

การจำลองสามมิติโดยอัตโนมัติอย่างเต็มที่เพื่อลดการคัดลอกและเสริมสวยใบหน้า



นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

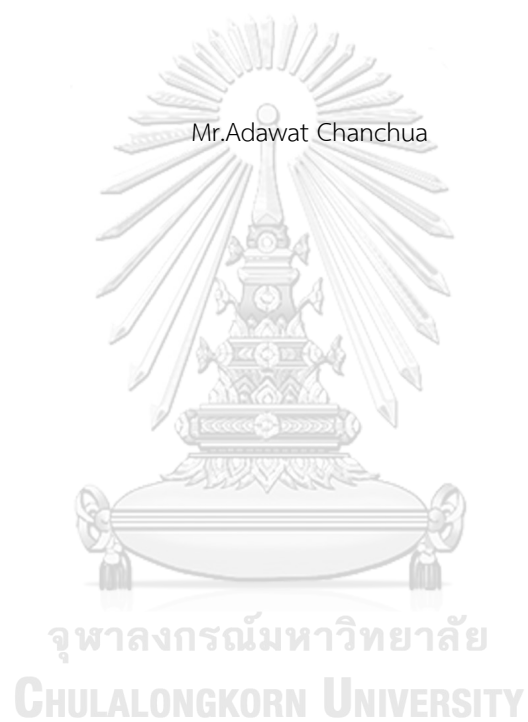
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2561

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

FULLY AUTOMATIC 3D FACIAL COSMETIC SURGERY SIMULATION



A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2018

Copyright of Chulalongkorn University

Thesis Title                      FULLY AUTOMATIC 3D FACIAL COSMETIC SURGERY SIMULATION  
By                                      Mr.Adawat Chanchua  
Field of Study                      Computer Science  
Thesis Advisor                      Associate Professor Nuttapong Chentanez, Ph.D.

---

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

THESIS COMMITTEE

.....                      Dean of the Faculty of Engineering  
(Professor Supot Teachavorasinskun, D.Eng.)

.....                      Chairman  
(Assistant Professor Nattee Niparnan, Ph.D.)

.....                      Thesis Advisor  
(Associate Professor Nuttapong Chentanez, Ph.D.)

.....                      Examiner  
(Pitchaya Sitthi-amorn, Ph.D.)

.....                      External Examiner  
(Borom Tunwattanapong, Ph.D.)



เอตาเวิร์ค จันทร์ เชื้อ: การจำลองสามมิติโดยอัตโนมัติอย่างเต็มที่เพื่อหัตถการ ศัลยกรรม เสริมสวย ใบหน้า.

(FULLY AUTOMATIC 3D FACIAL COSMETIC SURGERY SIMULATION) อ.ที่ปรึกษาหลัก : รศ. ดร. ญัฐพงศ์

ชินธเนศ, 37 หน้า.

ก่อนดำเนินการศัลยกรรมตกแต่งความงามใบหน้า โดยทั่วไปผู้รับบริการจะต้องเข้าขั้นตอนการสัมภาษณ์โดยใช้ภาพตัวอย่าง โดยภาพตัวอย่างประกอบด้วย ภาพก่อนและหลังการทำศัลยกรรมของบุคคลอ้างอิง ซึ่งอาจนำไปสู่ความเข้าใจที่คลาดเคลื่อนระหว่างผู้รับบริการและศัลยแพทย์ได้ วิทยานิพนธ์เล่มนี้นำเสนอการจำลองสามมิติโดยอัตโนมัติอย่างเต็มที่ทั้งกระบวนการ เพื่อจำลองใบหน้าหลังการทำศัลยกรรมของผู้เข้ารับบริการ เริ่มจากการนำเสนอวิธีการกำหนดตำแหน่งและการจัดเรียงภาพโดยอิงจากหัตถการในปัจจุบัน จากนั้นสร้างภาพเดลตา (Delta Image) เรายังนำเสนอภาพเดลตาเพื่อแก้ปัญหาการไม่สามารถหาชุดข้อมูลขนาดใหญ่ ของภาพใบหน้า ก่อนและหลังการทำศัลยกรรมได้อีกทั้งยังได้นำเสนอแบบจำลองโครงข่ายประสาทแบบคอนโวลูชันแบบโปรแกรมเข้ารหัสอัตโนมัติ (Convolutional Autoencoder) เพื่อเลือกใบหน้าที่คล้ายที่สุด นอกจากนี้ยังนำเสนอการจำลองใบหน้าหลังการทำศัลยกรรมโดยใช้หัตถการเดียวกับใบหน้าที่เลือก สุดท้าย เราตรวจสอบความสมเหตุสมผลโดยใช้แบบสำรวจเชิงจิตวิสัย และบันทึกจุดสำคัญของการพัฒนาไว้อีกด้วย

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

ภาควิชา : . . . วิศวกรรมคอมพิวเตอร์ . . . . . ลายมือชื่อนิสิต . . . . .

สาขาวิชา : . . วิทยาศาสตร์คอมพิวเตอร์ . . . . . ลายมือชื่อ อ.ที่ปรึกษาหลัก . . . . .

ปีการศึกษา . . . . . 2561 . . . . .



## 6170984821: MAJOR COMPUTER SCIENCE

KEYWORDS: RENDERING / FACIAL SURGERY / SIMULATION / MACHINE LEARNING

ADAWAT CHANCHUA : FULLY AUTOMATIC 3D FACIAL COSMETIC SURGERY SIMULATION. ADVISOR

: Associate Professor Nuttapong Chentanez, Ph.D., 37 pp.

Before cosmetic surgery procedures, the patients are commonly involved in the consultation process using interviews and reference images. The reference images typically consist of pre-post surgery images of other patients, leading to misunderstandings between patients and the surgeon. This thesis presents a fully automatic pipeline to simulate the whole face of post-surgery results. We first establish a 3D face registration and alignment based on the face surgery procedures of the current day and then generate the delta image. We proposed the delta image to solve the lack of dataset dilemma of the pre-post surgery face images. We also propose a convolutional autoencoder model to select the most similar face. Furthermore, we simulate the post-surgery results using surgery procedures retargeting. The last section validates the results using a subjective survey and shows the implementation notes.



Department : . . . . Computer Engineering . . . . Student's Signature . . . . .

Field of Study : . . . . Computer Science . . . . Advisor's Signature . . . . .

Academic Year . . . . . 2018 . . . . .

## Acknowledgements

I want to thank Associate Professor Nuttapong Chentanez, advisor of this thesis, for the accurate advice, mission and goal of this research. He always has the answer to needs, and his starting papers guild this research to reach the goal. I also would like to thank Intelligent System Lab (ISL) for the support and provides 3D face meshes for the experiments.

Thank Associate Professor Proadpran Punyabukkana and Associate Professor Thanarat Chalidabhongse for referring me to research computer graphics. Without the referring, it seems impossible for the computer graphics research for the part-time master student. Additionally, I would like to thank all of the lecturers in the department of computer engineering for the foundation courses, which built up knowledge and mindset for the research. Last but not least, thank all staff for the convenience and support during the study.

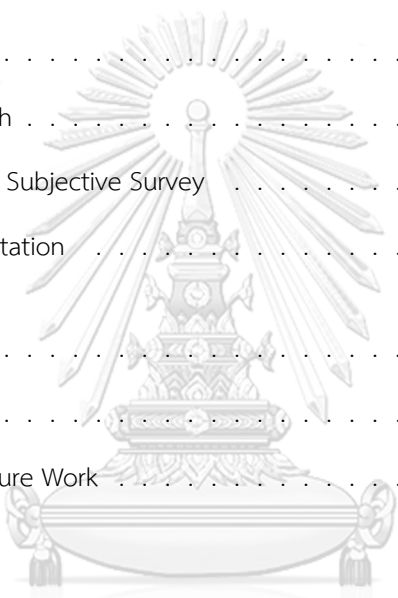
I thank my mother and family for their support, understanding of my situation, and encouragement. I also appreciated my classmates, even though they had graduated years before me. Without the good energy around me, it's impossible to finish this challenge and laborious research (for me). Also, thank my friends as the respondents for the subjective survey. They gave me much advice and trialled my annoyance survey many times.

Lastly, not scientific, even though I am a researcher, I would like to thank a series of destinies I will never forget. It started by I got lost in SIGGRAPH Asia 2017 in Bangkok. That conference sparked me by showing clever stuff by the groups of weird people, then ignited me to register for a master degree in computer science. Then I spent every night for three years doing this enjoyment thesis.

# Contents

	Page
Abstract (Thai) . . . . .	iv
Abstract (English) . . . . .	v
Acknowledgements . . . . .	vi
Contents . . . . .	vii
List of Tables . . . . .	ix
List of Figures . . . . .	x
<b>Chapter</b>	
<b>1 Introduction . . . . .</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Problem Statement . . . . .	1
1.3 Thesis Questions . . . . .	2
1.4 Scope of Work . . . . .	2
<b>2 Background . . . . .</b>	<b>4</b>
2.1 Vector Space, Coordinate and the Transformation between Them . . . . .	4
2.1.1 World, Model, Camera, and Screen, Space . . . . .	4
2.1.2 Cartesian and Barycentric Coordinate . . . . .	6
2.2 Point Correspondence . . . . .	7
2.3 Graphics Rendering Pipeline and Vertex Color Shader . . . . .	8
<b>3 Related Works . . . . .</b>	<b>14</b>
3.1 Cosmetic Surgery Simulation Methods . . . . .	14
3.2 Beautiful Faces Definition and Surgical Procedures . . . . .	14
3.3 Face Registration in 2D and 3D . . . . .	15
3.4 Similarity Models for Face Image Query . . . . .	16
3.5 Facial Retargeting . . . . .	16
<b>4 Methods . . . . .</b>	<b>19</b>

Chapter	Page
4.1 2D and 3D Registration . . . . .	21
4.1.1 2D Registration . . . . .	21
4.1.2 3D Registration . . . . .	22
4.2 Face Alignment and Delta Image . . . . .	24
4.3 Face Similarity Search . . . . .	25
4.4 Surgical Procedures Retargeting . . . . .	26
<b>5 Result . . . . .</b>	<b>28</b>
5.1 Face Similarity Search . . . . .	28
5.2 Attractiveness-Based Subjective Survey . . . . .	29
5.3 Robustness and Limitation . . . . .	29
<b>6 Discussion . . . . .</b>	<b>34</b>
6.1 Implementations . . . . .	34
6.2 Conclusions and Future Work . . . . .	34
<b>Chapter</b>	
<b>A List of Publications . . . . .</b>	<b>1</b>
A.1 International Conference Proceeding . . . . .	1
<b>Biography . . . . .</b>	<b>2</b>



## List of Tables

Table	Page
2.1 Four Minimally-invasive procedures, include deinitions . . . . .	13
3.1 Twenty-six dominant points, include abbreviation and definition. . . . .	17
3.2 Four Minimally-invasive procedures, include deinitions . . . . .	18
3.3 Five real procedures, include deinitions . . . . .	18
5.1 We conducted a subjective survey with ten questions based on attractiveness attributes. .	32



## List of Figures

Figure	Page
1.1 The left and right face images are from different faces, and suppose method A estimates feature points for both left and right faces so that we can match the correspondence points represented by green lines. . . . .	3
2.1 The left and right face images are from different faces, and suppose method A estimates feature points for both left and right faces so that we can match the correspondence points represented by green lines. . . . .	5
2.2 The red points represent sparse-corresponding points estimated using the method proposed from Kazemi and Sullivan (2014). . . . .	8
2.3 Three samples of 3D face mesh estimated the correspondence points using White et al. (2019). Each same vertex colour on the different face meshes represents the same segment. For example, the orange colour represents the nose segment. . . . .	9
2.4 The graphics rendering pipeline contains four stages; application, geometry processing, rasterization, pixel processing. . . . .	9
2.5 The example application stage contains a camera on the left of the image and still face mesh on the right. . . . .	10
2.6 Geometric processing contains four sub-stages; vertex shading, projection, clipping and screen mapping. . . . .	10
2.7 Rasterization substage contains triangle setup and triangle traversal. Pixel processing substage contains pixel shading and merging. . . . .	10
4.1 DeltaNet pipeline is an end-to-end method. $\alpha$ represents the 3D mesh in semantic order registered from the 3D mesh and 2D image. Then we proceed rigidly alignment ( $\delta_m$ ). $E_n$ represents the encoder model to get the most similar face in the dataset ( $\delta_m^+$ ), map to the surgery procedures image ( $\delta_p^+$ ). And finally, we obtain the retargeted image ( $\delta_m^*$ ) described by $\delta_m^* = \delta_m + \delta_p^*$ . . . . .	20
4.2 The top image shows the improvement correspondence points estimation in the eyes area. The bottom image shows the vector field altering by facial landmarks. . . . .	21
4.3 Delta Image of Pre Surgery Face (Left) and Surgery Procedures (Right) . . . . .	23

Figure	Page
4.4 Convolutional autoencoder network architecture for the similarity search shown with hyperparameters. We utilized the encoder with 16 variables shown in the decoded layer. . . . .	26
4.5 We need an additional process by overlaying the original image using D mesh. The input image (Left). The input image with overlaying mesh (Middle). The rendered result (Right). . . . .	27
5.1 Our DeltaNet pipeline simulates the post-surgery image in 2D and 3D (bottom) without any manual settings by giving an image or 3D mesh as the input (top). The images were not used during training. . . . .	28
5.2 Training loss for the face similarity model demonstrate that our model can generalize from the training set to validation set (top). The table shows loss values from the training set (Train Loss) and validation set (Val Loss) given activation functions, epochs and learning rate (Lr) as hyperparameters. We selected LeakyReLU as a popular choice for modern deep learning as it suffers less from the vanishing gradient problem compared to Sigmoid and Tanh (bottom). . . . .	31
5.3 Our confusion matrix shows the overall simulation performance of our pipeline based on attractiveness attributes. The columns of each matrix represent the scores on a controlled group, whereas the rows represent the scores on a simulated group. The order relating to each question in table 5.1. . . . .	33
5.4 The images on the left represent the original facial images, whereas the right represents the simulated post-surgery images. Two examples show that our model does not perform well with the teeth area (Top row). Also, the model does not perform well for the face shape that lacks the dataset, for example, the male face. . . . .	33
6.1 Our system's pipeline . . . . .	35

## CHAPTER I

### INTRODUCTION

#### 1.1 Overview

Facial cosmetic surgery has been growing exponentially during the last two decades as a globally new definition of beautification, and the average cost is lowering Nappi et al. (2016). However, the specialist uses frontal and nasofrontal images for reference during the planning process.

The reference images commonly consist of the patient's images before the surgical procedures and pre-post the same surgical procedures images of the other patients. Using the mentioned method might lead to two issues. First, the pre-post image pairs are not from the patient under advisement, which might lead to an unexpected result. Second, frontal and nasofrontal images can only represent two angles and limit details from the other angles.

The existing works we found in the market are Vectra3D XT Verhulst et al. (2018), which represent tools to scan, select, edit, and visualize the changes in 3D. However, the tool requires several manual interactions that require specialist skills. Such as manually correspondence points marking in Vectra3D during the registration step and manually selecting and morphing during the editing step.

#### 1.2 Problem Statement

Deep learning successfully solves many complex problems in human face images, ranging from the changing age or gender to simulating facial expressions. However, deep learning requires a large dataset to predict the results. In the same way, making a model to predict post-surgery images also requires a large enough dataset.

Finding the pre-post image pairs is a tedious task. The only open dataset we found is the research output from Singh et al. (2009), containing 1800 pre-post surgery images on 900 subjects. This dataset size is still far from our expectation for deep neural network training. The next question



is, can we define the most beautiful face?

Definition of the ideal human face has been attempted since ancient times. One of the most known theories is the golden ratio. Then later applied to the face domain with neoclassical canon and facial golden ratio mask. A comparative study from Burusapat and Lekdaeng (2019) measured face ratios from Miss Universe and Miss Universe Thailand between 2001 and 2015. The next question is how cosmetic surgery procedures affect facial ratios.

The research from Nappi et al. (2016) shows the list of cosmetic surgery procedures and the affected area. We found that only chin augmentation and cheekbones reshaping procedures noticeably affect the overall facial ratios.

We also found that the beautiful face definition might differ between regions, ethnicities, trends, and times. For example, Asian people requests cheekbones reshaping more than other ethnic groups to make their faces look smaller Nappi et al. (2016). Another problem, an input face probably has more than one plausible post-surgery result. Hence, the post-surgery simulation models must be able to make use of a small dataset while still reflex the diversity of beautifications.

### 1.3 Thesis Questions

1. How to utilize the deep learning model with very limited datasets of pre-post surgery facial images to create a system that can suggest a result image of post-surgery from a face photo?
2. How do we validate whether the suggested result face photos are accurate suggestions?

### 1.4 Scope of Work

- To propose a fully automatic end-to-end method for predicting a post-surgery facial image from a regular face photo.
- To validate the end-to-end method using a subjective survey.

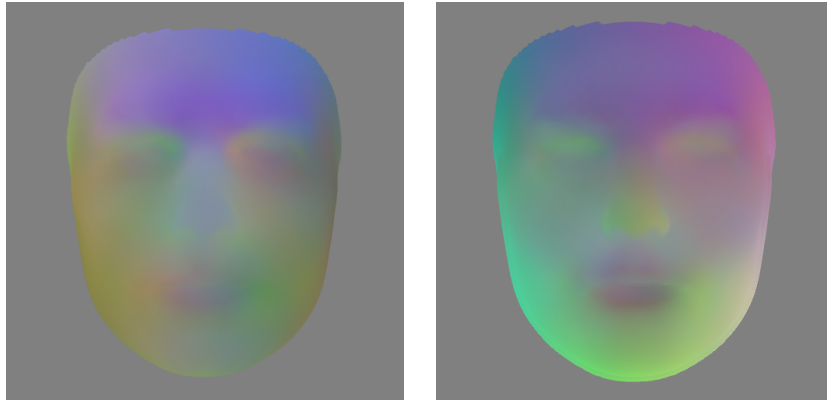


Figure 1.1: The left and right face images are from different faces, and suppose method A estimates feature points for both left and right faces so that we can match the correspondence points represented by green lines.



## CHAPTER II

### BACKGROUND

#### 2.1 Vector Space, Coordinate and the Transformation between Them

Depending on the purpose, we can represent a point in different spaces and coordinates in computer graphics. Cartesian is the most common and intuitive coordinate using three orthogonal axes as the base vectors, representing the three dimensions. Additionally, in Cartesian coordinates, we utilize the world, screen, model, and viewport space to solve the different problems.

Additionally, Barycentric is another coordinate we can use to solve a set of problems respected to a triangle, for example, point in triangle test and shading colours in a triangle.

Since the transformation between spaces and coordinates are essential for this research, we will discuss the basic concepts of each required space and coordinate and the conversion between them.

We will discuss only the essential topics for this research in this section. We use a book from Gortler (2012) as the foundation.

##### 2.1.1 World, Model, Camera, and Screen, Space

Before advancing, we can compare three-dimensional vector space with a digital camera. We use digital cameras to capture our world scenery into a two-dimensional plane. We can represent the world scenery using the Cartesian coordinate  $x$ ,  $y$  and  $z$  axes for the technical explanation. On the other hand, we can represent the image taken by digital cameras using only  $x$  and  $y$  axes.

In the same way, computer graphics simulate digital cameras with the simplest model called the pinhole camera model [cite] shown in figure [pinhole]. The physically-corrected pinhole camera has the film plane behind the small aperture, which produces the flipped image. Thus, we can reduce

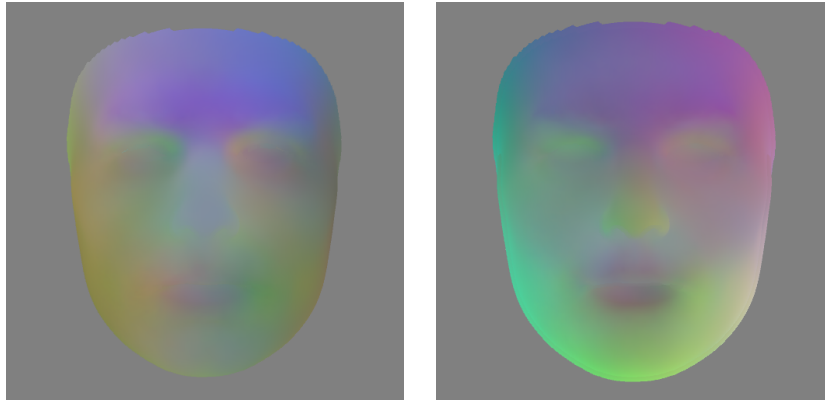


Figure 2.1: The left and right face images are from different faces, and suppose method A estimates feature points for both left and right faces so that we can match the correspondence points represented by green lines.

the complexity using mathematics by moving the plane in front of the aperture with the same distance shown in figure [frustum]. From the discussion, we can represent the output of this virtual camera model using a two-dimensional plane in x and y axes.

The virtual camera renders objects only in a definite area called camera frustum. Thus, we need a shared space to describe the relationship between the virtual camera and the rendered objects called world space.

World space is represented by Cartesian coordinates with a 3-dimensional axis using x, y and z. All objects in the rendering scene and the camera required the world space's position.

For example, in the world space given the camera position at  $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$  for x, y, z respectively. The camera facing toward z+, and the object position at  $\begin{bmatrix} 0 & 0 & 8 \end{bmatrix}$ . We will render the object at  $\begin{bmatrix} 0 & 0 & 7 \end{bmatrix}$  at the center of the the camera. However, the object in the 3D scene doesn't have only one point represented by the group of points called vertices connected in triangulation topology. All vertices of a single object have their local space called model space.

For one more example, suppose we need to paint colours on the face model to specify facial segments. We must first locate the model in front of the camera at the appropriate distance because the distance between the object and camera relates to the rendered object size. We can avoid this complicated by using an orthographic camera.

The pinhole camera model and camera frustum have one property called field of view (FoV). The greater FoV means you have a wider camera, but the object looks smaller than the smaller FoV in screen space. On the other hand, orthographic cameras, in a mathematical model, have FoV at 0, which means the object's size on screen space is the same regardless of the distance between the camera and the object in world space. As a result, we require the object to be in front of the camera irrespective of the distance.

Then we can calculate each vertex position on-screen space in two steps. First, convert the vertex point from model space to world space. And second, convert it from the world space to screen space.

Lastly, we can specify the vertex colour based on a facial segment on-screen space using painting tools.

### 2.1.2 Cartesian and Barycentric Coordinate

Our research uses the Barycentric coordinate for two reasons; First, point in triangle testing. Second, change the point's position proportionately when moving triangle points. However, we will discuss the basic concept of Barycentric coordinate in the first step.

Consider a triangle has three vertices at  $p_0$ ,  $p_1$  and  $p_2$ , and  $p$  is a point inside the given triangle. Barycentric coordinate represents  $p_0$  with value  $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$ ,  $p_1$  with value  $\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$  and  $p_2$  with value  $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$ . For  $p$ , we can describe using equation 2.1.

$$\begin{aligned}
 a &= [(p_1 - p_0) \times (p_2 - p_0)] * 0.5 \\
 u &= [(p_1 - p) \times (p_2 - p)] * 0.5/a \\
 v &= [(p_0 - p) \times (p_2 - p)] * 0.5/a \\
 w &= [(p_0 - p) \times (p_1 - p)] * 0.5/a
 \end{aligned}
 \tag{2.1}$$

The equation 2.1 expresses the position of  $p$  on Barycentric coordinate. Points  $u$ ,  $v$  and  $w$ , indicate the proportions of normalized areas in the triangle divided by point  $p$ . Additionally, Barycentric

coordinate is applicable to test a point in a triangle using  $u$ ,  $v$  and  $w$ . We have two ways to test; First,  $u$ ,  $v$ , and  $w$  must be greater than or equal to 0 and less than or equal to 1. Second,  $u + v + w$  must be equal to 1.

## 2.2 Point Correspondence

To simulate facial cosmetic surgery with a 3D face, we need to know the semantic of any point on the face, called point correspondence, whereas the same correspondence points between two 3D faces refer to the same semantic. For more specific, we can also describe correspondence points by equation 2.2.

$$(\exists p \in F_i)[\exists q \in F_j][F_i \rightarrow F_j](i \neq j) \quad (2.2)$$

Given  $F_i$  and  $F_j$  represent different faces,  $E_p$  and  $E_q$  are feature points on  $F_i$  and  $F_j$ , respectively,  $p$  and  $q$  are corresponding points on the 3D models. Suitable correspondence point methods should be accurate and covered throughout the 3D face. Based on density, point correspondence can be classified into sparse and dense.

Sparse point correspondence methods estimate the points partially from the 3D mesh. Kazemi and Sullivan (2014) and K3DM Gilani et al. (2018) are examples of sparse points estimation methods shown in figure 2.2.

On the other hand, Dense point correspondence estimates the points by covering all the areas of the 3D face. Deng et al. (2019), White et al. (2019) and Feng et al. (2018) are examples of dense points estimation methods shown in figure 2.3.

Moreover, dense correspondence estimation methods based on vertex indices enable us to convert to other formats, such as an image pixel representing depth since vertices in a 3D mesh can produce triangle faces. Then we can interpolate the pixel colour using Barycentric coordinate. However, using modern GPU and rendering pipeline, we can render the depth image using vertex colour shader. We will discuss briefly in rendering pipeline and vertex colour shader in the next

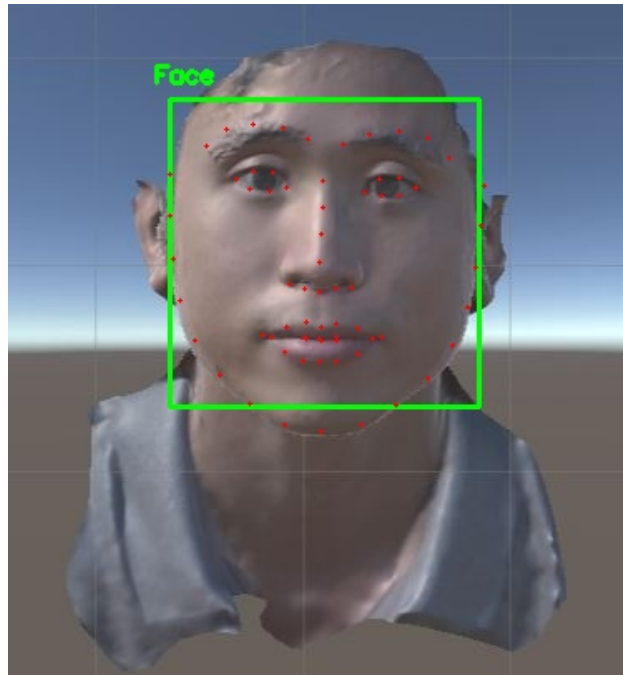


Figure 2.2: The red points represent sparse-corresponding points estimated using the method proposed from Kazemi and Sullivan (2014).

section 2.3.

### 2.3 Graphics Rendering Pipeline and Vertex Color Shader

We construct a single model from vertices in model space, convert it to the camera's space, and then render it onto the screen. We discussed this in section 2.1.1.

Accompany with the section refointCorrespondence we can convert corresponding points from 3D vertices to a depth image. Conversion to depth images enables us to utilize the convolutional neural networks (CNN) for solving image classification or similar image query problems. Even though we have the existing method to process 3D face as the input proposed by Ranjan et al. (2018), but still inconvenient and might take more processing time than the 2D image-based CNN models in the current days.

We can manually generate pixel-by-pixel depth images from 3D models given the colour attached to each vertex by using Barycentric coordinates. Nevertheless, using hardware rendering is the better option for many reasons. First, it is faster to use parallelization on Graphics Processing Unit (GPU), and second, it is convenient to use the shader programs. GPU gives the rendering speed on



Figure 2.3: Three samples of 3D face mesh estimated the correspondence points using White et al. (2019). Each same vertex colour on the different face meshes represents the same segment. For example, the orange colour represents the nose segment.

parallelizable tasks using the graphics rendering pipeline, also known simply as "The pipeline".

Following the book written by Akenine-Mller et al. (2018) in chapter 2, we can represent the basics of the pipeline consisting of four stages, application, geometry processing, rasterization, and pixel processing shown in figure 2.4. This section will discuss the pipeline focusing on face mesh and vertex colour shader.



Figure 2.4: The graphics rendering pipeline contains four stages; application, geometry processing, rasterization, pixel processing.

The application stage contains application logic running on Central Processing Unit (CPU). This application stage might be complex in some genres, such as video games, game logic, collision detection, etc. However, our application is placing the still face mesh in front of the camera shown in figure 2.5. The output of the application stage is rendering primitives, for example, points, lines, and triangles. In our case, we output the vertices' positions. We also attach colour in RGB for each vertex.

The geometry Processing stage computes per-triangle and per-vertex operations. So, this stage executes our vertex shader program. This stage can also compute with lighting information, but our purpose is to output the depth data, so we need to make sure we do not include lighting in our shader. Geometry Processing contains four sub-stage shown in figure 2.6. Briefly, Vertex shader outputs vertices colour with positioned in camera's space. Projection and clipping take the primitives from vertex shading, clip to process only the primitives inside the camera and outputs the positions



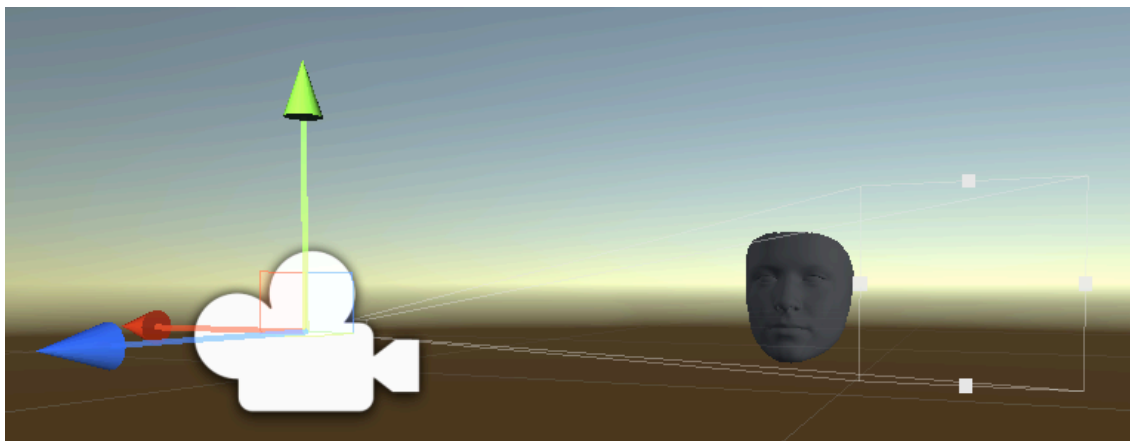


Figure 2.5: The example application stage contains a camera on the left of the image and still face mesh on the right.

of the triangles in 3D normalized device coordinates, which the detail in Akenine-Mller et al. (2018). Lastly, the Screen Mapping stage process the clipped primitives and output colours in the screen coordinate with the z-coordinates.



Figure 2.6: Geometric processing contains four sub-stages; vertex shading, projection, clipping and screen mapping.

Rasterization and Pixel Processing Stages process the primitives in screen coordinate and output the pixel colours on the screen. Rasterization also interpolates the pixel colours given the vertex colours with pixel shader, also called fragment shader. In our problem, the pixel shader input the interpolated colour from the previous stages and bypass the received colour in the following stages. The rasterization and pixel processing pipeline is shown in figure 2.7



Figure 2.7: Rasterization substage contains triangle setup and triangle traversal. Pixel processing substage contains pixel shading and merging.

Additionally, we also discuss our shader program source code using Cg program in Unity Game Engine.

```

1 Shader "Custom/VColorOpaque" {
2     Category {
3     Tags { "RenderType"="Opaque" }
4     Lighting Off
5     SubShader { Pass {
6         CGPROGRAM
7         #pragma vertex vert
8         #pragma fragment frag
9         #include "UnityCG.cginc"
10        struct appdata_t {
11            float4 vertex : POSITION;
12            fixed4 color : COLOR;
13        };
14        struct v2f {
15            float4 vertex : SV_POSITION;
16            fixed4 color : COLOR;
17        };
18        // Vertex Color Shader
19        v2f vert (appdata_t v) {
20            v2f o;
21            o.vertex = UnityObjectToClipPos(v.vertex);
22            o.color = v.color;
23            return o;
24        }
25        // Pixel Color Shader
26        fixed4 frag (v2f i) : SV_Target {
27            return i.color;
28        }
29        ENDCG
30    }}
31 }
32 }

```

Line 3: States that the opaque object

Line 4: Defines that lights settled in the application scene do not affect the surface colour

Line 7: Executes the vertex shader program with the function named vert

Line 8: Executes the fragment shader program with the function named frag

Line 10: appdata\_t is the struct receiving from the application scene containing vertex position and colour

Line 14: v2f is the struct passing from vertex shader program to fragment shader program containing the vertex in clip space and colour

Line 19: The vertex shader program converts the vertex position to clip space and redirects the vertex colour to the fragment shader program.

Line 26: The fragment shader program (Pixel shader) outputs the vertex colour received from the rasterization substage.



Procedure	Definition
Botulinum toxin injections	Injections of botulinum toxin weaken the target muscle. The final result is a reduction of facial wrinkles and also reduce the muscular size. Botox is one of the most known types which categorize in type A.
Dermal fillers	Dermal fillers can restore volume and fullness in the face. Injecting it can diminish facial lines to make the look-younger face, and can change the face shape as well.
Chemical peel	Chemical peel improves the texture of the face's skin by reducing sun exposure, acne, or just aging can leave texture uneven, wrinkled, spotted, or scarred.
Microdermabrasion	Microdermabrasion can improve the texture of the face's skin similarly with Chemical peel. The procedure is to remove the outer layer of the skin to allow newer skin to grow and replace the older skin.

จุฬาลงกรณ์มหาวิทยาลัย  
 CHULALONGKORN UNIVERSITY  
 Table 2.1: Four Minimally-invasive procedures, include definitions

## CHAPTER III

### RELATED WORKS

Our work focuses on constructing a fully automatic pipeline for facial cosmetic surgery simulation. We review other simulation methods with their advantages and disadvantages. We also review related methods we use in the pipeline in this section.

#### 3.1 Cosmetic Surgery Simulation Methods

Post-surgery visualization can reduce misunderstanding during the consultation process. Paper from Verhulst et al. (2018) compares the three 3D face scanning systems' accuracy and performance: Vectra XT, Artec Eva, and 3dMDface. Paper from Bottino et al. (2012) shows the post-surgery image for the planning stage.

#### 3.2 Beautiful Faces Definition and Surgical Procedures

Neoclassical canon and the golden facial ratio are two beauty measurement methods based on facial proportions. The research from Burusapat and Lekdaeng (2019) measures the proportions of Miss Universe and Miss Universe Thailand as the beautiful facial group. Another research group discuss the beautiful score based on attractiveness Scheib et al. (1999) Perrett et al. (1998) Little et al. (2011) Fink and Penton-Voak (2002) Little (2014) Perrett et al. (1999). Facial attractiveness research group commonly discussing based on subjective surveys. The mentioned facial proportions rely on dominant points mentioned in the table 3.1.

Focusing on the surgical procedures, the research from Nappi et al. (2016) includes all procedures and the affected segment of the face for each procedure. The study of all procedures can specify which area is affected by which procedure and how. Note that surgical procedures cannot change some properties of certain regions. For example, we can change the face ratio by applying chin augmentation and cheekbones reshaping, but we cannot shift the position of the lip and eyes up or down vertically.

Additionally, we can classify the surgery procedures into two groups from the research proposed by Nappi et al. (2016). First, the minimally-invasive procedures modify the specific area temporarily shown in table 3.2. Two well-known procedures are Botulinum toxin which can reduce facial wrinkles, and Dermal fillers, which can restore volume. The surgery procedures in this group are commonly absent for months or years.

Second, the real surgical procedures modify the face appearance permanently shown in table 3.3. Rhinoplasty, or nose reshaping, is the highest procedure in this group. However, we can restore some procedures in this group using surgical procedures.

### 3.3 Face Registration in 2D and 3D

2D Facial landmarks proposed by Kazemi and Sullivan (2014) with 194 correspondence points around facial contours and face contents using a regression tree. 2D and 3D facial landmarks methods proposed by Bulat and Tzimiropoulos (2017) using a deep learning model. The output of 3D models includes the correspondence points of the facial landmarks and a 3D face reconstruction. For the non-model methods, K3DM Gilani et al. (2018) registers the 3D face using a geometric feature matching algorithm. A Morphable for the Synthesis of 3D Faces (3DMM) Blanz and Vetter (1999), and Morphable Face Models-An Open Framework Gerig et al. (2018) embed face models in the dataset to single vector space. Physically-based methods are another technique to register the 3D faces. MeshMonk White et al. (2019) registers rigidly by force between nearest-neighbor vertex pairs and registers non-rigidly using viscoelasticity force. The concept of non-rigid and rigid registrations are described in Bro-Nielsen (1996).

To simulate facial cosmetic surgery with a 3D or 2D face, we need to know the semantics of any point on the face, called point correspondence. Where the same correspondence point between two 3D faces refers to the same semantic location Berretti et al. (2013). Sparse facial landmarks output the semantical points, whereas dense face alignment Liu et al. (2017) outputs the connected lines or spaces.

### 3.4 Similarity Models for Face Image Query

Image query utilizes deep learning to find the most similar image in the dataset. The perceptual distance function and cosine similarity function can measure the similarity between two images Li et al. (2003) Neumann and Gegenfurtner (2006). Deep learning models for the large dataset search using autoencoder Portenier et al. (2018) Siamese network Wiggers et al. (2019), and triplet loss Wang et al. (2014) give better results over using loss functions for the images with variance scale, rotation, including occlusion of the contents. However, deep learning models for similarity search require distance comparison in the n-dimension of the abstraction space. FAISS Johnson et al. (2019) proposes the methods for the billion-scale image search.

### 3.5 Facial Retargeting

Facial retargeting shows the good and fun results in digital avatar creating and facial expressions Chaudhuri et al. (2019) Zhang et al. (2020) Costigan et al. (2014) Ouzounis et al. (2017). We show how to apply retargeting methods in cosmetic surgery domain in this work.



Abbreviation	Definition
Vertex (v)	The highest seen point on the head.
Trichion (tr)	The junction of hairline and forehead in the midline.
Glabella (g)	The most prominent point of the forehead in the midline between the eyebrows.
Temporal (t)	The most prominent point of the forehead in the midline between the eyebrows.
Medial eyebrow (bm)	Medial border of the eyebrow (left and right).
Lateral eyebrow (bl)	Lateral border of the eyebrow (left and right).
Peak of the eyebrow (bp)	Lateral border of the eyebrow (left and right).
Nasion (n)	Point of greatest concavity of the nasal dorsum near a line level with the upper lid lash line.
Endocanthion (en)	Point of greatest concavity of the nasal dorsum near a line level with the upper lid lash line.
Exocanthion (ex)	Point of the lateral canthus where the upper and lower lids join (left and right).
Palpebrae superius (ps)	Superior border of the palpebrae (left and right).
Palpebrae inferius (pi)	Inferior border of the palpebrae (left and right).
Center of pupil (p)	The center point of the pupil.
Zygion (zy)	Lateral border of the cheek (left and right).
Pronasale (prn)	The highest point of the tip of the nose.
Ala (al)	The most lateral point on the rims of the alar wing of the nose (left and right).
Subnasale (sn)	The deepest point at the junction of the base of the columella and upper lip in the midline.
Cupid's peak (cp)	The peak of Cupid's bow.
Labial superius (ls)	The upper border of the upper lip.
Chelion (ch)	Oral commissure.
Labial inferius (li)	The lower border of the lower lip.
Stomion (sto)	The midline point where the upper lip touches the lower lip.
Sublabiale (sl)	The midline point at the junction of the lower border of the cutaneous lower lip and superior border of the chin, it is the deepest point of the labiomental groove.
Mandibular angle (ang)	The angle of the chin according to the line from the oral commissure.
Pogonion (pg)	The most prominent point of the chin in the midline.
Gnathion (gn)	The most inferior point of the lower border of the chin.

Table 3.1: Twenty-six dominant points, include abbreviation and definition.



Procedure	Definition
Botulinum toxin injections	Injections of botulinum toxin weaken the target muscle. The final result is a reduction of facial wrinkles and also reduce the muscular size. Botox is one of the most known types which categorize in type A.
Dermal fillers	Dermal fillers can restore volume and fullness in the face. Injecting it can diminish facial lines to make the look-younger face, and can change the face shape as well.
Chemical peel	Chemical peel improves the texture of the face's skin by reducing sun exposure, acne, or just aging can leave texture uneven, wrinkled, spotted, or scarred.
Microdermabrasion	Microdermabrasion can improve the texture of the face's skin similarly with Chemical peel. The procedure is to remove the outer layer of the skin to allow newer skin to grow and replace the older skin.

Table 3.2: Four Minimally-invasive procedures, include deinitions

Procedure	Definition
Nose reshaping (rhinoplasty)	Nose reshaping can enhance the proportions and facial harmony by adjusting the nose area.
Eyelid surgery (blepharoplasty)	Eyelid surgery improves the aspect of the eye region, including the eye's shape, either the upper or lower lids.
Facelift (rhytidectomy)	Lifting face and neck can improve visible signs of aging.
Brow lift (forehead lift)	The surgical procedure to reduce the horizontal wrinkle lines across the forehead and on the bridge of the nose between the eyes
Chin surgery (mentoplasty)	Either by enhancing with an implant or reducing on the bone, chin surgery can change the facial proportions. A surgeon may also recommend chin surgery to a patient having rhinoplasty.

Table 3.3: Five real procedures, include deinitions

## CHAPTER IV

### METHODS

We propose a fully 3D post-surgery simulation pipeline by retargeting a single image and only 400 pre-and-post 2D image pairs in the face similarity search process. Our two registration methods support both 3D meshes and 2D images as input. For 3D mesh, we proceed with rigid and non-rigid physically-based registration White et al. (2019) and improve the accuracy using facial landmarks model Kazemi and Sullivan (2014). For a 2D image input, we reconstruct a 3D mesh using Deng et al. (2019). Both 3D and 2D registration methods produce a 3D mesh containing vertices in the same semantically order.

Next, we align the registered 3D mesh rigidly using a gradient descent algorithm with mean squared error (MSE) of the selected vertices between the registered and mean mesh. We store the aligned face data in an RGB image with 512 x 512 pixels called "Delta Image". We also use a delta image to store the distance between pre and post-surgery mesh of the same face.

Then we select the most similar face using a convolutional autoencoder trained from delta images between pre-and-mean faces. Finally, we retarget surgical procedures using the delta image between the pre-to-post face of the same person to simulate the post-surgery result. The final result can be a 2D image or 3D mesh with texture.

The last section will discuss the verification by a subjective survey from the focus group to measure the results. Additionally, we also discuss the tools and resources we use throughout the pipeline.

Our pipeline consists of delta images generated from either 3D or 2D registration and alignment methods, similar face selection using image autoencoder, and retargeting procedures. We introduce the delta image as a significant part of similar face selection and retargeting methods.

We use the dense correspondence point of image pixels to represent the geometric features.

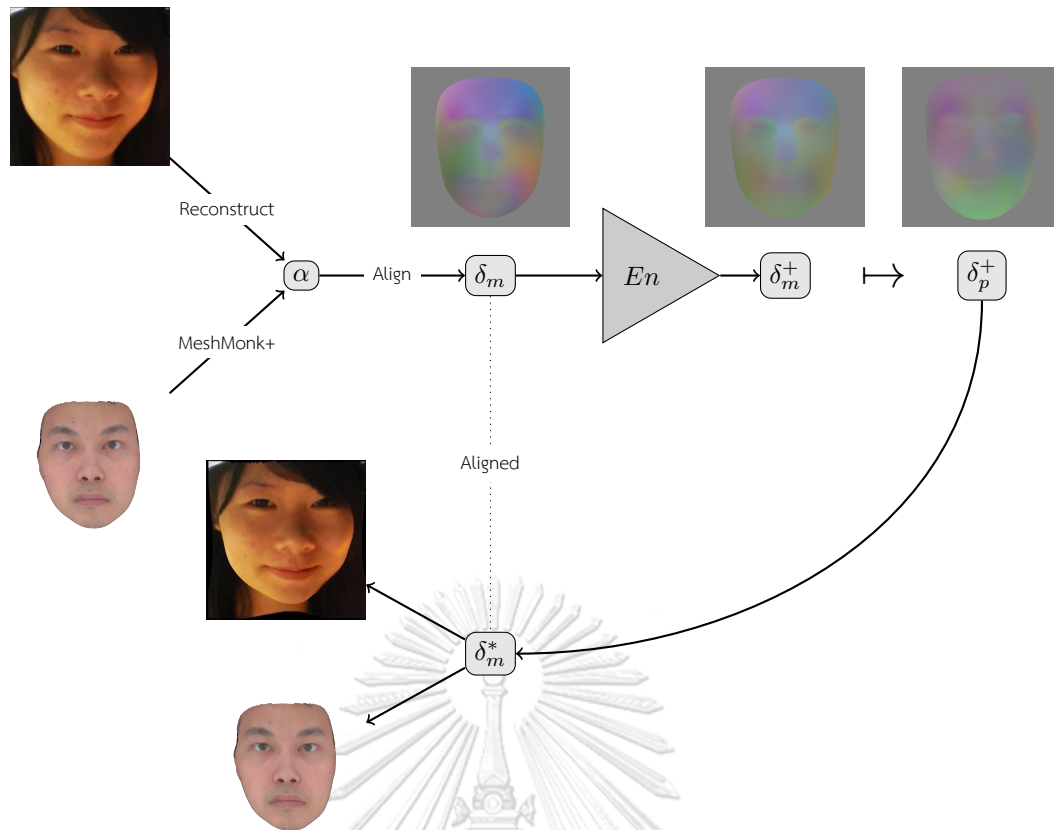


Figure 4.1: DeltaNet pipeline is an end-to-end method.  $\alpha$  represents the 3D mesh in semantical order registered from the 3D mesh and 2D image. Then we proceed rigidly alignment ( $\delta_m$ ).  $E_n$  represents the encoder model to get the most similar face in the dataset ( $\delta_m^+$ ), map to the surgery procedures image ( $\delta_p^+$ ). And finally, we obtain the retargeted image ( $\delta_m^*$ ) described by  $\delta_m^* = \delta_m + \delta_p^*$

In a similar face selection method, we propose a convolutional autoencoder with 400 mean-to-pre delta images. At the final step, we propose a single image retargeting procedure using a pre-to-post delta image.

Another advantage of using the delta image is that it supports 3D mesh and 2D images as the input. The registration stage for 3D mesh and 2D image required different methods. For 3D mesh, we propose physically-based face registration using MeshMonk White et al. (2019), then we improve the accuracy using the face landmarks model Kazemi and Sullivan (2014). Our proposed 3D face registration methods support 3D mesh with different polygon topology, Hence, can be used with any 3D scanning system. For 2D face images, we reconstruct a 3D face using a deep learning model Deng et al. (2019) to obtain the geometric features. Lastly, we propose an alignment procedure using a rigidly 3D face matching method in the gradient descent by selecting face areas that are not affected by any surgical procedures of the current day.

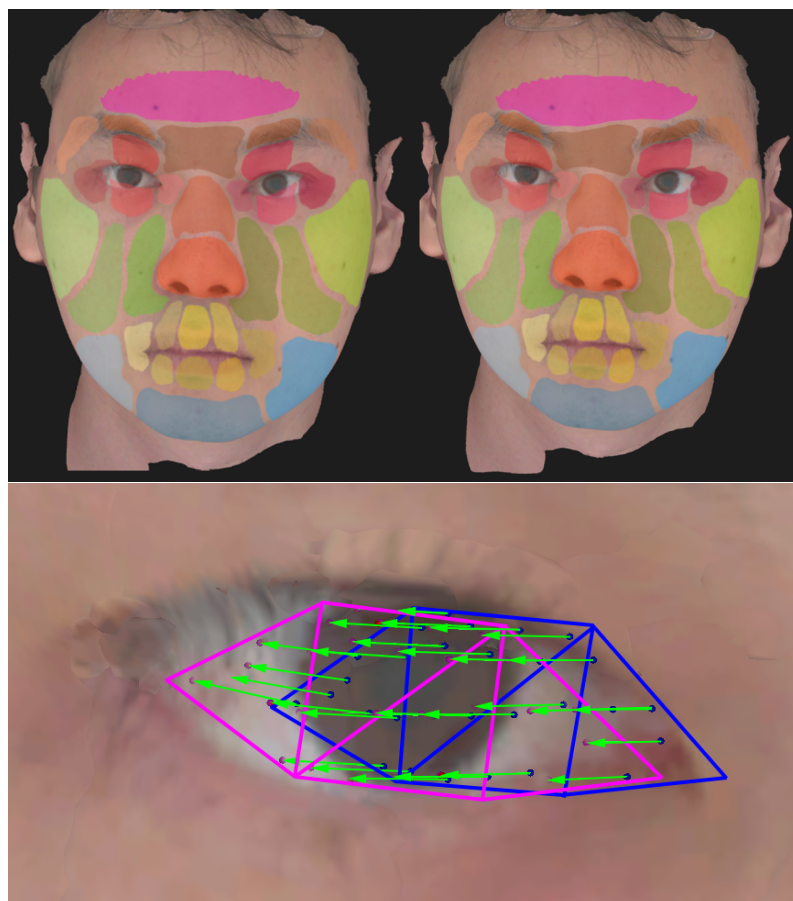


Figure 4.2: The top image shows the improvement correspondence points estimation in the eyes area. The bottom image shows the vector field altering by facial landmarks.

#### 4.1 2D and 3D Registration

We propose a registration and alignment method for 3D meshes and 2D images that output the same structure.

##### 4.1.1 2D Registration

The 2D face can refer to an RGB face image that comes with only optical features. However, the facial image is the only existing choice to obtain the pre-post surgery image pairs as the dataset. Additionally, the open dataset from the facial images is relatively small compared with other deep learning models. Consequently, utilizing a 3D reconstruction method from a facial image Deng et al. (2019) is a good way to estimate geometric information.

#### 4.1.2 3D Registration

Facial registration from geometric features can accurately capture the large area of the face, such as the nose, chin or the overall face shape, but struggles with capturing eyeballs, lids, and eyebrow area because of the tiny noticeable geometry differences. On the other hand, the optical features that consider light intensity perform better in some small areas. We propose the 3D mesh registration method by combining the two approaches.

Additionally, geometric based registration requires 3D mesh as the input. 3D meshes are commonly from the 3D face reconstruction models Zhang et al. (2015). Optical based registration from 3D mesh requires texture and rendering with natural light settings.

MeshMonk White et al. (2019) proposed 3D correspondence points estimation base on geometric features using rigid and non-rigid matching. The facial landmarks model Kazemi and Sullivan (2014) utilized the optical features and outputs 64 semantic points on the face contour and facial contents. We propose the methods to improve the accuracy in the eyes area by triangulating facial landmarks, correspondence points retargeting, and lastly, replacing specific vectors in the vector fields during MeshMonk's matching process.

We first set up the 3D scene with an orthographic camera toward the labelled face mesh and unlit the albedo texture shader. We found that face landmarks captured from screen space are plausible. Hence, we use screen space to manipulate the vertices.

On the screen space, we triangulate from the landmarks on the eye contour shown by the blue triangles in figure 4.2. Then we test whether vertices in each triangle by using the equation to convert from Cartesian to Barycentric coordinates. Depicted by the equation 4.1, any vertices which have  $u + v + w$  less than 1 are in the triangle.

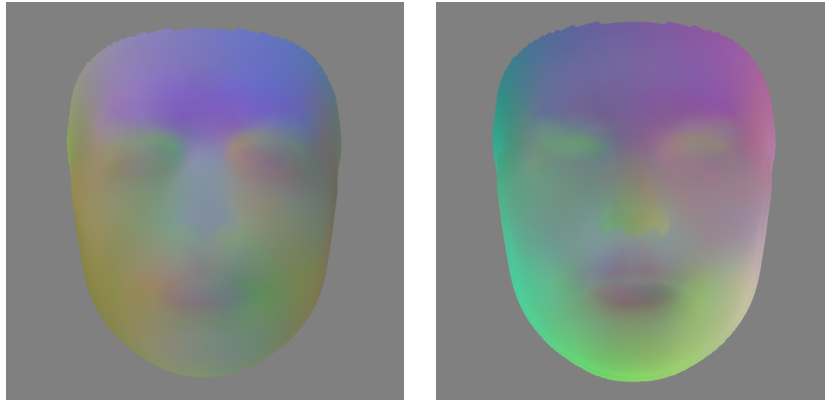


Figure 4.3: Delta Image of Pre Surgery Face (Left) and Surgery Procedures (Right)

$$\begin{aligned}
 a &= [(p_1 - p_0) \times (p_2 - p_0)] * 0.5 \\
 u &= [(p_1 - p) \times (p_2 - p)] * 0.5/a \\
 v &= [(p_0 - p) \times (p_2 - p)] * 0.5/a \\
 w &= [(p_0 - p) \times (p_1 - p)] * 0.5/a
 \end{aligned} \tag{4.1}$$

Where:

$p_0, p_1, p_2$  = the triangle vertices

$p$  = a point on the same space

$a$  = triangle area

$u, v, w$  = area of sub triangles divided by point  $p$

Equation 4.1 also describe the positioning of the vertex with respect to a triangle. In the retargeting step, we triangulate from the landmarks and replace the vertices position in the triangles from the labelled face using the equation to convert from Barycentric coordinate to Cartesian coordinate represented by  $p = up_0 + vp_1 + wp_2$ .

Lastly, the replacing vertices position can change the vector field in MeshMonk's 3D matching method. The figure 4.2 shows the improvement result around the eyes area.

## 4.2 Face Alignment and Delta Image

Both 3D and 2D face registration methods output the correspondence points represented by vertices. However, the meshes are slightly different in position, scale and rotation. We propose a rigid alignment method to minimize the variation. The surgery procedures from Nappi et al. (2016) show the affected areas. We noticed that the pupil of the eyes, the centre of the upper and lower lip, and the philtrum are neglectable affected by any procedures. Hence, we selected the vertices on such areas as reference points. We propose rigidly alignment to mean face by minimizing the error by functions from reference points between each correspondence point pairs in the gradient descent algorithm.

We perform three operations in each iteration:

1. Minimize the positioning error between target and mean face using mean square error (MSE).
2. Minimize the scaling error by creating a centroid of all referenced vertices and measure the error using MSE of the distance between the centroid to each vertex compared with mean face.
3. Minimize the rotational error by creating a centroid of all referenced vertices, then get Quaternion angle between the centroid and each vertex. Lastly, use quaternion interpolation (SLERP) approaching the Quaternion angle of each mean face vertex.

All error functions are multiplied by the learning rate ( $lr$ ) in each iteration.

Face registration and alignment methods output the correspondence points by the vertex indices on the 3D mesh to represent the semantical points. We have two reasons to store the corresponding data in the image; First, we can generate the connected space by interpolating the adjacent points. Second, we can utilize a convolutional neural network (CNN) to get the similarity. Moreover, the pixel-to-pixel similarity functions, MSE or cosine similarity, are another option for getting similar faces.

The alignment process output the euclidean distance of the same feature between the two meshes. We aim to store the same feature in the same pixel for any face. Hence, we store the

euclidean distance of features as vertex colour of the mean face mesh. Lastly, we render the output image using an unlit vertex colour shading technique to interpolate the colour between the triangle's vertices.

Delta image store three-dimensional Euclidean distance of each feature point represented by RGB colour channels where R represents  $\delta_x$ , G represents  $\delta_y$ , and B represents  $\delta_z$

The limitation of storing RGB images is that we are limited to representing between 0 and 255. So we need a linear conversion function shown by equation 4.2.

$$c = \left( \frac{\delta}{2 * \delta_{max}} \right) + 0.5 \quad (4.2)$$

$c$  = colour value for each channel

$\delta$  = distance between two face mesh

$\delta_{max}$  = a considered max value for the distance

From the equation 4.2, we can use a colour channel for the negative distance represented by the value less than 128. In contrast, a value greater than 128 represents a positive distance. Note that we set  $\delta_{max}$  to 1.58 centimetres.

### 4.3 Face Similarity Search

Figure 5.3 shows the delta image, whereas each pixel represents a feature. The exact pixel position from one face is the same feature as another face. Hence, the delta image enables us to estimate the most similar face by two approaches. In one approach, we can compare pixel-by-pixel using perceptual loss function or cosine similarity. In another approach, we can achieve the same task using a convolutional autoencoder.

We propose a convolutional autoencoder for the similarity model considering computation efficiency. Given the problem, the pixel-by-pixel comparison methods require computing all pixels multiples by all images in the dataset for a single test. In contrast, the convolutional autoencoder



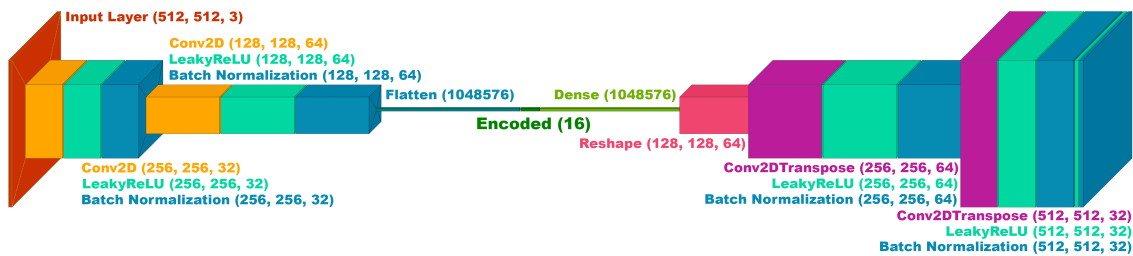


Figure 4.4: Convolutional autoencoder network architecture for the similarity search shown with hyperparameters. We utilized the encoder with 16 variables shown in the decoded layer.

estimates the position in the n-dimension of latent space represented by n-floating point numbers. Then, we can compute to get the most similar image by finding the nearest image in the latent space.

We train the network with the architecture shown in figure 4.4, training with 400 images using 200 epochs and 1e-6 as the learning rate.

#### 4.4 Surgical Procedures Retargeting

Once we have the surgery procedures information from the most similar face ( $\delta_p^+$ ), we can add it with the aligned face image ( $\delta_m$ ). The delta image of the post-surgery face ( $\delta_m^*$ ) computed with equation 4.3

$$\delta_m^* = \delta_m + \delta_p^+, \quad (4.3)$$

Specifically, the surgery procedures information represented by  $\delta_p^+$  requires conversion to the aligned space with respect to the mean face. From the aligned space shown in equation 4.2 we need to reverse the equation from c to  $\delta$  4.4.

$$\delta = (c - 0.5) * 2 / \delta_{max} \quad (4.4)$$

Our retargeting method can support both 3D meshes and 2D images. For 3D meshes, we can render the end result directly after modifying the vertices using data from a delta image. In contrast, 2D images require an additional process by overlaying the original image using D mesh. After we transform

the mesh to mean face space and retarget the surgery procedure, we need to invert, transform and render the result. 3D mesh overlaying is shown in figure 4.5

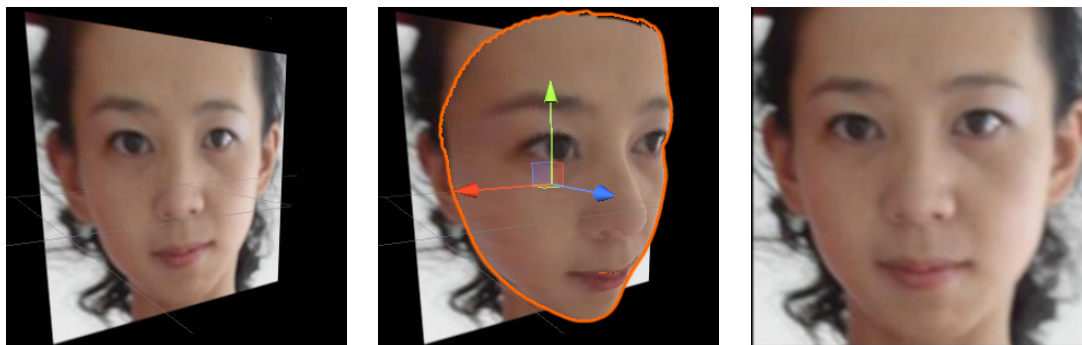


Figure 4.5: We need an additional process by overlaying the original image using D mesh. The input image (Left). The input image with overlaying mesh (Middle). The rendered result (Right).



## CHAPTER V

### RESULT

The figure 5.1 shows sample results using our proposed method. However, due to the extreme lack of pre-post surgery images, we cannot get likely results from some face images that might not be similar to the faces in the dataset. The section 5.3 explains the details of our limitations. Additionally, we do not properly verify the similarities from the face similarity search model because similarity measurement can be another big topic. Instead, we measure our end-to-end results using subjective surveys.



Figure 5.1: Our DeltaNet pipeline simulates the post-surgery image in 2D and 3D (bottom) without any manual settings by giving an image or 3D mesh as the input (top). The images were not used during training.

#### 5.1 Face Similarity Search

Our delta images eliminate the position, scale, and rotation using 3D alignment method. Additionally, we selected only the rest-face images in the dataset. As a result, our delta images represent a bijective mapping between the image pixels and geometric features of the face. Moreover, the same image pixel of the different faces represents the same feature. We train our model with a small dataset, 381 for the training set and 96 for the validation set. The error values during the training by figure 5.2 show the generalization between the training set and validation set.

However, generalization can become an issue because the dataset is relatively small. We will

discuss this issue more in the robustness section.

## 5.2 Attractiveness-Based Subjective Survey

To evaluate the simulated results, we conducted the subjectivity using ten questions based on the attractiveness attributes Scheib et al. (1999)Perrett et al. (1998)Little et al. (2011)Fink and Penton-Voak (2002)Little (2014)Perrett et al. (1999) with 24 respondents. The survey shows ten pre-post surgery face image pairs divided into simulated and controlled groups, with five pairs for each group. The simulated sample shows an actual human face image for the pre-surgery side and the simulated image from our pipeline for the post-surgery side. On the contrary, the controlled sample shows the actual human face for both pre-and-post surgery sides.

As the mechanism of our pipeline, select the most similar face and retarget the surgery procedures from the selected image pair. Hence, a simulated sample is mapped from a single controlled sample, whereas a controlled sample can map to many simulated samples. Then we analyze by comparing controlled and simulated pairs using a 3x3 confusion matrix shown in table 5.1. Each matrix represents all controlled-simulated pairs for the individual attribute.

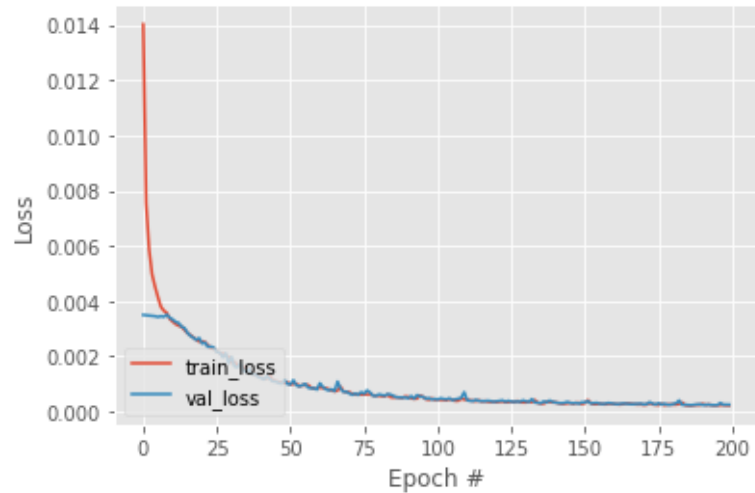
The matrices show that our model performs well in the face shape attribute as the majority selected "The post-surgery face look smaller" for both controlled and simulated groups. The matrices representing positive results for symmetry and face proportion attributes can imply that our simulated result looks plausibly realistic. The majority selected "The overall beauty of the post-surgery is better" or "No different" for both controlled and simulated groups. In the same way, the post-surgery face looks more feminist. In contrast, the controlled group looks younger, but the simulated group looks not different means our pipeline does not perform well in the ageing attribute. Lastly, our simulated sample does not perform well with nose shape, mouth shape, eyes shape and chin shape with slightly different distributions.

## 5.3 Robustness and Limitation

Our method contains information on the fully frontal face image with rest expression. Thus, we further evaluate with slightly rotated face images and with different facial expressions. Our models

produce plausible results with slightly rotated face images but show noticeable artefacts on the teeth area. Also, the model does not perform well for the face shape that is very different from those in the dataset, for example, the male face images shown in figure 5.4.





Activation Function	Epoch	Lr	Train Loss	Val Loss
LeakyReLU	200	1e-06	0.000235	0.000239
Sigmoid	200	1e-06	0.000178	0.000158
Tanh	200	1e-06	0.000199	0.000161
LeakyReLU	100	1e-05	0.000412	0.000427
LeakyReLU	400	1e-07	0.000818	0.000957

Figure 5.2: Training loss for the face similarity model demonstrate that our model can generalize from the training set to validation set (top).

The table shows loss values from the training set (Train Loss) and validation set (Val Loss) given activation functions, epochs and learning rate (Lr) as hyperparameters. We selected LeakyReLU as a popular choice for modern deep learning as it suffers less from the vanishing gradient problem compared to Sigmoid and Tanh (bottom).

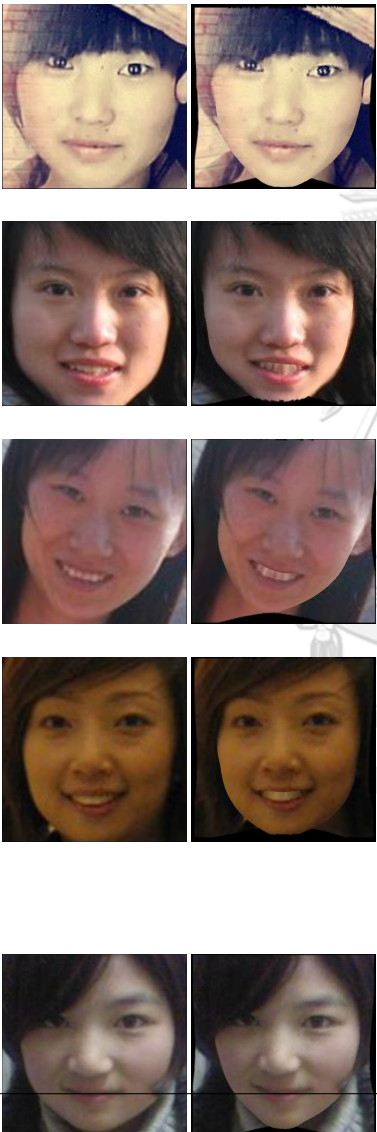
<p>Simulated samples</p>	<p>A respondent answers ten questions per sample focusing the changes on post surgery images.</p>
	<ol style="list-style-type: none"> <li>1 Is face look more slender or wider?</li> <li>2 If look more feminist or less?</li> <li>3 Is look younger or older?</li> <li>4 Is symmetry or asymmetry?</li> <li>5 Is nose look more slender or less?</li> <li>6 Is mouth look more slender or bigger?</li> <li>7 Is eyes look smaller or bigger?</li> <li>8 Is chin look taller or shorter?</li> <li>9 Is the overall face look more - proportional or less?</li> <li>10 Is the overall face look more - beautiful or less?</li> </ol>

Table 5.1: We conducted a subjective survey with ten questions based on attractiveness attributes.

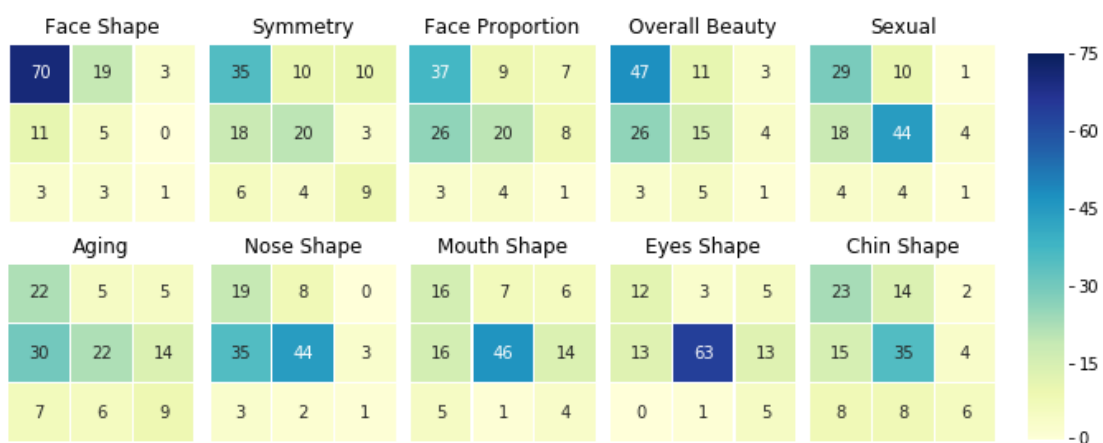


Figure 5.3: Our confusion matrix shows the overall simulation performance of our pipeline based on attractiveness attributes. The columns of each matrix represent the scores on a controlled group, whereas the rows represent the scores on a simulated group. The order relating to each question in table 5.1.



Figure 5.4: The images on the left represent the original facial images, whereas the right represents the simulated post-surgery images. Two examples show that our model does not perform well with the teeth area (Top row). Also, the model does not perform well for the face shape that lacks the dataset, for example, the male face.



## CHAPTER VI

### DISCUSSION

#### 6.1 Implementations

The entire simulation pipeline composes of five main models with different software environments. For example, at the time during research, 2D face reconstruction model requires Python 3.6 with Tensorflow 1.12, whereas the Face similarity search model requires Python 3.8 with Tensorflow 2.2.0. Moreover, the Surgical procedures retargeting renderer requires Unity3D, and the 3D face reconstruction model requires MATLAB.

Thus we design a straightforward and flexible environment that can run on a MacBook Pro 2015, shown in figure 6.1 by following steps.

Step 1: We manually paste pre-surgery images on the main Input directory at the top-left of the figure.

Step 2: The main shell script copies files to the Input directory of Model 1 and runs the shell script of Model 1.

Step 3: The separate shell script of Model 1 runs the model and paste the output files into its Output directory.

Step 4: The main shell script copies files from the Output directory of Model 1 to the main Output directory and runs the shell script of Model 2

Step 5: We repeat the process between the steps 2 and 4 until we finish the last model (Model N)

#### 6.2 Conclusions and Future Work

Our research simulates the facial surgery simulation pipeline, which requires a small dataset. We propose a 3D registration method to establish correspondence points throughout the face and align the 3D face rigidly in gradient descent. The output of the registration method is the surgical

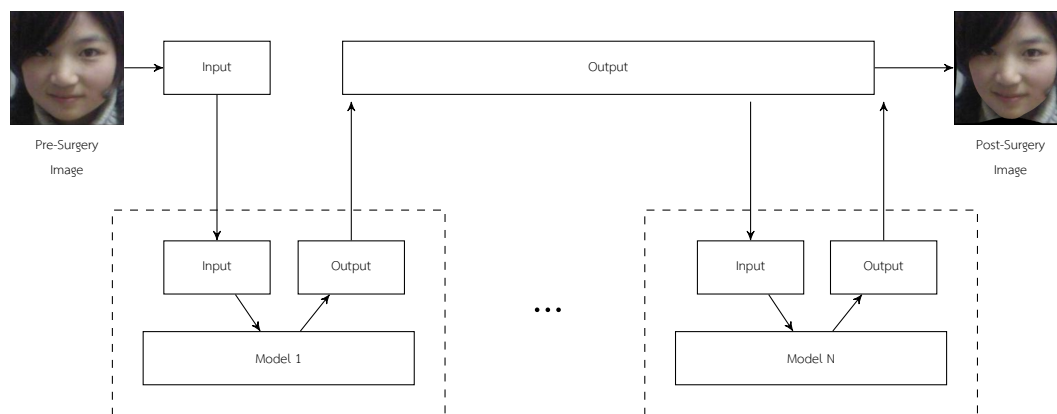


Figure 6.1: Our system's pipeline

procedures of the whole single face. We store the correspondence values in RGB images that make it convenient to extract spatial information. Then we use convolutional autoencoder to estimate the most similar face. Lastly, we retarget the surgical procedure from one face to the target face by adding from pre-to-post delta image. We also discuss our project structure to show how a prototype of the system may be implemented. We are excited to apply our system to more real person faces in the future.

## References

- Akenine-Mller, T., Haines, E., and Hoffman, N. 2018. *Real-Time Rendering, Fourth Edition*. A. K. Peters, Ltd., USA, 4th edition. ISBN 0134997832.
- Berretti, S., Werghi, N., del Bimbo, A., and Pala, P. 2013. Matching 3d face scans using interest points and local histogram descriptors. *Computers & Graphics* 37.5 (2013): 509–525.
- Blanz, V. and Vetter, T. 1999. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99*, pp. 187–194. USA: ACM Press/Addison-Wesley Publishing Co.
- Bottino, A., De Simone, M., Laurentini, A., and Sforza, C. 2012. A new 3-d tool for planning plastic surgery. *IEEE Transactions on Biomedical Engineering* 59.12 (2012): 3439–3449.
- Bro-Nielsen, M. 1996. *Medical Image Registration and Surgery Simulation*. PhD thesis.
- Bulat, A. and Tzimiropoulos, G. 2017. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). *2017 IEEE International Conference on Computer Vision (ICCV)* (2017): 1021–1030.
- Burusapat, C. and Lekdaeng, P. 2019. What is the most beautiful facial proportion in the 21st century? comparative study among miss universe, miss universe thailand, neoclassical canons, and facial golden ratios. *Plastic and Reconstructive Surgery Global Open* 7 (2019):
- Chaudhuri, B., Vesdapunt, N., and Wang, B. 2019. Joint face detection and facial motion retargeting for multiple faces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. :
- Costigan, T., Prasad, M., and McDonnell, R. 2014. Facial retargeting using neural networks. In *Proceedings of the Seventh International Conference on Motion in Games, MIG '14*, pp. 31–38. New York, NY, USA: Association for Computing Machinery.
- Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., and Tong, X. 2019. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *2019 IEEE/CVF Conference*

- on *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 285–295. :
- Feng, Y., Wu, F., Shao, X., Wang, Y., and Zhou, X. 2018. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*. :
- Fink, B. and Penton-Voak, I. 2002. Evolutionary psychology of facial attractiveness. *Current Directions in Psychological Science* 11.5 (2002): 154–158.
- Gerig, T., Forster, A., Blumer, C., Egger, B., Lüthi, M., Schönborn, S., and Vetter, T. 2018. Morphable face models - an open framework. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)* (2018): 75–82.
- Gilani, S. Z., Mian, A., Shafait, F., and Reid, I. 2018. Dense 3d face correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.7 (2018): 1584–1598.
- Gortler, S. J. 2012. *Foundations of 3D Computer Graphics*. The MIT Press. ISBN 0262017350.
- Johnson, J., Douze, M., and Jégou, H. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* (2019): 1–1.
- Kazemi, V. and Sullivan, J. 2014. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874. :
- Li, B., Chang, E., and Wu, Y. 2003. Discovery of a perceptual distance function for measuring image similarity. *Multimedia Systems* 8.6 (Apr 2003): 512–522.
- Little, A. C., Jones, B. C., and DeBruine, L. M. 2011. Facial attractiveness: evolutionary based research. *Philos Trans R Soc Lond B Biol Sci* 366.1571 (Jun 2011): 1638–1659.
- Little, A. C. 2014. Facial attractiveness. *WIREs Cognitive Science* 5.6 (2014): 621–634.
- Liu, Y., Jourabloo, A., Ren, W., and Liu, X. 2017. Dense face alignment. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 1619–1628. :
- Nappi, M., Ricciardi, S., and Tistarelli, M. 2016. Deceiving faces: When plastic surgery challenges face recognition. *Image and Vision Computing* 54 (2016): 71–82.
- Neumann, D. and Gegenfurtner, K. R. 2006. Image retrieval and perceptual similarity. *ACM Trans. Appl. Percept.* 3.1 (January 2006): 31–47.

- Ouzounis, C., Kiliyas, A., and Mousas, C. 2017. Kernel projection of latent structures regression for facial animation retargeting. In *Proceedings of the 13th Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS '17*, pp. 59–65. Goslar, DEU: Eurographics Association.
- Perrett, D. I., Lee, K. J., Penton-Voak, I., Rowland, D., Yoshikawa, S., Burt, D. M., Henzi, S. P., Castles, D. L., and Akamatsu, S. 1998. Effects of sexual dimorphism on facial attractiveness. *Nature* 394.6696 (Aug 1998): 884–887.
- Perrett, D. I., Burt, D., Penton-Voak, I. S., Lee, K. J., Rowland, D. A., and Edwards, R. 1999. Symmetry and human facial attractiveness. *Evolution and Human Behavior* 20.5 (1999): 295–307.
- Portenier, T., Hu, Q., Favaro, P., and Zwicker, M. 2018. Fine-grained retrieval with autoencoders. In *VISIGRAPP*. :
- Ranjan, A., Bolkart, T., Sanyal, S., and Black, M. J. 2018. Generating 3d faces using convolutional mesh autoencoders. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (ed.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, pp. 725–741. : Springer.
- Scheib, J. E., Gangestad, S. W., and Thornhill, R. 1999. Facial attractiveness, symmetry and cues of good genes. *Proc Biol Sci* 266.1431 (Sep 1999): 1913–1917.
- Singh, R., Vatsa, M., and Noore, A. 2009. Effect of plastic surgery on face recognition: A preliminary study. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 72–77. :
- Verhulst, A., Hol, M., Vreeken, R., Becking, A., Ulrich, D., and Maal, T. 2018. Three-dimensional imaging of the face: A comparison between three different imaging modalities. *Aesthetic Surgery Journal* 38 (2018): 579–585.
- Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. 2014. Learning fine-grained image similarity with deep ranking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1386–1393. :

- White, J. D., Ortega-Castrillon, A., Matthews, H., Zaidi, A. A., Ekrami, O., Snyders, J., Fan, Y., Penington, T., Van Dongen, S., Shriver, M. D., and Claes, P. 2019. Meshmonk : open-source large-scale intensive 3d phenotyping. (2019):
- Wiggers, K. L., Britto, A. S., Heutte, L., Koerich, A. L., and Oliveira, L. S. 2019. Image retrieval and pattern spotting using siamese neural network. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. :
- Zhang, J., Chen, K., and Zheng, J. 2020. Facial expression retargeting from human to avatar made easy. *IEEE Transactions on Visualization and Computer Graphics* (2020): 1–1.
- Zhang, S., Yu, H., Dong, J., Wang, T., Ju, Z., and Liu, H. 2015. Automatic reconstruction of dense 3d face point cloud with a single depth image. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1439–1444. :



## APPENDIX I

### LIST OF PUBLICATIONS

#### A.1 International Conference Proceeding

1. A. Chanchua and N. Chentanez, DeltaFace: Fully Automatic 3D Facial Cosmetic Surgery Simulation, 2021 25th International Computer Science and Engineering Conference (ICSEC), 2021, pp. 246-251, doi: 10.1109/ICSEC53205.2021.9684623.



## Biography

Adawat Chanchua received a bachelor degree of science in software engineering from Mae Fah Luang University in 2008. He and his team got a finalist in Nation Software Contest (NSC) 2009. He has more than ten years of experience in software development consisting of games, mobile applications, Android and websites. He was part of a team for the Final Fantasy 9 remaster. He is interested in computer graphics, computer vision and software development techniques.

