# โปรแกรมตรวจสอบลายนิ้วมือ

# FINGERPRINT VERIFIER SOFTWARE

นางสาวภัทรพร ยอดมณฑป
นายภาคภูมิ ศิริวุฒิ



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

# FINGERPRINT VERIFIER SOFTWARE

PATRAPORN YODMONDHOP

PARKPOOM SIRIVUTE

A THESIS SUBMITTED IN PARTIAL FUFILMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN ELECTRONICS ENGINEERING

SCHOOL OF GRADUATE STUDIES

KINGMONKUT'S INSTITUTE OF TECHNOLOGY LADGRABANG

ปริญญานิพนธ์ ปีการศึกษา 2550
ภาควิชา อิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง โปรแกรมตรวจสอบลายนิ้วมือ

ผู้จัดทำ

1. นางสาวภัทรพรพร ยอดมณฑป 47010551
2. นายภาคภูมิ ศิริวุฒิ 47010562

____________________อาจารย์ที่ปรึกษา

(ผศ.ดร.ยุทธนา คิดใจเดียว)

วันที่___/___/___

| | |
|---|---|
| **หัวข้อวิทยานิพนธ์** | โปรแกรมตรวจสอบลายนิ้วมือ |
| **นักศึกษา** | นางสาวภัทรพร ยอดมณฑป 47010551 |
| | นายภาคภูมิ ศิริวุฒิ 47010562 |
| **ปริญญา** | วิศวกรรมศาสตร์บัณฑิต |
| **สาขาวิชา** | วิศวกรรมอิเล็กทรอนิกส์ |
| **พ.ศ.** | 2551 |
| **อาจารย์ที่ปรึกษาวิทยานิพนธ์** | ผศ.ดร.ยุทธนา คิดใจเดียว อาจารย์ที่ปรึกษา |


## บทคัดย่อ

เครื่องสแกนลายนิ้วมือจะส่งข้อมูลของลายนิ้วมือไปยังคอมพิวเตอร์โดยผ่านพอร์ต I/O เพื่อที่จะนำไปใช้ในกระบวนการตรวจสอบเปรียบเทียบกับข้อมูลเดิมที่จัดเก็บไว้ใน Database โดยที่ภายในคอมพิวเตอร์จะมีซอฟแวร์ที่ได้ทำการสร้างขึ้นไว้จากโปรแกรม LabView สำหรับติดต่อกับผู้ใช้ได้สะดวก ไม่ซับซ้อนและใช้การได้ง่าย ซึ่งลำดับแรกรูปที่ได้จากการสแกน จะถูกส่งเข้าสู่กระบวนการ Preprocessing เป็นการปรับปรุงภาพที่ได้ให้เกิดความชัดมากขึ้น จากนั้นจะเข้าสู่ขั้นตอน Post-processing ในขั้นตอนนี้เป็นขั้นตอนการประมวลภาพในรูปแบบที่เราได้กำหนดไว้ ได้แก่ การทำให้บางแล้วเปลี่ยนภาพที่ได้ให้อยู่ในเวกเตอร์ทิศทาง การหาจุดกึ่งกลางและเดลต้า รวมถึงการหาจุดปลายและจุดแยกของลายนูน เพื่อเก็บไว้ทำการเปรียบเทียบ(Matching) กับลายนิ้วมือที่เก็บไว้ใน Database ว่ามีลักษณะเฉพาะเหมือนตรงกันหรือไม่ และวิธีในการเปรียบเทียบ(Matching)ในอัลกอริทึมนี้จะใช้วิธีของ Delaunay (วิธีสามเหลี่ยม) ขั้นตอนนี้จะใช้โปรแกรม MatLab ซึ่งเป็นโปรแกรมทางคณิตศาสตร์ในการช่วยคำนวณ พล็อตสามเหลี่ยม และประมวลผลการตรวจสอบลายนิ้วมือ

| | | |
|---|---|---|
| **Thesis Title** | Fingerprint Software Verifier | |
| **Student** | Ms.Patraporn Yodmondhop | 47010551 |
| | Mr.Parkpoom Sirivute | 47010562 |
| **Degree** | Bachelor of Engineering | |
| **Program** | Electronics of Engineering | |
| **Year** | 2008 | |
| **Thesis Advisor** | Assoc. Prof.Dr. Yuttana Kitjaidure | |

## Abstract

The project presents the fingerprint verification. The fingerprint data are fed into the computer via input/output port (I/O port) to compare with the stored data kept in Database by a program called "LabVIEW" as being not too complicated for the user to use it. First, the fingerprint data will be enhanced in the preprocessing to improve clarity, then in post-processing, the input will be thinning and transformed into directional vectors. After that, the core and delta including the specific Ridges End and Ridge Bifurcation are extracted. Finally, the MATLAB program is used for the matching by the Delaunay method.

**กิตติกรรมประกาศ**

โครงงานนี้สำเร็จลงได้ด้วยดีก็ด้วยความสนับสนุนช่วยเหลือและกำลังใจจากหลาย ๆ ฝ่ายด้วยกัน โดยการให้ข้อมูล อุปกรณ์สนับสนุน และชี้แนะข้อคิดเห็นต่างๆ ซึ่งมีประโยชน์อย่างมากในการดำเนินการ ไม่ว่าจะเป็นจากพี่ ๆ เพื่อน ๆ ในภาคอิเล็กทรอนิกส์และภาคอื่นๆ ที่ให้ความช่วยเหลือทุก ๆ ด้าน พี่ ๆ ปริญญาโทที่ช่วยกันให้คำปรึกษาและให้กำลังใจในการทำงาน โดยเฉพาะอย่างยิ่ง ผศ. ดร. ยุทธนา คิดใจเดียว ที่ให้คำปรึกษาตลอดเวลา และอาจารย์ทุก ๆ ท่านที่ให้คำปรึกษาเป็นอย่างดี

ผู้จัดทำจึงขอขอบคุณทุก ๆ ท่านมา ณ โอกาสนี้

ผู้จัดทำ

# สารบัญ

# สารบัญรูปภาพ

## สารบัญตาราง

# บทที่ 1
# บทนำ

โครงงานเครื่องสแกนลายนิ้วมือนี้ได้สร้างขึ้นเพื่อสแกนลายนิ้วมือและสามารถเปรียบเทียบลายนิ้วมือได้อย่างถูกต้อง โดยอุปกรณ์นี้สามารถจะนำไปประยุกต์ใช้งานรักษาความปลอดภัยต่างๆได้ เช่น ระบบประตูรักษาความปลอดภัย ระบบคอมพิวเตอร์รักษาความปลอดภัย

## 1.1. รายละเอียดโดยย่อของโครงงาน

โครงงานนี้เป็นโครงงานเครื่องสแกนลายนิ้วมือโดยใช้ไอซี MBF200 เป็นอุปกรณ์ในการสแกนลายนิ้วมือและใช้โปรแกรมบนคอมพิวเตอร์ทำการประมวลผลเปรียบเทียบลายนิ้วมือ โดยตัวอุปกรณ์สามารถติดต่อและส่งข้อมูลผ่านทางคอมพิวเตอร์โดยใช้พอร์ตอนุกรม ส่วนโปรแกรมบนคอมพิวเตอร์ใช้โปรแกรมแมทแล็ป (MATLAB) ในการเปรียบเทียบลายนิ้วมือ

## 1.2. วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาหลักการวิเคราะห์และเปรียบเทียบลายนิ้วมือ
2) เพื่อกำหนดอัลกอริทึมที่ง่ายต่อการนำไปสร้างเป็นเครื่องสแกนลายนิ้วมือ
3) เพื่อสร้างเครื่องสแกนลายนิ้วมือที่ใช้ในระบบรักษาความปลอดภัย
4) เพื่อพัฒนาเทคโนโลยีในระบบรักษาความปลอดภัย

## 1.3. ขอบเขตของโครงงาน

สร้างเครื่องสแกนลายนิ้วมือที่สามารถสแกนลายนิ้วมือบนโปรแกรมที่สร้างได้ และโดยสามารถเปรียบเทียบลายนิ้วมือได้อย่างถูกต้อง

## 1.4. ประโยชน์และผลที่คาดว่าจะได้รับจากโครงงานนี้

เนื่องจากในปัจจุบันการรักษาความปลอดภัยเป็นเรื่องสำคัญในชีวิตประจำวัน และลายนิ้วมือก็เป็นที่นิยมนำมาใช้ในการรักษาความปลอดภัย เพราะสะดวกในการตรวจสอบและมีความถูกต้องแม่นยำ ซึ่งอุปกรณ์ที่ได้สร้างขึ้นนี้สามารถนำไปประยุกต์ใช้งานได้หลายรูปแบบ

# บทที่ 2
# ความรู้เบื้องต้นเกี่ยวกับลายนิ้วมือ

เมื่อเราหงายมือขึ้นและสังเกตไปที่นิ้วมือแต่ละนิ้ว จะพบว่านิ้วนั้นมีลักษณะเป็นลายอยู่ เมื่อมองเข้าไปใกล้ๆ จะเห็นว่าลายนิ้วมือที่มีลักษณะประกอบไปด้วย 2 ส่วนใหญ่ คือ ส่วนที่เป็นลายนูน (Ridges) และร่อง (Valleys or Furrows) วิ่งวนเป็นลักษณะส่วนโค้งไปรอบๆช่วงปลายนิ้วมือสลับกันไประหว่างร่องและลายนูน ทำให้เกิดเป็นลายขึ้นมา ลายดังกล่าวนี้จะมีส่วนของจุดที่เป็น จุดเฉพาะ (Singular Point) คือ จุดกึ่งกลาง (Core) และ สันดอน (Delta) รวมทั้งลักษณะสำคัญ (Minutiae) ซึ่งก็คือ จุดปลายของลายนูน (Ridge Ending) และจุดแยกของลายนูน (Ridge Bifurcation) รูปแบบของลายนิ้วมือนี้จะมีอยู่หลายรูปแบบด้วยกัน ดังจะกล่าวในรายละเอียดต่อไปนี้

## 2.1. ลักษณะของจุดต่างๆบนลายนิ้วมือ

### 2.1.1. จุดกึ่งกลาง (Core)

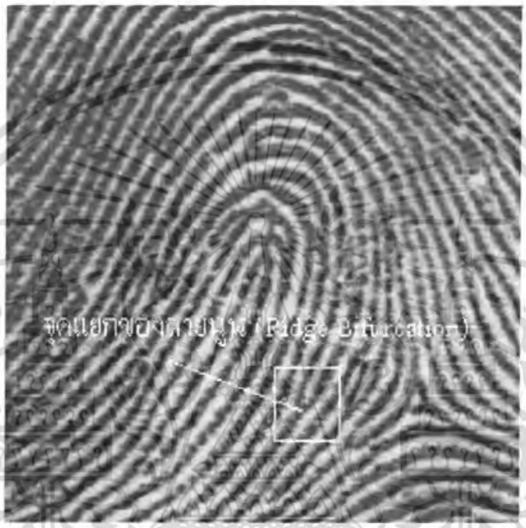
จุดบนเส้นโค้งของลายนูนต้องมีลักษณะเป็นโค้งขึ้น และเริ่มที่จะกลับตัว หรือเริ่มที่จะกลายเป็นโค้งลง แล้วจึงวิ่งสวนทางกลับกันกับในตอนแรกก่อนที่จะโค้งขึ้น เส้นโค้งลายนูนนี้ต้องเป็นโค้งลายนูนที่อยู่ชั้นในที่สุดของบริเวณที่มีลักษณะรูปแบบเป็นจุดกึ่งกลางนั้น เป็นลักษณะของเส้นโค้งที่โค้งขึ้นมาซ้อนๆกันหลายๆเส้น ในลักษณะของรูปมัดหวายจากโค้งวงกว้างลดลงมาเรื่อยๆ จนโค้งเล็กสุด (เส้นที่อยู่ในที่สุดอาจไม่เป็นเส้นโค้งก็ได้) ลักษณะของรูปแบบเส้นโค้งที่อยู่ในบริเวณที่มีลักษณะรูปแบบเป็นจุดกึ่งกลาง

รูปที่ 2.1 แสดงบริเวณที่มีลักษณะรูปแบบเป็นจุดกึ่งกลาง (Core)

2.1.2. ลักษณะสำคัญ (Minutiae)

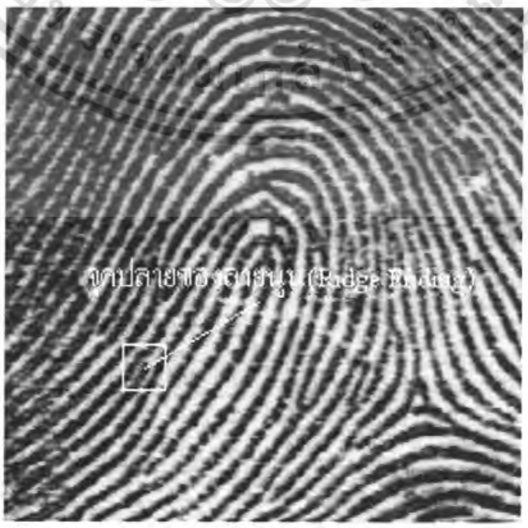
เป็นลักษณะสำคัญในรูปลายนิ้วมือ ประกอบด้วยส่วนประกอบสำคัญ 2 ชนิด ได้แก่

- จุดแยกของลายนูน (Ridge Bifurcation) เป็นจุดที่อยู่บนลายนูน ที่เกิดจากการแยกจากหนึ่งเส้นทางเป็นสองเส้นทาง ดังแสดงในรูปที่ 2.2




รูปที่ 2.2 แสดงจุดแยกของลายนูน (Ridge Bifurcation)

- จุดปลายของลายนูน (Ridge Ending) เป็นจุดที่อยู่บนลายนูน ในบริเวณปลายสุดของลายนูน ดังแสดงไว้ในรูปที่ 2.3




รูปที่ 2.3 แสดงจุดปลายของลายนูน (Ridge Ending)

## 2.2. ชนิดและรูปแบบลายนิ้วมือ

รูปแบบลายนูนในภาพลายนิ้วมือที่จับกลุ่มกันเป็นรูปแบบ สามารถแบ่งได้หลายชนิด ดังต่อไปนี้

2.2.1. กลุ่มเส้นโค้ง

ลักษณะของรูปแบบในภาพลายนิ้วมือมีลักษณะการวิ่งของลายนูนจากลักษณะที่ขนานกับพื้นราบ แล้วพุ่งโค้งขึ้น แล้วจึงมีการวิ่งในลักษณะขนานกับพื้นราบอีกครั้ง

รูปแบบของกลุ่มเส้นโค้งนั้นมี 2 รูปแบบ คือ

- กลุ่มเส้นโค้งราบ (Arch)
- กลุ่มเส้นโค้งกระโจม (Tented Arch)

โดยกลุ่มเส้นโค้งทั้งสองจะแตกต่างกันที่ลักษณะความสูงของส่วนโค้ง ดังตัวอย่างในรูปที่ 2.4 และ 2.5

รูปที่ 2.4 แสดงกลุ่มโค้งราบ (Arch)

รูปที่ 2.5 แสดงลักษณะของรูปแบบเส้นโค้งกระโจม (Tented Arch)

2.2.2. กลุ่มมัดหวาย (Loop)

1) กลุ่มมัดหวายเดี่ยว (Single Loop)

- กลุ่มมัดหวายเอียงขวา (Right Loop)

มีลักษณะของการวิ่งของลายนูนจากบริเวณด้านซ้ายมาที่บริเวณกลางนิ้วมือแล้ววิ่งโค้งขึ้นไปด้านบน แล้วจึงโค้งลงและวิ่งวนกลับไปตามทางเดิมที่ได้วิ่งมา เมื่อสังเกตด้วยตาเปล่าจะเห็นว่าเมื่อมีการจับกลุ่มเป็นรูปแบบคล้ายกับมัดหวาย แล้วจะเอนเอียงไปทางด้านขวา จึงได้เรียกว่า "กลุ่มมัดหวายเอียงขวา" และมีสันดอนอยู่ทางด้านขวาของรูปแบบ

รูปที่2.6 แสดงลักษณะของกลุ่มมัดหวายเอียงขวา (Right Loop)

- กลุ่มมัดหวายเอียงซ้าย (Left Loop)

รูปแบบจะคล้ายลักษณะของกลุ่มมัดหวายเอียงขวา คือ มีการวิ่งของลายนูนจากทางด้านขวาที่บริเวณตรงกลางของนิ้วมือ จากนั้นจะวิ่งขึ้นไปด้านบนแล้วโค้ง และวิ่งกลับไปตามทางเดิมที่วิ่งมา เมื่อทำการสังเกตจะพบว่ามีลักษณะรูปแบบคล้ายกับมัดหวาย และรูปแบบดังกล่าวจะมีการเอียงตัวไปทางด้านซ้าย เราจึงเรียกว่า "กลุ่มมัดหวายเอียงซ้าย" ปกติมักจะมีสันดอนอยู่ทางด้านซ้ายของรูปแบบเช่นกัน

รูปที่2.7 แสดงลักษณะของกลุ่มมัดหวายเอียงซ้าย (Left Loop)

2) กลุ่มมัดหวายคู่ (Double Loop)

เป็นกลุ่มของรูปแบบที่เกิดจากการรวมกันของกลุ่มมัดหวายเดี่ยวสองกลุ่มรวมตัวกันเป็นรูปแบบเดียว โดยลักษณะการวิ่งของลายนูนอาจมาจากทางเดียวกันแล้วรวมตัวกันเป็นลักษณะของกลุ่มมัดหวายเดี่ยวสองกลุ่ม โดยที่กลุ่มหนึ่งจะมีรูปแบบเป็นลักษณะของ โค้งขึ้นและกลุ่มมัดหวายอีกกลุ่มหนึ่งจะมีลักษณะโค้งลง เมื่อรวมกันเป็นกลุ่มมัดหวายคู่รูปแบบดังกล่าวอาจมีการเอียงซ้าย หรือเอียงขวา

รูปที่ 2.8 แสดงลักษณะรูปแบบของกลุ่มมัดหวายคู่ (Double Loop)

2.2.3. กลุ่มก้นหอย (Whorl)

ลักษณะการวิ่งของลายนูนมีรูปแบบเป็นลักษณะเส้นโค้งเป็นรูปวงกลมจากวงใหญ่ค่อยๆเล็กลงไปเรื่อยๆ จนเหลือวงกลมวงเล็กที่สุด

รูปที่ 2.9 แสดงลักษณะรูปแบบของกลุ่มก้นหอย (Whorl)

2.2.4. กลุ่มซับซ้อน (Accidental Whorl)

ลายนิ้วมือที่มีลักษณะพิเศษที่ไม่จัดเข้าเป็นลายนิ้วมือชนิดใดโดยเฉพาะ ประกอบด้วยลายนิ้วมือ 2 แบบมาผสมกัน และมีสันดอน 2 สันดอน หรือมากกว่า เช่นกรณีที่ไม่สามารถเข้ากับลายนิ้วมือกลุ่มที่กล่าวมาแล้วข้างต้นไม่ได้เลย โดยมีความยุ่งเหยิง และเป็นรูปแบบที่ไม่แน่นอน

รูปที่ 2.10 แสดงลักษณะรูปแบบกลุ่มซับซ้อน (Accidental Whorl)

# บทที่ 3 ***

## MBF200

## Solid State Fingerprint Sensor

### 3.1. คุณสมบัติ (Feature)

- มีความละเอียด 500 dpi
- มีขนาด 1.28 cm x 1.50 cm
- มีจำนวนแถว 300 แถว จำนวนหลัก 256 หลัก(0.76 เมกกะพิกเซล)
- ช่วงการทำงาน 3.3 V ถึง 5 V
- สแกนภาพเป็นแบบ Gray Scale
- ได้ข้อมูล 8 บิต จากการเปลี่ยน Analog เป็น Digital
- สามารถส่งข้อมูลออกได้ 3 ทาง คือ 1.USB Port 2. Serial Peripheral 3. Microprocessor
- ระบบสแกนลายนิ้วมือโดยอัตโนมัติ

3.1.1. การนำไปใช้งาน

- ในระบบรักษาความปลอดภัย
- ในบัตร Smart Card
- ระบุบุคคลสำหรับบัตร ATM และบัตรประชาชน
- ใช้เป็นรหัสผ่านใน Internet

3.1.2. การทำงาน

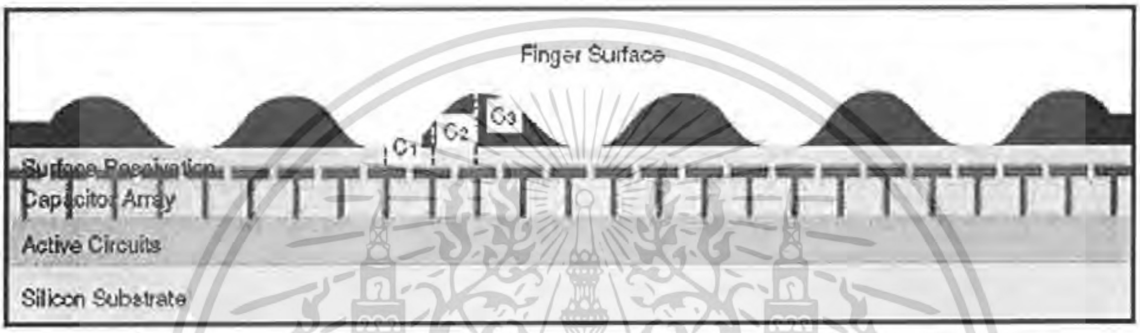
ตัวเซ็นเซอร์จะมีขนาด 300 แถว 256 หลัก ในหน้าตัดเซ็นเซอร์ การทำงานจะเกี่ยวข้องกับวงจร Sampling and Hold 2 ตัว โดยการทำงานจะจับภาพหรือส่งค่าทีละแถว เรียกว่า Row Capture ซึ่งจะประกอบด้วยการทำงาน 2 ขั้นตอน

ขั้นตอนแรกเซ็นเซอร์จะทำการกำหนดแถวที่จะทำการจับภาพและจะนำกระแสที่ขา VDD มาอัดประจุไปที่พื้นผิวของแถวที่จะจับภาพ และคงค่านี้ไว้ เรียกขั้นตอนนี้ว่า Precharge

*** บทที่ 3 อ้างอิงจากปริญญานิพนธ์ เรื่อง เครื่องสแกนลายนิ้วด้วยวิธีมุมและรัศมีที่จุดสำคัญ ปีการศึกษา 2549

ขั้นตอนที่ 2 แถวที่จับภาพจะทำการคายประจุคล้ายกับแหล่งจ่าย อัตราที่แผ่นเซ็นเซอร์จะใช้คายประจุจะกำหนดได้โดยรีจิสเตอร์ DCR (Discharge Current) หลังจากนั้นช่วงเวลาหนึ่ง (กำหนดโดยรีจิสเตอร์ DTR (Discharge Time)) และจะคงค่านี้ไว้ เรียกช่วงนี้ว่าช่วง Discharge

ผลต่างระหว่างค่า Precharge และ Discharge จะเป็นค่าของความจุไฟฟ้าในแต่ละเซลล์ หลังจากเสร็จสิ้นการ Row Capture แต่ละเซลล์ในแถวจะถูกกระบวนการ A/D เพื่อแปลงค่าเป็นตัวเลขส่งเป็นข้อมูลออกมา




รูปที่ 3.1 แสดงเทคโนโลยีและการทำงานของ Static-Capacity Semiconductor Sensor

จากรูปที่ 3.1 เป็นการมองจากภาพตัดขวาง จะเห็นได้ว่ามีตัวคาปาซิเตอร์จำนวนมากถูกจัดวางเรียงอยู่ที่ส่วนผิวชั้นบนของตัวชิป เมื่อนิ้วมือถูกวางลงบนผิว ค่าของคาปาซิเตอร์จะแปรเปลี่ยนไปตามสภาพความลึกตื้นของเส้นลายนิ้วมือ จากนั้นค่าแต่ละค่าของคาปาซิเตอร์จะถูกแปลงโดย A/D Converter 8 bits เป็นภาพขนาด 300x256 พิกเซล ซึ่งจะให้ความละเอียดของภาพลายนิ้วมือได้ 500 dpi

### 3.2. Pin Descriptions

**VDDA1, VDDA2** (ขา 1 และขา 7) เป็นขาจ่ายไฟใช้ +5V ให้กับส่วน Sensor

**VSSA1, VSSA2** (ขา 2 และขา 6) เป็นขากราวด์ให้กับส่วน Sensor

**VDD1, VDD2, VDD3** (ขา 25, 16 และ 39) เป็นขาจ่ายไฟ ใช้จ่าย +5V ให้กับส่วน Digital Logic และ I/O

**VSS1, VSS2, VSS3** (ขา 24, 15 และ 40) เป็นขากราวด์ใช้ +5V ให้กับส่วน Digital Logic และ I/O

**ISET** (ขา3) ต่อ R 200k ระหว่างขา VSSA1 กับ ISET เป็นขาอ้างอิงกระแสไฟฟ้าภายใน

**AIN** (ขา 4) เป็นขาที่ใช้รับสัญญาณจากภายนอกมาเปลี่ยนจาก Analog เป็น Digital โดยสามารถควบคุมได้จากการกำหนด AINSEL บิต ในรีจิสเตอร์ CTRLA ถ้าไม่ใช้ต้องต่อขานี้ลงกราวด์

**FSET** (ขา 5) ต่อขานี้กับ R 56k ลงกราวด์ เพื่อให้ Clock ภายในทำงาน

**XTAL1** (ขา 27) เป็นขาที่ใช้สำหรับต่อ Crystal เพื่อกำเนิด Clock แต่ถ้าใช้แหล่งกำเนิดภายนอกให้ต่อขานี้กับแหล่งกำเนิดสัญญาณนั้น

**XTAL2** (ขา 26) เป็นขาที่ใช้สำหรับต่อ Crystal เพื่อกำเนิด Clock แต่ถ้าใช้แหล่งกำเนิดภายนอกให้ขานี้ปล่อยลอย

**D[7:0]** (ขา 11-14, 17-20) เป็นขารับ-ส่งข้อมูล สามารถค้างค่าไว้ได้ ไม่ต้องใช้ขานี้ใน SPI, USB mode

**A0** (ขา 21) เป็น Address Input ถ้าA0 เป็น 0 (0 Volt) ใช้เป็นAddress Index Register ถ้า A0 เป็น 1 (5 Volt) ใช้เป็นData Buffer ไม่ต้องใช้ขานี้ใน SPI, USB mode

**$\overline{\text{RD}}$** (ขา 22) เป็นขาใช้สำหรับอ่านข้อมูลออกจากไอซีไปใช้งาน ทำงานเมื่อเป็น 0 (Active Low) ไม่ต้องใช้ขานี้ใน SPI, USB mode

**$\overline{\text{WR}}$** (ขา 23) เป็นขาใช้สำหรับเขียนข้อมูลเข้าไอซี ทำงานเมื่อเป็น 0 (Active Low) ไม่ต้องใช้ขานี้ใน SPI, USB mode

**$\overline{\text{CS0}}/\overline{\text{SCS}}$** (ขา 32) เป็นขาใช้สำหรับเลือก Mode การทำงาน ทำงานเมื่อเป็น0 (Active Low) ใช้ขานี้จะใช้คู่กับขา MODE1, MODE0

- MODE[1:0] = 00b (Microprocessor Bus Interface Mode)
  $\overline{\text{CS0}}/\overline{\text{SCS}}$ ให้เป็น 0 สำหรับเลือกเป็นอินพุทและให้ CS1 เป็น 1
- MODE[1:0] = 01b (SPI Slave Mode)
  $\overline{\text{CS0}}/\overline{\text{SCS}}$ ให้เป็น 0 สำหรับเลือกเป็นอินพุท ต่อ R-Pull Up ระหว่าง $\overline{\text{CS0}}/\overline{\text{SCS}}$ กับ VDD
- MODE[1:0] = 10b (USB Interface Mode, Using Internal ROM)
  ไม่มีหน้าที่ใช้งาน
- MODE[1:0] = 11b (USB Interface Mode, Using External ROM)
  $\overline{\text{CS0}}/\overline{\text{SCS}}$ สำหรับเลือกเป็นเอาท์พุทของไอซี โดยให้เป็น 0 สำหรับเลือก Serial ROM ต่อ R Pull Up ระหว่าง $\overline{\text{CS0}}/\overline{\text{SCS}}$ กับ VDD

**CS1/SCLK** (ขา 31) เป็นขาใช้สำหรับเลือก Mode การทำงาน ทำงานเมื่อเป็น 1 (Active High) ใช้ขานี้คู่กับขา MODE1, MODE0

- MODE[1:0] = 00b (Microprocessor Bus Interface Mode)
  CS1/SCLK ให้เป็น 1 สำหรับเลือกเป็นอินพุทและให้ $\overline{CS0}/\overline{SCS}$ เป็น 0
- MODE[1:0] = 01b (SPI Slave Mode)
  CS1/CLK ให้เป็น 0 สำหรับเลือกเป็นเอาท์พุท
- MODE[1:0] = 10b (USB Interface Mode, Using Internal ROM)
  ไม่มีหน้าที่ใช้งาน
- MODE[1:0] = 11b (USB Interface Mode, Using External ROM)
  CS1/SCLK ให้เป็น 0 สำหรับเลือกเอาท์พุท Serial Clock ต่อ R Pull Up ระหว่าง CS1/SCLK กับ VDD

**EXTINT** (ขา 30) เป็นขารับสัญญาณ Interrupt จากภายนอก

**$\overline{\text{INTR}}$** (ขา 28) เป็นขาส่งสัญญาณ

**$\overline{\text{WAIT}}$** (ขา 29) เป็นขาที่ส่งสัญญาณออกจากตัวไอซี ขณะที่อ่านค่าจากลายนิ้วมือจะมีสถานะเป็น 0 เมื่ออ่านค่าเสร็จ (แปลง A/D) มีสถานะเป็น 1

**MOSI** (ขา 33) เป็นขาเอาท์พุทของ SPI Mode โดยใช้คู่กับขา MODE1, MODE0 ไม่ต้องต่อขานี้ใน MCU Mode, USB Mode

- MODE[1:0] = 00b (Microprocessor Bus Interface Mode)
  ไม่มีหน้าที่ใช้งาน
- MODE[1:0] = 01b (SPI Slave Mode)
  ใช้เป็น Serial Data Intput ของไอซี
- MODE[1:0] = 10b (USB Interface Mode, Using Internal ROM)
  ไม่มีหน้าที่ใช้งาน
- MODE[1:0] = 11b (USB Interface Mode, Using External ROM)
  ใช้เป็น Serial Data Output ของไอซี

- **MISO** (ขา 34) เป็นขาอินพุทของ SPI Mode โดยใช้คู่กับขา MODE1, MODE0 ไม่ต้องต่อขานี้ใน MCU Mode, USB Mode
- MODE[1:0] = 00b (Microprocessor Bus Interface Mode)
  ไม่มีหน้าที่ใช้งาน
- MODE[1:0] = 01b (SPI Slave Mode)
  ใช้เป็น Serial Data Output ของไอซี
- MODE[1:0] = 10b (USB Interface Mode, Using Internal ROM)
  ไม่มีหน้าที่ใช้งาน
- MODE[1:0] = 11b (USB Interface Mode, Using External ROM)
  ใช้เป็น Serial Data Input ของไอซี

**P0** (ขา 9) เป็นขาเอาท์พุท ถูกควบคุมโดยบิต 0 ของ CTRLC register

**P1** (ขา 10) เป็นขาเอาท์พุท ถูกควบคุมโดยบิต 1 ของ CTRLC register

**DP** (ขา 38) เป็นขา Data USB D+ ใน USB Mode ต้องต่อ R 1.5k ระหว่างขา DP และ VDD3 (ใช้ 3.3V-3.6V) และต่อ R 43 โอห์ม อนุกรมกับขานี้ หรือต่อลงกราวนด์ใน MCU Mode, SPI Mode

**DM** (ขา 37) เป็นขา Data USB D- ใน USB Mode ต้องต่อ R 43 โอห์ม อนุกรมกับขานี้ หรือต่อลงกราวนด์ใน MCU Mode, SPI Mode

- MODE[1:0] (ขา 35, 36)
- MODE[1:0] = 00b (Microprocessor Bus Interface Mode)
- MODE[1:0] = 01b (SPI Slave Mode)
- MODE[1:0] = 10b (USB Interface Mode, Using Internal ROM)
- MODE[1:0] = 11b (USB Interface Mode, Using External ROM)

**TEST** (ขา 8) ใช้ทดสอบไอซี สำหรับโรงงานตรวจสอบเท่านั้น

**No Connect** (ขา 41-80)

### 3.3. Microprocessor Bus Interface

ในการทำงานโหมดนี้จะใช้ขา : D[7:0], A0, RD, WR, CS0, CS1, EXTINT, INTR, WAIT และขา XTAL1/XTAL2 เป็นขาสำหรับกำเนิดความถี่

ภายในไอซีตัวนี้มีขารับ-ส่งข้อมูล 8 ขา (D[7:0]) และมีขาสำหรับเลือกการทำงาน คือขา A0 เมื่อให้ขา A0 เป็น '0' เป็นการสั่งให้ชี้ Index Register ถ้าให้ขา A0 เป็น 1 เป็นการเขียนข้อมูลลง Index Register และมีขาอินพุทควบคุมอีก 4 ขา คือ CS0, CS1, RD, WR

นอกจากนี้ยังมีขาที่ใช้ในการควบคุมสถานะ 2 ขา คือ INTR, WAIT โดยที่ขา INTR ทำหน้าที่เป็นขาไว้ยืนยันเมื่อมีการเกิดการอินเตอร์รัพท์ขึ้น และขา WAIT จะเป็น low เมื่อ A/D Converter ถูกอ่านขณะมันกำลังทำงาน

ขา WAIT และ INTR ทั้งคู่จะเป็น High Impedance ขณะมันไม่ทำงาน และจะเป็น low เมื่อมัน Active

โหมด SPI และ USB จะไม่ได้ใช้เมื่อกำลังทำงานอยู่ในโหมด Microprocessor Bus Interface

ตารางที่ 3.1 Truth Table for the Microprocessor Bus Interface

| CS0 | CS1 | A0 | RD | WR | Mode | Data Lines |
|---|---|---|---|---|---|---|
| H | X | X | X | X | De-selected | High Impedance |
| X | L | X | X | X | De-selected | High Impedance |
| L | H | X | H | H | Standby | High Impedance |
| L | H | L | L | H | Read Index Register | Output |
| L | H | L | H | L | Write Index Register | Input |
| L | H | H | L | H | Read Index Register | Output |
| L | H | H | H | L | Write Index Register | Input |

### 3.4. Function Register

**RAH 0x00** เป็นบิตสูงของตัวชี้ตำแหน่งแถว ไว้ใช้สำหรับกำหนดแถวแรกที่จะทำการจับภาพในการ Read Mode Sub-Image

**RAL 0x01** เป็นบิตต่ำของตัวชี้ตำแหน่งแถว ไว้ใช้สำหรับกำหนดแถวแรกที่จะทำการจับภาพในการ Read Mode Sub-Image

**CAL 0x02** เป็นตัวชี้ตำแหน่งหลัก ไว้ใช้สำหรับกำหนดหลักแรกที่จะจับภาพในการ Read Mode Sub-Image

**REH 0x03** เป็นบิตสูงของตัวชี้ตำแหน่งแถว ไว้ใช้สำหรับกำหนดแถวสุดท้ายของการจับภาพในการ Read Mode Sub-Image

**REL 0x04** เป็นบิตต่ำของตัวชี้ตำแหน่งแถว ไว้ใช้สำหรับกำหนดแถวสุดท้ายที่จะทำการจับภาพในการ Read Mode Sub-Image

**CEL 0x05** เป็นตัวชี้ตำแหน่งหลัก ไว้ใช้สำหรับกำหนดหลักสุดท้ายที่จะจับภาพในการ Read Mode Sub-Image

**DTR 0x06** รีจิสเตอร์ควบคุมเวลาในการคายประจุ

**DCR 0x07** รีจิสเตอร์ควบคุมกระแสในการคายประจุ

**CTRLA 0x08** เขียนที่รีจิสเตอร์นี้เพื่อสั่งให้เริ่มการทำงาน การอ่านค่าจากรีจิสเตอร์นี้เพื่อรับค่าจาก A/D Converter และการเขียนรีจิสเตอร์นี้เพื่อกำหนดโหมดในการอ่านรูปภาพลายนิ้วมือ โดยมี 3 โหมด คือ

1) GETMG คือโหมดที่จะให้เซ็นเซอร์ทำการเริ่มต้นจับภาพตั้งแต่แถวแรก หลักแรก จนจบแถวสุดท้ายในคราวเดียวกัน
2) GETSUB คือโหมดที่ต้องกำหนดจุดเริ่มต้นและจุดสิ้นสุดในรีจิสเตอร์ RAH, RAL, CAL, REH, REL, CEL ก่อนทำการจับภาพในบริเวณที่กำหนด
3) GETROW คือโหมดที่จับภาพทีละแถว โดยต้องกำหนดแถวในรีจิสเตอร์ RAH, RAL เพื่อที่จะทำการจับภาพ

**CTRLB 0x09** เป็นรีจิสเตอร์ที่ทำหน้าที่ปิด/เปิดการทำงานของไอซี

**CTRLC 0x0A** เป็นตัวควบคุมขาเอาท์พุท P0 และ P1

**SRA 0x0B** เป็นรีจิสเตอร์ที่ Read Only ขานี้ทำหน้าที่เป็นเงาของรีจิสเตอร์ CTRLA เพื่อทำการตรวจสอบว่าเซ็นเซอร์กำลังทำงานอยู่ในโหมดอะไร

**PGC 0x0C** เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมอัตราขยายของเซ็นเซอร์

**ICR 0x0D** เป็นรีจิสเตอร์ที่ทำหน้าที่ปิด/เปิดการอินเตอร์รัพท์

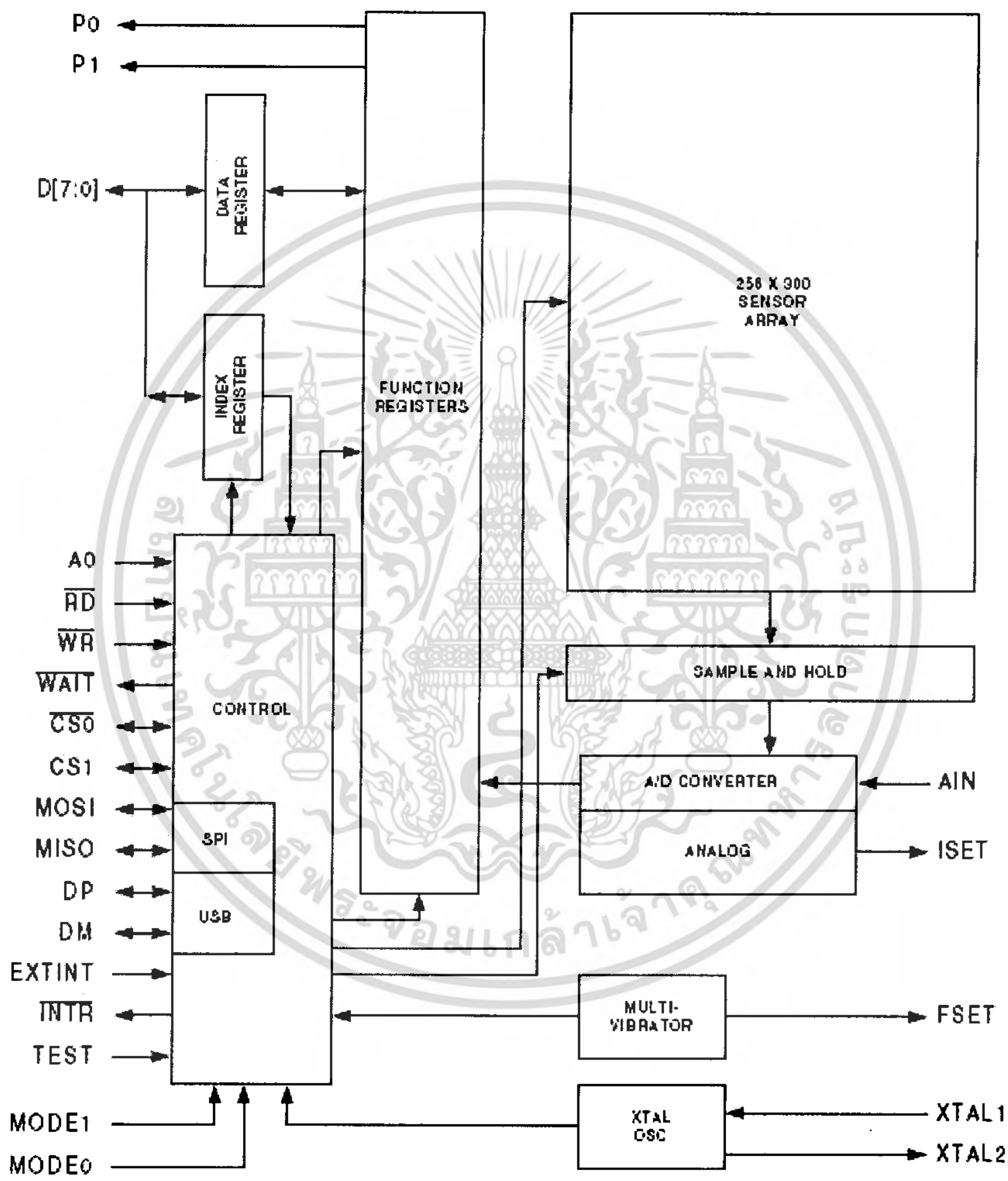
**ISR 0x0E** เป็นรีจิสเตอร์ที่ทำหน้าที่บอกสถานะของการอินเตอร์รัพท์

**THR 0x0F** เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมค่าขีดเริ่ม (Threshold) ในสถานะอินเตอร์รัพท์ในโหมด Auto Detection

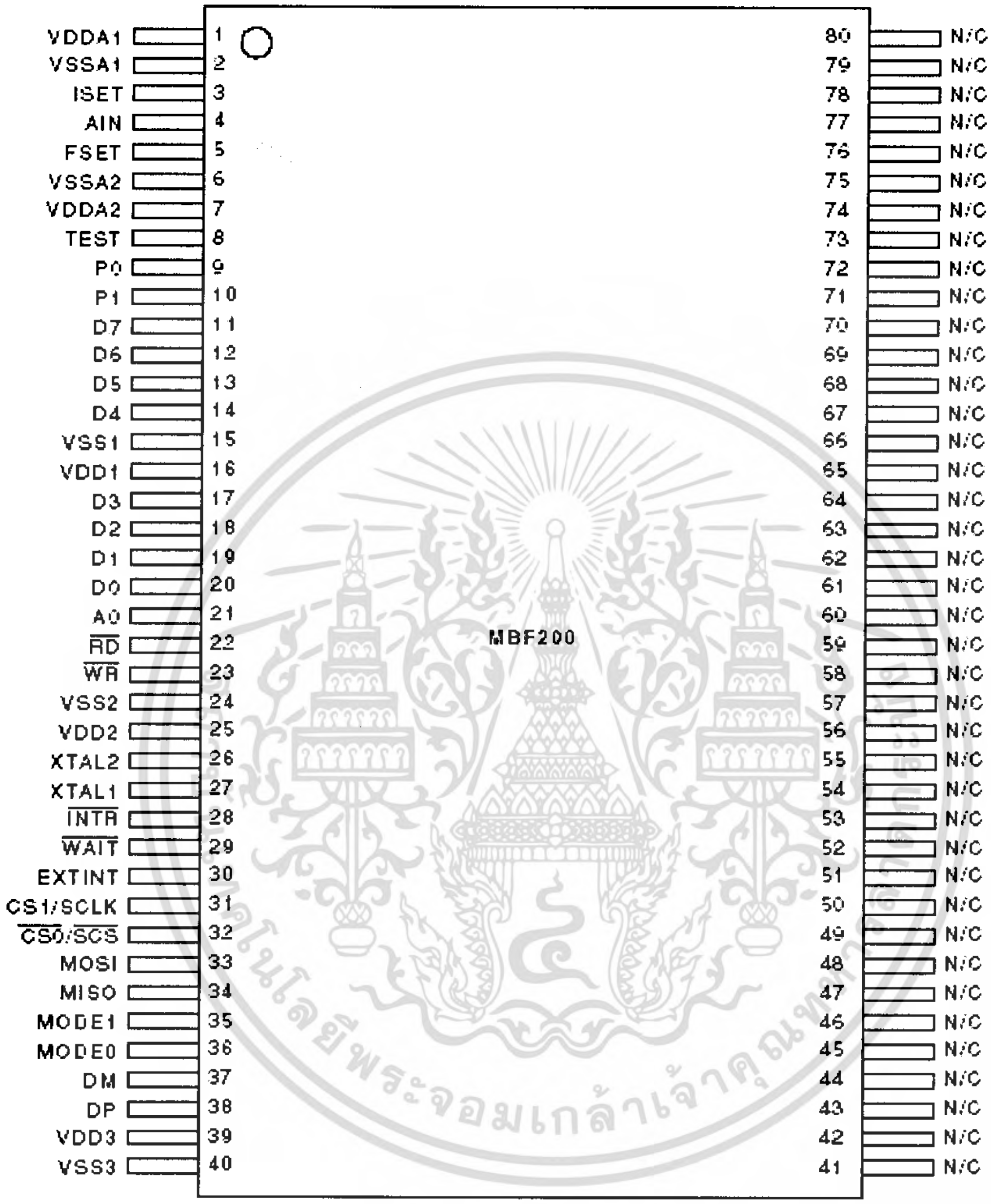
**CIDH 0x10** เป็นขา Read Only ทำหน้าที่เก็บไบต์สูงของค่ารุ่นของไอซี

**CIDL 0x11** เป็นขา Read Only ทำหน้าที่เก็บไบต์ต่ำของค่ารุ่นของไอซี

**TST 0x12** ใช้สำหรับทดสอบการทำงานของไอซี สงวนไว้ใช้สำหรับโรงงาน

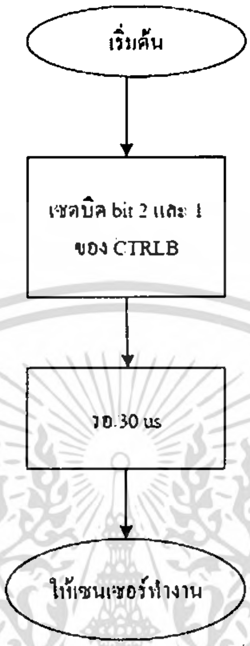



รูปที่ 3.2 แสดงแบบจำลองการทำงานของ MBF200

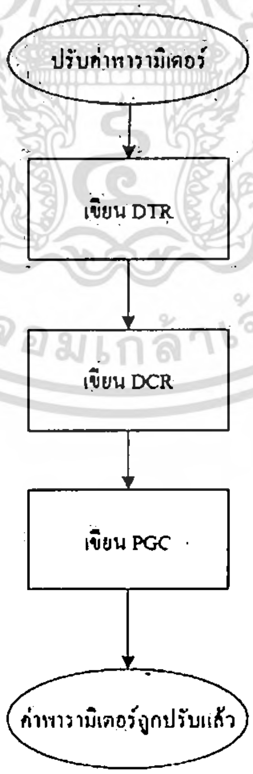



รูปที่ 3.3 แสดงขาสัญญาณของ MBF200

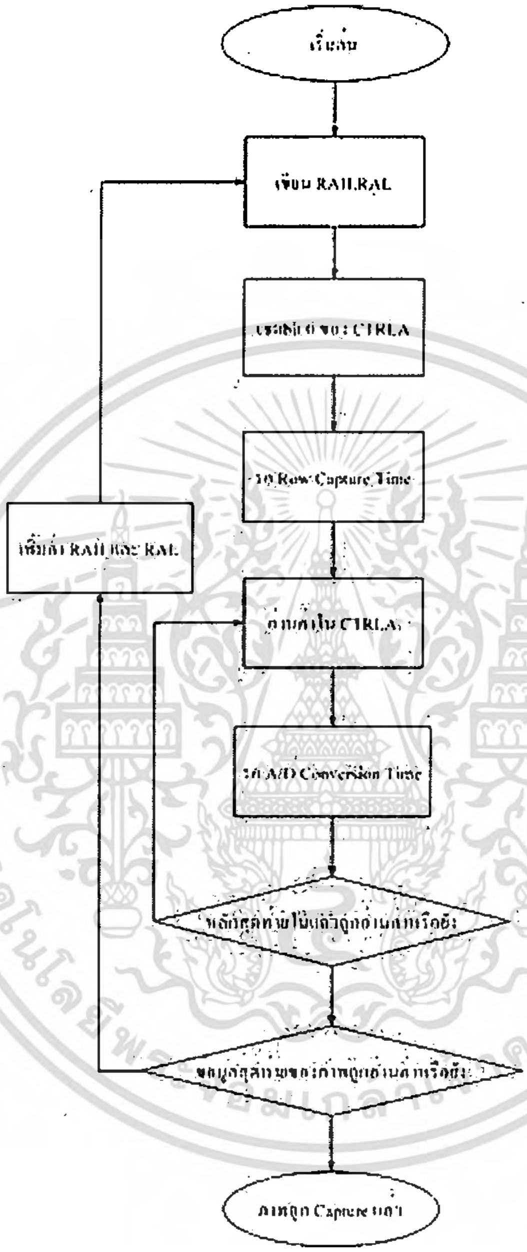
### 3.5. แผนภาพแสดงขั้นตอนการสั่งงานการทำงานของ MBF200




รูปที่ 3.4 แสดงการเลือกโหมดการทำงานของ MBF200




รูปที่ 3.5 แสดงการปรับค่าพารามิเตอร์ที่มีผลต่อความคมชัดของภาพ




รูปที่ 3.6 แสดงขั้นตอนการสแกนภาพ

# บทที่ 4
# ทฤษฎีและหลักการวิเคราะห์ลายนิ้วมือ

## 4.1. หลักการวิเคราะห์ลายนิ้วมือ

### 4.1.1. หลักการวิเคราะห์ลายนิ้วมือโดยทั่วไป

การวิเคราะห์ลายนิ้วมือของบุคคลโดยทั่วไปนั้น จะเริ่มด้วยการนำลายนิ้วมือของแต่ละบุคคลแต่ละนิ้วมาหาจุดลักษณะเฉพาะที่สำคัญ

กระบวนการแรกเริ่มของการตรวจพิสูจน์ลายนิ้วมือคือ การอ่านภาพลายนิ้วมือเข้ามาเก็บไว้ในฐานข้อมูล (Data base) โดยข้อมูลที่อ่านหรือสแกนเข้ามานั้นจะนำมาผ่านการประมวลผล (Processing) ก่อนแล้วจึงเก็บข้อมูลนั้นไว้เป็นต้นแบบหรือรหัสของผู้ใช้แต่ละคน

จากที่กล่าวมาแล้ว ก่อนที่จะนำลายนิ้วมือเข้าไปเก็บในฐานข้อมูลนั้น จะต้องผ่านขั้นตอนการประมวลผลเบื้องต้น (Preprocessing) และการประมวลผลภายหลัง (Postprocessing) ก่อน ในกระบวนการนี้จะทำให้ภาพที่ได้รับจากการสแกนเข้ามาเกิดความสมบูรณ์มากขึ้น เพราะเมื่อเครื่องสแกนได้รับการสแกนภาพเข้ามาแล้ว ภาพที่อ่านได้อาจไม่ชัดเจน พร่าเลือน ก็จะทำให้การประมวลผลในขั้นตอนถัดไปทำได้ด้วยความยากลำบากหรือทำไม่ได้ ซึ่งจะทำให้ผลที่ได้ก็อาจไม่ถูกต้องตามที่ควรจะเป็น เมื่อเกิดปัญหาเช่นนี้ในกระบวนการนี้จึงได้มีการกระทำหลายกระบวนการด้วยกันคือ การทำ Normalization, การหาทิศทางภาพ (Orientation), การกรองสัญญาณ (Gabor Filter), การทำลายเส้นให้บาง (Thinning), การหาจุดกึ่งกลางของลายนิ้วมือ (Core), การหาจุดสำคัญของลายนิ้วมือ(Minutiae) และอื่น ๆ อีกมากมาย ซึ่งกระบวนการจะมากหรือน้อยขึ้นอยู่กับว่าตัวอุปกรณ์นั้นมีการอ่านค่าลายนิ้วมือที่ได้ภาพออกมาละเอียดและสมบูรณ์แค่ไหน

เมื่อได้ลายนิ้วมือที่ผ่านการประมวลผลแล้ว ก็จะนำข้อมูลหรือภาพนี้ไปเก็บในฐานข้อมูล โดยภาพที่ถูกจัดเก็บไว้นี้จะถูกเก็บไว้เพื่อใช้ในการเปรียบเทียบกับลายนิ้วมือที่ได้รับการสแกนเข้ามาเมื่อนำตัวอุปกรณ์นี้มาใช้งาน

ขั้นตอนในการใช้งานก็จะคล้ายกับตอนอ่านลายนิ้วมือเข้ามาเก็บไว้ เพียงแต่การอ่านเข้ามาครั้งนี้ข้อมูลที่ได้จะถูกนำไปเก็บไว้ในหน่วยความจำชั่วคราว (RAM) ซึ่งในขั้นตอนต่อไปก็จะนำข้อมูลที่เก็บอยู่ในฐานข้อมูลกับส่วนที่เก็บในหน่วยความจำชั่วคราวนั้นมาเปรียบเทียบกัน (Matching) เมื่อได้ผลแล้วก็จะแสดงผลให้ทราบว่าเป็นลายนิ้วมือของบุคคลเดียวกันหรือไม่ ซึ่งสามารถนำไปใช้ประโยชน์ใน

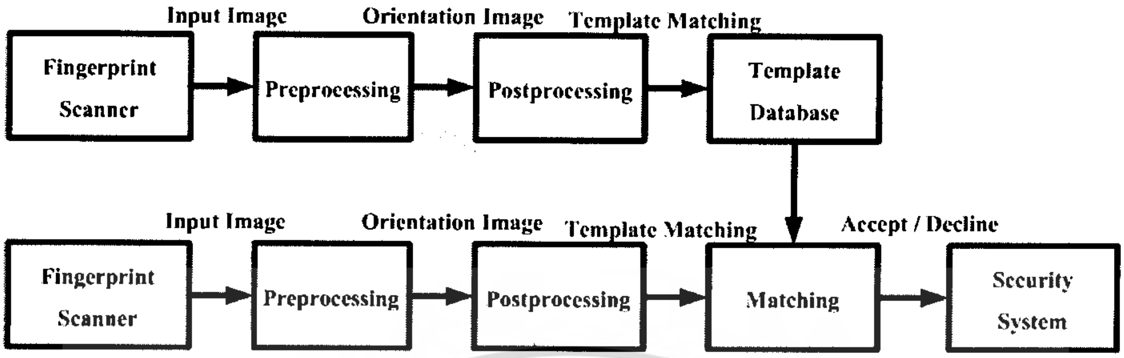
การพัฒนาระบบงานด้านต่างๆให้มีประสิทธิภาพขึ้นได้ เช่น งานด้านอาชญากรรมของกรมตำรวจ งานทะเบียนราษฎร์ของกรมการปกครอง หรืองานด้านความปลอดภัย


รูปที่ 4.1 แสดงขั้นตอนการเปรียบเทียบลายนิ้วมือโดยทั่วไป

4.1.2. หลักการวิเคราะห์ลายนิ้วมือสำหรับโครงงานนี้

สำหรับโครงงานนี้จะมีหลักการวิเคราะห์ลายนิ้วมือคล้ายคลึงกับหลักการ โดยทั่วไป เพียงแต่จะมีความแตกต่างกันในส่วนของการประมวลผลเบื้องต้น และการประมวลผลภายหลัง โดยที่ขั้นตอนหรืออัลกอริทึมของการประมวลผลเหล่านี้จำเป็นต้องพิจารณาให้เหมาะสมกับการนำไปสร้างเป็นเครื่องแสกนลายนิ้วมือที่สามารถประมวลผลบนตัวเครื่องได้เลย และการนำไปพัฒนาระบบรักษาความปลอดภัยต่อไป ดังแสดงในรูปที่ 5.2

ดังนั้นเพื่อเป็นไปตามวัตถุประสงค์ของโครงงาน ในภาคการศึกษานี้จะเป็นการทดสอบหาอัลกอริทึมที่ใช้ในการประมวลผลภาพที่เหมาะสมที่สุด ซึ่งอัลกอริทึมหรือขั้นตอนที่ได้นั้นจะเป็นไปตามหัวข้อที่ 4.2




รูปที่ 4.2 แสดงขั้นตอนการทำงานของการเปรียบเทียบลายนิ้วมือในโครงงานนี้

## 4.2. ขั้นตอนการเปรียบเทียบลายนิ้วมือ

ขั้นตอนของการเปรียบเทียบลายนิ้วมือที่ใช้ในโครงงานนี้จะประกอบด้วยส่วนของการประมวลผลเบื้องต้น(Preprocessing) การประมวลผลภายหลัง (Postprocessing) และการเปรียบเทียบลายนิ้วมือ (Matching)

### 4.2.1. การประมวลผลเบื้องต้น (Preprocessing)

#### 4.2.1.1.นอร์มัลไลเซชั่น Normalization

เนื่องจากภาพลายนิ้วมือที่ได้รับมาแต่ละภาพนั้น มีค่าเฉลี่ยของความสว่างของภาพไม่เท่ากัน เพื่อให้ทุกภาพมีค่าเฉลี่ยของระดับความสว่างของภาพที่เท่ากัน จึงต้องมีการนอร์มัลไลเซชั่นภาพเสียก่อนซึ่งภาพที่ได้จะง่ายต่อการฟิลเตอร์ภาพต่อไป

ให้ I(i,j) แสดงถึงค่าระดับสีเทาที่กำหนดตำแหน่งพิกเซล(i,j) , M และ VAR แสดงถึงค่าเฉลี่ยโดยประมาณและการเปลี่ยนแปลงของ I และ G(i,j) แสดงถึงค่าระดับสีเทาที่ตำแหน่งพิกเซล (i,j) ซึ่งผ่านการนอร์มัลไลเซชั่นแล้วได้ดังต่อไปนี้

$$G(i,j)=\begin{cases} M_0+\sqrt{\dfrac{VAR_0(I(i,j)-M)^2}{VAR}} & ,if \quad I(i,j)>M \\ M_0-\sqrt{\dfrac{VAR_0(I(i,j)-M)^2}{VAR}} & ,otherwise \end{cases}$$

โดย $I(i,j)$ เป็นค่าระดับเทา (gray-level value)ที่พิกเซล $(i,j)$

M เป็นค่า mean ของภาพ

VAR เป็นค่า variance ของภาพ

$G(i, j)$ เป็นค่า normalized ที่พิกเซล $(i, j)$

$M_0$ และ $VAR_0$ เป็นค่า mean และค่า variance ที่ต้องการตามลำดับ

การนอร์มัลไลเซชั่นเป็นการกระทำในระดับพิกเซลซึ่งจะไม่เปลี่ยนแปลงความชัดเจนของโครงสร้างเส้นนูนและร่องของภาพ

a) b)

รูปที่ 4.3 a) รูปต้นแบบ b) รูปที่ผ่านการทำnormalized

**4.2.1.2. Orientation Image (ภาพทิศทาง)**

การกำหนดทิศทางของภาพจะแสดงถึงลักษณะที่สำคัญของภาพลายนิ้วมือและการกำหนดค่าตำแหน่งที่คงที่สำหรับสันและร่องในบริเวณที่ใกล้เคียง โดยการพิจารณารายละเอียดของทิศทางของภาพลายนิ้วมือโดยมีขั้นตอนการทำดังนี้

4.2.1.2.1 ทำการคำนวณค่าเกรเดียนท์ $\partial_x(i,j)$ และ $\partial_y(i,j)$ ของแต่ละพิกเซล $(i,j)$ ซึ่งในรายงานนี้ได้ใช้วิธีการหาขอบภาพด้วยวิธีของSobel (Sobel Edge Detection) โดยการใช้หน้ากากขนาด 3x3 ไปทำการ convolution กับภาพของเรา ดังนี้

| | | |
|---|---|---|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

Gx

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

Gy

รูปที่4.4 หน้ากาก Sobel Convolution

4.2.1.2.2 ทำการประมาณทิศทางการกำหนดทิศทางบริเวณใกล้เคียงของแต่ละศูนย์กลางของแต่ละพิกเซล (i,j) โดยใช้สมการดังต่อไปนี้

$$V_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v)$$

$$V_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \left(\partial_x^2(u,v) - \partial_y^2(u,v)\right)$$

$$\theta(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{V_y(i,j)}{V_x(i,j)}\right)$$

โดยที่ $\theta(i,j)$ เป็นค่าประมาณของทิศทางที่ใกล้เคียง ณ ตำแหน่ง (i.j)

4.2.1.2.3 เนื่องจากสัญญาณรบกวนที่เกิดขึ้นกับลายนิ้วมือได้แก่ โครงสร้างของเส้นนูนและร่องมีส่วนที่ไม่ต้องการเกิดขึ้น ลักษณะลายอื่น ๆ ทำให้การประมาณทิศทางของสันบริเวณใกล้เคียง $\theta(i,j)$ ไม่ถูกต้องเสมอไป เนื่องจากการกำหนดทิศทางของสันบริเวณใกล้เคียงมีการเปลี่ยนแปลงอย่างช้า ๆ ในบริเวณที่ใกล้เคียงโดยไม่มีจุดเด่นปรากฏขึ้น ฟิลเตอร์กรองความถี่ต่ำผ่านสามารถที่จะใช้ในการปรับปรุงทิศทางที่ไม่ถูกต้องของสันบริเวณใกล้เคียงได้ ในการใช้ฟิลเตอร์กรองความถี่ต่ำผ่านนั้น ภาพที่กำหนดทิศทางแล้วจำเป็นที่จะต้องเปลี่ยนกลับไปในรูปแบบของเวกเตอร์ที่ต่อเนื่อง ซึ่งกำหนดไว้ดังนี้

$$\phi_x(i,j) = \cos(2\theta(i,j))$$
$$\phi_y(i,j) = \sin(2\theta(i,j))$$

โดย $\phi_x$ เป็น vector field ในแกน x

$\phi_y$ เป็น vector field ในแกน y

ฟิลเตอร์กรองความถี่ต่ำผ่านสามารถเขียนได้ดังนี้

$$\Phi_z^{'}(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_x(i-uw, j-vw)$$
$$\Phi_y^{'}(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_y(i-uw, j-vw)$$

โดย W คือ ฟิลเตอร์กรองความถี่ต่ำผ่านแบบ 2 มิติโดนจำนวนเต็มและ $\omega_\Phi \times \omega_\Phi$ คือขนาดของฟิลเตอร์เพื่อให้การใช้งานทำให้ภาพออกมาเรียบที่สุด ขนาดของฟิลเตอร์ที่ใช้ควรจะเป็น5x5

4.2.1.2.4 คำนวณทิศทางของสันที่อยู่ใกล้กัน ณ ตำแหน่ง(i,j) โดยใช้

$$O(i,j) = \frac{1}{2}\tan\left(\frac{\Phi'_y(i,j)}{\Phi'_x(i,j)}\right)$$


รูปที่ 4.5 แสดงภาพทิศทางของเส้นลายนิ้วมือ

**4.2.1.3. การกรองลดสัญญาณรบกวน (Filtering)**

เพื่อที่จะเอาสัญญาณรบกวนที่ไม่ต้องการออกไป และแสดงส่วนเส้นนูนและร่องของเส้นลายนิ้วมือให้เด่นชัดขึ้น จึงทำการกรองภาพลายนิ้วมือให้ทิศทางที่แตกต่างกัน โดยการใช้ Gabor filter ซึ่งเป็นแบนด์พาสฟิลเตอร์ ซึ่งสามารถที่จะทำการเลือกหรือกำหนดคุณสมบัติเกี่ยวกับทิศทางและความถี่ได้และมีความแน่นอนของการเชื่อมต่อที่ดีที่สุดทั้งในโดเมนของที่ว่างและความถี่สำหรับภาพลายนิ้วมือที่มีร่องและสันที่ขนานกัน คุณสมบัติการปรับค่าของ Gabor filter สามารถที่จะเอาสัญญาณรบกวนออกได้ และรักษาโครงร่างของร่องและสันที่แท้จริงเอาไว้ และการแบ่งข้อมูลจะจำกัดทิศทางที่จำเพาะของภาพ โดย Gabor filter มีสมการดังนี้

$$h(x, y : \phi, f) = \exp\left\{\frac{-1}{2}\left[\frac{x_\phi^2}{\delta_x^2} + \frac{y_\phi^2}{\delta_y^2}\right]\right\}\cos(2\pi f x_\phi)$$

$$x_\phi = x\cos\phi + y\sin\phi$$

$$y_\phi = -x\sin\phi + y\cos\phi$$

โดยที่ $\phi$ เป็นทิศทางของ Gabor filter

$f$ เป็นความถี่ของระนาบคลื่น sine

$\delta_x$ เป็นค่าแวร์เรี่ยน(ค่าเบี่ยงเบนมาตรฐาน) ตามแนวแกน x

$\delta_y$ เป็นค่าแวร์เรี่ยน(ค่าเบี่ยงเบนมาตรฐาน) ตามแนวแกน y

รูปที่ 4.6 แสดงเส้นลายนิ้วมือที่ผ่านกระบวนการกรองแล้ว

#### 4.2.1.4. การทำเส้นลายนิ้วมือให้บาง (Thinning)

นำภาพที่ผ่านกระบวนการกรองแล้ว มาทำให้เส้นบาง จนมีขนาด 1 พิกเซล ซึ่งจะนำภาพที่ได้จากการผ่านกระบวนการนี้ไปหาจุดสำคัญ (minutiae) ต่อไป

รูปที่ 4.7 แสดงภาพเส้นลายนิ้วมือที่ผ่านกระบวนการทำเส้นลายนิ้วมือให้บางแล้ว

### 4.2.2. การประมวลผลหลัง (Post processing)

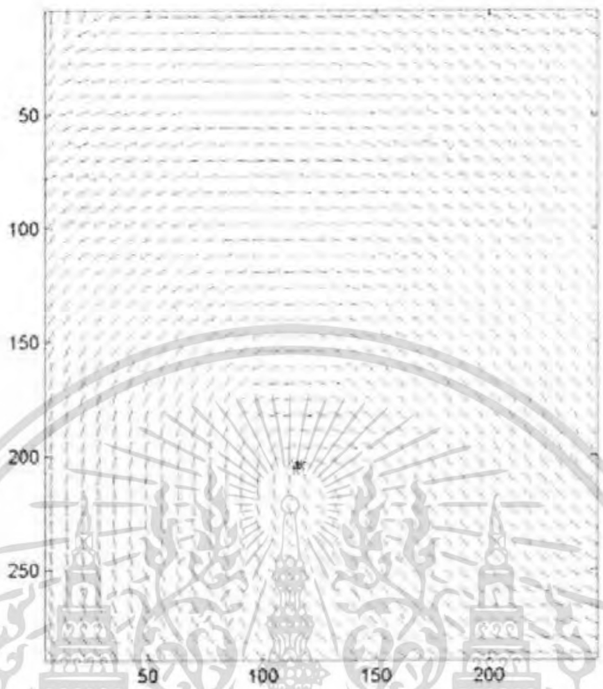
#### 4.2.2.1. การหาจุดกึ่งกลางของภาพทิศทาง

หาได้โดยนำภาพทิศทาง มาหาจุดที่คาดว่าจะเป็นจุดกึ่งกลางของภาพ (Core) ซึ่งจุดกึ่งกลางของภาพ จะเป็นจุดที่มีการเปลี่ยนแปลงของทิศทางมากที่สุด โดยสามารถพิจารณาได้จากสมการดังนี้

$$d_x = \sum_{j=-1}^{1} \sin(2\theta(i-1,j)) - \sum_{j=-1}^{1} \sin(2\theta(i+1,j))$$

$$d_y = \sum_{i=-1}^{1} \cos(2\theta(i,j-1)) - \sum_{i=-1}^{1} \cos(2\theta(i,j+1))$$

โดยที่ $\theta(i,j)$ เป็นภาพทิศทางของ Block $(i,j)$




รูปที่ 4.8 แสดงจุดกึ่งกลางของภาพลายนิ้วมือ

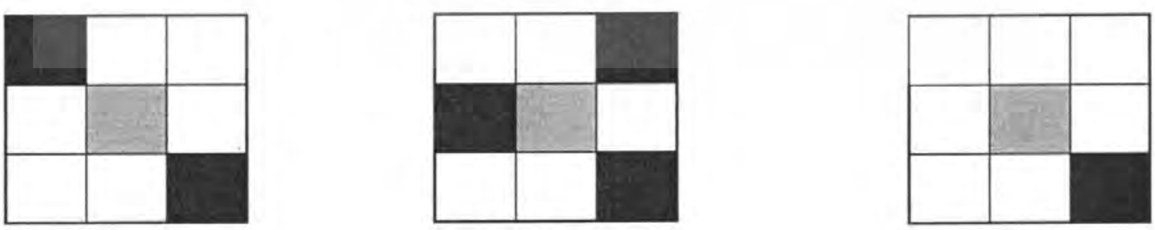
**4.2.2.2 การหาลักษณะสำคัญของลายนิ้วมือ (Minutiae Extraction)**

นำภาพที่ได้จากการ Thinning มาหาลักษณะสำคัญ (Minutiae) ตามสมการ

$$cn(p) = \frac{1}{2}\sum_{i=1\ldots8}\left|val(p_i) - val(p_{i+1})\right|$$

ซึ่ง $p_1, p_2, \ldots, p_8$ เป็นพิกเซลข้างเคียง 8 พิกเซล ของ p

Val(p) เป็นค่าของพิกเซล



cn(p)=2 cn(p)=3 cn(p)=1

ถ้า cn(p) = 1 จะถือว่าเป็นจุดปลายของลายเส้นนูน (Ridge endings)

cn(p) = 3 จะถือว่าเป็นจุดแยกของลายเส้นนูน (Ridge bifurcations)

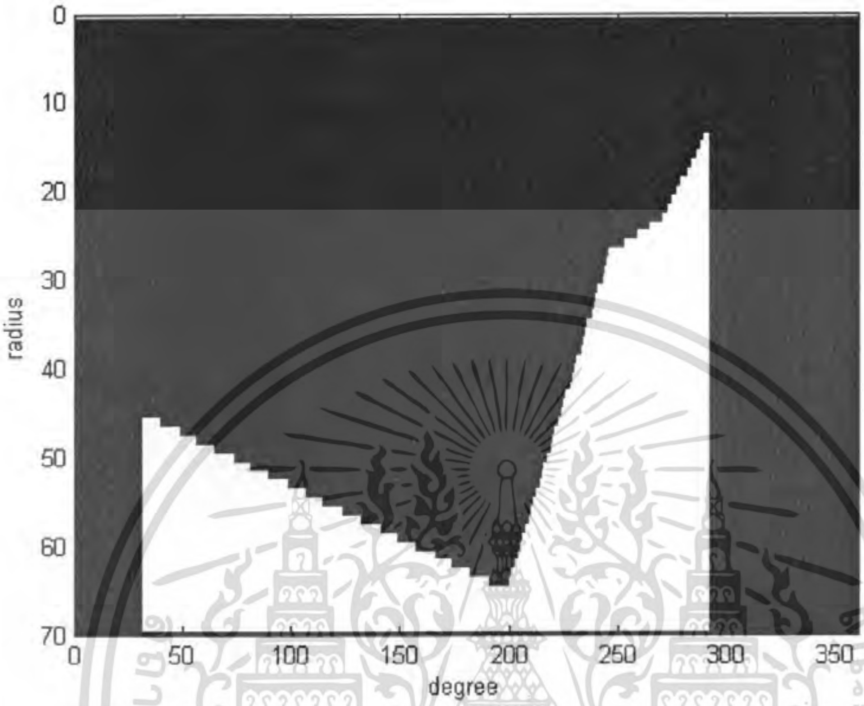
รูปที่ 4.9 แสดงจุดสำคัญที่ได้จากภาพลายนิ้วมือ

**4.2.2.3. วิธีสร้างข้อมูลจากจุดลักษณะเฉพาะ *****

**1) การเขียนกราฟความสัมพันธ์ระหว่างรัศมีและมุมของจุดกึ่งกลางจุดสำคัญ**

เขียนกราฟความสัมพันธ์ระหว่างรัศมีที่ลากจุดกึ่งกลางถึงจุดสำคัญและมุมที่จุดสำคัญทำกับแกน X เพื่อที่จะนำไปใช้ในกระบวนการ Matching ต่อไป


*** หัวข้อที่4.2.2.3.1 อ้างอิงจากปริญญานิพนธ์ เรื่อง เครื่องสแกนลายนิ้วด้วยวิธีมุมและรัศมีที่จุดสำคัญ ปีการศึกษา 2549




รูปที่ 4.10 แสดงรูปกราฟความสัมพันธ์ระหว่างรัศมีและมุมที่จุดสำคัญทำกับแกน X

**1.2) การทำฐานข้อมูล(Template Database)**

เมื่อเราได้ข้อมูลของรัศมีที่ลากระหว่างจุดกึ่งกลางถึงจุดสำคัญและมุมที่จุดสำคัญทำกับแกน X มาแล้วเราจะนำข้อมูลนี้มาผ่านกระบวนการหาค่าเฉลี่ยกลุ่มเพื่อเก็บเป็นฐานข้อมูลด้วยวิธี Hard C-Means โดยใช้สมการต่าง ๆ ดังนี้

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\left(\sum_{k,x_k \in G_i} \|x_k - C_i\|^2\right)$$

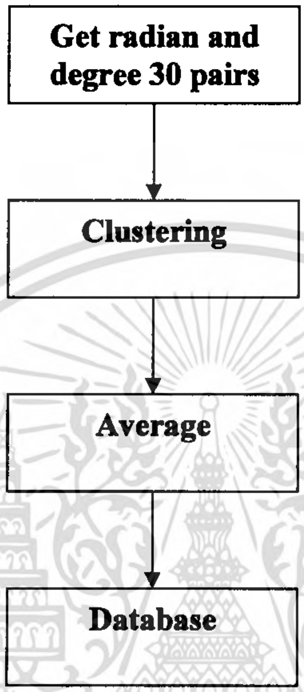
โดย J เป็นการหาค่าผลรวมของระยะทางระหว่างข้อมูลแต่ละจุดกับค่าเฉลี่ยของกลุ่มของทุกกลุ่มและ $C_i$ เป็นค่าเฉลี่ยของกลุ่มได้มาโดยการสุ่มเลือกจากข้อมูลแรกที่รับเข้ามา

$$u_{i,j} = \begin{cases} 1 & if \|X_j - C_i\|^2 \le \|X_j - C_k\|^2, for\ each\ k \ne i \\ 0 & otherwise \end{cases}$$

โดย U เป็นการคัดแยกว่าข้อมูลแต่ละจุดเป็นสมาชิกของกลุ่มนั้นหรือไม่ โดยถ้าเป็น 1 คือเป็นสมาชิกแต่ถ้าเป็น 0 คือไม่ใช่สมาชิก

$$C_i = \frac{1}{|G_i|}\sum_{k,x_k \in G_i} X_k$$

เป็นการหาจุดศูนย์กลางเฉลี่ยของข้อมูลแต่ละกลุ่ม เราจะวนทำกระบวนการนี้จนกว่าจะได้จุดศูนย์กลางเฉลี่ยที่คงที่และเก็บค่านั้นเป็นฐานข้อมูลของแต่ละบุคคล




รูปที่4.11 แสดงขั้นตอนการทำฐานข้อมูล

**1.3) การเปรียบเทียบลายนิ้วมือ (Matching)**

เมื่อเราเขียนกราฟความสัมพันธ์ระหว่างรัศมีที่ลากจุดกึ่งกลางถึงจุดสำคัญและมุมที่จุดสำคัญทำกับแกน X จะได้กราฟลักษณะต่าง ๆ ซึ่งรูปกราฟของบุคคลเดียวกันคาดว่าจะมีลักษณะกราฟใกล้เคียงกัน แต่รูปกราฟของบุคคลต่างกัน คาดว่าจะได้กราฟที่มีลักษณะต่างกัน โดยกระบวนการในการหาความสัมพันธ์นั้น เราจะใช้วิธี Template Matching

**1.4) วิธีการวัดความแม่นยำของระบบ**

ความแม่นยำของระบบไบโอเมตริกสามารถจะถูกเทียบวัดได้ค่าต่อไปนี้

1) False Rejection Rate : FRR คือ อัตราการปฏิเสธการผ่านกับผู้ใช้ที่ถูกต้อง
2) False Acceptance Rate : FAR คือ อัตราการหลุดรอดของผู้แปลกปลอมจากการตรวจจับ

โดยทั่วไปค่า FRR และ FAR เป็นค่าที่ค้านซึ่งกันและกันอยู่ เพราะเมื่อ FAR มีค่าสูง FRR ก็จะมีค่าต่ำไปโดยอัตโนมัติ

ในระบบรักษาความปลอดภัยด้วยลายนิ้วมือนี้ ค่า FRR และ FAR จะเป็นค่าที่สามารถถูกปรับได้ตามความต้องการของผู้ติดตั้งระบบว่าต้องการให้มีความปลอดภัยอยู่มากน้อยเพียงใด

**2) การหารูปทรงสามเหลี่ยมจากกลุ่มของจุดลักษณะเฉพาะ**

กลุ่มของจุดลักษณะเฉพาะของลายนิ้วมือที่ได้ทำการหาโดยผ่านกระบวนการต่างๆดังที่กล่าวมาแล้วนั้นสามารถที่จะนำมาพิจารณาในการนำมาทำการตรวจสอบภาพลายนิ้วมือได้ โดนจะเห็นได้ว่าจุดเหล่านี้จะกระจายกันอยู่บนระนาบและจะใช้จุดเหล่านั้นทำการหารูปทรงสามเหลี่ยม เพื่อที่จะนำมาเป็นเครื่องมือและจะมีประสิทธิภาพมากในการจัดการกับกลุ่มของจุดลักษณะเฉพาะที่กระจัดกระจายกันคือเราจะทำการประยุกต์มาใช้งานการตรวจสอบลายนิ้วมือได้โดยวิธีของ DT (Delaunay Triangulation)

**2.1) ขั้นตอนการสร้างรูปสามเหลี่ยมจากกลุ่มข้อมูลของจุดลักษณะเฉพาะโดยใช้ DT (Delaunay Triangulation)**

กระบวนการของ DT ที่สร้างรูปทรงสามเหลี่ยมจากกลุ่มของลักษณะเฉพาะในรูปที่4.16 นั้นแสดงขั้นตอนการสร้างสามเกลี่ยนมจากลักษณะเฉพาะ และในรูปที่4.17 แสดงผลของการหาจากจุดลักษณะเฉพาะบนลายนิ้วมือ โดยที่กระบวนการที่ทำนั้นถูกแบ่งเป็นสี่ขั้นตอนคือ

1.ขั้นตอนการเตรียมข้อมูล

ทำการคำนวณหารูปแบบของกริดจะเริ่มจากหาค่าต่ำสุด-สูงสุดของกรอบสี่เหลี่ยมจากกลุ่มข้อมูลที่มีอยู่จากนั้นทำการขยายกรอบของข้อมูลโดยการทำการเพิ่มออฟเซท คือค่า TOL

$$X_{min} = x_{min} - TOL$$

$$X_{max} = x_{max} + TOL$$

$$Y_{min} = y_{min} - TOL$$

$$Y_{max} = y_{max} + TOL$$

โดยที่

$x_{min}$ คือค่าของข้อมูลที่มีค่าต่ำสุดในทางแกนx

$x_{max}$ คือค่าของข้อมูลที่มีค่าสูงสุดในทางแกนx

$y_{min}$ คือค่าของข้อมูลที่มีค่าต่ำสุดในทางแกนy

$y_{max}$ คือค่าของข้อมูลที่มีค่ามากสุดในทางแกนy

ทำการคำนวณขนาดของกริดโดย

$$size = \sqrt{\frac{(xmax-xmin)(ymax-ymin)}{n}} \tag{4.1}$$

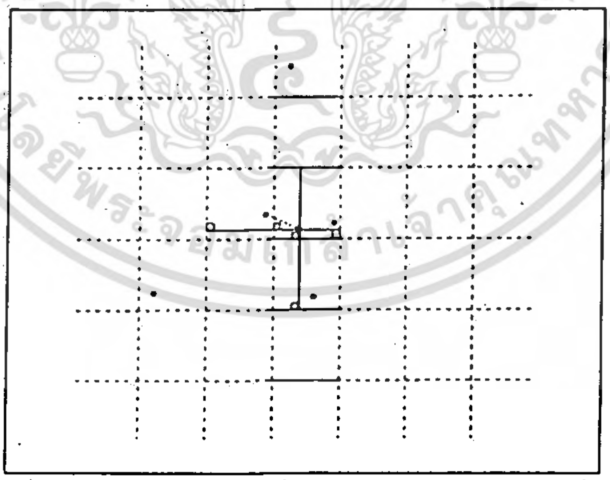
$$x\_res = \left|\frac{xmax-xmin}{size}\right| - 1 \tag{4.2}$$

$$y\_res = \left|\frac{ymax-ymin}{size}\right| - 1 \tag{4.3}$$

โดยที่ n คือจำนวนของจุดข้อมูลที่มี

2. ทำการหาจุดเริ่มต้นและเส้นเริ่มต้น

ทำการหาจุดลักษณะเฉพาะที่จะนำมาทำเป็นจุดเริ่มต้นโดยจะต้องทำการหาจากส่วนที่อยู่ตรงกลางของ กริดหรือทำการหาจุดจากบริเวณนั้นต่อจากนั้นทำการหาจุดลักษณะเฉพาะจุดที่สองซึ่งจุดๆนี้จะต้องมีระยะใกล้กับจุกแรกมากที่สุดของจุดข้างเคียงที่อยู่ใกล้กับจุดเริ่มต้น ต่อจากนั้นทำการเชื่อมจุดลักษณะเฉพาะจุดนี้เข้าเข้ากับจุดแรกดังรูปที่5.12 หลังจากที่ได้เลือกจุดเริ่มต้น P1 แล้วทำการหาเส้นเริ่มแรกโดนทำการหาจุดที่มีตำแหน่งใกล้ที่สุด P2 ต่อมาทำการเชื่อมเส้น P1 และ P2 เพื่อใช้สำหรับสร้างสามเหลี่ยมเริ่มต้น

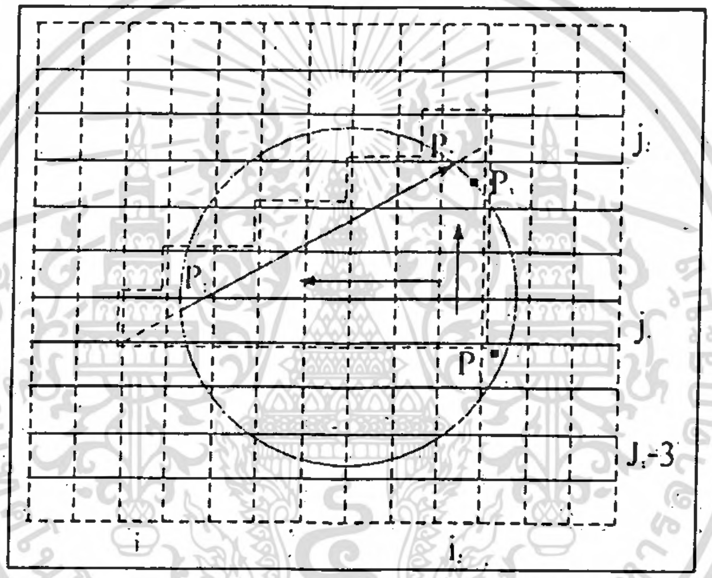


รูปที่ 4.12แสดงการหาจุดเริ่มต้นของการสร้างสามเหลี่ยม

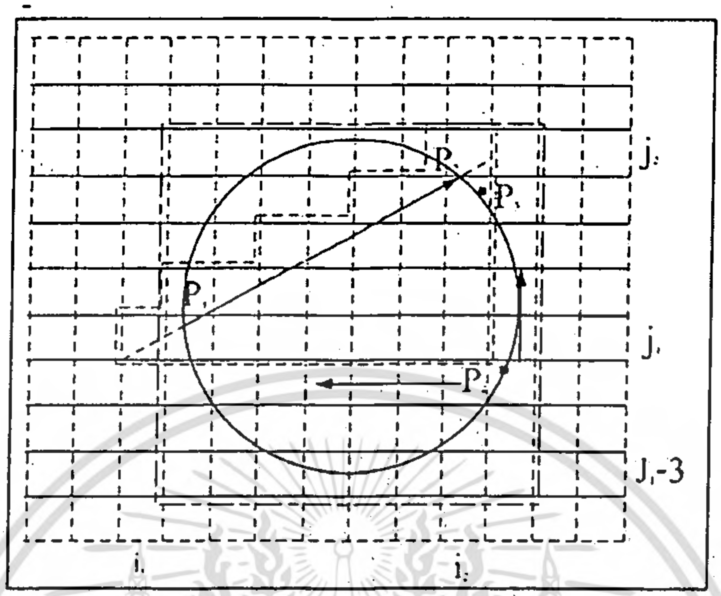
3. ทำการสร้างสามเหลี่ยม

ต่อมาทำการค้นหาจุดทางด้านขวามือของ P1,P2 หา เซลล์จากตำแหน่ง $(i_1,j_1)$ และ $(i_2,j_2)$ ทั้งสองในเซลล์เหล่านี้สามารถมีจุดในด้านขวามือ P1,P2 หาตำแหน่งเซลล์ที่ตัดผ่านเส้น $i_1$ ในทำนองเดียวกัน

หาตำแหน่งที่ตัดผ่าน $j_2$ ที่อยู่ทางขวามือของ P2 ต่อมาทำการสร้างสามเหลี่ยมโดยมีพื้นที่ปกคลุมแต่ละเซลล์ของจุดจาก $(_{i1}j_1)$ $(i_2j_2)$ และ $(i_3j_3)$ ดังแสดงในรูปที่ 5.13 ทำการเลือกเซลล์ทุกเซลล์ที่ P1,P2 ตัดผ่านหลังจากนั้นทำการตรวจดูข้างในเซลล์แต่ละเซลล์หาจุดที่มีค่ามุมมากที่สุดที่สามารถเป็นไปได้ เมื่อทำการหาจุดที่มีค่ามุมมากที่สุดแล้วทำการสร้างวงกลมที่กำหนดโดย P1,P2 แล้วทำการกำหนดค่าต่ำสุด-สูงสุดของกรอบสี่เหลี่ยมของวงกลมดังแสดงรูปที่3.14 หาข้อมูลที่อยู่ข้างในแถวและหลักเมื่อเสร็จสิ้นให้ทำเหมือนกับกระบวนการที่ผ่านมาและจะหยุดทำเมื่อไม่มีการพบข้อมูลข้างในแถวและหลักในกรอบ



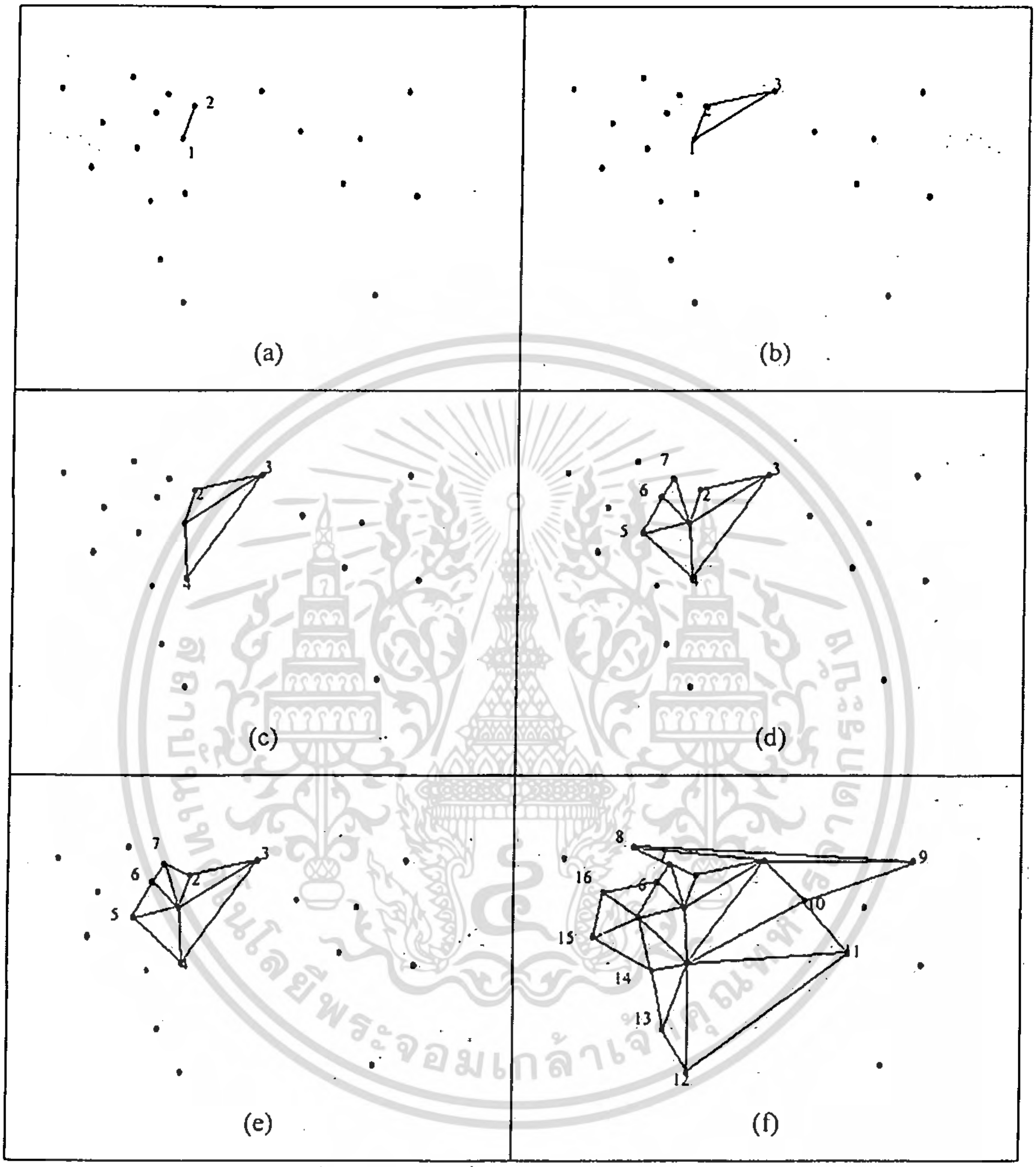
รูปที่4.13 แสดงขั้นตอนการสร้างเส้นเริ่มต้นของสามเหลี่ยม



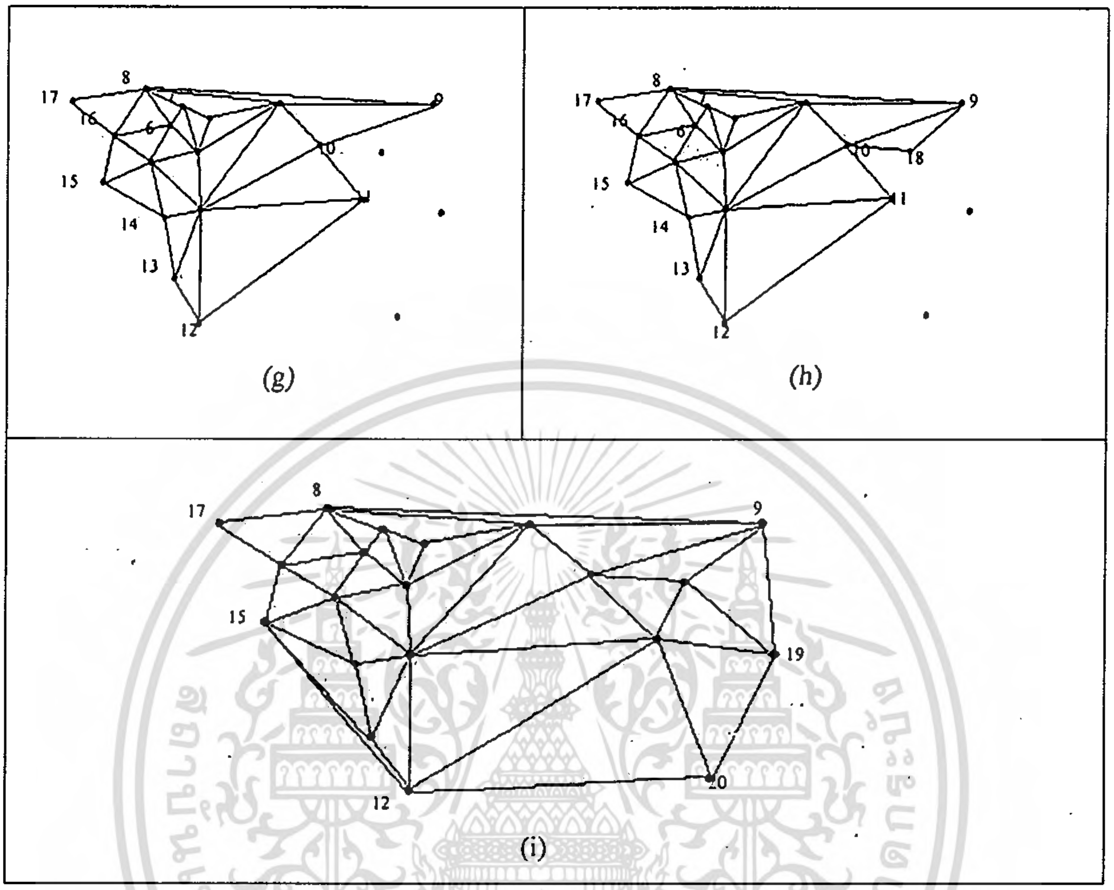

รูปที่4.14 แสดงการหาข้อมูลในเซลล์ที่อยู่ในกรอบสี่เหลี่ยม

4. ทำการวางสามเหลี่ยมเข้าด้วยกัน

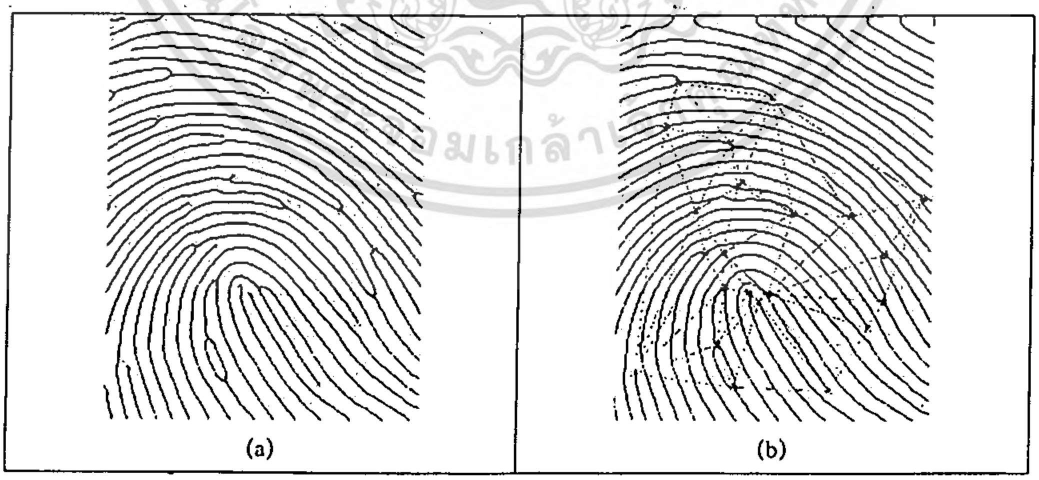
เมื่อทำตามกระบวนการต่างๆจนสร้างสามเหลี่ยมเรียบร้อยแล้วดังนั้นในการเชื่อมต่อสามเหลี่ยมจากกระบวนการทั้งหลายนั้นจะง่ายไม่วับซ้อนจะไม่เกิดช่องว่างและการข้ามของจุดเกิดขึ้น ทำการตรวจสอบกระบวนการทุกกระบวนการที่ได้ดำเนินการ ไปว่าสามเหลี่ยมเหล่านั้นเรียบร้อยและถูกต้องดังแสดงในรูปที่4.15




รูปที่4.15 แสดงขั้นตอนการสร้างสามเหลี่ยมจากกลุ่มของจุดลักษณะเฉพาะ




รูปที่4.16 แสดงขั้นตอนการสร้างสามเหลี่ยมจากกลุ่มของจุดลักษณะเฉพาะ(ต่อ)



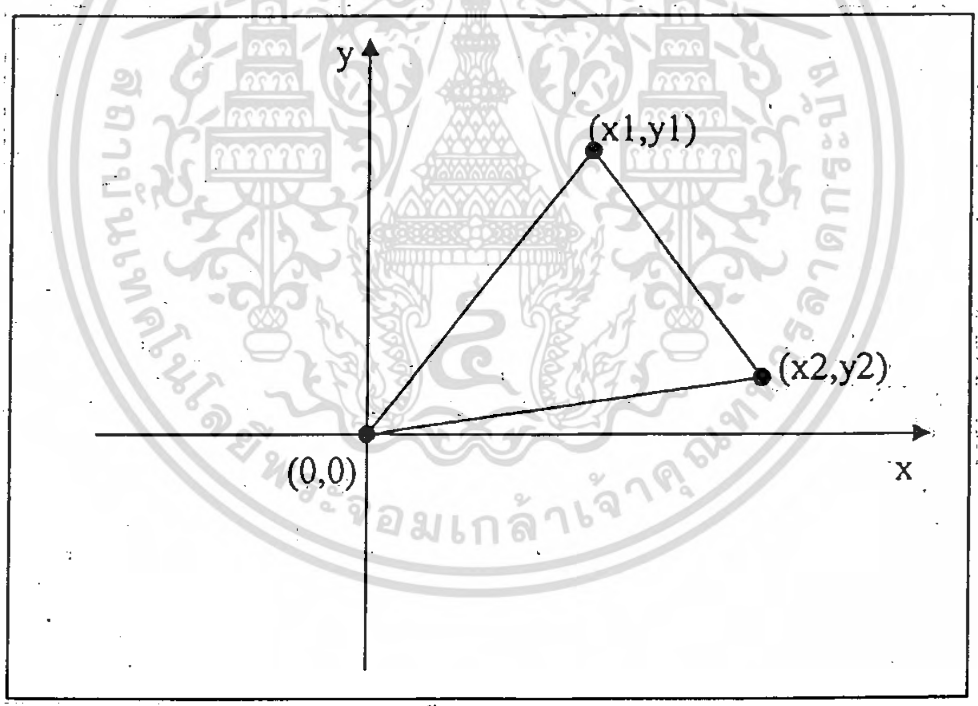

รูปที่4.17 แสดงผลของการหาจากจุดลักษณะเฉพาะบนลายนิ้วมือ a)แสดงจุดลักษณะเฉพาะบนลายนิ้วมือ b)แสดงสามเหลี่ยมที่ถูกสร้างจากจุดลักษณะเฉพาะ

**2.2) การเตรียมข้อมูลเพื่อใช้ในการเปรียบเทียบ**

หลังจากทำการนำภาพฉายลายนิ้วมือเพื่อทำการปรับปรุงเพื่อใช้ในภาพหลายนิ้วนั้นมีความสมบูรณ์ขึ้น เพื่อง่ายต่อกราฟจุดลักษณะเฉพาะของลายนิ้วมือ ต่อจากนั้นทำการหาสามเหลี่ยมโดยใช้วิธีการของ DT ซึ่งในการวิจัยนี้จะใช้รูปทรงสามเหลี่ยมที่ทำการหามาได้นำมาเพื่อใช้เพื่อทำการเปรียบเทียบลายนิ้วมือต่อไป

**2.2.1) การหาพื้นที่สามเหลี่ยม**

รูปทรงสามเหลี่ยมนั้นจะประกอบด้วยด้านทั้งสามด้านและจะมีมุมสามมุมซึ่งในการสร้างรูปทรงสามเหลี่ยมที่ผ่านมานั้นทุกๆสามเหลี่ยมจะถูกจุดจุดสามจุดประกอบกันขึ้นมาต่อหนึ่งสามเหลี่ยม และในการหาพื้นที่นี้ จะนำจุดมาทั้งสามจุดในแต่ละสามเหลี่ยมมาหาพื้นที่สามเหลี่ยมแต่ละอัน โดยในการหานี้ต้องทำการเลื่อนจุดให้จุดใดจุดหนึ่งให้มาอยู่ที่จุดกำเนิด (origin) ดังแสดงในรูปที่4.18




รูปที่ 4.18 การหาพื้นที่สามเหลี่ยมจากจุดสามจุด

โดยที่

$$area = \frac{|\Delta|}{2} \tag{4.4}$$

จาก

$$\Delta = \begin{vmatrix} x1 & y1 \\ x2 & y2 \end{vmatrix} \tag{4.5}$$

$$\Delta = |(x1 * y1) - (x2 * y2)| \tag{4.6}$$

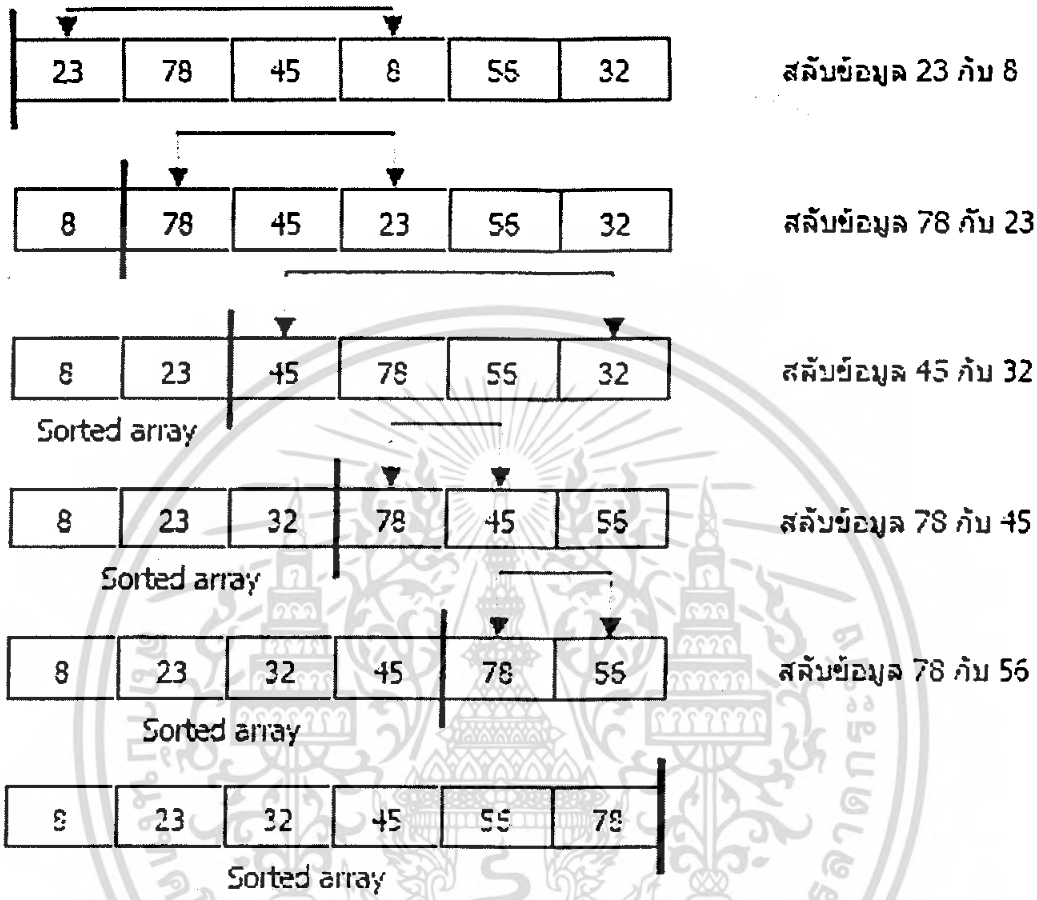
ในสมการที่ (4.4) เป็นสมการการหาพื้นที่สามเหลี่ยมโดยมีจุดใดจุดหนึ่งอยู่บนจุดกำเนิด

**2.2.2)การเรียงลำดับของของค่าตัวเลขแบบ (Selection Sort)**

การทำงานของ Selection sort เป็นการเปรียบเทียบข้อมูลที่อยู่ซ้ายสุดในแถว กับข้อมูลทางด้านขวาที่เล็กที่สุดในแถว ซึ่งถ้ามีข้อมูลนี้อยู่ ก็สลับข้อมูลนี้กับข้อมูลตัวแรกสุด จะเห็นได้ว่าข้อมูลที่เล็กอยู่ในตำแหน่งที่เหมาะสมแล้ว เมื่อได้แล้วเราก็เลื่อนไปทางขวาหนึ่งตำแหน่งและทำการเปรียบเทียบข้อมูล ณ ตำแหน่งนี้กับข้อมูลตัวที่เล็กที่สุดที่อยู่ในแถวทางด้านขวา ถ้ามีเราก็สลับตำแหน่งเรื่อยๆ จนกว่าจะหมดข้อมูลในแถว โดยสามารถสรุปขั้นตอนได้ดังนี้

1. หาข้อมูลที่เล็กที่สุดทางด้านขวาของข้อมูลตัวแรก
2. สลับตำแหน่งของข้อมูลถ้าหาเจอ
3. เลื่อนไปทางขวาหนึ่งตำแหน่ง
4. ทำขั้นตอนทั้งสามจนกว่าจะหมดข้อมูลในแถว

ดังตัวอย่างภาพที่แสดงขั้นตอนที่เกิดขึ้นของการ sort ด้วยการใช้ Selection Sort




รูปที่ 4.19 ขั้นตอนการเรียงลำดับด้วย Selection Sort

จากรูปที่ 4.19 เราเริ่มต้นด้วยการเปรียบเทียบ 23 กับ ข้อมูลที่เล็กที่สุดทางขวา ซึ่งก็คือ 8 แล้วจึงสลับข้อมูลทั้งสอง เสร็จแล้วก็เราก็เลื่อนไปทางขวาหนึ่งตำแหน่ง เปรียบเทียบ 78 กับข้อมูลที่เล็กที่สุดทางขวาอีก ซึ่งก็คือ 23 สลับตำแหน่งของ 23 และ 78 เลื่อนไปทางขวาอีกหนึ่งตำแหน่งเปรียบเทียบ 45 แล้วจึงสลับตำแหน่งของ 45 กับ 23 เราทำการเปรียบเทียบและสลับข้อมูลในลักษณะนี้ไปจนกระทั่งเราเสร็จสิ้นการสลับตำแหน่งของ 78 กับ 56 ซึ่งเป็นข้อมูลคู่สุดท้ายที่มีการสลับตำแหน่ง

**2.2.3) การทำฐานข้อมูล (Template Database)**

จากกระบวนการที่ผ่านมานั้นจะได้ข้อมูลของพื้นที่สามเหลี่ยม ทำการจัดเรียงลำดับค่าจากน้อยไปมากดังรูปที่ 4.20 แสดงตัวอย่างการเรียงค่าพื้นที่สามเหลี่ยมจากลักษณะเฉพาะของลายนิ้วมือ (minutiae) การนำพื้นที่สามเหลี่ยมมาใช้เนื่องจากพื้นที่สามเหลี่ยมเป็นค่าที่ไม่ผันแปรแบบสัมพัทธ์ (Relative Invariance) ดังนั้นค่าที่ไม่ผันแปรสัมพัทธ์ของสามเหลี่ยมต้นฉบับคือ

$$I = \Delta I' \tag{4.7}$$

เมื่อ

$I$ คือพารามิเตอร์สำหรับการแปลง

$\Delta$ คืออัตราส่วนระหว่างค่าที่ไม่ผันแปรแบบสัมพัทธ์

ค่าที่ไม่ผันแปรแบบสมบูรณ์ (Absolute Invariance) นั้นจะหาได้จากการกำจัดแฟกเตอร์ของค่าที่ไม่ผันแปรแบบสัมพัทธ์สองค่า เช่น

$$I_1 = \Delta I_1' \tag{4.8}$$

$$I_2 = \Delta I_2' \tag{4.9}$$

จะได้ค่าที่ไม่ผันแปรแบบสัมบูรณ์เป็น

$$\frac{I_2}{I_1} = \frac{I_2'}{I_1'} \tag{4.10}$$

โดยค่าอัตราส่วนที่ได้นี้สามารถนำมาตรวจสอบได้ว่าเป็นสามเหลี่ยมชุดเดียวกันหรือไม่

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 162.5 | 32 | 46 | 672.5 | ---------- | 80 | 92 | 380 | 144 |

(a) ข้อมูลพื้นที่สามเหลี่ยมที่ต้องการทำการเรียงลำดับ

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 32 | 46 | 64 | 78 | ---------- | 1266.5 | 1503 | 1725 | 3221 |

(b). การเรียงลำดับพื้นที่สามเหลี่ยมจากน้อยไปมาก

รูปที่4.20 แสดงตัวอย่างการเรียงลำดับพื้นที่สามเหลี่ยม

เมื่อได้ทำการเรียงลพดับค่าพื้นที่สามเหลี่ยมจากน้อยไปมากเรียบร้อยแล้วนำค่าที่ได้มาทำการหาอัตราส่วนดังแสดงในรูปที่4.21 ซึ่งค่าที่ได้นี้จะเป็นค่าที่นำไปเปรียบเทียบตรวจสอบกับค่าอินพุทอันอื่นว่าใช่ลายนิ้วมือเดียวกันหรือไม่

| | | | | | | |
|---|---|---|---|---|---|---|
| $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | ---------- | $Y_{m-1}$ | $Y_m$ |

รูปที่4.21 จำนวนลำดับที่เรียงจากน้อยไปมาก

ดังนั้นจะได้

$$X_1 = \frac{Y_2}{Y_1} \tag{4.11}$$

$$X_2 = \frac{Y_3}{Y_2} \tag{4.12}$$

$$X_3 = \frac{Y_3}{Y_4} \tag{4.13}$$

ทำการหาไปจนถึง

$$X_n = \frac{Y_{n+1}}{Y_n} \tag{4.14}$$

โดยที่สมการที่ 4.14 เป็นสมการที่ใช้ทำการหาอัตราส่วนของพื้นที่สามเหลี่ยม

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 32 | 46 | 64 | 78 | ---------- | 1266.5 | 1503 | 1725 | 3221 |

(a)

| | | | | | | |
|---|---|---|---|---|---|---|
| 0.69565 | 0.71875 | 0.82051 | 0.975 | ---------- | 0.8713 | 0.53555 |

(b)

รูปที่4.22 แสดงอัตราส่วนที่หาจากพื้นที่สามเหลี่ยม a) อินพุทพื้นที่สามเหลี่ยม b)อัตราส่วนพื้นที่

**2.4) การเปรียบเทียบหรือการตรวจสอบลายนิ้วมือ**

นำค่าเทมเพลต(Template) กับอินพุทที่ได้ทำการเตรียมไว้แล้วมาทำการหาค่าความผิดพลาด โดยที่ค่าความผิดพลาดนั้นจะต้องมีการตั้งระดับเทรชโฮล์ไว้เพื่อใช้ทำการตรวจสอบค่าความผิดพลาด โดยจะมีขั้นตอนการตรวจสอบดังนี้

1. ทำการหาค่าความผิดพลาดระหว่างค่าอัตราส่วนของเทมเพลตกับอัตราส่วนของอินพุทดังสมการ

$$error = \left(\frac{ratio_T - ratio_I}{ratio_T}\right) \times 100\% \tag{4.15}$$

โดยที่

$ratio_T$ คือ อัตราส่วนของเทมเพลต

$ratio_I$ คือ อัตราส่วนของอินพุท

โดยจะต้องทำการหาค่าความผิดพลาดของทั้งหมด

2. ทำการตั้งระดับค่าความผิดพลาดเพื่อทำการตรวจสอบหาค่าความผิดพลาดที่ได้ในแต่ละตำแหน่งว่ายอมรับได้หรือไม่ ต้องทำการตรวจสอบทีละตำแหน่ง

3. หาความยาวของตำแหน่งของค่าความผิดพลาดที่ยอมรับได้และติดกันมากที่สุดโดยเปรียบเทียบกับค่าความผิดพลาดที่กำหนดไว้ ต่อจากนั้นทำการนำค่าผิดพลาดในช่วงนั้นมาทำการหาค่าเฉลี่ยของค่าผิดพลาด

4. เมื่อทำขั้นตอนทั้งสามเสร็จแล้วจะต้องทำการเลื่อนตำแหน่งของอัตราส่วนของอินพุทหรือทำการหมุนวนแล้วทำตามขั้นตอนที่ผ่านมา จะหยุดเมื่อตำแหน่งที่เลื่อนหรือหมุนวนกลับมาอยู่ที่เดิม

5. นำค่าความยาวของทั้งหมดที่ได้มานำมาหาค่าความยาวที่มากที่สุดพร้อมกับนำค่าความผิดพลาดเฉลี่ยในช่วงนั้นมาทำการตรวจสอบว่าใช่หรือไม่ใช่ดังสมการ

$$m = \begin{cases} yes, error_{long} \geq T_1, error_{mean} < T_e \\ no, error_{long} < T_1 \end{cases} \tag{5.16}$$

เมื่อ

Error_long คือ ค่าความยาวของค่าความผิดพลาดที่ยอมรับได้ติดต่อกันมากที่สุด

Error_mean คือ ค่าความผิดพลาดเฉลี่ย

$T_1$ คือ ค่าเทรชโฮล์ของค่าความยาวอัตราส่วน

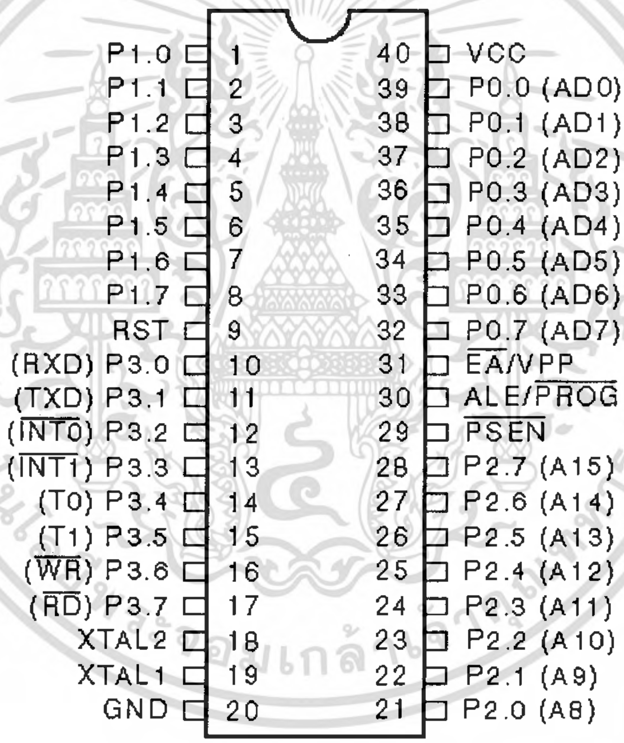
$T_2$ คือ ค่าเทรชโฮล์ของค่าความผิดพลาดเฉลี่ย

# บทที่ 5 ***
# อุปกรณ์ที่นำมาใช้งาน

## 5.1. ไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C51

มีโครงสร้างของหน่วยความจำสำหรับเก็บโปรแกรมเป็นแบบ Flash Memory ขนาด 4K Bytes ซึ่งสามารถโปรแกรมแล้วลบได้ 1,000 ครั้ง โดยใช้สัญญาณทางไฟฟ้าในการโปรแกรมและลบข้อมูล (Programmable and Erasable ROM: PEROM)

รายละเอียดไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C51 แสดงดังรูปที่ 4.1




รูปที่ 5.1 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C51

**การจัดขาของไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C51**

**Vcc** ใช้สำหรับต่อไฟเลี้ยง +5V

**GND** เป็นขากราวด์ ใช้สำหรับต่อกับกราวด์ของระบบ

*** บทที่ 5 อ้างอิงจากปริญญานิพนธ์ เรื่อง เครื่องสแกนลายนิ้วด้วยวิธีมุมและรัศมีที่จุดสำคัญ ปีการศึกษา 2549

**พอร์ต 0 (P0.0-P0.7)** มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุท สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุท สามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุทอิมพิแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุทได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรส และขาข้อมูล

**พอร์ต 1 (P1.0-P1.7)** มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุท สามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย

**พอร์ต 2 (P2.0-P2.7)** มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุทสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุทอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุทได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

**พอร์ต 3 (P3.0-P3.7)** มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุทสำหรับใช้งานทั่วไปโดยมีการต่อพูลอัพไว้ภายในแล้ว ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุทสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุทอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุทได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นตอนต่อไปนี้

**P3.0** ใช้เป็นขาอินพุทสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

**P3.1** ใช้เป็นขาอินพุทสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

**P3.2** ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัพท์จากภายนอกช่อง 0 หรือขา $\overline{\text{INT0}}$

**P3.3** ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัพท์จากภายนอกช่อง 1 หรือขา $\overline{\text{INT1}}$

**P3.4** ใช้เป็นขาอินพุทสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0

**P3.5** ใช้เป็นขาอินพุทสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 1 หรือขา T1

**P3.6** ใช้เป็นขาสัญญาณ $\overline{\text{WR}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

**P3.7** ใช้เป็นขาสัญญาณ $\overline{\text{RD}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

**RST รีเซต (Reset)** ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยใช้การป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

**$\overline{\text{ALE}}$/PROG (Address Latch Enable/Program pulse input)** เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

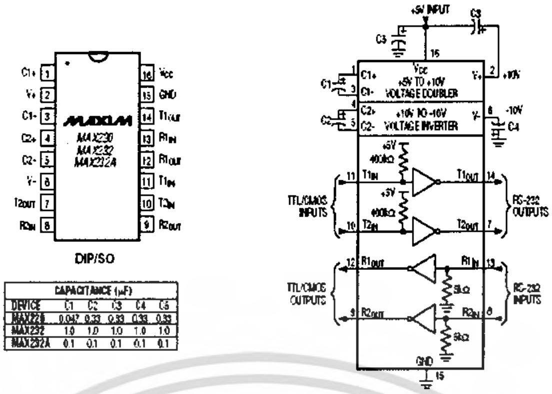
**$\overline{\text{PSEN}}$ (Program Store Enable)** ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้ง ในแต่ละแมชชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายในขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

**$\overline{\text{EA}}$/Vpp (External Access enable/Programming voltage input)** ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น “0” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น “1” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุทสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12V

**XTAL1 และ XTAL2** เป็นขาสำหรับต่อคริสตัลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

### 5.2 การเชื่อมต่อพอร์ตอนุกรมโดยใช้ไอซี MAX232

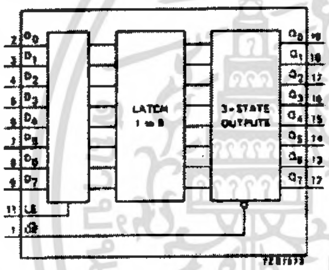
เนื่องจากไมโครคอนโทรลเลอร์ MCS-51 ใช้ไฟเลี้ยง +5V ทำให้เมื่อนำมาเชื่อมต่อกับพอร์ตอนุกรมของเครื่องคอมพิวเตอร์จึงไม่สามารถทำได้ เนื่องจากแรงดันในพอร์ตจะอยู่ระหว่าง 3V ถึง 12V ดังนั้นในการเชื่อมต่อจึงต้องใช้ไอซีไดร์เวอร์เฉพาะ ซึ่งในวงจรใช้ไอซีเบอร์ MAX232 ทำให้การรับส่งข้อมูลกับพอร์ตอนุกรมอยู่ภายใต้ระดับแรงดันทีทีแอลเหมือนกัน สายสัญญาณที่ใช้ในการเชื่อมต่อมีเพียง 3 เส้น คือ TxD, RxD และ GND



รูปที่ 5.2 ไอซี MAX232

## 5.3. การใช้งาน 74HC573

รายละเอียดไอซี 74HC573 แสดงดังรูปที่ 5.3

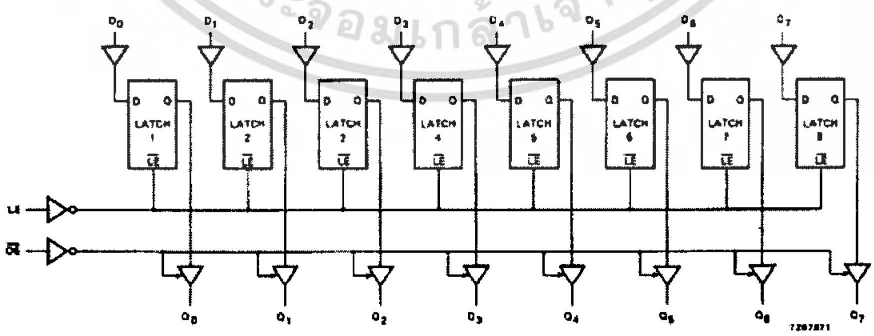


FUNCTION TABLE

| OPERATING MODES | INPUTS | | | INTERNAL LATCHES | OUTPUTS |
|---|---|---|---|---|---|
| | $\overline{OE}$ | LE | $D_N$ | | $Q_0$ to $Q_7$ |
| enable and read register (transparent mode) | L<br>L | H<br>H | L<br>H | L<br>H | L<br>H |
| latch and read register | L<br>L | L<br>L | l<br>h | L<br>H | L<br>H |
| latch register and disable outputs | H<br>H | L<br>L | l<br>h | L<br>H | Z<br>Z |

Notes

1. H = HIGH voltage level
   h = HIGH voltage level one set-up time prior to the HIGH-to-LOW LE transition
   L = LOW voltage level
   l = LOW voltage level one set-up time prior to the HIGH-to-LOW LE transition
   Z = high impedance OFF-state



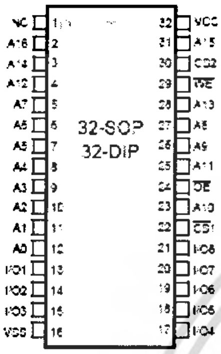
รูปที่ 5.3 ไอซี 74HC573

หน้าที่ของไอซี 74HC573 นั้นนำมาใช้เพื่อการ Latch ข้อมูลที่ได้มาจากตัวไมโครคอนโทรลเลอร์ MCS-51

## 5.4. การใช้งาน SRAM K6T1008C2E

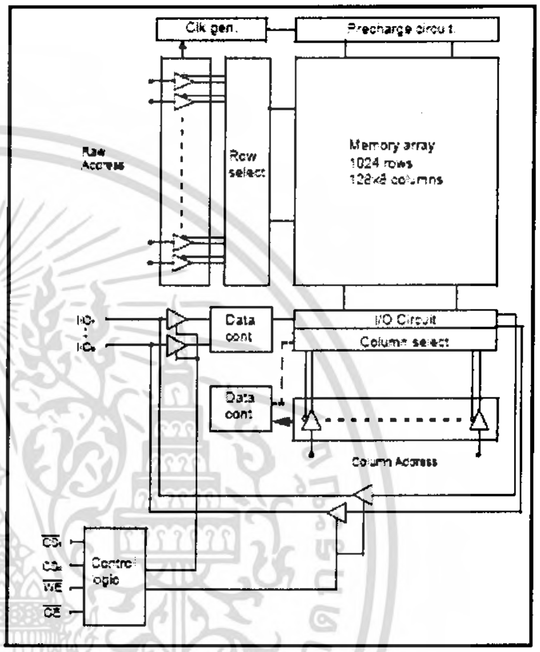
รายละเอียดไอซี SRAM K6T1008C2E แสดงดังรูปที่ 5.4

PIN DESCRIPTION




| Name | Function |
|---|---|
| $\overline{CS_1}$, $CS_2$ | Chip Select Input |
| $\overline{OE}$ | Output Enable Input |
| $\overline{WE}$ | Write Enable Input |
| $I/O_1$~$I/O_8$ | Data Inputs/Outputs |
| $A_0$~$A_{16}$ | Address Inputs |
| Vcc | Power |
| Vss | Ground |
| NC | No Connection |

FUNCTIONAL BLOCK DIAGRAM




รูปที่ 5.4 ไอซี SRAM K6T1008C2E

SRAM K6T1008C2E มีหน้าที่ในการเก็บข้อมูลที่ได้จากการสแกน เพื่อที่จะนำข้อมูลนี้ไปเปรียบเทียบกับข้อมูลที่เก็บในฐานข้อมูล

# บทที่ 6 ***

# การออกแบบอุปกรณ์เพื่อทำงานกับไอซี MBF200

## 6.1. หลักการและจุดประสงค์ในการออกแบบ

6.1.1. เพื่อให้สามารถใช้งานไอซี MBF200 ได้

6.1.2. เพื่อให้เครื่องสแกนลายนิ้วมือประมวลผลบนคอมพิวเตอร์ได้ ด้วยโปรแกรมLabVIEW และ MATLAB

6.1.3. เพื่อให้ชิ้นงานนี้สามารถประมวลผลการทำงานระบุบุคคลที่มีความผิดพลาดที่สามารถยอมรับได้

## 6.2. อุปกรณ์ที่นำมาใช้งาน

6.2.1. ไอซี MBF200

6.2.2. ไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C51

6.2.3. หน่วยความจำชั่วคราว (RAM)

6.2.4. ไอซี MAX232

6.2.5. ไอซี 74HC573

6.2.6. LCD 16*2

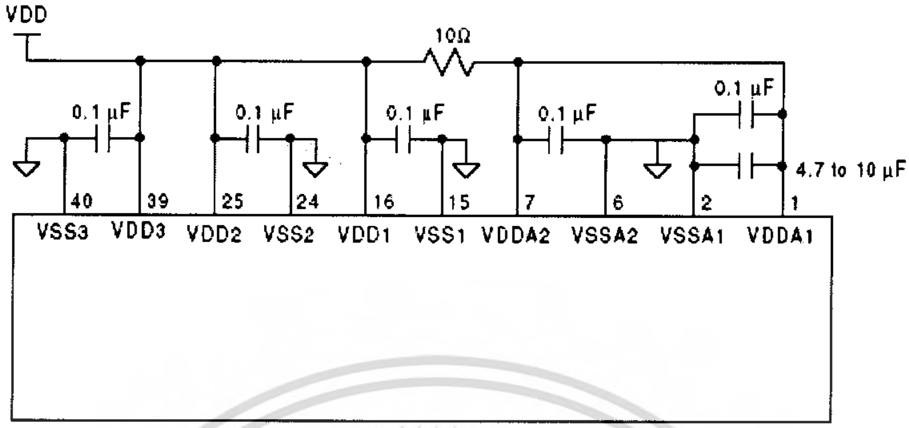
6.2.7. KEYPAD 4*3 DECODINGS

## 6.3 การออกแบบ

6.3.1. การออกแบบวงจรไอซี MBF200

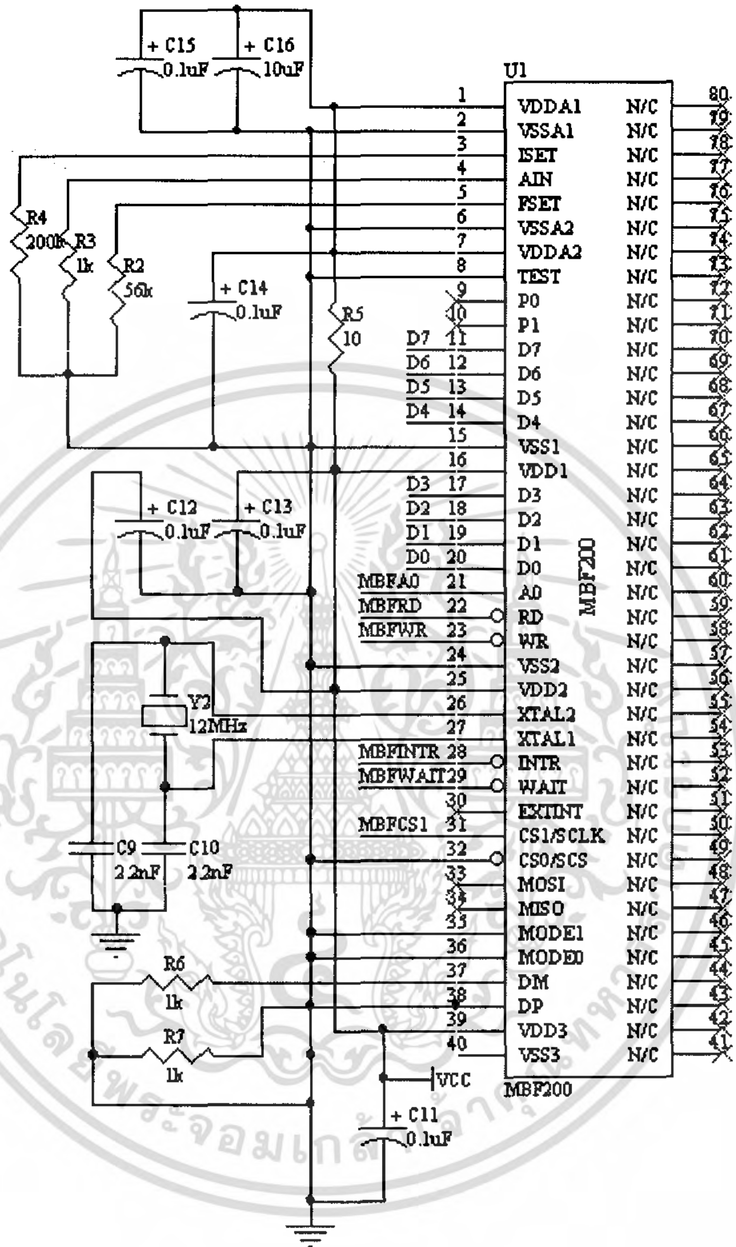
เนื่องจากตัวไอซี MBF200 มีการออกแบบการจ่ายไฟเลี้ยงให้อุปกรณ์ที่แน่นอนอยู่แล้ว ทำให้สามารถออกแบบตามที่ทางผู้ผลิตมีข้อมูลมาให้ซึ่งเป็นการต่อไฟเลี้ยงและกราวด์เพื่อจะลดสัญญาณรบกวน ทำให้ได้ภาพลายนิ้วมือที่ดีที่สุดจากเซ็นเซอร์ (Sensor) โดยวงจรจะแสดงดังรูปที่ 6.1

*** บทที่ 6 อ้างอิงจากปริญญานิพนธ์ เรื่อง เครื่องสแกนลายนิ้วด้วยวิธีมุมและรัศมีที่จุดสำคัญ ปีการศึกษา 2549




รูปที่ 6.1 แสดงการป้อนไฟเลี้ยงให้ไอซี MBF200

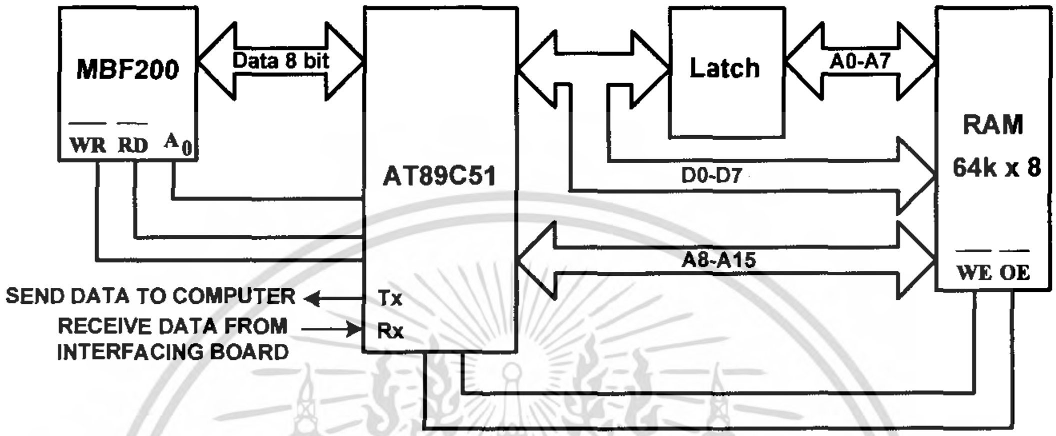
โดยการออกแบบได้ปฏิบัติตามกระบวนการที่ทางผู้ผลิตนั้นได้มีมาให้ เมื่อได้ออกแบบในส่วนของไฟเลี้ยงแล้วขั้นตอนต่อไปก็เริ่มออกแบบส่วนของตัวขาต่างๆ ของ MBF200 เพื่อให้ตัวไอซีมีการทำงานและสามารถใช้งานได้ตรงตามวัตถุประสงค์ ซึ่งในการออกแบบเพื่อใช้งานนี้ได้เลือกใช้การทำงานจากวงจรสร้างความถี่ภายนอก (External Crystal) โดยวงจรแสดงดังรูปที่ 6.2




รูปที่ 6.2 แสดงการออกแบบส่วนไอซี MBF200

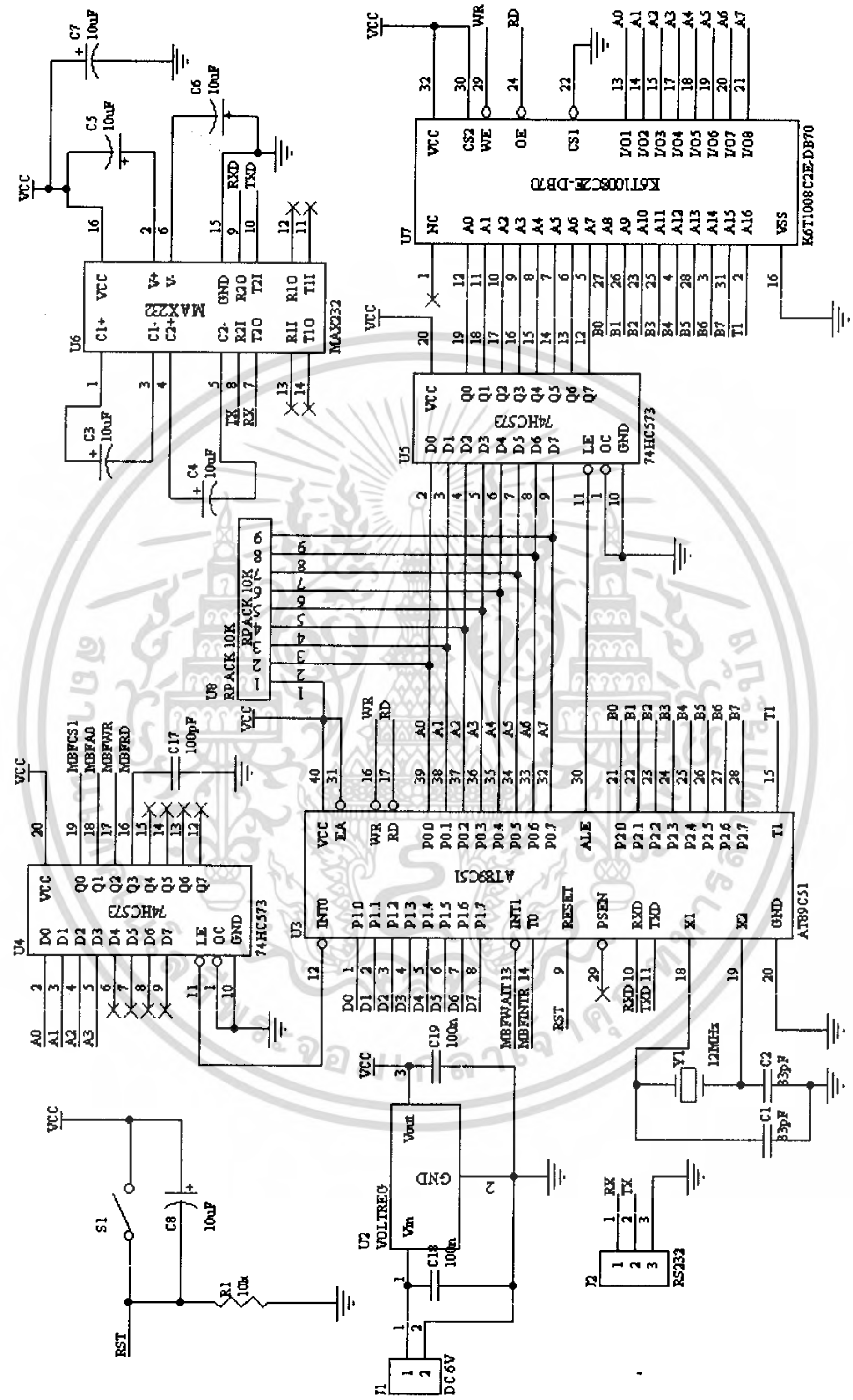
6.3.2. การออกแบบวงจรการทำงาน MCS-51, MBF200, RAM และ Serial port ให้ทำงานร่วมกัน

การออกแบบในส่วนของการนำเอาโมดูลต่าง ๆ มาทำงานร่วมกัน โดยการออกแบบให้ข้อมูลที่ได้มาจากไอซี MBF200 นั้นเข้ามาเก็บไว้ที่หน่วยความจำชั่วคราว แล้วส่งข้อมูลที่จัดเก็บไว้ในหน่วยความจำชั่วคราวไปยังคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม อธิบายได้ตามรูปที่ 6.3




รูปที่ 6.3 แสดงแผนภาพการออกแบบของวงจรที่ 1 SENSOR BOARD

จากรูปที่ 6.3 เป็นแผนผังการออกแบบตัวอุปกรณ์ของโครงงานนี้เมื่อนำหลักการดังกล่าวไปดำเนินการออกแบบและปฏิบัติจะได้วงจรการทำงานของอุปกรณ์นี้ดังรูปที่ 6.4 ซึ่งแต่ละส่วนได้แบ่งแยกการทำงานไว้เป็นสัดส่วน



รูปที่ 6.4 แสดงรูปการต่ออุปกรณ์ในวงจรที่ 1 SENSOR BOARD

วงจรประกอบด้วย 4 ส่วนหลักๆ คือ ไมโครคอนโทรลเลอร์, โมดูล MBF200, พอร์ตอนุกรมและหน่วยความจำชั่วคราวภายนอก

ในส่วนของไมโครคอนโทรลเลอร์เป็นหัวใจของการทำงานของวงจร เพราะจะต้องควบคุมการทำงานของส่วนอื่น ๆ ที่เหลือ ส่วนนี้จะใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มาเป็นตัวควบคุมการทำงาน

ในส่วนของหน่วยความจำชั่วคราวนั้นจะใช้หน่วยความจำชั่วคราวขนาด 64K x 8 bit ไมโครคอนโทรลเลอร์ตระกูล MCS-51 นั้นสามารถอ้างหน่วยความจำได้มากสุด 64K x 8 bit ซึ่งทำการเชื่อมต่อโดยใช้วงจรแลทช์เข้ามาช่วยและใช้พอร์ต P0 และ P2 ของไมโครคอนโทรลเลอร์ในการติดต่อ โดย P2 จะใช้อ้างแอดเดรสไบต์สูง ส่วน P0 จะใช้อ้างแอดเดรสไบต์ต่ำ นอกจากนั้นยังใช้ในการถ่ายโอนข้อมูลกับหน่วยความจำภายนอกด้วย

ในส่วนของการควบคุมโมดูล MBF200 นั้นสามารถควบคุมได้โดยการต่อขาสัญญาณ A0, $\overline{RD}$ และ $\overline{WR}$ ของไอซี MBF200 เข้ากับขาสัญญาณของไมโครคอนโทรลเลอร์และในการเปิดการใช้งานไอซี MBF200 นั้นจะต่อขา CS1 กับสัญญาณ 5 โวลต์ และต่อขา CS0 กับสัญญาณ 0 โวลต์ การเขียนและอ่านข้อมูลจาก MBF200 นั้นจะเป็นข้อมูลแบบ 8 บิตวิ่งขนานกันมาเข้าพอร์ต P1 ของไมโครคอนโทรลเลอร์ โดยใช้ขาสัญญาณ $\overline{RD}$ คุมการอ่านข้อมูล และ $\overline{WR}$ คุมการเขียนข้อมูล ส่วนขาสัญญาณ A0 นั้นใช้เลือกระหว่างการชี้ไปยัง Index register หรือ Data register ของ MBF200

ในส่วนของแหล่งจ่ายไฟของวงจรจะใช้ไฟเลี้ยงขนาด 5 โวลต์โดยใช้ไอซีเบอร์ 7805 แปลงระดับแรงดันจาก 9 โวลต์เป็น 5 โวลต์ อุปกรณ์ที่ได้สร้างขึ้นมีลักษณะดังรูปที่ 6.5

รูปที่ 6.5 แสดงอุปกรณ์ที่สร้างขึ้นเพื่อรับภาพลายนิ้วมือ

# บทที่ 7
# การออกแบบโปรแกรมเพื่อทำงานร่วมกับอุปกรณ์

## 7.1. จุดประสงค์ในการออกแบบ

7.1.1. เพื่อให้สามารถติดต่ออุปกรณ์ผ่านทางพอร์ตอนุกรมได้

7.1.2. เพื่อให้สามารถอ่านค่าและแสดงผลลายนิ้วมือที่ผ่านการสแกนได้

7.1.3. เพื่อให้สามารถจัดเก็บภาพลายนิ้วมือไว้ในคอมพิวเตอร์ได้

7.1.4. เพื่อให้สามารถวิเคราะห์หาจุดเด่นของภาพลายนิ้วมือได้

7.1.5. เพื่อให้เกิดความสะดวกในการใช้งาน

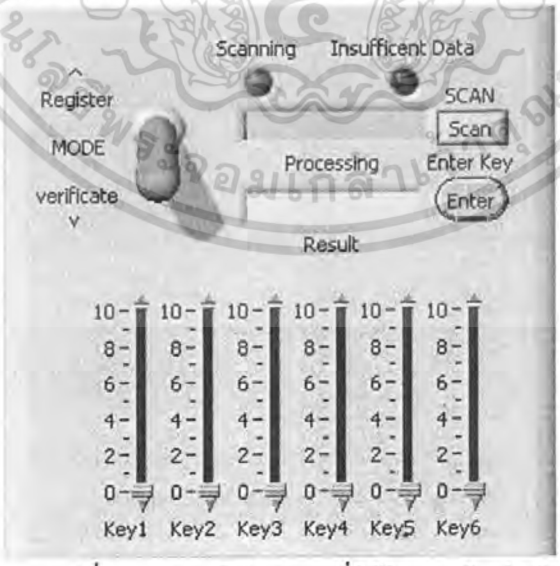
## 7.2. ซอฟต์แวร์ (Software) ที่นำมาออกแบบ

7.2.1. LabVIEW 8.2

7.2.2. MatLab R2006b

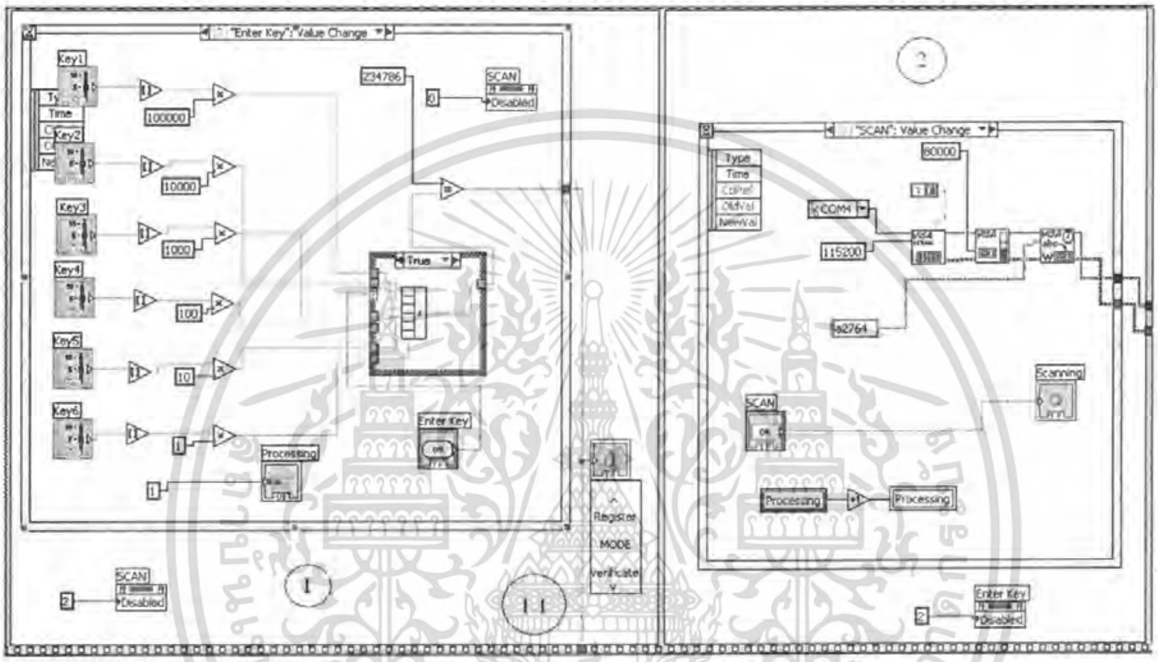
## 7.3. การออกแบบ

การออกแบบโปรแกรมที่ใช้ในการสแกนภาพลายนิ้วมือแสดงดังรูปที่ 7.1




รูปที่ 7.1 แสดงโปรแกรมที่ใช้ในการติดต่อกับผู้ใช้

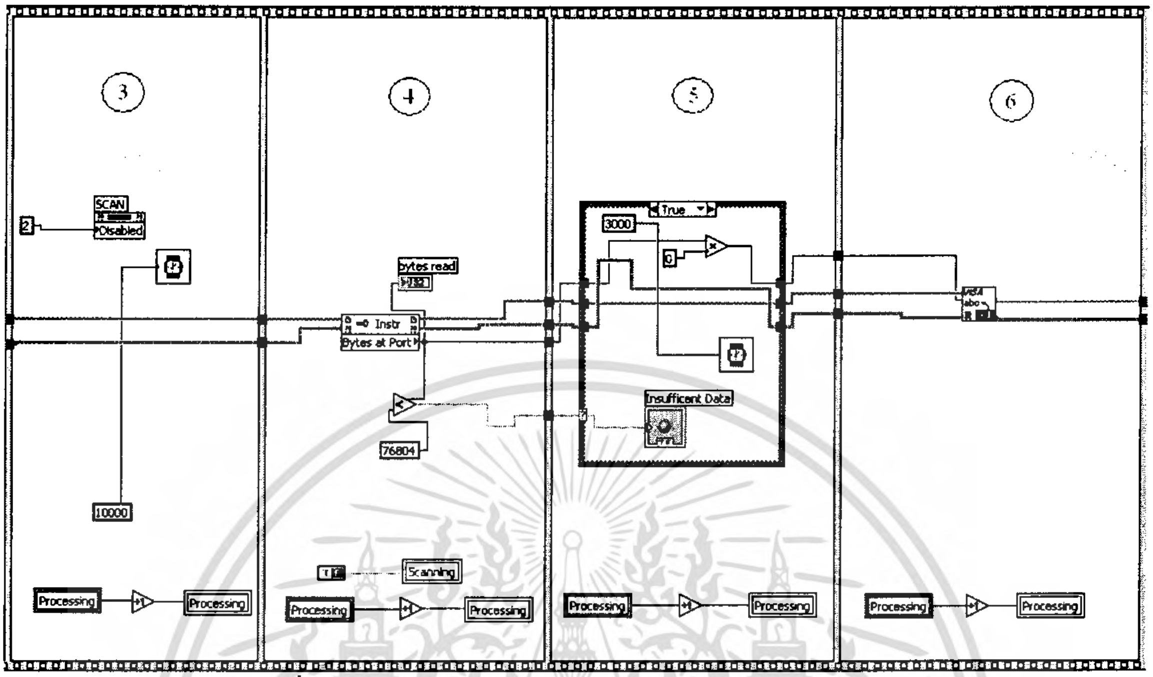
รูปด้านบนเป็นหน้าปัดที่ใช้ติดต่อกับผู้ใช้งาน ใช้การได้อย่างสะดวกเป็นขั้นตอน เข้าใจง่าย สามารถอธิบายได้ดังนี้ เนื่องจากโปรแกรมLabVIEWมีการส่งข้อมูลแบบ DATA FLOW ฉะนั้นเราจึงกำหนดSequence ให้เพื่อให้เกิดความเป็นลำดับขั้นตอน คือ




รูปที่ 7.2 แสดงการออกแบบโปรแกรมด้วย LabVIEW

Sequence1.) ป้อนPasswordว่าต้องการจะกำลังปฏิบัติการในModeใด(Verification or Register) โดยผ่านช่องทางที่1.1

Sequence2.) เริ่มทำการเปิดPort Serial เพื่อรับภาพที่จะสแกนเข้ามา และกดSCANเพื่อWRITE buffer เพื่อเป็นคำสั่งเริ่มทำการสแกน



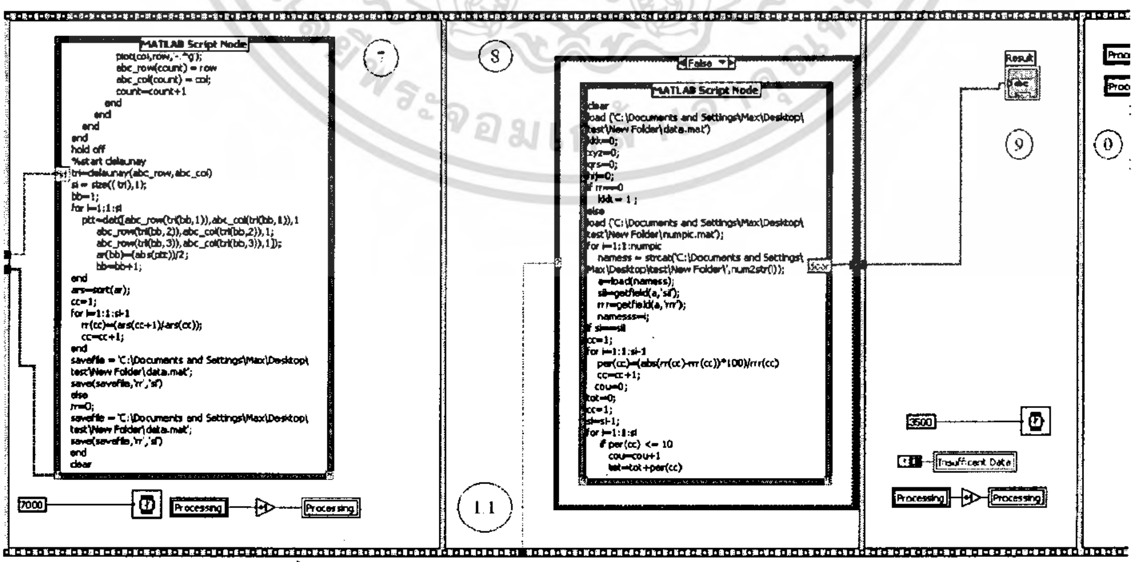

รูปที่ 7.3 แสดงการออกแบบโปรแกรมด้วย LabVIEW (ต่อ)

Sequence3.) Delay Times 10 วินาทีเครื่องสแกนกำลังทำการสแกน

Sequence4.) Check Property จำนวนข้อมูลที่สแกน เข้ามาว่าครบตามที่กำหนดหรือไม่

Sequence5.) ถ้าข้อมูลไม่ครบจะมีสัญญาณไฟสีแดง(Insufficient Data)เตือนขึ้น 3 วินาที

Sequence6.) READ DATA จากเครื่องสแกนเข้าสู่คอมพิวเตอร์




รูปที่ 7.4 แสดงการออกแบบโปรแกรมด้วย LabVIEW (ต่อ)

Sequence7.) เข้าสู่การประมวลผลของโปรแกรมMATLAB ในขั้นตอนของการPRE-POST Process

Sequence8.) เป็นส่วนที่ทำงานเปลี่ยนแปลงไปตามMODEการทำงานที่ได้เลือกไว้ จากช่องทางหมายเลข 1.1 (เส้นสีเขียว)

Verification Mode > ทำการMATCHING ระหว่างข้อมูลที่เข้ามากับ DATABASE

Register Mode > ทำการเซฟข้อมูลอัตราส่วนสามเหลี่ยมจากจุดสำคัญ เป็นFile นามสกุล.mat

Sequence9.) ส่วนนี้จะทำการแสดงผลที่ได้ผ่านกระบวนการมาทั้งหมดว่าสำเร็จหรือไม่

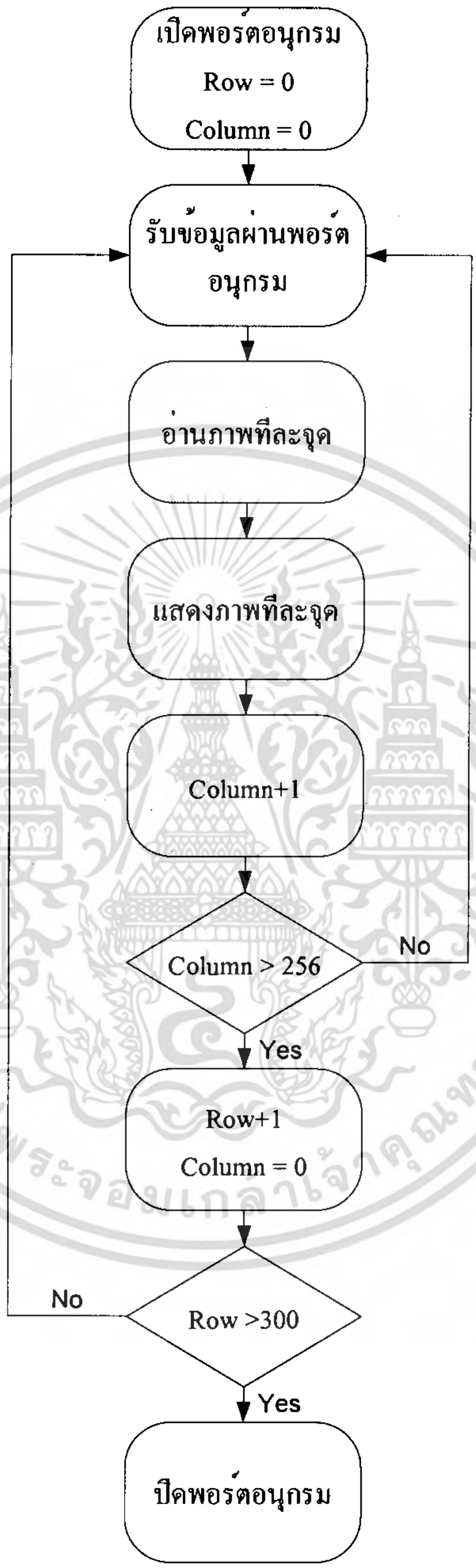
Sequence10.) ลำดับสุดท้ายเป็นการ Set Default ให้แก่หน้าปัดต่างๆเพื่อรอการเริ่มทำขั้นตอนในรอบต่อไป

Sequence ทั้งหมดจะถูกปิดล้อมด้วย While Loop เพื่อให้การทำงานของโปรแกรมนั้นเป็นไปอย่างต่อเนื่อง ไม่จำเป็นต้องหยุดการทำงานหากมีการเปลี่ยนแปลงMODEของการใช้งาน MODEการใช้งานโดยปกติจะSet Default ไว้ที่MODE Verification หากต้องการทำงานในMODE Register จะต้องป้อนPasswordที่ถูกต้องเพื่อที่จะสามารถลงทะเบียนเก็บข้อมูลของบุคคลนั้นไว้DATABASE ความจุของDATABASEนี้ขึ้นกับของความจุของHARDDISKของเครื่องๆนั้น หากเกิดการERROR จะมีหลอดPROCESSที่ทุกๆ Sequence เพื่อง่ายต่อการตรวจสอบแก้ไขข้อผิดพลาดว่า ERRORที่เกิดขึ้นนั้นอยู่ในขั้นตอนใด

## 7.4. แผนผังการทำงานของแต่ละขั้นตอนของโปรแกรมย่อย

7.4.1. โปรแกรมแสดงภาพที่สแกนผ่านพอร์ตอนุกรม

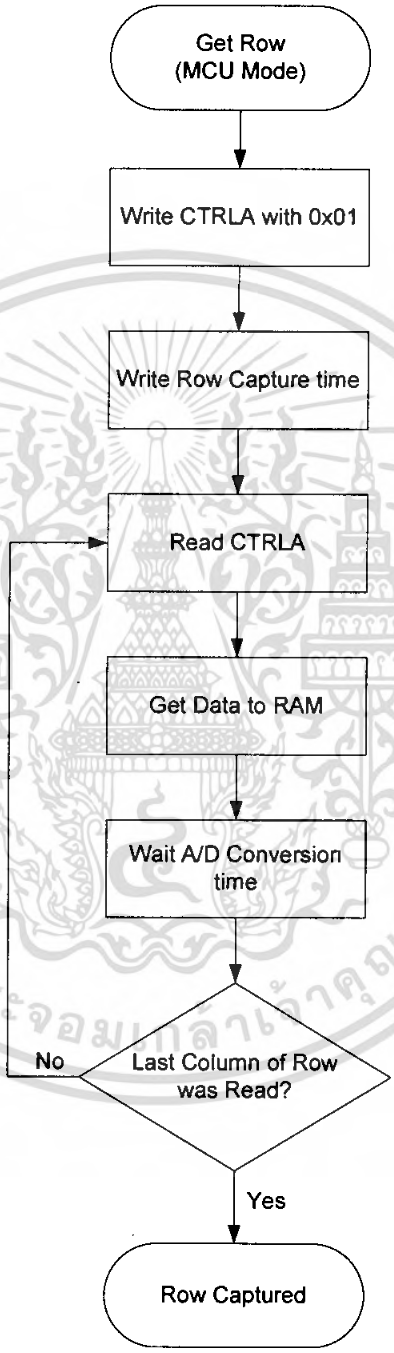
รายละเอียดในการเขียนตัวโปรแกรมแสดงดังแผนผังของรูปที่ 7.2




รูปที่ 7.5 แสดงแผนผังการออกแบบโปรแกรมแสดงภาพที่สแกนผ่านพอร์ตอนุกรม

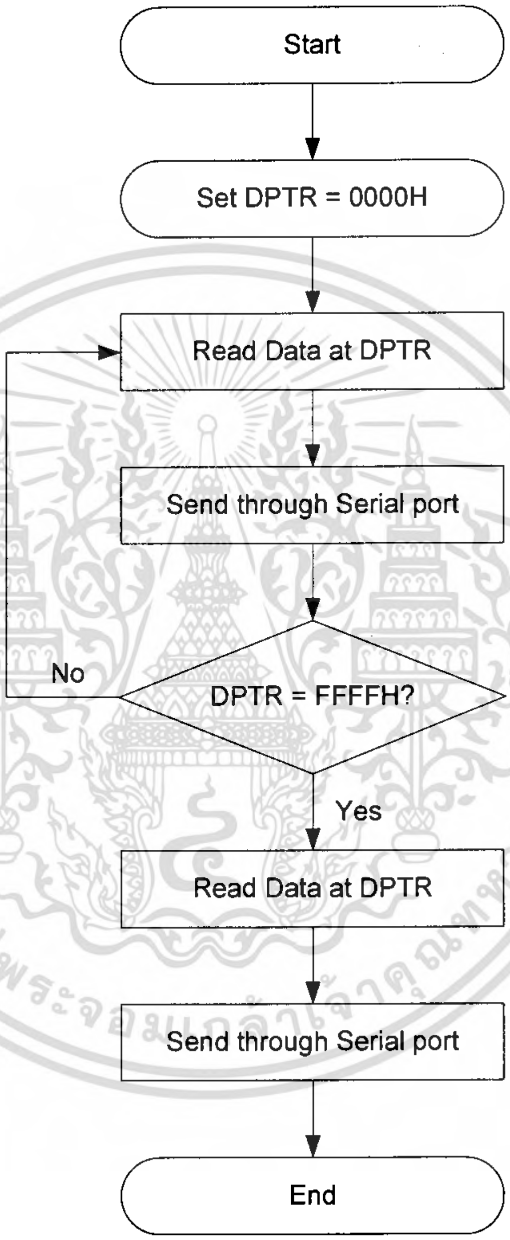
7.4.2. โปรแกรมควบคุมการทำงานการอ่านค่าจาก MBF200 โดยใช้ MCS-51

การอ่านข้อมูลมาจัดเก็บไว้ที่ตัวหน่วยความจำชั่วคราวโดยใช้ MCS-51 ในการจัดการมีรายละเอียดในการเขียนตัวโปรแกรมแสดงดังแผนผังของรูปที่ 7.3




รูปที่ 7.6 แสดงการทำงานของ MCS-51 ขณะดึงข้อมูลมาเก็บไว้ที่หน่วยความจำชั่วคราว

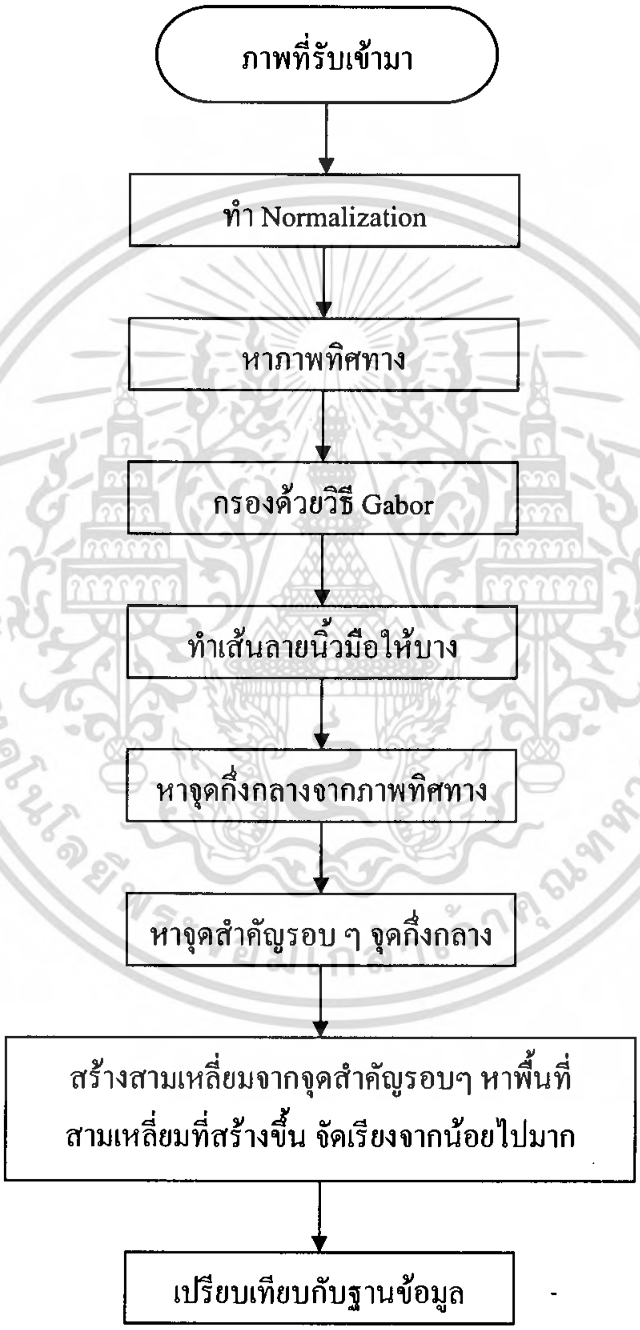
และการอ่านค่าข้อมูลลายนิ้วมือจากหน่วยความจำชั่วคราวส่งมาทางพอร์ตอนุกรม มีการทำงานดังนี้




รูปที่ 7.7 แสดงการทำงานของการดึงข้อมูลจากหน่วยความจำชั่วคราวไปยังคอมพิวเตอร์

7.4.3. โปรแกรมวิเคราะห์หาจุดเด่นของภาพลายนิ้วมือ

รายละเอียดในการเขียนตัวโปรแกรมด้วยโปรแกรมแล็บวิว(LabVIEW) โดยใช้การติดต่อประมวลผลด้วยโปรแกรมแมทแล็ป (Matlab) แสดงดังแผนผังของรูปที่ 7.5




รูปที่ 7.8 แสดงการทำงานของการวิเคราะห์หาจุดเด่นของภาพลายนิ้วมือ

# บทที่ 8

# การทดลองและผลการทดลอง

## 8.1. การทดลองการนำภาพลายนิ้วมือมาวิเคราะห์

โดยแบ่งการทำงานเป็นขั้นตอนดังนี้

8.1.1.การทำ Normalization

เนื่องจากภาพที่สแกนเข้ามาแต่ละภาพนั้น มีความเข้มต่างกัน จึงต้องมีการทำ Normalizeภาพ เพื่อให้ ความเข้มสีของภาพนั้นมีความเข้มประมาณเดียวกัน

(ก) (ข)

รูปที่ 8.1 ภาพลายนิ้วมือที่ได้จากการสแกน ( ก) แบบมัดหวาย (ข) แบบก้นหอย

(ก) (ข)

รูปที่ 8.2 ภาพลายนิ้วมือที่ผ่านการทำ Normalize ( ก) แบบมัดหวาย (ข) แบบก้นหอย

จากภาพผลการทดลองที่ได้จะพบว่าเมื่อผ่านการทำ Normalize แล้ว ภาพที่ได้จะมีความเข้มสีใกล้เคียงกัน

8.1.2.การทดลองการหาภาพทิศทาง (Orientation)


รูปที่ 8.3 ภาพทิศทางของลายนิ้วมือ รูปที่ 8.2 (ก)

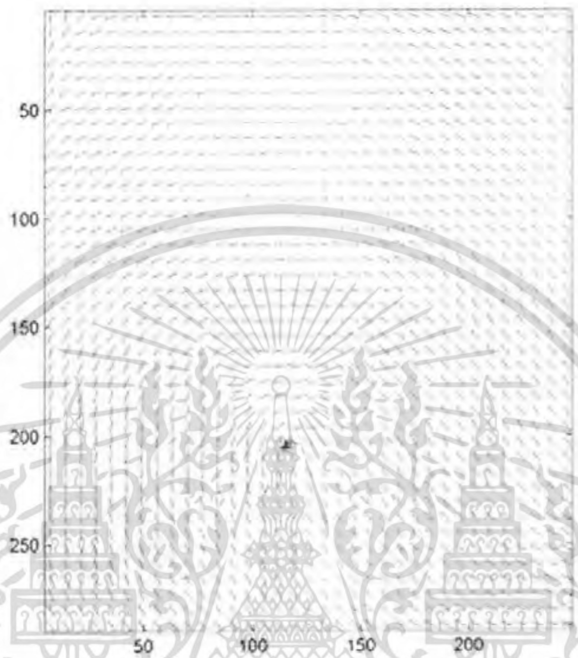
8.1.3.การทดลองการกรองลดสัญญาณรบกวน (Filtering)

รูปที่ 8.4 ภาพลายนิ้วมือที่ผ่านการกรองลดสัญญาณรบกวนของรูปที่ 8.2 (ก)

8.1.4.การทดลองการทำเส้นลายนิ้วมือให้บาง (Thinning)

รูปที่ 8.5 ภาพลายนิ้วมือที่ผ่านการทำเส้นให้บางของรูปที่8.4

8.1.5.การทดลองการหาจุดกึ่งกลางภาพทิศทาง (Core)



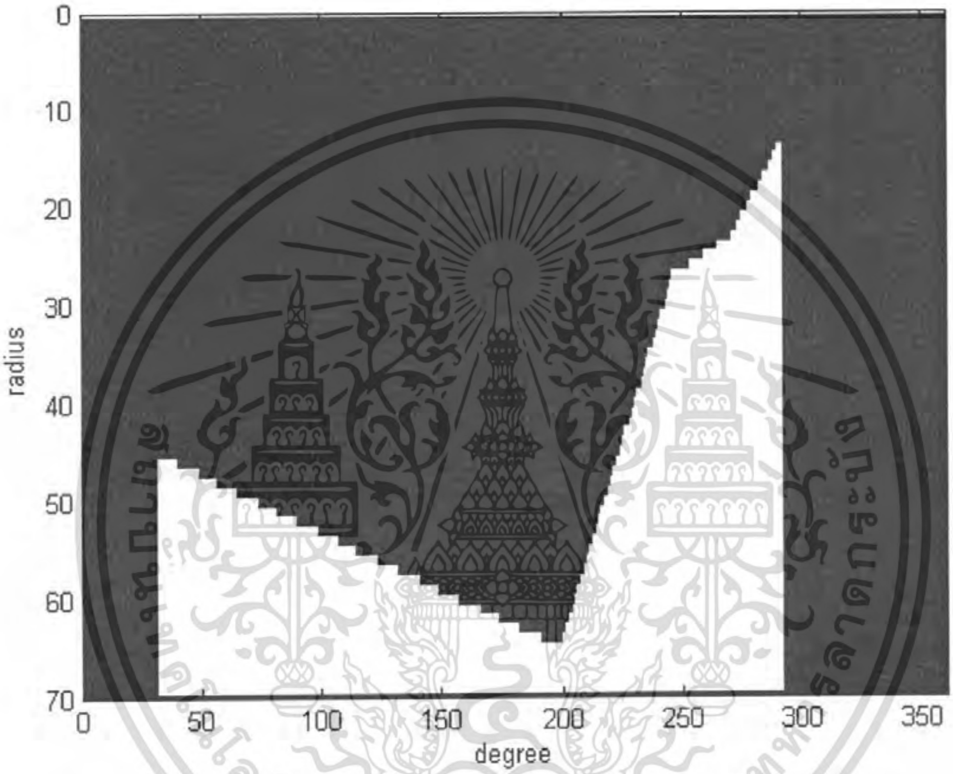

รูปที่ 8.6 ภาพแสดงตำแหน่งจุดกึ่งกลางที่ได้จากภาพทิศทาง รูปที่ 8.5

8.1.6.การทดลองการหาจุดสำคัญ (Minutiae Extraction)

**รูปที่ 8.7 ภาพแสดงตำแหน่งจุดสำคัญที่ได้จากภาพที่ผ่านการทำเส้นให้บาง รูปที่ 8.6**

**8.2.วิธีสร้างข้อมูลจากลักษณะเฉพาะ**

**8.2.1.การทดลองการเขียนกราฟความสัมพันธ์ระหว่างรัศมีที่ลากจุดกึ่งกลางถึงจุดสำคัญและมุมที่จุดสำคัญทำกับแกน X *****




รูปที่ 8.8 รูปกราฟความสัมพันธ์ระหว่างรัศมีที่ลากจากจุดกึ่งกลางถึงจุดสำคัญ และมุมที่จุดสำคัญทำกับแกน X

**8.2.1.1.**การทดลองการทำฐานข้อมูล **(Template Database)**ด้วยวิธี **Hard C-Means**

ตารางที่ 8.1. แสดงการหาค่าเฉลี่ยกลุ่มเพื่อเก็บเป็นฐานข้อมูลบุคคลที่ 1

| ลำดับที่ | | จุดที่ 1 | จุดที่ 2 | จุดที่ 3 | จุดที่ 4 | จุดที่ 5 | จุดที่ 6 | จุดที่ 7 | จุดที่ 8 | %error* |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | r | 13.6 | 45.4 | | | | - | | | 30.8532 |
| | θ | 186.5 | 253.68 | | | | | | | |
| 2 | r | 11.1 | 12.4 | 14.6 | 43.5 | 46.1 | 58 | | | 8.0159 |
| | θ | 341.41 | 291.78 | 174.47 | 248.66 | 134.86 | 73.195 | | | |

*** หัวข้อ8.2.1 อ้างอิงจากปริญญานิพนธ์ เรื่อง เครื่องสแกนลายนิ้วด้วยวิธีมุมและรัศมีที่จุดสำคัญ ปีการศึกษา 2549

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | r | 12.4 | 12.4 | 14.6 | 43.8 | 57.2 | 68.1 | 69.8 | | 5.7698 |
| | θ | 291.78 | 338.4 | 181.99 | 247.66 | 66.678 | 113.3 | 264.71 | | |
| 4 | r | 11.9 | 12.1 | 12.6 | 39.9 | 44.7 | 57.9 | 64.3 | 69.4 | 7.5317 |
| | θ | 297.79 | 341.91 | 172.96 | 250.17 | 134.86 | 67.681 | 278.74 | 119.32 | |
| 5 | r | 11.8 | 12.4 | 15.6 | 43.1 | 46.9 | 59.2 | | | 8.6071 |
| | θ | 347.43 | 291.78 | 178.48 | 250.17 | 137.37 | 72.694 | | | |
| 6 | r | 13.4 | 14.7 | 15.6 | 45.4 | 52 | 60.9 | 69.1 | | 6.3571 |
| | θ | 289.77 | 336.9 | 181.99 | 132.35 | 252.17 | 68.182 | 262.7 | | |
| 7 | r | 12.1 | 13.4 | 13.6 | 45.6 | 47.5 | 61.3 | 68.8 | | 5.0476 |
| | θ | 341.91 | 289.77 | 173.46 | 128.84 | 249.67 | 67.179 | 264.71 | | |
| 8 | r | 9.7 | 10.7 | 17.6 | 45.6 | 50.4 | 61.3 | | | 9.5437 |
| | θ | 348.43 | 278.24 | 173.96 | 245.66 | 134.86 | 76.203 | | | |
| 9 | r | 9.9 | 10.2 | 15.6 | 43.8 | 45.6 | 59.9 | | | 6.4325 |
| | θ | 284.76 | 338.9 | 178.48 | 244.65 | 130.35 | 65.675 | | | |
| 10 | r | 11.5 | 13 | 14.6 | 41.9 | 50.5 | 52.7 | 62.1 | | 7.004 |
| | θ | 336.4 | 285.76 | 173.46 | 243.65 | 254.18 | 250.17 | 63.67 | | |
| 11 | r | 8.9 | 11.1 | 15.6 | 45.3 | 63.5 | | | | 17.4841 |
| | θ | 286.77 | 341.41 | 178.48 | 125.84 | 260.19 | | | | |
| 12 | r | 10.2 | 11.8 | 15.6 | 42.5 | 58.6 | 69 | | | 5.0516 |
| | θ | 290.27 | 347.43 | 174.97 | 242.15 | 61.665 | 108.29 | | | |
| 13 | r | 11.1 | 12.4 | 14.6 | 44.7 | 47.8 | 58.5 | | | 6.293 |
| | θ | 341.41 | 291.78 | 181.99 | 131.35 | 252.17 | 66.177 | | | |
| 14 | r | 8 | 14 | 15 | 42.8 | 44.7 | 56.6 | 65.2 | 67.8 | 1.8849 |
| | θ | 288.27 | 344.92 | 165.44 | 244.15 | 131.35 | 63.169 | 261.7 | 110.29 | |
| 15 | r | 6.7 | 21.9 | 37 | 49.8 | 59.6 | 61.3 | 61.9 | | 16.3651 |
| | θ | 13.035 | 159.43 | 223.6 | 264.21 | 268.72 | 256.18 | 71.19 | | |
| 16 | r | 7.7 | 10.7 | 16 | 42.9 | 59.8 | 63.8 | | | 7.7143 |
| | θ | 281.25 | 348.93 | 165.94 | 241.14 | 63.67 | 258.69 | | | |
| 17 | r | 8.7 | 11.1 | 16.6 | 43.3 | 45.9 | 59.8 | 63.7 | | 4.1865 |
| | θ | 280.25 | 341.41 | 174.97 | 240.14 | 126.84 | 63.67 | 259.19 | | |
| 18 | r | 9.9 | 10.2 | 14.6 | 43.8 | 46 | 63.1 | 64 | | 3.5437 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | θ | 284.76 | 338.9 | 174.47 | 244.65 | 120.82 | 61.665 | 263.2 | | |
| 19 | r | 11.6 | 15.1 | 39.8 | 57.2 | 65 | | | | 13.5278 |
| | θ | 187.5 | 333.39 | 129.85 | 56.15 | 108.79 | | | | |
| 20 | r | 9.7 | 11.5 | 15.6 | 45.9 | 48.2 | 59.1 | 66.7 | | 5.4087 |
| | θ | 279.25 | 336.4 | 174.97 | 126.84 | 244.65 | 60.662 | 259.19 | | |
| 21 | r | 7.7 | 9.9 | 15 | 41.6 | 45 | 57.3 | 65.8 | 68.4 | 3.4524 |
| | θ | 281.25 | 343.92 | 165.44 | 238.64 | 129.35 | 60.161 | 258.69 | 109.29 | |
| 22 | r | 9.9 | 10.2 | 14.6 | 45 | 45.1 | 60.3 | | | 5.8254 |
| | θ | 284.76 | 338.9 | 174.47 | 127.84 | 244.15 | 62.667 | | | |
| 23 | r | 12.6 | 13.3 | 14.3 | 41.3 | 44.2 | 55.9 | | | 8.3135 |
| | θ | 187 | 329.88 | 288.77 | 131.85 | 246.66 | 64.673 | | | |
| 24 | r | 9.9 | 9.9 | 15 | 43.3 | 45 | 58.6 | 69 | | 5.4484 |
| | θ | 284.76 | 343.92 | 165.44 | 240.14 | 127.84 | 59.659 | 263.2 | | |
| 25 | r | 7.7 | 10.9 | 14.8 | 43.8 | 46.4 | 60.8 | | | 7.1706 |
| | θ | 281.25 | 344.92 | 168.45 | 244.65 | 123.83 | 61.665 | | | |
| 26 | r | 8 | 10.6 | 17.6 | 39.8 | 60.5 | 61.9 | | | 6.0992 |
| | θ | 288.27 | 352.44 | 173.96 | 240.14 | 66.678 | 263.7 | | | |
| 27 | r | 10.2 | 10.9 | 16.6 | 43.8 | 45.3 | 58.6 | 63.8 | 68 | 2.6349 |
| | θ | 290.27 | 344.92 | 173.96 | 241.64 | 125.84 | 59.659 | 258.69 | 105.28 | |
| 28 | r | 9.3 | 14.3 | 15 | 43.8 | 44.9 | 59.3 | 62.1 | 69.1 | 5.1389 |
| | θ | 292.28 | 340.91 | 165.44 | 241.64 | 123.33 | 56.651 | 257.19 | 102.27 | |
| 29 | r | 8.7 | 12.1 | 15.8 | 44.2 | 44.9 | 58.9 | 63.5 | 69.1 | 5.9524 |
| | θ | 280.25 | 341.91 | 168.95 | 240.64 | 123.33 | 54.145 | 255.68 | 102.27 | |
| 30 | r | 10.6 | 11.8 | 15.6 | 46.1 | 46.4 | 60 | | | 7.373 |
| | θ | 333.89 | 282.25 | 173.96 | 241.64 | 123.83 | 59.158 | | | |
| ข้อมูลเฉลี่ย | r | 10.9418 | 9.84211 | 15.5119 | 44.5488 | 45.5625 | 59.3771 | 66.083 | 68.2636 | |
| | θ | 341.906 | 284.945 | 174.019 | 246.918 | 126.777 | 63.7546 | 262.332 | 110.416 | |

%error* คือค่าที่ได้จากจากการเปรียบเทียบข้อมูลแต่ละข้อมูลกับ ข้อมูลเฉลี่ยที่เก็บเป็นฐานข้อมูลของแต่ละบุคคลคิดเป็นเปอร์เซ็นต์

**8.2.1.2 การทดลองการเปรียบเทียบลายนิ้วมือ**

8.2.1.2.1 การทดลองการเปรียบเทียบลายนิ้วมือของบุคคลเดียวกันและได้ผลว่าเป็นบุคคลเดียวกัน

(ก) (ข)

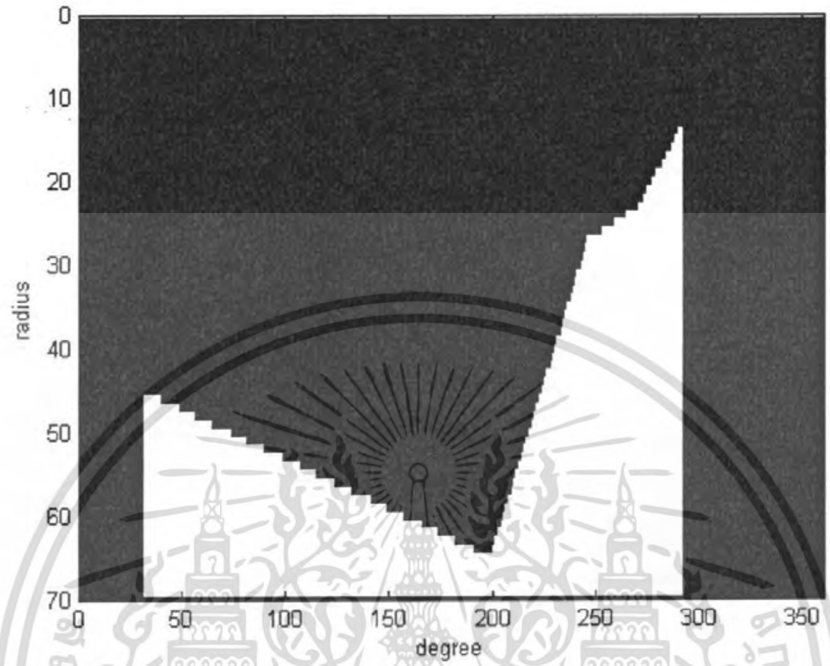
รูปที่ 8.9 รูปลายนิ้วมือของบุคคลที่ 1 (ก)ภาพที่ 1 (ข)ภาพที่ 2

ตารางที่ 8.2. แสดงค่ารัศมีและมุมของบุคคลที่ 1 (ภาพที่ 1) ซึ่งต่อไปนี้จะให้เป็นบุคคลอ้างอิง

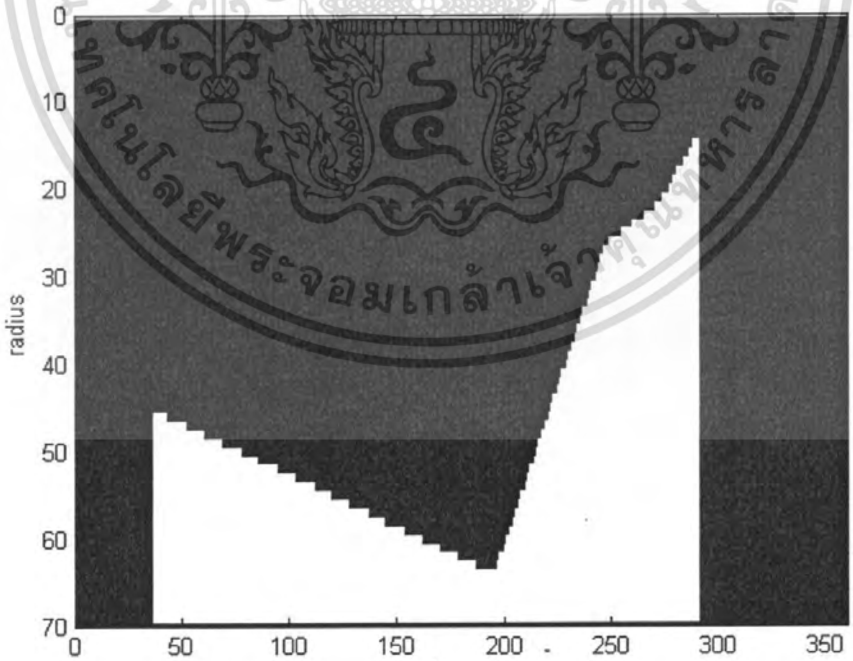
| | | | | | |
|---|---|---|---|---|---|
| **R** | 24.9 | 6 | 43.8 | 48.25 | 56.18 |
| **Θ** | 35.14 | 194.18 | 246.42 | 270.91 | 289.78 |

ตารางที่ 8.3. แสดงค่ารัศมีและมุมที่ ของบุคคลที่ 1 (ภาพที่ 2)

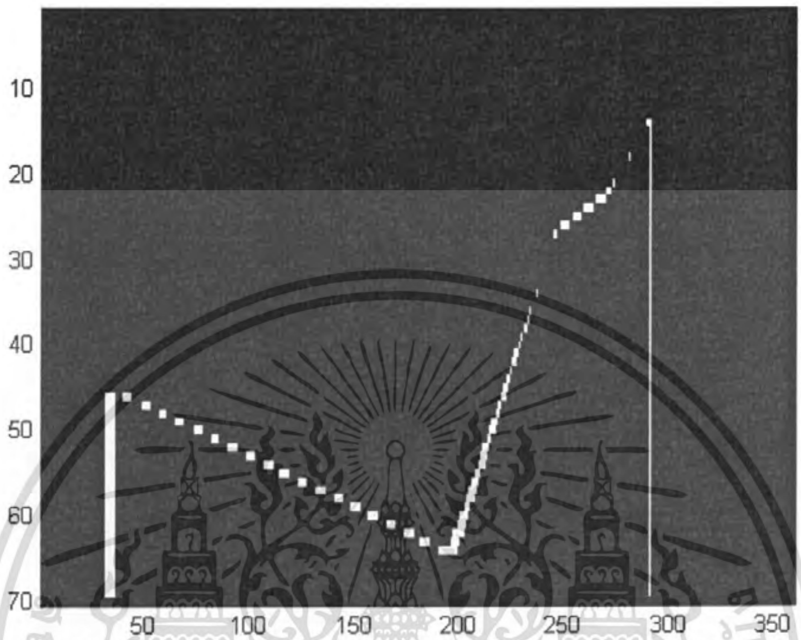
| | | | | | |
|---|---|---|---|---|---|
| **r** | 25 | 5 | 43 | 47 | 57 |
| **Θ** | 30.424 | 197.5 | 244.94 | 269.69 | 290.32 |




รูปที่ 8.10 รูปกราฟความสัมพันธ์ระหว่างรัศมี และมุมของบุคคลอ้างอิง




รูปที่ 8.11 รูปกราฟความสัมพันธ์ระหว่างรัศมี และมุมของบุคคลเดิม (ภาพที่ 2)




รูปที่ 8.12 เปรียบเทียบกราฟแสดงรัศมีและมุมของบุคคลคนเดียวกัน ได้ผลใกล้เคียงกัน ผลการเปรียบเทียบจะได้ว่าเป็นลายนิ้วมือของบุคคลเดียวกัน (Match)

8.2.1.2.2 การทดลองการเปรียบเทียบลายนิ้วมือของบุคคลเดียวกันและได้ผลต่างกัน

(ก) (ข)

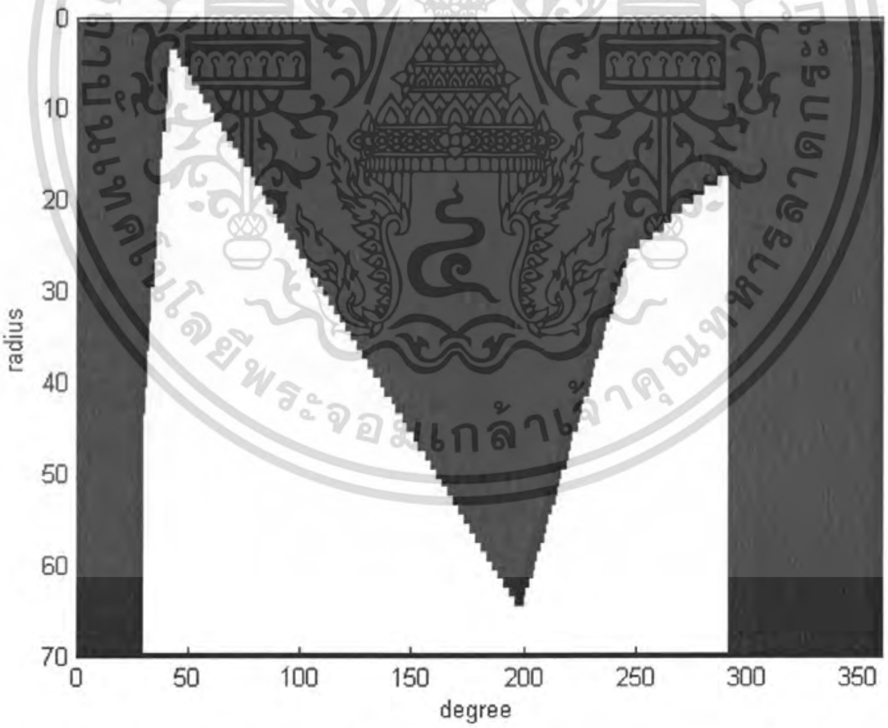
**รูปที่ 8.13 รูปลายนิ้วมือของบุคคลที่ 1 (ก)ภาพที่ 1 (ข)ภาพที่ 3**

ตารางที่ 8.4. แสดงค่ารัศมีและมุมของบุคคลที่ 1 (ภาพที่ 1) ซึ่งเป็นบุคคลอ้างอิง

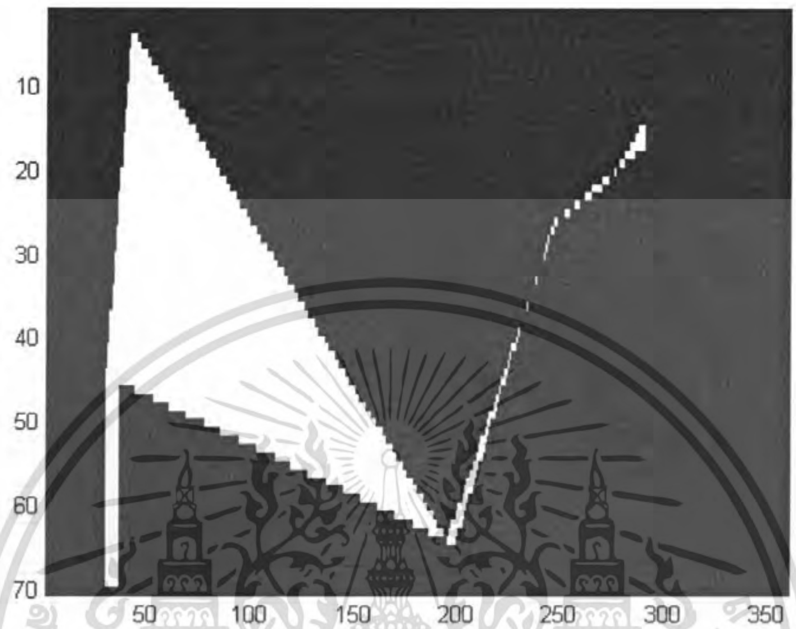
| r | 24.9 | 6 | 43.8 | 48.25 | 56.18 |
|---|---|---|---|---|---|
| θ | 35.14 | 194.18 | 246.42 | 270.91 | 289.78 |

ตารางที่ 8.5. แสดงค่ารัศมีและมุมของบุคคลที่ 1 (ภาพที่ 3)

| r | 26 | 67 | 5 | 44 | 53 |
|---|---|---|---|---|---|
| θ | 28.877 | 40.737 | 197.5 | 245.46 | 289.29 |




รูปที่ 8.14 รูปกราฟความสัมพันธ์ระหว่างรัศมี และมุมของบุคคลเดิม (ภาพที่ 3)




รูปที่ 8.15 เปรียบเทียบกราฟแสดงรัศมีและมุมของบุคคลคนเดียวกัน ได้ผลต่างกัน

ผลการเปรียบเทียบจะได้ว่าเป็นลายนิ้วมือของบุคคลต่างกัน (Mismatch) ถือเป็นความผิดพลาดแบบการปฏิเสธการผ่านของผู้ใช้ที่ถูกต้อง (FRR)

8.2.1.2.3 การทดลองการเปรียบเทียบลายนิ้วมือชนิดเดียวกันของบุคคลต่างกัน ได้ผลต่างกัน

(ก) (ข)

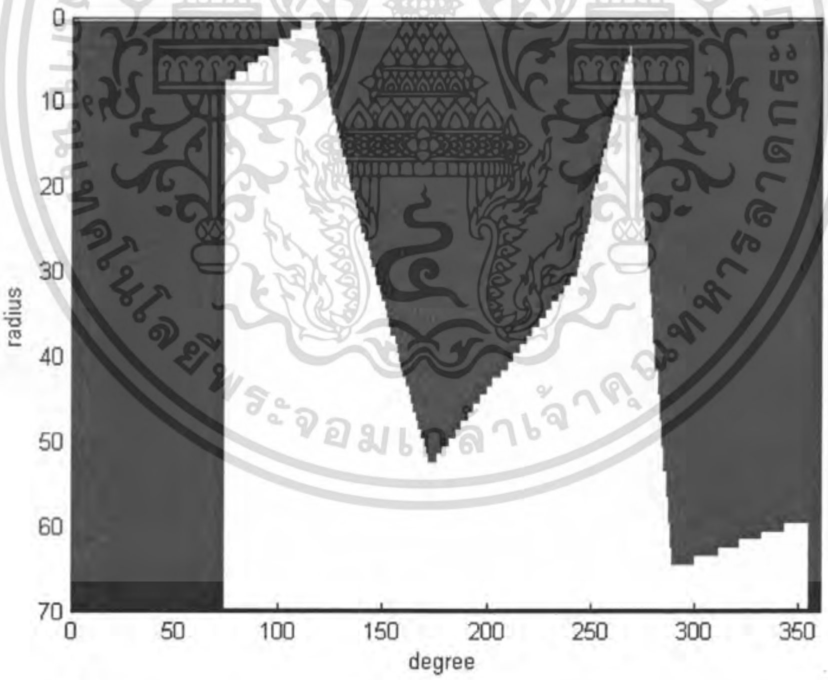
รูปที่ 8.16 (ก) รูปลายนิ้วมือชนิดเดียวกันของบุคคลที่ 1 (ข)รูปลายนิ้วมือชนิดเดียวกันของบุคคลที่ 2

ตารางที่ 8.6. แสดงค่ารัศมีและมุมของบุคคลที่ 1 (ภาพที่ 1) ซึ่งเป็นบุคคลอ้างอิง

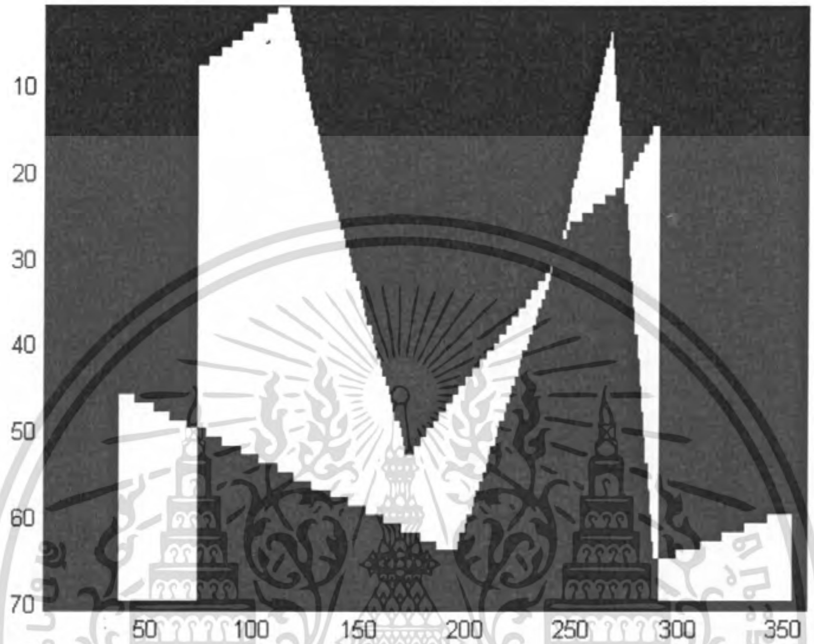
| r | 24.9 | 6 | 43.8 | 48.25 | 56.18 |
|---|---|---|---|---|---|
| Θ | 35.14 | 194.18 | 246.42 | 270.91 | 289.78 |

ตารางที่ 8.7. แสดงค่ารัศมีและมุมของบุคคลที่ 2 ที่มีลายนิ้วมือแบบเดียวกัน

| r | 62 | 70 | 49 | 17 | 40 | 67 | 5 | 11 |
|---|---|---|---|---|---|---|---|---|
| θ | 72.708 | 114.48 | 134.07 | 171.72 | 241.33 | 267.11 | 287.74 | 352.2 |




รูปที่ 8.17 รูปกราฟความสัมพันธ์ระหว่างรัศมี และมุมของบุคคลต่างกัน แต่เป็นลายนิ้วมือชนิดเดียวกันบุคคลที่ 2




รูปที่ 8.18 เปรียบเทียบกราฟแสดงรัศมีและมุมของลายนิ้วมือชนิดเดียวกันของบุคคลต่างกัน ผลการเปรียบเทียบจะได้ว่าเป็นลายนิ้วมือของบุคคลต่างกัน (Mismatch)

8.2.1.2.4 การทดลองการเปรียบเทียบลายนิ้วมือชนิดเดียวกันของบุคคลต่างกันได้ผลต่างกัน

(ก)

(ข)

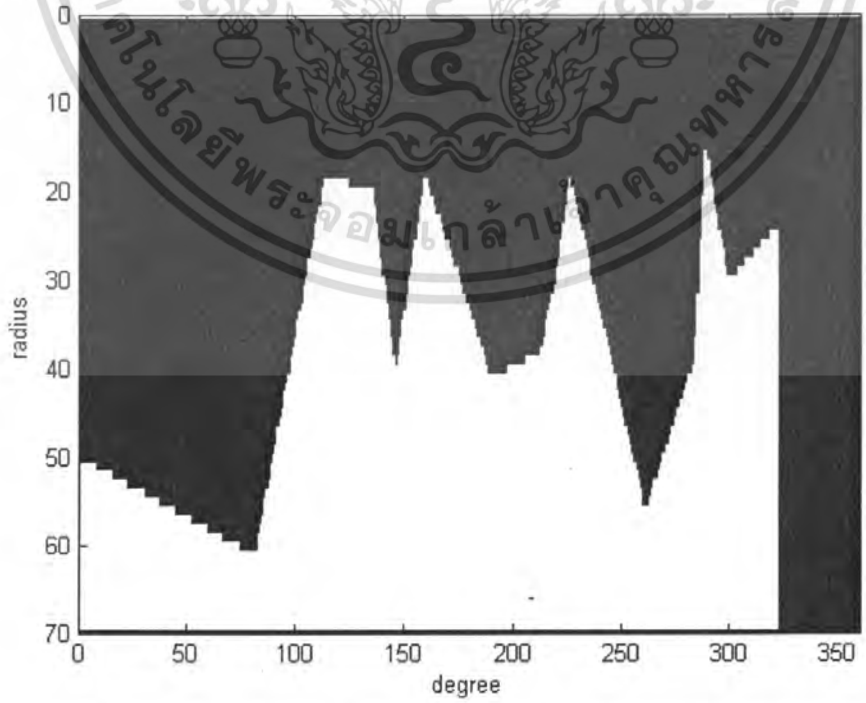
รูปที่ 8.19 (ก) รูปลายนิ้วมือต่างชนิดกันของบุคคลที่ 1 (ข)รูปลายนิ้วมือต่างชนิดกันของบุคคลที่ 3

ตารางที่ 8.8. แสดงค่ารัศมีและมุมของบุคคลที่ 1 (ภาพที่ 1) ซึ่งเป็นบุคคลอ้างอิง

| r | 24.9 | 6 | 43.8 | 48.25 | 56.18 |
|---|---|---|---|---|---|
| θ | 35.14 | 194.18 | 246.42 | 270.91 | 289.78 |

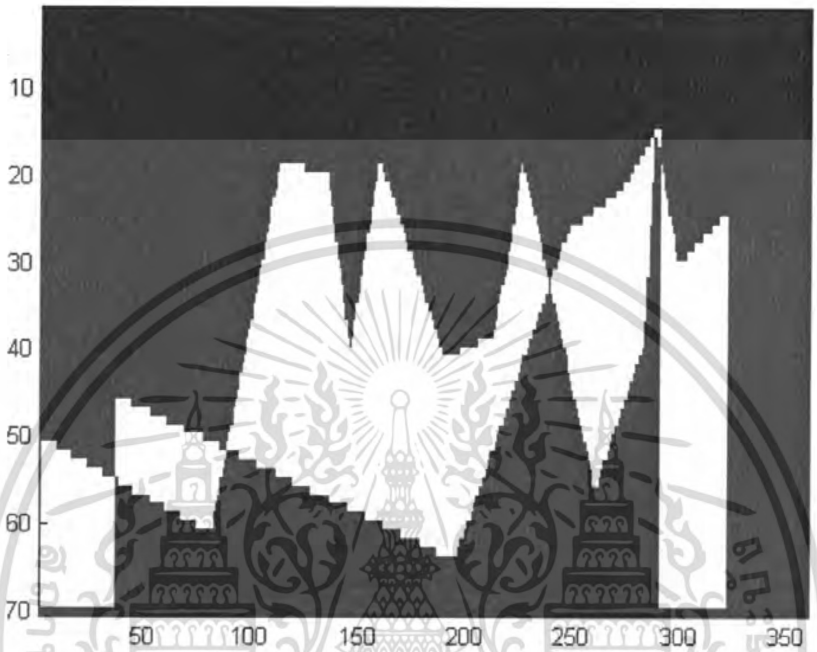
ตารางที่ 8.9. แสดงค่ารัศมีและมุมของบุคคลที่ 3 ที่มีลายนิ้วมือต่างชนิดกัน

| r | 20 | 9 | 52 | 50 | 30 | 52 | 29 | 32 |
|---|---|---|---|---|---|---|---|---|
| θ | 0 | 80.443 | 112.41 | 134.07 | 144.9 | 158.31 | 189.25 | 211.42 |
| r | 42 | 52 | 14 | 31 | 56 | 40 | 46 | |
| θ | 219.67 | 224.83 | 259.89 | 282.58 | 287.22 | 298.05 | 321.77 | |




รูปที่ 8.20 รูปกราฟความสัมพันธ์ระหว่างรัศมี และมุมของบุคคลต่างกัน

ที่เป็นลายนิ้วมือต่างชนิดกันบุคคลที่ 3

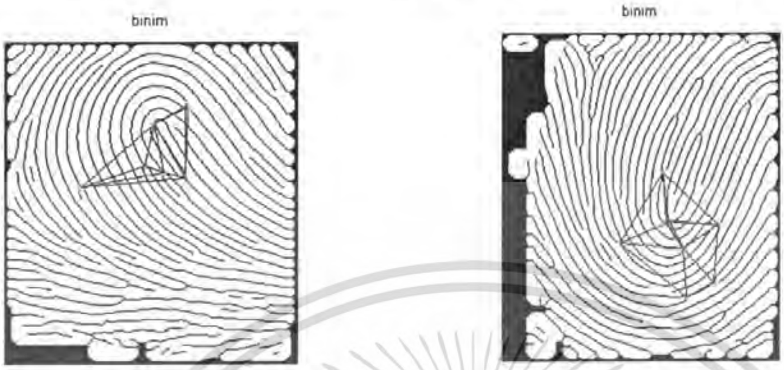



รูปที่ 8.21 เปรียบเทียบกราฟแสดงรัศมีและมุมของลายนิ้วมือต่างชนิดกันของบุคคลต่างกัน ผลการเปรียบเทียบจะได้ว่าเป็นลายนิ้วมือของบุคคลต่างกัน (Mismatch)

**8.2.2. การทดลองเปรียบเทียบลายนิ้วมือด้วยการสร้างสามเหลี่ยมจากกลุ่มของจุดลักษณะเฉพาะด้วยวิธีของ Delaunay**

ขั้นตอนนี้เป็นอีกวิธีที่สามารถนำมาใช้ในการเปรียบเทียบข้อมูลระหว่างข้อมูลใน Template กับข้อมูลinput ว่าใช่หรือไม่ใช่คนเดียวกันหรือไม่ ดังนี้

8.2.2.1 การสร้างสามเหลี่ยมจากจุดลักษณะเฉพาะ (Minutiae)




รูปที่8.22 แสดงกลุ่มของสามเหลี่ยมที่สร้างขึ้นจากMinutiae ตัวอย่างที่จะนำมาทำการทดลองโดยรูปซ้าย-ขวาจะเป็นTemplate 1 และ 2 ตามลำดับ

8.2.2.2 การหาพื้นที่สามเหลี่ยมแต่ละรูปและการจัดเตรียมTemplate

เมื่อทำการสร้างสามเหลี่ยมเรียบร้อยแล้วเราจะนำค่าจุดยอดของของสามเหลี่ยมที่เกิดขึ้นมาแทนสูตรหาพื้นที่สามเหลี่ยมนั่นคือ พื้นที่สามเหลี่ยมเท่ากับดีเทอมิแนนซ์ของจุดยอดทั้ง3วางเรียงตัวกันแล้วมีคอลัมน์ที่3 ของพิกัดเป็น $(x_i,y_i,z_i) = (x_i,y_i,1)$ เหตุที่ต้องใส่เป็น1 เพราะต้องทำให้เกิดเป็นแมทตริกจตุรัส 3x3 จึงสามารถหาค่าดีเทอมิแนนซ์ได้ เมื่อได้ค่าพื้นที่สามเหลี่ยมครบทั้งหมดแล้วนำมาเรียงเป็นลำดับจากน้อยไปมากโดยการใช้วิธี selection sort ของMat-lab เราจะได้arrayที่เป็นชุดของมูลของพื้นที่สามเหลี่ยมที่เรียงลำดับจากน้อยไปมากซึ่งนี้ยังนำมาใช้การไม่ได้ เราจึงต้องใช้อัตราส่วนของพื้นที่สามเหลี่ยมที่เรียงลำดับไว้แล้ว โดยที่จะนำค่าทางด้านขวาของตัวแรกซึ่งมีค่ามากกว่า1ลำดับเป็นตัวตั้งหารด้วยค่าของตัวที่น้อยกว่าดังสมการ $X_n = \frac{Y_{n+1}}{Y_n}$ เสร็จสิ้นขั้นตอนนี้เราจะได้Template ที่พร้อมใช้ในการเปรียบเทียบลายนิ้วมือ

8.2.2.3 การเปรียบเทียบลายนิ้วมือและขั้นตอนการตัดสินใจ

เราจะยกตัวอย่างTemplateจากรูป4.22 เปรียบเทียบกับข้อมูลที่ได้จากการSimulateลายนิ้วมือข้างเคียงที่เป็นประเภทเดียวกันคนเดียวกันหรือประเภทเดียวกันแต่คนละบุคคลกันดังนี้

| ข้อมูล Template อ้างอิง | 9.5 | 3.3158 | 4.5952 | 1.0743 | 1.0547 | 1.1463 | 2.1782 | 1.1368 |
|---|---|---|---|---|---|---|---|---|
| Input_1 | 1.1388 | 1.0278 | 1.7027 | 1.1914 | 1.3782 | | | |
| Error | 88.013 | 69.003 | 62.946 | 10.900 | 30.672 | | | |
| Input_2 | 9.5 | 3.3684 | 4.8594 | 1.0418 | 1.0123 | 1.7561 | 1.0903 | 1.4825 |
| Error | 0.000 | 1.586 | 5.749 | 3.025 | 4.020 | 53.197 | 49.945 | 30.410 |

ตารางที่8.10เปรียบเทียบ input ต่างๆกับtemplateที่1 (รูปที่4.22 ซ้ายมือ)

เราจะทำการเปรียบเทียบข้อมูลโดยการหาerror ที่ลำดับแต่ละลำดับ เมื่อหาแล้วจะนำค่าerrorที่อยู่ในค่าที่ยอมรับได้และติดกันมากที่สุด (จากตารางจะเห็นได้ว่าinput1 errorมีค่ามากเกินค่าที่ยอมรับได้และไม่มีที่สามารถยอมรับได้เลยแสดงให้เห็นว่าเป็นคนละบุคคลหรือinputที่รับมาไม่สมบรูณ์อย่างมาก) ในการทดลองนี้ตั้งค่า%Errorไว้ไม่เกิน 12% เพราะฉะนั้น ค่า Error_long จะเป็นช่วง 0.00-4.020 มีความยาวเป็น 5/8 column length (เรากำหนด $T_l$ = 50%ของcolumn) นำเปอร์เซ็นต์ในช่วงนี้มาหาค่าเฉลี่ยความผิดพลาดจะได้ Error_Mean = 2.8588% (เรากำหนด $T_e$ = 10%) ซึ่งInput2 นี้ถือได้ว่าเป็นข้อมูลที่สมบรูณ์มากเกิดค่าผิดพลาดเพียงเล็กน้อย

ขั้นตอนการตัดสินใจ Yes/No จะพิจารณาสมการดังที่ได้กล่าวไว้คือ

$$m = \begin{cases} yes, error_{long} \geq T_1, error_{mean} \leq T_e \\ no, error_{long} < T_1 \end{cases}$$

ในกรณีInput2 M = Yes จาก 5/8 >= 50%column length & 2.8588% < 10%

เพราะฉะนั้น ในCase Input2 จะถือว่า Match!!

<u>ตัวอย่างที่ 2</u>

<u>ตารางที่8.11เปรียบเทียบ input ต่างๆกับtemplateที่2 (รูปที่4.22 ขวามือ)</u>

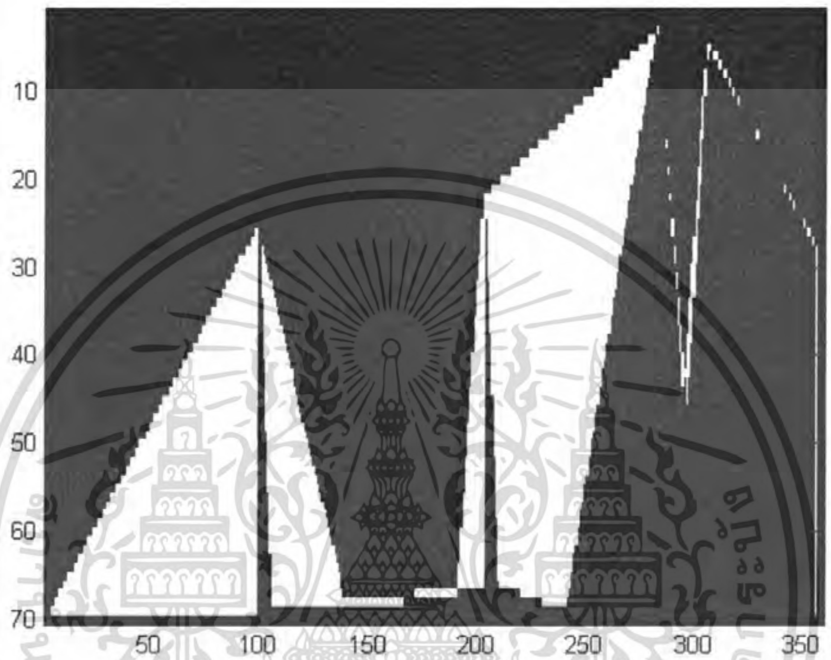
| ข้อมูล Template อ้างอิง | 2.6667 | 2.75 | 3.6364 | 1 | 1.2375 | 9.5354 | 1.1345 | 1.0803 | 1.2939 | 1.2211 | 1.0372 | 1.2621 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input_1 | 2 | 2 | 19.25 | 1.0649 | 1.2439 | 8.8431 | 1.214 | 1.0228 | 1.3402 | 1.1872 | 1.0988 | 1.2523 |
| Error | 25.001 | 27.273 | 429.37 | 6.490 | 0.517 | 7.260 | 7.007 | 5.323 | 3.578 | 2.776 | 5.939 | 0.776 |
| Input_2 | 1.1762 | 1.0228 | 1.3402 | 1.3045 | 1.0322 | 1.2133 | | | | | | |
| Error | 55.893 | 62.807 | 63.145 | 30.450 | 16.590 | 87.276 | | | | | | |

Input2 ถือว่าใช้การไม่ได้เพราะเกิดปัญหาในขั้นตอนการรับภาพหรือการหาจุดลักษณะเฉพาะทำให้เกิดอัตราส่วนของสามเหลี่ยมไม่ครบทำให้ไม่สามารถนำมาเปรียบเทียบได้ พิจารณาInput1 พบช่วงของค่าความผิดพลาดที่ยอมรับได้และติดกันมากที่สุดคือช่วง 6.49-0.776 % คือเป็นความยาว 9/12 column length นั่นหมายถึงเกิน50% column length เพราะฉะนั้น Error_long > =$T_1$ ผ่านเงื่อนไขแรก จากนั้นหาค่า

Error_Mean = (6.49+0.517+7.26+7.007+5.323+3.578+2.776+5.939+0.776) /9 = 4.407%

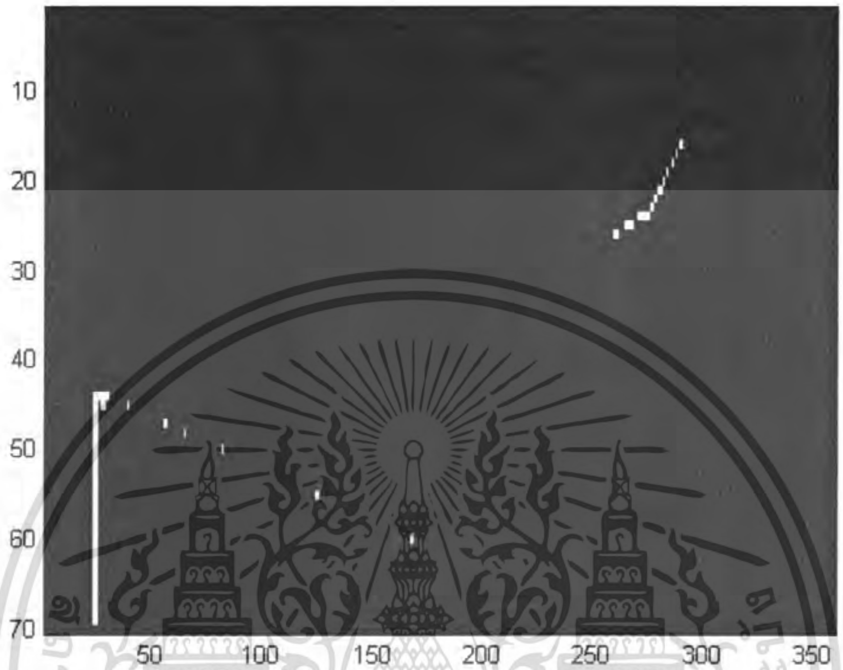
4.407% < 10% (Error_Mean < $T_2$) ผ่านเงื่อนไขที่สอง สรุปผล M = Yes [Match!!]

**8.2.3** เปรียบเทียบผลการเปรียบเทียบระหว่างวิธีความสัมพันธ์รัศมีและมุมสำคัญกับวิธีเทียบอัตราส่วนสามเหลี่ยม**Delaunay** ที่ลายนิ้วมือเดียว **Template** และ **Input** เดียวกัน




รูปที่8.23 รูปที่นำกราฟความสัมพันธ์รัศมีกับมุมสำคัญมาซ้อนกันเพื่อทำการเปรียบเทียบลายนิ้วมือ ของตาราง4.11 (Input1)

จากรูป8.23 เป็นรูปที่ได้จากrun Template เดียวกันกับตารางที่8.11และเป็นinputที่1 จากกราฟแสดงให้เห็นถึงความผิดพลาดที่เกิดขึ้นมีค่ามาก (ส่วนที่เป็นสีขาว)ถึง 24.8175 % ในวิธีแบบกราฟความสัมพันธ์รัศมีกับมุมสำคัญถือว่า Mismatch แต่ในวิธีของDelaunay ดังตัวอย่างที่2 ค่าError_Mean ที่เกิดขึ้นมีเพียง 4.407% ถือว่าMatch ซึ่งอันที่จริงแล้วลายนิ้วมือ Input1นี้เป็นบุคคลเดียวกันกับTemplate data แสดงถึงวิธี Delaunay มีความแม่นยำมากกว่า




รูปที่8.24 เป็นรูปที่นำกราฟความสัมพันธ์รัศมีกับมุมสำคัญมาซ้อนกันเพื่อทำการเปรียบเทียบลายนิ้วมือของตาราง8.10 (Input2)

จากรูปกราฟของTemplateกับInput2 เกือบซ้อนทับกันสนิทมีค่าความผิดพลาดเพียง 0.0357% เป็นค่าความผิดพลาดที่น้อยมากสรุปผลว่า Match แต่ส่วนวิธีของDelaunay นั้นได้ค่าความผิดพลาดเฉลี่ยที่คำนวณได้เท่ากับ 2.8588% ถึงแม้ว่าความแม่นยำจะจะน้อยกว่าวิธีกราฟความสัมพันธ์แต่ผลที่สรุปจากเงื่อนไขยังถือว่าลายนิ้วมือทั้งสองยัง Match กันอยู่ แสดงให้เห็นถึงความแม่นยำที่ตัดสินได้แม่นยำกว่าของวิธีDelaunay

**8.3 การทดลองหาค่า False Rejection Rate : FRR และ False Acceptance Rate : FAR**

โดยเราจะเริ่มหาค่าFARจากคน30คน โดยมีลายนิ้วมือของคนหนึ่งที่ใช้อ้างอิงเป็นลายนิ้วมือแบบมัดหวาย (30คนนี้ อาจจะไม่ใช้มัดหวายหรือลายใดลายหนึ่งเพียงอย่างเดียว)

หมายเหตุ O/C = Out of Case (ไม่เข้ากรณีที่สามารถทำการmatchได้ )

100% หมายถึง ค่าที่ERRORที่เกิดขึ้นเกินที่กำหนด จะSET ERRORให้เป็น100%

ตารางที่ 8.13 แสดงผลการหาค่า FAR จากการสแกนลายนิ้วมือของบุคคล 30 บุคคล เมื่อลายนิ้วมือที่อ้างอิงเป็นลายนิ้วมือแบบมัดหวาย

| ครั้งที่ | เปอร์เซ็นต์ความคลาดเคลื่อน (%) | Match / Mismatch |
|---|---|---|
| 1 | 100 | Mismatch |
| 2 | 100 | Mismatch |
| 3 | 100 | Mismatch |
| 4 | 100 | Mismatch |
| 5 | O/C | Mismatch |
| 6 | O/C | Mismatch |
| 7 | 100 | Mismatch |
| 8 | O/C | Mismatch |
| 9 | 100 | Mismatch |
| 10 | 100 | Mismatch |
| 11 | 100 | Mismatch |
| 12 | 100 | Mismatch |
| 13 | 100 | Mismatch |
| 14 | 100 | Mismatch |
| 15 | O/C | Mismatch |
| 16 | O/C | Mismatch |
| 17 | O/C | Mismatch |
| 18 | 100 | Mismatch |
| 19 | O/C | Mismatch |
| 20 | O/C | Mismatch |
| 21 | O/C | Mismatch |
| 22 | 100 | Mismatch |
| 23 | O/C | Mismatch |
| 24 | 100 | Mismatch |

| | | |
|---|---|---|
| 25 | O/C | Mismatch |
| 26 | O/C | Mismatch |
| 27 | 100 | Mismatch |
| 28 | O/C | Mismatch |
| 29 | O/C | Mismatch |
| 30 | 100 | Mismatch |
| **FAR=(0/30)*100=0%** | | |

ต่อไปเป็นการหาค่า **FRR** โดยการสุ่มเอาบุคคลจำนวนหนึ่งมาทำการเก็บข้อมูลของตนไว้แล้วใช้เครื่องสแกนทำการตรวจสอบลายนิ้วมือของตนเอง ทำทั้งหมด30ครั้งได้ผลดังตารางต่อไปนี้

ตารางที่ 8.14 แสดงผลการหาค่า FRR จากการสแกนลายนิ้วมือของบุคคลหลายคน จำนวน 30 ครั้ง (ข้อมูลในDATABASEเป็นข้อมูลที่ดีที่สุดของคนๆนั้น)

| ครั้งที่ | เปอร์เซ็นต์ความคลาดเคลื่อน (%) | Match / Mismatch |
|---|---|---|
| 1 | 0.8223 | Match |
| 2 | 0.0159 | Match |
| 3 | 2.5317 | Match |
| 4 | 100 | Mismatch |
| 5 | O/C | Mismatch |
| 6 | 4.6671 | Match |
| 7 | O/C | Mismatch |
| 8 | 100 | Mismatch |
| 9 | 2.2236 | Match |
| 10 | 100 | Mismatch |
| 11 | O/C | Mismatch |
| 12 | 1.1546 | Match |

| | | |
|---|---|---|
| 13 | 1.2930 | Match |
| 14 | 3.0847 | Match |
| 15 | O/C | Mismatch |
| 16 | 2.5266 | Match |
| 17 | 0.1865 | Match |
| 18 | 0.0054 | Match |
| 19 | 100 | Mismatch |
| 20 | 1.4362 | Match |
| 21 | 0.4994 | Match |
| 22 | 3.7854 | Match |
| 23 | 0.3135 | Match |
| 24 | 1.4484 | Match |
| 25 | 100 | Mismatch |
| 26 | O/C | Mismatch |
| 27 | O/C | Mismatch |
| 28 | 2.1126 | Match |
| 29 | 5.9524 | Match |
| 30 | 3.3313 | Match |
| **FRR=(11/30)*100= 36.67%**<br>**O/C = 6 ครั้ง : '100' = 5 ครั้ง** | | |

# บทที่ 9
# บทวิจารณ์และสรุป

## 5.1 สรุปผลการทดลอง

จากผลการทดลองที่8.1. การทดลองการนำภาพลายนิ้วมือมาวิเคราะห์สามารถสรุปได้ว่า เราได้ทำการปรับปรุงภาพที่รับเข้ามา (Image Enhancement) โดยทำการ Normalize ภาพ เพื่อนำภาพที่ได้ไปหาภาพทิศทาง (Orientation Image)ซึ่งเป็นกระบวนการประมาณทิศทางของเส้นนูนของภาพลายนิ้วมือ (จากเดิมภาพทิศทางมีแค่ 8 ทิศทางแต่ละทิศทางต่างกัน 22.5 ซึ่งเป็นทิศที่หยาบมาก ทำให้ไม่สามารถหา core ที่ถูกต้องได้) มีทิศทางตั้งแต่ $0^\circ - 180^\circ$ จากนั้นนำภาพทิศทางที่ได้ไปหาจุดกึ่งกลางภาพ (Core) ซึ่งได้ตำแหน่งใกล้เคียงกับจุดกึ่งกลางของลายนิ้วมือจริง ขั้นตอนต่อมาคือกระบวนการกรองลดสัญญาณรบกวนด้วยวิธีของ Gabor เพื่อให้ภาพมีลายเส้นนูนและร่องชัดเจนและถูกต้องยิ่งขึ้น เพื่อที่จะนำไปทำ Thinning เพื่อหาจุดสำคัญของภาพลายนิ้วมือต่อไป

ซึ่งจากการทดลองจะเห็นว่า ถ้าภาพลายนิ้วมือที่รับเข้ามาหากไม่มีความชัดเจน มีลักษณะของเส้นลายนิ้วมือที่ไม่สมบูรณ์มาก หรือภาพที่มีจุดกึ่งกลางภาพไม่อยู่ในบริเวณกลางภาพมาก ๆ ภาพที่ได้จากการปรับปรุงภาพ(Image Enhancement) และจุดกึ่งกลางภาพที่ได้จากการประมวลผลจะไม่ถูกต้องไปด้วย ดังนั้นอัลกอริทึ่ม (algorithm) นี้จะใช้ได้ดีกับภาพลายนิ้วมือที่ค่อนข้างชัดเจนและสมบูรณ์

จากผลการทดลองที่8.2.1 การทดลองการเปรียบเทียบลายนิ้วมือสามารถสรุปได้ว่า อัลกอริทึ่มที่ได้ออกแบบไว้สำหรับการเปรียบเทียบสามารถตรวจสอบและแยกแยะลายนิ้วมือของบุคคลได้ดี นั่นคือสามารถตรวจสอบลายนิ้วมือของบุคคลคนเดียวกันได้ สามารถแยกลายนิ้วมือรูปแบบเดียวกันของบุคคล 2 บุคคลได้ และสามารถแยกลายนิ้วมือที่มีรูปแบบต่างกันของบุคคล 2 บุคคลได้ โดยพิจารณาจากกราฟความสัมพันธ์ระหว่างระยะห่างระหว่างจุดกึ่งกลางกับจุดสำคัญ(r)กับมุมที่จุดสำคัญทำกับแกน x($\theta$) ซึ่งบุคคลแต่ละคนจะมีลักษณะแตกต่างกันออกไป

จากผลการทดลองที่8.2.2 การทดลองที่ใช้วิธีการสร้างสามเหลี่ยมจากจุดลักษณะเฉพาะ (Delaunay) เป็นอีกทางเลือกหนึ่งในการนำมาทำการเปรียบเทียบลายนิ้วมือถึงแม้ว่าจะมีการเตรียม Template ที่ยุ่งยากกว่าแต่เป็นข้อมูลที่อธิบายและเข้าใจได้ง่ายเนื่องจากเป็นอัตราส่วนเชิงคณิตศาสตร์แบบพื้นฐานมาเปรียบเทียบกันเป็นแบบลำดับแล้วประมาณช่วงที่สามารถยอมรับได้ซึ่งเป็นส่วนที่กำหนดไว้แล้ว ความยืดหยุ่นของ%Error นี่เป็นข้อดีของวิธีนี้อย่างหนึ่งเพราะกำหนดได้ง่ายและถ้าลด% Errorที่กำหนดนี้ให้มีความน้อยลงก็จะเป็นการเพิ่มความปลอดภัยภายในตัว

จากผลการทดลองที่ 8.2.3 เปรียบเทียบวิธีการการเปรียบเทียบทั้งสองแบบพบว่า แบบที่ใช้กราฟความสัมพันธ์ของรัศมีกับมุมสำคัญสามารถตัดสินลายนิ้วมือที่ไม่ซับซ้อนได้ดีกว่าแบบสามเหลี่ยม (Delaunay) แต่เมื่อเป็นInputที่มีความซับซ้อนมากขึ้นหรือมีความคมชัดน้อยลงวิธีที่ใช้กราฟความสัมพันธ์นี้อาจตัดสินผิดพลาดได้ ในขณะที่วิธีของDelaunayนั้นถึงแม้ว่า%Error ที่ออกมาจากลายนิ้วมือที่ไม่ซับซ้อนจะมีค่ามากกว่าวิธีกราฟความสัมพันธ์แต่ผลคำตัดสินยังคงเป็นmatch เช่นเดียวกับวิธีแรก เมื่อได้รับInput ที่มีความซับซ้อนของลายนิ้วมือมากขึ้นวิธีสามเหลี่ยมนี้ยังคงตัดสินได้ถูกต้องมากกว่าวิธีที่ใช้กราฟ แสดงให้เห็นถึงข้อดีในความแม่นยำสูงของวิธีสามเหลี่ยม(Delaunay) จึงเหมาะสำหรับที่ๆต้องใช้ระบบความปลอดภัยระดับสูงได้แก่ ธนาคาร หรือ กระทรวงที่เกี่ยวข้องกับความมั่นคงของชาติ ส่วนแบบกราฟความสัมพันธ์เหมาะสำหรับใช้กับหอพักหรือแบบทั่วไปตามสำนักงานและครัวเรือน

ผลการทดลองที่ 8.3 ค่าFAR(อัตราการหลุดรอดของผู้แปลกปลอมจากการตรวจจับ) มีค่าเป็นศูนย์ หมายถึงโอกาสที่จะสามารถปลอมแปลงลายนิ้วมือเพื่อให้รอดจากการตรวจสอบนั้นน้อยมากจนเป็นศูนย์ ด้วยเหตุนี้ทำให้เป็นข้อสนับสนุนของความมีความปลอดภัยที่สูงในระบบนี้ ส่วนค่าของFRR(การปฏิเสธการผ่านกับผู้ใช้ที่ถูกต้อง)ที่สูงถึง 36.67% ก็ไม่ใช่เรื่องผิดปกติแต่อย่างไร เพราะยิ่งมีความปลอดภัยสูงๆ ค่าของ DATA IN ที่ได้จากSENSOR SCANจะต้องยิ่งมีความชัดเจนและแน่นอนในทุกๆครั้งที่ทำการSCAN จากค่าในตาราง O/C และ ค่า 100% เกิดขึ้น 6 และ 5 ครั้งตามลำดับ ค่าO/Cนี่จะเกิดขึ้นเมื่ออัตราส่วนที่เป็นColumn(หรือ array)ระหว่าง DATA-IN กับ DATABASE มีค่าต่างกันมากๆ เช่น 8 ต่อ 3 หรือ 5 ต่อ 12 เป็นต้น ทั้งนี้เป็นเพราะเครื่องSENSORที่เก่ามาก เมื่อเกิดกรณีนี้ขึ้นโปรแกรม MATlab จะไม่ประมวลผลต่อเพราะถือว่าข้อมูลที่มีอยู่นั้นไม่เพียงพอที่จะนำมาเปรียบเทียบกัน ส่วนกรณี ERROR = 100% เกิดขึ้นเพราะมีจุดลักษณะสำคัญแปลกปลอมเกิดขึ้นอาจเป็นเพราะจุดขาวๆเกิดขึ้นระหว่างเส้นดำทำให้มีจุดลักษณะสำคัญมีเพิ่มขึ้นการสร้างสามเหลี่ยมและอัตราส่วนพื้นที่จึงเปลี่ยนแปลงตามกันไป ERRORที่ออกมาได้จึงเกินที่เราตั้งไว้คือ6% เราจะไม่เก็บค่าERRORที่เกินกำหนดนี้เราจึงSET ERRORให้ขึ้นเป็น100%

**9.2. แนวทางการพัฒนา**

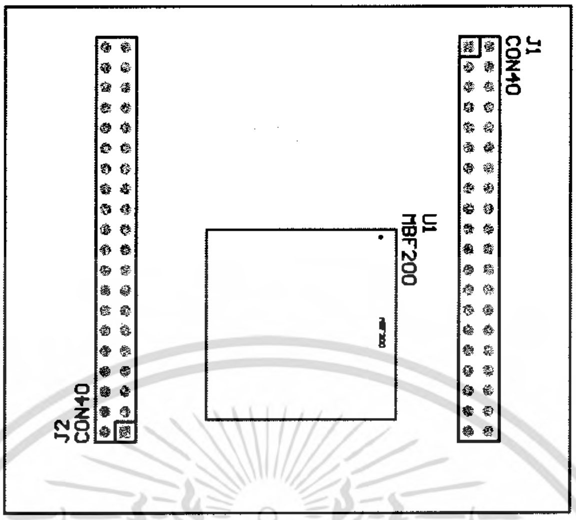
ในการลดค่าFRR สามารถกระทำได้โดยการเปลี่ยนSENSOR SCAN (MBF 200) ใหม่ เพราะปัจจุบันหน้าปัดสแกนมีลายขีดขวนมากมายจนภาพที่รับเข้ามามีจุดขาวมากมาย ส่วนจุดดำที่เกิดขึ้นสามารถแก้ได้โดยนำสารดูดความชื้นวางไว้ที่หน้าปัดสแกน และก่อนทำการสแกนผู้สแกนจะต้องมีการผลัดแป้งบางๆบริเวณนิ้วมือเพราะจะช่วยไม่ให้เกิดความมันและความชื้น

ในการพัฒนาระบบDATABASE อาจจะใช้โปรแกรมMY SQLมาช่วยเพื่อให้ใช้เวลาในการ MATCHINGนั้นน้อยลงเพราะในโปรแกรมนี้มีระบบ INDEX ที่สามารถระบุได้อย่างรวดเร็วว่า DATA-INที่เข้ามานั้นควรจะไปเปรียบเทียบกับตัวใดในDATABASE จะได้มิต้องเสียเวลาในการทำการ MATCHINGกับข้อมูลทุกๆตัวที่เก็บไว้หากเก็บไว้มากในระดับ200-300ข้อมูล วิธีของกลุ่มข้าพเจ้าอาจเสียเวลาเป็นนาที

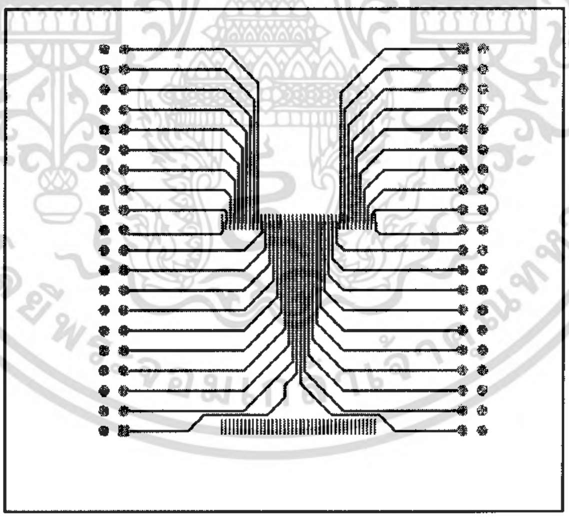
# บรรณานุกรม

1. เกรียงไกร โชวเจริญสุข, "Automation Fingerprint Verification System Using Directional Histogram and Neural Network for Equipment Controlling", วิทยานิพนธ์ ปีการศึกษา 2539
2. ขวัญชัย เรืองบุญคุ้มสุข, สุเมธ สำเร็จศิลป์, แสนศักดิ์ บวรจิโรภาส, "ระบบตรวจสอบลายนิ้วมือระบุบุคคล", ปริญญานิพนธ์ ปีการศึกษา 2546
3. ธานินทร์ พินทอง, ศรนรินทร์ นพรัตน์, "Fingerprint Recognition", ปริญญานิพนธ์ ปีการศึกษา 2539
4. ปกรณ์ มีรอด, สุมันตี พึ่งพิพัฒน์, อุดมศักดิ์ มีพัฒน์, "เครื่องสแกนลายนิ้วมือ",ปริญญานิพนธ์ ปีการศึกษา 2547
5. ปริญญานิพนธ์ เรื่อง *เครื่องสแกนลายนิ้วมือโดยใช้มุมและรัศมีที่จุดสำคัญ* ปีการศึกษา 2549
6. พิษณุนาถ จิวสิทธิประไพ, วสุ เฟื่องสวัสดิ์, "การเปรียบเทียบลายนิ้วมือและตรวจสอบลายนิ้วมือ", ปริญญานิพนธ์ ปีการศึกษา 2545
7. อารัมภ์ จันทร์ใย, โสรัตย์ อุนหะวราภร, "เครื่องตรวจสอบลายนิ้วมือสำหรับระบบรักษาความปลอดภัย", วารสารเซมิคอนดัคเตอร์อิเล็กทรอนิกส์ ฉบับที่ 244-245 มกราคม 2546
8. DAVIDE MALTONI, DARIO MAIO, ANIL K.KAIN, SALIL PRABHAKAR, "Handbook of Fingerprint Recognition", Springer
9. JIE ZHOU and JINWEI GU, "A Model-Based Method for the Computation of Fingerprints' Orientation Field", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 6, JUNE 2004
10. F.A. Afsar, M. Arif and M. Hussain, "Fingerprint Identification and Verification System using Minutiae Matching, National Conference on Emerging Technologies 2004
11. Xudong Jiang1, Manhua Liu2 and Alex Chichung Kot, "Reference Point Detection for Fingerprint Recognition", Proceedings of the 17th International Conference on Pattern Recognition (1CPR'04)

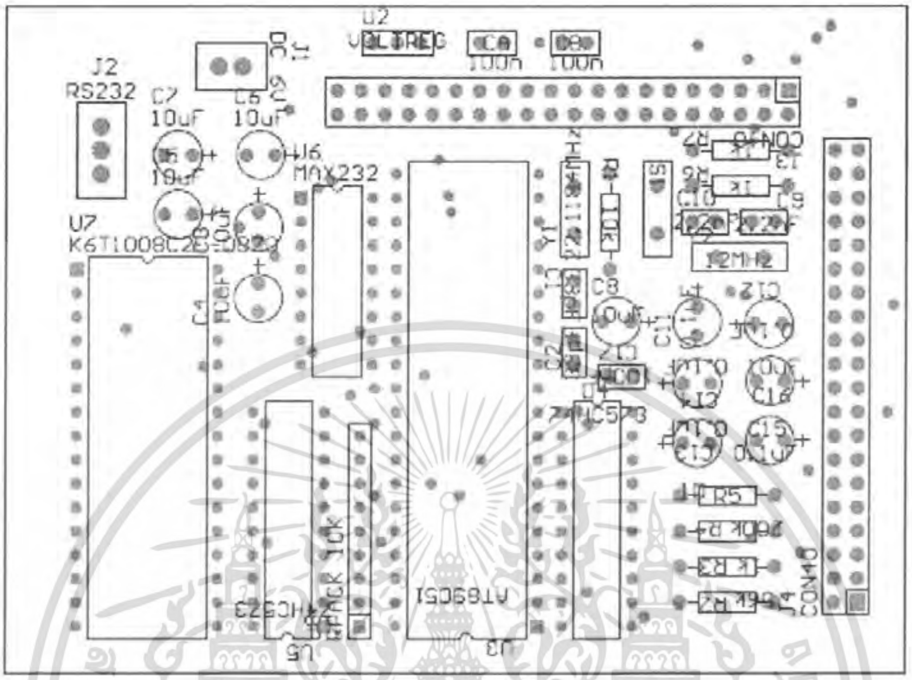
# ภาคผนวก



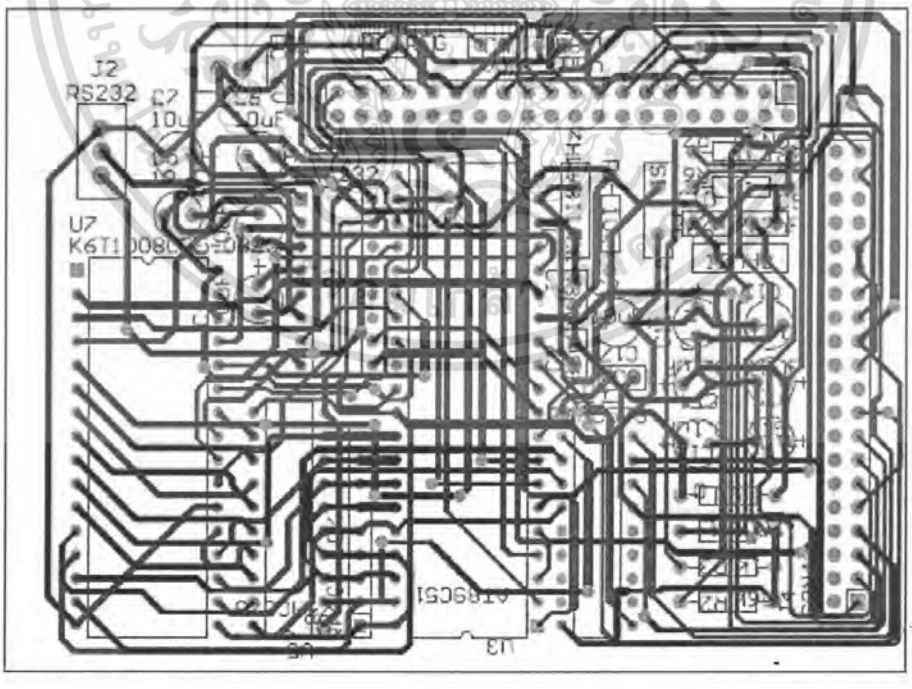

รูปที่ i เซนเซอร์สแกนลายนิ้วมือ



รูปที่ ii แสดงลายวงจรตัวเซนเซอร์




รูปที่ iii แสดงตำแหน่งการวางอุปกรณ์บอร์ดสแกน




รูปที่ iv แสดงลายวงจรบอร์ดสแกน