network comprises three types of nodes: belief nodes, goal nodes, and goal-achievement nodes. Weights were associated with goal-achievement nodes as a function of the agent's personality. The system simulated how emotions are triggered and decay over time. However, there were no variations in specific emotional expressions.

Bui, Heylen, Nijholt, and Poel (2001) presented a fuzzy rule based system to map emotional states to facial expressions. They described six channels of facial movements which may have conflicts with one another: manipulators to satisfy biological requirements, lip movements for phonemes, conversational signals to emphasize speech, emotional displays, gaze movements, and head movements to support eye contact or to point to something during the conversation. The dominance model of Cohen and Massaro (1993) was used for co-articulation of the phonemes. Movements from the other channels were combined, and conflicts in parameters were resolved with those of higher

priority dominating the ones with lower priority. The emotional component consisted of the six basic emotions described by Ekman and Friesen (1978), and could be varied by intensity. Consider the example of smiling while speaking the word "hello." Normally, the phoneme "o" requires a pursed shape of the lips while the smile requires the lip corners to be pulled outward. The speech phoneme will have the higher priority, unless the speaker is so happy that the words are not spoken naturally. Variations in emotional expression occurred in the mouth area with respect to the interaction of emotion and speech.

Ho and Huang (2004) developed a facial modeling system using a “coarse-to-fine” genetic algorithm. They used a GA to acquire the 3D coordinates of control points of a face image. The control points can then determine the 3D facial model based on the topological and geometric descriptions of the generic facial model. Thus, the input to the system is a 2D image, and the output is a 3D facial model. The topology of the 3D facial model is a set of well-designed triangular polygons that are constructed by control points. The topology is described as $M = (V,U,T)$ where V is the set of all coordinates for all control points and vertices, U is a set of functions describing the weighted linear combination of neighboring control points, and T is the set of control points and curvature parameters for each vertex. The geometric values are obtained from a set of training facial models. These include all the 3D coordinates of control points, all of the curvature parameters for every vertex generation function, all the weights in the interrelationship functions of set U, and the statistical model ratios. The ratios include data such as the distance between the two far corners of the eyes. The GA is used to

obtain a good set of geometric values for the generic facial model. The fitness function is the Euclidean distance between the feature vector obtained from the training images to the feature vector of the GA chromosomes. The feature vector is the concatenation of the NFDs of three projections at different angles. The chromosome has to be decoded, a facial model constructed, three facial models projected, and the NFDs of the three models computed to obtain the feature vector.

The chromosome was encoded as a vector $((m_1, d_1), (m_2, d_2), \ldots, (m_n, d_n))$ with $2*n$ parameters, where n is the number of control points, m_i is an integer from 0 to 26, and d_i is an integer from 1 to N_{part}, i ranges from 1 to n, m_i is the moving direction in 3D space, and N_{part} is the partition number of search space. A single-layered facial model consists of 24 control points and 33 vertices, while a two-layered model consists of 24 control points and 145 vertices.

The researchers present coarse-to-fine as global and local searches, respectively. They use a parameter radius of the search space as a tuning parameter. Several experiments were performed, and success was measured by the relative error and convergence time of the coarse-to-fine GA as compared to the standard GA.

Interactive Genetic Algorithms

IGAs have a history of artistic image creation, including graphic art and animation, 2D facial images, lighting, and virtual reality spaces (Takagi, 2001). The specific technique of using a NN surrogate function with a large population has often been used in the music domain to evolve sequences of music.

Halim and Al-Fiadh (2006) developed an interactive genetic algorithm to assist victims in identifying criminals by evolving facial components based on user feedback. The chromosome was comprised of six genes representing background, forehead, eyebrows, eyes, nose, and mouth. The alleles contained data regarding the size and location of the face part. The initial population was 16 individuals. The witness chose the best composite, and rated each face part. A fitness function using these ratings was applied to the population, and eight individuals were selected for mating. The less fit individuals were eliminated. Different crossover and mutation probabilities were used for each face part based on the provided rating. Each of the eight mating individuals was crossed with the selected individual to generate 16 offspring. The algorithm terminated when the witness was satisfied with the image or after 15 generations.

Tokui (2000) used a multilevel neural network as a fitness approximation function to evolve music composition. The system maintained two populations, one of which consisted of short pieces of rhythms, and the other represented sequences of the rhythm pieces. The two populations evolved separately. The NN was given the elements of the GA genotype as input and the estimated fitness value as output. The NN learned through these examples how a user was likely to rate a given individual. By choosing only individuals with a high NN output score, only most likely candidates were presented to the user for evaluation.

Sun, Gong, and Li (2009) used a support vector neural network as a fitness approximation function in the fashion design domain. They used a small population size of 8 to compare the performance with a standard IGA using only human evaluation.

Their IGA used tournament selection with size 2, one-point crossover with a probability of .85, and one-point mutation with mutation probability of .05. Results showed that the use of the fitness approximation function reduced the time until convergence by 40% and the user satisfaction with the results were greater. These results were more apparent as the population size increased.

An example of large parameter GA that uses a subjective fitness function is the music composition by Dahstedt (2007). He used a representation based on recursive binary trees. Each leaf of the genome tree contained a note or a list of notes. The branches either concatenate or merge the notes into larger musical structures. For each note, information is kept for onset time, pitch, amplitude, duration, and articulated duration. Each piece was limited to 15 seconds. Since notes can be full notes, quarter notes, half notes, or one/sixteenth notes, the number of leaf nodes in the genome tree varied significantly. Clearly, the search space is very large. Three approaches were used to initialize the population. First, all the parameters were randomly generated. Second, stored genomes from previous human-evaluated evolution runs were used. Third, music input from a keyboard was translated into a tree structure. Populations were between 25 and 50, and ten generations were used for the subjective GA. Several rules were developed to weed out pieces that were not likely to be rated very well by the human. These rules were based on observation and statistics gathered from the interactive selection process and perform the role of a surrogate fitness function. The authors concluded that the output was structurally complex and musically convincing.

Measuring Animation Quality

It is important to be able to objectively evaluate the quality of facial animation. Facial expressions are an important subject not only in computer animation field, but also in the cognitive sciences, where it has been studied extensively over the last few decades. Wallraven, Breidt, Cunningham, and Bulthoff (2005) presented a framework in which psychophysical experiments evaluate the perceptual quality of facial animations. To do so, they drew upon a large body of research in the field of psychology, psychophysics, and neurophysiology. They demonstrated that experimental methods from the perceptual sciences allow defining and measuring the perceived realism of computer-generated images.

The methodology presented uses untrained participants to judge aspects of the avatar expressions. Since humans are sensitive to a large range of subtle aspects of facial motion, animation evaluations must include a large enough set of perceptual evaluation criteria to fully capture why an expression is judged to be realistic. The most fundamental test is recognition of expressions. Other important measures that have been identified are intensity, sincerity, and typicality of the expression. Finally, the time required to identify or rate the expression was taken into consideration. These measures enable the formation of a complete picture of the perceptual processing of facial expressions. To demonstrate the effectiveness of psychophysical experiments, the researchers used several different variations of generating the avatar. Participants were presented with an animation sequence repeatedly until they pressed an indicator that they were ready to evaluate the expression, yielding a measure for reaction time.

The collected data were analyzed using standard analysis of variance methods, which yield the statistical significance of each variable for the different measures. Variables included animation style, and changes in shape, texture, motion, timing, and frequency. The researchers demonstrated that this technique can be used to indicate the perceptual impact of different information channels, revealing what the animator needs to focus on to achieve perceptual realism.

Chapter 3

Methodology

Overview

The proposed research is based primarily on genetic algorithms and neural networks, both of which are well-established machine learning methods. Before describing the specifics of the proposed project, the concepts that form the basis of the design decisions are described in the following section.

Genetic Algorithm

GAs belong to the class of stochastic search methods. Whereas most stochastic search methods operate on a single solution to the problem, GAs operate on a population of solutions.

A GA is a process that mimics evolution. The canonical GA evolves a population of chromosomes, each of which represents a potential solution to a problem. The genes of the chromosome encode a particular element of the problem. The possible values of the genes are called alleles. The fitness of each chromosome determines how likely it is to be selected for reproduction. The operations that are applied to create the next generation of chromosomes include selection, crossover, mutation, and replacement. Selection selects individuals from the current population for mating. Crossover consists

of exchanging material between two chromosomes. Mutation consists of replacing the value of a randomly chosen gene. Replacement merges the current population and its offspring to create the next generation.

There are no strong conclusions on when a genetic algorithm will outperform traditional methods (Mitchell, 1998). If the search space is large or not well structured, or if the fitness function is noisy, and if a global optimum is not required, GAs can be effective. A GA performs a parallel search guided by a fitness function. There are several parameters that can be adjusted to improve performance. Two components unique to each problem that are critical to the success of a genetic algorithm are genotype (chromosome representation) and the fitness function used to guide the evolution of the population of potential solutions. Other GA parameters have guidelines backed by research, but are also highly dependent on the specific problem.

The genes of the chromosome (genotype) in a population must map to a potential solution (phenotype). The size of the solution space tends to increase with each additional gene in the chromosome, but a many-to-one mapping mitigates this effect. If the number of genes is too small, the solution space may be inadequate to produce an acceptable solution. If the number of genes is too large, it may take too much time to evolve an acceptable solution. The design of the GA must consider the best balance between these two issues.

When a predictive fitness function is used together with the original fitness function, the fitness function to be optimized is $f(X)$ if the original fitness function is used, and $f(X)+E(X)$ if the predictive fitness function is used, as depicted in the equation

$$F(X) = \begin{cases} f(X) \\ f(X) + E(X) \end{cases}$$

where *E(X)* is the approximation error of the predictive function. The error in the predictive function is deterministic once it is constructed and so cannot be reduced by resampling. Instead the error has to be addressed by using the true fitness function rather than the approximation function. The design of the GA must consider the best balance between cheap but inaccurate predictive evaluations and expensive but true fitness evaluations.

The issue of integrating the two fitness functions is called evolution control or model management. The two basic categories are individual-based evolution control, in which a subset of the population are evaluated with the true fitness function in every generation, and generation-based evolution control, where all individuals are evaluated with the true fitness function in a subset of generations.

One common method of individual-based evolution control is that all individuals are evaluated using the predictive function, and then a number of individuals are chosen to be re-evaluated using the true fitness function. There are several methods that have been used for determining which individuals are re-evaluated. The selection can be random, it can be based on the best individuals according to the predictive function, it can choose the most uncertain individuals, it can choose the best from a cluster of individuals, or it can use a combination of quality and uncertainty. If we assume the prediction function is better than a random guess, it is natural to choose the best individuals to be

reevaluated. Usually, the number of individuals to be reevaluated is predefined and fixed during the evolution.

The design of the approximation function must balance the trade-off between approximation accuracy and model complexity. The model complexity should be controlled to avoid overfitting. In the case of neural networks, structural optimization can improve the accuracy of the predictive fitness function significantly. Proper training data can also have a significant effect on the accuracy of the predictive function.

A GA is sensitive to parameters such as population size, crossover and mutation probabilities. General guidelines for GA parameters were proposed by Harrold and Grefenstette, 1986. Several researchers have performed research on how to further optimize details of the GA, including population size (Harik, Cantú-Paz, Goldberg, and Miller, 1999), crossover techniques (De Jong and Spears, 1992), and selection methods (Goldberg and Deb, 1991). There is interdependence among GA parameters so they cannot be optimized independently. In particular, population size, crossover, and mutation rate interact nonlinearly with one another (Mitchell, 1998).

Initialization of the population is usually random but one may seed the population with individuals that meet some criterion. The evolution of new generations continues until an individual appears that meets the fitness criterion or until a predetermined maximum number of generations have been met. The fitness for an individual is usually determined algorithmically, using a mathematical optimization. In the case of a subjective goal, a human may be used to supply the fitness function.

The selection step in a GA reflects the evolutionary principle that the most fit individuals survive and reproduce. It is usually a random process biased by the fitness value so that the most fit individuals are more likely, but not guaranteed, to mate.

Combining the parents' genes using a crossover function produces a child chromosome. The child may also be modified with a random mutation with a specified probability that is typically quite small. In the canonical GA, these operations mimic what occurs in natural organisms. Often, natural genetics is more of an inspiration than a constraint, and many domain-specific operators have been invented.

In creating the next generation, some old individuals, usually the most fit, may survive intact, creating a generation gap between them and the younger individuals. The size of this gap, along with the size of the population, affects the speed of the search. A large population with no generation gap will cover the most ground, but might lose the best individual. A small population with a large gap will not lose the best individuals, but will take more time to explore the solution space.

Neural Network Classification

Neural networks can be trained to perform complex non-linear functions, such as pattern recognition. Our research employs six feedforward neural networks trained with back propagation. Input vectors (FACS intensity values) and the corresponding target output (emotion) were used to train the networks. The quality of the NNs was crucial to the effectiveness of the IGA system.

The problem of learning in a neural network can be framed as minimization of an error function E (Bishop, 1995). The error is a function of the weights and biases in a

network, which can be grouped together into a single W-dimensional weight vector $w_1..w_W$. One of the effective training algorithms is a variation of gradient descent called scaled conjugate gradient.

Gradient Descent

The gradient descent algorithm searches along the direction of steepest descent, and the weights are updated using

$$\Delta w^r = -\eta \nabla E^n \mid_{w^r}$$

where η is the learning rate, and provided it is sufficiently small, the value of E will decrease each step leading to a minima where the vector $\nabla E=0$. There are problems with convergence with the simple gradient descent algorithm, however. It is difficult to find a suitable value for η. The error surface may contain areas where most points do not point towards the minimum, resulting in a very inefficient procedure. The basic algorithm can be enhanced by adding a momentum term μ to smooth out the oscillations, and by updating the learning rate (Bishop, 1995).

Conjugate Gradient Descent

Another issue with gradient descent is choosing a suitable search direction. Suppose we have minimized along a line given by the local gradient vector. Choosing successive search directions can lead to oscillations while making little progress toward the minimum. For this problem, conjugate gradients are employed. Suppose a line search has been performed along the direction d^r starting from point w^r to give an error

minimum along the search path at the point w^{r+1}. The direction d^{r+1} is said to be conjugate to the direction d^{r} if the component of the gradient parallel to the direction d^{r}, which has been made zero, remains zero as we move along the direction d^{r+1}. It can be shown that the minimum of a general quadratic error function can be found in at most W steps using conjugate gradients (Bishop, 1995).

Scaled Conjugate Gradient

A basic problem with line search is that every line minimization involves several error function evaluations, each of which is computationally expensive. The procedure also involves a parameter whose value determines the termination criteria for each line search. The performance is sensitive to this value. The scaled conjugate gradient algorithm avoids the expense of line minimization by evaluating Hd_j where H is the Hessian matrix comprised of the second derivatives of the error

$$\frac{\partial^2 E}{\partial w_{ji} \partial w_{lk}}$$

However, it is necessary to ensure that H is positive definite so that the denominator doesn't become negative and thus increase the error. Adding a multiple of the unit matrix does this

$$H + \lambda I$$

where I is the unit matrix and $\lambda \geq 0$ is a scaling coefficient. The formula for the step length is then given by

$$\alpha_j = -\frac{d_j^T g_j}{d_j^T H_j d_j + \lambda_j \| d_j \|^2}$$

where d_j is the direction at step j, and g_j is the gradient vector at the jth step orthogonal to all previous conjugate directions. The suffix j on λ_j reflects that the optimum value for this parameter can vary on each iteration. Techniques like this are well known in standard optimization where they are called model trust regions. The model is only trusted in a small region around the current search point. The size of the trust region is controlled by λ_j so that for large λ_j the trust region is small. In regions where the quadratic approximation is good, the value of λj should be reduced, while if the quadratic approximation is poor, λ_j should be increased. This is achieved by considering the following comparison parameter

$$\Delta_j = \frac{2\{E(w_j) - E(w_j + \alpha_j d_j)\}}{\alpha_j d_j^T g_j}$$

The value of λ_j is then adjusted with

$$\text{If } \Delta_j > 0.75 \text{ then } \lambda_{j+1} = \frac{\lambda_j}{2}$$

$$\text{If } \Delta_j < 0.25 \text{ then } \lambda_{j+1} = 4\lambda_j$$

$$\text{Else } \lambda_{j+1} = \lambda_j$$

If $\Delta_j < 0$, the step would actually increase the error so the weights are not updated, but

instead the value of λ_j is increased and Δ_j is re-evaluated. Eventually an error decrease will occur since once λ_j becomes large enough, the algorithm will be taking a small step in the direction of the negative gradient. The two stages of increasing λ_j if required and adjusting λ_j are applied in succession after each weight update (Bishop, 1995).

Hidden Layers

There is no theoretical reason to ever use more than two hidden layers, and for the majority of practical problems, there is no reason to use more than one hidden layer. The problems with multiple hidden layers include longer training times, the gradient is more unstable, and the number of false minima increases dramatically (Masters, 1993).

Long training times, overfitting, and loss of generalization can be caused by too many hidden neurons. The network may learn insignificant aspects of the training set that are irrelevant to the general population. Too few neurons and the network is not able to learn the pattern at all. The number of required neurons is dependent on the complexity of the function to be learned, and is usually discovered through experimentation.

Neural Network Costs

A detailed analysis of the scaled conjugate gradient algorithm by the researcher who introduced the algorithm (Moller, 1993) shows that it has a calculation complexity of $O(6N^2)$ for each iteration, where N is the number of weights and biases in the network.

The computational complexity of the neural network is affected by both informational complexity and neural complexity. The informational complexity is the

number of examples required to approximate the function *f* within a given tolerance *ε*. The neural complexity is the number of neurons necessary to approximate *f* within *ε*.

Baum and Haussler (1989) considered the number of training samples required for multilayer feed-forward networks. For a network with *M* units and *W* weights, including biases, they gave an upper bound on the capacity of the network as

$$d_{VC} \leq 2W \log_2(eM)$$

where *e* is the base of natural logarithms. They used this to show that, if some number *N* of patterns, given by

$$N \geq \frac{W}{\varepsilon} \log_2(\frac{M}{\varepsilon})$$

can be learned so that *1-ε/2* are correctly classified, with $0 < \varepsilon <= 1/8$, then there is a high probability that 1-*ε* future examples will also be correctly classified. They further derived a rule of thumb that to correctly classify 1-*ε* of new examples requires a minimum number of training patterns of *W/ε*. Thus, for $\varepsilon = 0.1$, one would need around ten times the training examples as there are weights in the network. The Baum-Haussler rule of thumb is based on the worst-case bounds, so good generalization is possible with fewer training patterns.

Our research project used a neural network comprised of 30 hidden neurons, 39 input neurons, and 1 output neuron for a total of 1200 weights in the network. The average number of iterations to reach convergence was 118, with a range from 105 to 127 among the six networks. Using Moller's equation, the number of calculations would be in the neighborhood of $118 * 6 * 1200^2$ ($1.019X10^9$) for a training run. Our NNs used an

average of 793 training samples each ($8.085X10^{11}$ calculations). On a Macbook Pro with a 2.66 GHz Intel Core i7 processor, the running time of training a neural network was noticeable, at times taking as long as 30 seconds. This was a relatively small amount of time in relation to the overall time required to build and use the IGA system since the NN training was initiated by the user as needed and not automatically executed by the IGA.

Specific Research Methods to be Employed

IGA System Design Overview

Our research used an interactive GA to generate realistic variations in facial expressions. Given a target emotion, the IGA system evolves an animation sequence of a face expressing that emotion.

The chromosome represents a sequence of *n* genes, each of which encodes a face model that comprises the complete set of AUs. The genes are sequenced so that the *i*th gene represents the *i*th face model in the animation sequence. The number of face models does not evolve, but is a parameter that can be set by the user.

Since FACS does not incorporate temporal data, typical values were used for the rate of interpolation between two expressions, and for the duration of a static expression before beginning the next interpolation. The face models are key-framed every four frames, with the first face model always being the neutral expression. The last face model is repeated twice for eight additional frames to present a pause when looping the animation.

Table 1: Chromosome Mapping of n genes

Face Model 1	Face Model 2	…	Face Model n
$AU1_1 \ldots AU39_1$	$AU1_2 \ldots AU39_2$	…	$AU1_n..AU39_n$

AUk_i=*k*th action unit of the *i*th face model

In order to take advantage of the biodiversity in a large population, a surrogate fitness function in the form of a neural network was used to evaluate the entire population, and a subset of the most promising individuals according to the NN was

chosen for the user to evaluate. The NNs were trained to evaluate a single facial expression. Thus, the NN performs *n* evaluations on each chromosome, one evaluation for each face model.

Consider a chromosome with *n* genes. The chromosome is given a surrogate fitness value corresponding to how many of its genes are classified as the target emotion by the NN. The inputs to the NN are the values of the facial action units for a gene.

Table 2: NN Input/Output

Happiness NN Classifier	
Input	Output
Face Model 1 AUs	Y
Face Model 2 AUs	Y
.	.
.	.
.	.
Face Model n AUs	Y

The NN performs its fitness calculations on the genotype. However, the user assigns a fitness value on the phenotype of the individuals most highly rated by the NN. The phenotype is an animation sequence of facial expressions as displayed by the graphics software. The facial action units encoded in the gene map directly to blendshape control values. Blendshape controls comprise the blendshape animation rig that is used to generate the phenotype. The blendshape control values are constrained to allow a range of motion that is humanly possible. A small number n_s of animation sequences are displayed to the user for evaluation.

The evolution continues using conventional GA parameters until the user is satisfied with the results, the user aborts the run, or the maximum number of generations is reached.

The genetic algorithm can be described as follows: given a target emotion, evolve an animation sequence that exhibits variations in the specified expression. Initialize a large population with n_p individuals. The NN evaluates every gene in every chromosome in the population. The chromosome's surrogate fitness value is calculated by totaling the surrogate fitness values of each of its genes. A subset of n_s of the most highly fit individuals according to the NN are chosen for subjective evaluation, with eight being a promising value for n_s. These eight genotypes are converted to phenotypes for display. The user assigns fitness values to the displayed animations. The most highly fit chromosomes in the population are selected for breeding the next generation. Convergence occurs when the user is satisfied or when a fixed number of generations have been performed.

Table 3: IGA System Algorithm

1) User selects target emotion to be expressed
2) Initialize large Face Model GA population with n_p individuals
3) Apply surrogate fitness function to total population
4) Generate phenotypes for subset of n_s most-fit individuals
5) User evaluates phenotypes
6) Select mating population
7) Create next generation with crossover and mutation
8) Repeat from step 3 until convergence

The primary activities that were performed to build the IGA system are listed below, and described in more detail in the following sections.

Table 4: Primary Activities to Build IGA System

1) Prepare the blendshapes and controls
2) Define the genotype
3) Map the genotype to the phenotype
4) Define the constraints on genotype values
5) Design the genetic algorithm
6) Generate the phenotype
7) Prepare the training example database
8) Design the neural network
9) Design the human-computer interface
10) Implement GUI, genetic algorithm, neural network, and graphic software.

IGA System Design Details

Prepare the blendshapes and controls

Blendshapes are based on FACS to separate the facial movements into simple, distinct action units. Blendshapes can be combined to create different expressions. The blendshape controls change the weight of each blendshape. They are often depicted graphically as a slider to enable a user to manually increase or decrease the effect of the desired blendshape. In our project, a polygonal face model has 39 blendshape controls, each of which has a range between 0 and 1, where 0 is the value for the neutral expression.

Define the genotype

The chromosome is comprised of a user-selectable number of genes. Each gene encodes an expression. The genes are arranged in sequence so the *i*th gene encodes the face model of the *i*th keyframe in the animation. An expression is defined as a combination of facial action units based on the widely embraced FACS. Thus, each gene is comprised of action units. The facial action units define all visible changes that a face can perform.

Although a subset of action units can be used to recognize emotions, the entire set is needed to generate the full range of variety. It is a large problem, but previous research exists that defines a large chromosome to represent the face (Ho and Huang, 2001; Karungaru, Fukumi, and Akamatsu, 2007). To control the size of the solution space, the number of gene components involved in the evolution is a subset of the AUs on the gene. The AUs that are known to express the target emotion evolve, plus n_r genes that

are chosen at random. The number of random genes is a parameter that is set by the user. The remaining genes have fixed values for the neutral expression.

The chromosome is encoded as a vector *((*$AU1_1 \ldots AU39_1$*), (*$AU1_2..AU39_2$*), ..., (*$AU1_n..AU39_n$*))* where *n* is the number of face models, and AUk_i is *k*th action unit of the *i*th face model.

Map the genotype to the phenotype

The phenotype is an animation sequence of facial expressions in the Autodesk Maya 2010 graphics program. Maya has an API for C++ capable of driving the blendshape controls used in animation. After each new generation, the genetic algorithm provides the blendshape control values for *n* face models in n_s animations from the allele values for *n* genes in the best n_s chromosomes. When determining the best chromosomes, the GA gives priority to the subjective fitness value if one exists. Otherwise, the surrogate fitness provided by the NN is used.

The Maya script keyframes the blendshapes associated with the n_s visible face model groups using the blendshape control values provided by the GA. The values of the alleles map directly to the values of the blendshapes. Maya creates the animation phenotype by interpolating the blendshape values between keyframes.

Define the constraints on genotype values

Facial action units have limits on their values. To eliminate impossible facial expressions, constraints are placed on how far up an eyebrow can rise for example. The

constraints are placed on the blendshape controls. A value of 1 is the maximum range of the facial action unit.

Darwin claimed that all people express some emotions in the same ways in their faces. There have been several studies conducted that support this claim with respect to the basic emotions (Matsumoto and Ekman, 2008). Our genetic algorithm was optimized by selecting the specific facial action units for that target emotion and a small number of random action units to evolve. All other action units retain the values for the neutral expression. This eliminates a large range of useless solutions from evolving. Additionally, in the manually generated animations that are presented to the participants for comparison, only the specific facial action units for the target emotion were manipulated.

Figure 1: FACS Coding of Fear (Matsumoto and Ekman, 2008)

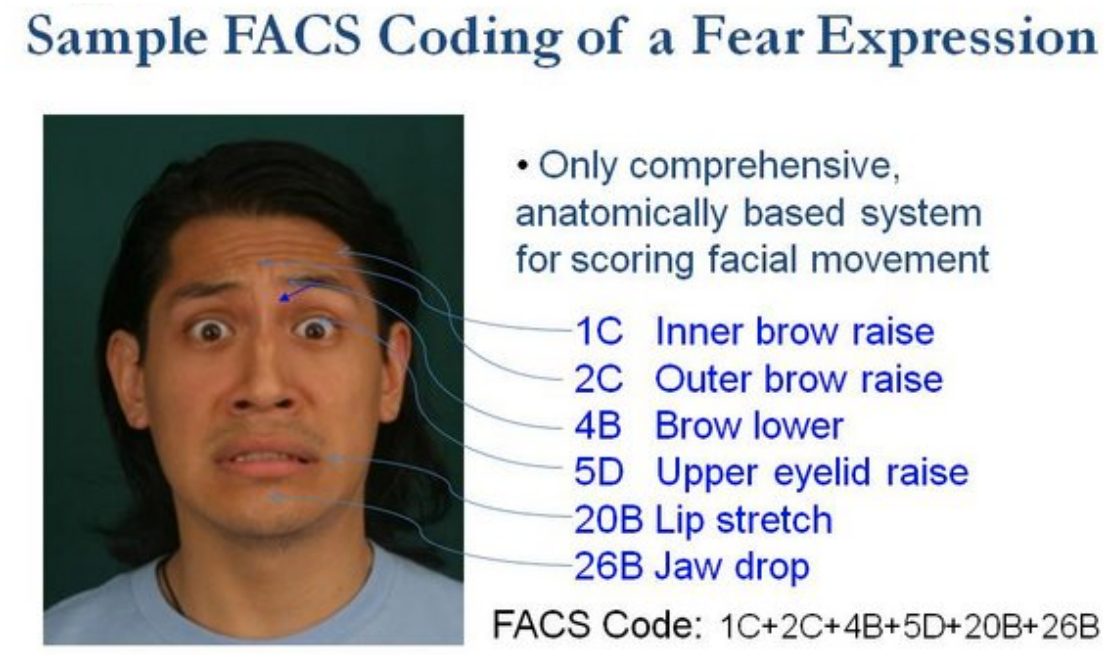


Table 5: AUs of Basic Emotions (Matsumoto and Ekman, 2008)

Emotion	AU #	FACS AU
Anger	4	Brow Lowerer
	5 and/or 7	Upper Lid Raiser or Lid Tightener
	22	Lip Funneler
	23	Lip Tightener
	24	Lip Pressor
Disgust	10	Upper Lip Raiser
	16	Lower Lip Depressor
	22	Lip Funneler
	25 or 26	Lips Part or Jaw Drop
Fear	1	Inner Brow Raiser
	2	Outer Brow Raiser
	4	Brow Lowerer
	5	Upper Lid Raiser
	7	Lid Tightener
	20	Lip Stretcher
	25 or 26	Lips Part or Jaw Drop
Happiness	6	Cheek Raiser
	12	Lip Corner Puller
Sadness	1	Inner Brow Raiser
	4	Brow Lowerer
	15	Lip Corner Depressor
	17	Chin Raiser
Surprise	1	Inner Brow Raiser
	2	Outer Brow Raiser
	5	Upper Lid Raiser
	25 or 26	Lips Part or Jaw Drop

Design the genetic algorithm

Initialization

The search space is very large, and even a large population of 100 is relatively small in comparison. In this circumstance, the initial population is very important, since it determines the starting point in search space. A randomly generated initial population may provide the maximal variation and coverage. However, if one wants to explore a certain neighborhood in the search space, it may be helpful to bias the initialization. This is sometimes done with previously evolved individuals, human input, or predefined constraints (Dahlstedt, 2008).

The action units that express specific emotions are a well-defined subset of the complete set of AUs defined by the FACS. The parts of the gene encoding the action units that express the target emotion will be referred to as the target gene components. The number of target gene components n_t varies with each emotion. In addition, a user selectable number n_r of random gene components evolve. The target gene components provide the predictable component of an expression, while the random gene components provide the random component. By varying the parameter n_r, the degree of randomness can be controlled.

During initialization, the target gene components and random gene components are given random values that represent the degree to which the corresponding action unit is activated. The remaining gene components maintain standard fixed values.

The population size is a configurable parameter, with 100 being used in most of our experiments.

Selection

The selection technique must be strong enough so that evolution is not too slow as to cause user fatigue or boredom, and yet not so strong that suboptimal individuals do not take over the population prematurely. It is important to balance the exploitation and exploration conflict. The slow growth rates that encourage thorough exploration may be impractical for an IGA. Due to user fatigue, it is likely preferable to emphasize exploitation and reduce convergence time. Since there are often multiple acceptable solutions in the facial expression domain, local optima are less of an issue.

Fitness-proportionate selection often puts too much emphasis on exploitation of highly fit strings early in the evolution, causing them to multiply rapidly and prevent sufficient exploration (Mitchell, 1998). Ranking and tournament methods have similar effects on selection, reducing the selection pressure when the fitness variance is high, and increasing it when the fitness variance is low.

Tournament selection is more efficient than ranking. The selection pressure can be adjusted by modifying the tournament size. As the tournament size increases, the convergence time is decreased (Goldberg and Deb, 1991). Additionally, it has been claimed that tournament selection achieves niching implicitly (Muhlenbein, 1989). Niching allows for two strings that are relatively equal in fitness to get relatively equal

samples, thus maintaining useful diversity and encouraging continued exploration. It compensates for the tendency of a highly fit individual rapidly taking over the population.

Our IGA system uses the tournament method to take advantage of the efficiency, adjustments, and niching aspects. If two individuals have the same fitness value, with one being provided by the surrogate fitness function and the other being provided by the subjective fitness function, the one with the subjective fitness value wins the tournament.

Replacement

The most common replacement strategies are generational and steady state. In the generational algorithm, the entire population is replaced each generation. The steady state algorithm replaces only a few individuals in each generation. The most common steady state replacement algorithms are replace-worst and replace-most-similar.

Most GAs described in the literature are generational. One technique that is common to a generational GA is called elitism, and keeps some of the most fit individuals in the successive generation. Steady-state selection is useful in GAs where incremental learning is important, and in which members of the population collectively rather than individually solve the problem at hand.

In our IGA system, generational replacement with elitism is used.

Crossover

Although single-point crossover is the most common in the literature, it has the disadvantages of having positional bias, being unable to combine all possible schemas, tending to preserve hitchhikers, and treating endpoint loci preferentially (Mitchell, 1998).

More aggressive crossover, such as 5-point or uniform, increases the diversity but also disrupts the desired schema from forming. Two-point crossover reduces positional bias and the endpoint effect, but there are still schemas that two-point crossover cannot combine. Some researchers promote the use of parameterized uniform crossover, in which an exchange occurs at each locus with a probability typically between 0.5 and 0.8. This has no positional bias but can prevent coadapted alleles from ever forming since it can be quite disruptive of the schema.

In our IGA system, two-point crossover is used, and the crossover rate is configurable. The crossover points occur between each gene so that the genes are indivisible units in the crossover operation.

Mutation

The mutation rate is very low, comparable to that used by other researchers. Mutation plays a minor role in maintaining diversity when the initial population is large enough. With smaller populations, mutation is the primary mechanism for generating diversity. Since the intent of using a surrogate function in the IGA is to enable a sufficiently large population, the traditional small mutation rate is used.

Mutation is applied to each gene with a probability p_m. When it is determined that a chromosome should undergo mutation, one of its alleles is chosen at random and given a random value across all of its genes.

Population Size

The population must be large enough to contain enough building blocks for a high-quality solution, and yet not so large that time is wasted processing unnecessary individuals and contributing to user fatigue in IGAs. The size is dependent on the number of desired alleles in the initial population, the size of the problem, and the selection intensity (Harik, Cantu-Paz, Goldberg, & Miller, 1999).

Ensuring there are a sufficient number of building blocks in the initial population is essential to a successful GA (Goldberg, Sastry, and Latoza, 2002). Subsequent steps of ensuring the growth of superior BBs, the mixing of BBs, and good decisions among competing BBs are all based on having an initial adequate supply (Goldberg, 2002).

Our IGA population size is a configurable parameter $n_{p,}$ which must be large enough to ensure a sufficient number of building blocks. The initial population is significantly larger than the typical IGA, which tends to use small populations to reduce user fatigue. In our IGA, user fatigue is addressed by selecting a subset of eight to be evaluated by the user, while the faster surrogate fitness function is applied to the total set.

Table 6: IGA Design Parameters

Initialization	Random values given to a subset $n_t + n_r$ alleles, where n_t refers to the number of target emotion alleles, and n_r refers to the randomly selected alleles.
Selection	Tournament of size 8, with subjective fitness given priority over surrogate fitness
Replacement	Generational with elitism.
Crossover	Two-point crossover with probability p_c where the crossover points are between genes.
Mutation	Random value given to random allele in set of genes with probability p_m
Population Size	n_p configurable
Termination	User satisfied (convergence), user aborted, or 20-generation limit.

Generate the phenotype

The phenotype of the IGA is an animated sequence of facial expressions. The animation is generated in the Autodesk Maya 3D Animation Software. The alleles of the gene are mapped to blendshape control values, which are used by the Maya Embedded Language (MEL). Maya makes an API to this scripting language available to C++.

Prepare the training example database

The NN that serves as the surrogate fitness function must be trained with examples for the target expression. It is common for research in facial expressions to

focus on the six basic emotions as described by Ekman and Friesen (1978). These basic emotions are happiness, sadness, disgust, anger, fear, and surprise.

There are twelve separate databases, one for positive and one for negative samples of each of the basic emotions. The database entries correspond to the alleles of a gene, which represent the facial action units of each face model. The databases were initialized with enough manually generated examples to attain at least an 80% success rate in classifying an expression. But in the initial runs, the random gene components created unrealistic expressions since the NNs were trained only to classify emotion. The NNs were not differentiating between valid and invalid expressions. A mechanism was introduced to add samples of realistic and unrealistic expressions to the NN training databases from the individual genes that were generated by the IGA. The user could choose to step through the individual face models of any animation and add one or more to the positive or negative training database for the target emotion. The user could also choose to retrain the NN during the IGA.

It should be noted that a chromosome corresponds to a sequence of genes, so the NN is evaluating the genes of the chromosome. The fitness value of the chromosome is the sum of the fitness value of each of its genes.

Design the neural network

Neural networks have been shown to be effective in classifying human facial expressions. NNs are a very powerful and general framework for representing non-linear mappings from several input variables to several output variables (Bishop, 1995). The mapping is based on parameters that are learned from a set of training data. A neural

network classifier can be seen as a function approximation, where the approximated functions are the probabilities of membership of the different classes expressed as functions of the input variables (Bishop, 1995).

Facial expression classification is a high-dimensional problem. As the number of input features increase relative to the size of the training samples, the classifier loses its ability to generalize. In the most extreme case, it becomes a look-up table that only matches patterns it has seen before. Another problem has been called 'the curse of dimensionality' in that the number of required examples grows exponentially with the number of input features. Thus, the feature extraction step has a profound effect on the pattern recognition system of high-dimensional data (Bishop, 1995).

When working with video sequences, techniques for reducing the dimensionality of the data include optical flow, principal component analysis, local feature analysis, and Gabor wavelet representations (Donato, Barlett, Hager, Ekman, and Sejnowski, 1999). The dimensionality of the data using these techniques is comparable to the dimensionality of our chromosome representation, which is still rather high.

It has been demonstrated that a subset of facial features contribute the most to the recognition of emotion, especially the eyes, eyebrows, and mouth features (Cunningham, Kleiner, Bilthoff, & Wallraven, 2004). Initially, the input to the NN was the alleles most relevant to the recognition of emotion. Unfortunately, this resulted in a population of genes that were comprised of valid values for the 17 NN inputs, but unrealistic values for the remaining alleles. The input was changed to include all the alleles, and the NN was trained to recognize valid emotional expressions.

Table 7: AUs Relevant For Emotion Recognition

AU #	FACS name
1	Inner Brow Raiser
2	Outer Brow Raiser
4	Brow Lowerer
5	Upper Lid Raiser
6	Cheek Raiser
7	Lid Tightener
10	Upper Lip Raiser
12	Lip Corner Puller
15	Lip Corner Depressor
16	Lower Lip Depressor
17	Chin Raiser
20	Lip Stretcher
22	Lip Funneler
23	Lip Tightener
24	Lip Pressor
25	Lips Part
26	Jaw Drop

Initially, we planned to use Matlab as the environment for the neural network classification. However, the student version of Matlab does not have an API for calling the Matlab scripts from an external program. Instead, we used the Fast Artificial Neural

Network Library (FANN), which was developed at the University of Copenhagen and is now available as an open source library (Nissen, S., 2003).

There were six neural networks trained for binary classification of each of the basic emotions, using a small proportion of positive examples. Each NN has 39 inputs and 1 output.

The goal underlying the network design is to discover the simplest network architecture possible so that overfitting is avoided, and generalization is maximized. Bishop (1995) refers to this as finding the balance between bias and variance. The procedure is to start with a network that is too small to learn the problem, and continue adding hidden neurons (and if necessary, hidden layers) until the error function is acceptable, and there is insignificant improvement from the previous trial.

Our final NN configuration had a single hidden layer and 30 hidden neurons. The NNs use the scaled conjugate gradient training algorithm, with MSE as the performance function.

Design the human-computer interface

The eight chromosomes with the highest fitness value as determined by the surrogate fitness function are converted to phenotypes and presented to the user. The user evaluated each of the eight animation sequences using discrete values from 1 to 10. The use of a small range of values reduces user fatigue since it is difficult for users to rate small differences among objects (Ohsaki, Takagi, and Ohya, 1998).

The practical number of generations when using an interactive GA is in the range from 10 to 20. The user identifies when a satisfactory solution is achieved. If the

evolution is not producing promising candidates, the user may stop early and declare failure to converge. Otherwise, at the end of 20 generations, the user identifies whether the most-fit candidate is acceptable.

Implement GUI, genetic algorithm, neural network, and graphic software.

The IGA system requires a framework to manipulate representations of 3D facial models so that they can be modified by a genetic algorithm, evaluated by a neural network, and displayed with an interactive graphical user interface (GUI). The GUI is capable of gathering input from the user and providing output for the genetic algorithm populations, the neural network training databases, and the statistics necessary for evaluating the effectiveness of the system.

The genetic algorithm was implemented with C++. C++ was chosen since it has an interface with the Maya graphics software. The Maya API provides the ability to create windows that can display the face models and animation sequences. C++ also has open source libraries available for the neural network. The NN library that was used is called Fast Artificial Neural Network (Nissen, S., 2003).

Table 8: Major Software Components of IGA System

- genetic algorithm
- neural network
- database of training examples
- graphic animation software
- database of predefined blendshapes with controls
- interactive GUI

Measuring the Quality of the IGA System

The purpose of this research is to investigate the effectiveness of an IGA in generating credible variations in facial expressions. An IGA's effectiveness is measured by satisfaction with the end results, and acceptable user fatigue.

Convergence Analysis

User fatigue is the physical and psychological burden inherent in evaluating solutions over a long period of time. It is directly correlated with the number of evaluations conducted and the rate of successful convergences.

The effectiveness of the IGA is dependent on the effectiveness of the surrogate fitness function. The questions to be answered are "did the surrogate fitness function improve time to convergence, and if so, by how much?" The answers indicate whether it was worth the additional effort and complexity to introduce a surrogate fitness function.

There are many examples in the literature where the surrogate fitness function exerted a significant improvement on convergence.

It is intuitive that the higher the fidelity of the approximation function, the more likely the user will be satisfied with the population subset chosen for subjective evaluation, thus reducing convergence time. If the error of the approximation model is large, then convergence time will be large. Therefore, it is possible to determine the quality of the approximation function from the relationship between the fitness values provided by the approximation and those provided by the user. The model error can be estimated as follows:

$$E(k) = \sqrt{\frac{1}{p}\sum_{i=1}^{p}(y_{user}(i) - y_{nn}(i))^2}$$

where k is the *kth* generation, p is the population size, y_{user} is the fitness value provided by the user, and y_{nn} is the fitness function provided by the neural network

$$y_{nn} = \sum_{j=1}^{H} v_j \theta(\sum_{i=1}^{n} w_{ij} x_i)$$

where H is the number of hidden nodes, n is the number of inputs, w_{ij} is the weight between the input layer and hidden layer, v_j is the weight between the hidden layer and output layer, and $\theta(z) = \frac{1}{1+e^{-z}}$

Participant Statistics

It is important to evaluate the quality of the IGA system objectively. This has been done in the facial animation literature by using a method from the psychological and behavioral sciences that is designed to measure human perception (Cunningham, Kleiner, Bulthoff, and Wallraven (2004); Wallraven, Breidt, Cunningham, and Bulthoff, 2005). Psychophysical experiments can be used to measure satisfaction with the animation sequences. The experimental data to be analyzed was collected in the form of questionnaires that are completed by participants. Examples of questions that have been asked in facial animation evaluation include identification, credibility, intensity, aesthetic preference, sincerity, and naturalness. It has been shown that all of these criteria, as well as reaction time, contribute to the perceived realism of the animation.

The most important task is to identify the expression from a list of possible expressions, with an option for none of the above. The non-forced choice methodology avoids inflated accuracy ratings found in the absence of a 'none of the above' option and avoids the subjectivity associated with free descriptions (Cunningham, Kleiner, Bilthoff, & Wallraven, 2004). In contrast to the other questions, identification provides an objective qualitative criterion.

Intensity and sincerity reflect a higher-level impression of facial expressions. The ratings are of particular interest in areas such as virtual sales, where it is important to have convincing facial expressions.

Participants were asked to judge naturalness as something that people normally do. Additionally, any strange artifacts in the graphics should also be regarded as unnatural.

Aesthetic preference is evaluated by showing the participant a manually created animation sequence and an IGA created sequence side-by-side. Participants are asked to answer the question "which sequence captures the essence of the expression better?" Although the question asks for a very subjective evaluation, participants are forced to choose one answer, which allows for a clean analysis of the data.

The software components required for the statistics gathering are:

- Graphic animation software
- Interactive GUI
- Statistics database

After the IGA system generated the animation sequences, the statistics were gathered as follows:

1) The set of animation sequences were selected for evaluation.
2) A random animation sequence was shown to the participant.
3) Participant provided answers for identification, intensity, sincerity, credibility, and naturalness.
4) Repeated from (2) until all 24 sequences were shown.
5) A pair of IGA/non-IGA animations was shown to participant.
6) Participant provided answer for aesthetic preference.
7) Repeated from (5) until all 12 pairs were shown.

Each participant was thus asked to evaluate 36 animation sequences. This is an arbitrary number and is fixed for consistency.

The animation sequences shown to the participants were selected from the IGA evolved solutions, as well as the examples that were manually created for evaluation. Participants did not know the source of the sample being evaluated. Analysis of the two sets of data provides some evidence of the effectiveness of the IGA as compared to the standard method of animation.

Animation sequences were presented on a computer monitor. The animation plays in a repeating loop with a pause at the end. The participant could replay the sequences as often as desired. Below the animation window was a window containing the questions to be completed by the participant. Answers were recorded into a database for later analysis. The time between the start of the animation and the start of the questionnaire completion was recorded.

The questions were defined onscreen as follows.

1. Select the facial expression being displayed:
   - Anger
   - Disgust
   - Fear
   - Happiness
   - Sadness
   - Surprise
   - None of the above

2. Rate Intensity on a scale from 1-7, with 1 being the least intense and 7 being the most intense. Intensity is the degree to which an emotion appears to be felt. For example, a low intensity anger appears as irritation, whereas a high intensity anger appears as rage.

3. Rate Naturalness on a scale from 1-7, with 1 being the least natural, and 7 being the most natural. An expression is natural if it is something people normally do. Any artistic or technical faults in the animation should be considered unnatural.

4. Rate Sincerity on a scale from 1-7, with 1 meaning the avatar appears to be faking or pretending, and 7 meaning the avatar appears to really mean the underlying emotion.

5. Rate Credibility on a scale from 1-7, with 1 being the least credible and 7 being the most credible. Credibility reflects how believable or realistic the animation is.

Figure 2: Participant Evaluation Menu Screenshot

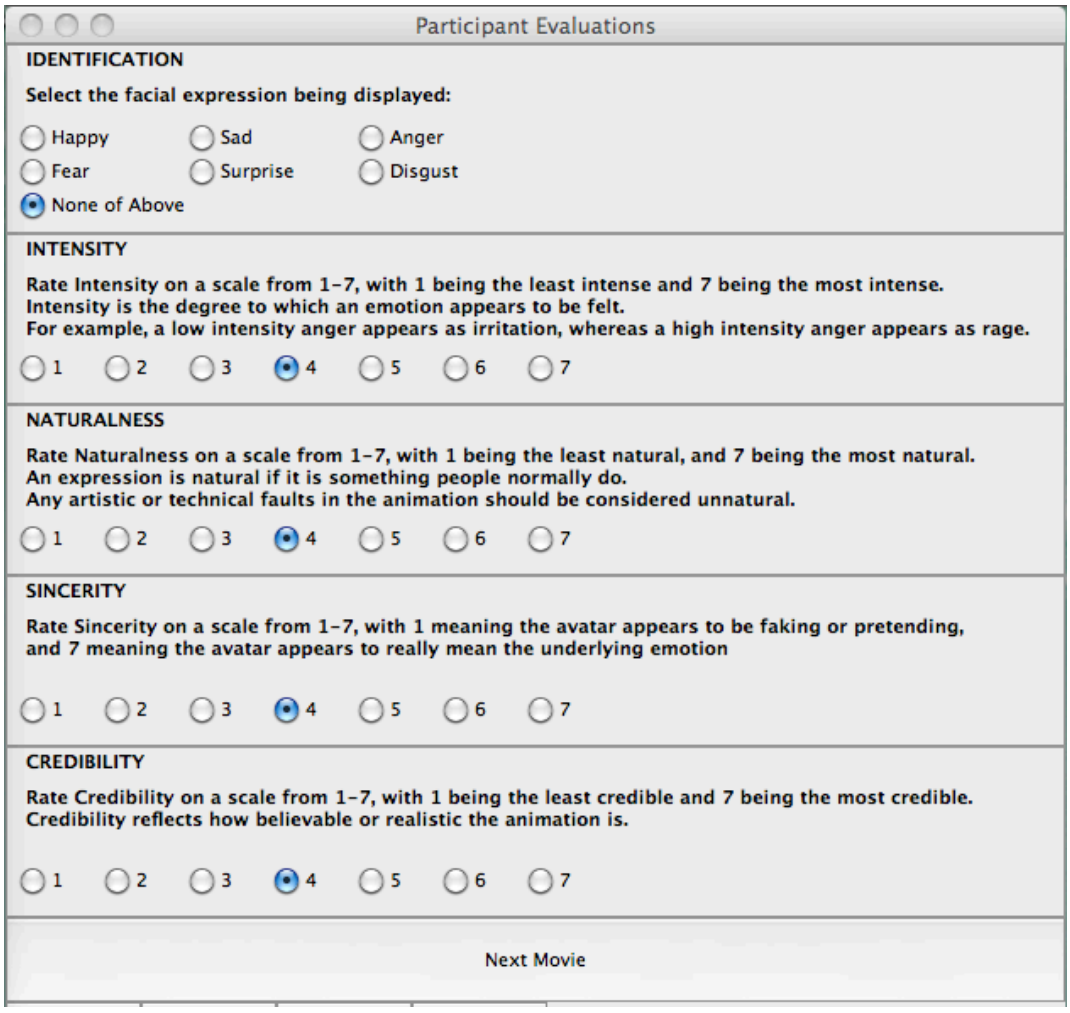
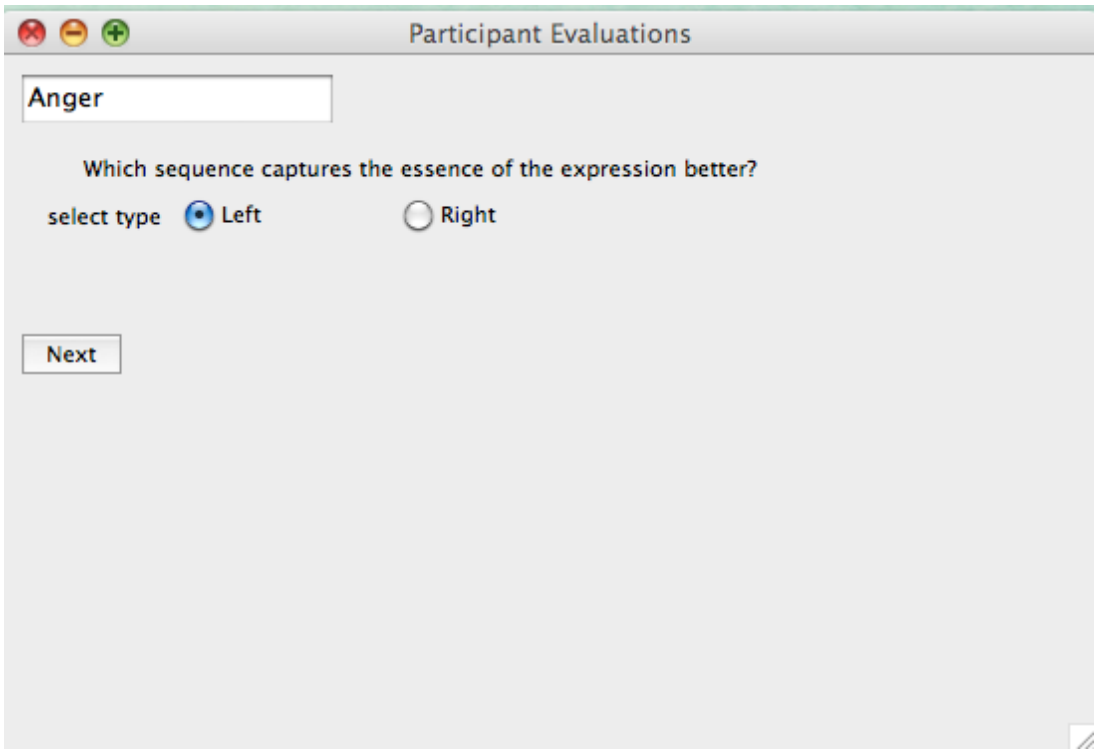


Figure 3: Participant Preferences Menu Screenshot



Neural Network Quality

The quality of the NN classification can be determined by the receiver operating characteristics (ROC). The ROC plot was generated from the confusion matrices and is used to compare classifiers during the design of the NN architecture.

The ROC curve is a plot of the true positive rate (TPR) versus the false positive rate (FPR) as the threshold varied. The most frequently used performance measure in ROC analysis is the area under the ROC curve, or AUC. The AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. A perfect fit would cluster the points in the upper-left corner.

To further clarify the ROC, consider a classifier that maps instances to one element of the set {positive, negative}. Given a classifier and an instance, there are four possible outcomes:

- If the instance is positive and it is classified as positive, it is a true positive.
- If the instance is positive and it is classified as negative, it is a false negative.
- If the instance is negative and it is classified as negative, it is a true negative.
- If the instance is negative and it is classified as positive, it is a false positive.

Metrics are commonly taken from an associated confusion matrix. Given a classifier and set of instances, a 2x2 confusion matrix is made using the counts of the correct and incorrect classifications of the instances. The diagonal represents the correct decisions, while the non-diagonal entries represent the errors between the classes. The true positive rate is then:

$$tp \text{ rate} \approx \text{true positives/ total positives}$$

and the false positive rate is:

$$fp \text{ rate} \approx \text{false positives/total negatives}$$

Points in ROC space are better the closer they are to the upper left corner. Classifiers on the left of the ROC graph may be thought of as conservative, in that they only make positive classifications with strong evidence. They make few false positives, but may miss many true positives as well. On the other hand, classifiers on the upper right side of an ROC graph may be thought of as liberal in that they make positive classifications on weak evidence so they classify nearly all the positives correctly but

have a high false positive rate. Any classifiers below the y=x line would be worse than random guessing (Fawcett, 2003).

The area under the ROC curve (AUC) has the statistical property that it is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. It is effective as a scalar value to compare the performance of two classifiers. The AUC is equal to the Wilcoxon statistic, which is used to test the hypothesis that the distribution of some variable (*x*) from one population (*p*) is equal to that of a second population (*n*).

$$H0 : x_p = x_n.$$

If this null hypothesis is rejected, we can calculate the probabilities $x_p > x_n$, $x_p < x_n$, or $x_p \neq x_n$. For a classifier, we want $p(x_p > x_n)$ to be as close to unity as possible. The AUC effectively measures $p(x_p > x_n)$ (Bradley, 1997).

Formats for Presenting Results

Genetic Algorithm Effectiveness

The effectiveness of an IGA is measured by its convergence rate and user satisfaction with the results. In each of our experiments, the IGA was run 10 times and the average number of convergences and average satisfaction rating given by the user is recorded. Tables containing the results of the experiments are provided. The summary data is also presented as a line graph.

The quality of the NN surrogate fitness function has a significant impact on the overall effectiveness of the IGA. The TPR and FPR data are reported in tables for each

of the six NNs. Also, the best and average surrogate fitness values for the final generation in an IGA run are presented along with the best and average subjective fitness values.

Genetic Algorithm Quality

Results of the participant questionnaires are presented as totals and averages in table form. The summary data is also presented as graphs.

Resources

The project was done on an Apple Macbook Pro running Mac OS X, version 10.6.4. This computer has a 2.66 GHz Intel Core i7 processor, with 4 GB of 1067 MHz DDR3 memory. The XCode development platform was used, along with the open source library FANN for the neural network. The graphics package that was used is Autodesk Maya 2010.

Eight participants were used to answer the questionnaires evaluating the animations.

Chapter 4

Results

Data Analysis

Genetic Algorithm Effectiveness

In this section, the experiments and the results obtained are described. In the first several experiments, one aspect of the IGA system was varied and the results were analyzed to find an optimal configuration. Then a set of experiments using this configuration was used in a set of ten runs for each emotion to obtain a final consistent set of data.

The first test of the IGA system enabled all of the AUs to be given random values during the population initiation. With this configuration, all of the animations presented were grossly exaggerated and unrealistic. It was clear that the number of AUs that are allowed to evolve must be limited.

The set of AUs associated with the target emotion were established by Matsumoto and Ekman (2008), and are listed in Table 5. The blendshape framework used for these experiments have a face model for each of the six basic emotions comprised of the set of AUs specific to that emotion. When the blendshape for a given emotion has a value of 0.5, all of the corresponding AUs are modified to display an intensity of 50%. The

remaining blendshapes correspond to single AUs, and include the individual AUs of the target emotion.

The next set of experiments restricted the AUs that could be given initial values to those associated with the target emotion, plus a limited number of random AUs. With each run of the GA, the subjective fitness values decreased. It was discovered that the initial set of NNs was giving a surrogate fitness value of 1 to every gene in the population. The surrogate fitness function was exhibiting no evolutionary pressure. In this case, the larger population may have been a disadvantage since the subjective fitness function represented a smaller percentage of the population. In these experiments, the IGA was not converging. It was theorized that the inability of the IGA to find an optimal area was most likely due to an inadequate surrogate fitness function. The NN input consisted of the subset of 17 AUs that have been shown to identify emotion. The AUs of a neutral expression have a value of 0. Since the AUs associated with the target emotion were initialized with random values, almost all of the genes were being assigned a surrogate fitness value of 1.

To improve the performance of the GA, the NN configuration was changed to accept all AUs as input, and the training sample sizes were increased by an order of magnitude. A large number of training samples were required to be representative of the entire face space. The average set of training samples for the six NNs had 88 samples when the input dimension was 17. The average set of training samples increased to 793 when the input dimension was increased to 39.

The next set of experiments varied the number of random AUs that were allowed to evolve. These were the first set of experiments with successful convergences, and were conducted using Happiness as the target. Convergence is defined as the point where a sufficient fitness level has been achieved and is manually indicated by the user. At this point, the individual with the highest subjective fitness value is saved for later evaluation.

The initial experiment set the number of random AUs to 7. Convergence did not occur within the ten trials that were conducted. When the number of random AUs was set to 5, convergence occurred in 5 out of 10 runs. When the number of random AUs was lowered to 3, convergence occurred in 3 out of 10 runs.

Due to research suggesting that 40% of the population needs to be evaluated with the more precise fitness function (Jin, Olhofer, and Sendhoff, 2001), an experiment was conducted using a population of 20, with 8 being evaluated subjectively. When the number of random AUs was set to 5, convergence occurred in 3 out of 10 runs. When the number of random AUs was lowered to 3, convergence also occurred in 3 out of 10 runs.

The set of experiments using the Happiness target clearly revealed that the number of random AUs that were allowed to evolve was critical to the success of the IGA. When the number of random AUs was too high (i.e. 7), none of the runs converged. When the number of random AUs was too low (i.e. 3), the variations were often barely noticeable. This is reflected in the fact that the average user satisfaction using 3 random AUs was significantly lower than when using 5 random AUs. It is possible that the optimal number of random AUs may be different for each target emotion.

Table 9: User Statistics: Happiness

Target: Happiness Population Size: Variable Number Random AUs: Variable Tournament Size: 4			
Population Size	Number Random AUs	Convergence Rate	Average User Satisfaction
20	5	30%	3.2
20	3	30%	2.8
100	7	0%	1.0
100	5	50%	3.8
100	3	30%	2.9

The next set of trials was conducted with the target Sadness. The first pair of trials varied the population size. Convergence rates of 70% and 50% were obtained using population sizes of 20 and 100, respectively.

It was observed that the initial eight phenotypes presented usually resulted in several highly fit chromosomes. But even when the IGA converged successfully, the majority of the phenotypes presented for evaluation had a very low subjective fitness. Moreover, with each successive generation, the highly fit chromosomes devolved into lower fit chromosomes.

The next set of experiments varied tournament size, based on the work of Goldberg and Deb (1991), which showed that an increase in tournament size would decrease the convergence time.

When the tournament size was increased to 8, the fittest chromosomes quickly spread through the population. This was evident by the appearance of chromosomes with

a high subjective fitness value appearing multiple times in the phenotypes. The undesirable chromosomes appeared to be weeded out more quickly. It only took a few runs to determine if the population was improving. As generations progressed, the number of phenotypes that were given high subjective ratings increased. Interestingly, multiple highly fit individuals often appeared more than once, supporting the claim by Muhlenbein (1989) that tournament selection inherently supports niching. The convergence rate improved from 50% for the runs with a tournament size of 4, to 80% for the runs with a tournament size of 8. The average subjective fitness rating across the successful convergences was 3.6 with a tournament size of 4, and 4.5 with a tournament size of 8 (see Appendix B).

Table 10: User Statistics: Sadness

Target: Sadness Population Size: Variable Number Random AUs: 5 Tournament Size: Variable			
Population Size	Tournament Size	Convergence Rate	Average User Satisfaction
20	4	70%	5.7
100	4	50%	4.9
100	8	80%	6.6

The next set of trials was conducted on the target Anger. In three sets of ten runs, the tournament size was varied for each set. With a tournament size of 8, the most fit candidates quickly dominated the population as was seen with the sadness target. Successful convergence was reached in each of the 10 runs.

With a tournament size of 6, the quality of the initial population was more important. It is noteworthy that if there were only one highly fit phenotype presented, it was less likely that the good genes would multiply. Instead, the good chromosomes were more likely to be corrupted by the less fit chromosomes. In general, the initial eight chromosomes needed to present at least 2 or 3 highly fit chromosomes for the highly fit chromosomes to spread throughout the population.

With a tournament size of 4, the highly fit chromosomes did not take over the population. Indeed, the eight highest fit chromosomes usually did not increase their level of fitness significantly.

Table 11: User Statistics: Anger

Target: Anger Population Size: 100 Number Random AUs: 5 Tournament Size: Variable		
Tournament Size	Convergence Rate	Average User Satisfaction
8	100%	7.0
6	60%	5.0
4	70%	5.3

The next set of experiments again varied population size, this time using a tournament size of 8 with the target disgust. The larger population of 100 resulted in a convergence rate of 80%, and the smaller population resulted in a convergence rate of 40%.

Table 12: User Statistics: Disgust

Target: Disgust Population Size: 20 Number Random AUs: 5 Tournament Size: 8		
Population Size	Convergence Rate	Average User Satisfaction
100	80%	6.70
20	40%	4.25

The initial trials for the targets fear and surprise used the most successful parameters that had been discovered in the previous experiments. The population was 100, the number of random AUs was 5, the crossover rate was .8, and the tournament size was 8. The experiments produced unacceptably low convergence rates. An analysis revealed that the NNs for these two emotions were producing near zero surrogate fitness values. This failure rate was addressed by attempting to improve the surrogate fitness function. The NNs for these three target emotions were trained with significantly more training samples, but the surrogate fitness values remained invalid and convergence rates remained low. Finally, an error was discovered that was corrupting the training samples for these three NNs. Once the error was corrected, the IGA was performing well for these target emotions.

To obtain a good consistent set of data across all six target emotions, a set of experiments was conducted using the most successful parameters that had been discovered. Further discussion of IGA results throughout this paper will refer to this final consistent set unless otherwise indicated. The IGA was run ten times for each

emotion for a total of 60 runs. The average convergence rate was 85%, ranging from 70% to 100%. In every run, the user ended the IGA within 5 generations, both for the success and failure convergence cases. In the cases of anger and fear, the IGA converged ten out of ten times. Details of the individual runs can be found in Appendix A.

Table 13: User Statistics: Final Consistent Set of Experiments

Target: Variable Population Size: 100 Number Random AUs: 5 Tournament Size: 8		
Target Emotion	Convergence Rate	Average User Satisfaction
Happy	70%	5.6
Sadness	70%	6.0
Anger	100%	7.0
Fear	100%	8.5
Surprise	90%	6.1
Disgust	80%	6.7

Surrogate and Subjective Fitness Analysis

Data was also collected to compare the two fitness functions. The average subjective fitness was calculated using the subjective fitness values for the eight chromosomes whose phenotypes were presented to the user for subjective evaluation. The average surrogate fitness was calculated over the entire population of chromosomes after the user ended the IGA run, whether convergence was successful or not. Results of

each run are included in Appendix B. The following discussion uses the averages of the ten runs contained in each table of Appendix B.

For the seven successful convergence cases for happiness, the best subjective fitness values averaged 6.14 across the final ten runs, ranging from 6 to 9. The corresponding best surrogate fitness values averaged 5.00. For the three unsuccessful runs, the best subjective fitness values averaged 4.00, and the corresponding surrogate fitness values averaged 4.98. The average surrogate fitness values across the entire population averaged 4.25 for the successful runs, and 4.87 for the unsuccessful runs. For the happiness case, there was no correlation between the surrogate fitness values and the likelihood of convergence.

For the seven successful convergence cases for sadness, the best subjective fitness value averaged 8.43 across the final ten runs, ranging from 7 to 9. The corresponding best surrogate fitness values averaged 4.89. For the unsuccessful runs, the best subjective fitness values averaged 3.00, and the corresponding surrogate fitness values averaged 4.18. The average surrogate fitness values across the entire population averaged 4.72 for the successful runs, and 3.94 for the unsuccessful runs. For the sadness case, there was support for correlation between the surrogate fitness values and convergence rate.

Anger had ten successful convergence cases. The best subjective fitness value averaged 7.80 across the final ten runs, ranging from 7 to 9. The corresponding best surrogate fitness values averaged 4.91. The average surrogate fitness values across the entire population averaged 4.75.

Fear also had ten successful convergence cases. The best subjective fitness value averaged 8.90 across the final ten runs, ranging from 7 to 10. The corresponding best surrogate fitness values averaged 4.96. The average surrogate fitness values across the entire population averaged 4.73.

For the nine successful convergence run for surprise, the best subjective fitness value averaged 7.33 across the final ten runs, ranging from 7 to 9. The corresponding best surrogate fitness values averaged 4.91. For the unsuccessful run, the best subjective fitness value was 2.00, and the corresponding surrogate fitness value was 4.94. The average surrogate fitness values across the entire population averaged 4.67 for the successful runs, and 4.84 for the unsuccessful runs. There was no correlation between the surrogate fitness values and the likelihood of convergence.

For the eight successful convergence cases for disgust, the best subjective fitness value averaged 8.25 across the final ten runs, ranging from 6 to 10. The two corresponding best surrogate fitness values averaged 4.35. For the unsuccessful runs, the best subjective fitness values averaged 4.00, and the corresponding surrogate fitness values averaged 4.24. The average surrogate fitness values across the entire population averaged 4.03 for the both the successful and unsuccessful runs. There was no correlation between the surrogate fitness values and the likelihood of convergence.

Table 14: Subjective and Surrogate Fitness Values

Comparison of Subjective and Surrogate Fitness Values Average over ten runs for each target emotion					
Target Emotion	Best Subjective Fitness	Average Subjective Fitness	Best Surrogate Fitness	Average Surrogate Fitness	Converged
Happiness	6.14	3.42	5.00	4.25	Yes
Happiness	4.00	3.12	4.98	4.87	No
Sadness	8.43	5.14	4.89	4.72	Yes
Sadness	3.00	1.50	4.18	3.94	No
Anger	7.80	5.20	4.91	4.75	Yes
Anger	-	-	-	-	No
Fear	8.90	6.00	4.96	4.73	Yes
Fear	-	-	-	-	No
Surprise	7.33	4.00	4.91	4.67	Yes
Surprise	2.00	1.12	4.94	4.84	No
Disgust	8.25	4.25	4.35	4.03	Yes
Disgust	4.00	1.69	4.24	4.03	No

Neural Network Classification

The final set of happiness training samples consisted of 1001 training samples and 250 testing samples. This neural network was trained until the mean square error (mse) reached .001. In the final neural network trained, the testing samples resulted in a true positive rate of 0.940 and a false positive rate of 0.164

Note that the true positive rate (TPR) is the fraction of true positives out of the positives, and the false positive rate (FPR) is the fraction of false positives out of the negatives.

Table 15: Neural Network: Happiness

Neural Network: Happiness	
Number Training Samples	1001
Number Testing Samples	250
False Positive	30
False Negative	4
True Negative	153
True Positive	63
True Positive Rate	0.940
False Positive Rate	0.164
Mean Square Error	0.001

The final set of sadness training samples consisted of 930 training samples and 232 testing samples. This neural network was also trained until the mse reached .001. In the final neural network trained, the testing samples resulted in a true positive rate of 0.880 and a false positive rate of 0.010.

Table 16: Neural Network: Sadness

Neural Network: Sadness	
Number Training Samples	930
Number Testing Samples	232
False Positive	2
False Negative	3
True Negative	205
True Positive	22
True Positive Rate	0.880
False Positive Rate	0.010
Mean Square Error	0.001

The final set of anger training samples consisted of 915 training samples and 228 testing samples. This neural network was trained until the mse reached .001. In the

final neural network trained, the testing samples resulted in a true positive rate of 0.923 and a false positive rate of 0.015.

Table 17: Neural Network: Anger

Neural Network: Anger	
Number Training Samples	915
Number Testing Samples	228
False Positive	3
False Negative	2
True Negative	199
True Positive	24
True Positive Rate	0.923
False Positive Rate	0.015
Mean Square Error	0.001

The final set of fear training samples consisted of 1005 training samples and 250 testing samples. This neural network was trained until the mse reached .001. In the final neural network trained, the testing samples resulted in a true positive rate of 0.903 and a false positive rate of 0.037.

Table 18: Neural Network: Fear

Neural Network: Fear	
Number Training Samples	1005
Number Testing Samples	250
False Positive	8
False Negative	3
True Negative	211
True Positive	28
True Positive Rate	0.903
False Positive Rate	0.037
Mean Square Error	0.001

The final set of surprise training samples consisted of 994 training samples and 248 testing samples. This neural network was trained until the mse reached .001. In the final neural network trained, the testing samples resulted in a true positive rate of 0.917 and a false positive rate of 0.004.

Table 19: Neural Network: Surprise

Neural Network: Surprise	
Number Training Samples	994
Number Testing Samples	248
False Positive	1
False Negative	2
True Negative	223
True Positive	22
True Positive Rate	0.917
False Positive Rate	0.004
Mean Square Error	0.001

The final set of disgust training samples consisted of 754 training samples and 188 testing samples. In the final neural network trained, the testing samples resulted in a true positive rate of 0.778 and a false positive rate of 0.

Table 20: Neural Network: Disgust

Neural Network: Disgust	
Number Training Samples	754
Number Testing Samples	188
False Positive	0
False Negative	2
True Negative	179
True Positive	7
True Positive Rate	0.778
False Positive Rate	0.00
Mean Square Error	0.001

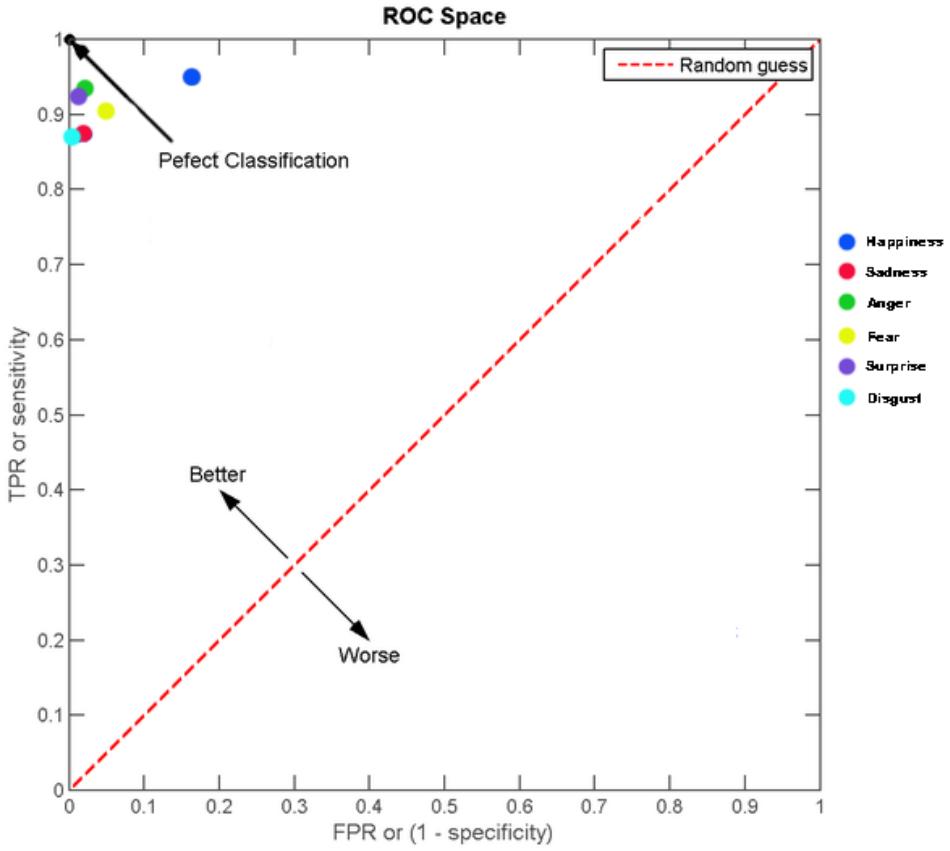
The true positive rate across all six NNs ranged from 0.778 to 0.940. The happiness NN, which had the poorest false positive rate, also produced one of the lowest convergence rates. It is notable that the happiness NN had a significantly higher percentage of positive training samples than the other NNs.

Table 21: Neural Networks: True Positive Rate/False Positive Rate

Target Emotion	True Positive Rate	False Positive Rate	Total Testing Samples	Percent Positive Samples	IGA Convergence Rate
Happiness	0.940	0.164	250	37%	70%
Sadness	0.880	0.010	232	10%	70%
Anger	0.923	0.015	228	12%	100%
Fear	0.903	0.037	250	14%	100%
Surprise	0.917	0.004	248	9%	90%
Disgust	0.778	0.000	188	4%	80%

The following graph shows the trade-offs between TPR and FPR in ROC space. Points closest to the upper left corner should have the best predictive power. The primary outlier is the happiness NN which had a relatively high FPR and resulted in one of the lowest convergence rates. The next points farthest from the upper left corner are from the sadness and disgust NNs, which also had the next two lowest convergence rates. The three best points in ROC space were the anger, surprise, and fear NNs, which also had the three best convergence rates. This suggests there may be a correlation between the quality of the NN and the convergence rate of the IGA.

Figure 4: ROC Plots for Six NNs



Genetic Algorithm Quality

The quality of the IGA system was measured with questionnaires administered to eight participants. Individual results from each participant are included in Appendix C. The results in this section use the averages from the individual questionnaires.

The first question the participants were asked was to identify the facial expression being displayed. The identification results varied widely across the emotions. Happiness

was identified 13 out of 16 times for both the IGA and manual animations. Fear, on the other hand, was only identified 6 and 7 times for the IGA and the manual animations, respectively. The difference between the number of correct identifications for the IGA and manual animations was negligible for 4 of the 6 emotions, but the manual animations had a significant lead in the cases of anger and sadness. The manual anger animations were identified correctly 56% of the time compared with 38% for the IGA animations. The manual sadness animations were identified correctly 38% of the time, compared with 69% for the IGA animations. Overall, when the numbers for all the emotions are combined, the IGA animations were correctly identified 53% of the time, and the manual animations were correctly identified 60% of the time.

There were many patterns in the misidentifications. For example, anger and disgust were often identified as each other. Fear was often misidentified as surprise. There were also patterns in individual responses. For example, participant 3 tended to see surprise, which comprised 47% of her misclassifications.

Table 22: Participant Responses: Incorrect Identifications per participant

Emotion	1	2	3	4	5	6	7	8
Anger	Disgust Disgust	Fear Happy Disgust	Surprise	Disgust Disgust	Happy	Fear None	Happy Happy Fear	Disgust
Disgust	Anger Sadness	Surprise Anger Anger	Surprise Anger Anger	Anger Happy	Anger	Sadness None Surprise	None Fear Fear	Fear Surprise
Fear	Sadness Sadness	Happy Surprise	Surprise Surprise Surprise Happy	Surprise Surprise		None None Surprise	Sadness Anger Sadness	None Surprise Anger
Happiness		None Surprise Sadness	Anger Surprise Surprise					
Sadness		Fear	Anger Happy	Happy Surprise	None Fear None		Surprise Disgust Happy	None None Fear Fear
Surprise		Happy	Anger Happy		Disgust	Happy Sadness		None Fear

The next question the participants were asked was to rate the intensity of the emotion, with 1 being the least intense and 7 being the most intense. Intensity was defined as the degree to which an emotion appears to be felt. For example, a low intensity anger appears as irritation, whereas a high intensity anger appears as rage. The average per emotion for the IGA animations ranged from 4.38 to 5.38. For the manual animations, the rating for intensity ranged from 3.62 to 4.56. In every case, the rating for the IGA animation was higher than for the corresponding manual animation. In all but one case, the difference was less than 1 point. The highest difference was the case of fear, which had a difference of 1.44. When all emotions are taken together, the total

average intensity rating for the IGA animations was 4.84, and the average rating for the manual animations was 3.93.

The participants were next asked to rate the naturalness of the expression. It was explained that an expression is natural if it is something people normally do. Also, any artistic or technical faults in the animations should be considered unnatural. In every emotion except sadness, the rating was higher for the IGA animations than the manual animations. The difference was less than 1 point in every case. Overall, the average rating for the IGA animations was 4.75, and the average for the manual animations was 4.54.

The next rating that was requested was for sincerity. This was described as a rating of 1 meaning the avatar appears to be faking or pretending, and 7 meaning the avatar appears to really mean the underlying emotion. In 4 of the emotions, the IGA animations had a higher average rating, and in 2 of the emotions, the manual animations had the higher average rating. In all but one emotion, the difference in average rating was less than 1 point. Fear had the highest difference, with 5.44 and 4.06 for the IGA and manual animations, respectively. When all emotions are totaled together, it shows that the overall average for the IGA animations was 4.98, and the average for the manual animations was 4.60.

Lastly, the participants were asked to rate the credibility of the animations. Credibility was defined as how believable or realistic the animation is. For anger and disgust, the average ratings were identical for the IGA and manual animations. Fear, happiness, and surprise had higher average rating for the IGA animations. Sadness was

the only emotion where the manual animations were rated as being more credible. When the totals for all emotions are combined, the overall average is 4.84 (69%) and 4.58 (65%) for the IGA and manual animations, respectively.

Table 23: Participant Responses: Totals per Emotion

Total Participant Responses						
Category: 16 Presented	Anger	Disgust	Fear	Happiness	Sadness	Surprise
IGA: Identification	6	7	6	13	6	13
Manual: Identification	9	7	7	13	11	11
IGA: Intensity	86	75	75	82	70	77
Manual: Intensity	73	65	52	71	58	58
IGA: Naturalness	75	77	80	76	63	85
Manual: Naturalness	72	73	68	66	76	81
IGA: Sincerity	77	76	87	80	68	90
Manual: Sincerity	73	78	65	68	76	82
IGA: Credibility	78	75	83	76	66	87
Manual: Credibility	78	75	66	70	69	82

Table 24: Participant Responses: Averages per Emotion

Total Participant Averages						
Category	Anger	Disgust	Fear	Happiness	Sadness	Surprise
IGA: Identification	0.38	0.44	0.38	0.81	0.38	0.81
Manual: Identification	0.56	0.44	0.44	0.81	0.69	0.69
IGA: Intensity	5.38	4.69	4.69	5.12	4.38	4.81
Manual: Intensity	4.56	4.06	3.25	4.44	3.62	3.62
IGA: Naturalness	4.69	4.81	5.00	4.75	3.94	5.31
Manual: Naturalness	4.50	4.56	4.25	4.12	4.75	5.06
IGA: Sincerity	4.81	4.75	5.44	5.00	4.25	5.62
Manual: Sincerity	4.56	4.88	4.06	4.25	4.75	5.12
IGA: Credibility	4.88	4.69	5.19	4.75	4.12	5.44
Manual: Credibility	4.88	4.69	4.12	4.38	4.31	5.12

Table 25: Participant Responses: Totals of all emotions

Total Participant Responses	
Category: 96 Presented	Totals
IGA: Identification	51
Manual: Identification	58
IGA: Intensity	465
Manual: Intensity	377
IGA: Naturalness	456
Manual: Naturalness	436
IGA: Sincerity	478
Manual: Sincerity	442
IGA: Credibility	465
Manual: Credibility	440

Table 26: Participant Responses: Averages of all emotions

Total Participant Averages	
Category: 96 Presented	Totals
IGA: Identification	0.53
Manual: Identification	0.60
IGA: Intensity	4.84
Manual: Intensity	3.93
IGA: Naturalness	4.75
Manual: Naturalness	4.54
IGA: Sincerity	4.98
Manual: Sincerity	4.60
IGA: Credibility	4.84
Manual: Credibility	4.58

The participants were also shown an IGA animation and manual animation side by side and asked which one they preferred. The IGA animation was randomly placed on the right or left to avoid any positional bias. The participants knew the emotion being displayed but did not know which animation was generated by the IGA. The preference results were quite skewed. In the case of anger and disgust, the manual animations had a

significant preference. In the case of fear, happiness, and surprise, the IGA animations had a significant preference. In the case of sadness, the IGA animations had a slight preference.

When the totals are averaged across all of the emotions, the preference rates are close, with 54% of the IGA animations being preferred, and 46% of the manual animations being preferred.

Table 27: Participant Responses: Preferences Totals per Emotion

IGA/Manual	Emotion	Preferred Out of 16 Total	Average
IGA	Anger	6	0.38
Manual	Anger	10	0.62
IGA	Disgust	4	0.25
Manual	Disgust	12	0.75
IGA	Fear	12	0.75
Manual	Fear	4	0.25
IGA	Happiness	10	0.62
Manual	Happiness	6	0.38
IGA	Sadness	9	0.56
Manual	Sadness	7	0.44
IGA	Surprise	11	0.69
Manual	Surprise	6	0.31

Table 28: Participant Responses: Preference Totals of all Emotions

IGA/Manual	Preferred Out of 96 Total	Average
IGA	52	0.54
Manual	46	0.46

Findings

In all of the experiments, the most significant factor in the successful convergence of the IGA was the quality of the first eight phenotypes presented for evaluation. If there were at least two or three good phenotypes, it was likely that the overall quality of presented phenotypes would improve. If there were no good phenotypes presented, the subjective fitness values offered no evolutionary pressure and the IGA was unlikely to present a good phenotype.

It was discovered early in the testing phase that the surrogate fitness function had a significant effect on the success of the IGA. The second set of experiments were not converging due to the NN always assigning a surrogate fitness value of 1. It was reinforced later when corrupted data prevented the IGA from converging due to the NN always assigning a surrogate fitness value near 0. In both cases, when the NN was fixed, the IGA was able to converge.

Two sets of experiments were performed with the user evaluating 40% of the population, as described by Jin, Olhofer, and Sendhoff (2001). In these experiments, which used a small population size of 20, the convergence rate was lower than when using a population of 100.

The final consistent set data was generated from six experiments of ten trials each. The average convergence rate was 85%, ranging from 70% to 100%. The average user satisfaction ranged from 5.6 to 8.5. There was a strong correlation between user satisfaction and convergence rate. Happiness and sadness had both the lowest

convergence rates and the lowest user satisfaction rates. Anger and fear had both the highest convergence rates and highest user satisfaction rates.

In a comparison of the NNs of the six target emotions, the NNs with the best TPR/FPR points in ROC space corresponded to the IGAs with the highest convergence rates. Similarly, the NNs with the ROC points farthest from the upper left corner corresponded to the IGAs with the lowest convergence rates.

The results of the participant questionnaires show that the manual and IGA animations compare favorably. In fact, the IGA animations had higher ratings on average than the manual animations in every category except identification. The credibility ratings for the IGA animations were the same or higher than those for the manual animations for every emotion. The preference ratings for the IGA animations were higher in 4 of the 6 emotions.

Summary of Results

Genetic Algorithm Effectiveness

A set of experiments was conducted, systematically changing one variable and performing 10 runs of the IGA. The results of each run were recorded in the form of whether the IGA converged and how satisfied the user was with the result. The results of each run are included in Appendix A.

Additionally, the best and average values of the subjective and surrogate fitness function were recorded. The results of each run are included in Appendix B.

The first experiment allowed all 39 AUs to be initialized and evolve, resulting in unrealistic phenotypes throughout the population.

The next set of experiments limited the number of AUs that could evolve, but the surrogate fitness values were 1 throughout the population. The NNs were changed to accept all AUs as input instead of the subset of AUs associated with emotion.

Three sets of experiments were conducted varying the number of random AUs for happiness. When the number of random AUs was 7, no convergences occurred. When the number of random AUs was 3, the variations were often barely visible. In two sets of experiments using 3 and 5 random AUs, the results were (30%, 30%), (30%, 50%) where the first percentile listed reflects the experiment with 3 random AUs.

Three sets of experiments were done varying the population size for happiness and sadness. The remaining variables were kept the same for each pair of experiments, but differed among the three sets. The convergence results of the experiments were (30%, 30%), (30%, 50%), (40%, 80%), where the first number was for a population size 20, and the second number was for a population size 100.

Two sets of experiments were conducted varying the tournament size for sadness and anger. Using a tournament size of 4 and 8, the results were (50%, 80%) in the first experiment. Using a tournament size of 4, 6, and 8, the results were (70%, 60%, and 100%).

Fear and surprise were not converging at this point due to an error in the NN training data. When the surrogate fitness value was 0 throughout the population, the IGA did not converge.

The best convergence rates were found using a population of 100, number of random AUs of 5, and tournament size of 8. Using these parameters, another set of six

experiments was conducted for a final consistent set of values. The only variable for this final set of 60 runs was the target emotion and the quality of its associated neural network. Convergence rates averaged 85%, ranging from 70% to 100%. User satisfaction ranged from 5.6 to 8.5 out of 10. There was a strong correlation between the convergence rate and average user satisfaction.

In 5 of the 6 trials in the final set of experiments, there was no correlation between surrogate fitness values and subjective fitness values or convergence rate. However, in the absence of a valid surrogate fitness function, the IGA failed to converge.

The TPR of the six NNs ranged from 0.778 to 0.940, and the FPR ranged from 0.00 to 0.164. There was some evidence suggesting a correlation between the TPR/FPR points in ROC space and IGA convergence rates. The number of training samples ranged from 754 to 1005.

Table 29: Summary of IGA Experiments

Summary of Experimental Results (Variables Shaded Grey)					
Emotion	Converge Rate	Avg User Satis.	Pop. Size	Number Random AUs	Tourn. Size
Happiness	30%	2.6	20	5	4
Happiness	30%	2.8	20	3	4
Happiness	0%	1.0	100	7	4
Happiness	50%	3.8	100	5	4
Happiness	30%	2.9	100	3	4
Sadness	70%	5.7	20	5	4
Sadness	50%	4.9	100	5	4
Sadness	80%	6.6	100	5	8
Anger	70%	5.3	100	5	4
Anger	60%	5.0	100	5	6
Anger	100%	7.0	100	5	8
Disgust	40%	7.25	20	5	8
Disgust	80%	6.7	100	5	8
Happiness	70%	5.6	100	5	8
Sadness	70%	6.0	100	5	8
Anger	100%	7.0	100	5	8
Disgust	80%	6.7	100	5	8
Fear	100%	8.5	100	5	8
Surprise	90%	6.1	100	5	8

Genetic Algorithm Quality

It has been shown that experimental methods from the perceptual sciences allow defining and measuring the perceived realism of computer-generated images (Wallraven, Breidt, Cunningham, and Bulthoff (2005). Specifically, the analysis of questionnaires administered to participants provides a quantitative method of measuring subjective data. The data show that identification rates were 53% for the IGA animations and 60% for the manual animations. The intensity, naturalness, sincerity, and credibility ratings were all slightly higher for the IGA animations than the manual animations overall. The credibility ratings for the IGA animations were the same or higher than the manual animations for every emotion. The credibility average was 69% for the IGA animations and 65% for the manual animations. The results from the preference questionnaire showed a preference for the manual animations for anger and disgust, and a preference for the IGA animations for the remaining four emotions. The preference average over all the emotions was 54% for the IGA animations and 46% for the manual animations.

Table 30: Participant Responses: Averages for all Emotions

Total Participant Averages	
Category: 96 Presented	Totals
IGA: Identification	0.53
Manual: Identification	0.60
IGA: Intensity	4.84
Manual: Intensity	3.93
IGA: Naturalness	4.75
Manual: Naturalness	4.54
IGA: Sincerity	4.98
Manual: Sincerity	4.60
IGA: Credibility	4.84
Manual: Credibility	4.58

Chapter 5

Conclusions, Implications, Recommendations, and Summary

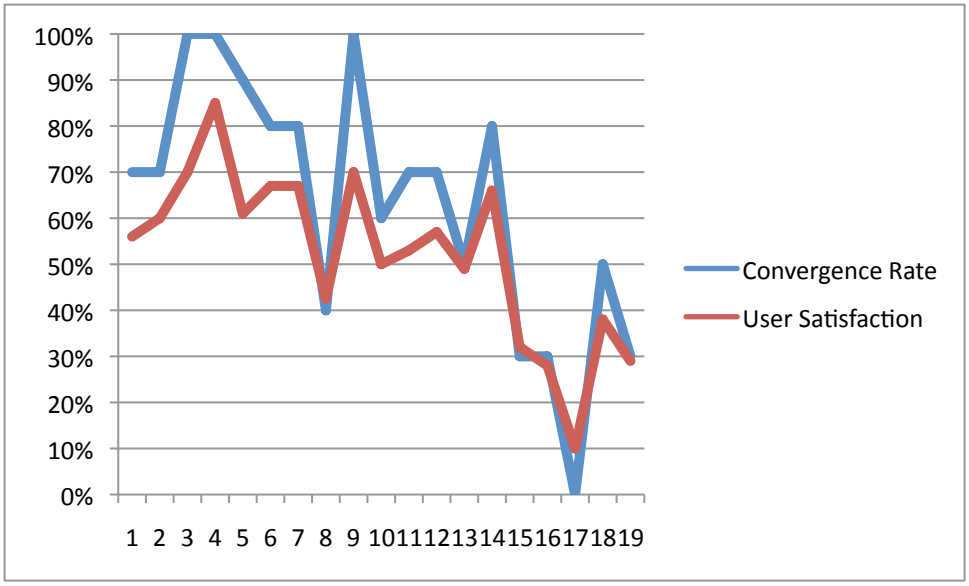
Conclusions

Our research set out to determine whether an IGA could effectively generate realistic variations in facial expressions. Realistic facial animations are comprised of both a random component and a predictable component to expressions. The roles of speaker and listener are the central focus of facial animation research. The projects that incorporate emotion generally rely on static emotional expressions, the predictable component of expressions. These components have been identified by FACS and studied extensively. Our research produced random variations by varying randomly selected AUs that are not associated with either the roles of speaker and listener or with the AUs identified with a specific emotion. The quality of the IGA system was measured by user satisfaction, percent of successful convergences, and evaluation by participants.

There was a strong correlation with user satisfaction and convergence rate. The graph below plots the convergence rate and user satisfaction. It shows the results of 19 experiments, with each experiment comprising ten runs. The first six points in the graph show the final consistent set of experiments. The user satisfaction rating reflected how

satisfied the user was with the best phenotype on a per-run basis. User satisfaction ranged from 5.6 to 8.5 on a scale of 1 to 10 in the final consistent set of experiments, and the convergence rate ranged from 70% to 100%. The quality of the IGA system from the perspective of the user was at least above average and was often quite favorable.

Figure 5: User Satisfaction and Convergence Rate



Participants were asked to identify the presented emotion and rate the intensity, naturalness, sincerity, and credibility on a scale from 1 to 7. Participants were not aware of which animations were manually generated and which ones were generated by the IGA system. In general, the ratings for naturalness, sincerity, and credibility showed a strong correlation with one another. The ratings for the IGA animations were slightly higher overall. In fact, the credibility ratings for the IGA animations were the same or higher than the manual animations in every emotion except sadness, where the manual animations had a slight lead. Thus, from the perspective of the participants, the

animations created by the IGA system were just as credible as the manually created animations.

Figure 6: Participant Evaluation - Comparison of IGA animations

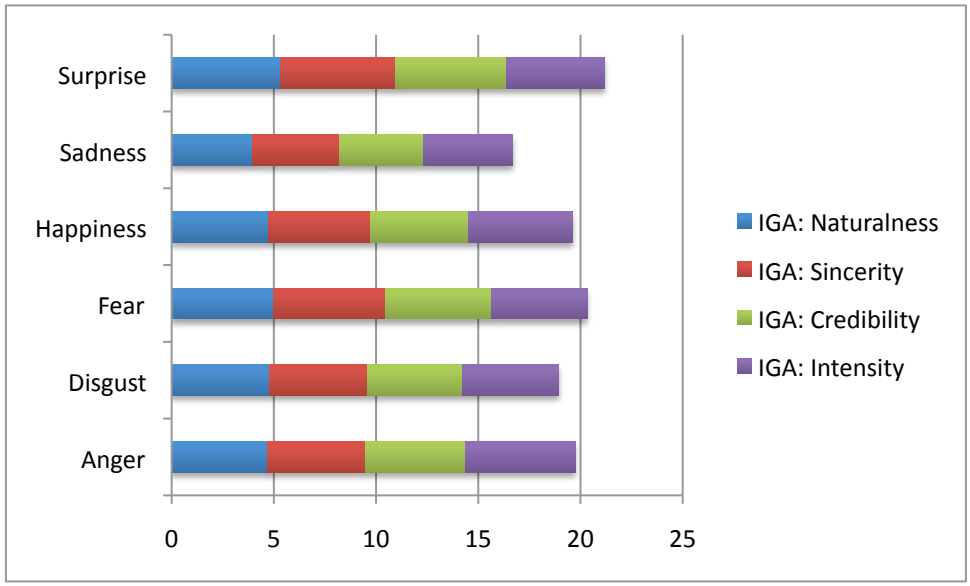


Figure 7: Participant Evaluation - Comparison of manual animations

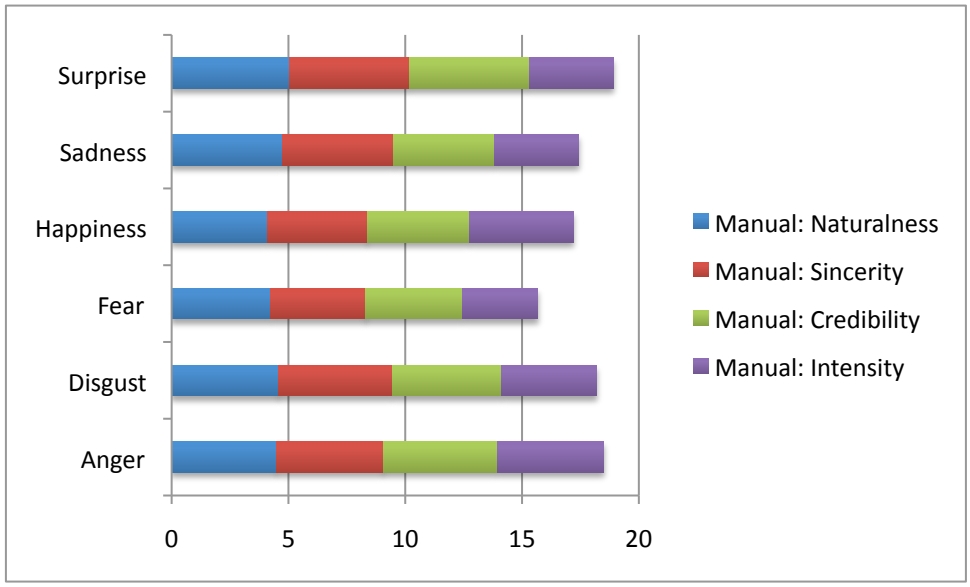
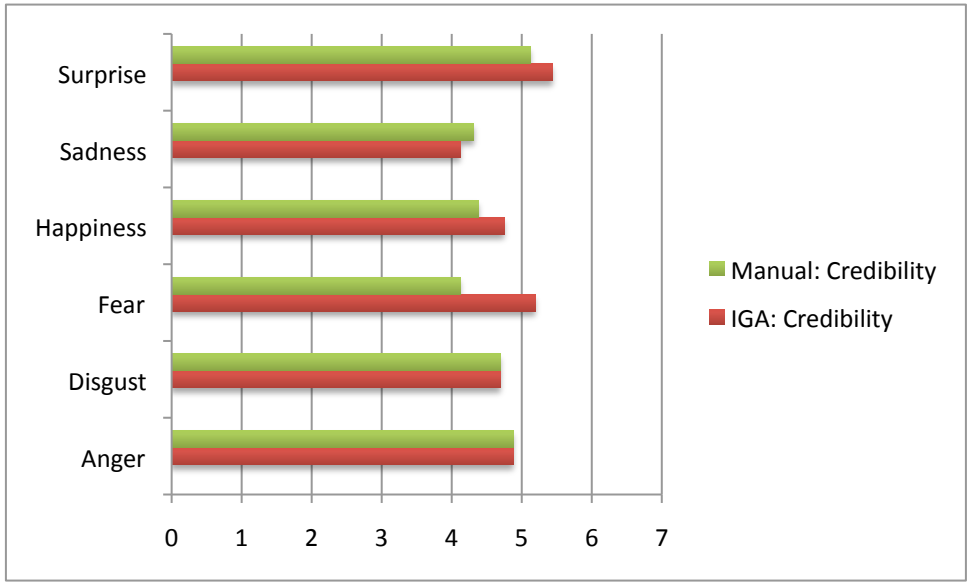
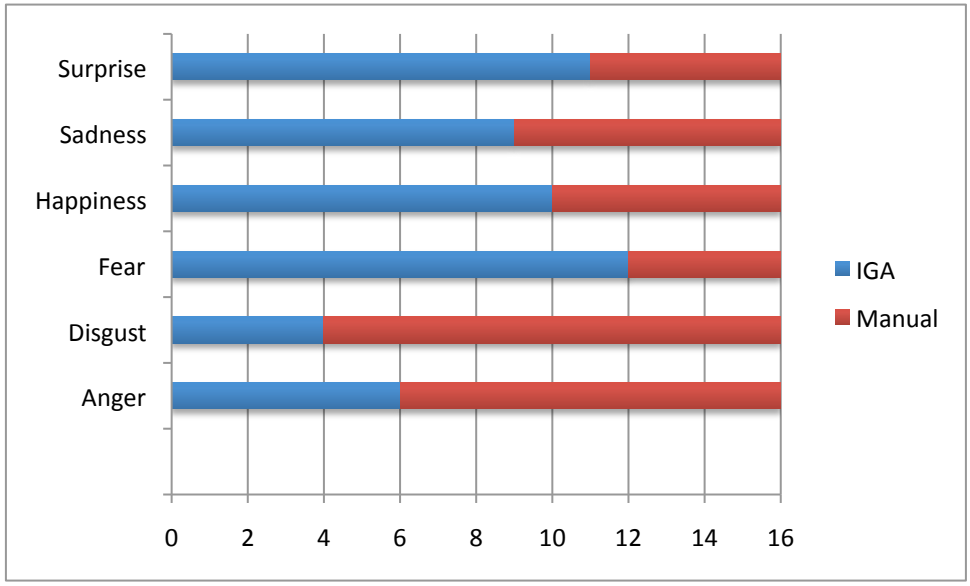


Figure 8: Participant Evaluation - Credibility Average Rating from all Participants



Participants were also presented with 12 animations that showed both an IGA and a manual animation, and asked which one they preferred. The participant did not know which one was manually generated and which one was generated by the IGA. The results were quite mixed, with the IGA animations being preferred in 4 of the 6 emotions. Overall, the IGA animations were chosen 54% of the time. This provides further support that the IGA system is able to evolve credible variations in facial expressions.

Figure 9: Participant Evaluation - IGA/Manual Preference



In addition to the results of the final consistent set of data, there were other interesting discoveries over the course of the project.

The NN training databases were initially populated with manually created face models over the course of several days. When it became apparent that many more samples were required, a mechanism was built into the IGA system so that the individual face models of any of the eight presented animations could be stepped through and added to either the positive or negative training sets for the target emotion. This transformed a task that would have taken many tedious hours into a task that only took minutes. Several 100's of samples could be added in less than 30 minutes. This illustrates the potential power of a system that can automatically generate creative content.

Another aspect of our research was to determine how well a neural network could serve as a surrogate fitness function for evaluating the quality of the chromosomes.

Other research (Tokui, 2000; Dahstedt, 2007) had used NNs in this capacity for evolving music, which is also very subjective. During the course of the experiments, it was clear that the NNs had a significant influence on quality of IGA. When the initial NNs were giving a value of 1 to all the genes, the IGA did not converge. Later when the NNs were giving a value of 0 to all the genes, the IGA did not converge. Over the course of ten runs in each of these failure scenarios the phenotypes presented for subjective evaluation were of very low quality. This suggests that the ability of the NN to select the 8 best individuals for subjective fitness evaluation was a crucial part of the IGA system.

The happiness, sadness, and disgust NNs had the lowest quality as measured by TPR and FPR, and had the corresponding lowest convergence rates for their IGAs. The anger, surprise, and fear NNs had the best TPR and FPR points in ROC space, and had the highest convergence rates for their IGAs. This suggests there is a correlation between the quality of the NN and the convergence rate of the IGA. It is also possible that the experiment samples of ten runs were too small to get a precise representation of the convergence rate. This is possible because the random AUs were different in each run. The actual AUs selected for random evolving had a significant effect on the ability of the IGA to converge. Some AUs were contraindicative of the target emotion, such as lowering the brows for surprise. Other combinations were difficult for any target, such as when both eye left and eye right were evolving.

One of the motivating factors behind using a surrogate fitness function was to increase the number of building blocks by using a large population. It has been shown that there must be a sufficient number of building blocks in the initial population to arrive

at an optimal solution (Harik et. al., 1999). Three experiments were conducted to compare the effects population size had on the IGA, showing a significant advantage for the larger population size. This suggests that the smaller population size did not provide a sufficient number of building blocks. It may also be that the smaller population GA performed poorly because the other GA parameters favored a larger population. It has been shown that the performance of a GA is influenced by a complex interaction of its parameters (Grefenstette, 1986).

One of our research questions was how to incorporate the two fitness functions. In our research, the chromosomes retained both of the fitness functions. If a chromosome were carried over into the next generation, it retained its subjective fitness value. If a new chromosome were created from crossover or mutation, its subjective fitness value was reset to zero. When selecting the mating population, a non-zero subjective fitness value was given priority over the surrogate fitness value. Biasing the selection is the most common method described in the literature, but it had been shown with computationally expensive fitness functions that 40% of the population needed to be evaluated with the more precise fitness function for optimal convergence (Jin, Olhofer, and Sendhoff, 2001). Our experiments yielded successful results evaluating only 8% of the population. This may be due to the fact that a subjective fitness function represents an area rather than a point in the solution space. It may also be that our solutions represented local optima rather than global optima.

One of the interesting sets of experiments varied the tournament size. When the tournament size was increased, the most fit chromosomes quickly spread throughout the

population and convergence results improved. Due to issues of user fatigue, it was important for the user to perceive improved phenotypes within the first few generations. In the case of an IGA with a surrogate fitness function and a large population, it appears that the emphasis is better placed on exploitation rather than exploration. It is possible that a more accurate surrogate fitness function would lessen this effect.

Implications

IGAs have been used to generate static 2-D and 3-D facial expressions. They have also been used to create animated art, and animated arms and legs. Our research has shown that it is possible to evolve credible variations in 3-D facial expressions for the six basic emotions. Due to the extensive manual intervention required to produce animations in current practice, an automated system of generating creative content could be very useful. As the speed and power of computer power improves, the demand for 3-D animations is likely to increase.

Once an IGA system has been constructed, it can produce variations in expressions in a fraction of the time that an individual can produce manually. Our research has shown that the participants liked the IGA animations as least as well as the manual animations and found them just as credible. One can imagine how an IGA system could be improved to incorporate speech sequences and other types of expressions and tied into a production workflow to enhance productivity.

In our experiments, a large population size yielded significantly better results than a smaller one. This is in contrast to the overwhelming majority of IGAs described in the literature, which use a small population and have the user evaluate each individual. The

large population is possible because of the surrogate fitness function screening for the best individuals to be subjectively evaluated. It is reasonable to believe that the more accurate the surrogate fitness function is, the more optimal the solution will be. Our research has shown one configuration of an IGA system for the limited domain of six facial expressions. While the NNs were an adequate surrogate fitness function, they were expensive to build and specific to each expression. It is not clear whether NNs are the best surrogate fitness function for face models. It is likely that our training sets were not optimal. More research into predictive mechanisms for FACS-based face models would be extremely important to furthering the IGA capability in this domain.

Recommendations

Our research accomplished its primary objective, but in the process it raised many interesting questions.

Most importantly, it is believed that the quality of the surrogate fitness function is critical to the success of the IGA. It would be interesting to experiment with the composition of the NN training sets and the NN parameters, and run a series of experiments with each configuration. These experiments could restrict the AUs that are allowed to evolve to be the same set rather than a random set for a more accurate comparison. The evolving AUs should be adequately represented in the NN training sets. With these restrictions in place, there should be relationships among the subjective and surrogate fitness values of the IGA, and the TPR/FPR values of the NN.

Along the same lines, it would also be interesting to compare other types of pattern classification systems serving as a surrogate fitness function. Bayesian networks, decision trees, or one of the other variants of NNs are examples of possible alternatives.

Our research showed that certain AUs could prevent the initial population from containing solutions of sufficient quality to rate above a minimal subjective fitness. The surrogate fitness function alone is not precise enough to improve the population in such cases and the IGA fails to produce an acceptable solution.. There is a subset of AUs unique to each emotion that destroys the credibility of the expression unless their value is very low. Additionally, some AU combinations have this detrimental effect even though individually they are fine. It is likely that the IGA would be more effective if the user were allowed to select which AUs were allowed to evolve for each run. This would eliminate the detrimental AUs from the target emotion. Alternatively, the detrimental AUs could be identified and restricted to a small range without involving the user. It would also be interesting if the IGA could learn over time which AUs were detrimental for the target emotion and incorporated into rules as was done by Dahstedt (2007) in the music domain.

Speech-driven animation is an important focus in the facial animation research. It would be interesting to specify a particular sequence of AUs to evolve the animation of a specific phrase. This would be particular useful for incorporating the IGA into a rule-based system such as that presented by Rosis, Pelachaud, Poggi, Carofiglio, and Carolis (2003). In a rule-based system, the selected AUs could be provided by the rules rather than manually specified by the user.

The GA parameters had a significant impact on the success of the IGA. Our experiments varied population size and tournament size with definite winners for each. These experiments were far from exhaustive. Different combinations of population size, tournament size, crossover rate, and mutation rate could reveal a more optimal configuration than the one we discovered.

Our IGA system creates a two-second animation. To be useful, these animations would still have to be manually connected to a timeline. It would be interesting to create the timeline automatically by evolving two separate populations, similar to the work done by Tokui (2000). One population would combine face models into two-second clips, while the second population would combine the two-second clips into longer animation sequences. Temporal data, such as transition and duration, might be a component of the second population. The user might specify a phoneme sequence, where the first population evolves each individual phoneme, and the second population evolves the entire phrase.

Summary

A major focus of research in computer graphics is the modeling and animation of realistic human faces. The film and game industries have a big influence on the demand for improved facial animation. Embodied conversational agents are becoming popular as front ends to web sites, and as part of many computer applications such as virtual training environments, tutoring systems, storytelling systems, portable personal guides, and

entertainment systems (Mana and Pianesi, 2006). These types of applications will require realistic and believable graphical renderings of facial expressions.

Modeling and animation of facial expressions is a very difficult task, requiring extensive manual manipulation by computer artists. One of the primary research goals for facial animation is a system that creates realistic animation while reducing the amount of manual manipulation.

There is a rich stream of research focused on enhancing computerized facial animation. There is also a large body of literature investigating the use of interactive genetic algorithms in generating creative works. Our research combines these two lines of research.

Facial modeling and animation are often done with a polygonal mesh based on the pioneering work of Parke (Parke, 1972). The vertices of the mesh are manipulated to create changes in the basic face, such as raising the brows. In one of the more popular animation techniques, a number of face models are created from the basic shape. Then distinct model variations are selected and key-framed to points throughout the scene, interpolating from one model to the next. Due to its efficiency and simplicity, the blendshape approach is widely used for key framing facial animation (Li and Deng, 2008).

There is a large body of research using rule-based systems that focus on the roles of speaker and listener. When these systems incorporate emotion, they suffer from static generation, with no variation in the given facial expression. Ho and Huang (2004) developed a facial modeling system based on a polygonal mesh, using a GA that

transformed a 2-D image into a 3-D model. But animation is a temporal sequence, and we can find better similarities in the music domain. Tokui (2000) used a multilevel neural network as a surrogate fitness function to evolve music composition, evolving two populations separately, one for short pieces and the other for longer sequences of the short pieces. Dahstedt (2007) represented music composition as recursive binary trees, and used an IGA that incorporated rules based on observation and statistics to serve as a surrogate fitness function.

GAs are complex non-linear algorithms (Harik, et al., 1999). They work by discovering, emphasizing, and recombining good building blocks of solutions in a highly parallel manner (Mitchell, 1998). This is known as the schema theorem and is fundamental to the analysis of genetic algorithms. There must be a sufficient number of building blocks in the initial population to arrive at an optimal solution. Otherwise, the chances of the GA converging to a good solution are small (Harik et. al., 1999).

The major problem of IGAs is human fatigue. This is typically dealt with by using small populations. Unfortunately, small populations suffer from the lack of genetic diversity, resulting in poor performance and a tendency to converge to a non-optimal solution. To find solutions of high quality, the population size must be increased as much as possible (Harik, Cantu-Paz, Goldberg, & Miller, 1999). One method that has been used to address this problem is a fitness prediction function, also called a surrogate function. This algorithm uses with a large population, applies a predictive fitness function to all the individuals, and then shows a small subset of the most likely candidates to the user for evaluation (Takagi, 2001; Jin, 2005).

In the case of a high-dimensional input space and a limited number of samples, a neural network is often used as a surrogate fitness function (Jin, 2005). Neural networks are well suited for complex pattern classifications, and have been used to classify facial expressions in a number of research projects. By using NNs as a fitness approximation function, it is possible to use larger populations for an IGA.

Our research project is an IGA system that evolves 3-D animation sequences of one of the six basic emotions. The FACS-based genome maps directly to the blendshape controls. A NN serves as a surrogate fitness function to enable the use of a large population. This is a unique approach to the important problem of automatic generation of facial animation.

The IGA system presents the eight most-fit phenotypes to the user, who provides a subjective fitness value. The subjective fitness values bias the selection process by giving the subjective fitness value priority over the surrogate fitness value. The chromosomes maintain both the surrogate and subjective fitness values into the next generation unless they are changed by crossover or mutation, at which point the subjective fitness value is reset to zero. Tournament selection is used to select the mating pairs.

The chromosome represents a sequence of n genes, each of which encodes a face model comprised of 39 blendshape controls. The genes are sequenced so that the *i*th gene represents the *i*th face model in the animation sequence. The number of face models does not evolve, but is a parameter that can be set by the user. The NNs were trained to

evaluate a single facial expression. Thus, the NN performs *n* evaluations on each chromosome, one evaluation for each face model.

Each experimental result reported is the average of ten runs of the IGA. Several experiments were conducted to test various parameter settings. Then, a final set of experiments was conducted with all six target emotions to get a final consistent set of data. The final set of data was generated with a tournament size of 8, crossover rate of 0.8, mutation rate of 0.001, population size of 100, number of random AUs of 5, and number of genes of 5.

Several experiments were conducted with a disabled surrogate fitness function and did not converge at all. This suggests that the ability of the NN to select the 8 best individuals for subjective fitness evaluation was a crucial part of the IGA system. In the final consistent set of experiments, there was a correlation between the TPR/FPR points in ROC space of the NNs and the convergence rate of their corresponding IGAs.

One of the motivating factors behind using a surrogate fitness function was to increase the number of building blocks by using a large population. Three experiments were conducted to compare the effects population size had on the IGA, showing a significant advantage for the larger population size. This suggests that the smaller population size did not provide a sufficient number of building blocks.

One of the interesting sets of experiments varied the tournament size. When the tournament size was increased, the most fit chromosomes quickly spread throughout the population and convergence results improved. Due to issues of user fatigue, it was important for the user to perceive improved phenotypes within the first few generations.

In the case of an IGA with a surrogate fitness function and a large population, it appears that the emphasis is better placed on exploitation rather than exploration.

In the final consistent set of experiments, convergence rates ranged from 70% to 100%, and user satisfaction ranged from 5.6 to 8.5 out of 10. There was a strong correlation between the convergence rate and average user satisfaction. The quality of the IGA system from the perspective of the user was at least above average and was often quite favorable.

The data show that identification rates were 53% for the IGA animations and 60% for the manual animations. The intensity, naturalness, sincerity, and credibility ratings were all slightly higher for the IGA animations than for the manual animations overall. In fact, the credibility ratings for the IGA animations were the same or higher than the manual animations for every emotion. The overall credibility averaged 69% for the IGA animations and 65% for the manual animations. The results from the preference questionnaire showed a preference for the manual animations for anger and disgust, and a preference for the IGA animations for the remaining four emotions. The preference total average over all the emotions was 54% for the IGA animations and 46% for the manual animations. The results of the questionnaires indicate that the IGA animations were as credible and liked as well as the manual animations.

Once an IGA system has been constructed, it can produce variations in expressions in a fraction of the time that an individual can manually. Our research has shown that an IGA is capable of generating credible variations in facial expressions. It

has also provided evidence that a large population with a surrogate fitness function has an advantage over a small population.

Our research has shown one configuration of an IGA system for the limited domain of six facial expressions. While the NNs were an adequate surrogate fitness function, they were expensive to build and specific to each expression. It is not clear whether NNs are the best surrogate fitness function for face models. It is likely that our training sets were not optimal. More research into predictive mechanisms for FACS-based face models would be extremely important to furthering the IGA capability in this domain.

Reference List

Baum, E., and Haussler, D. (1989). What size net gives valid generalization?. *Neural Comput.*1, 1 (March 1989), 151-160.

Biles, J.A. (1994). Genjam: A genetic algorithm for generating jazz solos. *In Proc. Int. Comput. Music Conf*. 131-187.

Bishop, C.M. (1995) Neural Networks for Pattern Recognition. New York: Oxford University Press.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30(7), 1145-1159.

Bui, T.D., Heylen, D.K.J., Nijholt, A. and Poel, M. (2001). Generation of Facial Expressions from Emotion Using a Fuzzy Rule Based System. *In Proceedings of the 14th Australian Joint Conference on Artificial Intelligence*, December 2001.

Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S., and Stone, M. (1994). Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. *In Proceedings of the 21st Annual Conference on Computer Graphics and interactive Techniques*, 413-420.

Cohen, M.M, and Massaro, D.W. (1993). Modeling coarticulation in synthetic visual speech. *In Models and Techniques in Computer Animation,* 139-156.

Cunningham, D. W., Kleiner, M., Bilthoff, H. H., and Wallraven, C. (2004). The components of conversational facial expressions. *In Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization*, 73, 143-150.

Dahlstedt, P. (2007). Autonomous Evolution of Complete Piano Pieces and Performances. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life*.

De Jong, K.A., Spears, W.M. (1992). A formal analysis of multi-point crossover in genetic algorithms. *Anals of Mathematics and Artificial Intelligence*, 5(1), 1-26.

Deng, Z. and Neumann, U. (2007). Data-Driven 3D Facial Animation. Springer-Verlag Press, October 2007.

Donato, G., Bartlett, M., Hager, J., Ekman, P., and Sejnowski, T. (1999). Classifying facial actions. *IEEE Pattern Analysis and Machine Intelligence*, 21, 10, 974 989.

Duda, R.O., Hart, P.E., Stork D.G. (2001) Pattern classification (2nd ed). New York: John Wiley & Sons, Inc.

Ekman, P. and Friesen, W.V. (1978). Facial action coding system. Consulting Psychologists Press.

Ekman, P. (2003). Darwin, Deception, and Facial Expression. *Annals of the New York Academy of Sciences, Vol. 1000,* pp. 205-221.

Fasel, B., and Luettin, J. (2003). Automatic facial expression analysis: a survey. *Pattern Recognition* 36, 259–275.

Fawcett, T. (2003). ROC graphs: notes and practical considerations for data mining researchers. *Technical Report HPL-2003–4*, Palo Alto, CA. HP Laboratories.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, 1 edition.

Goldberg, D.E. and Deb, K. (1991). A comparison of selection schemes used in genetic algorithms. *In G.E. Rawlins Ed., Foundations of Genetic Algorithms*, 69-93. San Mateo, Ca: Morgan Kaufmann.

Goldberg, D.E., Sastry, K., & Latoza, T. (2002). On The Supply of Building Blocks. *Paper presented at the Proceeding of GECCO* 2002.

Grefenstette, J. (1986). Optimization of control parameters for genetic algorithms. IEEE Trans. *Syst. Man Cybern*. SMC-16, 1, 122 128.

Griesser, R. T., Cunningham, D. W., Wallraven, C., and Bilthoff, H. H. (2007). Psychophysical investigation of facial expressions using computer animated faces. *In Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization*, 253, 11-18.

Halim, M. F. and Al-Fiadh, H. H. (2006). Facial Composite System Using Genetic Algorithm. *In Proceedings of the international Conference on Computer Graphics, Imaging and Visualization*, 262-266.

Harik, G., Cantu-Paz, E., Goldberg, D.E., and Miller, B.L. (1999). The Gambler's Ruin Problem, Genetic Algorithms, and the Sizing of Populations. *Transactions on Evolutionary Computation*, 7, 231-253.

Ho, S. and Huang, H. (2001). Facial modeling from an uncalibrated face image using a coarse-to-fine genetic algorithm, *Patten Recognition* 34, 1015-1031.

Jin, Y., Olhofer, M., and Sendhoff, B. (2001). Managing approximate models in evolutionary aerodynamic design optimization. *In Proceedings of IEEE Congress on Evolutionary Computation*, 1, 592– 599.

Jin, Y., Olhofer, M., and Sendhoff, B. (2002). A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494.

Jin, Y. (2005). A comprehensive survey of fitness approximation in evolutionary computation. *Soft Comput*. 9, 1, 3-12.

Karungaru, S. Fukumi, M., and Akamatsu, N. (2007). Automatic human faces morphing using genetic algorithms based control points selection. *International Journal of Innovative Computing, Information & Control*, 3(2), 247–256.

Li, Q. and Deng, Z. (2008). Orthogonal-Blendshape-Based Editing System for Facial Motion Capture Data. *IEEE Comput. Graph. Appl*. 28, 6, 76-82.

Llora, X., Sastry, K., Goldberg, D. E., Gupta, A., and Lakshmi, L. (2005). Combating user fatigue in iGAs: partial ordering, support vector machines, and synthetic fitness. *In Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, 1363-1370.

Mana, N. and Pianesi, F. (2006). HMM-based synthesis of emotional facial expressions during speech in synthetic talking heads. *In Proceedings of the 8th international Conference on Multimodal interfaces*, 380-387.

Masters, T. (1993). Practical Neural Network Recipes in C++. Academic Press Professional, Inc.

Matsumoto, D. and Ekman, P. (2008), Scholarpedia, 3(5):4237.

Mitchell, M. (1998). An introduction to genetic algorithms. Cambridge: MIT Press.

Moller, M.F. (1993). Original Contribution: A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.* 6, 4 (April 1993), 525-533.

Muhlenbein, H. (1989). Parallel genetic algorithms, population genetics and combinatorial optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, 416-421.

Nissen, S. (2003). Implementation of a fast artificial neural network library. *Technical report*, Department of Computer Science, University of Copenhagen, 2003.

Ohsaki, M., Takagi, H., and Ohya, K. (1998). An input method using discrete fitness values for interactive GA. *J. Intell. Fuzzy Syst*. 6, 1, 131-145.

Parke, F. I. (1972). Computer generated animation of faces. *In Proceedings of the ACM Annual Conference* - Volume 1, 451-457.

Parke, F. and Waters, K. (2008). Computer Facial Animation. Second. AK Peters Ltd.

Pearce, A., Wyvill, B., Wyvill, G., & Hill, D. (1986) Speech and expression: A computer solution to face animation. *Graphics Interface* 86.

Roble, D. and Bin Zafar, N. (2009). Don't Trust Your Eyes: Cutting-Edge Visual Effects. *Computer* 42, 7, 35-41.

Rosis, F. D., Pelachaud, C., Poggi, I., Carofiglio, V., and Carolis, B. D. (2003). From Greta's mind to her face: modelling the dynamics of affective states in a conversational embodied agent. *International Journal of Human-Computer Studies,* 59, 1-2.

Sun, X.Y., Gong, D., and Li, S. (2009). Classification and regression-based surrogate model-assisted interactive genetic algorithm with individual's fuzzy fitness. *In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, 907-914.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296.

Tokui N. (2000). Music Composition with Interactive Evolutionary Computation. In: Iba H (eds) *Proc. of 3rd International Conference on Generative Art*, Milan.

Wallraven, C., Breidt, M., Cunningham, D.W., and Bilthoff, H. (2005). Psychophysical evaluation of animated facial expressions. In *Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, 17-24.

Appendices

Appendix A: Convergence Results for Each IGA Run

The convergence results of each run are included in this appendix. The successful and failed convergence data were in separate files so their appearance in the charts is not necessarily in the order in which they occurred.

Table 31: Convergence Results: Happiness Experiment 1

Target: Happiness Population Size: 100 Number Random AUs: 7 Tournament Size: 4 Number NN Training Samples: 709	
Converged	User Satisfaction
No	1
No	1
No	1
No	1
No	1
No	1
No	1
No	1
No	1
No	1

Table 32: Convergence Results: Happiness Experiment 2

Target: Happiness Population Size: 20 Number Random AUs: 5 Tournament Size: 4 Number NN Training Samples: 709	
Converged	User Satisfaction
Yes	6
Yes	7
Yes	6
No	1
No	1
No	1
No	1
No	1
No	1
No	1

Table 33: Convergence Results: Happiness Experiment 3

Target: Happiness Population Size: 20 Number Random AUs: 3 Tournament Size: 4 Number NN Training Samples: 709	
Converged	User Satisfaction
Yes	6
Yes	8
Yes	6
No	1
No	1
No	1
No	1
No	1
No	1
No	1

Table 34: Convergence Results: Happiness Experiment 4

Target: Happiness Population Size: 100 Number Random AUs: 5 Tournament Size: 4 Number NN Training Samples: 709	
Converged	User Satisfaction
Yes	6
Yes	6
Yes	7
Yes	7
Yes	7
No	1
No	1
No	1
No	1
No	1

Table 35: Convergence Results: Happiness Experiment 5

Target: Happiness Population Size: 100 Number Random AUs: 3 Tournament Size: 4 Number NN Training Samples: 709	
Converged	User Satisfaction
Yes	6
Yes	7
Yes	8
No	4
No	1
No	4
No	1
No	1
No	1
No	1

Table 36: Convergence Results: Happiness Experiment 6

Target: Happiness Population Size: 100 Number Random AUs: 5 Tournament Size: 8 Number NN Training Samples: 1001	
Converged	User Satisfaction
Yes	9
Yes	6
Yes	8
Yes	7
Yes	6
Yes	7
Yes	7
No	1
No	4
No	1

Table 37: Convergence Results: Sadness Experiment 1

Target: Sadness Population Size: 20 Number Random AUs: 5 Tournament Size: 4 Number NN Training Samples: 698	
Converged	User Satisfaction
Yes	8
Yes	9
Yes	7
Yes	8
Yes	7
Yes	7
Yes	6
No	3
No	1
No	1

Table 38: Convergence Results: Sadness Experiment 2

Target: Sadness Population Size: 100 Number Random AUs: 5 Tournament Size: 4 Number NN Training Samples: 698	
Converged	User Satisfaction
Yes	6
Yes	9
Yes	9
Yes	8
Yes	9
No	4
No	1
No	1
No	1
No	1

Table 39: Convergence Results: Sadness Experiment 3

Target: Sadness Population Size: 100 Number Random AUs: 5 Tournament Size: 8 Number NN Training Samples: 698	
Converged	User Satisfaction
Yes	8
Yes	6
Yes	7
Yes	9
Yes	6
Yes	9
Yes	8
Yes	9
No	3
No	1

Table 40: Convergence Results: Sadness Experiment 4

Target: Sadness Population Size: 100 Number Random AUs: 5 Tournament Size: 8 Number NN Training Samples: 930	
Converged	User Satisfaction
Yes	9
Yes	9
Yes	7
Yes	9
Yes	9
Yes	4
Yes	7
No	1
No	1
No	1

Table 41: Convergence Results: Anger Experiment 1

Target: Anger Population Size: 100 Number Random AUs: 5 Tournament Size: 8 Number NN Training Samples: 731	
Converged	User Satisfaction
Yes	9
Yes	8
Yes	7
Yes	8
Yes	8
Yes	7
Yes	9
Yes	7
Yes	6
No	1

Table 42: Convergence Results: Anger Experiment 2

Target: Anger Population Size: 100 Number Random AUs: 5 Tournament Size: 6 Number NN Training Samples: 731	
Converged	User Satisfaction
Yes	6
Yes	8
Yes	7
Yes	6
Yes	7
Yes	7
No	5
No	2
No	1
No	1

Table 43: Convergence Results: Anger Experiment 3

Target: Anger Population Size: 100 Number Random AUs: 5 Tournament Size: 4 Number NN Training Samples: 731	
Converged	User Satisfaction
Yes	6
Yes	7
Yes	6
Yes	7
Yes	7
Yes	7
Yes	8
No	1
No	1
No	3

Table 44: Convergence Results: Anger Experiment 4

Target: Anger Population Size: 100 Number Random AUs: 5 Tournament Size: 8 Number NN Training Samples: 915	
Converged	User Satisfaction
Yes	8
Yes	8
Yes	7
Yes	8
Yes	7
Yes	7
Yes	6
Yes	7
Yes	6
Yes	6

Table 45: Convergence Results: Fear Experiment 1

Target: Fear Population Size: 100 Number Random AUs: 5 Tournament Size: 8 Number NN Training Samples: 1005	
Converged	User Satisfaction
Yes	8
Yes	8
Yes	9
Yes	9
Yes	9
Yes	9
Yes	6
Yes	9
Yes	9
Yes	9

Table 46: Convergence Results: Surprise Experiment 1

Target: Surprise Population Size: 100 Number Random AUs: 5 Tournament Size: 8 Number NN Training Samples: 1008	
Converged	User Satisfaction
Yes	8
Yes	9
Yes	8
Yes	8
Yes	6
Yes	6
Yes	9
Yes	8
Yes	1
No	4

Table 47: Convergence Results: Disgust Experiment 1

Target: Disgust Population Size: 100 Number Random AUs: 5 Tournament Size: 8 Number NN Training Samples: 754	
Converged	User Satisfaction
Yes	8
Yes	9
Yes	8
Yes	8
Yes	6
Yes	6
Yes	9
Yes	8
No	1
No	4

Appendix B: Surrogate and Subjective Fitness Values

This appendix is a compilation of the subjective and surrogate fitness values for each run in the final consistent set of experiments.

Table 48: Subjective and Surrogate Fitness: Happiness

Target: Happiness
Population Size: 100
Number Random AUs: 5
Tournament Size: 8
Number NN Training Samples: 1001

Best Subjective Fitness	Average Subjective Fitness	Subjective Standard Deviation	Best Surrogate Fitness	Average Surrogate Fitness	Surrogate Standard Deviation	Num Runs
9	4.0	3.57	5.00	4.99	0.01	2
6	3.0	2.15	4.99	4.96	0.03	4
8	5.0	2.00	4.99	4.79	0.12	3
7	3.0	2.92	5.00	5.00	0.00	4
7	1.0	2.12	5.00	5.00	0.00	4
8	4.0	2.83	5.00	5.00	0.00	4
7	4.0	2.52	5.00	4.97	0.31	4
1	1.0	0.00	5.00	4.93	0.04	5
5	2.75	1.85	4.93	4.69	0.08	5
6	1.62	1.65	5.00	5.00	0.00	4

Table 49: Subjective and Surrogate Fitness: Sadness

Target: Sadness
Population Size: 100
Number Random AUs: 5
Tournament Size: 8
Number NN Training Samples: 930

Best Subjective Fitness	Average Subjective Fitness	Subjective Standard Deviation	Best Surrogate Fitness	Average Surrogate Fitness	Surrogate Standard Deviation	Num Runs
9	7.0	2.35	4.95	4.21	0.61	1
9	6.0	1.58	4.87	4.83	0.02	3
9	4.0	2.57	4.95	4.93	0.02	3
9	6.0	3.43	4.84	4.53	0.38	3
9	6.0	2.81	4.93	4.88	0.04	3
7	3.0	2.92	4.75	4.75	0.00	10
7	4.0	2.45	4.95	4.92	0.01	3
7	2.5	2.60	4.76	4.75	0.09	9
1	1.0	0.00	4.85	4.83	0.01	5
1	1.0	0.00	2.96	2.23	0.25	5

Table 50: Subjective and Surrogate Fitness: Anger

Target: Anger
Population Size: 100
Number Random AUs: 5
Tournament Size: 8
Number NN Training Samples: 915

Best Subjective Fitness	Average Subjective Fitness	Subjective Standard Deviation	Best Surrogate Fitness	Average Surrogate Fitness	Surrogate Standard Deviation	Num Runs
8	7	0.94	4.90	4.74	0.12	3
8	4	3.12	4.95	4.85	0.09	4
9	3	3.35	4.98	4.84	0.24	3
8	6	2.09	5.00	4.98	0.02	3
7	3	2.76	4.92	4.91	0.05	5
8	6	1.54	4.95	4.85	0.07	3
7	5	1.17	4.89	4.59	0.36	3
8	7	0.79	5.00	4.99	0.01	3
8	6	1.32	5.00	4.99	0.01	4
7	5	1.66	4.51	3.76	0.54	3

Table 51: Subjective and Surrogate Fitness: Fear

Target: Fear
Population Size: 100
Number Random AUs: 5
Tournament Size: 8
Number NN Training Samples: 1005

Best Subjective Fitness	Average Subjective Fitness	Subjective Standard Deviation	Best Surrogate Fitness	Average Surrogate Fitness	Surrogate Standard Deviation	Num Runs
9	5	2.98	4.99	4.86	0.43	3
9	4	3.08	5.00	4.98	0.01	3
10	8	1.27	4.88	3.76	0.95	2
8	6	2.37	4.88	4.69	0.15	4
9	6	3.30	5.00	5.00	0.01	4
9	5	3.06	4.95	4.79	0.15	3
7	3	2.65	4.96	4.90	0.07	5
9	7	2.50	4.99	4.76	0.25	2
10	9	0.35	4.99	4.81	0.31	2
9	7	1.17	5.00	4.98	0.02	3

Table 52: Subjective and Surrogate Fitness: Surprise

Target: Surprise
Population Size: 100
Number Random AUs: 5
Tournament Size: 8
Number NN Training Samples: 1008

Best Subjective Fitness	Average Subjective Fitness	Subjective Standard Deviation	Best Surrogate Fitness	Average Surrogate Fitness	Surrogate Standard Deviation	Num Runs
8	3	2.69	4.94	4.92	0.01	3
9	6	2.29	4.94	4.08	0.65	2
7	5	1.87	4.70	3.59	0.52	3
7	3	1.94	4.95	4.94	0.00	6
7	4	1.77	4.90	4.90	0.01	4
7	3	2.92	4.95	4.90	0.13	4
7	3	2.60	4.92	4.88	0.02	5
8	6	2.09	4.90	4.88	0.02	3
6	3	1.77	4.95	4.94	0.01	3
2	1.12	0.33	4.94	4.84	0.09	2

Table 53: Subjective and Surrogate Fitness: Disgust

Target: Disgust
Population Size: 100
Number Random AUs: 5
Tournament Size: 8
Number NN Training Samples: 754

Best Subjective Fitness	Average Subjective Fitness	Subjective Standard Deviation	Best Surrogate Fitness	Average Surrogate Fitness	Surrogate Standard Deviation	Num Runs
9	6	2.37	4.44	4.39	0.03	4
10	6	3.64	4.35	4.01	0.24	3
8	2	3.12	4.28	3.65	0.46	3
9	5	3.12	4.40	4.33	0.05	3
7	3	2.81	4.31	3.81	0.45	4
6	4	1.80	4.25	4.17	0.11	4
9	5	2.60	4.36	3.63	0.56	2
8	3	3.10	4.38	4.28	0.06	5
1	1	0.00	4.19	3.81	0.14	4
7	2.38	2.69	4.29	4.24	0.03	5

Appendix C: Participant Evaluations

This appendix includes the responses from each of the participants for the first set of 24 animations.

Table 54: Participant 1 Responses

Anim. Num.	Origin (GA or Manual)	Target Emotion	Identified Emotion	Inten. (1-7)	Natur. (1-7)	Sinc. (1-7)	Cred. (1-7)	Time
1	Manual	1	1	7	6	7	6	09:12:32
2	Manual	5	5	4	6	7	6	09:13:01
3	GA	3	3	4	6	7	5	09:14:16
4	Manual	1	2	4	6	7	6	09:14:38
5	Manual	5	5	2	6	7	6	09:14:57
6	GA	2	2	6	6	6	6	09:15:11
7	GA	6	6	4	7	7	7	09:15:24
8	Manual	2	1	5	6	6	6	09:15:50
9	Manual	6	6	3	7	7	7	09:16:05
10	GA	2	2	5	6	6	5	09:16:30
11	Manual	1	2	5	5	5	5	09:17:13
12	GA	4	4	6	6	6	6	09:17:28
13	GA	2	5	2	6	6	6	09:17:38
14	Manual	5	5	6	6	6	7	09:17:56
15	Manual	6	6	2	6	6	7	09:18:09
16	GA	3	3	4	6	6	7	09:18:27
17	GA	4	4	6	6	7	7	09:18:39
18	GA	1	2	6	6	7	7	09:18:50
19	GA	3	5	3	6	7	6	09:19:07
20	Manual	4	4	7	6	7	7	09:19:17
21	Manual	4	4	2	5	7	7	09:19:42
22	GA	3	5	3	5	6	6	09:19:59
23	Manual	6	6	6	6	7	7	09:20:12
24	GA	5	5	6	6	7	7	09:20:30

Table 55: Participant 2 Responses

Anim. Num.	Origin (GA or Manual)	Target Emotion	Identified Emotion	Inten. (1-7)	Natur. (1-7)	Sinc. (1-7)	Cred. (1-7)	Time
1	Manual	1	1	4	3	3	4	15:15:40
2	Manual	5	5	5	4	4	4	15:16:25
3	GA	3	3	4	3	4	4	15:16:47
4	Manual	1	3	4	3	4	4	15:17:08
5	Manual	5	3	4	3	4	4	15:17:36
6	GA	2	6	4	4	5	4	15:17:55
7	GA	6	6	4	4	5	4	15:18:02
8	Manual	2	2	5	4	5	4	15:18:17
9	Manual	6	4	5	5	4	5	15:18:34
10	GA	2	1	4	3	4	4	15:19:00
11	Manual	1	4	5	3	3	4	15:19:34
12	GA	4	0	3	3	2	4	15:19:50
13	GA	2	1	3	4	5	3	15:20:10
14	Manual	5	5	3	5	3	4	15:20:43
15	Manual	6	6	5	6	7	5	15:21:15
16	GA	3	4	4	3	3	3	15:21:40
17	GA	4	4	4	5	6	5	15:22:03
18	GA	1	2	4	5	6	5	15:23:18
19	GA	3	3	4	5	2	5	15:23:30
20	Manual	4	6	4	5	5	5	15:23:40
21	Manual	4	5	5	4	4	4	15:23:59
22	GA	3	6	5	3	5	6	15:24:22
23	Manual	6	6	5	5	7	6	15:25:07
24	GA	5	5	5	5	7	6	15:25:14

Table 56: Participant 3 Responses

Anim. Num.	Origin (GA or Manual)	Target Emotion	Identified Emotion	Inten. (1-7)	Natur. (1-7)	Sinc. (1-7)	Cred. (1-7)	Time
1	Manual	1	1	4	4	7	7	15:31:46
2	Manual	5	5	4	7	7	3	15:33:37
3	GA	3	6	5	7	7	6	15:34:38
4	Manual	1	1	6	7	5	7	15:35:29
5	Manual	5	5	3	3	1	1	15:36:04
6	GA	2	6	6	7	7	7	15:36:43
7	GA	6	6	2	2	2	2	15:37:27
8	Manual	2	1	6	7	7	7	15:37:54
9	Manual	6	1	7	7	7	7	15:38:34
10	GA	2	2	7	7	7	7	15:38:53
11	Manual	1	1	7	7	7	7	15:39:16
12	GA	4	1	2	3	2	2	15:39:45
13	GA	2	1	7	7	7	7	15:40:03
14	Manual	5	1	1	1	3	3	15:40:36
15	Manual	6	6	1	5	5	5	15:41:16
16	GA	3	6	7	7	7	7	15:41:49
17	GA	4	6	7	1	1	1	15:42:28
18	GA	1	6	7	7	7	7	15::42:58
19	GA	3	6	3	3	2	1	15:43:35
20	Manual	4	6	2	1	2	1	15:43:57
21	Manual	4	4	1	1	1	3	15:44:23
22	GA	3	4	5	5	3	4	15:45:03
23	Manual	6	4	6	6	6	6	15:45:28
24	GA	5	4	6	1	1	1	15:45:51

Table 57: Participant 4 Responses

Anim. Num.	Origin (GA or Manual)	Target Emotion	Identified Emotion	Inten. (1-7)	Natur. (1-7)	Sinc. (1-7)	Cred. (1-7)	Time
1	Manual	1	1	6	5	4	7	15:54:02
2	Manual	5	5	5	4	4	4	15:55:25
3	GA	3	6	5	5	5	4	15:56:37
4	Manual	1	2	4	3	4	2	15:57:25
5	Manual	5	5	5	5	6	4	15:58:32
6	GA	2	1	6	6	6	4	15:58:32
7	GA	6	6	5	5	6	4	16:00:21
8	Manual	2	2	3	2	2	2	16:01:21
9	Manual	6	6	4	6	6	6	16:01:55
10	GA	2	2	5	5	6	6	16:02:18
11	Manual	1	1	5	5	5	5	16:02:52
12	GA	4	4	5	3	5	2	16:03:29
13	GA	2	4	5	5	6	6	16:03:57
14	Manual	5	4	5	6	6	6	16:04:19
15	Manual	6	6	4	7	7	7	16:04:45
16	GA	3	6	4	4	5	5	16:05:11
17	GA	4	4	4	6	6	5	16:06:00
18	GA	1	2	7	4	6	5	16:06:33
19	GA	3	3	3	3	3	3	16:07:02
20	Manual	4	4	3	5	5	5	16:07:31
21	Manual	4	4	2	6	6	6	16:07:52
22	GA	3	3	4	4	4	4	16:08:15
23	Manual	6	6	4	6	5	6	16:08:51
24	GA	5	6	2	4	4	3	16:09:16

Table 58: Participant 5 Responses

Anim. Num.	Origin (GA or Manual)	Target Emotion	Identified Emotion	Inten. (1-7)	Natur. (1-7)	Sinc. (1-7)	Cred. (1-7)	Time
1	Manual	1	1	3	1	1	2	16:18:48
2	Manual	5	5	3	5	5	6	16:19:27
3	GA	3	3	6	5	6	6	16:20:00
4	Manual	1	1	3	5	4	5	16:21:05
5	Manual	5	0	1	2	2	2	16:21:58
6	GA	2	2	6	2	3	5	16:22:31
7	GA	6	6	4	4	4	6	16:23:21
8	Manual	2	2	3	3	3	3	16:24:08
9	Manual	6	2	3	3	3	3	16:24:09
10	GA	2	1	1	2	1	2	16:25:07
11	Manual	1	4	3	1	1	1	16:25:35
12	GA	4	4	6	6	6	6	16:26:11
13	GA	2	2	1	2	3	3	16:26:56
14	Manual	5	3	4	5	6	6	16:27:25
15	Manual	6	6	1	2	3	4	16:27:55
16	GA	3	3	4	3	4	4	16:28:22
17	GA	4	4	6	3	4	3	16:28:51
18	GA	1	1	6	5	4	5	16:29:12
19	GA	3	3	2	5	5	5	16:29:57
20	Manual	4	4	5	2	2	2	16:30:31
21	Manual	4	4	3	3	3	4	16:31:02
22	GA	3	3	1	4	5	5	16:31:23
23	Manual	6	6	4	5	6	6	16:31:45
24	GA	5	0	3	3	4	3	16:32:33

Table 59: Participant 6 Responses

Anim. Num.	Origin (GA or Manual)	Target Emotion	Identified Emotion	Inten. (1-7)	Natur. (1-7)	Sinc. (1-7)	Cred. (1-7)	Time
1	Manual	1	1	5	7	6	6	16:39:05
2	Manual	5	5	5	5	6	6	16:40:16
3	GA	3	3	6	5	6	6	16:41:20
4	Manual	1	3	6	6	6	6	16:41:55
5	Manual	5	5	6	6	7	7	16:42:54
6	GA	2	5	6	6	6	6	16:43:29
7	GA	6	6	6	6	6	6	16:43:53
8	Manual	2	2	6	6	6	6	16:44:16
9	Manual	6	4	6	6	6	6	16:44:37
10	GA	2	0	2	2	2	2	16:45:25
11	Manual	1	0	2	2	2	2	16:45:49
12	GA	4	4	5	5	5	5	16:46:16
13	GA	2	6	5	5	5	5	16:46:58
14	Manual	5	5	5	5	5	5	16:47:19
15	Manual	6	5	5	5	5	5	16:47:43
16	GA	3	0	5	5	5	5	16:48:00
17	GA	4	4	5	6	6	6	16:48:26
18	GA	1	1	6	6	6	6	16:48:58
19	GA	3	0	2	2	2	2	16:49:49
20	Manual	4	4	5	5	5	5	16:50:13
21	Manual	4	4	6	6	6	6	16:50:41
22	GA	3	6	6	6	6	6	16:51:00
23	Manual	6	6	7	7	7	7	16:51:23
24	GA	5	5	7	7	7	7	16:51:41

Table 60: Participant 7 Responses

Anim. Num.	Origin (GA or Manual)	Target Emotion	Identified Emotion	Inten. (1-7)	Natur. (1-7)	Sinc. (1-7)	Cred. (1-7)	Time
1	Manual	1	4	5	3	3	4	16:58:20
2	Manual	5	6	4	7	7	7	16:59:01
3	GA	3	5	4	7	7	7	16:59:42
4	Manual	1	1	3	5	5	6	17:00:25
5	Manual	5	5	2	6	4	4	17:01:27
6	GA	2	2	7	7	7	7	17:02:01
7	GA	6	6	5	7	7	7	17:02:38
8	Manual	2	0	6	7	7	7	17:03:35
9	Manual	6	6	5	7	7	7	17:04:02
10	GA	2	3	2	7	3	4	17:04:59
11	Manual	1	4	5	4	3	4	17:05:33
12	GA	4	4	5	7	7	7	17:06:03
13	GA	2	3	1	2	2	3	17:06:59
14	Manual	5	2	2	2	2	3	17:07:24
15	Manual	6	6	1	2	2	2	17:07:51
16	GA	3	1	3	6	6	6	17:08:31
17	GA	4	4	7	7	7	7	17:09:06
18	GA	1	3	7	7	7	7	17:09:32
19	GA	3	3	3	6	6	6	17:10:13
20	Manual	4	4	5	6	6	6	17:10:50
21	Manual	4	4	5	6	6	6	17:11:13
22	GA	3	5	3	6	4	4	17:11:49
23	Manual	6	6	7	7	7	7	17:12:06
24	GA	5	4	7	1	1	1	17:12:38

Table 61: Participant 8 Responses

Anim. Num.	Origin (GA or Manual)	Target Emotion	Identified Emotion	Inten. (1-7)	Natur. (1-7)	Sinc. (1-7)	Cred. (1-7)	Time
1	Manual	1	1	5	5	4	3	17:22:15
2	Manual	5	0	3	5	3	3	17:23:14
3	GA	3	0	5	4	4	3	17:23:55
4	Manual	1	2	4	3	3	3	17:24:32
5	Manual	5	0	2	2	2	2	17:25:22
6	GA	2	3	5	3	3	2	17:26:01
7	GA	6	0	3	3	3	2	17:26:42
8	Manual	2	2	5	4	4	4	17:27:20
9	Manual	6	6	3	4	4	4	17:28:10
10	GA	2	6	3	4	4	4	17:28:27
11	Manual	1	1	5	4	5	5	17:28:57
12	GA	4	4	5	5	5	5	17:29:49
13	GA	2	2	2	3	4	3	17:30:33
14	Manual	5	3	3	3	3	2	17:31:07
15	Manual	6	6	3	3	3	2	17:31:28
16	GA	3	6	5	4	5	5	17:31:53
17	GA	4	4	6	4	5	5	17:32:21
18	GA	1	1	6	4	3	3	17:32:43
19	GA	3	1	3	3	3	2	17:33:10
20	Manual	4	4	7	4	2	2	17:33:46
21	Manual	4	4	7	3	2	2	17:34:10
22	GA	3	3	2	2	2	1	17:34:42
23	Manual	6	3	5	5	5	4	17:35:09
24	GA	5	3	5	3	3	2	17:35:41

Appendix D: Screenshots of animations

This appendix includes snapshots of the two-second animation clips that were shown to participants for evaluation. An example is shown for each emotional expression generated both manually and by the IGA.

Figure 10: Sadness - Manually Generated

Figure 11: Sadness - IGA Generated

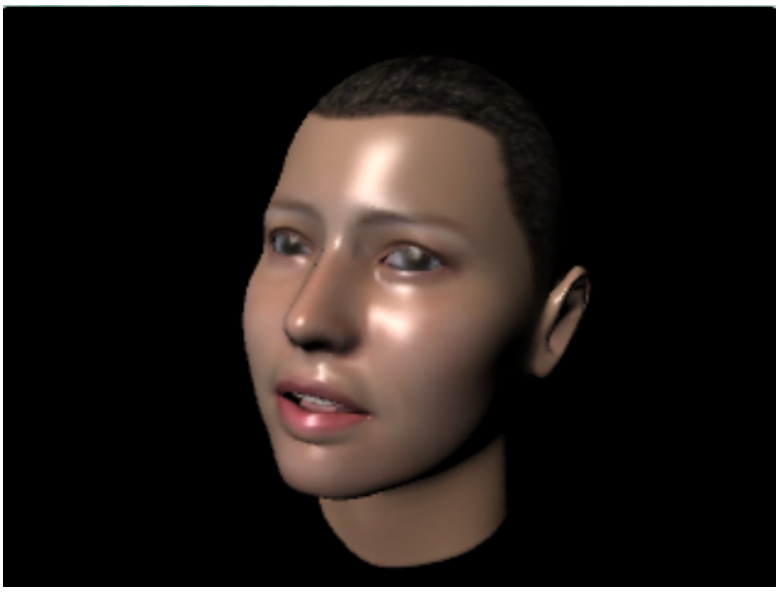


Figure 12: Surprise - Manually Generated

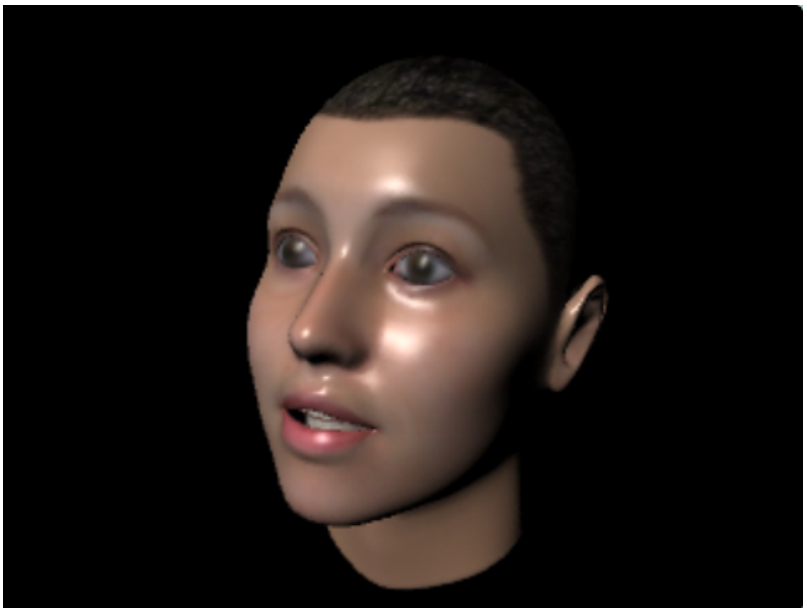


Figure 13: Surprise - IGA Generated

Figure 14: Disgust - Manually Generated

Figure 15: Disgust - IGA Generated

Figure 16: Fear - Manually Generated

Figure 17: Fear - IGA Generated

Figure 18: Happy - Manually Generated

Figure 19: Happy - IGA Generated

Figure 20: Anger - Manually Generated

Figure 21: Anger - IGA Generated

Appendix E: Screenshots of Animation Pairs

This appendix contains screenshots taken from the two-second animation clips that were shown for comparison to the participants. One example for each emotion is included.

Figure 22: Happy Comparison

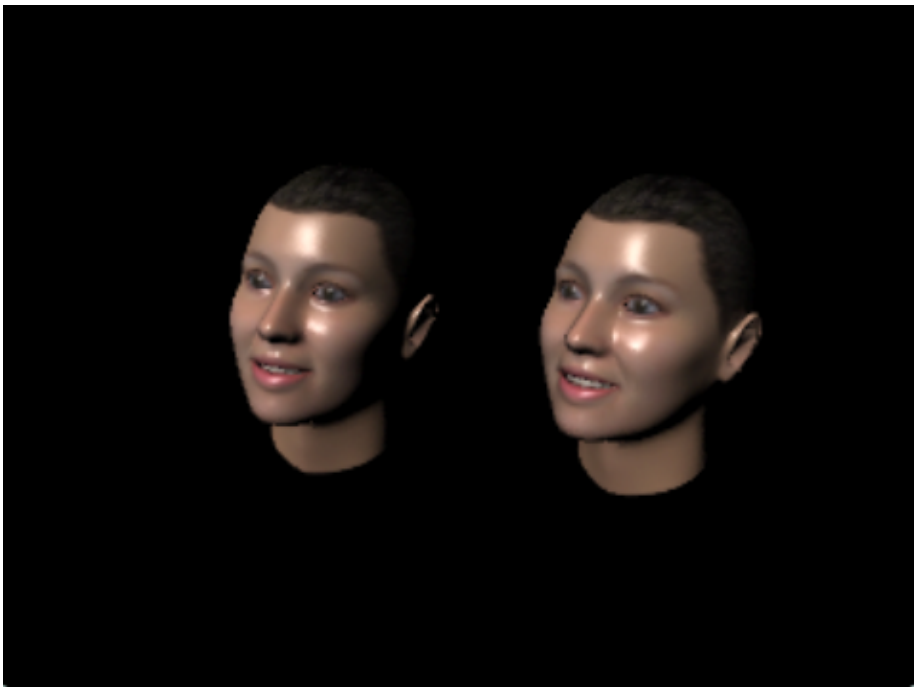


Figure 23: Anger Comparison

Figure 24: Disgust Comparison

Figure 25: Sad Comparison

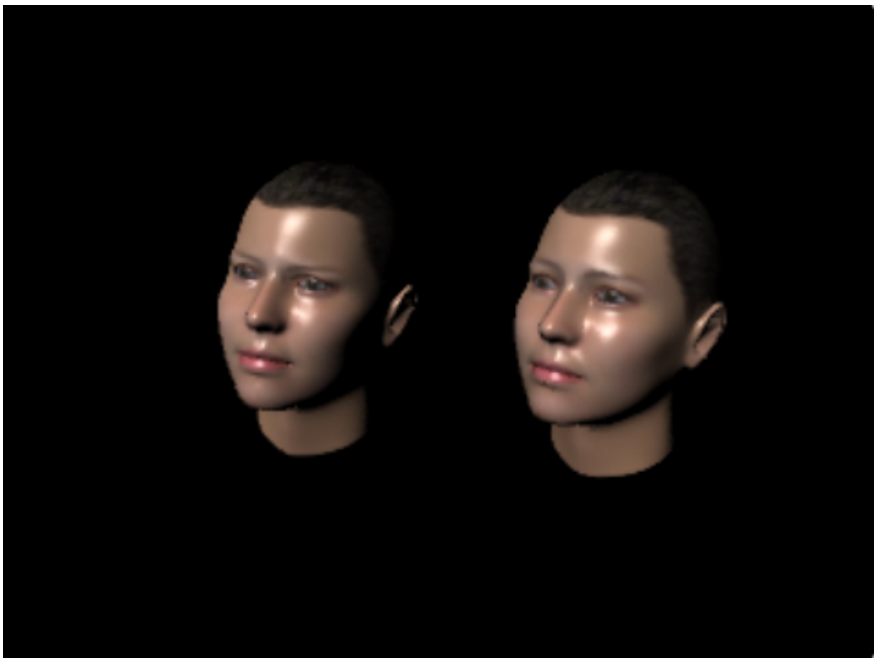


Figure 26: Fear Comparison

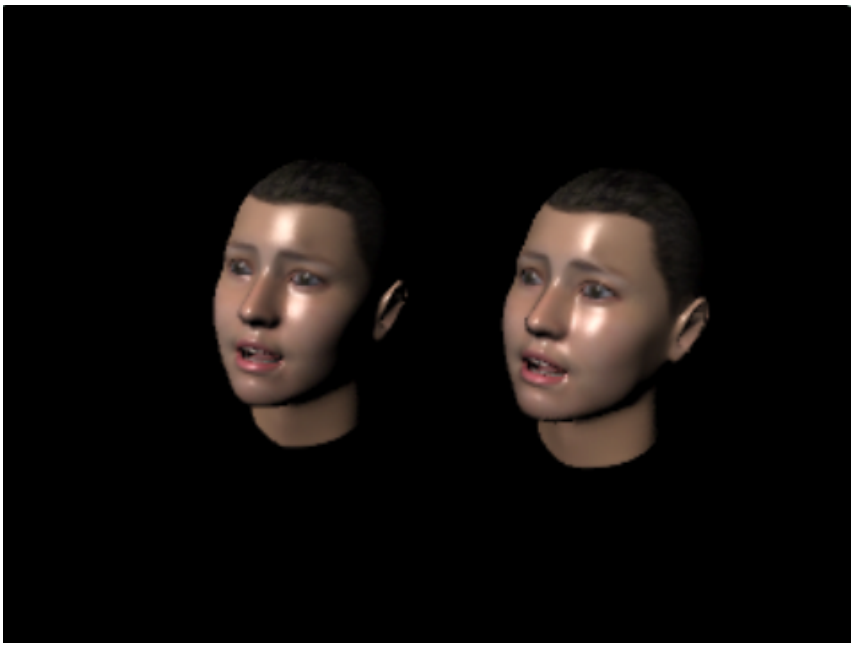


Figure 27: Surprise Comparison