



การตรวจลายนิ้วมือ

Fingerprint Analysis

โดย

นายนวัทร สุจิระชาติ 37014195

นายภิญโญ วงษ์สถาปนาลีศ 37014321

นางสาวศิริลักษณ์ ปล่องทอง 37014439

อาจารย์ที่ปรึกษา

ดร. สุรพันธุ์ เอื้อไพบูลย์

วัน เดือน ปี.....	23.คค.2541
เลขทะเบียน.....	039139
เลขเรียกหนังสือ.....	T 110378 46291 ก.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ **039139** คำ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานเรื่อง การตรวจสอบลายนิ้วมือ  
Fingerprint Analysis

จัดทำโดย นาย นวภัทร์ สุจิระชาโต 37014195  
นาย ภิญ โสญ วงษ์สถาปนาเลิศ 37014321  
น.ส. ศิริลักษณ์ ปล่องทอง 37014439

อาจารย์ที่ปรึกษา คร. สุรพันธ์ เอื้อไพบูลย์



รายงานฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ

อาจารย์ที่ปรึกษา

วันที่ 17 / เม.ย. 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การตรวจสอบลายนิ้วมือ

นาย นวภัทร์ สุจิราธาโต

นางสาว ศิริลักษณ์ ปล่องทอง

นาย ภิญโญ วงษ์สถาปนาเลิศ

ดร.สุรพันธ์ เอื้อไพบูลย์ อาจารย์ที่ปรึกษา

### บทคัดย่อ

ในปัจจุบัน การตรวจสอบลายนิ้วมืบทบาทมากขึ้น ในการแยกแยะลักษณะเฉพาะของแต่ละบุคคล ซึ่งสามารถนำไปใช้ในงานต่างๆ ได้มีประสิทธิภาพ เช่น การเก็บลายนิ้วมือ ก่อนการทำบัตรประชาชน เป็นลักษณะเฉพาะของแต่ละบุคคล เพราะเมื่อเวลาผ่านไป หน้าตาอาจเปลี่ยนไป อายุมากขึ้น แต่ลายนิ้วมือจะคงทนไม่เปลี่ยนแปลง ขณะเดียวกัน เทคโนโลยีด้านคอมพิวเตอร์ได้พัฒนาขึ้น การใช้คอมพิวเตอร์ในการเก็บข้อมูลและแยกแยะข้อมูลลายนิ้วมือสามารถทำได้ง่ายและรวดเร็ว โดยเฉพาะกับข้อมูลจำนวนมากๆ การใช้คอมพิวเตอร์ทำให้เหมาะสมกว่า ดังนั้นการตรวจสอบลายนิ้วมือโดยคอมพิวเตอร์ จะต้องมีความรวดเร็วและแม่นยำถูกต้อง

## Fingerprint Analysis

Mr Navapat Sujirachato

Miss Sirilak Plongthong

Mr Pinyo Wongsathanalert

Dr. Surapan Airphaiboon

1<sup>st</sup> Semester, Education Year 1997

### Abstract

Now, Fingerprint Analysis has a important role in classifying each person features. That makes process operates efficiently ,for example: collect fingerprint before making ID card, because when the person is older ,so will the face, but fingerprint is more durable. While the technology of computer is advancely developed, process finger print analysis by computer is easier and faster especially for an large amount of data ,the computer is more suitable. So finger print analysis by computer is designed to run accurately and quickly.

# สารบัญ

บทที่ 1 ความรู้เกี่ยวกับการประมวลผลภาพและการปรับปรุงภาพ	
1. ความรู้เกี่ยวกับการประมวลผลภาพ	1
1.1 การประมวลผลภาพ	1
1.2 การจดจำรูปแบบ	1
2. การประยุกต์ใช้ดิจิทัลอิมเมจโปรเซสซิ่งกับการจัดการลายนิ้วมือ	3
2.1 การปรับปรุงภาพ โดยใช้ filter	3
2.2 การทำโครงร่าง	6
บทที่ 2 การกำจัดเส้นกึ่ง โดยวิธีติดตามลายเส้น	
1. ทฤษฎี	8
2. อัลกอริทึมในการติดตามลายเส้น	14
บทที่ 3 การแบ่งประเภทลายนิ้วมือ	18
1. การประมาณรูปแบบของลายนิ้วมือ	19
2. จุดเด่นที่คัดออกมา	21
3. การวิเคราะห์รูปแบบ	22
บทที่ 4 โปรแกรมการทำงาน	25
1. การสร้างไฟล์ฐานข้อมูล	27
2. การเปรียบเทียบไฟล์กับฐานข้อมูล	28
3. การแบ่งชนิดของไฟล์ว่าเป็นลายนิ้วมือชนิดใด	30
บทที่ 5 ผลการทดลอง สรุปและวิจารณ์	
1. หน้าจอการทำงาน	32
2. ส่วนทำการเปรียบเทียบไฟล์ข้อมูล	33
3. ส่วนทำการแบ่งประเภทลายนิ้วมือ	33
4. ส่วนการแสดงผลภาพ	34
การทดลอง	35
ผลการทดลอง	45
สรุปและวิจารณ์	46

## ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ

## บทที่ 1 ความรู้พื้นฐานเกี่ยวกับการประมวลผลภาพและการปรับปรุงภาพ

รูปที่ 1.1 การใช้แว่นสายตาในการปรับปรุงการรับภาพจากสายตา	1
รูปที่ 1.2 ภาพลายนิ้วมือ	2
รูปที่ 1.3 ภาพแสดงการเปลี่ยนแปลงของค่าสีแบบ Low frequency	3
รูปที่ 1.4 ภาพแสดงการเปลี่ยนแปลงของค่าสีแบบ High frequency	3
รูปที่ 1.5ก. ข้อมูลภาพแบบ Low frequency ก่อนการ Convolution	4
1.5ข. ข้อมูลภาพหลังทำ Convolution	5
รูปที่ 1.6ก. ข้อมูลภาพแบบ Low frequency ก่อนการ Convolution	5
1.6ข. ข้อมูลภาพหลังทำ Convolution	5
รูปที่ 1.7 รูปจุดใกล้เคียงทั้ง 8 ทิศทางของ $P_1$	7
รูปที่ 1.8 การตรวจสอบเงื่อนไขจากรูปพบว่า $N(P_1)=4$ และ $S(P_1)=3$	7
รูปที่ 1.9 ภาพแสดงผลการทำโครงร่าง	
a ผลของการผ่านขั้นตอนแรก	7
b ผลของขั้นตอนที่สอง	7
c ผลสุดท้าย	7

## บทที่ 2 การกำจัดเส้นกึ่งโดยวิธีการติดตามลายเส้น

รูปที่ 2.1 a ภาพตัวเลขอารบิก ที่มีขนาดตัวอักษร 72 จุดภาพ	9
b ภาพตัวเลขอารบิก ที่ได้จากการนำภาพ 2.1a. มาผ่านการทำเส้นให้บาง	9
c ภาพตัวเลขอารบิก ที่มีเส้นกึ่ง	9
รูปที่ 2.2 ลักษณะเส้นกึ่ง	
a ก่อนการแก้ไข	9
b หลังการแก้ไข	9
รูปที่ 2.3 a วินโดว์ขนาด $3 \times 3$	9
b แสดงทิศทางในการสแกนจุดภาพจากด้านซ้ายไปขวา และจากบนลงล่าง	9
รูปที่ 2.4 ตารางแสดงค่าโทโปโลยีของจุดรูป	10

## สารบัญรูปร่าง

รูปที่ 2.5 a.	เทมเพลต 3 x 3 ของจุดภาพที่มีค่า $Nc^4 = 0$ หรือ $Nc^8 = 0$	11
b.	เทมเพลต 3 x 3 ของจุดภาพที่มีค่า $Nc^4 = 1$	11
c.	เทมเพลต 3 x 3 ของจุดภาพที่มีค่า $Nc^8 = 1$	11
d.	เทมเพลต 3 x 3 ของจุดภาพที่มีค่า $Nc^4 = 2$	11
e.	เทมเพลต 3 x 3 ของจุดภาพที่มีค่า $Nc^8 = 2$	11
f.	เทมเพลต 3 x 3 ของจุดภาพที่มีค่า $Nc^4 = 3$	11
g.	เทมเพลต 3 x 3 ของจุดภาพที่มีค่า $Nc^8 = 3$	11
รูปที่ 2.6	ตัวอย่างจุดภาพที่ไม่สามารถจำแนกได้โดยค่า $Nc^8$ หรือ $Nc^4$	12
รูปที่ 2.7	เทมเพลต 3 x 3 ที่แสดงถึงจุดเชื่อมต่อกันทั้งหมดถึง 12 กรณี	13
รูปที่ 2.8	แสดงเทมเพลต 3 x 3 ที่แสดงถึงจุดแยกในแต่ละกรณีเมื่อหมุนทวนเข็มนาฬิกาทีละ $90^\circ$	14
รูปที่ 2.9	เทมเพลต 3 x 3 ที่แสดงถึงจุดปลายทั้งหมดถึง 8 กรณี	14
รูปที่ 2.10 a.	ภาพตัวเลขอารบิกทั้ง 10 ที่มีลายเส้นกึ่งอยู่คือ 3 และ 4	16
b.	ภาพตัวเลขอารบิก 3 และ 4 บริเวณของเส้นกึ่งที่ต้องถูกกำจัด	16
c.	ภาพตัวเลขอารบิก 3 และ 4 บริเวณของเส้นกึ่งที่ต้องถูกกำจัดออกไป	16
รูปที่ 2.11 a.	ภาพตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ ที่มีตัวอักษรอยู่ 4 ตัว คือ M,N,V และ W ที่มีลายเส้นกึ่ง	17
b.	ภาพตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ M,N,V และ W บริเวณของเส้นกึ่งที่ต้องถูกกำจัด	17
c.	ภาพตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่ ทั้ง 4 ตัว คือ M,N,V และ W บริเวณของเส้นกึ่งที่ต้องถูกกำจัดออกไป	17
<b>บทที่ 3 การแบ่งประเภทของลายนิ้วมือ</b>		
รูปที่ 3.1	แสดงจุด Delta	18
รูปที่ 3.2	ชนิดของลายนิ้วมือและกลุ่มของตัวอักษรสำหรับแต่ละชนิดของลายนิ้วมือ	19
รูปที่ 3.3	รหัสทิศทาง	20
รูปที่ 3.4	ตัวจัดการแบบต่างๆ	20
รูปที่ 3.5	ลายนิ้วมือที่ถูกทำการสุ่ม	20

## สารบัญรูปภาพ

รูปที่ 3.6	ขั้นตอน ในเมทริกซ์จากการสุ่ม	23
รูปที่ 3.7	ตัวกรองและผลที่ได้จากตัวกรอง	23
รูปที่ 3.8	คุณสมบัติที่ใช้ในการแบ่งประเภท	24
รูปที่ 3.9	รูปแบบที่ได้จากการสุ่มตัวอักษรที่มีความหมาย	24
รูปที่ 3.10	รูปแบบต่างๆของลายนิ้วมือแบบขมวด	24

### บทที่ 4 โปรแกรมการทำงาน

รูปที่ 4.1	โฟลว์ชาร์ตการทำงานของโครงการ	26
รูปที่ 4.2	โฟลว์ชาร์ตการเก็บข้อมูลที่ได้จากการประมวลผล	28
รูปที่ 4.3	โฟลว์ชาร์ตขั้นตอนการเปรียบเทียบ	29
รูปที่ 4.5	โฟลว์ชาร์ตการทำเส้นทาง	30
รูปที่ 4.6	โฟลว์ชาร์ตการแบ่งชนิดของลายนิ้วมือ	31

### บทที่ 5 ผลการทดลอง สรุป และวิจารณ์

รูปที่ 5.1	หน้าจอเมื่อสั่งรันโปรแกรม	32
รูปที่ 5.2	ภาพลายนิ้วมือปกติ(ภาพที่ 1)	35
รูปที่ 5.3	ภาพลายนิ้วมือที่ผ่านการทำบาง(ภาพที่ 1)	35
รูปที่ 5.4	ภาพลายนิ้วมือที่กำจัดเส้นกึ่งแล้ว(ภาพที่ 1)	36
รูปที่ 5.5	ภาพลายนิ้วมือทำการหาทิศทางแล้ว(ภาพที่ 1)	36
รูปที่ 5.6	ภาพลายนิ้วมือปกติ(ภาพที่ 2)	37
รูปที่ 5.7	ภาพลายนิ้วมือที่ผ่านการทำบาง(ภาพที่ 2)	37
รูปที่ 5.8	ภาพลายนิ้วมือที่กำจัดเส้นกึ่งแล้ว(ภาพที่ 2)	38
รูปที่ 5.9	ภาพลายนิ้วมือทำการหาทิศทางแล้ว(ภาพที่ 2)	38

## สารบัญรูปภาพ

รูปที่ 5.10 ภาพลายนิ้วมือปกติ(ภาพที่ 3)	39
รูปที่ 5.11 ภาพลายนิ้วมือที่ผ่านการทำบาง(ภาพที่ 3)	39
รูปที่ 5.12 ภาพลายนิ้วมือที่กำจัดเส้นกึ่งแล้ว(ภาพที่ 3)	40
รูปที่ 5.13 ภาพลายนิ้วมือทำการหาทิศทางแล้ว(ภาพที่ 3)	40
รูปที่ 5.14 ภาพลายนิ้วมือปกติ(ภาพที่ 4)	41
รูปที่ 5.15 ภาพลายนิ้วมือที่ผ่านการทำบาง(ภาพที่ 4)	41
รูปที่ 5.16 ภาพลายนิ้วมือที่กำจัดเส้นกึ่งแล้ว(ภาพที่ 4)	42
รูปที่ 5.17 ภาพลายนิ้วมือทำการหาทิศทางแล้ว(ภาพที่ 4)	42
รูปที่ 5.18 ภาพลายนิ้วมือปกติ(ภาพที่ 5)	43
รูปที่ 5.19 ภาพลายนิ้วมือที่ผ่านการทำบาง(ภาพที่ 5)	43
รูปที่ 5.20 ภาพลายนิ้วมือที่กำจัดเส้นกึ่งแล้ว(ภาพที่ 5)	44
รูปที่ 5.21 ภาพลายนิ้วมือทำการหาทิศทางแล้ว(ภาพที่ 5)	44

## บทที่ 1

# ความรู้พื้นฐานเกี่ยวกับการประมวลผลภาพและการปรับปรุงภาพ

## (Introduction to image processing)

### 1 ความรู้พื้นฐานเกี่ยวกับการประมวลผลภาพ

#### 1.1 การประมวลผลภาพ

นิยามของ การประมวลผลภาพ (Image processing) คือการจัดการและวิเคราะห์ข้อมูลของภาพ ซึ่งหมายถึงข้อมูล 2 มิติ ของภาพจริง เราจะพบตัวอย่างของการประมวลผลภาพได้รอบตัว ตัวอย่างเช่นการใช้แว่นตา เพื่อแก้ไขภาพที่บิดเบือนให้ชัดเจนขึ้น โดยภาพที่ชัดเจนแล้วจะเข้าสู่สายตาเป็นภาพที่ถูกต้อง



รูปที่ 1.1 การใช้แว่นสายตาในการปรับปรุงการรับภาพจากตา

การประมวลผลภาพมี 3 แบบ

- 1 การประมวลผลเกี่ยวกับสายตา(Optical image processing)
- 2 การประมวลผลภาพทางอนาลอก(Analog image processing)
- 3 การประมวลผลภาพทางดิจิทัล(Digital image processing)

#### 1.2 การจดจำรูปแบบ(Pattern recognition)

คือขั้นตอนการแยกความแตกต่าง และการจัดกลุ่ม การจดจกรูปแบบ มีจำนวนกลุ่มที่มีความแตกต่าง(Class) ขึ้นอยู่กับรูปแบบการใช้งานต่างๆ เช่นการจดจำรูปแบบตัวอักษรภาษาอังกฤษ จะมีทั้งหมด 26 คลาส นั่นคือ ภาษาอังกฤษมีตัวอักษรที่แตกต่างกัน 26 ตัว ขณะเดียวกัน ถ้าต้องการแยกความแตกต่างระหว่าง ภาษาอังกฤษกับภาษารัสเซีย ก็จะถือว่ามึคลาสอยู่ 2 คลาส ในการใช้งานบางครั้งเราไม่สามารถรู้ จำนวนคลาสที่แน่นอนในขณะที่เริ่มต้นได้ แต่สามารถหาได้จากการสังเกตของรูปแบบที่เป็นตัวอย่าง โดยเราจะตรวจหาโอกาสที่จะพบรูปแบบให้มากที่สุดเท่าที่จะสามารถทำได้ ในปัจจุบัน นักวิทยาศาสตร์และวิศวกรนิยมใช้เครื่องมือชนิดนี้ในการทำงาน

หลักการเบื้องต้นของการจดจำ คือ การเปรียบเทียบระหว่างข้อมูลที่ต้องการเปรียบเทียบ กับแบบจำลอง ในที่นี้ชุดของแบบจำลองจะถูกเก็บไว้ในเครื่อง เมื่อมีข้อมูลเข้ามา ข้อมูลนี้จะถูกนำไปเปรียบเทียบกับแบบจำลองของข้อมูลในแต่ละประเภท ถ้าหากเปรียบเทียบออกมาได้ผลที่ตรงกัน ข้อมูลนี้จะถูกจัดให้อยู่ในกลุ่มเดียวกับแบบจำลองนั้น โดยทั่วไปแบบจำลองจะอยู่ในรูปข้อมูลทั่วไป ข้อเสียของการจดจำโดยการเปรียบเทียบกับแบบจำลอง คือ บางครั้งยากต่อการเลือกแบบจำลองในแต่ละประเภท

เราจะใช้วิธีทางคณิตศาสตร์ หลายอย่างในการใช้หารูปแบบ ซึ่งแบ่งได้ 2 วิธีคือ

1 Decision theoretic

2 Syntactic

Decision theoretic

ใช้แนวคิดทางสถิติ โดยทั่วไปใช้ลักษณะทางฟิสิกส์ มาแบ่งแยกเป็น ความกว้าง×ความยาว จึงแบ่งได้เป็น 2 ขั้นตอน ขั้นตอนแรก คือการวัด ความกว้าง×ความยาว ที่ได้คือลักษณะสำคัญของวัตถุ โดยการวัดก็จำเป็นต้องเลือกส่วนที่วัด อย่างไรก็ตาม ก็ไม่ค่อยมีทฤษฎีสับสนุนวิธีนี้เท่าใดนัก ขั้นตอนที่สอง คือ การแยกเข้ากลุ่ม โดยจะหาลักษณะที่สำคัญของแต่ละแบบมาเทียบกับข้อมูลที่ได้มา

Syntactic หรือ Structural approach

คือการวิเคราะห์ทางโครงสร้างเนื่องจากบางครั้งอินพุตที่นำมาจัดกลุ่ม มีลักษณะสำคัญมาก และข้อมูลมีความซับซ้อนมาก นอกจากนี้ยังมีความยากในการแบ่งแยกประเภท บางครั้งไม่สามารถใช้วิธี Decision theoretic ได้ เช่นรูปแบบลายนิ้วมือ รูปหน้าคน

## 2 การประยุกต์ใช้ DIGITAL IMAGE PROCESSING กับการจัดการลายนิ้วมือ

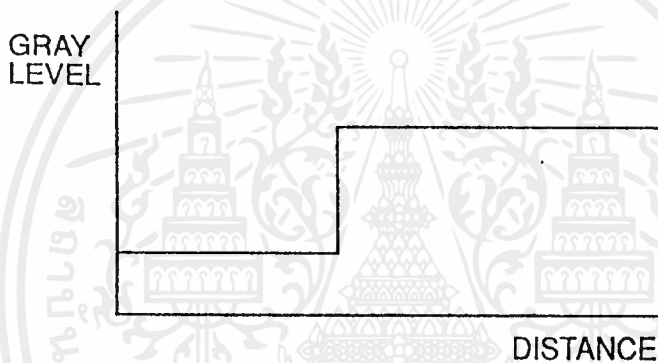
จุดประสงค์ของการนำ Digital image processing มาใช้เกี่ยวกับลายนิ้วมือ คือ ใช้ปรับปรุงภาพ คือ เพิ่มแสงเงาของภาพ พยายามทำให้ภาพมีความคมชัดมากยิ่งขึ้น เนื่องจาก ภาพที่ได้รับจากกล้องจะมีบางส่วนที่ได้รับแสงมากเกินไป หรือน้อยเกินไป ทำให้ไม่สามารถที่จะประมวลผลภาพได้

### 2.1 การปรับปรุงภาพ(Image Enhancement) โดยใช้ Lowpass filtering

ภาพแบบดิจิตอล ที่บริเวณขอบของวัตถุ จะมีคุณสมบัติของ High Frequency ดังนั้นการลดความมัว ของภาพทำได้โดยการส่วนประกอบของ High Frequency

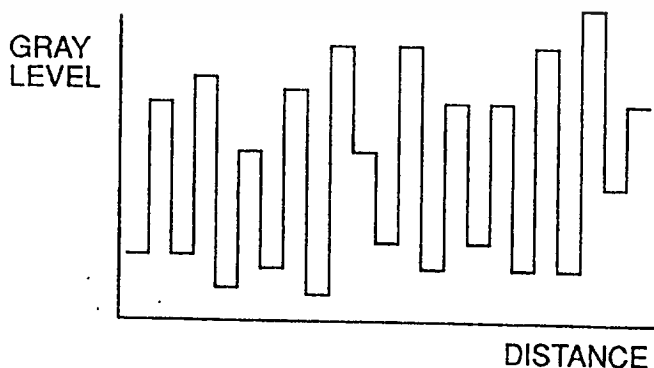
#### Lowpass filter โดยการ convolution

ในภาพแบบดิจิตอลซึ่งข้อมูลของภาพคือค่าสีในแต่ละตำแหน่งของภาพ ในบริเวณที่เป็นขอบของภาพก็คือ ในบริเวณที่ค่าสีของภาพเปลี่ยนจากค่าหนึ่งไปเป็นอีกค่าหนึ่ง ดังเช่นแสดงในรูปที่ 1.3



รูปที่ 1.3 ภาพแสดงการเปลี่ยนแปลงของค่าสีแบบ Low frequency

จากภาพเห็นว่าการเปลี่ยนแปลงเพียงครั้งเดียว(Low frequency) ซึ่งในกรณีนี้กล่าวได้ว่า ภาพมีความคมชัดโดยทั่วไปแล้วสัญญาณรบกวน(noise) ที่พบในภาพแบบดิจิตอลนั้น พบว่าบริเวณที่มีสัญญาณรบกวนข้อมูลของค่าสีของภาพ จะมีอัตราการเปลี่ยนแปลงสูง(high frequency) ดังแสดงในรูปที่ 1.4



รูปที่ 1.4 แสดงการเปลี่ยนแปลงของค่าสีแบบ High frequency

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำ Lowpass filter จะเป็นการกำจัดส่วนประกอบของภาพซึ่งมีอัตราการเปลี่ยนแปลงของค่าสีสูง ซึ่งจะเป็นการลดสัญญาณรบกวน แต่พบว่าจะทำให้ภาพที่ได้มีความพร่ามัวเพิ่มขึ้น

สำหรับการทำ Lowpass filter โดยใช้การคูณประสาน(convolution) จะมี Mask ที่ใช้ในการ Convolution หลายแบบ ตัวอย่างเช่น

$$\text{Mask1} \quad 1/16 \times \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\text{Mask2} \quad 1/9 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Mask3} \quad 1/10 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Mask4} \quad 1/16 \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

จาก Mask แบบต่างๆ ที่ได้แสดงจะพบว่า การคูณประสาน Mask กับข้อมูลซึ่งมีค่าคงที่ จะได้ผลลัพธ์คือข้อมูลเดิมไม่มีการเปลี่ยนแปลง สำหรับการคูณประสาน Mask กับข้อมูลภาพแบบ Low frequency แสดงในรูปที่ 1.5 และการคูณประสาน Mask กับข้อมูลภาพแบบ High frequency แสดงในรูปที่ 1.6

$$\begin{bmatrix} 150 & 150 & 150 & 150 & 150 \\ 150 & 150 & 150 & 150 & 150 \\ 150 & 150 & 150 & 150 & 150 \\ 150 & 150 & 150 & 150 & 150 \\ 150 & 150 & 150 & 150 & 150 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

150	150	150	150	150
150	150	150	150	150
150	150	150	150	150
150	150	150	150	150
125	125	125	125	125
25	25	25	25	25
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

ข.

รูปที่ 1.5 ก. ข้อมูลภาพแบบ Low frequency ก่อนการ Convolution

ข. ข้อมูลภาพหลังทำ Convolution

จากรูปที่ 1.5 พบว่าหากทำคูณประสานกันระหว่าง Mask กับภาพแบบ Low frequency จะได้ภาพที่มีอัตราการเปลี่ยนแปลงที่สูงขึ้นกว่าเดิม ซึ่งเป็นสาเหตุให้ภาพที่ได้พร่ามัว

150	150	150	150	150
1	1	1	1	1
150	150	150	150	150
1	1	1	1	1
150	150	150	150	150
1	1	1	1	1
150	150	150	150	150
1	1	1	1	1
150	150	150	150	150
1	1	1	1	0

ก.

100	100	100	100	100
50	50	50	50	50
100	100	100	100	100
50	50	50	50	50
100	100	100	100	100
50	50	50	50	50
100	100	100	100	100
50	50	50	50	50
100	100	100	100	100
50	50	50	50	50

ข.

รูปที่ 1.6 ก. ข้อมูลภาพแบบ High frequency ก่อนการ Convolution

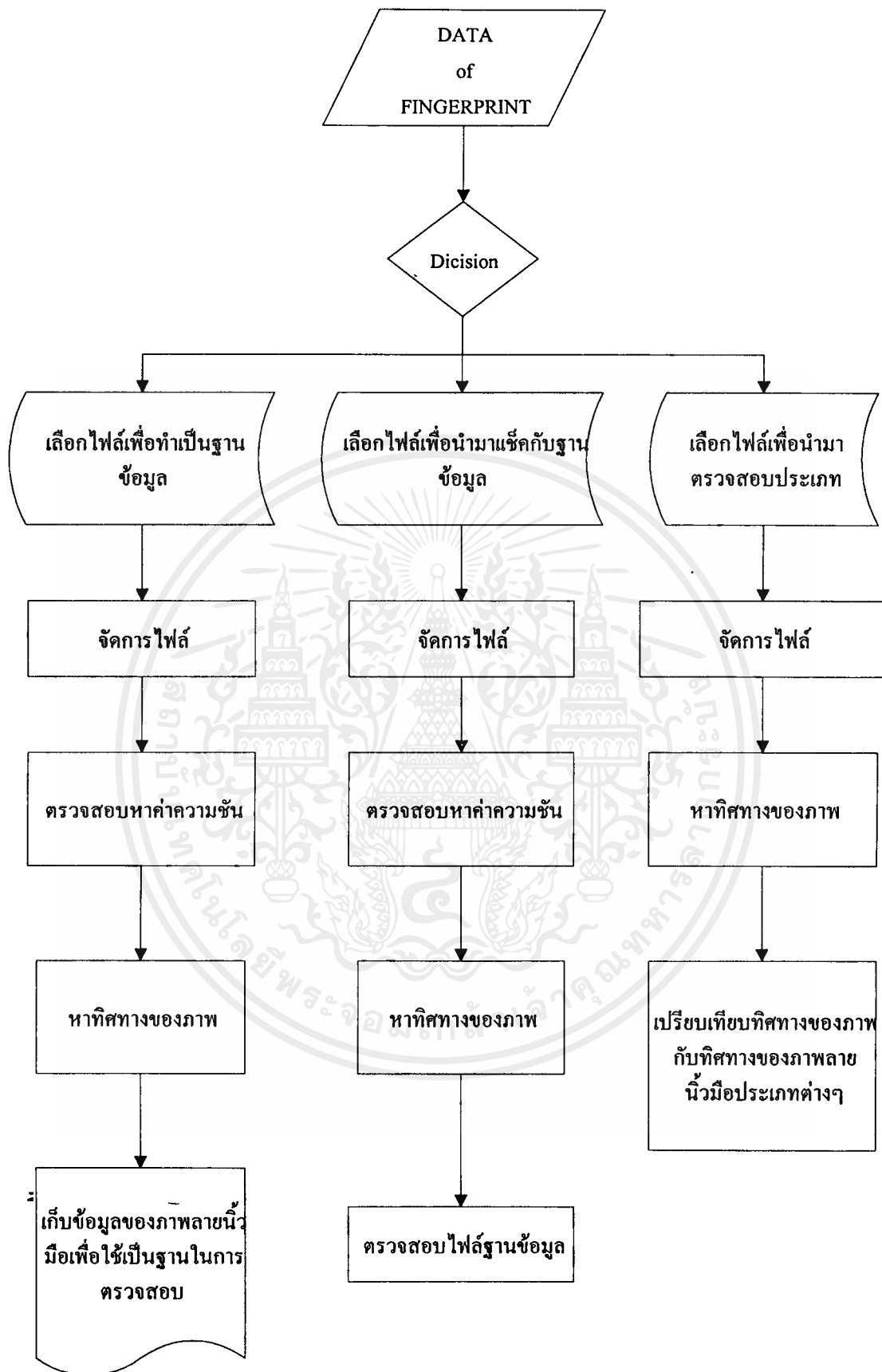
ข. ข้อมูลภาพหลังทำ Convolution

## บทที่ 4

### โปรแกรมการทำงาน

#### การทำงานของโปรแกรม

โปรแกรมของโครงการนี้จะเริ่มต้น เมื่อผู้ใช้ทำการเลือกไฟล์ที่ต้องการสร้างเป็นฐานของข้อมูลสำหรับตรวจสอบ เมื่อผู้ใช้ทำการเลือกไฟล์เรียบร้อยแล้ว โปรแกรมนี้จะนำเอาไฟล์นี้มาทำการปรับปรุงก่อนที่จะทำการตรวจสอบหาข้อมูล โดยจะทำการเปลี่ยนภาพจากภาพแบบ BMP ให้เป็นภาพแบบ Gray Scale ก่อน เพื่อที่จะนำภาพที่ได้นี้ไปเปลี่ยนเป็นภาพแบบ Binary Bit ( ภาพขาวดำ ) เหตุผลสำคัญที่จะต้องทำการเปลี่ยนเป็นภาพแบบ ขาว-ดำ เนื่องจากความง่ายในการตรวจสอบ และเปรียบเทียบข้อมูล แต่ภาพขาว-ดำที่ได้มานี้ จะยังไม่สามารถทำการตรวจสอบได้ทันที เนื่องจากไฟล์ข้อมูลนี้เป็นภาพของลายนิ้วมือที่มีความละเอียดอ่อนมากกว่าไฟล์ข้อมูลชนิดอื่น มากๆ ดังนั้นจึงจะต้องทำการปรับปรุงไฟล์ข้อมูลนี้ ให้อยู่ในสภาพที่สามารถทำการตรวจสอบได้ จึงต้องนำไฟล์ข้อมูลนี้มาทำการตกแต่งให้อยู่ในสภาพที่สามารถทำการตรวจสอบได้ เช่น การต่อเส้น, การลบเส้นกึ่ง, การลบจุดที่ไม่จำเป็นต่อเส้นของลายนิ้วมือ เป็นต้น หลังจากทำการประมวลผลต่างๆเรียบร้อยแล้วจึงจะสามารถนำไฟล์ข้อมูลของลายนิ้วมือนี้ไปทำการตรวจสอบได้ โดยข้อมูลที่ได้นี้จะนำไปใช้งานสำหรับฟังก์ชันการทำงานของโปรแกรม 3 ส่วนใหญ่ๆ คือ



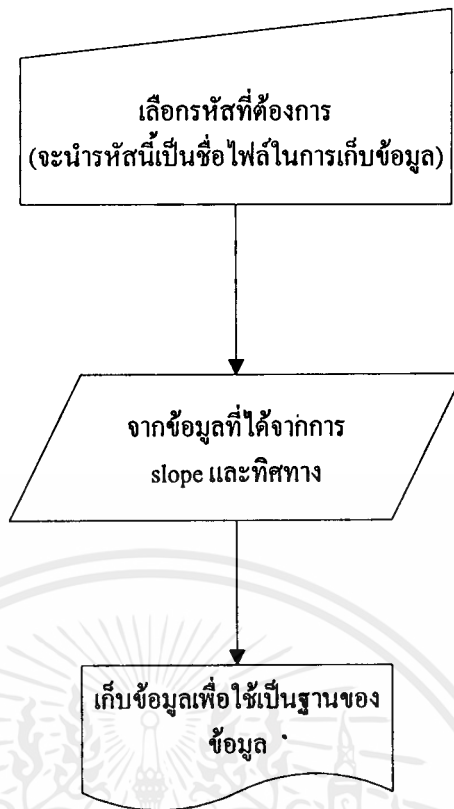
รูปที่ 4.1 Flow chart การทำงานของโครงการ

## 1. ส่วนของการสร้างไฟล์ฐานข้อมูล

ส่วนนี้เมื่อได้รับข้อมูลที่ได้หลังจากการประมวลผลต่างๆเรียบร้อยแล้ว จะนำเอาข้อมูลที่ได้มานี้มาทำการเก็บลักษณะเด่นของภาพของไฟล์ลายนิ้วมือนี้ ซึ่งลักษณะเด่นของภาพลายนิ้วมือนี้มีหลายแบบ หลายชนิด เช่น จุดสิ้นสุดของลายนิ้วมือ , ค่าความชันของเส้นลายนิ้วมือ , ตำแหน่งแยกของลายนิ้วมือ ซึ่งจะเป็นอะไรก็ได้ที่ทำให้เห็นหรือเป็นจุดสังเกตได้ง่ายว่าลายนิ้วมือของไฟล์ภาพที่ต่างกันมีลักษณะที่ต่างกัน ขึ้นอยู่กับผู้ที่ทำการศึกษา ทดลอง

สำหรับโครงการนี้จะเลือกใช้ลักษณะเด่นของเส้นลายนิ้วมือคือ ค่าความชัน และ ทิศทางของเส้นลายนิ้วมือ เหตุผลสำคัญที่เลือก 2 ฟังก์ชันนี้ก็คือ ค่าความชันของเส้นลายนิ้วมือนั้นจะมีค่าความแตกต่างที่เห็นได้อย่างชัดเจน และสำหรับเหตุผลที่เลือกใช้ทิศทางของเส้นลายนิ้วมือก็เนื่องมาจาก สามารถนำค่าที่ได้นี้ไปแบ่งเส้นลายนิ้วมือเป็นประเภทต่างๆ ได้อีกด้วย แต่สำหรับลักษณะเด่นอื่นๆที่ไม่เลือกนำมาใช้ก็เนื่องมาจากหลายๆสาเหตุ เช่น ตำแหน่งของจุดแยก หรือจุดสิ้นสุดของเส้นลายนิ้วมือนั้นอาจไม่ใช่ตำแหน่งที่แท้จริงของเส้นลายนิ้วมือก็ได้ อาจจะเป็นตำแหน่งของของ error ที่เกิดเนื่องมาจากสาเหตุต่างๆ เป็นต้น

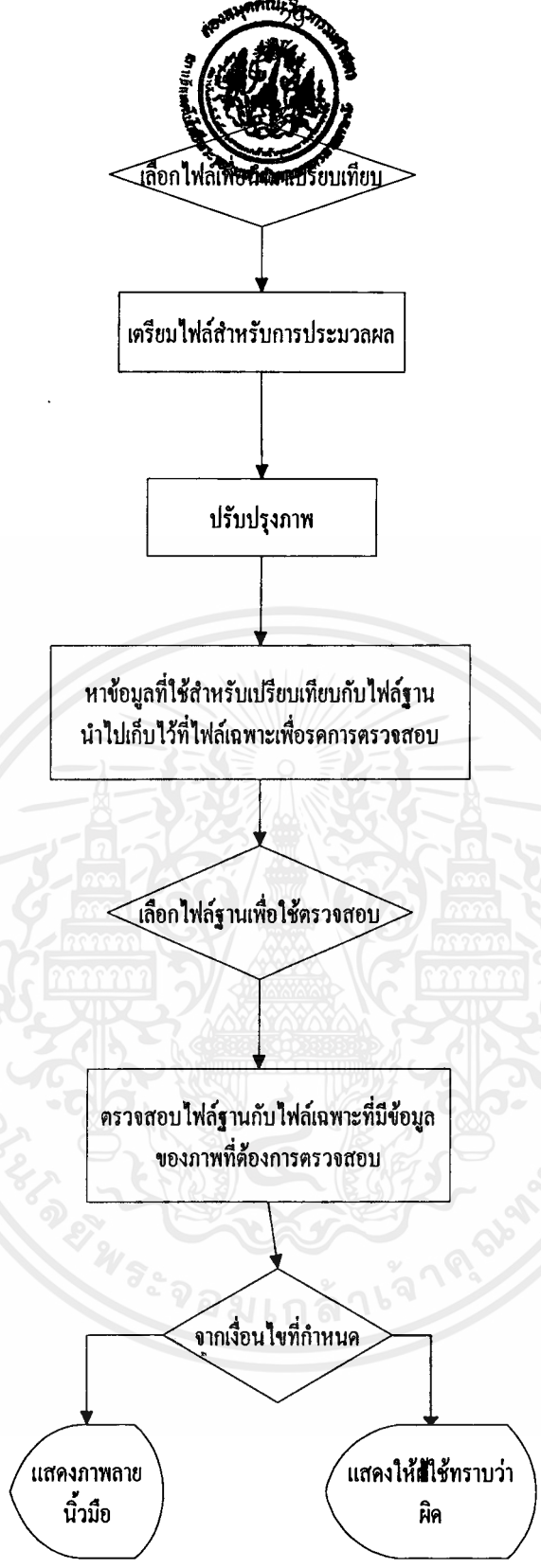
สำหรับการหาค่าความชันนั้น จะทำได้โดยการ สারণข้อมูลจากไฟล์ที่ต้องการทำเป็นฐานข้อมูล และนำไฟล์ที่ทำการสারণข้อมูลนี้มาไล่ตั้งแต่ตำแหน่งมุมซ้ายสุดของภาพ ไปยังตำแหน่งมุมขวาสุดของภาพ โดยระหว่างการไล่จุดต่างๆตามภาพนี้ จะทำการตรวจสอบไปด้วยว่า ณ.ที่ตำแหน่งนั้นเป็นจุดปลายใช่หรือไม่ ถ้าไม่ใช่ก็จะทำการไล่จุดตามภาพต่อไป แต่ถ้าที่ตำแหน่งนั้นเป็นจุดปลายของเส้นลายนิ้วมือ ก็จะทำการไล่ตั้งแต่ตำแหน่งของจุดปลายนั้น ไปยังจุดปลายที่เป็นจุดสิ้นสุดของเส้นนั้น โดยที่ระหว่างการไล่จุดไปตามเส้นนั้นก็จะทำการเก็บตำแหน่งเริ่มต้นของเส้นและตำแหน่งจุดปลายของเส้นไว้ด้วย พร้อมทั้งนับความยาวของเส้นนั้นไปด้วย เพื่อที่จะนำความยาวของเส้นมาแบ่งเป็น 3 ส่วนเพื่อทำการหาค่าความชันและเก็บไว้ เพื่อทำการเปรียบเทียบว่าภายในส่วนที่นำมาประมวลผลนี้มีค่าความชันหลักคือเท่าไร แล้วจึงนำไปเก็บไว้เพื่อสร้างเป็นฐานของข้อมูล โดยจะทำการคำนวณเช่นนี้ทั้งหมด 9 ครั้ง ภายในส่วนต่างๆของภาพ



รูปที่ 4.2 การเก็บข้อมูลที่ได้จากการประมวลผล

## 2. ส่วนของการเปรียบเทียบไฟล์กับฐานข้อมูล

สำหรับส่วนนี้จะทำการประมวลผลเช่นเดียวกับส่วนสร้างฐานข้อมูล แต่จะมีการเก็บไฟล์ไว้เพื่อทำการเปรียบเทียบ ณ. ไฟล์ที่เราทำการกำหนด เมื่อทำการสั่งให้โปรแกรมทำการตรวจสอบ ต่อจากนั้นผู้ใช้งานจะต้องทำการเลือกไฟล์ที่ต้องการเปรียบเทียบ ( ชื่อของไฟล์ฐานก็จะทำหน้าที่คล้ายๆกับเป็นรหัส ) กับไฟล์ที่ทำการเปิดขึ้นมา โปรแกรมก็จะทำการตรวจสอบข้อมูลในไฟล์ฐานกับข้อมูลของไฟล์ที่ผู้ใช้งานตรวจสอบ ถ้ามีค่าของข้อมูลตรงกันมากกว่าค่าที่กำหนดก็จะเสมือนว่าไฟล์นั้นเป็นไฟล์เดียวกับไฟล์ฐาน



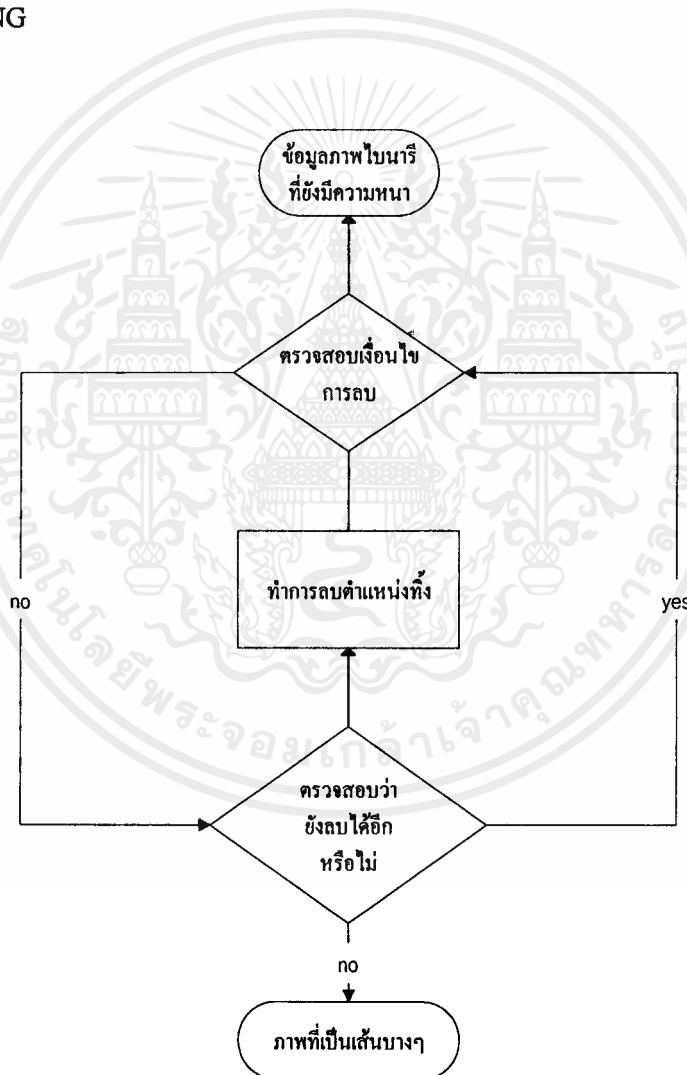
รูปที่ 4.3 แสดงขั้นตอนการเปรียบเทียบ

### 3. ส่วนของการแบ่งชนิดของไฟล์ว่าเป็นลายนิ้วมือชนิดใด

การแบ่งชนิดของลายนิ้วมือ โดยนำลายนิ้วมือที่ผ่านการทำตัวบาง มาแบ่งเป็นบล็อกขนาด  $8 \times 8$  แล้วนำมาหาทิศทางของลายนิ้วมือโดยประมาณ เราใช้ทั้งหมด 4 ทิศทาง คือ แนวตั้ง แนวนอน เฉียงซ้าย เฉียงขวา การหาทิศทางทำโดยนำ เมตริก 4 ชนิด มาคูณ(Convolution) บล็อกแต่ละบล็อก โดยเมตริกดังกล่าว จะเป็นตัวคอยตรวจสอบ ว่าทิศทางในแต่ละบล็อกเป็นอย่างไร สังเกตจากผลลัพธ์ที่เกิดจากการคูณ จะมีศูนย์เกิดขึ้น นับจำนวนศูนย์ที่เกิดขึ้น ว่าเมตริกชนิดใด เมื่อคูณกับภาพแล้ว ให้ผลลัพธ์เป็นศูนย์มากที่สุด แสดงว่า บล็อกมีทิศทางเช่นเดียวกับเมตริกนั้น เมื่อได้ทิศทางแล้ว นำมาวิเคราะห์ว่า ลักษณะเด่นของภาพ ตรงกับข้อกำหนดใด ก็จะได้ชนิดของลายนิ้วมือมาเรียบร้อย

#### Flow Chart การทำงานในส่วนต่างๆ

##### 1. การทำ THINNING



รูปที่ 4.5 flow chart การทำ thinning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. การหาทิศทาง



รูปที่ 4.6 flow chart การแบ่งชนิดของลายนิ้วมือ

## บทที่ 5

### ผลการทดลอง สรุป และวิจารณ์

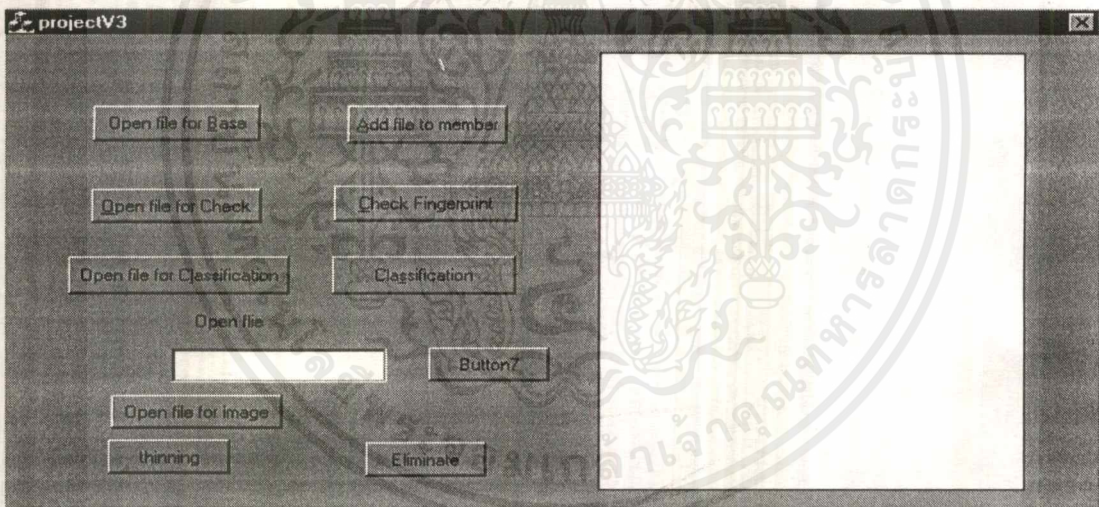
โครงการนี้จะเป็นการตรวจสอบไฟล์ของภาพลายนิ้วมือ และจะพยายามแบ่งประเภทของภาพลายนิ้วมือ โดยแบ่งการใช้งานออกเป็น 3 ส่วนคือ

1. ส่วนของการสร้างไฟล์ฐานของข้อมูล
2. ส่วนของการเปรียบเทียบไฟล์ข้อมูลกับไฟล์ฐาน
3. ส่วนของการแยกประเภทไฟล์ภาพลายนิ้วมือ

โดยการทำงานของแต่ละส่วนจะเป็นอิสระต่อกัน โดยผู้ใช้สามารถเลือกใช้งานส่วนใดก็ได้ การทำงานของทุกส่วนจะเริ่มจากการเลือกไฟล์เพื่อนำมาใช้ในการประมวลผล เมื่อผู้ใช้เลือกไฟล์ที่ต้องการนำมาประมวลเรียบร้อยแล้ว แล้วจึงทำการส่งให้คอมพิวเตอร์ประมวลผลตามฟังก์ชันที่ผู้ใช้ต้องการ

#### 1. หน้าจอเมื่อสั่งให้โปรแกรมทำงาน

เมื่อผู้ใช้ทำการเปิด โปรแกรมก็จะได้น้ำจอ ดังนี้



รูปที่ 5.1 หน้าจอเมื่อสั่งรันโปรแกรม

จากหน้าจอของโปรแกรมจะสามารถแบ่งกลุ่มการใช้งานเป็น 3 กลุ่มดังที่ได้กล่าวมาแล้ว และ แบ่งเป็นกลุ่มย่อยได้อีก 1 กลุ่มคือ

#### 1. ส่วนการสร้างไฟล์ฐานข้อมูล ได้แก่

- 1.1 Open file for Base เมื่อทำการใช้งานฟังก์ชันนี้ โปรแกรมจะทำการเรียกหน้าจอ open

file ของ windows เพื่ออนุญาตให้ผู้ใช้ทำการเลือกไฟล์ที่ผู้ใช้ต้องการทำเป็นฐานข้อมูลต่อจากนั้นจึงทำการเลือกฟังก์ชันต่อไป

1.2 Add file to Member เป็นฟังก์ชันการทำงาน โดยจะใช้งานต่อจาก ฟังก์ชัน Open file

for base เมื่อทำงานฟังก์ชันนี้ โปรแกรมจะทำการเรียกหน้าจอ Save file ของ Windows เพื่อให้ผู้ใช้ทำการกำหนดชื่อไฟล์ (เปรียบเสมือนการตั้งรหัส) แล้ว โปรแกรมจะทำการประมวลผลหาข้อมูลที่ต้องการเก็บเพื่อใช้ในการเปรียบเทียบ

2. ส่วนทำการเปรียบเทียบไฟล์ข้อมูล ได้แก่

2.1 Open file for Check เมื่อทำการใช้งานฟังก์ชันนี้ โปรแกรมจะทำการเรียกหน้าจอ

Open file ของ Windows เพื่อให้ผู้ใช้ทำการเลือกไฟล์ที่ต้องการตรวจสอบ แล้ว โปรแกรมจะทำการประมวลผลหาข้อมูลที่ต้องการใช้สำหรับการตรวจสอบนำไปเก็บไว้ใน buffer file เพื่อการเปรียบเทียบกับไฟล์ฐาน

2.2 Check fingerprint เป็นฟังก์ชันการทำงานต่อจากฟังก์ชัน Open file for check เมื่อ

ทำงาน โปรแกรมจะทำการเรียกหน้าจอ Save file ของ Windows เพื่อให้ผู้ใช้เลือกไฟล์ฐานที่ต้องการให้ไฟล์ที่เลือกมาข้างต้นตรวจสอบด้วย

3. ส่วนทำการแบ่งประเภทลายนิ้วมือ ได้แก่

3.1 Open file for Classification เมื่อทำการใช้งานฟังก์ชันนี้ โปรแกรมจะทำการเรียก

หน้าจอ Open file ของ Windows เพื่อให้ผู้ใช้ทำการเลือกไฟล์ที่ต้องการตรวจสอบ

3.2 Classification เป็นฟังก์ชันการทำงานต่อจากฟังก์ชัน Open file for

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Classification เมื่อทำงาน โปรแกรมจะทำการเรียกหน้าจอ Save file ของ Windows เพื่อแสดงทิศทางของเส้นลายนิ้วมือ

#### 4. ส่วนการแสดงผลที่ทำการ Image processing ได้แก่

4.1 **Open file for image** เมื่อทำการใช้งานฟังก์ชันนี้ โปรแกรมจะทำการเรียก

หน้าจอ Open file ของ Windows เพื่อให้ผู้ใช้ทำการเลือกไฟล์ที่ต้องการตรวจสอบ

4.2 **Thinning** เป็นฟังก์ชันการทำงานต่อจากฟังก์ชัน Open file for Image เมื่อ

ทำงาน โปรแกรมจะทำการเรียกหน้าจอ Save file ของ Windows เพื่อให้ผู้ใช้เลือกไฟล์ที่ต้องการเก็บข้อมูลไฟล์ที่แสดงภาพ Thinning

4.3 **Elimination** เป็นฟังก์ชันการทำงานต่อจากฟังก์ชัน Open file for Image

เมื่อทำงาน โปรแกรมจะทำการเรียกหน้าจอ Save file ของ Windows เพื่อให้ผู้ใช้เลือกไฟล์ที่ต้องการเก็บข้อมูลไฟล์ที่แสดงภาพ Elimination

ส่วนการแสดงผลที่ต้องการประมวล หรือภาพที่ได้หลังจากการประมวลผลจะแสดงในส่วน  
ของ จอภาพทางด้านหลัง

### การทดลอง

จากการทดลองได้ทำการทดลองกับภาพลายนิ้วมือ ทั้งหมด 22 ภาพ ในที่นี้จะแสดงเพียงบางภาพ ที่  
จำเป็นเท่านั้น

#### ภาพที่ 1

ภาพลายนิ้วมือปกติ



รูปที่ 5.2 ภาพลายนิ้วมือปกติ

ภาพลายนิ้วมือที่ผ่านการทำบาง

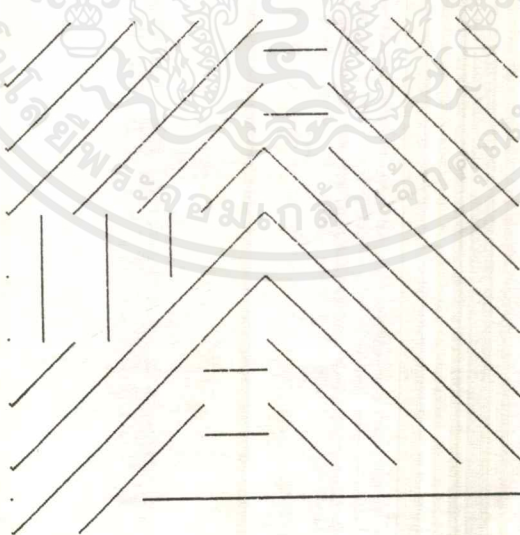


รูปที่ 5.3 ภาพลายนิ้วมือที่ผ่านการทำบาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 ภาพลายนิ้วมือที่กำจัดเส้นกึ่งแล้ว

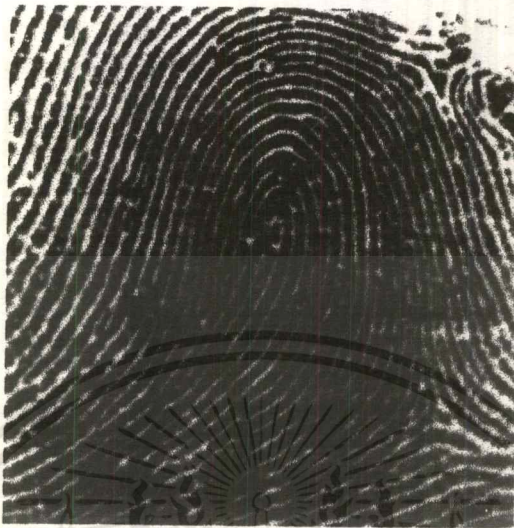


รูปที่ 5.5 ภาพลายนิ้วมือที่ทำการหาทิศทางแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาพที่ 2

## ภาพลายนิ้วมือปกติ



รูปที่ 5.6 ภาพลายนิ้วมือปกติ

## ภาพลายนิ้วมือที่ผ่านการทำบาง



รูปที่ 5.7 ภาพลายนิ้วมือที่ผ่านการทำบาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

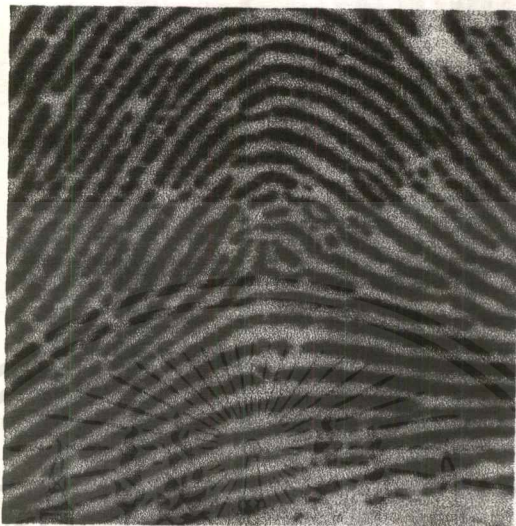


รูปที่ 5.9 ภาพลายนิ้วมือที่ทำการหาทิศทางแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาพที่ 3

## ภาพลายนิ้วมือปกติ



รูปที่ 5.10 ภาพลายนิ้วมือปกติ

## ภาพลายนิ้วมือที่ผ่านการทำบาง

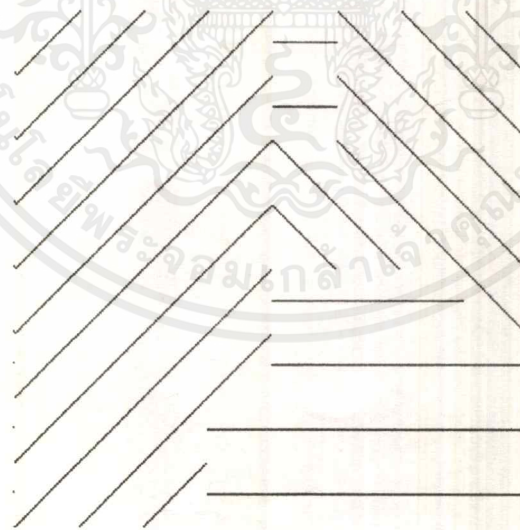


รูปที่ 5.11 ภาพลายนิ้วมือที่ผ่านการทำบาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12 ภาพลายนิ้วมือที่กำจัดเส้นกึ่งแล้ว

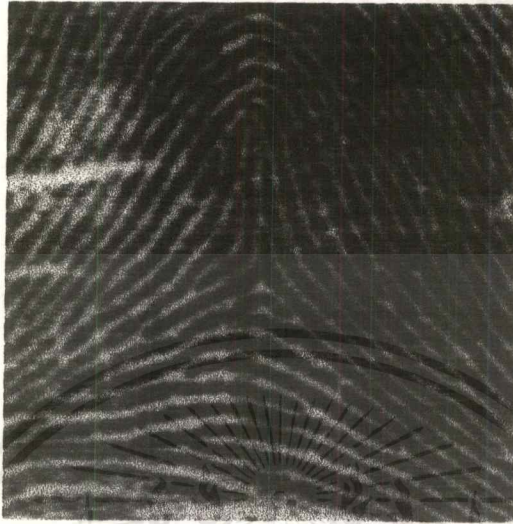


รูปที่ 5.13 ภาพลายนิ้วมือที่ทำการหาทิศทางแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

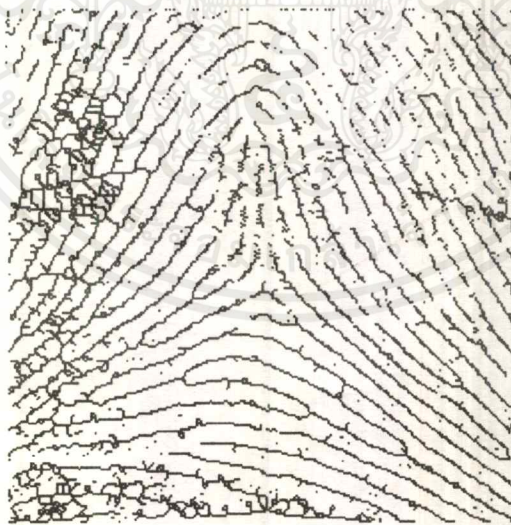
## ภาพที่ 4

## ภาพลายนิ้วมือปกติ



รูปที่ 5.14 ภาพลายนิ้วมือปกติ

## ภาพลายนิ้วมือที่ผ่านการทำบาง

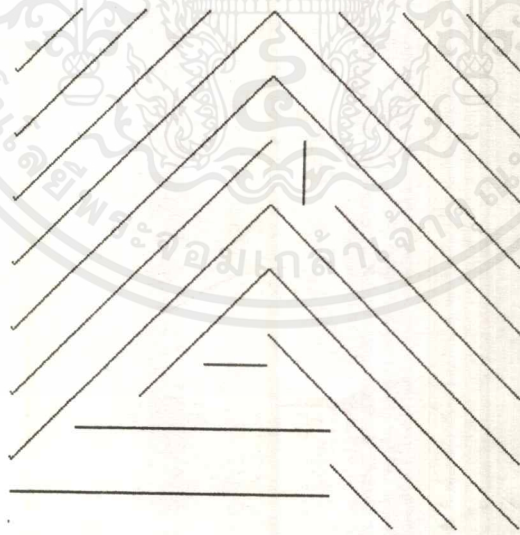


รูปที่ 5.15 ภาพลายนิ้วมือที่ผ่านการทำบาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.16 ภาพลายนิ้วมือที่กำจัดเส้นกึ่งแล้ว



รูปที่ 5.17 ภาพลายนิ้วมือที่ทำการหาทิศทางแล้ว

## ภาพที่ 5

## ภาพลายนิ้วมือปกติ



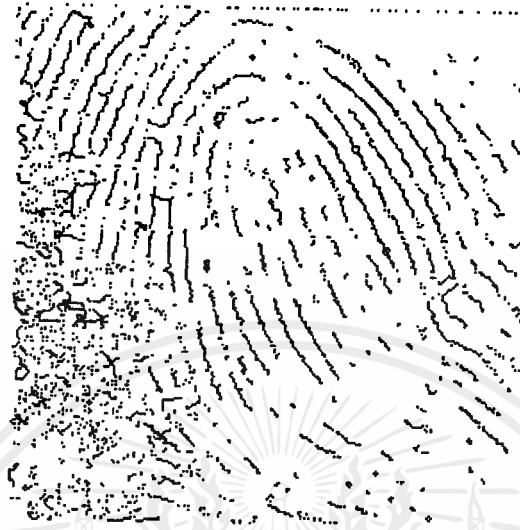
รูปที่ 5.18 ภาพลายนิ้วมือปกติ

## ภาพลายนิ้วมือที่ผ่านการทำบาง

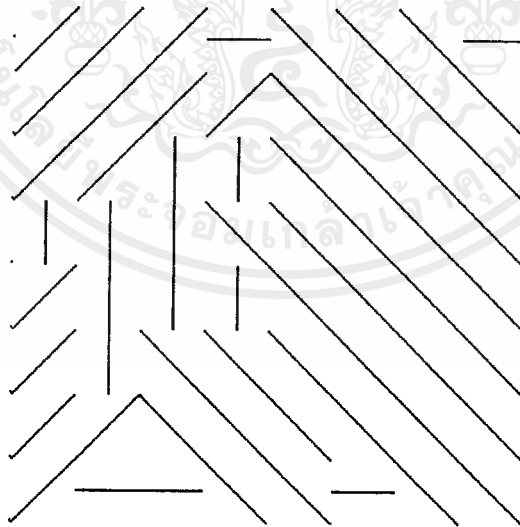


รูปที่ 5.19 ภาพลายนิ้วมือที่ผ่านการทำบาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.20 ภาพลายนิ้วมือที่กำจัดเส้นกึ่งแล้ว



รูปที่ 5.21 ภาพลายนิ้วมือที่ทำการหาทิศทางแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ผลการทดลอง

การทดลองการตรวจสอบลายนิ้วมือ โดยการนำภาพลายนิ้วมือ มาแบ่งแยกชนิด แล้วหาจำนวนภาพที่สามารถวิเคราะห์ได้อย่างถูกต้อง และ ภาพที่วิเคราะห์ผิดพลาด ไปหาเป็นเปอร์เซ็นต์ความผิดพลาด เนื่องจากการทดลองนี้มีภาพลายนิ้วมือเพียง 18 ภาพ ใน 18 ภาพนั้นประกอบไปด้วยลายนิ้วมือ 4ประเภท ซึ่งถือว่าจำนวนภาพน้อยมาก ทำให้เปอร์เซ็นต์ความถูกต้อง หรือ เปอร์เซ็นต์ความผิดพลาด ที่ได้ อาจไม่ตรงกับความจริง

ชนิดของลายนิ้วมือ	จำนวนภาพ	วิเคราะห์ ถูกต้อง	ผิดพลาด
1. เส้นโค้ง	4	4	0
2. กั้นหอย	3	3	0
3. ขดซ้าย	5	4	1
4. ขดขวา	6	5	1
รวม	18	16	2

ตารางแสดงจำนวนการวิเคราะห์ที่ผิดพลาด

จากผลการทดลอง พบการวิเคราะห์ที่ผิดพลาด 2 ภาพ จากภาพทั้งหมด 18 คิดเป็นเปอร์เซ็นต์การผิดพลาด 11เปอร์เซ็นต์ โดยภาพที่ผิดเป็นลายนิ้วมือที่เป็นขด

ความผิดพลาดที่เกิดขึ้นนี้ เนื่องจากการแยกชนิดลายนิ้วมือตามโครงสร้างนี้ ไม่ได้เป็นวิธีที่ชัดเจนนัก เพราะ โครงสร้างของภาพลายนิ้วมือมีมาก และหลากหลาย ในขณะที่โครงสร้างที่เรานำมาเป็นข้อกำหนด ว่าลายนิ้วมืออย่างนี้มีข้อกำหนดอย่างไร ก่อนข้างจำกัด การวิเคราะห์ให้ครอบคลุมทุกกรณี จึงมีข้อผิดพลาด

อย่างไรก็ตาม วิธีนี้ก็ยังมีข้อดีคือ การตรวจสอบใช้อัลกอริทึมที่ไม่ซับซ้อน และไม่ได้ใช้เวลาประมวลผลนาน ถ้าต้องการให้ครอบคลุมหลายกรณี ก็ควรมีข้อกำหนดมากขึ้น

## สรุปและวิจารณ์

การตรวจสอบลายนิ้วมือ มีขั้นตอนเริ่มจากการปรับปรุงภาพก่อน เพราะภาพที่ใช้งานจริงย่อมมีส่วนที่ชัดและไม่ชัด เพราะการรับลายนิ้วมือแต่ละครั้งได้ภาพไม่เหมือนกัน เพื่อให้ได้ผลลัพธ์เหมือนกัน จึงจำเป็นต้องทำให้ภาพที่ไม่ชัดมีความคมชัดมากขึ้น โดยใช้การกรองความถี่สูง (Highpass filter) นอกจากนี้ภาพที่มีส่วนที่รบกวน(noise) ก็กรองออกโดยการกรองความถี่ต่ำ ต่อมาต้องทำการปรับภาพเป็นขาวดำ(Threshold) เพราะการประมวลผลภาพ มักทำกับภาพที่เป็นไบนารี อย่างไรก็ตาม ในภาพที่แย่งจริงๆก็ต้องการความผิดพลาดเกิดขึ้นบ้าง

ขั้นตอนการทำให้เส้นบาง เนื่องจากเส้นลายนิ้วมือมีความหนาอยู่บ้าง แม้ว่าจะไม่มาก การทำให้เส้นบางจะทำให้ลายนิ้วมือ มีโครงสร้างเป็นเส้นที่หนาเพียง 1 พิกเซล เส้นที่บางจะเป็นประโยชน์ในการไล่เส้น(contour following) อย่างไรก็ตามภาพที่ได้จะมีเส้นกึ่ง เกิดขึ้น ถ้าภาพชัด เส้นกึ่งก็จะมีน้อย ถ้าภาพไม่ชัด เส้นกึ่งจะมีมาก

ขั้นตอนการลบเส้นกึ่ง ในขั้นตอนการทำให้เส้นบางจะมีเส้นกึ่งเกิดขึ้น โดยโครงสร้างของเส้นกึ่งมีหลายแบบ จากการศึกษากรณีต่างๆที่ใช้ในที่นี้สามารถลบเส้นกึ่งได้เป็นจำนวนมาก แม้ว่าจะไม่หมด แต่นับว่าใช้ได้ ภาพที่ได้มีเส้นกึ่งเล็กน้อย และ เส้นลายนิ้วมือบางส่วนขาดไปบ้าง และเส้นที่เป็นเส้นเล็กๆเหมือนnoise ก็ หายไปด้วย

ขั้นตอนการแสดงทิศทาง ทิศทางของเส้นลายนิ้วมือสามารถนำไปวิเคราะห์ชนิดของลายนิ้วมือ การแสดงทิศทางนี้แค่แสดง ทิศทางโดยประมาณ มีทั้งหมด 4 ทิศทางคือ แนวตั้ง แนวนอน เฉียงขวา เฉียงซ้าย โดยแบ่งภาพ เป็น 8x8 ส่วน แล้วนำแต่ละส่วนมาคิด จะได้ 1 ทิศทาง เราจะศึกษาชนิดของลายนิ้วมือ จากการศึกษาทิศทางในเมตริก 8x8 การแสดงทิศทางนี้อาจทำให้ละเอียดกว่านี้ก็ได้ แต่ในที่นี้ได้ทดลองแล้วพบว่า 8x8 มีความเหมาะสมแล้ว

ขั้นตอนการวิเคราะห์ นำทิศทางมาวิเคราะห์ โดยเราแบ่งมีลายนิ้วมือ 4 ชนิด คือ ก้นหอย โค้ง ขดซ้าย และขดขวา โดยใช้การศึกษาโครงสร้างในการวิเคราะห์ โดยการค้นหาโครงสร้างที่ประกอบกันเป็น ลูป และ โค้ง แล้วส่วนที่เป็นลูปสามารถวิเคราะห์แบ่งย่อยได้อีก 3 ชนิดที่กล่าวมา อย่างไรก็ตามก็มีข้อผิดพลาด เพราะการศึกษาโครงสร้างดังกล่าว บางครั้งพบลักษณะทั้ง 2 โครงสร้าง ในลายนิ้วมือเดียวกัน



ภาคผนวก

โปรแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// PicDlg.cpp : implementation file
//
#include "string.h"
#include "stdio.h"
#include "stdafx.h"
#include "Project2.h"
#include "PicDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////

// CPicDlg dialog

CPicDlg::CPicDlg(CWnd* pParent /*=NULL*/)
: CDialog(CPicDlg::IDD, pParent)
{
//{{AFX_DATA_INIT(CPicDlg)
// NOTE: the ClassWizard will add member initialization here
//}}AFX_DATA_INIT
}

void CPicDlg::DoDataExchange(CDataExchange* pDX)
{
CDialog::DoDataExchange(pDX);
//{{AFX_DATA_MAP(CPicDlg)
DDX_Control(pDX, IDC_TEGOBMPCTRL1, m_bmp);
//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CPicDlg, CDialog)
//{{AFX_MSG_MAP(CPicDlg)
ON_COMMAND(ID_FILE_OPEN, OnFileOpen)
ON_COMMAND(ID_PROCESS_EDGE, OnProcessEdge)
ON_COMMAND(ID_PROCESS_SHOWFILEFORBASE, OnProcessShowfileforbase)
ON_COMMAND(ID_PROCESS_ADDNEWMEMBER, OnProcessAddnewmember)
ON_COMMAND(ID_FILE_CheckFingerPrint, OnFILECheckFingerPrint)
ON_COMMAND(ID_FILE_SelectFileForCheck, OnFILESelectFileForCheck)
ON_COMMAND(ID_CLASSIFIED_CLASS, OnClassifiedClass)
ON_COMMAND(ID_CLASSIFIED_CHOOSFILEFORCLASSED, OnClassifiedChoosefileforcla
ssed)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////

// CPicDlg message handlers

BOOL CPicDlg::OnInitDialog()
{
CDialog::OnInitDialog();

// TODO: Add extra initialization here
CRect rect;
GetClientRect(&rect);
m_bmp.SetWindowPos(NULL,0,0,rect.right,rect.bottom,NULL);
return TRUE; // return TRUE unless you set the focus to a control
// EXCEPTION: OCX Property Pages should return FALSE
}

void CPicDlg::OnFileOpen()
{
// TODO: Add your command handler code here
char *HeaderDataDialogBox = "";
CFile BmpF;
char FileTitle[100];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OPENFILENAME ofn;
memset(&ofn, 0, sizeof(ofn));
ofn.lStructSize = sizeof(OPENFILENAME);
ofn.hwndOwner=NULL;
ofn.hInstance=NULL;
```

```
ofn.lpstrFilter =
    TEXT("Bitmap picture file *.bmp\0*.bmp\0All File *.*\0*.*\0\0");
```

```
ofn.lpstrCustomFilter=NULL;
ofn.nMaxCustFilter=0;
ofn.nFilterIndex=1;
ofn.lpstrFile=FileName;
ofn.nMaxFile=500;
ofn.lpstrFileTitle=FileTitle;
ofn.nMaxFileTitle=99;
ofn.lpstrInitialDir=NULL;
ofn.lpstrTitle=" Open BMP File";
ofn.Flags= OFN_FILEMUSTEXIST;
ofn.lpstrDefExt="BMP";
ofn.lCustData=NULL;
ofn.lpfHook=NULL;
ofn.lpTemplateName=NULL;
```

```
FileName[0]='\0';
```

```
GetOpenFileName(&ofn);
```

```
if (FileName[0]=='\0')
{
    return;
}
```

```
m_bmp.SetBmpFilename(FileName);
SetWindowText(FileTitle);
}
```

```
void CPicDlg::OnFileExit()
{
    // TODO: Add your command handler code here
    OnOK();
}
```

```
void CPicDlg::OnProcessEdge()
{
    // TODO: Add your command handler code here
    /*
        BmpG.Close();
        BmpF.Close();
        BmpFW.Close();
        m_bmp.SetBmpFilename(SFileName1);
        SetWindowText(FileTitle);
    */
}
```

```
void CPicDlg::OnProcessShowfileforbase()
{
    // TODO: Add your command handler code here
    char *HeaderDataDialogBox ="";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CFile BmpF;
char FileTitle[100];

OPENFILENAME ofn;
memset(&ofn, 0, sizeof(ofn));
ofn.lStructSize = sizeof(OPENFILENAME);
ofn.hwndOwner=NULL;
ofn.hInstance=NULL;

ofn.lpstrFilter =
    TEXT("Bitmap picture file *.bmp\0*.bmp\0All File *.*\0*.*\0\0");

ofn.lpstrCustomFilter=NULL;
ofn.nMaxCustFilter=0;
ofn.nFilterIndex=1;
ofn.lpstrFile=FileName;
ofn.nMaxFile=500;
ofn.lpstrFileTitle=FileTitle;
ofn.nMaxFileTitle=99;
ofn.lpstrInitialDir=NULL;
ofn.lpstrTitle=" Open BMP File";
ofn.Flags= OFN_FILEMUSTEXIST;
ofn.lpstrDefExt="BMP";
ofn.lCustData=NULL;
ofn.lpfnHook=NULL;
ofn.lpTemplateName=NULL;

FileName[0]='\0';

GetOpenFileName(&ofn);

if (FileName[0]!='\0')
{
    return;
}

m_bmp.SetBmpFilename(FileName);
SetWindowText(FileTitle);
//BmpF.Close();
}

void CPicDlg::OnProcessAddnewmember()
{
    // TODO: Add your command handler code here

    // TODO: Add your command handler code here
    char FileTitle[100];
    OPENFILENAME sfn;
    memset(&sfn, 0, sizeof(sfn));
    sfn.lStructSize = sizeof(OPENFILENAME);
    sfn.hwndOwner=NULL;
    sfn.hInstance=NULL;

    sfn.lpstrFilter =
        TEXT("Gray picture file *.tvt\0*.tvt\0All File *.*\0*.*\0\0");

    sfn.lpstrCustomFilter=NULL;
    sfn.nMaxCustFilter=0;
    sfn.nFilterIndex=1;
    sfn.lpstrFile=SFileName;
    sfn.nMaxFile=500;
    sfn.lpstrFileTitle=FileTitle;
    sfn.nMaxFileTitle=99;
    sfn.lpstrInitialDir=NULL;
    sfn.lpstrTitle=" Save Gray File";
    sfn.Flags= OFN_FILEMUSTEXIST;
    sfn.lpstrDefExt="TXT";
    sfn.lCustData=NULL;
    sfn.lpfnHook=NULL;
    sfn.lpTemplateName=NULL;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SFileName[0]='\0';

GetSaveFileName(&sfname);

if (SFileName[0]!='\0')
{
    return;
}

char IdBmp[2];
typedef struct
{
    //char IdBmp[2];
    long FileSize;
    short Reserved1[2];
    long HeaderSize;
    long InfoSize;
    long Width;
    long Depth;
    short biPlanes;
    short Bits;
    long biCompression;
    long biSizeImage;
    long biXPelsPerMeter;
    long biYPelsPerMeter;
    long biClrUsed;
    long biClrImportant;
} BmpHead;

BmpHead BmpHeadData;
BmpHead BmpHeadWrite;

// char *HeaderDataDialogBox = "";
unsigned char Gray8Bits1 = 0;
//double Gray8Bits2 = 0;

int X_width;
int Y_depth;

CFile BmpF;
CFile BmpG;
CFile TXIFW;

TXIFW.Open(SFileName,CFile::modeCreate | CFile::modeWrite);

if (BmpF.Open(FileName, CFile::modeRead)==FALSE )
{
    return;
}
if (BmpG.Open("C:\\WINDOWS\\GTEST.BMP", CFile::modeRead)==FALSE)
{
    return;
}

TXIFW.SeekToBegin();
BmpF.SeekToBegin();

BmpG.Read( IdBmp, 2);
BmpG.Read( &BmpHeadData, sizeof(BmpHead) );

BmpF.Read( IdBmp, 2);
BmpF.Read( &BmpHeadData, sizeof(BmpHead) );

/// Start Change to Gray ///
////////////////////////////////////
int a,count;
unsigned char* pimage;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pimage=(unsigned char *)malloc(BmpHeadData.Width1*BmpHeadData.Depth1);

BmpF.Seek( BmpHeadData.HeaderSize1, CFile::begin );

typedef struct {
    unsigned char Red24Bits;
    unsigned char Green24Bits;
    unsigned char Blue24Bits;
    //unsigned char RgbReserved;
} Rgb24Bits;

Rgb24Bits Rgb24Bits1;
Rgb24Bits Rgb24Bits2;

count =1;
for (Y_depth=1;Y_depth<=BmpHeadData.Depth1;Y_depth++)
{
    for (X_width=1;X_width<=BmpHeadData.Width1;X_width++)
    {
        BmpF.Read( &Rgb24Bits1, sizeof(Rgb24Bits));

        Gray8Bits1 = ( Rgb24Bits1.Red24Bits * 299 / 1000 ) + ( Rgb24Bits1.Green24Bits * 587 / 1000 ) + ( Rgb24Bits1.Blue24Bits * 114 /1000 );
        Rgb24Bits2.Red24Bits = Gray8Bits1;
        Rgb24Bits2.Green24Bits = Gray8Bits1;
        Rgb24Bits2.Blue24Bits = Gray8Bits1;

        pimage[count++]=Gray8Bits1;

    } // end for X_width loop
} // end for Y_depth loop

unsigned char Gray8Bits;
int x,y;

int mask5[3][3]={ {1,-2,1},
                  {-2,5,-2},
                  {1,-2,1} };

int j,b1,n[20],y2,y3,y4,x2,x3,x4,dx[20],dy[20],t,p1;
int lx[7],ly[7],sum1,c1,p,p2,q,m,s1[5],s2[5],wy[5],wx[5];

PicSize zoomimage2;

count=1;

for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{ Gray8Bits=pimage[count++];
  zoomimage2[y][x]=Gray8Bits;
}

unsigned long color_sum,color_average;
int diff_color;

unsigned char image32[32][32];
Convolution(mask5,zoomimage2);

color_sum=0;
for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{ color_sum=color_sum+zoomimage2[y][x];}

color_average=color_sum/65536;
diff_color=color_average-134;

for(y=0;y<=255;y++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 \*ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(x=0;x<=255;x++)
{ zoomimage2[y][x]=zoomimage2[y][x]-diff_color;}

for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{ Gray8Bits=zoomimage2[y][x];
  if(Gray8Bits>=152){Gray8Bits=255;}
  else{Gray8Bits=0;}
  zoomimage2[y][x]=Gray8Bits;
}

PicSize zoomimage3;
/*1
int max,p,q,zero;

int sampling[8][8]= { {1,1,1,1,1,1,1,1},
                      {1,1,1,1,1,1,1,1},
                      {1,1,1,1,1,1,1,1},
                      {1,1,1,1,1,1,1,1},
                      {1,1,1,1,1,1,1,1},
                      {1,1,1,1,1,1,1,1},
                      {1,1,1,1,3,3,3,3},
                      {1,1,1,1,4,4,3,3}
};

for(j=1;j<=8;j++)
for(i=1;i<=8;i++)
{
  p=0;q=0;
  for(y=(j-1)*32;y<=(j*32)-1;y++)
  {
    for(x=(i-1)*32;x<=(i*32)-1;x++)
    { image32[p][q]=zoomimage2[y][x];
      q++;
      if(q==32){q=0;}
    }
    p++;
  }

  max=0;
  zero=Convolution2(mask1,image32);if(zero>max){max=zero;sampling[j-1][i-1]=1;}
  zero=Convolution2(mask2,image32);if(zero>max){max=zero;sampling[j-1][i-1]=2;}
  zero=Convolution2(mask3,image32);if(zero>max){max=zero;sampling[j-1][i-1]=3;}
  zero=Convolution2(mask4,image32);if(zero>max){max=zero;sampling[j-1][i-1]=4;}
}

  show_sampling(zoomimage3,sampling); */

Thining(zoomimage2);
//eliminate(zoomimage2);

int x1,y1,ry[8],rx[8],sy[8],sx[8],flag,f,b;
int d,k,c,i[9],sum;

for(y=1;y<255;y++)
{
  for(x=1;x<255;x++)
  {
    y1=y; x1=x;
    if(zoomimage2[y][x]==0)
    {
      i[0]=zoomimage2[y][x+1];
      i[1]=zoomimage2[y-1][x+1];
      i[2]=zoomimage2[y-1][x];
      i[3]=zoomimage2[y-1][x-1];
      i[4]=zoomimage2[y][x-1];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

i[5]=zoomimage2[y+1][x-1];
i[6]=zoomimage2[y+1][x];
i[7]=zoomimage2[y+1][x+1];
i[8]=zoomimage2[y][x+1];
sum=0;
for(a=0;a<=7;a++)
{
b=i[a]-i[a+1];
if(b<0){b=b*(-1);}
sum=sum+b;
} //for(a)
if(sum==255*6)
{
c=0;
ry[c]=y; rx[c]=x; c++;
for(a=0;a<=7;a++)
{
if(i[a]==0)
{
switch(a)
{
case 0: ry[c]=y; rx[c]=x+1; c++;break;
case 1: ry[c]=y-1; rx[c]=x+1; c++;break;
case 2: ry[c]=y-1; rx[c]=x; c++;break;
case 3: ry[c]=y-1; rx[c]=x-1; c++;break;
case 4: ry[c]=y; rx[c]=x-1; c++;break;
case 5: ry[c]=y+1; rx[c]=x-1; c++;break;
case 6: ry[c]=y+1; rx[c]=x; c++;break;
case 7: ry[c]=y+1; rx[c]=x+1; c++;break;
} //switch
} //if(i[a]==0)
} //for(a)
for(a=1;a<c;a++)
{ flag=0;
d=0;
i[0]=zoomimage2[ry[a]][rx[a]+1];
i[1]=zoomimage2[ry[a]-1][rx[a]+1];
i[2]=zoomimage2[ry[a]-1][rx[a]];
i[3]=zoomimage2[ry[a]-1][rx[a]-1];
i[4]=zoomimage2[ry[a]][rx[a]-1];
i[5]=zoomimage2[ry[a]+1][rx[a]-1];
i[6]=zoomimage2[ry[a]+1][rx[a]];
i[7]=zoomimage2[ry[a]+1][rx[a]+1];
for(b=0;b<=7;b++)
{
if(i[b]==0)
{
switch(b)
{
case 0: sy[d]=ry[a]; sx[d]=rx[a]+1; d++;break;
case 1: sy[d]=ry[a]-1; sx[d]=rx[a]+1; d++;break;
case 2: sy[d]=ry[a]-1; sx[d]=rx[a]; d++;break;
case 3: sy[d]=ry[a]-1; sx[d]=rx[a]-1; d++;break;
case 4: sy[d]=ry[a]; sx[d]=rx[a]-1; d++;break;
case 5: sy[d]=ry[a]+1; sx[d]=rx[a]-1; d++;break;
case 6: sy[d]=ry[a]+1; sx[d]=rx[a]; d++;break;
case 7: sy[d]=ry[a]+1; sx[d]=rx[a]+1; d++;break;
} //switch
} //if(i[b]==0)
} //for(b)
f=0;
for(b=0;b<d;b++)
{
for(k=0;k<c;k++)
{
if((sy[b]==ry[k])&(sx[b]==rx[k]))
{
f++;
} //if
} //for(k)
} //for(b)
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(f!=d){ flag=1;}
        if(flag==0){zoomimage2[ry[a]][rx[a]]=255;}
    }//for(a)
} //if(sum==255*6)
} //if(zoomimage2==0)
y=y1; x=x1;
} //for(x)
} //for(y)

int KY[2],KX[2];
float fslope[9];

for(a=0;a<=8;a++)
{
    fslope[a]=0;
}

/// case 1
S_D(zoomimage2,68,68,KY,KX);
if(KX[1]==KX[0]){ fslope[0]=99; }
else{ fslope[0]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 2
S_D(zoomimage2,68,108,KY,KX);
if(KX[1]==KX[0]){ fslope[1]=99; }
else{ fslope[1]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 3
S_D(zoomimage2,68,148,KY,KX);
if(KX[1]==KX[0]){ fslope[2]=99; }
else{ fslope[2]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 4
S_D(zoomimage2,108,68,KY,KX);
if(KX[1]==KX[0]){ fslope[3]=99; }
else{ fslope[3]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 5
S_D(zoomimage2,108,108,KY,KX);
if(KX[1]==KX[0]){ fslope[4]=99; }
else{ fslope[4]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 6
S_D(zoomimage2,108,148,KY,KX);
if(KX[1]==KX[0]){ fslope[5]=99; }
else{ fslope[5]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 7
S_D(zoomimage2,148,68,KY,KX);
if(KX[1]==KX[0]){ fslope[6]=99; }
else{ fslope[6]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 8
S_D(zoomimage2,148,108,KY,KX);
if(KX[1]==KX[0]){ fslope[7]=99; }
else{ fslope[7]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 9
S_D(zoomimage2,148,148,KY,KX);
if(KX[1]==KX[0]){ fslope[8]=99; }
else{ fslope[8]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

```

```

////////////////////////////////////
////////////////////////////////////

```

```

    TXIFW.SeekToBegin;
    float ddl;
    for( a=0;a<=8;a++)
    {
        ddl= fslope[a];
        TXIFW.Write( &ddl, sizeof(ddl));
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    ///////////////////////////////////////////////////////////////////
    ///////////////////////////////////////////////////////////////////

    TXIFW.Close();
    BmpG.Close();
    BmpF.Close();
    m_bmp.SetBmpFilename(SFileName);
    SetWindowText(FileName);
}

```

```
void CPicDlg::OnFILESelectFileForCheck()
```

```

{
    // TODO: Add your command handler code here
    CFile BmpF;
    CFile BmpG;
    CFile TXIFW;

    char FileName[100];

    OPENFILENAME ofn;
    memset(&ofn, 0, sizeof(ofn));
    ofn.lStructSize = sizeof(OPENFILENAME);
    ofn.hwndOwner=NULL;
    ofn.hInstance=NULL;

    ofn.lpstrFilter =
        TEXT("Bitmap picture file *.bmp\0*.bmp\0All File *.*\0*\0");

    ofn.lpstrCustomFilter=NULL;
    ofn.nMaxCustFilter=0;
    ofn.nFilterIndex=1;
    ofn.lpstrFile=OFileName;
    ofn.nMaxFile=500;
    ofn.lpstrFileTitle=FileName;
    ofn.nMaxFileTitle=99;
    ofn.lpstrInitialDir=NULL;
    ofn.lpstrTitle=" Open BMP File";
    ofn.Flags= OFN_FILEMUSTEXIST;
    ofn.lpstrDefExt="BMP";
    ofn.lCustData=NULL;
    ofn.lpfnHook=NULL;
    ofn.lpTemplateName=NULL;

    FileName[0]='\0';

    GetOpenFileName(&ofn);

    if (OFileName[0]!='\0')
    {
        return;
    }

    TXIFW.Open("C:\\BASE\\TEST.TXT",CFile::modeCreate | CFile::modeWrite);

    if (BmpF.Open(OFileName, CFile::modeRead)==FALSE)
    {
        return;
    }
    if (BmpG.Open("C:\\WINDOWS\\GTEST.BMP", CFile::modeRead)==FALSE)
    {
        return;
    }

    char IdBmp[2];
    typedef struct

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    //char IdBmp[2];
    long FileSize1;
    short Reserved1[2];
    long HeaderSize1;
    long InfoSize1;
    long Width1;
    long Depth1;
    short biPlanes1;
    short Bits1;
    long biCompression1;
    long biSizeImage1;
    long biXPelsPerMeter1;
    long biYPelsPerMeter1;
    long biClrUsed;
    long biClrImportant;
} BmpHead1;

BmpHead1 BmpHeadData;
BmpHead1 BmpHeadWrite;

unsigned char Gray8Bits1 = 0;

int X_width;
int Y_depth;

TXIFW.SeekToBegin();
BmpF.SeekToBegin();

BmpG.Read( IdBmp, 2);
BmpG.Read( &BmpHeadData, sizeof(BmpHead1) );

BmpF.Read( IdBmp, 2);
BmpF.Read( &BmpHeadData, sizeof(BmpHead1) );

    /// Start Change to Gray ///
    ///////////////////////////////////
    int a,count;
    unsigned char* pimage;
    pimage=(unsigned char *)malloc(BmpHeadData.Width1*BmpHeadData.Depth1);

    BmpF.Seek( BmpHeadData.HeaderSize1, CFile::begin );

    typedef struct {
        unsigned char Red24Bits;
        unsigned char Green24Bits;
        unsigned char Blue24Bits;
        //unsigned char RgbReserved;
    } Rgb24Bits;

    Rgb24Bits Rgb24Bits1;
    Rgb24Bits Rgb24Bits2;

    count =1;
    for (Y_depth=1;Y_depth<=BmpHeadData.Depth1;Y_depth++)
    {
        for (X_width=1;X_width<=BmpHeadData.Width1;X_width++)
        {
            BmpF.Read( &Rgb24Bits1, sizeof(Rgb24Bits));

            Gray8Bits1 = ( Rgb24Bits1.Red24Bits * 299 / 1000) + ( Rgb24Bits1.Gr
een24Bits * 587 / 1000 ) + ( Rgb24Bits1.Blue24Bits * 114 /1000 );
            Rgb24Bits2.Red24Bits = Gray8Bits1;
            Rgb24Bits2.Green24Bits = Gray8Bits1;
            Rgb24Bits2.Blue24Bits = Gray8Bits1;

            pimage[count++]=Gray8Bits1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        } // end for X_width loop
    }
    //////////////////////////////////////
    unsigned char Gray8Bits;
    int x,y;

    int mask5[3][3]={ {1,-2,1},
                      {-2,5,-2},
                      {1,-2,1} };

    int j,b1,n[20],y2,y3,y4,x2,x3,x4,dx[20],dy[20],t,p1;
    int lx[7],ly[7],sum1,c1,p,p2,q,m,s1[5],s2[5],wy[5],wx[5];

    PicSize zoomimage2;

    count=1;

    for(y=0;y<=255;y++)
    for(x=0;x<=255;x++)
    { Gray8Bits=pimage[count++];
      zoomimage2[y][x]=Gray8Bits;
    }

    unsigned long color_sum,color_average;
    int diff_color;

    unsigned char image32[32][32];
    Convolution(mask5,zoomimage2);

    color_sum=0;
    for(y=0;y<=255;y++)
    for(x=0;x<=255;x++)
    { color_sum=color_sum+zoomimage2[y][x];}

    color_average=color_sum/65536;
    diff_color=color_average-134;

    for(y=0;y<=255;y++)
    for(x=0;x<=255;x++)
    { zoomimage2[y][x]=zoomimage2[y][x]-diff_color;}

    for(y=0;y<=255;y++)
    for(x=0;x<=255;x++)
    { Gray8Bits=zoomimage2[y][x];
      if(Gray8Bits>=152){Gray8Bits=255;}
      else{Gray8Bits=0;}
      zoomimage2[y][x]=Gray8Bits;
    }

    PicSize zoomimage3;

    Thining(zoomimage2);

    // eliminate2(zoomimage2);
    // x=(int)log10(100);
    int x1,y1,ry[8],rx[8],sy[8],sx[8],flag,f,b;
    int d,k,c,i[9],sum;
    for(y=1;y<255;y++)
    {
        for(x=1;x<255;x++)
        {
            y1=y; x1=x;
            if(zoomimage2[y][x]==0)
            {
                i[0]=zoomimage2[y][x+1];
                i[1]=zoomimage2[y-1][x+1];
                i[2]=zoomimage2[y-1][x];
            }
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

i[3]=zoomimage2[y-1][x-1];
i[4]=zoomimage2[y][x-1];
i[5]=zoomimage2[y+1][x-1];
i[6]=zoomimage2[y+1][x];
i[7]=zoomimage2[y+1][x+1];
i[8]=zoomimage2[y][x+1];
sum=0;
for(a=0;a<=7;a++)
{
  b=i[a]-i[a+1];
  if(b<0){b=b*(-1);}
  sum=sum+b;
}
if(sum==255*6)
{
  c=0;
  ry[c]=y; rx[c]=x; c++;
  for(a=0;a<=7;a++)
  {
    if(i[a]==0)
    {
      switch(a)
      {
        case 0: ry[c]=y; rx[c]=x+1; c++;break;
        case 1: ry[c]=y-1; rx[c]=x+1; c++;break;
        case 2: ry[c]=y-1; rx[c]=x; c++;break;
        case 3: ry[c]=y-1; rx[c]=x-1; c++;break;
        case 4: ry[c]=y; rx[c]=x-1; c++;break;
        case 5: ry[c]=y+1; rx[c]=x-1; c++;break;
        case 6: ry[c]=y+1; rx[c]=x; c++;break;
        case 7: ry[c]=y+1; rx[c]=x+1; c++;break;
      }
    }
  }
  for(a=1;a<c;a++)
  {
    flag=0;
    d=0;
    i[0]=zoomimage2[ry[a]][rx[a]+1];
    i[1]=zoomimage2[ry[a]-1][rx[a]+1];
    i[2]=zoomimage2[ry[a]-1][rx[a]];
    i[3]=zoomimage2[ry[a]-1][rx[a]-1];
    i[4]=zoomimage2[ry[a]][rx[a]-1];
    i[5]=zoomimage2[ry[a]+1][rx[a]-1];
    i[6]=zoomimage2[ry[a]+1][rx[a]];
    i[7]=zoomimage2[ry[a]+1][rx[a]+1];
    for(b=0;b<=7;b++)
    {
      if(i[b]==0)
      {
        switch(b)
        {
          case 0: sy[d]=ry[a]; sx[d]=rx[a]+1; d++;break;
          case 1: sy[d]=ry[a]-1; sx[d]=rx[a]+1; d++;break;
          case 2: sy[d]=ry[a]-1; sx[d]=rx[a]; d++;break;
          case 3: sy[d]=ry[a]-1; sx[d]=rx[a]-1; d++;break;
          case 4: sy[d]=ry[a]; sx[d]=rx[a]-1; d++;break;
          case 5: sy[d]=ry[a]+1; sx[d]=rx[a]-1; d++;break;
          case 6: sy[d]=ry[a]+1; sx[d]=rx[a]; d++;break;
          case 7: sy[d]=ry[a]+1; sx[d]=rx[a]+1; d++;break;
        }
      }
    }
    f=0;
    for(b=0;b<d;b++)
    {
      for(k=0;k<c;k++)
      {
        if((sy[b]==ry[k])&(sx[b]==rx[k]))
        {
          f++;
        }
      }
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        }//for(k)
        }//for(b)
        if(f!=d){ flag=1;}
        if(flag==0){zoomimage2[ry[a]][rx[a]]=255;}
    }//for(a)
} //if(sum==255*6)
} //if(zoomimage2==0)
y=y1; x=x1;
} //for(x)
} //for(y)

int KY[2],KX[2];
float fslope[9];

/// case 1
S_D(zoomimage2,68,68,KY,KX);
if(KX[1]==KX[0]){ fslope[0]=99; }
else{ fslope[0]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 2
S_D(zoomimage2,68,108,KY,KX);
if(KX[1]==KX[0]){ fslope[1]=99; }
else{ fslope[1]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 3
S_D(zoomimage2,68,148,KY,KX);
if(KX[1]==KX[0]){ fslope[2]=99; }
else{ fslope[2]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 4
S_D(zoomimage2,108,68,KY,KX);
if(KX[1]==KX[0]){ fslope[3]=99; }
else{ fslope[3]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 5
S_D(zoomimage2,108,108,KY,KX);
if(KX[1]==KX[0]){ fslope[4]=99; }
else{ fslope[4]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 6
S_D(zoomimage2,108,148,KY,KX);
if(KX[1]==KX[0]){ fslope[5]=99; }
else{ fslope[5]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 7
S_D(zoomimage2,148,68,KY,KX);
if(KX[1]==KX[0]){ fslope[6]=99; }
else{ fslope[6]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 8
S_D(zoomimage2,148,108,KY,KX);
if(KX[1]==KX[0]){ fslope[7]=99; }
else{ fslope[7]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

/// case 9
S_D(zoomimage2,148,148,KY,KX);
if(KX[1]==KX[0]){ fslope[8]=99; }
else{ fslope[8]=(float)(KY[1]-KY[0])/(KX[1]-KX[0]);}

```

```

////////////////////////////////////
////////////////////////////////////

```

```

TXIFW.SeekToBegin;
float dd1;
for( a=0;a<=8;a++)
{
    dd1= fslope[a];
    TXIFW.Write( &dd1, sizeof(dd1));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////////////////////////////////
    m_bmp.SetBmpFilename(OFFileName);
    SetWindowText(FileTitle);
    BmpF.Close();
    BmpG.Close();
    TXTFW.Close();
}

void CPicDlg::OnFILECheckFingerPrint()
{
    CFile TXTF;
    CFile TXTFW;
    char FileTitle[100];

    OPENFILENAME ofn;
    memset(&ofn, 0, sizeof(ofn));
    ofn.lStructSize = sizeof(OPENFILENAME);
    ofn.hwndOwner=NULL;
    ofn.hInstance=NULL;

    ofn.lpstrFilter =
        TEXT("Bitmap picture file *.txt\0*.txt\0All File *.*\0*.*\0");

    ofn.lpstrCustomFilter=NULL;
    ofn.nMaxCustFilter=0;
    ofn.nFilterIndex=1;
    ofn.lpstrFile=TFileName;
    ofn.nMaxFile=500;
    ofn.lpstrFileTitle=FileTitle;
    ofn.nMaxFileTitle=99;
    ofn.lpstrInitialDir=NULL;
    ofn.lpstrTitle=" Open BMP File";
    ofn.Flags= OFN_FILEMUSTEXIST;
    ofn.lpstrDefExt="TXT";
    ofn.lCustData=NULL;
    ofn.lpfnHook=NULL;
    ofn.lpTemplateName=NULL;

    FileName[0]='\0';

    GetOpenFileName(&ofn);

    if (TFileName[0]!='\0')
    {
        return;
    }

    if (TXTF.Open(TFileName, CFile::modeRead)==FALSE )
    {
        return;
    }
    if (TXTFW.Open("C:\\BASE\\Test.TXT", CFile::modeRead)==FALSE)
    {
        return;
    }

    float a,b;
    float slopeA[9],slopeB[9];
    //char f1,f2;
    //char directA[9],directB[9];
    int n,n1;

    char *HeaderDataDialogBox = "";
    TXTF.SeekToBegin();
    TXTFW.SeekToBegin();

    for(n=0;n<=8;n++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    TXIF.Read( &a,sizeof(a));
    slopeA[n]=a;
}

for(n=0;n<=26;n++)
{
    TXIFW.Read( &b,sizeof(b));
    slopeB[n]=b;
}

n=0;
for(n1=0;n1<=8;n1++)
{
    if(slopeA[n1]==slopeB[n1]){ n++;}
}

char G[12]="yotest2,BMP";
char string[80];

strcpy( string, "E:\\PATMAN\\" );
// strcat( string, "strcpy " );
strcat( string, G );
//strcat( string, "strcat!" );
//printf( "String = %s\n", string );

if(n>=5){ n=45;}

{
    m_bmp.SetBmpFilename(string);
}
else
{
    m_bmp.SetBmpFilename("E:\\PATMAN\\yotest2.BMP");
}

SetWindowText(FileTitle);
TXIF.Close();
TXIFW.Close();
}

void CPicDlg::OnClassifiedChoosefileforclased()
{
    // TODO: Add your command handler code here

    CFile BmpF;
    char FileTitle[100];

    OPENFILENAME ofn;
    memset(&ofn, 0, sizeof(ofn));
    ofn.lStructSize = sizeof(OPENFILENAME);
    ofn.hwndOwner=NULL;
    ofn.hInstance=NULL;

    ofn.lpstrFilter =
        TEXT("Bitmap picture file *.bmp\0*.bmp\0All File *.*\0*.*\0\0");

    ofn.lpstrCustomFilter=NULL;
    ofn.nMaxCustFilter=0;
    ofn.nFilterIndex=1;
    ofn.lpstrFile=FileName;
    ofn.nMaxFile=500;
    ofn.lpstrFileTitle=FileTitle;
    ofn.nMaxFileTitle=99;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ofn.lpstrInitialDir=NULL;
ofn.lpstrTitle=" Open BMP File";
ofn.Flags= OFN_FILEMUSTEXIST;
ofn.lpstrDefExt="BMP";
ofn.lCustData=NULL;
ofn.lpfnHook=NULL;
ofn.lpTemplateName=NULL;

FileName[0]='\0';

GetOpenFileName(&ofn);

if (FileName[0]!='\0')
{
    return;
}

m_bmp.SetBmpFilename(FileName);
SetWindowText(FileName);

}

void CPicDlg::OnClassifiedClass()
{
    // TODO: Add your command handler code here

char FileName[100];
OPENFILENAME sfn;
memset(&sfn, 0, sizeof(sfn));
sfn.lStructSize = sizeof(OPENFILENAME);
sfn.hwndOwner=NULL;
sfn.hInstance=NULL;

sfn.lpstrFilter =
    TEXT("Edge picture file *.bmp\0*.bmp\0All File *.*\0*\0*\0");

sfn.lpstrCustomFilter=NULL;
sfn.nMaxCustFilter=0;
sfn.nFilterIndex=1;
sfn.lpstrFile=SFileName;
sfn.nMaxFile=500;
sfn.lpstrFileTitle=FileName;
sfn.nMaxFileTitle=99;
sfn.lpstrInitialDir=NULL;
sfn.lpstrTitle=" Save Edge File";
sfn.Flags= OFN_FILEMUSTEXIST;
sfn.lpstrDefExt="BMP";
sfn.lCustData=NULL;
sfn.lpfnHook=NULL;
sfn.lpTemplateName=NULL;

SFileName[0]='\0';

GetSaveFileName(&sfn);

if (SFileName[0]!='\0')
{
    return;
}

char IdBmp[2];
typedef struct
{
    long FileSize1;
    short Reserved1[2];
    long HeaderSize1;
    long InfoSize1;
    long Width1;
    long Depth1;
    short biPlanes1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

short Bits1;
long biCompression1;
long biSizeImage1;
long biXPelsPerMeter1;
long biYPelsPerMeter1;
long biClrUsed;
long biClrImportant;
} BmpHead1;

```

```

BmpHead1 BmpHeadData;
BmpHead1 BmpHeadWrite;

```

```

int X_width,Y_depth,x,y,count,i,j;
unsigned char* pimage;
unsigned char* outimage;

```

```

char *HeaderDataDialogBox = "";
unsigned char Gray8Bits;

```

```

int mask5[3][3]={
    {1,-2,1},
    {-2,5,-2},
    {1,-2,1} };

```

```

CFile BmpF;
CFile BmpFW;
CFile BmpG;

```

```

BmpFW.Open(SFileName,CFile::modeCreate | CFile::modeWrite );

```

```

if (BmpF.Open(FileName, CFile::modeRead)==FALSE )

```

```

{
    return;
}

```

```

if (BmpG.Open("C:\\Windows\\gtest.BMP", CFile::modeRead)==FALSE)

```

```

{
    return;
}

```

```

BmpF.SeekToBegin();

```

```

BmpG.Read( IdBmp, 2);
BmpG.Read( &BmpHeadData, sizeof(BmpHead1) );

```

```

BmpF.Read( IdBmp, 2);
BmpF.Read( &BmpHeadData, sizeof(BmpHead1) );

```

```

BmpHeadWrite.FileSize1 = BmpHeadData.FileSize1;
BmpHeadWrite.Reserved1[0] = 0;
BmpHeadWrite.Reserved1[1] = 0;
BmpHeadWrite.HeaderSize1 = 1078;
BmpHeadWrite.InfoSize1 = 40;
BmpHeadWrite.Width1 =256;
BmpHeadWrite.Depth1 =256;
BmpHeadWrite.biPlanes1 = 1;
BmpHeadWrite.Bits1 = 8;
BmpHeadWrite.biCompression1 = 0;
BmpHeadWrite.biSizeImage1 = 0;
BmpHeadWrite.biXPelsPerMeter1 = BmpHeadData.biXPelsPerMeter1;
BmpHeadWrite.biYPelsPerMeter1 = BmpHeadData.biYPelsPerMeter1;
BmpHeadWrite.biClrUsed = 0;
BmpHeadWrite.biClrImportant = 0;

```

```

BmpFW.Write( IdBmp, 2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BmpFW.Write( &BmpHeadWrite, sizeof(BmpHead1) );

for (Y_depth=54;Y_depth <= 1078;Y_depth++)
{
    BmpG.Read( &Gray8Bits, 1);
    BmpFW.Write( &Gray8Bits, 1);
}
/// Start Change to Gray ///
////////////////////////////////////
int a;
unsigned char Gray8Bits1=0;
pimage=(unsigned char *)malloc(BmpHeadData.Width1*BmpHeadData.Depth1);

BmpF.Seek( BmpHeadData.HeaderSize1, CFile::begin );

typedef struct {
    unsigned char Red24Bits;
    unsigned char Green24Bits;
    unsigned char Blue24Bits;
    //unsigned char RgbReserved;
} Rgb24Bits;

Rgb24Bits Rgb24Bits1;
Rgb24Bits Rgb24Bits2;

count =1;
for (Y_depth=1;Y_depth<=BmpHeadData.Depth1;Y_depth++)
{
    for (X_width=1;X_width<=BmpHeadData.Width1;X_width++)
    {
        BmpF.Read( &Rgb24Bits1, sizeof(Rgb24Bits));

        Gray8Bits1 = ( Rgb24Bits1.Red24Bits * 299 / 1000) + ( Rgb24Bits1.Green24Bits * 587 / 1000 ) + ( Rgb24Bits1.Blue24Bits * 114 / 1000 );
        Rgb24Bits2.Red24Bits = Gray8Bits1;
        Rgb24Bits2.Green24Bits = Gray8Bits1;
        Rgb24Bits2.Blue24Bits = Gray8Bits1;

        pimage[count++]=Gray8Bits1;

    } // end for X_width loop
} // end for Y_depth loop

////////////////////////////////////
;

PicSize zoomimage2;
count=1;
for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{ Gray8Bits=pimage[count++];
  zoomimage2[y][x]=Gray8Bits;
}

unsigned long color_sum,color_average;
int diff_color;

//Convolution(mask5,zoomimage2);
color_sum=0;
for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{ color_sum=color_sum+zoomimage2[y][x];}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
color_average=color_sum/65536;
diff_color=color_average-134;
```

```
for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{ zoomimage2[y][x]=zoomimage2[y][x]-diff_color;}
```

```
for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{ Gray8Bits=zoomimage2[y][x];
  if(Gray8Bits>=152){Gray8Bits=255;}
  else{Gray8Bits=0;}
  zoomimage2[y][x]=Gray8Bits;
}
```

```
int sampling2[8][8];
int sampling[8][8];
direction(zoomimage2,sampling2);
for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{ zoomimage2[y][x]=255;}
show_sampling(zoomimage2,sampling2);
```

```
for(y=0;y<=7;y++)
for(x=0;x<=7;x++)
{ sampling[7-y][x]=sampling2[y][x];}
```

```
int loop,curve,strtX,strtY;
curve=0;
for(y=0;y<=7;y++)
for(x=0;x<=7;x++)
{ if( (sampling[y][x]==4&&sampling[y][x+1]==3&&sampling[y+1][x-1]==4&&sampling[x+1][y+1]==3)
  || (sampling[y][x]==4&&sampling[y][x+1]==1&&sampling[y][x+2]==3&&sampling[y+1][x-1]==4&&sampling[y+1][x+3]))
  { curve=curve+1;}
```

```
int line_3,line_4,stop,nextX,nextY;
```

```
loop=0;
for(y=0;y<=7;y++)
for(x=0;x<=7;x++)
{
```

```
  if(sampling[y][x]==4&&sampling[y][x+1]==3)
```

```
  {
    strtX=x;strtY=y;i=6;line_3=0;stop=0;
    while(i>=0&&stop==0)
```

```
    {
      if(sampling[strtY][strtX]==4)
      {
        nextX=strtX-1;nextY=strtY+1;
```

```
        if(sampling[strtY+1][strtX]==3)
        {nextX=strtX;nextY=strtY+1;line_3=line_3+1;}
        if(nextX<0||nextX>7||nextY<0||nextY>7){stop=1;}
        if(nextX<0){nextX=strtX;stop=0;}
      }
    }
```

```
    if(sampling[strtY][strtX]==3)
    { line_3=line_3+1;
      nextX=strtX+1;nextY=strtY+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(sampling[strY+1][strX]==4)
        {nextX=strX;nextY=strY+1;}
        if(nextX<0|nextX>7|nextY<0|nextY>7){stop=1;}
        if(nextX>7){nextX=strX;stop=0;}
    }

    if(sampling[strY][strX]==1)
    { nextX=strX+1;nextY=strY;
      if(nextX<0|nextX>7|nextY<0|nextY>7){stop=1;}
    }

    if(sampling[strY][strX]==2)
    { nextX=strX;nextY=strY+1;
      if(nextX<0|nextX>7|nextY<0|nextY>7){stop=1;}
    }

    i--;
    if(line_3>2){stop=1;}
    strX=nextX;strY=nextY;
} ///////////////while

```

```

strX=x+1;strY=y;i=6;line_4=0;stop=0;
while(i>=0&&stop==0)
{
    if(sampling[strY][strX]==4)
    { line_4=line_4+1;
      nextX=strX-1;nextY=strY+1;

      if(sampling[strY+1][strX]==3)
      {nextX=strX;nextY=strY+1;}
      if(nextX<0|nextX>7|nextY<0|nextY>7){stop=1;}
      if(nextX<0){nextX=strX;stop=0;}
    }

    if(sampling[strY][strX]==3)
    {
      nextX=strX+1;nextY=strY+1;

      if(sampling[strY+1][strX]==4)
      {nextX=strX;nextY=strY+1;line_4=line_4+1;}
      if(nextX<0|nextX>7|nextY<0|nextY>7){stop=1;}
      if(nextX>7){nextX=strX;stop=0;}
    }

    if(sampling[strY][strX]==1)
    { nextX=strX+1;nextY=strY;
      if(nextX<0|nextX>7|nextY<0|nextY>7){stop=1;}
    }

    if(sampling[strY][strX]==2)
    { nextX=strX;nextY=strY+1;
      if(nextX<0|nextX>7|nextY<0|nextY>7){stop=1;}
    }

    i--;
    if(line_4>2){stop=1;}
    strX=nextX;strY=nextY;
} ///////////////while

```

```

}//////////if

```

```

    if(line_3>2||line_4>2){loop=loop+1;}
}////////// for

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*
for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{
    if(zoomimage2[y][x]==255){zoomimage2[y][x]=0;}
    else{zoomimage2[y][x]=1;}
} */

for(y=0;y<=255;y++)
for(x=0;x<=255;x++)
{
    Gray8Bits=zoomimage2[y][x];
    BmpFW.Write(&Gray8Bits,1);
}

BmpG.Close();
BmpF.Close();
BmpFW.Close();
m_bmp.SetBmpFilename(SFileName);
SetWindowText(FileTitle);
// MessageBox("this is fingerprint");
if(loop>=1){MessageBox("loop");}
else if(curve>1){MessageBox("tented");}
else{MessageBox("other"); }
}

////////////////////////////////////

////////////////////////////////////
void CPicDlg::Convolution(Mask maskk,PicSize Inimage)
////////////////////////////////////
{
    int a,b,c,d,color;
    PicSize result;

    for (b=1;b<=254;b++)
    {
        for (a=1;a<=254;a++)
        {
            color=0;
            for ( c=-1;c<=1;c++)
            {
                for (d=-1;d<=1;d++)
                { color=color+Inimage[b+c][a+d]*maskk[1+c][1+d];}

            }

            result[b][a]=color;
        }
    }

    for (b=0;b<=255;b++)
    {
        for (a=0;a<=255;a++)
        { color=result[b][a];
            Inimage[b][a]=color;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}
}

////////////////////////////////////
void CPicDlg::Thining(PicSize zoomimage2)
////////////////////////////////////
{
int flag,data,M1,M2,N,S,s1,p1,p2,p3,p4,p5,p6,p7,p8,p9,x,y;
unsigned char tempimage[256][256],zoomimage[256][256];

flag=1;
while(flag--1)
{
for(y=0;y<=255;y++)
{
for(x=0;x<=255;x++)
{
data=zoomimage2[y][x];
tempimage[y][x]=data;
}
} //Now,temp store data

flag=0;
for(y=0;y<=255;y++)
{
for(x=0;x<=255;x++)
{
p1=zoomimage2[y][x];
if(p1==255)
{
p2=zoomimage2[y-1][x];
p3=zoomimage2[y-1][x+1];
p4=zoomimage2[y][x+1];
p5=zoomimage2[y+1][x+1];
p6=zoomimage2[y+1][x];
p7=zoomimage2[y+1][x-1];
p8=zoomimage2[y][x-1];
p9=zoomimage2[y-1][x-1];
M1=p2*p4*p6;
M2=p4*p6*p8;
N=(p2+p3+p4+p5+p6+p7+p8+p9)/255;
S=0;s1=0;
s1=p3-p2;if(s1==255){S=S+255;}s1=0;
s1=p4-p3;if(s1==255){S=S+255;}s1=0;
s1=p5-p4;if(s1==255){S=S+255;}s1=0;
s1=p6-p5;if(s1==255){S=S+255;}s1=0;
s1=p7-p6;if(s1==255){S=S+255;}s1=0;
s1=p8-p7;if(s1==255){S=S+255;}s1=0;
s1=p9-p8;if(s1==255){S=S+255;}s1=0;
s1=p2-p9;if(s1==255){S=S+255;}s1=0;
if((M1==0)&&(M2==0)&&(S==255)&&(N<=6&&N>=2))
{ tempimage[y][x]=0;
flag =1; }
}
} //end for
} //end for

for(y=0;y<=255;y++)
{
for(x=0;x<=255;x++)
{ data=tempimage[y][x];
zoomimage[y][x]=data;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(y=0;y<=255;y++)
{
  for(x=0;x<=255;x++)
  {
    p1=tempimage[y][x];
    if(p1==255)
    {
      p2=tempimage[y-1][x];
      p3=tempimage[y-1][x+1];
      p4=tempimage[y][x+1];
      p5=tempimage[y+1][x+1];
      p6=tempimage[y+1][x];
      p7=tempimage[y+1][x-1];
      p8=tempimage[y][x-1];
      p9=tempimage[y-1][x-1];
      M1=p2*p4*p8;
      M2=p2*p6*p8;
      N=(p2+p3+p4+p5+p6+p7+p8+p9)/255;
      S=0;
      s1=p3-p2;if(s1==255){S=S+255;}s1=0;
      s1=p4-p3;if(s1==255){S=S+255;}s1=0;
      s1=p5-p4;if(s1==255){S=S+255;}s1=0;
      s1=p6-p5;if(s1==255){S=S+255;}s1=0;
      s1=p7-p6;if(s1==255){S=S+255;}s1=0;
      s1=p8-p7;if(s1==255){S=S+255;}s1=0;
      s1=p9-p8;if(s1==255){S=S+255;}s1=0;
      s1=p2-p9;if(s1==255){S=S+255;}s1=0;
      if(M1==0 && M2==0 && S==255 && (N<=6&&N>=2))
      { flag=1;
        zoomimage[y][x]=0;}
    }
  } //end for
} //end for

for(y=0;y<=255;y++)
{
  for(x=0;x<=255;x++)
  { data=zoomimage[y][x];
    zoomimage2[y][x]=data;
  }
}

} //end for thinning

///restore data
for(y=0;y<=255;y++)
{
  for(x=0;x<=255;x++)
  { data=zoomimage2[y][x];
    if(data==255){data=0;}
    else{data=255;}
    zoomimage2[y][x]=data;
  }
}

```

```

////////////////////////////////////
void CPicDlg::eliminate(PicSize image)
////////////////////////////////////
{ int width,height,x,y,data,x1,x2,x3,x4,x5,x6,x7,x8,Nc41,Nc42,Nc81,Nc82;
  int stop,next,Oldx,Oldy,branch;
  int i,j,k,l;//for debug
  PicSize temp;

  width=256;height=256;

  for(y=0;y<=height-1;y++)
  for(x=0;x<=width-1;x++)
  { temp[y][x]=5;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

data=image[y][x];
if(data==0){image[y][x]=1;}
if(data==255){image[y][x]=0;}
}

for(y=1;y<=height-2;y++)
for(x=1;x<=width-2;x++)
{
data=image[y][x];
if(data==1)
{
x1=image[y][x+1];x2=image[y-1][x+1];x3=image[y-1][x];x4=image[y-1][x-1];
x5=image[y][x-1];x6=image[y+1][x-1];x7=image[y+1][x];x8=image[y+1][x+1];

if(x1+x2+x3+x4+x5+x6+x7+x8==0){image[y][x]=0;}

if(x1+x2+x3+x4+x5+x6+x7+x8==1)
{ stop=1;branch=0;next=0;

k=x+1-y;
while((branch<=7)&&(stop==1))
{
x1=image[y][x+1];if(y==Oldy&&Oldx==1+x&&next==1){x1=0;}
x2=image[y-1][x+1];if(Oldy==y-1&&Oldx==1+x&&next==1){x2=0;}
x3=image[y-1][x];if(Oldy==y-1&&Oldx==x&&next==1){x3=0;}
x4=image[y-1][x-1];if(Oldy==y-1&&Oldx==x-1&&next==1){x4=0;}
x5=image[y][x-1];if(y==Oldy&&Oldx==x-1&&next==1){x5=0;}
x6=image[y+1][x-1];if(Oldy==y+1&&Oldx==x-1&&next==1){x6=0;}
x7=image[y+1][x];if(Oldy==y+1&&Oldx==x&&next==1){x7=0;}
x8=image[y+1][x+1];if(Oldy==y+1&&Oldx==1+x&&next==1){x8=0;}

Nc41=(x1-x1*x2*x3)+(x3-x3*x4*x5)+(x5-x5*x6*x7)+(x7-x7*x8*x1);
Nc42=(x2-x2*x3*x4)+(x4-x4*x5*x6)+(x6-x6*x7*x8)+(x8-x8*x1*x2);
x1=1-x1;x2=1-x2;x3=1-x3;x4=1-x4;x5=1-x5;x6=1-x6;x7=1-x7;x8=1-x8;
Nc81=(x1-x*x2*x3)+(x3-x3*x4*x5)+(x5-x5*x6*x7)+(x7-x7*x8*x1);
Nc82=(x2-x2*x3*x4)+(x4-x4*x5*x6)+(x6-x6*x7*x8)+(x8-x8*x1*x2);

if((Nc41==2&&Nc42==1&&Nc82==3)|| (Nc41==1&&Nc42&&Nc81==3)||
(Nc41==3|Nc81==3|Nc82==3)|| (Nc41==3&&Nc42==0)||
(Nc41==0&&Nc82==2&&Nc42==2))
{ stop=0;}
if((Nc41==3&&Nc81==2&&Nc81==3)||Nc82==2)
{ stop=0;}
x1=1-x1;x2=1-x2;x3=1-x3;x4=1-x4;x5=1-x5;x6=1-x6;x7=1-x7;x8=1-x8;
temp[y][x]=0;Oldx=x;Oldy=y;
if(x1==1){y=y;x=x+1;}
if(x2==1){y=y-1;x=x+1;}
if(x3==1){y=y-1;x=x;}
if(x4==1){y=y-1;x=x-1;}
if(x5==1){y=y;x=x-1;}
if(x6==1){y=y+1;x=x-1;}
if(x7==1){y=y+1;x=x;}
if(x8==1){y=y+1;x=x+1;}
next=1;
branch=branch+1;
x1=image[y][x+1];if(y==Oldy&&Oldx==1+x&&next==1){x1=0;}
x2=image[y-1][x+1];if(Oldy==y-1&&Oldx==1+x&&next==1){x2=0;}
x3=image[y-1][x];if(Oldy==y-1&&Oldx==x&&next==1){x3=0;}
x4=image[y-1][x-1];if(Oldy==y-1&&Oldx==x-1&&next==1){x4=0;}
x5=image[y][x-1];if(y==Oldy&&Oldx==x-1&&next==1){x5=0;}
x6=image[y+1][x-1];if(Oldy==y+1&&Oldx==x-1&&next==1){x6=0;}
x7=image[y+1][x];if(Oldy==y+1&&Oldx==x&&next==1){x7=0;}
x8=image[y+1][x+1];if(Oldy==y+1&&Oldx==1+x&&next==1){x8=0;}

if(x1+x2+x3+x4+x5+x6+x7+x8!=1){stop=0;}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} //while
x=k;y=l;
if(stop==0)
{ for(j=0;j<=height-1;j++)
  for(i=0;i<=width-1;i++)
  {
    if(temp[j][i]==0){image[j][i]=0;}
  }
}

} //end point
} //data==1
} //for

for(y=0;y<=height-1;y++)
for(x=0;x<=width-1;x++)
{ data=image[y][x];
if(data==0){image[y][x]=255;}
if(data==1){image[y][x]=0;}
}

}

////////////////////////////////////
void CPicDlg::eliminate2(PicSize image)
////////////////////////////////////
{
  unsigned char i,data ;
  int height,width,x,y;

  height=256;width=256;

  for(y=0;y<=height-1;y++)
  for(x=0;x<=width-1;x++)
  {
    data=image[y][x];
    if(data==0){image[y][x]=1;}
    if(data==255){image[y][x]=0;}
  }

  find_separate(image);

  for(y=0;y<=height-1;y++)
  for(x=0;x<=width-1;x++)
  {
    if(image[y][x]==7)
    { for(i=1;i<=8;i++)
      {
        if(i==1&&image[y][x+1]==1)
        { if(select_and_slope(image,x+1,y,i,x,y)==1)
          {delete_select(image);}
          else{clear_select(image);}
        }

        if(i==2&&image[y-1][x+1]==1)
        { if(select_and_slope(image,x+1,y-1,i,x,y)==1)
          {delete_select(image);}
          else{clear_select(image);}
        }

        if(i==3&&image[y-1][x]==1)
        { if(select_and_slope(image,x,y-1,i,x,y)==1)
          {delete_select(image);}
          else{clear_select(image);}
        }

        if(i==4&&image[y-1][x-1]==1)
        { if(select_and_slope(image,x-1,y-1,i,x,y)==1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {delete_select(image);}
    else{clear_select(image);}
    }

    if(i==5&&image[y][x-1]==1)
    { if(select_and_slope(image,x-1,y,i,x,y)==1)
      {delete_select(image);}
    else{clear_select(image);}
    }

    if(i==6&&image[y+1][x-1]==1)
    { if(select_and_slope(image,x-1,y+1,i,x,y)==1)
      {delete_select(image);}
    else{clear_select(image);}
    }

    if(i==7&&image[y+1][x]==1)
    { if(select_and_slope(image,x,y+1,i,x,y)==1)
      {delete_select(image);}
    else{clear_select(image);}
    }

    if(i==8&&image[y+1][x+1]==1)
    { if(select_and_slope(image,x+1,y+1,i,x,y)==1)
      {delete_select(image);}
    else{clear_select(image);}
    }

} //for
} ////if
} //////////for

for(y=0;y<=height-1;y++)
for(x=0;x<=width-1;x++)
{ data=image[y][x];
  if(data==0){image[y][x]=255;}
  if(data==1||data==7){image[y][x]=0;}
}
}

```

```

////////////////////////////////////
int CPicDlg::select_and_slope(PicSize image,unsigned char strtX,unsigned char s
trtY,int pos,unsigned char BifrX,unsigned char BifrY)
{

```

```

    unsigned char x,y,oldX,oldY,count,distance,temp,k,stop;
    int height,width,delete_it,find_end;
    unsigned char window[9],image1[3][3],no_pixel,data;
    int n[8][8]={
        {8,6,4,2,7,5,3,1},
        {6,5,7,8,4,1,3,2},
        {2,8,6,4,1,7,5,3},
        {8,1,7,2,6,5,3,4},
        {4,2,8,6,3,1,7,5},
        {2,1,3,4,8,7,5,6},
        {6,4,2,8,5,3,1,7},
        {4,3,5,6,2,7,1,8},
    };

```

```

};

int i,j;
float slope;
int store[2],realXY[2];

```

```

k=pos-1;
width=256;height=256;

```

```

for(j=-1;j<=1;j++)
for(i=-1;i<=1;i++)
{ data=image[BifrY+j][BifrX+i];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    image1[1+j][1+i]=data;
    if (data==1){image[BifrY+j][BifrX+i]=4;}
}
    image[strtY][strtX]=1;

    x=strtX;y=strtY;
    oldX=x;oldY=y;
    count=0;distance=0,stop=0;

if(image[y][x]==1)
{
while(count<=15&&stop==0)
{
    y=oldY;x=oldX;no_pixel=0;delete_it=0;find_end=0;

for(j=0;j<=7;j++)
{ i=n[k][j];

    if(i==1)
{ window[i]=image[y][x+1];
    if(window[i]==1)
    { no_pixel=1;
      image[y][x+1]=3;
      store[1]=y;store[0]=x+1;
      temp=(strtX-store[0])*(strtX-store[0])+(strtY-store[1])*(strtY-store[1]);
      if(temp>=distance)
      {distance=temp;realXY[1]=store[1];realXY[0]=store[0];}
    }
} //i==1

    if(i==2)
{ window[i]=image[y-1][x+1];
    if(window[i]==1)
    { no_pixel=1;
      image[y-1][x+1]=3;
      store[1]=y-1;store[0]=x+1;
      temp=(strtX-store[0])*(strtX-store[0])+(strtY-store[1])*(strtY-store[1]);
      if(temp>=distance)
      {distance=temp;realXY[1]=store[1];realXY[0]=store[0];}
    }
} //i==2

    if(i==3)
{ window[i]=image[y-1][x];
    if(window[i]==1)
    { no_pixel=1;
      image[y-1][x]=3;
      store[1]=y-1;store[0]=x;
      temp=(strtX-store[0])*(strtX-store[0])+(strtY-store[1])*(strtY-store[1]);
      if(temp>=distance)
      {distance=temp;realXY[0]=store[0];realXY[1]=store[1];}
    }
} //i==3

    if(i==4)
{ window[i]=image[y-1][x-1];
    if(window[i]==1)
    { no_pixel=1;
      image[y-1][x-1]=3;
      store[1]=y-1;store[0]=x-1;
      temp=(strtX-store[0])*(strtX-store[0])+(strtY-store[1])*(strtY-store[1]);
      if(temp>=distance)
      {distance=temp;realXY[0]=store[0];realXY[1]=store[1];}
    }
} //i==4

    if(i==5)
{ window[i]=image[y][x-1];
    if(window[i]==1)
    { no_pixel=1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(realXY[0]!=strtX)
{ slope=(float)(realXY[1]-strtY)/(realXY[0]-strtX);}

if(realXY[0]-strtX<=1&&realXY[0]-strtX>=-1){slope=99;}
if(realXY[1]-strtY<=1&&realXY[1]-strtY>=-1){slope=0;}
}
if((delete_it==1&&count<=2)||((slope==99||slope==0)&&count<9)){delete_it=1;}
else{delete_it=0;}
return delete_it;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void CPicDlg::delete_select(PicSize image)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
{
    int x,y,height,width;

height=256;width=256;

for(y=0;y<=height-1;y++)
for(x=0;x<=width-1;x++)
{ if(image[y][x]==4||image[y][x]==5||image[y][x]==3)
{image[y][x]=0;}
}

}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void CPicDlg::find_separate(PicSize image)
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
{
    int height,width,x,y;
    int x1,x2,x3,x4,x5,x6,x7,x8,Cb,r1,r2,r3,r4,r5,r6,r7,r8;

height=256;width=256;

for(y=1;y<=height-2;y++)
for(x=1;x<=width-2;x++)
{ if(image[y][x]==1)
{ x1=image[y-1][x];x2=image[y-1][x+1];x3=image[y][x+1];x4=image[y+1][x+
1];
x5=image[y+1][x];x6=image[y+1][x-1];x7=image[y][x-1];x8=image[y-1][x-1];

r1=x2-x1;if(r1<0){r1=r1*(-1);}
r2=x3-x2;if(r2<0){r2=r2*(-1);}
r3=x4-x3;if(r3<0){r3=r3*(-1);}
r4=x5-x4;if(r4<0){r4=r4*(-1);}
r5=x6-x5;if(r5<0){r5=r5*(-1);}
r6=x7-x6;if(r6<0){r6=r6*(-1);}
r7=x8-x7;if(r7<0){r7=r7*(-1);}
r8=x1-x8;if(r8<0){r8=r8*(-1);}
Cb=r1+r2+r3+r4+r5+r6+r7+r8;
if(Cb==6)
{ image[y][x]=7;}

} //if
} //for

}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int CPicDlg::find_separate3(PicSize image)
{ //begin
    unsigned char findit,a,b,sum;
    int height,width,x,y;

    findit=0;
height=256;width=256;
for(y=0;y<=height-1;y++)
for(x=0;x<=width-1;x++)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    store[1]=y;store[0]=x-1;
    temp=(strtX-store[0])*(strtX-store[0])+(strtY-store[1])*(strtY-store[1]);
    if(temp>=distance)
    {distance=temp;realXY[0]=store[0];realXY[1]=store[1];}
}
} //i==5

if(i==6)
{ window[i]=image[y+1][x-1];
if(window[i]==1)
{ no_pixel=1;
image[y+1][x-1]=3;
store[1]=y+1;store[0]=x-1;
temp=(strtX-store[0])*(strtX-store[0])+(strtY-store[1])*(strtY-store[1]);
if(temp>=distance)
{distance=temp;realXY[0]=store[0];realXY[1]=store[1];}
}
} //i==6

if(i==7)
{ window[i]=image[y+1][x];
if(window[i]==1)
{ no_pixel=1;
image[y+1][x]=3;
store[1]=y+1;store[0]=x;
temp=(strtX-store[0])*(strtX-store[0])+(strtY-store[1])*(strtY-store[1]);
if(temp>=distance)
{distance=temp;realXY[0]=store[0];realXY[1]=store[1];}
}
} //i==7

if(i==8)
{ window[i]=image[y+1][x+1];
if(window[i]==1)
{ no_pixel=1;
image[y+1][x+1]=3;
store[1]=y+1;store[0]=x+1;
temp=(strtX-store[0])*(strtX-store[0])+(strtY-store[1])*(strtY-store[1]);
if(temp>=distance)
{distance=temp;realXY[0]=store[0];realXY[1]=store[1];}
}
} //i==8
} //for

if(no_pixel==1)
{
image[realXY[1]][realXY[0]]=5;
count=count+1;
oldX=realXY[0];oldY=realXY[1];
if(find_point7(image,oldX,oldY)==1){stop=1;delete_it=1;}
if(find_endpoint(image,oldX,oldY)==1){stop=1;delete_it=1;}
}
else
{ if(count>0){delete_it=1;}
stop=1;
}
} //while
} //if

for(j=-1;j<=1;j++)
for(i=-1;i<=1;i++)
{ image[BifrY+j][BifrX+i]=image1[1+j][1+i];}

if(no_pixel==1||delete_it==1)
{
image[strtY][strtX]=4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
////////////////////////////////////
```

```
int CPicDlg::Convolution2(Mask maskk,unsigned char Inimage[32][32])
{ int a,b,c,d,color;
  int zero;

  zero=0;

  for (b=1;b<=30;b++)
  for (a=1;a<=30;a++)
  {
    color=0;
    for ( c=-1;c<=1;c++)
    for (d=-1;d<=1;d++)
    { color=color+Inimage[b+c][a+d]*maskk[1+c][1+d];}

    if(color==0){zero=zero+1;}
  }

  return zero;
}
```

```
////////////////////////////////////
void CPicDlg::show_sampling(PicSize zoomimage2,int sampling[8][8])
////////////////////////////////////
```

```
{
int p,q,x,y,i,j;
for(j=1;j<=8;j++)
for(i=1;i<=8;i++)
{
  p=0;q=0;
  for(y=(j-1)*32;y<=(j*32)-1;y++)
  { for(x=(i-1)*32;x<=(i*32)-1;x++)
  {
    if(sampling[j-1][i-1]==4)
    { if(p==q){zoomimage2[y][x]=0;}
    else{zoomimage2[y][x]=255;}
    }
    if(sampling[j-1][i-1]==1)
    { if(p==16){zoomimage2[y][x]=0;}
    else{zoomimage2[y][x]=255;}
    }
    if(sampling[j-1][i-1]==2)
    { if(q==16){zoomimage2[y][x]=0;}
    else{zoomimage2[y][x]=255;}
    }
    if(sampling[j-1][i-1]==3)
    { if(q==31-p){zoomimage2[y][x]=0;}
    else{zoomimage2[y][x]=255;}
    }

    q++;
    if(q==32){q=0;}
  }
  p++;
}
}
```

```
////////////////////////////////////
//// find start and destination //////////////////////////////////////
////////////////////////////////////
```

```
void CPicDlg::S_D(unsigned char zoomimage2[256][256],int aly,int alx,int KY[2],
int KX[2])
{
int comXY[40][40];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int a1,n11[50],y,x,a,count,c1,flag,i[8];
int storeX,storeY,sum,max,maxn,c;
int Y1[100],Y2[100],X1[100],X2[100];
int cx,cy,fx1[2],fy1[2],s11[2];

for(y=0;y<=39;y++)
{
for(x=0;x<=39;x++)
{
comXY[y][x]=255;
}
}

a=0;a1=0;

// a1x=90; //////////////// for start x
// a1y=90; //////////////// for start y
for(y=a1y;y<=a1y+39;y++)
{
for(x=a1x;x<=a1x+39;x++)
{
if(zoomimage2[y][x]==0){comXY[a][a1]=0;}
a1++;
} //end for
a++;a1=0;
} //end for

for(y=0;y<=39;y++)
{
for(x=0;x<=39;x++)
{
comXY[0][x]=255;
comXY[y][0]=255;
comXY[39][x]=255;
comXY[y][39]=255;
} //end for
} //end for

////////////////////////////////////
/// find start and destination ///
////////////////////////////////////

count=0;c1=0;flag=1;a1=0;
for(y=1;y<=39;y++)
{
for(x=1;x<=39;x++)
{
if(comXY[y][x]==0)
{
flag=1;
i[0]=comXY[y-1][x+1];
i[1]=comXY[y-1][x];
i[2]=comXY[y-1][x-1];
i[3]=comXY[y][x-1];
i[4]=comXY[y+1][x-1];
i[5]=comXY[y+1][x];
i[6]=comXY[y+1][x+1];
i[7]=comXY[y][x+1];
sum=i[0]+i[1]+i[2]+i[3]+i[4]+i[5]+i[6]+i[7];
if(sum==255*7)
{
X1[c1]=x;
Y1[c1]=y;
n11[c1]=1;
comXY[y][x]=255;
for(a=0;a<=7;a++)
{
if(i[a]==0)
{
switch(a)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        case 4: storeY=cy+1;
                storeX=cx-1;n11[c1]++;
                break;
        case 5: storeY=cy+1;
                storeX=cx;n11[c1]++;
                break;
        case 6: storeY=cy+1;
                storeX=cx+1;n11[c1]++;
                break;
        case 7: storeY=cy;
                storeX=cx+1;n11[c1]++;
                break;
    }
} //end if
} //end for
} //end if sum=255*7
///// case 3 /////
if(sum==255*6)
{
    a1=0;
    comXY[cy][cx]=255;
    for(a=0;a<=7;a++)
    {
        if(i[a]==0)
        {
            switch(a)
            {
                case 0: fy1[a1]=cy-1;
                        fx1[a1++]=cx+1;n11[c1]++;
                        break;
                case 1: fy1[a1]=cy-1;
                        fx1[a1++]=cx;n11[c1]++;
                        break;
                case 2: fy1[a1]=cy-1;
                        fx1[a1++]=cx-1;n11[c1]++;
                        break;
                case 3: fy1[a1]=cy;
                        fx1[a1++]=cx-1;n11[c1]++;
                        break;
                case 4: fy1[a1]=cy+1;
                        fx1[a1++]=cx-1;n11[c1]++;
                        break;
                case 5: fy1[a1]=cy+1;
                        fx1[a1++]=cx;n11[c1]++;
                        break;
                case 6: fy1[a1]=cy+1;
                        fx1[a1++]=cx+1;n11[c1]++;
                        break;
                case 7: fy1[a1]=cy;
                        fx1[a1++]=cx+1;n11[c1]++;
                        break;
            }
        } //end if
    } //end for
    for(s=0;a<a1;a++)
    {
        cx=fx1[a];cy=fy1[a];
        i[0]=comXY[cy-1][cx+1];
        i[1]=comXY[cy-1][cx];
        i[2]=comXY[cy-1][cx-1];
        i[3]=comXY[cy][cx-1];
        i[4]=comXY[cy+1][cx-1];
        i[5]=comXY[cy+1][cx];
        i[6]=comXY[cy+1][cx+1];
        i[7]=comXY[cy][cx+1];
        s11[a]=i[0]+i[1]+i[2]+i[3]+i[4]+i[5]+i[6]+i[7];
    } //end for
    for(a=0;a<a1-1;a++)
    {
        if(s11[a]<s11[a+1])
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

storeX=fx1[a];
storeY=fy1[a];
comXY[fy1[a+1]][fx1[a+1]]=255;
} //end if
if(s11[a]==s11[a+1])
{
storeX=fx1[a+1];
storeY=fy1[a+1];
comXY[fy1[a]][fx1[a]]=255;
}
if(s11[a]>s11[a+1])
{
storeX=fx1[a+1];
storeY=fy1[a+1];
comXY[fy1[a]][fx1[a]]=255;
}
} //end for
} //end if sum=255*6
///// case 4 /////
if(sum<=255*5)
{
Y2[c1]=cy;
X2[c1++]=cx;
comXY[cy][cx]=255;
flag=0;
} //
} //end while
} //end if sum=255*7
if(sum==255*6)
{
X1[c1]=x;
Y1[c1]=y;
n11[c1]=1;
comXY[y][x]=255;
a1=0;
for(a=0;a<=7;a++)
{
if(i[a]==0)
{
switch(a)
{
case 0:fy1[a1]=y-1;
fx1[a1++]=x+1;n11[c1]++;
break;
case 1:fy1[a1]=y-1;
fx1[a1++]=x;n11[c1]++;
break;
case 2:fy1[a1]=y-1;
fx1[a1++]=x-1;n11[c1]++;
break;
case 3:fy1[a1]=y;
fx1[a1++]=x-1;n11[c1]++;
break;
case 4:fy1[a1]=y+1;
fx1[a1++]=x-1;n11[c1]++;
break;
case 5:fy1[a1]=y+1;
fx1[a1++]=x;n11[c1]++;
break;
case 6:fy1[a1]=y+1;
fx1[a1++]=x+1;n11[c1]++;
break;
case 7:fy1[a1]=y;
fx1[a1++]=x+1;n11[c1]++;
break;
} //end switch
} //end if
} //end for

for(a=0;a<a1;a++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  cx=fx1[a];cy=fy1[a];
  i[0]=comXY[cy-1][cx+1];
  i[1]=comXY[cy-1][cx];
  i[2]=comXY[cy-1][cx-1];
  i[3]=comXY[cy][cx-1];
  i[4]=comXY[cy+1][cx-1];
  i[5]=comXY[cy+1][cx];
  i[6]=comXY[cy+1][cx+1];
  i[7]=comXY[cy][cx+1];
  s11[a]=i[0]+i[1]+i[2]+i[3]+i[4]+i[5]+i[6]+i[7];
} //end for
for(a=0;a<a1-1;a++)
{
  if(s11[a]<s11[a+1])
  {
    storeX=fx1[a];
    storeY=fy1[a];
    comXY[fy1[a+1]][fx1[a+1]]=255;
  } //end if
  if(s11[a]==s11[a+1])
  {
    storeX=fx1[a+1];
    storeY=fy1[a+1];
    comXY[fy1[a]][fx1[a]]=255;
  }
  if(s11[a]>s11[a+1])
  {
    storeX=fx1[a+1];
    storeY=fy1[a+1];
    comXY[fy1[a]][fx1[a]]=255;
  }
} //end for
while(flag==1)
{
  cx=storeX;cy=storeY;
  i[0]=comXY[cy-1][cx+1];
  i[1]=comXY[cy-1][cx];
  i[2]=comXY[cy-1][cx-1];
  i[3]=comXY[cy][cx-1];
  i[4]=comXY[cy+1][cx-1];
  i[5]=comXY[cy+1][cx];
  i[6]=comXY[cy+1][cx+1];
  i[7]=comXY[cy][cx+1];
  sum=i[0]+i[1]+i[2]+i[3]+i[4]+i[5]+i[6]+i[7];
  ///// case 1 /////
if(sum==255*8)
{
  comXY[cy][cx]=255;
  Y2[c1]=cy;
  X2[c1]=cx;
  c1++;
  flag=0;
} //end if sum 255*8
  ///// case 2 /////
if(sum==255*7)
{
  comXY[cy][cx]=255;
  for(a=0;a<=7;a++)
  {
    if(i[a]==0)
    {
      switch(a)
      {
        case 0: storeY=cy-1;
                storeX=cx+1;n11[c1]++;
                break;
        case 1: storeY=cy-1;
                storeX=cx;n11[c1]++;
                break;
        case 2: storeY=cy-1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        storeX=cx-1;n11[c1]++;
        break;
    case 3: storeY=cy;
        storeX=cx-1;n11[c1]++;
        break;
    case 4: storeY=cy+1;
        storeX=cx-1;n11[c1]++;
        break;
    case 5: storeY=cy+1;
        storeX=cx;n11[c1]++;
        break;
    case 6: storeY=cy+1;
        storeX=cx+1;n11[c1]++;
        break;
    case 7: storeY=cy;
        storeX=cx+1;n11[c1]++;
        break;
    }
} //end if
} //end for
} //end if sum=255*7
///// case 3 /////
if(sum==255*6)
{
    a1=0;
    comXY[cy][cx]=255;
    for(a=0;a<=7;a++)
    {
        if(i[a]==0)
        {
            switch(a)
            {
                case 0: fy1[a1]=cy-1;
                    fx1[a1++]=cx+1;n11[c1]++;
                    break;
                case 1: fy1[a1]=cy-1;
                    fx1[a1++]=cx;n11[c1]++;
                    break;
                case 2: fy1[a1]=cy-1;
                    fx1[a1++]=cx-1;n11[c1]++;
                    break;
                case 3: fy1[a1]=cy;
                    fx1[a1++]=cx-1;n11[c1]++;
                    break;
                case 4: fy1[a1]=cy+1;
                    fx1[a1++]=cx-1;n11[c1]++;
                    break;
                case 5: fy1[a1]=cy+1;
                    fx1[a1++]=cx;n11[c1]++;
                    break;
                case 6: fy1[a1]=cy+1;
                    fx1[a1++]=cx+1;n11[c1]++;
                    break;
                case 7: fy1[a1]=cy;
                    fx1[a1++]=cx+1;n11[c1]++;
                    break;
            }
        } //end if
    } //end for
    for(a=0;a<a1;a++)
    {
        cx=fx1[a];cy=fy1[a];
        i[0]=comXY[cy-1][cx+1];
        i[1]=comXY[cy-1][cx];
        i[2]=comXY[cy-1][cx-1];
        i[3]=comXY[cy][cx-1];
        i[4]=comXY[cy+1][cx-1];
        i[5]=comXY[cy+1][cx];
        i[6]=comXY[cy+1][cx+1];
        i[7]=comXY[cy][cx+1];
        s11[a]=i[0]+i[1]+i[2]+i[3]+i[4]+i[5]+i[6]+i[7];
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }//end for

    for(a=0;a<a1-1;a++)
    {
        if(s11[a]<s11[a+1])
        {
            storeX=fx1[a];
            storeY=fy1[a];
            comXY[fy1[a+1]][fx1[a+1]]=255;
        }//end if
        if(s11[a]==s11[a+1])
        {
            storeX=fx1[a+1];
            storeY=fy1[a+1];
            comXY[fy1[a]][fx1[a]]=255;
        }
        if(s11[a]>s11[a+1])
        {
            storeX=fx1[a+1];
            storeY=fy1[a+1];
            comXY[fy1[a]][fx1[a]]=255;
        }
    }
} //end for
} //end if sum=255*6
///// case 4 /////
if(sum<=255*5)
{
    Y2[c1]=cy;
    X2[c1++]=cx;
    comXY[cy][cx]=255;
    flag=0;
} //
} //end while

} //end if sum=255*6
} //end if comXY==0
} //end for
} //end for

if(c1==1)
{
    max=0;
}
maxn=0;
if(c1>1)
{
    for(c=0;c<c1-1;c++)
    {
        if(n11[c]<n11[c+1])
        {
            if(n11[c+1]>maxn)
            {
                maxn=n11[c+1];
                max=c+1;
            } //end if
        } //end if
        if(n11[c]>n11[c+1])
        {
            if(n11[c]>maxn)
            {
                maxn=n11[c];
                max=c;
            } //end if
        } //end if
    } //end for
} //end if

KY[0]=Y1[max];
KY[1]=Y2[max];
KX[0]=X1[max];
KX[1]=X2[max];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void CPicDlg::direction(unsigned char zoomimage2[256][256],int sampling[8][8])
{
    int max,p,q,zero,y,x,i,j;
    unsigned char image16[32][32];

    int mask1[3][3]={    {-1,0,1},
                        {-2,0,2},
                        {-1,0,1} };

    int mask2[3][3]={    {-1,-2,-1},
                        {0,0,0},
                        {1,2,1}  };

    int mask3[3][3]={    {0,-1,-2},
                        {1,0,-1},
                        {2,1,0}  };

    int mask4[3][3]={    {2,1,0},
                        {1,0,-1},
                        {0,-1,-2} };

    for(j=1;j<=8;j++)
    for(i=1;i<=8;i++)
    {
        p=0;q=0;
        for(y=(j-1)*32;y<=(j*32)-1;y++)
        {
            for(x=(i-1)*32;x<=(i*32)-1;x++)
            { image16[p][q]=zoomimage2[y][x];
              q++;
              if(q==32){q=0;}
            }
            p++;
        }

        max=0;
        zero=Convolution2(mask1,image16);if (zero>max){max=zero;sampling[j-1][i-1]=1;}
        zero=Convolution2(mask2,image16);if (zero>max){max=zero;sampling[j-1][i-1]=2;}
        zero=Convolution2(mask3,image16);if (zero>max){max=zero;sampling[j-1][i-1]=3;}
        zero=Convolution2(mask4,image16);if (zero>max){max=zero;sampling[j-1][i-1]=4;}

    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. J.H. Wegstein and J.F. Rafferty ,”Matching fingerprints by computer,” NBS Tech. Note 466, U.S. Government Printing Office, 1968
2. W.J. Hankley and J.T. Tou,”Automatic fingerprints interpretation and classification via contentual analysis and topological coding,” in Pictorial Pattern Recognition, G.C. Cheng R.S. Ledley, D.K. Pollack, and A.Rosenfeld, Eds. Washington, DC: Thompson, 1968, p. 411-456
3. B. Moayer and K.S. Fu, “A syntactic approach to fingerprint pattern recognition,” Pattern Recognition, vol. 7 , pp. 1-23 , 1975.
4. A. Grasselli , “On the automatic classification of fingerprint Some considerations of the linguistic interpretation of pictures,” in Nethodologies of Pattern Recognition, S. Qatanabe, Ed. New York Academic, 1969, pp. 253-273
5. B. Kruse, “ A parallel picture processing machjine,” IEE Trans. Comput., vol. C-22, pp. 1075-1087, Dec. 1973
- 6.