



วิทยานิพนธ์

ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติสำหรับการลดความซับซ้อนของ
พื้นผิวสามมิติ

**MULTIDIMENSIONAL BINARY CLUSTERING TREE FOR 3D
SURFACE SIMPLIFICATION**

นายธเนศ สุพันธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

พ.ศ. 2550



ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิทยาศาสตร์มหาบัณฑิต (วิทยาการคอมพิวเตอร์)

ปริญญา

วิทยาการคอมพิวเตอร์

วิทยาการคอมพิวเตอร์

สาขา

ภาควิชา

เรื่อง ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติสำหรับการลดความซับซ้อนของพื้นผิวสามมิติ

Multidimensional Binary Clustering Tree for 3D Surface Simplification

นามผู้วิจัย นายธนศ สุพันธ์

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(อาจารย์ชาคริต วัชโรภาส, Ph.D.)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

(อาจารย์ยัวร์เศรษฐ สุวรรณิก, วศ.ด.)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

(อาจารย์ฐิติวรรณ ศรีนาค, Ph.D.)

หัวหน้าภาควิชา

(ผู้ช่วยศาสตราจารย์อุมาพร ศิริธรานนท์, M.S.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

(รองศาสตราจารย์วินัย อัจจงหาญ, M.A.)

คณบดีบัณฑิตวิทยาลัย

วันที่ เดือน พ.ศ.

วิทยานิพนธ์

เรื่อง

ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติสำหรับการลดความซับซ้อนของพื้นผิวสามมิติ

Multidimensional Binary Clustering Tree for 3D Surface Simplification

โดย

นายธเนศ สุพันธ์

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)

พ.ศ. 2550

ชเนศ สุพันธ์ 2550: ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติสำหรับการลดความซับซ้อนของ
พื้นผิวสามมิติ ปริญญาวิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์) สาขาวิทยาการ
คอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก:
อาจารย์ชาคริต วัชรโรภาส, Ph.D. 63 หน้า

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อนำเสนออัลกอริทึมลดความซับซ้อนของพื้นผิวที่เรียกว่า
ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ (Multidimensional Binary Clustering Tree) ซึ่งสามารถลด
ความซับซ้อนของกลุ่มจุดบนพื้นผิวได้อย่างรวดเร็ว โดยเปรียบเทียบกับอัลกอริทึมลดความ
ซับซ้อนของพื้นผิวอื่นๆ ซึ่งประกอบด้วย การจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง (Divisive
Hierarchical Clustering) การจัดกลุ่มแบบค่อยๆเพิ่มขึ้น (Incremental Clustering) และการจัดกลุ่ม
แบบเค-มีน (K-Mean Clustering)

ชุดข้อมูลทดสอบที่ใช้ในการวิจัย คือพื้นผิวของแบบจำลองสามมิติที่ประกอบด้วยกลุ่มจุด
ซึ่งได้จากเครื่องสแกนเนอร์สามมิติ และทำการลดความซับซ้อนของชุดข้อมูลด้วยอัลกอริทึมลด
ความซับซ้อนของพื้นผิวต่างๆ เพื่อเปรียบเทียบประสิทธิภาพในเรื่องของเวลาการทำงาน จากนั้น
จึงทำการเปรียบเทียบเวลาที่ใช้ในการสร้างพื้นผิว และคุณภาพของพื้นผิวต่อไป

ผลการวิจัยพบว่า อัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติสามารถลดความซับซ้อน
ของพื้นผิวได้เร็วที่สุด เมื่อเปรียบเทียบกับอัลกอริทึมอื่นๆ ที่ใช้ในงานวิจัยนี้ เวลาในการสร้าง
พื้นผิวจากชุดข้อมูลที่ผ่านการลดความซับซ้อนของพื้นผิวด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ
การจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง และการจัดกลุ่มแบบเค-มีนนั้น ใช้เวลาใกล้เคียงกัน ส่วน
การลดความซับซ้อนของพื้นผิวด้วยการจัดกลุ่มแบบค่อยๆเพิ่มขึ้นใช้เวลาในการลดความซับซ้อน
และการสร้างพื้นผิวมากที่สุด

Thanet Suphan 2007: Multidimensional Binary Clustering Tree for 3D Surface Simplification. Master of Science (Computer Science), Major Field: Computer Science, Department of Computer Science. Thesis Advisor: Mr. Chakrit Watcharopas, Ph.D. 63 pages.

This research objective is to presents the surface simplification algorithm, Multidimensional Binary Clustering Tree (MBC Tree) that can quickly simplify surface and is still comparable to other algorithms such as Divisive Hierarchical Clustering (DHC), Incremental Clustering (INC), and K-Mean Clustering (KMC).

The data sets in this research are 3D points sampled from real surface using a modern 3D scanner device. The data sets are then simplified with surface simplification process to compare effectiveness among simplification algorithms. Thereafter, we compare surface reconstruction time and surface quality between the resulting surfaces and original one.

The results in this research indicate that MBC Tree outperforms other algorithms shown in the research on simplification process and uses less amount of simplification time. Whereas, the surface reconstruction time taken by MBC Tree, DHC and KMC are still comparable. The results also show that spends the most amount time on both simplification and reconstruction.

Student's signature

Thesis Advisor's signature

/ /

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณ อาจารย์ชาคริต วัชโรภาส ประธานกรรมการที่ปรึกษา ที่ได้ช่วยเหลือในการวางแผนงานวิจัยในวิทยานิพนธ์ฉบับนี้ ตลอดจนการให้คำปรึกษาแนะนำและตรวจแก้ไขข้อบกพร่องต่างๆ ขอกราบขอบพระคุณ อาจารย์วัชรเศรษฐ สุวรรณิก กรรมการร่วม อาจารย์จิตติวรรณ ศรีนาค กรรมการร่วม รศ. ดร.อนงค์นาฏ ศรีวิหค ประธานการสอบ และ รศ. นท. ดร.ประสงค์ ปราณิตพลกรัง ผู้ทรงคุณวุฒิภายนอก ที่กรุณาให้คำปรึกษาแนะนำและช่วยเหลือในการทำวิทยานิพนธ์ให้สำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณ ห้องปฏิบัติการกราฟิกส์ มหาวิทยาลัยสแตนฟอร์ด ที่กรุณาเอื้อเพื่อข้อมูลทดสอบสำหรับการวิจัย ขอขอบพระคุณ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ สำหรับทุนสนับสนุนงานวิจัยหรือวิทยานิพนธ์ โครงการปริญญาโท สาขาวิทยาการคอมพิวเตอร์ ภาคพิเศษ

ขอขอบคุณ คุณศศิธร มงคลศรีพัฒนา คุณสุวัชชัย ชื่นเมืองปักษ์ และ คุณวงกต พจน์พงษ์สวรรค์ รุ่นพี่ ที่กรุณาให้รายละเอียดบางประการเกี่ยวกับการจัดกลุ่มข้อมูลแบบต่างๆ ขอขอบคุณ คุณพ่อ คุณแม่ พี่สาว และทุกๆ คนที่คอยให้ความช่วยเหลือ ให้กำลังใจ ซึ่แนะและสนับสนุนการทำวิทยานิพนธ์จนสำเร็จลุล่วงได้

ธเนศ สุพันธ์
พฤษภาคม 2550

สารบัญ

	หน้า
สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(3)
คำนำ	1
วัตถุประสงค์	2
ประโยชน์ที่คาดว่าจะได้รับ	2
ขอบเขตและข้อจำกัด	2
การตรวจเอกสาร	3
ทฤษฎีที่เกี่ยวข้อง	3
งานวิจัยที่เกี่ยวข้อง	20
อุปกรณ์และวิธีการ	23
อุปกรณ์	23
วิธีการ	23
ผลและวิจารณ์	27
ผล	27
วิจารณ์	48
สรุปและข้อเสนอแนะ	50
สรุป	50
ข้อเสนอแนะ	50
เอกสารและสิ่งอ้างอิง	51
ภาคผนวก	54
ประวัติการศึกษา และการทำงาน	63

สารบัญตาราง

ตารางที่		หน้า
1	ความแตกต่างระหว่างต้นไม้ค้นหาแบบทวิภาคหลายมิติและต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ	10
2	รายละเอียดชุดข้อมูลสำหรับการทดลอง	23
3	รายละเอียดชุดการทดลองสำหรับการวัดเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว	25
4	การกำหนดค่าพารามิเตอร์ของโปรแกรมซูเปอร์โคโคน	25
5	การกำหนดค่าพารามิเตอร์ของโปรแกรมเมโท	26
6	เวลาที่ใช้ในการลดความซับซ้อนของพื้นผิวบนชุดข้อมูลต่างๆ เมื่อขนาดกลุ่มข้อมูลคงที่ (8 จุด)	27
7	ขนาดของชุดข้อมูลที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลคงที่ (8 จุด)	28
8	เวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว Dragon เมื่อขนาดกลุ่มเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด	30
9	ขนาดของชุดข้อมูล Dragon ที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด	31
10	เวลาที่ใช้ในการสร้างพื้นผิวด้านฉบับและพื้นผิวที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่ากับ 8 จุด	33
11	เวลาที่ใช้ในการสร้างพื้นผิวที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด	37
12	ความผิดพลาดที่เกิดขึ้นกับพื้นผิวต่างๆ เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่ากับ 8 จุด	41
13	ความผิดพลาดที่เกิดขึ้นกับพื้นผิว Dragon เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด	45

สารบัญภาพ

ภาพที่	หน้า
1 อัลกอริทึมต้นไม้ค้นหาแบบทวิภาคหลายมิติ	3
2 ลักษณะการเลือกตัวกลาง (Median) จากข้อมูลทั้งหมด	4
3 ลักษณะการเลือกตัวกลาง (Median) จากกลุ่มข้อมูล	5
4 ลักษณะต้นไม้ที่ได้จากอัลกอริทึมต้นไม้ค้นหาแบบทวิภาคหลายมิติ	5
5 อัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ	6
6 ลักษณะจุดศูนย์กลาง (Centroid) ที่คำนวณได้จากข้อมูลทั้งหมด	7
7 ลักษณะจุดศูนย์กลาง (Centroid) ที่คำนวณได้จากกลุ่มข้อมูล	8
8 ลักษณะกลุ่มข้อมูลที่จัดได้จากอัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ	8
9 ลักษณะกลุ่มจุดผลลัพธ์ที่ได้จากอัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ	9
10 อัลกอริทึมการจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง	10
11 ลักษณะจุดศูนย์กลาง (Centroid) ที่คำนวณได้จากข้อมูลทั้งหมด	11
12 ลักษณะจุดศูนย์กลาง (Centroid) ที่คำนวณได้จากกลุ่มข้อมูล	12
13 ลักษณะกลุ่มข้อมูลที่แบ่งได้จากอัลกอริทึมการจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง	12
14 ลักษณะกลุ่มจุดผลลัพธ์ที่ได้จากอัลกอริทึมการจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง	13
15 อัลกอริทึมการจัดกลุ่มแบบค่อยเพิ่มขึ้น	14
16 แสดงจุดเริ่มต้น (Seed) ที่สุ่มได้จากข้อมูลทั้งหมด	14
17 ลักษณะกลุ่มข้อมูลกลุ่มแรกที่จัดได้	15
18 ลักษณะการกำหนดจุดเริ่มต้นใหม่	15
19 ลักษณะกลุ่มข้อมูลที่จัดได้จากอัลกอริทึมการจัดกลุ่มแบบค่อยเพิ่มขึ้น	16
20 ลักษณะกลุ่มจุดผลลัพธ์ที่ได้จากอัลกอริทึมการจัดกลุ่มแบบค่อยเพิ่มขึ้น	16
21 อัลกอริทึมการจัดกลุ่มแบบเค-มีน	17
22 ลักษณะจุดศูนย์กลางของแต่ละกลุ่มที่สุ่มจากข้อมูล	18
23 ลักษณะกลุ่มข้อมูลที่จัดได้จากอัลกอริทึมการจัดกลุ่มแบบเค-มีน	18
24 ลักษณะจุดศูนย์กลางของแต่ละกลุ่มที่คำนวณได้	19
25 ลักษณะของกลุ่มจุดผลลัพธ์ที่ได้จากอัลกอริทึมการจัดกลุ่มแบบเค-มีน	19

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
26 การลดความซับซ้อนของพื้นผิวประเภทการจัดกลุ่มที่สัมพันธ์กัน	22
27 ขั้นตอนการทดลองในงานวิจัย	24
28 เปรียบเทียบเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว เมื่อขนาดชุดข้อมูลนำเข้าเปลี่ยนแปลงและขนาดกลุ่มข้อมูลคงที่ (8 จุด)	28
29 เปรียบเทียบขนาดชุดข้อมูลที่ผ่านการลดความซับซ้อน เมื่อขนาดชุดข้อมูลนำเข้าเปลี่ยนแปลงและขนาดกลุ่มข้อมูลคงที่ (8 จุด)	29
30 เปรียบเทียบที่ใช้ในการลดความซับซ้อนของพื้นผิว เมื่อขนาดชุดข้อมูลนำเข้าคงที่ (Dragon) และขนาดกลุ่มข้อมูลเปลี่ยนแปลง	30
31 เปรียบเทียบขนาดชุดข้อมูล Dragon ที่ผ่านการลดความซับซ้อน เมื่อขนาดชุดข้อมูลนำเข้าคงที่ (Dragon) และขนาดกลุ่มข้อมูลเปลี่ยนแปลง	32
32 เปรียบเทียบที่ใช้ในการสร้างพื้นผิวด้านฉบับและพื้นผิวที่ผ่านการลดความซับซ้อนด้วยกลุ่มข้อมูลเท่ากับ 8 จุด	33
33 ลักษณะพื้นผิว Bunny ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด	34
34 ลักษณะพื้นผิว Horse ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด	34
35 ลักษณะพื้นผิว Armadillo ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด	35
36 ลักษณะพื้นผิว Skeleton Hand ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด	35
37 ลักษณะพื้นผิว Dragon ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด	35
38 ลักษณะพื้นผิว Happy Buddha ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด	36

สารบัญภาพ (ต่อ)

ภาพที่	หน้า	
39	ลักษณะพื้นผิว Turbine Blade ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด	36
40	เปรียบเทียบเวลาที่ใช้ในการสร้างพื้นผิว Dragon ที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด	38
41	ลักษณะพื้นผิว Dragon ที่ลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ	38
42	ลักษณะพื้นผิว Dragon ที่ลดความซับซ้อนด้วยการจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง	39
43	ลักษณะพื้นผิว Dragon ที่ลดความซับซ้อนด้วยการจัดกลุ่มแบบค่อยเพิ่มขึ้น	39
44	ลักษณะพื้นผิว Dragon ที่ลดความซับซ้อนด้วยการจัดกลุ่มแบบเค-มีน	40
45	เปรียบเทียบความผิดพลาดที่เกิดขึ้นกับพื้นผิวที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด	41
46	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Bunny เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด	42
47	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Horse เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด	42
48	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Armadillo เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด	43
49	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Skeleton Hand เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด	43
50	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด	43
51	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Happy Buddha เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด	44
52	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Turbine Blade เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด	44

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
53	เปรียบเทียบความผิดพลาดที่เกิดขึ้นกับพื้นผิว Dragon ที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด	45
54	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อผ่านการลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ	46
55	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อผ่านการลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบลำดับชั้นจากบนลงล่าง	46
56	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อผ่านการลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบค่อยเพิ่มขึ้น	47
57	ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อผ่านการลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบเค-มิน	47

คำอธิบายสัญลักษณ์และคำย่อ

Avg.	=	Average Distance
DHC	=	Divisive Hierarchical Clustering
INC	=	Incremental Clustering
KMC	=	K-Mean Clustering
Max.	=	Maximum Distance
MBC Tree	=	Multidimensional Binary Clustering Tree

ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติสำหรับการลดความซับซ้อนของพื้นผิวสามมิติ

Multidimensional Binary Clustering Tree for 3D Surface Simplification

คำนำ

ปัจจุบันพื้นผิวของแบบจำลองสามมิติที่ได้จากเครื่องสแกนเนอร์สามมิตินั้น ประกอบด้วยกลุ่มจุดที่มีขนาดใหญ่โตขึ้นเรื่อยๆ (Lindstrom, 2001) ทั้งนี้เนื่องมาจากงานด้านคอมพิวเตอร์กราฟิกส์มีความจำเป็นต้องใช้งานพื้นผิวของแบบจำลองสามมิติที่มีความละเอียดสูงขึ้นนั่นเอง ด้วยเหตุนี้จึงส่งผลให้การแสดงผลแบบจำลองสามมิติของโปรแกรมประยุกต์ด้านคอมพิวเตอร์กราฟิกส์ต้องใช้เวลานาน ใช้หน่วยความจำมาก และโต้ตอบกับผู้ใช้งานโปรแกรมค่อนข้างช้า (Eric and Garland, 2001) ดังนั้นโปรแกรมประยุกต์ด้านคอมพิวเตอร์กราฟิกส์จึงมีการนำเอาขั้นตอนวิธีการลดความซับซ้อนของพื้นผิวเข้ามาช่วย เพื่อทำให้โปรแกรมประยุกต์ด้านคอมพิวเตอร์กราฟิกส์สามารถแสดงผลได้เร็วขึ้น ใช้หน่วยความจำน้อยลงและโต้ตอบกับผู้ใช้งานได้อย่างทันท่วงที

การลดความซับซ้อนของพื้นผิว (Surface Simplification) คือการทำให้พื้นผิวที่มีความซับซ้อนสูง มีความซับซ้อนลดน้อยลง เพื่อเป็นการเพิ่มความเร็วในการแสดงผลของโปรแกรมโต้ตอบทางด้านคอมพิวเตอร์กราฟิกส์ (Lindstrom and Turk, 1998) ขั้นตอนวิธีการลดความซับซ้อนของพื้นผิวมีด้วยกันมากมายหลายวิธี แต่โดยทั่วไปแล้ว ขั้นตอนวิธีการลดความซับซ้อนของพื้นผิวแบ่งออกเป็น 3 ประเภทใหญ่ๆ ได้แก่ การกำจัดจุดที่ไม่จำเป็น (Vertex Decimation) การจัดกลุ่มจุดที่สัมพันธ์กัน (Vertex Clustering) และการหดหรือรวบเส้นขอบที่ไม่จำเป็น (Edge Contraction) ซึ่งขั้นตอนวิธีการลดความซับซ้อนของพื้นผิวแต่ละประเภทก็มีข้อดีและข้อเสียแตกต่างกันไป (Garland and Heckbert, 1997)

งานวิจัยนี้นำเสนอขั้นตอนวิธีการลดความซับซ้อนของพื้นผิวของแบบจำลองสามมิติที่เรียกว่า ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ (Multidimensional Binary Clustering Tree) หรือ เอ็มบีซีทีรี (MBC Tree) ซึ่งเป็นการลดความซับซ้อนของพื้นผิวประเภทการจัดกลุ่มจุดที่สัมพันธ์กันเข้าไว้ด้วยกัน โดยขั้นตอนวิธีนี้พัฒนาและปรับปรุงมาจากขั้นตอนวิธีต้นไม้ค้นหาแบบทวิภาคหลายมิติ (Multidimensional Binary Search Tree) หรือเค-ดีทีรี (Kd-tree) ซึ่งถูกออกแบบมาเพื่อใช้ในการค้นหาข้อมูลที่มีความสัมพันธ์กันหรือใกล้ชิดกัน (Bentley, 1975)

วัตถุประสงค์

1. เพื่อนำเสนออัลกอริทึมลดความซับซ้อนของพื้นผิวด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติที่สามารถทำงานได้อย่างรวดเร็ว
2. เพื่อเปรียบเทียบเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิวระหว่างต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ และการจัดกลุ่มแบบอื่นๆ
3. เพื่อเปรียบเทียบเวลาที่ใช้ในการสร้างพื้นผิวที่สร้างจากข้อมูลที่ผ่านการลดความซับซ้อนมาแล้ว
4. เพื่อเปรียบเทียบคุณภาพของพื้นผิวระหว่างพื้นผิวด้านลบและพื้นผิวที่ผ่านการลดความซับซ้อน

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ขั้นตอนการลดความซับซ้อนของพื้นผิวด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติที่ทำงานได้อย่างรวดเร็ว
2. เพื่อเป็นประโยชน์ในการศึกษาค้นคว้างานทางด้านการลดความซับซ้อนของพื้นผิวประเภทการจัดกลุ่มจุดที่สัมพันธ์กัน

ขอบเขตและข้อจำกัด

1. ทดลองกับชุดข้อมูล Bunny, Horse, Armadillo, Skeleton Hand, Dragon, Happy Buddha และ Turbine Blade
2. ใช้เวลาในการประมวลผลและคุณภาพของพื้นผิวในการวัดประสิทธิภาพ โดยเปรียบเทียบกันระหว่างต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ การจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง การจัดกลุ่มแบบค่อยเพิ่มขึ้น และการจัดกลุ่มแบบเค-มีน

การตรวจเอกสาร

ทฤษฎีที่เกี่ยวข้อง

การลดความซับซ้อนของพื้นที่มีความสำคัญต่อโปรแกรมโต้ตอบทางด้านคอมพิวเตอร์กราฟิกส์ กล่าวคือ สามารถทำให้การแสดงผลเร็วขึ้น ใช้หน่วยความจำของเครื่องคอมพิวเตอร์น้อยลง ส่งผลให้โปรแกรมสามารถโต้ตอบกับผู้ใช้ได้อย่างทันท่วงที

งานวิจัยนี้เป็นการนำเสนอขั้นตอนการลดความซับซ้อนของพื้นที่ผิวประเภทการจัดกลุ่มจุดที่สัมพันธ์กันเข้าไว้ด้วยกัน ดังนั้นบทนี้จะอธิบายถึงอัลกอริทึมต่างๆ ที่นำมาใช้ในงานวิจัยนี้ ซึ่งมีดังต่อไปนี้

1. การลดความซับซ้อนของพื้นผิวด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ

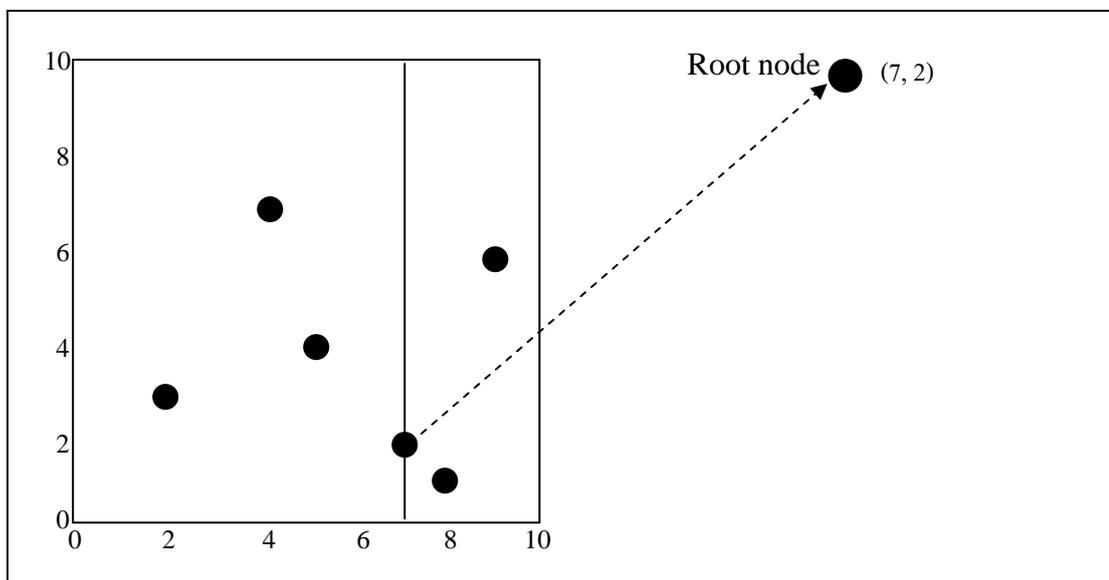
การลดความซับซ้อนของพื้นผิวด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติพัฒนาและปรับปรุงมาจากอัลกอริทึมต้นไม้ค้นหาแบบทวิภาคหลายมิติหรือเค-ดีทรี ซึ่งพัฒนาและนำเสนอ โดย Bentley (1975) เพื่อใช้ในการค้นหาข้อมูลที่มีความสัมพันธ์กันหรือใกล้ชิดกัน อัลกอริทึมต้นไม้ค้นหาแบบทวิภาคหลายมิติแสดงไว้ดังภาพที่ 1

```
kdtree (list_of_points pointList, int depth)
  if pointList is empty
    return null
  else
    int axis = depth mod k //dimension of data
    select median from pointList
    tree_node node
    node.location = median
    node.leftChild = kdtree(pointList before median, depth+1)
    node.rightChild = kdtree(pointList after median, depth+1)
    return node
```

ภาพที่ 1 อัลกอริทึมต้นไม้ค้นหาแบบทวิภาคหลายมิติ

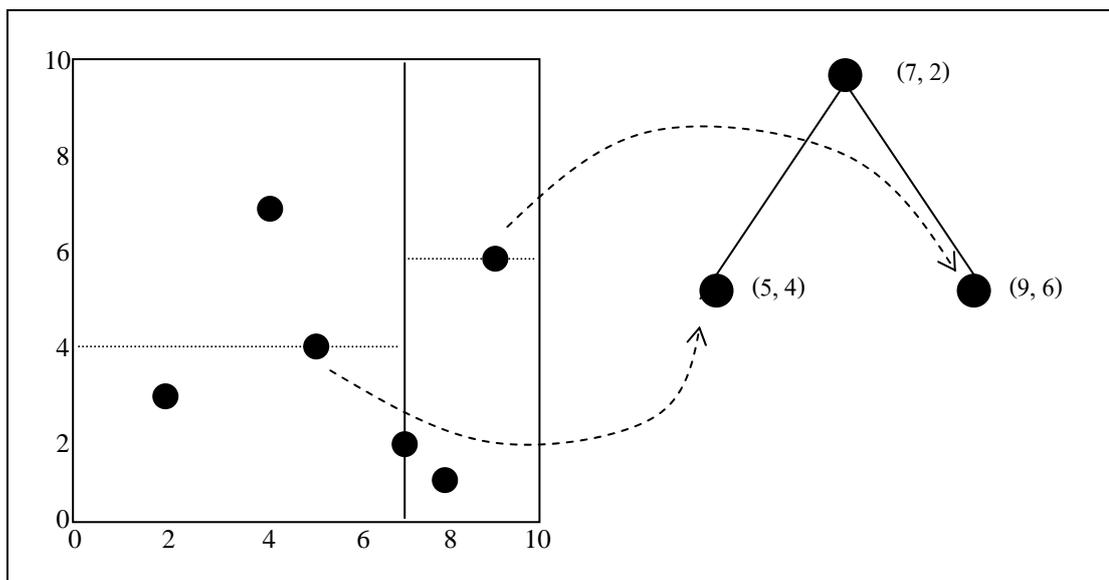
จากอัลกอริทึมข้างต้นมีลักษณะการทำงานแบบเรียกซ้ำ (Recursive) และใช้เวลาในการประมวลผลเท่ากับ $O(n \log n)$ เมื่อ n คือขนาดของข้อมูลที่ต้องการค้นหา (Bentley, 1975)

ตัวอย่างการทำงานกับข้อมูลสองมิติของอัลกอริทึมต้นไม้ค้นหาแบบทวิภาคหลายมิติ แสดงไว้ดังภาพต่อไปนี้



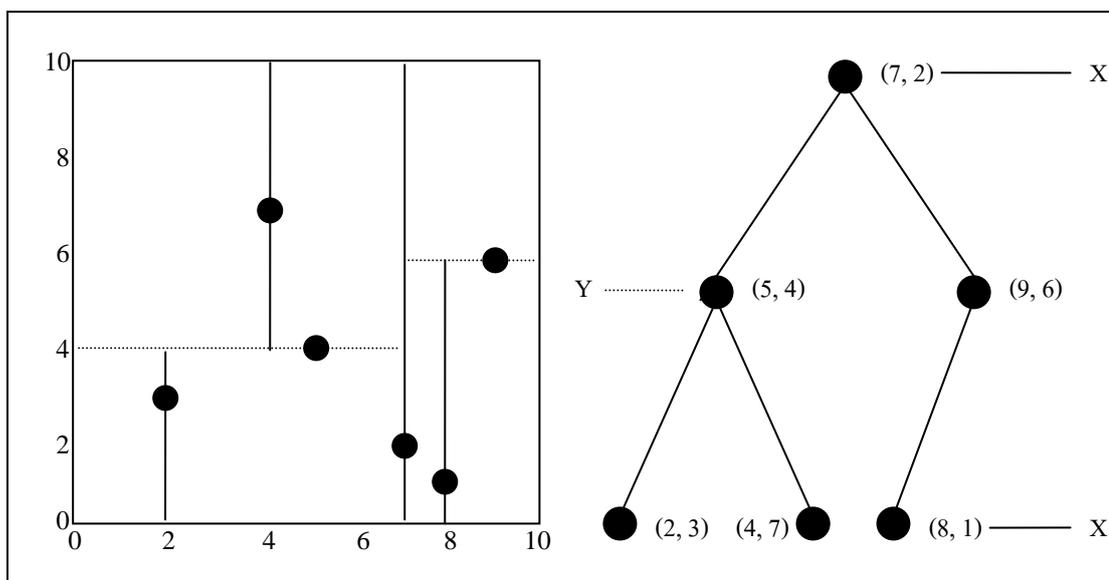
ภาพที่ 2 ลักษณะการเลือกตัวกลาง (Median) จากข้อมูลทั้งหมด

จากภาพที่ 2 แสดงการเลือกตัวกลาง (Median) จากข้อมูลที่ถูกจัดเรียงตามแนวแกน X (Axis) ซึ่งจะเห็นว่าตัวกลางของข้อมูลมีด้วยกัน 2 ตัว คือ (5, 4) และ (7, 2) ซึ่งในที่นี้เราเลือก (7, 2) ซึ่งมีค่าในแนวแกน X มากกว่า ให้เป็นตัวกลางของข้อมูลและให้เป็นโหนดราก (Root node) จากนั้นเราใช้ตัวกลางนี้แบ่งข้อมูลออกเป็นสองกลุ่ม ได้แก่ กลุ่มข้อมูลก่อนตัวกลางและกลุ่มข้อมูลหลังตัวกลาง ดังภาพที่ 2



ภาพที่ 3 ลักษณะการเลือกตัวกลาง (Median) จากกลุ่มข้อมูล

จากภาพที่ 3 แสดงการเลือกตัวกลางจากกลุ่มข้อมูลทั้งสองกลุ่ม เพื่อสร้างเป็นโหนดทางซ้าย (Left Child) และโหนดทางขวา (Right Child) การเลือกตัวกลางใช้วิธีการเช่นเดียวที่กล่าวมาแล้วข้างต้น แต่จะจัดเรียงข้อมูลตามแนวแกน Y แทน ดังภาพที่ 3



ภาพที่ 4 ลักษณะต้นไม้ที่ได้จากอัลกอริทึมต้นไม้ค้นหาแบบทวิภาคหลายมิติ

จากภาพที่ 4 แสดงลักษณะต้นไม้ที่สร้างจากอัลกอริทึม จากต้นไม้เราสังเกตเห็นได้ว่า ในกรณีข้อมูลสองมิตินี้ โหนดแต่ละชั้นของต้นไม้จะเป็นตัวกลางที่เลือกจากข้อมูลที่จัดเรียงตามแนวแกน X และ Y สลับกันไปเรื่อยๆ

จากอัลกอริทึมต้นไม้ค้นหาแบบทวิภาคหลายมิติข้างต้น เราแก้ไขและปรับปรุงการทำงานของอัลกอริทึมเดิมเพื่อให้ได้อัลกอริทึมสำหรับการลดความซับซ้อนของพื้นที่ผิวที่เรียกว่า ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ โดยมีอัลกอริทึมดังภาพที่ 5

```

mbctree (list_of_points pointList, int depth)
  if pointList is empty
    return null
  else
    tree_node node
    calculate centroid of pointList
    int axis = depth mod k //dimension of data
    if size of pointList less than or equal to clustSize
      node.location = centroid
      node.leftChild = null
      node.rightChild = null
    else
      node.location = centroid
      node.leftChild = mbctree(pointList before centroid, depth+1)
      node.rightChild = mbctree(pointList after centroid, depth+1)
    return node

```

ภาพที่ 5 อัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ

จากภาพที่ 5 อัลกอริทึมนี้มีลักษณะการทำงานแบบเรียกซ้ำ (Recursive) และใช้เวลาในการประมวลผลเท่ากับ $O(n \log n)$ เมื่อ n คือขนาดของกลุ่มข้อมูลที่ต้องการลดความซับซ้อน

จากอัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติข้างต้น การเปลี่ยนแปลงที่เพิ่มเข้ามาได้แก่ การตรวจสอบขนาดกลุ่มข้อมูล (clustSize) ซึ่งกำหนดโดยผู้ใช้งานและการคำนวณจุดศูนย์กลาง (Centroid) ซึ่งคำนวณได้จากสมการต่อไปนี้

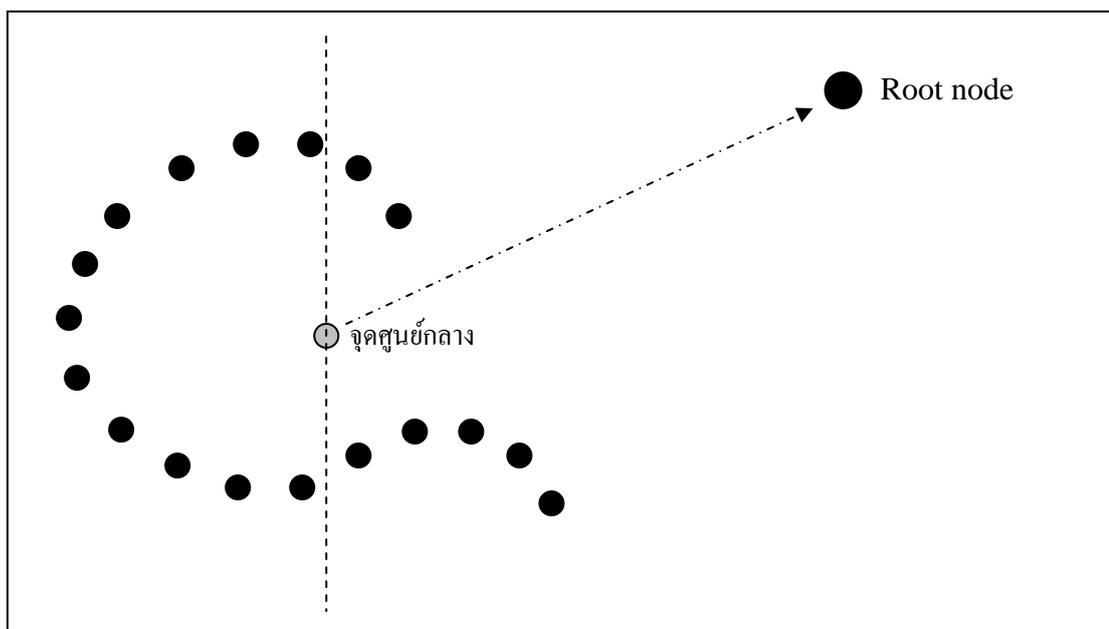
$$Centroid(x, y, z) = \left(\frac{\sum_{i=1}^n X_i}{n}, \frac{\sum_{i=1}^n Y_i}{n}, \frac{\sum_{i=1}^n Z_i}{n} \right) \quad (1)$$

เมื่อ n คือขนาดของกลุ่มข้อมูลที่ต้องการลดความซับซ้อน

i คือจุดที่พิจารณา

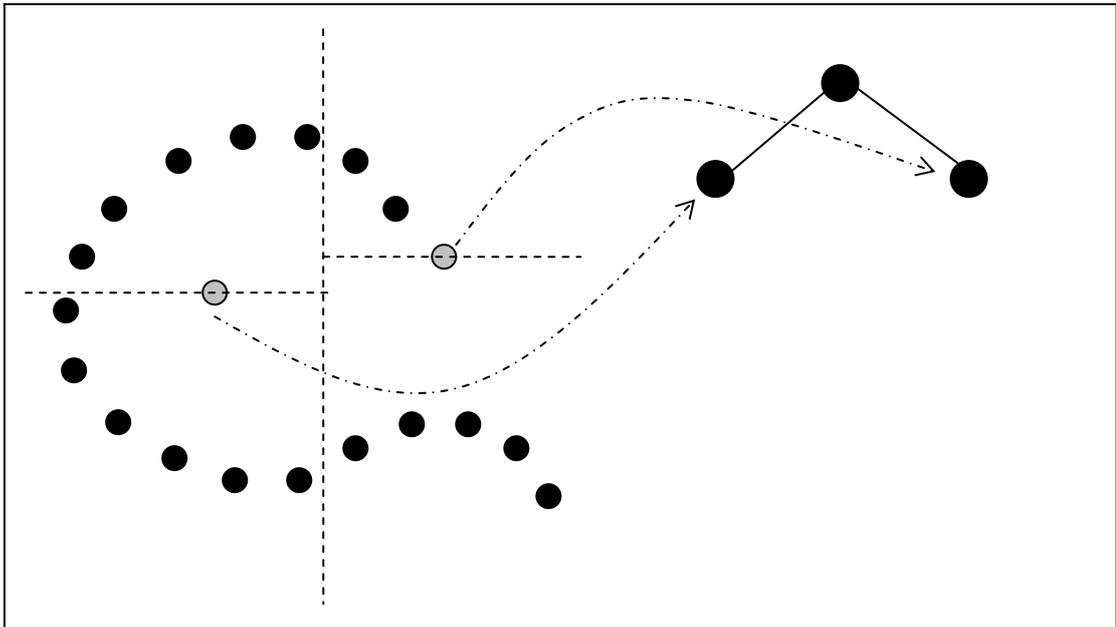
X, Y, Z คือค่าที่แกน X, Y และ Z ตามลำดับ

ตัวอย่างการใช้งานกับข้อมูลสองมิติของอัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ แสดงไว้ดังภาพต่อไปนี้



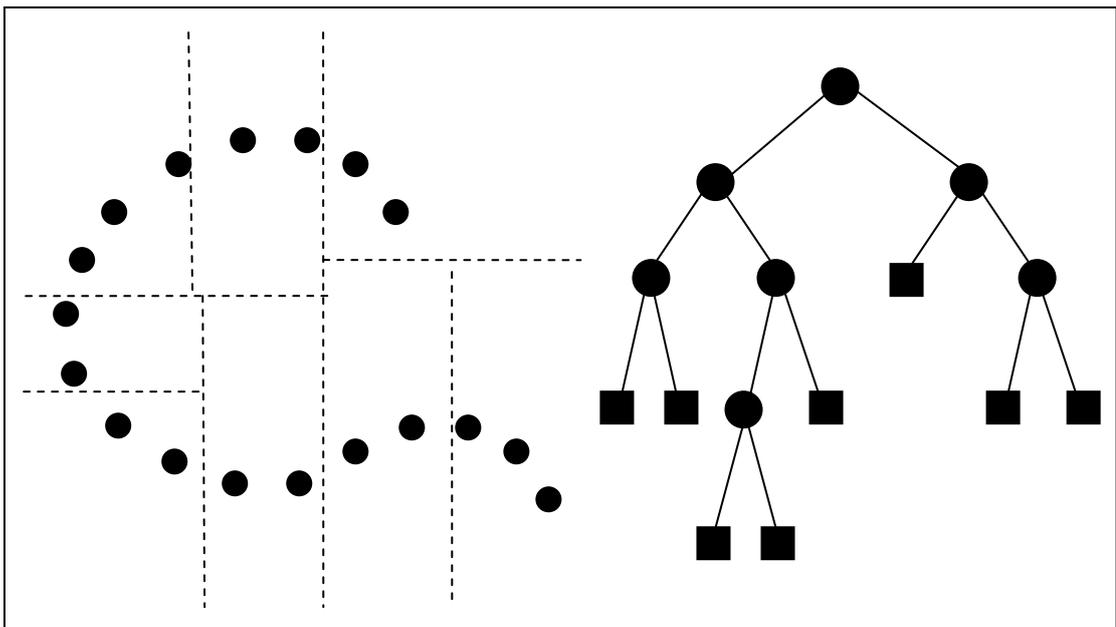
ภาพที่ 6 ลักษณะจุดศูนย์กลาง (Centroid) ที่คำนวณได้จากข้อมูลทั้งหมด

จากภาพที่ 6 แสดงจุดศูนย์กลางที่คำนวณจากข้อมูลทั้งด้วยสมการ (1) จากนั้นเปรียบเทียบค่าที่แนวแกน X (Axis) ของข้อมูลแต่ละตัว (สีดำ) กับจุดศูนย์กลาง (สีเทา) เพื่อแบ่งข้อมูลออกเป็นสองกลุ่ม ได้แก่ กลุ่มข้อมูลที่อยู่ทางด้านซ้ายของจุดศูนย์กลางและกลุ่มข้อมูลที่อยู่ทางด้านขวาของจุดศูนย์กลาง ดังภาพที่ 6



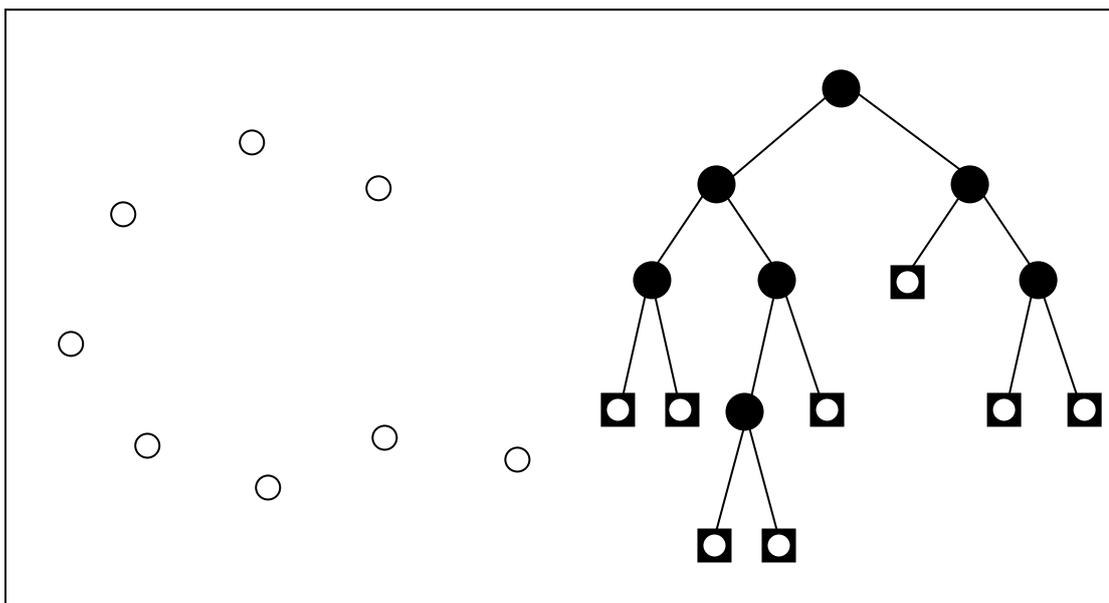
ภาพที่ 7 ลักษณะจุดศูนย์กลาง (Centroid) ที่คำนวณได้จากกลุ่มข้อมูล

จากรูปที่ 7 แสดงจุดศูนย์กลางที่คำนวณได้จากกลุ่มข้อมูลทั้งสองกลุ่ม (จุดสีเทา) ด้วยสมการ (1) ซึ่งจะถูกนำไปสร้างเป็นโหนดทางซ้าย (Left Child) และโหนดทางขวา (Right Child) ข้อมูลแต่ละกลุ่มจะถูกแบ่งออกเป็นสองกลุ่มด้วยวิธีการเดิม แต่จะเปลี่ยนมาพิจารณาค่าที่แกน Y แทน ดังภาพที่ 7



ภาพที่ 8 ลักษณะกลุ่มข้อมูลที่จัดได้จากอัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ

จากภาพที่ 8 แสดงลักษณะกลุ่มข้อมูลที่จัดได้จากอัลกอริทึม เมื่อการทำงานของอัลกอริทึมจบลง โดยรูปสี่เหลี่ยมที่ใบของต้นไม้คือ กลุ่มข้อมูล จะสังเกตเห็นว่าการแบ่งข้อมูลในแต่ละชั้นของต้นไม้จะสลับแกน X และ Y ไปเรื่อยๆ (กรณีข้อมูลสองมิติ) จนขนาดของแต่ละกลุ่มข้อมูลมีขนาดเท่ากับที่กำหนดไว้ (ตัวแปร `clustSize` กำหนดขนาดของแต่ละกลุ่มข้อมูลดังแสดงไว้ในภาพที่ 8)



ภาพที่ 9 ลักษณะกลุ่มจุดผลลัพธ์ที่ได้จากอัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ

จากภาพที่ 9 แสดงลักษณะผลลัพธ์ (จุดสีขาว) ซึ่งได้จากจุดศูนย์กลางของแต่ละกลุ่มข้อมูล หรือ โหนดใบของต้นไม้ (Leaf node) จะเห็นได้ว่าผลลัพธ์ที่ได้มีจำนวนของข้อมูลลดลง ในขณะที่ลักษณะของข้อมูลยังคงคล้ายข้อมูลเดิม ผลลัพธ์นี้จะถูกนำไปสร้างเป็นพื้นผิวต่อไป

จากลักษณะการทำงานของอัลกอริทึมทั้งสองที่กล่าวมาข้างต้น สามารถสรุปความแตกต่างระหว่างอัลกอริทึมทั้งสองได้ดังตารางที่ 1

ตารางที่ 1 ความแตกต่างระหว่างต้นไม้ค้นหาแบบทวิภาคหลายมิติและต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ

ลักษณะ	ต้นไม้ค้นหาแบบทวิภาคหลายมิติ	ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ
โหนด	ได้จากตัวกลาง (Median) ของข้อมูลหลังจากจัดเรียงข้อมูลแล้ว	ได้จากการคำนวณจุดศูนย์กลางตามสมการ (1)
การเรียกซ้ำ	เรียกซ้ำจนกระทั่งข้อมูลทั้งหมดถูกสร้างเป็นโหนดบนต้นไม้	เรียกซ้ำจนกระทั่งทุกกลุ่มข้อมูลมีขนาดตามที่กำหนดไว้

2. การลดความซับซ้อนของพื้นที่ผิวด้วยการจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง

การลดความซับซ้อนของพื้นที่ผิวด้วยการจัดกลุ่มแบบลำดับชั้นจากบนลงล่างนำเสนอโดย Mark และคณะ (2002) อัลกอริทึมดังกล่าวมีการทำงานจากบนลงล่าง (Top-down) โดยจะแบ่งข้อมูลออกเป็นสองกลุ่ม ตรวจจับกลุ่มข้อมูลมีขนาดใหญ่กว่าขนาดที่กำหนดไว้ ซึ่งจะแบ่งตามแนวแบ่งข้อมูล อัลกอริทึมการจัดกลุ่มแบบลำดับชั้นจากบนลงล่างแสดงไว้ดังภาพที่ 10

```

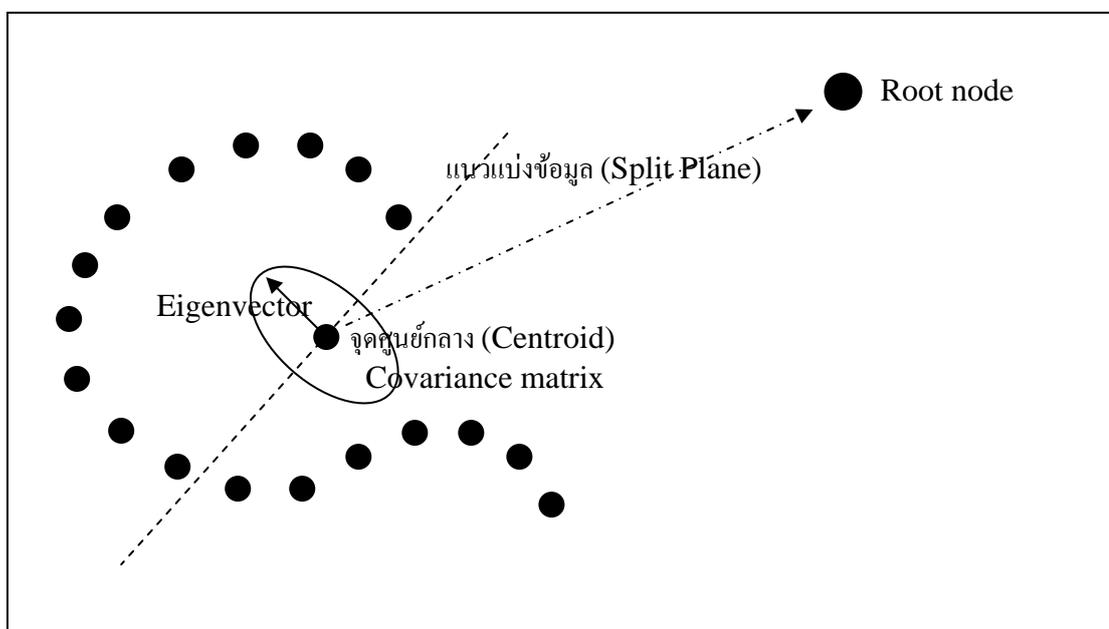
dhc (list_of_points pointList)
  if pointList is empty
    return null
  else
    tree_node node
    calculate centroid of pointList
    if size of pointList less than or equal to clustSize
      node.location = centroid
      node.leftChild = null
      node.rightChild = null
    else
      calculate covarianceMatrix of pointList
      calculate eigenvector of covarianceMatrix
      define splitPlane by centroid and eigenvector
      node.location = centroid
      node.leftChild = dhc(pointList before splitPlane)
      node.rightChild = dhc(pointList after splitPlane)
    return node

```

ภาพที่ 10 อัลกอริทึมการจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง

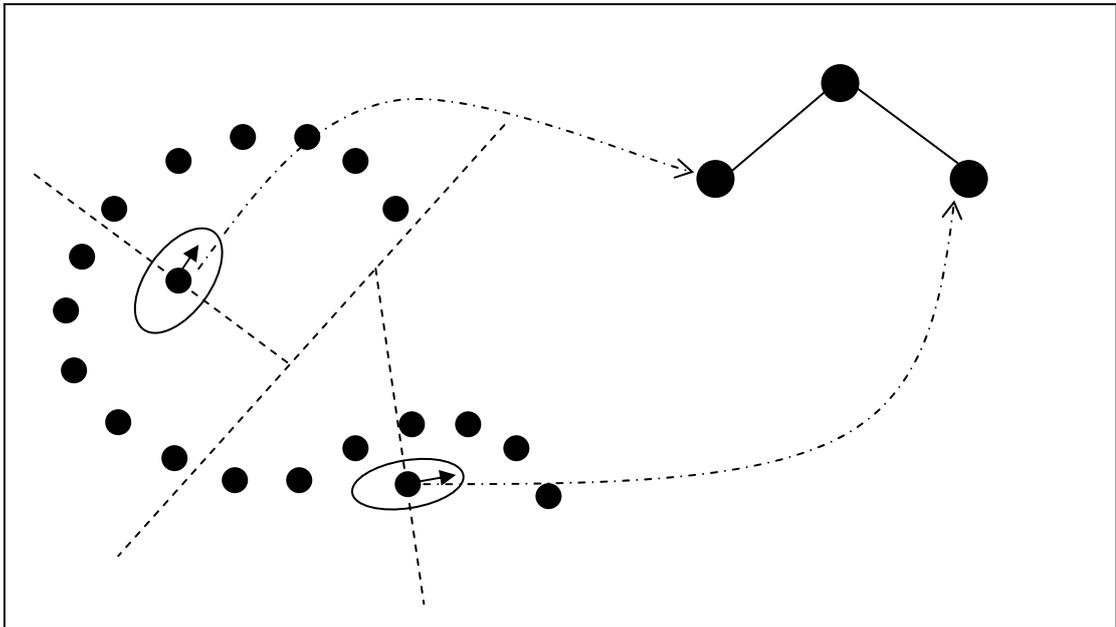
จากภาพที่ 10 อัลกอริทึมนี้มีลักษณะการทำงานแบบเรียกซ้ำ (Recursive) และใช้เวลาในการประมวลผลเท่ากับ $O(n \log n)$ เมื่อ n คือขนาดของข้อมูลที่ต้องการลดความซับซ้อน (Pauly *et al.*, 2002) การคำนวณจุดศูนย์กลางของข้อมูลสามารถคำนวณได้จากสมการ (1) เช่นกัน ส่วนการคำนวณ Covariance matrix และ Eigenvector สามารถดูรายละเอียดได้จากงานวิจัยของ Christopher และคณะ (2003)

ตัวอย่างการทำงานกับข้อมูลสองมิติของอัลกอริทึมการจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง แสดงไว้ดังภาพต่อไปนี้



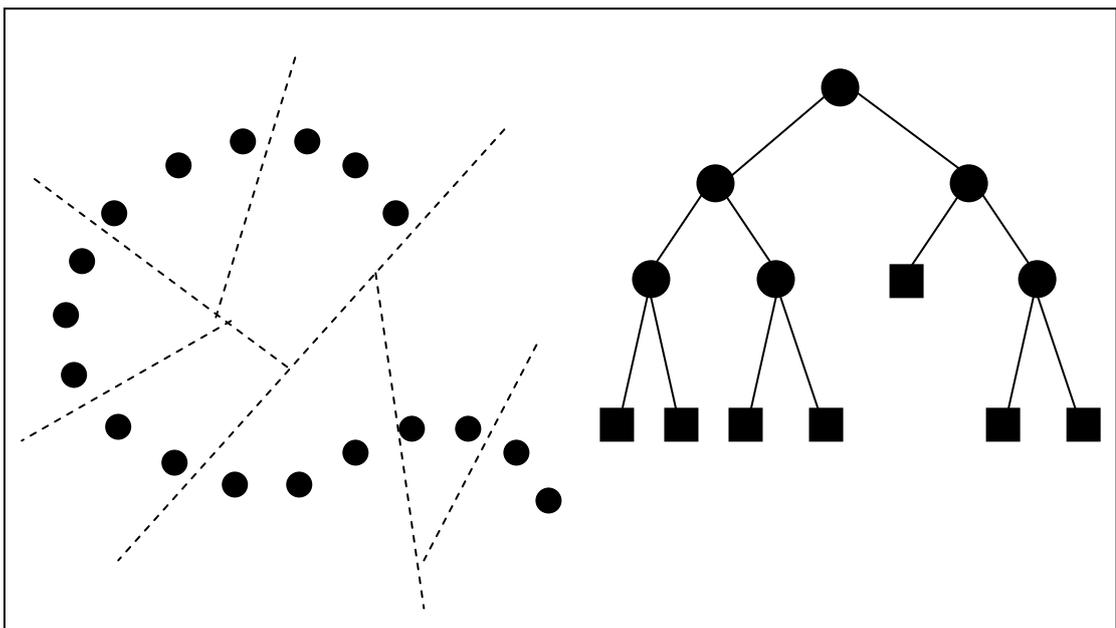
ภาพที่ 11 ลักษณะจุดศูนย์กลาง (Centroid) ที่คำนวณได้จากข้อมูลทั้งหมด

จากภาพที่ 11 แสดงลักษณะจุดศูนย์กลางที่คำนวณได้จากสมการ (1) ซึ่งจะถูกนำไปสร้างเป็น โหนดราก (Root node) และแนวแบ่งข้อมูล (Split plane) ที่กำหนดได้จากจุดศูนย์กลางและ Eigenvector ที่คำนวณได้จาก Covariance matrix จะเห็นว่าข้อมูลถูกแบ่งออกเป็นสองกลุ่มคือ กลุ่มข้อมูลที่อยู่ด้านเดียวกับ Eigenvector และกลุ่มข้อมูลที่อยู่ด้านตรงกันข้ามกับ Eigenvector ดังภาพที่



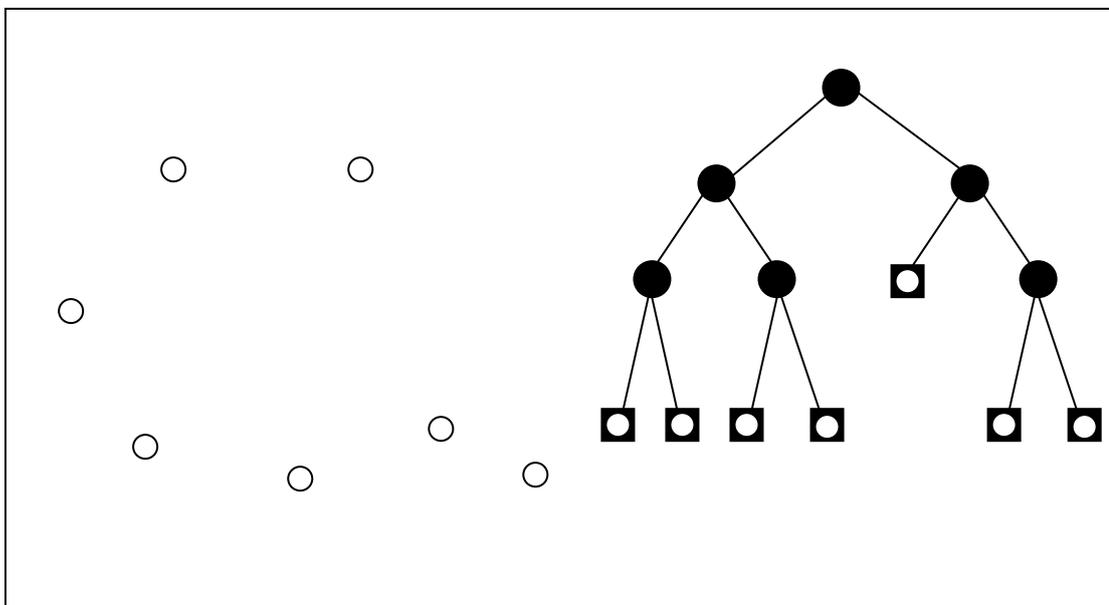
ภาพที่ 12 ลักษณะจุดศูนย์กลาง (Centroid) ที่คำนวณได้จากกลุ่มข้อมูล

จากภาพที่ 12 แสดงลักษณะจุดศูนย์กลางของแต่ละกลุ่มข้อมูลจากสมการ (1) เพื่อนำไปสร้างเป็นโหนดทางซ้าย (Left Child) และ โหนดทางขวา (Right Child) ข้อมูลจะถูกแบ่งออกเป็นสองกลุ่มข้อมูลตามแนวแบ่งข้อมูลที่คำนวณได้ของกลุ่มข้อมูลนั้นๆ ดังภาพที่ 12



ภาพที่ 13 ลักษณะกลุ่มข้อมูลที่แบ่งได้จากอัลกอริทึมการจัดกลุ่มแบบลำดับขั้นจากบนลงล่าง

จากภาพที่ 13 แสดงลักษณะกลุ่มข้อมูลที่จัดได้จากอัลกอริทึม เมื่ออัลกอริทึมทำงานจบลง โดยรูปสี่เหลี่ยมที่ใบของต้นไม้คือ กลุ่มข้อมูล



ภาพที่ 14 ลักษณะกลุ่มจุดผลลัพธ์ที่ได้จากอัลกอริทึมการจัดกลุ่มแบบลำดับขั้นจากบนลงล่าง

จากภาพที่ 14 แสดงลักษณะผลลัพธ์ (จุดสีขาว) ที่ได้จากจุดศูนย์กลางของแต่ละกลุ่มข้อมูล ซึ่งถูกจัดเก็บอยู่ในโหนดใบของต้นไม้ (Leaf node) ลักษณะผลลัพธ์ที่ได้จะเป็นกลุ่มข้อมูลใหม่ที่สามารถใช้ประมาณชุดข้อมูลเดิม แต่กลุ่มข้อมูลชุดใหม่นี้มีจำนวนข้อมูลลดลง ดังภาพที่ 14

3. การลดความซับซ้อนของพื้นที่ผิวด้วยการจัดกลุ่มแบบลำดับขั้นจากบนลงล่าง

การลดความซับซ้อนของพื้นที่ผิวด้วยการจัดกลุ่มแบบค่อยเพิ่มขึ้นนำเสนอโดย Mark และ คณะ (2002) อัลกอริทึมดังกล่าวจะเริ่มจากการสุ่มจุดเริ่มต้น (Seed) จากข้อมูลขึ้นมาหนึ่งจุด จากนั้น จะทำการจัดข้อมูลตัวที่อยู่ใกล้เข้ากลุ่มตามจำนวนที่กำหนด ทำเช่นนี้ไปเรื่อยๆ จนข้อมูลทุกตัวถูก จัดเข้ากลุ่ม อัลกอริทึมการจัดกลุ่มแบบค่อยเพิ่มขึ้นแสดงไว้ดังภาพที่ 15

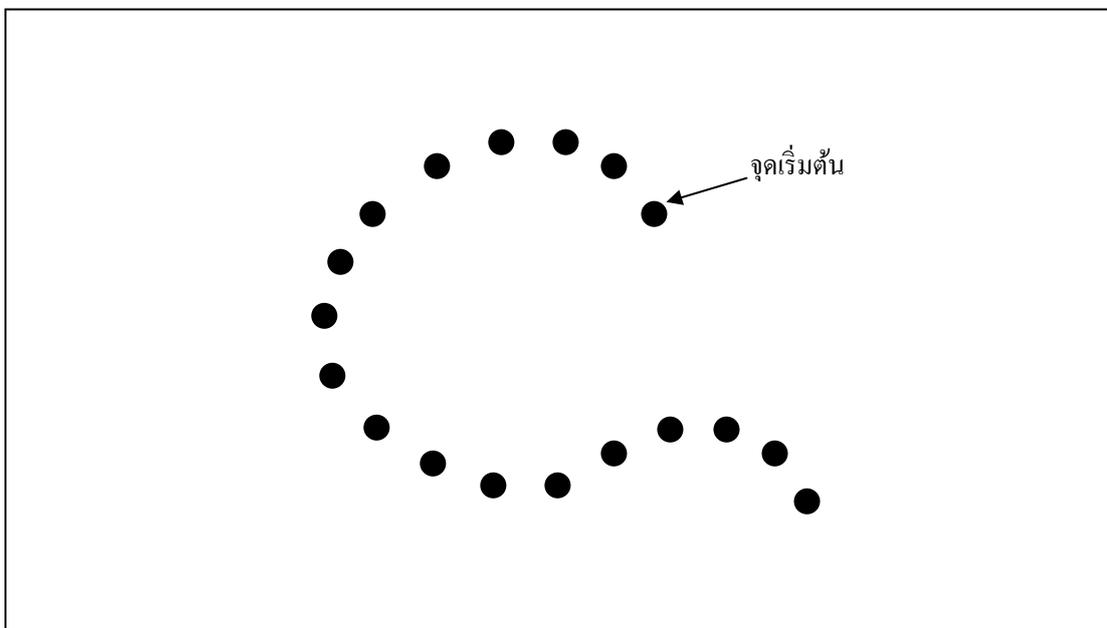
incremental

1. Decide a value of c .
2. Decide *seed* from all objects.
3. Calculate *distance* between *seed* and objects.
4. Decide the class memberships of the c objects by assigning them to the nearest *seed*.
5. If all objects decide the class membership, goto 6. Otherwise goto 2.
6. Calculate *centroid* from class memberships as new simplified points.

ภาพที่ 15 อัลกอริทึมการจัดกลุ่มแบบค่อยเพิ่มขึ้น

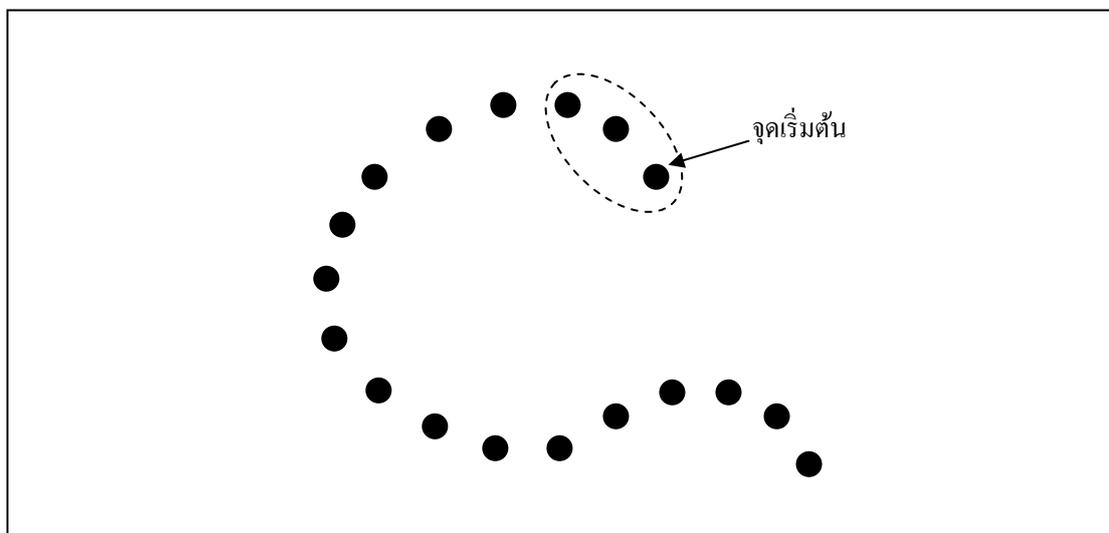
จากภาพที่ 15 อัลกอริทึมนี้ใช้เวลาในการประมวลผลเท่ากับ $O(n \log n)$ เมื่อ n คือขนาดของข้อมูลที่ต้องการลดความซับซ้อน (Pauly, 2002)

ตัวอย่างการทำงานกับข้อมูลสองมิติของอัลกอริทึมการจัดกลุ่มแบบค่อยเพิ่มขึ้น แสดงไว้ดังภาพต่อไปนี้



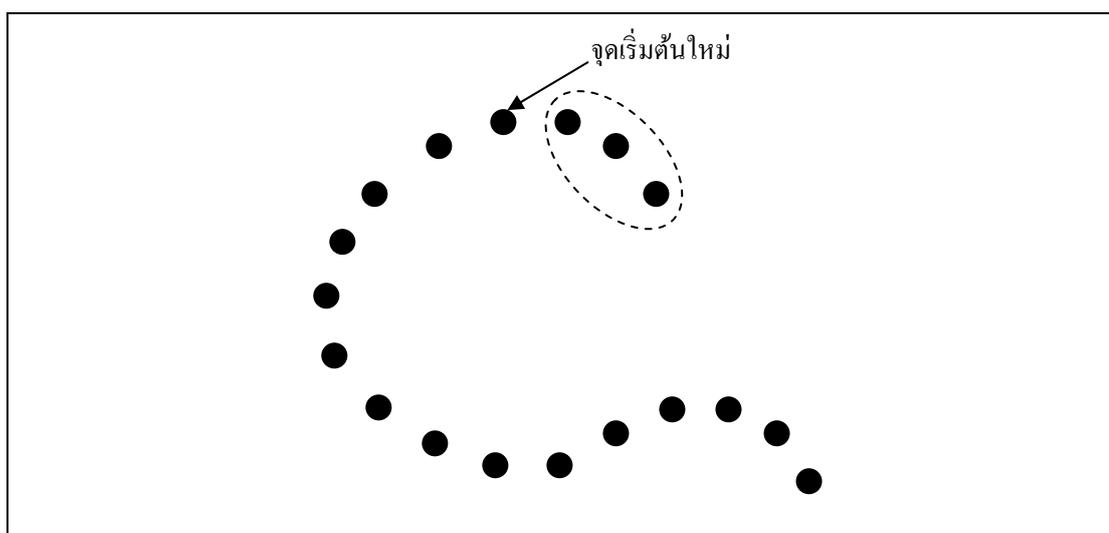
ภาพที่ 16 แสดงจุดเริ่มต้น (Seed) ที่สุ่มได้จากข้อมูลทั้งหมด

จากภาพที่ 16 แสดงลักษณะจุดเริ่มต้น (Seed) ที่สุ่มได้จากข้อมูล จากนั้นการวัดระยะห่างระหว่างจุดเริ่มต้นและข้อมูลแต่ละตัว เพื่อจัดข้อมูลตัวที่อยู่ใกล้กับจุดเริ่มต้นเข้ากลุ่มตามจำนวนที่กำหนดไว้



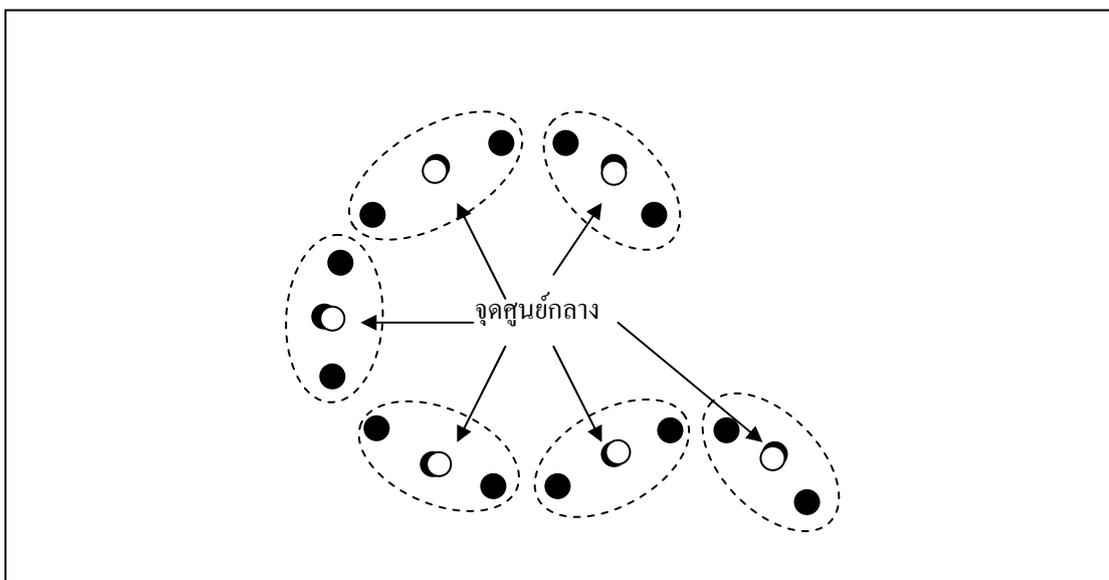
ภาพที่ 17 ลักษณะกลุ่มข้อมูลกลุ่มแรกที่จัดได้

จากภาพที่ 17 แสดงลักษณะกลุ่มข้อมูลที่จัดได้ตอนเริ่มต้น จากนั้นจะต้องทำการกำหนดจุดเริ่มต้นใหม่ โดยเลือกจากจุดที่อยู่ใกล้กับกลุ่มข้อมูลที่เพิ่งจัดไปมากที่สุด



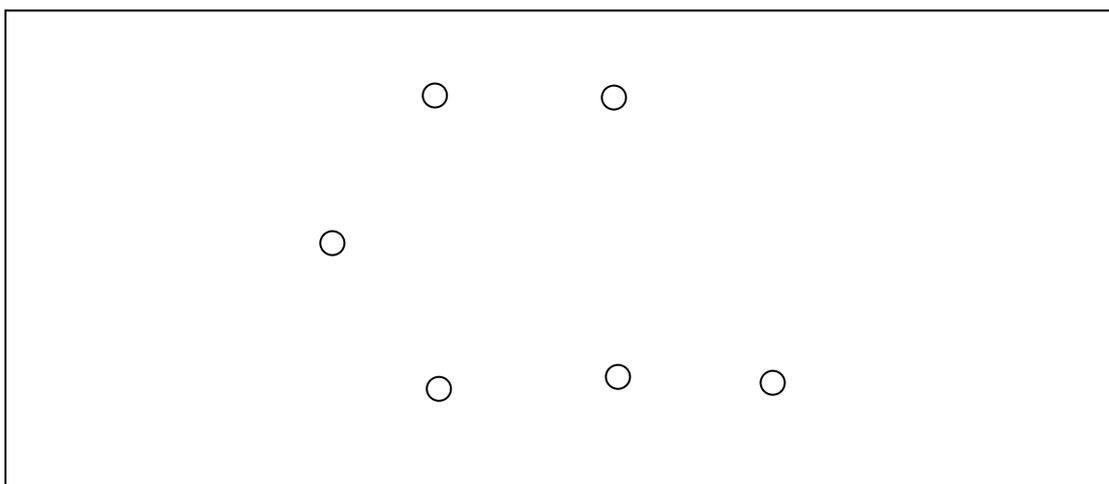
ภาพที่ 18 ลักษณะการกำหนดจุดเริ่มต้นใหม่

จากภาพที่ 18 แสดงลักษณะการกำหนดจุดเริ่มต้นใหม่ ซึ่งอยู่ใกล้กับกลุ่มข้อมูลที่เพิ่งจัดได้ จากนั้นทำการวัดระยะห่างระหว่างข้อมูลแต่ละตัวและจุดเริ่มต้นใหม่ เพื่อจัดข้อมูลเข้ากลุ่ม ทำเช่นนี้เรื่อยไปจนกระทั่งข้อมูลทุกตัวถูกจัดเข้ากลุ่ม



ภาพที่ 19 ลักษณะกลุ่มข้อมูลที่จัดได้จากอัลกอริทึมการจัดกลุ่มแบบค่อยเพิ่มขึ้น

จากภาพที่ 19 แสดงลักษณะกลุ่มข้อมูลที่จัดได้จากอัลกอริทึม จากนั้นคำนวณจุดศูนย์กลาง (Centroid) ของข้อมูลแต่ละกลุ่มตามสมการ (1) เพื่อสร้างจุดที่จะเป็นผลลัพธ์ (จุดสีขาว)



ภาพที่ 20 ลักษณะกลุ่มจุดผลลัพธ์ที่ได้จากอัลกอริทึมการจัดกลุ่มแบบค่อยเพิ่มขึ้น

จากภาพที่ 20 แสดงลักษณะผลลัพธ์ที่ได้จากอัลกอริทึม ผลลัพธ์ที่ได้มีจำนวนจุดที่น้อยลง และสามารถถูกนำมาใช้ประมาณข้อมูลต้นฉบับได้

4. การลดความซับซ้อนของพื้นที่ผิวด้วยการจัดกลุ่มแบบเค-มีน

เทคนิคการจัดกลุ่มข้อมูลโดยอัลกอริทึมเค-มีนนั้นได้รับการพัฒนาและนำเสนอโดย Mac Queen (1967) ซึ่งเป็นขั้นตอนในการจัดกลุ่มข้อมูลที่สมาชิกภายในแต่ละกลุ่มนั้นมีระยะอยู่ใกล้จุดศูนย์กลางหรือตัวแทนของกลุ่ม ขั้นตอนการจัดกลุ่มข้อมูลโดยอัลกอริทึมเค-มีนนั้นประกอบด้วย การกำหนดจำนวนกลุ่มเริ่มต้น การกำหนดตัวแทนกลุ่ม การจัดข้อมูลแต่ละตัวเข้ากลุ่ม และการปรับปรุงตัวแทนกลุ่มในแต่ละกลุ่มหรือคำนวณจุดศูนย์กลางใหม่

ถึงแม้อัลกอริทึมการจัดกลุ่มแบบเค-มีนถูกนำมาใช้ในการจัดกลุ่มข้อมูล แต่ปัญหาที่พบคือการกำหนดจำนวนจุดเริ่มต้น ซึ่งส่งผลกระทบต่อประสิทธิภาพของการจัดกลุ่ม หรือการที่กลุ่มบางกลุ่มนั้นมีจำนวนสมาชิกน้อยเกิน หรือไม่มีสมาชิกในกลุ่มเลย

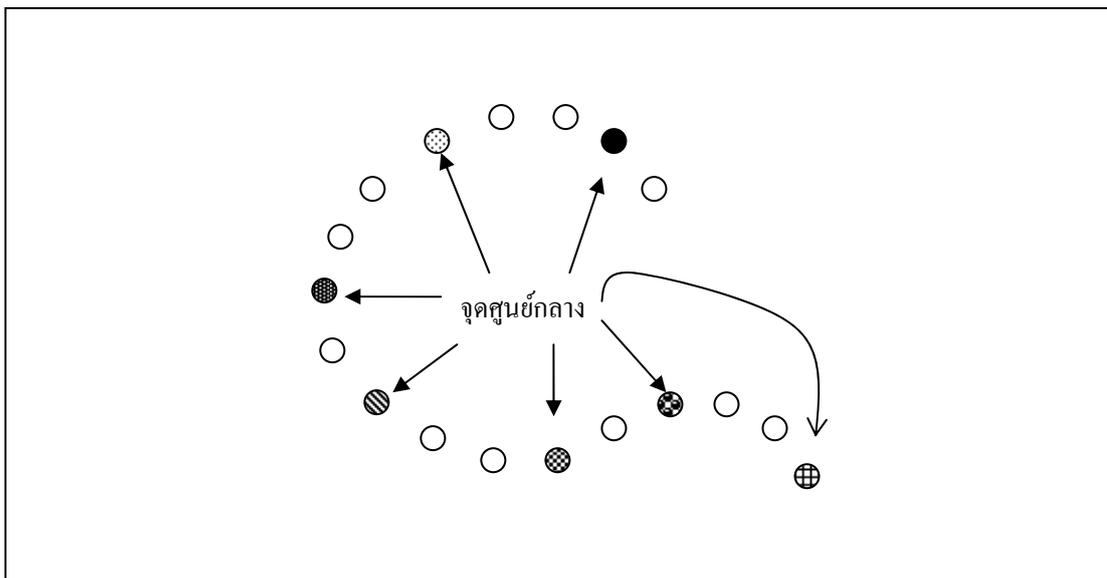
kmean

1. Decide on a value for k .
2. Initialize the k cluster centers (randomly, if necessary).
3. Decide the class memberships of the n objects by assigning them to the nearest cluster center.
4. Re-calculate the k cluster centers, by assuming the memberships found above are correct.
5. If none of the n objects changed membership in the last iteration, goto 6. Otherwise goto 3.
6. Calculate *centroid* from the class memberships as new simplified points.

ภาพที่ 21 อัลกอริทึมการจัดกลุ่มแบบเค-มีน

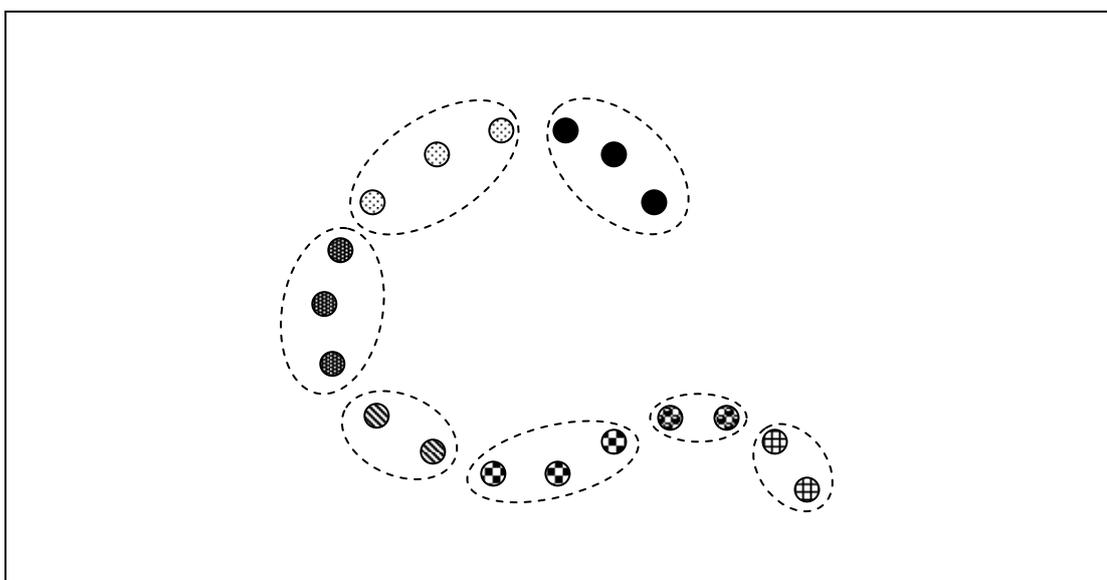
จากภาพที่ 21 อัลกอริทึมนี้ใช้เวลาในการประมวลผลเท่ากับ $O(knrd)$ เมื่อ k คือจำนวนกลุ่มข้อมูลที่ผู้ใช้กำหนด n คือจำนวนข้อมูลที่ต้องการลดความซับซ้อน r คือจำนวนรอบในการทำงาน และ d คือมิติของข้อมูล (Macqueen, 1967)

ตัวอย่างการทำงานกับข้อมูลสองมิติของอัลกอริทึมการจัดกลุ่มแบบเค-มีน แสดงไว้ดังภาพต่อไปนี



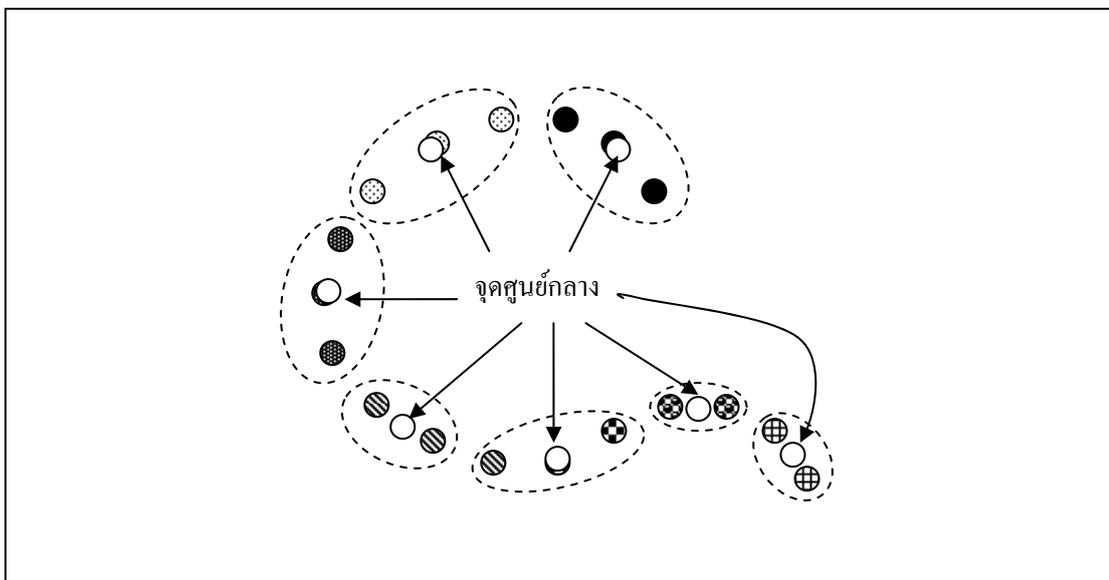
ภาพที่ 22 ลักษณะจุดศูนย์กลางของแต่ละกลุ่มที่สุ่มจากข้อมูล

จากภาพที่ 22 แสดงจุดศูนย์กลางที่สุ่มจากข้อมูลทั้งหมดตามจำนวนกลุ่มที่กำหนด (k) ในที่นี้ k มีค่าเท่ากับ 7 จากนั้นวัดระยะห่างระหว่างข้อมูลแต่ละตัวกับจุดศูนย์กลางแต่ละจุด แล้วทำการจัดข้อมูลแต่ละตัวเข้ากลุ่มที่ข้อมูลตัวนั้นอยู่ใกล้



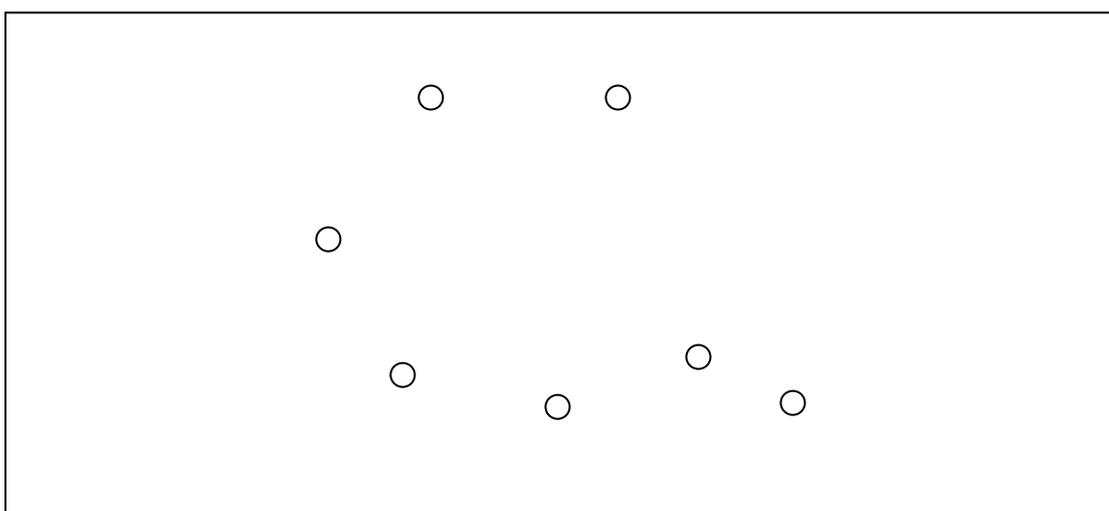
ภาพที่ 23 ลักษณะกลุ่มข้อมูลที่จัดได้จากอัลกอริทึมการจัดกลุ่มแบบเค-มีน

จากภาพที่ 23 แสดงลักษณะข้อมูลที่ถูกจัดเข้ากลุ่มที่ข้อมูลนั้นๆ อยู่ใกล้ เมื่อจำนวนสมาชิกในแต่ละกลุ่มคงที่แล้ว จำนวนจุดศูนย์กลางของแต่ละกลุ่ม ดังภาพที่ 24



ภาพที่ 24 ลักษณะจุดศูนย์กลางของแต่ละกลุ่มที่คำนวณได้

จากภาพที่ 24 แสดงลักษณะจุดศูนย์กลางของแต่ละกลุ่มที่คำนวณได้จากสมการ (1) เพื่อสร้างเป็นกลุ่มจุดผลลัพธ์ (จุดสีขาว) ดังภาพที่ 25



ภาพที่ 25 ลักษณะของกลุ่มจุดผลลัพธ์ที่ได้จากอัลกอริทึมการจัดกลุ่มแบบเค-มีน

งานวิจัยที่เกี่ยวข้อง

1. การลดความซับซ้อนของพื้นผิว (Surface Simplification)

การลดความซับซ้อนของพื้นผิวเป็นเทคนิคที่ถูกนำมาใช้ในโปรแกรมประยุกต์ทางด้านคอมพิวเตอร์กราฟิกส์ เพื่อเพิ่มความเร็วและลดการใช้หน่วยความจำในการแสดงผลแบบจำลองสามมิติบนคอมพิวเตอร์ ซึ่งเทคนิคนี้มีอยู่ด้วยกันหลายประเภท

ปี ค.ศ. 1997 Low และ Tan ได้นำเสนอวิธีการลดความซับซ้อนของพื้นผิวประเภทการจัดกลุ่มข้อมูลที่สัมพันธ์กัน (Vertex Clustering) โดยวิธีการที่นำเสนอนี้ ใช้จำนวนครั้งในการคำนวณน้อยลง ให้อัตราการลดความซับซ้อนที่สูงขึ้น และมีการทำงานแบบวนซ้ำ (Iterative)

ต่อมา Lindstrom และ Turk (1998) ได้นำเสนอวิธีการลดความซับซ้อนของพื้นผิวด้วยการลดจำนวนเส้นเชื่อมระหว่างจุดบนพื้นผิวของแบบจำลองสามมิติ จากงานวิจัยนี้พบว่า วิธีการที่เขานำเสนอสามารถลดความซับซ้อนของพื้นผิวได้เร็วที่สุด และใช้หน่วยความจำน้อยที่สุดเช่นกัน

หลังจากนั้น ในปี ค.ศ. 2002 Mark และคณะ ได้เสนอการเปรียบเทียบประสิทธิภาพของอัลกอริทึมการลดความซับซ้อนของพื้นผิวด้วยวิธีการต่างๆ ได้แก่ Incremental Clustering, Hierarchical Clustering, Iterative Simplification และ Particle Simulation ในงานวิจัยนี้ได้ข้อสรุปว่า อัลกอริทึม Hierarchical Clustering (Top-down) นั้นใช้เวลาในการทำงานน้อยที่สุด ส่วนอัลกอริทึม Iterative Simplification ให้พื้นผิวผลลัพธ์ที่มีคุณภาพสูงที่สุด

2. การสร้างพื้นผิว (Surface Reconstruction)

การสร้างพื้นผิวจากกลุ่มจุดบนพื้นผิวที่ได้จากเครื่องสแกนเนอร์สามมิตินั้นมีอยู่ด้วยกันหลายวิธี ในส่วนนี้จะกล่าวถึงงานวิจัยที่นำเสนอการสร้างพื้นผิวจากกลุ่มจุดบนพื้นผิวที่ใช้หลักการของไวโรนอยไดอะแกรม

ปี ค.ศ. 1999 Amenta และ Bern ได้เสนออัลกอริทึมสร้างพื้นผิวจากกลุ่มจุดตัวอย่างที่ประยุกต์การทำงานมาจากไวโรนอยไดอะแกรม ซึ่งเดิมใช้กับข้อมูลสองมิติมาใช้กับข้อมูลสามมิติ

โดยอัลกอริทึมที่พวกเขาแนะนำเสนอนั้นง่ายต่อ โปรแกรมและนำไปใช้งาน แต่ทำงานได้กับกลุ่มจุด ตัวอย่างที่มีขนาดจำกัด

ต่อมาในปี ค.ศ. 2000 Amenta และคณะ ได้เสนอวิธีการสร้างพื้นผิวจากกลุ่มจุดตัวอย่างที่เรียกว่า Cocone โดยอัลกอริทึมนี้ใช้การคำนวณโวลุ่มนอยไดอะแกรมเพียงรอบเดียวเท่านั้น ซึ่งต่างจากอัลกอริทึมอื่นๆ ที่ต้องใช้การคำนวณโวลุ่มนอยถึงสองรอบ แต่อัลกอริทึมนี้ก็สามารถทำงานได้กับกลุ่มจุดตัวอย่างที่ขนาดไม่ใหญ่นัก

หลังจากนั้น Dey และคณะ (2001) ได้เสนอวิธีการสร้างพื้นผิวจากกลุ่มจุดตัวอย่างที่เรียกว่า Supercocone ซึ่งพัฒนาต่อเนื่องมาจาก Cocone (Amenta, 2000) โดยได้เพิ่มขั้นตอนการแบ่งกลุ่มจุดตัวอย่างออกเป็นส่วนย่อยๆ ในลักษณะของออกทรี (Octree) ก่อนการสร้างพื้นผิวของกลุ่มจุดตัวอย่างแต่ละส่วนด้วยอัลกอริทึม Cocone ดังนั้นอัลกอริทึมนี้จึงสามารถทำงานกับกลุ่มจุดตัวอย่างที่มีขนาดมากกว่าหนึ่งล้านจุดได้

3. งานวิจัยอื่นๆ ที่เกี่ยวข้อง

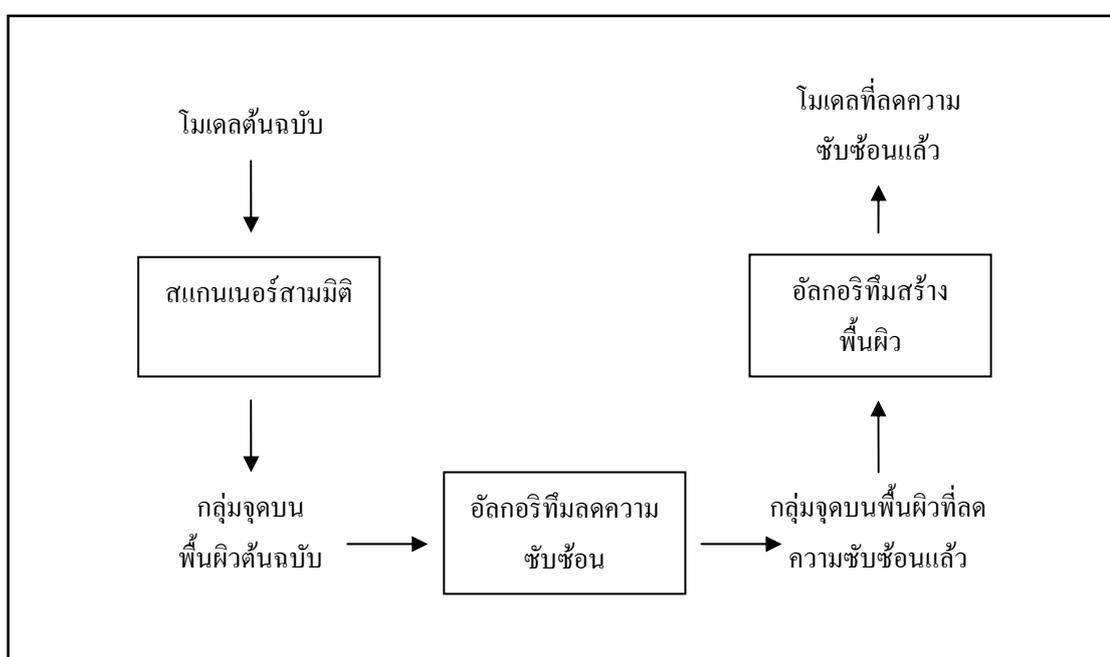
ในปี ค.ศ. 1975 Bentley ได้พัฒนาและนำเสนอโครงสร้างข้อมูลต้นไม้ใหม่ เพื่อใช้ในการค้นหาข้อมูลที่มีความสัมพันธ์กันหรืออยู่ใกล้เคียงกัน จากงานวิจัยนี้ทำให้ได้โครงสร้างข้อมูลต้นไม้ที่เรียกว่า Multidimensional Binary Search Tree หรือ Kd-tree ซึ่งใช้เวลาในการสร้างต้นไม้ไม่บ่อย และค้นหาข้อมูลใกล้เคียงได้อย่างรวดเร็ว

ต่อมา Cignoni และคณะ (1998) ได้นำเสนอวิธีการวัดความผิดพลาดที่เกิดขึ้นจากการลดความซับซ้อนของพื้นผิว โดยการเปรียบเทียบความแตกต่างระหว่างพื้นผิวดั้งเดิมและพื้นผิวที่ผ่านการลดความซับซ้อนแล้ว ผลที่ได้จากงานวิจัยนี้ คือ โปรแกรมเมโทร ซึ่งเป็นโปรแกรมช่วยวัดความผิดพลาดที่เกิดขึ้นพื้นผิวที่ผ่านการลดความซับซ้อน สามารถแสดงผลได้ทั้งสองแบบคือ ตัวเลขรายงานความผิดพลาดและแบบพื้นผิวแสดงความผิดพลาด

จากงานวิจัยที่กล่าวมาข้างต้นจึงเป็นที่มาของงานวิจัยนี้ ซึ่งนำเสนออัลกอริทึมลดความซับซ้อนของพื้นผิวประเภทการจัดกลุ่มที่สัมพันธ์กัน (Vertex Clustering) โดยอัลกอริทึมที่นำเสนอนี้ได้ประยุกต์มาจากการทำงานของอัลกอริทึมต้นไม้ค้นหาแบบทวิภาคหลายมิติ (Bentley, 1975)

และเรียกอัลกอริทึมนี้ว่า ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ ซึ่งทำงานโดยตรงกับกลุ่มจุดบนพื้นผิว โดยไม่ผ่านกระบวนการกรองใดๆ ก่อน อีกทั้งยังเป็นอัลกอริทึมที่มีการทำงานไม่ซับซ้อนและง่ายต่อการโปรแกรม

อัลกอริทึมที่นำเสนอนี้ เป็นอัลกอริทึมที่ทำงานโดยตรงกับกลุ่มจุดบนพื้นผิวที่ได้จากเครื่องสแกนเนอร์สามมิติ ก่อนจะถูกนำไปสร้างเป็นพื้นผิว เพื่อใช้ในการแสดงผลต่อไป ขั้นตอนการลดความซับซ้อนของพื้นผิวประเภทการการจัดกลุ่มที่สัมพันธ์กันแสดงไว้ดังภาพที่ 26



ภาพที่ 26 การลดความซับซ้อนของพื้นผิวประเภทการการจัดกลุ่มที่สัมพันธ์กัน

อุปกรณ์และวิธีการ

อุปกรณ์

- ฮาร์ดแวร์
 - CPU: Intel Pentium IV 2.4 GHz
 - Main Memory: 1 GB
 - Hard Disk: 80 GB
- ซอฟต์แวร์
 - Java Development Toolkit: JDK 1.6
 - IDE: NetBeans 5.5
 - Model Viewer: GeomView 1.81
 - Operating System: OpenSUSE 10.2

วิธีการ

1. ชุดข้อมูลสำหรับการทดลอง

ชุดข้อมูลที่ใช้ในการวิจัยนี้เป็นกลุ่มจุดบนพื้นผิวของแบบจำลองสามมิติ ซึ่งมีรายละเอียดของชุดข้อมูลแสดงไว้ในตารางที่ 2

ตารางที่ 2 รายละเอียดชุดข้อมูลสำหรับการทดลอง

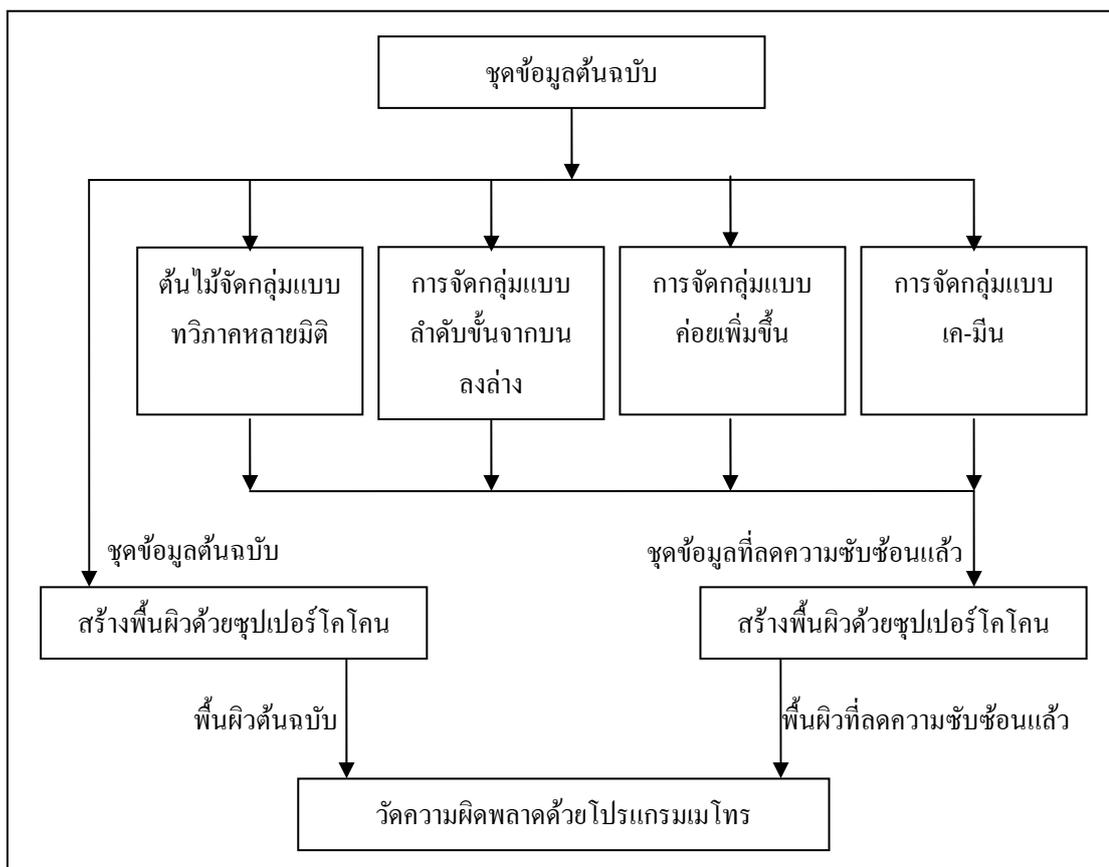
ชื่อชุดข้อมูล	ขนาดของชุดข้อมูล (จุด)
Bunny	35,947
Horse	48,480
Armadillo	172,793
Skeleton Hand	327,323
Dragon	437,647

ตารางที่ 2 (ต่อ)

ชื่อชุดข้อมูล	ขนาดของชุดข้อมูล (จุด)
Happy Buddha	543,652
Turbine Blade	882,945

2. ขั้นตอนการทดลอง

การวิจัยนี้แบ่งการทดลองออกเป็น 3 ส่วน ได้แก่ การวัดเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว การวัดเวลาที่ใช้ในการสร้างพื้นผิว และการวัดคุณภาพของพื้นผิว ซึ่งแต่ละส่วนมีรายละเอียด ดังต่อไปนี้



ภาพที่ 27 ขั้นตอนการทดลองในงานวิจัย

2.1 การวัดเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว

การทดลองส่วนนี้เป็นการวัดเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิวด้วยขั้นตอนวิธีการต่างๆ ซึ่งได้แก่ ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ การจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง การจัดกลุ่มแบบค่อยเพิ่มขึ้น และการจัดกลุ่มแบบเค-มีน การทดลองจะใช้ข้อมูลเข้าที่จัดอยู่ใน 2 ลักษณะ ลักษณะแรกใช้ชุดข้อมูลทั้งหมด 7 ชุด และกำหนดขนาดของกลุ่มข้อมูลเท่ากับ 8 จุด ส่วนลักษณะที่สองใช้ชุดข้อมูลชุดเดียว คือ Dragon และกำหนดขนาดของกลุ่มข้อมูลที่ต้องการเป็น 8, 16, 32, 64, 128, 256, 512 และ 1024 จุดตามลำดับ การทดลองแต่ละชุดมีรายละเอียดดังตารางที่ 3

ตารางที่ 3 รายละเอียดชุดการทดลองสำหรับการวัดเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว

การทดลองที่	ชุดข้อมูลนำเข้า	ขนาดของกลุ่มข้อมูล (จุด)
1	Bunny, Horse, Armadillo, Skeleton Hand, Dragon, Happy Buddha, Turbine Blade	8
2	Dragon	8, 16, 32, 64, 128, 256, 512, 1024

ผลลัพธ์ที่ได้จากการทดลอง ในส่วนนี้ จะถูกนำไปเป็นข้อมูลในการสร้างพื้นผิวด้วยโปรแกรมซูเปอร์โคโคนในการทดลองส่วนต่อไป

2.2 การวัดเวลาที่ใช้ในการสร้างพื้นผิว

การทดลองส่วนนี้เป็นการวัดเวลาที่ใช้ในการสร้างพื้นผิวของโปรแกรมสร้างพื้นผิวซูเปอร์โคโคน โดยข้อมูลนำเข้าที่ใช้ ได้แก่ ชุดข้อมูลต้นฉบับทั้ง 7 ชุด และผลลัพธ์ที่ได้จากการทดลองในส่วนที่ 1 จากนั้นทำการกำหนดค่าพารามิเตอร์ของโปรแกรมโคโคนดังตารางที่ 4

ตารางที่ 4 การกำหนดค่าพารามิเตอร์ของโปรแกรมซูเปอร์โคโคน

supercocone [option] file1 file2
file1: ชื่อไฟล์ชุดข้อมูลนำเข้า
file2: ชื่อไฟล์พื้นผิวผลลัพธ์
[option]: ตัวเลือกของโปรแกรมซูเปอร์โคโคน
-P: สร้างไฟล์พื้นผิวในรูปแบบ Ply

ผลลัพธ์ที่ได้ในส่วนนี้ จะถูกนำไปวัดคุณภาพของพื้นผิวด้วยโปรแกรมเมโทรในการทดลองส่วนต่อไป

2.3 การวัดคุณภาพของพื้นผิว

การทดลองส่วนนี้ เป็นการวัดคุณภาพของพื้นผิวด้วยโปรแกรมเมโทร โดยข้อมูลนำเข้าที่ใช้ในการทดลองส่วนนี้เป็นพื้นผิวผลลัพธ์ที่ได้จากการทดลองส่วนที่ 2 และทำการกำหนดค่าพารามิเตอร์ของโปรแกรมเมโทรดังตารางที่ 5

ตารางที่ 5 การกำหนดค่าพารามิเตอร์ของโปรแกรมเมโทร

metro file1 file2 [option]

file1: ไฟล์พื้นผิวที่ต้องการเปรียบเทียบไฟล์ที่ 1

file2: ไฟล์พื้นผิวที่ต้องการเปรียบเทียบไฟล์ที่ 2

[Option]: ตัวเลือกของโปรแกรมเมโทรใช้หรือไม่ก็ได้

-e: ไม่ต้องทำ edge sampling

-f: ไม่ต้องทำ face sampling

-c: บันทึกไฟล์เปรียบเทียบระหว่างพื้นผิวพื้นผิว

ผลและวิจารณ์

ผล

ผลการทดลองถูกแบ่งเป็น 3 ส่วน ได้แก่ เวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว เวลาที่ใช้ในการสร้างพื้นผิวและคุณภาพของพื้นผิว โดยแต่ละส่วนมีรายละเอียด ดังต่อไปนี้

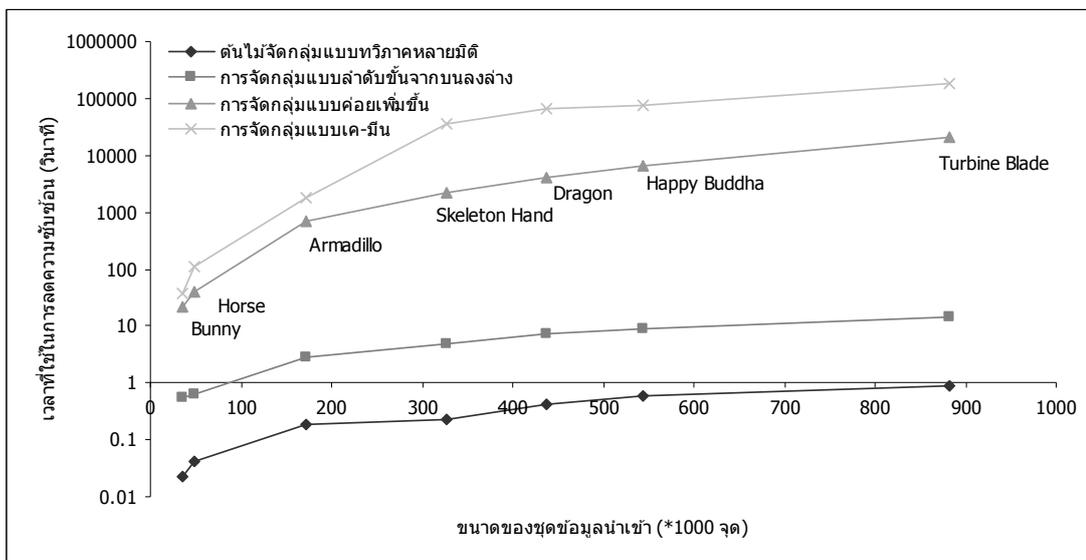
1. เวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว

ผลการทดลองจากการวัดเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว เมื่อข้อมูลนำเข้าเป็นชุดข้อมูล Bunny, Horse, Armadillo, Skeleton Hand, Dragon, Happy Buddha และ Turbine Blade และกำหนดขนาดของกลุ่มข้อมูลที่ต้องการเท่ากับ 8 จุด แสดงไว้ในตารางที่ 6

ตารางที่ 6 เวลาที่ใช้ในการลดความซับซ้อนของพื้นผิวบนชุดข้อมูลต่างๆ เมื่อขนาดกลุ่มข้อมูลคงที่ (8 จุด)

ชื่อชุดข้อมูล	เวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว (วินาที)			
	MBC Tree	DHC	INC	KMC
Bunny	0.023	0.536	21.944	37.946
Horse	0.041	0.643	39.678	113.718
Armadillo	0.189	2.818	704.218	1769.771
Skeleton Hand	0.224	4.917	2268.503	36442.026
Dragon	0.428	7.438	4156.233	64734.198
Happy Buddha	0.604	8.765	6434.919	77486.547
Turbine Blade	0.915	14.460	20475.203	177846.236

จากตารางที่ 6 แสดงเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิวของโมเดลต่างๆ ด้วยอัลกอริทึมที่นำเสนอในวิจัยนี้ จากข้อมูลในตารางสามารถแสดงกราฟการเปรียบเทียบเวลาที่ใช้ในการลดความซับซ้อนได้ดังภาพที่ 28



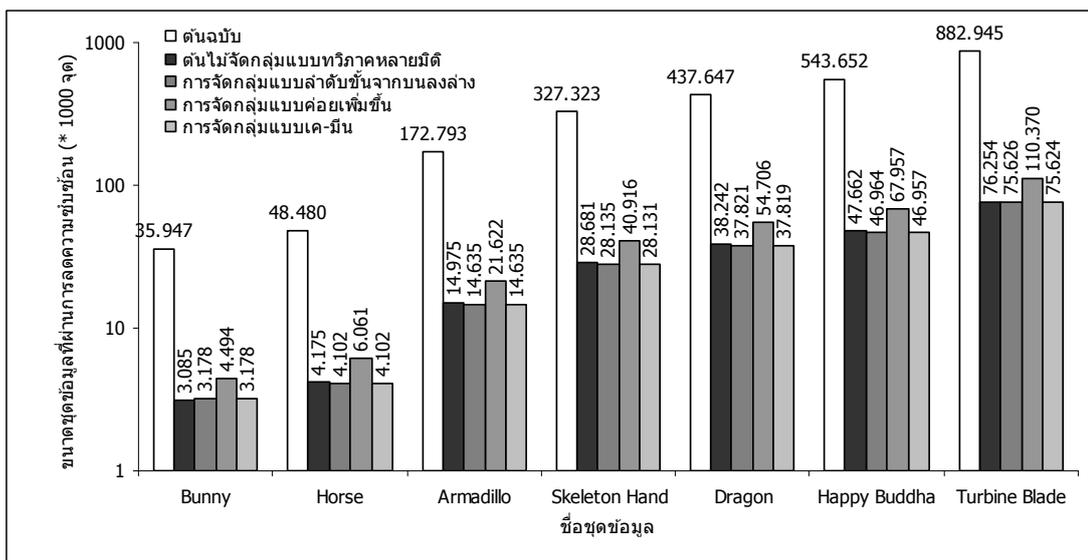
ภาพที่ 28 เปรียบเทียบเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว เมื่อขนาดชุดข้อมูลนำเข้าเปลี่ยนแปลงและขนาดกลุ่มข้อมูลคงที่ (8 จุด)

จากภาพที่ 28 แสดงให้เห็นว่า ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติใช้เวลาในการลดความซับซ้อนของพื้นผิวน้อยที่สุด รองลงมาคือ การจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง การจัดกลุ่มแบบค่อยเพิ่มขึ้น และการจัดกลุ่มแบบเค-มีน ตามลำดับ

ตารางที่ 7 ขนาดของชุดข้อมูลที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลคงที่ (8 จุด)

ขนาดชุดข้อมูล ต้นฉบับ (จุด)	ขนาดชุดข้อมูล (จุด) เมื่อลดความซับซ้อนด้วย			
	MBC Tree	DHC	INC	KMC
35,947	3,085	3,178	4,494	3,178
48,480	4,175	4,102	6,061	4,102
172,793	14,975	14,635	21,622	14,635
327,323	28,681	28,135	40,916	28,131
437,647	38,242	37,821	54,706	37,819
543,652	47,662	46,964	67,957	46,957
882,945	76,254	75,626	110,370	75,624

จากตารางที่ 7 แสดงขนาดของชุดข้อมูลที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมที่นำเสนอในงานวิจัยนี้ จากข้อมูลในตารางนี้แสดงกราฟเปรียบเทียบขนาดของชุดข้อมูลที่ผ่านการลดความซับซ้อนได้ดังภาพที่ 29



ภาพที่ 29 เปรียบเทียบขนาดชุดข้อมูลที่ผ่านการลดความซับซ้อน เมื่อขนาดชุดข้อมูลนำเข้าเปลี่ยนแปลงและขนาดกลุ่มข้อมูลคงที่ (8 จุด)

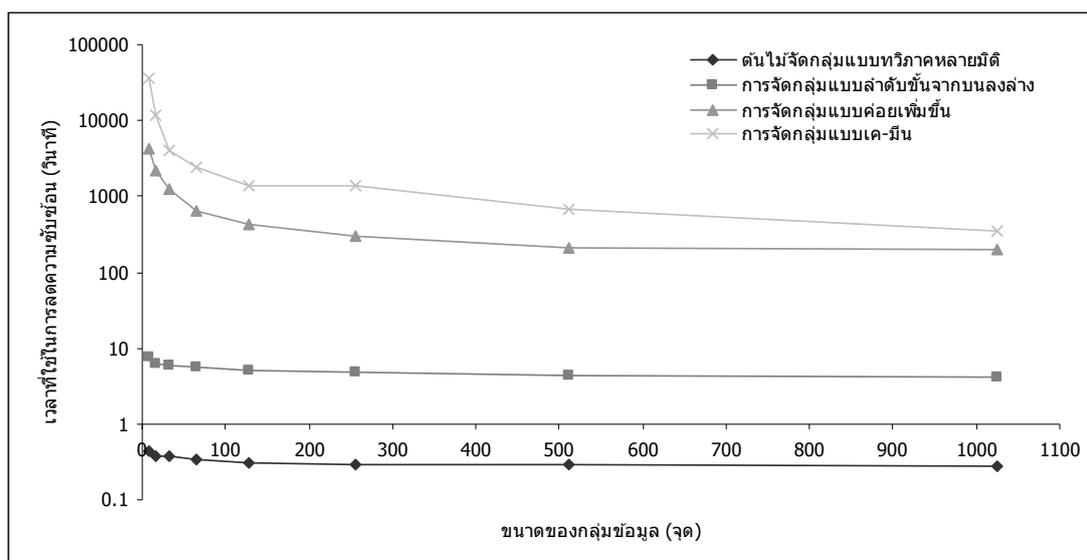
จากภาพที่ 29 แสดงให้เห็นว่าขนาดของชุดข้อมูลที่ผ่านการลดความซับซ้อนแล้วนั้นมีขนาดเล็กลงเมื่อเปรียบเทียบกับขนาดของต้นฉบับ ขนาดของชุดข้อมูลที่ผ่านการลดความซับซ้อนด้วยการจัดกลุ่มแบบค่อยเพิ่มขึ้นมีขนาดใหญ่ที่สุดเมื่อเปรียบเทียบกับขนาดของชุดข้อมูลที่ผ่านการลดความซับซ้อนด้วยการจัดกลุ่มแบบอื่นๆ

ผลการทดลองจากการวัดเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว เมื่อข้อมูลนำเข้าเป็นชุดข้อมูล Dragon ขนาด 437,647 จุด และกำหนดขนาดกลุ่มข้อมูลที่ต้องการเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด ตามลำดับ แสดงไว้ในตารางที่ 8

ตารางที่ 8 เวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว Dragon เมื่อขนาดกลุ่มเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด

ขนาดกลุ่ม ข้อมูล (จุด)	เวลาที่ใช้ในการลดความซับซ้อนของพื้นผิว (วินาที)			
	MBC Tree	DHC	INC	KMC
8	0.428	7.438	4156.233	36442.026
16	0.368	6.329	2206.049	11651.558
32	0.377	5.928	1228.346	4081.912
64	0.335	5.510	650.822	2480.363
128	0.301	5.047	425.655	1386.575
256	0.297	4.723	297.806	1364.544
512	0.291	4.350	209.824	670.772
1024	0.277	4.157	201.908	354.047

จากตารางที่ 8 แสดงเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิวของชุดข้อมูล Dragon ด้วยอัลกอริทึมที่นำเสนอในงานวิจัยนี้ จากข้อมูลในตารางแสดงกราฟเปรียบเทียบเวลาที่ใช้ในการลดความซับซ้อนได้ดังภาพที่ 30



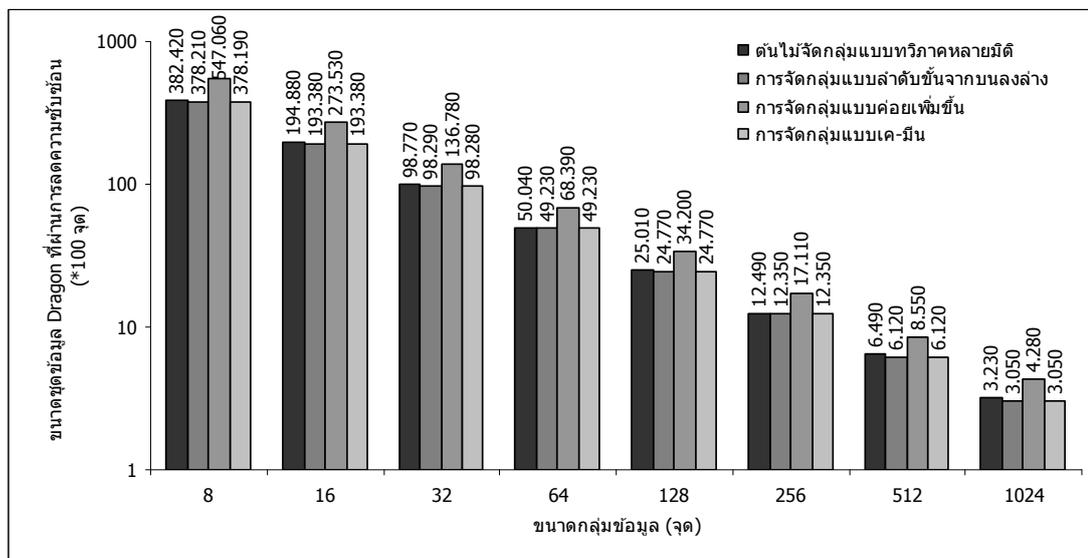
ภาพที่ 30 เปรียบเทียบที่ใช้ในการลดความซับซ้อนของพื้นผิว เมื่อขนาดชุดข้อมูลนำเข้าคงที่ (Dragon) และขนาดกลุ่มข้อมูลเปลี่ยนแปลง

จากภาพที่ 30 แสดงให้เห็นว่า เมื่อขนาดของกลุ่มข้อมูลเพิ่มมากขึ้น อัลกอริทึมลดความซับซ้อนต่างๆ ใช้เวลาในการลดความซับซ้อนน้อยลง โดยอัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติใช้เวลาในการทำงานน้อยที่สุด รองลงมาคือ การจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง การจัดกลุ่มแบบค่อยเพิ่มขึ้น และการจัดกลุ่มแบบเค-มิน ตามลำดับ

ตารางที่ 9 ขนาดของชุดข้อมูล Dragon ที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด

ขนาดกลุ่มข้อมูล (จุด)	ขนาดชุดข้อมูล Dragon เมื่อลดความซับซ้อน (จุด)			
	MBC Tree	DHC	INC	KMC
8	38,242	37,821	54,706	37,819
16	19,488	19,338	27,353	19,338
32	9,877	9,829	13,678	9,828
64	5,004	4,923	6,839	4,923
128	2,501	2,477	3,420	2,477
256	1,249	1,235	1,711	1,235
512	649	612	855	612
1024	323	305	428	305

จากตารางที่ 9 แสดงขนาดของชุดข้อมูลที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมที่นำเสนอในงานวิจัยนี้ จากข้อมูลในตารางสามารถแสดงกราฟเปรียบเทียบขนาดของชุดข้อมูล Dragon ที่ผ่านการลดความซับซ้อนได้ดังภาพที่ 31



ภาพที่ 31 เปรียบเทียบขนาดชุดข้อมูล Dragon ที่ผ่านการลดความซับซ้อน เมื่อขนาดชุดข้อมูลนำเข้าคงที่ (Dragon) และขนาดกลุ่มข้อมูลเปลี่ยนแปลง

จากภาพที่ 31 แสดงให้เห็นว่า เมื่อขนาดกลุ่มข้อมูลใหญ่ขึ้น ขนาดของชุดข้อมูล Dragon ที่ลดความซับซ้อนแล้วจะมีขนาดเล็กลง โดยชุดข้อมูล Dragon ที่ผ่านการลดความซับซ้อนด้วยการจัดกลุ่มแบบค่อยเพิ่มขึ้นมีขนาดใหญ่ที่สุด เมื่อเปรียบเทียบกับ การลดความซับซ้อนด้วยอัลกอริทึมอื่นๆ

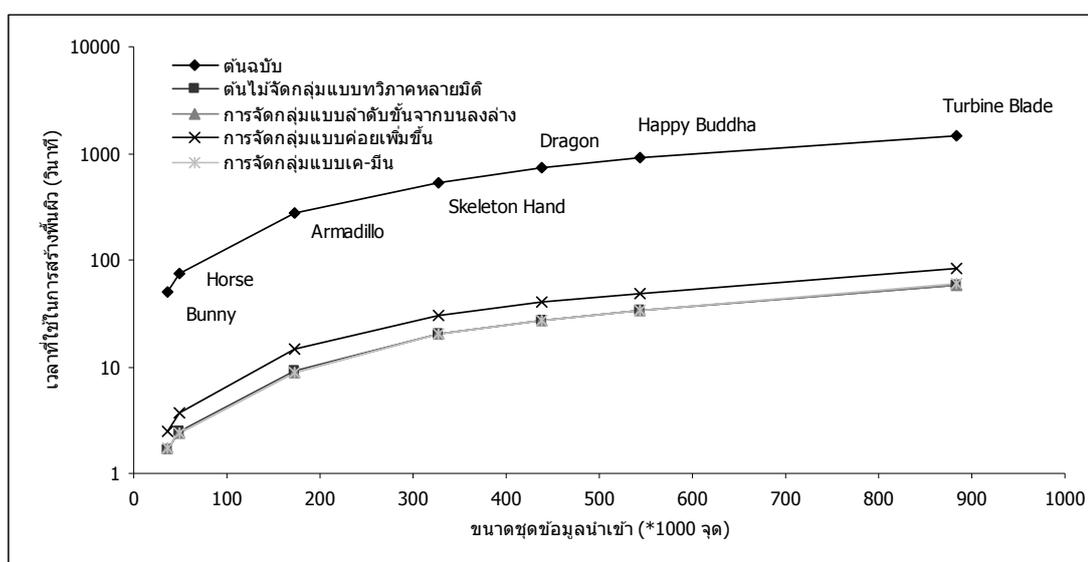
2. เวลาที่ใช้ในการสร้างพื้นผิว

ผลการทดลองจากเวลาที่ใช้ในการสร้างพื้นผิวด้วยโปรแกรมซูบเปอร์โคโคนของชุดข้อมูลต้นฉบับและชุดข้อมูลที่ผ่านการลดความซับซ้อนของพื้นผิวแล้วด้วย ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ การจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง การจัดกลุ่มแบบค่อยเพิ่มขึ้น และการจัดกลุ่มแบบเค-มิน โดยกำหนดขนาดของกลุ่มข้อมูลที่ต้องการเท่ากับ 8 จุด แสดงไว้ในตารางที่ 10

ตารางที่ 10 เวลาที่ใช้ในการสร้างพื้นผิวด้านฉบับและพื้นผิวที่ผ่านการลดความซับซ้อนด้วยขนาด
กลุ่มข้อมูลเท่ากับ 8 จุด

ชื่อชุดข้อมูล	เวลาที่ใช้ในการสร้างพื้นผิว (วินาที)				
	ต้นฉบับ	MBC Tree	DHC	INC	KMC
Bunny	49.766	1.691	1.701	2.453	1.724
Horse	75.690	2.439	2.374	3.643	2.377
Armadillo	272.072	8.981	8.821	14.625	8.782
Skeleton Hand	538.219	19.959	20.146	30.224	20.094
Dragon	747.616	27.091	27.174	39.786	26.736
Happy Buddha	911.892	33.424	33.709	48.884	33.699
Turbine Blade	1489.901	57.554	58.689	84.817	61.213

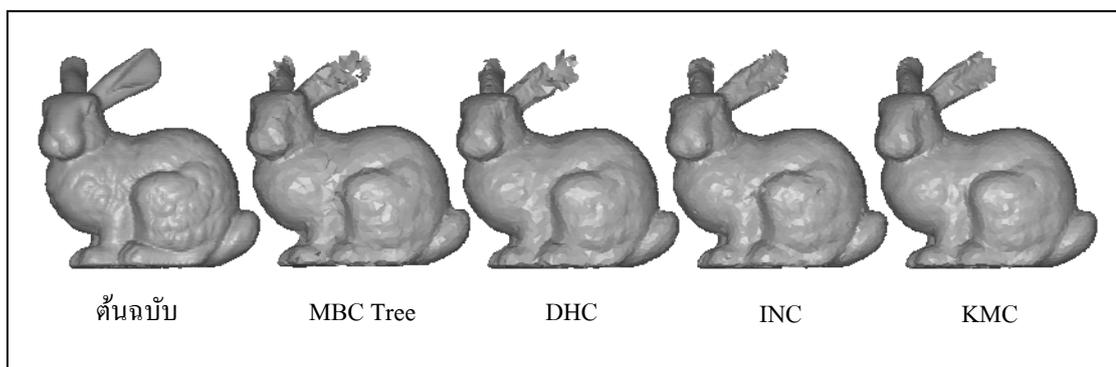
จากตารางที่ 10 แสดงเวลาที่ใช้ในการสร้างพื้นผิวจากชุดข้อมูลต้นฉบับและชุดข้อมูลผ่านการลดความซับซ้อนด้วยอัลกอริทึมที่นำเสนอ ในงานวิจัยนี้ จากข้อมูลในตารางสามารถแสดงกราฟเปรียบเทียบเวลาที่ใช้ในการสร้างพื้นผิวได้ดังภาพที่ 32



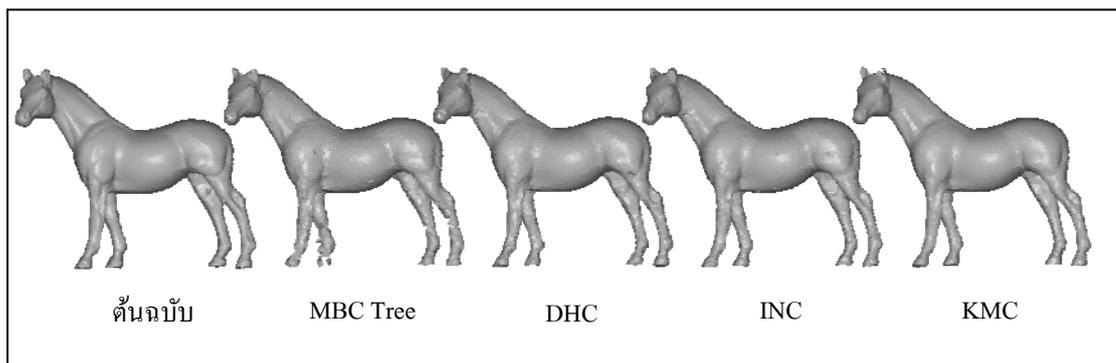
ภาพที่ 32 เปรียบเทียบที่ใช้ในการสร้างพื้นผิวด้านฉบับและพื้นผิวที่ผ่านการลดความซับซ้อนด้วย
กลุ่มข้อมูลเท่ากับ 8 จุด

จากภาพที่ 32 แสดงให้เห็นว่า เวลาที่ใช้ในการสร้างพื้นผิวของชุดข้อมูลต้นฉบับใช้เวลาสูงมาก ในขณะที่ใช้ในการสร้างพื้นผิวของพื้นผิวที่ผ่านการลดความซับซ้อนแล้วใช้เวลาน้อยลง โดยพื้นผิวที่ผ่านการลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ การจัดกลุ่มแบบลำดับขึ้นจากบนลงล่างและการจัดกลุ่มแบบเค-มีนใช้เวลาใกล้เคียงกัน

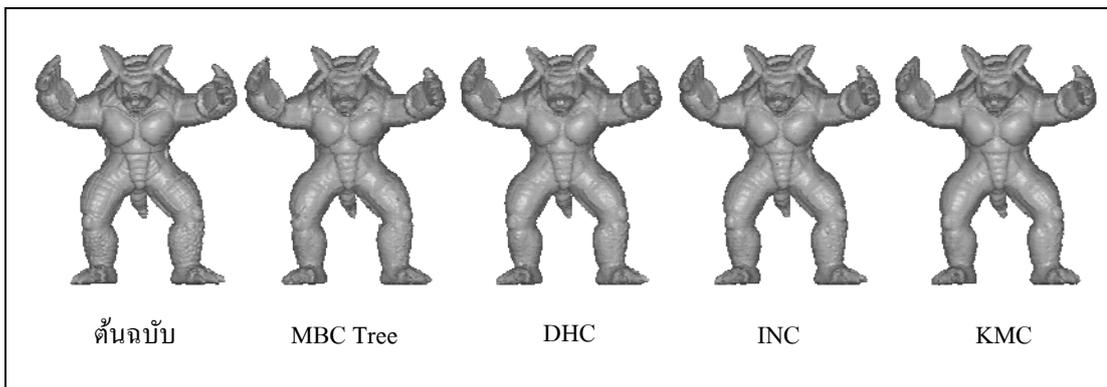
ลักษณะของพื้นผิวผลลัพธ์จากผลการทดลองในตารางที่ 10 ที่สร้างด้วยโปรแกรมซูเปอร์โคโคนแสดงไว้ดังภาพต่อไปนี้



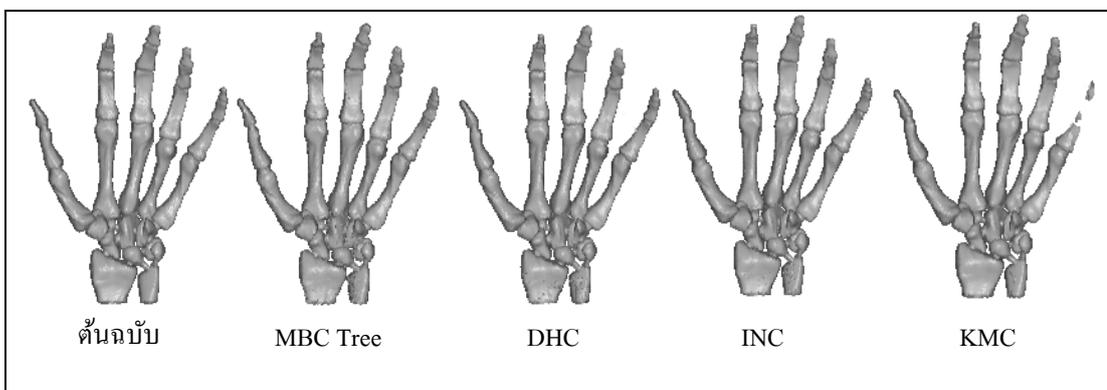
ภาพที่ 33 ลักษณะพื้นผิว Bunny ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด



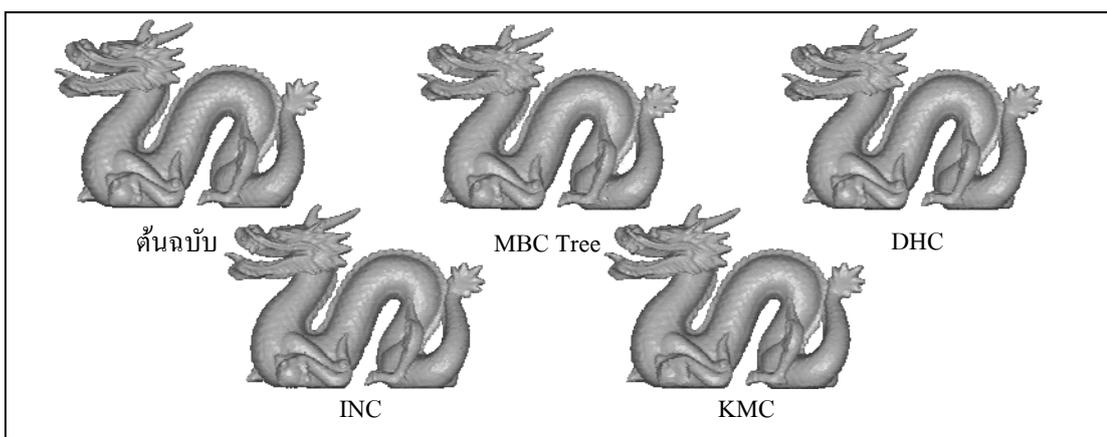
ภาพที่ 34 ลักษณะพื้นผิว Horse ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด



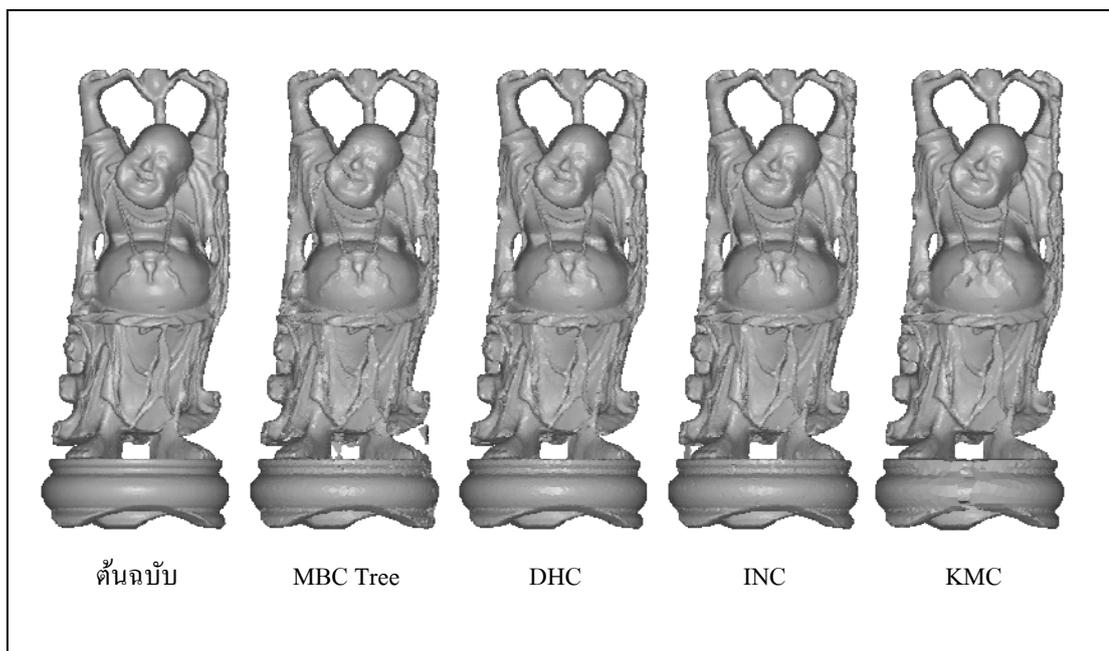
ภาพที่ 35 ลักษณะพื้นผิว Armadillo ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด



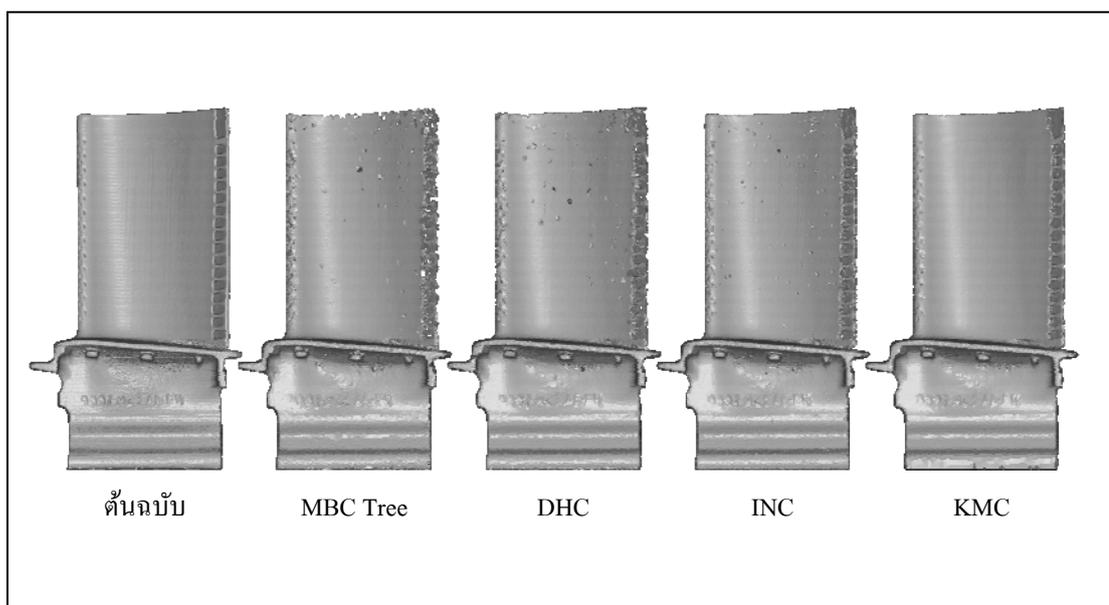
ภาพที่ 36 ลักษณะพื้นผิว Skeleton Hand ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด



ภาพที่ 37 ลักษณะพื้นผิว Dragon ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมต่างๆ เมื่อกำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด



ภาพที่ 38 ลักษณะพื้นผิว Happy Buddha ที่ผ่านการลดความชื้นด้วยอัลกอริทึมต่างๆ เมื่อ กำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด



ภาพที่ 39 ลักษณะพื้นผิว Turbine Blade ที่ผ่านการลดความชื้นด้วยอัลกอริทึมต่างๆ เมื่อ กำหนดขนาดกลุ่มข้อมูลเท่า 8 จุด

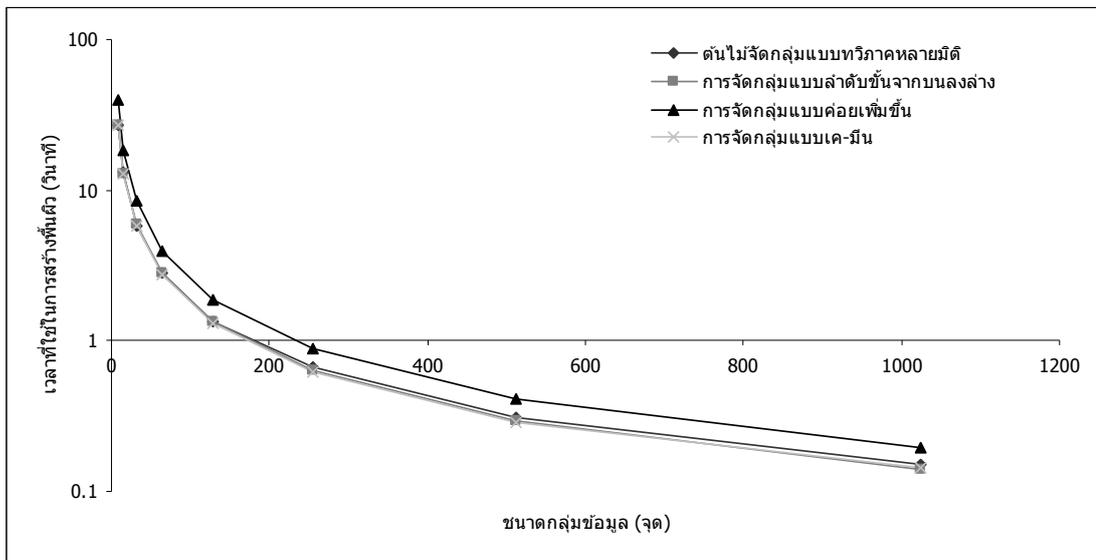
จากภาพที่ 33 - 39 แสดงลักษณะพื้นผิว Bunny, Horse, Armadillo, Skeleton Hand, Dragon, Happy Buddha และ Turbine Blade ตามลำดับ โดยแสดงเปรียบเทียบกันระหว่างพื้นผิวด้านจับและพื้นผิวผ่านการลดความซับซ้อนด้วยการจัดกลุ่มแบบต่างๆ ซึ่งสร้างขึ้นด้วยโปรแกรมซูเปอร์โคโคน

ผลการทดลองจากเวลาที่ใช้ในการสร้างพื้นผิวด้วยโปรแกรมซูเปอร์โคโคนของชุดข้อมูล Dragon ขนาด 437,647 จุด ที่ผ่านการลดความซับซ้อนของพื้นผิวด้วย ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ การจัดกลุ่มแบบลำดับขั้นจากบนลงล่าง การจัดกลุ่มแบบค่อยเพิ่มขึ้น และการจัดกลุ่มแบบเค-มีน โดยกำหนดขนาดกลุ่มข้อมูลที่ต้องการเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด ตามลำดับ แสดงไว้ในตารางที่ 11

ตารางที่ 11 เวลาที่ใช้ในการสร้างพื้นผิวที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด

ขนาดกลุ่มข้อมูล (จุด)	เวลาที่ใช้ในการสร้างพื้นผิว (วินาที)			
	MBC Tree	DHC	INC	KMC
8	27.091	27.283	39.786	26.736
16	13.072	12.820	18.398	12.802
32	5.813	5.951	8.534	5.828
64	2.829	2.787	3.930	2.777
128	1.350	1.334	1.866	1.318
256	0.670	0.638	0.887	0.617
512	0.312	0.294	0.414	0.283
1024	0.152	0.140	0.195	0.145

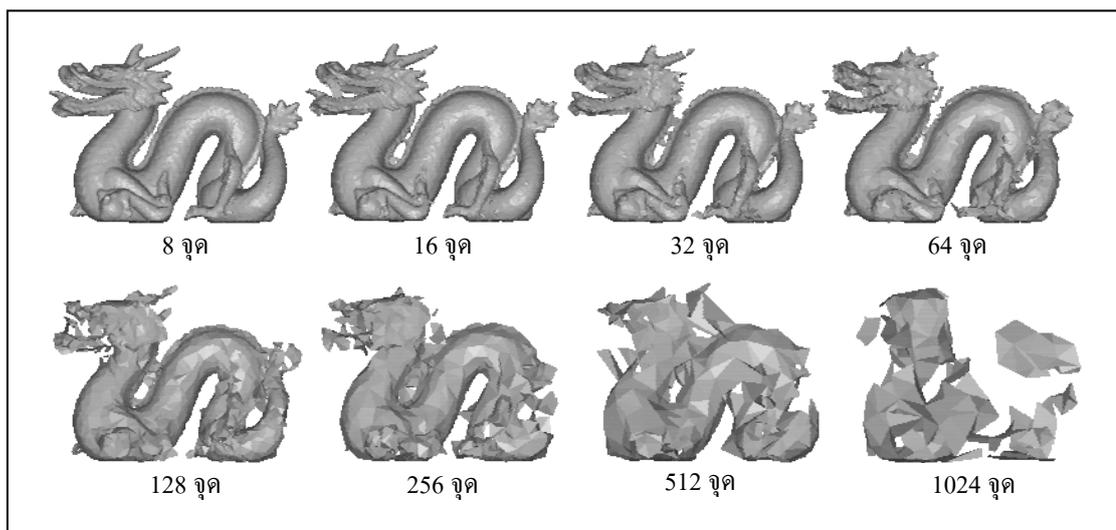
จากตารางที่ 11 แสดงเวลาที่ใช้ในการสร้างพื้นผิวที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมที่นำเสนอในงานวิจัยนี้ จากข้อมูลในตารางสามารถแสดงกราฟเปรียบเทียบเวลาที่ใช้ในการสร้างพื้นผิวได้ดังภาพที่ 40



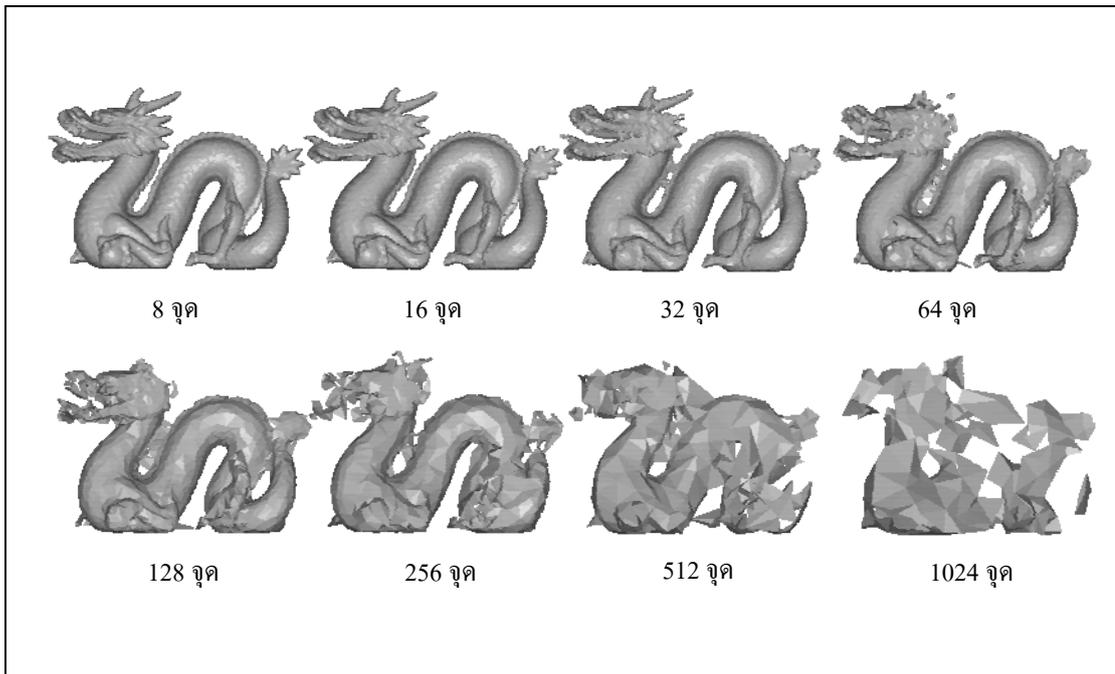
ภาพที่ 40 เปรียบเทียบเวลาที่ใช้ในการสร้างพื้นผิว Dragon ที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด

จากภาพที่ 40 แสดงให้เห็นว่า เวลาที่ใช้ในการสร้างพื้นผิวจากพื้นผิวที่ลดความซับซ้อนด้วยการจัดกลุ่มแบบค่อยเพิ่มขึ้นใช้เวลามากที่สุด ในขณะที่ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ การจัดกลุ่มแบบลำดับชั้นจากบนลงล่างและการจัดกลุ่มแบบเค-มีนใช้เวลาใกล้เคียงกัน

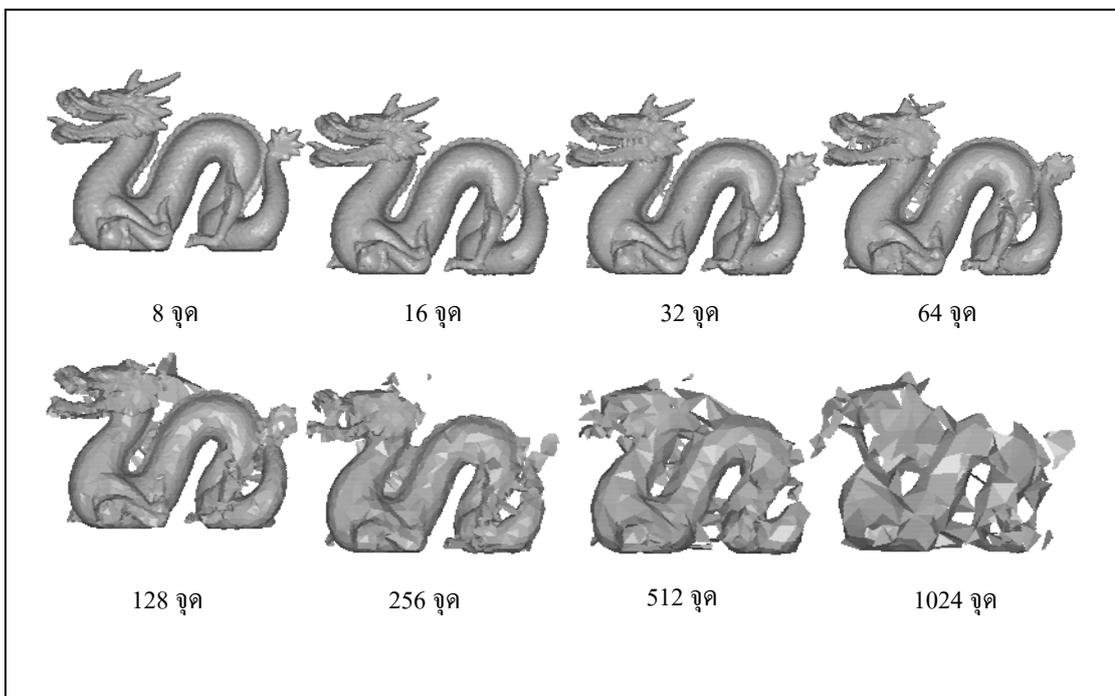
ลักษณะของพื้นผิวผลลัพธ์จากผลการทดลองในตารางที่ 11 ที่สร้างด้วยโปรแกรมซูเปอร์โคโคเนแสดงไว้ดังภาพต่อไปนี้



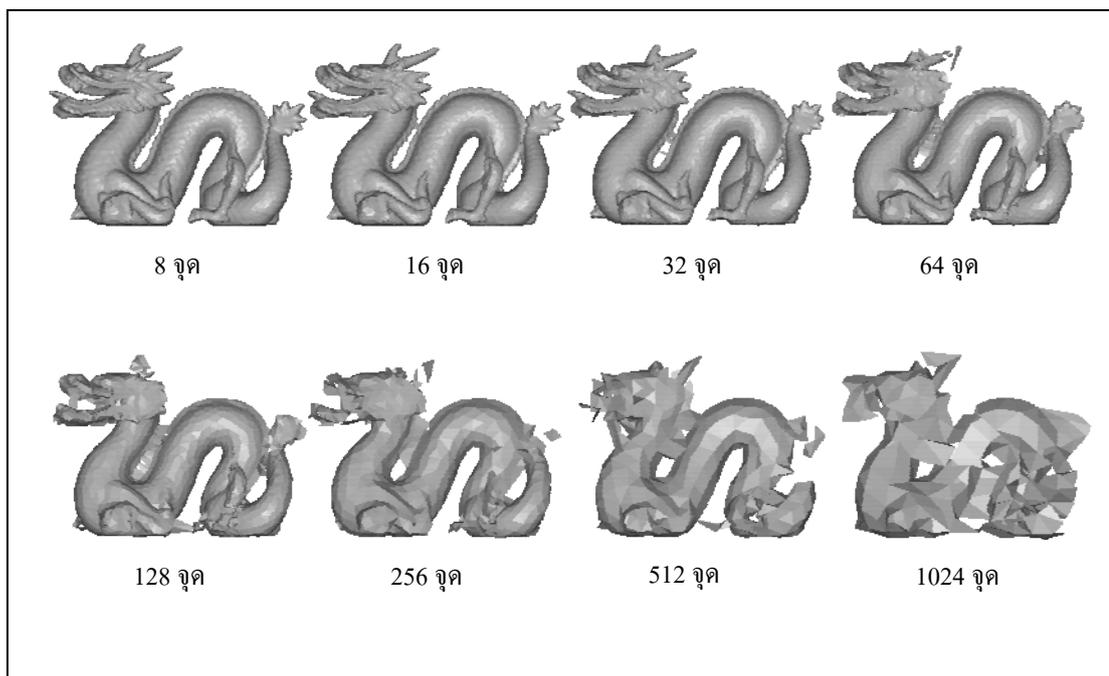
ภาพที่ 41 ลักษณะพื้นผิว Dragon ที่ลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ



ภาพที่ 42 ลักษณะพื้นผิว Dragon ที่ลดความซับซ้อนด้วยการจัดกลุ่มแบบลำดับขั้นจากบนลงล่าง



ภาพที่ 43 ลักษณะพื้นผิว Dragon ที่ลดความซับซ้อนด้วยการจัดกลุ่มแบบค่อยๆเพิ่มขึ้น



ภาพที่ 44 ลักษณะพื้นผิว Dragon ที่ลดความซับซ้อนด้วยการจัดกลุ่มแบบเค-มิน

จากภาพที่ 41 – 44 แสดงลักษณะของพื้นผิว Dragon ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมที่นำเสนอในงานวิจัยนี้ โดยกำหนดขนาดกลุ่มข้อมูลเท่า 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด

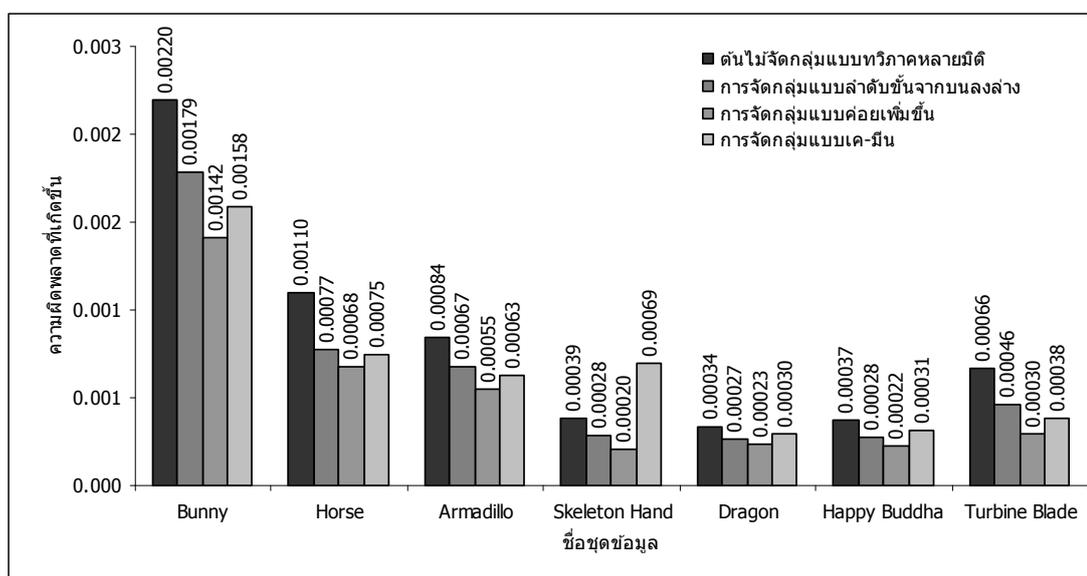
3. คุณภาพของพื้นผิว

ผลการทดลองจากการวัดความผิดพลาดที่เกิดขึ้นด้วยโปรแกรมเมโทรระหว่างพื้นผิวดั้งเดิมและพื้นผิวที่ผ่านการลดความซับซ้อนของพื้นผิวด้วย ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ การจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง การจัดกลุ่มแบบค่อยเพิ่มขึ้น และการจัดกลุ่มแบบเค-มิน โดยกำหนดขนาดกลุ่มที่ต้องการเท่ากับ 8 จุด แสดงไว้ในตารางที่ 12

ตารางที่ 12 ความผิดพลาดที่เกิดขึ้นกับพื้นผิวต่างๆ เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูล เท่ากับ 8 จุด

ชื่อชุดข้อมูล	MBC Tree		DHC		INC		KMC	
	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.
Bunny	0.04382	0.00220	0.04001	0.00179	0.02507	0.00142	0.04478	0.00158
Horse	0.03145	0.00110	0.01617	0.00077	0.02522	0.00068	0.02517	0.00075
Armadillo	0.02647	0.00084	0.02205	0.00067	0.01271	0.00055	0.02343	0.00063
Skeleton Hand	0.01452	0.00039	0.01054	0.00028	0.00726	0.00020	0.05014	0.00069
Dragon	0.01222	0.00034	0.00882	0.00027	0.01935	0.00023	0.01460	0.00030
Happy Buddha	0.01667	0.00037	0.01211	0.00028	0.01047	0.00022	0.01627	0.00031
Turbine Blade	0.02233	0.00066	0.01716	0.00046	0.01681	0.00030	0.02220	0.00038

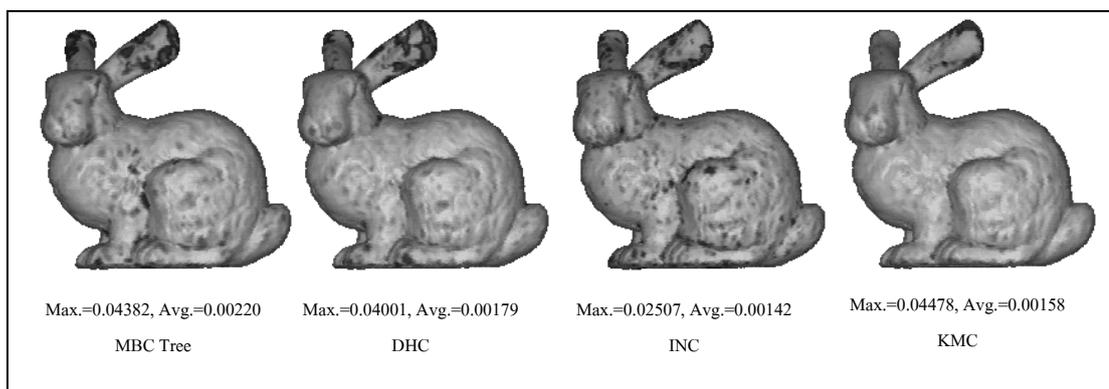
จากตารางที่ 12 แสดงระยะห่างระหว่างพื้นผิวด้านลบและพื้นผิวที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมที่นำเสนอในงานวิจัยนี้ จากข้อมูลในตารางสามารถแสดงกราฟเปรียบเทียบความผิดพลาดที่เกิดขึ้นกับพื้นผิวที่ลดความซับซ้อนได้ดังภาพที่ 45



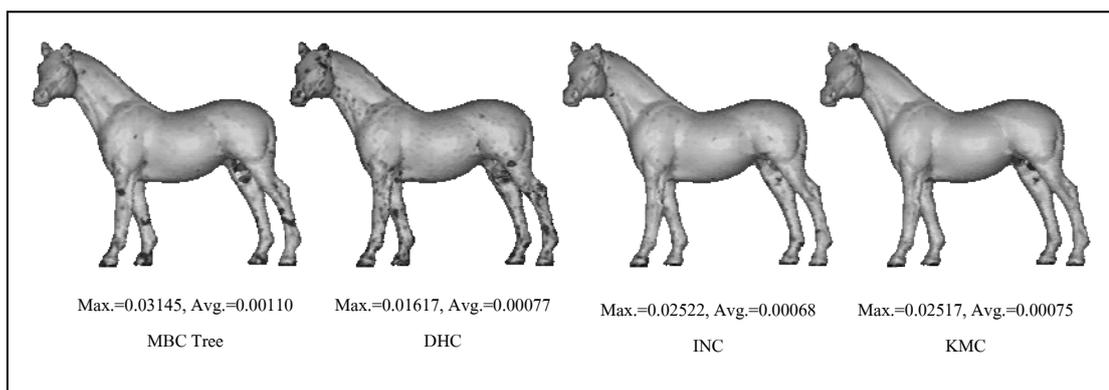
ภาพที่ 45 เปรียบเทียบความผิดพลาดที่เกิดขึ้นกับพื้นผิวที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด

จากภาพที่ 45 แสดงให้เห็นว่า ความผิดพลาดที่เกิดขึ้นกับพื้นผิวที่ผ่านการลดความซับซ้อนด้วยการจัดกลุ่มแบบค่อยเพิ่มขึ้นมีค่าน้อยที่สุด เมื่อเปรียบเทียบกับค่าความผิดพลาดที่เกิดขึ้นจากการลดความซับซ้อนด้วยอัลกอริทึมอื่นๆ

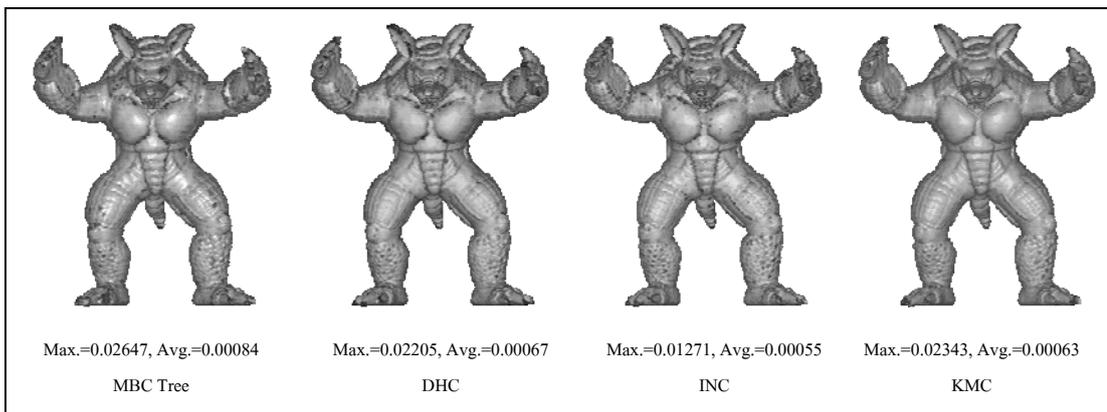
ลักษณะของพื้นผิวแสดงความผิดพลาดจากผลการทดลองในตารางที่ 12 ที่สร้างด้วยโปรแกรมเมโทรแสดงไว้ดังภาพต่อไปนี้



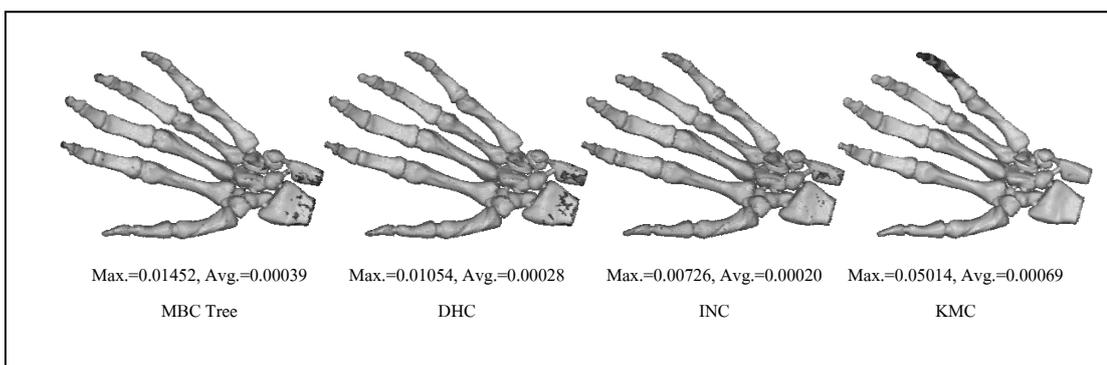
ภาพที่ 46 ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Bunny เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด



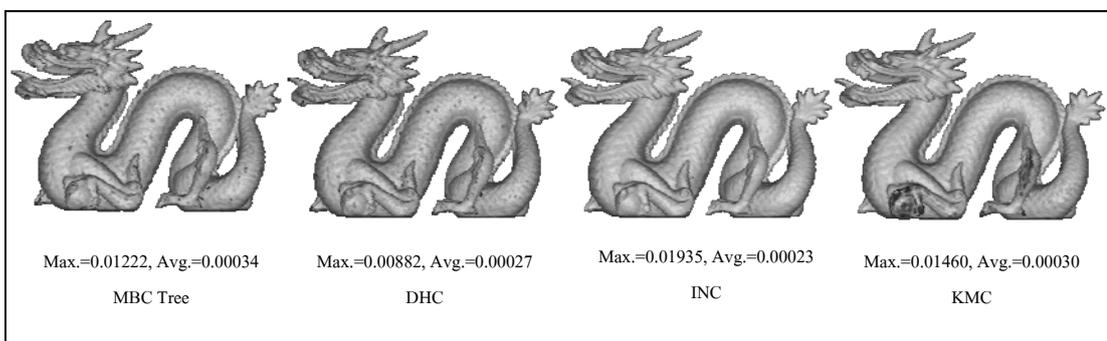
ภาพที่ 47 ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Horse เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด



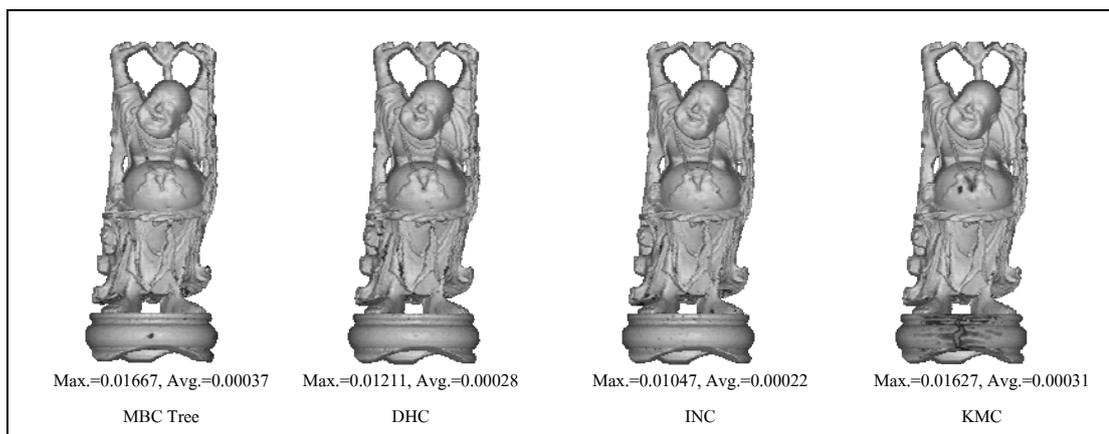
ภาพที่ 48 ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Armadillo เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด



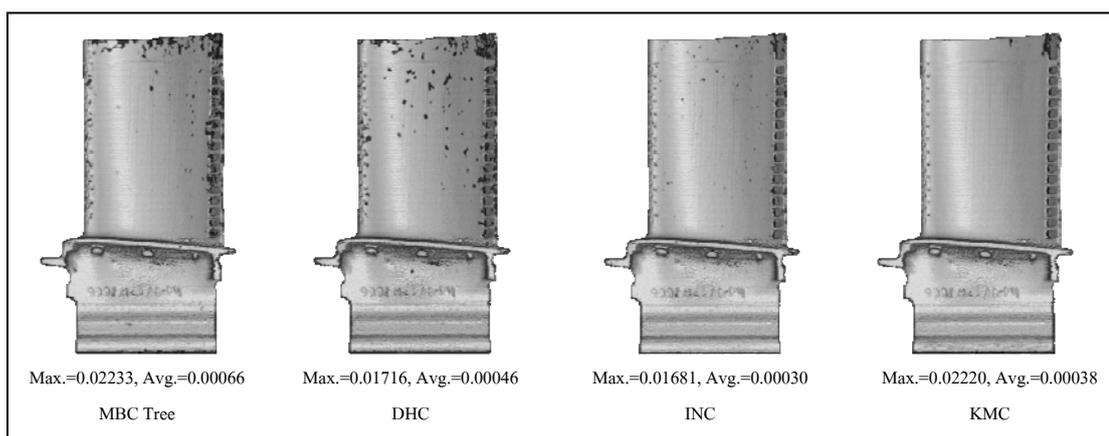
ภาพที่ 49 ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Skeleton Hand เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด



ภาพที่ 50 ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด



ภาพที่ 51 ลักษณะความผิดปกติที่เกิดขึ้นบนพื้นผิว Happy Buddha เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด



ภาพที่ 52 ลักษณะความผิดปกติที่เกิดขึ้นบนพื้นผิว Turbine Blade เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด

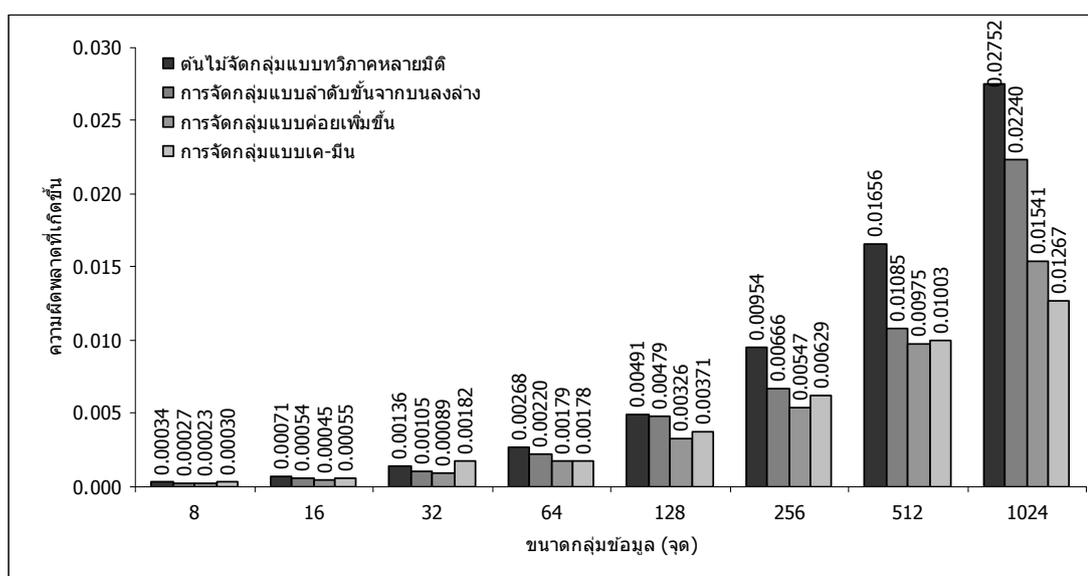
จากภาพที่ 46 - 52 แสดงความผิดปกติที่เกิดขึ้นบนพื้นผิวต่างๆ เมื่อผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8 จุด โดยบริเวณสีดำบนพื้นผิวคือ พื้นผิวบริเวณที่มีความผิดปกติเกิดขึ้นหรือเกิดความแตกต่างระหว่างพื้นผิวด้านนบับและพื้นผิวที่ผ่านการลดความซับซ้อน

ผลการทดลองจากการวัดความผิดปกติที่เกิดขึ้นด้วยโปรแกรมเมโทรระหว่างพื้นผิว Dragon ด้านนบับและพื้นผิว Dragon ที่ผ่านการลดความซับซ้อนของพื้นผิวด้วยต่างๆ ที่นำเสนอในงานวิจัยนี้ โดยกำหนดขนาดกลุ่มข้อมูลเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด ตามลำดับ แสดงไว้ในตารางที่ 13

ตารางที่ 13 ความผิดพลาดที่เกิดขึ้นกับพื้นผิว Dragon เมื่อลดความซับซ้อนด้วยขนาดกลุ่มข้อมูล
เท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด

ขนาดกลุ่ม ข้อมูล (จุด)	MBC Tree		DHC		INC		KMC	
	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.
8	0.01222	0.00034	0.00882	0.00027	0.01935	0.00023	0.01460	0.00030
16	0.02918	0.00071	0.01322	0.00054	0.01149	0.00045	0.03579	0.00055
32	0.03547	0.00136	0.02230	0.00105	0.02353	0.00089	0.03053	0.00182
64	0.07080	0.00268	0.04721	0.00220	0.03975	0.00179	0.03711	0.00178
128	0.04454	0.00491	0.15333	0.00479	0.06474	0.00326	0.13868	0.00371
256	0.13152	0.00954	0.06403	0.00666	0.06446	0.00547	0.08835	0.00629
512	0.14878	0.01656	0.11572	0.01085	0.08749	0.00975	0.09365	0.01003
1024	0.13736	0.02752	0.13495	0.02240	0.10554	0.01541	0.09681	0.01267

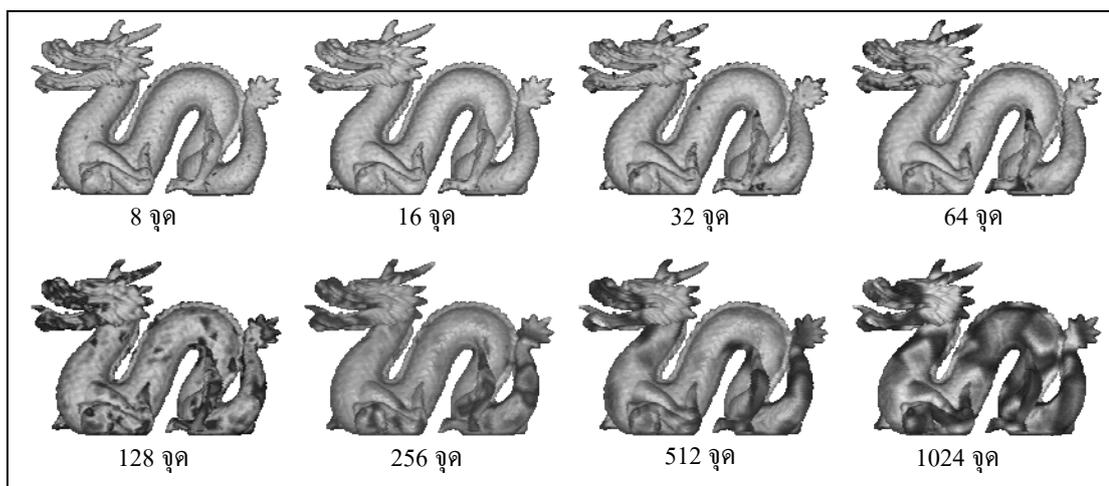
จากตารางที่ 13 แสดงระยะห่างระหว่างพื้นผิว Dragon ต้นฉบับและพื้นผิว Dragon ที่ลดความซับซ้อนด้วยอัลกอริทึมที่นำเสนอในงานวิจัยนี้ จากข้อมูลในตารางสามารถสร้างกราฟแสดงการเปรียบเทียบความผิดพลาดที่เกิดขึ้นได้ดังภาพที่ 53



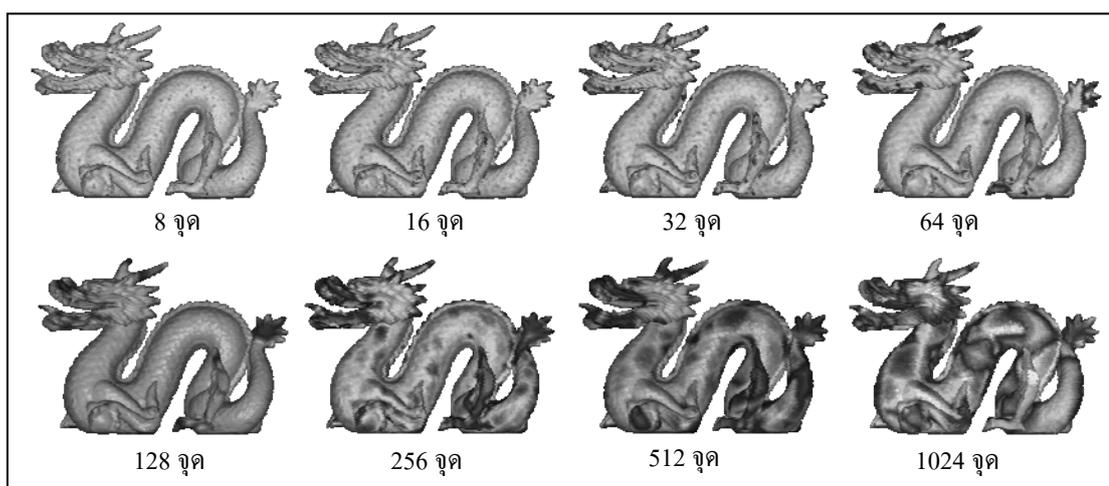
ภาพที่ 53 เปรียบเทียบความผิดพลาดที่เกิดขึ้นกับพื้นผิว Dragon ที่ผ่านการลดความซับซ้อนด้วยขนาดกลุ่มข้อมูลเท่า 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด

จากภาพที่ 53 แสดงให้เห็นว่า เมื่อลดความซับซ้อนของพื้นผิวด้วยขนาดกลุ่มข้อมูลที่ใหญ่ขึ้น ความผิดพลาดที่เกิดขึ้นกับพื้นผิวก็จะเพิ่มมากขึ้น โดยพื้นผิวที่ลดความซับซ้อนด้วยอัลกอริทึมการจัดกลุ่มแบบค่อยเพิ่มขึ้นมีความผิดพลาดเกิดขึ้นน้อยที่สุด เมื่อเปรียบเทียบกับอัลกอริทึมอื่นๆ

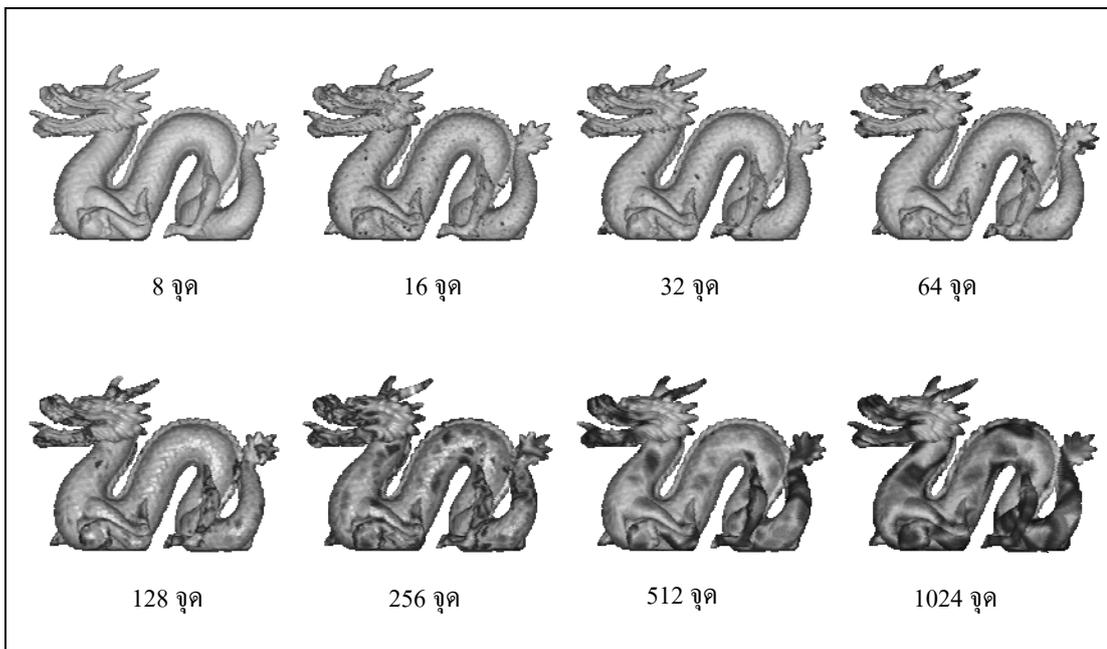
ลักษณะของพื้นผิวแสดงความผิดพลาดจากผลการทดลองในตารางที่ 13 ที่สร้างด้วยโปรแกรมเมโทรแสดงไว้ดังภาพต่อไปนี้



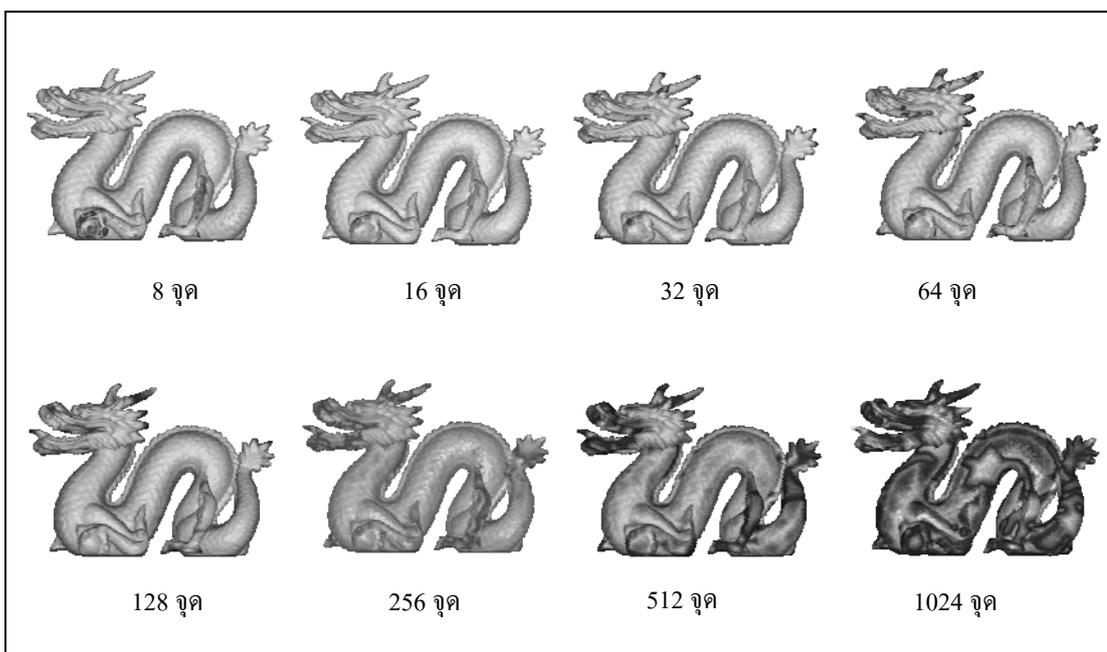
ภาพที่ 54 ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อผ่านการลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ



ภาพที่ 55 ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อผ่านการลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบลำดับขั้นจากบนลงล่าง



ภาพที่ 56 ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อผ่านการลดความซับซ้อนด้วย
ต้นไม้จัดกลุ่มแบบค้อยเพิ่มขึ้น



ภาพที่ 57 ลักษณะความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อผ่านการลดความซับซ้อนด้วย
ต้นไม้จัดกลุ่มแบบเค-มิน

จากภาพที่ 54 - 57 แสดงความผิดพลาดที่เกิดขึ้นบนพื้นผิว Dragon เมื่อผ่านการลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ การจัดกลุ่มแบบลำดับชั้น การจัดกลุ่มแบบค่อยเพิ่มขึ้น และการจัดกลุ่มแบบเค-มีน ตามลำดับ โดยกำหนดขนาดกลุ่มข้อมูลที่ต้องการเท่ากับ 8, 16, 32, 64, 128, 256, 512 และ 1024 จุด บริเวณสีดำนบนพื้นผิวคือ พื้นผิวบริเวณที่มีความผิดพลาดขึ้นหรือเกิดความแตกต่างระหว่างพื้นผิวดั้งฉบับและพื้นผิวที่ผ่านการลดความซับซ้อน

วิจารณ์

งานวิจัยนี้เสนออัลกอริทึมการลดความซับซ้อนของพื้นผิว ที่เรียกว่า ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ โดยทำการเปรียบเทียบประสิทธิภาพการทำงานและคุณภาพของพื้นผิวลัพธ์กับอัลกอริทึมลดความซับซ้อนอื่นๆ ได้แก่ การจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง การจัดกลุ่มแบบค่อยเพิ่มขึ้น และการจัดกลุ่มแบบเค-มีน

จากผลการวัดประสิทธิภาพการทำงานในเรื่องของเวลาที่ใช้ในการลดความซับซ้อนของพื้นผิวพบว่า ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติใช้เวลาในการลดความซับซ้อนของพื้นผิวน้อยที่สุดเมื่อเปรียบเทียบกับอัลกอริทึมอื่นๆ ที่นำเสนอในงานวิจัยนี้ โดยเฉพาะเมื่อเทียบกับอัลกอริทึมการจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง ซึ่งเป็นอัลกอริทึมที่ใช้เวลาในการลดความซับซ้อนที่น้อยที่สุดจากการนำเสนอของ Mark และคณะ (2002) พบว่า อัลกอริทึมต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติใช้เวลาในการลดความซับซ้อนของพื้นผิวน้อยกว่า ที่เป็นเช่นนี้เนื่องจากอัลกอริทึมที่นำเสนอนี้มีการทำงานที่ไม่ซับซ้อน ใช้จำนวนครั้งในการคำนวณน้อยกว่า จึงส่งผลให้อัลกอริทึมที่นำเสนอนี้ใช้เวลาในการประมวลผลที่น้อยกว่า

จากผลการวัดเวลาที่ใช้ในการสร้างพื้นผิวจากชุดข้อมูลต้นฉบับและชุดข้อมูลที่ผ่านการลดความซับซ้อนแล้วพบว่า การสร้างพื้นผิวจากชุดข้อมูลต้นฉบับใช้เวลาค่อนข้างมาก เมื่อเปรียบเทียบกับชุดข้อมูลที่ลดความซับซ้อนแล้ว ที่เป็นเช่นนี้เนื่องจากในขั้นตอนการสร้างพื้นผิวด้วยซูเปอร์โคโคนั้นต้องอาศัยการสร้างเวอรานอยตีโดอะแกรมจากจุดทั้งหมดบนพื้นผิว ซึ่งต้องใช้หน่วยความจำเป็นจำนวนมาก ดังนั้นชุดข้อมูลที่ลดความซับซ้อนแล้วจึงใช้เวลาน้อยในการสร้างพื้นผิว เพราะเป็นชุดข้อมูลที่ประกอบด้วยกลุ่มจุดที่มีขนาดเล็กกว่าชุดข้อมูลต้นฉบับ จากผลการทดลองส่วนนี้พบว่า พื้นผิวที่ลดความซับซ้อนด้วยการจัดกลุ่มแบบค่อยเพิ่มขึ้นใช้เวลามากที่สุดใน

การสร้างพื้นผิว เนื่องจากชุดข้อมูลที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมนี้มีขนาดใหญ่ที่สุดนั่นเอง

จากผลการวัดความผิดพลาดที่เกิดขึ้นกับพื้นผิวที่ผ่านการลดความซับซ้อนแล้วด้วยโปรแกรมเมโทรพบว่า ขั้นตอนการลดความซับซ้อนของพื้นผิวทุกขั้นตอนให้พื้นผิวที่มีความผิดพลาดหรือระยะห่างจากต้นฉบับเพียงเล็กน้อยเท่านั้น โดยพื้นผิวที่ผ่านการลดความซับซ้อนด้วยการจัดกลุ่มแบบค่อยเพิ่มขึ้นนั้นให้พื้นผิวผลลัพธ์ที่มีความผิดพลาดน้อยที่สุด กล่าวคือมีระยะห่างระหว่างพื้นผิวต้นฉบับและพื้นผิวที่ผ่านการลดความซับซ้อนน้อยที่สุด เมื่อเปรียบเทียบกับอัลกอริทึมลดความซับซ้อนอื่นๆ

ต้น ไม่จัดกลุ่มแบบทวิภาคหลายมิติที่นำเสนอในงานวิจัยนี้ ถือเป็นอัลกอริทึมลดความซับซ้อนของพื้นผิวอีกอัลกอริทึมหนึ่ง ซึ่งสามารถนำไปใช้งานร่วมกับโปรแกรมประยุกต์ด้านคอมพิวเตอร์กราฟิกส์ ตัวอย่างเช่น โปรแกรมวิวแบบจำลองสามมิติ (3D Model Viewer) โดยอัลกอริทึมนี้จะทำหน้าที่ลดความซับซ้อนของกลุ่มจุดบนพื้นผิวของแบบจำลองสามมิติก่อนที่โปรแกรมวิวแบบจำลองสามมิติจะเริ่มแสดงแบบจำลองสามมิตินั้นๆ บนจอภาพ เมื่อความซับซ้อนของกลุ่มจุดบนพื้นผิวของแบบจำลองสามมิติน้อยลง จึงส่งผลให้โปรแกรมวิวแบบจำลองสามมิติสามารถแสดงผลได้รวดเร็วขึ้น อีกทั้งยังลดการใช้งานหน่วยความจำของเครื่องคอมพิวเตอร์อีกด้วย

สรุปและข้อเสนอแนะ

สรุป

ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติเป็นอัลกอริทึมลดความซับซ้อนของพื้นที่ผิวที่ทำงานได้รวดเร็วกว่าอัลกอริทึมลดความซับซ้อนของพื้นที่ผิวอื่นๆหลายเท่า พื้นที่ผิวที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมนี้ใช้เวลาในการสร้างพื้นผิวผลลัพธ์น้อยกว่า เมื่อเปรียบเทียบกับเวลาที่ใช้เวลาในการสร้างพื้นผิวผลลัพธ์ของต้นฉบับ แต่อัลกอริทึมนี้ให้พื้นผิวผลลัพธ์ที่มีคุณภาพยังไม่สูงนัก เมื่อเปรียบเทียบกับคุณภาพของพื้นผิวผลลัพธ์ที่ผ่านการลดความซับซ้อนด้วยอัลกอริทึมอื่นๆ ที่ใช้ในงานวิจัยนี้

ข้อเสนอแนะ

จากผลการทดลองแสดงให้เห็นว่า ต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติใช้เวลาในการลดความซับซ้อนของพื้นที่ผิวน้อยกว่าอัลกอริทึมอื่นๆ แต่พื้นผิวผลลัพธ์ที่ได้คุณภาพยังไม่สูงนัก ดังนั้นแนวทางการพัฒนาต่อไปในอนาคตคือ การปรับปรุงคุณภาพของพื้นผิวผลลัพธ์ที่สร้างจากกลุ่มจุดที่ลดความซับซ้อนด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติให้สูงขึ้น

เอกสารและสิ่งอ้างอิง

- ธนศ สุพันธ์ และ ชาคริต วัชโรภาส. 2007. การลดความซับซ้อนของจุดบนพื้นผิวด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ, น. 137-141. ใน งานประชุมวิชาการ มหาวิทยาลัยรังสิต ประจำปีการศึกษา 2549 (RSU Research Conference 2007) ครั้งที่ 1. มหาวิทยาลัยรังสิต, กรุงเทพฯ.
- ธนศ สุพันธ์ และ ชาคริต วัชโรภาส. 2007. การลดความซับซ้อนของจุดบนพื้นผิวด้วยการปรับปรุงต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ, น. 433-438. ใน The 4th International Joint Conference on Computer Science and Software Engineering (JCSSE 2007) ครั้งที่ 4. มหาวิทยาลัยขอนแก่น, ขอนแก่น.
- Bentley, J. L. 1975. Multidimensional binary search trees used for associative searching, pp. 509 - 517. In *Communications of the ACM Volume 18 , Issue 9 (September 1975)*. ACM Press, New York, NY, USA.
- Brodsky, D. and B. Watson. 2000. Model simplification through refinement, pp. 221-228. In *Graphics Interface 2000 Conference* . Canadian Human-Computer Communications Society 2000, Montreal, Canada.
- Cignoni, P., C. Rocchini and R. Scopigno. 1998. Metro: Measuring Error on Simplified Surfaces. *Computer Graphics Forum* 2 (17): 167-174.
- Co, C. S., B. Heckel, H. Hagen, B. Hamann and K. I. Joy. 2003. Hierarchical Clustering for Unstructured Volumetric Scalar Fields, p. 43. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* . IEEE Computer Society, Washington, DC, USA.
- Dey, T. K., J. Giesen and J. Hudson. 2001. Delaunay based shape reconstruction from large data, pp. 19 - 27. In *the IEEE 2001 symposium on parallel and large-data visualization and graphics* . IEEE Press, Piscataway, NJ, USA.

- Garland, M. and P. S. Heckbert. 1997. Surface simplification using quadric error metrics, pp. 209-216. *In International Conference on Computer Graphics and Interactive Techniques 24. ed.* ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.
- Garland, M. and Y. Zhou. 2005. Quadric-based simplification in any dimension, pp. 209-239. *In ACM Transactions on Graphics (TOG) Volume 24 , Issue 2 (April 2005) .* ACM Press, New York, NY, USA.
- Granberg, C. J. and L. Li. 2005. Hierarchical clustering of large volumetric datasets, pp. 425-428. *In Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia 3. ed.* ACM Press, New York, NY, USA.
- Lindstrom, P. 2000. Out-of-core simplification of large polygonal models, pp. 259-262. *In International Conference on Computer Graphics and Interactive Techniques 27. ed.* ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.
- Lindstrom, P. 2001. A memory insensitive technique for large model simplification, pp. 121-126. *In Proceedings of the conference on Visualization '01 .* IEEE Computer Society, Washington, DC, USA.
- Lindstrom, P. and G. Turk. 1998. Fast and memory efficient polygonal simplification, pp. 279-286. *In Proceedings of the conference on Visualization '98 .* IEEE Computer Society Press, Los Alamitos, CA, USA.
- Low, K. and T. Tan. 1997. Model simplification using vertex-clustering, pp. 75-ff. *In Proceedings of the 1997 symposium on Interactive 3D graphics .* ACM Press, New York, NY, USA.

- Macqueen, J. 1967. Some Methods for Classification and Analysis of Multivariate Observations, pp. 281-297. *In Berkeley Symposium on Mathematical Statistics and Probability 5*. Berkeley: University of California Press.
- Moening, C. and N. A. Dodgson. 2003. A New Point Cloud Simplification Algorithm, pp. 396-261. *In 3rd International Conference on Visualization, Imaging, and Image Processing (VIIP 2003) 3. ed.* Visualization, Imaging, and Image Processing, Benalmadena, Spain.
- Pauly, M., M. Gross and L. P. Kobbelt. 2002. Efficient simplification of point-sampled surfaces, pp. 163-170. *In Proceedings of the conference on Visualization '02* . IEEE Computer Society, Washington, DC, USA.
- Shaffer, E. and M. Garland. 2001. Efficient adaptive simplification of massive meshes, pp. 127-134. *In Proceedings of the conference on Visualization '01* . IEEE Computer Society, Washington, DC, USA.
- Suphan, T. and C. Watcharopas. 2007. Multidimensional Binary Clustering Tree for Surface Simplification of Point-Sampled Surface, p. 1202-1204. *In Proceeding of the 2007 Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) International Conference 4. ed.* Mae Fah Luang University, Chiang Rai.

ภาคผนวก

1. ตัวอย่างเมธอดลดความซับซ้อนของพื้นผิวด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ

```

private static void mbctree(int depth, int clusterSize, int head, int tail) {
    // input: ArrayList(Point3D) as input data of point cloud
    // output: ArrayList(Point3D) as output data of simplified point
    if (tail - head + 1 <= clusterSize) {
        if (tail - head + 1 > 0)
            output.add(getCentroid(head, tail));
        return;
    }

    int h = head, t = tail;
    Point3D current = input[h], centroid = getCentroid(h, t);

    switch (depth % 3) {
        case 0:
            while (h <= t) {
                if (current.x >= centroid.x) {
                    Point3D temp = input[t];
                    input[t] = current;
                    --t;
                    current = temp;
                }
            }
            else {
                input[h] = current;
                ++h;
                current = input[h];
            }
    }
    break;
}

```

case 1:

```
while (h <= t) {  
    if (current.y >= centroid.y) {  
        Point3D temp = input[t];  
        input[t] = current;  
        --t;  
        current = temp;  
    }  
    else {  
        input[h] = current;  
        ++h;  
        current = input[h];  
    }  
}  
break;
```

case 2:

```
while (h <= t) {  
    if (current.z >= centroid.z) {  
        Point3D temp = input[t];  
        input[t] = current;  
        --t;  
        current = temp;  
    }  
    else {  
        input[h] = current;  
        ++h;  
        current = input[h];  
    }  
}
```

```

        break;
    }

    if (t - head + 1 <= clusterSize) {
        if (t - head + 1 > 0)
            output.add(getCentroid(head, t));
        return;
    }
    else
        mbctree(++depth, clusterSize, head, t);

    if (tail - h + 1 <= clusterSize) {
        if (tail - h + 1 > 0)
            output.add(getCentroid(h, tail));
        return;
    }
    else
        mbctree(++depth, clusterSize, h, tail);
}

```

2. ตัวอย่างเมฆลดความซับซ้อนของพื้นผิวด้วยการจัดกลุ่มแบบลำดับชั้นจากบนลงล่าง

```

private static void dhc(int clusterSize, int head, int tail) {
    // input: ArrayList(Point3D) as input data of point cloud
    // output: ArrayList(Point3D) as output data of simplified point
    if (tail - head + 1 <= clusterSize) {
        if (tail - head + 1 > 0)
            output.add(getCentroid(head, tail));
        return;
    }
}

```

```

int h = head, t = tail;
Point3D current = input[h], centroid = getCentroid(h, t);
Matrix covarianceMatrix = getCovarianceMatrix(h, t, centroid);
EigenvalueDecomposition eigenVector = new EigenvalueDecomposition(covarianceMatrix);
Matrix NormalVector = getMaxNormalVector(eigenVector.getD(), eigenVector.getV());
double splitPlane = matrixDotProduct(point2matrix(centroid), NormalVector);

while (h <= t) {
    if (matrixDotProduct(point2matrix(current), NormalVector) >= splitPlane) {
        Point3D temp = input[t];
        input[t] = current;
        --t;
        current = temp;
    }
    else {
        input[h] = current;
        ++h;
        current = input[h];
    }
}

if (t - head + 1 <= clusterSize) {
    if (t - head + 1 > 0)
        output.add(getCentroid(head, t));
    return;
}
else
    dhc(clusterSize, head, t);

if (tail - h + 1 <= clusterSize) {

```

```

    if (tail - h + 1 > 0)
        output.add(getCentroid(h, tail));
    return;
}
else
    dhc(clusterSize, h, tail);
}

```

3. ตัวอย่างเมธอดลดความซับซ้อนของพื้นผิวด้วยการจัดกลุ่มแบบค่อยเพิ่มขึ้น

```

private static void inc(int clusterSize) {
    // input: ArrayList(Point3D) as input data of point cloud
    // output: ArrayList(Point3D) as output data of simplified point
    PriorityQueue q = new PriorityQueue();
    Point3D cur = input.remove(0);

    while (input.size() > clusterSize) {
        for (int i = 0; i < input.size(); i++)
            q.add(i, 100.0 - Point3D.sqrDist(cur, input.get(i)));

        ArrayList<Point3D> temp = new ArrayList<Point3D>();
        temp.add(cur);
        int curIdx = 0;
        int[] remIdx = new int[clusterSize];
        for (int i = 0; i < remIdx.length; i++) {
            remIdx[i] = Integer.parseInt(q.remove().toString());
            if (i == remIdx.length - 1)
                curIdx = remIdx[i];
        }
        Arrays.sort(remIdx);
    }
}

```

```

for (int i = remIdx.length - 1; i >= 0; i--) {
    if (curIdx != remIdx[i])
        temp.add(input.remove(remIdx[i]));
    else
        cur = input.remove(remIdx[i]);
}

output.add(getCentroid(temp));
q.clear();
}
input.add(cur);
output.add(getCentroid(input));
}

```

4. ตัวอย่างเมฆลดความซับซ้อนของพื้นผิวด้วยการจัดกลุ่มแบบเค-มีน

```

private static void kmc(int numCluster) {
    // data: double[][] as input data of point cloud
    // output: ArrayList(Point3D) as output data of simplified point
    int totalData = data[0].length;
    double[][] sum = new double[4][numCluster];
    double[][] centroid = new double[3][numCluster];
    // initial centroid
    for (int i = 0; i < 3; i++) {
        int ii = 0;
        for (int j = 0; j < numCluster; j++) {
            centroid[i][j] = data[i][ii];
            ii = ii + step;
        }
    }
}

```

```
// start cluster
int cluster = 0;
boolean isStillMoving = true;

while (isStillMoving) {
    isStillMoving = false;
    // move point to its cluster
    for (int i = 0; i < totalData; i++) {
        double minDist = Double.MAX_VALUE;
        double x = data[0][i];
        double y = data[1][i];
        double z = data[2][i];

        for (int j = 0; j < numCluster; j++) {
            double newDist = dist(x, y, z, centroid[0][j], centroid[1][j], centroid[2][j]);
            if (newDist < minDist) {
                minDist = newDist;
                cluster = j;
            }
        }

        if (data[3][i] != cluster) {
            data[3][i] = cluster;
            isStillMoving = true;
        }
    }
    // cluster change
    if (isStillMoving) {
        // calculate new centroid of each cluster
        sum = new double[4][numCluster];
```

```
for (int i = 0; i < totalData; i++) {
    sum[0][(int)data[3][i]] += data[0][i];
    sum[1][(int)data[3][i]] += data[1][i];
    sum[2][(int)data[3][i]] += data[2][i];
    sum[3][(int)data[3][i]] += 1;
}

for (int i = 0; i < numCluster; i++) {
    centroid[0][i] = sum[0][i] / sum[3][i];
    centroid[1][i] = sum[1][i] / sum[3][i];
    centroid[2][i] = sum[2][i] / sum[3][i];
}
}
}

// output points
for (int i = 0; i < numCluster; i++) {
    Point3D p = new Point3D(centroid[0][i], centroid[1][i], centroid[2][i]);
    if (! p.toString().equals("NaN NaN NaN"))
        output.add(p);
}
}
```

ประวัติการศึกษาและการทำงาน

ชื่อ –นามสกุล	นายธนศ สุพันธ์
วัน เดือน ปี ที่เกิด	วันที่ 2 มีนาคม 2525
สถานที่เกิด	อำเภอเชียงแสน จังหวัดเชียงราย
ประวัติการศึกษา	วท.บ. (วิทยาการคอมพิวเตอร์) มหาวิทยาลัยเวสเทิร์น
ตำแหน่งหน้าที่การงานปัจจุบัน	อาจารย์พิเศษ
สถานที่ทำงานปัจจุบัน	มหาวิทยาลัยเวสเทิร์น
ผลงานตีพิมพ์	<ol style="list-style-type: none">1. การลดความซับซ้อนของจุดบนพื้นผิวด้วยต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ, น. 137-141. ใน งานประชุมวิชาการ มหาวิทยาลัยรังสิต ประจำปี การศึกษา 2549 (RSU Research Conference 2007) ครั้งที่ 1. มหาวิทยาลัยรังสิต, กรุงเทพฯ.2. การลดความซับซ้อนของจุดบนพื้นผิวด้วยการปรับปรุงต้นไม้จัดกลุ่มแบบทวิภาคหลายมิติ, น. 433-438. ใน The 4th International Joint Conference on Computer Science and Software Engineering (JCSSE2007) ครั้งที่ 4. มหาวิทยาลัยขอนแก่น, ขอนแก่น.3. Multidimensional Binary Clustering Tree for Surface Simplification of Point-Sampled Surface, p. 1202-1204. In Proceeding of the 2007 Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) International Conference 4. ed. Mae Fah Luang University, Chiang Rai.
ทุนการศึกษาที่ได้รับ	ทุนสนับสนุนงานวิจัยหรือวิทยานิพนธ์ โครงการปริญญาโท สาขาวิทยาการคอมพิวเตอร์ ภาคพิเศษ (พ.ศ. 2549)