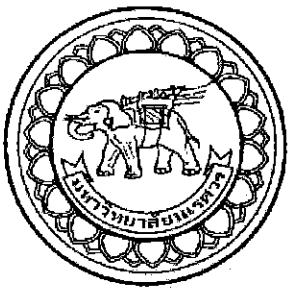
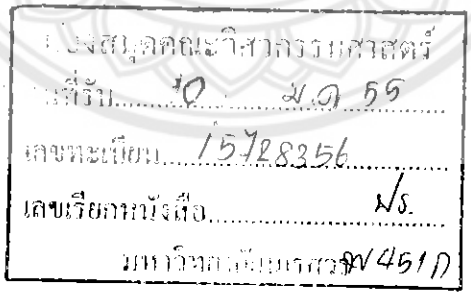


การแสดงโครงสร้างของรูป 3 มิติ

Visualizing the skeletonized 3D images

นางสาวพลอยไพลิน ไหววิจิตร รหัส 49361263

นางสาววนารัตน์ จุฬพันธ์ทอง รหัส 49361652



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การแสดงโครงสร้างของรูป 3 มิติ
ผู้ดำเนินโครงงาน	นางสาวพลอยไพลิน ไหววิจิตร รหัส 49361263
	นางสาววนารัตน์ จุฬพันธ์ทอง รหัส 49361652
อาจารย์ที่ปรึกษา	ดร.สุรเดช จิตประไพกุลศาล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2552

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ

(รองศาสตราจารย์ ดร.ไพศาล มุณีสว่าง)

.....กรรมการ

(ดร.วรลักษณ์ คงเด่นฟ้า)

หัวข้อโครงงาน	การแสดงโครงสร้างของรูป 3 มิติ
ผู้ดำเนินโครงงาน	นางสาวพลอยไพลิน ไหววิจิตร รหัส 49361263
	นางสาววนารัตน์ จุฬพันธ์ทอง รหัส 49361652
อาจารย์ที่ปรึกษา	ดร.สุรเดช จิตประไพกุลศาล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2552

.....

บทคัดย่อ

โครงงานนี้ให้ความรู้เรื่องการประมวลผลภาพดิจิทัลเพื่อแสดงโครงสร้างของรูป 3 มิติ ซึ่งผู้ศึกษาใช้ภาษาจาว่าในการพัฒนาโปรแกรม โดยมี algorithm ที่ใช้หาโครงสร้างของภาพทั้งหมด 3 algorithm ได้แก่ Distance & Transform, Thinning และ Voronoi จากการทดสอบปรากฏว่าโปรแกรมสามารถหาโครงสร้างของภาพออกมาได้ทั้ง 3 algorithm โดยแต่ละ algorithm นั้น จะมีข้อดีข้อเสียแตกต่างกันไป เช่น algorithm แบบ Thinning จะได้ภาพผลลัพธ์ที่มีคุณสมบัติ topological แต่จะขาดคุณสมบัติ geometrical ส่วน algorithm แบบ Distance & Transform จะได้ภาพผลลัพธ์ที่มีคุณสมบัติ geometrical แต่จะขาดคุณสมบัติ topological และ algorithm แบบ Voronoi จะได้ภาพผลลัพธ์ที่มีทั้งคุณสมบัติ topological และ geometrical

Project Title	Visualizing the skeletonized 3D images
Name	Miss Phloiphilin Waiwichit ID. 49361263
	Miss Wanarat Juraphanthong ID. 49361652
Project Advisor	Suradet Jitprapaikulsarn, Ph.D.
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2009

Abstract

This project use digital image processing to determine the skeleton of 3D images. The program was develop in Java programming language. Three algorithms have used: 1) Distance & Transform, 2) Thinning and 3) Voronoi. From our observation three algorithms produce skeletons with different properties. The skeletons from the Thinning algorithm have topological property but lack geometrical property. The skeletons from the Distance & Transform algorithm have geometrical property but lack topological property. The skeletons from the Voronoi algorithm have topological property and geometrical property.

กิตติกรรมประกาศ

ในการทำโครงงานวิศวกรรมครั้งนี้ คณะผู้จัดทำขอกราบขอบพระคุณบิดา มารดาที่คอยสนับสนุนและให้กำลังใจตลอดมา และขอกราบขอบพระคุณ ดร.สุรเดช จิตประไพกุลศาล ที่ได้ให้คำปรึกษาโครงงานนี้ ทั้งทฤษฎี แนวคิด ตลอดจนขั้นตอนการปฏิบัติงานต่างๆ และขอกราบขอบพระคุณ ผศ.ดร.ไพศาล มุณีสว่าง และ ดร.วรลักษณ์ คงเด่นฟ้า ที่ให้คำแนะนำและติชม เพื่อแก้ไขในส่วนของโปรแกรมและรายงานโครงงาน ตลอดจนเสียสละเวลาเพื่อทำการตรวจสอบการทำงานและชี้แนวทางในการแก้ไขปัญหาโครงงานนี้

และทางคณะผู้จัดทำใคร่ขอกราบอภัยบุคคลที่มีส่วนเกี่ยวข้องในการทำโครงงานนี้แต่ไม่ได้กล่าวนามและใคร่ขอขอบพระคุณ มา ณ ที่นี้ด้วย

นางสาวพลอยไพลิน ไหววิจิตร

นางสาววนารัตน์ จุฬพันธ์ทอง

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ก

บทคัดย่อภาษาอังกฤษ.....ข

กิตติกรรมประกาศ.....ค

สารบัญ.....ง

สารบัญตาราง.....ฉ

สารบัญรูป.....ช

บทที่ 1 บทนำ.....1

1.1 ที่มาและความสำคัญ.....1

1.2 วัตถุประสงค์.....1

1.3 ขอบข่าย.....1

1.4 ขั้นตอนดำเนินการ.....2

1.5 แผนการดำเนินงาน.....2

1.6 ผลที่คาดว่าจะได้รับ.....3

1.7 งบประมาณ.....3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....4

2.1 Skeletonization [2].....4

2.2 Java Swing [8].....16

บทที่ 3 ขั้นตอนและการดำเนินงาน.....22

3.1 การจัดการกับรูปภาพก่อนการประมวลผล.....23

สารบัญ(ต่อ)

หน้า

3.2 การประมวลผลภาพ 23

3.3 การจัดการกับรูปภาพหลังการประมวลผล 29

บทที่ 4 การออกแบบซอฟต์แวร์ 30

4.1 คุณสมบัติของซอฟต์แวร์ 30

4.2 ขั้นตอนการประมวลผล 30

4.3 ขอบเขตของระบบ 31

4.4 การออกแบบซอฟต์แวร์ 31

บทที่ 5 ผลการทดสอบ 38

5.1 ทดสอบการทำงานแต่ละ algorithm 38

5.2 ผลการเปรียบเทียบประสิทธิภาพการทำงาน 45

5.3 ทดสอบว่าสามารถแสดงผลแบบ GUI ได้ 46

บทที่ 6 สรุปผล 47

6.1 สรุปผลการทดลอง 47

6.2 ปัญหาที่พบ 48

6.3 ข้อเสนอแนะ 48

เอกสารอ้างอิง 49

ประวัติผู้เขียนโครงงาน 50

สารบัญตาราง

ตารางที่		หน้า
1.1	ขั้นตอนการดำเนินโครงงาน	2
2.1	การเปรียบเทียบเทคนิค skeletonization	7
5.1	ตารางแสดงผลการทดสอบ algorithm Distance & Transform – Plus กับรูปภาพต่างๆ	39
5.2	ตารางแสดงผลการทดสอบ algorithm Distance & Transform – Multiply กับรูปภาพต่างๆ	40
5.3	ตารางแสดงผลการทดสอบ algorithm Distance & Transform – Full กับรูปภาพต่างๆ	41
5.4	ตารางแสดงผลการทดสอบ algorithm Distance & Transform – L กับรูปภาพต่างๆ	42
5.5	ตารางแสดงผลการทดสอบ algorithm Voronoi กับรูปภาพต่างๆ	43
5.6	ตารางแสดงผลการทดสอบ algorithm Thinning กับรูปภาพต่างๆ	44
5.7	ตารางเปรียบเทียบประสิทธิภาพการทำงานของแต่ละ algorithm	45
6.1	ตารางเปรียบเทียบประสิทธิภาพการทำงานของแต่ละ algorithm	48

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างของรูปสี่เหลี่ยม: ทั้งจุด A และ B เป็นโครงสร้าง แต่จุด C ไม่ได้เป็นส่วนหนึ่งของโครงสร้าง (โครงสร้างถูกแสดงด้วยเส้นสีดำหนา)..... 4

2.2 โครงสร้างของกล่องตัน: โครงสร้างใน 3 มิติจะบรรจุแผ่นพื้นผิว (รูปเดิม (ซ้าย) และโครงสร้าง (ขวา))..... 5

2.3 โครงสร้างที่เหมือนกันอาจมาจากวัตถุที่แตกต่างกัน..... 5

2.4 ความเสถียรภาพ: รูปสี่เหลี่ยมที่ขอบเขตถูกรบกวน..... 5

2.5 โครงสร้างแสดงสัดส่วนของวัตถุโดยเฉพาะและโครงสร้าง topological ของวัตถุ..... 6

2.6 โครงสร้างถูกแสดงด้วยสี่เหลี่ยมสีชมพู (ซ้าย) และแผนที่ระยะทางจาก 4 จุดใกล้เคียง (ขวา)..... 7

2.7 แผนที่ระยะทางจากกระดานหมากรุก หรือ 8 จุดใกล้เคียง (ซ้าย) และแผนที่ระยะทางจาก (3,4)-chamfer (ขวา)..... 8

2.8 ตัวอย่าง distance transformationรูป binary เดิม (ซ้าย) และแผนที่ระยะทาง (ขวา) (แผนที่ระยะทางแสดงพื้นผิวที่เป็นจุดสันตามโครงสร้าง)..... 8

2.9 ตัวอย่างของ Voronoi diagrams10 จุดกำเนิด (ซ้าย) และ Voronoi diagram (ขวา)..... 9

2.10 Voronoi diagrams สามารถคำนวณได้ตามโครงสร้างที่เพิ่มขึ้นหากความหนาแน่นของจุดขอบเขตไปรวมกันตรงไหนมากที่สุดใน Voronoi diagram นั่นคือ skeleton..... 9

2.11 จุดขอบเขตของรูปสี่เหลี่ยมซึ่ง set เป็นจุดกำเนิด..... 10

2.12 โครงสร้าง (แสดงด้วยเส้นสีแดง) ถูกประมาณด้วย subgraph ของ Voronoi diagram..... 10

2.13 ตัวอย่างของกระบวนการ Thinning เซลล์ที่เข้มที่สุดเป็นเซลล์ที่ได้จากการคำนวณ skeleton..... 11

2.14 ตัวอย่างของกระบวนการลดรูป 2 มิติ ที่ไม่รักษา topology..... 12

สารบัญรูป(ต่อ)

รูปที่		หน้า
2.15	ตัวอย่างการทำการลดรูป 3 มิติ ซึ่งไม่รักษา topology รูปเดิม (ซ้าย) และผลลัพธ์ของการลดรูป (ขวา)	12
2.16	ตัวอย่างของ Thinning ที่รักษารูปร่าง รูปเดิม (บน) และผลลัพธ์ของ Thinning (ล่าง) ข้อความยังสามารถอ่านได้	13
2.17	ตัวอย่างของกระบวนการ Thinning ที่แตกต่างกัน รูปเดิม (ซ้ายบน) และพื้นผิวแกนกลาง (ขวาบน) เส้นแกนกลาง (ซ้ายล่าง) และแก่น topological (ขวาล่าง)	14
2.18	ส่วนของอวัยวะมนุษย์: รูปเดิม (ซ้าย) และเส้นแกนกลาง (ขวา)	15
2.19	ม้าหมากรุก (ซ้าย) และ พื้นผิวแกนกลาง (ขวา)	15
3.1	แผนภาพการแสดงโครงสร้างรูป 3 มิติ	22
3.2	การเก็บข้อมูลจุดภาพให้อยู่ในรูปแบบของ array 3 มิติ	23
3.3	ตำแหน่งรอบข้างที่ใช้พิจารณา	24
3.4	ตำแหน่งพื้นฐานใน class A, B, C, D	26
3.5	ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class A	27
3.6	ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class B	27
3.7	ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class C	27
3.8	ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class D	28
3.9	ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class modified D	28
3.10	การแปลงข้อมูล array 3 มิติให้อยู่ในรูปแบบของรูปภาพ 3 มิติ	29
4.1	ขั้นตอนการประมวลผลของระบบ	30
4.2	มุมมองในการออกแบบซอฟต์แวร์	31

สารบัญรูป(ต่อ)

รูปที่		หน้า
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	หน้าต่างซอฟต์แวร์การแสดงผลโครงสร้างของรูป 3 มิติ	46

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันการวิเคราะห์งานทางด้านโครงสร้างนั้นได้เข้ามามีบทบาทในชีวิตประจำวันมากมายหลากหลายรูปแบบ อาทิเช่น การวิเคราะห์โครงสร้างของอาคาร เพื่อสร้างแกนกลางหาจุดสมดุล เป็นเครื่องมือช่วยวิเคราะห์งานทางด้านการแพทย์ สนับสนุนช่วยเหลือการผ่าตัดให้มีความถูกต้อง เหมาะสมและปลอดภัยต่อผลกระทบข้างเคียง ซึ่งอาจกล่าวได้ว่าสิ่งนี้ได้กลายเป็นอีกหนึ่งองค์ประกอบสำคัญที่ใช้ในการวางแผนการทำงานกับบุคคลหลากหลายอาชีพทั้งทางตรงและทางอ้อม ด้วยตระหนักและเห็นความสำคัญในจุดนี้ โครงงานนี้จึงได้ทำการศึกษาและพัฒนาซอฟต์แวร์การแสดงโครงสร้างของรูป 3 มิติ ให้มีประสิทธิภาพ มีความถูกต้องและเหมาะสมกับการเลือกใช้งานให้มากที่สุด

1.2 วัตถุประสงค์

1.2.1 เพื่อพัฒนาซอฟต์แวร์ที่สามารถคำนวณหาโครงสร้างจากรูป 3 มิติได้

1.3 ขอบข่าย

โปรแกรมที่พัฒนาขึ้นจะสามารถทำงานได้ดังนี้

1.3.1 สามารถประมวลผลได้บนระบบปฏิบัติงาน Windows Xp ได้

1.3.2 สามารถแสดงผลแบบ GUI ได้

1.3.3 สามารถแปลงรูป 3 มิติให้แสดงผลในรูปแบบโครงสร้างของรูปได้

1.4 ขั้นตอนดำเนินการ

1.4.1 กำหนดขอบเขตของโครงงาน

1.4.2 วางแผนการทำงาน

1.4.3 ศึกษาข้อมูลและทำความเข้าใจเกี่ยวกับเนื้อหาและขั้นตอนในการแสดงโครงสร้างของรูป 3 มิติ

1.4.4 ออกแบบขั้นตอนการพัฒนาซอฟต์แวร์

1.4.5 พัฒนาซอฟต์แวร์

1.4.6 ทดสอบและปรับปรุงแก้ไขการทำงานของซอฟต์แวร์

1.4.7 จัดทำเอกสารและคู่มือการใช้งาน

1.4.8 ส่งโครงงานฉบับสมบูรณ์

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 ขั้นตอนการดำเนินโครงงาน

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. 2553		ก.พ. 2553		มี.ค. 2553	
	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
1. กำหนดขอบเขตของโครงงาน	←	→																		
2. วางแผนการทำงาน			←	→																
3. ศึกษาข้อมูลและทำความเข้าใจเกี่ยวกับเนื้อหาและขั้นตอนในการแสดงโครงสร้างของรูป 3 มิติ					←			→												
4. ออกแบบขั้นตอนการพัฒนาซอฟต์แวร์							←		→											
5. พัฒนาซอฟต์แวร์									←				→							
6. ทดสอบและปรับปรุงแก้ไขการทำงานของซอฟต์แวร์													←		→					

ตารางที่ 1.1(ต่อ)

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. 2553		ก.พ. 2553		มี.ค 2553	
	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
7. จัดทำเอกสารและคู่มือการใช้งาน																	↔	↔		
8. ส่งโครงงานฉบับสมบูรณ์																			↔	

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 มีความรู้ความเข้าใจในเรื่องการแสดงผลโครงสร้างของรูป 3 มิติ

1.6.2 สามารถพัฒนาซอฟต์แวร์การแสดงโครงสร้างของรูป 3 มิติได้

1.7 งบประมาณ

1.7.1 ค่าพิมพ์เอกสารและถ่ายเอกสาร	1000	บาท
1.7.2 ค่าจัดทำรูปเล่ม	1000	บาท
รวมเป็นเงิน	2000	บาท

<u>หมายเหตุ</u> ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 Skeletonization [2]

2.1.1 แนะนำ

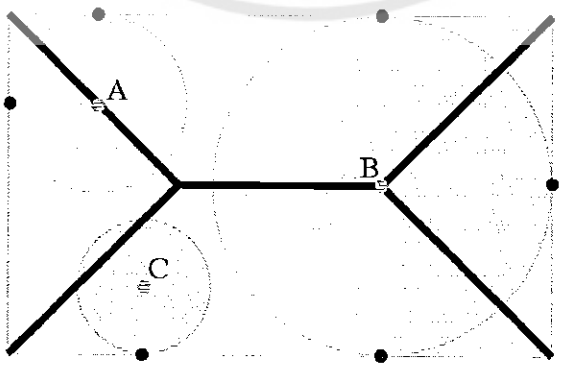
จุดมุ่งหมายของการหาโครงสร้าง คือ การหาลักษณะรูปร่างพื้นฐานเดิมของวัตถุ

2.1.2 การหาโครงสร้างและคุณสมบัติ

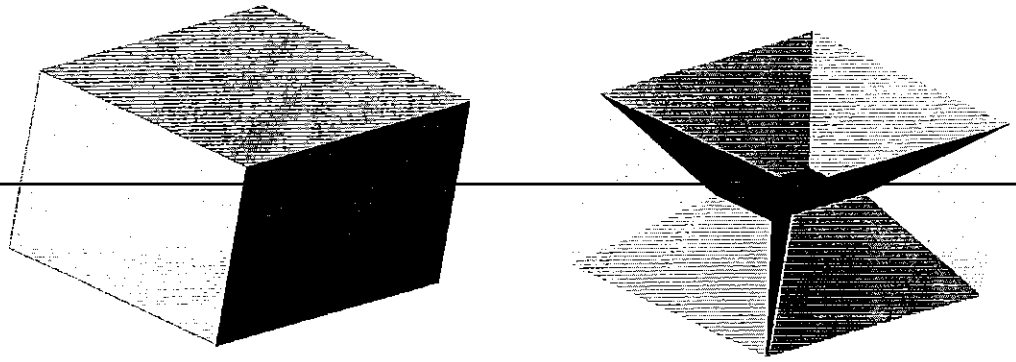
ความคิด skeleton ริเริ่มโดย H. Blum เป็นผลลัพธ์ของการแปลงรูปแกนกลาง (Medial Axis Transform: MAT) หรือ การแปลงสัดส่วนแกน (Symmetry Axis Transform: SAT) การหา MAT หาจากจุดขอบเขตใกล้ชิดของแต่ละจุดที่อยู่ในวัตถุ จุดชั้นในเป็นส่วนหนึ่งของ skeleton หากมีอย่างน้อย 2 จุดขอบเขตใกล้ชิด

นิยามมากมายของการหาโครงสร้างถูกเปรียบเทียบกับการเผาไหม้ของไฟ ขอบเขตของวัตถุถูกสมมติเป็นไฟ และตัวโครงสร้างคือจุดที่ไฟวิ่งมาพบกันและดับลง

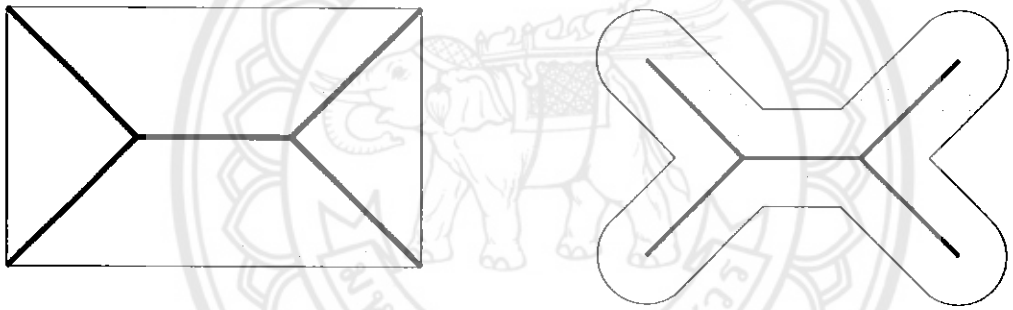
โครงสร้างเป็นตำแหน่งของศูนย์กลางของการเขียนวงกลมที่กว้างที่สุด (ตัวอย่าง แผ่นดิสก์ และลูกบอล ใน 2 มิติและ 3 มิติ ตามลำดับ) การเขียนวงกลมจะกว้างที่สุดหากไม่มีวงกลมอื่นมาครอบมันได้



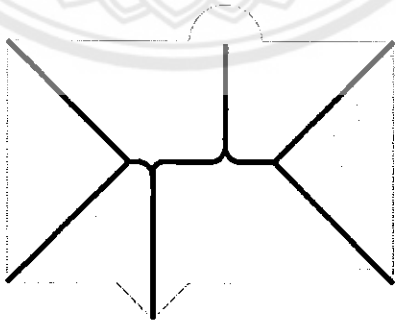
รูปที่ 2.1 โครงสร้างของรูปสี่เหลี่ยม: ทั้งจุด A และ B เป็นโครงสร้าง แต่จุด C ไม่ได้เป็นส่วนหนึ่งของโครงสร้าง (โครงสร้างถูกแสดงด้วยเส้นสีดำหนา)



รูปที่ 2.2 โครงสร้างของกล่องตัน: โครงสร้างใน 3 มิติจะบรรจุแผ่นพื้นผิว (รูปเดิม (ซ้าย) และโครงสร้าง (ขวา))



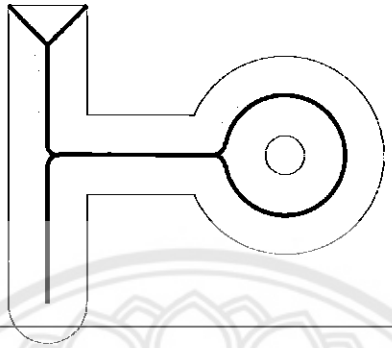
รูปที่ 2.3 โครงสร้างที่เหมือนกันอาจมาจากวัตถุที่แตกต่างกัน



รูปที่ 2.4 ความเสถียรภาพ: รูปสี่เหลี่ยมที่ขอบเขตถูกรบกวน

การเติมคุณสมบัติสำคัญ

โครงสร้างมีความบาง ซึ่งบรรจุจุดที่มีจำนวนน้อยกว่าวัตถุเดิมมาก (วัตถุประสงค์เพื่อลดรูปวัตถุ)



รูปที่ 2.5 โครงสร้างแสดงสัดส่วนของวัตถุโดยเฉพาะและโครงสร้าง topological ของวัตถุ

2.1.3 เทคนิค Skeletonization

การหาโครงสร้าง (ตัวอย่าง โครงสร้างจากรูปภาพ digital binary) ให้ลักษณะรูปร่างพื้นฐานเดิม ซึ่งเป็นกระบวนการเริ่มแรกในการเปลี่ยนแปลง raster-to-vector หรือ การจำตัวอย่าง

เทคนิค skeletonization มีอยู่ 3 ประเภท ดังนี้

- ค้นหาแนวสันในแผนที่ระยะทางของจุดขอบเขต
- คำนวณ Voronoi diagram ที่เกิดจากจุดขอบเขต
- หาชั้น ด้วยการกัดเซาะชั้น เรียกว่า Thinning

ในพื้นที่ digital การประมาณโครงสร้างที่แท้จริงเท่านั้นที่สามารถทำได้ โดยมี 2 ปัจจัยที่ควรพิจารณา ได้แก่

- Topological (การรักษา topology ของรูปเดิม โดยไม่มีการฉีกขาด)
- Geometrical (การบีบโครงสร้างให้อยู่ตรงกลางวัตถุ และแตกแขนงโครงสร้างไปตามรูปทรงเรขาคณิต)

ตารางที่ 2.1 การเปรียบเทียบเทคนิค skeletonization

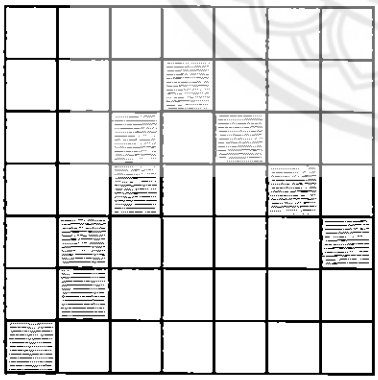
method	geometrical	topological
Distance & Transform	yes	no
Voronoi	yes	yes
Thinning	no	yes

2.1.3.1 Distance transformation

พื้นฐานการหาโครงสร้างด้วยวิธี distance transformation มี 3 ขั้นตอน ดังนี้

- รูปเดิม (binary) ถูกแปลงเข้ามาเป็น element คุณลักษณะ element เป็นไปตามขอบเขตวัตถุ
- แผนที่ระยะทางถูกสร้างขึ้น ซึ่งแต่ละ element จะให้ระยะทางที่ใกล้ element ที่สุด
- แนวสัน (ที่ตั้งปลายสุด) เป็นจุดโครงสร้าง

ผลลัพธ์แผนที่ระยะทาง โดย distance transformation ขึ้นอยู่กับระยะทางที่เลือก



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 2.6 โครงสร้างถูกแสดงด้วยสี่เหลี่ยมสีชมพู (ซ้าย) และแผนที่ระยะทางจาก 4 จุดใกล้เคียง (ขวา)

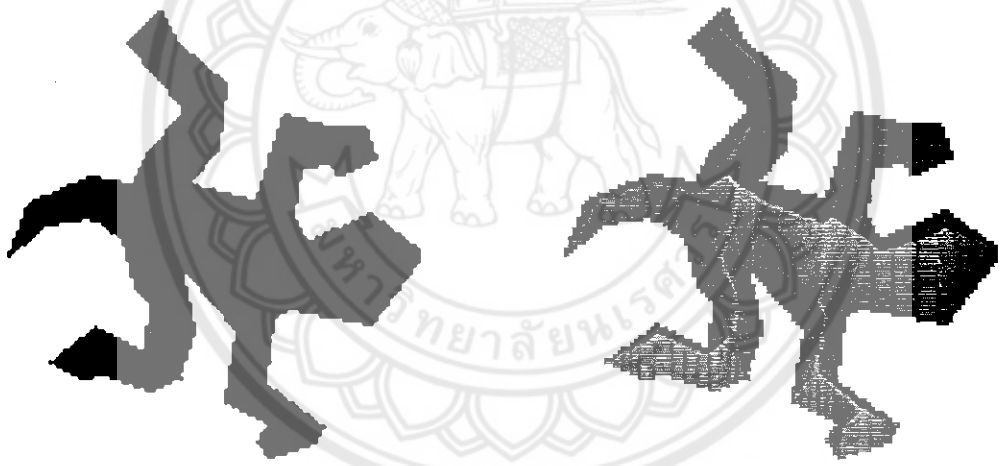
2	2	1	1	1	2	2
2	1	1	0	1	1	2
2	1	0	1	0	1	1
1	1	0	1	1	0	1
1	0	1	1	1	1	0
1	0	1	2	2	1	1
0	1	1	2	2	2	2

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 2.7 แผนที่ระยะทางจากกระดานหมากรุก หรือ 8 จุดใกล้เคียง (ซ้าย) และแผนที่ระยะทางจาก (3,4)-chamfer (ขวา)



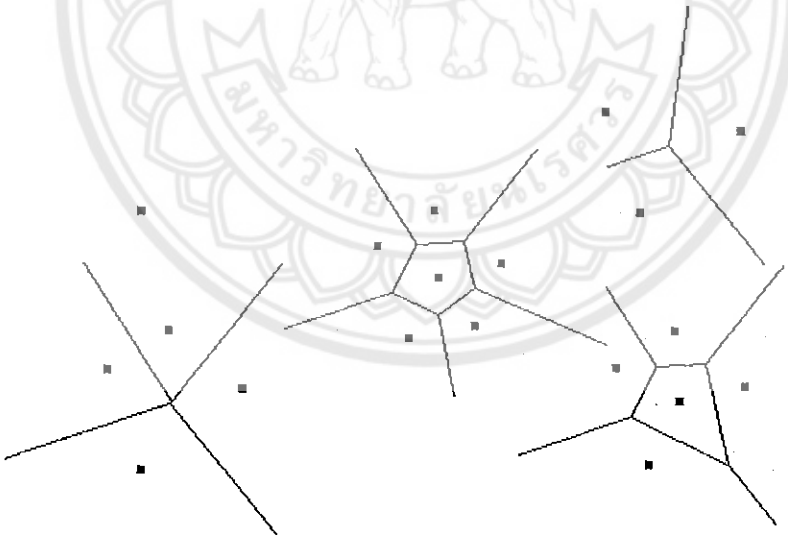
รูปที่ 2.8 ตัวอย่าง distance transformationรูป binary เดิม (ซ้าย) และแผนที่ระยะทาง (ขวา) (แผนที่ระยะทางแสดงพื้นผิวที่เป็นจุดสันตามโครงสร้าง)

Distance transformation สามารถหาได้ด้วยเวลาใกล้เคียง O(n) ในมิติที่ไม่มีขอบเขตจำกัด(ซึ่ง "n" คือจำนวน element รูปภาพ) วิธีนี้ตรงตามจุดประสงค์ของ geometrical แต่ไม่เป็นไปตามจุดประสงค์ของ topological

2.1.3.2 Voronoi diagram

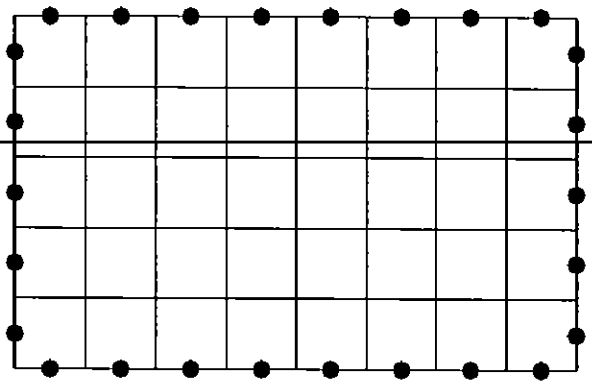
Voronoi diagram เป็นการแยกจุด (เรียกว่าจุดกำเนิด) เป็นเสมือนกำแพงกั้นพื้นที่ในเซลล์ ในแต่ละเซลล์จะบรรจุจุดกำเนิด 1 จุดและตำแหน่งของจุดอื่นๆ ซึ่งใกล้เคียงจุดกำเนิดนี้มากกว่าจุดกำเนิดอื่น

รูปที่ 2.9 ตัวอย่างของ Voronoi diagrams 10 จุดกำเนิด (ซ้าย) และ Voronoi diagram (ขวา)

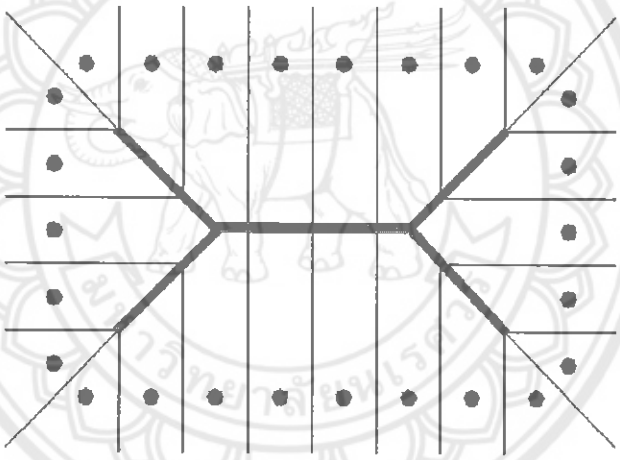


รูปที่ 2.10 Voronoi diagrams สามารถคำนวณได้ตามโครงสร้างที่เพิ่มขึ้นหากความหนาแน่นของจุดขอบเขตไปรวมกันตรงไหนมากที่สุดใน Voronoi diagram นั่นคือ skeleton

ตัวอย่างโครงสร้าง Voronoi



รูปที่ 2.11 จุดขอบเขตของรูปสี่เหลี่ยมซึ่ง set เป็นจุดกำเนิด

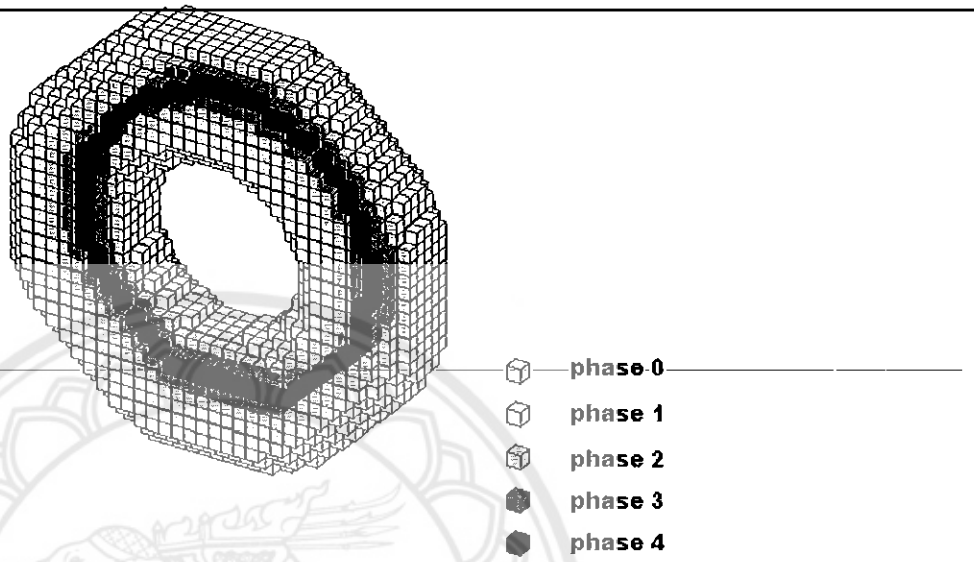


รูปที่ 2.12 โครงสร้าง (แสดงด้วยเส้นสีแดง) ถูกประมาณด้วย subgraph ของ Voronoi diagram

Voronoi diagram เป็นไปตามจุดประสงค์ของ topological และ geometrical ทั้งคู่ แต่มันเป็นกระบวนการที่สิ้นเปลือง โดยเฉพาะวัตถุที่ใหญ่และซับซ้อน

2.1.3.3 Thinning

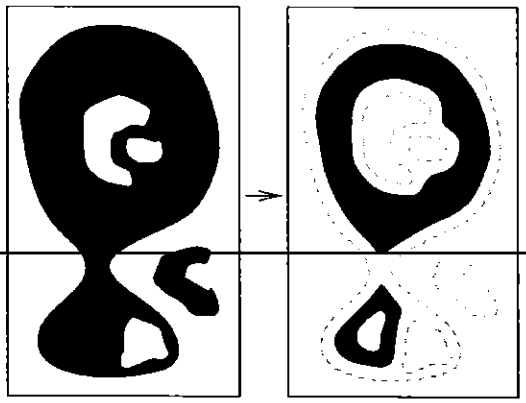
มันเป็นเทคนิคการลดรูปวัตถุซ้ำเรื่อยๆ สำหรับแบบจำลองการแพร่ไฟในพื้นที่ digital



รูปที่ 2.13 ตัวอย่างของกระบวนการ Thinning เซลล์ที่เข้มที่สุดเป็นเซลล์ที่ได้จากการคำนวณ skeleton

คุณสมบัติที่เป็นประโยชน์ของ Thinning

- รักษา topology (รักษา topology ของวัตถุเดิม)
- รักษารูปร่าง (ลักษณะสำคัญ ซึ่งทำให้จำวัตถุนั้นได้)
- บีบโครงสร้างให้อยู่ตรงกลางวัตถุ
- ก่อให้เกิดโครงสร้างที่มีความกว้าง 1 เซลล์

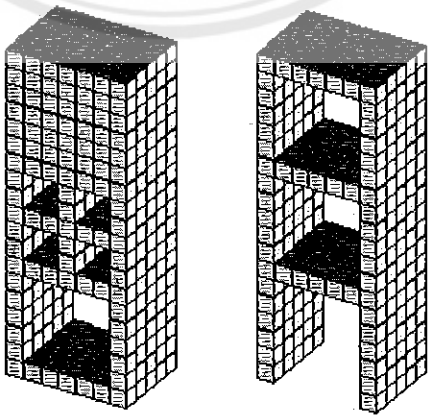


รูปที่ 2.14 ตัวอย่างของกระบวนการลดรูป 2 มิติ ที่ไม่รักษา topology

มันไม่รักษา topology เมื่อ

- วัตถุไม่ต่อเนื่องกัน
- วัตถุถูกลบหมด
- โพรงถูกสร้างขึ้น (สีขาวที่ถูกล้อมด้วยวัตถุ)
- โพรงถูกกลืนหายไปกับพื้นหลัง
- โพรงถูกรวมเข้าด้วยกัน

มีอีก 1 ความคิดที่ถูกเรียกว่า รู ในภาพ 3 มิติ รู (ลักษณะเหมือนโดนัท) ถูกแทนด้วยจุดสีขาว แต่ไม่ใช่โพรง การรักษา topology ถูกกำจัดหรือสร้างรูที่ไม่ได้รับการยอมรับ

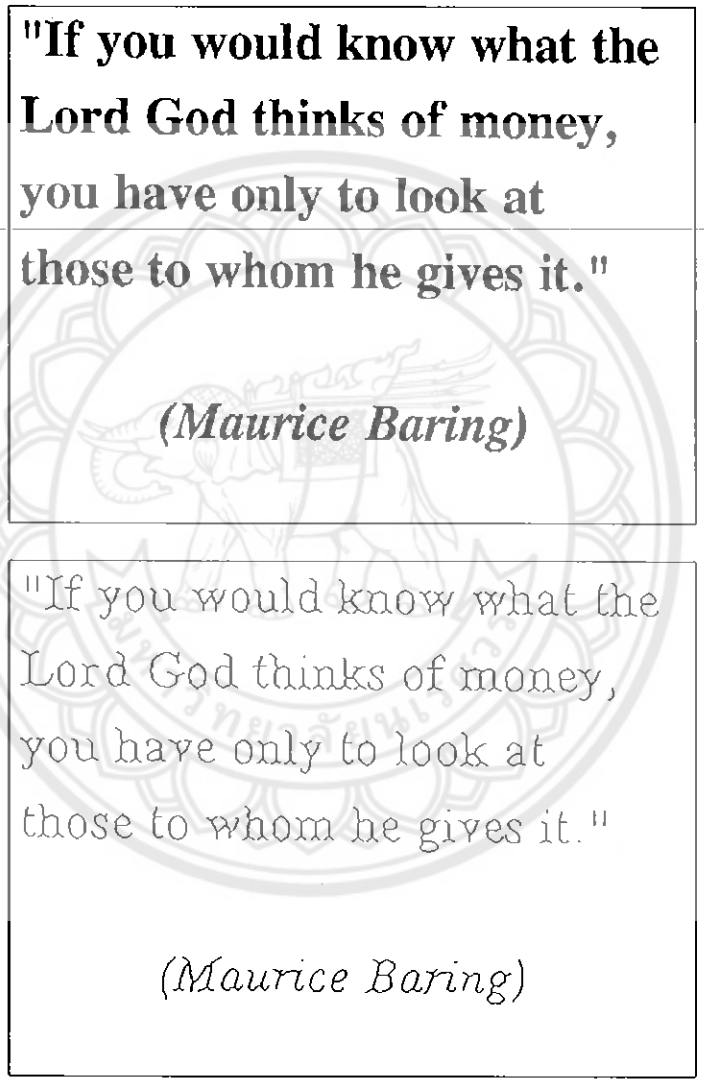


รูปที่ 2.15 ตัวอย่างการทำการลดรูป 3 มิติ ซึ่งไม่รักษา topology

รูปเดิม (ซ้าย) และผลลัพธ์ของการลดรูป (ขวา)

มันไม่รักษา topology เมื่อ

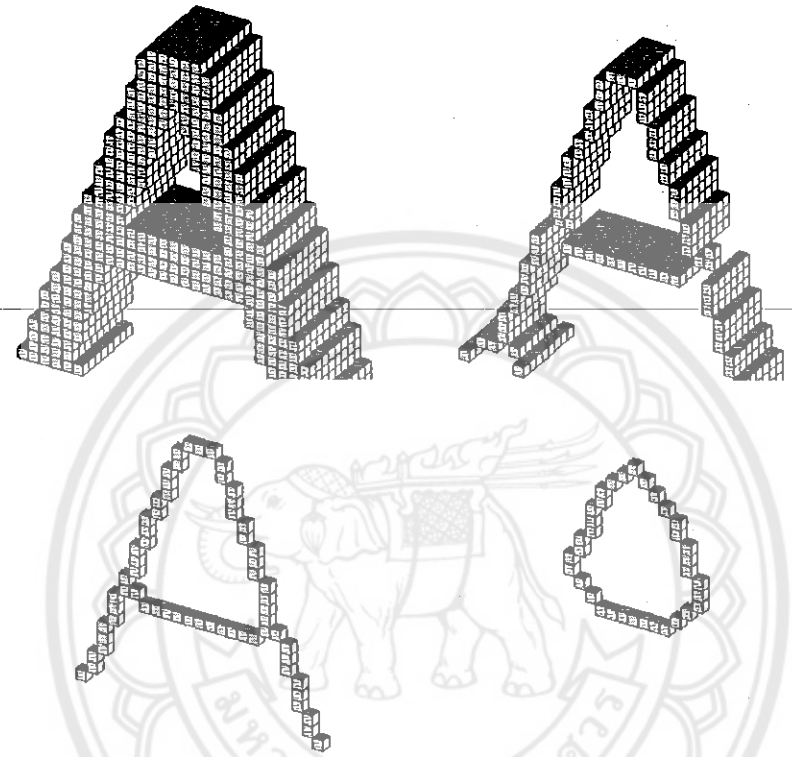
- รูถูกสร้าง
- รูถูกกำจัด
- 4 รูถูกรวมเข้าด้วยกัน



รูปที่ 2.16 ตัวอย่างของ Thinning ที่รักษารูปร่าง

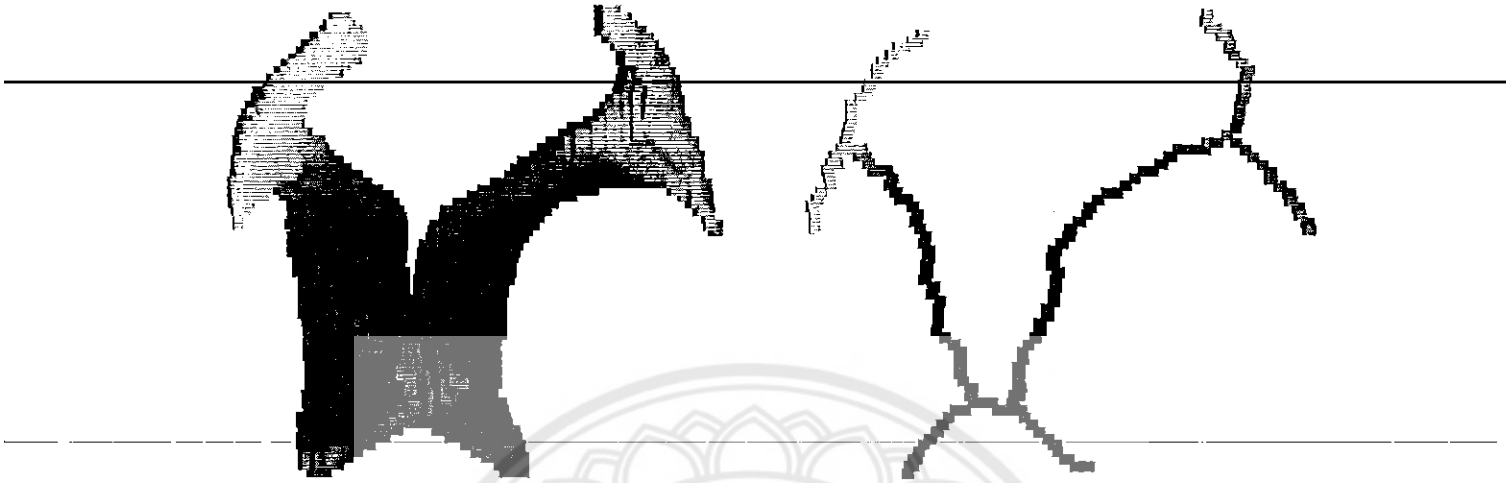
รูปเดิม (บน) และผลลัพธ์ของ Thinning (ล่าง) ข้อความยังสามารถอ่านได้

ผลผลิตของ skeleton 3 มิติ ขึ้นอยู่กับการเลือกจุดสุดท้ายที่จะรักษาไว้: curve thinning รักษาจุดสุดท้าย surface thinning ไม่ลบพื้นผิดจุดสุดท้าย และ shrinking (รักษาแก่น topology ของวัตถุ ไม่พิจารณาจุดสุดท้าย)

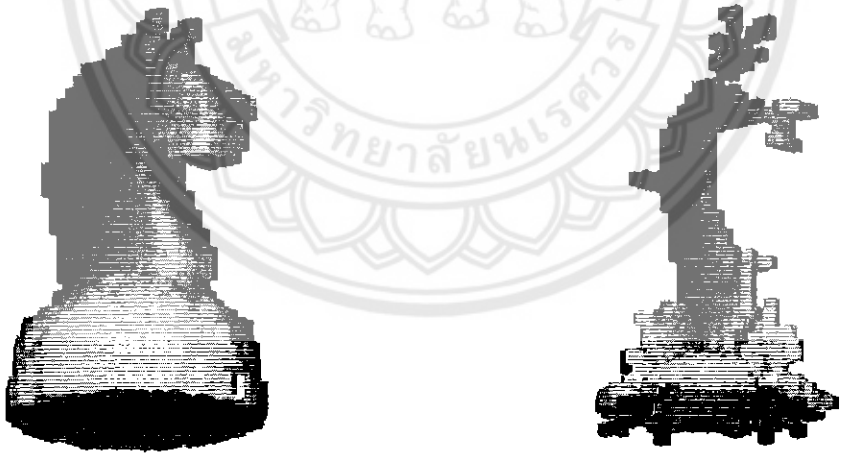


รูปที่ 2.17 ตัวอย่างของกระบวนการ Thinning ที่แตกต่างกันรูปเดิม (ซ้ายบน) และพื้นผิวแกนกลาง (ขวาบน) เส้นแกนกลาง (ซ้ายล่าง) และแก่น topological (ขวาล่าง)

ผลลัพธ์ของ Thinning 3 มิติ



รูปที่ 2.18 ส่วนของอวัยวะมนุษย์: รูปเดิม (ซ้าย) และเส้นแกนกลาง (ขวา)



รูปที่ 2.19 ม้าหมากรุก (ซ้าย) และ พื้นผิวแกนกลาง (ขวา)

2.2 Java Swing [8]

AWT (Abstract Windowing Toolkit) เป็นเครื่องมือสร้าง Graphic User Interface (GUI) และกำหนดส่วนประกอบ (Component) ต่างๆให้กับ GUI โดยภาษา Java ซึ่งจะประกอบไปด้วย 3 ส่วนคือ

1. Components เช่น ปุ่ม (button) กล่องรับข้อความ (textfield) ข้อความป้าย (label) เป็นต้น
2. Containers เป็น Component ใหญ่ที่จะจัดเก็บ Component ย่อยอื่นๆ ตัวอย่าง Containers เช่น Applet, Container, Dialog, Frame, Panel, Scrollane เป็นต้น
3. Layout Managers เป็นการจัดเรียง Component ที่อยู่ใน Containers เช่น FlowLayout, BorderLayout, GridLayout, CardLayout และ GridBagLayout

แต่ Component ต่างๆ ใน AWT มีข้อจำกัดบางอย่างดังนั้นจึงมีการพัฒนา Package ใหม่ชื่อ Swing ขึ้นมาสนับสนุนของเดิมให้ดีขึ้น เราจะใช้ Component ต่างๆใน Swing ซึ่งเป็น Class ที่ถูกจัดเก็บไว้ใน Package "javax.swing" เช่น

Swing Component class	Component
javax.swing.JButton	Button
javax.swing.JCheckbox	Checkbox
javax.swing.JDialog	Dialog
javax.swing.JFrame	Frame
javax.swing.JLabel	Label
javax.swing.JMenu	Menu
javax.swing.JPanel	Panel
javax.swing.JScrollbar	Scrollbar
javax.swing.JScrollPane	Scroll pane
javax.swing.JTextArea	Text Area
javax.swing.JTextField	Text Field
javax.swing.JComboBox	Combo Box
javax.swing.JList	List
javax.swing.JPasswordField	Password Filed
javax.swing.JTabbedPane	Tab

2.2.1 ขั้นตอนการสร้าง GUI

2.2.1.1 สร้าง Object ของ Class Containers และกำหนดพื้นที่ในการแสดงผลด้วย method getContentPane()

รูปแบบ Containers *Object1* = getContentPane() ;

2.2.1.2 สร้าง Object ของ Class Component ใน Swing

รูปแบบ *JComponent Object2* = new *JComponent*() ;

*Component เปียนแทน button, label ,list เป็นต้น

2.2.1.3 เพิ่ม Component ลงใน Container โดยใช้ method add()

รูปแบบ *Object1.add(Object2)* ;

2.2.1.4 กำหนดการจัดเรียง Component ใน Container

รูปแบบ *Object1*.setLayout (*LayoutManager*) ;

* *LayoutManager* เป็น Object ของ Class LayoutManager เช่น FlowLayout , BorderLayout , GridLayout, CardLayout และ GridBagLayout

2.2.2 Swing Component

2.2.2.1 JButton

รูปแบบการสร้าง Object ใน class JButton

```
JButton namebutton = new JButton("Label");
```

2.2.2.2 JLabel

รูปแบบการสร้าง Object ใน class JLabel

```
JLabel nameLabel = new JLabel("Label");
```

2.2.2.3 JCheckBox

รูปแบบการสร้าง Object ใน class JCheckBox

```
JCheckbox namecheckbox = new JCheckbox("Label",Group,Checkedstatus );
```

**Group คือการจัดกลุ่มของ checkbox ถ้าไม่มีกลุ่มให้เป็น nul*

***CheckedStatus เป็นการบอกสถานะว่ามีการเช็คที่ช่องหรือป่าว ถ้าเป็น true ก็จะเช็ค ถ้าเป็น false ก็จะไม่เช็ค*

2.2.2.4 JRadioButton

รูปแบบการสร้าง Object ใน class JRadioButton

```
JRadioButton namebutton = new JRadioButton("Label",status);
```

รูปแบบการจัดกลุ่มของ ButtonGroup

```
ButtonGroup nameGroup = new ButtonGroup;
```

รูปแบบการนำ JRadioButton ลงไปในกลุ่ม ButtonGroup

```
nameGroup.add(namebutton);
```

2.2.2.5 JComboBox

รูปแบบการสร้าง Object ใน class JComboBox

```
JComboBox nameCombo = new JComboBox(ListData[]);
```

2.2.2.6 JList

รูปแบบการสร้าง Object ใน class JList

```
JList namelist = new JList(ListData[] );
```

2.2.2.7 JTextField

รูปแบบการสร้าง Object ใน class JTextField

```
JTextField nameTextField = new JTextField("Label",size );
```

2.2.2.8 JTabbedPane

รูปแบบการสร้าง Object ใน class JTabbedPane

```
JTabbedPane nameTabbedPane = new JTabbedPane();
```

วิธีใช้ *nameTabbedPane.add("nametab",Component);*

2.2.3 Layout Manager

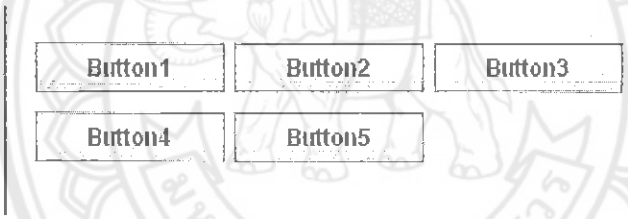
2.2.3.1 FlowLayout เป็นการเรียง Component แบบง่าย โดยจะเรียงจากซ้ายมือไปขวามือตามแนวนอน พอหมดบรรทัดก็จะขึ้นใหม่ ปกติถ้าไม่กำหนดอะไร (defaults) Layout แบบ FlowLayout จะกำหนดระยะห่างระหว่าง Component เป็นระยะห่าง 5 พิกเซล และจะกำหนดให้ Component ทั้งหมดอยู่ตรงกลาง แต่สามารถปรับระยะห่าง หรือตำแหน่งของ Component

```
FlowLayout(align, int HorizontalSpacing,int VerticalSpacing);
```

*align คือตำแหน่งที่จะวาง component เช่น right ,left ,center

* HorizontalSpacing คือค่าระยะห่างระหว่าง Component แนวนอน

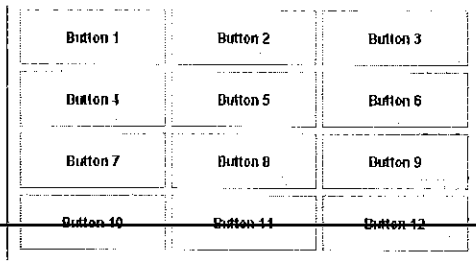
* VerticalSpacing คือค่าระยะห่างระหว่าง Component แนวตั้ง



รูปที่ 2.20 ตัวอย่าง FlowLayout(FlowLayout.LEFT, 5, 10);

2.2.3.2 GridLayout เป็นการเรียง Component ในรูปแบบตาราง โดยจะจัดเรียง component ในลักษณะ row กับ column โดยจะแบ่งเท่าๆกันหมดทุก cell และจะเรียง component จากซ้ายมือไปขวามือตามแนวนอน พอหมดบรรทัดก็จะขึ้นใหม่ ในการจัดเรียงแบบนี้จะต้องกำหนด row และ column ก่อน(ถ้าไม่กำหนด default จะให้เป็น 1 component ต่อ 1column) อีกทั้งยังสามารถปรับระยะห่าง หรือตำแหน่งของ Component ได้

```
GridLayout(int row,int column,int HorizontalSpacing,int VerticalSpacing);
```



รูปที่ 2.21 ตัวอย่าง GridLayout(4,3,5,5);

2.2.3.3 BorderLayout เป็นการเรียง Component บน Container ที่ถูกแบ่งเป็น 5 ส่วน ตามทิศต่างๆ เช่น North , South , West , East ,Center เมื่อต้องการจะนำ Component ไปลงบน Container จะต้องกำหนด ทิศทางด้วย(จะทำต้อง Add Component นั้นเอง)

วิธีใช้ ***container.add(Component,Borderlayout.position)***

Position คือ North , South , West , East ,Center

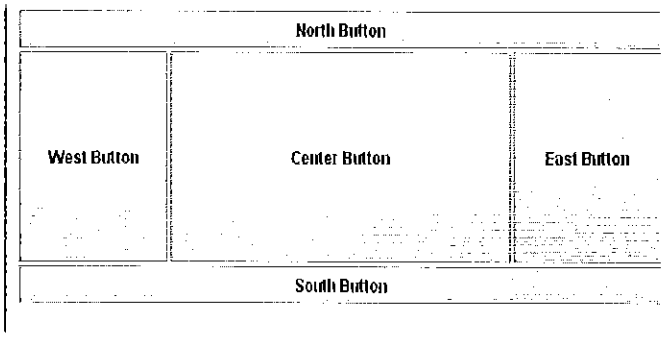
```
BorderLayout(int HorizontalSpacing,int VerticalSpacing);
```

ตัวอย่าง container.add(Northbutton,Borderlayout.North);

container.add(Centerbutton,Borderlayout.Center);

.

.



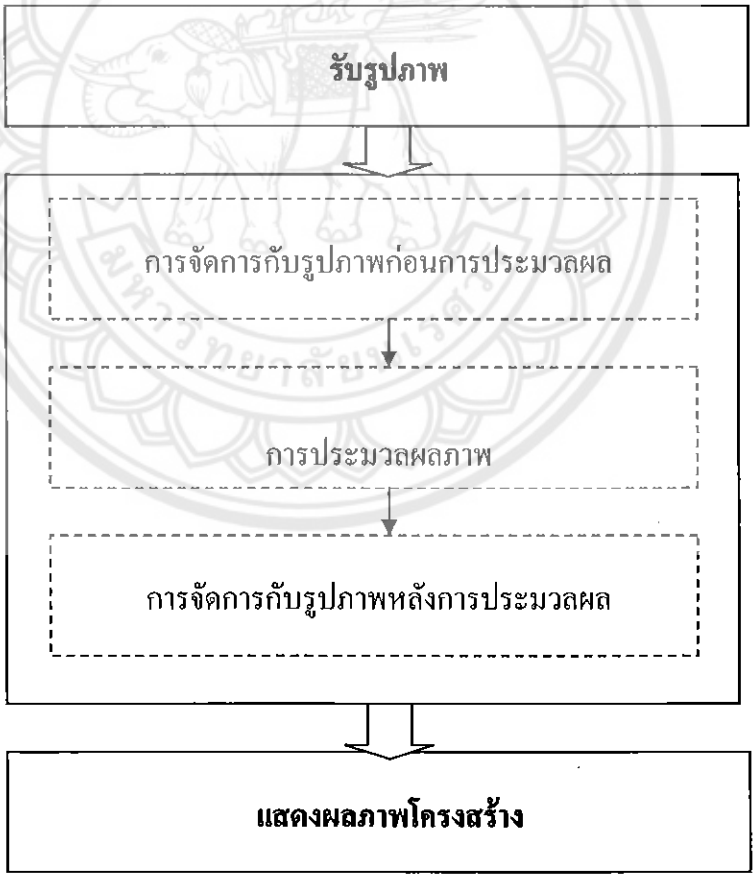
รูปที่ 2.22 ตัวอย่าง ***container.add(Component,Borderlayout.position)***

บทที่ 3
ขั้นตอนและการดำเนินงาน

บทนี้จะกล่าวถึงวิธีการออกแบบและพัฒนาระบบสำหรับโครงงานนี้ เพื่อความสะดวกในการทำความเข้าใจผู้พัฒนาจึงแบ่งระบบออกเป็น 3 ส่วนหลักๆ ดังนี้

- การจัดการกับรูปภาพก่อนการประมวลผล
- การประมวลผลภาพ
- การจัดการกับรูปภาพหลังการประมวลผล

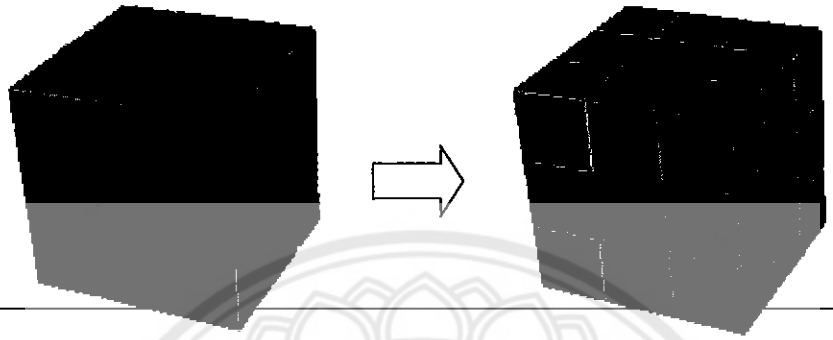
แผนภาพแสดงลำดับการทำงาน



รูปที่ 3.1 แผนภาพการแสดงโครงสร้างรูป 3 มิติ

3.1 การจัดการกับรูปภาพก่อนการประมวลผล

ในการประมวลผลรูปภาพ 3 มิติ จำเป็นที่ต้องทำการเก็บข้อมูลจุดภาพให้อยู่ในรูปแบบของ array 3 มิติ โดยใช้หลักการแทนจุดภาพด้วยระบบ Binary เพื่อความสะดวกในขั้นตอนการประมวลผล



รูปที่ 3.2 การเก็บข้อมูลจุดภาพให้อยู่ในรูปแบบของ array 3 มิติ

3.2 การประมวลผลภาพ

ขั้นตอนในการประมวลผลรูปภาพนั้น ใช้เทคนิคการแสดงผลโครงสร้าง 3 algorithm หลัก ได้แก่

- Distance & Transform
- Voronoi
- Thinning

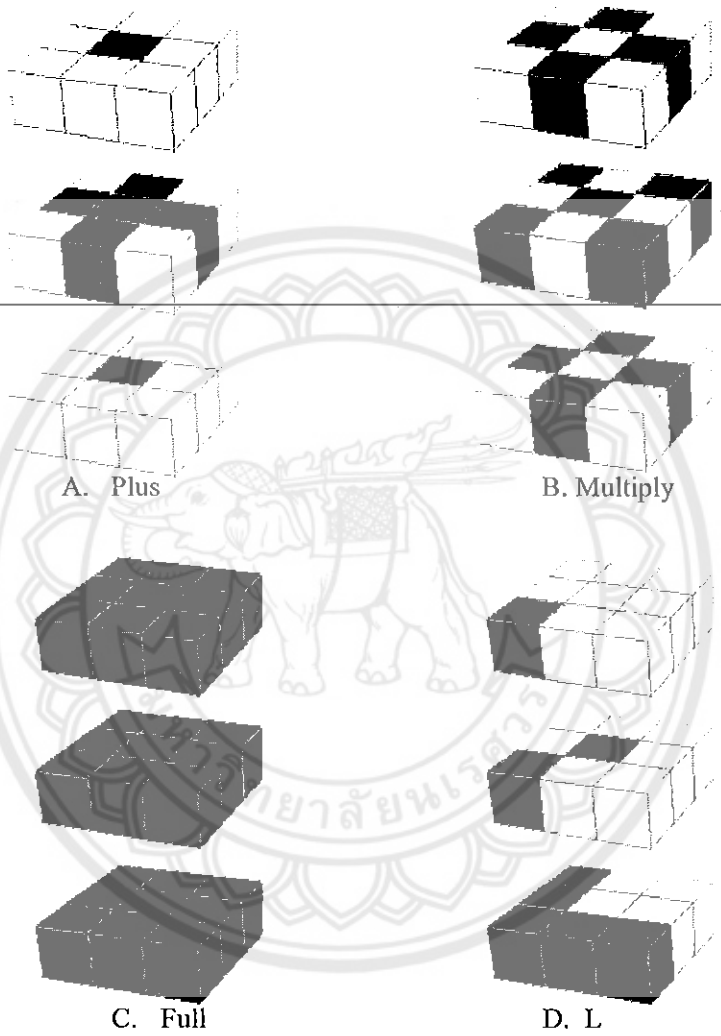
3.2.1 Distance & Transform [6]

ในการประมวลผลแบบ Distance & Transform จะแบ่งขั้นตอนออกเป็น 2 ส่วน ได้แก่

- การสร้างแผนภาพระยะทาง
- การเลือกตำแหน่งของโครงสร้าง

3.2.1.1 การสร้างแผนภาพระยะทาง

ขั้นตอนนี้จะทำการเปลี่ยนแปลงค่าของ array แต่ละตำแหน่ง ด้วยการพิจารณาตำแหน่งต่างๆ รอบข้างของตำแหน่งที่พิจารณา ดังนี้



รูปที่ 3.3 ตำแหน่งรอบข้างที่ใช้พิจารณา

<u>ขั้นตอนการทำงาน</u>

พิจารณาตำแหน่งที่มีค่าไม่เท่ากับศูนย์

- หาค่าน้อยที่สุดในตำแหน่งรอบข้างของตำแหน่งที่พิจารณา
- ค่าน้อยที่สุดบวกด้วยหนึ่งคือค่าของตำแหน่งนี้ในตารางใหม่
- ทำซ้ำขั้นตอนเดิมให้ครบทุกตำแหน่ง
- ทำจนกว่าตารางจะไม่มีการเปลี่ยนแปลงค่าอีก

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

3.2.1.2 การเลือกตำแหน่งของโครงสร้าง

ขั้นตอนนี้จะทำการเลือกตำแหน่งที่เป็นโครงสร้างของรูปภาพ ด้วยการพิจารณาตำแหน่งต่างๆ รอบข้างของตำแหน่งที่พิจารณา

ขั้นตอนการทำงาน

พิจารณาตำแหน่งที่มีค่าไม่เท่ากับศูนย์

- หาค่ามากที่สุดในตำแหน่งรอบข้างของตำแหน่งที่พิจารณาและตำแหน่งที่พิจารณา
- หากค่ามากที่สุดที่คำนวณได้มีค่าเท่ากับค่าในตำแหน่งที่พิจารณา กำหนดค่าในตำแหน่งนั้นให้เท่ากับหนึ่ง (เป็นตำแหน่งของโครงสร้าง)
- ทำซ้ำขั้นตอนเดิมให้ครบทุกตำแหน่ง

3.2.2 Voronoi

ในการประมวลผลแบบ Voronoi จะใช้หลักการในการสร้างแผนภาพระยะทาง เช่นเดียวกับการประมวลผลแบบ Distance & Transform (หัวข้อ 3.2.1.1) แต่ใช้หลักในการเลือกตำแหน่งของโครงสร้างแตกต่างกัน

3.2.2.1 การเลือกตำแหน่งของโครงสร้าง

ขั้นตอนนี้ Voronoi จะใช้หลักการเลือกตำแหน่งของโครงสร้างด้วยการพิจารณารัศมีระยะทางรอบทิศ

ขั้นตอนการทำงาน

พิจารณาตำแหน่งที่มีค่าไม่เท่ากับศูนย์

- หาค่าน้อยที่สุดในตำแหน่งรอบข้างของตำแหน่งที่พิจารณา
- พิจารณาดูว่าตำแหน่งรอบข้างของตำแหน่งที่พิจารณานั้นมีค่าน้อยที่สุดเท่ากับหรือมากกว่า 2 ตำแหน่งหรือไม่ ถ้าใช่กำหนดค่าในตำแหน่งนั้นให้เท่ากับหนึ่ง (เป็นตำแหน่งของโครงสร้าง) โดยค่าแผนภาพระยะทางที่ใช้พิจารณานี้เป็นการอ้างอิงค่าการหาพื้นที่ขอบเขตและค่าที่น้อยที่สุดที่ใช้อ้างอิงในการเลือกตำแหน่งของโครงสร้างก็คือเส้นขอบที่ควรพิจารณาตามหลักทฤษฎีของ Voronoi
- ทำซ้ำขั้นตอนเดิมให้ครบทุกตำแหน่ง

i 572 8356
2/5
W 4511
2552

3.2.3 Thinning [7]

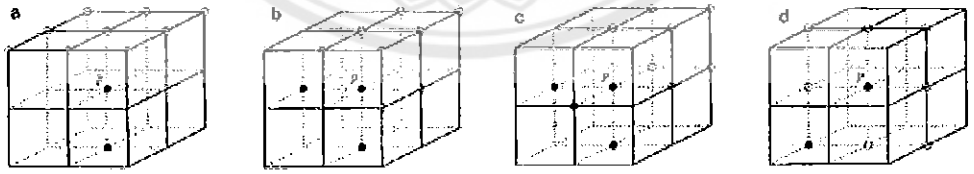
ในการประมวลผลแบบ Thinning จะใช้หลักการในการสร้างแผนภาพระยะทาง เช่นเดียวกับการประมวลผลแบบ Distance & Transform (หัวข้อ 3.2.1.1) แต่ใช้หลักในการเลือกตำแหน่งของโครงสร้างแตกต่างกัน

3.2.3.1 การเลือกตำแหน่งของโครงสร้าง

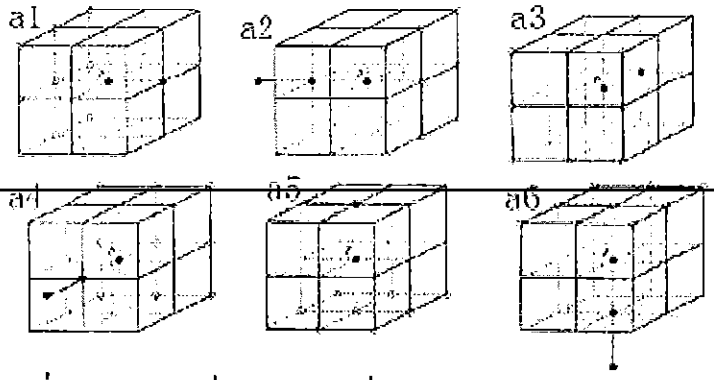
ขั้นตอนนี้ Thinning จะใช้หลักการเลือกตำแหน่งของโครงสร้างด้วยการพิจารณาเงื่อนไขต่างๆ ดังต่อไปนี้

- 26 ตำแหน่งรอบข้าง มีตำแหน่งที่มีค่ามากกว่าศูนย์อยู่ 1 ตำแหน่งใช่หรือไม่ ถ้าใช่เลือกตำแหน่งนั้นเป็นตำแหน่งของโครงสร้าง
- 26 ตำแหน่งรอบข้าง มีตำแหน่งที่มีค่ามากกว่าศูนย์อยู่ 2 ตำแหน่งดังนี้ ใช่หรือไม่
  - ตำแหน่งหน้าและขวา หรือ ตำแหน่งหน้าและบน
  - ตำแหน่งหลังและซ้าย หรือ ตำแหน่งบนและซ้าย
  - ตำแหน่งหลังและล่าง หรือ ตำแหน่งขวาและล่าง

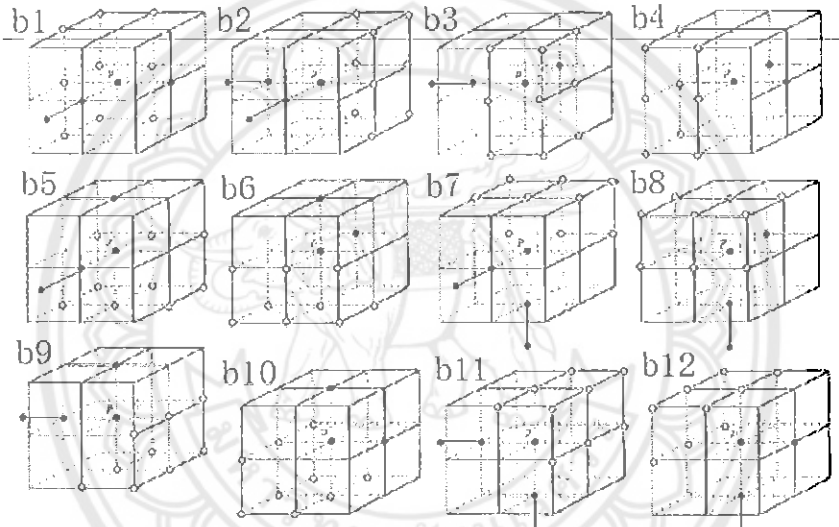
  ถ้าใช่เลือกตำแหน่งนั้นเป็นตำแหน่งของโครงสร้าง
- หากไม่เป็นดังกล่าวข้างต้น หากตรงตามเงื่อนไขใดเงื่อนไขในรูปด้านล่างนี้ (รูปที่ 3.5, 3.6, 3.7, 3.9) ให้ลบตำแหน่งนั้นออกทันที (ตำแหน่งนั้นไม่ใช่ส่วนหนึ่งของโครงสร้าง)



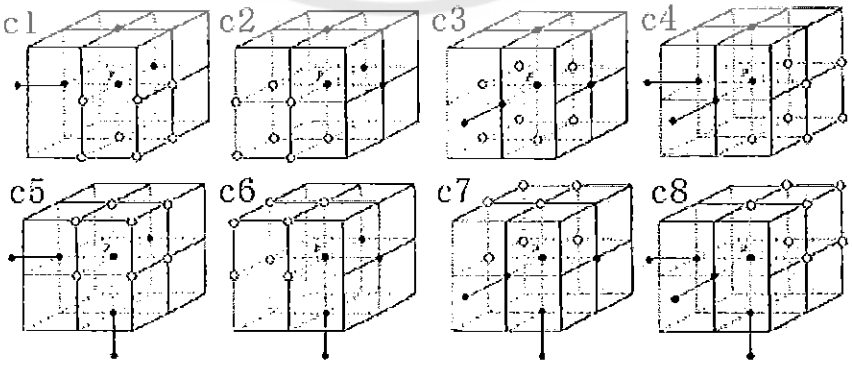
รูปที่ 3.4 ตำแหน่งพื้นฐานใน class A, B, C, D



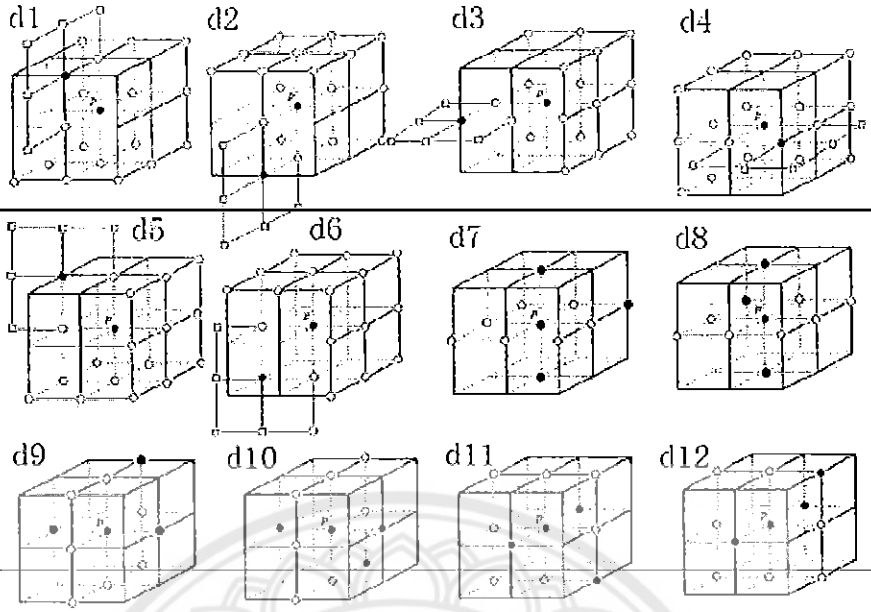
รูปที่ 3.5 ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class A



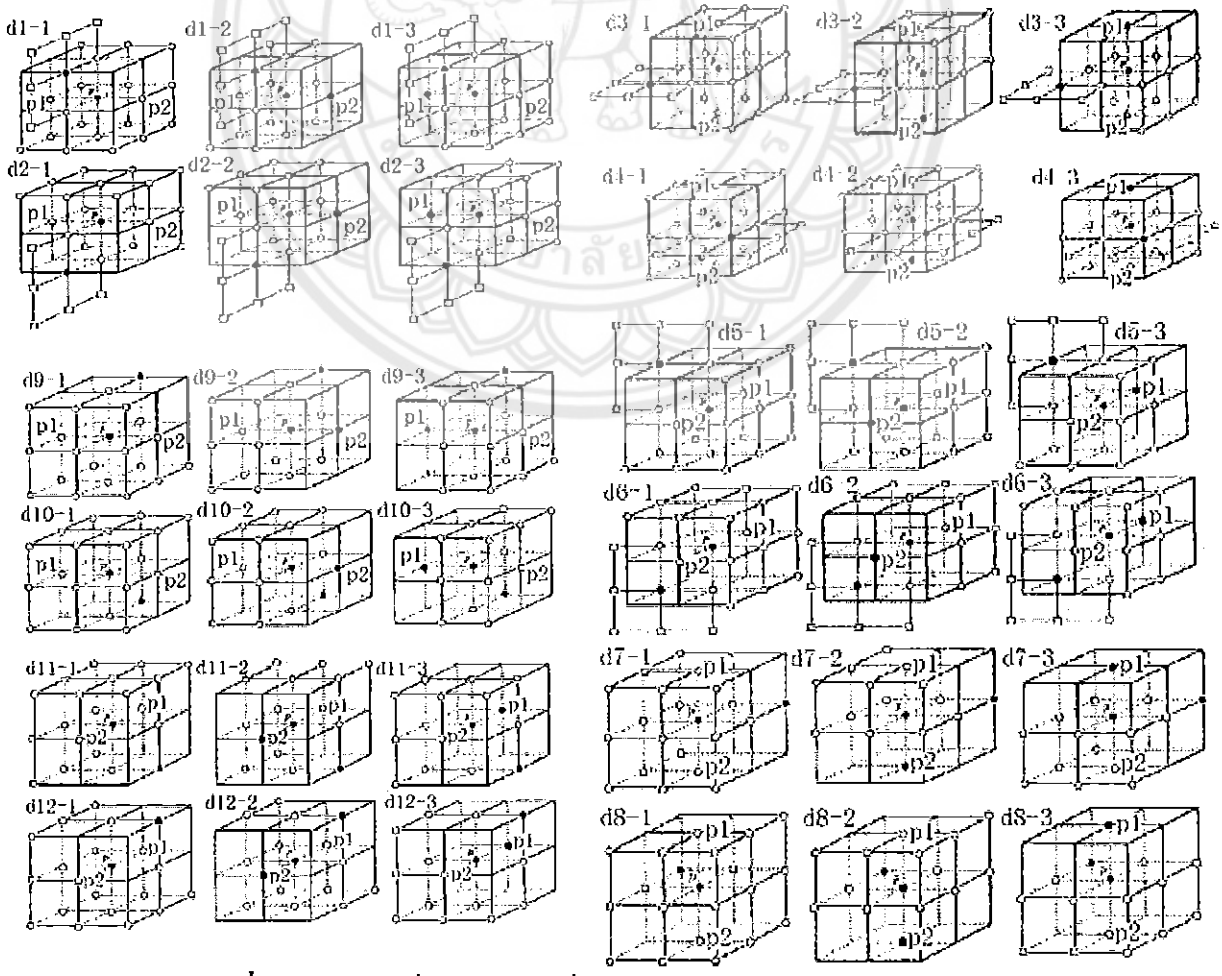
รูปที่ 3.6 ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class B



รูปที่ 3.7 ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class C



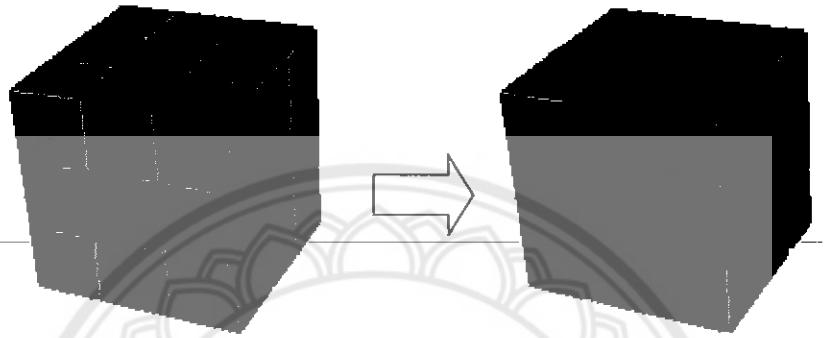
รูปที่ 3.8 ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class D



รูปที่ 3.9 ตำแหน่งที่ไม่ใช่ส่วนหนึ่งของโครงสร้างใน class modified D

3.3 การจัดการกับรูปภาพหลังการประมวลผล

ในการแสดงผลโครงสร้างของรูปภาพ 3 มิติ นั้น เนื่องจากค่าที่ได้จากการประมวลผลใช้หลักการ Binary ซึ่งเก็บค่าไว้ใน array 3 มิติ จึงจำเป็นต้องทำการแปลง array 3 มิติ ให้กลายเป็นรูปภาพของโครงสร้าง 3 มิติดังเดิม



รูปที่ 3.10 การแปลงข้อมูล array 3 มิติให้อยู่ในรูปแบบของรูปภาพ 3 มิติ

บทที่ 4

การออกแบบซอฟต์แวร์

เพื่อความสะดวกที่ผู้อื่นจะทำการศึกษาและประยุกต์ใช้ซอฟต์แวร์ที่พัฒนาขึ้น บทนี้จะแนะนำระบบซอฟต์แวร์ ดังนี้

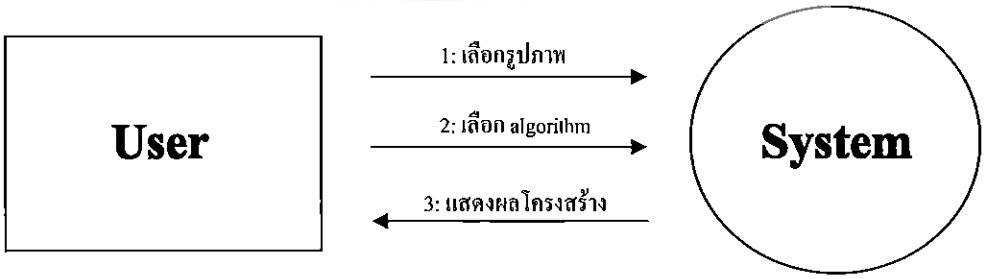
- คุณสมบัติของซอฟต์แวร์
- ขั้นตอนการประมวลผล
- ขอบเขตของระบบ
- การออกแบบซอฟต์แวร์

4.1 คุณสมบัติของซอฟต์แวร์

โปรแกรมสำหรับแสดงผลโครงสร้างของรูปภาพ 3 มิติ มีคุณสมบัติต่างๆ ดังนี้

- สามารถเลือกภาพที่ต้องการได้
- สามารถเลือก algorithm ที่ต้องการการประมวลผลได้
- สามารถบันทึกรูปภาพผลลัพธ์ออกมาในรูปแบบภาพนามสกุล .JPG ได้

4.2 ขั้นตอนการประมวลผล



รูปที่ 4.1 ขั้นตอนการประมวลผลของระบบ

การทำงานของระบบเริ่มจากผู้ใช้ทำการเลือกรูปภาพที่ต้องการแสดงผล จากนั้นเลือก algorithm ที่ต้องการสั่งให้ระบบทำการประมวลผล เมื่อระบบได้รับคำสั่ง ซอฟต์แวร์ก็จะนำภาพที่รับเข้ามานั้นมาทำการประมวลผลหาโครงสร้างตามหลัก algorithm แล้วส่งภาพผลลัพธ์ที่ได้ให้กับผู้ใช้

4.3 ขอบเขตของระบบ

โปรแกรมสำหรับแสดงผลโครงสร้างของรูปภาพ 3 มิติ มีขอบเขตของระบบดังนี้

- สามารถรับรูปภาพได้เฉพาะไฟล์นามสกุล .TXT และ .OBJ
- สามารถประมวลผลโดยรองรับรูปภาพได้เฉพาะรูปที่มีลักษณะเป็น Polygon
- สามารถบันทึกรูปภาพโครงสร้างของรูป 3 มิติได้เฉพาะไฟล์นามสกุล .JPG

4.4 การออกแบบซอฟต์แวร์

การออกแบบซอฟต์แวร์นั้นควรออกแบบให้ครอบคลุมทั้ง 4 มุมมองดังรูปด้านล่าง

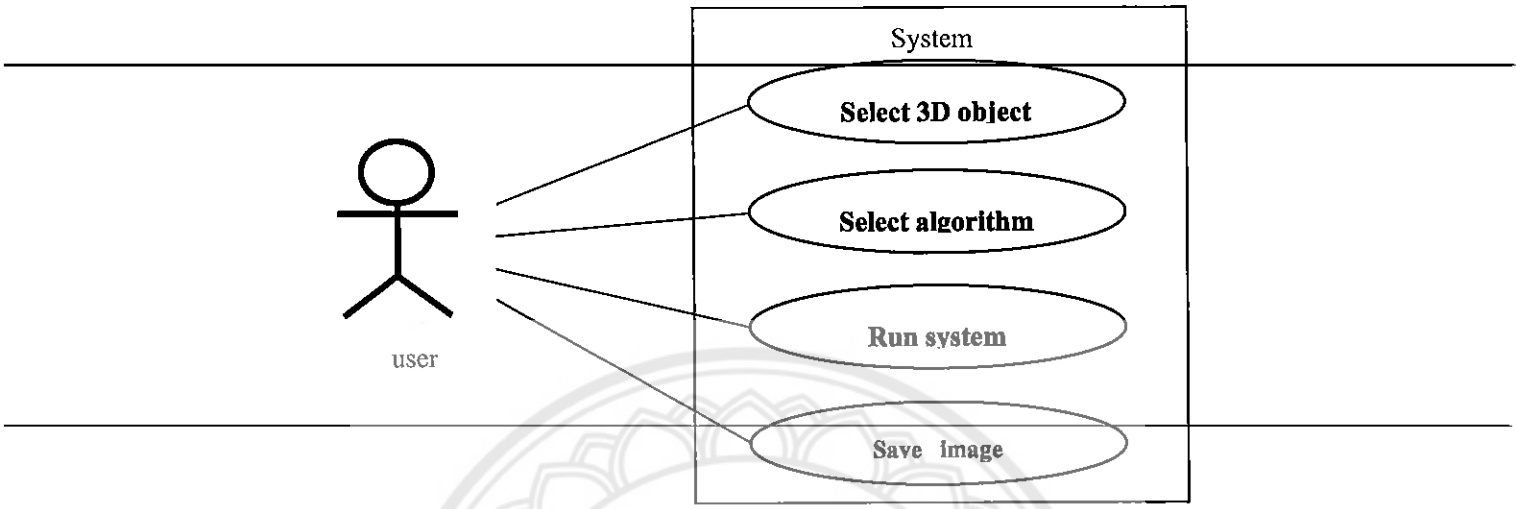
	Dynamic	Static
External	- Use case diagram - Sequence diagram	Component diagrams
Internal	Activity diagrams	Class diagrams

รูปที่ 4.2 มุมมองในการออกแบบซอฟต์แวร์

โดยแต่ละมุมมองมีความหมายดังต่อไปนี้

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

4.4.1 ยูสเคลส ไดอะแกรม (Use case diagram)



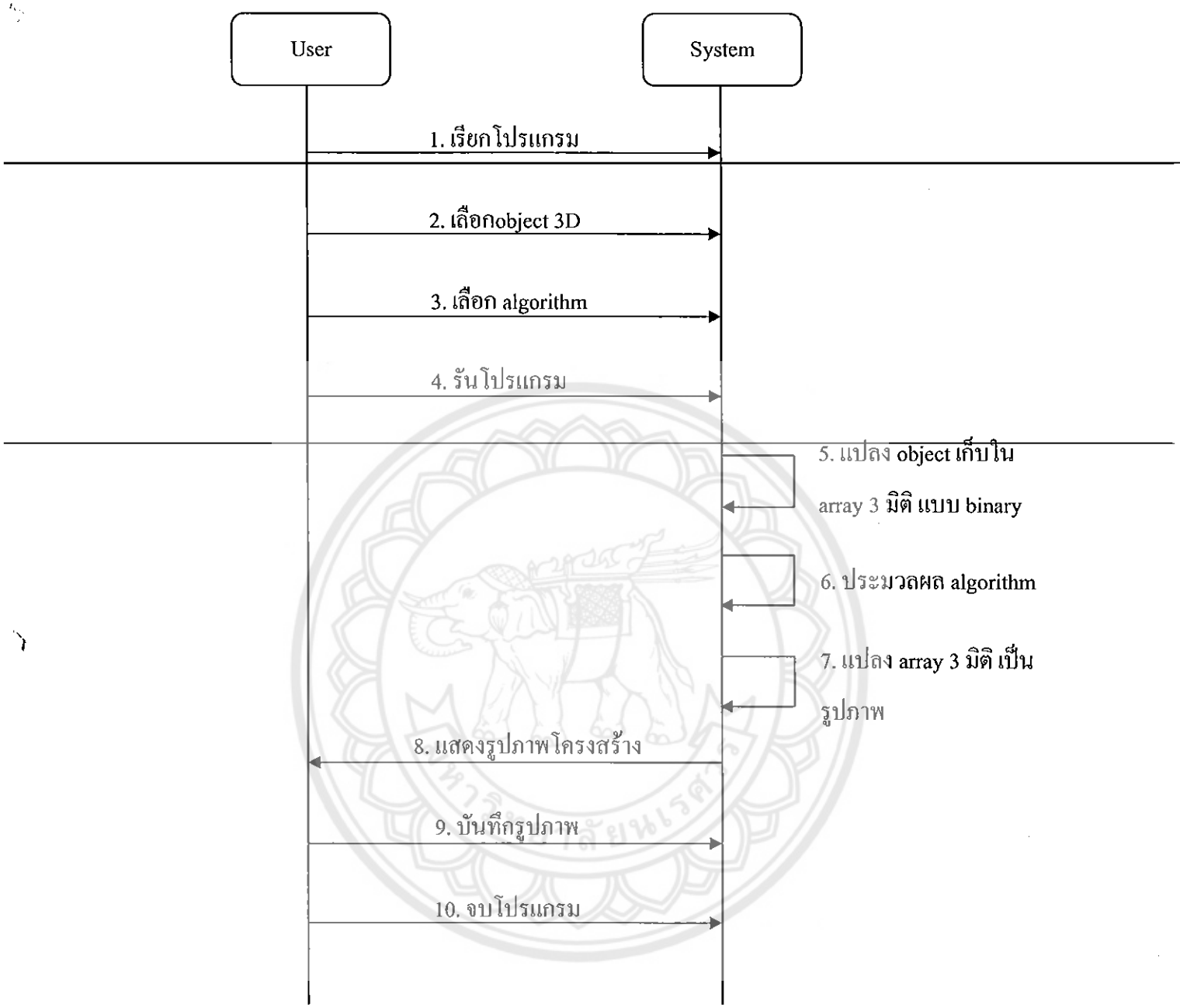
รูปที่ 4.3 Use case diagram

การติดต่อกับผู้ใช้นั้นจะมี 4 ขั้นตอนคือ

1. ผู้ใช้ทำการเลือก object ที่ต้องการประมวลผล
2. ผู้ใช้ทำการเลือก algorithm
3. ผู้ใช้สั่งให้ระบบทำการประมวลผล
4. ผู้ใช้ทำการบันทึกรูปภาพผลลัพธ์

4.4.2 ซีเควน ไดอะแกรม (Sequence diagram)

ลำดับการทำงานของโปรแกรมเริ่มจากการที่ผู้ใช้เรียกโปรแกรมขึ้นมา จากนั้นทำการเลือก object ที่ต้องการหาโครงสร้างของภาพ แล้วก็ทำการเลือก algorithm ที่ต้องการใช้ในการประมวลผล จากนั้นผู้ใช้ก็สั่งให้โปรแกรมเริ่มทำการประมวลผล โดยโปรแกรมจะเริ่มประมวลผลโดยการแปลง object โดยใช้หลักการแทนจุดภาพด้วยระบบ binary แล้วทำการเก็บข้อมูลของรูปภาพเป็นโครงสร้างของ array 3 มิติ จากนั้นก็ทำการประมวลผลตามอัลกอริทึมที่ผู้ใช้เลือก จากนั้นก็ทำการแปลงข้อมูลจากarray 3 มิติ กลับมาเป็นรูปภาพแล้วแสดงรูปภาพผลลัพธ์ออกทางหน้าจอ ถ้าผู้ใช้ต้องการบันทึกภาพผลลัพธ์ระบบก็จะทำการบันทึกภาพเก็บไว้ในหน่วยความจำ ตามที่ผู้ใช้กำหนด ดังรูปที่ 4.4



รูปที่ 4.4 Sequence diagram

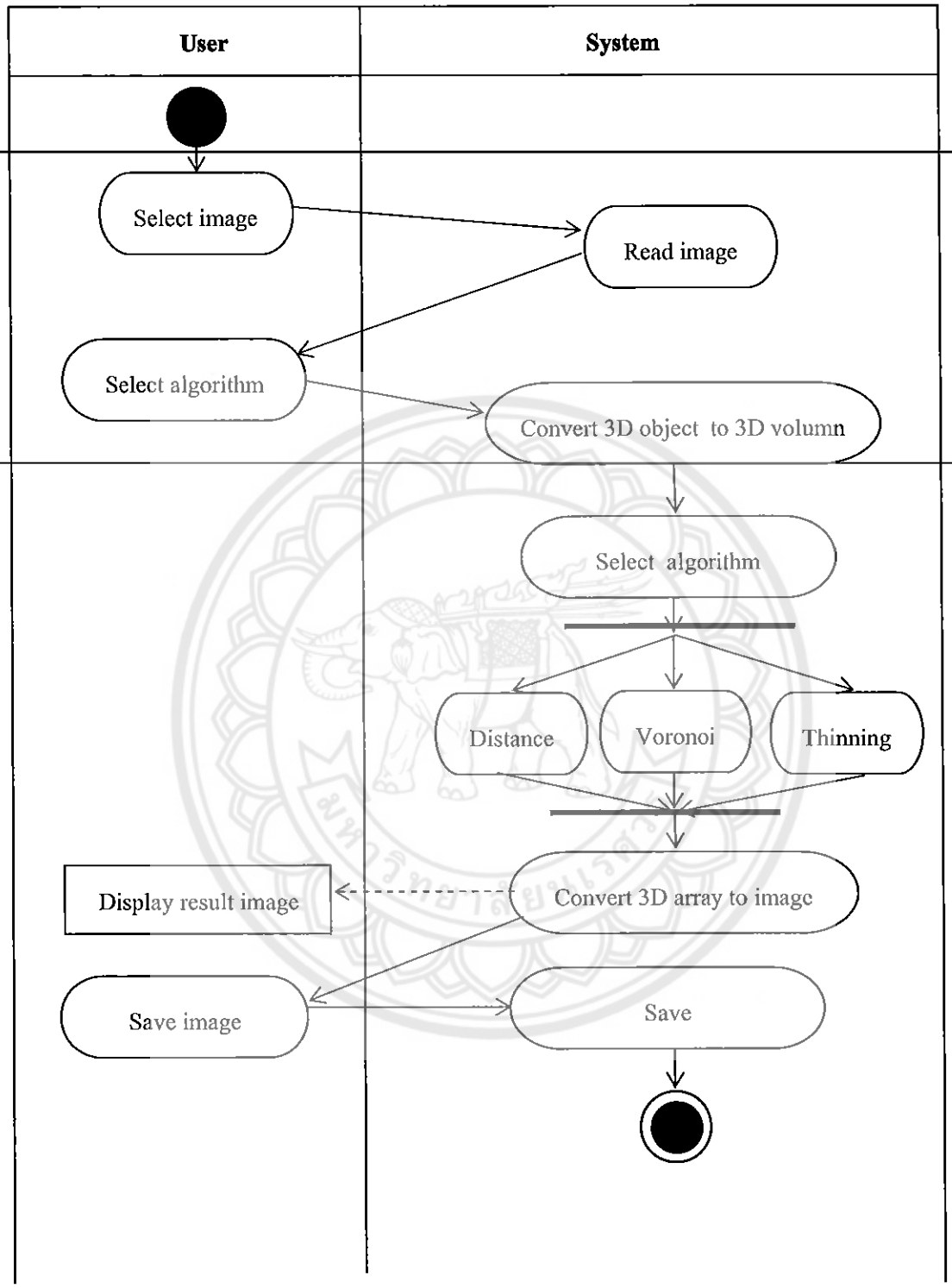
4.4.3 คอมโพเนนท์ไดอะแกรม (Component Diagram)

เนื่องจากโปรแกรมไม่ได้ติดต่อ หรือเรียกใช้ function ภายนอก ดังนั้นจึงมีเพียง component ของโปรแกรมหลักเพียง component เดียว

รูปที่ 4.5 Component diagram

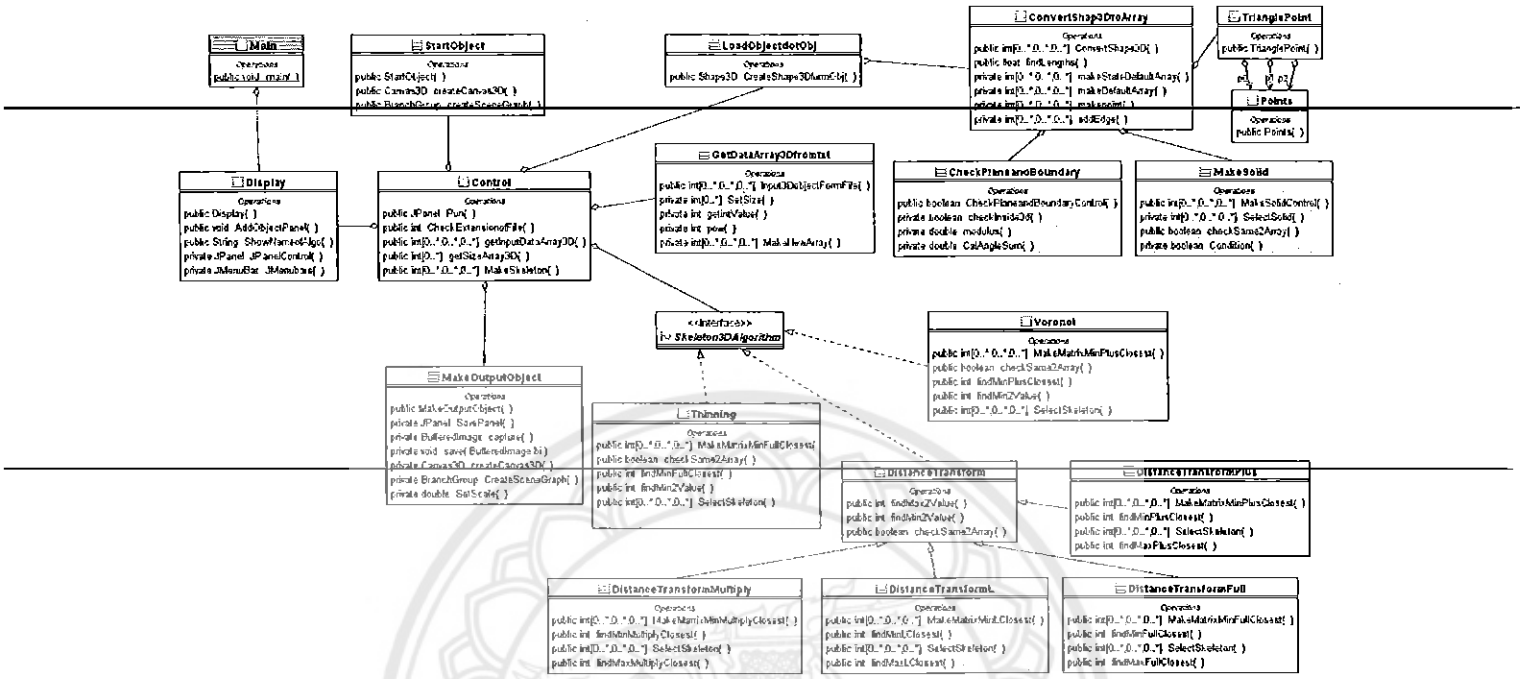
4.4.4 แอคทีวิตีไดอะแกรม (Activity diagram)

ลำดับกิจกรรมต่างๆ ของโปรแกรมจะเริ่มจากผู้ใช้ทำการเลือก object ที่ต้องการประมวลผล จากนั้นโปรแกรมจะทำการอ่านไฟล์ object มาเก็บไว้ในหน่วยความจำ จากนั้นผู้ใช้ทำการเลือก algorithm ที่ต้องการแล้วสั่งให้โปรแกรมเริ่มทำการประมวลผล จากนั้นระบบทำการแปลง object แล้วทำการเก็บข้อมูลของรูปภาพในรูปแบบของ array 3 มิติ จากนั้นระบบทำการประมวลผลตาม algorithm ที่ผู้ใช้เลือกจนเสร็จ จากนั้นระบบจึงทำการแปลง array 3 มิติกลับเป็นรูปภาพแล้วแสดงรูปภาพผลลัพธ์ออกมาทางหน้าจอ จากนั้นผู้ใช้ทำการเลือกชื่อและที่อยู่ในหน่วยความจำของรูปภาพที่ต้องการบันทึกแล้วสั่งให้โปรแกรมเริ่มทำการบันทึกข้อมูล จากนั้นโปรแกรมจะทำการบันทึกข้อมูลรูปภาพไว้ในหน่วยความจำที่กำหนด



รูปที่ 4.6 Activity diagram

4.3.5 คลาสไดอะแกรม (Class diagram)



รูปที่ 4.7 Class diagram

ใน Class diagram จะประกอบด้วย 20 class ดังนี้

1. Main ทำหน้าที่เป็น class หลักซึ่งเลือกรูปแบบของ GUI ที่ใช้ติดต่อกับผู้ใช้
2. Display เป็น GUI ซึ่งใช้ติดต่อกับผู้ใช้
3. Control ทำหน้าที่ในการควบคุมการทำงานทั้งหมดของโปรแกรม
4. StartObject เป็น object แรกที่แสดงบน panel เมื่อทำการเปิดโปรแกรม
5. MakeOutputObject ทำหน้าที่สร้าง skeleton object หรือรูปภาพผลลัพธ์ของโปรแกรม
6. LoadObjectdotObj ทำหน้าที่ในการโหลดไฟล์ .obj
7. GetDataArray3Dfromtxt ทำหน้าที่ในการโหลดไฟล์ .txt
8. ConvertShap3DtoArray ทำหน้าที่แปลง object เป็น array 3 มิติ
9. MakeSolid ทำหน้าที่สร้าง solid ให้แก่ object
10. CheckPlaneandBoundary ทำหน้าที่สร้าง surface ให้แก่ object
11. Points เป็น class สำหรับเก็บข้อมูลค่าจุดของ object (ค่า x y และ z)
12. TrianglePoint เป็นclass สำหรับเก็บกลุ่มข้อมูลของจุดที่ประกอบกันเป็น surface object
13. Skeleton3DAlgorithm เป็น interface class สำหรับ algorithm

14. Voronoi ทำหน้าที่ประมวลผลด้วย Voronoi algorithm
15. Thinning ทำหน้าที่ประมวลผลด้วย Thinning algorithm
16. DistanceTransform เป็น base class สำหรับ Distance & Transform algorithm
17. DistanceTransformPlus ทำหน้าที่ประมวลผลด้วย Distance & Transform - Plus algorithm
18. DistanceTransformFull ทำหน้าที่ประมวลผลด้วย Distance & Transform - Full algorithm
19. DistanceTransformMultiply ทำหน้าที่ประมวลผลด้วย Distance & Transform - Multiply algorithm
20. DistanceTransformL ทำหน้าที่ประมวลผลด้วย Distance & Transform - L algorithm

บทที่ 5

ผลการทดสอบ

ในบทนี้จะกล่าวถึงการทดสอบใช้งานซอฟต์แวร์ที่ได้ออกแบบและพัฒนาขึ้น โดยทำการแบ่งการทดสอบออกเป็น 2 ส่วน ดังนี้

1. ทดสอบการทำงานแต่ละ algorithm
2. ทดสอบว่าสามารถแสดงผลแบบ GUI ได้

5.1 ทดสอบการทำงานแต่ละ algorithm

โปรแกรมการแสดงผลโครงสร้างของรูป 3 มิติ สามารถทำงานด้วย algorithm ดังนี้

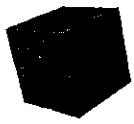
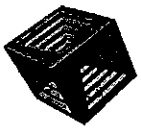
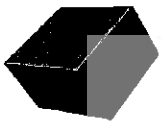
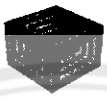
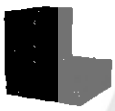
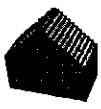
- Distance & Transform
  - Distance & Transform - Plus
  - Distance & Transform - Mutiply
  - Distance & Transform - Full
  - Distance & Transform - L
- Voronoi
- Thinning

โดยการทดสอบแต่ละ algorithm นั้นจะทำการทดสอบกับรูปภาพ 3 มิติ ดังนี้

- รูปทรงสี่เหลี่ยมจัตุรัสตัน
- รูปทรงสี่เหลี่ยมผืนผ้าตัน
- รูปทรงสี่เหลี่ยมกลวง
- รูปทรงตัวแอลตัน
- รูปทรงพีระมิด1
- รูปทรงพีระมิด2
- รูปบ้าน
- รูปใบพัด

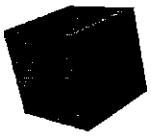
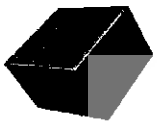
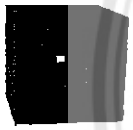
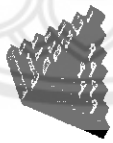
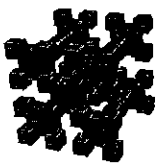
5.1.1 Distance & Transform - Plus

ตารางที่ 5.1 ตารางแสดงผลการทดสอบ algorithm Distance & Transform – Plus กับรูปภาพต่างๆ

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

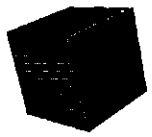
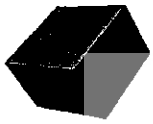
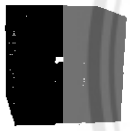
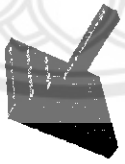
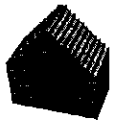
5.1.2 Distance & Transform - Multiply

ตารางที่ 5.2 ตารางแสดงผลการทดสอบ algorithm Distance & Transform – Multiply กับรูปภาพต่างๆ

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

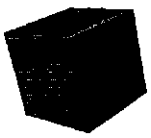
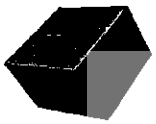
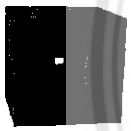
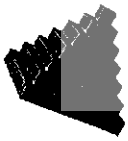
5.1.3 Distance & Transform - Full

ตารางที่ 5.3 ตารางแสดงผลการทดสอบ algorithm Distance & Transform – Full กับรูปภาพต่างๆ

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

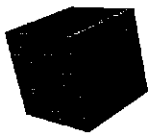
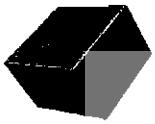
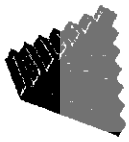
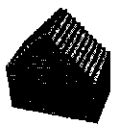
5.1.4 Distance & Transform - L

ตารางที่ 5.4 ตารางแสดงผลการทดสอบ algorithm Distance & Transform – L กับรูปภาพต่างๆ

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

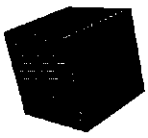
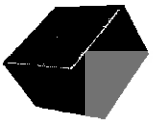
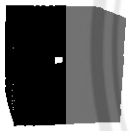
5.1.5 Voronoi

ตารางที่ 5.5 ตารางแสดงผลการทดสอบ algorithm Voronoi กับรูปภาพต่างๆ

ภาพต้นแบบ	ภาพผลลัพธ์	Topological	Geometrical
		Yes	Yes
		Yes	Yes
		Yes	Yes
		Yes	Yes
		Yes	Yes
		Yes	Yes
		Yes	Yes
		Yes	Yes

5.1.6 Thinning

ตารางที่ 5.6 ตารางแสดงผลการทดสอบ algorithm Thinning กับรูปภาพต่างๆ

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

5.2 ผลการเปรียบเทียบประสิทธิภาพการทำงาน

จากผลการเปรียบเทียบประสิทธิภาพการประมวลผลโครงสร้างของรูปภาพ 3 มิติจำนวนทั้งหมด 8 รูป algorithm ที่ให้ผลลัพธ์โครงสร้างที่ดีที่สุดคือ Voronoi เนื่องจากรักษาทั้ง geometrical และ topology ของทุกรูป รองลงมาคือ Distance & Transform – Plus, Thinning, Distance & Transform – Multiply, Distance & Transform – Full และ Distance & Transform – L ตามลำดับ ซึ่งสามารถสรุปผลเป็นตารางเปรียบเทียบประสิทธิภาพการทำงานของแต่ละ algorithm ได้ดังตารางที่ 5.7

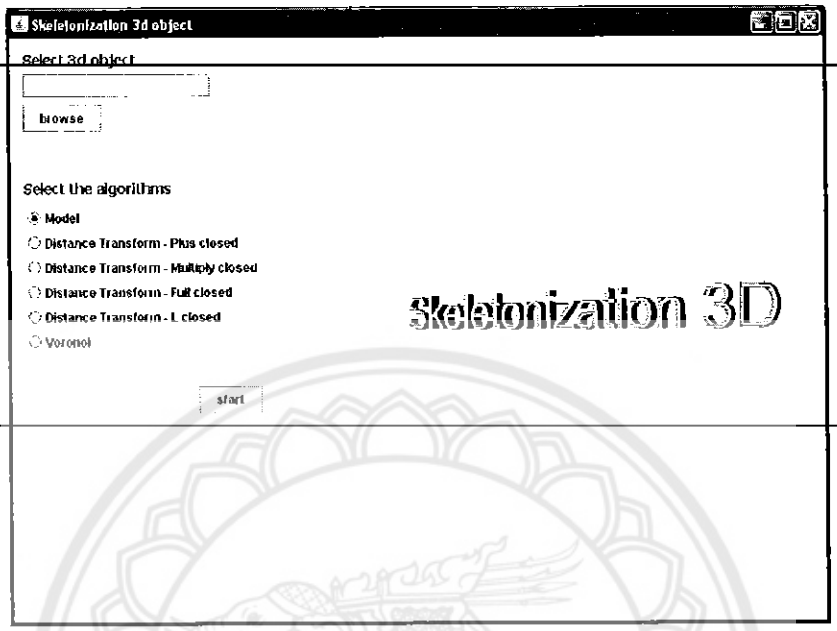
ตารางที่ 5.7 ตารางเปรียบเทียบประสิทธิภาพการทำงานของแต่ละ algorithm

ลำดับที่	algorithm	ประสิทธิภาพการทำงาน	
		รักษา topology	รักษา geometrical
1	Voronoi	8	8
2	Distance & Transform - Plus	7	8
3	Thinning	7	3
4	Distance & Transform - Multiply	5	8
5	Distance & Transform - Full	5	4
6*	Distance & Transform - L	0	0

การเปรียบเทียบประสิทธิภาพจะยึดหลักการรักษา topology เป็นอันดับแรก ถัดมาจึงพิจารณาการรักษา geometrical เรียงตามลำดับ ซึ่งหากรูปภาพที่นำมาประมวลผลมีจำนวนหรือรูปแบบที่แตกต่างออกไป การเปรียบเทียบประสิทธิภาพอาจเปลี่ยนแปลงได้ตามความเหมาะสม

* Distance & Transform – L ถูกกำหนดขึ้นเพื่อนำมาใช้ในการพิจารณาโครงสร้างด้านในของรูปภาพเพื่อให้เห็นความแตกต่างได้ชัดเจนมากขึ้นเท่านั้น ผลลัพธ์ที่ได้จึงไม่สอดคล้องตรงตามทฤษฎี

5.3 ทดสอบว่าสามารถแสดงผลแบบ GUI ได้



รูปที่ 5.1 หน้าต่างซอฟต์แวร์การแสดงผลโครงสร้างของรูป 3 มิติ

จากการพัฒนาซอฟต์แวร์การแสดงผลโครงสร้างของรูป 3 มิติ ผู้พัฒนาได้ทำการแสดงผลซอฟต์แวร์ที่ได้ออกมาในรูปแบบของ GUI ซึ่งจะทำให้ผู้ใช้สามารถเข้าใจหลักการทำงาน อีกทั้งยังเพิ่มความสะดวกต่อการใช้งานมากยิ่งขึ้น

บทที่ 6

สรุปผล

6.1 สรุปผลการทดลอง

ผลการศึกษาและพัฒนาโปรแกรมการแสดงโครงสร้างของรูป 3 มิติ ด้วยเทคนิคการแสดงโครงสร้างในรูปแบบต่างๆ จะเห็นได้ว่าแต่ละเทคนิคก็จะมีการวิเคราะห์ มีหลักการพิจารณาที่แตกต่างกันไป ดังนี้

- Distance & Transform จะอาศัยหลักการสร้างแผนภาพระยะทางด้วยการพิจารณาตำแหน่งรอบข้างในรูปแบบต่าง ๆ เพื่อหาโครงสร้างของรูป
  - Distance & Transform – Plus closed พิจารณาตำแหน่งรอบข้างในรูปแบบเครื่องหมายบวก
  - Distance & Transform – Multiply closed พิจารณาตำแหน่งรอบข้างในรูปแบบเครื่องหมายคูณ
  - Distance & Transform – Full closed พิจารณาตำแหน่งรอบข้างทั้งหมด
  - Distance & Transform – L closed พิจารณาตำแหน่งรอบข้างในรูปแบบตัวแอล

  ซึ่งเทคนิค Distance & Transform จะยังคงรักษา geometrical แต่ไม่รองรับการรักษา topology ในบางรูป
- Voronoi จะอาศัยหลักการหาขอบเขตพื้นที่ของแต่ละจุดกำเนิด และนำเส้นขอบเขตที่แบ่งได้นั้นมาพิจารณาหาโครงสร้างของรูป เทคนิคนี้จะรองรับและรักษาทั้ง geometrical และ topology แต่มีข้อเสียในเรื่องของการสิ้นเปลืองกระบวนการทำงานเป็นอย่างมาก หากภาพที่ต้องการแปลงโครงสร้างมีความใหญ่และซับซ้อน
- Thinning จะอาศัยหลักการลดขนาดของรูปในทุกด้านลงเรื่อยๆ เทคนิคนี้จะยังคงรักษา topology แต่ไม่รักษา geometrical

ซึ่งประสิทธิภาพการทำงานของแต่ละ algorithm สามารถสรุปผลได้ตารางที่ 6.1

ตารางที่ 6.1 ตารางเปรียบเทียบประสิทธิภาพการทำงานของแต่ละ algorithm

ลำดับที่	algorithm	ประสิทธิภาพการทำงาน	
		รักษา topology	รักษา geometrical
1	Voronoi	Yes	Yes
2	Distance & Transform - Plus	No	Yes
3	Thinning	Yes	No
4	Distance & Transform - Multiply	No	Yes
5	Distance & Transform - Full	No	Yes
6*	Distance & Transform - L	No	No

6.2 ปัญหาที่พบ

6.2.1 Application นี้ ในส่วนของการรับภาพและแสดงผลจำเป็นต้องใช้หน่วยความจำในการเก็บข้อมูลเป็นอย่างมาก ดังนั้น เมื่อเราเปิดหน้าต่างและทำการประมวลผลภาพหลายๆ รอบ หรือกดให้ทำการประมวลผลไวเกินไป จะทำให้หน่วยความจำที่ใช้ไม่เพียงพอ

6.2.2 หากรูปภาพที่รับเข้ามามีขนาดใหญ่มากๆ หรือมีความซับซ้อนของภาพสูง จะทำให้การประมวลผลภาพช้าลงและอาจทำให้ระบบเกิดการค้างได้ในบางกรณี

6.2.3 สามารถประมวลผลโดยรองรับรูปภาพได้เฉพาะรูปที่มีลักษณะเป็น Polygon

6.3 ข้อเสนอแนะ

6.3.1 ในการรับภาพและแสดงผลควรใช้การเรียกคืนหน่วยความจำ เอง โดยไม่ต้องรอให้ระบบคืนหน่วยความจำอัตโนมัติ เพื่อแก้ไขปัญหาหน่วยความจำไม่เพียงพอ

6.3.2 ควรพัฒนาโปรแกรมให้สามารถรองรับการประมวลผลรูปภาพได้ทุกรูปแบบ

6.3.3 จากการศึกษาการแสดงโครงสร้างของรูป 3 มิตินั้นมีความหลากหลายทางด้านเทคนิคที่ใช้เป็นอย่างมาก แม้จะใช้หลักการเดียวกันแต่หากเราพิจารณาในตำแหน่งหรือมุมมองที่ต่างกัน โครงสร้างที่ได้ก็จะเปลี่ยนแปลงไป ซึ่งไม่มีกฎเกณฑ์หรือมาตรฐานใดมาวัดความถูกต้องและแม่นยำได้ การเลือกใช้จึงขึ้นอยู่กับความเหมาะสมกับชิ้นงาน

เอกสารอ้างอิง

[1] Hong Zhang and Y.Daniel Liang. **Computer graphics using Java 2D and 3D**. USA: Pearson Education Ltd, Inc. 2007

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

[3] Leen Ammeraal . **Commputer graphics for Java programmers**. Great Britain: Bookcraft(Bath) Ltd, Inc. 1998

[4] Paul Bourke. **"Determining if a point lies on the interior of a polygon."** [online]. Available: http://local.wasp.uwa.edu.au/~pbourke/geometry/insidepoly. November 1987

[5] Rogers and David F. **Mathematical elements for computer graphics**. Singapore: McGraw-Hill Book Co. 1989

[6] Ruye Wang . "**A Skeletonization Algorithm.**" [online]. Available: http://fourier.eng.hmc.edu/e161/lectures/morphology/node3.html. 2004

[7] Steve Smith. "**The Thinning Algorithm.**" [online]. Available: http://users.fmrib.ox.ac.uk/~steve/susan/thinning/node2.html. 2009

[8] Suranaree University of technology. "**Java Swing.**" [online]. Available: http://ccsmail.sut.ac.th/e-ru/teacher/file/file243.doc. 2009

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวพลอยไพลิน ไหววิจิตร

ภูมิลำเนา 134/1 ม.3 ต.ท่างิ้ว อ.บรรพตพิสัย จ.นครสวรรค์

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนบรรพตพิสัยพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail : **joy_tawanyen@hotmail.com**

ชื่อ นางสาววนารัตน์ จุฬพันธ์ทอง

ภูมิลำเนา 632/1 ถ.ราชดำเนิน1 ต.ในเมือง อ.เมือง จ.กำแพงเพชร

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนกำแพงเพชรพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail : **tidz_pohskidjai@hotmail.com**