AUTOMATIC FACE REPLACEMENT IN PHOTOGRAPHS

การเปลี่ยนหน้าอัตโนมัติในรูปถ่าย

BY

MR. JIRAYUS LAEMTHONG 4988089
MR. WORACHET SAE-LIM 4988275
MR. DAMRI SAMRETWIT 4988277

ADVISOR
DR. RAWESAK TANAWONGSUWAN

A Senior Project Submitted in Partial Fulfillment of
the Requirements for

THE DEGREE OF BACHELOR OF SCIENCE
(INFORMATION AND COMMUNICATION TECHNOLOGY)

Faculty of Information and Communication Technology
Mahidol University

2010

ACKNOWLEDGEMENT

We would like to thank our friends who volunteered to participate in our testing and evaluation, and gave us advice to improve our system.

Mr. Jirayus Laemthong
Mr. Worachet Sae-lim
Mr. Damri Samretwit

AUTOMATIC FACE REPLACEMENT IN PHOTOGRAPHS

JIRAYUS LAEMTHONG 4988089 SCIT/B
WORACHET SAE-LIM 4988275 SCIT/B
DAMRI SAMRETWIT 4988277 SCIT/B

B.Sc. (INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR : DR. RAWESAK TANAWONGSUWAN

ABSTRACT

The face is one important human identity. People want to be good-looking in their photographs. With digital technology, people can take their personal photos in digital format. Digital photos can be used for image processing later on. There are many applications which deal with human faces such as face recognition, face modeling, facial animation, face editing, etc. In this project, we focus on the facial editing aspect. We developed a system called "Automatic Face Replacement in Photographs". This system is able to replace a face in an image with another face to get a new realistic image. The steps of face replacement are face detection, face contour creation, alignment, color adjustment, and blending. We implemented this system with OpenCV library and dnAnalytics linear solver. We also performed a user study to consider the quality of the results. This system helps people reduce time spent on editing faces and to easily get realistic face images.

KEY WORDS : FACE REPLACEMENT / SEAMLESS CLONING /
IMAGE-BASED RENDERING / COMPUTATIONAL PHOTOGRAPHY

45 P.

การเปลี่ยนหน้าอัตโนมัติในรูปถ่าย

จิรายุส แหลมทอง 4988089 SCIT/B
วรเชษฐ์ แซ่ลิ่ม 4988275 SCIT/B
ดำริห์ สำเร็จวิทย์ 4988277 SCIT/B

วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษาโครงการ : ดร. รวีศักดิ์ ธนวงศ์สุวรรณ

บทคัดย่อ

ใบหน้าเป็นส่วนสำคัญที่ใช้ในการระบุรูปพรรณของมนุษย์ มนุษย์เราส่วนใหญ่ต้องการดูดีในรูปถ่ายของตัวเอง ด้วยเทคโนโลยีดิจิตอล มนุษย์เราสามารถถ่ายภาพในรูปแบบดิจิตอล ซึ่งรูปดิจิตอลนั้น สามารถนำมาใช้ในการประมวลผลภาพต่อไปได้ โปรแกรมที่เกี่ยวข้องกับใบหน้ามนุษย์มีหลากหลายรูปแบบ อาทิเช่น การจดจำใบหน้า การสร้างโมเดลใบหน้า การทำภาพเคลื่อนไหวของใบหน้า การตัดต่อใบหน้า และอื่น ๆ ในโครงงานนี้ เรามีความสนใจในการตัดต่อใบหน้า เราได้พัฒนาระบบที่เรียกว่า "การเปลี่ยนหน้าอัตโนมัติในรูปถ่าย" ระบบนี้สามารถเปลี่ยนใบหน้าจากรูปหนึ่งด้วยอีกใบหน้าหนึ่งและได้รูปผลลัพธ์ที่เหมือนจริง ขั้นตอนของการเปลี่ยนหน้า ได้แก่ การตรวจจับใบหน้า การสร้างเค้าโครงใบหน้า การจัดตำแหน่งของใบหน้า การปรับค่าสี และการทำภาพให้กลมกลืน เราใช้ระบบ OpenCV และตัวแก้ไขสมการเชิงเส้น dnAnalytics เพื่อทำระบบให้สมบูรณ์ เราได้ทำการศึกษาผู้ใช้เพื่อประเมินผลลัพธ์ของระบบ ระบบนี้สามารถช่วยให้ผู้ใช้งานลดเวลาในการตัดต่อใบหน้าในรูปภาพและได้ผลลัพธ์ที่เหมือนจริงโดยง่ายดาย

45 หน้า

CONTENTS

LIST OF FIGURES

CHAPTER1
INTRODUCTION

This chapter describes the motivation, problem statement, scope of this project and objective of the Face Replacement System. It also describes the organization of this document.

1.1 Motivation

"One Picture is Worth Ten Thousand Words." This phrase can also mean that people love photography. When people go outside, travel in many places, they never forget the camera to take their photos. Nowadays, digital photography has become popular because of the cheaper cost and the convenience in taking and getting photos; photo film cameras are now obsolete.

People do not only collect their own photos, but they now love to share them with others, often through social networks, and this is called photo sharing. In order to share (or post) the photos on any social network, such as Windows Live, Facebook, MySpace, hi5, etc, users have to upload their photos on the Internet. Any internet user is able to search those photos by using a search engine in order to find the photos of someone they want to see.

However, sometimes photos are taken and accidentally include someone that the photographer did not want in that photo. The person who was accidentally shot in the photo may not even know. So, one important thing we should consider when photo sharing is privacy protection. One way to protect the privacy of the persons who were accidentally taken in the photo is to conceal their identity, especially their faces.

EveryScape.com is a web service which provides three-dimensional, photo-realistic experiences of cities & towns, streets & sidewalks, and building exteriors & interiors. As

shown in figure 1.1, this service provides the privacy protection which conceals the facial identity of people who are accidentally shot.

Figure 1.1: EveryScape (http://www.everyscape.com)

People also love to have their photos to be satisfied, which means they should be good-looking in the photos they have taken. In group photos, the common situation occurs where people are not ready yet; eye closed, no smile, etc. So, these photos will be unsatisfactory. Not only these situations, but in others as well, people want to have fun with editing their photos to get them to look like a superstar, famous people, etc.

Figure 1.2: PhotoFunia (http://www.photofunia.com)

PhotoFunia.com is an online photo editing tool which provides the automatic creation of users' photos with many effects and montages.

People mostly consider how to look good such as to have a perfect body shape, to have nice skin especially, skin color, hair dressing, and even the dress; someone may try to wear the same clothes as famous people.

Figure 1.3: Before and After Manually Retouched Photo
(http://www.dietweightloss.co.uk)

Often, there is one human part which people mostly consider, the face. The face can describe the expression, emotion, appearance, and beauty of humans. However, sometimes people are not satisfied with their photos; "It looks terrible", "Why so serious?", "I don't like this smile", etc. This is especially true in the event or the place that they will never be in again. They may not have a chance to take the photos like that again.

So, what people need is an image editing tool to edit their images to be more satisfied by improving unsatisfactory human appearance in their photos. There are many techniques using software tools that are now available to do image editing, but most people have no skill in image editing. Also, the cost of these tools are really expensive.

This project aims to help the users to create their satisfactory images by face replacement technique; to replace the imperfect face by a better face from another photo. It

will be easy-to-use, and the result will come out as the final photo which users are really waiting for.

1.2 Problem Statement

As we had described in the previous section, privacy issues, difficulty to do try-on (superimpose someone's face on that of popular person), and unsatisfied appearance in the digital photographs are the problems that could be solved by digital image manipulation methods.

Most people want to edit their photos on their own but they have no skill in image editing. They will have to take much time to learn how to use these tools. Although they can do it well, they will take much time on doing it. In some image editing software such as Adobe Photoshop, they have to do it manually on each photo which will take much time or even some may give up on doing it because they lack art skill.

So, we need a system that will be easy to use and able to save time compared to the manually image editing software. And, a system that not only enables people to edit their photos in shorter time, but the result should also look real. The system should be able to combine one face to another head at the appropriate position & pose, and the color & lighting should be consistent.

1.3 Objectives of the Project

The purposes of this project are:

- To create an automatic face replacing tool. The tool is targeted for personal use, so it can be used by anyone, even the users with less image editing skill. The result should be created in reasonable time.
- To create a face replacing tool which can produce a novel image that may never be taken again in the real events. It should look somewhat real.

1.4 Scope of the Project

This project aims to create an automatic tool to do image re-compositing on a coupling of a face and a head to get a new image. The system will need two input images. The first one is the source image of the face and the other one is the image of the target head that users want to place the face on.

The approach of the system is to do alignment in 2D space. We are not going to create a 3D face model because it is hard to create one from only a single image and it may require too much user intervention. For some cases, some effort from users may be needed to help the system to do alignment.

We assume the pose of the face to be a frontal pose. The system will not support an out-of-plane pose because even the commercial face detector cannot correctly estimate the pose of the face from a single image. The system relies on the users to select the input images which have a similar pose, direction of light and shadow, color, and image resolution. For the extreme difference, the system might not be able to create the realistic result.

The result of the compositing of two images should be a picture that is blended seamlessly. The color will be changed from the source face image to the target head image.

This project is inspired by the research Face Swapping: Automatically Replacing Faces in Photographs by Dmitri Bitouk et al. In comparing Bitouk's Face Swapping and our system, it should be noted that, first, they swap any face in large image database to be out of context so, their system can automatically select the face arbitrarily, but our system takes images from users. Users can select the images from their personal photo collection. Second, they use a commercial face detector which has high quality face detection, but our system uses an open source face detector. Third, their system is designed to select only similar faces, whereas our system can take any face the users want. Fourth, their system uses simple feathering method to blend images, but our system needs to use better methods because our system cannot guarantee that the images are similar.

1.5 Expected Benefits

The system would enable the possibility to create new images that may never be taken again in the real events by any users. The system should be easy-to-use software which users do not need to have any image editing skills.

Furthermore, this system can be used as a privacy protector which would be useful when users would like to protect the privacy of people who accidentally appear in the photos by replacing the face identity before the photo is published.

1.6 Organization of the Document

This document consists of 6 chapters including:

1. Introduction - This chapter describes the motivation, problem statement, scope of this project and objectives of the Face Replacement System. It addresses some key questions. Why would we like to develop this system? What are the benefits of the system? It also describes the organization of this document.

2. Background - This chapter reviews the challenges which we have faced in order to develop the Face Replacement System. There are several methods with short descriptions which reference several related works.

3. Methodology - This chapter describes the chosen techniques in more detail, including mathematics formulae, and methods which are used in the system.

4. Implementation - This chapter shows the steps of work for the Face Replacement System and how we have developed the project.

5. Testing and Evaluation - This chapter shows experimental results of the testing and evaluation of using the Face Replacement System.

6. Conclusion - The last section of the document talks about benefits, problems and limitations and also future works of the Face Replacement System.

CHAPTER2
BACKGROUND

This project is inspired by the research "Face Swapping: Automatically Replacing Faces in Photographs" by Dmitri Bitouk et al[1]. In order to develop the Face Replacement System, there are several steps needed to be done.

The challenge of each step is to use practical methods to achieve the desired result. There are several approaches, developed by other researchers, which we select to use in the system. The following are the challenges we will discuss.

2.1 Getting Face Contour

In this project, we need to define a boundary of both the source face region and the target face region. In order to clone a face patch to fill in another head region, the two regions should be identical in shape. There are many approaches to define these regions.

One approach is to draw the face boundary manually by the user. This requires a lot of user intervention. It is a standard method that is provided in many image editing tools. It is also hard to set some constraints like the number of vertices.

Another technique is to select a region by color. This approach needs less user intervention than the previous approach. One example is being used by Adobe Photoshop as a tool named "Magic Wand Tool". The tool will start the region from the point where the user specifies, then expand to nearby pixels until the color difference from the starting point exceeds the difference threshold. It is hard to get the same shape in both the source and the target regions. Interactive Digital Photomontage [7] uses a similar approach with graph-cut optimization.

An automatic approach might be implemented using face detection technology if the input images are high resolution and the face detector can extract feature points. The system may simply draw lines connecting those feature points automatically.

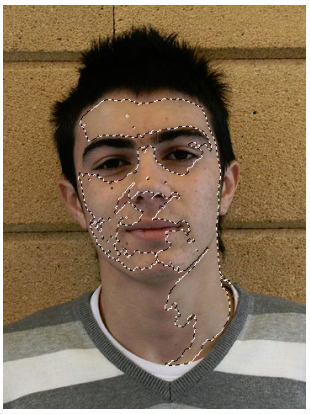


Figure 2.1: Magic Wand Tool in Adobe Photoshop

Another approach is to create a predefined shape for the face boundary. According to the fact that the human face has only little variation, i.e. two eyes, one nose, and one mouth are located at almost the same position on anyone's face, we may extract common characteristics from many faces to define a standard face boundary. We do not need a perfect boundary which separates a face from the neck and hair. We just need a boundary to achieve smooth blending, so we can create any boundary that roughly covers eyes, nose, and mouth, but excludes neck, ears, and hair. This approach can be applicable for a low resolution image in which only the eyes can be detected by the face detector. We may need human interaction only once to draw boundaries of training faces, and then find an average shape in relation to distance and direction of eyes, then we get converged initiation boundary data. Although Face Swapping [1] does not clearly discuss the approach to initialize the face contour, their figure, as shown in figure 2.2, shows the same contour shape for every face. So, Bitouk's approach is probably similar to the approach above.

When the contour is applied to the input faces, we may further modify contour to avoid crossing any prominent details.

2.2 Alignment

In order to do face alignment, we need to define the face position of both the source face and the target face to replace the face on the corresponding position. In the simplest

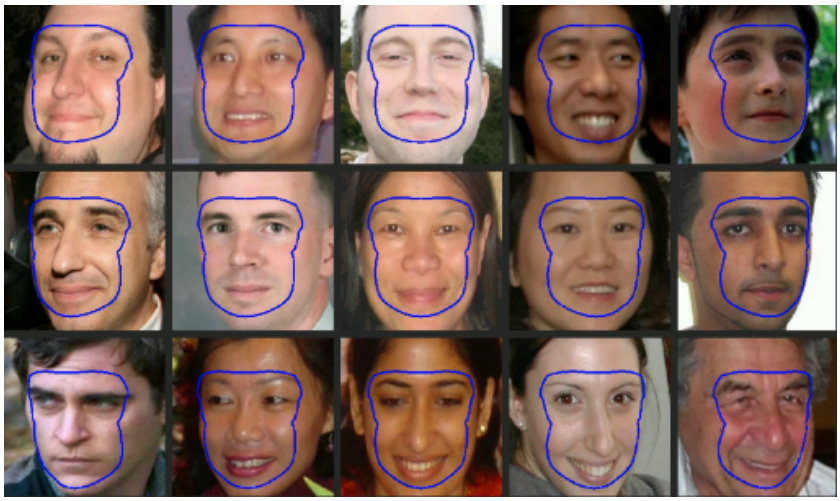


Figure 2.2: Face Contours

way, users can manually indicate the position of eyes, nose, and mouth.

In recent past decades, artificial intelligence has been developed for many applications. One of them is face detection. "OMRON OKAO" and "OpenCV tool" are examples of face detectors that can extract feature points of a face.

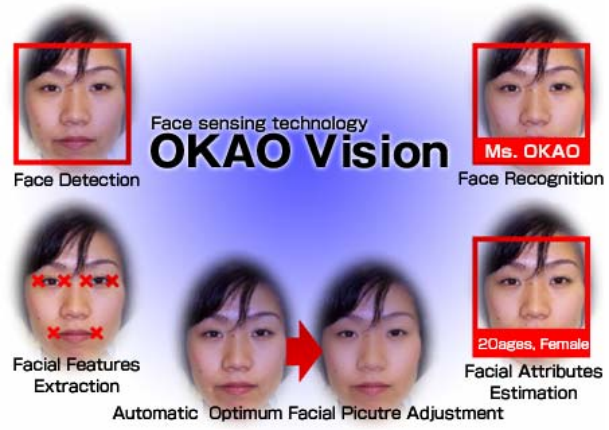


Figure 2.3: Omron OKAO

After we get the feature points, we can calculate the distance between eyes, displacement between source and target face, etc.

Furthermore, some face detectors can detect the direction of pose, in pitch and yaw angles. However, it is still not accurate when detecting the direction of pose from a single image.

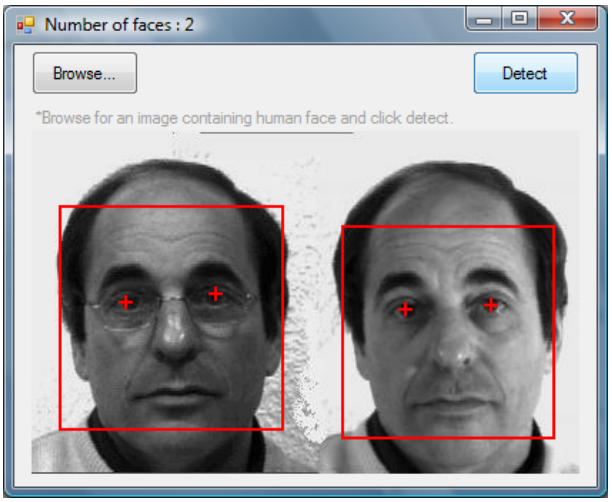


Figure 2.4: Face Detection Implementation by Zeeshan Ejaz Bhatti using OpenCV

For the situation when the face detector is not able to detect any direction of pose, another method is to infer the pose from the position of eyes. So, the face pose can be rotated to be matched with the head pose which has been inferred. This method will also be able to infer the pose from the position of the mouth and then further adjust the face pose by skewing.

As the face patch does not lay on the appropriate position, thus, the alignment process needs to be done using the information we have discussed. There are standard methods in image processing to do transforming such as rotate, scale, skew, and translate an image in two dimensions.

2.3 Relighting

In order to get a more realistic result, we need two faces with similar color that can be blended smoothly. We need to do the relighting process to make the source input face have similar color and lighting as the target.

The first approach is to do a color transfer between the source input and target input images [2]. They change the average of color and rescale the color distribution of the source image in order to have the color characteristic match the target image. However, the lighting characteristic such as level of shine cannot be transferred.

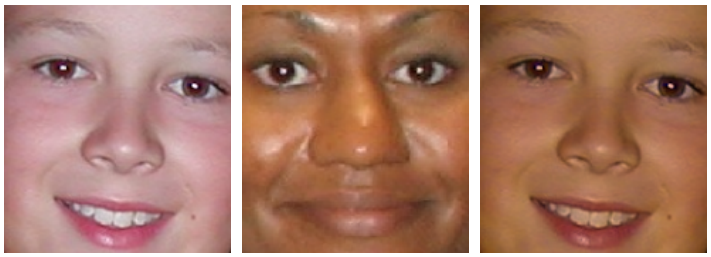


Figure 2.5: Source Color, Target Color, and the Output of the Color Transfer Technique

In Face Swapping [1], they use nine spherical harmonics proposed by Ramamoorthi and Hanrahan which can create a realistic lighting result.

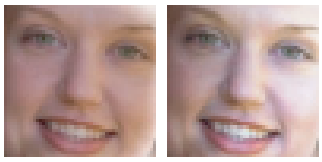


Figure 2.6: The Original Lighting Condition and the Relighted Output of the Spherical Harmonics Technique

2.4 Blending

Blending is the process of mixing two images together and then making the boundary seamless. This process could finally help the result to be more realistic.

In Face Swapping [1], they apply feathering the face along the boundary between the images that are already similar; the well-selected images which have a similar boundary selected from a large image database.

For Poisson Image Editing [6], they use a gradient-domain technique, solving a large linear system. This method can blend any two images together and mostly get the satisfied results.

For Mean-Value Coordinates [3], this method does not work on the gradient-domain. They improve the Poisson method efficiency by avoiding solving a large system, and also get a result that is similar to the Poisson method.

CHAPTER3
METHODOLOGY

This chapter describes the techniques we have selected for designing and implementing the Face Replacement System in more detail, and also the techniques we have discovered on our own. This chapter contains the system functions including explanation diagrams.

3.1 Face Replacement System Functions

The Face Replacement System can be described as these following five main functions.

3.1.1 Face Detection using OpenCV

The first function of the system is face detection. Firstly, the region of each face is detected by the face detector. The face detector detects the existence and the location of the faces in each image. Then, we continue to detect eyes and mouth from this face to get their positions.

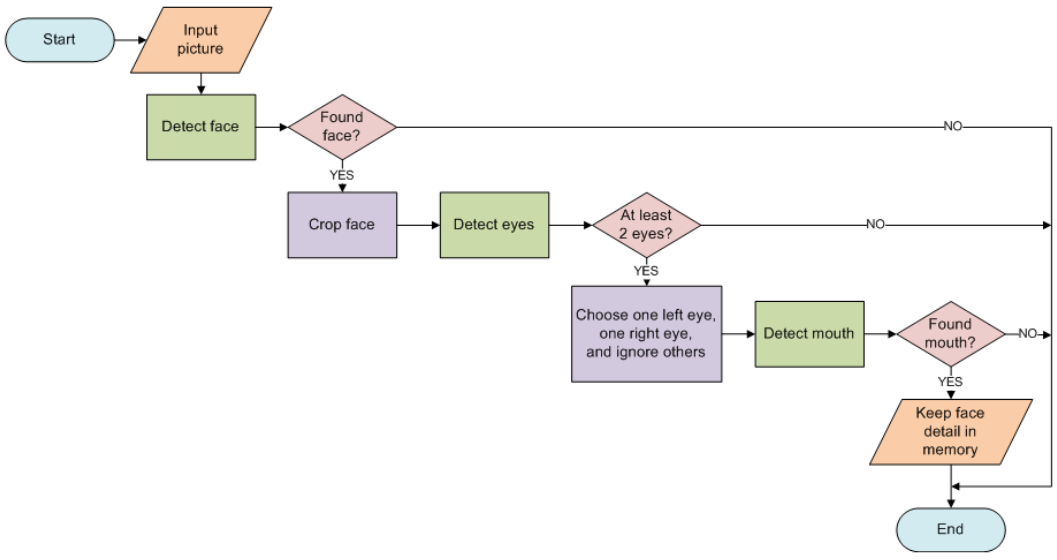


Figure 3.1: Face Detection Flow Chart

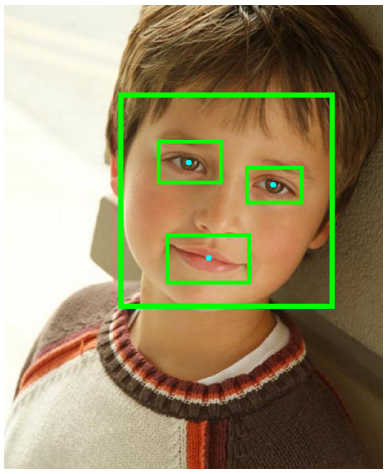


Figure 3.2: Face Detection

3.1.2 Face Contour Creation

The second function of the system is face contour creation. Our approach is to infer the face contour from main face features that can be obtained easily; two eyes and a mouth. We assume that these features are placed at a similar position for all people, so we may use these features to infer other face features that the face detector cannot detect; for this case, the face contour.

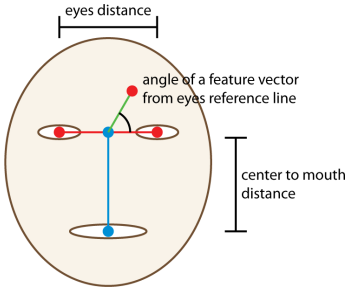


Figure 3.3: Angle from the Reference Line

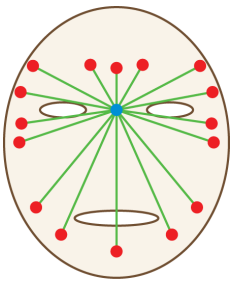


Figure 3.4: Face Contour Vectors from the Center Point

Creating a contour begins with getting the eyes and mouth position from the face detector, then calculating the center position between two eyes, and then creating the vector from the center position by using the predefined relationship between the eyes-mouth and face contour to get 16 contour positions around the face area and formed to be a face contour. Each predefined relationship defines the length of the vector from the center and the angle value from the reference line (the line between eyes) to the contour vector.

The raw data of absolute positions of each face has no meaning. The average of absolute positions cannot be used to infer any relationship because faces are placed at different positions, have different pose angles, and have different resolutions. The predefined relationship is gathered by calculating the relative position which is independent from those differences.

To improve the correctness of the predefined relationship, we find the average of each face contour position. As the human face is bi-symmetric, we also find the average between the left and right part of each face.

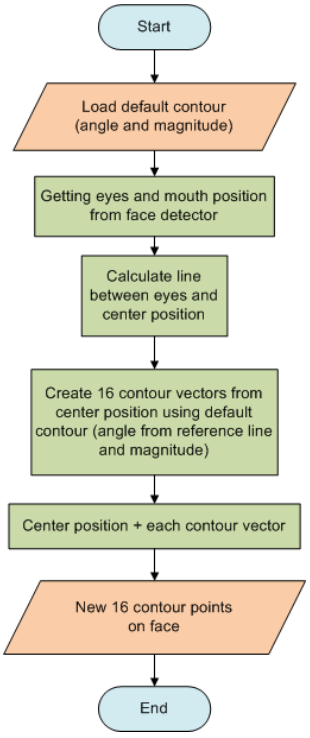


Figure 3.5: Face Contour Creation Flow Chart

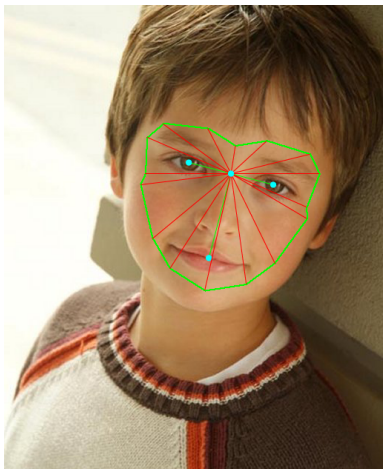


Figure 3.6: Contour Creation

3.1.3 Alignment

The third function of the Face Replacement System is alignment. Alignment is another important factor in order to create a more realistic result.

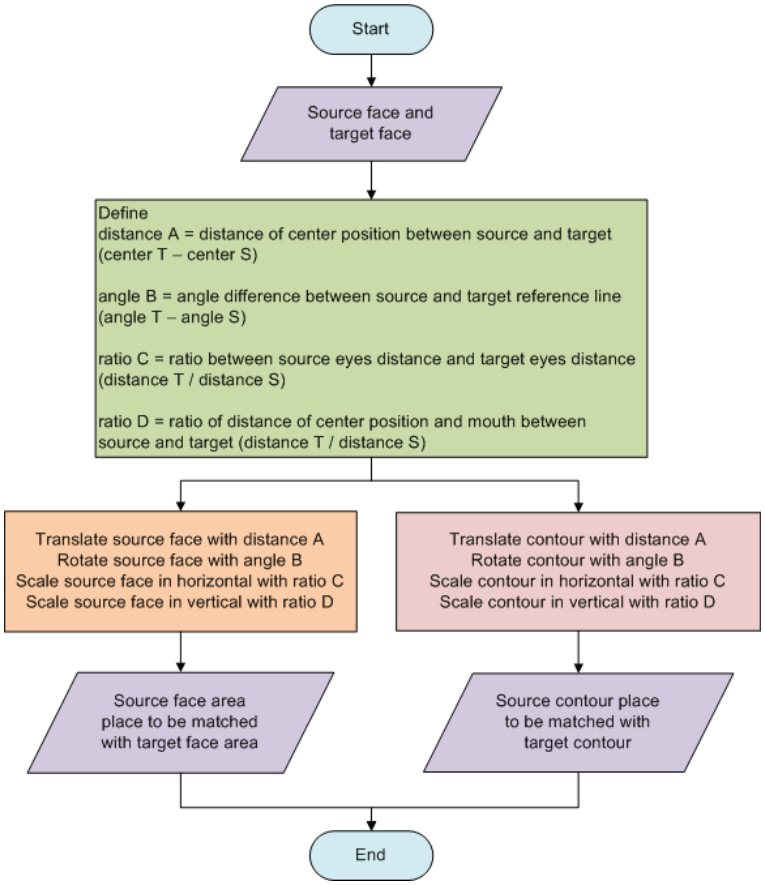


Figure 3.7: Alignment Flow Chart

The purpose of alignment is to have the eyes and mouth position of the source face to be at the same position as the target face. First, we get the eyes and mouth position of each face from the face detector. To do alignment, we translate the center position of the source face to match the center position of the target face. For rotation, we have to calculate the pose angle difference, and then compensate the pose angle of the source face to match the pose angle of the target face. For scaling, there are two parts. The first part is scaling on the horizontal direction in proportion to the distance between two eyes. The other part is scaling on the vertical direction in proportion to the distance between the center position and mouth. When those three transformations are applied, the alignment will match the source face to the target face.

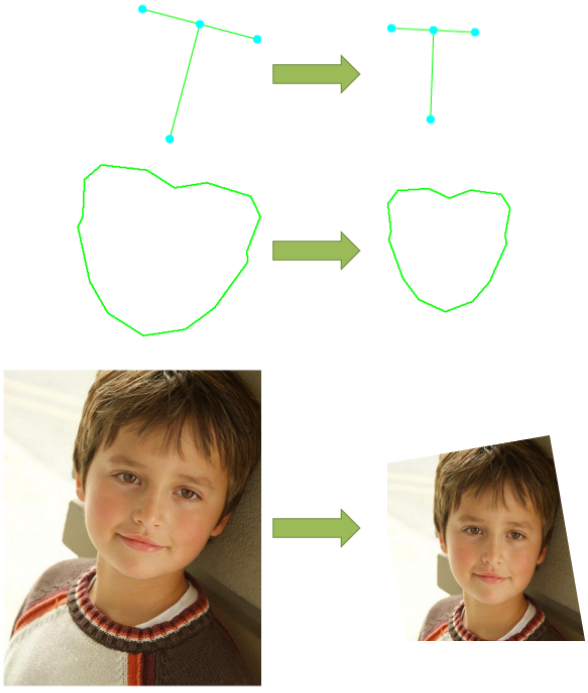


Figure 3.8: Alignment

3.1.4 Color Adjustment

The fourth function of the system is color adjustment. While the blending process tries to adjust the color of all pixels on the face patch to compensate color differences between

source and target pixels along the contour, the face patch will be adjusted regardless of color space limitation. Because a color space has a finite boundary, it is possible for the face patch to be saturated, e.g. too bright. The color transfer technique is optionally used to reduce those effects by mapping the color range statistically before passing to the blending process. Then the blending process will receive the face patch in which the colors are already almost the same, and then make changes only to minor differences on the boundary area.

Because the lighting problems are still found in the results, we have modified the color transfer to map colors by their percentiles. This approach can transfer some characteristics, i.e. the proportion of dark pixels and bright pixels will be preserved. The distribution of color is not just relocated to the same range but also be cloned as the same shape as shown in figure 3.9.

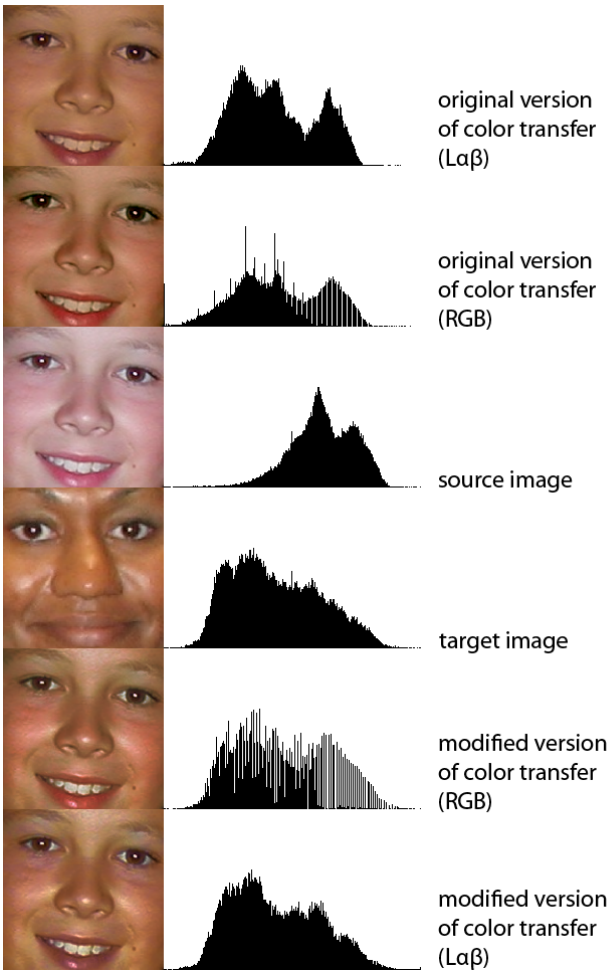


Figure 3.9: Color Transfer Techniques Comparison with Histogram

(a) Before (b) After

Figure 3.10: Color Adjustment Result

3.1.5 Blending

The last function of the system is blending. This part attempts to blend two pictures that may have different colors. We try to cancel the color discontinuity between inner and outer pixels along the compositing boundary. The pixels inside the boundary have to be adjusted to match the pixels outside the boundary.

(a) Before (b) After

Figure 3.11: Blending Result

There are 3 inputs for this process, which are source face image as foreground, target face image as background, and mask as a boundary information. The approach begins with rendering the mask of the source image from the contour, and then aligning the source face and the mask to be the same alignment as the target face. The next step is to blend the color of the source and target image along the boundary of the masking area, and the result

will be as shown in the figure 3.12.

Figure 3.12: Poisson Seamless Cloning Example

The techniques we have selected to do this process are Poisson Cloning and Mean-Value Coordinates Cloning.

3.1.5.1 Poisson Cloning

Poisson Cloning is one of the methods developed by Pérez et al[6] for doing seamless cloning between images.

The process begins with calculating the gradient field by calculating the second derivative of color value of those N pixels inside the boundary using the Laplacian operator.

$$\Delta . = \frac{\partial^2 .}{\partial x^2} + \frac{\partial^2 .}{\partial y^2}$$

Figure 3.13: Laplacian Operator

The process subtracts the nearby values both in the X axis and Y axis of the foreground in order to get the first derivatives, then the process does the same with the first derivatives to get the second derivatives of both axes, then it adds both second derivatives together to get the gradient field as shown in figure 3.14

Then, we map these gradient values of those N pixels into the vector b. The pixels at the boundary of the contour will be subtracted by the color value of its neighbor pixels of the target face image in order to set the constraint that the pixels inside the boundary must match with pixels outside the boundary.

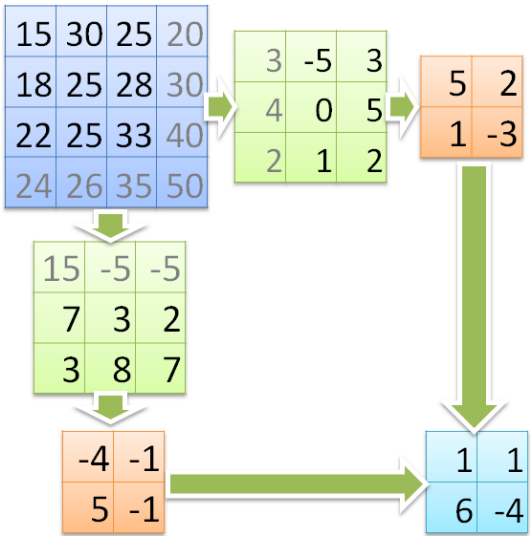


Figure 3.14: Laplacian Operation with the Matrix of Pixel Values in the Image

Next, we express the influence relationship of each inside pixel to its neighbor pixels into the matrix A. Each pixel inside the boundary area will correspond with a column or a row of the matrix. The total N pixels will result in the matrix A sized N×N. Each column of the matrix will consist of a relationship between the pixels itself and its neighbor pixels. We have to solve the linear equation Ax = b to get the vector x which yields N color values of the final result, then we map those N values back to inside pixels.

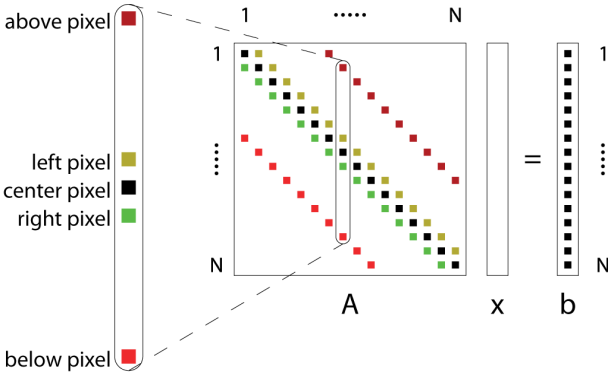


Figure 3.15: Solving Poisson Equation

3.1.5.2 Mean-Value Coordinates Cloning

Mean-Value Coordinates Cloning is another method developed by Farbman et al[3] to do seamless cloning between images. This method needs the same inputs as Poisson Cloning.

The idea of this method is to create an error correction to cancel color discontinuity around the border. Rather than solving a large linear system, this method finds the interpolation of correction at each interior pixel which is corresponding to Mean-Value Coordinates. Mean-Value Coordinates will determine different mixtures of values along the boundary according to the coordinate by defining a weighted combination of each pixel. Angle and distance, which are calculated from coordinates, are parameters that are used to determine the weighted combination. The weighted combination value of each pixel tells how much the portion of each color difference around the contour will affect to the interior pixel.

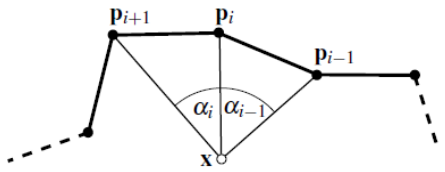


Figure 3.16: Mean-Value Coordinates Cloning

From the figure 3.16, Pixel x will be affected by the all pixels p on the contour with weights. Each pixel p has corresponding color differences between source and target pixels. The larger angle between pixel p, pixel x, and the next contiguous pixel p means the dominant of the p in that area, so we assign more weight to affect the pixel x. Also the shorter distance between pixel x and each pixel p, the more the weight will be affected by the color differences. If pixel x is located at the center of the inside area, it is likely to have equal weights for all pixels p, so it will be adjusted by a neutral mixture of all pixels p. The pixel x near the contour is likely to be biased to the nearest pixel p, so it will be mostly adjusted by the color difference of that pixel p to make the border looks seamless by achieving almost the same color.

CHAPTER4
IMPLEMENTATION

This chapter describes how to implement the Face Replacement System. It includes system environments that describe the requirements of the Face Replacement System, the process of implementing a user interface, and also the Face Replacement System.

4.1 System Environment

Operating System and Utilities Application requirements

- Operating System
  - Microsoft Windows with .NET Framework 3.5 SP1
- Integrated Development Environment
  - Microsoft Visual Studio 2008
  - Microsoft Expression Blend 3
- Additional Library
  - OpenCV library with Emgu.CV-2.0.1.0 wrapper
  - dnAnalytics 2009.8

4.2 Implementation of the User Interface

Windows Presentation Foundation (WPF) is a next-generation presentation system for building Windows client applications. We use WPF in order to create the graphical user interface which looks more attractive and easy to use. The main functionality of our interface is the ability to drag and drop the items. For this interface, users can drag photograph files

into the application window. The photo will show on the interface. The application will detect faces automatically and shows the line of face contour. To do the face replacement, users drag a face to any other face they want. The replacement will be done automatically and the result will be shown when it finishes. Users can keep a face to use across photos by dragging a face into the box below. Later on, users can drag a face from the box to another face in a photo that the users want.

Figure 4.1: The Early Version of the User Interface of the Face Replacement System

Sometimes, the face detector fails to detect the precisely correct position, and results in an incorrect face contour. The edit button is available for editing the face contour manually by the users. Users can improve the correctness of the contour by adjusting the contour to cover only the facial skin and avoid having the contour cross any details.

4.3 Implementation of the Face Replacement Engine

We have implemented the Face Replacement System using C# language because it is a high-level language and there is the unsafe code feature which allows some portion of code to be executed with low overhead as low-level language. It is also a current flagship

language from Microsoft that can integrate with most Microsoft Technology such as WPF.

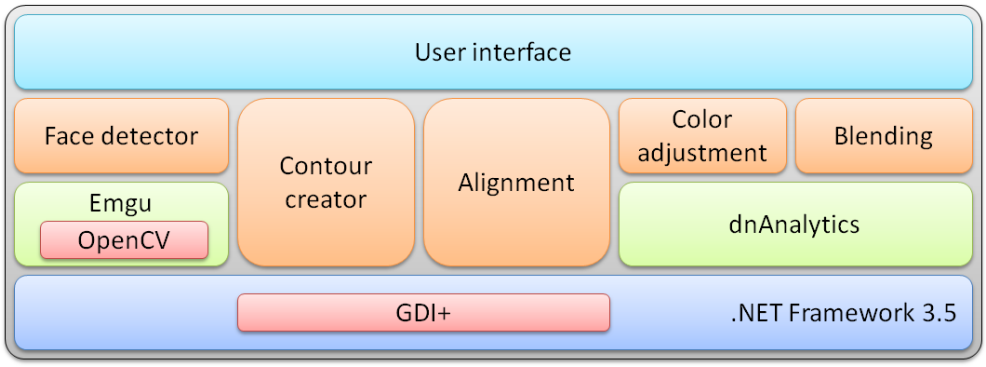


Figure 4.2: The Face Replacement System Architecture Diagram

For the face detection part, we have developed the functionality of face detection from Emgu, which is the wrapper of OpenCV. There are cascade trained files used for detecting face, eyes, and mouth, which are published by Modesto Castrillon Santana [8]. We have implemented the face detection to be able to detect the face region. Then, we divide the subregions and use them to detect eyes and mouth. We divide the subregions to be top-left for left eye, top-right for right eye, and below for mouth. There is a high possibility that the eye detection will give a false positive result. We assume that if it detects more than one eye position at each side of eyes, we will choose the one which is the nearest to 30% from each side.

As we did not use the commercial face detector as used in Bitouk's paper [1], which is able to detect the face with out-of-plane pose, we have considered only 2D plane poses.

For the face contour creation part, there are trigonometric functions available. There are Point and Vector classes which can be used for calculating the vector length and angle between vectors in a 2D plane.

We have developed a separate program for training a face contour. This program was written to help collect face contour training information. This program is not intended for end users because training face contours is not an end users' task. The face contour training program will be described in the section "Preparation of contour initiation data" later in this chapter.

For the alignment part, there is the Matrix class, which is used for transformation and is in the GDI+ library, which is wrapped in the .NET Framework, so we can define a set of operations such as rotate, translate, and scale. Then, we can use the matrix to transform the image and contour.

The color adjustment part has been implemented as suggested in the paper. We have also implemented our modified version which clones color distribution by percentiles from an image to another image.

For the blending part, there are two algorithms, Poisson and Mean-Value Coordinates. In the early stage of development, we have integrated the source code of the Poisson Image Editing, which is the open source project provided online by Tommer Leyvand [4]. The Poisson Image Editing source code is written in unmanaged C++ but our system was developed in .NET Framework environment. So, we have to study how it works and port the system from this source code in order to get the same functionality.

The contour obtained from the contour creation part will be used as loose selection input for Poisson cloning. Therefore, we have to render the contour to be mask area. We use GDI+ library to render this masking. As the original Poisson Image Editing source code is developed in C++, it uses TAUCS as a linear sparse matrix solver in order to solve the Poisson equation.

As we have implemented the system on the .NET platform, a problem occurs when we try to combine our system with old static library, which is an unmanaged code. So, we have to select another linear solver that can work with our system. We have selected dnAnalytics, which is an open source numerical library written in C#, and also supports sparse matrix.

Another blending algorithm is Mean-Value Coordinates algorithm. In the later stage, we have developed this algorithm according to the research paper. dnAnalytics also contains statistics facilities so we use it in some steps of this calculation.

4.4 Preparation of Contour Initiation Data

We have developed a separate program for face contour training. The purpose of this program is to extract information regarding face contour, including the position of eyes and mouth, and sixteen contour points around a face, gathered from many faces to define contour initiation data. The program will collect raw data as absolute positions. This raw data will be converted to relative positions from main face features to face contour positions, be averaged to be a converged relationship, and then will be used as a predefined relationship to construct a face contour that can be approximately matched with any face.

The face images that we have used for training face contours are from the Face of Tomorrow collection by photographer Mike Mike on Flickr istanbulmike's photostream [5]. We have randomly selected 124 images which are approximately sized 400×500 pixels in order to do the face contour training.

The usage of this program is to record the feature positions from a large collection of face images. At first, trainers have to define the position of left eye, right eye, and mouth sequentially. Then, the 16 contour points will appear on the face image which might not exactly match to the face contour. What trainers have to do is to manually edit each contour point to be at the appropriate position as the face contour, and then save the contour points information into a file.

Figure 4.3: The Face Contour Training Program

CHAPTER5
TESTING AND EVALUATION

This chapter describes how we perform the usability testing to compare the time requirements between using an image retouching tool and the Face Replacement System, score the user satisfaction, and evaluate user interface. We also conducted an output evaluation to score the output satisfaction.

5.1 Usability Testing

We performed this testing by comparing the time to do face replacement between retouches by using the image retouching tool and the Face Replacement System. We selected the popular image retouching tool, Adobe Photoshop. We invited 7 volunteers who are our ICT friends and have experience, but not expertise, in Adobe Photoshop to perform this testing. The machine we used for this testing was a Dell Optiplex 330, using Intel Core2Duo E8400 3.0 GHz with 2 GB of RAM. The test volunteers had to replace faces from the images they had selected on their own by firstly using Adobe Photoshop, then using the Face Replacement System. Then, we recorded the time the volunteers spent for replacing faces using Adobe Photoshop and the time spent using the Face Replacement System.

After finishing this testing, the volunteers had to score their user satisfaction, which depended on the time requirements and the output images they created from using both Adobe Photoshop and the Face Replacement System. A percentage score was based on our four scales of measurement; Unacceptable (0%), Acceptable (33.33%), Good (66.67%), and Perfect (100%).

From this testing, the mean average required time for using Adobe Photoshop was 623.29 seconds, whereas the mean average required time for using the Face Replacement System was 245.43 seconds.

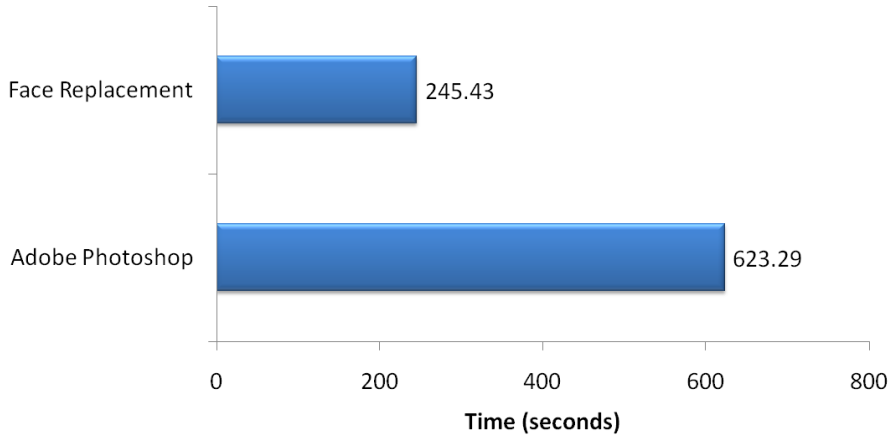


Figure 5.1: The Graph Shows the Mean Average Required Time for both Adobe Photoshop and the Face Replacement System

For the user satisfaction, the mean average satisfaction score for Adobe Photoshop was 47% (in Acceptable level), whereas the mean average satisfaction score for the Face Replacement System was 70 % (in Good level).

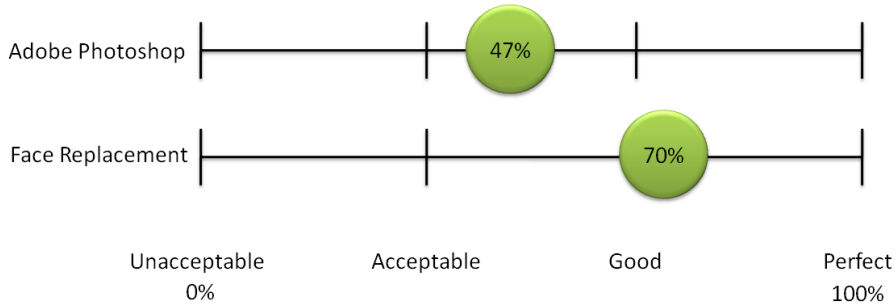


Figure 5.2: The Diagram Shows the Mean Average User Satisfaction Scores of both Adobe Photoshop and Face Replacement System

From the results, we concluded that users can do face replacing by using the Face Replacement System approximately 2.5 times faster than the image retouching tool. We also concluded that users are more satisfied with the required time and the results from the Face Replacement System rather than doing it on their own by using the image retouching tool.

These are the examples of the output images which the volunteers created from both Adobe Photoshop and Face Replacement System. Note that the volunteers picked their own images to do this testing.

(a) Adobe Photoshop (b) Face Replacement System

Figure 5.3: The Output Images Created from Adobe Photoshop and the Face Replacement System

As shown in figure 5.3, the volunteer can locate the appropriate position of the facial features, but has difficulty in adjusting color and lighting when using Adobe Photoshop. However, our system can deal effectively with the color and lighting. We can see from other output images created from the Face Replacement System that the alignment is not as good as a human can do from Adobe Photoshop because of the lower accuracy of the face detector and the different appearances between the selected faces. Also, the volunteers are not familiar with system functionality, so they are not able to adjust the contour to be appropriate.

We also conducted the user interface evaluation by interviewing the volunteers about the system usage, including the user interface design. Figure 5.4 shows the user interface which is used for this evaluation.

We received the feedback from the volunteers as follows:

- The program does not provide the instruction for the users who use our system for the first time.
- The color design of the interface is too translucent.
- The option for selecting the techniques to do face replacement is described with technical terms which are difficult to understand for common users.
- Users misunderstand the contour editing feature in that they thought they had to adjust

those three pivots into eyes and mouth position, but the feature is intended for adjusting the contour in such a way that it does not to cross any face features.

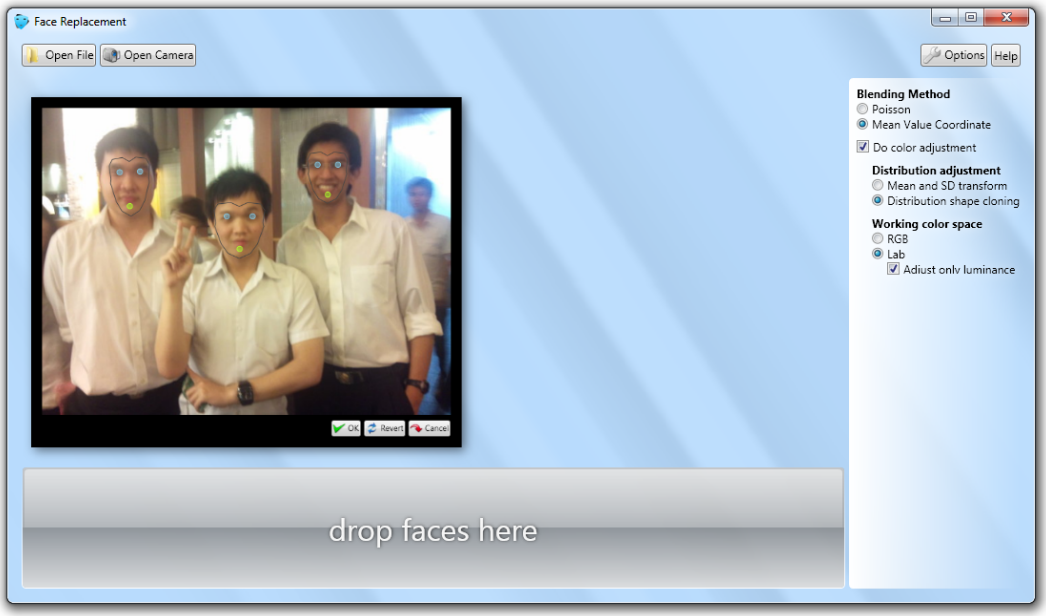


Figure 5.4: The User Interface Version that Volunteers used for this Evaluation

From the feedback, we improved the design and some functions of the system to be more effective as following:

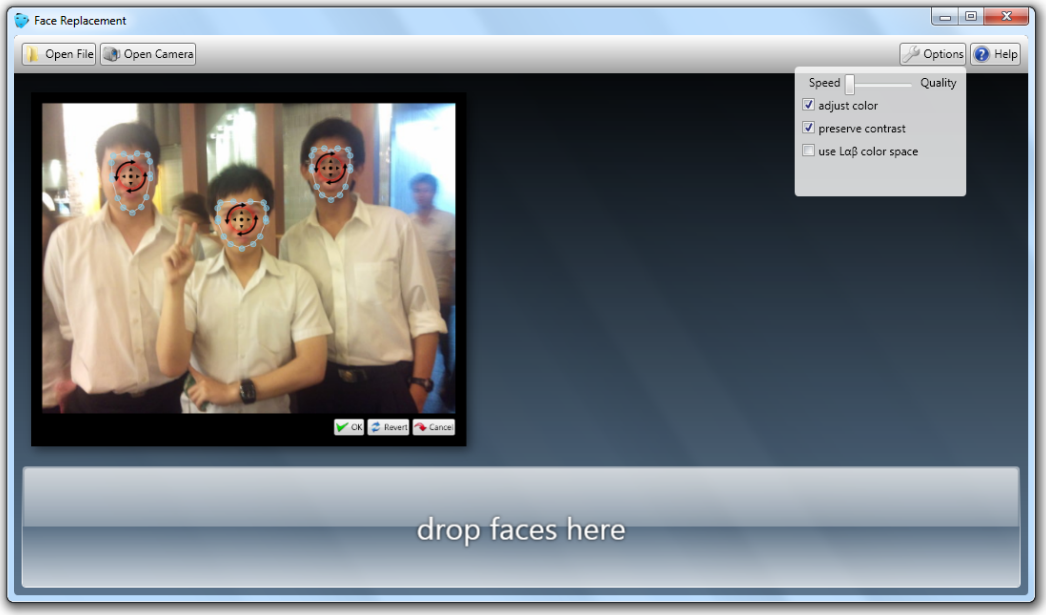


Figure 5.5: The Recent Version of User Interface

- We have created a Help feature providing the instruction of the program.
- We have adjusted the color to be more opaque to look like the working space.
- We have adjusted the option menu to be described with simple words and easy for common users to understand.
- We have adjusted the contour editing feature to meet the user expectation. Users can translate the contour by grabbing the center circle pad, rotate by sliding around the center circle pad, and scale by moving each point of the contour points directly.

5.2 Output Evaluation

We conducted an evaluation to know how realistic the output images from the Face Replacement System will be compared to the output images from the image retouching tool. We invited another 5 volunteers to evaluate all the output images from the previous testing and score them using the percentages of the four scales of measurement as in the previous evaluation. The volunteers did not know which method was used to create each output image.

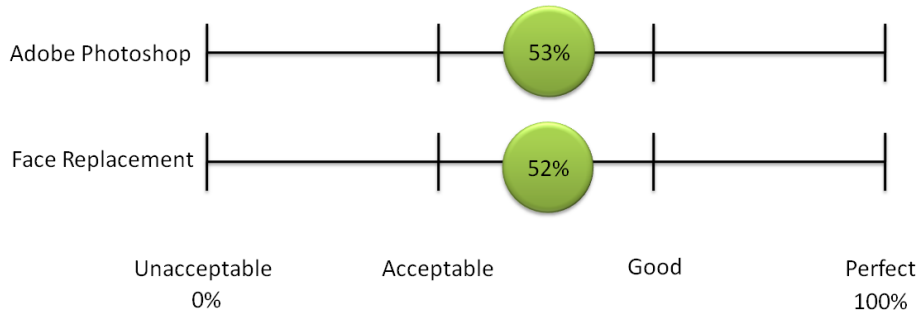


Figure 5.6: The Diagram Shows the Mean Average Quality Scores of both Adobe Photoshop and the Face Replacement System

From this evaluation, the quality score for the output images created using the image retouching tool was 53.54%, whereas the score for the output images created using our system was 52%. Both scores are in the Acceptable level.

We concluded that the face replacing output images from both methods are acceptable for participants. Both the quality scores of the output images using the image retouching tool and the Face Replacement System are approximately the same, which suggests our system can achieve the same output quality as human can do.

Note that the various image appearances the volunteers picked to do testing on and the various levels of image retouching skill can affect the required time and the quality of output images. The results may have been affected with these minor uncontrolled test variables.

CHAPTER6
CONCLUSIONS

This chapter describes the benefits, problems and limitations of the Face Replacement System. It also describes future works which can extend the research.

6.1 Benefits

6.1.1 Benefits to Project Developers

- To learn and practice C# language on .NET Framework.
- To learn about OpenCV library and understand the concept of face detection.
- To practice and apply the knowledge from Computer Graphics class.
- To practice and apply the concept of color adjustment technique; Color Transfer between Images, in a new way.
- To understand the concept of blending techniques; Poisson Cloning, and the state of the art technique; Mean-Value Coordinate Cloning.
- To learn and practice how to create a modern design of user interface by using Windows Presentation Foundation.

6.1.2 Benefits to Users

- To help users reduce the required time for face editing and to easily get realistic face images.
- To be able to do face de-identification in order to do privacy protection that is more pleasing than using a mosaic on the face.
- To be able to do try-on on various dressings with their face identity.

- To be able to put a user's face into the event or the photograph of a location where they will never be again.

6.2 Problems and Limitations

- The face images which will be used in the system should be in-plane pose, not with out-of-plane pose, because our system can only deal with 2D operations. The in-plane pose should not be extreme because the face detector might not be able to detect the face.
- The face images should not be blocked by any obstruction, i.e. no wearing glasses, no hair covering the face, etc.
- The images which have the extreme difference of lighting condition on each face might get unsatisfied results, i.e. different direction of light projected on the face, extreme different color dynamics or contrast, etc.
- The system is not fast enough to run with real-time video stream. One of the reasons is C# language is a high-level language which has a high overhead.
- Although the system can run on 64-bit version of Windows, it is not a true 64-bit application because OpenCV is implemented as 32-bit.
- The system still cannot achieve 100% accuracy on face detection because of some circumstances such as a too-small size of face, eyes are too small, etc. which also affects the correctness of face contour creation process.
- Due to the limitation of 32-bit addressing scheme, the linear solver cannot deal with a matrix larger than 65536×65536 elements. This means that when applying the Poisson blending method, the total pixels of the face should not be more than 65536 pixels or approximately 256×256 pixels.
- A machine that has a low performance graphics processor might experience a slow response and jaggedness on user interface rendering.

6.3 Future Works

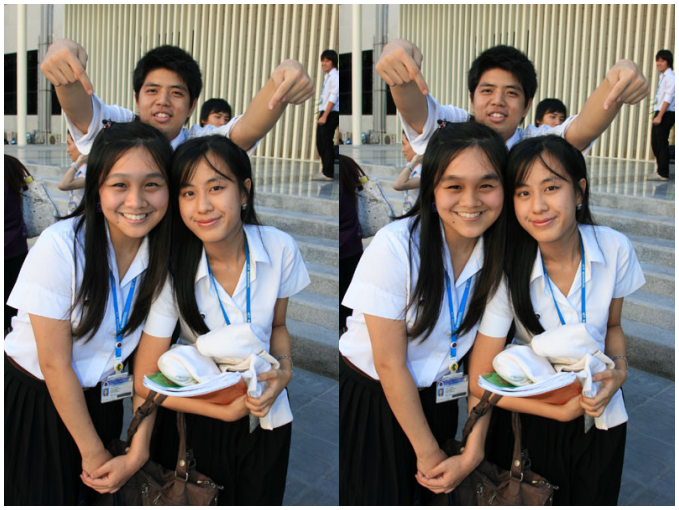
- To develop the system to be able to replace faces between images with out-of-plane posture by 3D operation.
- To develop the system to be able to process in real time such as in a video stream.
- To develop the system to have fully automatic face contour editing not to cross any face features.
- To develop the system so that it is tolerant of obstruction on the face region, i.e. hair, glasses, etc.
- To develop the system to be able to replace each face feature; eyes, nose, and mouth separately.

APPENDIX

APPENDIXA
TESTING AND EVALUATION RESULTS

A.1 Usability Testing

Volunteer 1



(a) Adobe Photoshop (b) Face Replacement

Figure A.1: The Output Images created by Volunteer 1

Method	The required time	The user satisfaction score
Adobe Photoshop	355 seconds	70%
Face Replacement	99 seconds	60%

User interface comment from Volunteer 1

If possible, the program should have ability to adjust the offset, or to scale the contour.

Volunteer 2

(a) Adobe Photoshop (b) Face Replacement

Figure A.2: The Output Images created by Volunteer 2

Method	The required time	The user satisfaction score
Adobe Photoshop	535 seconds	40%
Face Replacement	174 seconds	65%

User interface comment from Volunteer 2

The objects behind the interface windows should not be seen. There should be a kind of Pop-Up to tell the instruction.

Volunteer 3

(a) Adobe Photoshop (b) Face Replacement

Figure A.3: The Output Images created by Volunteer 3

Method	The required time	The user satisfaction score
Adobe Photoshop	1069 seconds	40%
Face Replacement	323 seconds	60%

User interface comment from Volunteer 3

The icons behind the interface can be seen when run on a desktop.

Volunteer 4

(a) Adobe Photoshop (b) Face Replacement

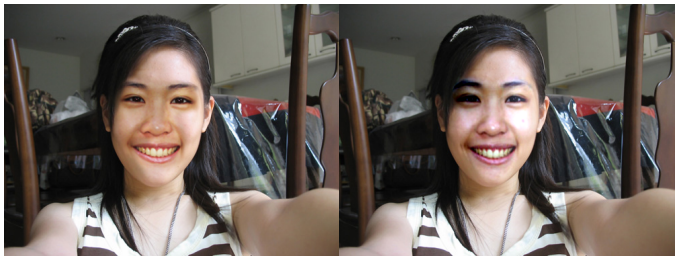
Figure A.4: The Output Images created by Volunteer 4

Method	The required time	The user satisfaction score
Adobe Photoshop	1016 seconds	10%
Face Replacement	293 seconds	91%

User interface comment from Volunteer 4

There are no instructions to use the program. Other users who use it for the first time will have no idea how to use it.

Volunteer 5



(a) Adobe Photoshop (b) Face Replacement

Figure A.5: The Output Images created by Volunteer 5

Method	The required time	The user satisfaction score
Adobe Photoshop	352 seconds	45%
Face Replacement	317 seconds	85%

User interface comment from Volunteer 5

The Save button should be on the above panel. The image on the Edit button should also be changed. There should be a 'Getting Started' feature.

Volunteer 6

(a) Adobe Photoshop (b) Face Replacement

Figure A.6: The Output Images created by Volunteer 6

Method	The required time	The user satisfaction score
Adobe Photoshop	575 seconds	60%
Face Replacement	287 seconds	55%

User interface comment from Volunteer 6

Users may want to zoom the image to easily edit the contour. Users may have no idea how to use some features, such as users can double click on the face to revert the contour.

Volunteer 7

(a) Adobe Photoshop (b) Face Replacement

Figure A.7: The Output Images created by Volunteer 7

Method	The required time	The user satisfaction score
Adobe Photoshop	461 seconds	65%
Face Replacement	225 seconds	65%

User interface comment from Volunteer 7

Some users may have no idea about the techniques they have to select in the option menu. To correctly edit the contour is a bit difficult to achieve.

Volunteer	Adobe Photoshop	Face Replacement
1	355	99
2	535	174
3	1069	323
4	1016	293
5	352	317
6	575	287
7	461	225
Average	625.29	245.42

The table above shows the required time results (in seconds) from all volunteers and the mean average required time from both methods. The table below shows the user satisfaction scores (in %) from all volunteers and the mean average scores from both methods.

Volunteer	Adobe Photoshop	Face Replacement
1	70%	60%
2	40%	65%
3	40%	60%
4	10%	91%
5	45%	85%
6	60%	55%
7	65%	65%
Average	47%	70%

A.2 Output Quality Evaluation

Figure	Volunteer					
	1	2	3	4	5	Average
A.1 (a)	100%	100%	97%	100%	100%	99.4%
A.1 (b)	70%	80%	15%	65%	90%	64%
A.2 (a)	50%	5%	5%	50%	30%	28%
A.2 (b)	100%	70%	65%	40%	70%	69%
A.3 (a)	0%	10%	10%	0%	25%	9%
A.3 (b)	30%	10%	5%	0%	30%	15%
A.4 (a)	0%	20%	25%	0%	30%	15%
A.4 (b)	100%	90%	45%	55%	40%	66%
A.5 (a)	80%	70%	95%	65%	80%	78%
A.5 (b)	50%	80%	85%	60%	90%	73%
A.6 (a)	80%	65%	67%	100%	70%	76.4%
A.6 (b)	90%	60%	30%	60%	70%	62%
A.7 (a)	95%	75%	55%	50%	70%	69%
A.7 (b)	20%	20%	5%	0%	30%	15%

The table above shows the results of output quality satisfaction scores and the mean average score of each output image. From these results, the mean average quality scores for both methods are shown in the table below.

Method	Average quality score
Adobe Photoshop	53%
Face Replacement	52%

REFERENCES

[1] D. Bitouk, N. Kumar, S. Dhillon, P. N. Belhumeur, and S. K. Nayar. Face swapping: Automatically replacing faces in photographs. *ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH)*, August 2008.

[2] Bruce Gooch Erik Reinhard, Michael Ashikhmin and Peter Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34 — 41, September/October 2001.

[3] Zeev Farbman, Gil Hoffer, Yaron Lipman, Daniel C. Or, and Dani Lischinski. Coordinates for instant image cloning. *ACM Trans. Graph.*, 28:1—9, 2009.

[4] Tommer Leyvand. Advanced Topics in Computer Graphics: Poisson Image Editing. `http://leyvand.com/research/adv-graphics/ex1.htm`, October 2009.

[5] Mike Mike. Face of Tomorrow - Flickr istanbulmike's photostream. `http://www.flickr.com/photos/istanbulmike`, October 2009.

[6] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH'03)*, 22(3):313—318, 2003.

[7] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 2004.

[8] Modesto Castrillon Santana. Facial feature xml cascades for OpenCV. `http://mozart.dis.ulpgc.es/Glas/modesto_eng.html`, August 2009.

BIOGRAPHY

NAME Mr. Jirayus Laemthong

DATE OF BIRTH 22 March 1988

PLACE OF BIRTH Chaing Mai, Thailand

INSTITUTIONS ATTENDED Nawamintrachinutit Satriwitthaya Putthamonthon School, 2006

High School Diploma

Mahidol University, 2010

Bachelor of Science (ICT)

NAME Mr. Worachet Sae-lim

DATE OF BIRTH 10 November 1987

PLACE OF BIRTH Bangkok, Thailand

INSTITUTIONS ATTENDED Wat Nuannoradit School, 2006

High School Diploma

Mahidol University, 2010

Bachelor of Science (ICT)

NAME Mr. Damri Samretwit

DATE OF BIRTH 1 September 1987

PLACE OF BIRTH Bangkok, Thailand

INSTITUTIONS ATTENDED Triamudomsuksa Pattanakarn School, 2006

High School Diploma

Mahidol University, 2010

Bachelor of Science (ICT)