FACE RECOGNITION BASED ON FACIAL LANDMARKS DETECTION

ANIWAT JUHONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN BIOMEDICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2018

KMITL-2018-EN-M-045-168

FACE RECOGNITION BASED ON FACIAL LANDMARKS DETECTION

ANIWAT JUHONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIMENTS FOR DEGREE OF

MASTER OF ENGINEERING IN BIOMEDICAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

KMITL-EN-M-XXX-XXX

การจดจำใบหน้าด้วยวิธีการตรวจหาจุดสำคัญบนใบหน้า

อนิวัฒน์ จูห้อง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

KMITL-EN-M-XXX-XXX

หัวข้อวิทยานิพนธ์	การจดจำใบหน้าด้วยวิธีการตรวจหาจุดสำคัญบนใบหน้า
นักศึกษา	นายอนิวัฒน์ จูห้อง
รหัสนักศึกษา	59601121
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมชีวการแพทย์
พ.ศ.	2561
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร. ชูชาติ ปิณฑวิรุจน์

บทคัดย่อ

ลักษณะทางไบโอแมททริกซ์ (biometrics) เช่น ลายมือ ใบหน้า มือ ลายนิ้วมือ ม่านตา และอื่นๆ ได้รับความนิยมใช้ในระบบรักษาความปลอดภัยมากกว่า วิธีการใช้ รหัสผ่าน หรือ บัตรผ่านเข้าออก เนื่องจาก ไอโอแมทริกซ์เป็นระบบที่มีความน่าเชื่อถือที่ไม่สามารถจะขโมย ดัดแปลง หรือทำซ้ำได้ โดยในงานวิจัยนี้เรานำเสนอไบโอแมทริกซ์สำหรับการระบุตัวบุคลลโดยใช้การจุดตรวจจุดสำคัญบนใบหน้า ได้แก่จุดของตาทั้งสองขา จมูก และปาก โดยในส่วนแรกจะทำการระบุตำแหน่งที่เราสนใจ (Region of interest, ROI) ด้วย วิธี Haar cascade สำหรับการหาตำแหน่งของตา ปาก และจมูก หลังจากนั้นเราจะหาจุดสำคัญบนใบหน้า โดยการเปลี่ยนภาพไบนารี่ ด้วยการทำ Thresholding ในการที่จำกัดพื้นที่ของคิ้วเราจะทำการใช้การโปรเจคชั่นในแกนแนวนอน ซึ่งจะทำให้เราได้เฉพาะพื้นที่ของตา จากนั้นก็ทำการโปรเจคชั่นในแนวตั้งซึ่งจะทำเราให้ข้อมูลตำแหน่งของค่าพิเซลตำแหน่งแรกและตำแหน่งสุดท้ายของตาซึ่งจะนำมาใช้เป็นจุดสำคัญของตา โดยการโปรเจคชั่นในแนวตั้งก็จะถูกใช้ในการจุดสำคัญของจมูกและปากอีกเช่นกัน การได้มากซึ่งจุดสำคัญบนใบหน้าที่สอดคล้องกันของภาพใบหน้าอ้างอิงและภาพใบหน้าที่ต้องการตรวจสอบ จะถูกนำไปผ่านกระบวนการแปลงทางเรขาคณิตเพื่อทำการซ้อนทับกันของภาพทั้งสอง จากนั้น canny edge จะถูกใช้เพื่อทำการของภาพเส้นขอบเพื่อทำการหาค่าความผิดพลาดของการซ้อนทับกันของภาพ ในการทดสอบเราได้ใช้ภาพใบหน้าจากคลังข้อมูลโดยผลลัพธ์นั้นมีความแม่นยำที่สูงมาก (93 %)

Thesis Title	Face recognition based on facial landmarks detection
Student	Mr. Aniwat Juhong
Student ID	59601121
Degree	Master of Engineering
Program	Biomedical Engineering
Year	2018
Thesis Advisor	Assoc.Prof. Dr. Chuchart Pintavirooj

Abstract

Biometric characteristics such as palm print, face, hand and finger geometry, fingerprint, Iris, etc. are mostly popular used in security systems over the traditional secure measures, password or ID cards. The biometric systems are more reliable because they cannot easily be lost, stolen, shared and duplicated. In this research we propose the biometric for person identification based on facial landmark pattern. Our landmarks are those associated with eyes mouth and nose. To extract facial landmarks, we first use Haar cascade algorithm to detect the face region of interest (ROI) following by Haar cascade algorithm for the eye, mouth and nose ROI determination. To find landmark associated with the eyes, we convert eye ROI image to binary image using thresholding algorithm. To exclude the eyebrow region, we apply horizontal projection. The project data will then be used to separate the eyebrow region from the eye region. To detect eye-related landmark, vertical projection is applied. With the vertical projection data, the outermost pixel can be identified and the associated eye landmark can be determined. The similar technique can then be used to identify landmarks associated with the nose and mouth area. Given the correspond landmarks on the reference face and the query face, geometric transformation can be determined using normal equation bases on minimized mean squared errors. The two faces are then aligned. To provide the quantitative measurement, the two aligned face are converted to edge image using canny edge algorithm. The distance map error between the two aligned edge facial images is then used to identify the query face. The purposed algorithm for person identification was tested on the face database resulting in a very high accuracy (approximately 93%).

Acknowledgements

Firstly, I would like to express my sincere gratitude my advisor Assoc. Prof. Dr. Chuchart Pintavirooj for the continuous support of my master study and related research, for his patient, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master study. And I also would like to thank my family and my friends for supporting me.

Aniwat Juhong

Contents

List of table

Table page

List of Figures

Figure page

List of Figures (Cont.)

Chapter 1
Introduction

1.1 Statement and Significance of the problems

Biometrics is the measurement and statistical analysis of people's unique physical and behavioral characteristic. The technology is mainly used for identification and access some control systems. There are two main types of biometric identifiers depending on either behavioral characteristics or physiological characteristics [1].

Behavioral identifiers include the unique ways in which individuals act, recognition of typing patterns, walking gait and other gestures. Some of these behavioral identifiers can be used to provide continuous authentication instead of a single one-off authentication check.

Physiological identifiers relate to the composition of the user being authenticated and include facial recognition, fingerprints, finger geometry, iris recognition, vein recognition, retina scanning, voice recognition and DNA matching.

In addition, this research we propose a novel algorithm for face recognition, it is a technology matches the unique characteristics for the purpose of personal identification to apply in many industries. For example, law enforcement agencies use face recognition to keep communities safer, retailers could prevent crime and violence by using it, or it is applied in airport to improve traveler's convenience and security. In traditional methods, face recognition could sometime find mismatch of face images because there are invariance factors result from human pose, hairstyle, acne arising, fatter etc. Thus, face recognition based on facial geometric landmarks was developed and seems promising to solve these problems.

Nowadays, there are many techniques in face recognition, and they are divided into two main methods by using acquisition tools which are 2D and 3D. Moreover, some researches combined both 2D and 3D together to get extremely high accuracy result. However, 3D techniques need to use advance and costly equipment for getting face images [2]. Hence, in general purpose of face recognition, 3D is overkill technique, while 2D is sufficient to obtain satisfying result.

The traditional methods of face recognition in 2D can be divided into two techniques. Firstly, face recognition based on grey image alignment such as Principal Component Analysis(PCA) as known as eigenface method [4], linear discriminant analysis(LDA). This method has to use more information to train the system and also requires high computation. Secondly, Face recognition is based on facial landmarks and geometric pattern. The recognition based on this method mostly depends on the number of facial landmarks that the accuracy of result can increase, when we increase the number of facial landmarks for using in face recognition [3]. Therefore, this method need to extract landmarks as much as possible, albeit it is somewhat difficult to extract the landmarks.

Figure 1.1 Eigen face image based on alignment of face image [4]

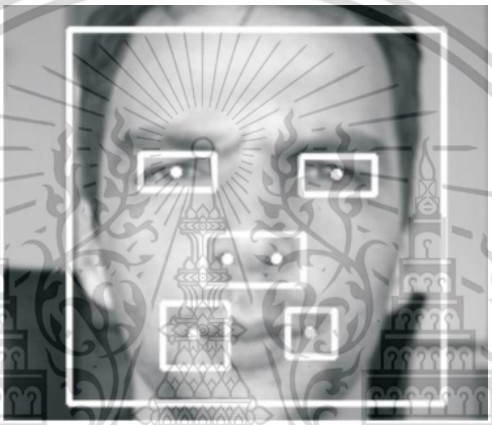


Figure 1.2 Facial landmarks detection [3]

The idea of this thesis is to aggregate these both methods, which are image registration and geometric features from facial landmarks, in face recognition.

1.2 Goal and objective

This research proposes a novel technique for face recognition by using the facial landmarks extraction and facial image registration. Facial landmarks are extracted by radon transform, these landmarks are applied for facial image registration and define facial features to identify persons.

1.3 Hypothesis

Each of people has the different facial landmarks, therefore these landmarks can be applied for face recognition using the novel technique that was developed in this research.

1.4 Scope of Research

This research developed 2D technique to identify persons by using facial biometric. The technique extracts landmark associated with facial anatomical landmark including nose, eye and mouth. The extracted landmark can be used to estimate geometric transformation, which is an important process in face image registration. The 2D query face images are then aligned the reference image. Distance map error between the edge images of the two aligned image in region of interest and distance of each facial component (in pixel unit) are used for person identification.

1.5 Chapter Organizing

The succeeding chapters are fine organized as the following: -
Chapter 2 Theoretical framework
Chapter 3 Image registration
Chapter 4 Research Methodology
Chapter 5 Experiment and Results
Chapter 6 Conclusions & suggestions

Chapter 2
Theoretical framework

Image processing is a method to perform some operations on an image, in order to obtain an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be the image or some characteristics/features associated with that image. Currently, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines.

There are two types of method which are used for image processing namely, analogue and digital image processing. Analogue image processing can be applied for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display information extraction.

2.1 Digital Image Definitions

A digital image f[m,n] described in a 2D discrete space is transformed from an analog image f(x,y) by sampling process which is frequently referred to as digitization. Digital image can represent in two dimensional image as a finite set of digital values which are called pixels. Pixel values basically represent gray levels, color, heights, etc. For instance, a pixel value in the rage: [0, 255] that represents grey level as the gray level value raging from black (0) to white (255) as shown in figure 2.1.

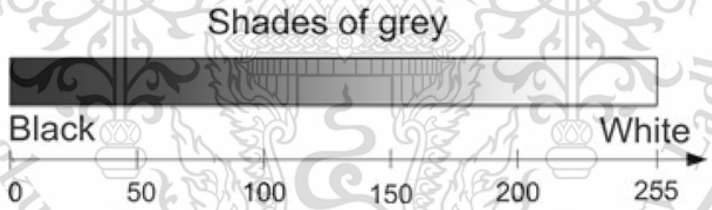


Figure 2.1 The relationship between the intensity values and the different shades of gray [5]

2.2 Digital Image types

There are four main images using in image processing which are intensity image, binary image, index image, and RGB image.

2.2.1 Intensity images

An intensity image is a data matrix, I, whose values represent intensities within some range. the intensity image can be represented as a single matrix, with each element of the matrix corresponding to one image pixel. The matrix can be of class double, uint8, or uint16. Intensity images are rarely saved with a colormap. The elements in the intensity matrix represent various intensities, or gray levels, where the intensity 0 usually represents black and the intensity 1, 255, or 65535 usually represents full intensity, or white.

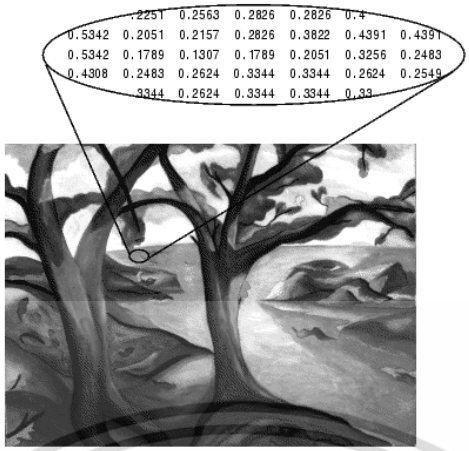


Figure 2.2.1 Pixel Value in an Intensity Image Define Gray Levels [6]

2.2.2 Binary image.

In a binary image, each pixel assumes one of only two discrete values. Typically, these two values correspond to on (1) and off (0) as shown in figure (2.3). Binary images are often produced by thresholding a grayscale or color image, in order to separate an object in the image from the background. The color of the object (usually white) is referred to as the foreground color. The rest (usually black) is referred to as the background color. However, depending on the image which is to be thresholded, this polarity might be inverted, in which case the object is displayed with 0 and the background is with a non-zero value.

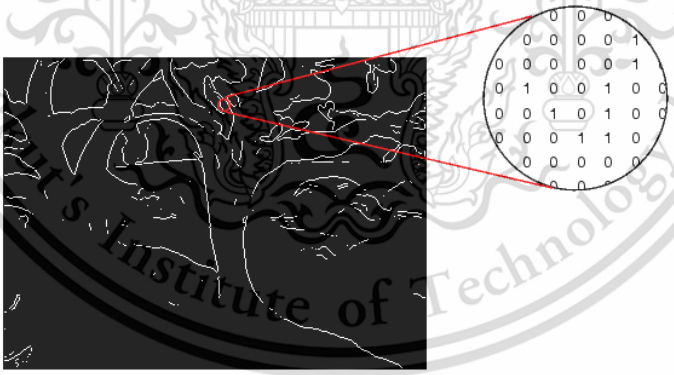


Figure 2.2.2 Pixels in a binary image have two possible value: 0 or 1 [6]

2.2.3 Indexed Images

An index image consists of a data matrix, X, and a colormap matrix. The data matrix can be of class uint8, uint16, or double. The color matrix is an m-by-3 array of class double containing floating point value within the range [0, 1]. Each row of map specifies the red, green, and blue components of a single color. An indexed image applied direct mapping of pixel values to colormap values. The color of each image pixel is defined by using the corresponding value of X as an index into map. The value 1 points to the first row in map, the value 2 points to the second row, and so on.

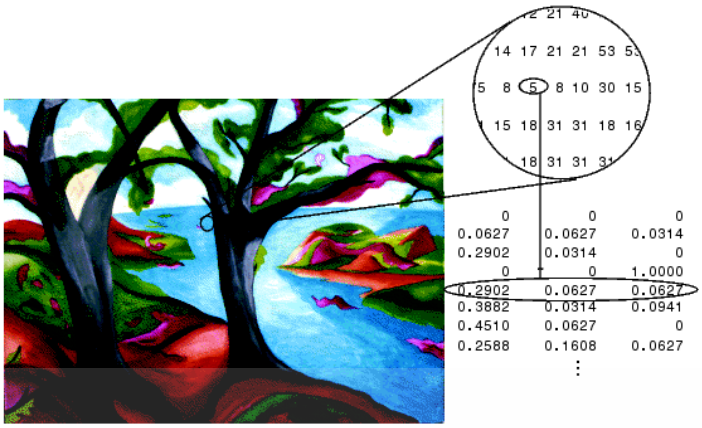


Figure 2.2.3 Pixel values are indices to a colormap in indexed images [6]

2.2.4 RGB image

An RGB image is an m-by-n-by-3 data array that defines red, green, and blue color component for each individual pixel. The color of each pixel is determined by the combination of the red, green, and blue intensities stored in each color plane at the pixel's location. Graphics file formats store RGB images as 24-bit images, where the red, green, and blue components are 8 bits each. This yields a potential of 16 million colors.

Figure 2.2.4 The Color Planes of an RGB Image [6]

2.3 The Region of Interest (ROI)

As digital cameras are sold in larger and larger numbers the development within sensor technology has resulted in many new products including larger and larger numbers of pixels within one sensor. This is normally defined as the size of the image that can be captured by a sensor, i.e., the number of pixels in the vertical direction multiplied by the number of pixels in the horizontal direction. Having a large number of pixels can result in high quality images and has made, for example, digital zoom a reality.

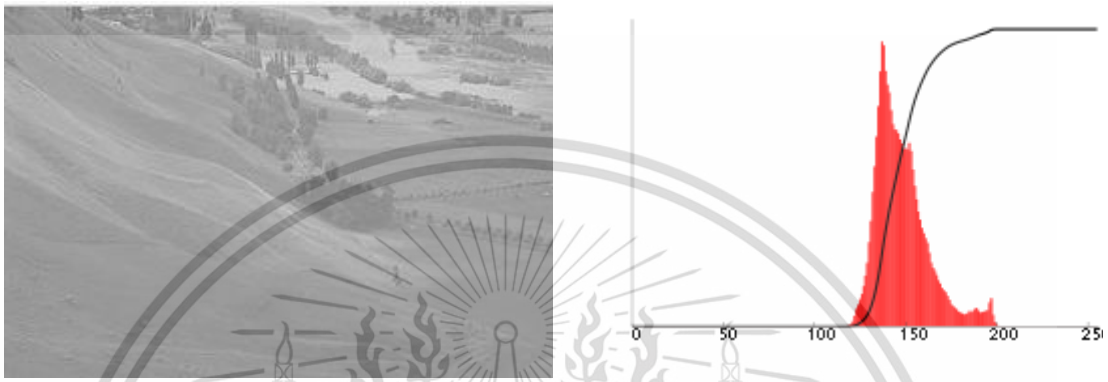
When it comes to image processing, a larger image size is not always a benefit. Unless you are interested in tiny details or require very accurate measurements in the image, you are better off using a smaller sized image. The reason being that when we start to process images we have to process each pixel, i.e., perform some math on each pixel. In addition, due to the large number of pixels, that quickly adds up to somewhat a large number of mathematical operations, which in turn means a high computational load on your computer. For example, when we get image which is 500 x 500 pixels. That means we have 500 x 500 = 250,000 pixels. Now say that we are processing video with 50 images per second. That means that you have to process 50 x 250,000 = 12,500,000 pixels per second. Therefore, your algorithm requires 10 mathematical operations per pixel, then in total our computer has to do 10 x 12,500,000 = 125,000,000 operation per second. That is a significantly number even for currently powerful computers. Hence, when we choose our camera do not make the mistake of thinking that bigger is always better [7].

Apart from picking a camera with a reasonable size, we should also consider introducing a region-of-interest (ROI). An ROI is simply a region (normally a rectangle) within the image which defines the pixels of interest. Those pixels not included in the region are ignored altogether and less processing is therefore required. An ROI is illustrated in figure 2.3.

Figure 2.3 The white rectangle defines a region-of-interest (ROI) [7]

2.4 Histogram Equalization

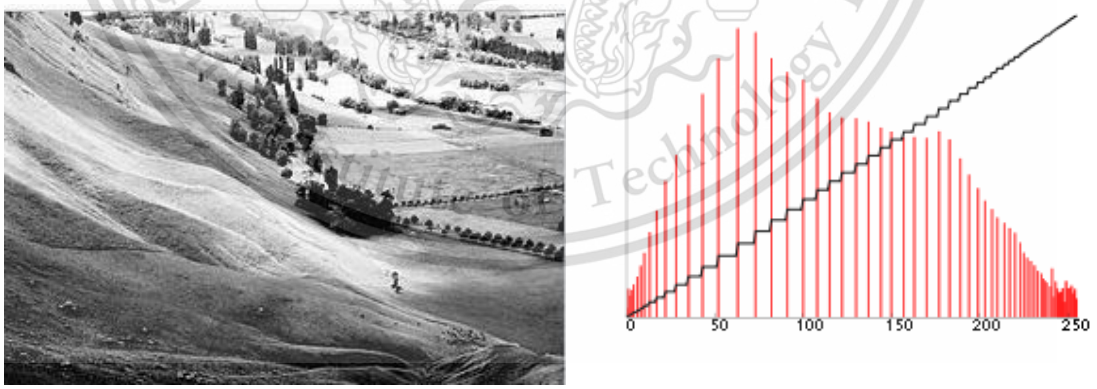
Histogram is a graphical representation of the intensity distribution of an image. In simple terms, it represents the number of pixels for each intensity value considered. In the figure 2.30, X-axis represents the total scale (black at the left and white at the right), and Y-axis represents the number of pixels in an image. Here, the histogram shows the number of pixels for each brightness level (from black to white), and when there are more pixels, the peak at the certain brightness level is higher. [8]



(a) (b)

Figure 2.4.1 (a) Before Histogram equalization and (b) Corresponding histogram (red) and cumulative histogram (black) [8]

Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast.

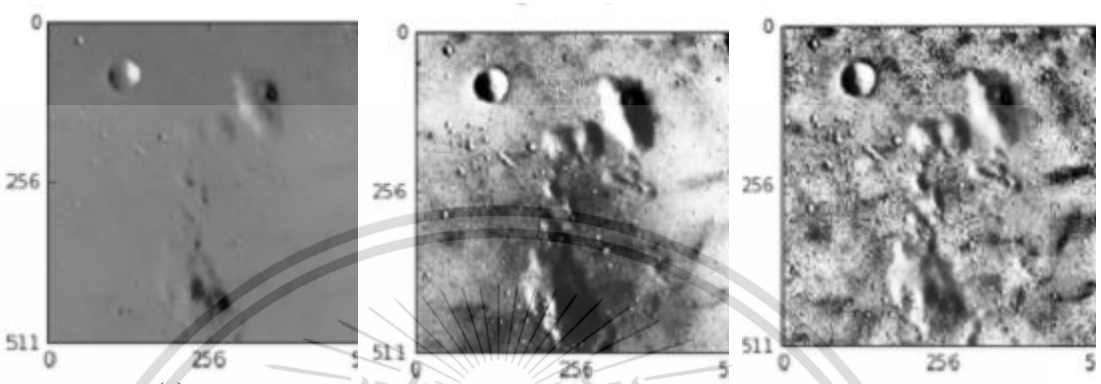


(a) (b)

Figure 2.4.2 (a) After Histogram Equalization, (b) Corresponding histogram(red) and cumulative histogram (black) [8]

A color histogram of an image represents the number of pixels in each type of color component. Histogram equalization cannot be applied separately to the Red, Green and Blue components of the image as it leads to dramatic changes in the image's color balance. However, if the image is first converted to another color space, like HSL/HSV color space, then the algorithm can be applied to the luminance or value channel without resulting in changes to the hue and saturation of the image.

Adaptive Histogram Equalization differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image.



(a) (b) (c)

Figure 2.4.3 (a) Initial image, (b) Histogram equalized (c) Adaptive histogram equalized[8]

Contrast Limited AHE (CLAHE) differs from adaptive histogram equalization in its contrast limiting. In the case of CLAHE, the contrast limiting procedure is applied to each neighborhood from which a transformation function is derived. CLAHE was developed to prevent the over amplification of noise that adaptive histogram equalization can give rise to.

(a) (b) (c)

Figure 2.4.4 (a) Initial image, (b) Histogram equalized (c) Contrastive limited adaptive equalized image [8]

2.5 Image Thresholding

Thresholding consists of segmentation an image into two regions: a particle region and background region. In its most simple form, this process works by setting to white all pixels that belong to gray-level interval, called the threshold interval, and setting all other pixels in the image to black. The resulting image is referred to as a binary image. For color images, three thresholds must be specified, one for each color component. [6] The threshold can be chosen manually or by using automated techniques. Manual threshold selection is normally done by trial and error, using a histogram as a guide. Figure 2.35 shows a threshold chosen to isolate the brightest particles form an image [9].

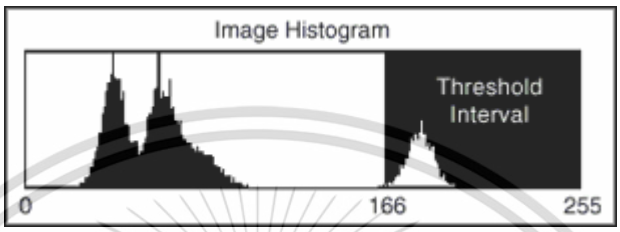
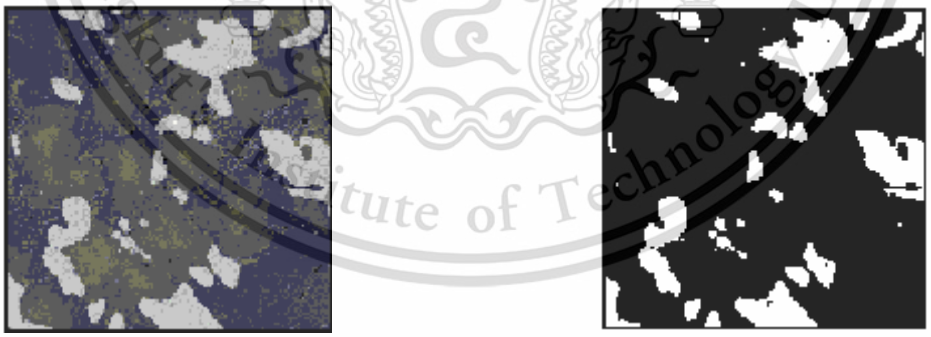


Figure 2.5.1 Image Histogram [9]

Automated thresholding techniques select a threshold which optimizes a specified characteristic of the resulting images. These techniques include clustering, entropy, metric, moments, and interclass variance. Clustering is unique in that it is a multi-class thresholding method. In other words, instead of producing only binary images it can specify multiple threshold levels which result in images with three or more gray-level value.

Thresholding is the most common method of segmenting images into particle regions and background regions. A typical processing procedure would start with filtering or other enhancements to sharpen the boundaries between objects and their background. Then, the objects are separated from the background using thresholding.

The binary image resulting from the above thresholding operation is shown in Figure 2.52.



(a) (b)

Figure 2.5.2 (a) Original Image (b) Threshold binary image [9]

The light particles are separated in the binary image, while the various darker shades of gray are all set to black. Various morphological operations can be used to further refine the image by filling small holes, deleting small particles and joining or separating adjacent particles.

2.6 Haar cascade Algorithm

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Here we will work with face detection. Initially, the algorithm needs a lot of positive each facial components (images of faces, both of eyes, nose and mouth) and negative images (images without faces, both of eyes, nose and mouth) to train the classifier. Then we need to extract features from it. For this, Haar features shown in figure 2.36 are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle. [10]

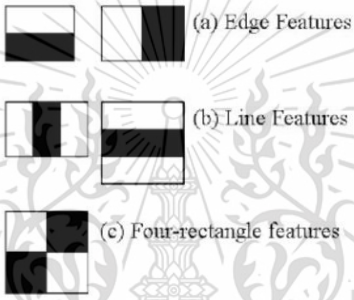


Figure 2.6.1 Haar feature [10]

All possible sizes and locations of each kernel is used to calculate plenty of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find sum of pixels under white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels. But among all these features we calculated, most of them are irrelevant. For example, consider the image below (Figure 2.6.2). Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by Adaboost. [10]

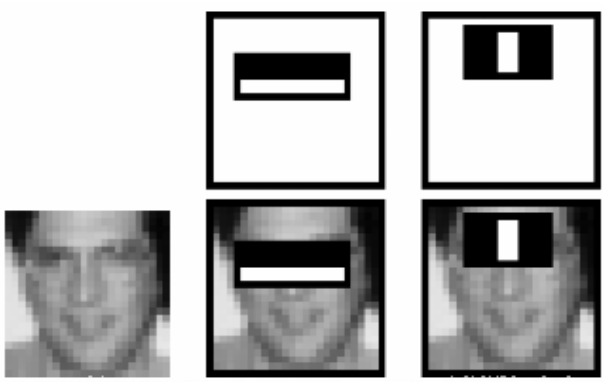


Figure 2.6.2 Face detection using Haar feature [10]

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated. Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).

Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone cannot classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with approximately 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).

Eventually, we take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not. In an image, most of the image region is non-face region. Therefore, it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot. Do not process it again. Instead focus on region where there can be a face. This way, we can find more time to check a possible face region.

For this they introduced the concept of Cascade of Classifiers. Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

Authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in first five stages. (Two features in the above image is actually obtained as the best two features from Adaboost).

2.7 Radon Transformation

The radon function computes projections of an image matrix along specified directions. A projection of a two-dimensional function f(x,y) is a set of line integrals. The radon function computes the line integrals from multiple sources along parallel paths, or beams, in a certain direction. The beams are spaced 1-pixel unit apart. To represent an image, the radon function takes multiple, parallel- beam projections of the image from different angles by rotating the source around the center of the image. The following figures shows a single projection at specified rotation angle. [11]

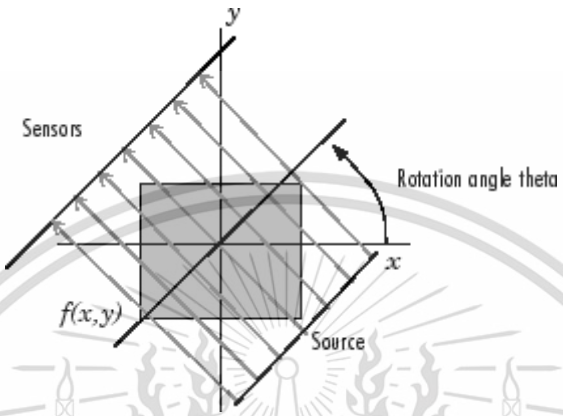


Figure 2.7.1 Parallel-Beam Projection at Rotation Angle Theta [11]

For example, the line integral of f(x,y) in the vertical direction is the projection of f(x,y) onto the x-axis, the line integral in the horizontal direction is the projection of f(x,y) onto the y-axis. The following figure shows horizontal and vertical projections for a simple two-dimensional function.

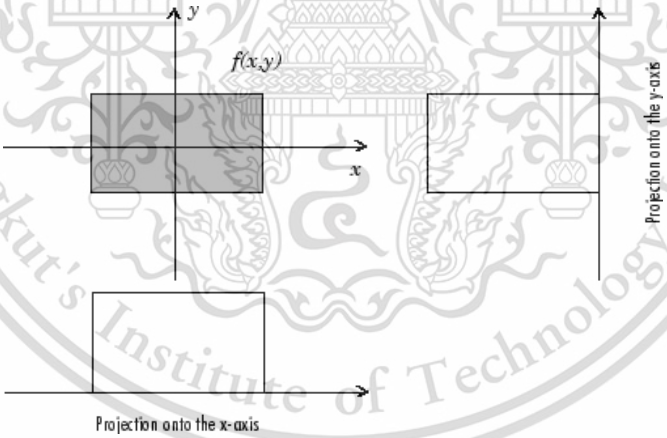


Figure 2.7.2 Horizontal and vertical Projection of a simple function [11]

Projections can be computed along any angle theta (θ). In general, the Radon transform of f(x,y) is the line integral of f parallel to the y'-axis.

$$R_\theta(x') = \int_{-\infty}^{\infty} f(x' \cos\theta - y' \sin\theta, x' \sin\theta - y' \cos\theta)\, dy'$$

Where

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.29}$$

The following figure (Geometry of the Radon transform) illustrates the geometry of the radon transform.

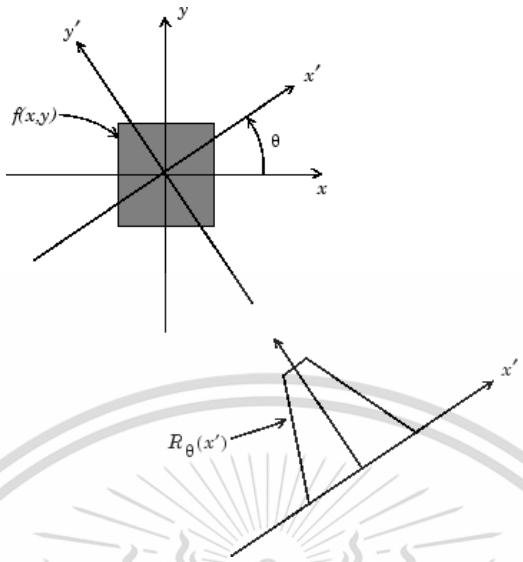


Figure 2.7.3 Geometry of the Radon Transform [11]

2.8 Canny edge detection

Canny Edge Detection is a popular edge detection algorithm. It was developed by John F. Canny in 1986. It is a multi-stage algorithm and we will go through each stages. [13]

2.8.1 Noise Reduction

Since edge detection is susceptible to noise in the image, first step is to remove the noise in the image with a 5x5 Gaussian filter.

2.8.2 Finding Intensity Gradient of the Image

image is then filtered with a Sobel kernel in both horizontal and vertical direction to get first derivative in horizontal direction (Gx) and vertical direction (Gy). From these two images, we can find edge gradient and direction for each pixel as follows:

$$\text{Edge Gradient(G)} = \sqrt{G_x^2 + G_y^2} \qquad (2.39)$$

$$\text{Angle}(\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \qquad (2.40)$$

Gradient direction is always perpendicular to edges. It is rounded to one of four angles representing vertical, horizontal and tow diagonal

2.8.3 **Non-maximum Suppression**

After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge. For this, at every pixel, pixel is checked if it is a local maximum in its neighborhood in the direction of gradient. Check the image below:

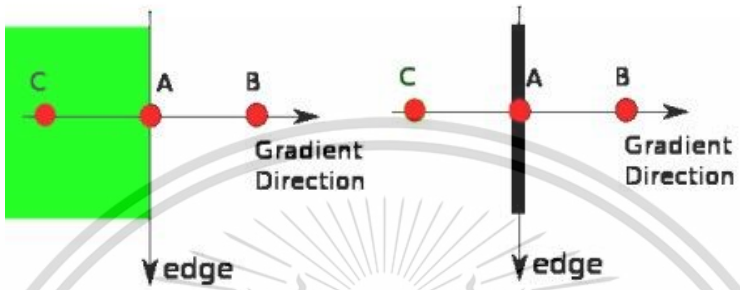


Figure 2.8.1 Direction of gradient [13]

Point A is on the edge (in vertical direction). Gradient direction is normal to the edge. Point B and C are in gradient directions. So point A is checked with point B and C to see if it forms a local maximum. If Therefore, it is considered for next stage, otherwise, it is suppressed (put to zero). In short, the result you get is a binary image with "thin edges".

2.8.4 **Hysteresis Thresholding**

This stage decides which are all edges are really edges and which are not. For this, we need two threshold values, minVal and maxVal. Any edges with intensity gradient more than maxVal are sure to be edges and those below minVal are sure to be non-edges, so discarded. Those who lie between these two thresholds are classified edges or non-edges based on their connectivity. If they are connected to "sure-edge" pixels, they are considered to be part of edges. Otherwise, they are also discarded. See the image below:

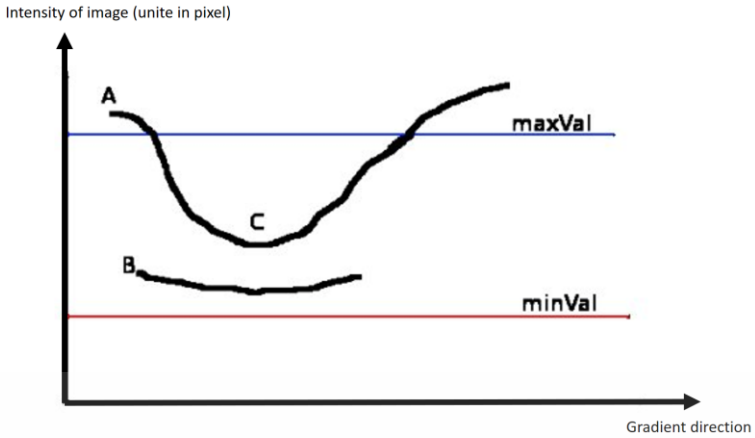


Figure 2.8.2 Hysteresis Thresholding [13]

The edge A is above the maxVal, so considered as "sure-edge". Although edge C is below maxVal, it is connected to edge A, so that also considered as valid edge and we get that full curve. But edge B, although it is above minVal and is in same region as that of edge C, it is not connected to any "sure-edge", so that is discarded. So it is very important that we have to select minVal and maxVal accordingly to get the correct result. This stage also removes small pixels noises on the assumption that edges are long lines. Therefore, what we finally get is strong edges in the image.

Chapter 3
Image registration

3.1 Introduction of image registration

Image registration is the process of estimating an optical geometric transformation to bring homologous points of two images as close as possible. Image registration can be classified into iterative-based method and feature-based method.

An iterative image registration depends on intensity of pixel data for 2D images, and intensity of voxel data for 3D images. The concept of this method is to repeat operations in many times. In each time, parameters are adjusted until getting the best result or minimal error in order to avoid the error from local minima. In addition, this method needs to use initial alignment before iterative process to register image correctly. The initial alignment also needs to apply with many parameters, which are computed error, as shown in research of Wood and et al [14,15].

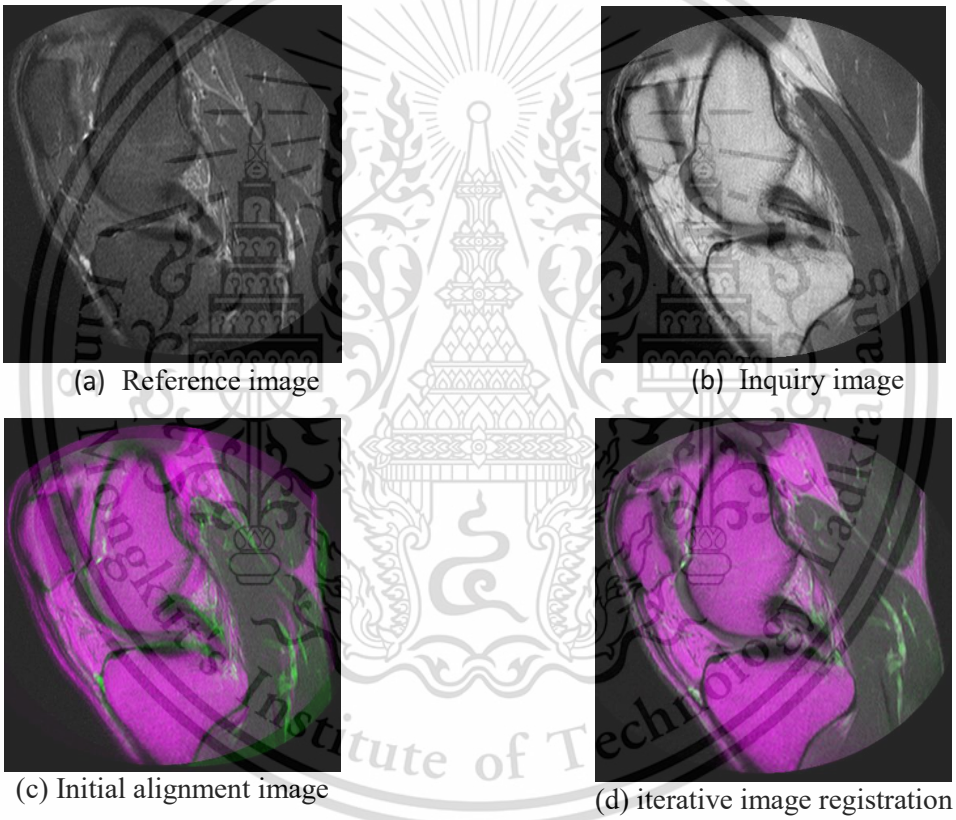


Figure 3.1 iterative image registration [16]

A feature-based image registration has a massive advantage in terms of getting less time to process. The first step of this method is to extract feature points of reference image and inquiry image. These feature points are then utilized with transformation parameters to do geometric transformations for getting registered images. The geometric transformations can be divided into two techniques that are linear geometric transformation and non-linear geometric transformation. The linear geometric transformations consist of scaling, translation, rotation, and shearing. while the other has features more than the linear geometrics transformations features, particularly bending feature.

Image registration based on linear geometric transformations consist of four steps. The first step is the geometrics landmarks extraction of both reference image and inquiry image. The second step is to find corresponding points. The third step, these points are used to find "T" matrix of geometric transformations. The last step, "T" matrix obtaining from the third step is applied for image registration and alignment. The additional information about geometric transformations is explained in topic 3.3.

3.2 Geometric invariant and features

Geometric invariant is most applied in many image processing applications [3] by using the properties of geometric invariant such as area, volume, angle, and so on. The figure 3.3 shows the geometric invariant features of face image which are six areas and 18 internal angles. The corresponding features are used to compute an error for image registration. For example, the first error is defined by the subtraction of ratio of the first area and the second area as shown in equation (3.1). This ratio is an invariant feature and we also do like this with every invariant features: ratios of internal angles, ratios of other areas, and so on. Eventually, we get the average error of image registration from all geometric invariant features.

$$\text{The first error} = \left| \frac{A1}{A2} - \frac{A1'}{A2'} \right| \quad (3.1)$$

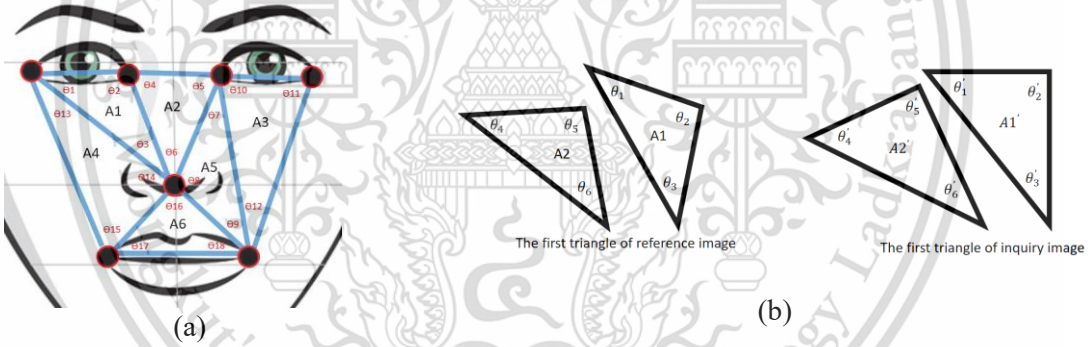


Figure 3.2.1 Areas and internal angles of facial triangles

In addition, when we cannot know the exact order of geometrics features, the corresponding features might be got from sorting by size or value of geometric features. For instance, the biggest area of reference image corresponds with the biggest area of inquiry image or the second biggest of the reference corresponds with the second biggest are of inquiry image.

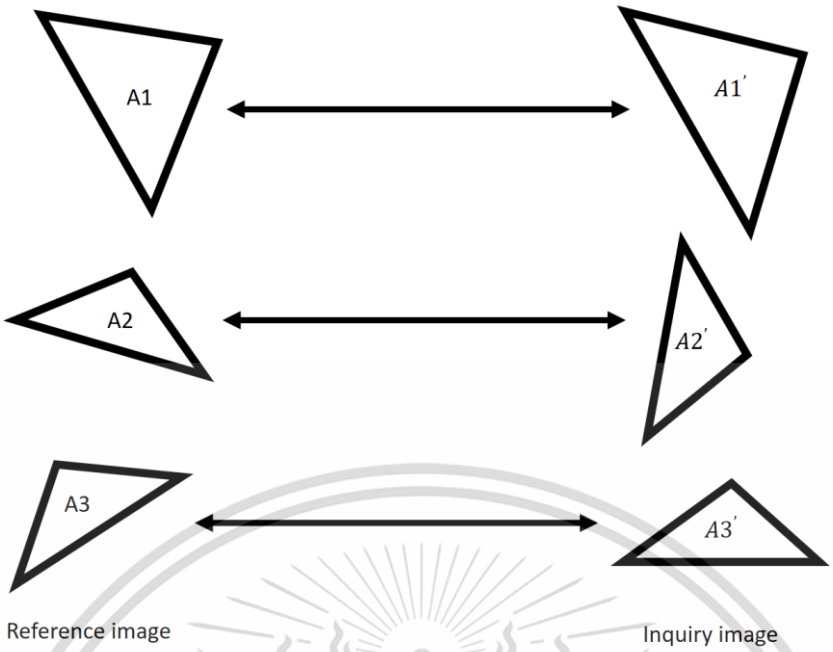
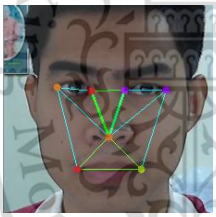
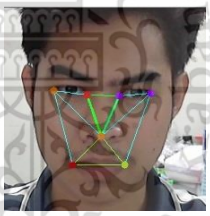


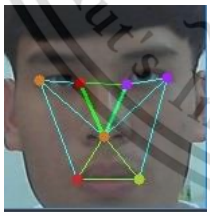
Figure 3.2.2 corresponding triangle of both reference image and inquiry image sorted by the areas of triangles



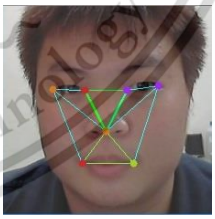
(a) The reference image



(b) The first inquiry image



(c) The second inquiry image



(d) The fourth inquiry image

Figure 3.2.3 Facial landmarks extraction used in geometric invariant coding

Under rigid transformation, triangle side and area is absolute invariance [2,3]. Beyond the rigid transformation, similarity and affine transformation, the triangle side lengths and angles are no longer preserved. The area of the corresponding triangles, however, becomes a relative invariant, with the two corresponding areas being related to each other through the determinant of the linear transformation matrix *T*, in the affine map transform *(T,b)* where *b* is the translation vector. If the area patches of the sequence of triangles on the template are [A(1), A(2),A(3)…, A(n)] then the corresponding area patches $A_a(k)$ of the sequence of triangles on the query under affine transformation are related to those of the template in accordance with the following relative invariant.

$$A_a(k) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} A(k), k = 1,2,3,...,n \qquad (3.2)$$

Where $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$ is the determinant of affine transformation matrix. As the linear transformation matrix is unknown, absolute affine invariants are constructed out of the area relative invariants by taking the ratio of two triangles to cancel out the dependence of the area relative invariant on the determinant of affine transformation matrix. By taking the ratio of the consecutive elements in the sequence, the set of absolute invariants in (3.3) and (3.4) are obtained

$$I(k) = \frac{A(k)}{A((k+1) \bmod n)}, k = 1,2,3,...,n \qquad (3.3)$$

and

$$I_a(k) = \frac{A_a(k)}{A_a((k+1) \bmod n)}, k = 1,2,3,...,n \qquad (3.4)$$

In the case of noise free measurement, the absolute invariant of the query equals that of the template, i.e. $I_a(k) = I(k)$ and in the presence of noise and occlusion, each of $I_a(k)$ will have a counter part with $I(k)$, with that counterpart easily determines through a circular shift involving n comparison where n is the number of invariants. To allow for noise and small deviations from an affine map, we allow a small error percentage difference between corresponding invariants to allow for only small difference between the area patches before declare them as matching. This may reduce the length of the matched triangle sequence. The lower the error percentage is the more strict the matching. Experimentally, an error percentage of 5% was applied. We adopt a run length method to decide on the correspondence between the two ordered set of triangles. For every starting point on the sequence, the run length method computes a sequence of consecutive invariants satisfying the criterion

$$\%e^2 = \frac{(I_a(j) - I(i))^2}{I(i)^2} \times 100 < \zeta \qquad (3.5)$$

We declare the match on the longest string (M) of triangles that yields minimum averaged error of $\frac{\sum_i \%e_i^2}{M}$.

3.3 Geometric image transformations

Geometric transformations are ubiquitously applied for image registration and the removal of geometric distortion. Typically, it is used in many applications such as construction of mosaics, geographical mapping, stereo and video, biometrics, and so on. The transformations can be classified into two main approaches: spatial transformation, and affine transformation.

A spatial transformation of an image is a geometric transformation of the image coordinate system. It is often significant to perform a spatial transformation to align image that were taken at different time or with different sensors, correct image for lens distortion, correct of camera orientation, image morphing, etc.

An affine transformation is any transformation that preserves collinearity as all points lying on a line initially still lie on a line after transformation and ration of distance that the midpoint of line segment remains the midpoint after transformation. Basically, an affine transformation is composition of scaling, translation, shear x by y, shear y by x, and rotation. These are defined following matrixes (3.6), (3.7), (3.8), (3.9), and (3.10), respectively.

Scaling is defined as

$$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Where S_x, and S_y are the respective scaling parameter in x and y axis.

Figure 3.3.1 Scaling of image

Note: A reflection can be performed with scaling parameters (Sx,Sy)=(−1,1) or (1,−1)

Translation is defined as

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} \quad (3.7)$$

Where T_x, and T_y are the Translation parameter in x and y axis respectively.

Figure 3.3.2 Translation of image

Shear *x* is defined as

$$\begin{bmatrix} 1 & Q_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

Where Q_x is the shear x by y parameter

Figure 3.3.3 Shear x by y of image

Shear *y* is defined as

$$\begin{bmatrix} 1 & 0 & 0 \\ Q_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

Where Q_y is the shear *x* parameter

Figure 3.3.4 Shear y by x of image

Rotation is defined as

$$\begin{bmatrix} cos\,\theta & -sin\,\theta & 0 \\ sin\,\theta & cos\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

Where θ is the angle of rotation around the origin point

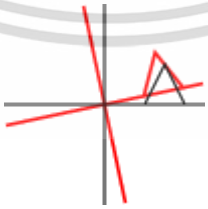


Figure 3.3.5 Rotation of image

T is the affine transformation matrix which can be derived from multiple result of scaling matrix, shear *x* by *y* matrix, shear *y* by *x* matrix, and rotation matrix as shown in equation (3.11), and it is reduced form to be simple matrix as represented in equation (3.12).

$$T = S.Sh_x.Sh_y.R.Tr$$

$$\mathrm{T} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} \begin{bmatrix} 1 & Q_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ Q_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

$$T = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

When we get all corresponding points of inquiry image and reference image, we can do affine transformation

$$Y = T.X \quad (3.13)$$

Where X is the matrix of corresponding points of inquiry image, Y is the matrix of corresponding points of reference image.

$$X = \begin{bmatrix} x_1 & x_2 & \dots x_n \\ y_1 & y_2 & \dots y_n \\ 1 & 1 & \dots 1 \end{bmatrix} \quad (3.14)$$

$$Y = \begin{bmatrix} x'_1 & x'_2 & \dots x'_n \\ y'_1 & y'_2 & \dots y'_n \\ 1 & 1 & \dots 1 \end{bmatrix} \quad (3.15)$$

In practical terms, we have to get T matrix firstly before doing affine transformation. Therefore, we can use the corresponding point of both reference image and inquiry image to estimate *T* matrix, which is shown in equation (3.12). Where X is the matrix of corresponding points of inquiry image and Y is the matrix of corresponding points of reference image as shown in equation (3.14), and (3.15) respectively.

To begin with square error(ϵ^2) that is defined by.

$$\epsilon^2 = (Y - TX)^T (Y - TX) \quad (3.16)$$

Find the derivative of ϵ^2 by *dT*, and the square error account for zero.

$$\frac{d\epsilon^2}{dT} = -2X^T(Y - TX) = 0$$

$$X^T Y = X^T T X$$

$$Y^T X = T^T X^T X$$

$$T^T = (Y^T X)(X^T X)^{-1}$$

$$T = (X^T X)^{-1}(X^T Y) \quad (3.17)$$

Chapter 4
Research methodology

The overview of the algorithm in this research is shown in figure 3.1. Firstly, we use Haar cascade algorithm to detect region of interest (ROI) of facial components that consist of both of eyes, nose and mouth. Secondly, all facial landmarks are defined by radon transform. Given the correspond landmarks on the reference face and the inquiry face, geometric transformation can be determined using normal equation based on minimized mean squared error. The two faces are then aligned. To provide the quantitative measurement, the two aligned faces are converted to edge image using canny edge algorithm. The distance map error between the two aligned edge facial images is then used to be one of features for face recognition. Another feature is distance of each facial landmarks. Both features are used to identify the inquiry face. Facial region of interest, facial landmarks extraction, image registration, canny edge detection, distance map error and distance of each facial landmarks are explained in 4.1, 4.2, 4.3, and 4.4, respectively.

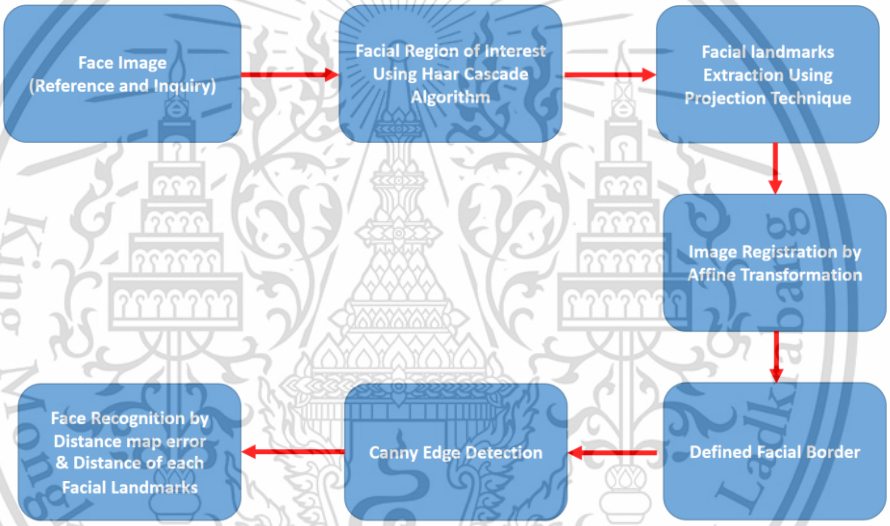
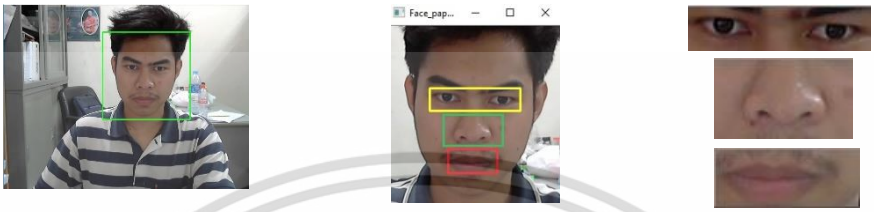


Figure 4.1 The Overview of process in this work

4.1 Defined Facial Region of interest using Haar cascade Algorithm.

The first goal of our approach is to defined region of interest(ROI) of 2D face image by Haar cascade algorithm. Haar cascade is used to define ROI in two steps. The first step, the algorithm is used to find the number and position of face and crops the ROI of face automatically as shown the result in figure 4.2 (a)-(b). The second step, we use Haar cascade again to get ROI of facial components which are both of eyes, nose, and mouth as represented in figure 4.2 (b)-(c).



(a) (b) (c)

Figure 4.2 Defined ROI of facial components by Haar cascade algorithm

4.2 Facial landmarks extraction based on projection technique.

To find landmark associated with the eye, we can covert eye ROI image to binary image using thresholding algorithm. The result is illustrated in figure 4.3 (b) which includes the binary region associated with eye and eyebrow. Therefore, we have to exclude the eyebrow region by horizontal projection. The projection data is used to distinguish between eye region and eyebrow region. To detect eye related landmark, vertical projection is applied. With the outermost pixel can be identified and the associated eye landmark can determine as shown in yellow dot in figure 4.3 (c)

(a) (b) (c)

Figure 4.3 Projection technique to find landmarks of eye

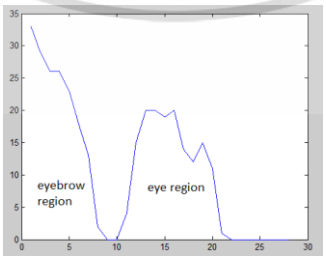


Figure 4.4 Projection data of Fig 4.3 (b)

To extract the nose related landmark, we convert the nose ROI to binary image and apply vertical projection. The outermost pixel can be identified and the associated nose landmark, as shown in figure 4.5. Figure 3.6 shows the similar algorithm that is applied to detect mouth-related landmark.

(a) (b) (c)

Figure 4.5 Projection technique to find landmarks of nose

(a) (b) (c)

Figure 4.6 Projection technique to find landmarks of mouth

4.3 Image registration using facial landmarks and affine transform.

Image registration is used for person identification. To align query face image against the reference image in the database, the fiducial points are extracted. The fiducial points on the query face and the reference face are then used to estimate the affine transformation matrix based on the corresponding key points by equation (3.13). Where T is the affine transformation matrix. X and Y are the corresponding key points of reference and inquiry facial image respectively. The registration of the inquiry against the reference landmark point are shown in Fig. 4.7.

(a) Face registration of corresponding person

(b) Face registration of different person

Figure 4.7 Face Registration (Circle dot: Landmark of reference image, Cross dot) Landmark of query image**.**

4.4 Person Identification.

To identify person based on extracted fiducial point the face image that used to align the reference image against the query image, the face region of interest is determined using facial border point. Facial border point, *P1*, *P2, P3, P4, P5* and *P6*, are shown in figure 4.8. The definitions of facial border points are as follows:

P1 is defined by the coordinate $(X1-\frac{X5-X1}{2}, Y1)$ (4.1)

P2 is defined by the coordinate $(X6-\frac{X5-X6}{2}, Y6)$ (4.2)

P3 is defined by the coordinate $(\frac{X5+X6}{2}, \frac{Y6+Y7}{2}+1.3\times(\frac{Y6+Y7}{2}-Y5))$ (4.3)

P4 is defined by the coordinate $(X7+\frac{X7-X5}{2}, Y7)$ (4.4)

P5 is defined by the coordinate $(X4+(\frac{X4-X5}{2}), Y4)$ (4.5)

P6 is defined by the coordinate $(\frac{X2+X3}{2}, \frac{Y2+Y3}{2.5})$ (4.6)

where
X1, Y1 is the coordinate of left eye outer landmark point (LE1),
X2, Y2 is the coordinate of left eye inner landmark point (LE2),
X3, Y3 is the coordinate of right eye outer landmark point (RE1),
X4, Y4 is the coordinate of right eye inner landmark point (RE2),
X5, Y5 is the coordinate of nose landmark point (N),
X6, Y6 is the coordinate of mouth right landmark point (M1),
X7, Y7 is the coordinate of mouth left landmark point (M2),

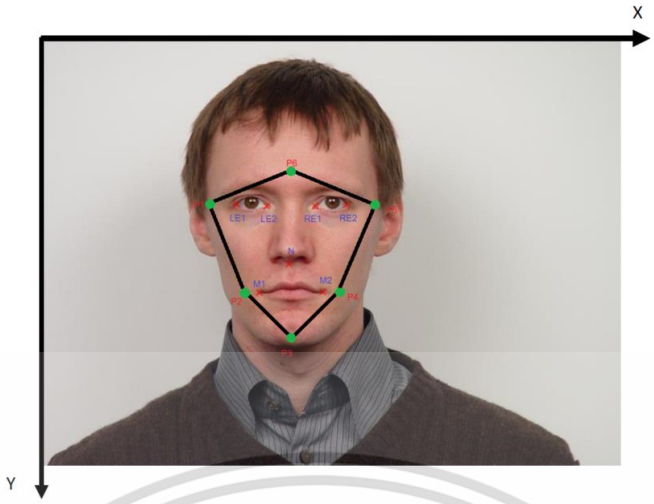
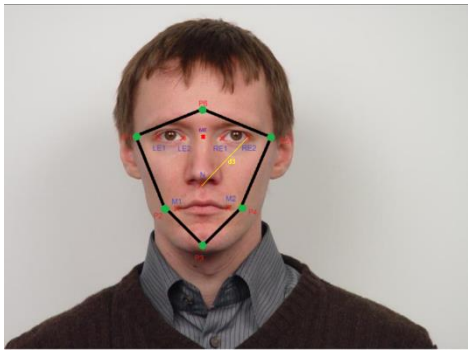


Fig 4.8 Facial border points which are defined by facial land marks [12]

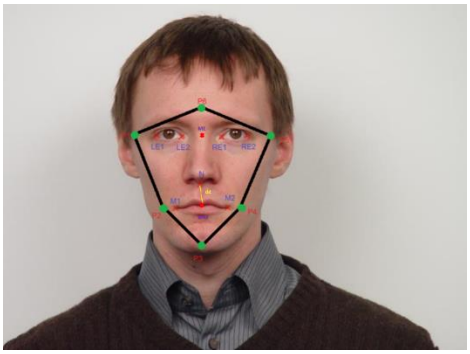
After obtaining facial border images, we used two features for face recognition. The first feature is the distance between facial landmarks formed as the seven feature vectors, and this vectors are defined in Figure 4.8 (a)-(h). The second feature is the distance map error worked out from alignment between reference image and inquiry image. Eventually, total error is computed from multiple results of error in the first feature and the second feature as shown in equation (4.8).

(a) distance from outer corner of right eye to central of mouth landmarks

(b) distance from central of inter of both of eyes to nose landmark

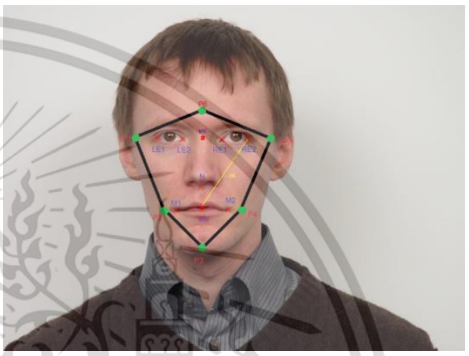


(c) distance form outer corner of left eye to central of mouth landmark



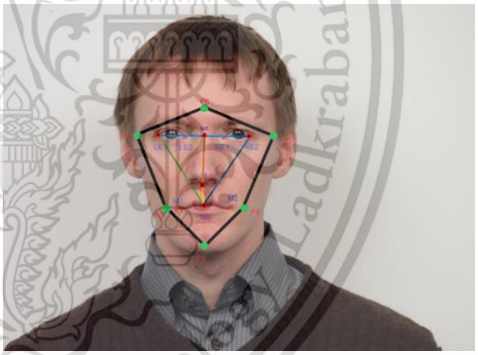
(d) distance from nose landmark to central of mouth landmark

(e) distance from outer corner of right eye to central of mouth landmark



(f) distance from outer right eye to central of mouth landmark

(g) distance from outer right eye to outer left eye landmark.



(h) all facial distances

Figure 4.9 Distance of Each facial landmarks

Where,

d1 is the distance from outer corner of right eye to central of mouth landmarks
d2 is the distance from central of inter of both of eyes to nose landmark
d3 is the distance form outer corner of left eye to central of mouth landmark
d4 is the distance from nose landmark to central of mouth landmark
d5 is the distance from outer corner of right eye to central of mouth landmark
d6 is the distance from outer right eye to central of mouth landmark
d7 is the distance from outer right eye to outer left eye landmark.

For reference face image, $d_{n1}, d_{n2}, d_{n3}\ and\ d_{n4}$, these are the 4 biggest of {*d1, d2, d3, d4, d5, d6* and *d7*}

For inquiry face image,
$d_{n1}^{'}, d_{n2}^{'}, d_{n3}^{'}\ and\ d_{n4}^{'}$, these are the 4 biggest of $\{d_1^{'}, d_2^{'}, d_3^{'}, d_4^{'}, d_5^{'}, d_6^{'}$ and $d_7^{'}\}$

$$|D| = \frac{|d_{n1}-d_{n1}^{'}|+|d_{n2}-d_{n2}^{'}|+|d_{n3}-d_{n3}^{'}|+|d_{n4}-d_{n4}^{'}|}{4} \quad (4.7)$$

For $|D|$, we compute absolute difference of feature and estimate the average of the 4 biggest error. Eventually, the multiple of $|D|$ and distance map error is used to be feature to identify person.

$$\text{Total error} = |D|.|Distance\ map\ error| \quad (4.8)$$

Chapter 5
Experiment and result

We tested our algorithm using the face images from DTU database [17] and KMITL database. We performed two experiments including intra-subject and inter subject experiment. The intra-subject experiment is used to verify the robustness of the purpose technique with facial image from different geometric transformation. The inter-subject experiment is to apply our technique in face recognition.

5.1 Inter-subject experiment using DTU database

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
S1	**68.8**	169.1	137.8	183.7	117.8	118.2	146.9	304.8	143.2	257.2	200.6	216.8
S2	234.71	**86.9**	258.6	320.4	185.4	240.8	294.5	470.5	162.57	339.7	319.2	374.4
S3	213.9	261.3	**37.67**	247.7	209.3	195.6	173.4	294.3	220.7	291.5	243.5	263.4
S4	150	201.9	142.49	**55.28**	171.76	126.2	115.8	185	150.6	174.7	155	142
S5	292.7	266.4	258.7	417.2	**67.4**	268.85	309.2	467.8	199.5	386.5	271.2	465.9
S6	203.4	228.7	186.1	263.6	144.3	**80**	183.447	298.05	152.3	270.87	252.2	312.9
S7	148.6	209.5	122.9	157.3	164.9	152.7	**58.9**	152.5	151.7	150.9	136.7	155.6
S8	239	422.9	183	185.1	250.7	234.8	139.5	**40.1**	260.9	159	151.3	255.5
S9	216.7	209.7	188.7	288.5	146.6	174.3	255	327.7	**89.7**	253.9	217.9	352.7
S10	204.4	304.9	174.2	250.8	203.9	204.9	134	173.3	171.5	**46.5**	165.4	237.6
S11	498.9	516	384.1	506.5	318.7	464.9	385.2	526.2	422.3	465.3	**94.4**	451.2
S12	173.3	254.2	151.3	142	208.3	194.9	128.5	182.7	207.9	182.4	150.3	**92.4**

Table 5.1 shows the result of total error obtaining from equation (3.9) of face images from DTU database. In this experiment, we test with 12 different frontal hum faces (144 images). The diagonal elements of this table yield the minimum total error as it is corresponding person testing.

5.2 Inter-subject experiment using KMITL database.

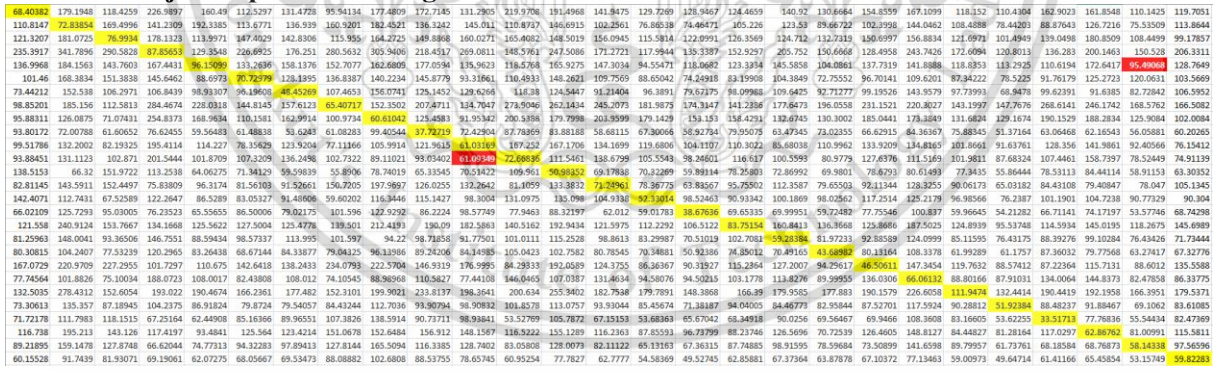


Table 5.2 shows the result of total error obtaining from equation (3.9) of face images from our database. In this experiment, we test with 27 different frontal hum faces (729 images). The diagonal elements of this table yield the minimum total error as it is corresponding person testing. However, in this experiment missed matching of two images, thus accuracy is approximately 93 %.

5.3 Intra-Subject experiment

	First subject (Inquiry image 1)	First subject (Inquiry image 2)	First subject (Inquiry image 3)	First subject (Inquiry image 4)	Standard deviation (STD)
First subject (Reference image)	68.8078	37.6713	55.2819	67.3765	5.0464

Table 5.3 shows the result of total error obtaining from equation (3.9) of face images from DTU database. In this experiment, we test with five face images as the same person.

	Second subject (Inquiry image 1)	Second subject (Inquiry image 2)	Second subject (Inquiry image 3)	Second subject (Inquiry image 4)	Standard deviation (STD)
Second subject (Reference image)	37.6713	43.7351	48.1136	46.2886	4.5164

Table 5.4 shows the result of total error obtaining from equation (3.9) of face images from DTU database. In this experiment, we test with five face images as the same person.

	Third subject (Inquiry image 1)	Third subject (Inquiry image 2)	Third subject (Inquiry image 3)	Third subject (Inquiry image 4)	Standard deviation (STD)
Third subject (Reference image)	67.3765	85.006	61.2081	70.5413	5.0447

Table 5.5 shows the result of total error obtaining from equation (3.9) of face images from DTU database. In this experiment, we test with five face images as the same person.

Results table 5.3 – 5.5, these represent the intra-subject experiments, and the results show that the average of standard deviation of total error for these experiments is roughly 4.869.

5.4 Sample of some raw face images

Figure 5.1 shows some frontal face images from DUT database. These images are used for testing with the algorithm in this research. There are 12 people (12x12=144 face images) in this database.

Figure 5.1 some instances of raw face images from DTU database for inter-subject experiment

Figure 5.2 shows some frontal face images from KMITL database. These images are used for testing with the algorithm in this research. There are 27 people (27x27=729 face images) in this database.

Figure 5.2 some instances of raw face images from KMITL database for inter-subject experiment

Figure 5.3 shows frontal face images of the same person from DUT database. These images are used for intra-subject experiment with the algorithm in this research.

Figure 5.3 some instances of raw face images from DTU database for intra-subject experiment

5.5 Some examples of testing on the algorithm of this thesis from DTU database

5.5.1 The first subject

Figure 5.4 shows ROI of facial border images from DTU database before applying image registration. These ROI images are defined from facial landmarks.

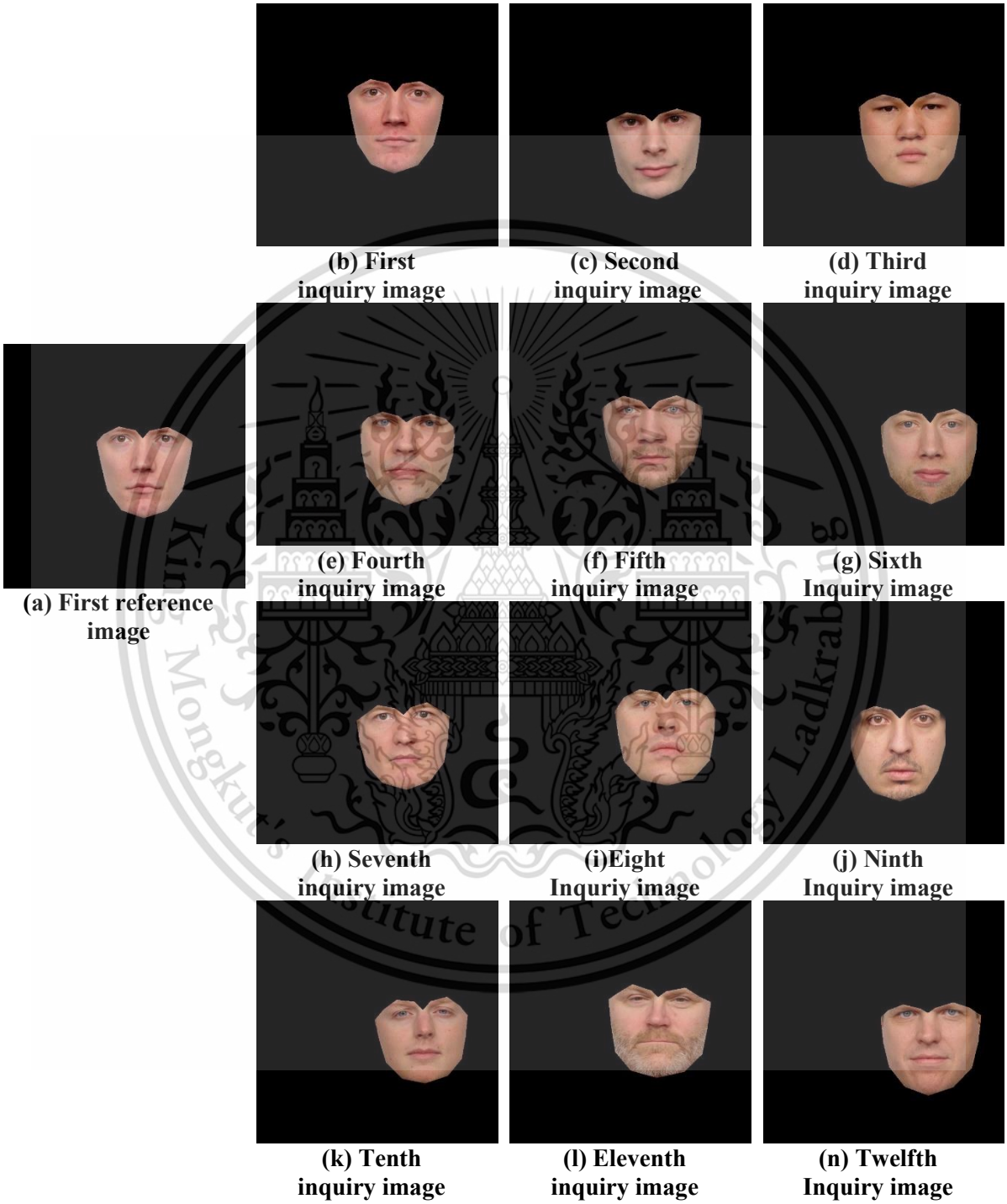


Figure 5.4 ROI of facial border images before applying image registration

Figure 5.5 shows ROI of facial border images from DTU database after applying geometric transformations.

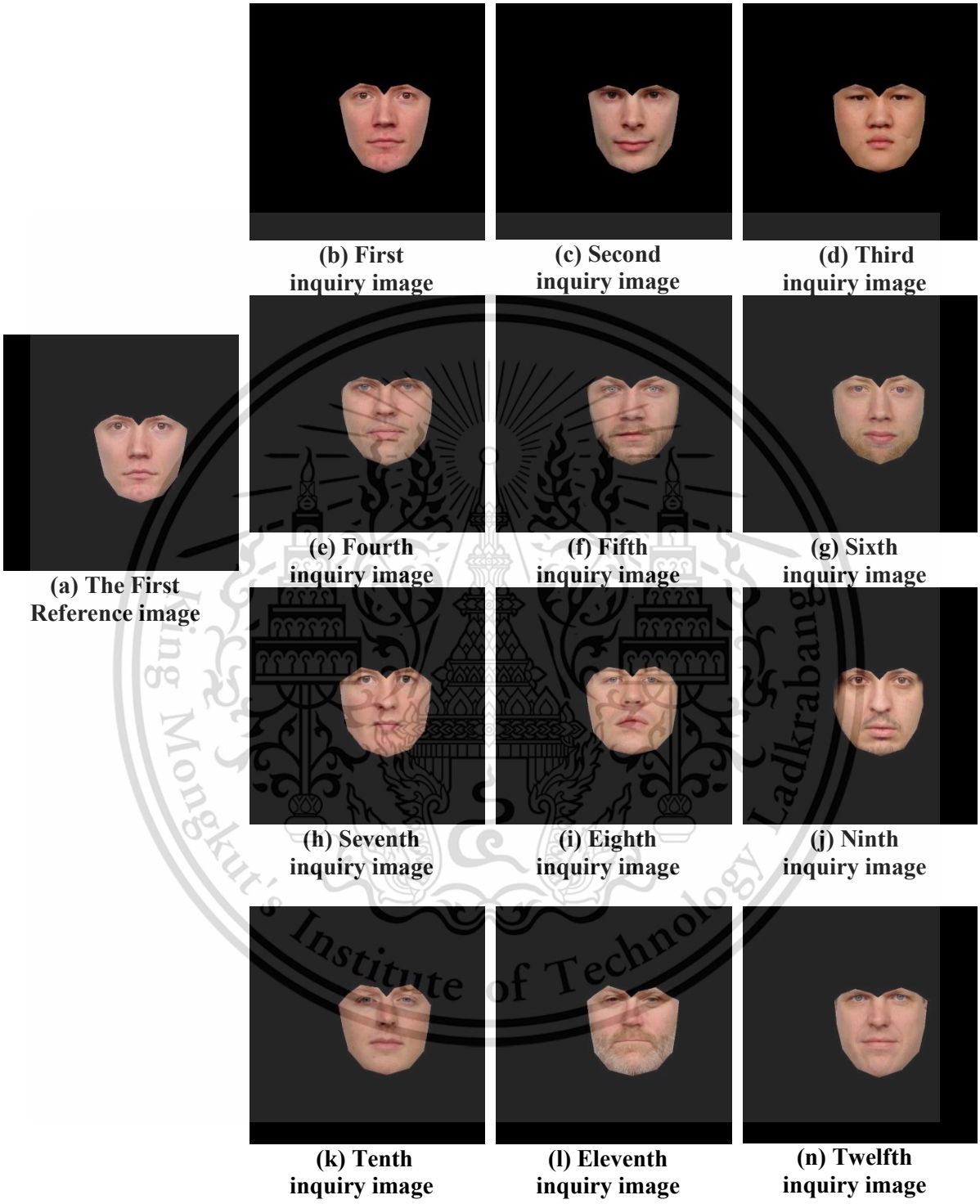


Figure 5.5 ROI of facial border images after applying image registration

Figure 5.6 represents alignment of reference image and inquiry images after transforming into edge images. The red edge image is the reference image and the greed edge image is the inquiry image.

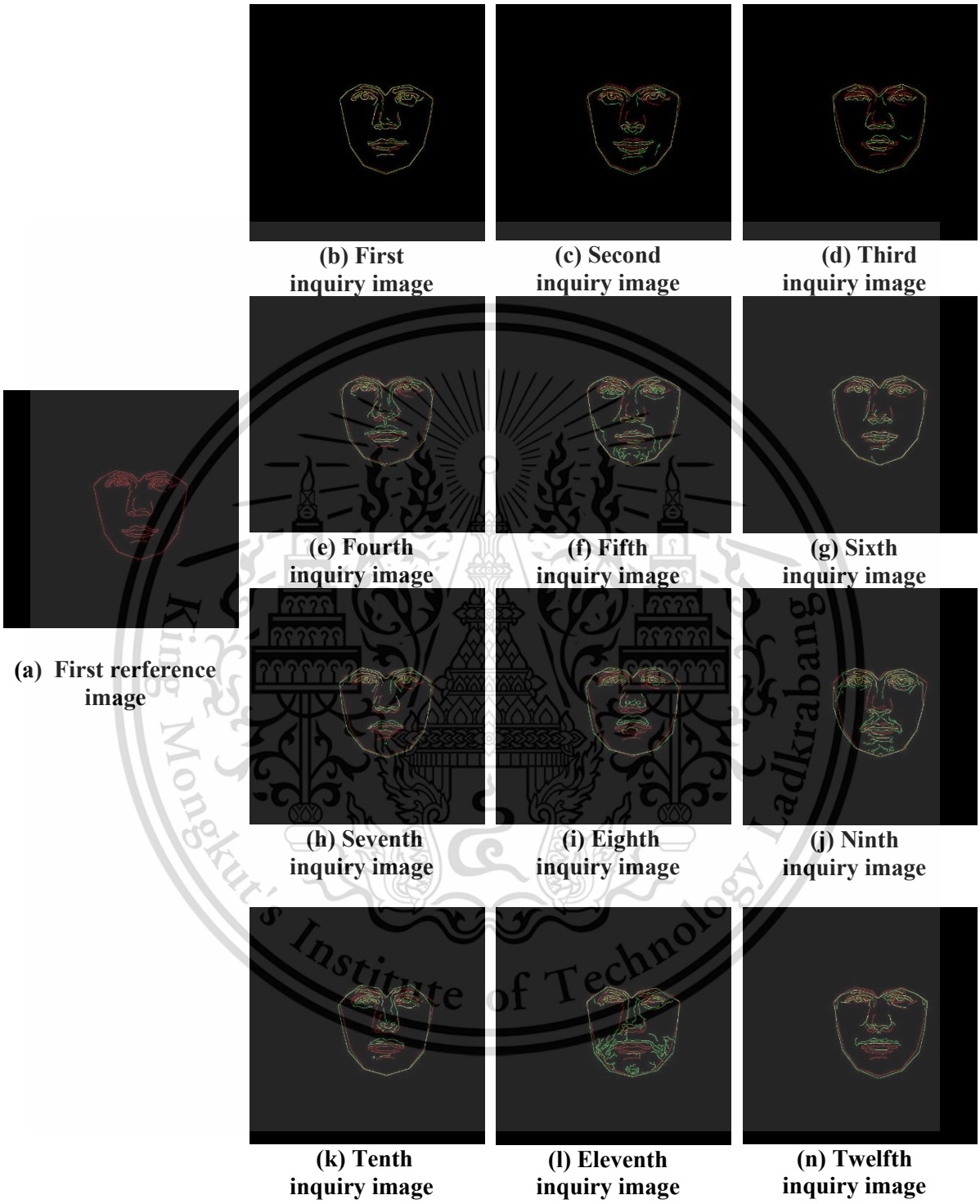


Figure 5.6 alignment of reference image and inquiry images after transforming into edge images

5.5.2 The second subject

Figure 5.7 shows ROI of facial border images from DTU database before applying image registration. This images. These ROI images are defined from facial landmarks.

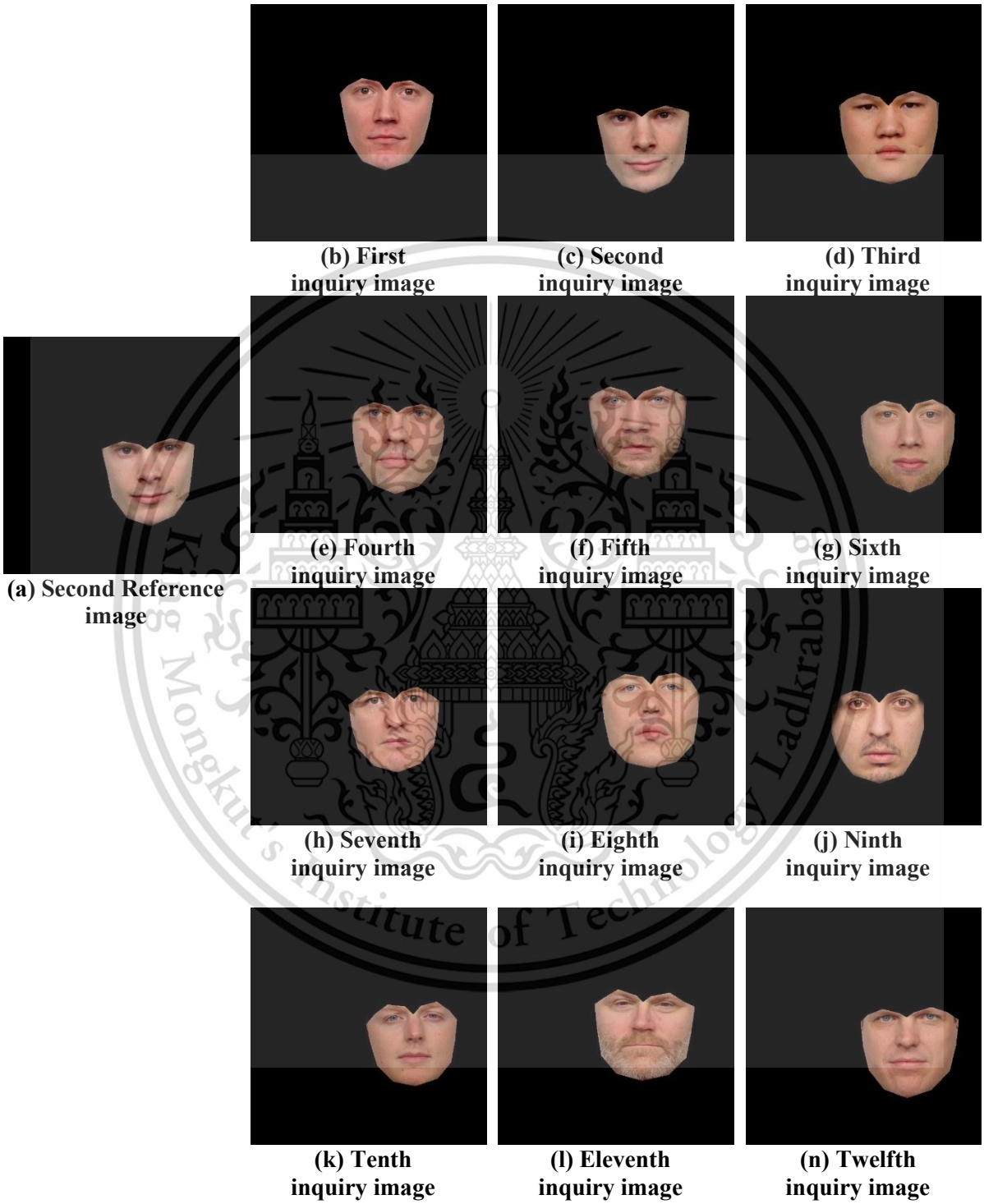


Figure 5.7 ROI of facial border images before applying image registration

Figure 5.8 shows ROI of facial border images from DTU database after applying geometric transformations.

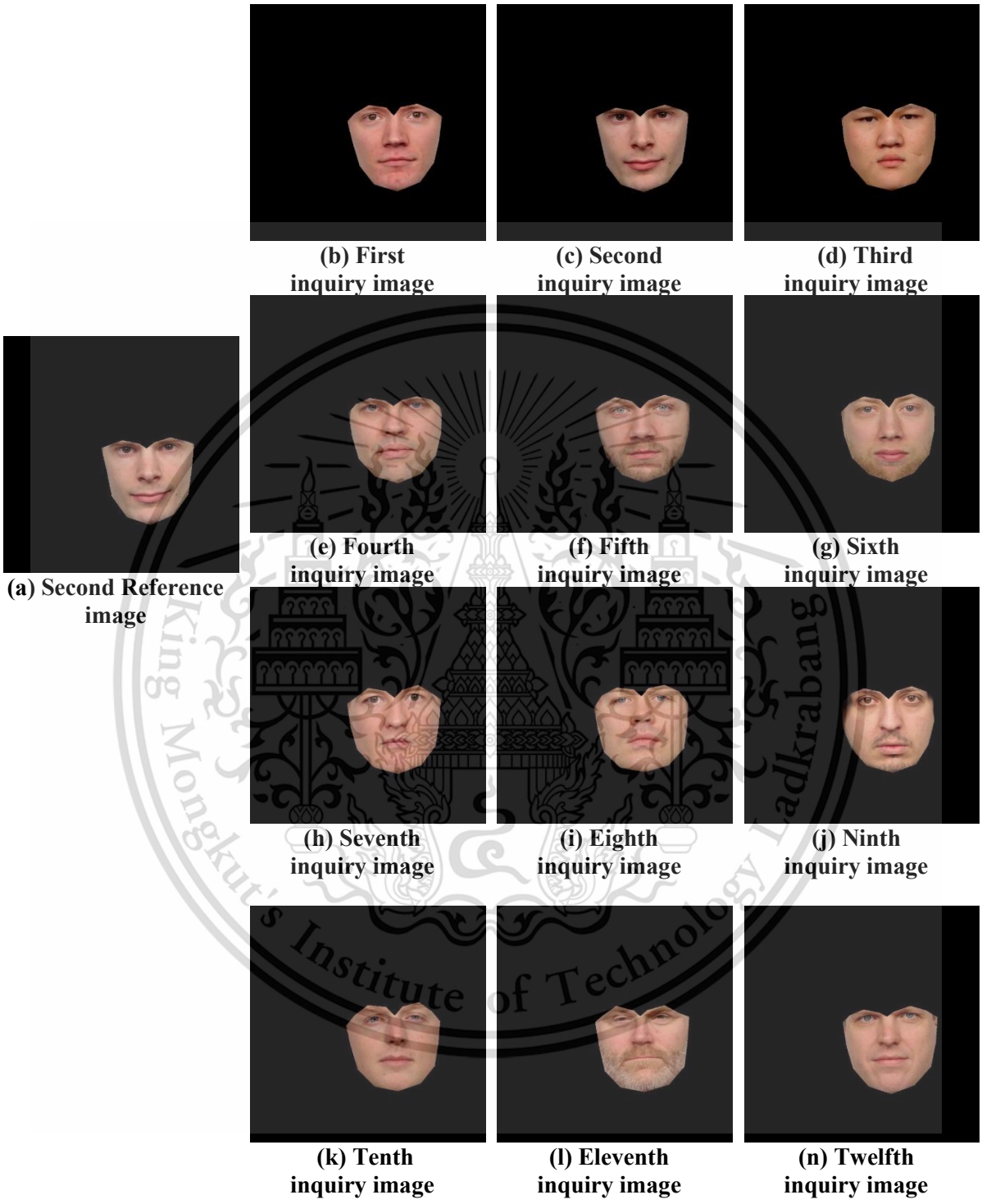


Figure 5.8 ROI of facial border images after applying image registration

Figure 5.9 represents alignment of reference image and inquiry images after transforming into edge images. The red edge image is the reference image and the greed edge image is the inquiry image.

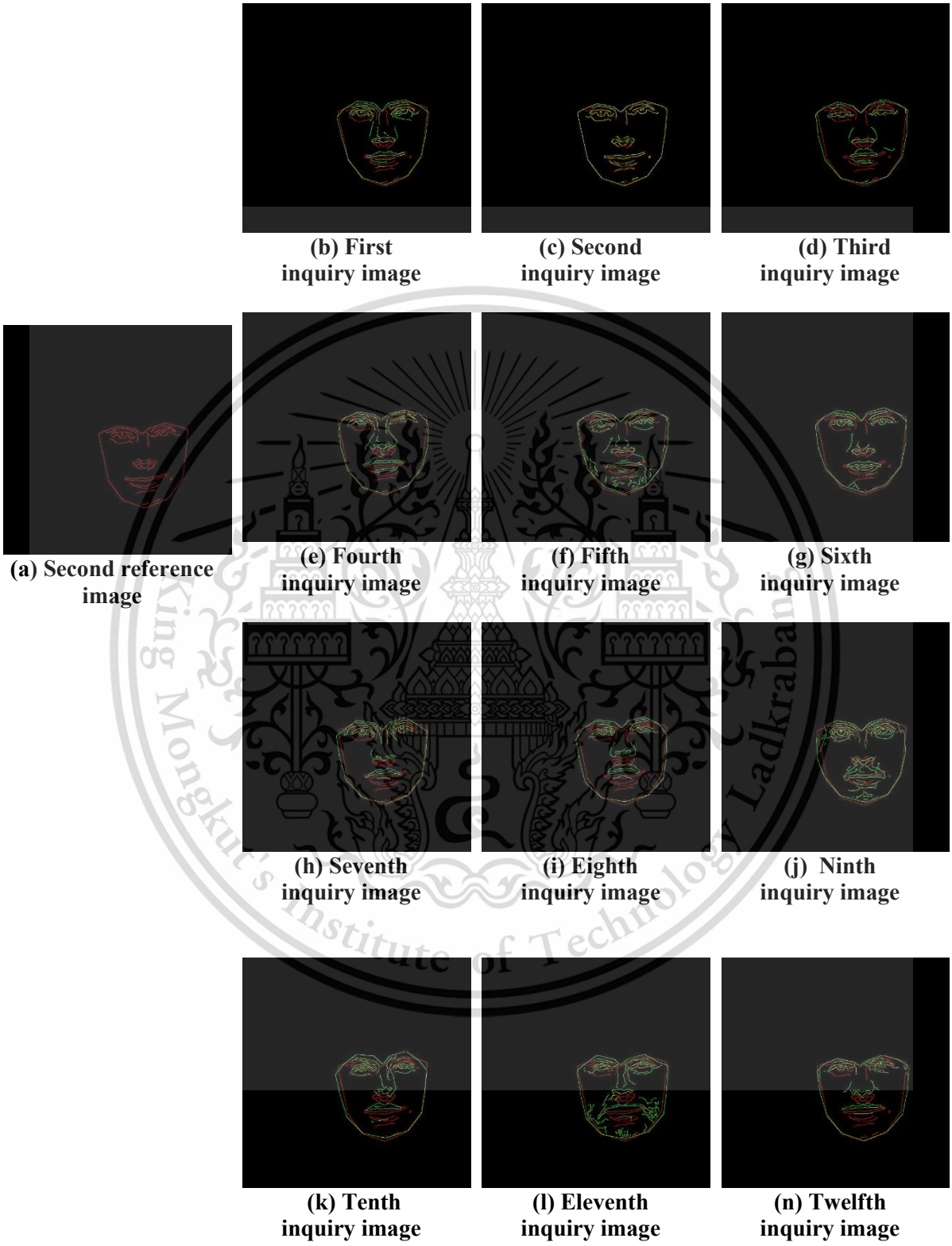


Figure 5.9 alignment of reference image and inquiry images after transforming into edge images

5.5.3 The third subject

Figure 5.10 shows ROI of facial border images from DTU database before applying image registration. This images. These ROI images are defined from facial landmarks.

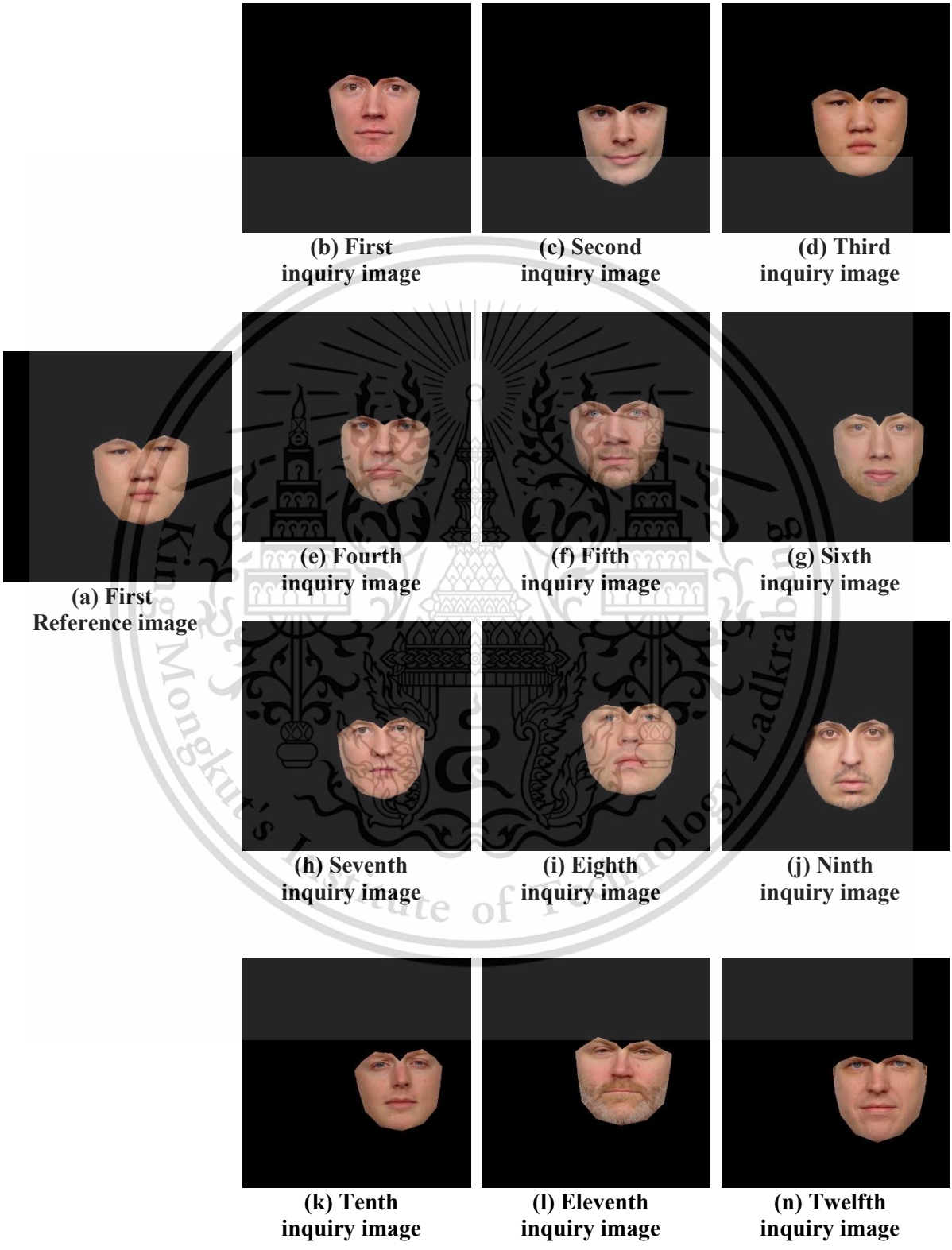


Figure 5.10 ROI of facial border images before image registration

Figure 5.11 shows ROI of facial border images from DTU database after applying geometric transformations.

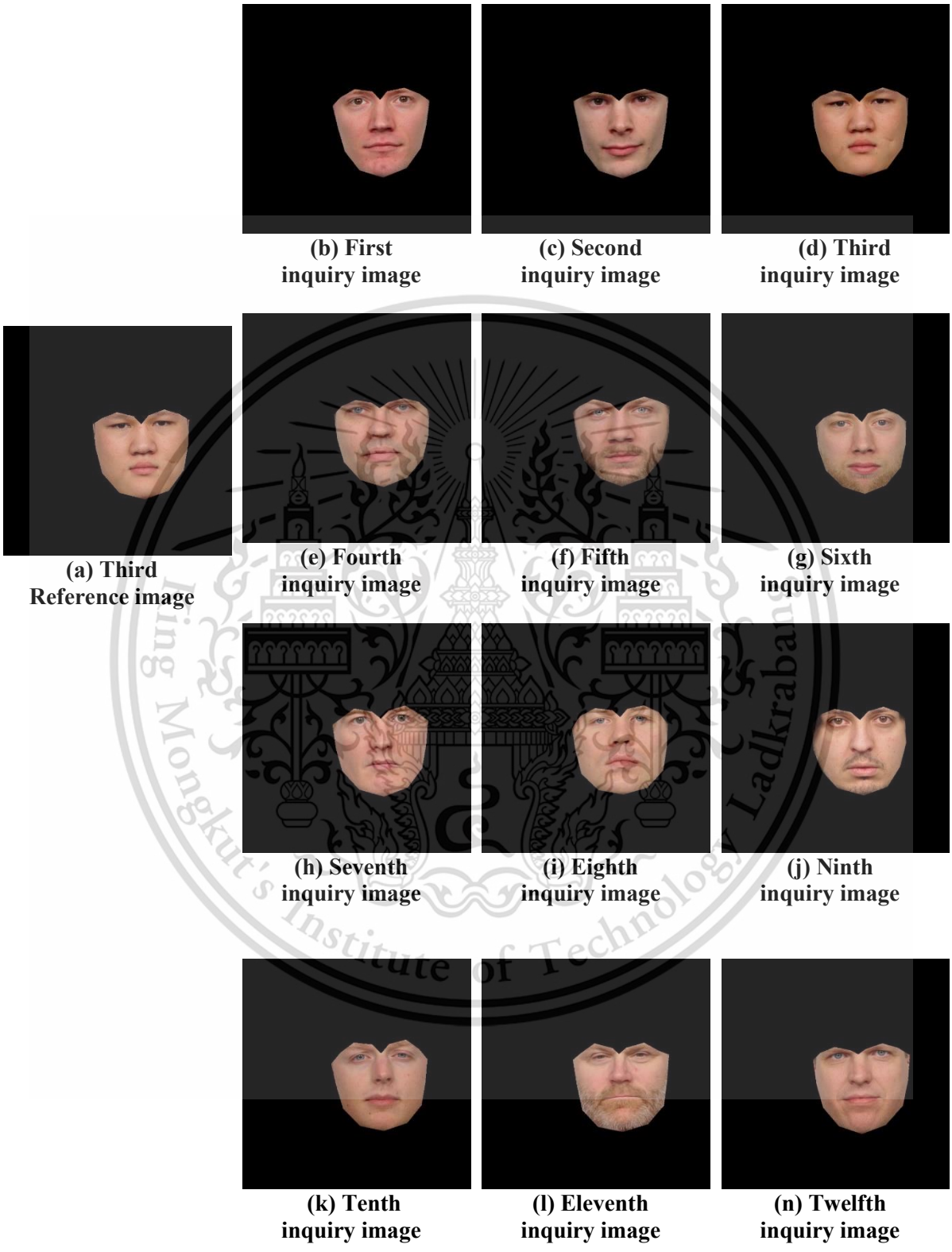


Figure 5.11 ROI of facial border images after image registration

Figure 5.12 represents alignment of reference image and inquiry images after transforming into edge images. The red edge image is the reference image and the greed edge image is the inquiry image.

Fig 5.12 alignment of reference image and inquiry images after transforming into edge images

5.5.4 Some examples of testing on the algorithm of this thesis from KMITL database

Figure 5.13 shows ROI of facial border images from KMITL database before applying image registration. This images. These ROI images are defined from facial landmarks.

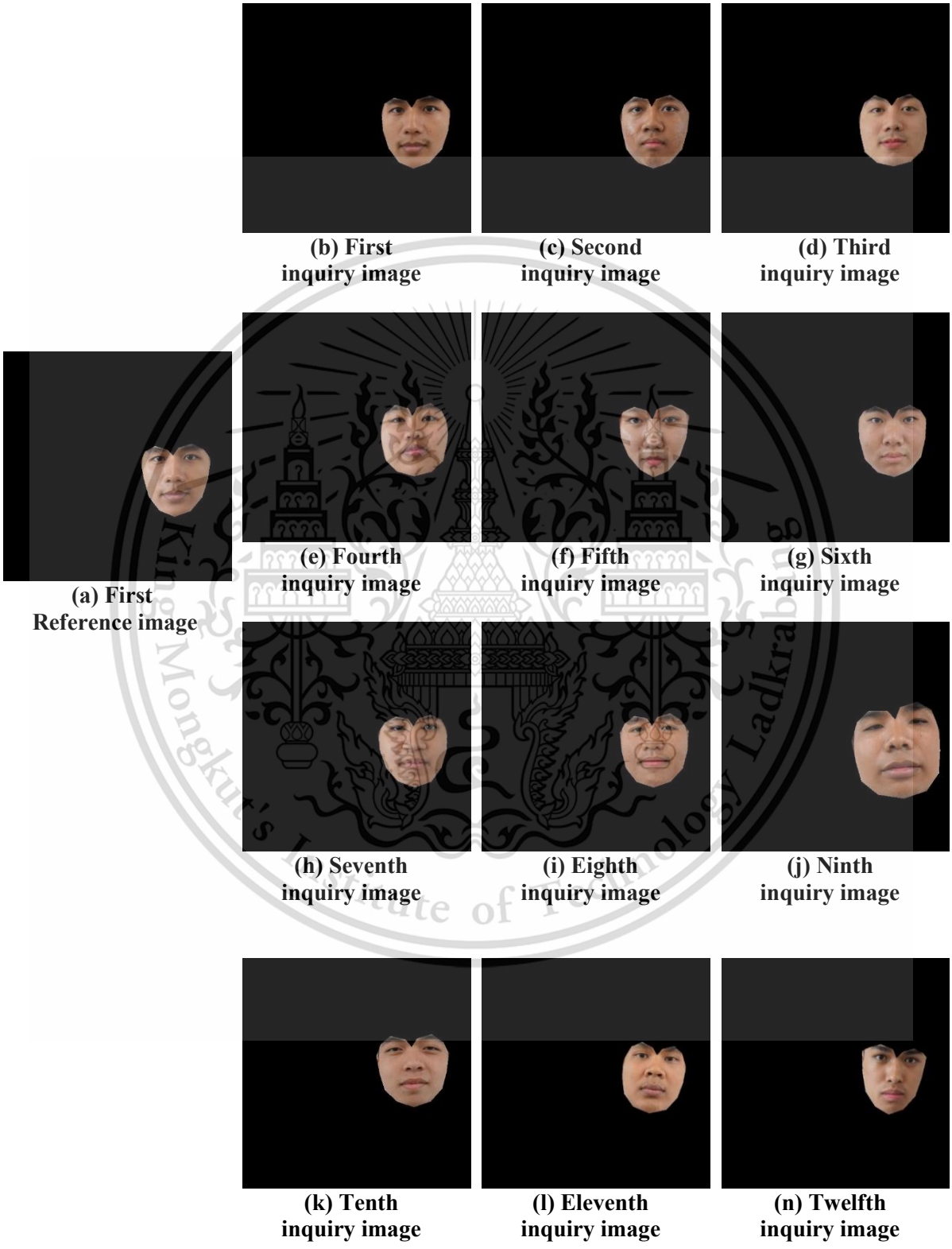


Figure 5.13 ROI of facial border images before image registration

Figure 5.14 shows ROI of facial border images from KMITL database after applying geometric transformations.

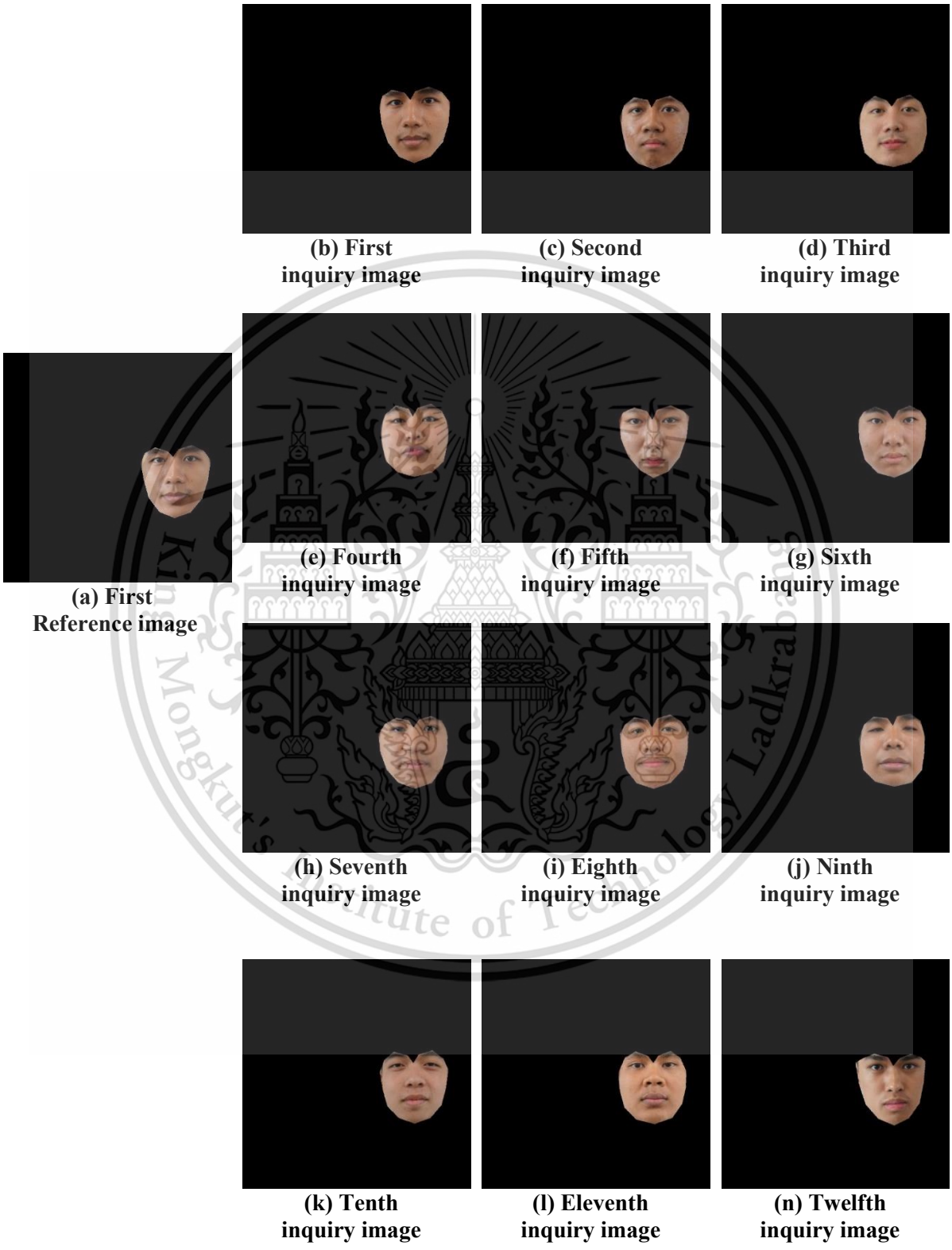


Figure 5.14 ROI of facial border images after image registration

Figure 5.15 represents alignment of reference image and inquiry images after transforming into edge images. The red edge image is the reference image and the greed edge image is the inquiry image.

Figure 5.15 alignment of reference image and inquiry images after transforming into edge images

Chapter 6
Conclusion and discussion

6.1 Conclusion

Face recognition technology has come a long way in the last twenty years ago. Nowadays, machines are able to automatically verify identity information for secure transaction, for surveillance and security tasks, and for access any control system. Face recognition is one of the several techniques for recognizing people. There are several methods that can be used for that purpose. In conventional methods are using PCA or Eigen-face. In addition, there are various techniques which are more simple to implement and good performance. The face recognition based facial landmarks extraction algorithm is one of those algorithms. As we show this algorithm has very good performance and accuracy.

This research presents a novel technique for face recognition related to facial landmark extraction, canny edge detection, distance map error and distance of each facial landmarks. The projection technique is applied to extract facial landmarks after that these landmarks are used to defined facial border for obtaining face region only. The canny edge is used to get edge image from face region image. distance map error is then performed with edge image. After that, distance of each facial landmarks will be computed. Eventually, the multiple of distance map error and absolute difference of the average of the four biggest error of distance of each facial landmarks are used to be feature of person identification. The result of this algorithms is successful and very promising.

6.2 Discussion

Although this research has very promising result, it should have more extensive experiments. The future work, deep learning will be integrated with this research to implement face recognition by accurately. Furthermore, all facial landmarks also can be applied with facial expression recognition.

References

[1] Definition of Biometric [Online]. Available : http://searchsecurity.techtarget.com/definition/biometrics

[2] Maur´ıcio P. Segundo et al. , “Real-time 3D face recognition using low-cost acquisition device”, A thesis submitted to the Department of Informatics, University Federal do Parana.

[3] G.M. Beumer, Q. Tao, A.M. Bazen, and R.N.J. Veldhuis, “A landmark paper in face recognition”, University of Twente, EEMSC, Signals and Systems P.O. box 217, 7500 AE, Enschede, The Netherlands.

[4] Eigen face image based on alignment of face image [Online]. Available: http://www.kixor.net/school/2008spring/comp776/assn3

[5] The relationship between the intensity values and the different shades of gray [Online]. Available: http://what-when-how.com/introduction-to-video-and-image-processing/image-acquisition-introduction-to-video-and-image-processing-part-2/

[6] Intensity Image define gray level [Online]. Available: https://edoras.sdsu.edu/doc/matlab/toolbox/images/intro4.html

[7] Principle of digital image processing [Online]. Available: http://what-when-how.com/introduction-to-video-and-image-processing/image-acquisition-introduction-to-video-and-image-processing-part-1/

[8] Histogram equalization [Online]. Available: https://towardsdatascience.com/histogram-equalization-5d1013626e64

[9] Thresholding an Image [Online]. Available: http://www.ni.com/tutorial/2916/en/

[10] Face Detection Using Haar Cascades [Online]. Available: https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html

[11] Radon Transformation [Online]. Available: https://www.mathworks.com/help/images/radon-transform.html

[12] Affine Transformation [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/rotate.htm

[13] Canny edge detection [Online]. Available: https://docs.opencv.org/3.3.0/da/d22/tutorial_py_canny.html

[14] R.P. Wood, Sr.R. Cherry, and J.C. Mazziotta. “Rapid Automated Algorithm for Aligning and Reslicing Pet Images.” J Compute Assist Tomogr, vol. 16, no.4, pp.620-633, 1992.

[15] R.P.Woods, J.C. Mazziotta, and S.R. Cherry. “MRI-PET Registration with automated Algorithm,” J comput Assist Tomogr., vol. 17, no. 4, pp. 536-546,1993.

[16] iterative image registration [Online]. Available:
https://www.mathworks.com/help/images/registering-multimodal-mri-images.html

[17] DTU database of face images [Online]
"http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=394"

Appendix A
Geometric transformation

The geometric transformation is a widely used technique to study common properties or feature between images using the same coordinate basis. In this chapter we depicted about Homogeneous coordinates and Affine transformation that related with this research.

A.1 Homogeneous Coordinates

Homogeneous Coordinates Homogeneous coordinates or projective coordinates, introduced by August Ferdinand Moebius in 1827 are a system of coordinates used in projective geometry. Projective geometry is a technique widely used in computer graphics and image processing to perform geometric transformation in a simple way using transformation matrices. It relies on adding a dimension to the working space to be able to perform transformations using matrices.

For an image, the working space is bi dimensional and will be extended to 3 dimensions by the use of homogeneous coordinates:

$$\mathbf{X} = \begin{pmatrix} x \\ y \end{pmatrix} \quad \Rightarrow \quad \mathbf{X_h} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{a.1}$$

In the case of three dimensional space, dimension will be increased to 4 with the same technique:

$$\mathbf{X} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \Rightarrow \quad \mathbf{X_h} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{a.2}$$

A.2 Affine Transformations

In geometry, an affine transformation is a function that maps an object from an affine space to another and which preserve structures. Indeed, an affine transformation preserves lines or distance ratios but change the orientation, size or position of the object. The set of affine transformation is composed of various operations. Translations which modify object position in the image. Homothetic Transformations composed of the contraction and dilatation of an object, both scaling operations. The transection (shear mapping) which modify position of an object. Rotation which allows to rotate an object according to it's axis. And a whole set of transformation produced by combining all of the previous.

The use of homogeneous coordinates is the central point of affine transformation which allow us to use the mathematical properties of matrices to perform transformations. So, to transform an image, we use a matrix: $T \in M4(R)$ providing the changes to apply.

$$\mathbf{T} = \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & T_x \\ a_{21} & a_{22} & a_{23} & T_y \\ a_{31} & a_{32} & a_{33} & T_z \\ \hline P_x & P_y & P_z & 1 \end{array}\right] \quad \text{(a.3)}$$

The vector $[T_x, T_y, T_z]$ represents the translation vector according the canonical vectors. The vector $[P_x, P_y, P_z]$ represents the projection vector on the basis. The square matrix composed by the a_{ij} elements is the affine transformation matrix.

In image processing due to the bi dimensional nature of images we will only use a reduced version of the previous matrix:

$$\mathbf{T} = \left[\begin{array}{cc|c} a_{11} & a_{12} & T_x \\ a_{21} & a_{22} & T_y \\ \hline P_x & P_y & 1 \end{array}\right] \quad \text{(a.3)}$$

We will also consider that our Projection vector : $[P_x, P_y]$ is the null vector. It's important to notice that this matrix form is strictly the same as the one given in the book. The mathematical proof is obvious and is based on the matrix product properties.

A.2.1 Forward Transformation

The technique that consists of scanning all the pixels in the original image, and then computing their position in the new image is called the Forward Transformation. This technique has a lot of issues that we will present when computing the different types of transformations on an image. In each section we will present the forward mapping and explain why in most cases it's not working well.

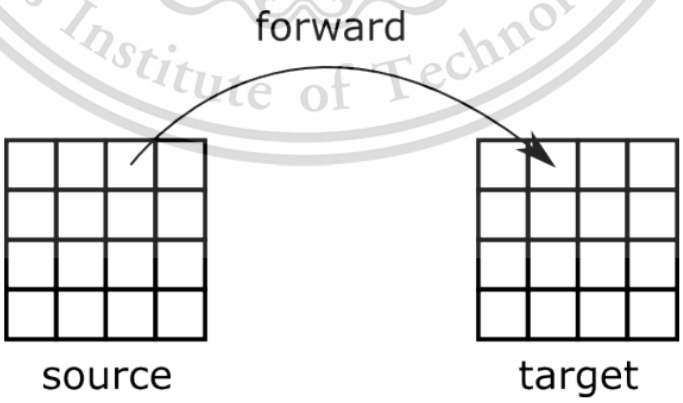


Figure A.1 Forward Transformation

A.2.2 Inverse Transformation

The Reverse Transformation is the inverse of the forward transformation. Indeed, this technique consists in scanning each pixel of the new image and apply the mapping to find the pixel in the original image. Then we need to use interpolation with the surrounding pixels to compute the new intensity

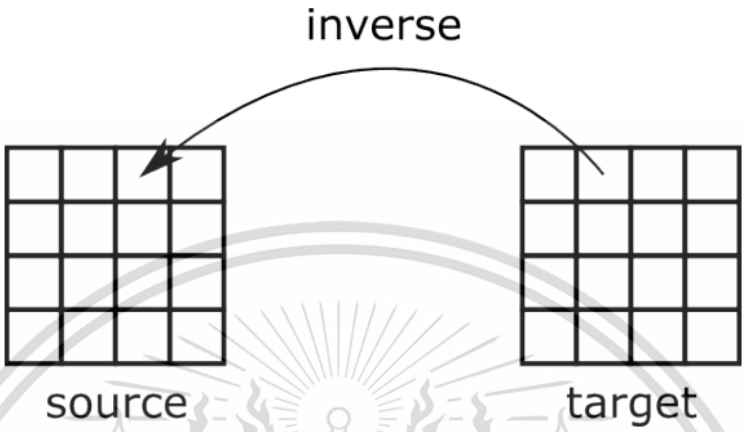


Figure A.2 Inverse Transformation

A.2.3 General Affine Transformation

The general affine transformation can be defined with only six parameters. Indeed if we list the ”degrees of freedom” or the possible parameters of the matrix we have:

θ : the rotation angle
Tx : the x component of the translation vector
Ty : the y component of the translation vector
Sx : the x component of the scaling vector
Sy : the y component of the scaling vector
shx : the x component of the shearing vector
shy : the y component of the shearing vector

This description can be reduced to 6 independent parameters. Indeed, it appears that the shearing y component can be perform by cascading a 90◦ rotation, an x shearing and again another rotation of −90◦.

Autobiography

Name Mr.Aniwat Juong **Birthdates** January 30, 1994
Address 77 M.20, WAING CHAI district, Chiang Rai province.
Email: Aniwat_BMEKMITL@hotmail.com

Education Bachelor of Engineering in Biomedical Engineering
Department of Biomedical, Faculty of Engineering
King Mongkut's Institute of technology Ladkrabang

Publications

[1] **A. Juhong**, C. Pintavirooj "**Filtering ECG Using IIR filter on Arduino**", The 7th Biomedical Engineering International Conference BMEiCON 2014 Fukuoka Japan, 26-28 November 2014.

[2] **A. Juhong**, C. Pintavirooj "**Simulated CT scanner for Educational Purpose**", The 9th Biomedical Engineering International Conference BMEiCON 2016 Luang Prabang Laos, December 6-8, 2016.

[3] **A. Juhong**, C.Pintavirooj, "**Face Recognition Based on Facial Landmark Detection**", The 10th Biomedical Engineering International Conference BMEiCON 2017 Hokkaido Japan, August 31 – September 2 , 2017.

[4] **A. Juhong**, T. Treebupachatsakul, C.Pintavirooj, "**Smart Eye Tracking System**", International Workshop on Advanced Image Technology 2018 (IWAIT 2018) Chiang Mai Thailand, January 7-10, 2018.

[5] **A. Juhong**, B. Purahong, S. Suwan, C. Pintavirooj, "**Biometrics Based on Facial Landmark with Application in Person Identification**", World Congress on Medical Physic and Biomedical Engineering 2018, Prague, Czech Republic, June 3-8, 2018.

[6] **A. Juhong**, W. Piyawattanametha, "**An E. coli screening machine in water via fluorescence detection**", Biomedical Engineering International Conference (BMEiCON 2018) Chiang Mai, Thailand, November 21-24, 2018.