HARVEY MUDD COLLEGE **Computer Science**

Final Report for
MITRE Corporation

Digital Aging

June 15, 2016

Team Members
Vincent Fiorentini (Project Manager, Spring)
Hana Kim
Josh Petrack
Hannah Young (Project Manager, Fall)

Advisor
Lisa Kaczmarczyk

Liaisons
Nick Orlans
Jay Doyle
Adam Day
Sarah Doyle

Abstract

The MITRE Corporation manages Federally Funded Research and Development Centers for the United States government. Harvey Mudd College is a liberal arts college of science, engineering, and mathematics in Claremont California. This work is a Harvey Mudd Clinic project supported by the MITRE Innovation Program, and is the fourth such project between the Harvey Mudd Clinic program and The MITRE Corporation.

Of ongoing interest at MITRE are the interrelated fields of biometrics, forensics, and identity, which utilize techniques and technologies such as DNA, fingerprints, facial images, and other methods of identifying people for purposes of both identification and authentication. A challenge in this area is how to aid the search for missing children. According to public data, there have been over 450,000 reports of missing children every year for the past several years (National Center for Missing and Exploited Children, 2016). Forensic artists have been working on manual digital aging techniques to aid the search and possible re-identification of missing children. However, the limitations of manual techniques led to questions of how the process might be automated. To address this issue, our Clinic team has researched existing methods of automated aging and then implemented an algorithm to simulate aging in face images, particularly for children. The algorithm takes a face image and produces an output image of the subject aged by the desired number of years. We then evaluated the perceived ages and photorealism of our output images with a user study to analyze the effectiveness of the algorithm.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Company Background

The MITRE Corporation manages and operates research and development centers sponsored by the federal government. Among their fields of research are the interrelated fields of biometrics, forensics, and identity that seek to develop techniques and technologies for identifying and authenticating humans. Significant advances in these fields have been made in recent years, but one area that continues to pose a challenge is maintaining accurate identification of a person as they age. Further research in this area, specifically aging a picture of a long-missing person to show how they might look now, can aid in the search for missing children. Although forensic artists can manually age a face image, MITRE wishes to assist with evaluation, validation, and possible steps toward automating the process. This work demonstrates an automated geometric growth model derived from Farkas measurements that represent a class of age related changes in young subjects.

1.2 Problem Background

The identification of missing children, especially those who have been missing for longer periods of time, continues to be a challenging and unsolved problem. Although photos provide a way for the public to help identify and find missing children, the task of recognizing the missing subject becomes more difficult if the photographs used are several years old. The most common approach to provide a more accurate photograph of what the subject might look like today is to have a skilled forensic artist manually modify the old image of the missing individual to make the subject appear older. While this process, also known as manual age progression, provides one solution to the problem, it also comes with several limitations. The process of manual aging is usually very time-intensive and difficult to verify the accuracy of because different artists might change different features on the same subject. Different images may also require completely different aging techniques because aging is such an individualized process and different factors may need to be taken into account for different individuals. We seek to offer an improved solution to this problem.

1.3 Problem Statement

Our Clinic team has automated digital face image aging and evaluated the photorealism of these modified images.

The goal of this project was to research the existing methods of digital aging to create our algorithm that simulates aging, and then to evaluate our algorithm with a user study on Amazon Mechanical Turk. In particular, we have created an algorithm that can be accessed through Adobe Photoshop (TM) to simulate the effects of aging on face images of adolescent subjects ages 6 through 19. This algorithm is applicable to the use case of helping identify missing children.

1.4 Project Requirements

In our initial statement of work, our team promised to complete and deliver a number of items by certain deadlines. We have successfully completed all deliverables by the stated times.

1.4.1 Objectives

We have completed the following objectives:

1. Decide what image processing techniques we will use to simulate aging. In order to do this, we researched various existing methods for age-progressing face images to gain an understanding of different relevant approaches. Such methods include those used in forensic art, manual aging via image processing software, and prior work on automated age progression.

2. Create an Adobe Photoshop (TM) script that simulates the effects of aging on a face image. When given a face image, the script returns an output image artificially aged by a given amount. The script takes the subject's initial age and sex as input information. We focus on simulating aging on face images of adolescent subjects.

3. Create a user manual that explains how to use the script. This manual contains information on how the script works, how to run the script, what input the script expects, and the intended use cases for the script.

4. Use Amazon Mechanical Turk to evaluate the photorealism and apparent aging of the script's results. This evaluation has been and can be further used to inform changes to the algorithm. With this evaluation, we have assessed how much our script can aid in existing identification challenges.

In addition, our team accomplished the following stretch goals:

1. We provided a command-line interface with which the user can interact with our algorithm. Through this command-line interface, we offer the ability to use our algorithm in batch runs.

2. Our algorithm also can perform age regression in addition to age progression.

1.4.2 Schedule

The team has completed all milestones on schedule. These milestones are as follows:

- By October 16, 2015, we were able to use Photoshop manually to age a face image. We also decided on the image processing techniques to use in our script. Tasks for this milestone included the following: completing beginner Photoshop tutorials, reading academic papers regarding image processing techniques for aging face images, and using Photoshop manually to age a face image.

- By November 23, 2015, we decided on an architecture for our script and implemented the internal interfaces for different modules in the script. Details of this architecture can be found in Appendix D.

- By December 14, 2015, we completed the initial prototype of our script using JavaScript. When given a face image, the script returns that face image digitally aged, though with a possibly low level of photorealism. We also prepared our script and user manual so that a non-Clinic-team user could run the script. Details of the script's functionality can be found in Chapter 3.
- By January 18, 2016, MITRE reviewed our script and user manual and suggested revisions and improvements.
- By February 1, 2016, we reviewed feedback returned by the MITRE liaisons regarding our previous milestone. We planned how to incorporate changes to our script and manual based on the feedback.
- By March 14, 2016, we improved the photorealism of our script. In addition, we set up a system for collecting data regarding the photorealism of our script via Mechanical Turk.
- By April 14, 2016, we completed improvements to our script as suggested by the Mechanical Turk data. We also completed an analysis of the Mechanical Turk data. A full discussion of the Mechanical Turk study is in Section 5.2.
- By May 3, 2015, we completed the user manual, which can be found in Appendix A.

1.4.3 Deliverables

Over the course of this Clinic project, our team has delivered the following items to MITRE:

1. A Photoshop script that alters a given face image so as to simulate aging of the subject.
2. Data that evaluates the photorealism and apparent age of the face images produced by our script. This data evaluates how humans perceive the photorealism of the modified images as well as the perceived age of the subject.
3. A user manual for our script. This manual explains how a user with a compatible version of Photoshop can run our script on a face image to simulate aging.

4. A standalone Windows executable that runs our age progression algorithm. This executable can be run in an automated batch process to produce comparisons for automated facial recognition tools.

Chapter 2

Research

In this section, we review a sampling of existing research methods for simulating facial aging. We also review some image processing tools.

2.1 Existing Methods

We explored different methods from the literature to simulate face image aging. Before committing to a final algorithm, the Clinic team spent time researching the main methods proposed by various other researchers regarding how to digitally age faces. We hoped to draw inspiration from one of these methods and implement something similar. After reviewing a wide variety of papers on the topic, the team narrowed our search down to four papers that seemed most relevant to the problem definition. The models that these algorithms are based on can be categorized in one of two ways–as either a Textural Model or a Craniofacial Growth Model.

2.1.1 Textural Models

Textural Models are based on identification of textures in the face that represent aging, such as wrinkles. They are also data-driven in their design.

1. Kemelmacher-Schlizerman et al. from the University of Washington utilize a concept called collection flow (Kemelmacher-Shlizerman et al., 2014). This method does not rely on the image in question being a face but instead relies on a more general theory of progression. The method identifies changes in pixel-darkness for many faces at a

given age to develop an average face for that age. That average is then applied to the face being aged. Doing so may have a blurring effect.

2. The model presented in "A Multi-Resolution Dynamic Model for Face Aging Simulation" by Suo et al. uses a tree based on different features, as well as hidden markov models (Suo et al., 2007). It uses probability sampling. This allows for creation of multiple different aged face images. However, it is designed to be more useful with adult faces than youth faces.

3. Lanitis et al.'s "Toward Automatic Simulation of Aging Effects on Face Images" is a seminal work (Lanitis et al., 2002). It uses a statistical model that breaks the face down into parameters and creates an aging function based on those. Again, it is geared towards adult faces.

2.1.2 Craniofacial Growth Models

Craniofacial growth is the normal way in which the bone structure of a human face ages. This means changes such as the face lengthening, eyes coming closer together, etc. These changes are different in child age progression and adult age progression. This is because the aging process is much more drastic during early childhood and through adolescence.

Ramanathan and Chellappa from the University of Maryland use a craniofacial model to simulate face aging (Ramanathan and Chellappa, 2006). They compute transformations based on anthropometric data collected from faces similar to the face in question. Their method uses identification of landmarks to note where features currently are. Then, it uses data from craniofacial norms to identify where those features would be at a certain age. It calculates growth parameters to describe the change from the original image to the aged image. Additionally, Ramanathan and Chellappa first explain a simplified version of their model, then add complexity, which we viewed as desirable because it would allow us to first emulate their simplified model, and then add complexity as we improved our algorithm.

2.2 Image Processing Tools

One requirement of our project was making our algorithm accessible through an existing image processing tool. This is important to MITRE because it is desirable that our work eventually become available to a wider community of people without technical knowledge of our model or of computation in general, such as forensic artists. We considered GIMP and Photoshop.

2.2.1 GIMP

GIMP, which stands for GNU Image Manipulation Program, is an open-source image-processing tool that is easily extensible through plug-ins and scripting. It has gained popularity because it is free to the public and has many features similar to those of Photoshop. While GIMP stood out as one possibility for our project because of its ease of accessibility to the general public, we ultimately decided against it for several reasons. Although GIMP supports powerful scripting, the language for scripting is C and the scripting is difficult to build on Windows. In addition, even though it has gained popularity, it is still not as well known compared to Photoshop.

2.2.2 Photoshop

Our team also considered using Photoshop for a few reasons. First, Photoshop is currently the industry-standard when it comes to image processing. Many people are familiar with the interface and features, and there is an extensive amount of documentation and tutorials available to the public. This better fits our desire for our algorithm to be accessible to larger audiences. Photoshop offers three scripting languages: JavaScript, AppleScript, and VBScript. Of these, we decided that JavaScript would be the best option because it is cross-platform and more natively supported by Photoshop. Adobe has also published a detailed manual of the tools available to Photoshop script developers, which allowed us to quickly determine what we could do.

2.3 Conclusions

After reviewing the literature surrounding this area and the image-processing tools available to us, we opted for an algorithm based on a craniofacial growth model like the one presented by Ramanathan and Chellappa. This model was within the scope of our project, applies well to children, and should allows incremental improvements.

We opted to make a Photoshop script because of Photoshop's popularity, its ease of use, and its powerful tools offered in the API which seemed to more similarly match our needs for our script.

Chapter 3

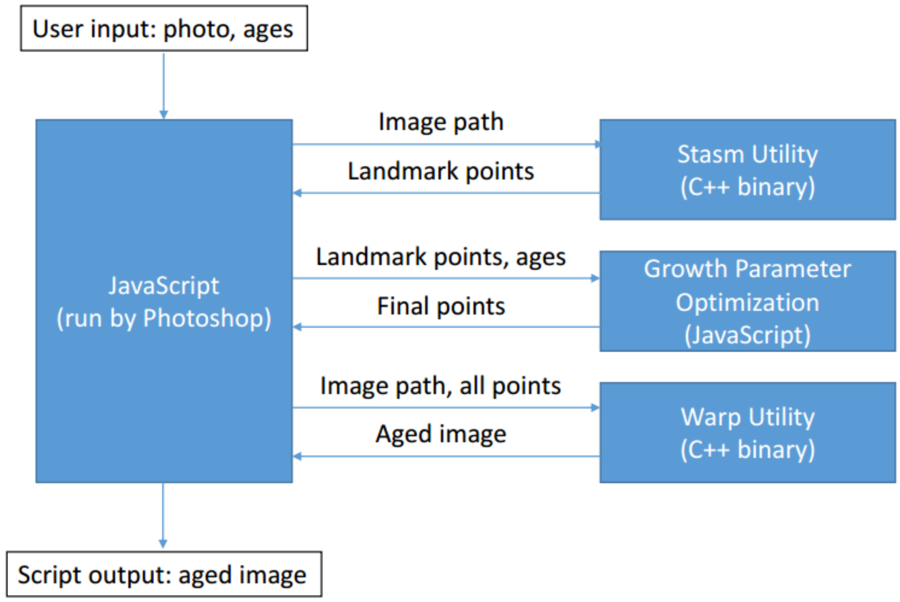
Project Prototype

In the Fall semester, we created a prototype version of our algorithm. This prototype had many of the functions of our final algorithm, but used a more simplistic model for aging. In this section we first present an overview of our model. Then, we discuss landmark detection, the different models we have used, and warping. Finally, we discuss other considerations in development of the prototype.

3.1 Overview

After carrying out our research during the first semester, we created our initial aging algorithm. The algorithm is based on the first model presented by Ramanathan and Chellappa in their paper (Ramanathan and Chellappa, 2006). The algorithm we created is designed to first detect a point at the center of the forehead to act as an "origin". Using this point, it then stretches the rest of the face outward based on each point's angle from a vertical line through the center of the image. To accomplish this, the algorithm detects a number of key points around the face, warps those points, and then interpolates the rest of the image according to the transformations on those points. This algorithm was implemented as a Photoshop script using JavaScript. Refer to Figure 3.1 for an overview of how these different parts of the script work together.

Figure 3.1 An overview of the architecture of our prototype.



3.2 Landmark Detection

The first step that this algorithm takes is to find key points around the face, so that it can determine the origin for warping and have a sampling of points around the face to warp outward. To do this, it uses the Stasm utility (Milborrow, 2009). Stasm is a publicly available utility based off of OpenCV libraries which, given a front-view image of a face with a neutral expression, will return the positions of 77 key features around the face. These are features such as inner corners of eyes, cheekbones, and jawline. These are landmarks that the Stasm creators have deemed important. Stasm can work on images that are not taken from a direct frontal view or that have subjects with other expressions, but may suffer in accuracy. Stasm is useful to us because it returns a large number of key points, so we have freedom to modify face images at a fairly intricate level of detail.

In Figure 3.2, we can see two different depictions of the Stasm points identified. In the left image, there are lines drawn between them, while on the right we see the same points just numbered. It is noteworthy that Ramanathan and Chellappa's method did not use Stasm for this purpose; it used a different method of detecting facial landmarks and used fewer

Figure 3.2 Example of the 77 Stasm points detected in an image.

Source: *www.milbo.org/stasm-files/stasm4.pdf*

total landmarks. We elected to use Stasm instead of attempting to use the particular method used by Ramanathan and Chellappa, as Stasm is a high-quality alternative for detecting facial landmarks.

Our script calls an executable version of the Stasm utility, which then stored the coordinates of the resulting landmarks to a text file. The script then reads from this text file to determine the coordinates of the points. The center of the forehead is not one of the points that Stasm returns, so to determine it, the script averages the coordinates of points to its left and right at approximately the same height.

3.3 Fluid-Filled Sphere Model

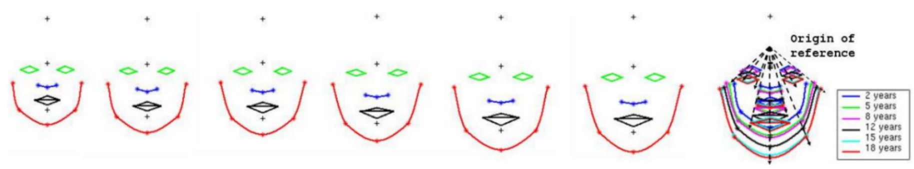
Ramanathan and Chellappa identify three key constants as necessary for a model of craniofacial growth to look realistic:

- The angular coordinates of each point with respect to the origin is held constant, that is, points only move directly towards or away from the origin.

- Bilateral symmetry is maintained.
- The function warping the image is continuous.

The fluid-filled sphere model is based on a model of the human head as a fluid-filled sac that holds all three of these properties. As people age, the sac gets larger, but this growth is directed radially outward with points closer to the horizontal center of the face and points farther from the origin moving farther.

Figure 3.3 Facial growth based on the fluid-filled sphere model.



This model is straightforward as it only needs a few simple equations to describe it. As a result, we viewed this model as a good first step to implement so that we could test the various parts of our system together. A visual explanation of what the model does to a face as it ages is shown in figure 3.3.

However, the model is too simple to account for actual craniofacial growth accurately. In general, images produced with this method appeared somewhat older, but at the cost of realism, as the images were typically ballooned outward based on how much they were aged. This was true both in the paper and in our own implementation of the algorithm. As a result, proportions in the resulting faces were not accurate to those in real faces. We decided to only implement this algorithm as a first step and eventually improve the algorithm based on the rest of Ramanathan and Chellappa's work. After presenting this simplified model, Ramanathan and Chellappa present a more complex and accurate method based on the simplified model. As this model was more complicated and we wanted to have a working prototype by the end of the semester, we decided to implement this more complex model in the Spring semester.

3.4 Warping

The warping utility is given input points and target points to warp from and to, respectively. In order to accomplish this displacement of points, we leverage an existing algorithm that uses the OpenCV library and warps images based on moving least squares using linear functions (Schaefer et al., 2006). The three types of transformations available were affine, similarity, and rigid. An affine transformation preserves points, lines, and planes; a similarity transformation is meant for shape preserving; and a rigid transformation preserves length. The paper that the algorithm was based on, called "Image Deformation Using Moving Least Squares," suggested using rigid transformation for realistic shapes (Schaefer et al., 2006). After testing all three types of transformations, we concluded that the best results did indeed come from the rigid transformation, and thus chose this option for our script. (The other two transformations resulted in bumpiness around the edges of the face.)

Figure 3.4 The effects of the warping utility.

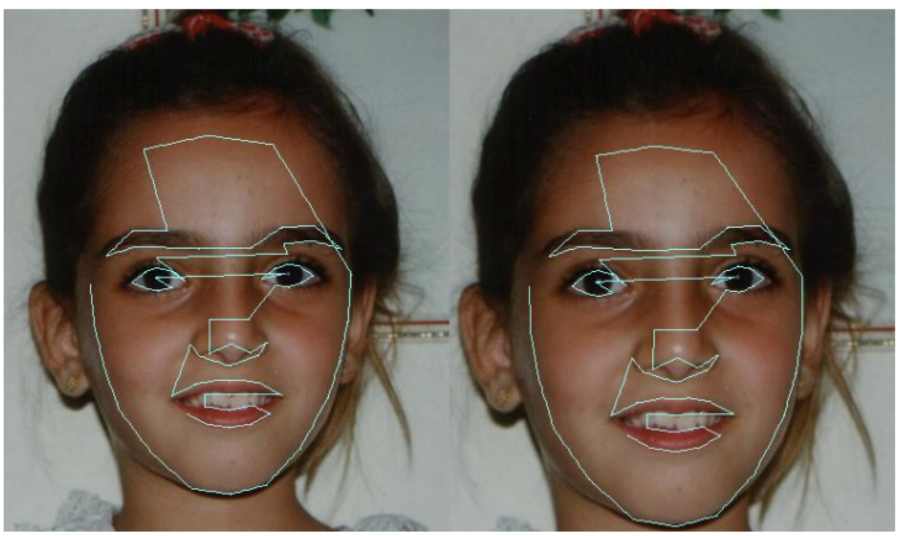


Figure 3.4 shows an example output from the prototype before border points and rescaling. This figure shows the input image (with Stasm landmarks) on the left and the age-progressed output image warped from the input image (also with Stasm landmarks) on the right.

3.5 Other Considerations

With this prototype algorithm, we noticed warping in other parts of the image, such as the forehead, ears, neck, and background. Because we only want the face to appear aged and the rest of the head and background to remain the same, we added "border points" around the face to prevent anything outside of the face from being warped.

We added border points at the circumference of the face in order to reduce warping beyond the face. We accomplished this by getting the Stasm points at the edges of the face (points 1 through 15) and shifting these points by a small constant factor (0.03) from the center of the face obtained from averaging the four points at the four corners of the face: the top-most (point 14), the bottom-most (point 6), the left-most (point 1), and the right-most (point 11). For reference, these points can be see in Figure 3.2. We added this new set of border points in addition to the 77 points originally recognized on the face by Stasm and told our algorithm to map these border points to themselves. In other words, we do not warp any of these border points, which would consequently tell the points beyond these border points not to warp themselves either. This change fixed the problem of having warping artifacts outside of the face region.

We also noticed that the fluid-filled sphere model often resulted in the face in the image growing beyond the four borders of the image frame. This was to be expected because of the nature of the fluid-filled sphere model—all points are warped outward from a specific origin point in the middle of the face, and for larger differences between input age and target age, all points are displaced by a large amount and may be forced to go beyond the borders of the image.

To prevent the face growing beyond the borders of the image frame, we added rescaling. This rescaling first finds the height of the original face and the height of the age-progressed face by calculating the distance between a Stasm point known to be at the top of the face and another Stasm point known to be at the bottom of the face. Then, we rescale the set of destination points by the ratio of the original height to the output height about the center of the destination points, which ensures the height of the age-progressed face matches the height of the original face. This results in a reasonable amount of growth in the face while still allowing the head to remain within the borders of the image frame.

Chapter 4

Final Project

In the Spring semester, we improved our aging algorithm and converted our project to a command-line executable written in C++.

This chapter discusses the structure of the C++ project, the Photoshop script we created to interface with this project, and the capability of this project to be executed in batch runs.

4.1 C++ Project

Our main project is an executable built from a Visual Studio C++ project that can be called from the command line in Windows or via a JavaScript script we wrote for Photoshop. We chose this setup because the Stasm library and the image warping library we used are only available as C++. Both of these libraries rely on OpenCV's C++ library. For more information about OpenCV, their website can be found at opencv.org.

The executable has three main components: landmark detection, growth parameter optimization, and warping.

4.1.1 Landmark Detection

We detect facial landmarks by using the Stasm library, a publicly available tool built on OpenCV. We used this tool in our original prototype, and our final project uses Stasm in the same way. Once we use Stasm to detect the landmarks on a face image, we use the landmarks to calculate the origin of the face, which we consider to be between the eyebrows. After that, the algorithm determines where each landmark would most likely be if the subject were aged by the given amount. Finally, we warp the image from the original set of landmarks to the age-progressed set of landmarks. The primary way in which our final project differs from the prototype is that the final project uses a more sophistocated algorithm to determine where each landmark should be in the age-progressed image.

4.1.2 Growth Parameter Optimization Model

Growth parameter optimization is the name we have given to our model's process for determining where each facial landmark should be translated. This model is largely based on the method described by Ramanthan and Chellappa, with many changes. We now present a detailed explanation of our growth parameter optimization algorithm.

Growth Parameters

As with the original simplified model, the growth parameter optimization model maintains three key invariants for each point and for the face as a whole: each point's angular coordinate, the face's bilateral symmetry, and continuity. However, with the original model, each landmark was simply scaled outward from the origin by a factor depending on its angular coordinate and the amount by which the subject was being aged. With the new model, we compute an individual growth parameter for each relevant Stasm point. As of now, we only use a subset of the Stasm points for reasons that will be explained in the next sections. These individual growth parameters tell the algorithm how far each point should be moved along the line between the origin and it. A growth parameter of 0 means that a point will stay stationary, while a growth paramter greater than 0 will multiply that point's distance from the origin by a factor of one plus the point's growth parameter.

In the simplified model, effectively, every point had a growth parameter depending only on the amount the face was being aged and that point's angular coordinate. With our new model, growth parameters can be much more varied, which allows us to more accurately capture the aging process. By freely allowing the algorithm to determine the best growth parameters, the simplified model becomes much more general.

Data Set

In order to calculate the growth parameters, we use empirical data to determine how much certain distances in the face increase as individual people age. It is difficult to simply measure where the points should go directly, as there is no reference of distance within a face image. For instance, if we determined that the tip of the nose should move 1 centimeter away from the center of the forehead in absolute distance, we would not be able to make that modification easily because we do not know how many pixels are in a centimeter. However, Ramanathan and Chellappa's method allows us to determine how much different pairs of points should move apart relative to the distance that they began with.

As such, our data for this purpose is from Farkas's *Anthropometric Facial Proportions in Medicine* (Farkas and Munro, 1987). The appendix we used gives average distances in millimeters for various pairs of points around the face, for male and female Caucasians from age 0 (subjects less than three months old) to 19 or more. Typically, the distances are reported for age 0, age 1/2, each age from 1 to 18, and a group for age 19-25. In order to use this data more easily, we ignored the data for age 1/2 and used the data for ages 19-25 as data for age 19. Since our focus is on the aging of children by several years, these assumptions have only a small adverse effect on the capabilities of our algorithm.

We were able to identify a set of roughly thirty distances within the Farkas data that have three properties we deemed necessary for our project. These properties are necessary for our algorithm to work with the landmarks detected by Stasm in frontal face images and are as follows:

- The distance had to be reliably measurable from frontal face images, so we could not use any data pertaining to ears, for instance.
- The points involved in the distance had to be close to Stasm points so that we could approximate their distances by looking at the distances between the corresponding Stasm points.
- The distance had to be a straight-line measurement, rather than an angular measurement or a curved distance.

Many of the distances that have these properties lacked data for age 0 or for ages 0-5. Because of this missing data, our project currently only supports aging from age 6 or older. As a workaround for this limitation, our project shifts the original and target ages if given an original age younger than 6. For example, if our algorithm is told to age a subject from 4 to 7, we instead age from 6 to 9. In this way, the subject is aged by the same number of years as desired but not with the desired original age. We deemed this workaround necessary because we saw certain aberrations in some images that were aged without utilizing the distances that only had data for ages 6 and older. In particular, the jawline was not straight for some of these output images.

While we only chose distances that can be measured from frontal images, these distances are not perfect for our needs because our reference data measures them in three dimensions. For instance, a distance from the tip of the nose to the cheek may be larger in reality than in a 2-dimensional projection because the tip of the nose is forward from the cheeks. To quantify this bias, we compared the 3D and 2D distances in one subject's face to the data provided by Farkas. This comparison can be found in Appendix C. To summarize this comparison, our example showed relatively minimal differences between the 3D and 2D distances we measured, indicating that our use of 3D reference data in 2D image space is acceptable.

Minimizing Least Squares

In order to utilize the data from Farkas, we use a nonlinear optimization process to determine the best growth parameters to match the empirical data. Given a set of growth parameters, our algorithm computes the new distances between each pair of points that has a corresponding distance from Farkas. It then calculates the ratio between this distance in the resulting image and the distance in the input image, and compares this ratio to the corresponding ratio caluclated from the Farkas data by taking the difference. This gives us the error for that distance ratio.

The objective of the optimization process is to minimize the sum-squared-error over all distance ratios for which we have data from Farkas. In order to carry out the optimization process, we use Alglib, an open-source library that implements a number of algorithms for optimization and similar problems. In particular, we use its minlm optimization, in which the lm stands for Levenberg-Marquadt, the underlying algorithm in the optimization. Ramanathan and Chellappa also made use of the Levenberg-Marquadt algorithm, but it is unclear exactly what they inputted into the algorithm to optimize.

In addition to this calculation, we also enforce two additional constraints. The first of these is symmetry; in order to enforce symmetry, for any pair of Stasm points that are symmetric across the face, we only use the value that was calculated for the Stasm point on the left side of the face, and use it to warp the points on both sides. However, our optimization currently makes use of the Farkas data which is typically somewhat asymmetric even for symmetric distances. This symmetry enforcement step is crucial, as without it we have more total points involved in distances than we have distances. This leads our system to be underconstrained, which led to highly unpredictable results. As soon as we implemented symmetry enforcement, we noticed a marked improvement in the consistency of our results' realism.

The second additional constraint is point grouping. In order to prevent unnatural warping of the eyes, mouth and nose, we group all points in each of these three sets together and average the calculated growth parameters for each group to get a growth parameter to use for all points in the group. Without these two additional constraints, our results were clearly not photorealistic. With these constraints, it is much more difficult to tell that the images our algorithm produces have been digitally modified since they are more photorealistic.

4.1.3 Warping

As with our landmark detection with Stasm, our method for warping points to their new locations is identical to the method we used in our initial prototype. This warping method is described in Section 3.4. Our warping method moves the points to their new locations and interpolates to determine where other points should go. It uses the new point locations calculated from our updated model.

4.1.4 Other Considerations

Although this algorithm based on growth parameter optimization gives us significantly more power than the simple fluid-filled sphere model of the original prototype, the algorithm alone has some undesirable side effects and does not completely achieve the desired effect. We now explain what these side effects are and how our project addresses these issues.

Border Points and Rescaling

One side effect of our algorithm is that it distorts the image outside of the face. This occurs when the landmarks at the edge of the face all move outward, which results in the rest of the image being enlarged in the warping process.

To address this issue, the algorithm calculates static "border points" around the edge of the face. These points are calculated by taking several points known to be at the edge of the face and scaling them outward from the origin by a small amount. These border points are then added both to the input landmarks and to the output landmarks in the warping step, which ensures the warping only occurs within the boundary of the face.

Another side effect of our algorithm is that the entire face tends to increase in size as a result of the age progression algorithm, which often results in the bottom of the face getting cut off by the bottom of the image.

To prevent the image from ballooning in this way, we rescale the output facial landmarks so that their width matches the width of the original landmarks. This ensures the face will not greatly increase in size.

Post Hoc Changes

We found that our algorithm did not appear to be aging as drastically as desired. To address this issue, we created two changes that can optionally be performed after the optimization step to increase the effect of the algorithm. Both of these steps increase the apparent effect of the aging, but potentially do this at the cost of photorealism. If either of these optional features are used with very large parameters, the resulting image will definitely look less realistic. Both of these parameters can be used to improve the algorithm but should be used carefully, as they may have deleterious effects on photorealism.

The first of these effects is to simply scale the calculated growth parameters by some constant factor. This increases the effect of the script because larger growth parameters correspond to modifying the face more. Even for fairly small factors, this can have a very noticeable effect. We recommend that this option be utilized with a factor of 1.35, as this value worked well in our testing and produced images that looked more drastically aged but still realistic. The default value for this parameter is 1, which does nothing.

The second of these effects is to rescale the face about its center by a ratio based on the difference between the input and output ages. This feature can be used to increase the size of the face. Similar to the first effect, this option has a noticeable effect even with a very small change, in this case, with an even smaller value. Again based on our testing, we recommend a value of 0.0025 for this parameter (which will be multiplied by the age differential by the algorithm to produce the actual value that is used). The default value for this parameter is 0, which does nothing.

4.2 Photoshop Script

We created a JavaScript script that can be called from within Adobe Photoshop Creative Cloud 2015. This script is responsible for taking in user inputs with Photoshop's user interface, calling the C++ executable, and placing the age-progressed image on a layer above the input image. Our prototype Photoshop script originally contained the age-progression algorithm, but we rewrote the C++ executable to contain everything needed for the age progression algorithm.

For more information on the Photoshop script, consult the user manual in Appendix A or the code documentation in Appendix D.2.

4.3 Batch Runs

Since our project is built as a command-line executable, it can be used in automated batch runs. Running our project on batches of images and using automatic tools to compute the similarity of our output images to actual ground truth images can help evaluate the effectiveness of our algorithm. With longitudinal data (that is, several images of the same subject at different ages), this combination of batch processing and automatic evaluation can be used to age the younger images and automatically compare the age-progressed output images to the actual ground truth images of the subject at the older ages.

This is one method that can be used to validate that the algorithm is having the desired effect of making individuals look older and, in particular, more similar to how they will look in a number of years. This is important for the use-case of missing children because the goal of the algorithm in this case is to make the image look more similar to how the child looks years later so that they can be identified.

Chapter 5

User Study

We performed a user study on Amazon Mechanical Turk to gauge the effectiveness of our algorithm by asking participants to provide feedback on perceived age and photorealism of the script's results. Mechanical Turk simplifies recruiting and paying workers, making it an ideal platform for our purposes. In this section, we overview our design of the user study, present the results, and discuss the impact of these results.

5.1 Design

We recruited 80 Mechanical Turk Workers in good standing to take surveys regarding our aging progress. We did not specify ages, gender, or ethnicity of our workers. Once their answers were approved, each participant was paid $3.50.

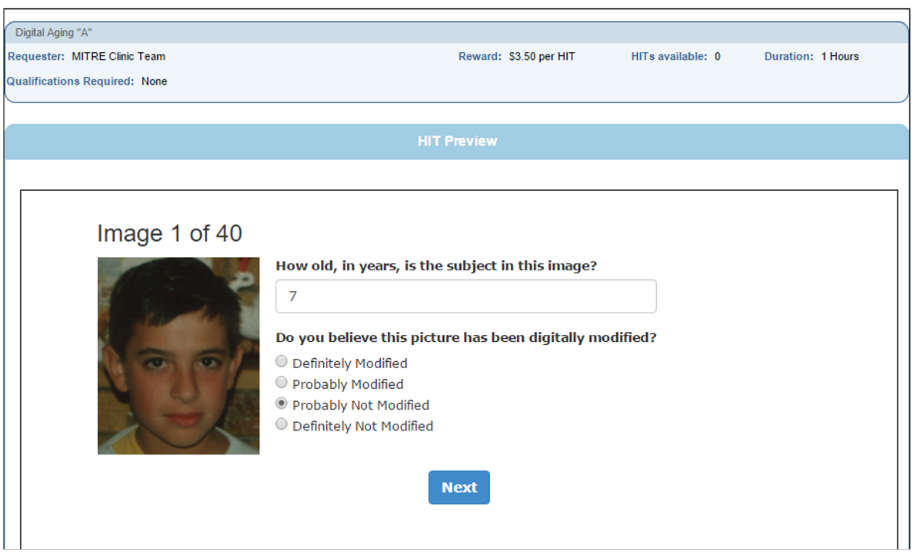
For the content of our user study, we chose 40 sets of face images from the FG-NET database. These are all of Caucasian males and females under the age of 18. There are two images of one person for each set: the ground truth image at the input age and the ground truth image at the output age. We also added the digitally aged image from input age to output age to each set. FG-NET is a database of images of people at various different years, which allowed us to compare the output of our script to actual photographs of subjects at the desired output age (Panis et al., 2015).

We broke up the user study into two separate surveys, with each version showing exactly 40 face images: 20 age-progressed face images (of 20 different subjects), 10 face images at the young/starting age (of 10 other subjects), and 10 face images at the old/target age (of the final 10 subjects).

Each of the surveys is taken by 40 Mechanical Turk Workers, for a total of 80 Workers.

The template for both surveys is the same: The user first sees the consent form, and must press the 'continue' button in order to proceed. Next, they see an instructions page that explains that they will be seeing 40 face images and answering two questions for each image: 1) How old does the subject in the image appear to be, and 2) How modified does this image appear on a 4-point scale. Once they hit continue, the first image and set of questions are shown. For each image, the user cannot press on the next button until they have answered both questions (how old and how modified). No image is shown more than once across the two versions of the survey. A sample page of the survey is shown in Figure 5.1.

Figure 5.1 Worker view from the Mechanical Turk study.



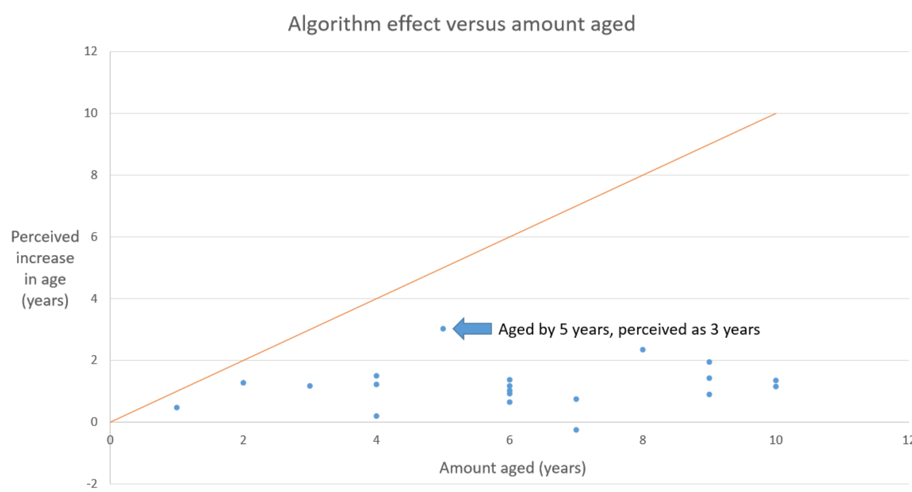
5.2 Results

We received 40 responses to each of our surveys, for 80 responses total. Because of the limitations of Mechanical Turk, we do not have data on the respondents themselves. On average, the images produced by our script were viewed as older than the input images by 24% of the amount that we specified they should be aged. For example, in the point indicated in Figure 5.2, we can see that this particular image was viewed as having aged by

3 years instead of 5 years. We carried out two-tailed t-tests for each pair of images to determine if the mean ages reported by our participants for each digitally aged subject and the subject's original image were statistically different. For 12 out of 20 pairs of images (8 male pairs and 4 female pairs), the difference was statistically significant at the 95% confidence level ($p < .05$). Realism scores were converted from the four multiple-choice text responses (see Figure 5.1) to a four-point scale ('Definitely not modified' being 1, and 'Definitely modified' being 4). We carried out testing for these values as well, and did not find the values to be statistically significantly different for those images that were aged by our script versus those that were not.

Our results are summarized for these 20 pairs of images in Figure 5.2. The orange vertical line represents the line where simulated age equals perceived age, the desired result for perfect simulation.

Figure 5.2 Summary of user study results.



5.3 Discussion

We now discuss the nature and implication of these results. We found that our algorithm works best for small age increases but overall doesn't increase the apparent ages of subjects enough. However, our algorithm does have a noticeable effect, and does not significantly sacrifice photorealism for this effect. We consider this a success, as it indicates that our script is capable of simulating aging at least to some degree without compromising realism. These results were used to inform our work on the script for the last few weeks of work, and led us to implement our post-hoc changes. These were features such as increasing the scale by which we moved the different facial landmarks and are discussed in section 4.1.4. As such, these results of the user study do not reflect the effect of those features. We believe with the implementation of those features, our output images will more closely match the actual age that they should appear as, while still maintaining good photorealism.

Chapter 6

Future Work

6.1 Project Strengths

As the project currently stands, we have both met and exceeded our original expectations in terms of the project deliverables. First, and most importantly, our algorithm shows statistically significant aging without compromising photorealism, as demonstrated through the data from our Mechanical Turk study. This was the highest priority of this Clinic project. Next, our code also meets good standards in terms of being modular and extensible, so that certain features not yet implemented (such as removal of baby fat) can be easily added in the future. Finally, the program can be run from both the command line and Photoshop. If the user chooses the former, then they can also execute batch commands of the program which enables large-scale testing. If the user is less technical and chooses the latter, the program is still easily accessible through the Photoshop user interface.

6.2 Limitations

There are several limitations to our algorithm. One of the issues we faced involves data limitations. We did not have access to a wide variety of demographics in either the Farkas data, which is limited to Caucasians, or the FG-NET images as most of the images in the database are of Caucasians. Given that this demographic data is instrumental to the aging algorithm of our program, it means that our program currently does not have the precision for non-Caucasians that it should. We are also unsure about the quality and quantity of some of the Farkas data, as data for certain ages is missing and the facial landmarks used by Farkas are slightly different than those used in Stasm's detection. Another concern is that Farkas data is in 3D space, but we calculate landmark distances based on 2D space, which may mean that landmark distances that are not vertically symmetric are not accurate distances for our purposes.

All of these issues can potentially be addressed all at once by implementing an automatic data collector. This collector could look through any database of face images annotated with age, sex and ethnicity, and use Stasm on each image and manually record the distances between each pair of the 77 landmarks. This would allow us to collect data on any face images we have available where we know the necessary data about the subject, and would bypass the problem of 3D and 2D space by working directly in 2 dimensions.

Another limitation is that the FarkasData.json file and StasmData path are not directly built into the executable, but instead their paths are passed in as optional arguments to the .exe. Thus, if either of these paths are incorrect, the script may result in a compilation error. This can be addressed by streamlining the script and integrating these features into the executable in the future.

6.3 Possible Improvements

In addition to those changes directly addressing our model's main limitations, our model can be improved in several areas. First, our algorithm does not correct for pose variations. These variations could be accounted for in the future by using data from the 3D camera. For instance, we might be able to determine the initial angle of the face using Stasm landmarks, then rotate all landmarks, perform growth parameter optimization as usual, and finally rotate all landmarks back. We also recommend the addition of textural features (wrinkles, baby fat removal, etc.) as these will increase the photorealism of our output images. We may also be able to improve our algorithm by using the landmark distances from parent(s)' faces to inform the growth parameters.

As for improving the user study, we would suggest performing additional testing. It appears based on our data that our script may be more effective for images of boys than for images of girls, although this conclusion is not entirely clear. The main weakness revealed in this study was that our algorithm was not aging its subjects drastically enough, which we have since addressed with post-hoc modifications. Further study is warranted to determine if these modifications are effective, either through additional user studies or through other means (such as with automated tools). Testing may also help to determine the best values of these post-hoc parameters to best balance the increased aging effect with the potential loss of photorealism.

Chapter 7

Acknowledgments

We would like to thank our sponsor, The MITRE Corporation, for giving us this socially impactful project and for providing useful technical advice and resources throughout the project. In particular, we would like to thank our liaisons Nick Orlans, Jay Doyle, Adam Day, and Sarah Doyle. Their technical advice and guidance were extremely helpful throughout this project, and their feedback upon testing our algorithm was valuable for our next improvements. We would like to thank our advisor, Lisa Kaczmarczyk, for her guidance, advice, and feedback. She helped us remain organized and focused throughout the course of the year. Finally, we would like to thank the clinic staff, particularly Geoff Kuenning and DruAnn Thomas, for making this project possible.

Appendix A

Photoshop Script User Manual

This document explains what the Photoshop script does, how to install and use it, and who might want to use it and why.

A.1 What the Script Does

The script can be run through Adobe Photoshop. When run, it takes in the following inputs:

1. An image of a child's face in `.jpg` or `.png` format
2. The age of the child in that image in years
3. A desired output age, also in years
4. The sex of the child

The script will then simulate aging on the provided image, showing the input image on one layer and the age-progressed image on a second layer.

A.2 How to Install and Use the Script

To use the script, you must have a copy of Photoshop Creative Cloud 2015 installed on a Windows computer. To install and use:

1. Extract the .zip containing the script to your computer to create a `script` folder
2. Open Photoshop Creative Cloud 2015

3. Within Photoshop, go to File > Scripts > Browse, and within the `script` folder select `age-progression.jsx`

4. Follow the prompts of the script.

Note that after aging an image, if you wish to use the script again, you should close the remaining image before doing so, as using the script while an image is open may result in errors.

A.3 Uses of the Script

The script's primary use is to aid in the search for missing children by automatically producing images of what a child who has been missing for several years might look like today. For this purpose, the script can be used by forensic artists as a starting point or as an additional reference, or the script can be used to produce a final product on its own. The script is most effective at aging pictures of children from age at least six up to age at most nineteen.

Appendix B

User Study Data

Here, we show the full tables of results from the user study. We show the results from both batches, for both boys and girls. The data is sorted by the subjects' identifiers (b1 through b20 for boys, g1 through g20 for girls). Because of the nature of the study, each image was presented as the original image in one study and the aged version in the other. Which version of the image that data column represents is noted. Each row corresponds to the ages given by a single participant from Amazon Mechanical Turk. For realism data, responses were converted to a four-point scale (definitely modified = 1, probably modified = 2, probably not modified = 3, definitely not modified = 4).

Table B.1 User study data for batch 1, male.

ID	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16	b17	b18	b19	b20
Mod.?	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
Input	8	9	11	8	5	12	5	18	6	6	11	16	5	9	6	5	8	17	7	7
Output	14	-	13	-	10	-	7	-	10	-	14	-	15	-	11	-	18	-	15	-
	16	11	10	8	10	15	9	17	11	9	9	18	9	10	7	8	11	18	13	8
	9	6	9	8	3	12	5	17	8	6	7	29	4	6	4	4	7	28	13	7
	16	8	11	10	6	12	14	21	11	7	12	17	9	13	12	7	14	17	14	10
	12	5	7	9	6	14	8	16	10	10	12	16	6	12	7	5	12	16	10	9
	14	6	8	7	7	12	9	16	9	8	8	18	7	10	6	5	8	19	16	9
	13	5	6	6	4	10	5	17	5	5	7	17	3	6	4	4	6	17	11	6
	9	8	6	5	6	12	6	14	7	4	11	14	6	8	6	5	6	16	13	9
	12	7	9	7	5	13	8	18	8	5	6	33	4	7	4	5	7	17	15	6
	13	6	9	9	5	13	7	15	7	4	7	20	6	6	5	5	7	16	13	7
	11	7	8	10	6	12	10	17	10	8	8	37	8	8	6	6	8	15	16	9
	11	7	8	7	6	12	10	17	9	7	8	37	5	9	4	6	7	16	12	7
	12	10	9	10	10	14	10	20	11	9	10	19	10	11	9	6	12	14	9	10
	10	11	12	8	8	15	8	17	8	8	10	16	11	12	7	9	12	18	15	9
	18	9	10	7	5	11	9	17	10	13	12	28	8	11	5	4	12	19	14	9
	14	7	8	7	5	16	9	16	10	7	7	18	8	11	6	5	12	17	14	8
	13	9	7	9	9	15	9	17	10	10	8	28	11	11	6	6	13	16	13	8
	14	9	12	7	7	12	8	23	8	7	7	18	7	10	9	7	15	17	17	11
	14	10	13	9	7	14	9	19	12	8	9	23	6	12	6	6	10	18	19	9
	14	7	8	8	3	14	8	18	11	8	10	40	6	6	8	6	10	16	15	10
	13	7	9	6	4	13	6	18	12	6	10	17	11	9	5	4	7	18	13	8
	16	8	10	12	6	12	12	18	12	8	10	20	8	12	7	9	11	18	25	10
	12	7	8	8	4	12	8	16	6	5	8	24	5	9	6	4	10	18	14	6
	19	5	7	8	4	14	8	23	9	7	7	33	6	10	6	5	9	18	14	10
	12	8	12	11	6	13	12	17	10	8	12	42	7	10	8	5	11	16	16	10
	13	6	7	5	5	13	7	17	8	5	7	17		8	4	3	12	23	12	6
	9	8	10	6	4	11	7	16	7	7	7	17	5	8	5	5	8	16	12	6
	10	6	7	7	6	11	8	16	10	5	11	23	8	10	6	5	8	15	12	7
	17	8	10	8	6	15	8	20	8	9	10	22	8	9	7	5	10	20	16	10
	13	6	10	8	9	15	14	26	10	5	11	44	9	10	5	4	9	25	15	11
	12	8	10	10	5	12	9	16	8	5	10	28	6	7	7	6	7	15	14	8
	11	7	8	7	4	11	8	16	8	7	7	15	5	7	5	5	12	16	12	7
	9	6	7	7	5	14	9	20	7	7	8	32	7	8	7	5	7	18	17	8
	11	7	10	9	7	13	11	31	8	7	8	26	6	8	6	8	9	16	14	9
	14	6	8	8	8	12	9	13	9	7	7	13	7	11	9	5	9	16	7	11
	18	7	10	7	5	20	10	28	7	6	9	27	6	7	5	6	13	27	16	6
	11	9	10	9	8	14	9	15	9	11	11	17	8	10	7	8	10	16	13	10
	18	8	10	8	8	18	12	19	10	12	12	35	12	12	10	6	10	22	12	8
	14	9	12	8	8	15	9	17	12	12	12	17	9	13	7	7	8	18	14	12
	13	12	10	10	11	14	12	17	11	12	14	34	9	9	9	11	11	16	17	12
	11	7	9	7	4	12	7	18	8	5	10	39	7	7	5	6	7	16	16	8

Table B.2 User study data for batch 1, female.

ID	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16	g17	g18	g19	g20
Mod.?	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
Input	5	6	7	16	9	17	9	6	9	6	8	16	9	5	6	12	6	18	13	15
Output	16	-	14	-	12	-	15	-	14	-	12	-	16	-	10	-	17	-	18	-
	9	11	13	20	10	25	10	8	11	9	11	17	12	9	9	14	10	20	17	22
	5	11	11	12	7	27	10	4	9	6	5	15	12	7	3	9	6	25	30	33
	7	9	12	17	11	24	11	6	9	7	12	20	13	8	6	12	7	20	18	19
	8	12	11	14	11	16	10	6	11	11	9	16	12	9	7	10	11	15	14	17
	6	22	9	12	12	36	10	6	9	8	10	31	11	8	7	10	10	16	16	17
	5	12	6	9	8	16	7	5	11	5	5	17	6	6	6	12	5	16	16	17
	4	6	9	9	8	14	8	4	4	7	11	48	12	8	6	7	6	16	13	14
	6	15	8	14	9	19	6	4	13	5	7	16	7	11	4	14	6	16	17	17
	6	12	9	11	11	15	6	7	12	8	8	14	8	7	5	12	7	16	13	14
	9	11	10	12	10	19	9	8	11	9	11	22	11	10	6	12	9	18	17	13
	7	12	9	13	10	18	7	6	11	9	9	19	11	9	5	11	10	16	10	16
	10	14	12	26	12	39	9	9	10	12	10	15	12	8	9	13	9	23	16	18
	9	9	12	12	7	18	9	8	14	8	10	16	13	9	8	14	7	16	16	20
	9	17	11	15	14	18	11	4	13	12	13	17	14	6	5	12	7	18	17	21
	6	11	12	16	12	20	9	6	14	9	7	17	9	7	6	10	9	17	15	19
	7	17	11	15	14	27	10	6	13	12	9	25	12	11	8	15	9	28	20	18
	7	8	10	15	10	16	12	6	7	6	8	15	8	8	6	11	7	21	11	19
	7	15	10	12	10	23	10	7	11	9	7	25	12	9	6	17	8	20	18	21
	6	14	12	15	14	35	10	6	12	8	8	30	10	7	6	10	10	24	16	20
	5	9	9	15	10	18	7	5	7	7	7	17	8	10	3	12	7	18	18	18
	10	15	12	13	12	26	10	7	9	10	11	30	9	9	8	15	11	16	15	18
	6	15	10	10	10	40	10	6	9	7	11	20	8	6	5	10	6	21	16	18
	6	14	13	14	16	34	8	5	10	15	10	23	11	8	6	20	12	33	27	19
	9	10	9	14	12	20	9	8	13	9	11	30	9	9	8	13	8	16	14	16
	6	15	10	13	11	21	8	7	12	8	6	23	9	5	7	13	9	17	19	25
	5	7	8	13	8	16	8	5	8	8	7	37	10	8	6	9	8	16	15	14
	5	6	10	12	7	15	7	5	8	9	8	12	11	7	7	12	8	15	12	13
	6	20	12	12	13		7	5	11	8	8	30	12	10	8	14	8	17	16	35
	8	13	8	14	12	21	12	4	13	5	8	32	11	9	8	11	7	29	33	22
	9	9	12	13	8	19	10	9	12	6	9	17	12	9	5	9	7	23	13	20
	8	14	8	10	12	17	6	5	8	8	8	15	9	9	7	15	9	15	15	16
	6	11	9	11	12	26	9	6	10	6	11	18	11	8	4	10	8	17	16	17
	7	17	11	13	10	28	9	5	8	9	11	46	10	9	5	13	8	15	19	22
	9	9	5	13	12	14	6	7	11	8	10	13	8	7	6	13	9	18	14	15
	5	10	8	13	11	24	12	4	5	7	13	23	11	11	4	14	8	21	21	21
	8	10	10	11	12	15	8	7	12	9	9	16	10	10	8	12	10	16	14	16
	6	14	12	14	8	30	10	8	12	10	11	35	10	8	7	15	8	20	20	25
	7	14	13	10	13	17	12	8	12	12	11	15	11	8	7	13	12	17	15	33
	9	12	13	15	11	22	9	8	13	9	11	27	11	9	7	12	10	25	14	22
	8	9	12	12	10	26	11	6	11	9	13	23	9	8	5	8	6	19	14	20

Table B.3 User study data for batch 2, male.

ID	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16	b17	b18	b19	b20
Mod.?	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y
Input	14	9	13	8	10	6	5	10	6	6	14	7	5	6	11	5	8	12	7	7
Output	-	18	-	17	-	12	-	18	-	12	-	16	-	9	-	11	-	17	-	16
	13	7	10	6	5	7	6	6	6	6	11	11	4	4	8	4	5	12	7	13
	15	9	13	11	8	12	9	10	10	10	14	10	7	9	7	5	8	14	12	10
	14	6	8	5	6	6	5	6	5	5	11	8	4	6	5	5	6	12	8	6
	16	7	9	8	8	7	8	8	7	8	13	7	4	6	8	5	8	15	10	9
	18	11	15	11	7	7	8	11	10	9	17	14	6	10	12	8	10	13	13	13
	16	11	11	9	8	8	9	9	7	8	15	11	6	8	10	7	9	14	11	12
	13	8	8	9	9	8	8	7	7	7	11	12	4	6	8	6	11	11	13	10
	14	11	7	8	6	11	7	12	8	9	14	11	5	10	9	9	7	14	13	12
	18	10	12	11	12	9	9	8	7	12	14	13	7	9	11	7	9	13	10	9
	15	5	12	7	6	8	8	6	6	7	14	10	6	6	5	7	7	13	13	10
	15	9	12	8	9	8	8	9	8	7	14	11	6	7	12	5	9	12	13	11
	15	10	12	11	7	9	9	11	8	10	13	10	5	9	7	9	8	11	10	10
	16	9	12	7	7	10	8	12	7	6	16	13	7	7	6	5	6	13	12	13
	17	12	13	9	8	12	9	11	9	11	13	12	9	11	13	8	13	16	17	13
	18	9	12	9	8	10	8	8	8	8	13	10	10	8	9	6	8	13	12	11
	18	8	13	11	12	8	9	12	7	10	16	13	6	11	11	7	12	16	13	14
	17	10	14	7	8	8	10	10	9	11	16	12	6	10	8	6	6	15	14	9
	16	7	10	9	8	7	8	8	6	9	14	11	6	6	7	5	8	14	10	12
	12	8	8	7	6	6	6	7	6	6	11	7	6	7	6	6	8	8	8	6
	14	6	11	8	6	7	5	8	4	8	15	8	6	8	5	5	6	14	7	8
	15	9	12	10	9	7	7	9	6	10	13	12	5	8	9	7	11	15	12	12
	16	7	12	8	8	8	6	10	6	8	16	9	5	7	8	6	6	15	7	9
	19	10	14	11	10	11	10	11	11	9	16	12	6	9	10	7	8	14	14	12
	17	12	11	12	9	7	9	10	10	10	14	11	6	8	9	8	10	14	14	11
	14	7	12	8	8	8	8	10	8	6	15	12	5	10	8	6	8	13	12	12
	16	11	13	9	10	8	7	10	10	8	14	12	6	9	11	7	11	16	14	14
	18	9	13	10	8	8	8	12	10	10	14	12	6	10	8	7	9	13	13	9
	16	9	10	8	5	7	7	9	8	9	12	10	4	7	8	7	8	15	11	8
	14	9	12	9	9	10	8	11	7	9	16	12	7	9	9	8	9	17	12	9
	18	9	12	8	10	8	7	10	7	11	14	11	5	7	6	8	9	13	14	10
	15	12	13	14	12	12	10	12	11	6	12	11	12	10	10	8	8	9	13	11
	16	7	12	6	6	7	6	6	5	7	14	8	5	6	6	5	7	10	10	7
	14	10	13	10	8	12	7	12	9	11	14	13	7	11	12	7	10	14	12	13
	14	8	8	6	4	5	5	6	5	3	13	6	4	5	4	4	4	7	8	8
	17	6	12	6	5	7	6	11	5	6	16	8	5	6	6	5	7	17	10	8
	17	11	13	10	9	10	9	13	8	10	16	12	10	13	11	6	10	14	13	12
	17	12	13	12	12	11	7	12	8	12	16	12	8	9	12	7	8	16	14	12
	17	8	8	8	7	8	6	8	9	6	14	11	6	8	8	5	7	13	12	10
	15	10	12	12	9	8	9	10	10	10	14	10	7	10	10	9	10	12	16	12
	13	11	13	8	7	6	6	6	6	8	15	12	5	8	9	5	9	13	12	12

Table B.4 User study data for batch 2, female.

ID	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16	g17	g18	g19	g20
Mod.?	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y
Input	16	6	14	12	9	7	9	6	14	6	8	5	16	5	6	12	17	12	13	11
Output	-	12	-	16	-	17	-	7	-	13	-	16	-	12	-	18	-	18	-	15
	15	9	16	11	8	5	5	5	13	5	8	7	16	6	3	10	12	9	13	12
	16	15	17	12	12	8	7	5	14	10	9	14	12	13	8	11	13	11	14	13
	14	11	16	12	7	6	6	4	14	6	5	6	9	6	4	11	12	6	15	8
	15	13	16	13	6	6	7	4	16	7	5	9	12	8	4	12	10	7	12	14
	19	15	18	15	14	10	11	8	16	8	12	12	14	10	6	10	15	9	13	14
	21	10	16	12	9	7	6	8	13	8	7	10	17	9	5	11	13	9	14	17
	16	15	15	11	10	6	7	7	14	5	9	12	12	9	6	12	13	9	13	15
	15	12	16	13	9	5	8	5	13	7	5	10	13	11	6	12	12	8	15	11
	22	15	24	13	12	8	9	8	14	11	10	16	12	10	10	11	16	9	14	11
	18	15	16	11	7	8	6	6	13	9	8	11	16	11	6	12	14	10	13	12
	17	13	16	13	10	7	9	8	16	8	7	12	15	9	5	14	15	8	14	16
	16	15	17	13	9	7	9	7	16	9	10	12	13	10	7	12	14	8	13	11
	18	18	21	12	7	5	5	6	14	9	7	12	16	11	5	17	13	6	15	23
	19	16	22	14	11	8	10	11	18	11	8	13	15	10	11	17	16	12	14	16
	18	12	17	15	9	7	7	7	15	7	10	12	13	10	7	13	16	7	14	15
	19	15	17	14	14	6	6	10	16	12	10	13	15	10	6	14	15	6	15	15
	17	12	16	11	8	8	9	8	17	9	8	13	16	7	5	14	16	11	17	13
	18	13	20	13	11	7	8	5	13	6	6	11	11	8	5	14	15	5	14	20
	13	11	15	10	8	6	6	6	12	5	8	8	10	7	5	10	10	7	8	9
	16	9	16	9	7	4	4	7	11	6	11	11	16	5	4	9	17	6	13	15
	22	13	14	12	8	7	7	8	16	6	8	11	18	5	5	16	13	9	12	17
	18	12	17	14	8	7	8	5	15	9	8	10	14	7	5	13	16	6	11	24
	18	12	17	14	8	7	8	7	16	8	9	11	17	11	7	14	13	11	15	18
	17	11	19	13	9	5	9	7	15	6	5	12	13	11	7	15	14	9	15	12
	17	16	17	13	13	10	10	6	13	11	10	13	14	10	6	15	17	8	15	16
	17	16	25	13	11	5	7	5	15	8	7	12	14	9	6	16	13	9	14	16
	19	12	26	11	9	8	7	6	15	11	9	11	14	10	6	12	14	11	14	11
	17	15	18	10	8	6	9	5	16	6	5	6	15	8	7	12	14	11	13	10
	18	20	16	13	9	8	9	7	18	12	8	13	17	10	7	10	14	8	14	20
	18	9	17	13	12	8	10	6	14	6	10	14	16	10	7	14	14	6	14	14
	14	15	16	14	9	10	11	7	12	11	11	11	14	12	10	15	14	9	13	17
	17	9	17	9	10	6	6	5	11	7	9	8	9	10	5	12	11	6	15	8
	16	12	17	13	12	6	10	7	14	12	9	12	14	9	7	13	15	9	15	13
	17	14	19	9	7	6	6	4	12	7	4	8	7	6	3	12	13	4	11	10
	17	17	16	11	6	4	5	6	17	6	5	15	15	5	3	17	14	5	17	17
	17	18	18	15	11	9	8	8	15	11	12	13	17	11	7	15	14	10	14	15
	21	12	24	12	11	9	8	9	17	12	9	13	17	12	9	16	15	10	13	15
	18	14	15	14	12	5	6	5	14	6	8	12	12	10	5	14	13	10	13	9
	17	14	16	14	12	9	10	9	13	6	8	9	18	9	6	17	15	8	15	14
	16	13	17	13	10	7	8	8	16	10	8	6	16	9	5	15	15	10	13	13

Table B.5 Realism data for batch 1, male.

ID	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16	b17	b18	b19	b20
Mod.?	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
Input	8	9	11	8	5	12	5	18	6	6	11	16	5	9	6	5	8	17	7	7
Output	14	-	13	-	10	-	7	-	10	-	14	-	15	-	11	-	18	-	15	-
	2	3	4	3	2	3	3	3	3	3	4	3	3	3	3	3	2	2	3	3
	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	3	3	3	4	4
	2	3	4	3	2	2	2	2	3	3	3	3	2	3	2	4	3	3	3	3
	3	3	2	4	4	3	4	3	4	2	4	3	2	4	4	3	3	3	3	4
	2	4	4	3	2	3	3	2	3	2	2	2	4	3	3	2	2	2	3	3
	2	3	3	3	2	3	3	3	3	3	2	2	2	3	3	3	3	2	3	3
	3	2	2	3	3	3	3	1	3	2	3	3	3	3	3	3	2	4	3	1
	3	3	3	3	2	3	3	3	3	2	3	3	2	3	3	3	3	3	2	3
	4	4	4	3	4	4	4	2	4	2	4	2	1	4	3	1	4	2	3	4
	2	3	3	3	3	2	3	3	3	2	3	2	3	3	3	2	3	2	2	3
	3	4	3	4	4	3	4	2	3	3	2	3	2	3	4	3	3	1	4	3
	4	4	3	4	4	4	4	4	4	3	4	2	3	4	2	4	4	4	4	4
	3	3	4	4	3	2	3	2	4	3	2	3	1	2	3	1	3	2	1	2
	2	2	3	3	3	3	3	3	3	3	3	3	3	3	1	2	2	3	3	2
	2	3	4	3	3	2	3	4	2	2	3	3	2	3	3	2	2	2	3	4
	3	3	3	3	2	3	3	3	3	3	3	2	2	3	3	3	3	2	3	3
	3	3	2	3	3	3	3	2	3	3	3	3	3	2	2	2	3	2	3	3
	3	2	3	2	3	3	2	2	2	3	3	3	3	3	2	2	3	3	2	3
	2	3	3	3	2	2	2	3	3	3	2	2	1	3	3	2	2	2	3	2
	2	3	4	3	2	2	3	3	2	3	2	4	3	3	3	4	2	2	4	3
	3	2	3	2	2	3	2	3	3	2	2	2	1	3	2	2	3	3	2	2
	2	3	3	2	1	2	3	4	2	1	1	4	2	3	3	2	1	1	4	3
	2	4	3	4	4	1	3	2	3	2	3	2	1	4	3	2	3	1	3	4
	3	4	4	3	3	3	4	3	4	3	2	4	4	3	3	4	3	3	4	3
	2	4	4	2	3	2	3	3	3	3	3	3	2	3	4	4	2	2	3	3
	2	3	3	3	3	4	3	4	2	2	2	4	2	3	3	2	4	4	3	4
	2	3	4	4	1	1	1	3	4	2	1	1	2	4	1	1	3	4	3	1
	1	3	2	2	4	4	3	4	4	4	3	4	1	2	2	2	4	4	3	3
	4	3	4	3	3	4	3	4	4	3	3	4	3	3	3	3	3	4	3	3
	2	3	1	4	2	1	3	3	1	3	1	4	3	3	1	1	2	1	3	2
	3	3	4	3	3	3	3	2	3	3	4	2	2	3	4	2	4	4	3	4
	3	3	3	3	3	3	3	3	3	2	3	2	2	3	3	2	3	3	2	3
	3	3	3	2	3	2	3	3	3	2	3	2	2	3	3	2	3	3	3	2
	1	4	3	2	1	1	1	1	2	1	1	1	3	1	1	2	2	2	2	2
	2	4	4	4	4	2	3	2	3	3	4	2	2	4	3	4	4	1	4	3
	3	3	4	3	3	2	3	2	3	2	4	2	2	3	2	2	3	2	2	3
	3	3	4	3	1	3	2	3	3	2	2	3	2	3	3	2	3	3	3	2
	2	4	3	2	4	2	3	4	4	4	4	4	3	2	2	2	4	1	4	4
	4	3	3	3	3	2	3	3	2	3	2	3	2	3	3	3	3	3	3	3
	4	2	2	3	3	4	2	3	4	3	4	2	2	2	3	3	1	2	3	2

Table B.6 Realism data for batch 1, female.

ID	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16	g17	g18	g19	g20
Mod.?	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
Input	5	6	7	16	9	17	9	6	9	6	8	16	9	5	6	12	6	18	13	15
Output	16	-	14	-	12	-	15	-	14	-	12	-	16	-	10	-	17	-	18	-
	3	2	3	2	3	3	4	2	3	3	3	3	3	4	3	3	2	3	3	2
	4	3	4	4	3	3	4	3	4	3	3	4	4	4	4	4	4	3	4	2
	3	2	2	2	2	2	3	2	2	2	3	3	3	3	3	2	3	3	2	2
	4	3	3	4	3	2	3	1	3	2	4	3	4	4	4	3	2	3	2	3
	3	1	3	4	2	1	2	2	4	3	2	2	2	4	4	3	2	3	2	2
	3	2	3	3	3	3	3	3	3	3	2	3	3	2	3	2	2	3	2	3
	3	2	3	4	3	3	2	1	2	2	4	3	2	2	2	3	2	2	4	3
	3	3	3	3	3	2	3	2	3	2	3	3	3	3	3	2	2	3	3	2
	4	4	1	3	4	4	3	4	3	2	2	2	4	3	3	4	4	4	4	4
	2	3	2	4	2	3	3	1	3	2	3	3	2	3	3	2	2	3	3	3
	3	1	3	2	2	2	3	1	2	3	2	2	2	3	3	2	2	1	2	2
	4	4	4	4	3	2	4	4	4	3	4	4	2	4	3	4	2	4	3	4
	2	4	3	3	4	2	3	1	2	3	3	3	3	4	3	1	3	2	2	1
	3	3	2	3	3	3	3	2	3	3	3	3	3	3	3	2	3	3	3	3
	3	2	2	3	3	2	3	2	1	2	3	3	2	3	3	3	3	3	3	2
	3	2	3	3	3	2	3	3	2	2	3	2	2	2	3	2	3	3	2	2
	2	3	3	3	2	3	3	3	2	3	3	3	2	3	2	3	3	3	3	3
	3	3	2	2	2	2	3	2	2	2	3	2	2	3	3	2	3	3	2	3
	3	3	2	2	2	3	3	3	2	3	2	3	3	4	3	3	3	3	3	2
	3	1	3	3	3	1	3	3	3	3	2	3	3	4	3	1	1	2	2	1
	3	2	2	3	3	3	3	2	3	2	4	2	3	4	2	1	3	3	3	3
	3	1	3	1	2	1	2	2	3	4	1	4	3	3	2	2	2	3	2	1
	4	2	2	3	2	2	4	2	4	1	2	3	2	2	3	3	3	2	2	2
	2	2	4	4	2	2	1	4	3	4	2	3	2	2	3	3	3	3	3	4
	3	2	3	3	3	2	4	4	2	3	3	2	3	3	3	3	2	2	2	2
	3	4	2	2	3	4	2	1	3	1	4	3	3	2	3	4	4	4	4	4
	3	1	2	4	2	1	1	2	4	4	1	3	3	2	4	1	1	1	3	1
	2	4	2	2	2	3	4	2	3	3	2	4	3	4	2	2	3	4	3	3
	3	4	3	4	2	4	3	2	3	4	3	3	3	3	3	3	3	4	4	4
	2	2	3	2	3	1	2	1	3	3	2	4	1	2	3	1	2	2	4	1
	3	4	3	3	2	1	4	3	4	4	3	2	2	4	3	4	2	2	2	3
	3	3	3	3	3	2	3	1	3	2	3	2	2	3	2	3	2	3	2	3
	3	3	2	2	2	2	2	3	2	3	3	3	2	3	3	2	3	3	2	3
	2	2	1	1	2	3	3	2	1	2	3	1	3	3	2	2	3	2	3	1
	4	2	4	4	2	1	3	4	3	2	2	1	3	4	4	4	1	2	1	2
	3	4	2	3	2	1	3	3	3	3	3	2	3	3	3	3	2	2	2	2
	3	2	3	3	3	2	3	3	2	3	2	2	3	3	3	3	3	3	3	2
	2	2	2	3	3	4	2	3	3	3	4	4	4	4	2	4	4	4	3	2
	3	2	4	3	2	3	3	3	3	2	3	2	2	2	3	2	3	3	2	2
	2	2	3	3	4	2	2	2	2	3	2	3	3	1	2	4	4	4	2	1

Table B.7 Realism data for batch 2, male.

ID	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16	b17	b18	b19	b20
Mod.?	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y
Input	14	9	13	8	10	6	5	10	6	6	14	7	5	6	11	5	8	12	7	7
Output	-	18	-	17	-	12	-	18	-	12	-	16	-	9	-	11	-	17	-	16
	2	3	3	3	1	4	3	2	3	2	3	3	4	1	1	1	3	2	3	3
	3	4	3	3	3	1	3	4	2	3	3	3	4	2	3	2	3	4	3	3
	2	3	4	4	3	3	4	2	3	3	2	3	4	3	1	3	2	2	2	3
	2	3	3	3	2	3	3	3	3	3	2	3	3	2	2	3	2	3	3	2
	2	3	3	3	2	3	3	4	4	3	2	3	4	3	1	3	2	2	3	3
	2	3	3	3	2	2	3	3	4	2	2	3	4	1	2	1	3	3	3	2
	3	3	2	3	2	1	4	2	3	1	3	1	3	1	1	1	1	3	3	2
	3	3	4	3	3	4	3	4	3	3	3	3	4	2	3	3	3	2	3	3
	3	3	2	3	2	3	3	3	4	3	2	3	3	2	3	3	3	2	4	3
	3	3	3	2	1	1	3	2	3	1	2	2	3	2	3	1	2	2	3	2
	1	4	4	4	2	3	3	3	3	3	3	3	4	2	2	3	2	2	3	3
	3	2	4	4	1	1	4	3	3	2	4	3	4	3	3	2	4	3	3	2
	2	3	3	2	2	2	3	3	4	1	3	2	4	1	2	2	3	2	3	4
	4	4	4	3	1	2	2	3	4	1	3	3	4	4	1	1	3	2	1	2
	1	1	3	4	2	1	4	2	3	1	4	3	4	2	1	2	2	2	2	2
	3	3	3	3	2	2	3	3	3	2	3	3	3	2	2	3	3	3	3	3
	3	2	2	2	2	1	3	2	3	3	3	3	3	1	2	4	2	2	3	3
	4	3	4	4	3	3	4	4	4	3	2	4	4	4	4	4	3	2	4	3
	2	3	3	4	3	4	3	3	3	4	1	4	4	3	3	4	3	4	4	4
	2	2	3	3	3	2	2	3	2	2	2	2	3	2	2	2	2	3	2	3
	3	3	2	4	1	2	3	2	2	2	2	1	3	1	3	1	2	3	3	1
	1	4	2	3	1	2	3	4	3	2	2	3	3	2	3	1	3	1	2	3
	2	2	3	3	2	2	3	2	3	2	2	3	3	3	2	2	2	2	3	2
	3	2	4	3	2	3	2	3	3	2	2	2	4	2	2	2	3	2	2	2
	4	4	4	4	4	3	3	3	4	4	2	4	4	4	3	4	4	3	4	4
	4	3	3	4	1	1	3	4	4	2	3	3	4	3	2	2	2	2	2	2
	2	3	3	3	1	3	3	3	4	4	3	3	3	4	3	3	4	3	3	2
	3	3	3	3	2	3	3	2	3	3	2	3	3	3	3	2	2	2	3	3
	3	3	3	3	3	3	3	2	3	3	2	3	3	2	3	3	2	3	3	3
	4	4	3	2	1	2	4	2	4	3	4	2	4	1	2	4	3	3	3	4
	2	3	2	3	2	1	3	3	3	2	2	2	2	2	2	3	2	3	3	2
	3	3	3	3	3	3	3	3	2	3	2	2	2	2	3	2	3	3	3	3
	4	3	4	3	3	1	4	4	3	2	4	1	4	2	1	1	4	4	4	2
	4	3	4	4	2	3	2	3	3	1	2	2	2	1	3	2	3	2	2	3
	2	2	4	4	2	4	2	2	3	3	2	3	4	3	3	2	2	1	2	2
	3	3	3	3	1	2	3	3	3	2	3	3	3	2	2	2	2	2	3	3
	2	4	4	3	2	3	3	3	3	3	2	3	3	3	3	2	2	2	4	3
	4	3	4	2	2	2	3	2	3	2	3	1	3	2	1	1	3	3	3	1
	4	4	2	4	4	4	4	4	4	4	4	4	4	4	4	2	4	4	4	3
	4	4	4	3	2	3	3	4	4	3	3	3	4	2	4	3	2	2	3	4

Table B.8 Realism data for batch 2, female.

ID	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16	g17	g18	g19	g20
Mod.?	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y
Input	16	6	14	12	9	7	9	6	14	6	8	5	16	5	6	12	17	12	13	11
Output	-	12	-	16	-	17	-	7	-	13	-	16	-	12	-	18	-	18	-	15
	3	2	2	2	2	2	4	3	2	3	3	2	1	3	3	1	3	3	1	3
	4	3	2	2	3	3	2	4	2	3	3	4	3	3	3	3	3	3	2	3
	3	2	2	2	2	3	3	4	2	4	3	2	2	2	4	2	2	3	2	3
	3	3	1	3	2	2	3	3	3	3	2	3	3	3	3	3	3	2	2	4
	4	3	2	3	3	3	3	2	3	4	3	3	3	2	3	3	3	3	3	4
	3	3	2	3	2	2	2	3	4	4	3	1	2	2	2	2	4	4	3	3
	4	3	2	3	3	3	2	1	1	2	3	3	1	3	2	4	4	4	3	4
	3	3	3	3	3	3	3	3	2	3	3	3	3	4	4	3	3	2	2	2
	3	2	2	2	3	2	4	3	3	3	3	3	3	2	4	3	3	3	2	2
	2	2	3	3	2	1	3	1	3	4	2	1	2	2	2	2	3	3	2	2
	3	2	2	2	2	2	4	3	3	3	3	3	3	3	4	2	3	4	3	2
	4	2	4	2	3	2	3	2	1	2	4	2	3	3	4	3	3	2	3	3
	2	1	4	2	3	3	3	3	2	3	3	2	3	3	4	2	3	4	3	3
	4	3	1	4	4	3	3	1	1	2	3	2	3	4	4	1	4	1	1	4
	3	3	3	2	4	2	3	2	3	3	2	2	3	4	3	2	4	4	4	3
	3	3	2	3	3	2	2	3	3	3	3	3	3	3	3	3	3	3	3	2
	3	2	2	2	2	3	1	3	3	3	2	3	3	3	3	3	3	4	3	4
	3	3	2	3	3	3	4	3	3	4	4	3	4	4	4	3	3	4	3	3
	3	1	1	2	4	2	2	4	3	2	1	3	3	4	1	2	2	1	4	2
	2	2	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
	2	1	1	3	1	2	2	4	2	3	4	2	3	4	4	3	3	1	1	1
	3	3	2	1	1	3	2	2	4	3	4	3	3	3	2	2	4	2	2	4
	3	3	2	2	2	2	2	2	3	3	3	2	3	2	2	2	3	3	2	3
	2	3	2	2	3	3	2	2	2	3	2	2	3	2	2	2	2	3	2	2
	4	4	2	3	3	2	4	4	4	4	4	4	3	3	4	4	3	4	4	4
	4	3	1	4	3	3	4	2	3	3	4	2	3	3	3	2	4	2	3	4
	4	1	3	3	2	2	3	2	2	3	2	1	2	4	3	2	3	2	2	2
	2	3	2	3	3	3	2	2	2	2	3	3	3	3	2	3	2	3	3	3
	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3
	4	1	3	2	3	3	3	3	4	4	2	4	4	3	3	1	3	4	3	1
	3	3	3	3	3	3	3	2	3	3	3	3	2	3	2	3	2	2	3	3
	3	3	3	3	3	3	2	3	3	3	3	3	3	2	3	3	3	3	3	3
	3	2	1	1	4	1	3	3	2	3	4	2	3	4	3	4	4	2	4	4
	2	3	2	3	4	3	2	3	2	2	2	2	2	2	2	3	2	4	2	3
	3	3	2	2	3	3	3	3	2	4	3	3	3	3	3	2	3	3	2	2
	2	2	1	2	3	3	3	2	2	3	2	1	3	3	3	2	3	3	3	2
	3	1	2	2	4	2	4	4	4	3	2	2	2	4	4	2	3	2	2	2
	3	2	2	1	2	1	3	2	3	2	3	3	2	3	3	3	3	3	3	4
	4	4	2	1	4	2	1	4	4	4	2	4	4	4	4	4	4	4	4	4
	3	2	2	4	4	2	4	2	2	2	3	2	3	4	3	3	4	4	4	4

Appendix C

Landmark Distance Comparisons

We now present comparisons of the distances between facial landmarks as measured in a 2D image and in a 3D image. Our growth parameter optimization algorithm works in 2D image space, but our reference data was collected from physical measurements in 3D space. To quantify how much error is introduced by using 3D distances as if they were 2D distances, we collected data for the same subject in 3D and in 2D for direct comparison.

We used the 3dMDface System (TM), a set of cameras that captures a three-dimensional face image, to collect data to better understand the landmark distances used in our growth parameter optimization algorithm. This system is a setup of several conventional cameras and several distance-finding infrared cameras that simultaneously capture several images of a subject, which are then automatically stitched together to create a 3D model of the subject. This system is primarily targeted toward medical uses, such as for comparing a patient's face before and after surgery. As a result, the 3D images have high geometrical accuracy.

The software for this system, 3dMD Vultus (TM), offers templates for landmark analysis for the 3D images. Of the templates offered, the best template to use is Facial Proportions (and we can modify this template to take out points we don't need or add points that we do need). There is a feature that automatically detects landmarks in the pre-determined Reserved Facial Landmarks list. 28 of the Stasm points used in our script are part of this list, and we can manually add any missing landmark points. In the 3Dmd Vultus program, we can get measurements from every recognized point to every other recognized point, so we were able to find all 3D distances corresponding to the 2D distances currently used in our algorithm. Note that the data points in the 3D image are in 3D space and hence have XYZ coordinates, whereas the distance measurements are linear (and measured in millimeters).

To show how the 3D distances compare to the corresponding 2D distances, the table below compares three sets of distances for one particular subject: the average distances from our Farkas reference data, the corresponding distances in a 2D screenshot of a 3D face image, and the distances in the 3D image itself (measured in 3D space). The subject whose face we used is Caucasian, male, and age 21. The Farkas average data is for a male in the age range 19-25.

The first column of Table C.1 is the distance number, which is simply an index we use to refer to each pair of facial landmarks. The second column is the pair of Stasm points used in the row's distance (specifically, the indices of these Stasm points, from 0 to 76 inclusive). The third column is the average distance between those landmarks as reported in the Farkas reference data for a male aged 19-25. The fourth column is the distance between these two landmarks as measured from a 2D image of the subject. The fifth column is the distance between these two landmarks as measured from a 3D image of the subject.

As Table C.1 shows, the distances taken from the 2D image tend to be similar to the distances taken from the 3D image. This indicates that our use of the Farkas reference data (which was collected in 3D space) in 2D image space is not a large source of error in the growth parameter optimization step of our algorithm.

Table C.1 Landmark distance comparisons.

Dist. num.	Stasm points	Farkas avg. (mm)	2D dist. (mm)	3D dist. (mm)
1	(18, 25)	115.9	118	102.16
2	(1, 11)	146.8	148	139.77
3	(4, 8)	105.6	106	119.17
4	(6, 14)	187.2	170	166.42
5	(6, 56)	72.6	72	72.17
6	(6, 67)	50.7	46	50.60
7	(6, 74)	33.1	34	40.19
8	(1, 6)	148.5	128	117.51
9	(6, 11)	148.2	126	120.10
10	(30, 40)	33.3	38	39.00
11	(34, 44)	91.2	94	87.91
12	(30, 34)	31.3	36	23.87
13	(40, 44)	31.3	36	24.17
14	(4, 34)	103.9	100	97.65
15	(8, 44)	103.6	98	96.12
16	(32, 36)	10.8	10	11.77
17	(42, 46)	10.8	10	11.58
18	(54, 58)	34.9	38	30.71
19	(55, 57)	32.8	34	31.27
20	(51, 53)	6.9	16	21.41
21	(56, 58)	23.2	26	30.71
22	(54, 56)	23.2	26	31.27
23	(61, 63)	10.4	14	15.27
24	(59, 65)	54.5	56	51.54
25	(56, 67)	22.3	24	22.90
26	(56, 62)	15.9	16	17.39
27	(62, 67)	8.0	5	7.68
28	(70, 74)	9.3	10	8.34
29	(57, 60)	17.0	20	21.91
30	(55, 64)	17.0	21	22.09
31	(1, 59)	114.5	72	81.09
32	(11, 65)	113.6	72	80.21

Appendix D

Code Documentation

This section contains more technical documentation about the code delivered in this project. While the code itself (located on the CD Rom and fully documented with comments) is the most directly informative, this documentation provides a slightly higher-level overview of what one may expect to find in the different files.

D.1 Project Prototype

In the first semester, we created an aging prototype to implement the fluid-filled sphere model of craniofacial growth from Photoshop (see Section 3.3). This prototype is comprised of a JavaScript file and two separate C++ projects that each compile to a binary. For a visual overview of the prototype's architecture, see Figure 3.1.

The final project does not directly use any of these files. The two C++ projects used in the prototype, `StasmUtil` and `WarpUtil`, have been subsumed by the final, unified executable, `AgeProgression`. The prototype JavaScript has been adapted to call the new executable in the final project.

D.1.1 Photoshop Script

The prototype JavaScript file, `aging-prototype.jsx`, implements the fluid-filled sphere model. After prompting the user for the input and desired output ages, the JavaScript calls the Stasm Utility to get the landmarks of the subject, calls an internal function to calculate the output landmarks, and calls the Warp Utility to warp the input image to match the output landmarks. The script then opens the output image and draws lines connecting the output landmarks.

D.1.2 Stasm Utility

The Stasm Utility is a binary executable we created that is called via the command line from JavaScript. This project, `StasmUtil`, is built from C++ in Visual Studio, using the Stasm library and OpenCV library (Milborrow, 2009). As command-line arguments, it takes the input image path, the output file path, and a path to Stasm data files. The executable writes, to a text file at the given output file path, a list of 77 integer coordinates, one for each Stasm landmark.

The C++ project uses Stasm version 4.1.0 and OpenCV version 2.4.11.

D.1.3 Fluid-Filled Sphere Model

The prototype JavaScript takes the 77 input landmarks and the number of years by which to age the subject. It then uses the fluid-filled sphere model of craniofacial growth to compute the new locations of each landmark (Ramanathan and Chellappa, 2006). This model increases the distance from each landmark to a point in the middle of the forehead as a function of the angle between the landmark and the middle point and of the desired increase in age (age delta). The `modifyLandmark(point, origin, kvalue)` function computes this new landmark and is called for each landmark.

D.1.4 Warp Utility

The Warp Utility, much like the Stasm Utility, is a binary executable we created that is called via the command line from JavaScript. The project, `WarpUtil`, is built from C++ in Visual Studio, using an existing warping library and the OpenCV library. The warping library can be found at https://github.com/cxcxcxcx/imgwarp-opencv, implemented from the paper "Image Deformation Using Moving Least Squares" (Schaefer et al., 2006). As command-line arguments, the executable takes an input image path, an output image path, a list of source points, and a list of destination points. The executable saves the warped output image to the given output image path.

D.2 Photoshop Script

While all of the key components of our final project are implemented in the C++ project, we also have provided a JavaScript file that can be called as a script from Photoshop to interface between the user and the executable built from the C++ project. This script, `age-progression.jsx`, is called from Photoshop in a similar manner as the prototype JavaScript described in the previous section. This script more or less acts as an interface to the `AgeProgression.exe` executable, as it does not hold much implementation of the actual aging model. This script prompts the user for the name of the file to open, the input age, target age, and sex. It also passes race, a temporary output image path, a Stasm data path, and a Farkas data path to the executable. After the output image is created and loaded into the Photoshop document, the script deletes the output image to clean up after itself. This version of the Photoshop script does not draw the Stasm landmarks, although a flag can be passed into the executable to draw the input and output landmarks onto the output image itself.

D.3 C++ Project

The central part of our final project is the C++ Project, `AgeProgression`. We now describe the steps required to clone and build the project and provide an overview of the most important files and functions in the project.

An example of a command used to run the executable is as follows:

```
AgeProgression.exe
--original-age 6 --target-age 19 --sex female --race caucasian
--input-image-path "example-files\girl.JPG"
--output-image-path "example-files\girl-out.JPG"
--stasm-data-path "stasm-data"
--original-landmarks-path "example-files\face-original-landmarks.txt"
--target-landmarks-path "example-files\face-output-landmarks.txt"
```

The full list of flags is as follows:

```
--original-age, --target-age, --sex, --race, --input-image-path,
--output-image-path, --stasm-data-path, [--farkas-data-path],
[--original-landmarks-path], [--target-landmarks-path], [--scale-k],
```

`[--scale-face], [--draw-landmarks]`

The flags in brackets are optional.

D.3.1 How to Build

Here we describe how the `AgeProgression` project is set up in Visual Studio and how to build it. This project has only been developed on Windows 10 computers but should work in other Windows environments.

Dependencies

We now explain the requirements to build the project. The steps to get the project set up on a new computer are as follows:

1. Install Visual Studio 2015 Community. Copy the code repository and open the `AgeProgression.sln` project file with Visual Studio; download the update that Visual Studio prompts you to download.

2. Download `http://www.milbo.users.sonic.net/stasm/download.html` Stasm 4.1.0 and extract it to `C:` (so you have `C:\stasm4.1.0`).

3. Download `http://opencv.org/downloads.html` OpenCV 2.4.11 and extract it to `C:` (so you have `C:\opencv`).

4. Restart the computer, then open the project in Visual Studio. Set the build settings to Release and x86, and the project should compile. The resulting executable should run (as long as there is a `stasm-data` folder and the folder with the executable has the four DLLs (`opencv_core2411.dll`,`opencv_highgui2411.dll`,`opencv_imgproc2411.dll`, and `opencv_objdetect2411.dll`) and `FarkasData.json`).

Make sure to build the release version of the project or else the DLLs might not work (since they are for release builds, not debug builds). The project settings include modifications to the VC++ Directories and the Linker Input Additional Dependencies for Stasm and OpenCV.

Libraries

Our project uses four pre-existing libraries: Stasm 4.1.0 for landmark detection, ALGLIB for performing a least-squares regression in our growth parameter optimization process, a library called imgwarp-opencv for Moving Least Squares warping, and OpenCV for image loading and manipulation.

Additional information about each of these libraries can be found at the following locations:

- Stasm: http://www.milbo.users.sonic.net/stasm/
- ALGLIB: http://www.alglib.net/
- imgwarp-opencv: https://github.com/cxcxcxcx/imgwarp-opencv
- OpenCV: http://opencv.org/

D.3.2 Key Components

We now explain the four most important files in our C++ project and their most important functions.

Age Progression

The file `age_progression.cpp` has the `main` function of our project. This file takes in information about a subject and outputs an age-progressed face image. The `main` function performs the following steps:

1. Parses all the command-line arguments.
2. Calls `getLandmarkCoords` in `stasm_util.cpp` to get the original Stasm landmarks.
3. If given a path to write the original landmarks, does so by calling `writeLandmarkCoords` from `stasm_util.cpp`.
4. Calls `ageProgressLandmarks` in `param_util.cpp` to get the output landmarks, which might be a subset of the given 77 landmarks.
5. Calls `scalePoints` in `rescale_util.cpp` to rescale the output landmarks to match the width of the input landmarks.
6. Calls `getBoundaryPoints` in `rescale_util.cpp` to add boundary points to the input and output landmarks.
7. If given a path to write the target landmarks, does so by calling `writeLandmarkCoords` from `stasm_util.cpp`.

8. Calls `warp` in `warp_util.cpp` to warp the original image based on the input landmarks and output landmarks.

9. Saves the output image.

Stasm Util

The `stasm_util` file is responsible for getting Stasm landmarks and writing them to a text file.

The file `stasm_util.h` has two public functions: `getLandmarkCoords` and `writeLandmarkCoords`. The function `getLandmarkCoords` calls the Stasm library's `stasm_search_single` function to get the detected landmarks. The function `writeLandmarkCoords` is a simple helper function for outputting a vector of landmarks to a text file.

Param Util

The `param_util` file implements the core part of our algorithm: it uses growth parameter optimization to calculate age-progressed landmark positions.

The file `param_util.h` has one main public function: `ageProgressLandmarks`. This function first checks the given originalAge and targetAge to make sure they are both within the range we have data for, which is 6 years of age to 19 years of age. It then calls `processFarkasData` in `param_util.cpp` to parse the `FarkasData.json` file containing all the craniofacial landmark norms from Farkas et al. After reducing the given list of all 77 landmarks to only those useful for our warp, i.e. only those that are used in a landmark distance we have data for, the function calculates the optimal growth parameter for each landmark.

The growth parameter for a landmark, also referred to as a "k-value", is a number indicating how much the distance between the landmark and a point in the center of the subject's forehead should increase. The function uses the ALGLIB library to minimize the least squares error produced by a list of k-values and `getLandmarkDistErrors`. The `getLandmarkDistErrors` function embodies our algorithm's assumptions about what sets of landmark positions are ideal for a subject using several helper functions. The overview of this growth parameter optimization model is given in section 4.1.2, but the summarized explanation is that the growth parameters should change such that the ratio of each landmark distance in the input image to the corresponding reference landmark distance for the input age is the same as the ratio of the landmark distance in the output image to the corresponding reference landmark distance for the output age.

This optimization step usually takes up the majority of the running time of our algorithm, but the amount of time it takes varies considerably from subject to subject due to the stochastic nature of the minimization step. Once this optimization is complete, the function calculates the output position of each landmark based on its output k-value by calling `applyGrowthParameters`. The function then returns this list of age-progressed landmarks.

Warp Util

The `warp_util` file is responsible for performing a warp from a set of input landmarks to a set of target landmarks.

The file `warp_util.h` has one public function: `warp`. This function calls the `setAllAndGenerate` function from the `imgwarp_mls.cpp` file of the warping library we found. This function takes an image, a vector of input landmarks, and a vector of target landmarks, and it returns a new image that is created by warping the given image from the input landmarks to the output landmarks.

Appendix E

Glossary

3dMDface System (TM): A camera that can capture high-resolution 3-dimensional images.

Age Progression: The process of making the subject of an image look older.

Age Regression: The process of making the subject of an image look younger.

Amazon Mechanical Turk: A platform for designing and running tasks for human workers to complete.

Anthropometric Data: Quantitative measurements of the human body, particularly distances between certain landmarks on an individual's face.

Craniofacial Growth: The growth of bones in the face, leading to structural changes and different shapes and proportions without significantly affecting textures.

Facial Landmarks: Key points on the face that can be used to identify different regions of the face.

FG-NET: A database containing images of male and female faces at various ages. It is longitudinal, i.e., it has multiple images of each subject at different ages.

Ground Truth: An actual picture of a subject at the desired output age, which can be compared to the result of a modified image aged to the same age.

Growth Parameter: A value that indicates how far a point should be displaced, relative to its distance from the origin. A parameter of 0 indicates no movement.

Growth Parameter Optimization: The process by which our algorithm determines what growth parameter values will best make an output image match empirical data for the appropriate age.

Levenberg-Marquadt: A least-squares algorithm for minimizing some objective function based on a set of input parameters.

OpenCV: An open-source library for computer vision.

Stasm: A utility that, given an image of a face, can locate 77 key landmarks.

Warping: The process of taking an image, moving key landmark points to new locations, and interpolating other points to create a modified image.

Bibliography

Farkas, Leslie G, and Ian R Munro. 1987. *Anthropometric facial proportions in medicine*. Charles C. Thomas Publisher.

Kemelmacher-Shlizerman, Ira, Supasorn Suwajanakorn, and Steven M Seitz. 2014. Illumination-aware age progression. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 3334–3341. IEEE.

Lanitis, Andreas, Chris J Taylor, and Timothy F Cootes. 2002. Toward automatic simulation of aging effects on face images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24(4):442–455.

Milborrow, Stephen. 2009. Active shape models with stasm. *Stasm Version* 3.

Milborrow, Stephen, and Fred Nicolls. 2014. Active shape models with sift descriptors and mars. In *VISAPP (2)*, 380–387.

National Center for Missing and Exploited Children. 2016. Key facts. http://www.missingkids.org/KeyFacts. Accessed April 29, 2016.

Panis, Gabriel, Andreas Lanitis, Nicholas Tsapatsoulis, and Timothy F Cootes. 2015. An overview of research on facial aging using the fg-net aging database. *IET Biometrics* .

Ramanathan, Narayanan, and Rama Chellappa. 2006. Modeling age progression in young faces. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, 387–394. IEEE.

Schaefer, Scott, Travis McPhail, and Joe Warren. 2006. Image deformation using moving least squares. In *ACM transactions on graphics (TOG)*, vol. 25, 533–540. ACM.

Suo, Jinli, Feng Min, Songchun Zhu, Shiguang Shan, and Xilin Chen. 2007. A multi-resolution dynamic model for face aging simulation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 1–8. IEEE.