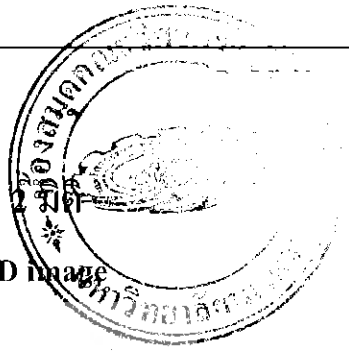


การแสดงโครงสร้างของรูป 2 มิติ

Visualizing the skeletonized 2D image

นายปริญญา วิชาวงษ์ รหัส 49361089

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 19 มิ.ย. 2555
เลขทะเบียน 1573 8064
เลขเรียกหนังสือ ว.
มหาวิทยาลัยนเรศวร ป/4581 7 2552

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน การแสดงโครงสร้างของรูป 2 มิติ

ผู้ดำเนินโครงงาน นายปริญญา วิชาวงษ์ รหัส 49361089

อาจารย์ที่ปรึกษา ดร. สุรเดช จิตประไพกุลศาล

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2552

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงาน

.....ประธานกรรมการ

(ดร. สุรเดช จิตประไพกุลศาล)

.....กรรมการ

(รองศาสตราจารย์ ดร. ไพศาล มุณีสว่าง)

.....กรรมการ

(ดร. วรลักษณ์ ดงเด่นฟ้า)

หัวข้อโครงงาน	**การแสดงโครงสร้างของรูป 2 มิติ**
ผู้ดำเนินโครงงาน	**นายปริญญา วิชาวงษ์**
อาจารย์ที่ปรึกษา	**ดร. สุรเดช จิตประไพกุลศาล**
สาขาวิชา	**วิศวกรรมคอมพิวเตอร์**
ภาควิชา	**วิศวกรรมไฟฟ้าและคอมพิวเตอร์**
ปีการศึกษา	**2552**

บทคัดย่อ

โครงงานนี้ใช้ความรู้เรื่องการประมวลผลภาพดิจิทัลเพื่อแสดงโครงสร้างของภาพ 2 มิติ ซึ่งผู้ศึกษาใช้ภาษาจาวาในการพัฒนาโปรแกรม โดยมีอัลกอริทึมที่ใช้หาแกนของภาพทั้งหมด 3 อัลกอริทึม ได้แก่ distance & transform, thinning และ Voronoi จากการทดสอบปรากฏว่าโปรแกรมสามารถหาแกนของภาพออกมาได้ทั้ง 3 อัลกอริทึม โดยแต่ละอัลกอริทึมนั้นจะมีข้อดีข้อเสียแตกต่างกันออกไป เช่น อัลกอริทึมแบบ thinning จะได้ภาพผลลัพธ์ที่มีคุณสมบัติ Topological แต่จะขาดคุณสมบัติ Geometrical ส่วนอัลกอริทึมแบบ distance & transform และ Voronoi จะได้ภาพผลลัพธ์ที่มีคุณสมบัติ Geometrical แต่จะขาดคุณสมบัติ Topological

Project Title	Visualizing the skeletonized 2D image
Name	Mr. Parinya Wichawong
Project Advisor	Dr. Suradet Jitprapaikulsarn
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic Tear	2009

Abstract

This project use digital image to determine the skeleton of 2D images. The program was develop in Java programming language. Three algorithm were use: 1) distance & transform 2) thinning 3) Voronoi From our observation three algorithm produce skeletons winth different properties. The skeletons from the thinning algorithm have Topological properties but lack Geometrical properties. The skeletons from the distance & transform algorithm have Geometrical properties but lack Topological properties. The skeletons from the Voronoi algorithm have Geometrical properties bot lack Topological properties.

กิตติกรรมประกาศ

โครงงานนี้มิอาจสำเร็จได้ด้วยดีหากขาดการสนับสนุน และคำแนะนำจากอาจารย์ที่ปรึกษาโครงงาน ขอขอบคุณอาจารย์ที่ปรึกษา ดร. สุรเดช จิตประไพกุลศาล ที่กรุณาให้คำปรึกษารวมถึงชี้แนะสาเหตุของปัญหาที่พบระหว่างทำโครงงาน ขอขอบคุณคณะกรรมการทุกท่านที่ให้ข้อคิดเห็นและคำแนะนำ และขอขอบคุณอาจารย์ทุกท่านที่ให้คำแนะนำต่างๆ จนทำให้โครงงานบรรลุผลสำเร็จตามเป้าหมาย

นอกเหนือจากบุคคลที่กล่าวมาข้างต้นแล้ว ข้าพเจ้าขอขอบคุณบุพการีที่ให้การสนับสนุนทางการศึกษา และกำลังใจที่มีให้มาโดยตลอด ข้าพเจ้าขอระลึกถึงพระคุณและกราบขอบพระคุณท่านมา ณ ที่นี้

ท้ายที่สุดข้าพเจ้าขอขอบคุณเพื่อนๆ ทุกคนที่ให้คำปรึกษา คำแนะนำ และเป็นกำลังใจให้กันเสมอมา ขออำนาจคุณพระศรีรัตนตรัยจงดลบันดาลให้ทุกท่านที่กล่าวมาประสบแต่ความสุขและมีสุขภาพพลานามัยแข็งแรงสมบูรณ์เทอญ

นายปริญญา วิชาวงษ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูปภาพ ..... ซ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของงาน 1

1.4 ขั้นตอนการดำเนินงาน 2

1.5 ระยะเวลาการดำเนินงาน 2

1.6 ผู้รับผิดชอบโครงการ 3

1.7 ผลที่คาดว่าจะได้รับ 3

1.8 งบประมาณในการดำเนินโครงการ 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 การประมวลผลภาพดิจิทัล 4

2.1.1 ภาพดิจิทัล 4

2.1.2 รูปร่างของภาพ 5

2.1.3 มาตรฐานของสี 6

2.2 Skeletonization 7

สารบัญ (ต่อ)

หน้า

2.2.1 คุณสมบัติของ skeleton 7

2.2.2 เทคนิค skeletonization 9

บทที่ 3 การออกแบบและพัฒนาระบบ 14

3.1 การจัดการกับรูปภาพก่อนการประมวลผล 14

3.1.1 การแปลงภาพสีเป็นโทนสีเทา 14

3.1.2 การแปลงภาพโทนสีเทาเป็นขาวดำ 15

3.1.3 การเก็บค่าของรูปภาพไว้ในรูปแบบอาเรย์ 2 มิติ 15

3.2 การประมวลผล 15

3.2.1 Distance & transform 16

3.2.2 Voronoi 22

3.2.2 Thinning 26

3.3 การจัดการกับรูปภาพหลังการประมวลผล 29

บทที่ 4 การออกแบบซอฟต์แวร์ 30

4.1 ความต้องการของระบบ 30

4.2 ขอบเขตของระบบ 30

4.3 การออกแบบซอฟต์แวร์ 31

4.3.1 ยูสเคลส ไดอะแกรม (Use case diagram) 32

4.3.2 ซีเควน ไดอะแกรม (Sequence diagram) 32

4.3.3 คอมโพเนนท์ ไดอะแกรม (Component Diagram) 34

4.3.4 แอคทีวิตี ไดอะแกรม (Activity diagram) 34

4.3.4 คลาส ไดอะแกรม (Class diagram) 36

บทที่ 5 ผลการทดสอบ 38

5.1 ทดสอบการทำงานแต่ละอัลกอริทึม 38

สารบัญ (ต่อ)

หน้า

5.1.1 Full distance & transform 39

5.1.2 Diagonal distance & transform 41

5.1.3 Cross distance & transform 43

5.1.4 Thinning 45

5.1.5 Voronoi 47

5.2 ทดสอบว่าสามารถแสดงผลแบบ GUI ได้ 51

5.2.1 การแสดงผลหน้าหลักของโปรแกรมเมื่อโปรแกรมเริ่มทำงาน 51

5.2.2 การเปิดไฟล์รูปภาพ 52

5.2.3 การบันทึกรูปภาพ 53

5.3 ทดสอบว่าสามารถประมวลผลบนระบบปฏิบัติการ Windows Xp ได้ 53

บทที่ 6 บทสรุป 54

6.1 สรุปผลการทดสอบ 54

6.2 ปัญหาและแนวทางแก้ไข 55

6.3 ข้อเสนอแนะ 55

เอกสารอ้างอิง 56

ภาคผนวก 57

ประวัติผู้เขียนโครงงาน 94

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงาน 2

2.1 ตารางเปรียบเทียบเทคนิค skeletonization แบบต่างๆ 9

5.1 ตารางผลการทดสอบอัลกอริทึม full distance & transform 39

5.2 ตารางผลการทดสอบอัลกอริทึม diagoanl distance & transform 41

5.3 ตารางผลการทดสอบอัลกอริทึม cross distance & transform 43

5.4 ตารางผลการทดสอบอัลกอริทึม thinning 45

5.5 ตารางผลการทดสอบอัลกอริทึม voronoi 47

5.6 ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม 49

5.7 เปรียบเทียบประสิทธิภาพแต่ละอัลกอริทึม 51

5.8 ทดสอบการทำงานบนระบบปฏิบัติการ Windows XP sp3 53

6.1 สรุปการทำงานแต่ละอัลกอริทึม 55

ก-1 ตารางผลการทดสอบอัลกอริทึม full distance & transform 60

ก-2 ตารางผลการทดสอบอัลกอริทึม diagoanl distance & transform 67

ก-3 ตารางผลการทดสอบอัลกอริทึม cross distance & transform 74

ก-4 ตารางผลการทดสอบอัลกอริทึม thinning 81

ก-5 ตารางผลการทดสอบอัลกอริทึม voronoi 88

ข-1 ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม 95

สารบัญรูปภาพ

รูปที่ หน้า

2.1 แบบจำลองสี RGB 7

2.2 จุดที่เป็น-ไม่เป็น ส่วนหนึ่งของ skeleton 8

2.3 ภาพไม่เหมือนกันแต่ได้ skeleton เหมือนกัน 8

2.4 skeleton ที่ได้จากสี่เหลี่ยมที่พื้นผิวไม่สม่ำเสมอ 8

2.5 การใช้จุดที่อยู่ด้านข้างจุดนั้นๆ สี่ตัวเป็นตัวกำหนด distance map 10

2.6 distance map ซึ่งใช้จุดโดยรอบตัวมันทั้งแปดตัวเป็นตัวกำหนด 10

2.7 จุดต้นกำเนิดและ Voronoi diagram ของจุดต้นกำเนิด 11

2.8 Volonoi diagram ของภาพที่มีจำนวนจุดต้นกำเนิดที่แตกต่างกัน 11

2.9 skeleton ที่เกิดจากเส้นย่อยๆ ของ Volonoi diagram 12

2.10 skeleton ที่ได้จากการทำ thinning 12

3.1 ตัวอย่างก่อน-หลังการแปลง color เป็น grayscale 14

3.2 ตัวอย่างก่อน-หลังการแปลงภาพ grayscale เป็น binary 15

3.3 การแทนค่าจุดภาพด้วยอาเรย์ 2 มิติ 15

3.4 แผนผังการทำงานขั้นตอนการกำหนดค่าเริ่มต้นให้ Integer Array 17

3.5 รูปแบบของตำแหน่งรอบข้างที่ใช้ในการพิจารณา 18

3.6 แผนผังการทำงานขั้นตอนการเปลี่ยนแปลงค่าตามระยะทาง 19

3.7 แผนผังการทำงานขั้นตอนการกำหนดจุดที่เป็นส่วนหนึ่งของ skeleton 21

3.8 โครงสร้างของแต่ละตำแหน่งในอาเรย์ (Voronoi) 22

3.9 แผนผังการทำงานการกำหนดค่าเริ่มต้นให้อาเรย์ (Voronoi) 23

3.10 ขั้นตอนการกำหนดค่าระยะทางด้านซ้ายของอาเรย์ 24

3.11 อาเรย์ที่กำหนดค่าระยะทางแต่ละตำแหน่งแล้ว 25

3.12 ตำแหน่งที่มีค่า count >= 2 25

สารบัญรูปภาพ (ต่อ)

รูปที่ หน้า

3.13 ตำแหน่งที่มีตำแหน่งข้างเคียงที่มีระยะทางของฝั่งตรงกันข้ามเท่ากัน 26

3.14 เงื่อนไข B(p1) ที่ไม่สอดคล้องกับกฎ 2 <= B(p1) <= 6 27

3.15 เงื่อนไข A(p1) ที่ไม่สอดคล้องกับกฎ A(p1) = 1 27

3.16 A(p2) = 2 และ p2.p4.p8 = 0 28

3.17 ตำแหน่งในแนวนอนที่มีความหนา 2 พิกเซล 28

3.18 A(p2) = 2 และ p2.p4.p8 = 0 28

3.19 ตำแหน่งในแนวตั้งที่มีความหนา 2 พิกเซล 29

4.1 Context Diagram ของระบบ 30

4.2 มุมมองในการออกแบบซอฟต์แวร์ 31

4.3 Use case diagram 32

4.4 Sequence diagram 33

4.5 Component diagram 34

4.6 Activity diagram 35

4.7 Class diagram 36

5.1 GUI หลักของโปรแกรม 52

5.2 GUI ส่วนของการเปิดไฟล์รูปภาพ 52

5.3 GUI หลักส่วนของการบันทึกไฟล์รูปภาพ 53

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันความรู้เรื่องการประมวลผลภาพ (Image Processing) มีการนำมาใช้ประโยชน์อย่างแพร่หลาย เช่น การวิเคราะห์ความสมบูรณ์ของเม็ดเลือดแดง การตีความจากรูปภาพสิ่งพิมพ์เป็นตัวหนังสือเป็นต้น ซึ่งเป็นประโยชน์ในหลายๆ ด้านทั้งด้านการแพทย์ การคมนาคม การทหาร ฯลฯ

การแสดงโครงสร้าง (skeleton) ของภาพถือเป็นศาสตร์อย่างหนึ่งของการประมวลผลภาพ ซึ่งสามารถนำไปใช้ประโยชน์ในงานต่างๆ ที่ต้องการวิเคราะห์โครงสร้างของวัตถุจากภาพ เช่น การวิเคราะห์รูปตึกสูงๆ เพื่อหาจุดศูนย์ถ่วงของตึก หรืออาจนำไปช่วยในการวินิจฉัย หรือช่วยในการตัดสินใจของแพทย์ เช่น ช่วยให้แพทย์สามารถดูลักษณะการขดตัวของลำไส้ได้ชัดเจนยิ่งขึ้น

แต่การแสดงโครงสร้างของภาพ 2 มิตินั้น สามารถทำได้หลายวิธี ซึ่งในแต่ละวิธีก็มี ข้อดี ข้อเสีย และความเหมาะสมกับลักษณะของงานที่แตกต่างกัน เช่น งานที่ต้องการแกนที่มีความต่อเนื่องของแกนเป็นหลัก หรือแกนที่รักษาโครงสร้างทางเรขาคณิตของวัตถุไว้เสมอ ด้วยเหตุนี้ผู้จัดทำจึงมีแนวคิดที่จะศึกษา และพัฒนาการแสดงโครงสร้างของภาพ 2 มิติหลายๆ วิธีการ และวิเคราะห์ความแตกต่าง รวมไปถึงข้อดีข้อเสียของของแต่ละวิธีการเพื่อเป็นประโยชน์ในการนำไปใช้ประโยชน์ต่อไป

1.2 วัตถุประสงค์

1.2.1 เพื่อพัฒนา software ที่สามารถคำนวณหา skeleton จากภาพสองมิติได้

1.3 ขอบเขตของงาน

โปรแกรมที่พัฒนาขึ้นจะสามารถทำงานได้ดังนี้

1.3.1 สามารถประมวลผลได้บนระบบปฏิบัติงาน Windows Xp ได้

1.3.2 สามารถแสดงผลแบบ GUI ได้

1.3.3 สามารถแปลงจากรูปภาพปกติไปเป็นแกนของรูปได้

1.4 ขั้นตอนการดำเนินงาน

1.4.1 กำหนดขอบเขตของงาน

1.4.2 วางแผนการทำงาน

1.4.3 สืบค้นข้อมูลและทำความเข้าใจในเนื้อหาที่จำเป็นในการทำโครงงาน

1.4.4 ออกแบบการทำงานของ software

1.4.5 พัฒนาส่วนของ software

1.4.6 ทดสอบและปรับปรุงแก้ไขการทำงานของ software

1.4.7 จัดทำเอกสารและคู่มือการใช้งาน

1.4.8 ส่งโครงงานฉบับสมบูรณ์

1.5 ระยะเวลาการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	สัปดาห์ – เดือน – ปี																	
	มิ.ย. 2552		ก.ค. 2552		ส.ค. 2552.		ก.ย. 2552		ต.ค. 2552		พ.ย. 2552		ธ.ค. 2552		ม.ค. 2552		ก.พ. 2552	
	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
1. กำหนดขอบเขตของงาน																		
2. วางแผนการทำงาน																		
3. ข้อมูลและทำความเข้าใจในเนื้อหาที่จำเป็นในการทำโครงงาน																		
4. ออกแบบการทำงานของ software																		
5. พัฒนาส่วนของ software																		

ตารางที่ 1.1 (ต่อ) แผนการดำเนินงาน

6. ทดสอบและปรับปรุงแก้ไขการทำงานของ software	
7. จัดทำเอกสารและคู่มือการใช้งาน	
8. ส่งโครงงานฉบับสมบูรณ์	

1.6 ผู้รับผิดชอบโครงการ

นายปริญญา วิชาวงษ์

1.7 ผลที่คาดว่าจะได้รับ

1.7.1 มีความรู้ความเข้าใจในเรื่องการประมวลผลภาพ

1.7.2 มีความรู้ความเข้าใจในเรื่องการหาแกนของภาพ

1.7.3 สามารถผลิต software ที่สามารถใช้ประโยชน์ได้จริง

1.7.4 สามารถนำความรู้และประสบการณ์จากการทำโครงงานไปประยุกต์ใช้กับงานอื่นๆ ได้

1.8 งบประมาณในการดำเนินโครงการ

1. เอกสารและตำรา	500	บาท
2. ค่าจัดทำรูปเล่มโครงงาน	600	บาท
3. ค่าใช้จ่ายอื่น ๆ	400	บาท
รวมทั้งหมด	1,500	บาท

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 การประมวลผลภาพดิจิทัล (Digital Image Processing)

ความพยายามทางด้านการประมวลผลภาพได้เริ่มขึ้นในปี 1964 ณ ห้องแลป Jet Propulsion (Pasasena California) ซึ่งได้นำการบวนการการประมวลผลภาพมาใช้ในการพิจารณาภาพถ่ายดาวเทียมของดวงจันทร์ ต่อมาได้มีการตั้งสาขาทางวิทยาศาสตร์สาขาใหม่มีชื่อว่า Digital image processing หลังจากนั้นงานทางด้านการประมวลผลภาพก็พัฒนาขึ้นเรื่อยๆ และใช้กันอย่างกว้างขวางสำหรับงานหลายๆ ด้านตัวอย่าง เช่น การสื่อสารทางโทรทัศน์ การพิมพ์ กราฟฟิก การแพทย์ และการค้นคว้าทางวิทยาศาสตร์

Digital image processing จะเกี่ยวกับการแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิทัล (Digital format) ซึ่งสามารถที่จะนำเอาข้อมูลนี้จัดผ่านกระบวนการต่างๆ ด้วยคอมพิวเตอร์ได้ในระบบของดิจิทัล อินพุตและเอาพุตของระบบจะอยู่ในรูปแบบดิจิทัลเท่านั้น

2.1.1 ภาพดิจิทัล (Digital Image) [1]

ภาพดิจิทัลเป็นการแสดงผลภาพในลักษณะสองมิติในหน่วยที่เรียกว่าพิกเซล ซึ่งสามารถนิยามเป็นฟังก์ชันสองมิติ f(x,y) โดยที่ x และ y เป็นพิกัดของภาพ และแอมพลิจูดของ f ที่พิกัด (x,y) ใดๆ ภายในภาพคือค่าความเข้มแสงของภาพ (Intensity) ที่ตำแหน่งนั้นๆ และเมื่อ x,y และแอมพลิจูดของ f เป็นค่าจำกัด (Finite value) จึงเรียกรูปภาพนี้ว่าเป็นภาพดิจิทัล (Digital Image) และถ้ากำหนดให้ภาพ f(x,y) มีขนาด M แถวและ N คอลัมน์ และพิกัดของจุดกำเนิด (Origin) ของภาพคือที่ตำแหน่ง (x,y) = (0,0) แล้ว จะสามารถเขียนสมการให้อยู่ในรูปเมทริกซ์ได้ดังนี้

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (1.1)$$

ค่าแต่ละค่าที่อยู่ในเมทริกซ์จะเรียกว่าพิกเซล (Pixel) โดยตำแหน่ง (0,0) จะอยู่ทางด้านซ้ายมือสุดด้านบนของภาพ การจัดลำดับตำแหน่งของจุดภาพจะเรียงจากซ้ายไปขวาในแต่ละเส้นจุด และจัดลำดับของเส้นจุดจะเรียงจากบนลงล่าง การเก็บค่าของความเข้มแสงของภาพดิจิทัลลงหน่วยความจำในลักษณะเส้นจุด (raster) นี้จะเรียกภาพบิตแมป (bit-maped image) หรือภาพแรสเตอร์ (raster image)

การสร้างภาพดิจิทัลสามารถสร้างได้จากอุปกรณ์รับภาพเช่น กล้องดิจิทัล (digital cameras) หรือ เครื่องกราดภาพ (scanners) เป็นต้น ภาพดิจิทัลยังสามารถสร้างโดยการสังเคราะห์จากสิ่งที่ไม่ใช่ข้อมูลภาพ เช่น ฟังก์ชันทางคณิตศาสตร์ หรือแบบจำลองเรขาคณิตแบบสามมิติซึ่งการสร้างภาพลักษณะนี้เป็นส่วนหนึ่งในงานด้านคอมพิวเตอร์กราฟิกส์ (computer graphics)

2.1.2 รูปร่างของภาพ (Image Shape) [2]

วัตถุที่มีอยู่ตามธรรมชาติและที่มนุษย์สร้างขึ้นมีรูปร่างมากมายแตกต่างกันไป ในศาสตร์ของการประมวลผลภาพนั้นการกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เป็นวิธีที่นิยมใช้กันมากที่สุดเนื่องจากทำให้การอ่านภาพการจัดเก็บข้อมูลภาพในหน่วยความจำและการแสดงภาพออกทางอุปกรณ์ต่างๆ เป็นไปได้อย่างมีประสิทธิภาพ

การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอาเรย์ (array) โดยค่าในแต่ละช่องของอาเรย์แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่องอาเรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ (ตามหัวข้อที่ 2.1.1)

สมมุติให้ Image เป็นตัวแปรแบบอาเรย์ขนาด M x N (M แถว และ N คอลัมน์) ที่ใช้เก็บภาพขนาด M x N จุด (M จุดในแนวนอน และ N จุดในแนวตั้ง) ค่าสี (หรือความสว่างในกรณีที่เป็นภาพ grey level) ของจุดภาพในแถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ Image(5,4) จะเห็นว่าเราใช้ตำแหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอาเรย์

จากการใช้หน่วยความจำเพื่อการเก็บภาพในลักษณะที่กล่าวมา เนื้อที่ในการเก็บภาพสามารถคำนวณได้จาก M x N x g เมื่อ g เป็นจำนวนเต็มที่แทนจำนวนบิตของข้อมูลในแต่ละจุดภาพ เช่น ถ้า g มีค่าเท่ากับ 8 บิตเราจะสามารถเก็บความแตกต่างของระดับสีที่เป็นไปสูงสุด 256 ระดับ ค่า M และ N จะเป็นตัวบอกถึงความละเอียดของภาพ สำหรับคอมพิวเตอร์ทั่วไปในระบบ VGA (Video Graphic Array) จะมีขนาด 640 x 480, 800 x 600 และ 1024 x 768 จุด เป็นต้น การกำหนดความละเอียดจะขึ้นอยู่กับงานที่จะใช้ ในงานบางอย่างใช้ความละเอียดแค่ 30 x 50 จุด ก็พอแล้วแต่ในงานบางชนิด ใช้ความละเอียดถึง 1000 x 1000 จุด ก็ยังไม่พอ

ปกติแล้วในการเก็บข้อมูลภาพโดยเครื่องมือต่างๆ จะเก็บตามมาตรฐานของโทรทัศน์ซึ่งมีอัตราส่วน x ต่อ y เท่ากับ 4:3 สำหรับเครื่องมือเก็บข้อมูลภาพที่ไม่เป็นไปตามอัตราส่วน 4:3 เมื่อนำภาพนี้ไปแสดงในจอภาพมาตรฐานจะทำให้ภาพที่แสดงนั้นมีขนาดของจุดภาพไม่เป็นสี่เหลี่ยมจัตุรัสเช่นในบางระบบอาจจะใช้ความละเอียดในการแสดงเท่ากับ 640 x 512 ซึ่งจะทำให้ขนาดของจุดภาพที่ได้มีขนาดของด้านกว้างมีความยาวมากกว่าด้านสูง ซึ่งลักษณะดังกล่าวนี้เป็นหัวข้อที่ต้องสนใจสำหรับการเขียนโปรแกรมทางด้านกราฟฟิกและการจัดการข้อมูล

จำนวนสีสูงสุดที่เป็นไปได้ของแต่ละจุดภาพขึ้นอยู่กับจำนวนบิตที่ใช้ เมื่อมีการกำหนดให้ขนาดของบิตต่อจุดมากขึ้นจะทำให้จำนวนของสีมากขึ้นด้วย เช่น

1 บิต = 2^1 = 2 สี

2 บิต = 2^2 = 4 สี

4 บิต = 2^4 = 16 สี

8 บิต = 2^8 = 256 สี

16 บิต = 2^{16} = 65536 สี เป็นต้น

สำหรับการแสดงข้อมูลภาพที่มีขนาด 1 บิตและ 8 บิตนั้นจะมีการทำงานที่จะใกล้เคียงกัน เนื่องจากหน่วยประมวลผลจะไม่สามารถจัดการกับข้อมูลที่เป็นบิตเดี่ยวๆ ได้ดังนั้นในการแสดงข้อมูลออกทางจอภาพตัวโปรเซสเซอร์จะทำการก็อปปี้ข้อมูลทั้ง 8 บิต (1 Byte) ส่งให้กับจอภาพซึ่งในกรณีที่ Pixel มีขนาด 1 บิต เมื่อโปรเซสเซอร์จะทำงานกับบิตแรกที่ต้องการแล้วก็จะทำการคัดลอก (copy) ข้อมูลชุดใหม่ทันทีโดยที่ไม่เกี่ยวกับข้อมูลอีก 7 บิตที่เหลือส่วนในกรณี Pixel ที่มีขนาด 8 บิต โปรเซสเซอร์จะทำการคัดลอกข้อมูลจุดใหม่ก็ต่อเมื่อโปรเซสเซอร์ทำงานกับทุกบิตแล้ว

ตัวอย่างสำหรับระบบที่มีความละเอียดเท่ากับ 800 x 600 และมีขนาด 16 บิตต่อ Pixel จะสามารถแสดงสีได้ทั้งหมด 65536 ระดับและต้องใช้เนื้อที่ในการเก็บเท่ากับ 800 x 600 x 16 บิต

2.1.3 มาตรฐานของสี

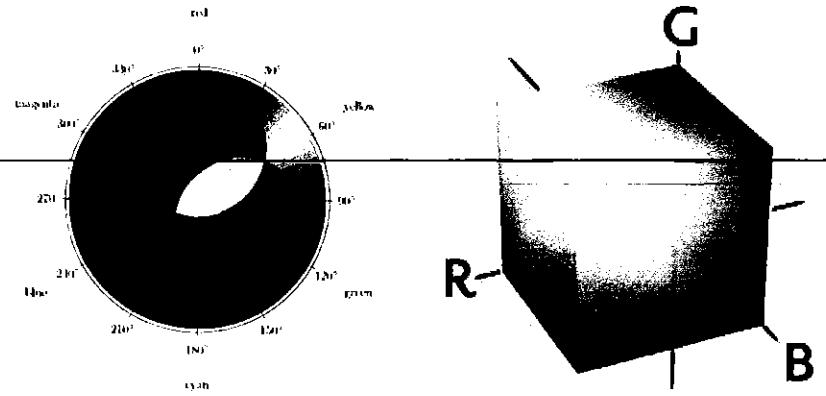
มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกัน ทั้งนี้จะขึ้นอยู่กับการนำไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน เช่น ในระบบ RGB จะมีแกนสีคือ แกนสีแดง, เขียว, และน้ำเงินในระบบ HLS จะมีแกนเป็น ค่าสี (hue), ความสว่าง (lightness), และความบริสุทธิ์ของสี (saturation) ตัวอย่างระบบสีที่นิยมใช้กัน ได้แก่ ระบบ RGB, HSV (Hue Saturation Value) และ HLS (Hue Lightness Saturation)

ในส่วนนี้ผู้จัดทำจะสนใจเฉพาะมาตรฐานสีแบบ RGB เท่านั้น

2.1.3.1 ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง (R) เขียว (G) และน้ำเงิน (B) โดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ในการใช้งาน

โดยปกติแล้วคอมพิวเตอร์จะเก็บข้อมูล 1 ไบต์ต่อ 1 สีทำให้แต่ละค่าของสีนั้นมีได้ตั้งแต่ 0 ถึง 256 สีดังนี้ ($0 \leq R < 255$), ($0 \leq G < 255$), และ ($0 \leq B < 255$) เมื่อนำทั้งสามสีมาผสมกันก็จะได้สีทั้งหมด 16 ล้านสี



รูปที่ 2.1 แบบจำลองสี RGB

อ้างอิง http://media.texample.net/tikz/examples/PNG/rgb-color-mixing.png

http://learn.colorotate.org/media.colorotate.org/c/rgb.jpg

2.2 Skeletonization [3]

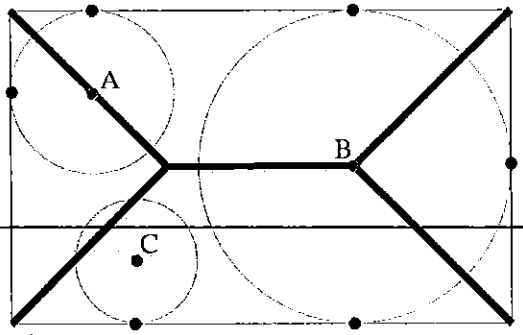
จุดมุ่งหมายของ skeletonization คือ การหารูปทรงซึ่งมีลักษณะเด่นเพื่อจะใช้อ้างแทนรูปร่าง หรือรูปทรงของวัตถุทั่วๆ ไป

2.2.1 คุณสมบัติของ skeleton

ความคิดเรื่อง skeleton ถูกกล่าวถึงโดย H.Blum เช่น ผลของ Medial Axis Transform (MAT คือ การเปลี่ยนรูปมาเป็นแกนกลาง) หรือSymmetry Axis Transform (SAT คือ การเปลี่ยนรูปมาเป็นแกนซึ่งได้สัดส่วนที่ตรงกัน) MAT จะหาบริเวณที่ใกล้เคียงกับจุดกลางมากที่สุดสำหรับทุกจุดที่อยู่ในวัตถุนั้นๆ

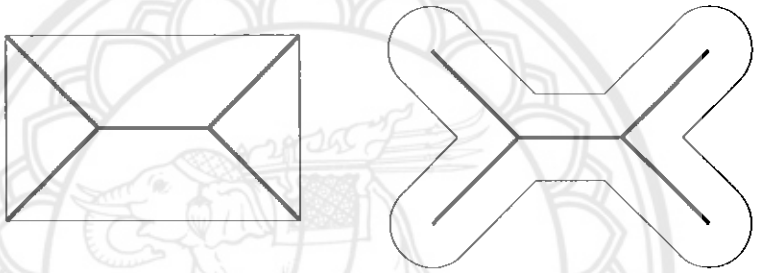
คำนิยามของ Skeleton

- Skeleton เป็นวิถีของจุดในเรขาคณิตซึ่งจะเคลื่อนไปตามสมการที่กำหนดให้
- Skeleton เป็นตำแหน่งกึ่งกลางรูปทรงกลม เช่น ได้จุดหนึ่งจุดสำหรับวัตถุรูปจาน



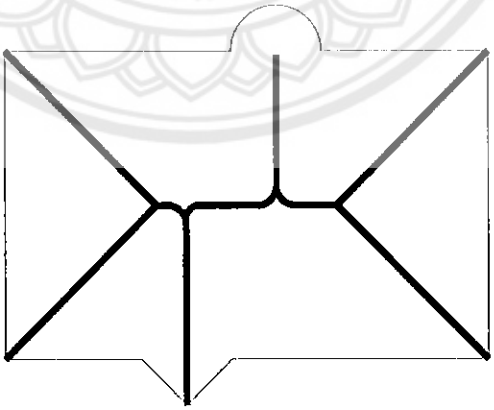
รูปที่ 2.2 จุดที่เป็น-ไม่เป็น ส่วนหนึ่งของ skeleton

อ้างอิง http://www.inf.u-szeged.hu/~palagyi/skel/skel.html



รูปที่ 2.3 ภาพไม่เหมือนกันแต่ได้ skeleton เหมือนกัน

อ้างอิง http://www.inf.u-szeged.hu/~palagyi/skel/skel.html



รูปที่ 2.4 skeleton ที่ได้จากสี่เหลี่ยมที่พื้นผิวไม่สม่ำเสมอ

อ้างอิง http://www.inf.u-szeged.hu/~palagyi/skel/skel.html

2.2.2 เทคนิค skeletonization

สามเทคนิคใหญ่ๆของ skeletonization คือ

- Distance & Transform
- Voronoi
- Thinning

การทำ skeletonization ที่มีความถูกต้องที่สุดนั้นจะต้องสอดคล้องกับคุณสมบัติ ดังนี้

- Topological คือคุณสมบัติทางรูปร่างที่ไม่แปรเปลี่ยนภายใต้การดึง ยืด หด หรือบีบ โดยไม่มีการฉีก การเจาะ หรือการเชื่อมติดใหม่
- Geometrical คือการบีบอัด skeleton ให้อยู่ในใจกลางของวัตถุ และแตกออกไปตามความสำคัญในเรื่องการแปลงรูปตามวิชาเรขาคณิต

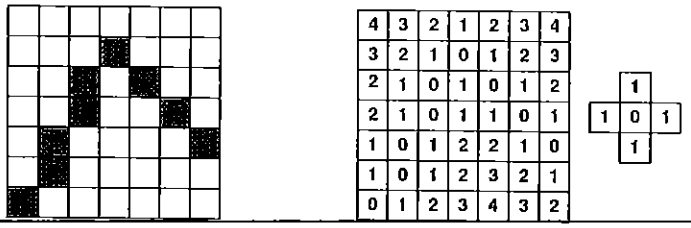
ตารางที่ 2.1 ตารางเปรียบเทียบเทคนิค skeletonization แบบต่างๆ

Method	Geometrical	Topological
Distance & transform	Yes	no
Voronoi-skeleton	Yes	yes
Thinning	No	yes

2.2.2.1 การแปลงรูปตามระยะทาง (Distance & transformation)

Skeletonization อยู่บนพื้นฐานของการเปลี่ยนรูปตามระยะทาง ซึ่งจำเป็นต้องมีขั้นตอนและกระบวนการ 3 ข้อดังนี้

- แปลงภาพต้นฉบับเป็นแบบไบนารี ให้มันมีส่วนที่นูนและไม่นูน ส่วนที่นูนจะเป็นส่วนเขตแดนของวัตถุ
- Distance map จะสร้างและกำหนดค่าแต่ละส่วนใน map ตามระยะทางจนกระทั่งได้ส่วนที่นูนที่สุด
- สันแนวที่ตรวจพบนั้นจะเป็นจุดของ skeleton



4	3	2	1	2	3	4
3	2	1	0	1	2	3
2	1	0	1	0	1	2
2	1	0	1	1	0	1
1	0	1	2	2	1	0
1	0	1	2	3	2	1
0	1	2	3	4	3	2

	1	
1	0	1
	1	

รูปที่ 2.5 การใช้จุดที่อยู่ด้านข้างจุดนั้นๆ สี่ตัวเป็นตัวกำหนด distance map

อ้างอิง http://www.inf.u-szeged.hu/~palagyi/skel/skel.html

2	2	1	1	1	2	2
2	1	1	0	1	1	2
2	1	0	1	0	1	1
1	1	0	1	1	0	1
1	0	1	1	1	1	0
1	0	1	2	2	1	1
0	1	1	2	2	2	2

1	1	1
1	0	1
1	1	1

8	7	4	3	4	7	8
7	4	3	0	3	4	7
6	3	0	3	0	3	4
4	3	0	3	3	0	3
3	0	3	4	4	3	0
3	0	3	6	7	4	3
0	3	6	7	8	7	6

4	3	4
3	0	3
4	3	4

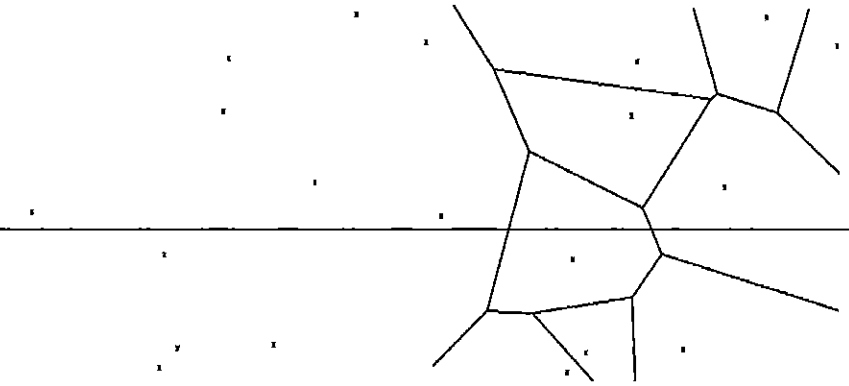
รูปที่ 2.6 distance map ซึ่งใช้จุดโดยรอบตัวมันทั้งแปดตัวเป็นตัวกำหนด

อ้างอิง http://www.inf.u-szeged.hu/~palagyi/skel/skel.html

Distance & Transformation สามารถกระทำได้โดยใช้ระยะเวลา O(n) ตามขนาดของภาพ (เมื่อ n เป็นจำนวนหน่วยย่อยๆ ของภาพ) วิธีการนี้จะทำตามข้อกำหนดของหลัก geometrical แต่จะไม่รับประกันได้ว่าจะถูกต้องตามแบบแผนของ topological หรือไม่

2.2.2.2 Voronoi diagram

Voronoi diagram นั้นทำโดยการแบ่งจุดออกเป็นส่วนๆ และแบ่งช่องระหว่างจุด โดยทุกๆ ช่องควรจะมีแนวของเส้นแบ่งระหว่างจุด แล้วส่วนของแนวที่มาบรรจบกันนั้นจะเป็นส่วนหนึ่งของ skeleton



จุดต้นกำเนิด voronoi diagram

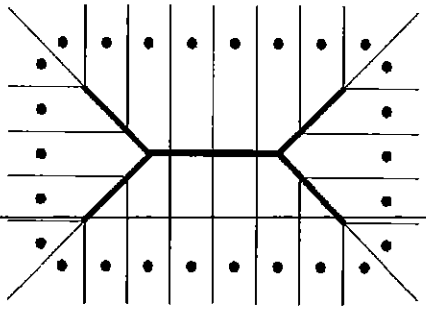
รูปที่ 2.7 จุดต้นกำเนิด และ Voronoi diagram ของจุดต้นกำเนิด

อ้างอิง http://www.inf.u-szeged.hu/~palagyi/skel/skel.html

รูปที่ 2.8 Volonoi diagram ของภาพที่มีจำนวนจุดต้นกำเนิดที่แตกต่างกัน

อ้างอิง http://www.inf.u-szeged.hu/~palagyi/skel/skel.html

ถ้าหากความหนาแน่นของจุดแบ่งเขตแดนมีค่ามากๆ (จุดต้นกำเนิดมีมากๆ) แล้วส่วนของ volonoi diagram ที่มาบรจบกันนั้นคือ เส้นแกน (skeleton)



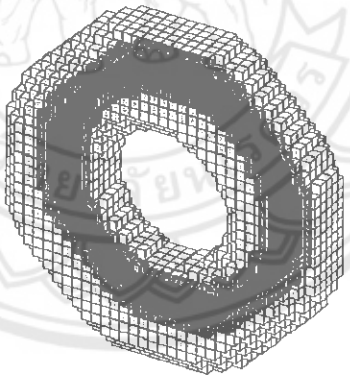
รูปที่ 2.9 skeleton ที่เกิดจากเส้นย่อยๆ ของ Volonoi diagram

อ้างอิง http://www.inf.u-szeged.hu/~palagyi/skel/skel.html

volonoi diagram จะคลอบคลุมทั้ง topological และ geometrical แต่มันก็จะสิ้นเปลือง กระบวนการโดยเฉพาะอย่างยิ่งกับวัตถุที่มีขนาดใหญ่และมีความซับซ้อนมาก

2.2.2.3 Thinning

เป็นเทคนิคการลดความหนาของวัตถุสำหรับสร้างแบบจำลองของวัตถุนั้นๆ ขึ้นมา



รูปที่ 2.10 skeleton ที่ได้จากการทำ thinning

อ้างอิง http://www.inf.u-szeged.hu/~palagyi/skel/skel.html

Thinning มีคุณสมบัติข้อดีดังนี้

- มันจะรักษาโครงสร้างเดิมไว้ (Topology)
- มันจะรักษารูปทรงเอาไว้
- มันจะบีบอัดจนได้ skeleton อยู่ใจกลางของวัตถุ
- ความกว้างของ skeleton นั้นจะเท่ากับ 1pixel เท่านั้น

บทที่ 3

การออกแบบและพัฒนาระบบ

บทนี้จะกล่าวถึงวิธีในการออกแบบ และพัฒนาระบบสำหรับโครงงานนี้ เพื่อความสะดวกในการทำความเข้าใจผู้พัฒนาจึงแบ่งระบบออกเป็น 2 ส่วนหลักๆ ดังนี้

- การจัดการกับรูปภาพก่อนการประมวลผล
- การประมวลผลภาพ
- การจัดการกับรูปภาพหลังการประมวลผล

3.1 การจัดการกับรูปภาพก่อนการประมวลผล

เนื่องจากการประมวลผลภาพเพื่อหาแกนของภาพ (skeletonization) นั้นภาพที่ใช้ในการประมวลผลจะต้องเป็นแบบไบนารี (binary) เท่านั้น แต่ภาพที่รับเข้ามานั้นอาจเป็นภาพสี ดังนั้นก่อนขั้นตอนการประมวลผลเราจึงจำเป็นต้องมีกระบวนการในการแปลงภาพสีไปเป็นภาพแบบขาวดำ และเก็บข้อมูลของภาพให้อยู่ในรูปแบบอาเรย์ 2 มิติตามขั้นตอนต่อไปนี้

3.1.1 การแปลงภาพสีเป็นโทนสีเทา

เพื่อความสะดวกในการประมวลผลเพื่อตรวจจับวัตถุในภาพจึงทำการแปลงภาพจากภาพสีเป็นภาพโทนสีเทา

รูปที่ 3.1 ตัวอย่างก่อน-หลังการแปลง color เป็น grayscale

3.2.1 Distance & transform [4]

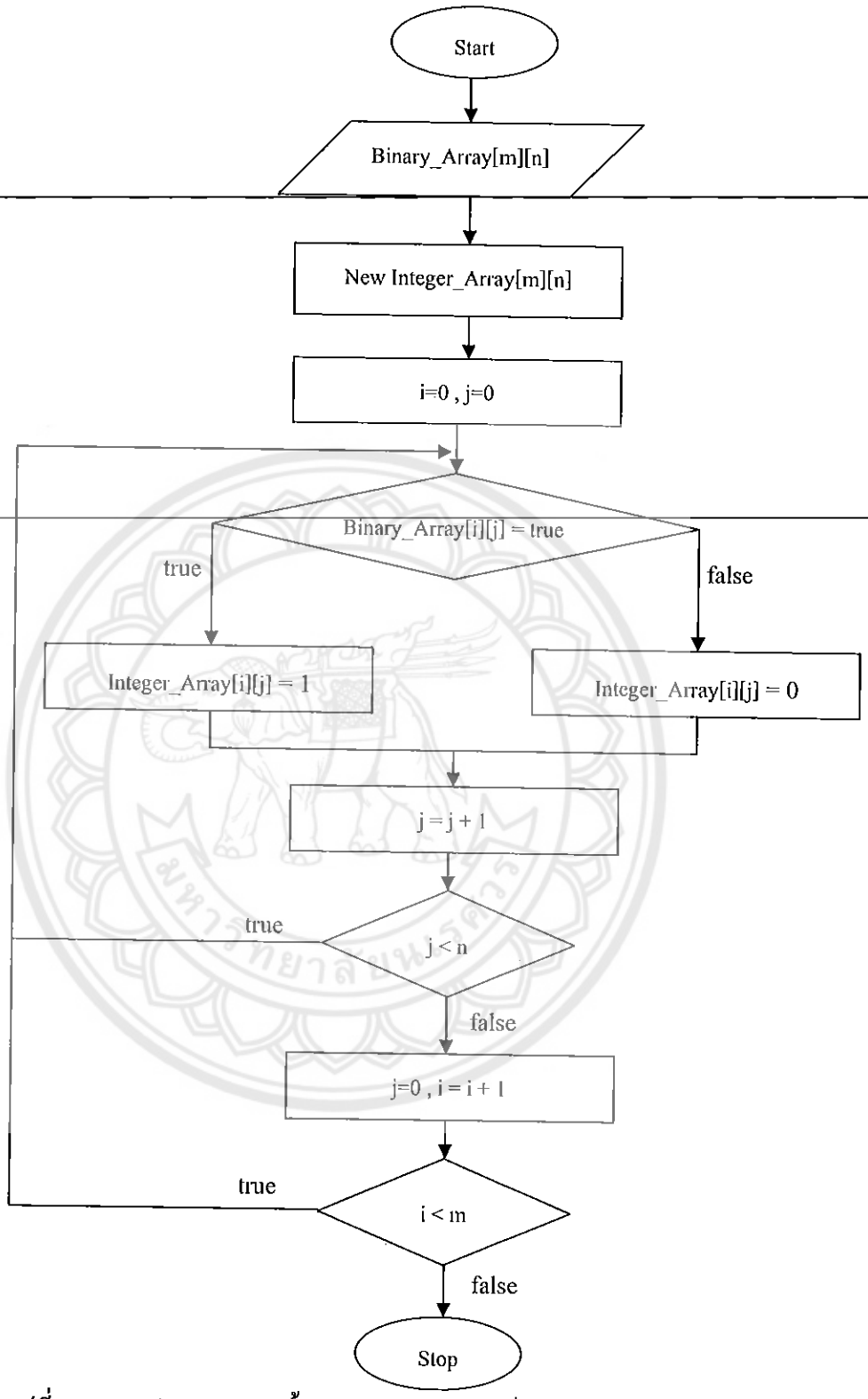
ในการประมวลผลแบบ Distance & Transform นั้นมี 3 ขั้นตอนหลักๆ ดังนี้

- กำหนดค่าเริ่มต้นให้อาเรย์ 2 มิติซึ่งเก็บค่าจำนวนเต็ม
- เปลี่ยนแปลงค่าแต่ละตำแหน่งของอาเรย์
- หาจุดที่เป็นส่วนหนึ่งของ skeleton

3.2.1.1 กำหนดค่าเริ่มต้นให้อาเรย์ 2 มิติซึ่งเก็บค่าจำนวนเต็ม

การประมวลผลด้วยวิธี Distance & Transform นั้นมีความจำเป็นต้องใช้โครงสร้างแบบอาเรย์ 2 มิติเก็บข้อมูลเป็นจำนวนเต็มเพื่อที่จะสามารถกำหนดค่าระยะทางของแต่ละจุดในอาเรย์ซึ่งมีขั้นตอนการทำดังนี้

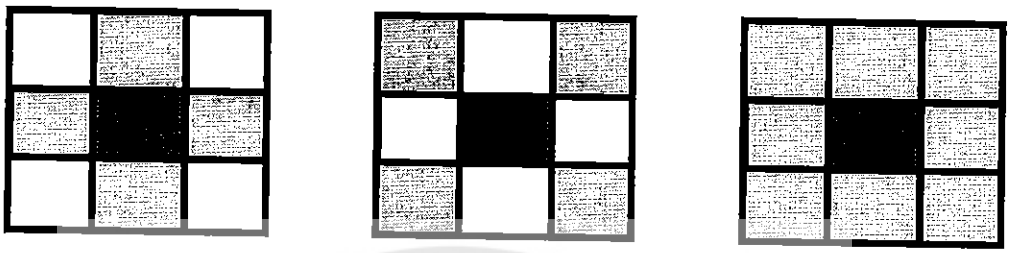
- สร้าง Integer Array ที่เก็บจำนวนเต็มซึ่งมีขนาดเท่ากับ Boolean Array เพื่อเก็บข้อมูลของรูปภาพ
- ถ้า Boolean Array[i][j] = true ให้ Integer Array[i][j] = 1
- ถ้า Boolean Array[i][j] = false ให้ Integer Array[i][j] = 0



รูปที่ 3.4 แผนผังการทำงานขั้นตอนการกำหนดค่าเริ่มต้นให้ Integer Array

3.2.1.2 เปลี่ยนแปลงค่าในแต่ละตำแหน่งของอาเรย์

ในการประมวลผลเพื่อแปลี่ยนแปลงค่าระยะทางจุดต่างๆ ในอาเรย์นั้นจะใช้ตำแหน่งรอบข้างจุดนั้นในการพิจารณา โดยจะใช้จุด 4 หรือ 8 ตำแหน่งซึ่งมี 3 รูปแบบ ดังนี้

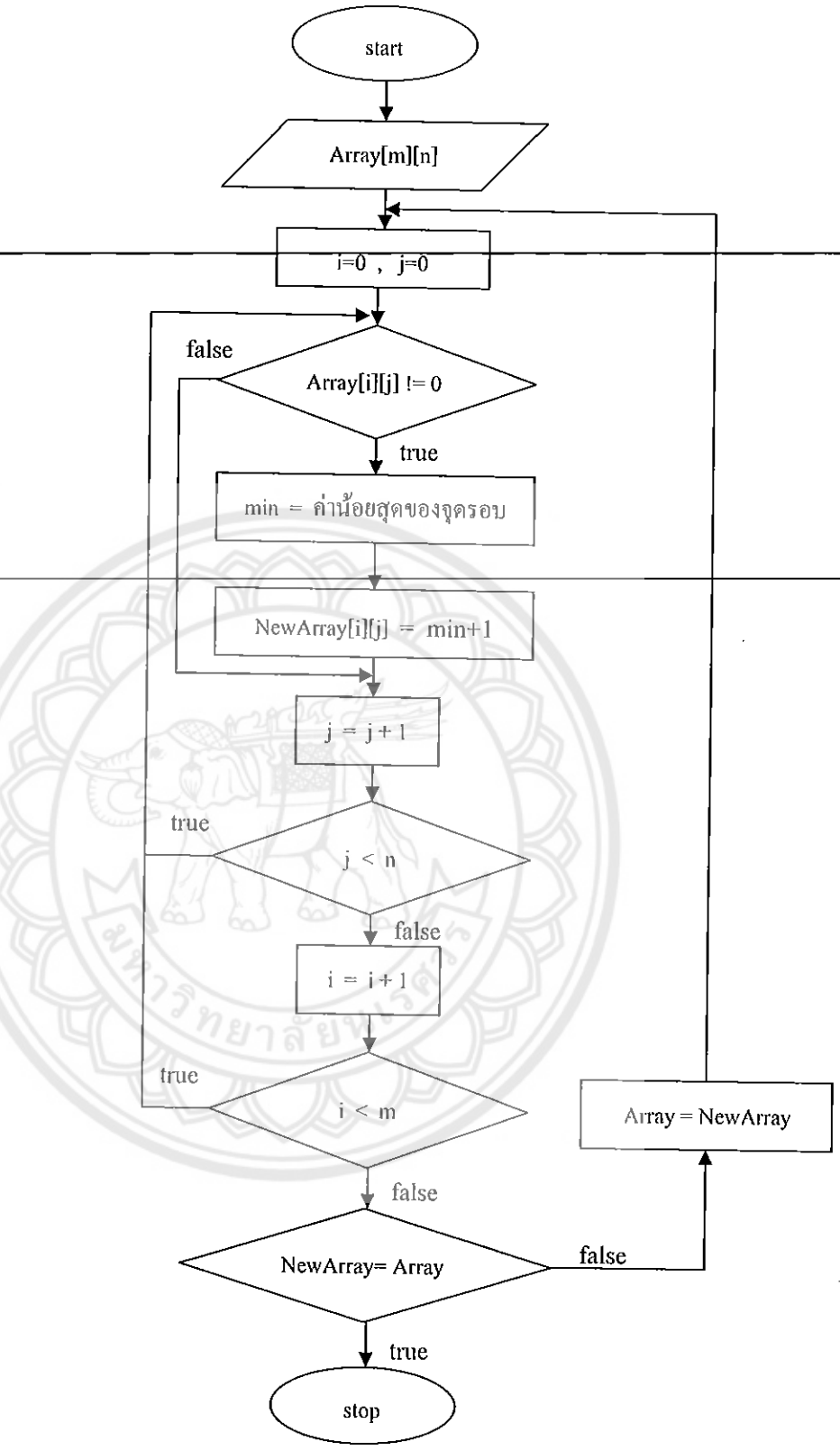


A. cross B. Diagonal C. Full

รูปที่ 3.5 รูปแบบของตำแหน่งรอบข้างที่ใช้ในการพิจารณา

ขั้นตอนการทำงาน

- วนไปทุกตำแหน่งที่ไม่ใช่ 0
- หาค่าน้อยสุด (min) จากตำแหน่งรอบข้าง
- กำหนดค่าตำแหน่งนั้นๆ ในรอบถัดไปเท่ากับ min+1
- ทำซ้ำตั้งแต่ข้อแรกจนกว่าจะไม่มีการเปลี่ยนแปลงค่าในอาเรย์

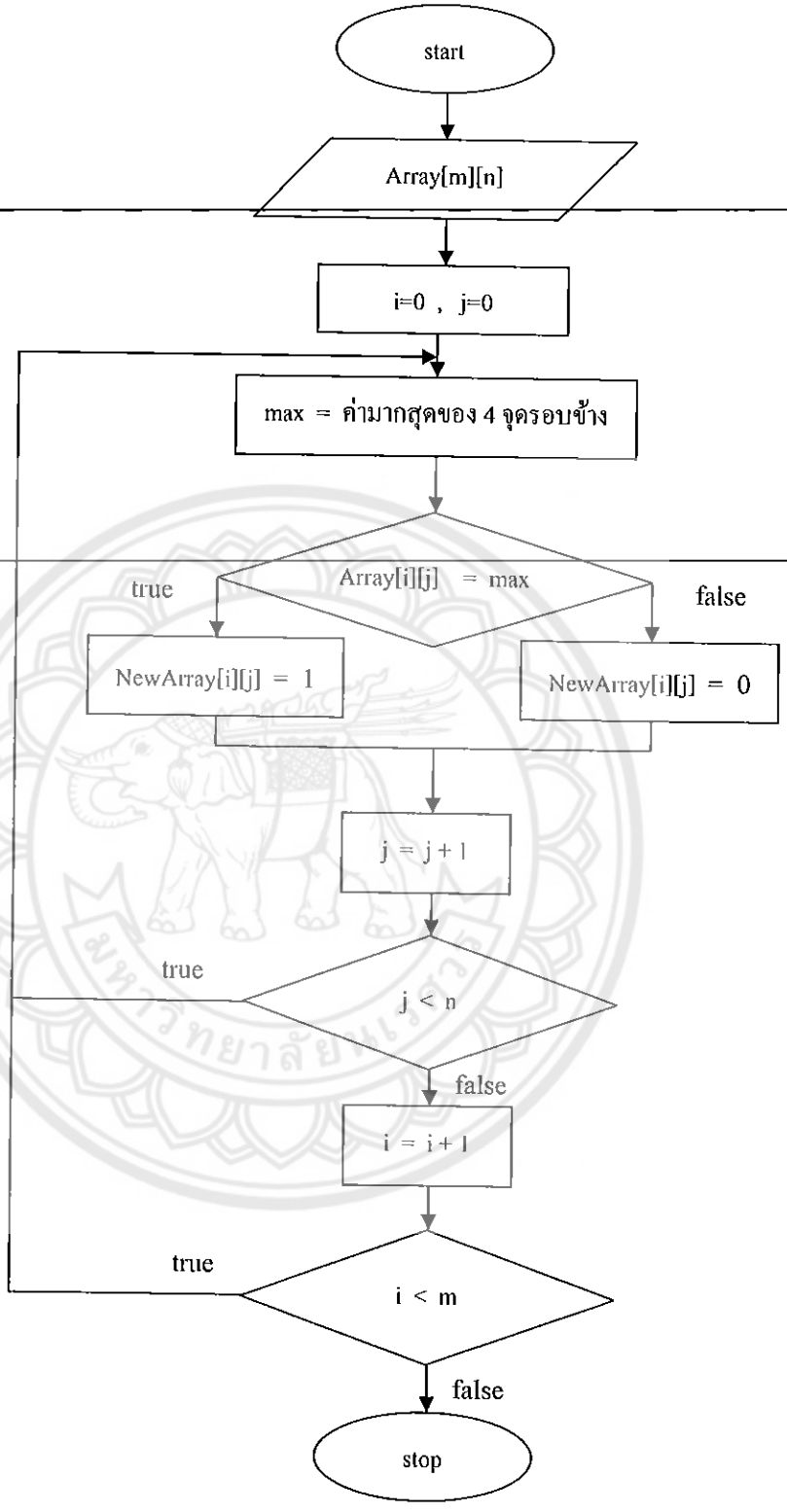


รูปที่ 3.6 แผนผังการทำงานขั้นตอนการเปลี่ยนแปลงค่าตามระยะทาง

3.2.1.2 หาตำแหน่งที่เป็นส่วนหนึ่งของ skeleton

ขั้นตอนนี้เป็นขั้นตอนสุดท้ายของวิธี Distance & Transform โดยจะพิจารณาค่าตำแหน่งในอาเรย์ว่าตำแหน่งใดเป็นส่วนหนึ่งของ skeleton (กำหนดให้เป็นสีดำ) หรือไม่ได้เป็นส่วนหนึ่งของ skeleton (กำหนดให้เป็นสีขาว) ซึ่งมีขั้นตอนการพิจารณาดังต่อไปนี้

- วนไปทุกตำแหน่งในอาเรย์
- หาค่ามากสุด (max) จากตำแหน่งรอบข้างรูปแบบเดียวกับตอนเปลี่ยนแปลงค่าในอาเรย์
- ถ้าค่าระยะทางตำแหน่งที่ i , j มีค่าเท่ากับ max ให้ตำแหน่งนั้นมีค่าเป็น 1 (สีดำ)
- ถ้าเงื่อนไขข้อก่อนหน้านี้ไม่เป็นจริงแล้ว ให้ตำแหน่งนั้นมีค่าเป็น 0 (สีขาว)



รูปที่ 3.7 แผนผังการทำงานขั้นตอนการกำหนดจุดที่เป็นส่วนหนึ่งของ skeleton

3.2.2 Voronoi

ในการประมวลผลแบบ Voronoi นั้นโครงสร้างของแต่ละตำแหน่งจะต้องเก็บข้อมูลระยะทางทั้ง 4 ทิศรวมทั้งสถานะของตำแหน่งนั้นๆ ได้ดังตัวอย่างในรูปที่ 3.8

รูปที่ 3.8 โครงสร้างของแต่ละตำแหน่งในอาเรย์ (Voronoi)

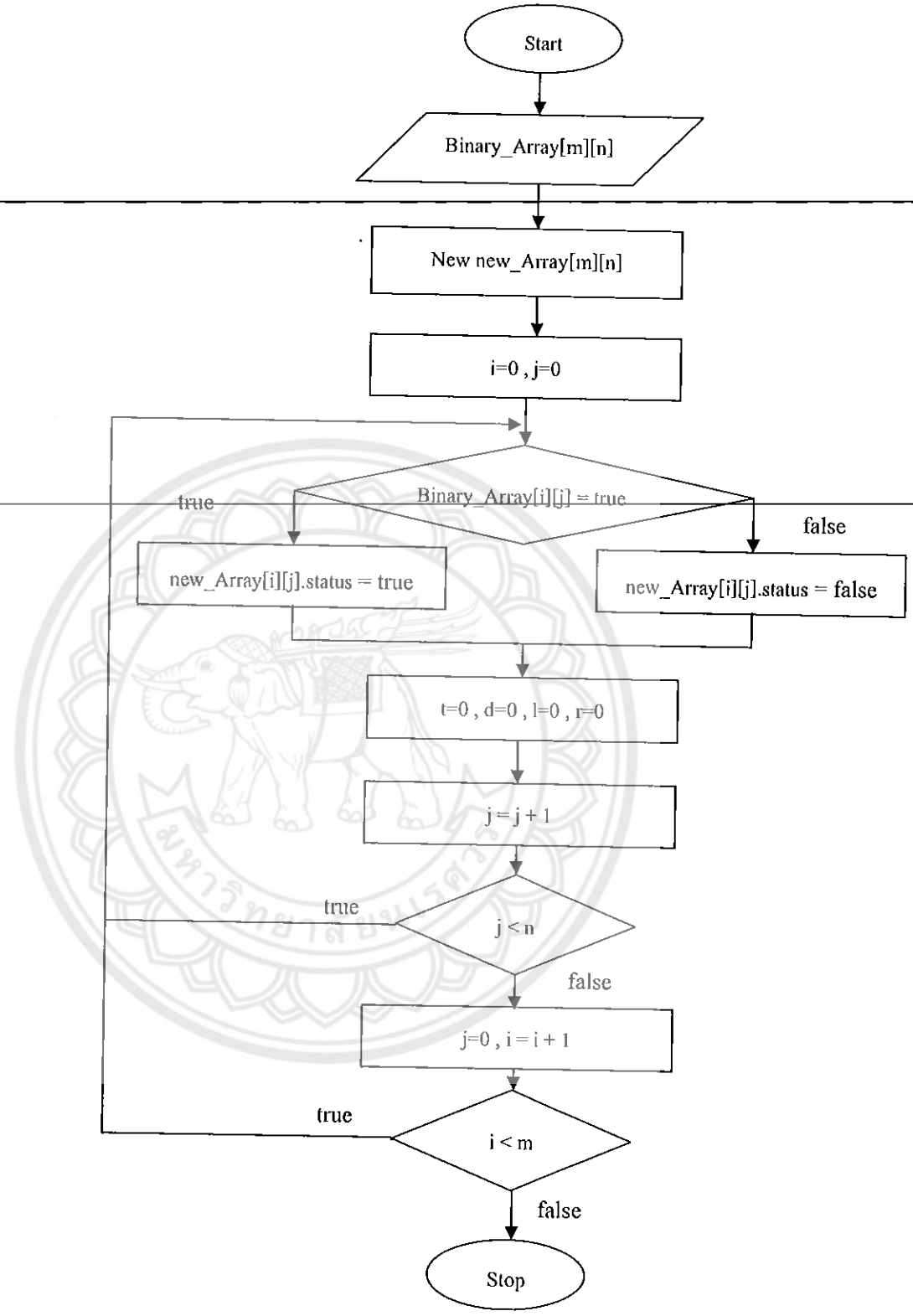
จากรูปที่ 3.8 แต่ละส่วนมีนิยามดังต่อไปนี้

- Status คือ ค่าสถานะของแต่ละตำแหน่งจะสามารถมีค่าได้เพียง true/false เท่านั้น
- top คือค่าระยะทางจากขอบภาพด้านบนถึงตำแหน่งนั้นๆ
- down คือค่าระยะทางจากขอบภาพด้านล่างถึงตำแหน่งนั้นๆ
- left คือค่าระยะทางจากขอบภาพด้านซ้ายถึงตำแหน่งนั้นๆ
- right คือค่าระยะทางจากขอบภาพด้านขวาถึงตำแหน่งนั้นๆ โดยการทำงานแบบ Voronoi นั้นจะแบ่งออกเป็น ขั้นตอนดังนี้
- กำหนดค่าเริ่มต้นให้อาเรย์ 2 มิติ
- เปลี่ยนแปลงค่าในแต่ละตำแหน่งของอาเรย์
- กำหนดตำแหน่งที่เป็น skeleton

3.2.2.1 กำหนดค่าเริ่มต้นให้อาเรย์ 2 มิติ

ขั้นตอนนี้เป็นการกำหนดค่าเริ่มต้นให้กับแต่ละตำแหน่งในอาเรย์ซึ่งมีกฎในการกำหนดดังนี้

- ตำแหน่งที่เป็นสีดำให้ status = true
- ตำแหน่งที่เป็นสีขาวให้ status = false
- top = 0 , down = 0 , left = 0 , right = 0



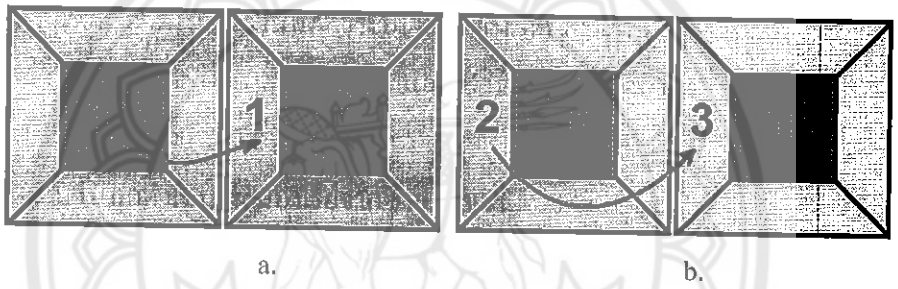
รูปที่ 3.9 แผนผังการทำงานการกำหนดค่าเริ่มต้นให้อาเรย์ (Voronoi)

3.2.2.2 เปลี่ยนแปลงค่าในแต่ละตำแหน่งของอาเรย์

ในขั้นตอนนี้จะทำการกำหนดค่าระยะทางทั้ง 4 ทิศของแต่ละตำแหน่ง คือ ด้านบน ด้านล่าง ด้านซ้าย และด้านขวา ที่อยู่ในวัตถุโดยมีขั้นตอนและวิธีการกำหนดค่าดังนี้

วิธีการกำหนดค่า (ด้านซ้าย)

- วนไปทุกตำแหน่งในอาเรย์จากซ้ายไปขวา
- ถ้า Array[i][j].status = true และ Array[i][j-1].status = false
  - Array[i][j].left = 1
- ถ้า Array[i][j].status = true และ Array[i][j-1].status = true
  - Array[i][j].left = Array[i][j-1].left + 1



a. b.

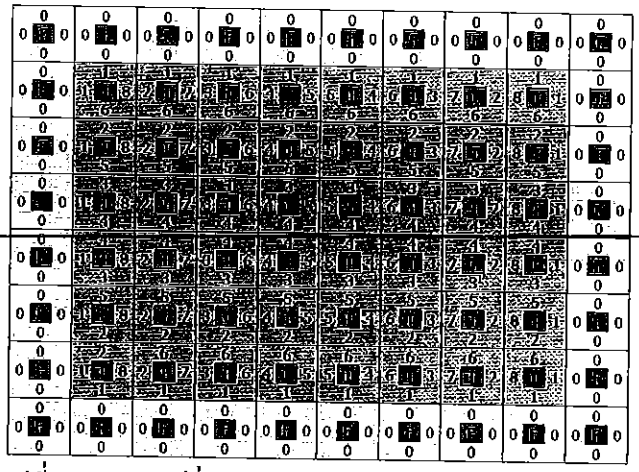
รูปที่ 3.10 ขั้นตอนการกำหนดค่าระยะทางด้านซ้ายของอาเรย์

a. Array[i][j].status = true และ Array[i][j-1].status = false

b. ถ้า Array[i][j].status = true และ Array[i][j-1].status = true

กำหนดค่าระยะทางให้กับ 3 ทิศที่เหลือ โดยใช้วิธีเดียวกับด้านซ้าย ซึ่งลักษณะการวนรอบและกฎที่ใช้เพื่อกำหนดค่าจะแตกต่างกันไปตามทิศที่ต้องการกำหนด เช่น การกำหนดค่าระยะทางด้านขวาก็จะวนไปทุกตำแหน่งในอาเรย์จากขวาไปซ้าย และ Array ที่ใช้เป็นเงื่อนไขในการกำหนดค่าระยะทางให้กับ Array[i][j] คือ Array[i][j+1] เป็นต้น

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

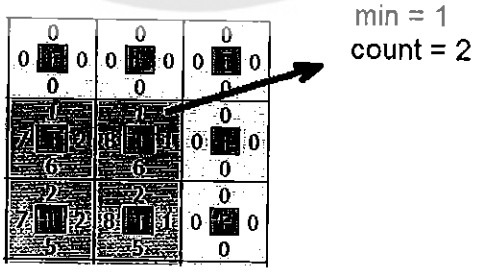


รูปที่ 3.11 อาเรย์ที่กำหนดค่าระยะทางแต่ละตำแหน่งแล้ว

3.2.2.3 กำหนดตำแหน่งที่เป็น skeleton

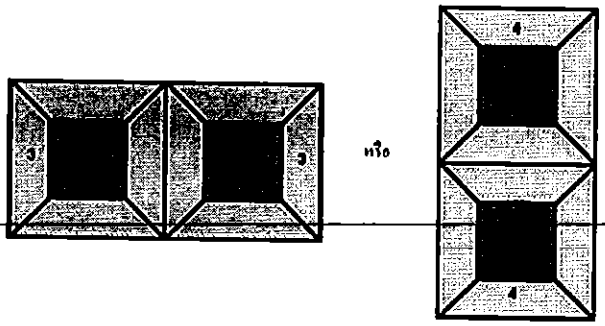
ในขั้นตอนนี้จะเป็นการพิจารณาแต่ละตำแหน่งว่าเป็นส่วนหนึ่งของ skeleton หรือไม่โดยมีขั้นตอนในการพิจารณาดังนี้

- วนไปทุกตำแหน่งในอาเรย์
- min = ค่าระยะทางที่น้อยที่สุดจากทั้ง 4 ทิศ ของตำแหน่งใดๆ
- count = จำนวนค่าระยะทางที่มีค่าเท่ากับ min ของตำแหน่งรอบข้างทั้ง 4 ทิศ
- ถ้า count >= 2 ให้ตำแหน่งนั้นมีค่า status = true (ส่วนหนึ่งของ skeleton)
- ถ้าตำแหน่งนั้นมีตำแหน่งข้างเคียงที่มีระยะทางของฝั่งตรงกันข้ามเท่ากัน ให้ตำแหน่งนั้นมีค่า status = true



รูปที่ 3.12 ตำแหน่งที่มีค่า count >= 2

15738064
ว.ร.
ว4580
2552



รูปที่ 3.13 ตำแหน่งที่มีตำแหน่งข้างเคียงที่มีระยะทางของฝั่งตรงกันข้ามเท่ากัน

- ถ้าเงื่อนไขข้อด้านบนไม่เป็นจริงแล้ว ให้ตำแหน่งนั้นมีค่า status = false (ไม่ได้เป็นส่วนหนึ่งของ skeleton)

3.2.2 Thinning [5]

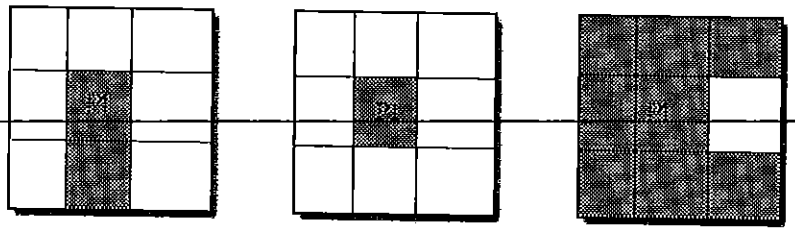
เป็นวิธีการหาแกนของภาพโดยวิธีการลดความกว้างของรูปภาพลงจนเหลือเฉพาะส่วนที่เป็นแกนของภาพซึ่งมีขั้นตอนการทำงานดังนี้

- วนไปทุกตำแหนงในอาเรย์
- ตรวจสอว่าตำแหน่งนั้นเป็นส่วนหนึ่งของ skeleton หรือไม่
- ถ้าตำแหน่งนั้นไม่ใช่ส่วนหนึ่งของ skeleton ให้กำหนดตำแหน่งนั้นมีค่าเป็น false
- ทำซ้ำตั้งแต่ข้อแรกจนกว่าอาเรย์จะไม่มีการเปลี่ยนแปลงค่า

การที่จะลบตำแหน่งนั้นออกได้จะต้องสอดคล้องกับเงื่อนไขดังนี้

- $2 <= B(p1) <= 6$
- $A(p1) = 1$
- $p2.p4.p8 = 0$ or $A(p2) != 1$
- $p2.p4.p6 = 0$ or $A(p4) != 1$

3.2.2.1 เงื่อนไข 2 <= B(p1) <= 6



B(p1) = 1 B(p1) = 0 B(p1) = 7

รูปที่ 3.14 เงื่อนไข B(p1) ที่ไม่สอดคล้องกับกฎ 2 <= B(p1) <= 6

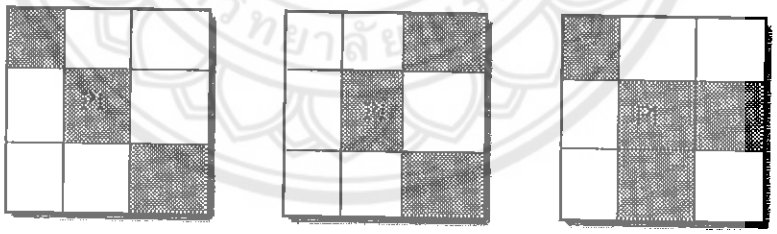
การที่จะลบตำแหน่งนั้นออกได้อย่างน้อยต้องสอดคล้องกับเงื่อนไขว่า 2 <= B(p1) <= 6 เมื่อ B(p1) คือ จำนวนตำแหน่งข้างเคียงตำแหน่ง p1 ที่มีค่าเป็น 1

B(p1) = 1 แสดงว่าตำแหน่งนั้นเป็นปลายของแกน (skeleton) ไม่สามารถลบได้

B(p1) = 0 แสดงว่าตำแหน่งนั้นเป็นแกน (skeleton) ไม่สามารถลบได้

B(p1) >= 7 แสดงว่าตำแหน่งนั้นอยู่ในรูปแบบที่อาจจะเป็นส่วนหนึ่งของแกน (skeleton) ได้ดังนั้นจึงยังไม่สามารถลบตำแหน่งนั้นได้

3.2.2.2 เงื่อนไข A(p1) = 1

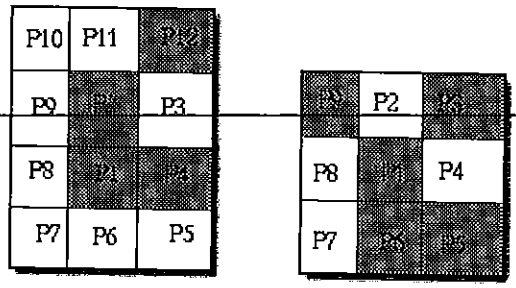


A(p1)=2 A(p1)=2 A(p1)=3

รูปที่ 3.15 เงื่อนไข A(p1) ที่ไม่สอดคล้องกับกฎ A(p1) = 1

เงื่อนไขนี้จะทำการตรวจสอบว่ามีลำดับ 01 จากลำดับ p2, p3, p4, …, p9, p2 ว่าเท่ากับ 1 หรือไม่ ซึ่งถ้า A(p1) มากกว่า 1 แสดงว่าตำแหน่งนั้นเป็นส่วนกลางของแกน (skeleton) ถ้าลบออกผลลัพธ์ที่ได้จะขาดคุณสมบัติ topological

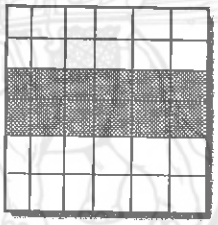
3.2.2.3 p2.p4.p8 = 0 or A(p2) != 1



A(p2) = 2 p2.p4.p8 = 0

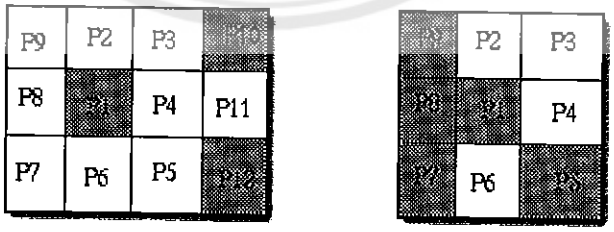
รูปที่ 3.16 A(p2) = 2 และ p2.p4.p8 = 0

เงื่อนไขนี้จะทำการรักษาแกนในแนวนอนที่มีความหนา 2 พิกเซลซึ่งอาจถูกลบออกหมด ให้เหลือไว้ 1 พิกเซล โดยจะรักษาพิกเซลที่อยู่ด้านล่างไว้



รูปที่ 3.17 ตำแหน่งในแนวนอนที่มีความหนา 2 พิกเซล

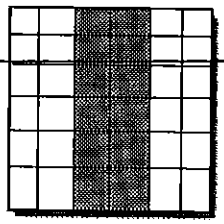
3.2.2.4 p2.p4.p6 = 0 or A(p4) != 1



A(p4) = 2 p2.p4.p8 = 0

รูปที่ 3.18 A(p2) = 2 และ p2.p4.p8 = 0

เงื่อนไขนี้จะทำการรักษาแกนในแนวตั้งที่มีความหนา 2 พิกเซลซึ่งอาจถูกลบออกหมดให้เหลือไว้ 1 พิกเซล โดยจะรักษาพิกเซลที่อยู่ด้านซ้ายไว้



รูปที่ 3.19 ตำแหน่งในแนวตั้งที่มีความหนา 2 พิกเซล

มหาวิทยาลัยนเรศวร

บทที่ 4
การออกแบบซอฟต์แวร์

เพื่อความสะดวกที่ผู้อื่นจะทำการศึกษาซอฟต์แวร์ที่พัฒนาขึ้น บทนี้จะอธิบายสิ่งต่างๆ ที่จำเป็นต่อการพัฒนาซอฟต์แวร์ดังนี้

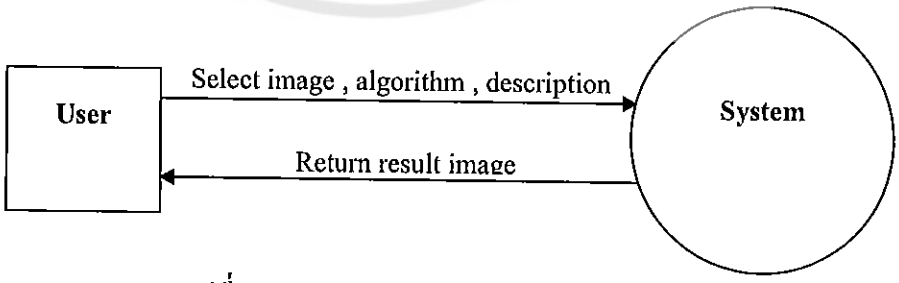
- ความต้องการของระบบ (Requirement Specification)
- ขอบเขตของระบบ
- การออกแบบซอฟต์แวร์

4.1 ความต้องการของระบบ (Requirement Specification)

โปรแกรมสำหรับหาแกนของภาพ (skeletonization) 2 มิติ มีความต้องการของระบบต่างๆ ดังนี้

- สามารถเลือกภาพที่ต้องการได้
- สามารถเลือก algorithm ที่ต้องการในการประมวลผลได้
- สามารถบันทึกรูปภาพผลลัพธ์ได้

4.2 ขอบเขตของระบบ



รูปที่ 4.1 Context Diagram ของระบบ

การทำงานของระบบเริ่มจากผู้ใช้ ทำการส่งรูปภาพที่ต้องการเข้าสู่ระบบ จากนั้นผู้ใช้ทำการเลือก algorithm และรายละเอียดต่างๆ ที่ต้องการแล้วสั่งให้ระบบทำการประมวลผล จากนั้นระบบจะทำการประมวลผลเพื่อหาแกนของภาพ (skeletonization) และส่งภาพผลลัพธ์ให้กับผู้ใช้ต่อไป

4.3 การออกแบบซอฟต์แวร์

การออกแบบซอฟต์แวร์นั้นควรออกแบบให้ครอบคลุมทั้ง 4 มุมมองดังรูปด้านล่าง

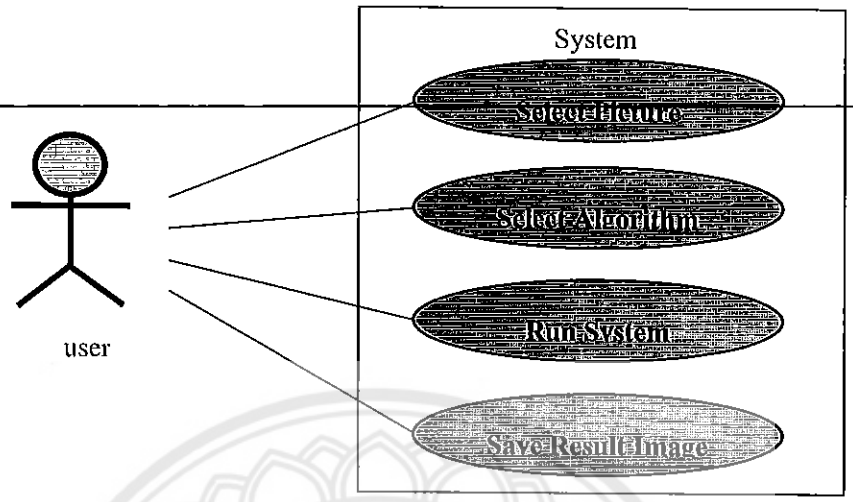
	Dynamic	Static
External	- Use case diagram - Sequence diagram	Component diagrams
Internal	Activity diagrams	Class diagrams

รูปที่ 4.2 มุมมองในการออกแบบซอฟต์แวร์

โดยแต่ละมุมมองมีความหมายดังต่อไปนี้

- External-Dynamic แสดงการติดต่อภายนอกที่สามารถมองเห็นได้ (การโต้ตอบระหว่างผู้ใช้กับโปรแกรม)
- External-Static แสดงโครงสร้างและส่วนประกอบต่างๆ อย่างคร่าวๆ
- Internal-Dynamic แสดงพฤติกรรมและสถานะต่างๆ ของระบบ
- Internal-Static แสดงโครงสร้างและส่วนประกอบต่างๆ ภายในระบบ

4.3.1 ยูสเคลส ไดอะแกรม (Use case diagram)



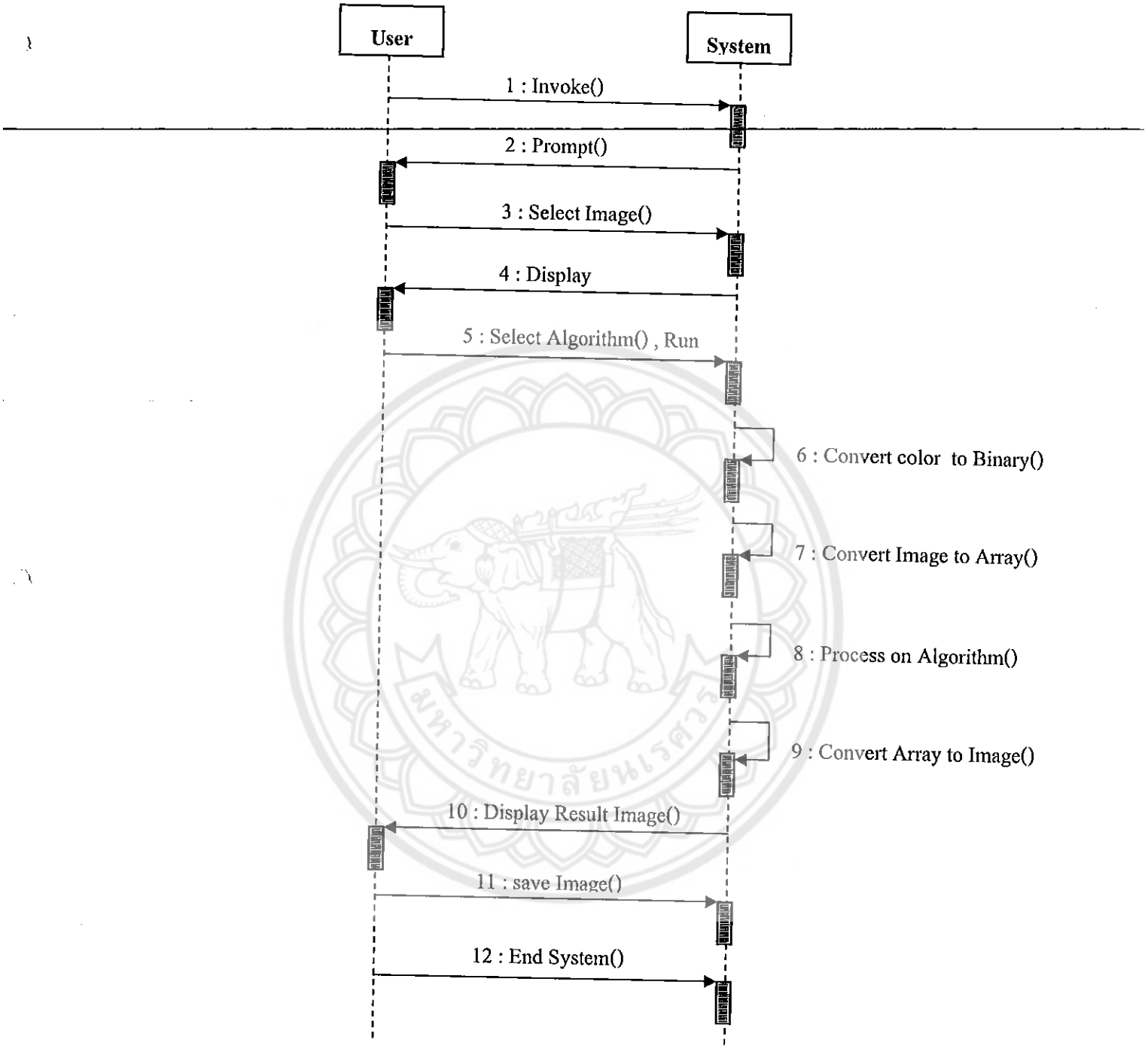
รูปที่ 4.3 Use case diagram

การติดต่อกับผู้ใช้นั้นจะมี 4 ขั้นตอนคือ

1. ผู้ใช้ทำการเลือกภาพที่ต้องการประมวลผล
2. ผู้ใช้ทำการเลือกอัลกอริทึม
3. ผู้ใช้สั่งให้ระบบทำการประมวลผล
4. ผู้ใช้ทำการบันทึกรูปภาพผลลัพธ์ไว้ในหน่วยความจำ

4.3.2 ซีเควน ไดอะแกรม (Sequence diagram)

ลำดับการทำงานของโปรแกรมนี้เริ่มจากการที่ผู้ใช้เรียกโปรแกรมขึ้นมา จากนั้นทำการเลือกรูปภาพที่ต้องการหาแกนของภาพ จากนั้นระบบจะทำการแสดงรูปภาพออกทางหน้าจอ เมื่อได้ภาพที่ต้องการแล้วผู้ใช้ก็ทำการเลือกอัลกอริทึมที่ต้องการใช้ในการประมวลผล จากนั้นผู้ใช้ก็สั่งให้โปรแกรมเริ่มทำการประมวลผล ซึ่งโปรแกรมจะเริ่มประมวลผลโดยการแปลงภาพให้เป็นแบบแบบขาวดำ แล้วทำการเก็บข้อมูลของรูปภาพเป็นโครงสร้างของอาเรย์ 2 มิติ จากนั้นก็ทำการประมวลผลตามอัลกอริทึมที่ผู้ใช้เลือก จากนั้นก็ทำการแปลงข้อมูลจากอาเรย์กลับมาเป็นรูปภาพแล้วแสดงรูปภาพผลลัพธ์ออกทางหน้าจอ ถ้าผู้ใช้ต้องการบันทึกภาพผลลัพธ์ระบบก็จะทำการบันทึกภาพเก็บไว้ใน memory ตามที่ผู้ใช้กำหนด ดังรูปที่ 4.4



รูปที่ 4.4 Sequence diagram

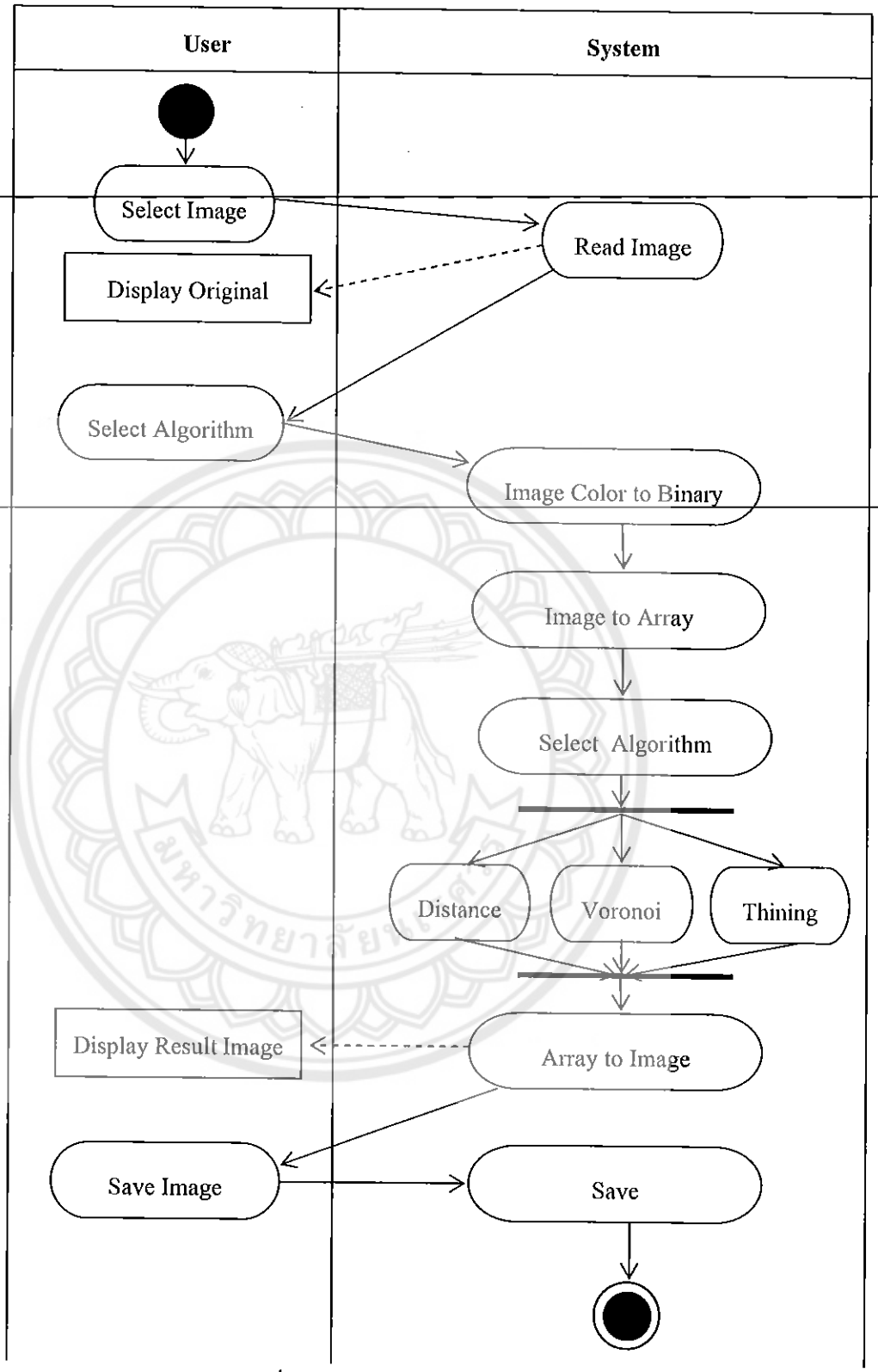
4.3.3 คอมโพเนนท์ ไดอะแกรม (Component Diagram)

เนื่องจากโปรแกรมไม่ได้ติดต่อ หรือเรียกใช้ function ภายนอก ดังนั้นจึงมีเพียง component ของโปรแกรมหลักเพียง component เดียว

รูปที่ 4.5 Component diagram

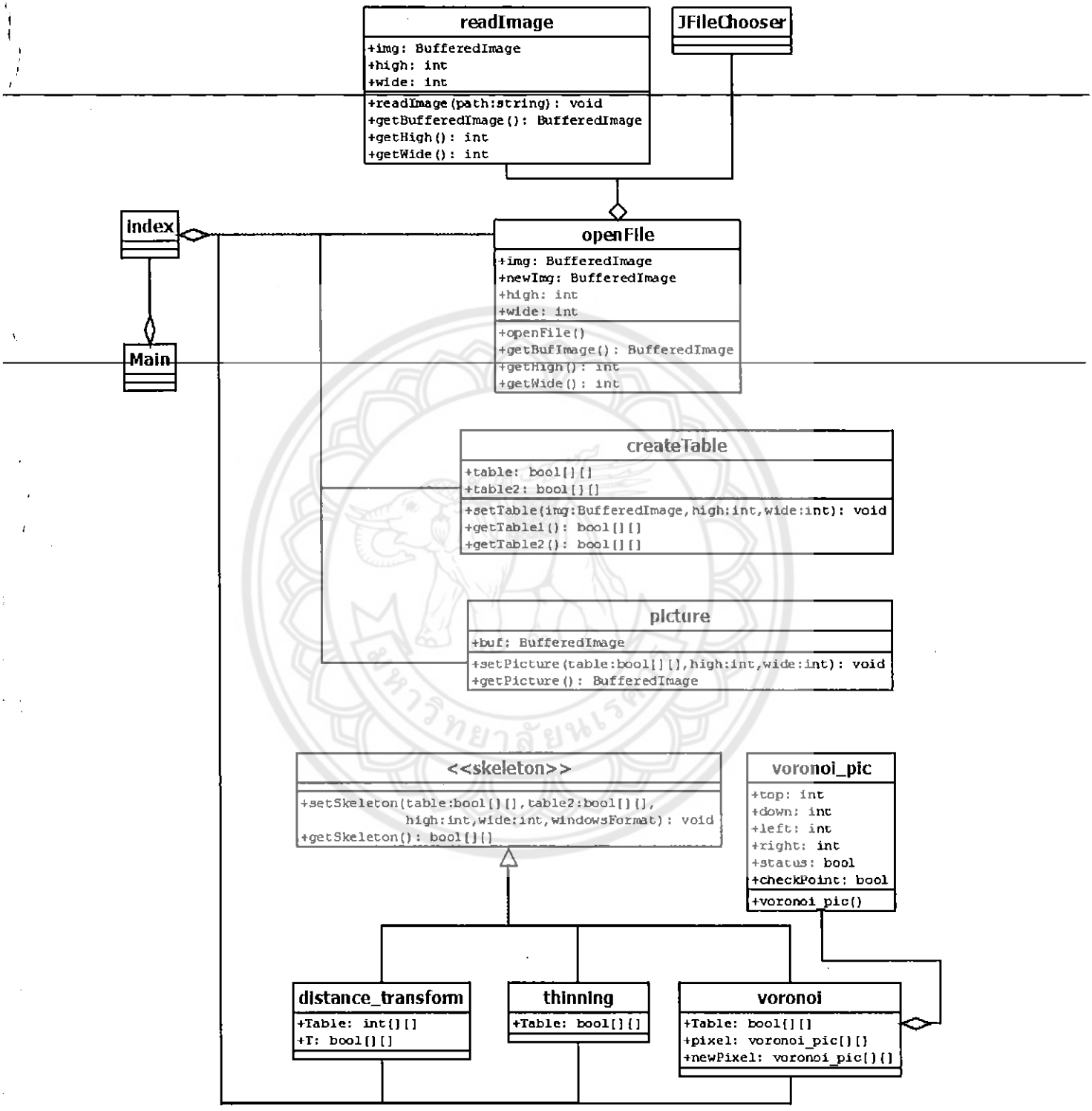
4.3.4 แอคทีวิตี้ไดอะแกรม (Activity diagram)

ลำดับกิจกรรมต่างๆ ของโปรแกรมจะเริ่มจากผู้ใช้ทำการเลือกรูปภาพที่ต้องการประมวลผล จากนั้นโปรแกรมจะทำการอ่านไฟล์รูปภาพมาเก็บไว้ในหน่วยความจำแล้วแสดงรูปภาพออกทางหน้าจอ จากนั้นผู้ใช้ทำการเลือกอัลกอริทึมที่ต้องการแล้วสั่งให้โปรแกรมเริ่มทำการประมวลผล จากนั้นระบบทำการแปลงภาพให้กลายเป็นโทนสีเทา และเป็นไบนารีตามลำดับแล้วทำการเก็บข้อมูลของรูปภาพในรูปแบบของอาเรย์ 2 มิติ จากนั้นระบบทำการประมวลผลตามอัลกอริทึมที่ผู้ใช้เลือกจนเสร็จ จากนั้นระบบจึงทำการแปลงข้อมูลจากอาเรย์ 2 มิติกลับเป็นรูปภาพแล้วแสดงรูปภาพผลลัพธ์ออกมาทางหน้าจอ จากนั้นผู้ใช้ทำการเลือกชื่อ และที่อยู่ในหน่วยความจำของรูปภาพที่ต้องการบันทึกแล้วสั่งให้โปรแกรมเริ่มทำการบันทึกข้อมูล จากนั้นโปรแกรมจะทำการบันทึกข้อมูลรูปภาพไว้ในหน่วยความจำที่กำหนด



รูปที่ 4.6 Activity diagram

4.3.4 คลาสไดอะแกรม (Class diagram)



รูปที่ 4.7 Class diagram

ใน Class Diagram จะประกอบด้วย 12 Class หลักๆ ดังนี้

1. Main ทำหน้าที่เป็น Class หลักซึ่งเลือกรูปแบบของ GUI ที่ใช้ติดต่อกับ user
2. index เป็น GUI ซึ่งใช้ติดต่อกับผู้ใช้
3. readImage ทำหน้าที่ในการอ่านไฟล์รูปภาพมาเก็บเป็นชนิด BuferedImage
4. JFileChooser ทำหน้าที่แสดงหน้าต่างสำหรับเปิดหรือบันทึกไฟล์
5. openFile สำหรับเปิดไฟล์โดยจะเรียกใช้ JFileChooser และ readImage
6. createTable สำหรับแปลงข้อมูลรูปภาพให้อยู่ในรูปแบบของอาเรย์ 2 มิติ
7. picture สำหรับแปลงอาเรย์ 2 มิติให้กลายเป็นรูปภาพ
8. skeleton เป็นอินเตอร์เฟสคลาสสำหรับอัลกอริทึม
9. voronoi_pic เป็นคลาสสำหรับเก็บข้อมูล 1 ตำแหน่งของอัลกอริทึมแบบ voronoi
10. voronoi เป็นอัลกอริทึมรูปแบบหนึ่งที่ใช้ในการประมวลผล
11. distance_transform เป็นอัลกอริทึมรูปแบบหนึ่งที่ใช้ในการประมวลผล
12. thinning เป็นอัลกอริทึมรูปแบบหนึ่งที่ใช้ในการประมวลผล

บทที่ 5

ผลการทดสอบ

การทดสอบการทำงานของโปรแกรมนั้นแบ่งออกเป็น 2 การทดสอบหลักๆ ดังนี้

1. ทดสอบการทำงานแต่ละอัลกอริทึม
2. ทดสอบว่าโปรแกรมสามารถแสดงผลแบบ GUI ได้หรือไม่
3. ทดสอบว่าโปรแกรมสามารถประมวลผลบนระบบปฏิบัติการ Windows Xp ได้หรือไม่

5.1 ทดสอบการทำงานแต่ละอัลกอริทึม

เนื่องจากอัลกอริทึมที่ใช้ในการประมวลผลนั้นมีหลายอัลกอริทึม ผู้จัดทำจึงทำการทดสอบทีละอัลกอริทึม ดังนี้

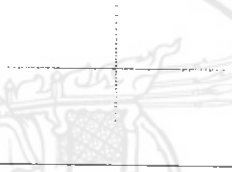
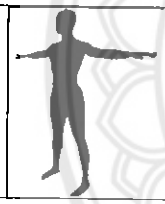
- Full distance & transform
- Diagonal distance & transform
- Cross distance & transform
- Thinning
- Voronoi

โดยการทดสอบแต่ละอัลกอริทึมนั้นจะทำการทดสอบกับรูปภาพแบบ 2 มิติ เพื่อทดสอบความถูกต้อง โดยรูปภาพที่ใช้สำหรับทดสอบการทำงานของอัลกอริทึมนั้นมี 50 รูป ได้แก่

- รูปสี่เหลี่ยมผืนผ้าสีดำพื้นสีขาว
- รูปสี่เหลี่ยมผืนผ้าสีดำถูกซ้อนทับด้วยรูปสี่เหลี่ยมผืนผ้าที่เล็กกว่าสีขาว และมีพื้นสีขาว
- รูปสี่เหลี่ยมผืนผ้าสีดำถูกซ้อนทับด้วยวงรีสีขาว และมีพื้นสีขาว
- รูปอักษรตัวที (T) สีดำพื้นสีขาว
- รูปวงกลมสีดำพื้นสีขาว
- รูปวงรีสีดำพื้นสีขาว
- รูปมนุษย์สีดำพื้นสีขาว ฯลฯ

5.1.1 Full distance & transform

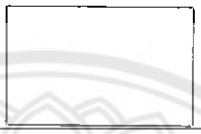
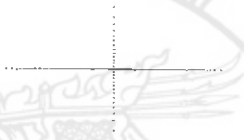
ตารางที่ 5.1 ตารางผลการทดสอบอัลกอริทึม full distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	No
		No	No
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	No
		No	Yes

อัลกอริทึมแบบ full distance & transform สามารถประมวลผลได้แกนของภาพตามที่คาดหวังไว้ แม้ว่าผลลัพธ์ที่ได้ส่วนใหญ่นั้นจะไม่มีคุณสมบัติของ Topological และ Geometrical แต่ก็ถือว่าผลลัพธ์ออกมาอยู่ในระดับที่น่าพอใจ

5.1.2 Diagonal distance & transform

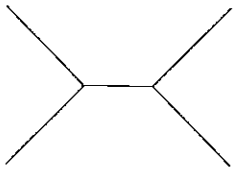
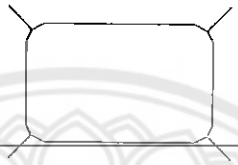
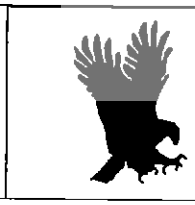
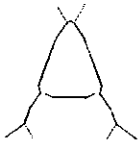
ตารางที่ 5.2 ตารางผลการทดสอบอัลกอริทึม diagoanl distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	No
		No	No
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	No
		No	Yes

อัลกอริทึมแบบ diagoanl distance & transform สามารถประมวลผลได้แกนของภาพตามที่คาดหวังไว้ แม้ว่าผลลัพธ์ที่ได้ส่วนใหญ่นั้นจะไม่มีคุณสมบัติของ Topological และ Geometrical แต่ก็ถือว่าผลลัพธ์ออกมาอยู่ในระดับที่น่าพอใจ

5.1.3 Cross distance & transform

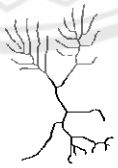
ตารางที่ 5.3 ตารางผลการทดสอบอัลกอริทึม cross distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	Yes
		Yes	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
A		No	Yes
9		No	Yes

อัลกอริทึมแบบ cross distance & transform สามารถประมวลผลได้แกนของภาพตามที่คาดหวังไว้ ภาพผลลัพธ์ส่วนใหญ่จะมีคุณสมบัติของ Geometrical แต่ขาดคุณสมบัติ Topological ซึ่งเป็นไปตามทฤษฎีของ distance & transform

5.1.4 Thinning

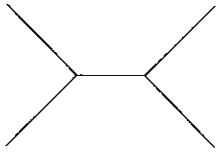
ตารางที่ 5.4 ตารางผลการทดสอบอัลกอริทึม thinning

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	No
		Yes	No
		Yes	No
		Yes	Yes
		Yes	Yes
		Yes	Yes
A		Yes	No
9		Yes	Yes

อัลกอริทึมแบบ thinning สามารถประมวลผลได้แกนของภาพตามที่คาดหวังไว้ ซึ่งภาพผลลัพธ์ส่วนใหญ่จะมีคุณสมบัติของ Topological แต่ขาดคุณสมบัติของ Geometrical ซึ่งเป็นไปตามทฤษฎีของอัลกอริทึมแบบ thinning

5.1.5 Voronoi

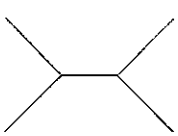
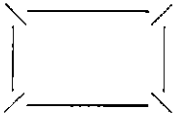
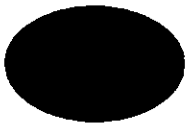
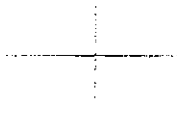
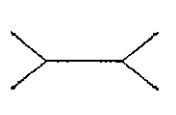
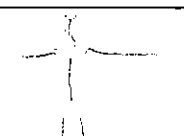
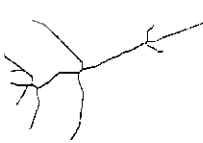
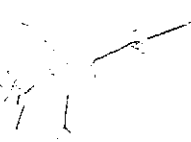
ตารางที่ 5.5 ตารางผลการทดสอบอัลกอริทึม voronoi

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	Yes
		No	Yes
		Yes	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes

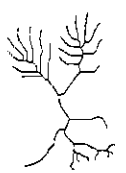
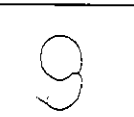
มหาวิทยาลัยพะเยา

อัลกอริทึมแบบ Voronoi สามารถประมวลผลได้แกนของภาพตามที่คาดหวังไว้ ซึ่งภาพผลลัพธ์ส่วนใหญ่จะมีคุณสมบัติของ Geometrical แต่ขาดคุณสมบัติของ Topological และจะมีจุดภาพที่ไม่พึงประสงค์ (Noise) เพิ่มเข้ามาในภาพผลลัพธ์ ซึ่งเกิดจากภาพที่เป็นเส้นโค้งซึ่งแท้ที่จริงแล้วเส้นโค้งที่เห็นเกิดจากจุดสี่เหลี่ยมวางต่อกันไม่ได้เป็นเส้นโค้งอย่างแท้จริง

ตารางที่ 5.6 ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม

ภาพต้นแบบ	Full distance & transform	Diagonal distance & transform	Cross distance & transform	Thinning	Voronoi
					
					
					
					
					

ตารางที่ 5.6 (ต่อ) ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม

						
A						
9						

50

ตารางที่ 5.7 เปรียบเทียบประสิทธิภาพแต่ละอัลกอริทึม

อัลกอริทึม	Topological (%)	Geometrical (%)
Full distance & transform	6	42
Diagonal distance & transform	6	42
Cross distance & transform	10	100
Thinning	100	36
Voronoi	8	100

จากตาราง อัลกอริทึมที่มีค่าเฉลี่ยระหว่าง Topological และ Geometrical ดีที่สุดคือ อัลกอริทึม Thinning รองลงมาคือ Cross distance & transform, Voronoi, Diagonal distance & transform และ Full distance & transform ตามลำดับ

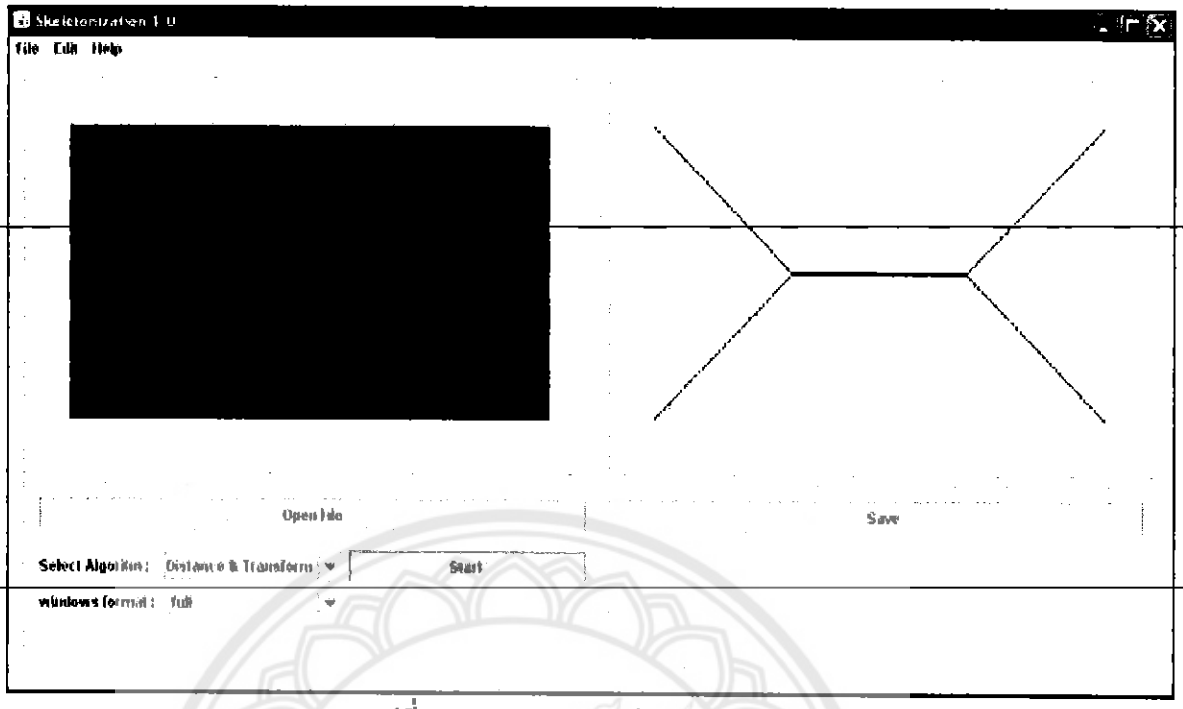
5.2 ทดสอบว่าสามารถแสดงผลแบบ GUI ได้

ในขั้นตอนทดสอบการแสดงผลแบบ GUI นั้นจะมีลำดับการทดสอบดังนี้

1. การแสดงผลหน้าหลักของโปรแกรม
2. การเปิดไฟล์รูปภาพ
3. การบันทึกรูปภาพ

5.2.1 การแสดงผลหน้าหลักของโปรแกรมเมื่อโปรแกรมเริ่มทำงาน

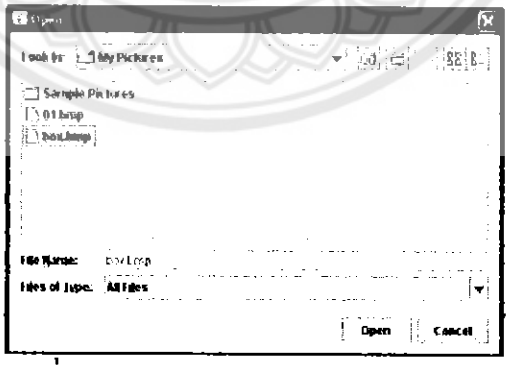
จากการทดสอบได้ผลว่าหน้าหลักของโปรแกรมสามารถแสดงผลได้ดี โดยเมื่อผู้ใช้เปิดไฟล์ภาพเรียบร้อยแล้วหน้าหลักของโปรแกรมก็จะทำการแสดงรูปภาพที่ผู้ใช้เลือกออกทางหน้าจอด้านซ้าย จากนั้นเมื่อผู้ใช้เลือกอัลกอริทึม และสั่งให้โปรแกรมเริ่มประมวลผล เมื่อโปรแกรมทำการประมวลผลเสร็จสิ้นจะทำการแสดงรูปภาพผลลัพธ์ออกทางหน้าจอฝั่งขวา ซึ่งไม่พบข้อผิดพลาดในการทดสอบ



รูปที่ 5.1 GUI หลักของโปรแกรม

5.2.2 การเปิดไฟล์รูปภาพ

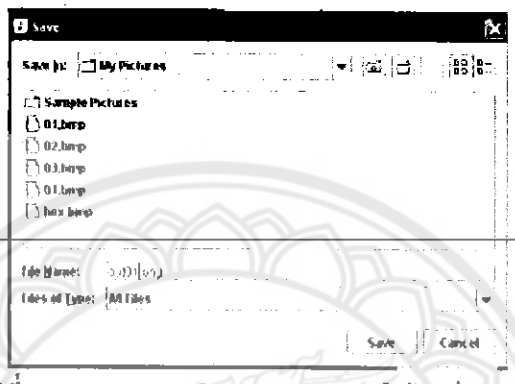
จากการทดสอบได้ผลว่า GUI ส่วนของการเปิดไฟล์ภาพของโปรแกรมสามารถทำงานได้ดี โดยผู้ใช้สามารถเลือกรูปภาพจาก directory ที่ต้องการ จากนั้นทำการคลิ๊กปุ่ม Open เป็นอันเสร็จสิ้นการเปิดไฟล์รูปภาพ ซึ่งไม่พบข้อผิดพลาดในการทดสอบ



รูปที่ 5.2 GUI ส่วนของการเปิดไฟล์รูปภาพ

5.2.3 การบันทึกรูปภาพ

จากการทดสอบได้ผลว่า GUI ส่วนของการบันทึกไฟล์ภาพของโปรแกรมสามารถทำงานได้ดี โดยผู้ใช้สามารถเลือก directory ที่ต้องการบันทึก จากนั้นทำการตั้งชื่อไฟล์พร้อมนามสกุล จากนั้นทำการคลิ๊กปุ่ม Save เป็นอันเสร็จสิ้นการบันทึกไฟล์รูปภาพ ซึ่งไม่พบข้อผิดพลาดในการทดสอบ



รูปที่ 5.3 GUI หลักส่วนของการบันทึกไฟล์รูปภาพ

5.3 ทดสอบว่าสามารถประมวลผลบนระบบปฏิบัติการ Windows Xp ได้

ตารางที่ 5.8 ทดสอบการทำงานบนระบบปฏิบัติการ Windows Xp Sp3

การทดสอบ	ผลการทดสอบ
GUI หน้าหลักของโปรแกรม	ผ่าน
GUI หน้า open file	ผ่าน
GUI หน้า save file	ผ่าน
การทำงานของอัลกอริทึม full distance & transform	ผ่าน
การทำงานของอัลกอริทึม diagonal distance & transform	ผ่าน
การทำงานของอัลกอริทึม cross distance & transform	ผ่าน
การทำงานของอัลกอริทึม thinning	ผ่าน
การทำงานของอัลกอริทึม Voronoi	ผ่าน

บทที่ 6
บทสรุป

โครงงานนี้ใช้ทฤษฎีของการประมวลผลภาพเพื่อหาแกนของภาพ 2 มิติ โดยมีอัลกอริทึมที่ใช้ในการประมวลผล 3 อัลกอริทึม ประกอบด้วย distance & transform, thinning และ Voronoi โดยอัลกอริทึม distance & transform นั้นแบ่งออกเป็น 3 อัลกอริทึมย่อย คือ full distance & transform, diagonal distance & transform และ cross distance & transform รวมทั้งสิ้นแล้วมีทั้งหมด 5 อัลกอริทึม

ผู้ศึกษาใช้ภาษาจาวา (Java) ในการพัฒนา เนื่องจากเป็นภาษาเชิงวัตถุมีความยืดหยุ่นสูง สามารถทำงานได้ในคอมพิวเตอร์ต่างระบบ และมีคลาสจำนวนมากให้ใช้

6.1 สรุปผลการทดสอบ

จากผลการทดสอบโปรแกรมที่พัฒนาขึ้นสามารถหาแกนของภาพได้ตามที่คาดหวังไว้ โดยอัลกอริทึมที่ได้ผลลัพธ์เฉลี่ยระหว่าง Topological และ Geometrical เป็นที่น่าพอใจที่สุดคือ thinning รองลงมาคือ cross distance & transform, Voronoi, diagonal distance & transform และ full distance & transform ตามลำดับ ส่วนอัลกอริทึมที่ได้ผลลัพธ์ที่มีคุณสมบัติ Topological ดีที่สุด คือ อัลกอริทึมแบบ thinning และอัลกอริทึมที่ได้ผลลัพธ์ที่มีคุณสมบัติ Geometrical ดีที่สุดคือ อัลกอริทึมแบบ cross distance & transform และ Voronoi โดยแต่ละอัลกอริทึมสามารถสรุปได้ดังนี้

อัลกอริทึม thinning นั้นได้ผลลัพธ์ค่อนข้างถูกต้องที่สุด โดยแกนของภาพที่ได้มีคุณสมบัติ Topological แต่ขาดคุณสมบัติ Geometrical ซึ่งสอดคล้องกับทฤษฎีของ thinning และได้แกนของภาพตามที่คาดหวังไว้ทุกรูป

อัลกอริทึมแบบ cross distance & transform นั้นแกนของภาพที่ได้ส่วนมากจะมีคุณสมบัติ Geometrical แต่จะขาดคุณสมบัติ Topological ซึ่งสอดคล้องกับทฤษฎีของ distance & transform และได้แกนของภาพตามที่คาดหวังไว้ทุกรูป

อัลกอริทึมแบบ full distance & transform และ diagonal distance & transform นั้นแกนของภาพที่ได้ส่วนมากแม้จะขาดคุณสมบัติทั้ง Geometrical และ Topological แต่ก็ก็สามารถประมวลผลได้แกนตามที่คาดหวังไว้ทุกรูป

อัลกอริทึม Voronoi นั้นแกนของภาพที่ได้ส่วนมากจะมีคุณสมบัติ Geometrical แต่ขาดคุณสมบัติ Topological แต่มีปัญหา คือมีเส้นที่ไม่ต้องการเพิ่มเข้ามาในรูปภาพผลลัพธ์ซึ่งเป็นข้อจำกัดของอัลกอริทึมแบบ voronoi ซึ่งพัฒนาโดยแนวคิดการวัดระยะทางจากขอบ 4 ทิศ

ตารางที่ 6.1 สรุปการทำงานแต่ละอัลกอริทึม

อัลกอริทึม	Topological	Geometrical
Full distance & transform	No	No
Diagonal distance & transform	No	No
Cross distance & transform	No	Yes
Thinning	Yes	No
Voronoi	No	Yes

อัลกอริทึมแบบต่างๆ นั้นจะมีข้อดีข้อเสียแตกต่างกันออกไป ซึ่งการเลือกใช้อัลกอริทึมนั้นผู้ใช้จะต้องเลือกตามความเหมาะสมในการนำไปใช้

6.2 ปัญหาและแนวทางแก้ไข

โปรแกรมที่พัฒนาขึ้นยังไม่ได้ทำส่วนของการตรวจจับ และแยกวัตถุออกจากภาพพื้นหลังที่ไม่ต้องการ ดังนั้นก่อนการประมวลผลภาพใดๆ ต้องนำภาพไปลบส่วนของพื้นหลังให้เป็นสีขาวทั้งหมดเสียก่อนด้วยโปรแกรมสำเร็จรูปที่ใช้จัดการกับรูปภาพทั่วไปก่อน เช่น โปรแกรม Adobe Photoshop หรือโปรแกรม paint เป็นต้น ซึ่งต้องเพิ่มส่วนดังกล่าวเข้ามาในโปรแกรมต่อไป

อัลกอริทึมที่พัฒนาขึ้นยังไม่มีอัลกอริทึมใดที่ที่ประมวลผลรูปภาพใดๆ แล้วให้ผลลัพธ์ที่มีคุณสมบัติครบทั้ง Topological และ Geometrical เลย ซึ่งจะต้องพัฒนา หรือแก้ไขปรับปรุงอัลกอริทึมให้มีความถูกต้องสมบูรณ์มากขึ้นต่อไป

6.3 ข้อเสนอแนะ

1. อาจนำอัลกอริทึมเดิมไปปรับปรุงให้สามารถประมวลผมแบบขนานเพื่อให้โปรแกรมสามารถประมวลผลได้เร็วขึ้น

เอกสารอ้างอิง

[1] - . "ภาพดิจิทัล." [online]. Available: http://th.wikipedia.org/wiki/ภาพดิจิทัล. 2009

[2] - . "Introduction to Digital Image Processing" [online].
Available: http://fivedots.coe.psu.ac.th/~montri/Teaching/image/chap1.pdf. 2009

[3] Kálmán Palágyi. "Skeletonization." [online].
Available: http://www.inf.u-szeged.hu/~palagyi/skel/skel.html. 2009

[4] - . "Distance Transform" [online].
Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/distance.htm 2010

[5] Prof. Godfried Toussaint. "Hilditch's Algorithm for Skeletonization" [online].
Available: http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton.html 2010

ภาคผนวก

ภาคผนวก ก

การทดสอบอัลกอริทึม

การทดสอบที่ 1 ทดสอบการทำงานของอัลกอริทึม full distance & transform

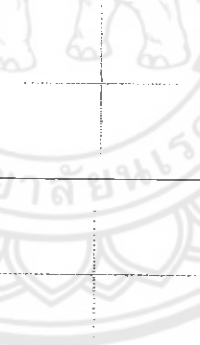
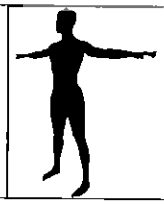
วัตถุประสงค์: เพื่อทดสอบการทำงานของอัลกอริทึมแบบ full distance & transform

รายละเอียด: ทำการทดสอบกับภาพ 4 แบบคือ ภาพรูปทรงพื้นฐานทางเรขาคณิต, ภาพตัวอักษรภาษาอังกฤษ, ภาพตัวเลขอารบิก, ภาพวัตถุทั่วไป และทดสอบบนระบบปฏิบัติการ Windows XP sp3

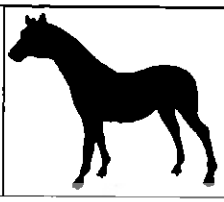
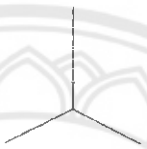
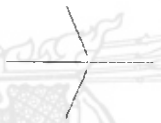
ผลที่คาดหวัง: ได้แกนของภาพซึ่งขาดคุณสมบัติ Topological และ Geometrical

ผลที่ได้จริง: ได้ผลตามที่คาดหวังไว้

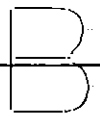
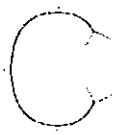
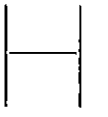
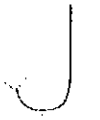
ตารางที่ ก-1 ตารางผลการทดสอบอัลกอริทึม full distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	No
		No	No
		Yes	No
		No	No
		No	Yes
		No	Yes
		No	Yes
		No	Yes

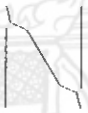
ตารางที่ ก-1 (ต่อ) ตารางผลการทดสอบอัลกอริทึม full distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		No	Yes
		No	Yes
		No	Yes
		No	No
		No	No
		No	Yes
		No	Yes
		No	No
		No	No

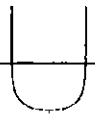
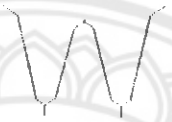
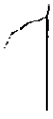
ตารางที่ ก-1 (ต่อ) ตารางผลการทดสอบอัลกอริทึม full distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
B		No	No
C		No	Yes
D		No	No
E		No	No
F		No	No
G		No	Yes
H		No	No
I		Yes	No
J		No	Yes

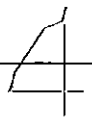
ตารางที่ ก-1 (ต่อ) ตารางผลการทดสอบอัลกอริทึม full distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
K		No	No
L		No	No
M		No	No
N		No	No
O		No	Yes
P		No	No
Q		No	Yes
R		No	No
S		No	Yes

ตารางที่ ก-1 (ต่อ) ตารางผลการทดสอบอัลกอริทึม full distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
U		No	No
V		No	No
W		No	No
X		No	No
Y		No	No
Z		No	No
1		No	No
2		No	Yes
3		No	Yes

ตารางที่ ก-1 (ต่อ) ตารางผลการทดสอบอัลกอริทึม full distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
4		No	No
5		No	Yes
6		No	Yes
7		No	No
8		No	Yes
9		No	Yes

การทดสอบที่ 2 ทดสอบการทำงานของอัลกอริทึม diagonal distance & transform

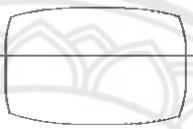
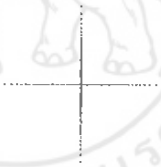
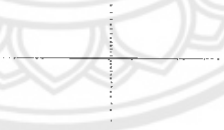
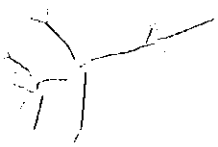
วัตถุประสงค์: เพื่อทดสอบการทำงานของอัลกอริทึมแบบ diagonal distance & transform

รายละเอียด: ทำการทดสอบกับภาพ 4 แบบคือ ภาพรูปทรงพื้นฐานทางเรขาคณิต, ภาพตัวอักษรภาษาอังกฤษ, ภาพตัวเลขอารบิก, ภาพวัตถุทั่วไป และทดสอบบนระบบปฏิบัติการ Windows XP sp3

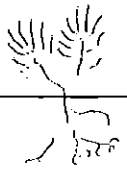
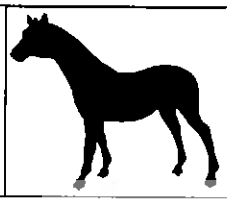
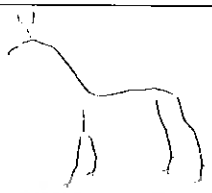
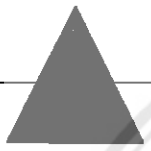
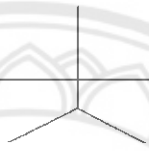
ผลที่คาดหวัง: ได้แกนของภาพซึ่งขาดคุณสมบัติ Topological และ Geometrical

ผลที่ได้จริง: ได้ผลตามที่คาดหวังไว้

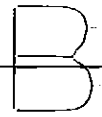
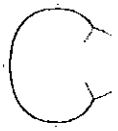
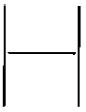
ตารางที่ ก-2 ตารางผลการทดสอบอัลกอริทึม diagoanl distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	No
		No	No
		Yes	No
		No	No
		No	Yes
		No	Yes
		No	Yes
		No	Yes

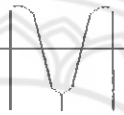
ตารางที่ ก-2 (ต่อ) ตารางผลการทดสอบอัลกอริทึม diagoanl distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		No	Yes
		No	Yes
		No	Yes
		No	No
		No	No
		No	Yes
		No	Yes
		No	No
		No	No

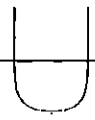
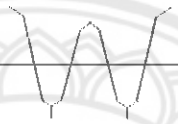
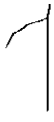
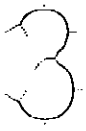
ตารางที่ ก-2 (ต่อ) ตารางผลการทดสอบอัลกอริทึม diagoanl distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
B		No	No
C		No	Yes
D		No	No
E		No	No
F		No	No
G		No	Yes
H		No	No
I		Yes	No
J		No	Yes

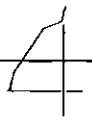
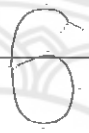
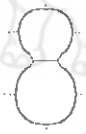
ตารางที่ ก-2 (ต่อ) ตารางผลการทดสอบอัลกอริทึม diagoanl distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
K		No	No
L		No	No
M		No	No
N		No	No
O		No	Yes
P		No	No
Q		No	Yes
R		No	No
S		No	Yes

ตารางที่ ก-2 (ต่อ) ตารางผลการทดสอบอัลกอริทึม diagoanl distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
U		No	No
V		No	No
W		No	No
X		No	No
Y		No	No
Z		No	No
1		No	No
2		No	Yes
3		No	Yes

ตารางที่ ก-2 (ต่อ) ตารางผลการทดสอบอัลกอริทึม diagoanl distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
4		No	No
5		No	Yes
6		No	Yes
7		No	No
8		No	Yes
9		No	Yes

การทดสอบที่ 3 ทดสอบการทำงานของอัลกอริทึม cross distance & transform

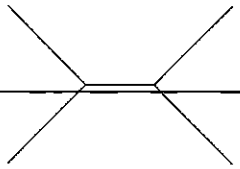
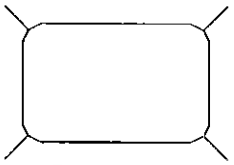
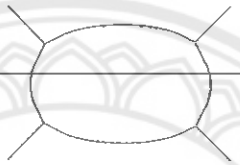
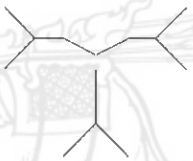
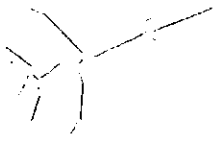
วัตถุประสงค์: เพื่อทดสอบการทำงานของอัลกอริทึมแบบ cross distance & transform

รายละเอียด: ทำการทดสอบกับภาพ 4 แบบคือ ภาพรูปทรงพื้นฐานทางเรขาคณิต, ภาพตัวอักษรภาษาอังกฤษ, ภาพตัวเลขอารบิก, ภาพวัตถุทั่วไป และทดสอบบนระบบปฏิบัติการ Windows XP sp3

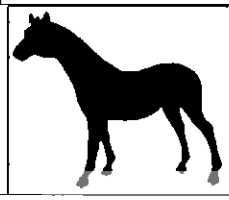
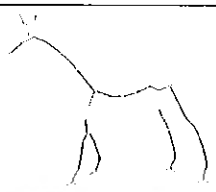
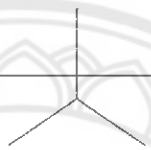
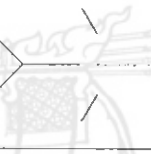
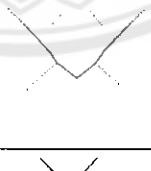
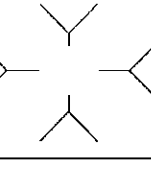
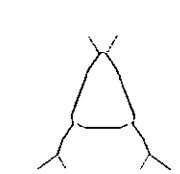
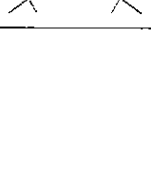
ผลที่คาดหวัง: ได้แกนของภาพซึ่งมีคุณสมบัติ Topological แต่ขาดคุณสมบัติ Geometrical

ผลที่ได้จริง: ได้ผลตามที่คาดหวังไว้

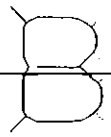
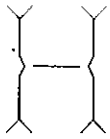
ตารางที่ ก-3 ตารางผลการทดสอบอัลกอริทึม cross distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	Yes
		Yes	Yes
		Yes	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes

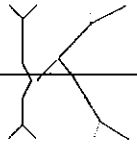
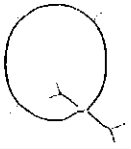
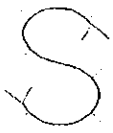
ตารางที่ ก-3 (ต่อ) ตารางผลการทดสอบอัลกอริทึม cross distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes

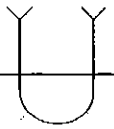
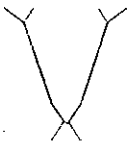
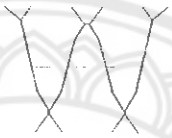
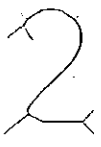
ตารางที่ ก-3 (ต่อ) ตารางผลการทดสอบอัลกอริทึม cross distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
B		No	Yes
C		No	Yes
D		No	Yes
E		No	Yes
F		No	Yes
G		No	Yes
H		No	Yes
I		Yes	Yes
J		No	Yes

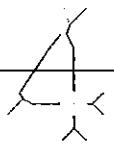
ตารางที่ ก-3 (ต่อ) ตารางผลการทดสอบอัลกอริทึม cross distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
K		No	Yes
L		Yes	Yes
M		No	Yes
N		No	Yes
O		No	Yes
P		No	Yes
Q		No	Yes
R		No	Yes
S		No	Yes

ตารางที่ ก-3 (ต่อ) ตารางผลการทดสอบอัลกอริทึม cross distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
U		No	Yes
V		No	Yes
W		No	Yes
X		No	Yes
Y		No	Yes
Z		No	Yes
1		No	Yes
2		No	Yes
3		No	Yes

ตารางที่ ก-3 (ต่อ) ตารางผลการทดสอบอัลกอริทึม cross distance & transform

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
4		No	Yes
5		No	Yes
6		No	Yes
7		No	Yes
8		No	Yes
9		No	Yes

การทดสอบที่ 4 ทดสอบการทำงานของอัลกอริทึม thinning

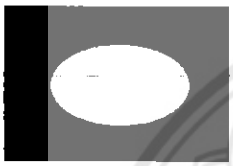
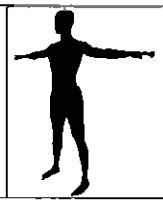
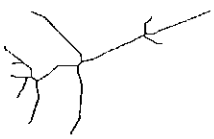
วัตถุประสงค์: เพื่อทดสอบการทำงานของอัลกอริทึมแบบ thinning

รายละเอียด: ทำการทดสอบกับภาพ 4 แบบคือ ภาพรูปทรงพื้นฐานทางเรขาคณิต, ภาพตัวอักษรภาษาอังกฤษ, ภาพตัวเลขอารบิก, ภาพวัตถุทั่วไป และทดสอบบนระบบปฏิบัติการ Windows XP sp3

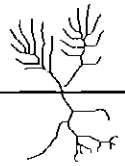
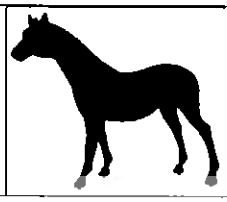
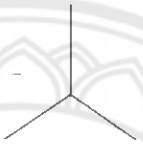
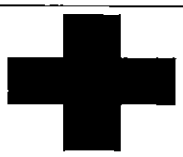
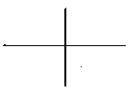
ผลที่คาดหวัง: ได้แกนของภาพซึ่งมีคุณสมบัติ Geometrical แต่ขาดคุณสมบัติ Topological

ผลที่ได้จริง: ได้ผลตามที่คาดหวังไว้

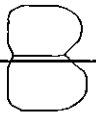
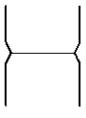
ตารางที่ ก-4 ตารางผลการทดสอบอัลกอริทึม thinning

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	No
		Yes	No
		Yes	No
		Yes	No
		Yes	No
		Yes	No
		Yes	Yes
		Yes	Yes

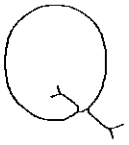
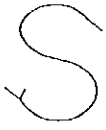
ตารางที่ ก-4 (ต่อ) ตารางผลการทดสอบอัลกอริทึม thinning

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	Yes
		Yes	Yes
		Yes	Yes
		Yes	No
		Yes	No
		Yes	Yes
		Yes	No
		Yes	No
		Yes	No

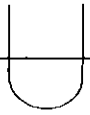
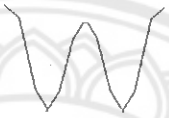
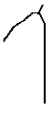
ตารางที่ ก-4 (ต่อ) ตารางผลการทดสอบอัลกอริทึม thinning

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
B		Yes	No
C		Yes	Yes
D		Yes	No
E		Yes	No
F		Yes	No
G		Yes	Yes
H		Yes	No
I		Yes	No
J		Yes	Yes

ตารางที่ ก-4 (ต่อ) ตารางผลการทดสอบอัลกอริทึม thinning

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
K		Yes	No
L		Yes	No
M		Yes	No
N		Yes	No
O		Yes	Yes
P		Yes	No
Q		Yes	Yes
R		Yes	No
S		Yes	Yes

ตารางที่ ก-4 (ต่อ) ตารางผลการทดสอบอัลกอริทึม thinning

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
U		Yes	No
V		Yes	No
W		Yes	No
X		Yes	No
Y		Yes	No
Z		Yes	No
1		Yes	No
2		Yes	Yes
3		Yes	Yes

ตารางที่ ก-4 (ต่อ) ตารางผลการทดสอบอัลกอริทึม thinning

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
4		Yes	No
5		Yes	Yes
6		Yes	Yes
7		Yes	No
8		Yes	Yes
9		Yes	Yes

การทดสอบที่ 5 ทดสอบการทำงานของอัลกอริทึม Voronoi

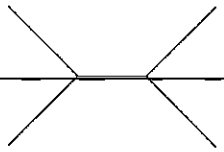
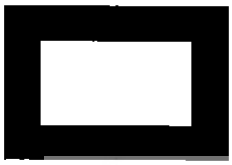
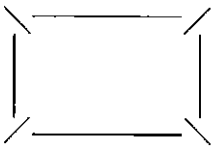
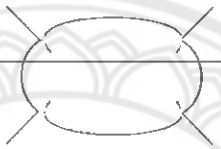
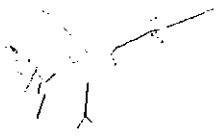
วัตถุประสงค์: เพื่อทดสอบการทำงานของอัลกอริทึมแบบ Voronoi

รายละเอียด: ทำการทดสอบกับภาพ 4 แบบคือ ภาพรูปทรงพื้นฐานทางเรขาคณิต, ภาพตัวอักษรภาษาอังกฤษ, ภาพตัวเลขอารบิก, ภาพวัตถุทั่วไป และทดสอบบนระบบปฏิบัติการ Windows XP sp3

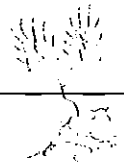
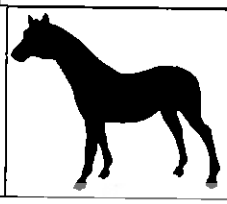
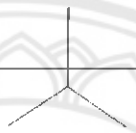
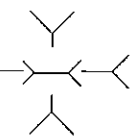
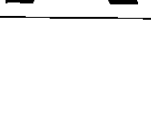
ผลที่คาดหวัง: ได้แกนของภาพซึ่งมีคุณสมบัติ Geometrical แต่ขาดคุณสมบัติ Topological

ผลที่ได้จริง: ได้ผลตามที่คาดหวังไว้

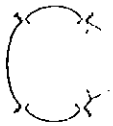
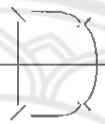
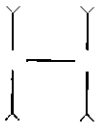
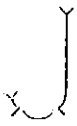
ตารางที่ ก-5 ตารางผลการทดสอบอัลกอริทึม voronoi

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	Yes
		No	Yes
		No	Yes
		No	Yes
		Yes	Yes
		Yes	Yes
		No	Yes
		No	Yes

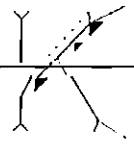
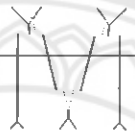
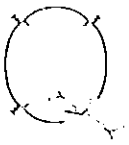
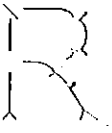
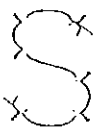
ตารางที่ ก-5 (ต่อ) ตารางผลการทดสอบอัลกอริทึม voronoi

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes
		No	Yes

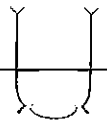
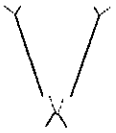
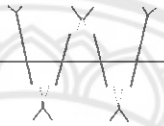
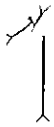
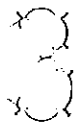
ตารางที่ ก-5 (ต่อ) ตารางผลการทดสอบอัลกอริทึม Voronoi

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
B		No	Yes
C		No	Yes
D		No	Yes
E		No	Yes
F		No	Yes
G		No	Yes
H		No	Yes
I		Yes	Yes
J		No	Yes

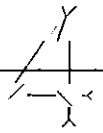
ตารางที่ ก-5 (ต่อ) ตารางผลการทดสอบอัลกอริทึม Voronoi

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
K		No	Yes
L		No	Yes
M		No	Yes
N		No	Yes
O		No	Yes
P		No	Yes
Q		No	Yes
R		No	Yes
S		No	Yes

ตารางที่ ก-5 (ต่อ) ตารางผลการทดสอบอัลกอริทึม Voronoi

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
U		No	Yes
V		No	Yes
W		No	Yes
X		No	Yes
Y		No	Yes
Z		No	Yes
1		No	Yes
2		No	Yes
3		No	Yes

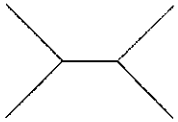
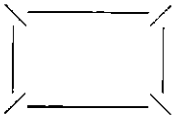
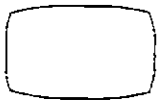
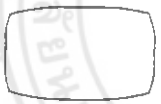
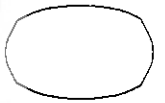
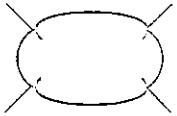
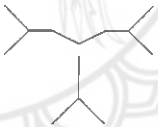
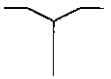
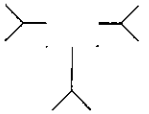
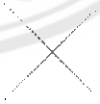
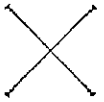
ตารางที่ ก-5 (ต่อ) ตารางผลการทดสอบอัลกอริทึม Voronoi

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
4		No	Yes
5		No	Yes
6		No	Yes
7		No	Yes
8		No	Yes
9		No	Yes

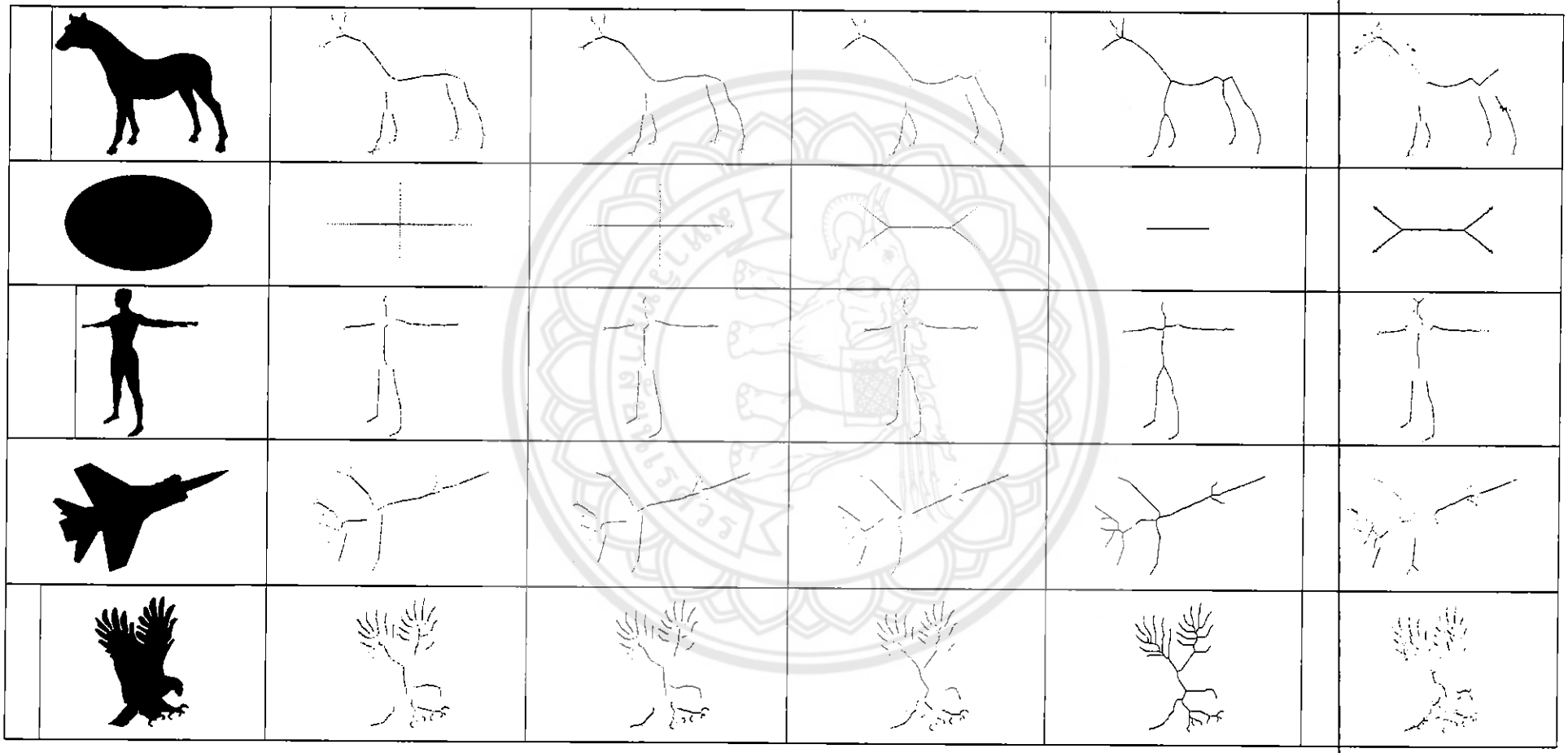
ภาคผนวก ข

การเปรียบเทียบการทำงานแต่ละอัลกอริทึม

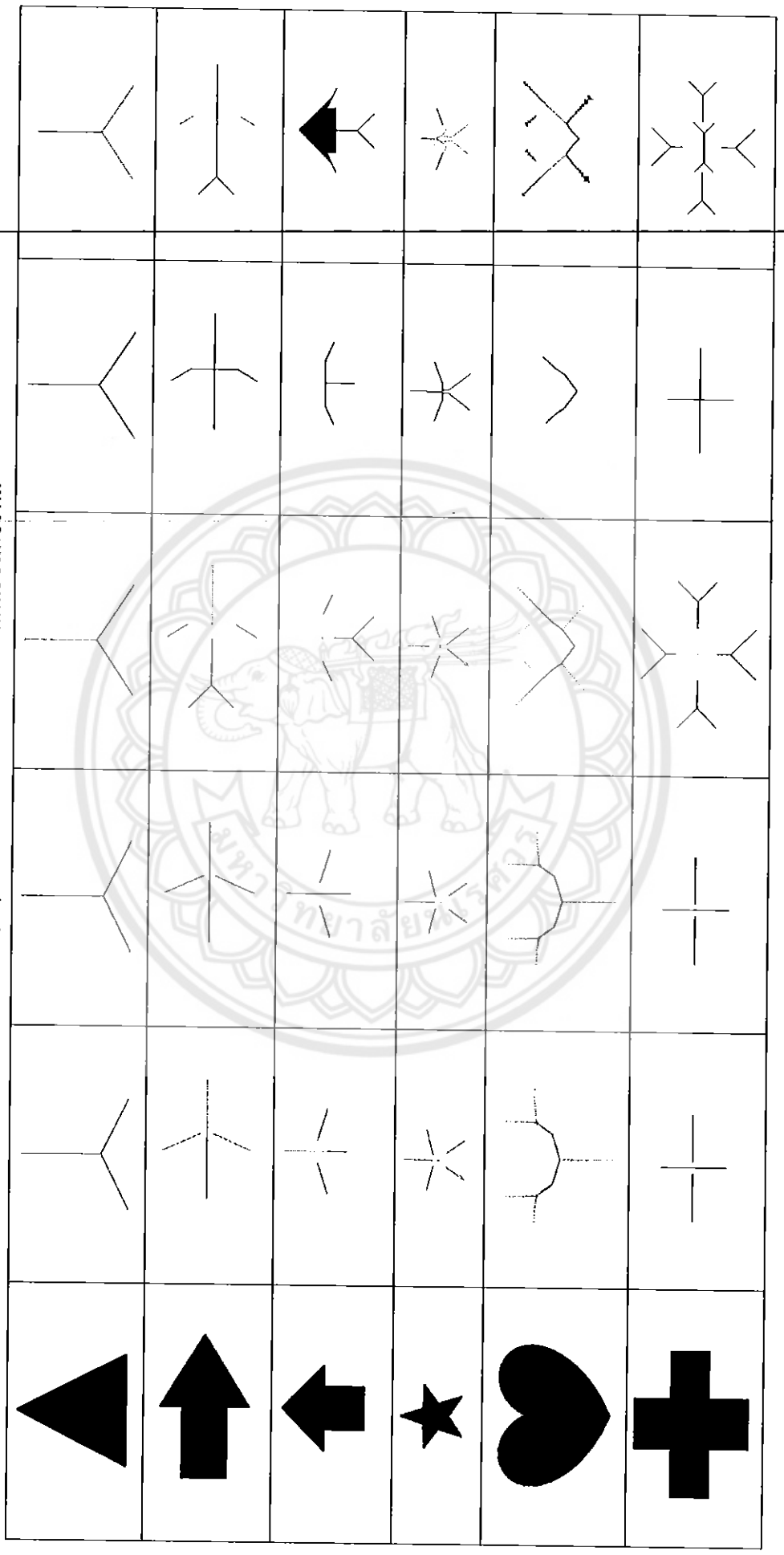
ตารางที่ ข-1 ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม

ภาพต้นแบบ	Full distance & transform	Diagonal distance & transform	Cross distance & transform	Thinning	Voronoi
					
					
					
					
					

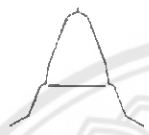
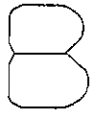
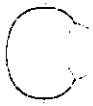
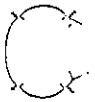
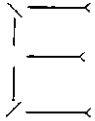
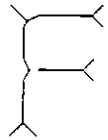
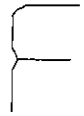
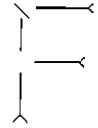
ตารางที่ ข-1 (ต่อ) ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม



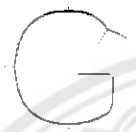
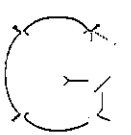
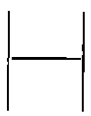
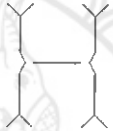
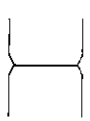
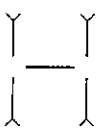
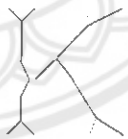
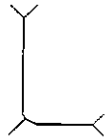
ตารางที่ ข-1 (ต่อ) ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม



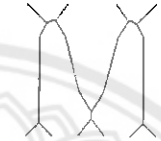
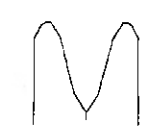
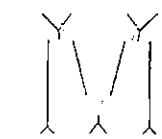
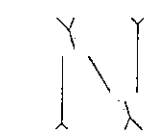
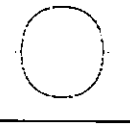
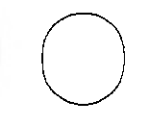
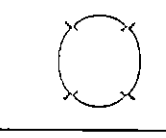
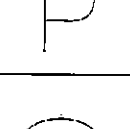
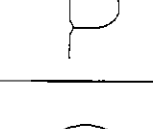
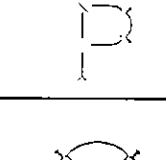
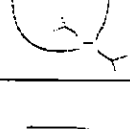
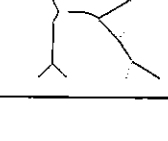
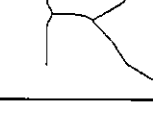
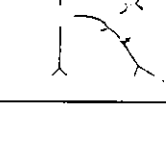
ตารางที่ ข-1 (ต่อ) ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม

A					
B					
C					
D					
E					
F					

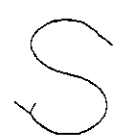
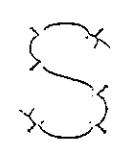
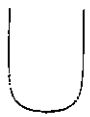
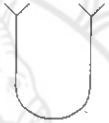
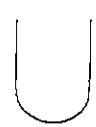
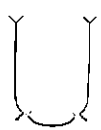
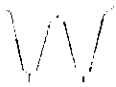
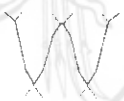
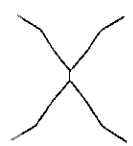
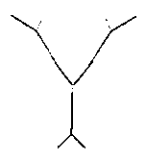
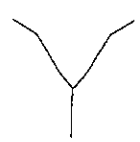
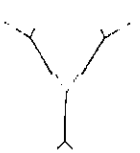
ตารางที่ ข-1 (ต่อ) ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม

G					
H					
I					
J					
K					
L					

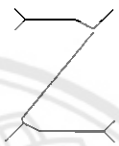
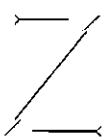
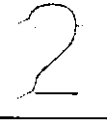
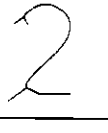
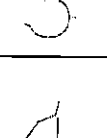
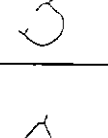
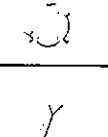
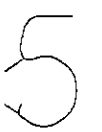
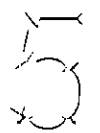
ตารางที่ ข-1 (ต่อ) ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม

M					
N					
O					
P					
Q					
R					

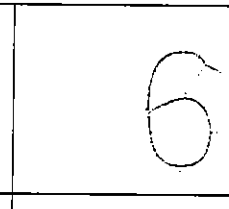
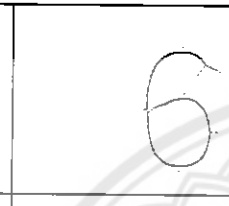
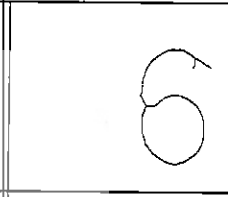
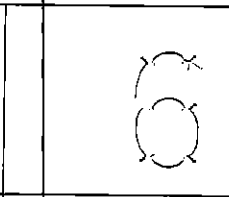
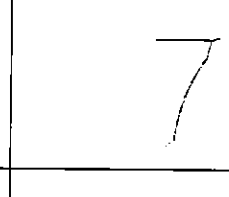
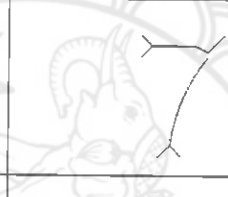
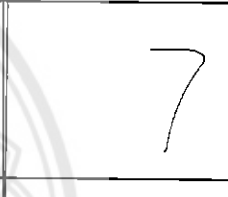
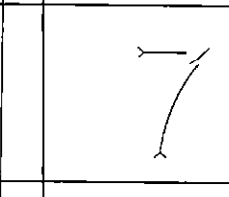
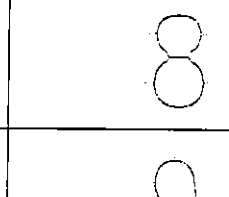
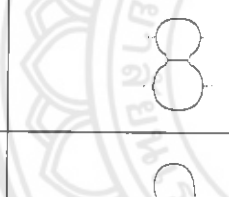
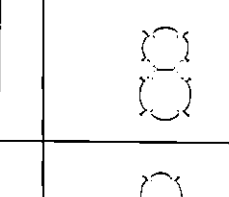
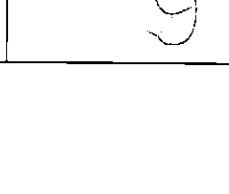
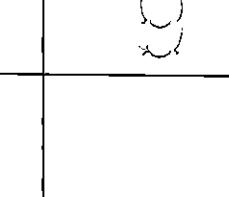
ตารางที่ ข-1 (ต่อ) ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม

S					
U					
V					
W					
X					
Y					

ตารางที่ ข-1 (ต่อ) ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม

Z					
1					
2					
3					
4					
5					

ตารางที่ ข-1 (ต่อ) ตารางผลการเปรียบเทียบผลแต่ละอัลกอริทึม

6					
7					
8					
9					

ประวัติผู้เขียนโครงงาน

ชื่อ นายปริญญา วิชาวงษ์

ภูมิลำเนา 246 ม.6 ต. คณฑี อ.เมือง จ.กำแพงเพชร

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนคณฑีพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : parinya_w@windowslive.com , conan_engi@hotmail.com