2012

Imitating individualized facial expressions in a human-like avatar through a hybrid particle swarm optimization - tabu search algorithm

Evan Husk
University of Central Florida

IMITATING INDIVIDUALIZED FACIAL EXPRESSIONS IN A HUMAN-LIKE AVATAR THROUGH A HYBRID PARTICLE SWARM OPTIMIZATION – TABU SEARCH ALGORITHM

by

EVAN R. HUSK

A thesis submitted in partial fulfillment of the requirements
for the Honors in the Major Program in Computer Engineering
in the College of Engineering and Computer Science
and in The Burnett Honors College
at the University of Central Florida
Orlando, Florida

Fall Term 2012

Thesis Chair: Dr. Avelino Gonzalez

ABSTRACT

This thesis describes a machine learning method for automatically imitating a particular person's facial expressions in a human-like avatar through a hybrid Particle Swarm Optimization – Tabu Search algorithm. The muscular structures of the facial expressions are measured by Ekman and Friesen's Facial Action Coding System (FACS). Using a neutral face as a reference, the minute movements of the Action Units, used in FACS, are automatically tracked and mapped onto the avatar using a hybrid method. The hybrid algorithm is composed of Kennedy and Eberhart's Particle Swarm Optimization algorithm (PSO) and Glover's Tabu Search (TS). Distinguishable features portrayed on the avatar ensure a personalized, realistic imitation of the facial expressions. To evaluate the feasibility of using PSO-TS in this approach, a fundamental proof-of-concept test is employed on the system using the OGRE avatar. This method is analyzed in-depth to ensure its proper functionality and evaluate its performance compared to previous work.

DEDICATION

For my parents, friends, and family, thank you for your encouragement, support, and love.

For my committee members, Avelino Gonzalez, Sumanta Pattanaik, and Andrew Nevai, thank you for your wisdom, guidance, and patience throughout this process.

For my wife Kelli, thank you for being my strongest supporter with your kind words of encouragement and love.

ACKNOWLEDGMENTS

I wish to express my deepest thanks and gratitude to all of those who have helped me throughout this process. I would especially like to thank Dr. Avelino Gonzalez and Dr. Sumanta Pattanaik for the numerous hours spent working on this project. I thank Dr. Andrew Nevai for his words of wisdom and time spent serving on the committee. The patience and understanding of this committee is what made this Honors in the Major thesis possible. I would also like to thank Denise Crisafi and Kelly Astro for their unselfishness and willingness to help. Finally, I would like to thank my family and friends for their encouragement, support, and patience throughout this process.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION AND BACKGROUND

This thesis describes a technique to imitate personalized facial expressions in an avatar through a hybrid Particle Swarm Optimization – Tabu Search algorithm. Although avatars used in current practices may accurately represent a particular human subject in terms of detail and realism, they are limited in their ability to mimic personalized expressions of that human. Instead, these avatars are limited to generic facial expressions that do not always accurately simulate the intended expression of a specific human [1, 2, 3]. The lack of individualized facial expressions in life-like avatars expresses a need for a mechanism that can translate personal human expressions onto an avatar.

The mechanism described in this thesis seeks to accomplish this translation process in an automated and efficient manner. The machine learning algorithm analyzes a reference face of an avatar that is used as a target solution. The algorithm automatically adjusts a set of parameters used to control the avatar's facial "muscles" in an attempt to replicate the target face of the avatar based on purely visual likeness. In order to apply this algorithm to the avatar, the muscular structure and features of the avatar's face are defined and tracked using a form of the Facial Action Coding System (FACS) [4]. By designating Action Units (AUs) involved with the face, the movements and muscular contractions of the face can be detected and tracked to help evaluate the slight differences that make an avatar's facial expressions unique. By using a form of the Particle Swarm Optimization (PSO) algorithm [5] paired with a Tabu Search (TS) [6, 7], the tendencies and slight movements can be quantified and mapped to help expose the slight differences between the working and target avatar faces. The machine learning aspect of PSO and TS allows for an automatic execution of this translation process.

1.1 LONG-TERM OBJECTIVE

The long-term goal of this research is to facilitate the transfer of personal facial expressions of a specific human being onto an avatar intended to represent that individual. This person's expressions are to be extracted from a video clip of the specific human being displaying an emotion. The detection and tracking of the movements of features throughout the video will be accomplished by automatically defining and analyzing the movement of the AUs used in the FACS system. Once the AUs have been detected and key facial features have been mapped, the individualized characteristic of the subject will be translated to an avatar which is designated to closely resemble the particular human. Performing the analysis with the hybrid PSO-TS algorithm will allow for the quantification of the features. This, in turn, will permit the translation of these features directly onto the avatar. The slight muscular movements will be imitated by the avatar, ideally revealing an exact replica of the expression originally depicted by the human subject. Thus, the avatar will express the unique characteristics of the human's facial expressions instead of producing a generic emotional expression based on simply classifying the emotional state of the subject.

Although facial features are constantly being detected and extracted from the video segment in the process described above, it should be noted that the goal of this investigation is not to recognize a face within the video. Instead, it is assumed that a particular face already exists within the clip and is featured prominently within it. Moreover, it is assumed that certain constraints on the tilt, position, and overall alignment of the subject's head will be enforced. The background of the video segment is assumed to be uncluttered and in an ideal state. This permits a non-hindered feature and movement extraction process. Therefore, the key focus of this thesis is to extract distinct facial features and movements that make the subject's face unique. However, the facial expressions of the human subject will not be photographed while naturally expressing an emotion. Instead, the subject will be asked to express a specific emotion upon request, thereby introducing a measure of artificiality.

The avatar used in this long-term work can be defined as a virtual representation of a particular human. The avatar shall include specific features of the person in order to express realism and personalization. An example of an avatar, as defined in this thesis, can be seen in Figure 1. This avatar represents Dr. Alexander Schwarzkopf in a realistic and personalized manner and is generally referred to as the "Alex Avatar." The similarities between the photograph of Dr. Schwarzkopf and the Alex Avatar clearly show that this specific avatar was created to represent Dr. Schwarzkopf, and no one else. Thus, the avatar will only be used to mimic the facial expressions of Dr. Schwarzkopf.

It should be noted that the use of a human-like avatar is strictly part of the long-term objective of this research, and this thesis only incorporates the use of the OGRE avatar to serve as a proof-of-concept test. The mechanism developed in this thesis will serve as an intermediate step in the process of transferring personal facial expressions of a specific human being onto an avatar intended to represent that individual [8]. The *target* OGRE avatar face will serve as the photograph or video of the human subject displaying a facial expression. The personal characteristics of the human subject's facial expressions can be manually transferred onto the target avatar face via sliders that control the musculature of the OGRE avatar [8]. The mechanism then transfers the specific features of the target face onto the *working* avatar face. The working avatar face serves as the surrogate for the human-like avatar to be used in our long-term objective. By using a pixel-by-pixel approach to compare the target and working faces of the OGRE avatar, our method can be transferred to a similar system that uses a photograph of a human and a human-like avatar.

Figure 1: The "Alex Avatar" of Dr. Schwarzkopf (Reproduced without permission [9])

1.1.1 INDIVIDUALIZED FACIAL EXPRESSIONS

Considerable advances have been made in personalizing facial expressions. The works described in [10] and [11] have been geared towards recognizing slight movements that make certain expressions personal. Zalewski and Gong [10] note that no two emotional expressions are alike. That is, no two people smile in the exact same manner because there are miniscule details that make each unique. Research that considers this individualization factor of unique facial expression has been lacking. Thus, a need for making facial expressions more personal has been established.

The individuality of expressions is also considered when developing avatars to mimic facial expressions [10]. However, using avatars to express individualized facial expressions has been lightly explored. In the literature consulted [12, 13, 14], the systems that incorporated avatars were used to help detect and classify facial expressions, not imitate and recreate them. This fact establishes novelty of the more specific field of using avatars to express recreated images. There have been limited advances in individualized facial expressions, and there is a need for personalized avatars in numerous applications [8, 10, 11, 15, 16]. Therefore, this thesis is geared toward avatars imitating a human's facial expression in a personalized fashion. It should be noted that an actual human subject was not used. Instead, the concept was tested using the OGRE avatar [8]. The target avatar face is used to recreate a unique facial expression that has the potential to mimic any human's unique expression. Thus, the target avatar face can be considered to be the equivalent of a photograph of a particular human's expression.

1.1.2 WEAKNESSES IN CURRENT MULTI-PARAMETER CONVERGENCE

The motivation behind this project was to establish a more robust method for the techniques currently used in personalized facial expression imitation in avatars [1]. The foundation of this thesis is based on the framework described in [1], where the authors approach the problem of multi-parameter convergence in an avatar by using the Particle Swarm Optimization algorithm. The multi-parameter convergence is based on the convergence of multiple slider values used to control the musculature of the avatar's face. Each parameter presents a slider used to control a specific set of Action Units on the avatar. As the particles converge, the slider values are systematically altered by PSO in an attempt to find the optimal solution. In this specific case, the optimal solution is the target face of the avatar depicting certain expressions [1].

The authors successfully applied the technique to five pre-defined facial expressions. However, a significant problem is exposed in the conclusion of the paper regarding the convergence of the features depicted in a fifth facial expression (the angry face) of the OGRE avatar [1, 8]. After running PSO using a population size of 50, the angry face did not converge to exhibit the proper features, despite a less than 1% error [1].

This problem exposes a limitation in the Particle Swarm Optimization algorithm. PSO is limited in its ability to diversify and branch out to other potential solutions in the search space once the swarm has started to converge [17, 18]. Although PSO may be a viable solution when optimizing parameters, it has the potential to get trapped in local minima. In turn, the algorithm may prematurely converge toward a local optimum, potentially leaving portions of the search space unexplored [17, 18, 19]. This is clearly the case in [1] when PSO prematurely converged to a local optimal solution and leaves the portion of the search space containing the true global optimal solution unexplored.

To avoid the premature convergence of PSO toward inadequate, local optimal solutions, many have paired PSO with other algorithms to form a superior, hybrid algorithm [12, 13, 18, 19, 20, 21, 22, 23]. To take advantage of PSO's global search and minimize the changes of premature convergence, the authors in [19] and [21] hybridize PSO with a local search technique known as Tabu Search [6, 7]. This hybrid model takes advantage of PSO's global search as well as Tabu Search's local search to determine the true global optimal solution within the search space [19]. The approach in this thesis also seeks to take advantage of these key strengths offered by each of these algorithms by forming a hybrid algorithm, based on previous research, geared to optimizing a set of parameters that depict an emotional state on an avatar [18, 19, 20, 21, 22, 23].

1.2 HYBRID APPROACH TO MULTI-PARAMETER CONVERGENCE

To increase the robustness and overcome the limitations of the method described in [1], this research sought to develop a hybrid Particle Swarm Optimization – Tabu Search algorithm that takes advantage of TS's local search techniques and PSO's global search to achieve a more efficient hybrid algorithm that can be directly applied to multi-parameter optimization. Along with a hybrid PSO-TS algorithm, a tool to compare the overall efficiency of these algorithms was also developed as part of this thesis. The OGRE Algorithm Comparison Tool (OACT), shown in Figure 2, serves as the test bed for the algorithms. The OACT was developed in a modular manner to allow for the addition of other algorithms. Each algorithm is used to translate a working avatar face (left) to match a target avatar face (right). Distinguishable facial features from the target avatar displaying an emotion, determined by a preset configuration of parameters, are continually compared to the working avatar face. Performing the analysis with a hybrid PSO-TS algorithm allows the parameters to be manipulated automatically to translate the depicted facial features of the target avatar onto the face of the working avatar. Thus, the avatar expresses the unique characteristics of the target avatar's facial expressions and avoids producing a generic emotional expression based on simply classifying the emotional state.

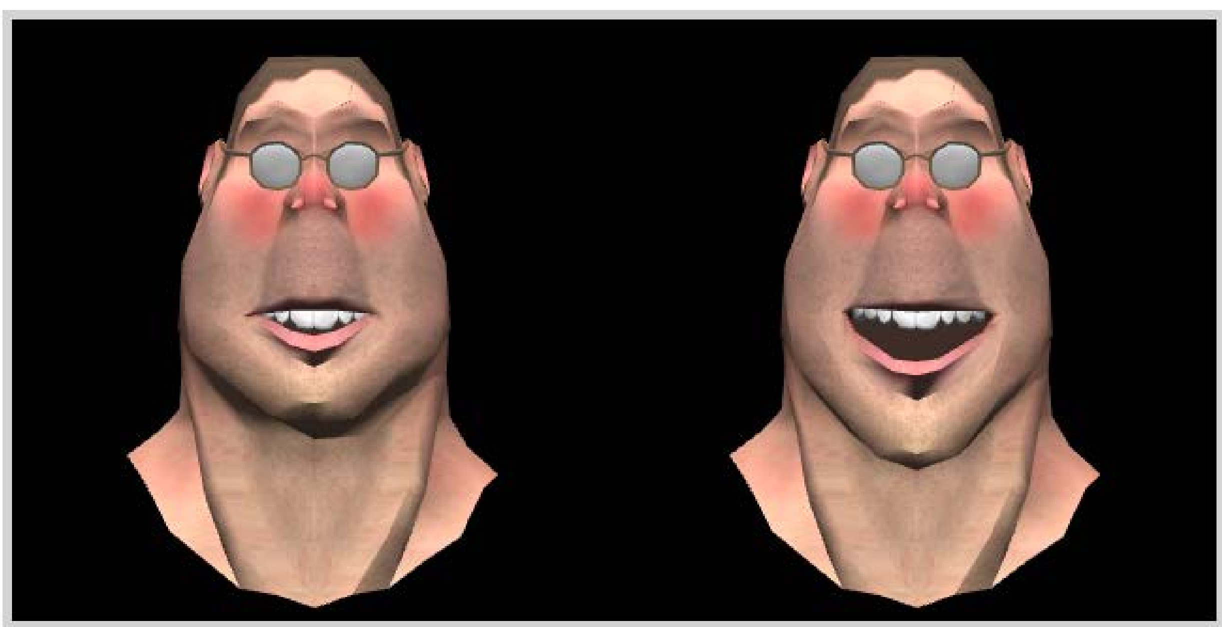


Figure 2: OGRE [8] Tool: Working Avatar (left) and Target Avatar (right)

1.3 HISTORICAL BACKGROUND

In order to provide a perspective of the long-term advances made in the field of facial recognition and expression detection, a brief history of its origins and progressions is necessary. Understanding these will also provide the reader with a perspective on how the basics of facial recognition and expression detection can be translated into facial expression imitation systems. The technical area of facial expressions is discussed first, followed by an in-depth review of attempts made in the past to extend the state of the art. The latter provides insight on what has already been accomplished as well as previous methods used to successfully achieve those goals. Issues and obstacles that have hindered further advancements are also discussed.

1.3.1 MEASURING FACIAL EXPRESSIONS: FACS

Ekman [24] gives an overview on the classifications of emotions and describes many psychological aspects of emotional expressions. The link between humans and expressions is established, providing information on how each individual expresses his/her current emotional state. It has been found that although cultures differ in many aspects, basic emotional expressions are, for the most part, all universally understood and expressed in the same manner [24]. A discussion of how to effectively measure and quantify emotions is presented and discussed in [24], whose authors claim that basic emotions are often expressed the same way cross-culturally. Therefore, a system can be set in place that can accurately quantify an individual's facial expressions based on their current emotion [24]. One such system, developed by Ekman and Friesen, is the Facial Action Coding System (FACS) [4].

In 1978, Ekman and Friesen [4] developed FACS to allow for a quantitative representation of facial expressions. At the time, the system was needed to further study emotional expressions and social interaction between humans [25]. Ekman and Friesen noticed that certain muscular contractions were the basis of rearranging the face in a way to express an emotional state. As depicted in Figure 3, the facial structure was decomposed into segments to monitor movements in specific portions of the face. Each section of the face is labeled and defined by a boundary. Further analysis of the structure and anatomy of the face provided insight on how to specifically map the muscular movement when expressing an emotion. Thus, they decided to base their system on these muscular contractions by pinpointing and tracking key locations on the face that encountered significant changes in the process of expressing an emotion [25].

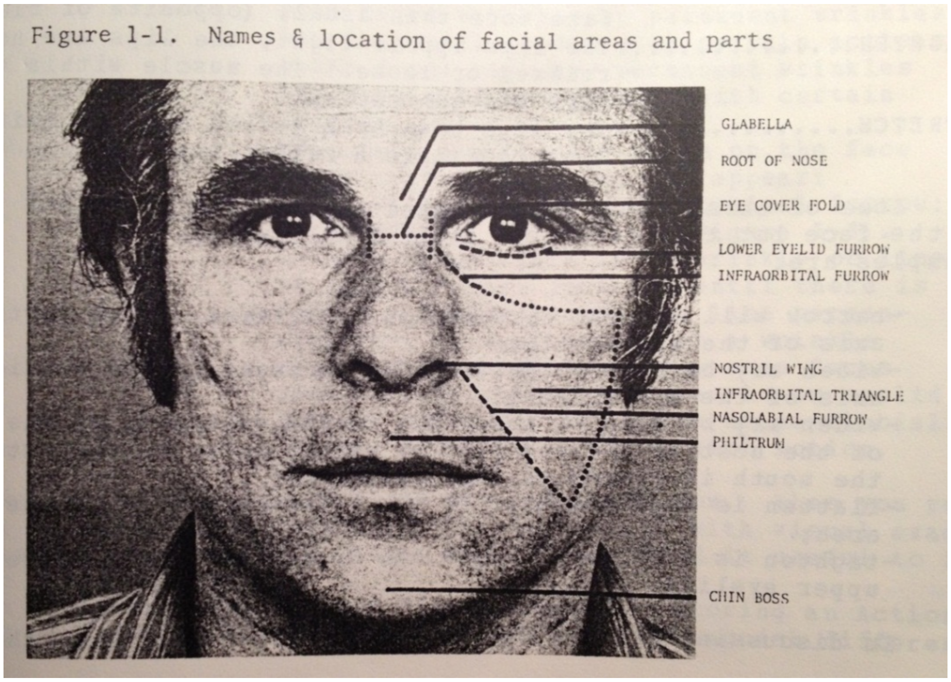


Figure 3: Facial Segments based on FACS (Reproduced without permission [4])

These specific locations, which accurately express the independent motion of features in the face, are more formally referred to as Action Units (AUs). The FACS system incorporates 44 AUs to accurately track and measure the facial differences encountered when expressing an emotion [15]. Figure 4 shows the upper AUs of the face and how those AUs are defined to track specified underlying muscles. Expert FACS coders thoroughly examine which AUs undergo changes when expressing certain emotions. Once these changes are noted, these expert coders are able to explicitly define which AUs were used to reproduce the expression so that FACS can correctly categorize the emotion [25]. FACS is based on six basic, universally understood emotions: anger, disgust, fear, happiness, sadness, and surprise [15]. On the basis of these basic emotions, FACS can, for the most part, be applied to multiple demographics to reduce the cross-cultural interdependency of the system.

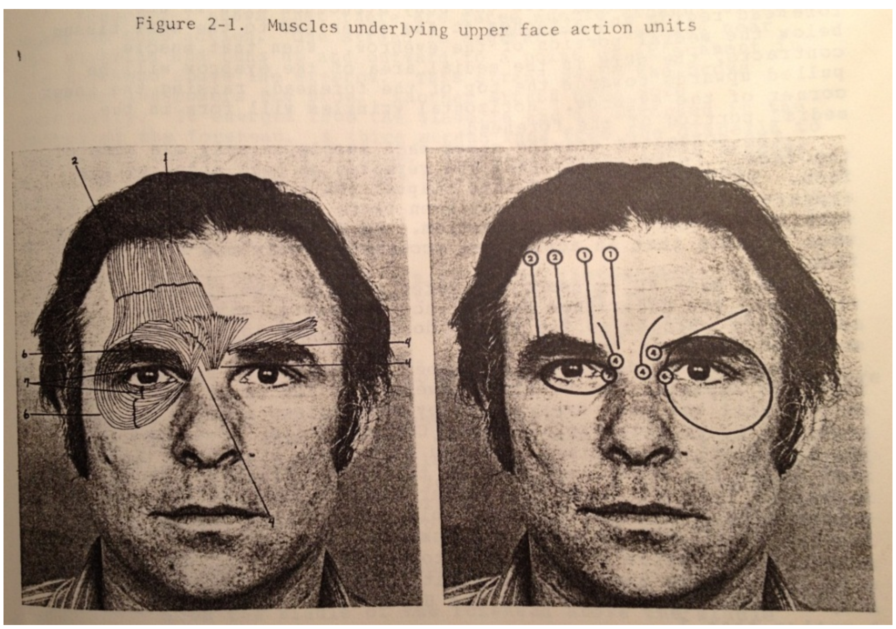


Figure 4: FACS: Upper Action Units (Reproduced without permission [4])

1.3.2 MAPPING ACTION UNITS

In order to correctly map the AUs defined in FACS, an algorithm is typically used to incorporate machine learning in the process. The use of this algorithm allows the mapping process to take place efficiently and effectively. An algorithm that has been found to be successful in search and optimization problems is Particle Swarm Optimization (PSO) [5]. Variations of PSO have been used in many applications that require an optimal solution [26]. In [26], modifications are made to incorporate PSO in emotion detection problems, which are directly related to the field of facial detection. Another search algorithm used in optimization problems is Tabu Search (TS) [6, 7]. Unlike PSO, Tabu Search is a local search algorithm that takes advantage of a memory structure to log solutions that were previously visited [6, 7, 19, 20]. Because of its local search technique and use of memory, TS has been hybridized with other algorithms to increase the robustness of certain applications [18, 19, 20, 21, 22, 23].

1.3.2.1 A BRIEF OVERVIEW OF PSO

PSO was developed in 1995 by Eberhart and Kennedy to simulate the actions of birds flying in a flock in search of food [5, 26]. It was developed as a population-based search algorithm. The algorithm essentially tracks potential solutions, more formally known as *particles*, as they swarm around the search space [26]. Adjustments are made to the positions of the particles within the search space based on their experienced successes [26]. Thus, the particles are said to be "flown" into a search space and swarm within the space, converging toward an optimal solution based on the successes of the interacting particles [5, 26]. PSO is discussed more extensively in Chapter 4 when the approach to the problem is described.

1.3.2.2 A BRIEF OVERVIEW OF TABU SEARCH

Glover developed Tabu Search in 1986 to aid in the optimization of difficult mathematical problems [6, 7]. TS is a meta-heuristic that takes advantage of a memory log, formally known as a Tabu List (TL), to help guide the search as it explores the search space. The TL is comprised of solutions that were previously visited by the search. As the search continues, the TL is consistently checked to ensure that previously visited solutions are not revisited. Thus, the search avoids becoming trapped in local optimal solutions and is able to converge toward the global optimal solution [6, 7, 19, 20, 23]. Tabu Search is more extensively examined when the approach to the problem is described in Chapter 4.

1.4 ACCOMPLISHMENTS IN THE FIELD

Since its origin, the field of facial recognition and expression detection has seen numerous accomplishments. Much research has been dedicated to optimizing the original methods and systems used in facial detection. This includes modifications to the original FACS [15] as well as adjustments made in algorithms to increase the accuracy and performance of facial recognition systems [10, 12, 26, 27]. The progressions of these former successes are discussed in the following sections.

1.4.1 PROGRESSION OF FACS

Much research has been done to improve FACS by adding minor modifications to the original Ekman and Friesen system [4]. These slight alterations have not effected or changed the original framework of the system. Rather, it has simply been enhanced to allow for a more robust system. The modifications discussed in the following sections have been used in formal systems based on FACS, and are not considered to be an in-depth, state-of-the-art review. The sections simply describe previous findings that have allowed the original version of FACS to continually improve.

1.4.1.1 AUTOMATED ACTION UNIT DETECTION

Much of the work using the original form of FACS has been geared towards automatically recognizing the AUs used in FACS. One such system is described in [16], where AUs are automatically mapped in a two-step process. Once a relationship is established between the AUs and the six basic emotions, the Longest Common Subsequence (LCS) is used to account for erroneously mapped AUs. Their method finds and compares AUs in organized sequences and LCS indicates which subsequences contain incorrectly mapped AUs [16]. Another system based on a minor modification of FACS is found in [27] where machine learning methods are directly applied to the FACS system to allow for more accuracy when attempting to automatically recognize facial expressions. One advantage of systems that automatically detect and track AUs is that an expert FACS coder is not needed to explicitly define the AUs in each of the considered images. Automated systems for detecting AUs have also focused on incorporating a cross-file format approach that incorporates both single image files as well as video files [28]. This approach is useful because many researchers are attempting to incorporate various types of inputs and formats to expand their system's use across numerous applications [28].

1.4.1.2 INTEGRATING 3D TECHNIQUES IN FACS

3D techniques have also been incorporated into FACS in addition to the traditional 2D approach [10]. Instead of using a strict 2D image, the 3D model is solely based on identifying and classifying the depth of the facial features [10]. Once the features are detected, their movements are mapped, in the same way AUs are, so a facial emotion can be classified [10]. These slight, yet, significant modifications to the FACS system allow for a continual expansion on a longstanding, rigorous system without drastically changing the approach. These early advances represent significant milestones in the evolution of FACS.

1.4.2 EVOLUTION IN TRADITIONAL ALGORITHMS

Advances in algorithms used in facial expression detection are required to incorporate a system that can support many detection methods and features. The advances in these algorithms, which have already been rigorously tested and proven to work well with emotion detection, allows for a more accurate and timely approach to this problem [26]. As for PSO [5], slight modifications were made to the original concept in [26] to directly apply an already successful algorithm to a facial expression detection system. This approach proved that traditional algorithms not already being used for this application can be modified and applied directly to expression detection. On the other hand, despite minute changes to apply the algorithm to specific applications, Tabu Search has remained rather unchanged. Instead of manipulating the original TS algorithm, many authors have decided to combine it with another algorithm to a form a hybrid model [18, 19, 20, 21, 22, 23].

Analyzing and focusing on these algorithms' key features has led to an attempt to form hybrid algorithms [12, 13, 18, 19, 20, 21, 22, 23]. The combination of such algorithms to form a more robust hybrid model has increased the performance of systems that were already in place [12]. By combining two or more algorithms, weaknesses in some methods can be masked by strengths in others. This allows the hybrid algorithms to increase the robustness and ensure the reliability of mechanisms applied to specific applications [18, 19, 20, 21, 22, 23]. These hybrid algorithms are extensively discussed in Chapter 2.

1.4.3 HANDLING NON-IDEAL INPUT

Considering that the majority of real-world applications in facial detection do not provide ideal conditions for the image or video, dealing with a non-ideal input is often a major reality. Minor, yet, important accomplishments have been made to improve recognition systems when considering a non-ideal image or video. For example, in [29], a cluttered background is considered in the input image. The authors note that clutter within the input image can cause the system to mistake other items included in the picture for facial features thereby causing incorrect detection and mapping of AUs [29]. The fact that the system described in [29] can handle clutter in the background of the test image is considered to be a substantial accomplishment. In [16], the development of a system that can handle high intensity illuminations is described. In the next section, illuminations and other obstacles in the field are discussed.

1.5 PREVIOUS ATTEMPTS AND THEIR LIMITATIONS

There are still major obstacles to overcome in facial recognition. Because of the many constraints placed on the system, it is hard to develop a robust system that can handle variable input. This varying input can come in the form of facial variation and imperfections as well as head position and tilt. Moreover, these variations are not limited to those listed.

Attempting to identify an emotion from an image that does not contain a frontal-face view has proven to be difficult. Efforts to account for slight head tilt and rotation have allowed some systems [10, 13, 14, 16] to properly handle these issues in small degrees. However, if the head of the subject is tilted or misaligned beyond a certain degree, some systems fail to correctly identify a face in the image [16]. Nonetheless, in order to properly identify a subject's face in a provided image, it must be assumed that the face is actually detectable. That is, if no feature of the face is identifiable, then the system cannot hope to perform properly.

Other factors that may pose a problem include the possible facial imperfections of the subject being considered. Facial blemishes pose a significant obstacle. The majority of the systems reviewed did not discuss how to handle facial imperfections, such as blemishes and wrinkles. In

[27], all of the selected test subjects were from a younger age range and had no such imperfections. These systems were tested with a sample population that is not representative of the entire population. Subjects from different backgrounds were also excluded from the sample populations described in [11], [12], [14], and [15]. In [15], the dataset on which their system was tested was composed of Caucasians and people of European descent. Therefore, failing to consider other demographics tainted their results.

1.6 SUMMARY

First, an overview of the key objectives of this thesis was stated, providing general information on the methods used in the process of reaching the long-term goal. Throughout this chapter, facial expression systems and algorithms, such as FACS, PSO, and TS, were discussed. A history of each was provided, and the way in which each operates was briefly defined. These are described in detail in Chapter 4. The modifications made and progressions taken to improve and expand upon the original systems were explained to provide insight into the state of the art. Moreover, additional accomplishments along with potential obstacles were described. This provides a perspective on what has been done, and which holes are left to fill in facial expression recognition and imitation.

Further information is provided in the following chapters that describe the process this thesis takes to reach the end goal. In Chapter 2, a more in-depth, state-of-the-art review is conducted to describe the current work in the field of personalized facial expression and the imitation of those expressions in an avatar. Chapters 3 and 4 establish the problem definition and the specific steps involved in the approach used to address the problem. In Chapter 5, a prototype is described using the steps explained in Chapter 4. In Chapter 6, testing and evaluation of the prototype developed is discussed, followed by a summary of the process in Chapter 7. Chapter 7 discusses conclusions made from the observations and results obtained. Future research is also discussed to allow future research to expand on the findings of this thesis.

CHAPTER 2: LITERATURE REVIEW

To better understand the accomplishments and advances that have been made in facial recognition and imitation, a state-of-the-art review is necessary. The following sections provide the reader with a description of the state-of-the-art and exposes some of the gaps found in the research. This discussion verifies that the approach presented in this thesis is novel and makes a significant advancement. It also justifies the need for a realistic, human-like avatar that can imitate personalized facial expressions, as part of our long-term goal.

2.1 RELEVANT ASPECTS TO BE REVIEWED

There are three significant aspects of the approach presented in this thesis: 1) using the Particle Swarm Optimization (PSO) algorithm [5] and Tabu Search [6, 7] to map Action Units (AUs) on the avatar, 2) adding a personalization factor to facial expressions to clearly distinguish one person's expression from another in an avatar, and 3) recreating those expressions in an avatar. The next sections provide an in-depth examination of recent research that focuses on the three factors previously mentioned. PSO and TS are extensively analyzed in Chapter 4 when the approach to the problem is discussed. It should be noted that not only are state-of-the-art processes and approaches described in the following sections, but potential gaps in the research are also discussed to give the reader perspective on which aspects of the field still remain open.

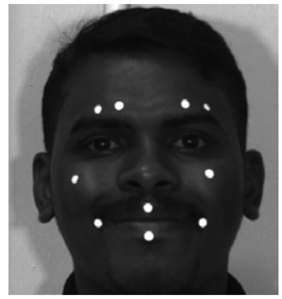
2.2 PARTICLE SWARM OPTIMIZATION ALGORITHM

Recent systems [12], [13], [26] have been successful in directly applying PSO to mapping AUs to detect and, in some cases, imitate facial expressions. While some authors attempt to apply the original PSO algorithm directly [26], others reconfigure and manipulate PSO to develop hybrid versions that meet the specific requirements of their system [12], [13]. Systems that incorporate PSO are examined in the proceeding subsections to gain perspective on state-of-the-art methods.

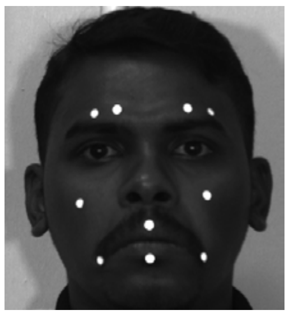
2.2.1 GUIDED PARTICLE SWARM OPTIMIZATION

Ghandi, Nagarajan, and Desa [26] modify the original PSO algorithm to incorporate a method that can be directly applied to emotional detection. The modified algorithm, Guided Particle Swarm Optimization (GPSO), identifies and maps the movements of small luminous markers that are placed on the test subject's face. As depicted in Figure 5, the luminous markers are placed in the same positions as the defined Action Units. Once the markers are in place, the algorithm is run on a segmented video, and each swarm continually updates the particle's positions with each iteration. Each swarm is continually compared to a database to classify the resulting facial expression. Once the algorithm has found a matching target, it classifies the emotional state of the subject in the video and announces success [26].

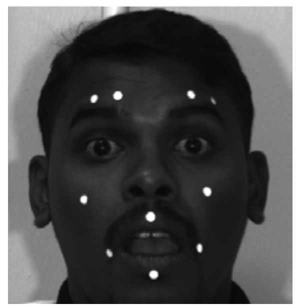
Neutral



Happy



Sad



Surprise

Figure 5: Positions of AUs in different emotions (Reproduced without permission [26])

This approach [26] is unique in that the positions of the AUs are known prior to the execution of the algorithm. Thus, the GPSO algorithm can arrive at an accurate solution in a computationally effective manner because it has a predetermined knowledge of the placement of the AUs. The combination of the two methods that make up the GPSO algorithm are 1) having the AUs represent one of the particles in each of the swarms and 2) applying a modified velocity equation. The GPSO algorithm allows the particles to be guided towards the path of the AUs in a more efficient manner, which allows for a more accurate detection of facial expressions [26].

The GPSO algorithm had success rates ranging from 85% to 95% when detecting three of the six basic emotions: happy, sad, and surprise [26]. Subjecting the algorithm to only three of the six basic emotions could potentially be the reason for such a good detection rate. Thus, the high efficiency of the algorithm claimed by the authors could be skewed and/or misinterpreted. Failing to incorporate all six basic emotions limits the applications of this system. This includes the recreation of expressions in a potential avatar, which is important to the approach presented in this thesis. Not considering all six basic emotions and failing to incorporate an avatar to recreate facial expressions are two major differences between this approach [26] and the work of this thesis. Another potential weakness of the system in [26] is that only a small number of Action Units are used. This limits the potential of adding a personalization factor because slight muscular movements in the face may not be detected.

2.2.2 SWARM INTELLIGENCE OVERVIEW

The previous subsections provide insight and perspective on how recent research has manipulated and used algorithms to provide unique approaches to facial detection. It should be noted that the recent discussion concerning the PSO algorithm, as in [26], and the hybrid methods used in [12] and [13], is solely to describe the use of algorithms to map Action Units to classify a facial expression. The specific applications in which the systems are used (i.e. facial recognition) are not the sole focus of this section. Instead, focusing on the algorithms and methods used to compose hybrid algorithms provides concepts that can be applied to imitating a facial expression. Focusing on the combination and optimization of the algorithms allows the systems discussed in [12] and [13] to increase the robustness and detection speed of the systems. Analyzing the previous systems that incorporate the use of PSO, exposes gaps in the approaches. The approach in this thesis fills these gaps by applying specific algorithms to imitating facial expressions in an avatar.

2.3 TABU SEARCH

Although there have not been any particular papers that have directly applied TS to facial recognition systems, a number of systems [30, 31, 32, 33] have used Tabu Search for parameter optimization. Other systems [18, 19, 20, 21, 22, 23] have focused on the hybridization of TS with other algorithms to increase the robustness of their systems. The following sections analyze systems that used Tabu Search in parameter optimization as well as hybrid TS methods.

In [31] the authors focus on the global optimization of parameters by applying a form of Tabu Search. The system follows the basic approach of Glover's original TS method [6, 7], but focuses on exhaustively searching throughout the entire search space, as opposed to a single search within a local neighborhood. First, a diversification procedure takes place to find the areas of the search space that contain the most promising solutions. Once this diversification process is complete, TS searches within each predefined neighborhood in a traditional manner, seeking the optimal solution of each neighborhood. Once solutions are found in each neighborhood, they are compared globally to ensure the best solution throughout the entire search space is found [31].

The authors of [32] take a similar approach of applying Tabu Search to system geared toward global optimization. However, instead of applying their methods solely to a set of discrete parameters, they focus on the optimization of both discrete and nonlinear settings. Historically, Tabu Search was typically used for discrete applications, which made it difficult to apply the search to continuous sets of data that were not discretely decomposed [6, 7]. In [32], the focus is on adjusting the TL and TS to account for nonlinear sets of data and coupling TS with different scatter and directional searches to solve nonlinear problems. Much like that of [31], this combination of scatter and directional searches allows the technique in [32] to search throughout the entire search space and locate the global optimal solution in a nonlinear system.

The optimization of nonlinear parameters, as discussed in [31, 32], are directly incorporated in the approach presented in this thesis. However, a more robust method can be developed by hybridizing TS with another algorithm to increase the performance of this system. In the following section, state-of-the-art hybrid methods are analyzed to provide the reader with perspective on how the combination of multiple algorithms can increase the robustness of a system.

2.4 HYBRID METHODS

The combination of algorithms, to form a more robust hybrid model, has increased the performance of systems [12]. Much effort has been geared to developing hybrid methods by combining two or more algorithms [12, 13, 18, 19, 20, 21, 22, 23]. These hybrid algorithms allow the weak aspects of one algorithm to be compensated by the strengths of another. This allows for a more robust and reliable mechanism readily applicable to a specific application [18, 19, 20, 21, 22, 23]. In the following subsections, state-of-the-art systems that utilize hybrid methods are analyzed.

2.4.1 HYBRID TABU SEARCH – PARTICLE SWARM OPTIMIZATION

Many authors have developed hybrid PSO-TS algorithms that take advantage of PSO's global search technique and TS's local search to efficiently search throughout the entire search space for the optimal solution [18, 19, 20, 21, 22, 23]. The authors of [19, 20, 23] focus on optimizing their system by masking PSO's weakness of premature convergence by hybridizing PSO with TS.

The hybrid TS-PSO developed by the authors of [20] takes advantage of PSO's global search as well as TS's local search in each iteration to significantly increase the chance of finding the global optimal solution. The flowchart in Figure 6 provides an overview of each iteration of the algorithm. During each iteration, the particle search population is randomly divided into two subpopulations. One half of the population updates the position and velocity of each particle via PSO, while the other half of the population searches locally for the best solutions using TS. After each half is updated, the two subpopulations are recombined, and the particle best and global best particles are updated. The Tabu List is also updated to contain the solutions that have already been visited within the search space [20]. The hybrid TS-PSO algorithm's success rates were higher than those of the Genetic, Particle Swarm Optimization, and Tabu Search algorithms [20]. The authors further test their algorithm on Cluster Analysis in another paper [23]. They use the same hybrid model, but apply the algorithm to real world problem of clustering. The results resemble those of their original paper [20] and on average their hybrid TSPSO algorithm had a lower computation time than the Genetic Algorithm and Combinatorial Particle Swarm Optimization when applied to Cluster Analysis [23].

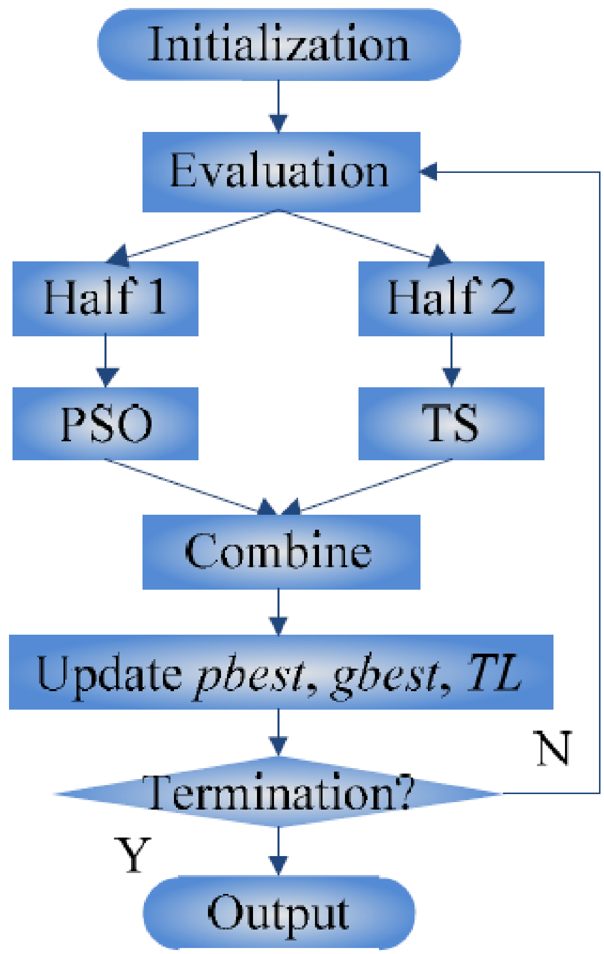


Figure 6: TS-PSO Flowchart (Reproduced without permission [23])

The authors of [19] take a similar approach to [20] and [23] when hybridizing PSO and TS. However, rather than merely combining the original PSO and TS algorithms as in [20] and [23], the authors [19] modified both PSO and TS before forming their hybrid model to directly apply their algorithm to a thermal problem called the T-junction problem. To avoid PSO's premature convergence toward low quality local optima, two Tabu Lists are implemented. The first Tabu List is used to handle the infeasible solutions encountered throughout the search. That is, if the new solution obtained by the search does not meet the certain criteria required, it is added to the TL of infeasible solutions and not revisited. The second TL is used to help diversify the search by finding local optimal solutions in local neighborhoods while PSO searches globally throughout the entire search space. Diversifying the search by using the second TL is what allows the hybrid algorithm to find the true optimal solution because it forces the algorithm to perform local searches within PSO's global search. This technique prevents PSO from prematurely converging toward poor quality optimal solutions [19].

2.4.2 HYBRID TAGUCHI-PARTICLE SWARM OPTIMIZATION

The Hybrid Taguchi-Particle Swarm Optimization (HTPSO) algorithm [12] combines the fast searching and solution finding aspect of Particle Swarm Optimization (PSO) with the iterative procedure of the Taguchi method. The HTPSO algorithm uses facial texture and surface information to detect different faces and their expressions. In the proposed method, the process is started by defining the parameter factors and determining how much each parameter influences the performance characteristic. Once the definitions are set, the algorithm takes in both 2D and 3D images and attempts to recognize the face in each image, regardless of the person's current facial expression. The procedure used to incorporate 3D images in the proposed system [12] can be manipulated and applied to other systems. Specifically, it can be applied to the approach described in this thesis, which seeks to make use of 2D avatars to recreate facial expressions. Although the approach is novel in the sense of combining two methods to form an improved hybrid algorithm, the approach fails to incorporate any specific applications involving the detection or recreation of facial expressions in an avatar. Not being able to apply the system [12] to a specific application involving the recreation of facial expressions limits its use outside of facial recognition. The system also fails to incorporate any emotional state detection, which is often a common application in facial detection systems. Because these factors are ignored, the HTPSO algorithm is restricted to basic facial recognition systems.

2.4.3 HYBRID ANT COLONY AND PARTICLE SWARM OPTIMIZATION

The innovative approach discussed in [13] describes a hybrid algorithm that combines the Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms to identify six basic facial expressions in 3D images. The classifications of the six emotions are based on Ekman's [4] six prototypical emotions used in his Facial Action Coding System. The hybrid algorithm takes advantage of ACO's ability to solve shortest distance problems and PSO's swarm intelligence to accurately and efficiently classify facial expressions [13]. The process resembles the hybrid integration used in [12]. However, unlike the approach used in [12], the method used by combining ACO and PSO is specifically applied to facial expression recognition rather than simply facial detection. Although the hybrid concept and properties of each of the algorithms is similar, the application used by each individual system is different and unique.

This approach [13] incorporates a reliable method to recognize 3D images using face depth information and rules established by both ACO and PSO. Both PSO and Ant-Miner, a modification of the original ACO algorithm, are used to establish facial emotion classification rules by which the emotions are mapped. The classification rules discovered throughout the process are expressed in the form of “if-then” rules [13]. The consequent of each rule represents a classification of one of the six possible facial expressions, while the antecedent is represented by three types: attribute, operator, and value. The rules are stored sequentially, and the first valid rule that fires is what actually classifies the facial expression. If no rule is triggered, a default class classifies the expression [13].

The approach described in [13] combines two traditional algorithms to further increase the success rate and obtain a more efficient classification of facial expressions. The fact that the system is capable of handling 3D images provides assurance that the system can be modified to incorporate an avatar or other 3D rendering to not only be used to recognize facial expressions, but also recreate them. Although the main focus of this paper is on 3D images, failing to incorporate 2D images in the approach limits the applications of the system. To make the system more robust, the authors can incorporate an analysis of 2D images to reconfirm the conclusions made by the 3D analysis. Neglecting to incorporate facial expression imitation in the approaches previously discussed in [12] and [13], exposes gaps in the research. The approach presented in this thesis seeks to fill those gaps by attempting to recreate facial expressions in an avatar while indirectly applying some of the same concepts expressed in these systems [12], [13].

2.5 PERSONALIZATION OF FACIAL EXPRESSIONS

The main aspect of the approach presented in this thesis is the personalization of facial expressions. Many of the systems previously discussed, such as those described in [12] and [13], fail to consider that each human expresses his/her emotions and feelings in a unique manner [4]. That is, although many expressions are generic in nature, there are always some minute details that distinguish one person's expression from that of another. Though the majority of facial expressions can be generally classified, a personalization factor allows for a more realistic reconstruction and imitation of an individual's emotional state. Failing to consider this personalization factor defaults into a generic, "cookie-cutter" reconstruction and imitation of a person's facial expression.

The following subsections expose gaps in the research by discussing systems that fail to identify unique characteristics that make facial expressions unique. Personalization is needed to establish realism when interacting with avatars resembling human beings. Otherwise, the generated representation of that person is bound to be unconvincing, leading to an unnatural interaction with the avatar. The following subsections seek to discuss, in detail, the methods used to account for more personalized facial expressions in their systems. However, it should be noted that the majority of the systems described in this chapter, do not focus on the recreation of said expressions. Instead, they simply attempt to recognize and classify them. Nonetheless, the approaches and methods used are important to further understand and expose the gaps remaining in facial expression imitation.

2.5.1 PERSONALIZATION USING ACTION UNITS

Cosker, Krumhuber, and Hilton [15] present a novel Dynamic 3D FACS Dataset (D3DFACS), based on the original Facial Action Coding System. The dataset consists of 10 test subjects each contributing between 19 and 97 different Action Unit-based video segments, for a total of 519 AU sequences. Samples of the D3DFACS dataset can be seen in Figure 7. The top two rows show the six different camera views from six of the 10 participants. The bottom two rows illustrates the 3D mesh data in both its textured and untextured forms, and show the corresponding UV texture maps [15].

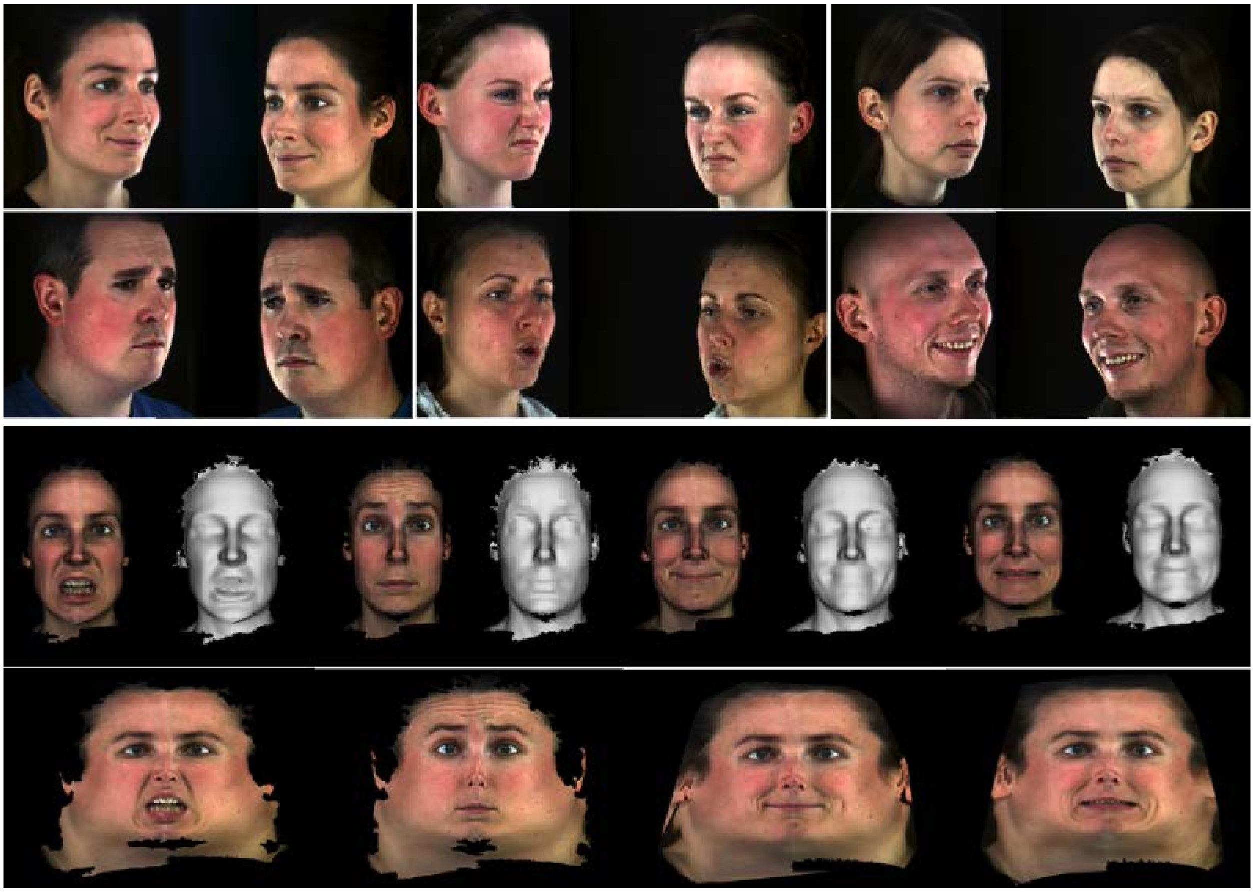


Figure 7: Examples from the D3DFACS dataset (Reproduced without permission [15])

After each human test subject was given basic information on the Facial Action Coding System, they expressed each of the six prototypical facial expressions. Each facial expression was recorded by six different cameras within five to ten seconds, depending on the intricacy of the emotion expressed. The test subjects were instructed to express each emotion naturally with slight head tilt so the dataset was capable of testing illuminations and slight variations in head position [15]. However, the actual, finalized dataset is not of significant importance in this state-of-the-art review. Instead, the process and approach used to record and map each of the 519 facial expressions is of interest.

To make each expression as unique as possible, the path that each AU took throughout the expression was recorded and monitored for a more intricate tracking [15]. This made each facial expression unique, deviating from the generic, “cookie-cutter” emotional expressions. Expert FACS coders were used in the mapping of the emotions to explicitly define the AUs prior to each recording. By defining each AU based on each individual’s facial structure rather than on a generic placement, the resulting emotional expressions were unique and personalized [15]. This unique process can be indirectly applied to the approach described in this thesis by strictly defining the AUs based on each test subject’s facial structure.

One of the differences between [15] and the approach presented in this thesis is that an expert FACS coder is required to explicitly define the individual AUs before each test in [15] Although this method allows for the maximum personalization of each expression, it is not a time-efficient approach. Having the coders define each AU also limits the use of the system in other applications that do not have access to expert FACS coders. The method described in this thesis takes into account that each person’s facial structure is different than others. However, it does not employ an expert FACS coder to define the individual AUs for each test subject. Nonetheless, having each AU intricately mapped to allow for a unique expression is important to the work presented in this thesis.

2.5.2 ERRONEOUSLY MAPPED ACTION UNITS

Another system that uses Action Units associated with the Facial Action Coding System to add a personalization factor is described in [16]. It maps multiple detected AUs to the six basic emotions in a similar way as the system described in [15]. The specific AUs considered in this system are defined and depicted in Figure 8. Each frame in the figure shows how the AUs defined on a facial structure can be manipulated with certain muscular contractions associated with specific emotional states.

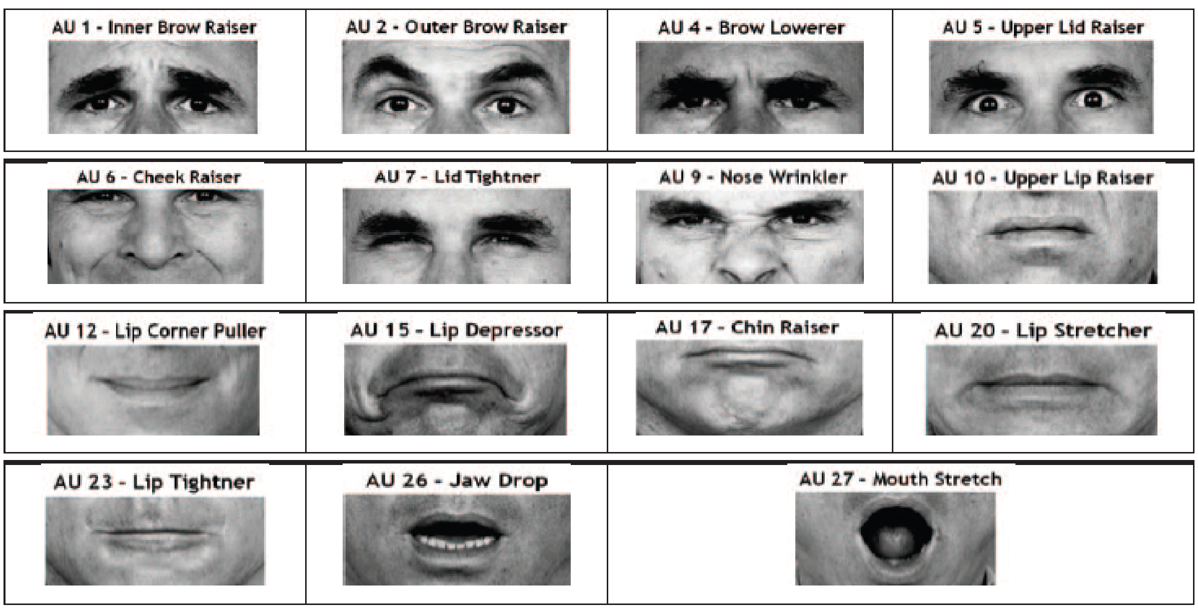


Figure 8: Visual samples of facial Action Units (AUs) (Reproduced without permission [16])

Unlike the system in [15], the method used in [16] is more rigorous because it takes a two-step approach to account for noise in the form of mismapped AUs. In the first step, a statistical analysis, performed on the mapping of the expression, establishes a relationship between the defined AUs and the six emotions being tested. In the second step, the Longest Common Subsequence (LCS) is used to determine the similarity between the strings formed in step one. The use of LCS accounts for incorrect AUs, caused by illumination, which may lead to an incorrect mapping of an emotion [16].

The difference between the system previously described [16] and the approach described in this thesis is the use of LCS to eliminate incorrectly mapped AUs. Although the approach in this thesis accounts for erroneous AUs, it does not incorporate the use of LCS. Instead, mismapped AUs are handled manually, rather than automatically. Eliminating incorrect AUs is still necessary, especially when considering all of the AUs necessary to incorporate a personalization factor. It should also be noted that unlike [16], this thesis focuses on the recreation of facial expressions rather than on their recognition.

2.6 RECREATION OF EXPRESSIONS IN AN AVATAR

The systems [10], [11], [14] reviewed in the following subsections focus on methods that incorporate the use of an avatar to make facial recognition systems more robust. They do not focus on the recreation of facial expressions in an avatar. Nonetheless, portions of the methods analyzed in [10], [11], and [14] can be applied to the approach presented in this thesis to imitate facial expression in a human-like avatar.

2.6.1 EXPRESSION RECOGNITION USING AN AVATAR

A unique approach to generating emotional expressions in an avatar is presented by Yang and Bhanu [14]. The authors developed an approach that encompasses the creation of an Emotion Avatar Image (EAI). Instead of directly attempting to recognize a facial expression given a segment of video, the authors create an EAI. Figure 9 provides a flow chart that illustrates the process. As depicted in the flow chart, each EAI serves as a clear representation of each frame throughout the video, which can be used as a static image. The use of local binary pattern (LPB) allows for a description of the textures involved in each frame. In the next stage of the flow chart, both LPB and local phase quantization (LPQ) are used to extract features from the EAI. Once the features are extracted from the EAI, they are compared to an avatar reference face using the SIFT flow algorithm. In short, a temporary EAI is created for each frame of the reference video. Then, each EAI is compared to the reference avatar, and the SIFT flow algorithm makes note of the differences in the faces and classifies the emotion [14].

Unlike the system proposed by Yang and Bhanu [14], the system described in this thesis uses an avatar to recreate facial expressions, not classify them. In order to further understand this approach, the process in [14] can essentially be reversed. Instead of taking the avatar and classifying the emotional state, the system described in this thesis takes the emotional state and imitates that emotional expression in an avatar. Using a form of an EAI to convert the segment of video into static images can help accomplish the reconstruction of the expression. The EAI allows the recreation process of the expression to be segmented into rudimentary steps by focusing on one EAI at a time.

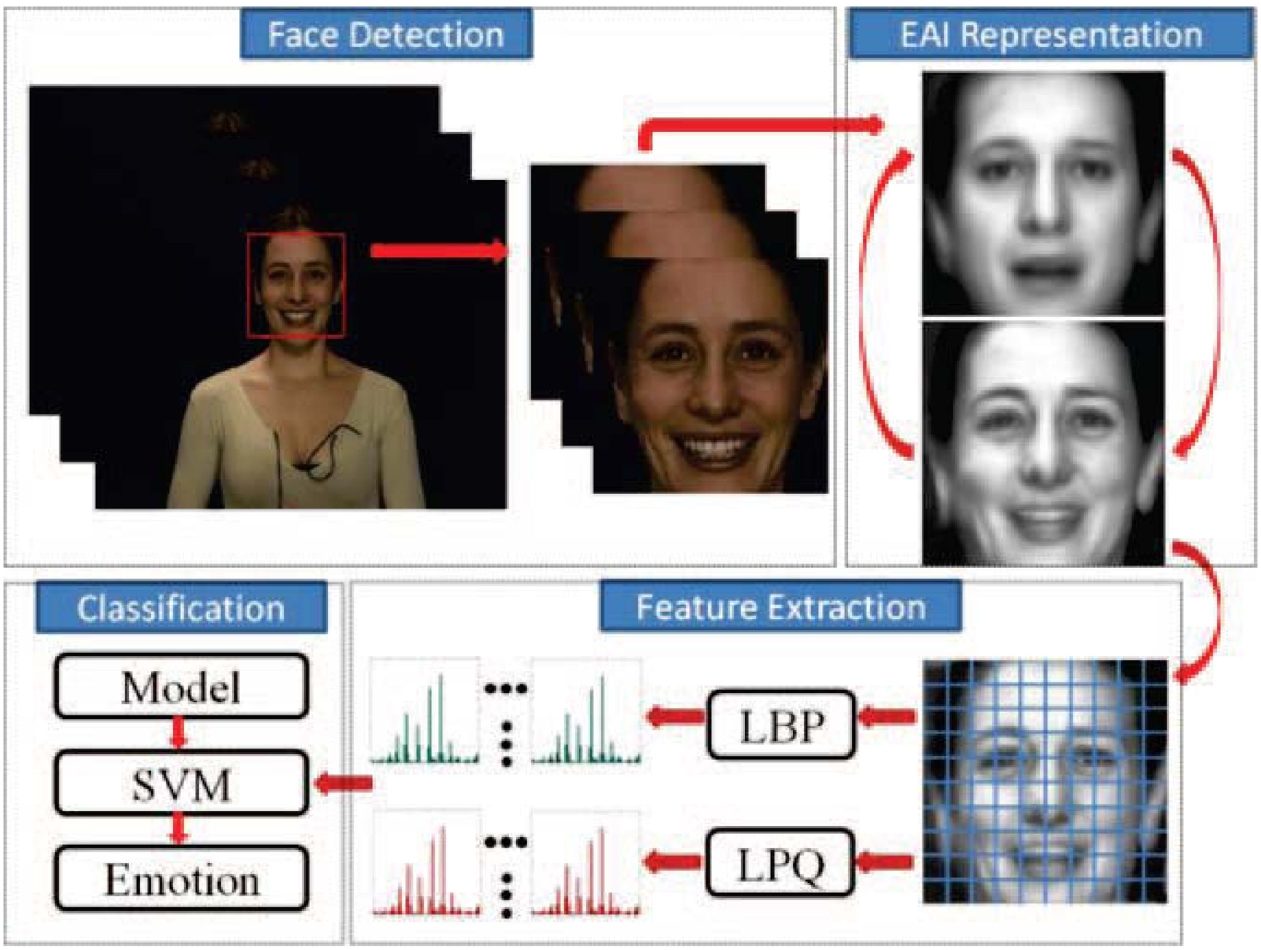


Figure 9: Overview of the detection process (Reproduced without permission [14])

2.6.2 GENERATION OF EMOTIONAL STATES

The system in [11] incorporates an avatar that expresses emotions in a natural environment (i.e. a daily conversation). The authors describe how emotional facial expressions can be expressed in an avatar using utterances as an input. That is, the avatar attempts to take on the emotion expressed in a given sound bite. The paper focuses on the naturalness of the emotion expressed in the avatar. Rather than having a set number of basic, “cookie-cutter” emotional expressions, the system [11] adds a personalized aspect in the recreation of the emotion by considering unique features of the face. To accomplish this personalization of expressions, the authors take into account the subtle differences in expressions. This avatar is not only capable of expressing basic emotions, but it can also convey subtle emotions that are often difficult to explicitly categorize. The use of the Facial Action Coding System allows the system to map and recreate the identified expression in the avatar. The authors manually controlled 30 Action Units to achieve desired emotional expressions. The method used to control the AUs is illustrated in Figure 10. The values expressed on the sliders, located underneath the image of the avatar, represent the intensity of each subcategory (defined on each side of the sliders). Manually manipulating the AUs in this manner allows the user to create a more realistic and personalized facial expression in the avatar.

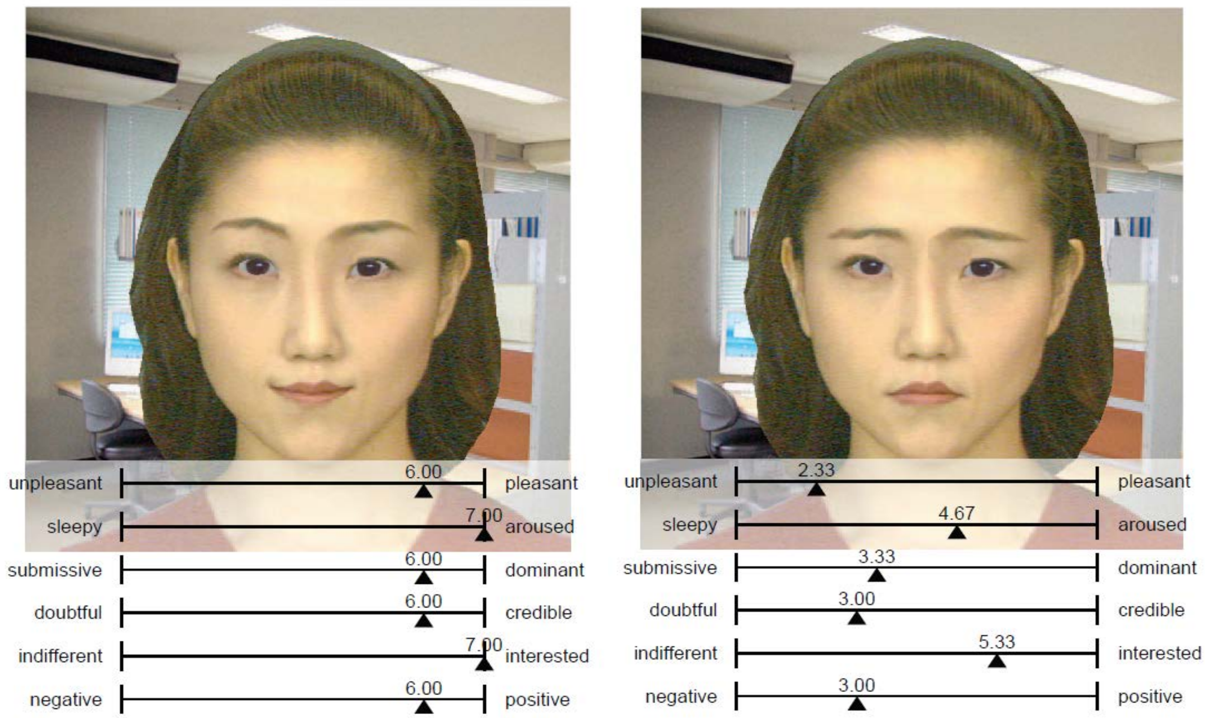


Figure 10: Expression generated from an emotion (Reproduced without permission [11])

The use of sliders to manually control the facial expressions of the avatar are considered in our work when attempting to personalize the expressions. The sliders used in the system described in this thesis are not limited to a small number of subcategories as in [11]. Instead, sliders for each mouth shape and individual emotional state are manipulated to provide the user with full control of the position of each AU and the musculature of the avatar's face. A major difference between the approach discussed in this thesis and the system in [11] is the use of utterances as an input. The tone of the speaker signifies his/her emotional state, and the system attempts to express that emotion on the avatar's face. To accurately and holistically define an emotional state, more factors should be taken into consideration. The authors [11] make the assumption that certain phrases are expressed in the same pitch and tone by all people. That is, one subject expresses his/her anger in a voice identical to that of another. The audible inputs can also be garbled and misinterpreted, which further constrains the system's potential. This limits the accuracy and robustness of the system [11]. Therefore, audible inputs were not considered in the scope of this thesis.

2.6.3 SMOOTH TRANSITIONS IN FACIAL EXPRESSIONS

To ensure a realistic interaction, the individuality of facial expressions must continually be considered when attempting to transfer the emotional expressions from a segment of video to a human-like avatar. It is also important to retain a lifelike expression while transitioning from one emotional state to another. Zalewski and Gong [10] describe a hierarchical approach where the face under consideration is broken down into its key components. In the first level, the low level component Action Units are analyzed to provide a general mapping of the face. In the second level, logic is used to combine subcomponent information, extracted from the lower level, to model the final expression. As opposed to a holistic recognition approach, the proposed system in [10] accounts for minute changes that allow for a more personalized facial recreation. Once the facial expression is broken down into its basic structures, the system attempts to model the face in a 3D avatar. Examples of the generated avatars, based on the test subject's face, are provided in Figure 11. The frame number and severity of the expression (in percentage) are expressed at the bottom of each frame. In order to properly examine the face and define its significant parts, key features of the Facial Action Coding System are used, which directly resembles a portion of the method described in this thesis [10].

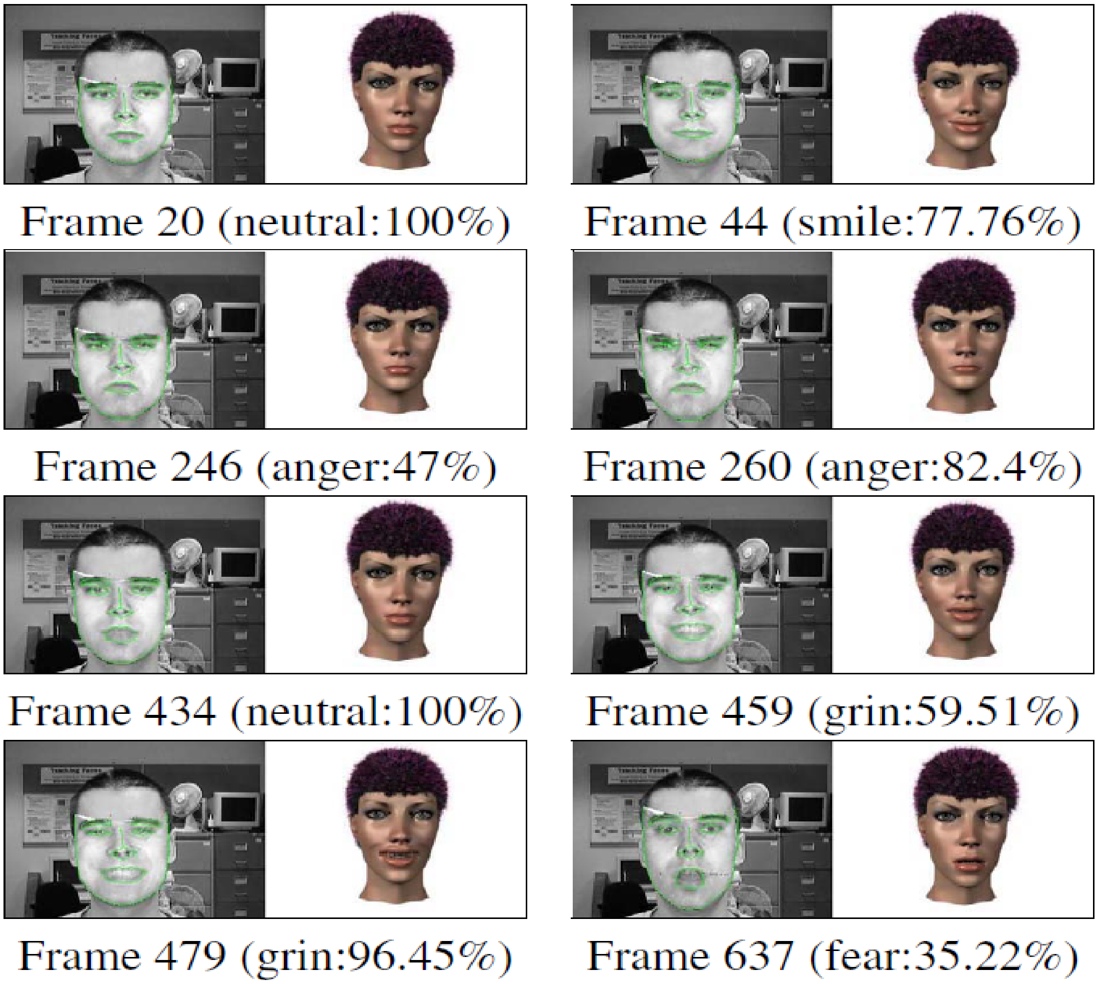


Figure 11: Sample frames of expression classification (Reproduced without permission [10])

The authors directly focus and address the issues associated with a realistic and smooth transition from one expression to the next [10]. They note that in order to achieve a realistic animation in an avatar, smooth gradual changes are required in the transition phases. To procure this smooth transition in their system, they produce a set of Rate of Change (ROC) curves that depict the casual transition from one emotion to another. The ROC curves are analyzed during the transition from one expression to another, and if sudden spikes are noticed in the curves, a smoothing filter is applied to make the transition smoother [10].

The long-term goal presented in this thesis focuses on the personalization of each facial expression and incorporates a smooth transition from one expression to the next. The methods used in [10], specifically the ROC graph, can be indirectly applied to our long-term goal to help monitor the realism of the transition process. However, one of the major differences in the long-term approach presented in this thesis and the authors' [10] approach is that our work does not seek to recognize a human face in real time. Instead, the values corresponding to the Action Units are inputted manually by the authors of [10]. These values are then translated to a human-like avatar. Nonetheless, the process described [10] to transition from one emotional state to the next is important to successfully imitate a realistic facial expression in a human-like avatar.

2.7 SUMMARY

First, the three topics relevant to the approach presented in this thesis were discussed. These topics are as follows: using the Particle Swarm Optimization algorithm and Tabu Search to map Action Units when using the Facial Action Coding System, adding a personalization factor to facial expressions to clearly distinguish one person's expression from another, and recreating those expressions in an avatar. Throughout this chapter, multiple state-of-the-art systems and their methods were discussed to provide background on the material necessary to gain an understanding of modern systems that resemble the methods of the work presented in this thesis.

Methods that could be directly or indirectly applied to this work were discussed in detail, stating the advantages of using portions of these techniques. Descriptions of the limitations, differences, and potential gaps were provided to ensure novelty in the approach presented in this thesis. After reviewing the literature, it is clear that no such system exists that imitates personalized facial expressions in a human-like avatar. The work presented in this thesis fills this gap by expanding on the methods discussed in this state-of-the-art review. The following chapters establish the problem definition and describe the specific steps involved in the approach to the problem.

CHAPTER 3: PROBLEM DEFINITION

This chapter defines both the general and specific problems related to this thesis. Along with concisely describing the problems, a hypothesis is stated. The contributions listed provide the reader with a perspective on the impact of our work that can be used by others. Finally, a discussion of novelty, significance, and usefulness concludes the chapter.

3.1 GENERAL PROBLEM

Historically, classifying facial expressions has been a continual challenge within computer vision and, more specifically, facial recognition systems [34]. Holistic classification of facial expressions requires a consistent method of measuring the facial structure and movements [34]. Ekman and Friesen's Facial Action Coding System (FACS) [4] provides a consistent method of dissecting and measuring the muscular structure of the face. The FACS system measures muscular contractions of the face by using Action Units (AUs). The AUs pinpoint and track important locations on the individual's face that experience significant changes throughout the process of expressing an emotion. With a neutral face as a reference, the displacement of the AUs can be measured to classify the expression of the human [4].

Once the facial expression can be quantified, its particular features can be translated onto an avatar. The slight variations in these features are what make the expression personal and allow for a more realistic recreation when mapped onto the avatar. Thus, the general problem of this thesis is transferring personalized facial expressions of a specific human being onto an avatar intended to represent that individual. It should be noted that in this thesis the OGRE [8] avatar is used in place of the photograph of the human subject as well as the human-like avatar to serve as an intermediate, proof-of-concept test.

3.2 SPECIFIC PROBLEM

The specific problem addressed in this thesis is the limitations of PSO when applied to a multi-parameter optimization problem. The weaknesses of PSO are exposed in [1] when the parameter optimization mechanism fails to accurately converge to the target face of the avatar. In the specific case of converging to the "angry face" in [1], PSO continually fails to find the optimal solution within the search space because it gets trapped in local minima. The premature convergence of PSO inhibits the algorithm from searching throughout the entire search space. Instead, the algorithm assumes that it has found the global optimal solution and satisfies the stopping criteria [19, 20]. Thus, the specific goal of this thesis is to establish a more robust multi-parameter optimization method that utilizes a form of PSO, paired with another algorithm, to properly converge to a global optimal solution.

3.3 GENERAL APPROACH

To incorporate machine learning in the translation process, an algorithm is used to map the changes in the AUs [5, 6, 7, 26]. Eberhart and Kennedy's Particle Swarm Optimization (PSO) algorithm [5] tracks potential solutions in the search space. Particles are "flown" into the search space, and each particle swarms throughout the search space looking for a solution. Because the particles communicate with each other based on their successes, they converge toward the optimal solution. Using PSO in conjunction with FACS allows the AUs to be efficiently mapped to an expression [26].

However, because of the limitations of PSO, as described above in the Specific Problem section, it must be paired with another algorithm to avoid prematurely converging to local optima. Thus, PSO will be hybridized with Glover's Tabu Search [6, 7] to avoid PSO's premature convergence toward local optima. This will combine PSO's global search with TS's local search technique to ensure the entire search space is thoroughly examined, and the true global optimal solution is found [19, 20].

3.4 HYPOTHESIS

The hybrid PSO-TS algorithm will overcome the limitations of the PSO algorithm used in prior research and will perform the translation by avoiding PSO's premature convergence to local optima.

3.5 CONTRIBUTIONS

The following are tangible contributions offered by the research presented in this thesis:

- A new hybrid method that combines PSO and TS to imitate personalized facial expressions in an avatar through machine learning.
- A prototype that demonstrates the system's ability to learn an individual's expressions and mimic those expressions in an avatar.
- Data that can be used by others to compare results.
- A report containing a description of the process, materials, methods, and results found while conducting this research.

3.6 CRITERIA FOR ACCEPTABILITY

To warrant a thesis, the specific problem must be novel and significant. Furthermore, from an engineering perspective, the results stemming from this research must also be useful. In the proceeding subsections an overview of the consulted literature is provided to justify novelty, and the difficulty and usefulness of our approach is discussed. The examination of these three factors ensures our work meets the criteria for acceptability.

3.6.1 NOVELTY

A literature review was conducted to determine how other authors approached similar problems. The majority of the literature focused on facial recognition systems [10, 11, 12, 13, 14, 26] that were able to classify the six basic facial expressions [4]. In order to classify facial expressions, some systems [12, 13, 26] used the Particle Swarm Optimization algorithm, while other systems [10, 11, 14] used an avatar as an intermediate step in the classification process. The technical literature reviewed exposed the gaps that remain in the research that this thesis fills.

A common theme found in papers that took a similar approach to our work was the use of the Particle Swarm Optimization algorithm. Directly applying modified PSO algorithms to the problem of facial classification is seen in [12], [13], and [26]. Few attempted to approach the problem by using a hybrid algorithm to increase the robustness of previous methods [19, 20]. Other reviewed approaches [10, 11, 14] demonstrate solutions that incorporate the use of an avatar as an intermediate step in the classification process. These approaches [10, 11, 14] generated a generic, “cookie-cutter” expression in an avatar to aid their systems in accurately classifying the emotional state of the human test subject based on the expression depicted in the avatar.

However, the sole focus of these systems [10, 11, 12, 13, 14, 26] is on the classification of facial expressions. They do not attempt to recreate personalized facial expressions in an avatar, as described in this thesis. After consulting the technical literature, no systems were found that resemble the problem addressed in this thesis.

3.6.2 SIGNIFICANCE

Our work utilizes a complex, hybrid optimization algorithm in combination with the Facial Action Coding System to automatically reproduce a personalized facial expression in an avatar. A significant amount of effort is required to meet this goal. This process involves the complicated integration of multiple methods to extend the state-of-the-art. Because the process described is far from trivial and the results from this research make a significant contribution to the field, it can be stated that the difficulty of this work warrants an Honors in the Major thesis.

3.6.3 USEFULNESS

A useful method to imitate personalized facial expressions in an avatar through machine learning is provided. From a short-term perspective, a prototype demonstrating the system's ability to learn and individual's expressions and mimic those expressions in an avatar can be used by others to further their own research. Using personalized human avatars in computerized applications and games can use the results of this research to provide a more realistic interaction with the user. In the long-term, applications that involve virtual humans, potentially used to replace absent professors or employees, can use this research.

CHAPTER 4: APPROACH

Now that the literature has been thoroughly reviewed and the problem has been described, the approach used to imitate personalized facial expressions in an avatar through a hybrid Particle Swarm Optimization – Tabu Search algorithm is discussed. This chapter extensively discusses the approach required to solve the specific problem of this thesis which, in turn, serves as an intermediate step to solving the general problem. First, PSO and TS are extensively discussed to describe the definitions and equations used in the methods and implementations presented in this thesis. The conceptual background and theory of each algorithm is also discussed. Second, the purpose behind developing a hybrid algorithm is stated. Limitations and strengths of the two algorithms are discussed, and an approach to overcoming their weaknesses is presented. Third, the theory behind the hybridization of the two algorithms is explained to provide the reader with an understanding of the overall architecture of the hybrid model described in this thesis. Finally, a general discussion follows to explain how the method is integrated with our OGRE Algorithm Comparison Tool (OACT) to solve the problem described in this thesis. The specifications and parameters used in the implementation of the algorithms and OACT are discussed in Chapter 5.

4.1 EXTENSIVE EXPLANATION OF ALGORITHMS EMPLOYED

An algorithm is conceived to implement an automated machine learning method to map Action Units in a mechanism that utilizes the Facial Action Coding System. The Particle Swarm Optimization algorithm has been found to be successful in global optimization problems that include a large search space [5, 26]. Like PSO, Tabu Search has also been used in systems to successfully optimize a set of parameters [6, 7, 19, 20]. To better understand the underlying concepts behind PSO and TS and how they are incorporated in this thesis, an in-depth description of each algorithm is provided next.

4.1.1 IN-DEPTH ANALYSIS OF PSO

The purpose behind PSO is to develop a population-based search algorithm that can efficiently search throughout the entire search space for the global optimal solution. PSO is a stochastic global search technique that simulates the actions of birds flying in a flock in search of food [5, 26, 35]. That is, a group of individual agents, more formally known as particles, are "flown" into the search space and collectively swarm around looking for the optimal solution [5, 26]. As the particles search for the global optimal solution, they continually communicate with each other. Adjustments are made to the positions and velocities of each particle based on their experienced successes while searching. Thus, the entire swarm of particles continually converges toward an optimal solution based on the successes of the interacting particles. Because each particle updates its position and velocity based on its own successful experience, as well as the successful experiences of its neighbors, it is said to use the concept of social interaction while problem solving [26, 36]. The overall concept behind PSO is to have numerous individual agents collaboratively searching for the same optimal solution within a search space while continually communicating with the group to update the swarm on the progress of the search [5, 26, 36]. This characteristic of PSO makes it appealing to use in our system because of the multi-dimensional search space of our problem.

In the basic PSO algorithm, as used in this thesis, the swarm is made up of *i* particles in the search space. Each particle is composed of a position *x* and a velocity vector *v*. The position of the individual particle represents the potential solution in the *N*-dimensional search space [5, 23, 36]. For example [23], the position and velocity of the *i*th particle are represented as

$$x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iN}) \tag{1}$$

$$v_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iN}) \tag{2}$$

where position *x* and velocity *v* are composed of N-dimensional components. The flowchart in Figure 12 helps describe the architecture and flow of PSO.

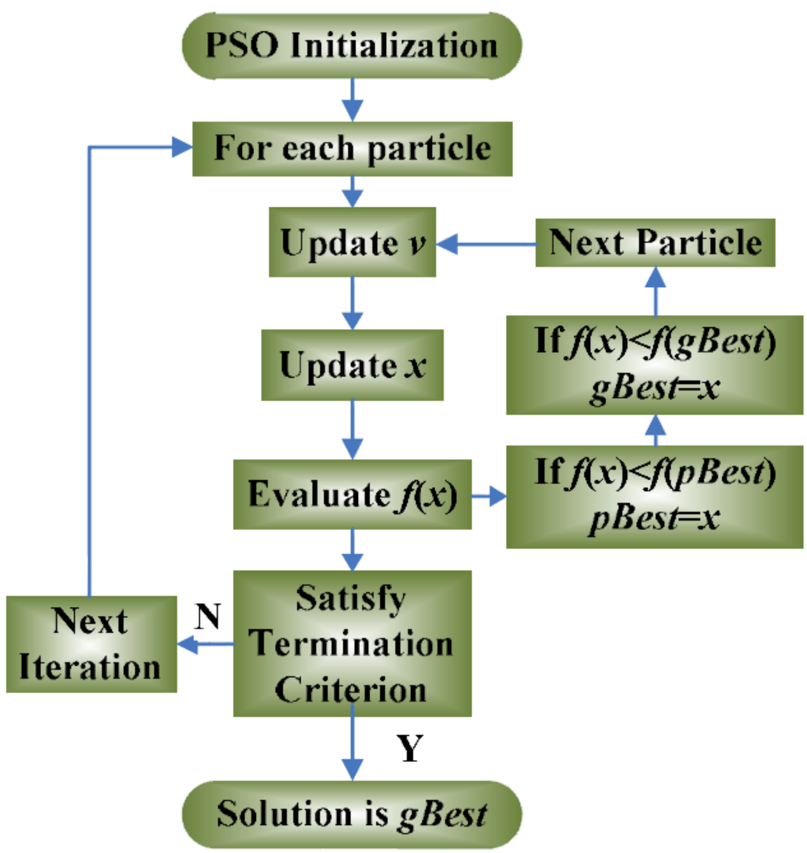


Figure 12: PSO Flowchart (Reproduced without permission [23])

The positions of the particles are randomly initialized to candidate solutions within the search space [23]. Each particle adjusts its position and velocity based on its personal experience and the experience of the surrounding particles. The adjustments are made based on the local and global optimal positions [36]. These positions are more formally referred to as *pbest* and *gbest*. *Pbest* refers to the *personal best* of the particle thus far in the search, while *gbest* refers to the *global best* of the entire swarm of particles. Both *pbest* and *gbest* are based on a fitness function used to rate the solutions found in the search space. The solutions are only updated if a better solution is found by the search. Throughout each iteration, the particles communicate their fitness with each other, and the individual *pbests* of all the particles are compared to determine the *gbest* of the entire swarm. At the end of each iteration, *gbest* represents the best fitness that any particle experienced throughout that single iteration. Each particle then adjusts its velocity based on the values of *pbest* and *gbest*. The update is determined by the fitness, distance, and a predefined "inertial" parameter as shown in equations (3) and (4) below [1, 5, 18, 26, 35, 36]. The search procedure of PSO is depicted in Figure 13 where the particles' movements are described. The modified velocity is influenced by the original velocity, the velocity towards *pbest*, and the velocity towards *gbest*. The original positions of the particles, marked x_i and x_j, are then updated to x_{i+1} and x_{j+1} based on the modified velocity. The formulas used to update the new velocities v_{new} and positions x_{new} of the swarm in this thesis are adapted from [1] and [35] and are defined as

$$v_{new} = C_0 * v_{previous} + C_1 * R_1\left(x_{gbest} - x_{previous}\right) + C_2 * R_2\left(x_{pbest} - x_{previous}\right) \qquad (3)$$

$$x_{new} = x_{previous} + v_{previous} \qquad (4)$$

where C_0, C_1, and C_2 represent constant values used as a inertial (weighting) factor, R_1 and R_2 are random numbers between 0 and 1 used to "swarm" the particles, x_{gbest} is the global best position of the swarm, x_{pbest} is the personal best position of the individual particle, and $v_{previous}$ and $x_{previous}$ represent the particle's previous velocity and position, respectively. The swarm is updated each iteration in this manner until a stopping criteria is met, at which point the search is terminated and *gbest* is returned representing the optimal solution found in the search space [23, 35, 36]. The specific methods used to implement PSO into the mechanism described in this thesis are discussed in Chapter 5. This includes a discussion of the constant and random values as well as the confined ranges of the equations.

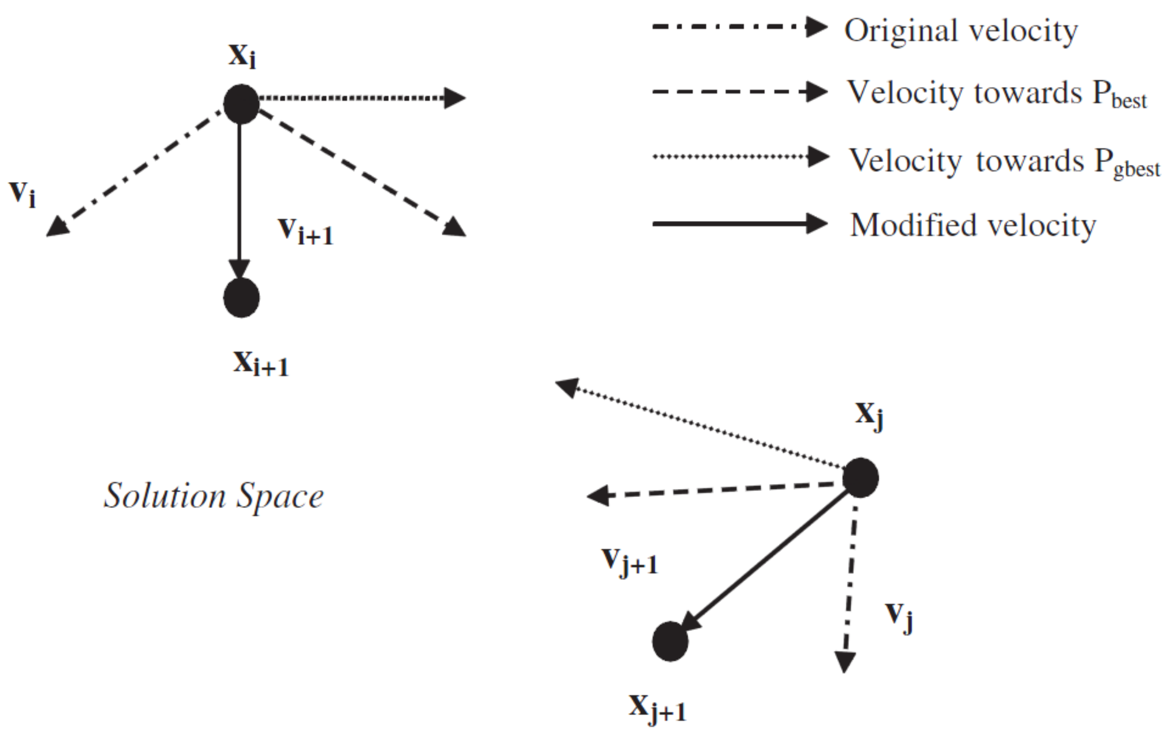


Figure 13: Search Procedure of PSO (Reproduced without permission [18])

4.1.2 IN-DEPTH ANALYSIS OF TS

Unlike PSO, TS is a local search technique that utilizes a memory structure to keep track of its search history. The purpose behind Tabu Search is to avoid repeatedly checking candidate solutions in the search space as to avoid getting stuck in local optima. This local search technique allows for an efficient means of searching for the solution throughout the search space by iteratively examining nearby regions. Varying from its intended use in mathematics, many systems have incorporated TS to provide an efficient means of converging toward an optimal solution within a search space. This includes iteratively adjusting a set of parameters to find the optimal solution to the problem, which can be directly applied to the problem described in this thesis [6, 7, 19, 20, 23, 30, 31, 32, 33].

It should be noted that the Tabu Search implemented in this thesis does not make use of a set of equations as described in section 4.1.1 above. Therefore, an algorithm with a set of equations is not provided to describe TS. Instead of using complex equations, the local search technique of TS is based on examining nearby neighborhoods. This meta-heuristic algorithm, as implemented in this thesis, searches in a local neighborhood N within the search space. The search proceeds by moving from solution s to solution s' in N [23]. To keep track of the solutions visited in N, Tabu Search takes advantage of a memory log to keep track of its search history. This memory log, more formally known as a Tabu List (TL), helps guide the search as it explores the search space for the optimal solution. The TL is constantly updated to keep track of previous solutions that have been visited in the past i iterations, where i is the number of previous solutions stored, more formally referred to as the Tabu Tenure [23]. Throughout the course of the search, the TL is checked to ensure a past solution is not revisited. Therefore, the search avoids becoming trapped in local optima and converges toward the true global optimal solution. The overall concept behind TS is to use a memory structure to keep track of the search history to avoid becoming stuck in a cycle of repeatedly checking previously visited solutions [6, 7, 19, 20, 23, 30, 31, 32, 33]. The specific methods used to implement TS into the mechanism described in this thesis are discussed in Chapter 5. This includes a discussion of the parameters and memory structure associated with the Tabu Tenure and Tabu List as well as a description of how the N neighborhoods are defined.

4.2 PURPOSE OF HYBRID PSO-TS ALGORITHM

In this section the strengths and weaknesses of PSO and TS are discussed to point out the key elements required to develop the hybrid PSO-TS model described in this thesis. Previous attempts of optimizing a set of parameters have exposed weaknesses in both PSO and TS [1, 19, 20, 23]. One of these methods [1] is extensively analyzed to determine exactly what causes the PSO algorithm to fail in a specific application. To overcome these limitations, the modifications made to these algorithms to form a hybrid PSO-TS algorithm are provided.

Although PSO can successfully search throughout a large search space and find an optimal solution, it often prematurely converges toward local optima. This means the algorithm often is subject to partial optimization in that it does not find the true global optimal solution. Instead, it finds the local optimal solution within the swarm [1, 19, 20, 23, 36]. A specific example of this premature convergence that directly relates to the problem presented in this thesis is seen in [1]. The authors' mechanism in [1] uses PSO to map the Action Units on the OGRE [8] avatar using much of the same process presented in this thesis. The PSO mechanism successfully converges to the optimal solution for the happy, sad, fear, and surprise test faces [1]. However, the weakness of PSO is exposed when PSO fails to accurately converge to the target angry face of the OGRE avatar [1, 8]. In the specific case of converging to the angry face in [1], PSO continually fails to find the optimal solution within the search space because it gets trapped in a local minimum. The premature convergence of PSO inhibits the algorithm from searching throughout the entire search space. Instead, the algorithm finds a local optimal solution that satisfies the stopping criteria and causes PSO to terminate, despite never finding the true optimal solution [19, 20].

The purpose of developing a hybrid PSO-TS algorithm is to overcome the limitations of PSO and establish a more robust multi-parameter optimization method geared toward facial expression recreation in an avatar. To overcome the limitations associated with PSO's premature convergence in [1], PSO is hybridized with Tabu Search. As previously mentioned, Tabu Search is a local search algorithm that makes use of a memory structure known as a Tabu List. The Tabu List is used to diversify the search by ensuring previous solutions are not revisited. This diversification process makes TS an appropriate choice to pair with PSO. By forming a hybrid PSO-TS algorithm, PSO's premature convergence is overcome by TS's local, diversifying search method. Likewise, TS's limitation in only finding solutions in nearby neighborhoods is masked by PSO global search technique [19, 20]. Thus, the weaknesses of both algorithms are masked by the strengths of the other. This hybrid technique ensures the entire search space is thoroughly examined, and the true global optimal solution is found.

4.3 HYBRID PSO-TS ARCHITECTURE

This section provides a general overview of the theory behind the hybridization of PSO and TS. A discussion of why each algorithm is applicable to the mechanism described in this thesis is provided. The general architecture of the algorithm is discussed to provide the reader with an understanding of how the two methods are combined. The actual implementation and an in-depth discussion of architecture are provided in Chapter 5 to further explain the design of the hybrid PSO-TS algorithm.

The hybrid PSO-TS algorithm is formed by combining a slightly modified form of PSO with a standard Tabu Search. The basic architecture of the algorithm is depicted in Figure 14 below. The flowchart provides a top-level overview of the architecture of the algorithm. As depicted in the flowchart, PSO serves as the driving force of the PSO-TS algorithm. Unlike other models where the search population is randomly halved and TS and PSO are independently operated [20, 23], this PSO-TS approach embeds TS within PSO. TS is implanted in PSO to serve as a local search technique that helps diversify the search. The diversification of the search helps PSO avoid its limitation of premature convergence toward local optima. The PSO algorithm drives the overall search and begins the process by randomly searching throughout the solution space.

PSO explores the entire search space for the optimal solution. Once initial *pbests* and a global *gbest* are obtained, the *pbests* are passed to Tabu Search to explore the nearby area of the swarm. TS takes in these *pbests* and establishes a local search boundary centered on each *pbest*. The search examines the nearby area for a potential better solution. If a better solution is found, TS returns this updated solution to PSO. PSO then updates the swarm based on the best solution found thus far. The *pbest* and *gbest* along with the particles' velocities and positions are updated. The process continues in this manner until the stopping criteria is met. Overall, the combination of PSO and TS to form a hybrid PSO-TS algorithm joins the strength of PSO's global search with TS's local search to help diversify the search to overcome PSO's premature convergence to poor quality local optima [19, 20].

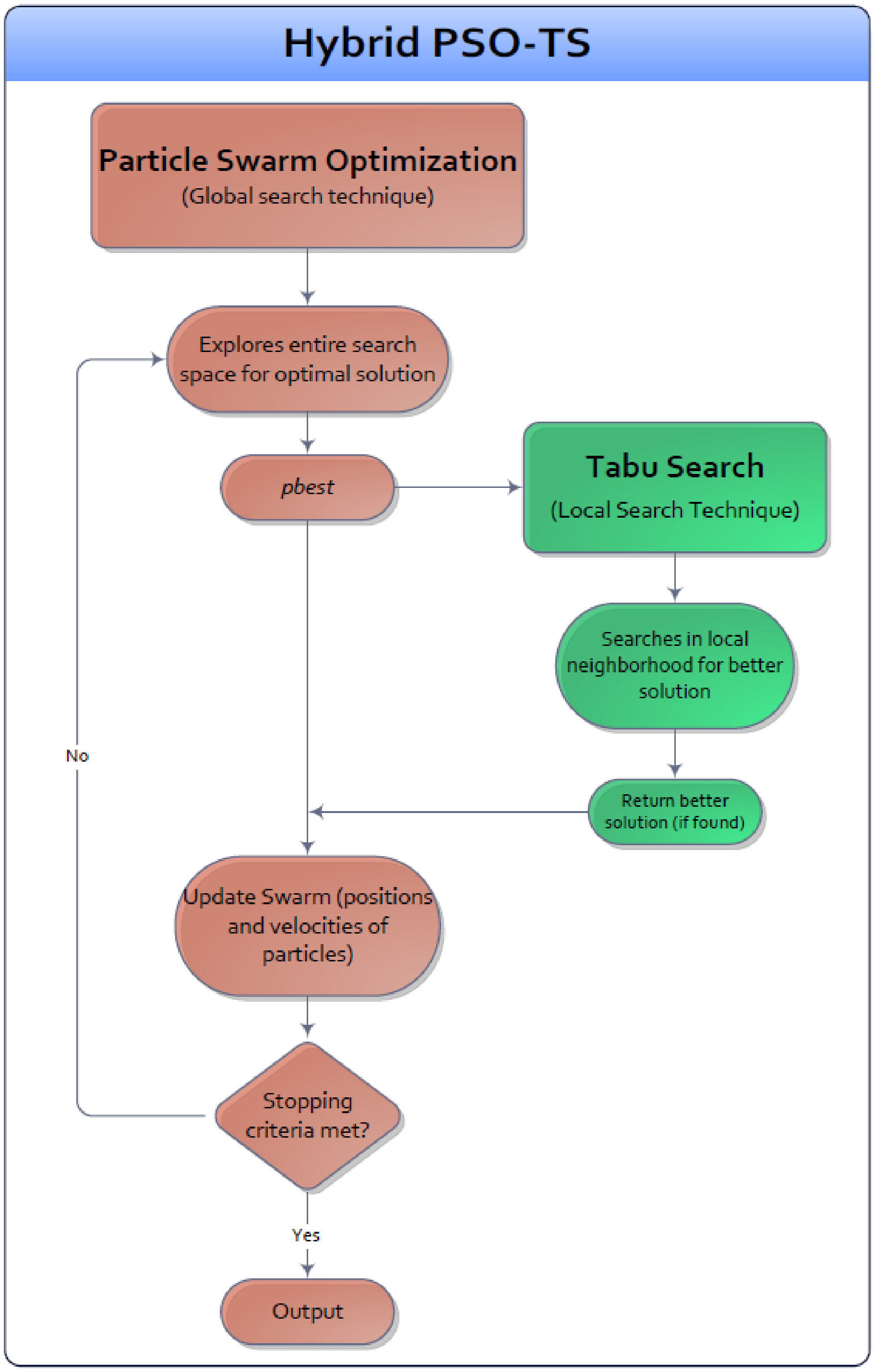


Figure 14: Top Level Flowchart of PSO-TS

4.4 GENERAL OVERVIEW OF OGRE APPLICATION FOR PSO-TS

A general discussion to explain how this hybrid method is directly applied to the OGRE Algorithm Comparison Tool ensues. This section provides a brief overview on how the PSO-TS algorithm and OACT are used to solve the problem presented in this thesis. An overview of the purpose of OACT is provided. The exact implementation and in-depth analysis of the architecture of the integrated system are provided in Chapter 5.

To rationalize the use of the hybrid PSO-TS algorithm applied to the long-term goal of imitating a personalized facial expression in a human-like avatar using machine learning techniques, an intermediate step utilizing the OGRE [8] avatar is taken in this thesis. This intermediate step uses a WebGL tool that utilizes OGRE models and textures. In OACT, the "target" face of the avatar is used in place of a photograph of a human displaying a facial expression. The personal characteristics of this photograph can be manually translated onto the target avatar face via the sliders that control the musculature of the OGRE face [8]. The "working" avatar face then attempts to translate the specific features of the target face by using a pixel-by-pixel comparison. The algorithm continually adjusts the parameters of the working face to vary the musculature of the avatar until the optimal solution has been found. In this case, the optimal solution is a match between the target and working faces of OACT [8]. An extensive discussion of this OGRE tool is provided in Chapter 5. This discussion includes specific definitions of the tool as well as the integration process of combining the OGRE avatar and the hybrid PSO-TS algorithm.

CHAPTER 5: PROTOTYPE AND IMPLEMENTATION

This chapter describes the implementation of the hybrid PSO-TS algorithm and the OGRE Algorithm Comparison Tool (OACT) as presented in the previous chapter. The implementation of this prototype was completed in two main phases. In the first phase, the OGRE facial models and textures were extracted from the downloaded OGRE demo [8]. These were used to develop the OACT WebGL application as previously described in Chapter 4. The basic PSO algorithm, as described in [1], was integrated into OACT. In the second phase, the hybrid PSO-TS algorithm was implemented and integrated into the OGRE tool. An extensive description of these phases, including the implementation of OACT and PSO-TS, are provided in the following subsections.

5.1 PHASE I DEVELOPMENT

In Phase I of development, the basic architecture of the OGRE Algorithm Comparison Tool was formed. The same PSO algorithm used in [1] was implemented to ensure the previous research reported in [1] could be reproduced in the tool we created. The basic slider animations that control the Action Units of the OGRE avatar were reproduced in our tool. All basic functionalities of OACT were checked to make sure it correctly resembled and functioned as the original OGRE Facial Animation demo [8]. In the proceeding subsections, the steps taken in Phase I development are explained in detail.

5.1.1 OGRE ALGORITHM COMPARISON TOOL

To begin the Phase I development, the OGRE Official Demos Distribution v1.7.0 (Windows) was downloaded and extensively examined to understand the underlying methods used to manipulate the Action Units that controlled the OGRE avatar's facial structure [8]. A screenshot of the OGRE Facial Animation demo is depicted in Figure 15. The OGRE face is centered in the middle of the screen with 18 sliders available in the upper left-hand corner. Once the "Manual" option is selected, the OGRE face stops its automated movements and provides the user with full control of the facial musculature of the avatar. The sliders are individually labeled to describe their corresponding action. For example, in Figure 15 the sliders for "Happy," "A," and "O" are manipulated to imitate a smile in the avatar. It is important to note that the "Happy," "Sad," and "Angry" sliders control multiple Action Units on the avatar face. To be more specific, when adjusting any of the "Expressions" sliders, multiple positions of the face are manipulated, as opposed to a specific portion of the face. On the other hand, the "Mouth Shapes" sliders represent specific phonemes in the avatar and only control a small number of Action Units on the OGRE face. These sliders can be used to adjust intricate details of the avatar's face to represent a personalized facial expression. To reiterate, the "Expressions" sliders can be used to setup the face in a generic facial emotion, and the "Mouth Shapes" can then be manipulated to fine-tune the intricate details of the face to allow for a more personal facial expression.

Figure 15: OGRE Facial Animation Demo [8]

The overall architecture of the downloaded Facial Animation demo was transferred via JavaScript to a WebGL application to allow for the same manipulation of the OGRE face via a set of 18 sliders. Our OACT provides a similar layout to the Facial Animation demo. As in our tool, the authors of [1] also based their PSO mechanism on the OGRE Facial Animation demo. However, unlike that of [1], the OGRE Algorithm Comparison Tool developed in this thesis is not directly embedded in, or dependent on, the OGRE demo. Instead, our tool is an independent, standalone WebGL application that is not directly reliant on the underlying functionality of the OGRE Facial Animation demo.

When developing OACT, the facial models and textures used in the Facial Animation demo were directly implanted in the WebGL application to ensure the same OGRE avatar face was used. Unlike the OGRE Facial Animation demo and the mechanism used in [1], our tool uses two OGRE avatar faces. These faces are set in a horizontal format to allow the application to provide real-time feedback through the expressions on the OGRE avatars. The layout of our tool is provided in Figure 16 below. The face on the left is defined as the “working face,” and the face on the right is defined as the “target face.” The musculature of the target face can be manipulated via the sliders on the right side of the tool. There are a total of 19 sliders, only 18 of which are used to manipulate the avatar face. The neutral slider under the “Expressions” heading is incorporated strictly to resemble the updated version of the OGRE demo, to allow for future improvements of the tool. The 18 sliders can take in a value between zero and one, corresponding to the intensity of the corresponding expression. Like the OGRE Facial Animation demo, the expression sliders “happy,” “sad,” and “mad” are used to control a number of Action Units on the OGRE face. Each of these sliders can be used to create an overall generic expression of its corresponding emotion. The “Shape Keys” represent the phonemes associated with the avatar. These sliders can be adjusted to control the precise individual movements of Action Units on the avatar to create a more personalized representation of a personal facial expression. This format directly resembles the overall architecture of the Facial Animation OGRE demo.

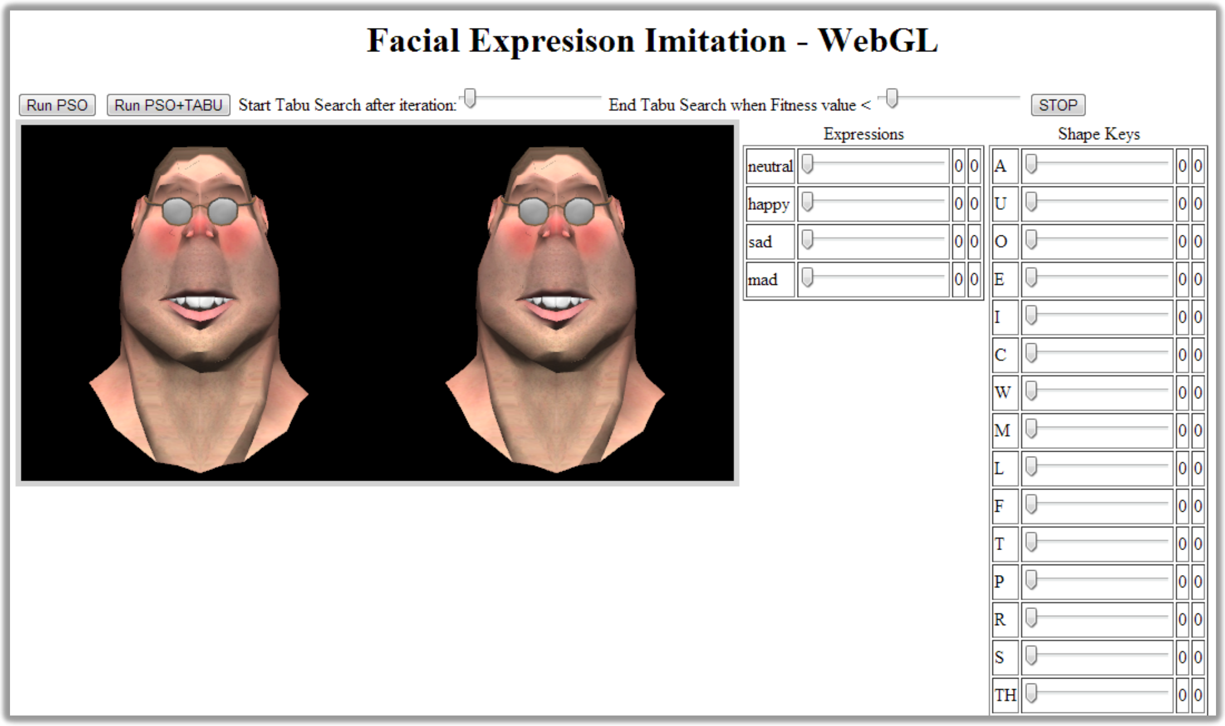


Figure 16: OGRE [8] Algorithm Comparison Tool

Deviating from the standard layout of the OGRE demo, some instruments were added to aid in testing. Placed at the top of the OACT are three buttons and two sliders. The “Run PSO” and “Run PSO+TABU” buttons run their respective algorithms. It should be noted that the modular design of the hybrid PSO-TS algorithm, further discussed in later sections, allows the algorithms to run independently. Thus, the “Run PSO” button will only run PSO and the “Run PSO-TABU” button will run the hybrid PSO-TS algorithm. The “STOP” button immediately halts the search process and displays the best facial match up to that point in the search. The “Start Tabu Search after iteration:” slider is used to activate Tabu Search after a specified number of iterations. The “End Tabu Search when Fitness <” slider is used to terminate the local Tabu Search once a fitness threshold is reached. Both of these sliders are only applicable to the PSO-

TS search method. Again, these additional buttons and sliders are used for testing purposely only and do not directly control the avatar's facial structure.

An important factor considered when developing OACT was the search status and convergence feedback. Once either search is activated, real time feedback of the progress of the search is provided through the working avatar face. Keeping in mind that the "working" face is on the left and the "target" face is on the right, the avatar faces in Figure 17 demonstrate the progression of the search as it converges toward the optimal solution. The top set of OGRE faces shows a working avatar face that is greatly deformed. At this point, the search has just begun and the algorithm is just beginning to explore the search space. As the search finds better solutions, it continually converges toward the target face as seen in the middle set of avatar faces. Finally, once the optimal solution has been found, the working and target avatar faces match, as depicted in the bottom set of OGRE faces.

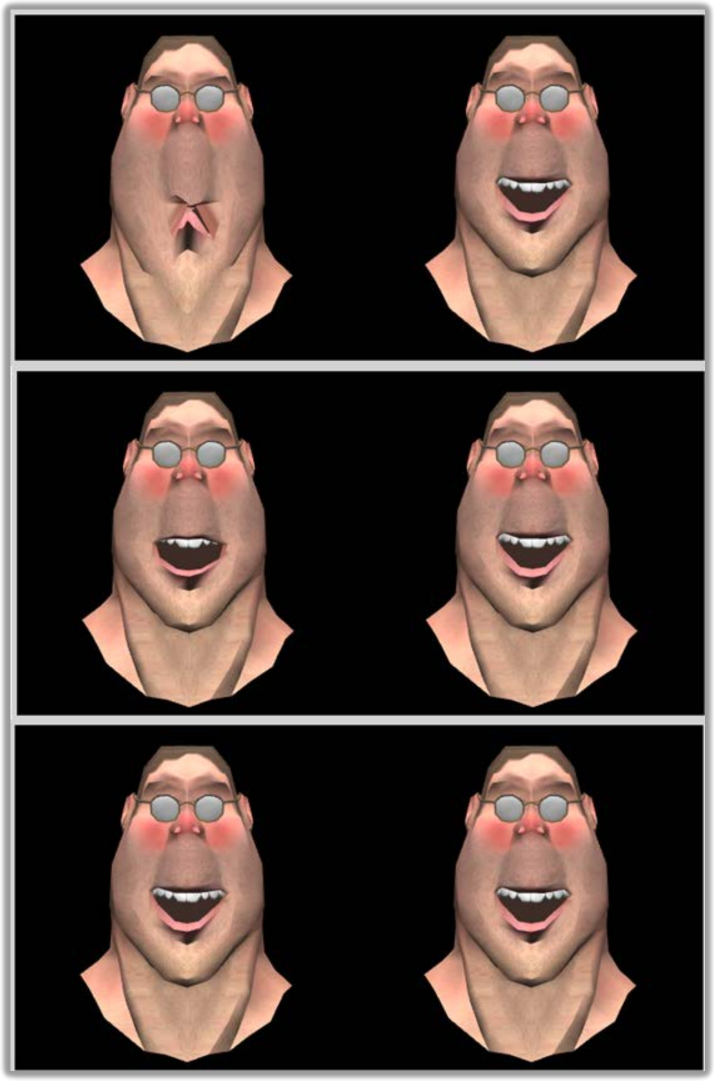


Figure 17: OGRE [8] Convergence Feedback

5.1.2 IMPLEMENTATION OF PSO

Now that the functionality of the OGRE Algorithm Comparison Tool has been extensively described, the implementation behind the PSO algorithm that allows it to run is discussed. This section describes the specific steps taken to develop PSO in the same manner as the authors of

[1]. Again, the same approach of developing PSO was taken to ensure the results of [1] could be reproduced. Thus, the same architecture and equations were used to develop the PSO algorithm used in this thesis. The following description of the implementation of PSO is directly adapted from [1].

The PSO algorithm implemented in this thesis is embedded within OACT. It utilizes the same layout and architecture as described in Chapter 4 and [1]. Thus, this chapter discusses the specifics of the algorithm, rather than reiterating the key concepts and theory behind PSO. Pseudo code of PSO is provided in Figure 18 to give an overview of the execution of the algorithm.

```
SWARM_SIZE = 50;
NUM_ITERATIONS = 100;

while(iteration < NUM_ITERATIONS)
{
        for(i=0; i<SWARM_SIZE; i++)
        {
                checkFitness(18 parameters)

                for(j=0; j<SWARM_SIZE; j++)
                {
                        check pbest and gbest;
                        save bests;
                }
        }
        for(i=0; i<SWARM_SIZE; i++)
                update positions and velocities of swarm;
}
```

Figure 18: PSO Pseudo Code (Adapted from [1])

The PSO algorithm commences the search by randomly initializing 50 particles. These 50 particles represent the *swarm size* of the search. Although the particles' initial positions and velocities are random, they are constrained to ensure each particle is initialized within the search space. Each 18-dimensional particle presents a potential solution in the search space. The 18-dimensional space corresponds to the 18 sliders used to control the OGRE face. PSO searches throughout the search space for the optimal solution for 250 iterations. During each iteration, the fitness value is checked to determine if a better solution has been found. A loop continually goes through each particle in the swarm and checks the local best *pbest* of each particle and the global best *gbest* of the swarm. The fitness functions used to determine the *pbests and gbest* resembles those used in [1] and are shown in Figure 19. The fitness function utilizes a *computeImageDifference* method to calculate the pixel-by-pixel difference. This pixel-by-pixel difference is defined as the pixel difference between the working and target avatar faces. This difference is then graded on a scale from zero to 4 to grade the difference between the working and target avatar faces. It should be noted that the initial fitness is dependent on the random initialization of the swarm. Therefore, the initial fitness values significantly vary based on the random initialization of PSO.

```
function computeImageDifference()
{
    drawScene();
    var w=gl.canvas.width/2;
    var h=gl.canvas.height;
    var  lpixelValues=new Uint8Array(4*w*h), rpixelValues=new Uint8Array(4*w*h);
    gl.readPixels(0,0,w,h, gl.RGBA, gl.UNSIGNED_BYTE, lpixelValues);
    gl.readPixels(w,0,w,h, gl.RGBA, gl.UNSIGNED_BYTE, rpixelValues);
    var diff=0;
    for(pixel=0;pixel<w*h;pixel++){
        diff+=Math.abs(lpixelValues[pixel*4+0]-rpixelValues[pixel*4+0]);
        diff+=Math.abs(lpixelValues[pixel*4+1]-rpixelValues[pixel*4+1]);
        diff+=Math.abs(lpixelValues[pixel*4+2]-rpixelValues[pixel*4+2]);
    }
    diff /= 255*w*h*3;
    diff *= 100;
    return diff;
}
```

Figure 19: PSO computeImageDifference

Once the fitness function determines the *pbest* solutions for that iteration, they are recorded, and the positions and velocities of the particles are updated using the equations shown in Figure 20 [1]. The original PSO equations (3) and (4) discussed in Chapter 4 are provided directly above Figure 20 to establish a comparison of the equations to clearly illustrate the values used. The swarm consistently converges toward the optimal solution until the stopping criteria is reached. In this case, the stopping criteria is designated as the number of iterations.

$$v_{new} = C_0 * v_{previous} + C_1 * R_1\left(x_{gbest} - x_{previous}\right) + C_2 * R_2\left(x_{pbest} - x_{previous}\right) \quad (3)$$

$$x_{new} = x_{previous} + v_{previous} \quad (4)$$

```
newVel[j] = 0.9*vx[j]+(Math.random()*2)*(pBest[j] - lx[j])+(Math.random()*2)*(gbest[j] - lx[j]);
newPos[j] = lx[j]+newVel[j]*0.25;
```

Figure 20: PSO Position and Velocity Equations

5.2 PHASE II DEVELOPMENT

Once the results of [1] were reproduced in the OGRE Algorithm Comparison Tool using PSO, the development of the hybrid PSO-TS algorithm began. The sole purpose of hybridizing Tabu Search with PSO is to establish a local search technique that can be used to diversify the search and find the true optimal solution within the search space. This prevents PSO from prematurely converging toward local optima. The concepts and theory behind this hybridization were extensively discussed in Chapter 4. Thus, the purpose of this chapter is to provide the specific details associated with implementing PSO-TS. The specifics of how the TS algorithm is implanted into PSO are discussed and an in-depth analysis of the PSO-TS architecture is provided next.

5.2.1 INTEGRATION OF PSO-TS

To incorporate a local search within PSO, Tabu Search was implanted directly into the PSO code. The architecture of this hybrid algorithm is shown in Figure 21. It should be noted that the PSO-TS architecture and flowchart were adapted from [19]. The flowchart describes how the two algorithms communicate to effectively converge to the optimal solution within the search space. As shown in the flowchart, PSO is the main driver of the PSO-TS algorithm. Overall, it controls and directs the search based on its global search technique. After PSO initializes the swarm and generates an initial solution, the swarm's fitness is evaluated via the fitness function as previously defined in the Phase I Development section. Once the best neighbor *pbest* is found, PSO passes this *pbest* to TS to perform a local search. The local search is established around this current *pbest* solution. TS begins by initializing a Tabu List to keep track of visited solutions. Neighborhoods are then set up around each solution that is not already in the Tabu List. The neighborhoods establish the boundaries for the local search technique of TS using *hyper-spheres,* which are based on the hyper-rectangles used in [19]. A hyper-sphere is defined as an n-dimensional sphere that covers an area of the search space containing potential solutions. A formal definition ensues:

A hyper-sphere centered around C with radius R is the area containing any number of solutions S such that the distance between S and C is less than R.

The hyper-sphere's center *C* is established using the *pbest* passed in from PSO. The radius *R* of each hyper-sphere is a constant value of 0.25. The neighborhoods are locally searched for a better solution. If one is found, it is added to the Tabu List to ensure it is not revisited and *pbest* is updated. If a more optimal solution is not found, no changes are made, and PSO continues.

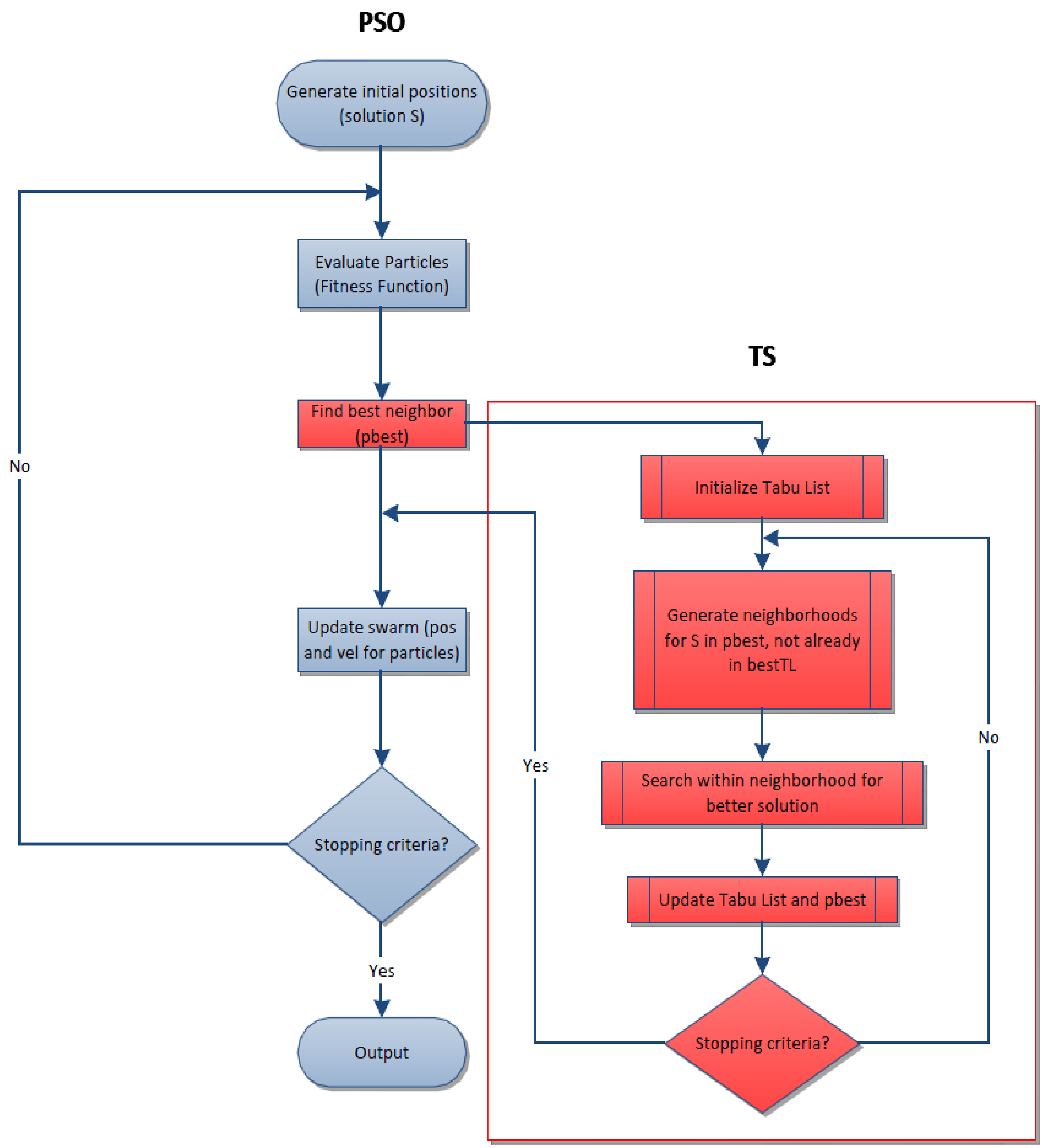


Figure 21: In-Depth Flowchart of PSO-TS

The code used to perform the local searches and update the particles is shown in Figure 22. As shown in the code, TS performs 25 local searches within nearby neighborhoods. It should be noted that the Tabu List is checked every time PSO calls TS via the inTabuList function. This ensures that previously visited solutions are not revisited. Since the particles' positions and velocities are not discrete values, the Manhattan Distance is used to determine if the solution resides in the Tabu List by checking within a close range around the visited solutions. TS then executes the set amount of local searches by randomly checking solutions within the hyper-sphere. If the random position found while checking in the hyper-sphere is found to be outside of the valid slider bounds [0-1], the value is reset to the respective bound. That is, any number greater than one will be reset to one and any number less than zero will be reset to zero. Only solutions that have a lower fitness function than the current best fitness function are considered. In this case, the number of local searches serves as the stopping criteria for the local Tabu Search. Once the local search is complete, TS returns the updated *pbests* to PSO. PSO then continues its global search in its native manner. Again, since PSO is the driver of the PSO-TS algorithm, the PSO-TS search does not terminate until the stopping criteria for PSO is met. Thus, throughout each iteration of PSO, TS will perform a local search based on the current local bests of PSO.

```
function updateLocalBestForTabuSearch(index)
{
    var length = 0.25;  // neighborhood hypersphere radius: 0.25
    var nSearches = 25; // Number of searches in the neighborhood
    var pBest=[];
    swarm.pBestLoc[index].get(pBest);
    bestFitness=swarm.pBest[index]
    var newPos=[],newFitness;
    var newBestPos=[],newBestFitness=bestFitness;

    var tabuList = [];

    function inTabuList(){
        var threshold = nDim*0.05;
        for (var i=0; i<tabuList.length;i++){
            var dist = 0;
            for (var j=0;j<nDim;j++){
                dist = dist + Math.abs(tabuList[i][j]-newPos[j]); // Manhattan distance
            }
            if (dist < threshold){
                console.log("Tabulist Hit with dist "+dist);
                return true;
            }
        }
        return false;
    }
    var count =0;
    for (var i=0; i<nSearches;i++){
        for (var j=0;j<nDim;j++){
            newPos[j] = (Math.random()-0.5)*length+pBest[j];
            // Keep the coordinates within [0-1] bound
            if (newPos[j] < 0) newPos[j] = 0;
            else if (newPos[j] > 1) newPos[j] = 1;
        }
        if (!inTabuList()){
            tabuList.push(newBestPos);
            newFitness = fitnessFunc(newPos);
            if (newFitness < newBestFitness){
                newBestPos=newPos.slice(0);     // store newPos as newBestPos
                newBestFitness = newFitness;
                count++;
            }
        }
    }
    if (count>0){
        swarm.pBestLoc[index].set(newBestPos);
        swarm.pBest[index] = newBestFitness;
    }
}
```

Figure 22: TS's Local Search Using Hyper-Spheres

5.3 SUMMARY

This chapter provides an extensive review of the implementation of the prototype used to solve the problem presented in Chapter 3. The analysis is based on the modular approach described in Chapter 4. The modular design of PSO-TS makes it easy to turn the local TS on and off within the PSO search. That is, TS can be started and stopped while PSO continually searches throughout the search space. To further optimize PSO-TS, TS is only initiated once PSO has become stuck in local optima. This method diversifies the search and allows PSO-TS to continue its search and explore the entire search space despite briefly prematurely converging. PSO-TS takes advantage of PSO's search speed and utilizes the efficiency of TS's local search technique to optimize the performance of the hybrid algorithm. This modular development strategy allows our mechanism to be directly transferred from the OGRE application presented in this thesis to a photo of a human and a human-like avatar, which is the long-term objective of this research. This modular design is also ideal for testing purposes because the two searches can be independently run to determine the difference in performance. In the next chapter, the mechanism is tested to determine whether or not our work makes an improvement on the previous method of [1].

CHAPTER 6: TESTING AND EVALUATION

By testing and comparing PSO-TS to the original PSO algorithm as implemented in [1], a measure of how well our new hybrid model can converge to an optimal solution within a multi-parameter search space is obtained. This chapter opens with an overview of how the testing is organized by describing the three-phase approach taken. A detailed description of the design of each testing phase used to evaluate our work is given. To better organize and explain the findings of each phase, the results are presented and evaluated in their respective phase subsections. Finally, the hypothesis is evaluated using the conclusions drawn from both phases of testing.

6.1 TESTING APPROACH

The OGRE Algorithm Comparison Tool is used in all of the experiments as a test bed to compare PSO-TS to the original PSO used in [1]. The tests were carried out on a desktop PC with the following specifications:

- Operating System: Windows 7 Professional 64-bit
- Processor: Intel Core i7 – 3770k @ 4.4GHz
- RAM: 16 GB DDR3 1600MHz
- Graphics Card: Nvidia GeForce GTX 680 2GB

Since the purpose of this research is to expand and improve upon previous research [1], the same testing steps and procedures used in [1] are implemented in our tests. The testing of our prototype was completed in three phases. To ensure a fair comparison between the two methods, a baseline is established in Phase I by reproducing the results of [1] using our PSO algorithm in OACT. In Phase II, a direct comparison is made between the results obtained using the hybrid PSO-TS algorithm presented in this thesis and those obtained by the PSO-based mechanism of [1]. Because of some unexpected results encountered in the first two phases of testing regarding the robustness of the methods employed, Phase III internally evaluates the performance of our system by directly comparing our PSO algorithm to PSO-TS. The comparison of the two algorithms sheds some light on the reliability of PSO-TS.

6.2 PHASE I TESTING: ESTABLISHING A BASELINE

The baseline testing exposes any potential differences between our PSO implementation and that reported in [1], certifying that the PSO algorithm is indeed an independent variable. This indicates that any improvement found in PSO-TS's performance cannot be attributed to a difference in the implementation of PSO alone. Instead, only our hybrid architecture and TS's local search can be credited for any improvement in convergence performance.

6.2.1 BASELINE TESTING APPROACH

To test the convergence performance of PSO in [1], the authors implemented a testing procedure that used a "photograph" of the OGRE avatar as a target face. This testing method resembles that used in our OACT, where the "photograph" in [1] is referred to as the "target" OGRE face. A pixel-by-pixel fitness function is used in both methods to recreate five, preset emotional expressions of the OGRE avatar established in [1]. Thus, lower fitness function values indicate a smaller pixel difference between the target and working avatar faces. The authors of [1] excluded a "disgust" facial expression and only used five of the six basic emotional expressions for their testing [15]. To directly compare our results to those of [1], we used the same five emotional faces. As shown in Figure 23, the five faces created in [1] to represent individualized expressions are: anger, happiness, sadness, fear, and surprise. These five emotional expressions were created by the authors of [1] to be used as a standard of measurement to grade the performance of their PSO algorithm. In our testing, the slider values used to create these five OGRE faces are reproduced and used in the same manner as [1] to determine the differences in performance between the PSO and PSO-TS algorithms. That is, the "working" OGRE face created by the algorithms is compared on a pixel-by-pixel manner to these same five faces to determine if the expressions were successfully recreated.

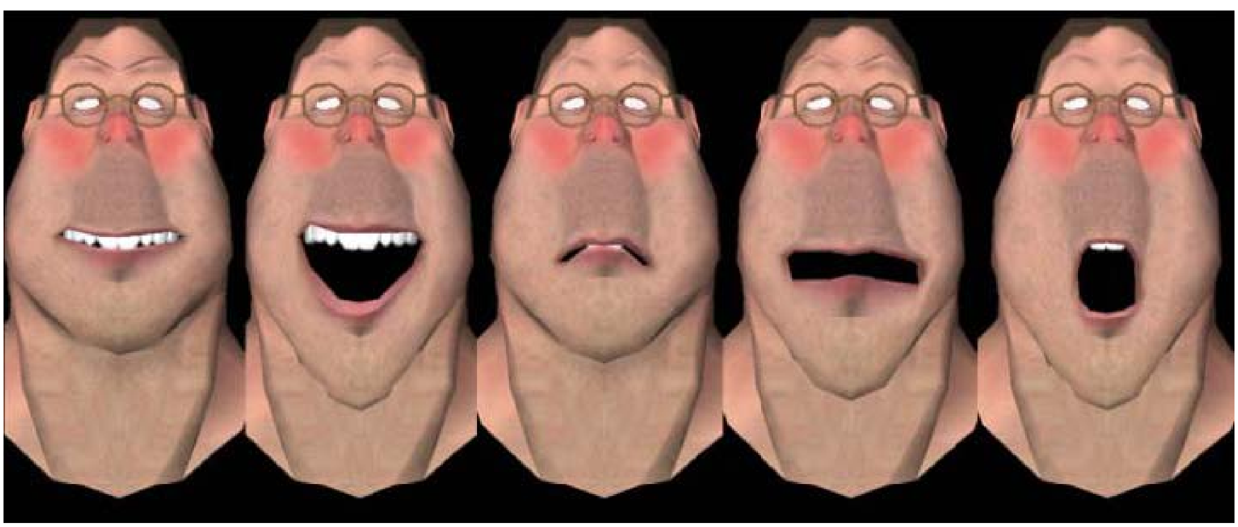
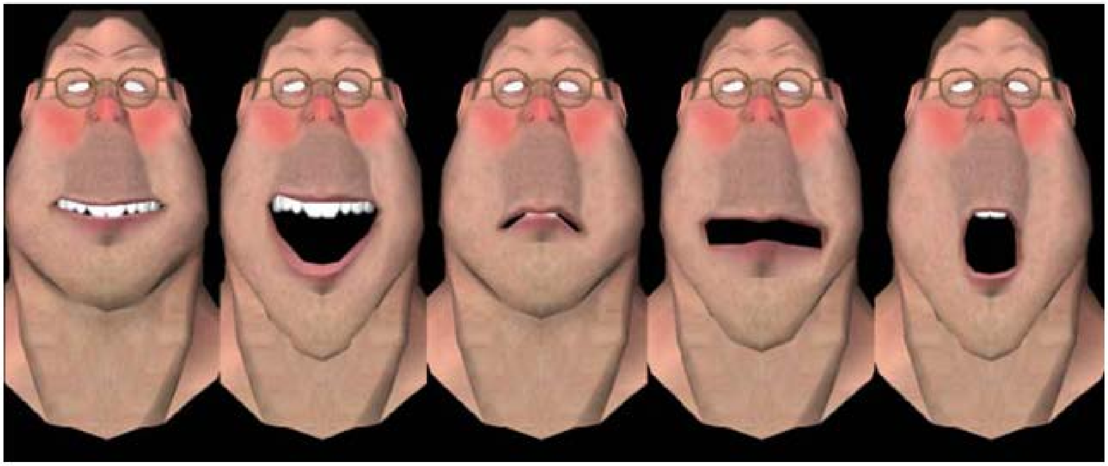


Figure 23: Left to Right: Anger, Happiness, Sadness, Fear, Surprise (Reproduced without permission [1])

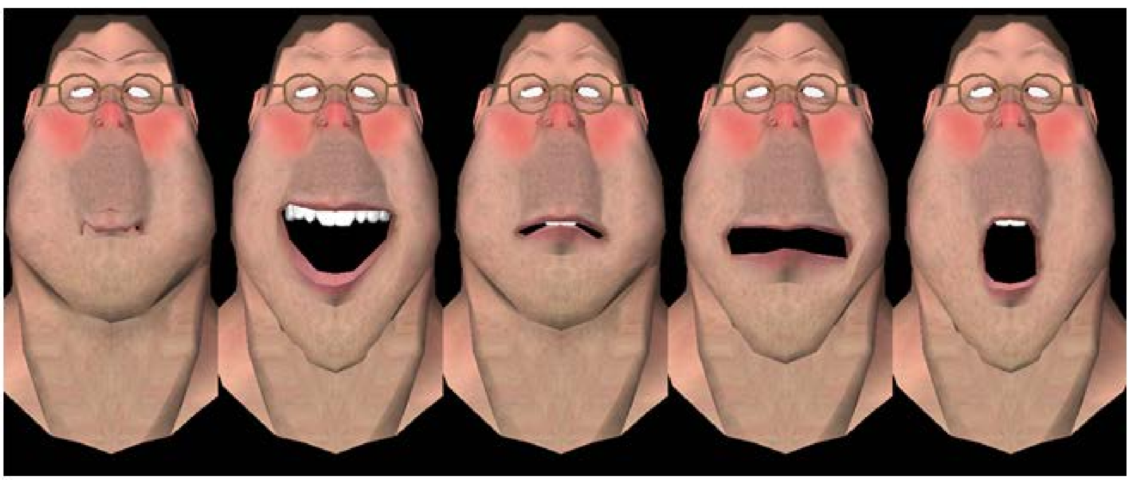
As stated in Chapters 4 and 5, our PSO algorithm is fully based on that of [1]. For testing, the same fitness function used in [1] was implemented in our PSO. This fitness function uses the same pixel-by-pixel difference for the fitness function. The PSO algorithm was run on our OACT for 250 iterations, using a swarm size of 50. Again, these are the exact PSO parameters that were used for testing in [1]. It should be noted that this thesis seeks to improve upon the original tests and findings of the PSO algorithm in [1] that used a population size of 50. Therefore, the tests in [1] that were geared toward forcing a successful convergence on the angry face by increasing the population size is disregarded. Those specific tests increased the population size of PSO to 100 to force a successful convergence on the angry face. This change doubled the runtime to four hours, and seems to have only been applied to establish a successful method to force a convergence on the angry face. Thus, we only compare our results with the tests in [1] that used a swarm size of 50. Each of the five preset OGRE faces in [1] is used for our tests. After the 250 iterations are complete, the avatar face produced by PSO is visually compared to the original "target" face for accuracy, and the fitness function is used to grade the success of the convergence.

6.2.2 BASELINE TESTING RESULTS

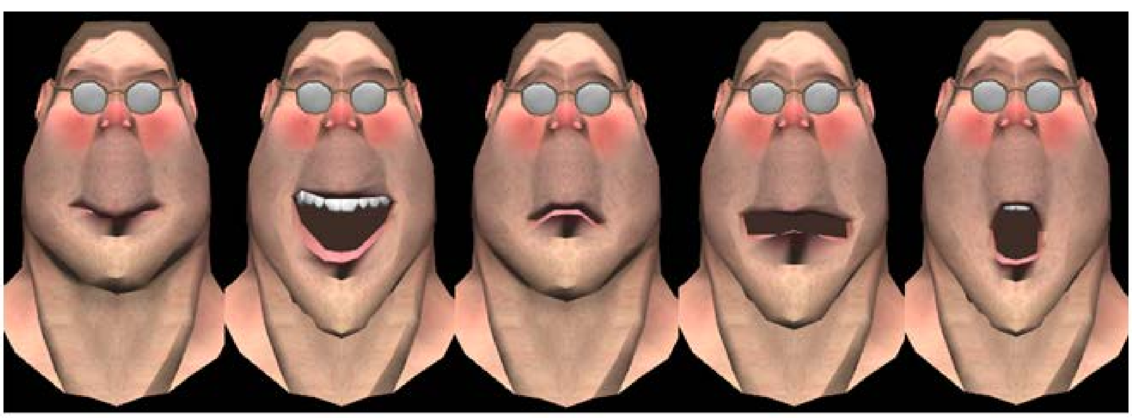
Figure 24 below compares the five OGRE faces obtained from this part of the testing. The five original OGRE test faces from Figure 23 are provided again at the top of Figure 24 so a direct visual comparison of the results can be easily made by the reader. The middle section of faces are those obtained from [1] using their PSO method, and the bottom section of faces are those acquired from our PSO algorithm using OACT. Before analyzing these results, it should be noted that there are some slight differences in the faces that are independent of the performance of PSO. These differences stem from the updated OGRE models and textures used to create OACT. To clearly explain these differences, a discussion of the sad face ensues. As seen in Figure 24, the sad face produced by [1] is almost identical to the one produced by our OACT. However, despite a proper convergence by PSO, some slight differences are visible around the mouth of the avatar. Unlike the sad face produced by [1], the teeth are hard to see in our resulting sad face due to the low contrast between the mouth opening and teeth.



Target Faces



PSO [1]



OACT's PSO

Figure 24: Emotional Faces Generated by PSO: Anger, Happiness, Sadness, Fear, Surprise (left to right)

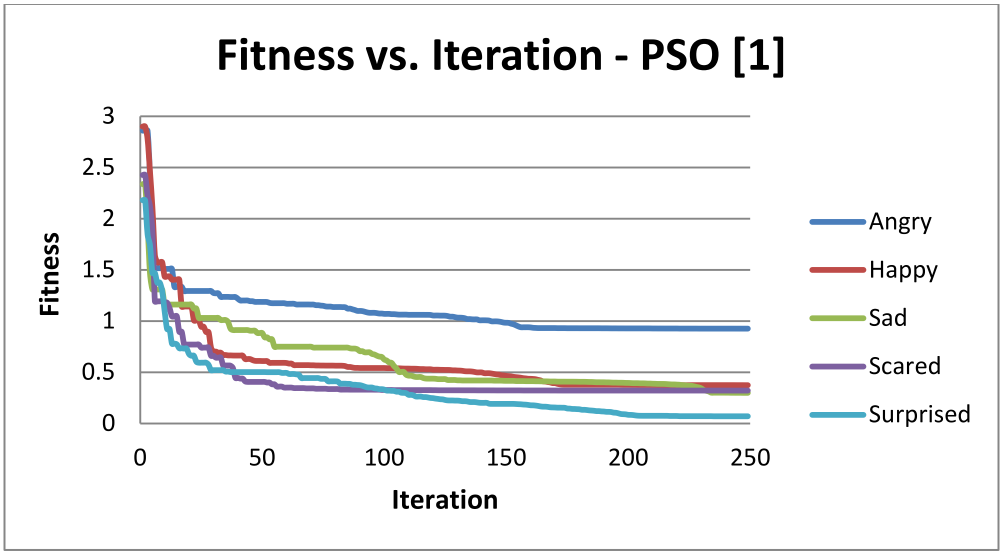
As depicted in Figure 24, all of the OGRE test faces were recreated in a similar fashion except for the angry face. As mentioned in [1], the angry face failed to converge to the optimal solution and throughout the convergence became stuck with the mouth closed. Although there are slight differences in the angry faced produced in [1] and ours, the same outcome of a closed mouth was generated due to PSO's premature convergence. Thus, the same outcome can be said to have been produced for all five OGRE test faces.

To further verify this baseline, we compare the fitness function outputs obtained by both of the PSO algorithms. The graphs in Figure 25 show the PSO convergence of each system. The top graph represents the fitness function results obtained by [1], and the bottom graph depicts the results acquired using our PSO algorithm in OACT. Figure 25 shows that each OGRE test face successfully converged to a minimum fitness function (not necessarily zero), indicating a low pixel-by-pixel difference in the target and working faces. Although the trend for both graphs is the same, there are some differences that need to be addressed.

One obvious difference between the graphs is the initial fitness value obtained in the first iteration of each face. The results from [1] indicate an initial fitness value that was greater than 2 in all cases. However, in our results, we obtained an initial fitness value around 1.5. To justify and explain this difference, the initialization of PSO is readdressed. As discussed in Chapters 4 and 5, PSO is randomly initialized to begin the search. That is, the search begins with a random solution within the search space. Therefore, there is no specific reason for the differences in the initial fitness functions other than PSO's random initialization to begin the search. Another main difference in the graphs is the rate of convergence. Despite the two PSO algorithms being implemented in the same manner and using the same parameters (iterations, swarm size, etc.), the convergence of PSO on our OACT was faster than that of [1]. As depicted in the graphs, our PSO algorithm converged to an optimal solution between iteration 50 and 100. However, the results of [1] show PSO converging to an optimal solution after iteration 150. The differences in convergence speed can be attributed to the overall architecture of the two systems. In [1], PSO was directly implanted into the original OGRE demo. Since PSO was based on a system that was already in place, PSO utilized the underlying functionality of the OGRE demo. On the contrary, our system was built from scratch and only made use of the OGRE models and textures. Thus, the JavaScript implementation of our OACT made certain that our tool was independent of the underlying functionality of the OGRE demo. The difference in these development styles can be attributed to the slight variances in the initial fitness function and the rate of convergence shown in Figure 25.

The time to complete 250 iterations of PSO also varied. In OACT, the average time to complete the 250 iterations for each of the five OGRE test faces was 3.569 minutes. As reported in [1], the time to complete 250 iterations of PSO with a population size of 50 was 2 hours. This is a drastic difference in runtime, despite the similarities in the two PSO algorithms. Some of the runtime difference can be attributed to the different computer systems used for testing. However, the majority of the time difference is due to the different architectures of the two systems. That is, because [1]'s system used the underlying functionality of the OGRE demo, more processing power was required for each iteration. This was avoided by our OGRE

Algorithm Comparison Tool because it was independently developed and did not use the same underlying functionality of the OGRE demo. Nonetheless, the trend of convergence is the same, despite these differences. Thus, a baseline of comparison was established and any differences found in the convergence performance of PSO-TS in the next section can be attributed to the hybrid algorithm itself, not the differences in the implementations of PSO.



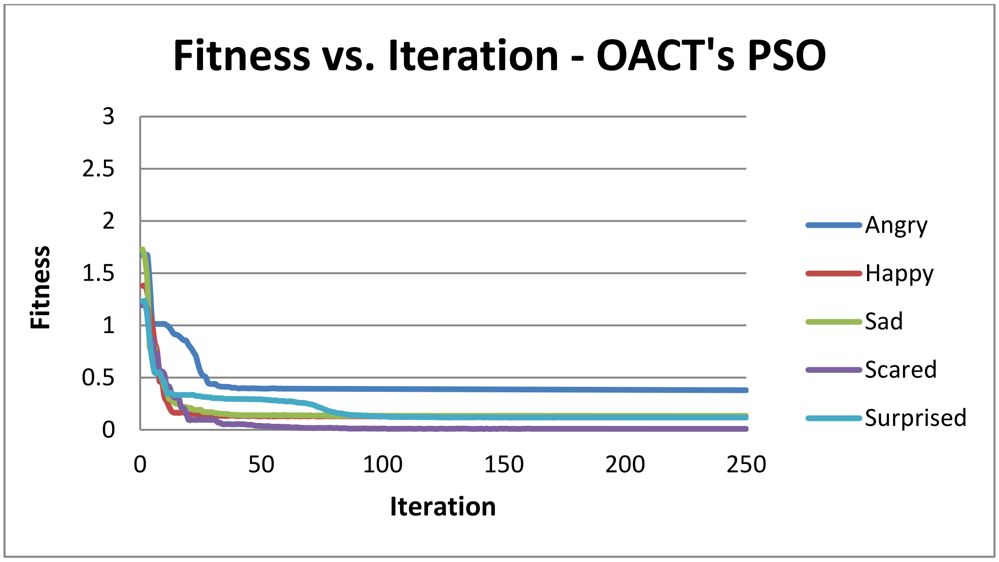


Figure 25: Fitness vs. Iteration Graph: PSO Comparison

6.3 PHASE II TESTING: PSO-TS PERFORMANCE

Now that a baseline has been established, we address the specific problem of this thesis by comparing our hybrid PSO-TS algorithm with the PSO results of [1]. Since our hybrid PSO-TS algorithm also uses a fully implemented PSO algorithm, it should be noted that the same PSO settings are used for both. The same approach and process used in the previous section for baseline testing is repeated in this section.

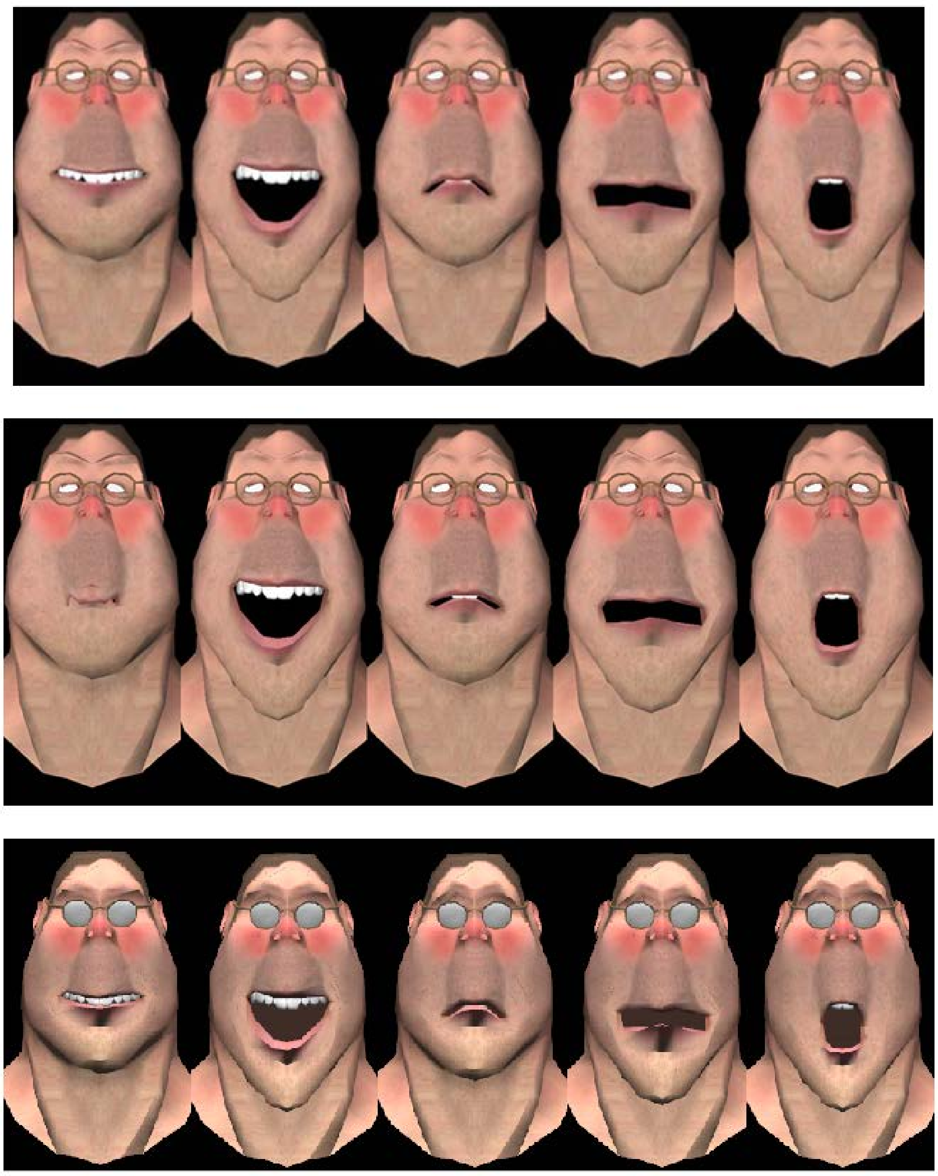
6.3.1 PSO-TS TESTING APPROACH

PSO is reconfigured with a swarm size of 50 particles and run for 250 iterations. The same fitness function, utilizing a pixel-by-pixel comparison, is used for these tests, and the same five preset OGRE faces are used to compare the algorithms' convergence performance. By comparing our PSO-TS algorithm running on OACT with the results obtained in [1], a measure of the performance increase of PSO-TS is obtained. In the next section, the five OGRE faces produced by PSO-TS are presented and analyzed. The fitness function and convergence speed of PSO-TS are also examined.

6.3.2 PSO-TS TESTING RESULTS

Figure 26 below compares the five OGRE faces obtained while testing PSO-TS. The five original OGRE test faces from Figure 23 are included at the top of Figure 26 to allow the reader to make a direct visual comparison of the results. The middle section of faces are the results of

[1]'s PSO method. The resulting faces of PSO-TS are included at the bottom of the figure. As in the baseline testing section, it should be noted that some slight differences in the faces are present. These differences are independent of the performance of the algorithm and only stem from the different OGRE models and textures used in OACT.



Target Faces

PSO [1]

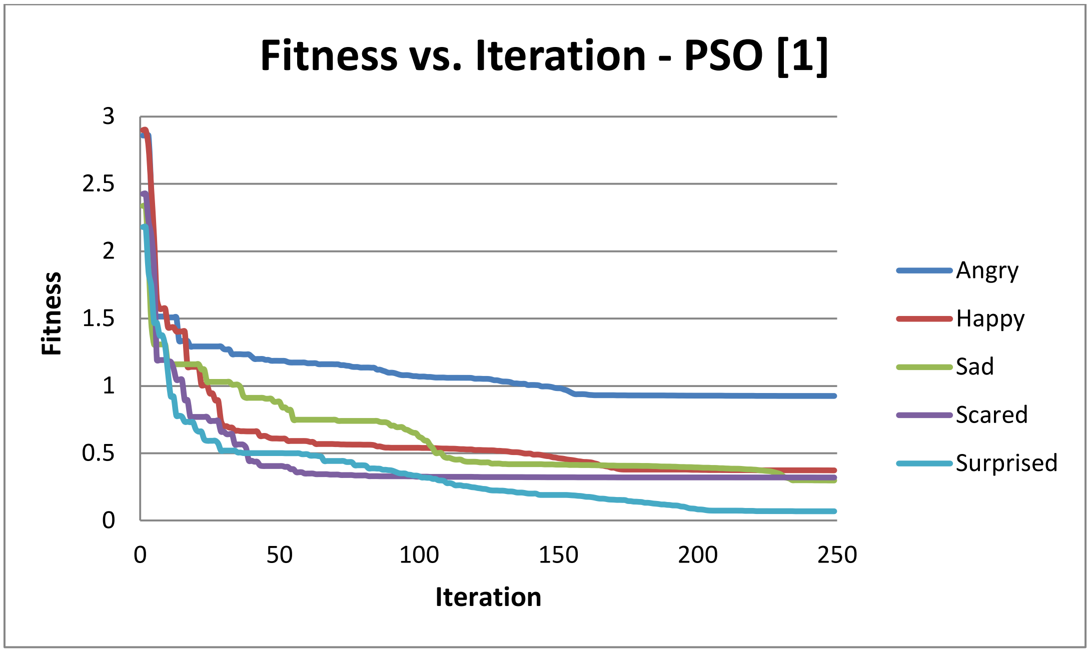
PSO-TS

Figure 26: Emotional Faces Generated by PSO-TS: Anger, Happiness, Sadness, Fear, Surprise (left to right)

As depicted in Figure 26, all of the OGRE test faces were reproduced identically by PSO-TS. The angry face that failed to properly converge in the baseline testing was successfully formed using PSO-TS. Since all of the PSO parameters remained the same for this section of testing, the performance increase can only be attributed to the hybrid architecture of the algorithm and TS's local search technique. TS successfully diversified the search and did not allow PSO to prematurely converge. The diversification of the search provided by TS is further discussed in the next section when the fitness function graphs are presented and analyzed.

To further verify this performance increase, the fitness function outputs of both algorithms are compared. The graphs in Figure 27 show the convergence performance of each algorithm. The top graph depicts the fitness function results obtained by [1], and the bottom graph displays the results acquired using PSO-TS in OACT. Figure 27 shows that when using PSO-TS, each OGRE test face successfully converged to zero. This convergence of the fitness function indicates an almost nonexistent pixel-by-pixel difference between the target and working avatar faces.

An important observation was made throughout testing that regards PSO-TS's convergence to zero. While converging, PSO-TS seemed to get stuck multiple times throughout the search, and the fitness function would stop improving for a number of iterations. This tendency can be seen in the PSO-TS fitness graph provided in Figure 27. The angry face's fitness function leveled out between iterations 25 and 50, and it seemed the algorithm was subject to PSO's premature convergence. However, TS's local search technique diversified the search by forcing the algorithm to explore nearby neighborhoods within the search space. Thus, as shown in the graph, the angry face's fitness function continually converged to zero and PSO-TS eventually found the optimal solution.



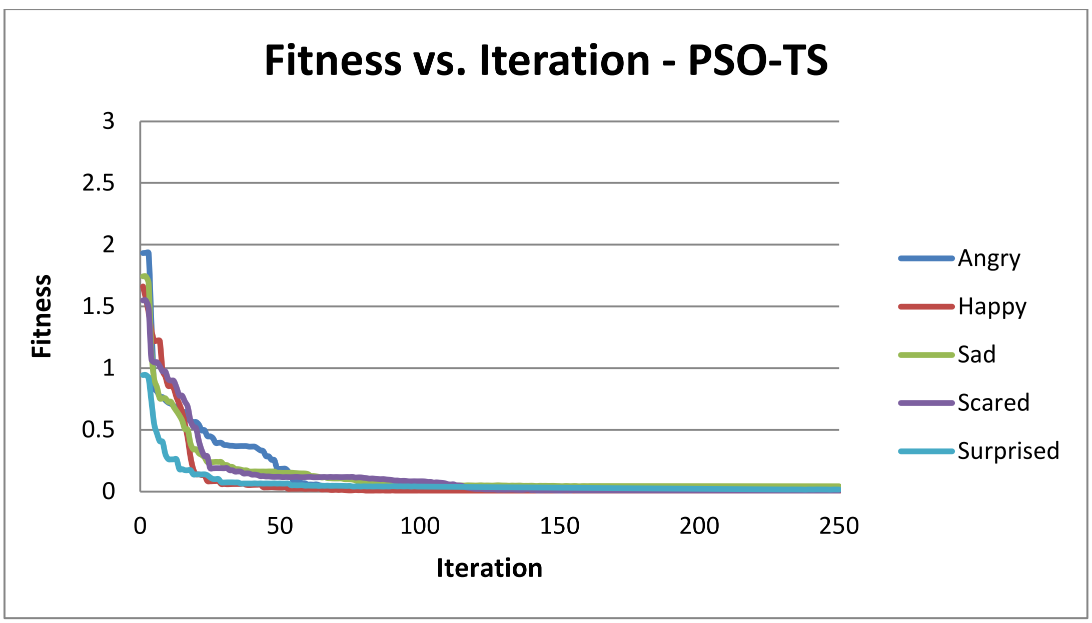


Figure 27: Fitness vs. Iteration Graph: PSO-TS Comparison

Two improvements on the previous results [1] are established with this test. Most importantly, all faces converged to zero with almost no pixel-by-pixel difference, which is a substantial improvement on the previous system described in [1]. Second, the number of iterations required to converge to the optimal solution was reduced by almost half. On average, PSO-TS converged to the optimal solution between iteration 75 and 125, at which point the differences in the OGRE faces were undetectable because of the exceptionally low pixel difference between the images. However, it should be noted that the low number of iterations required for convergence by PSO-TS can be credited to TS's local search within each PSO iteration. Essentially, two searches are being conducted each iteration. Therefore, the chance of finding a more optimal solution each iteration is greatly increased.

Despite the significant improvement in performance by PSO-TS, the time to complete the 250 iterations greatly increased. In OACT, the average time for PSO-TS to complete the 250 iterations for each of the five OGRE test faces was 8.809 minutes, which is more than double the 3.569 minutes required by PSO. Nonetheless, PSO-TS still outperformed PSO in all other aspects and successfully converged to zero in all test cases.

6.4 PHASE III TESTING: SHEDDING LIGHT ON RELIABILITY

Due to some unexpected convergence issues with PSO encountered in the first two phases of testing, Phase III testing internally examines the performance of our system by comparing our PSO algorithm with PSO-TS. In Phase I and Phase II testing, PSO prematurely converged more often than expected. As reported in [1], PSO's premature convergence was expected when testing the angry OGRE face. However, PSO often prematurely converged when testing the other OGRE test faces as well. Although results were obtained to compare our PSO algorithm to that of [1] and establish a baseline for testing, further analysis of the algorithms and their convergence performance is necessary to further establish PSO-TS's superiority over PSO. Thus, the robustness of the two algorithms is evaluated in this section. The section opens with a description of the testing approach regarding the premature convergence of the algorithms. The results are then presented and analyzed to determine the performance difference with respect to the algorithms' reliability and robustness.

6.4.1 INTERNAL TESTING APPROACH

Referring back to Figure 25, the premature convergence of individual OGRE test faces is evident. The graph representing the fitness function outputs of [1] shows all test faces except surprise prematurely converging to a value other than zero. Our PSO results, also visible in Figure 25, depict the same trend. Although PSO converged to fitness values lower than those in [1], the algorithm still failed to converge to zero in all test cases except the scared test face. Despite no visual evidence of any differences in the target and working avatar faces, this failure to converge to zero indicates there is indeed a pixel difference in the images. To compare the robustness of each algorithm, five individual test cases are set up using each OGRE test face. In each of the five test cases, a single OGRE test face is used to test the convergence reliability of PSO and PSO-TS. Each algorithm is run on the test face under consideration a total of ten times. A successful convergence for Phase III testing is defined as a proper convergence to the facial parameters that reveal a recreation of the intended OGRE test face. If the algorithm becomes "stuck" and reveals any visual differences between target test face and the working avatar face, the test is considered a failure.

6.4.2 INTERNAL TESTING RESULTS

After running the tests, it was evident that PSO-TS is significantly more reliable than PSO. Reliable can be defined as converging to the proper solution within the search space without getting stuck in a local minimum. Throughout the first two phases of testing, PSO often became stuck in a local optimum revealing a distorted face as shown in Figure 28.

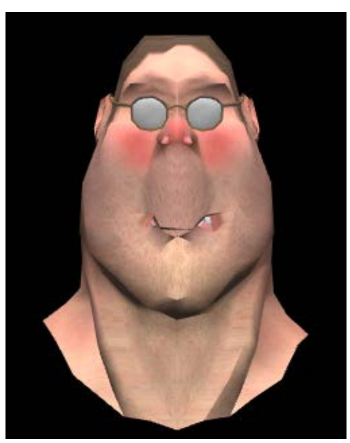


Figure 28: Distorted OGRE Face using OACT

As shown in Figure 29, PSO-TS handled each OGRE test face exceptionally well when compared to PSO. Each bar on the graph represents the number of times the algorithm successfully converged, based on the testing definitions presented in the previous section. The five OGRE test faces are provided at the bottom of the graph to indicate which specific face was used for that sequence of testing. Two conclusions can be made from this graph. First, PSO-TS is much more reliable and robust when it comes to facial recreation and parameter optimization. Not only does it provide a more accurate solution by avoiding a premature convergence, it is consistent and does not fail as often as PSO.

The second conclusion drawn from this graph is that some of the OGRE test faces are easier to generate than others. Because the slider values are randomly initialized using PSO, some test faces often get stuck at the beginning of the search. This is because there are some combinations of slider values that are difficult to recover from and reproduce through random generation. To further explain this concept, the convergence performances of the angry and sad faces are considered. There are many combinations of sliders in the OGRE system that produce an open mouth in the avatar. Seeing as there are not many sliders that keep the mouth closed while exposing teeth, there is a good chance that the random initialization of PSO will start the search with an open mouth and no exposed teeth. Yet, the exact opposite of this is required by the sad and angry faces. A closed mouth with exposed teeth is required to successfully generate these faces. Thus, if PSO initially generates an open mouth expression, there is a good chance PSO will converge to another region of the search space that does not contain the optimal solution for the sad or angry faces. This would make it harder for the algorithm to recover from this poor initialization and break out of this local optimum. This random initialization to an open mouth by PSO is the reason why the sad and angry faces are more difficult to reproduce than the others. This tendency is seen in the graph of Figure 29. Problems with improper convergence arose when testing the angry and sad faces, and the convergence rate of both algorithms dropped while testing these faces. Nonetheless, despite timing differences, PSO-TS still outperformed PSO in all cases and proved to be the more robust algorithm.

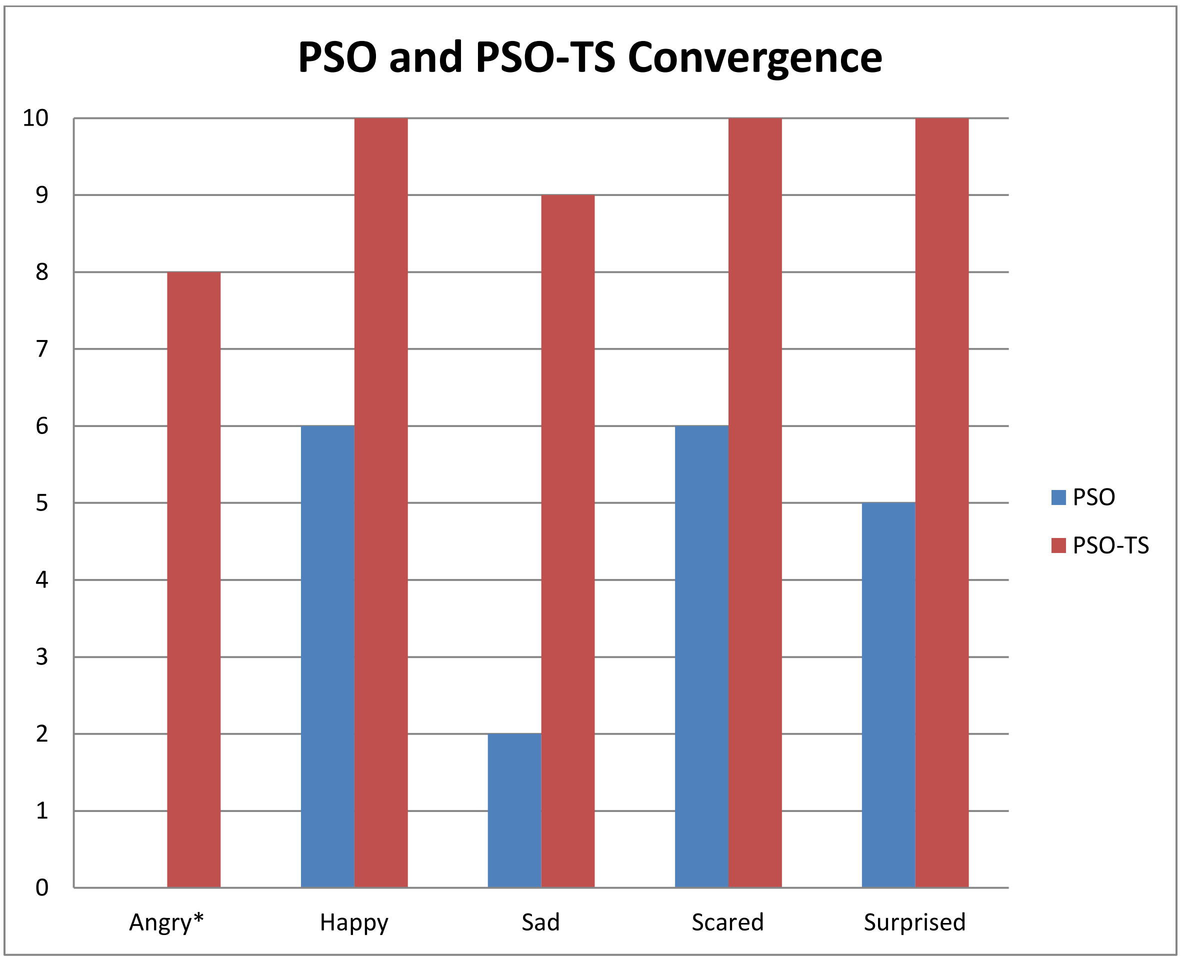


Figure 29: PSO and PSO-TS Convergence Performance (*Angry: PSO was zero)

6.5 CONCLUSIONS

By establishing a baseline for testing in Phase I, all improvements in performance by PSO-TS can be directly credited to the hybrid architecture and TS's local search technique. PSO-TS testing in Phase II shows that PSO-TS greatly improved the convergence performance when applied to the five OGRE test faces included in [1]. Thus, the performance of our hybrid algorithm greatly improved upon the previous research presented in [1]. Internally evaluating OACT and PSO-TS in Phase III further established PSO-TS's superiority over PSO. It was shown in the performance convergence comparison tests that PSO-TS consistently outperformed PSO in all aspects. Despite timing differences, the hybrid algorithm proved to be more robust and reliably converged to the optimal solution in all five OGRE test faces. Thus, it is clear that PSO-TS has greatly improved the performance of the multi-parameter optimization system described in [1].

The conclusions drawn from testing the specific problem of this thesis signify PSO-TS's superiority. However, the general problem and hypothesis still need to be evaluated. With respect to the hypothesis, our PSO-TS algorithm overcame the limitations of PSO's premature convergence and successfully transferred an emotional expression displayed on the target avatar to the working avatar. Thus, the hypothesis of this thesis is verified and the general problem of transferring our work to a similar system is likely to result in future research. Because of our modular approach, this machine learning technique proves to be a good candidate to facilitate the transfer of our method to a similar mechanism that incorporates a photograph of a human being and a human-like avatar.

CHAPTER 7: CONCLUSION

This chapter summarizes the thesis as a whole, encompassing a summary, conclusion, and discussion of future work. The summary revisits the key aspects of the thesis, including the problem addressed, the current solutions, the approach presented, and the results obtained. The conclusion provides an analysis of the scalability of this project to the real world. Finally, the future work section discusses the long-term goal of this thesis and presents an outlook on what others can accomplish based on the findings of this project.

7.1 SUMMARY

Throughout this thesis, a machine learning method for automatically imitating a particular person's facial expressions in a human-like avatar was discussed. Because of the complexity of this problem, an intermediate step was taken using an OGRE avatar. A literature review was conducted to further understand how to develop a mechanism to facilitate the translation of a facial expression onto an avatar. The literature review analyzed key subjects concerned with emotion detection and the quantification of facial expressions. This involved and in-depth analysis of Action Units associated with the Facial Action Coding System to understand how the muscular structures of facial expressions are measured. With a neutral face as a reference, the minute movements of the AUs can be tracked and mapped onto an avatar. Thus, intricate details of a specific human's facial expression can be reproduced in a personalized fashion. Previous works involving emotional states of avatars were reviewed and the algorithms used to map emotional expressions onto the avatars were analyzed. The Particle Swarm Optimization and

Tabu Search algorithms were studied to determine their strengths and weaknesses when applied to applications involving parameter optimization. Specifically, a previous OGRE mechanism that used PSO to converge toward personalized facial expressions was analyzed.

This review exposed the limitations of PSO, and the specific problem of overcoming PSO's premature convergence was addressed. The approach taken to overcome these limitations was to form a hybrid PSO-TS algorithm that takes advantage of PSO's global search and TS's local search. This allows the hybrid algorithm to search throughout the entire search space without getting stuck in local optima. If PSO begins to converge toward a local optimum, TS forcefully diversifies the search by exploring nearby neighborhoods for a better solution.

To evaluate the feasibility of the approach, a proof-of-concept test involving the OGRE avatar was employed. A WebGL application was developed using JavaScript to measure the performance increase of PSO-TS compared to PSO. The OGRE Algorithm Comparison Tool served as a test bed to examine the differences in the algorithms and quantify the performance increase of PSO-TS. Testing showed that PSO-TS improved upon the previous OGRE PSO mechanism and successfully found the global optimal solution in all test cases. As well as demonstrating quick convergence speeds, rigorous testing also proved PSO-TS to be a much more reliable method when applied to multi-parameter optimization in a large search space. Overall, PSO-TS outperformed PSO in every aspect except time required to execute all 250 iterations of the algorithm. Thus, the hybrid algorithm verified the hypothesis presented and overcame the premature convergence of PSO.

7.2 CONCLUSIONS

From the results obtained, it can be concluded that PSO-TS greatly outperforms PSO in this specific application. However, a different perspective is taken in this section and our work's scalability to the real world is analyzed. The conclusion drawn from the observations made throughout this thesis indicate that PSO alone is not a reliable means for successfully converging to a true optimal solution within a large multi-parameter search space. Although PSO seemed to successfully converge by visual inspection of the OGRE test faces, the analysis of the fitness function outputs for each test proved otherwise. This is an important aspect to be considered when dealing with real-world problems because a partially optimal solution is not acceptable in many cases. Instead, the true global optimal solution is required. When dealing with a search space that contains higher dimensions than the one presented in this thesis, PSO will inevitably fail in almost all applications. Its premature convergence allows the algorithm to assume it has found the optimal solution in the search space and prematurely terminate the search.

On the other hand, PSO-TS successfully converged to zero in all cases. Thus, when applied to a real-world application, it is more likely to adequately find the true optimal solution and avoid premature convergence, despite a large search space. In systems that require a higher dimensional search space than what was considered in this thesis, PSO-TS will more than likely succeed because of its robust, hybrid architecture. As far as directly applying our PSO-TS algorithm to other applications, the transfer should be rather unproblematic due to our modular approach and design.

7.3 FUTURE RESEARCH

Because the mechanism produced in this thesis serves as an intermediate step in the long-term goal, there is much room for improvement and future research. By using PSO-TS and the modular architecture of our OGRE Algorithm Comparison Tool, future researchers can easily transfer our mechanism to an updated system geared toward facial expression recreation in an avatar. With respect to the long-term goal presented in this thesis, the updated mechanism can use a photograph of a human subject in place of our target avatar face and a human-like avatar intended to represent that individual in place of our working avatar.

After verifying PSO-TS's robustness, a more detailed human-like avatar may be used. Because of PSO-TS's reliable convergence performance, the algorithm should be able to properly converge to the optimal solution, despite a higher dimensional search space than the one used in this thesis. This allows for a more intricate, realistic recreation of facial expressions in the like-like avatar. PSO-TS's architecture also allows the mapping of additional Action Units. Thus, the musculature of the avatar can be intricately controlled to reveal a more realistic recreation of the intended facial expression. Overall, the findings of this thesis allow for the expansion of our system into a pixel-by-pixel based mechanism that can facilitate the transfer of a personalized facial expression of a specific human being in the form of a photograph or video onto a detailed human-like avatar intended to represent that individual.

LIST OF REFERENCES

[1] C. Puklavage, A. Perila, A. Gonzalez, and M. Georgiopoulos, "Imitating Personalized Facial Expressions in an Avatar through Machine Learning," Intelligent Systems Laboratory at the University of Central Florida, 2010.

[2] Z. Ruttkay, J. Hendrix, P. Hagen, A. Lelievre, H. Noot and B. Ruiter, "A Facial Repertoire for Avatars," presented at Workshop on Interacting Agents, 2000.

[3] S. Krinidis and I. Pitas, "Statistical Analysis of Human Facial Expressions," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 3, pp. 241-260, 2010.

[4] P. Ekman and W. Friesen, *Facial action coding system: A technique for the measurement of facial movement*. Palo Alto, CA: Consulting Psychologists Press, 1978.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, pp. 1942-1948, 1995.

[6] F. Glover, "Tabu search: Part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190-206, 1989.

[7] F. Glover, "Tabu search: Part II," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4-32, 1990.

[8] OGRE – Open Source 3D Graphics Engine, 2012, http://www.ogre3d.org.

[9] S. Lee, "Immortal avatars: Back up your brain, never die", 2010, http://www.evl.uic.edu/core.php?mod=4&type=4&indi=705

[10] L. Zalewski and S. Gong, "2D statistical models of facial expressions for realistic 3D avatar animation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 217-222, 2005.

[11] H. Mori, K. Oshima, and M. Nakamura, "Generating avatar's facial expressions from emotional states in daily conversation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5720-5723, 2011.

[12] C. J. Lin, C. H. Chu, C. Y. Lee, and Y. T. Huang, "2D/3D face recognition using neural networks based on hybrid taguchi-particle swarm optimization," *Eighth International Conference on Intelligent Systems Design and Applications*, vol. 2, pp. 307-312, 2008.

[13] I. Mpiperis, S. Malassiotis, V. Petridis, and M. G. Strintzis, “3D facial expression recognition using swarm intelligence,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2133–2136, 2008.

[14] S. Yang and B. Bhanu “Facial expression recognition using emotion avatar image,” *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, pp. 866-871, 2011.

[15] D. Cosker, E. Krumhuber, and A. Hilton, “A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling,” *IEEE International Conference on Computer Vision (ICCV)*, pp. 2296-2303, 2011.

[16] S. Velusamy, H. Kannan, B. Anand, A. Sharma, and B. Navathe, “A method to infer emotions from facial action units,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2028-2031, 2011.

[17] Y. Dong, J. Tang, B. Xu, and D. Wang, “An Application of Swarm Optimization to Nonlinear Programming,” *Journal Computers & Mathematics with Applications*, vol. 49, no. 11-12, pp. 1655-1668, 2005.

[18] R. Thangaraj, M. Pant, A. Ajith, and B. Pascal, “Particle Swarm Optimization: Hybridization Perspectives and Experimental Illustrations,” *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5208-5226, 2011.

[19] A. Bekrar, S. Chaabane, D. Trentesaux, A. Bornschlegell, J. Pelle, and S. Harmand, “Hybrid PSO-tabu search for constrained non-linear optimization problems,” presented at International Conference on Swarm Intelligence, Cergy, France, 2011.

[20] Y. Zhang and L. Wu, “A Hyrbid TS-PSO Optimization Algorithm,” *Journal of Convergence Information Technology*, vol.6, no.5, pp. 169-174, 2011.

[21] J. Li, Q. Pan, S. Xie, B. Jia, and Y. Wang, “A hybrid particle swarm optimization and tabu search algorithm for flexible job-shop scheduling problem,” *International Journal of Computer Theory and Engineering*, vol. 2, no. 2, pp. 189-194, 2010.

[22] B. Mhamdi, K. Grayaa, and T. Aguili, “Hybrid of Particle Swarm Optimization, Simulated Annealing and Tabu Search for the Reconstruction of Two-Dimensional Targets from Laboratory-Controlled Data,” *Progress In Electromagnetics Research B*, vol. 28, pp. 1-18, 2011.

[23] Y. Zhand and L. Wu, “Tabu Search Particle Swarm Optimization used in Cluster Analysis,” *Journal of Science*, vol. 1, no. 1, pp. 6-12, 2012.

[24] P. Ekman, “Facial expression and emotion,” *The American Psychological Association, Inc.*, vol. 48, no. 4, pp. 384-392, 1993.

[25] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, “Classifying facial actions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974-989, 1999.

[26] B. M. Ghandi, R. Nagarajan, and H. Desa, “Particle swarm optimization algorithm for facial emotion detection,” *IEEE Symposium on Industrial Electronics & Applications*, vol. 2, pp. 595-599, 2009.

[27] M. S. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. Movellan, “Machine learning methods for fully automatic recognition of facial expressions and facial actions,” *IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 592-597, 2004.

[28] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett, “The computer expression recognition toolbox (CERT),” *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, pp. 298-305, 2011.

[29] A. Teller and M. Veloso, “Algorithm evolution for face recognition: What makes a picture difficult,” *IEEE International Conference on Evolutionary Computation*, vol. 2, pp. 608-613, 1995.

[30] G. Morley and W. Grover, “Tabu Search Optimization of Optical Ring Transport Networks,” *Proceedings IEEE Globecom 2001*, vol. 4, pp. 2160-2164, 2001.

[31] R. Chelouah and P. Siarry, “Tabu Search Applied to Global Optimization,” *European Journal of Operational Research*, vol. 123, no. 2, pp. 256-270, 2000.

[32] F. Glover, “Tabu Search for Nonlinear and Parametric Optimization (with links to genetic algorithms),” *Discrete Applied Mathematics*, vol. 49, pp. 231-255, 1994.

[33] J. Cheng and R. Fournier, “Structural Optimization of Atomic Clusters by Tabu Search in Descriptor Space,” *Theoretical Chemistry Accounts*, vol. 112, pp. 7-15, 2004.

[34] J. Ghent and J. McDonald, “Holistic facial expression classification,” presented at SPIE 5823, Opto-Ireland 2005: Imaging and Vision, Dublin, Ireland, 2005.

[35] L. Diosan and M. Oltean, “Evolving the Structure of the Particle Swarm Optimization Algorithms,” *Proceedings of the 6th European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 25-36, 2006.

[36] Q. Bai, “Analysis of Particle Swarm Optimization Algorithm,” *Computer and Information Science*, vol. 3, no. 1, pp. 180-184, 2010.