

การตรวจจับการชนกันโดยวิธีทางอนุภาคด้วยหน่วยประมวลผลกราฟิก



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2551

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

PARTICLE-BASED COLLISION DETECTION USING GRAPHIC PROCESSING UNITS



Mr. Thiti Rungcharoenpaisal

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

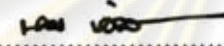
Academic Year 2008

Copyright of Chulalongkorn University


510925

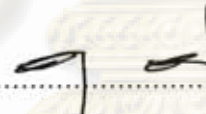
หัวข้อวิทยานิพนธ์ การตรวจจับการชนกันโดยวิธีทางอนุภาคด้วยหน่วยประมวลผลกราฟิก
โดย นายฐิติ รุ่งเจริญไพศาล
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. พิษณุ คนองชัยยศ


คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นำวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาโทบัณฑิต


..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศนिरูวงศ์)

คณะกรรมการสอบวิทยานิพนธ์


..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.สมชาย ประสิทธิ์จูตระกูล)


..... อาจารย์ที่ปรึกษาวิทยานิพนธ์
(ผู้ช่วยศาสตราจารย์ ดร.พิษณุ คนองชัยยศ)


..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ภาวดี สมภักดี)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

นายฐิติ รุ่งเจริญไพศาล : การตรวจจับการชนกันโดยวิธีทางอนุภาคด้วยหน่วยประมวลผลกราฟิก. (PARTICLE-BASED COLLISION DETECTION USING GRAPHIC PROCESSING UNITS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ.ดร.พิษณุ คนองชัยยศ, 56หน้า.

งานวิจัยนี้นำเสนอ วิธีการตรวจจับการชนกันของแบบจำลองรูปหลายเหลี่ยมความละเอียดสูง ที่สามารถเปลี่ยนแปลงรูปร่างได้ เพื่อนำไปใช้ในการจำลองที่ต้องการความละเอียดสูงเช่นการจำลองการผ่าตัด งานคอมพิวเตอร์แอนิเมชัน เป็นต้น โดยวิธีนี้จะมีประสิทธิภาพที่ดีเมื่อเทียบกับวิธีที่อาศัยการสร้างปริมาตรมาครอบวัตถุจริง เพื่อใช้กรองรูปหลายเหลี่ยมที่ไม่มีการชนออกไป ซึ่งเมื่อวัตถุจริงเปลี่ยนแปลงรูปร่าง จะต้องมีการปรับปรุงรูปร่างของปริมาตรตามไปด้วย ทำให้ต้องใช้เวลามากหากวัตถุมีความละเอียดสูง และวิธีนี้ยังมีประสิทธิภาพเพิ่มขึ้นเมื่อเทียบกับวิธีตรวจจับแบบอนุภาคทั่วไปตรงที่ สามารถให้คำตอบละเอียดว่าชนกันที่จุดยอดใดแตกต่างจากวิธีเดิมที่ให้คำตอบเพียงว่ามีการชนหรือไม่

โดยวิธีที่นำเสนอนี้จะนำเข้าสู่ข้อมูลของวัตถุจริง เพื่อแบ่งวัตถุออกเป็นส่วน ๆ และแต่ละส่วนจะกำหนดตัวควบคุมอนุภาคเพื่อปล่อยอนุภาคตรวจจับในส่วนที่น่าจะชนกัน โดยอนุภาคจะเข้าหากันเพื่อตรวจจับว่าส่วนใดที่เข้าใกล้กัน จากนั้นจึงนำรูปหลายเหลี่ยมทั้งหมดในส่วนที่ตรวจจับมาโดยอนุภาค มาตรวจสอบชั้นละเอียดพร้อม ๆ กันแบบขนานโดยอาศัยการทำงาน of หน่วยประมวลผลกราฟิก สุดท้ายจากผลการทดลองทำให้เห็นว่าวิธีนี้สามารถให้ผลลัพธ์ที่ดีและรวดเร็วขึ้นเมื่อเทียบกับวิธีที่ใช้ปริมาตรครอบคลุมในฉากที่มีรูปหลายเหลี่ยมจำนวนมากในแต่ละวัตถุจริง

ศูนย์วิทยทรัพยากร

ภาควิชาวิศวกรรมคอมพิวเตอร์

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์

ปีการศึกษา 2551

ลายมือชื่อนิสิต

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์

ฐิติ รุ่งเจริญไพศาล
 ผศ.ดร.พิษณุ คนองชัยยศ

5070261321 : MAJOR COMPUTER SCIENCE

KEYWORDS : COLLISION DETECTION / GPU/ PARTICLE

THITI RUNGCHAROENPAISAL: PARTICLE-BASED COLLISION DETECTION
USING GRAPHIC PROCESSING UNITS. ADVISOR: ASST. PROF. PIZZANU
KANONGCHAIYOS, 56 pp.

• This research presents a method for a collision detection of high resolution polygonal models, which can be deformed, for using in a complex work such as a *surgical simulation and a computer animation*. This method has a better efficiency comparing to bounding volume hierarchical methods which have to take time in managing their bounding volume when an object deforms especially in a high resolution object. Moreover, this method improves the previous particle-based method which can give only the true or false answers by giving the exact answers such as *all distances and points which are colliding*.

The work flow of this method is, quantizing each object to partitions and putting an emitter in each partition which is used for detecting a potential in collision on its own partition. If it is potential, the emitter will emit and control the number of particles which are used for selecting only the partitions that can be collided. Finally, *all polygons, in the selected partitions, are tested for finding the collided polygons simultaneously in exact phase by using the power of stream processing of GPU*. From an experiment, this method shows that it can give exact results, fast processing and using fewer resources when comparing to the previous particle-based method and bounding volume hierarchical methods.

Department : Computer Engineering
Field of Study: Computer Science.....
Academic Year : 2008.....

Student's Signature
Advisor's Signature

จุฬาลงกรณ์มหาวิทยาลัย

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จเรียบร้อยได้ด้วยดีเพราะได้รับคำแนะนำ และ ให้คำปรึกษาจาก ผู้ช่วยศาสตราจารย์ ดร.พิชณู คนองชัยยศ ซึ่งเป็นผู้ชี้แนะแนวทางในการสร้างสรรค์องค์ความรู้ใหม่ ให้เกิดเป็นวิทยานิพนธ์ฉบับนี้ขึ้น นอกจากนี้ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ ซึ่งได้แก่ รองศาสตราจารย์ ดร.สมชาย ประสิทธิ์จตุระกุล ผู้ซึ่งเป็นประธาน รองศาสตราจารย์ภาวดี สมภักดี ผู้ซึ่งเป็นกรรมการ ที่ได้สละเวลาอันมีค่ามาใช้ในการชี้แจงถึงข้อบกพร่อง รวมถึงแนวทางการแก้ไข และข้อแนะนำดี ๆ ให้กับข้าพเจ้า

ขอขอบพระคุณ คุณพ่อ คุณแม่ รวมไปถึงพี่ ๆ ที่คอยให้คำแนะนำและกำลังใจดี ๆ ในการเผชิญหน้ากับอุปสรรคที่เกิดขึ้น เพื่อให้ข้าพเจ้าได้มีพลังในการฟันฝ่าอุปสรรคต่าง ๆ เพื่อที่จะได้ ก้าวเดินมายืนอยู่ตรงจุดยอดนี้ได้อย่างภาคภูมิใจ และยิ่งไปกว่านั้นต้องขอขอบคุณต่อการสนับสนุนทางด้านเงินทุนในการศึกษาจนจบในระดับบัณฑิตศึกษา

สุดท้ายนี้ ขอขอบพระคุณเพื่อน ๆ พี่ ๆ และน้อง ๆ ทุกคนที่คอยให้กำลังใจและ คำปรึกษาที่ดี ในการใช้ชีวิตอย่างมีความสุขในรั้วจุฬาลงกรณ์มหาวิทยาลัยแห่งนี้



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ	ช
สารบัญภาพ.....	ฅ
สารบัญตาราง.....	ฎ
บทที่ 1 บทนำ.....	1
ความเป็นมาและความสำคัญของปัญหา.....	1
วัตถุประสงค์ของการวิจัย.....	4
ขอบเขตของการวิจัย	4
ประโยชน์ที่คาดว่าจะได้รับ.....	4
ขั้นตอนการดำเนินการวิจัย	4
โครงสร้างของวิทยานิพนธ์.....	5
ผลงานที่ตีพิมพ์จากวิทยานิพนธ์.....	5
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง	6
2.1 แนวคิดและทฤษฎี.....	6
2.1.1 การตรวจจับการชนกัน (Collision Detection).....	6
2.1.2 การทำงานของหน่วยประมวลผลกราฟิก (Graphic Processing Units).....	10
2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	13
บทที่ 3 ระเบียบขั้นตอนวิธีที่เสนอ	17
ระเบียบขั้นตอนวิธี	17
3.1 การเตรียมการประมวลผลก่อนตรวจสอบการชนกัน.....	21
3.2 การตรวจจับการชนกันขั้นตัดคู่วัตถุและคู่พื้นที่.....	24
3.3 การตรวจจับการชนกันขั้นละเอียด	28
การวิเคราะห์ค่าความซับซ้อนของวิธีที่เสนอ	35
บทที่ 4 การทดลองและผลการทดลอง.....	37
4.1 เครื่องมือที่ใช้และปัจจัยแวดล้อม	37

4.2 วัสดุและฉากที่นำมาใช้ในการทดลอง.....	37
4.3 การวัดประสิทธิภาพด้านความแม่นยำ.....	39
4.3.1 จากประเภทวัตถุนิ่ง.....	39
4.3.2 จากประเภทวัตถุเคลื่อนที่และมีการเปลี่ยนแปลงรูปร่าง.....	40
4.4 การวัดประสิทธิภาพด้านความเร็ว.....	41
4.4.1 จากประเภทวัตถุนิ่ง.....	41
4.4.2 จากประเภทวัตถุเคลื่อนที่แต่ไม่มีการเปลี่ยนแปลงรูปร่าง.....	41
4.4.3 จากประเภทวัตถุเคลื่อนที่และมีการเปลี่ยนแปลงรูปร่าง.....	42
4.5 วิเคราะห์ผลการทดลอง.....	44
4.5.1 ประสิทธิภาพด้านความแม่นยำ.....	45
4.5.2 ประสิทธิภาพด้านความเร็ว.....	46
4.5.3 สรุปผลการทดลอง.....	48
บทที่ 5 สรุปผลการวิจัย และแนวทางการพัฒนาต่อ.....	50
สรุปผลการวิจัย.....	50
แนวทางในการพัฒนาต่อ.....	51
รายการอ้างอิง.....	54
ประวัติผู้เขียนวิทยานิพนธ์.....	56

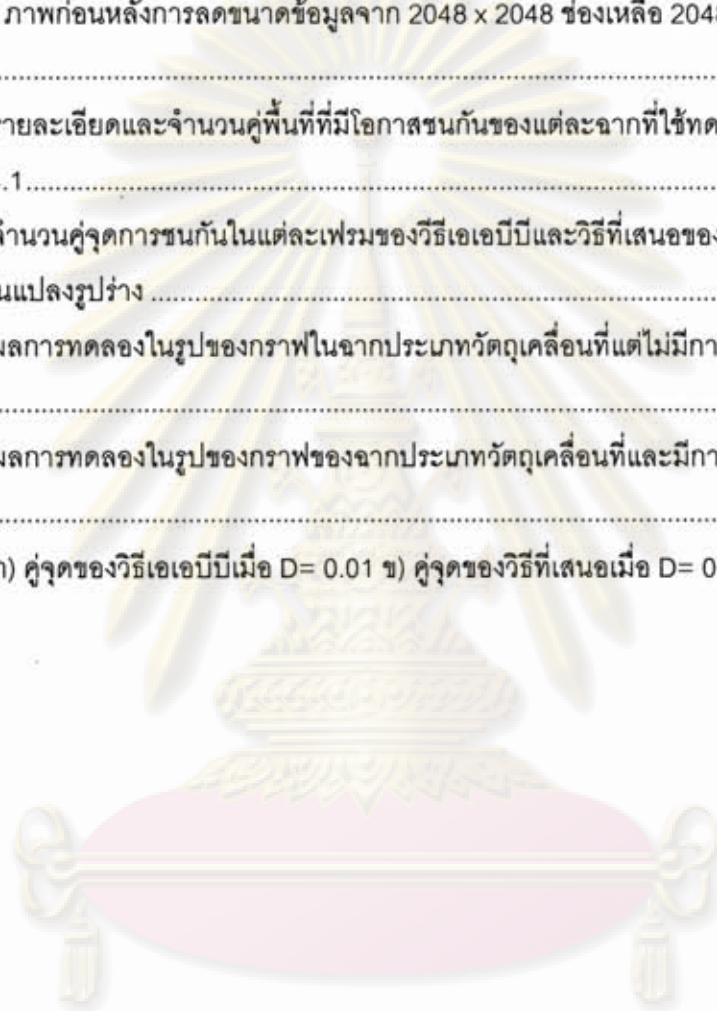


ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

รูปที่ 2-1 การตรวจจับการชนกันของวัตถุในแต่ละขั้นตอน.....	7
รูปที่ 2-2 การสร้างวัตถุตัวแทน 3 ลำดับชั้น.....	7
รูปที่ 2-3 การตรวจสอบการชนกันระหว่างวัตถุที่สี่เหลี่ยมและวัตถุสีแดงด้วยวิธีบีวีเอส.....	8
รูปที่ 2-4 ก) ตารางกริดที่มีขนาดคงที่ ข) ตารางกริดที่มีขนาดไม่คงที่.....	9
รูปที่ 2-5 การตรวจจับการชนกันของวัตถุด้วยวิธีกวาดและตัดออก.....	9
รูปที่ 2-6 การตรวจจับการชนกันของวัตถุด้วยวิธีการหาคุณสมบัติใกล้เคียงที่สุด.....	10
รูปที่ 2-7 ขั้นตอนการทำงานของหน่วยประมวลผลกราฟิก.....	10
รูปที่ 2-8 ข้อมูลของตัวเลข 2 กลุ่มที่จัดเตรียมไว้ในหน่วยความจำลายผิว 1 มิติ 2 ลายผิว.....	12
รูปที่ 2-9 ความละเอียดขนาด 5 x 5 ของของสี่เหลี่ยมจัตุรัส.....	12
รูปที่ 2-10 ชุดคำสั่งที่ใช้ในการประมวลในแต่ละช่อง.....	12
รูปที่ 2-11 ผลลัพธ์หลังการประมวลตามชุดคำสั่งที่กำหนด.....	13
รูปที่ 3-1 ภาพผลลัพธ์หลังผ่านการเตรียมข้อมูลของ 2 วัตถุ.....	21
รูปที่ 3-2 ขั้นตอนการแบ่งวัตถุออกเป็นส่วน ๆ.....	22
รูปที่ 3-3 ขั้นตอนการกำหนดตำแหน่งและรัศมีการทำงานของตัวควบคุมอนุภาค.....	23
รูปที่ 3-4 รูปบนเป็นตัวอย่างของวัตถุที่ใช้จำลอง และรูปล่างคือการตรวจหาการชนกันโดยใช้ตัวควบคุมอนุภาคและอนุภาค.....	24
รูปที่ 3-5 ขั้นตอนการกำหนดตำแหน่งของอนุภาค.....	25
รูปที่ 3-6 ขั้นตอนการเคลื่อนที่ของอนุภาค.....	26
รูปที่ 3-7 ขั้นตอนการเลือกคู่พื้นที่ที่มีโอกาสเกิดการชนกัน.....	27
รูปที่ 3-8 การประมวลผล 3 คู่พื้นที่ที่มีโอกาสชนกันพร้อม ๆ กันในช่องประมวลผลขนาด 2048 X 2048 ช่อง.....	28
รูปที่ 3-9 ขั้นตอนการเลือกหน่วยประมวลผลของแต่ละคู่พื้นที่.....	30
รูปที่ 3-10 การบันทึกข้อมูลของคู่พื้นที่ AB ลงในหน่วยความจำลายผิว TA, TB, VA, VB.....	31
รูปที่ 3-11 หน่วยความจำลายผิว 1 มิติที่บันทึกข้อมูลของ 16 คู่พื้นที่.....	31
รูปที่ 3-12 การเข้าถึงข้อมูลในหน่วยความจำลายผิวของแต่ละช่องประมวลผลเพื่อใช้ในการตรวจสอบการชนกัน.....	33
รูปที่ 3-13 การบันทึกจุดศูนย์กลางลายผิว TA และ VB ที่ผ่านค่า D ลงในหน่วยความจำลายผิวขาออก.....	33

รูปที่ 3-14 การบันทึกจุดของลายผิว TB และ VA ที่ผ่านค่า D ลงหน่วยความจำลายผิวขาออก.	33
รูปที่ 3-15 ตัวอย่างผลลัพธ์ของจุดที่ผ่านค่า D ในช่องสี่เหลี่ยมของคู่พื้นที่หนึ่ง ๆ	33
รูปที่ 3-16 การหาตัวชี้ข้อมูลแต่ละแถวของลายผิวที่เก็บตัวชี้.....	34
รูปที่ 3-17 ขั้นตอนการลดขนาดข้อมูลก่อนอ่านกลับในหนึ่งคู่พื้นที่.....	35
รูปที่ 3-18 ภาพก่อนหลังการลดขนาดข้อมูลจาก 2048 x 2048 ช่องเหลือ 2048 x 2 ช่องจำนวน 2 ลายผิว.....	35
รูปที่ 4-1 รายละเอียดและจำนวนคู่พื้นที่ที่มีโอกาสชนกันของแต่ละฉากที่ใช้ทดลองในการทดลองที่ 3.1 และ 4.1.....	40
รูปที่ 4-2 จำนวนจุดการชนกันในแต่ละเฟรมของวิธีเอเอบีบีและวิธีที่เสนอของฉากวัตถุเคลื่อนที่ และเปลี่ยนแปลงรูปร่าง	41
รูปที่ 4-3 ผลการทดลองในรูปของกราฟในฉากประเภทวัตถุเคลื่อนที่แต่ไม่มีการเปลี่ยนแปลงรูปร่าง	43
รูปที่ 4-4 ผลการทดลองในรูปของกราฟของฉากประเภทวัตถุเคลื่อนที่และมีการเปลี่ยนแปลงรูปร่าง	44
รูปที่ 4-5 ก) จุดของวิธีเอเอบีบีเมื่อ D= 0.01 ข) จุดของวิธีที่เสนอเมื่อ D= 0.01.....	45



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่ 3-1 ผลการทดลองหาเวลาที่ใช้ในการอ่านข้อมูลขนาดต่าง ๆ จากหน่วยประมวลผล กราฟิกกลับมาสู่หน่วยประมวลผลกลาง.....	29
ตารางที่ 3-2 จำนวนช่องข้อมูลที่ต้องอ่านกลับของจำนวนรูปทรงสามเหลี่ยมต่อพื้นที่ขนาดต่าง ๆ	29
ตารางที่ 4-1 วัตถุและรายละเอียดของวัตถุที่นำมาใช้ในการทดลอง	39
ตารางที่ 4-2 ผลการทดลองด้านความแม่นยำกับจากวัตถุหนึ่ง.....	40
ตารางที่ 4-3 ผลการทดลองด้านความเร็วกับจากวัตถุหนึ่ง	41
ตารางที่ 4-4 ผลการทดลองของจากประเภทวัตถุเคลื่อนที่แต่ไม่มีการเปลี่ยนแปลงรูปร่าง	42
ตารางที่ 4-5 ผลการทดลองด้านความเร็วกับวัตถุเคลื่อนที่และเปลี่ยนแปลงรูปร่าง	42



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการตรวจจับการชนกัน เป็นหนึ่งในปัญหาที่สำคัญในงานด้านคอมพิวเตอร์กราฟิก เพราะสามารถนำไปประยุกต์ใช้ในการจำลองการชนกันของวัตถุต่าง ๆ ในโลกเสมือนจริง ซึ่งมีประโยชน์อย่างมากในงานที่ไม่สามารถทำการทดลองการชนกันเหล่านั้นในโลกจริง ๆ ได้ หรือเป็นการยากหากต้องทำ ตัวอย่างเช่น การจำลองการชนกันของรถ การระเบิดของกำแพง การระเบิดของตึก การจำลองการผ่าตัด เป็นต้น

นอกจากนี้ยังมีอีกหนึ่งอุตสาหกรรมที่มีการใช้กันอย่างแพร่หลายก็คือ อุตสาหกรรมเกม ซึ่งมีการเติบโตของจำนวนการชนกันของวัตถุในฉากหนึ่ง ๆ สูงขึ้นเรื่อย ๆ [1] นั้นหมายความว่า จำเป็นต้องมีขั้นตอนวิธีการตรวจจับการชนกันที่ดีขึ้นและเร็วเพียงพอที่จะใช้กับเกมทั้งในปัจจุบันและในอนาคต

ซึ่งโดยปกติขั้นตอนวิธีพื้นฐานที่สุดในการตรวจจับการชนกันนั้นคือ เมื่อมีวัตถุอยู่จำนวน M วัตถุและแต่ละวัตถุมีจุดยอด (vertex) มีด้าน (edge) และมีหน้าของรูปทรง (face) อยู่ทั้งหมด N อัน หากต้องการตรวจสอบว่าบนวัตถุใด ๆ ของแต่ละ M วัตถุนี้มีการชนซึ่งกันและกันหรือไม่ จะเริ่มจากการนำจุดยอดแรกของวัตถุชิ้นแรกมาตรวจสอบกับจุดยอด ด้าน และหน้าของรูปทรงทั้งหมดของวัตถุอื่น ๆ ต่อมาจึงนำด้าน หรือหน้าของรูปทรงหรือจุดยอดต่อมาตรวจสอบแบบเดิมอีกครั้ง ซึ่งจะไล่ทำแบบนี้ไปเรื่อย ๆ จนครบทุกจุดยอด ทุกด้าน ทุกหน้าของรูปทรงบนวัตถุชิ้นแรก แล้วจึงเริ่มทำแบบเดียวกันนี้กับวัตถุชิ้นที่สองจนกระทั่งถึงวัตถุชิ้นสุดท้าย จะเห็นได้ว่าตามวิธีนี้ค่าความซับซ้อนของขั้นตอนวิธีเป็น $\Theta(M^2N^2)$ ดังนั้นวิธีนี้จึงไม่เหมาะสำหรับที่จะใช้ตรวจจับการชนกันที่ต้องการความเร็ว เมื่อ M และ N มีขนาดใหญ่ ซึ่งจำเป็นอย่างมากในงานที่ต้องการความละเอียดสูง

โดยทั่วไปจึงมีขั้นตอนวิธี เพื่อใช้ในการประมวลผลล่วงหน้า (Pre-Processing) ก่อนทำการค้นหาวัตถุที่มีการชนกัน โดยอาศัยการสร้างโครงสร้างข้อมูลของวัตถุตัวแทนซึ่งมีความละเอียดต่ำกว่าวัตถุจริงมาครอบวัตถุจริงเพื่อลดขนาดของ M และ N แล้วจึงใช้วัตถุตัวแทนนี้ตัดวัตถุตัวแทนด้วยกันที่ไม่มีโอกาสเกิดการชนกันออกไป ต่อมาจึงตรวจสอบอย่างละเอียดกับวัตถุจริงที่อยู่ภายในวัตถุตัวแทนที่มีโอกาสเกิดการชนกันในภายหลัง แต่ปัญหาของวิธีนี้จะเกิดขึ้นเมื่อนำไปใช้กับ วัตถุอ่อน (soft-body) [3]

โดยวัตถุที่เกิดการชนกันนั้นสามารถแบ่งออกได้เป็นสองชนิดคือ วัตถุแข็ง (rigid-body) และ วัตถุอ่อน โดยสำหรับวัตถุแข็งแล้ว การตรวจจับการชนกันจำนวนมากสามารถทำได้เร็วกว่าวัตถุอ่อน เนื่องจากไม่ต้องการความละเอียดของผลลัพธ์เท่ากับวัตถุอ่อน และไม่ต้องมีการปรับปรุงหรือสร้างโครงสร้างข้อมูลของวัตถุตัวแทนใหม่ เพราะวัตถุแข็งไม่มีการเปลี่ยนแปลงรูปร่างเหมือนที่เกิดขึ้นกับวัตถุอ่อนที่สามารถเปลี่ยนแปลงรูปร่างได้

ดังนั้นจึงได้มีขั้นตอนวิธีหนึ่งมาแก้ปัญหานี้คือ การหาคุณสมบัติ คือ จุดยอด ด้านและหน้าของรูปทรงที่ใกล้ที่สุดระหว่างวัตถุ (closest-features) [2, 4, 5] ซึ่งไม่ต้องมีการปรับปรุงหรือสร้างโครงสร้างข้อมูลใหม่เพราะไม่มีการเตรียมโครงสร้างข้อมูลขึ้นมาก่อน แต่จะอาศัยจุดยอดเริ่มต้นในการค้นหาขึ้นมาภายในแต่ละวัตถุ แล้วพยายามค้นหาคำตอบโดยขยับไปยังจุดยอดที่มีโอกาสเกิดการชนกันมากที่สุดไปเรื่อย ๆ จนกระทั่งเจอคำตอบ แล้วจึงเก็บคำตอบที่ได้นี้ไว้ใช้เป็นจุดยอดเริ่มต้นในครั้งต่อ ๆ ไป ซึ่งตามธรรมชาติของการชนกันของวัตถุหากเกิดการชนกันในเวลาก่อนหน้า เวลาต่อมาจุดยอดที่จะน่าจะเกิดการชนกันอีกก็ใกล้เคียงกับจุดยอดเดิม หรือจุดยอดข้างเคียงทำให้เวลาในการค้นหาคำตอบลดลง

แต่ปัญหาของวิธีการหาคุณสมบัติที่ใกล้ที่สุดคือ หลายครั้งคำตอบไม่ได้มีเพียงคำตอบเดียว กล่าวคือ วัตถุสามารถเกิดการชนได้ในหลาย ๆ จุดยอดบนตัววัตถุพร้อม ๆ กัน ดังนั้นการสุ่มจุดยอดเริ่มต้นในการค้นหาเพียงจุดยอดเดียวจึงไม่เพียงพอกับสถานการณ์นี้ รวมทั้งหากการสุ่มจุดยอดเริ่มต้นสุ่มไปได้จุดยอดเริ่มต้นที่ไม่ดี และวัตถุที่กำลังตรวจสอบเป็นวัตถุที่มีขนาดใหญ่มีความซับซ้อนมาก ก็ต้องใช้เวลาในการค้นหาเพิ่มขึ้น ดังนั้นเวลาที่ใช้ในการค้นหาข้อมูลในแต่ละครั้งจึงไม่แน่นอนเหมือนกับวิธีที่มีการสร้างโครงสร้างข้อมูลตัวแทน

ดังนั้นจึงมีงานวิจัย [6-10] ที่พยายามแก้ปัญหาจุดยอดเริ่มต้นที่เหมาะสม และปัญหาคำตอบหลายคำตอบ ของวิธีการหาคุณสมบัติที่ใกล้ที่สุด โดยอาศัยการสร้างโครงสร้างข้อมูลตัวแทนขึ้นมาช่วยในขั้นตอนเพื่อกรองพื้นที่ที่น่าจะเป็นจุดยอดเริ่มต้นที่ไม่ดีออกไป แล้วจึงอาศัยสุ่มจุดยอดเริ่มต้นจากบริเวณเหล่านั้น เพื่อใช้ในการค้นหาคำตอบต่อ ๆ ไป นอกจากนั้นยังมีการประมาณค่าจำนวนคำตอบที่จะเกิดขึ้นเพื่อใช้สร้างจุดยอดที่ใช้ค้นหาใหม่ ๆ และหลาย ๆ จุดยอดไปพร้อม ๆ กัน ซึ่งเป็นประโยชน์ในเรื่องของการแลกเปลี่ยนระหว่างความเร็วกับความถูกต้อง นั่นคือหากต้องการความเร็วก็ต้องลดปริมาณจุดยอดที่ใช้ในการค้นหาคำตอบลงแต่ก็ต้องแลกมากับความแม่นยำที่ลดลงเพราะได้คำตอบมาเพียงบางส่วนเท่านั้น ซึ่งหากอาศัยการวัดผลจากตาของมนุษย์ก็พอจะยอมรับในความแม่นยำที่ลดลงได้เพื่อแลกมาซึ่งความเร็วเพราะเห็นความแตกต่างได้ไม่ชัดเจนนัก [7]

แต่เนื่องจากงานวิจัยที่กล่าวมานั้น [6-10] ยังมีการนำโครงสร้างข้อมูลเข้ามาใช้ร่วมด้วย ดังนั้นปัญหาในการต้องปรับปรุงหรือสร้างโครงสร้างข้อมูลตัวแทนขึ้นมาใหม่ ทุกครั้งที่วัตถุอ่อนมีการเปลี่ยนแปลงรูปร่างจึงเพิ่มตามขึ้นมา ถึงแม้ว่าจะไม่เท่ากับวิธีที่ใช้โครงสร้างข้อมูลตัวแทนอย่างเดียวก็ตาม นอกจากนี้ปัญหาที่สำคัญที่สุดของวิธีการหาคุณสมบัติที่ใกล้เคียงที่สุด ก็คือ การที่ต้องคำนวณระยะทางจำนวนมากเพื่อใช้ในการค้นหาจากจุดยอดเริ่มต้นไปยังคำตอบที่มีการชนกัน ซึ่งในความเป็นจริงแล้วแต่ละจุดยอดที่ใช้ในการค้นหาแต่ละคำตอบ หากสามารถทำงานพร้อม ๆ กัน หรือค้นหาคำตอบคู่ขนานกันไปได้อย่างมีประสิทธิภาพสูงขึ้นมา แต่การที่จะทำแบบนี้ได้จำเป็นต้องมีหน่วยประมวลผลที่สามารถทำการประมวลผลแบบคู่ขนานได้ ซึ่งในปัจจุบันหน่วยประมวลผลที่สามารถทำงานแบบขนาน และเป็นที่ยอมรับก็คือหน่วยประมวลผลกราฟิก หรือ จีพียู (GPU: Graphic Processing Units)

จึงได้มีงานวิจัยจำนวนมากที่ได้มีการนำหน่วยประมวลผลกราฟิก มาช่วยในการประมวลผลการตรวจสอบการชนกัน [11] แต่ปัญหาที่สำคัญก็คือ หน่วยประมวลผลกราฟิกไม่ได้ ออกแบบมาเพื่อ งานประมวลผลทั่วไป (general purpose) [1] แต่ออกแบบมาเพื่อใช้ในการแสดงผลหรือเป็นการส่งข้อมูลจากหน่วยประมวลผลกลางหรือซีพียู (CPU) ไปยังหน่วยประมวลผลกราฟิกในทางเดียว ดังนั้นหากงานที่เป็นงานประมวลผลทั่วไปบางประเภทที่ต้องการข้อมูลที่ได้ประมวลผลจากหน่วยประมวลผลกราฟิกกลับคืนมาสู่หน่วยประมวลผลกลาง จะส่งผลให้เกิด ปัญหาความล่าช้าของการเดินทางของข้อมูล และอาจซ้ำกว่าการใช้เพียงหน่วยประมวลผลกลาง อย่างเดียวในการประมวลผล เพราะในปัจจุบันหน่วยประมวลผลกลางเองก็มี หน่วยประมวลผลแบบหลายแกน (multi-core) ซึ่งสามารถจะทำงานที่ต้องการการประมวลผลแบบขนานได้ดี รวมถึงในเรื่องของการตรวจจับการชนกัน [12]

เพราะฉะนั้นสามารถสรุปปัญหาที่เกิดขึ้นกับวิธีการหาคุณสมบัติที่ใกล้เคียงที่สุดได้ดังนี้ คือ ปัญหาการคำนวณระยะทางจำนวนมาก ปัญหาไม่รู้จำนวนคำตอบของการชนกัน ปัญหาจุดยอดเริ่มต้นในการค้นหาคำตอบที่เหมาะสม และปัญหาการเปลี่ยนแปลงรูปร่าง ซึ่งเมื่อนำปัญหาทั้งหมดมาพิจารณาร่วมกับประสิทธิภาพของการทำงานคู่ขนานของหน่วยประมวลผลกราฟิก ก็จะสามารถช่วยแก้ปัญหาได้บางส่วนในเรื่องของการคำนวณ แต่ก็จะต้องแก้ปัญหาข้อจำกัดในด้าน ฮาร์ดแวร์เพิ่มขึ้นมา ดังนั้นถ้าหากสามารถแก้ปัญหาทั้งหมดที่กล่าวมานี้ได้ ก็จะทำให้ไม่จำเป็นต้องอาศัยการสร้างวัตถุตัวแทนขึ้นมาใช้ตรวจสอบการชนกัน และเหมาะสมกับการใช้กับวัตถุอ่อนหรือวัตถุที่มีความละเอียดสูงเป็นอย่างมาก

วัตถุประสงค์ของการวิจัย

วิทยานิพนธ์นี้เสนอวิธีการตรวจจับการชนของวัตถุที่ไม่มีมีการเปลี่ยนแปลงรูปร่างและมีการเปลี่ยนแปลงรูปร่างโดยเฉพาะกับวัตถุความละเอียดสูง ซึ่งสามารถให้ผลลัพธ์ที่ถูกต้องมีความแม่นยำที่เพียงพอเพื่อใช้สำหรับการจำลองการตอบสนองของการชนกัน

ขอบเขตของการวิจัย

1. งานวิจัยนี้สามารถนำมาใช้กับวัตถุที่ไม่มีมีการเปลี่ยนแปลงโครงสร้างเช่น การเพิ่มหรือลด จุดยอด การเปลี่ยนการเชื่อมต่อระหว่างจุดยอด เป็นต้น
2. งานวิจัยนี้เป็นการตรวจจับการชนกันภายนอกวัตถุไม่รวมไปถึงการตรวจจับการชนกันภายในวัตถุเดียวกัน
3. งานวิจัยนี้จะเน้นในส่วนของ การตรวจจับการชนกันขั้นคัดพื้นที่ (narrow phase) และขั้นคัดละเอียด (exact phase)
4. งานวิจัยนี้เป็นการตรวจจับการชนกันแบบไม่ต่อเนื่อง (discrete collision detection)
5. งานวิจัยนี้ใช้ได้กับวัตถุที่ประกอบจากรูปทรงสามเหลี่ยมหรือโพลีกอน (polygon)
6. ในขั้นคัดละเอียดจะเป็นการตรวจสอบระหว่างจุดยอดกับหน้าของรูปทรงสามเหลี่ยม ไม่รวมไปถึงการตรวจจับระหว่างด้านกับด้าน
7. การวัดความแม่นยำจะเป็นการนับจำนวนคู่จุดที่ใกล้กันเกินกว่าค่าที่กำหนดไว้ของสองวัตถุของงานวิจัยนี้เทียบกับวิธีเอเอบีบี

ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำวิธีการตรวจจับการชนที่นำเสนอนี้ ไปใช้ในการตรวจจับการชนกันที่ต้องการความละเอียดสูงและต้องใช้เวลาในการประมวลผลด้วยความเร็วอย่างเช่น งานจำลองการผ่าตัด งานจำลองการชนกันของรถ งานด้านคอมพิวเตอร์แอนิเมชัน เป็นต้น

ขั้นตอนการดำเนินการวิจัย

1. ศึกษาวิธีการตรวจสอบการชนกันแนวกว้าง
2. ศึกษาวิธีการตรวจสอบการชนกันแนวแคบ
3. ศึกษาวิธีการประมวลผลแบบขนาน
4. ศึกษาการทำงานและวิธีการใช้งาน หน่วยประมวลผลกราฟิก

5. ทำการออกแบบขั้นตอนวิธีวิเคราะห์ปัญหาและความต้องให้ได้ตามวัตถุประสงค์
6. ทดสอบวิธีการที่นำเสนอ
7. วิเคราะห์ผลการทดลอง
8. สรุปผลและเรียบเรียงวิทยานิพนธ์

โครงสร้างของวิทยานิพนธ์

โครงสร้างของวิทยานิพนธ์ฉบับนี้ประกอบไปด้วย 5 บทหลักคือ บทนำ ทฤษฎีและงานวิจัยที่เกี่ยวข้อง ระเบียบขั้นตอนวิธีที่เสนอ การทดลองกับผลการทดลอง และ บทสรุปกับแนวทางในการพัฒนาต่อ

ในบทแรกจะกล่าวถึงประวัติความเป็นมาและปัญหา วัตถุประสงค์ ขอบเขตในการวิจัย ขั้นตอนการวิจัย ประโยชน์ที่คาดว่าจะได้รับ โครงสร้างของวิทยานิพนธ์ และผลงานที่ตีพิมพ์จากวิทยานิพนธ์ ต่อมาในบทที่ 2 ได้อธิบายถึงทฤษฎีในการตรวจจับการชนกัน และกล่าวถึงงานวิจัยที่เกี่ยวข้อง บทที่ 3 ได้กล่าวถึงระเบียบขั้นตอนวิธีของงานวิจัยที่เสนอและมีการเปรียบเทียบให้เห็นข้อแตกต่างกับวิธีก่อนหน้าต่อมาในบทที่ 4 ได้อธิบายถึงผลลัพธ์จากการวิจัย และสุดท้ายในบทที่ 5 ได้อธิบายถึงบทสรุปของงานวิจัย และแนวทางในการพัฒนาต่อ เพื่อปรับปรุงให้งานวิจัยมีประสิทธิภาพ และครอบคลุมปัญหาต่าง ๆ มากยิ่งขึ้นต่อไป

ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตอบรับให้ตีพิมพ์เป็นบทความทางวิชาการ ในหัวข้อเรื่อง "A Collision Detection Algorithm Using Adaptive Particle Sensor" โดย ชูติ รุ่งเจริญไพศาล และ พิษณุ คนองชัยยศ ในงานประชุมวิชาการของ "2008 IEEE International Conference on Robotics and Biomimetics (ROBIO2008)" ที่ประเทศไทยในระหว่าง วันที่ 21 - 26 กุมภาพันธ์ 2552 และ เรื่อง "Using Adaptive Particle Sensor and GPU for a Collision Detection in High Resolution Polygonal Objects" โดย ชูติ รุ่งเจริญไพศาล และ พิษณุ คนองชัยยศ ในงานประชุมวิชาการของ "The 6th International Joint Conference on Computer Science and Software Engineering (JCSSE2009)" ที่ประเทศไทยในระหว่าง วันที่ 13 - 15 พฤษภาคม 2552

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

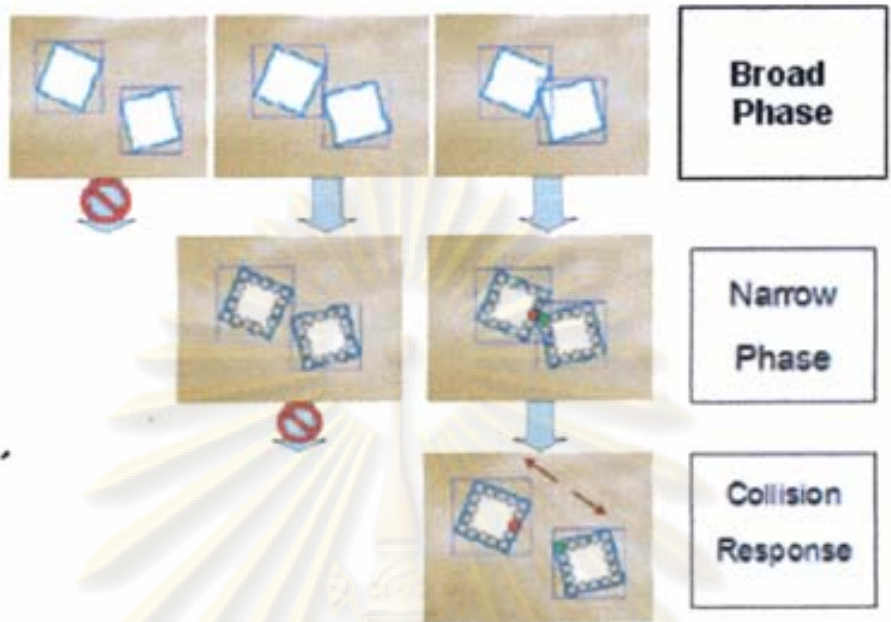
ในบทนี้จะอธิบายถึงแนวคิดและทฤษฎี รวมไปถึงเอกสารงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ เพื่อให้เห็นถึงพื้นหลังของงานวิจัย ปัญหา และแนวทางในการแก้ไขปัญหาในรูปแบบต่างๆ

2.1 แนวคิดและทฤษฎี

ในหัวข้อนี้จะเป็นการอธิบายถึงแนวคิดการตรวจจับการชนกันในรูปแบบต่างๆ และการทำงานของหน่วยประมวลผลกราฟิกพร้อมทั้งตัวอย่างการนำหน่วยประมวลผลกราฟิกไปใช้กับงานประมวลผลทั่วไปที่ไม่เกี่ยวข้องกับการแสดงภาพ ซึ่งมีรายละเอียดดังต่อไปนี้

2.1.1 การตรวจจับการชนกัน (Collision Detection)

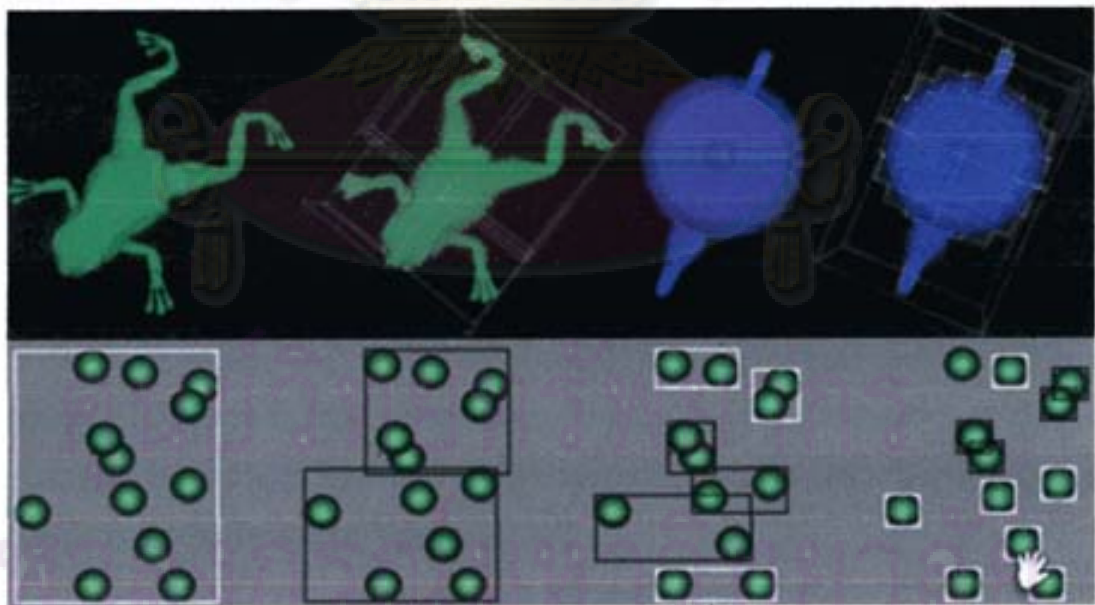
ในการตรวจจับการชนกันนั้นจะแบ่งการทำงานออกเป็นสองส่วนหลักดังรูปที่ 2-1 คือ การตรวจจับอย่างกว้างหรือขั้นคัดคู่วัตถุ (broad phase) ซึ่งเป็นส่วนที่ทำการคัดเลือกเอาเฉพาะคู่ของวัตถุที่มีโอกาสเกิดการชนกันและตัดวัตถุอื่น ๆ ที่ไม่มีโอกาสเกิดการชนกันออกไป และ การตรวจจับการชนกันอย่างแคบหรือขั้นคัดพื้นที่ (narrow phase) คือนำคู่ของวัตถุที่มีโอกาสเกิดการชนกันมาหาว่ามีการชนกันจริงหรือไม่ และชนกันที่จุดใด ระยะที่ใกล้ที่สุดหรือระยะที่ทั้งสองวัตถุเหลื่อมกันมีขนาดเท่าไร ในทิศทางใด เพื่อใช้ในขั้นตอนที่ต่อจากการตรวจจับการชนคือ การตอบสนองการชนกัน ซึ่งในขั้นตอนนี้จะใช้เวลาในการประมวลผลมาก ดังนั้นถ้าหากในขั้นตอนการตรวจจับอย่างกว้างที่ใช้เวลาในการประมวลผลน้อยกว่าไม่สามารถตัดเอาวัตถุที่ไม่เกิดการชนกันออกไปได้ ก็จะส่งผลให้ขั้นตอนการตรวจจับอย่างแคบต้องทำงานหนักมากขึ้น แต่ในทางกลับกันถ้าหากต้องการให้ การตรวจจับอย่างกว้าง ทำงานได้ดีก็จำเป็นต้องให้เวลาในการคำนวณของการตรวจจับอย่างกว้าง เพิ่มมากขึ้นเช่นกัน [1]



รูปที่ 2-1 การตรวจจับการชนกันของวัตถุในแต่ละขั้นตอน

ซึ่งวิธีที่ใช้ในการตรวจสอบการชนกันในปัจจุบัน สามารถแบ่งออกและมีรายละเอียดดังต่อไปนี้

1) การตรวจจับการชนกันด้วยวิธีบีวีเอชส์ (BVHs : Bounding Volume Hierarchies)



รูปที่ 2-2 การสร้างวัตถุตัวแทน 3 ลำดับชั้น

วิธีนี้จะอาศัยการสร้างวัตถุตัวแทนขึ้นมาแทนวัตถุจริง โดยจะเริ่มสร้างโครงสร้างต้นไม้ให้มีลักษณะที่รากของต้นไม้จะเก็บขอบเขตที่ครอบคลุมวัตถุหรือจุดยอดทั้งหมด และลูกของต้นไม้ก็จะ

ครอบคลุมจุดยอดลดลงมาตามลำดับชั้นจนกระทั่งถึงใบต้นไม้ที่เก็บตัววัตถุหรือจุดยอดจริง เพียงวัตถุหรือจุดยอดเดียวดังรูปที่ 2-2

ดังนั้นหากมีการตรวจสอบว่ากลุ่มจุดยอดของวัตถุที่หนึ่งมีการชนกับ กลุ่มจุดยอดของวัตถุที่สอง หรือไม่ ก็สามารถตรวจสอบจากวัตถุตัวแทนของรากของต้นไม้ของแต่ละกลุ่มก่อน ซึ่งหากไม่มีการชนกันก็สามารถที่จะตัดการพิจารณากลุ่มจุดยอดของคู่วัตถุนี้ออกไปได้ โดยที่ไม่ต้องมีการตรวจสอบถึงที่ละจุดยอดของวัตถุจริงภายในกลุ่ม

แต่หากวัตถุตัวแทนของรากต้นไม้ไม่มีการชนกันก็จะไล่พิจารณาไปยังลูกของต้นไม้ทั้งของทั้งสองวัตถุ ไปตามลำดับชั้นจนกระทั่งพบว่า มีหรือไม่มีวัตถุตัวแทนใดมีโอกาสเกิดการชนกัน เหมือนดังรูปที่ 2-3 ที่ทำการตรวจสอบว่าวัตถุกลุ่มสีแดงมีการชนกับวัตถุกลุ่มสีเขียวหรือไม่และมีโอกาสเกิดการชนกันที่ใดภายในกลุ่ม โดยเส้นที่บ่งแทนขอบเขตของวัตถุจริงเพียงจุดยอดเดียวและเส้นประแทนขอบเขตของวัตถุตัวแทนที่ครอบวัตถุจริงหลายจุดยอดอยู่ วิธีนี้จึงเป็นวิธีที่นิยมใช้ในการตรวจสอบอย่างแคบเพื่อคัดหาจุดที่ชนกันจริง ๆ

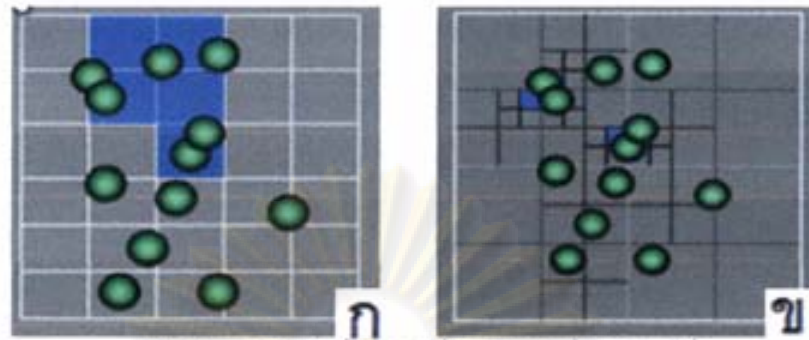


รูปที่ 2-3 การตรวจสอบการชนกันระหว่างวัตถุสีเขียวและวัตถุสีแดงด้วยวิธีบีวีเอส

2) การตรวจจับการชนกันด้วยวิธีตารางกริด (Grid)

วิธีนี้จะอาศัยการสร้างตารางกริดขึ้นมา โดยตารางกริดจะมีสองแบบด้วยกันคือ แบบที่มีขนาดของช่องกริดเท่ากันทุกช่องและแบบที่มีขนาดของกริดไม่เท่ากัน โดยในแต่ละช่องกริดจะมีการเก็บวัตถุหรือจุดยอด ที่อยู่ในขอบเขตของช่องกริดไว้ ทำให้เมื่อทำการตรวจสอบการชนกันหากพบว่าช่องกริดใดมีการเก็บวัตถุหรือจุดยอดได้มากกว่าหนึ่ง ก็แสดงว่ามีโอกาสเกิดการชนกันสูงในบริเวณช่องกริดนั้น ๆ ดังรูปที่ 2-4 ต่อมาจึงทำการตรวจสอบชั้นละเอียดเฉพาะวัตถุจริงที่อยู่ในช่องกริดที่มีโอกาสเกิดการชนกันภายในช่องกริดเท่านั้น

วิธีนี้นิยมใช้ทั้งในการตรวจจับอย่างกว้างเพื่อคัดคู่วัตถุที่มีโอกาสเกิดการชนกัน และการตรวจจับอย่างแคบเพื่อหาจุดที่ชนกันจริง ๆ นอกจากนี้ยังสามารถนำมาใช้ร่วมกับวิธีบีวีเอส โดยใช้วิธีบีวีเอสในการตรวจจับอย่างแคบ และวิธีนี้ในการตรวจจับอย่างกว้าง

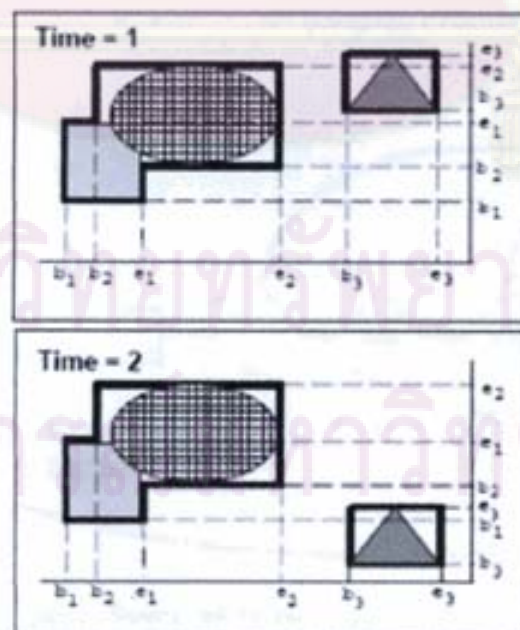


รูปที่ 2-4 ก) ตารางกริดที่มีขนาดคงที่ ข) ตารางกริดที่มีขนาดไม่คงที่

3) การตรวจจับการชนกันด้วยวิธีกวาดและตัดออก (Sweep and Prune)

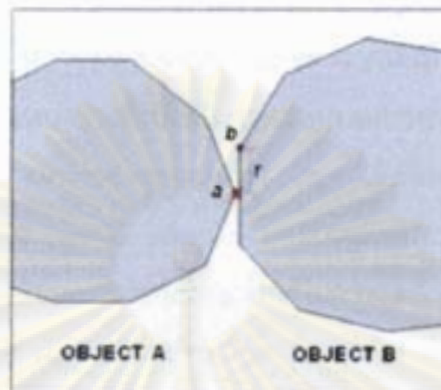
หลักการสำคัญของวิธีนี้คือ ถ้าหากวัตถุใด ๆ มีการชนกันหรือการเชื่อมกันใน 3 มิติ แล้วเมื่อแยกดูทีละมิติ ในทุกมิติที่แยกออกมา (X, Y, Z) ก็ย่อมมีการชนกันหรือเชื่อมกัน

ดังนั้นวิธีนี้จึงทำการสร้างรายการขึ้นมา 3 รายการเพื่อเก็บตำแหน่งจุดยอดของวัตถุตัวแทนของของวัตถุต่าง ๆ ใน แต่ละแกนแต่ละมิติและเพื่อความเร็วในการคำนวณวัตถุทั้งหมดจะถูกแทนที่ด้วยวัตถุตัวแทนรูปทรงกล่องสี่เหลี่ยมหรือเอเอบีบี (AABB: Axis-aligned Bounding Boxes) ที่มีขนาดครอบวัตถุนั้น ๆ ได้พอดี หลังจากนั้นจะทำการเรียงลำดับรายการในแต่ละรายการเพื่อตรวจดูว่ามีวัตถุใดบ้างที่มีการเชื่อมกัน โดยหากเชื่อมกันครบทั้ง 3 แกน นั่นก็หมายความว่าวัตถุใด ๆ นั้นมีโอกาสเกิดการชนกันสูงดังรูปที่ 2-5 ซึ่งวิธีนี้เป็นวิธีที่ใช้ในการตรวจจับอย่างกว้างเพื่อคัดคู่วัตถุที่มีโอกาสเกิดการชนกันขึ้นเป็นหลัก



รูปที่ 2-5 การตรวจจับการชนกันของวัตถุด้วยวิธีกวาดและตัดออก

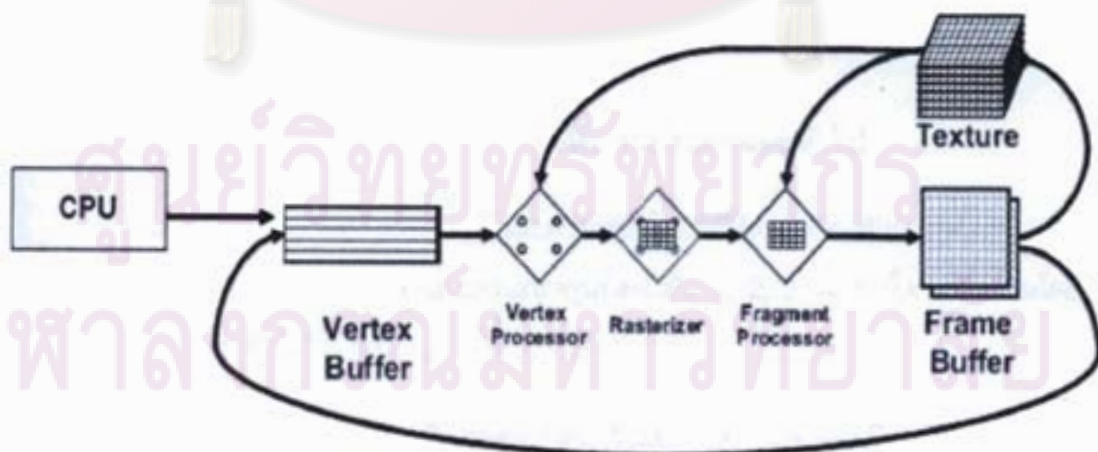
4) การตรวจจับการชนกันด้วยวิธีการหาคุณสมบัติที่ใกล้ที่สุด (Closest-Features)



รูปที่ 2-6 การตรวจจับการชนกันของวัตถุด้วยวิธีการหาคุณสมบัติที่ใกล้ที่สุด

วิธีการนี้จะเป็นการตรวจจับการชนกันอย่างแคบ ซึ่งจะอาศัยการสุ่มคุณสมบัติขึ้นมาในคู่วัตถุที่มีโอกาสเกิดการชนกัน โดยคุณสมบัติจะประกอบไปด้วย จุดยอด ด้าน และ หน้าของรูปทรง ซึ่งจะมีการคำนวณระยะทางระหว่างคู่คุณสมบัติที่เลือก เพื่อให้เปรียบเทียบกับคู่คุณสมบัติที่จะเลือกใหม่จากจุดยอด ด้าน หรือ หน้าของรูปทรง เพื่อนำมา โดยหากระยะทางของคู่คุณสมบัติเพื่อนบ้านได้สั้นกว่าระยะทางของคู่คุณสมบัติเดิมก็จะเลือกคู่คุณสมบัตินั้น เป็นคู่คุณสมบัติต่อไป ทำให้สุดท้ายแล้วจะได้คู่คุณสมบัติที่มีค่าใกล้กันมากที่สุด หรือ เกิดการชนกันดังรูปที่ 2-6 ที่ได้คู่คุณสมบัติเป็นจุดยอดกับด้าน

2.1.2 การทำงานของหน่วยประมวลผลกราฟิก (Graphic Processing Units)



รูปที่ 2-7 ขั้นตอนการทำงานของหน่วยประมวลผลกราฟิก

หลักการการทำงานของหน่วยประมวลผลกราฟิก เพื่อใช้ในการแสดงผลภาพไปสู่จอมอนิเตอร์ดังรูปที่ 2-7 มีรายละเอียดการทำงานดังต่อไปนี้ อันดับแรกจะรับจุดยอดของวัตถุต่าง ๆ จาก หน่วยประมวลผลกลาง มาเก็บไว้ในหน่วยความจำบนการ์ดจอ จากนั้นทำการแปลง ค่าตำแหน่งของจุดยอดนั้น ๆ มาให้เป็นค่าตำแหน่งอ้างอิงที่ใช้ในการแสดงผลสำหรับจอมอนิเตอร์ ซึ่งในขั้นตอนนี้สามารถทำการโปรแกรมค่าตำแหน่งของจุดยอดต่าง ๆ ได้เพิ่มเติม นอกเหนือจากการแปลงตำแหน่งอ้างอิงเพียงอย่างเดียว (vertex shader) ต่อมาหน่วยประมวลผลกราฟิกจะทำการแปลงข้อมูลที่มีความต่อเนื่องให้เป็นข้อมูลที่ไม่ต่อเนื่องเพื่อใช้ในการแสดงผลผ่านจอมอนิเตอร์ (rasterizer) ซึ่งหลังจากทำการแปลงเสร็จเรียบร้อยแล้วจะได้ช่องสี่เหลี่ยม (fragment) จำนวนเท่ากับความละเอียดของการแสดงผลเพื่อใช้ในการกำหนดสีต่าง ๆ ให้ช่องสี่เหลี่ยมนั้น ๆ (fragment shader) โดยสีในแต่ละช่องจะถูกจัดเก็บในรูปของเวกเตอร์ขนาด 4 มิติ คือ (แดง R, เขียว G, น้ำเงิน B, A) และนอกจากจะโปรแกรมสีขึ้นมาเองแล้วยังสามารถนำค่าสีมาจากหน่วยความจำลายผิว (texture buffer) ได้อีกทางหนึ่ง

สุดท้ายพอกำหนดสีให้กับทุกช่องแล้วข้อมูลก็ถูกนำไปแสดงผล ผ่านทางหน่วยความจำแสดงผล (frame buffer) หรือสามารถนำไปเก็บไว้ใน หน่วยความจำลายผิวเพื่อนำไปใช้ต่อไป

จากลักษณะการทำงานในขั้นตอนของ การโปรแกรมค่าตำแหน่งของจุดยอด และ การกำหนดสี ซึ่งสามารถทำงานพร้อม ๆ กันไปได้ในทุก ๆ จุดยอด และ ทุก ๆ ช่องสี่เหลี่ยม อย่างเป็นอิสระต่อกันนี้เอง ทำให้เกิดการ ทำงานแบบขนานขึ้น ซึ่งหากนำไปดัดแปลงเพื่อใช้กับงานประมวลผลทั่วไปนอกจากใช้ในการเลื่อนจุดยอดและการกำหนดสีแล้ว ก็จะทำให้งานประมวลผลทั่วไปนั้นสามารถได้รับประโยชน์จากการประมวลผลแบบขนานของหน่วยประมวลผลกราฟิก ซึ่งมีประโยชน์อย่างมากกับงานที่มีลักษณะของข้อมูลที่เป็นอิสระต่อกันแต่ทำงานอย่างเดียวกัน

(SIMD: Single Program Multiple Data)

ตัวอย่างการนำหน่วยประมวลผลกราฟิกไปใช้ในงานประมวลผลทั่วไป

กำหนดให้หาค่าความต่างน้อยที่สุดและค่าบวกกันมากที่สุดระหว่างตัวเลขสองกลุ่ม โดยให้ตัวเลขกลุ่มแรกคือ 5, 6, 7, 8, 9 และตัวเลขกลุ่มที่สองคือ 1, 2, 3, 4, 5 ซึ่งการทำงานโดยใช้หน่วยประมวลผลกราฟิกในการแก้ปัญหาที่มีขั้นตอนดังต่อไปนี้

1. ทำการจัดเตรียมข้อมูลของตัวเลขสองกลุ่มไว้ในหน่วยความจำลายผิวหนึ่งมิติจำนวน 2 ลายผิว โดยในแต่ละช่องจะมีตัวเลขตัวชี้ (Index) กำกับอยู่ในแต่ละช่องดังรูปที่ 2-8

0	1	2	3	4	5
5	6	7	8	9	

รูปที่ 2-8 ข้อมูลของตัวเลข 2 กลุ่มที่จัดเตรียมไว้ในหน่วยความจำลายฉิว 1 มิติ 2 ลายฉิว

2. กำหนดให้หน่วยประมวลผลกลางส่งจุดยอด 4 จุดยอดของสี่เหลี่ยมจัตุรัสไปยังหน่วยประมวลผลกราฟิก พร้อมทั้งกำหนดความละเอียดของการแสดงผล หรือ กำหนดจำนวนช่องสี่เหลี่ยมจำนวน 5×5 ช่อง เพื่อใช้ในการประมวลผลตามขั้นตอนในรูปที่ 2-7 ซึ่งจะได้ผลลัพธ์ตามรูปที่ 2-9

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

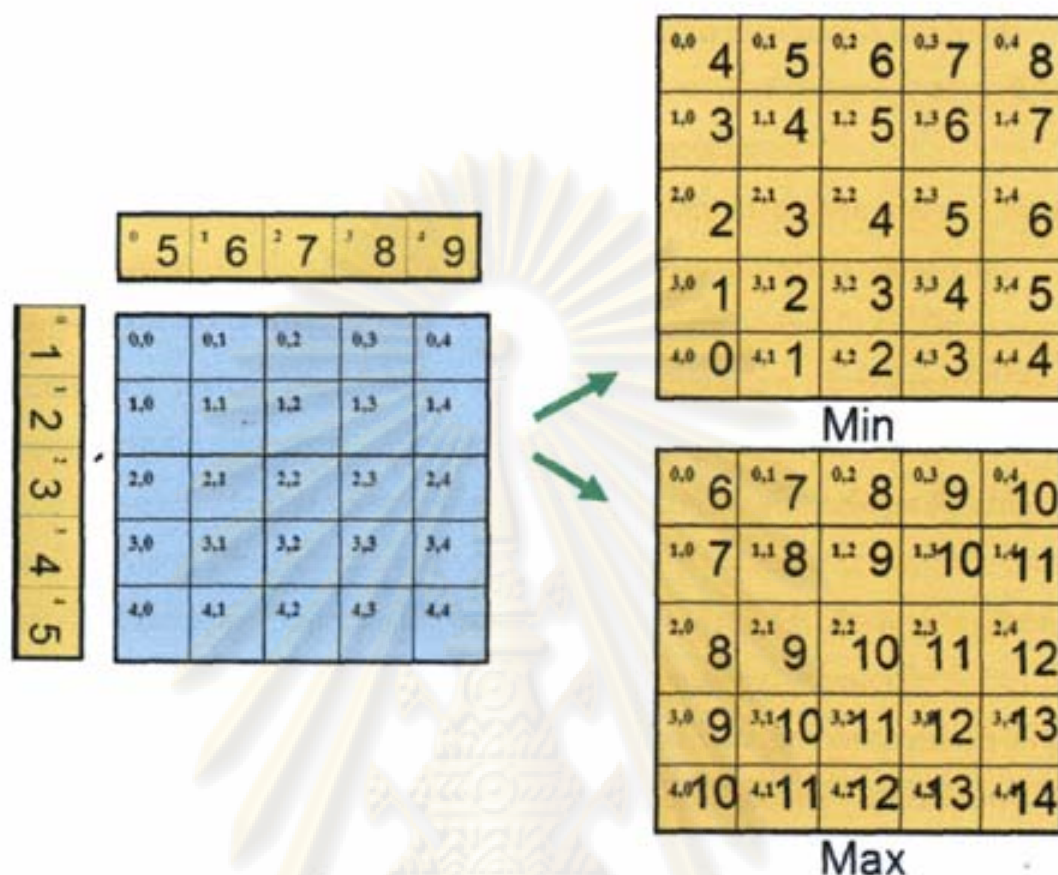
รูปที่ 2-9 ความละเอียดขนาด 5×5 ช่องของสี่เหลี่ยมจัตุรัส

3. กำหนดวิธีการที่ใช้ในการคำนวณค่าสีในแต่ละช่องสี่เหลี่ยม เมื่อกำหนดให้ i และ j คือตัวชี้ของแถวและคอลัมน์ของแต่ละช่อง ทำให้ได้ชุดคำสั่งดังรูปที่ 2-10

$$\begin{aligned} \text{Max} &= \text{TextureGroupOne}[i] + \text{TextureGroupTwo}[j] \\ \text{Min} &= \text{TextureGroupOne}[i] - \text{TextureGroupTwo}[j] \end{aligned}$$

รูปที่ 2-10 ชุดคำสั่งที่ใช้ในการประมวลในแต่ละช่อง

ซึ่งจะทำให้ได้ผลลัพธ์ออกมาสองผลลัพธ์ เก็บไว้ในหน่วยความจำลายฉิวสองมิติจำนวนสอง ลายฉิวดังรูปที่ 2-11



รูปที่ 2-11 ผลลัพธ์หลังการประมวลผลตามชุดคำสั่งที่กำหนด

4. ทำการอ่านข้อมูลจากหน่วยความจำลายผิวกลับมาไปยังฝั่งหน่วยประมวลผลกลางเพื่อหาค่าน้อยที่สุดจากลายผิว *Min* และค่ามากที่สุดจากลายผิว *Max* ซึ่งเป็นคำตอบของปัญหานี้

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

งานวิจัยที่นำหน่วยประมวลผลกราฟิก มาช่วยในการตรวจจับการชนกันในช่วงหัวถึงทกปีที่ผ่านมา สามารถแบ่งออกเป็นสองวิธีหลัก ๆ คือ วิธีทางภาพ [13-16] และ วิธีทางวัตถุ [17, 18] โดย วิธีทางภาพ เป็นการตรวจจับการชนกันของวัตถุโดยอาศัยขั้นตอนในการประมวลผลของภาพที่กำลังจะออกสู่หน่วยความจำแสดงผล โดยตรวจจากค่าความลึก (z-buffer) ของช่องสีเหลี่ยมนั้น ๆ รวมทั้งมี การตรวจผ่านการซ้อนทับของช่องสีเหลี่ยม (occlusion query) ซึ่งล้วนเป็นคำสั่งที่หน่วยประมวลผลกราฟิกจัดสรรให้ใช้ ข้อดีของวิธีนี้ก็คือ สามารถนำไปใช้ได้กับทุกวัตถุทุกรูปทรงรวมทั้งเป็นการทำงานในขั้นการตรวจจับอย่างกว้างที่มีประสิทธิภาพมาก จึงเหมาะกับการนำไปใช้กับวัตถุอ่อนอย่างยิง แต่ข้อเสียหลัก ๆ ก็คือเนื่องจากการแปลงข้อมูลจากที่มีความต่อเนื่องมาสู่ข้อมูลที่ไม่มีความต่อเนื่อง ดังนั้นหากช่วงของความไม่ต่อเนื่องนั้นไม่ละเอียดมากพอหรือห่างกันเกินไป ก็อาจส่งผลให้การตรวจจับการชนกันมีความผิดพลาด รวมทั้งวิธีนี้ยังไม่สามารถ

บอกรายละเอียดได้ว่าเกิดการชนกันที่จุดยอดใด เหลื่อมกันเป็นระยะทางเท่าไร และในทิศทางไหน ส่งผลให้ต้องมีการอ่านข้อมูลกลับมาสู่ หน่วยประมวลผลกลาง เพื่อทำขั้นตอน การตรวจจับอย่าง แคมต่อไป ซึ่งทำให้เกิดปัญหาความล่าช้าของการเดินทางของข้อมูล

ส่วนวิธีทางวัตถุจะเน้นไปในการนำวิธีการที่ใช้ในหน่วยประมวลผลกลางอย่าง บีวีเอส [17] ไปพัฒนาต่อเพื่อให้เหมาะสมกับการทำงานแบบขนานของหน่วยประมวลผลกราฟิก โดยหลักการคือสร้างต้นไม้เก็บไว้ในหน่วยความจำลายฉิว และทำการท่องไปในต้นไม้แบบแนวกว้าง (breadth first search) เพื่อทำให้เกิดการทำงานแบบขนานมากที่สุด ซึ่งข้อดีของวิธีนี้ก็คือจะมีความแม่นยำสูงกว่าวิธีทางภาพ เนื่องจากไม่มีการแปลงข้อมูลไปเป็นแบบไม่ต่อเนื่อง รวมทั้งยังสามารถหาจุดยอด ระยะและทิศทางที่เกิดการชนกันได้ ไม่จำเป็นต้องส่งข้อมูลกลับไปทำ การตรวจจับอย่างแคม ที่หน่วยประมวลผลกลาง แต่ข้อเสียก็คือ ไม่เหมาะกับการตรวจจับวัตถุอ่อน เท้ากับวิธี วิธีทางภาพ เพราะต้องมีการสร้างต้นไม้ทุกครั้ง ซึ่งเป็นวิธีที่เสียเวลามากแม้ว่าทำใน หน่วยประมวลผลกราฟิกแล้วก็ยังต้องอาศัยการประมวลผลพอสมควร ในขณะที่อีกวิธี [18] เลือกที่จะเก็บเฉพาะใบของต้นไม้ แทนที่จะเก็บต้นไม้ทั้งหมดลงในหน่วยความจำลายฉิวทำให้เวลาที่ วัตถุอ่อนเปลี่ยนแปลงรูปร่างเพียงทำการปรับปรุขนาดของใบไม้ก็เพียงพอ แต่ข้อเสียของวิธีนี้คือ ต้องทำการส่งข้อมูลของจุดยอดต่าง ๆ ที่อยู่ในใบไม้ที่ตรวจสอบแล้วว่ามี การชนกันกลับไปให้หน่วย ประมวลผลกลาง เพื่อให้สามารถประมวลผลหารายละเอียดการชนที่มากพอจะนำไปใช้ในขั้นตอน การตอบสนองการชน รวมทั้งวิธีนี้ยังตรวจสอบได้เฉพาะ การตรวจจับอย่างแคมหรือขั้นคัทพื้นที่ได้ เพียงทีละคู่วัตถุเท่านั้น

และอีกวิธีที่ได้นำหน่วยประมวลผลกราฟิกมาใช้งานคือ วิธีกาที่นำอนุภาคจำนวนมากมา แทนที่วัตถุจริง [15] และทำการตรวจจับการชนกันเหมือนการตรวจจับการชนใน ระบบอนุภาค (particle system) ข้อดีของวิธีนี้ก็คือ โดยธรรมชาติของตัวอนุภาคเอง เหมาะกับการทำงานของ หน่วยประมวลผลกราฟิก รวมทั้งยังสามารถนำไปใช้ร่วมกับการจำลองของไหลด้วยวิธีอนุภาคได้ ง่ายและสมจริง แต่ข้อจำกัดของวิธีนี้คือยังไม่มีการนำไปดัดแปลงใช้กับวัตถุอ่อน โดยใช้ได้เพียงกับ วัตถุแข็ง รวมทั้งถ้าต้องการความแม่นยำสูงก็ต้องมีการเพิ่มจำนวนของ อนุภาค มากขึ้นซึ่งส่งผลให้ การทำงานช้าลงแม้จำนวนอนุภาคเมื่อเทียบกับจุดยอดของวัตถุแล้วจะยังมีจำนวนน้อยกว่าก็ตาม

นอกจากนั้นยังมีวิธีการตรวจจับการชนกันที่เหมาะสมกับวัตถุอ่อน แต่ยังไม่มีการปรับใช้ กับหน่วยประมวลผลกราฟิก นั่นก็คืองานวิจัยที่ปรับปรุงวิธีการหาคุณสมบัติที่ใกล้ที่สุด [2] ให้เป็น วิธีที่เหมาะสมกับวัตถุอ่อน [6-10]

ได้แก่ วิธีแรก วิธีการสุ่มเพิ่มจุดยอดเริ่มต้นค้นหาการชนกันให้มากกว่าหนึ่งจุดยอด [10] ทำให้โอกาสที่จะได้จุดยอดเริ่มต้นที่ไม่ดีมีน้อยลง แต่วิธีนี้ก็ยังไม่สามารถแก้การคำนวณเพื่อใช้

ค้นหาคำตอบปริมาณมากได้ซึ่งหากเป็นวัตถุที่มีขนาดใหญ่และมีความละเอียดมาก วิธีการหาจำนวนและตำแหน่งจุดยอดเริ่มต้นที่เป็นการสุ่มทำให้มีโอกาสที่จะได้ตำแหน่งและจำนวนที่ไม่ดี และส่งผลต่อเวลาที่ต้องใช้คำนวณหาคำตอบที่ต้องเพิ่มมากขึ้น

วิธีอาศัยการกำหนดวัตถุตัวแทนซึ่งมีความละเอียดหลายระดับ (Multi-Resolution) [6] ขึ้นมาแทนตัววัตถุจริง ๆ ทำให้เวลาที่วัตถุอยู่ใกล้กันก็จะใช้ความละเอียดอย่างหยาบ ทำให้จำนวนจุดยอดที่ต้องเดินทางหาคำตอบน้อยลง รวมทั้งจำนวนจุดยอดที่ต้องสุ่มมาเพื่อเป็นจุดยอดเริ่มต้นในการค้นหาก็น้อยลงตามไปด้วย ซึ่งเมื่อวัตถุเข้ามาใกล้กันจึงมีการปรับความละเอียดให้มีความละเอียดมากขึ้น เพื่อใช้ในการค้นหาคำตอบที่เกิดจากการชนกันจริง ๆ นอกจากนี้ยังมีการเก็บคำตอบที่ดีไว้รวมทั้งมีการเพิ่มจุดยอดเริ่มต้นใหม่ ๆ ลงไป เพื่อใช้ในการค้นหาการชนในบริเวณอื่น ๆ ทำให้แก้ปัญหาการคำนวณที่มากเกินไปและมีจุดยอดเริ่มต้นที่ใช้ในการค้นหาคำตอบที่เพียงพอและทั่วถึงรอบทั้งวัตถุ แต่ข้อเสียของวิธีนี้ก็คือหากวัตถุอ่อนมีการเปลี่ยนแปลงรูปร่างไปจะต้องมีการแก้ไขโครงสร้างของวัตถุตัวแทนตามไปด้วย ซึ่งทำให้ต้องประสบปัญหาคล้ายกันกับการปรับปรุงหรือสร้างโครงสร้างใหม่แบบที่เกิดขึ้นกับวิธีบีวีเอส

วิธีแก้ปัญหาจำนวนจุดยอดเริ่มต้นค้นหาไม่เพียงพอ และตำแหน่งไม่เหมาะสม [7, 8] โดยการนำวิธีบีวีเอสเข้ามาช่วยกรองพื้นที่ที่ไม่น่าจะเกิดการชนกันออกไป จากนั้นจึงค่อยกำหนดจุดยอดเริ่มต้นทั้งหลายให้กับเฉพาะพื้นที่ที่มีโอกาสชนกัน ข้อดีของวิธีนี้คือ เร็วกว่าการใช้วิธีบีวีเอสเพียงอย่างเดียว เนื่องจากมีการเก็บคำตอบที่ดีเอาไว้ไม่ต้องหาใหม่ทุกครั้ง และเร็วกว่าวิธีการหาคุณสมบัติที่ใกล้ที่สุดเพียงอย่างเดียว เพราะได้จุดยอดเริ่มต้นในการค้นหาที่ดีกว่า แต่ข้อเสียของวิธีนี้ก็จะยังเป็นข้อเสียเดียวกันกับวิธีบีวีเอสแม้ว่าการปรับปรุงโครงสร้างจะง่ายกว่าเพราะมีลูกในต้นไม้น้อยกว่าอันเนื่องมาจากการใช้วิธีการหาคุณสมบัติที่ใกล้ที่สุดเข้าช่วยก็ตาม

วิธีสุดท้าย [9] วิธีแก้ไขปัญหาค่าตำแหน่งของจุดยอดเริ่มต้นในการค้นหา โดยอาศัยการแบ่งพื้นผิววัตถุออกเป็นส่วน ๆ ซึ่งจำนวนส่วนที่เหมาะสมจะขึ้นอยู่กับเลขคิสลิงนัมเบอร์ (Kissing Number) ซึ่งเป็นจำนวนตัวเลขที่ระบุว่ารูปร่างทรงกลมหนึ่ง ๆ สามารถโดนสัมผัสด้วยทรงกลมที่มีขนาดเท่ากันได้กี่ทรงกลมพร้อม ๆ กัน และเมื่อแบ่งวัตถุออกเป็นส่วน ๆ ตามตัวเลขนี้แล้วกำหนดจุดยอดเริ่มต้นให้แก่แต่ละพื้นที่ ก็จะทำให้ได้จุดยอดเริ่มต้นที่ดีและใกล้กับตำแหน่งที่คำตอบอยู่หากมีการชนกันในบริเวณนั้น แต่ปัญหาของวิธีนี้ก็คือ หากวัตถุอ่อนมีการเปลี่ยนแปลงรูปร่างจนเกินที่จุดยอดเริ่มต้นจุดยอดเดียวต่อหนึ่งพื้นที่จะเพียงพอต่อคำตอบก็ต้องมีการคำนวณและแบ่งพื้นที่ใหม่ ซึ่งสุดท้ายจะกลายเป็นปัญหาที่คล้ายกับวิธีบีวีเอสอีกนั่นเอง แต่ต่างกันตรงที่วิธีบีวีเอสต้องมีการปรับปรุงโครงสร้างข้อมูลเพราะต้องใช้โครงสร้างข้อมูลใหม่นั้นในการหาคำตอบ ในขณะที่วิธีนี้ต้องแบ่งพื้นที่ใหม่เพราะจำนวนจุดยอดที่ใช้ค้นหาคำตอบไม่เพียงพอ ซึ่งถ้าสามารถ

ทำให้มีจุดยอดที่ใช้ค้นหามากพอ ก็จะทำให้วิธีนี้ไม่จำเป็นต้องแบ่งพื้นที่ใหม่ แต่วิธีนี้ก็ยังมีปัญหาที่สำคัญอีกอย่างก็คือผลลัพธ์ของวิธีนี้ยังไม่เพียงพอที่จะใช้ในการจำลองการตอบสนองการชนกันได้ในทันที

จากงานวิจัยที่เกี่ยวข้องดังกล่าวมาข้างต้น จึงสามารถกล่าวโดยสรุปได้ว่า เมื่อไม่ต้องการใช้วิธีที่ต้องมีการจัดการกับวัตถุตัวแทน ก็จำเป็นต้องมีการแลกเปลี่ยนระหว่างความแม่นยำกับความเร็ว โดยใช้วิธีทางระบบอนุภาค หรือ วิธีที่ปรับปรุงจากการหาคุณสมบัติใกล้เคียงที่สุด ดังนั้นงานวิจัยที่เสนอนี้จึงมุ่งประเด็นไปที่การลดข้อจำกัดในเรื่องของการแลกเปลี่ยนระหว่างความแม่นยำกับความเร็ว เมื่อไม่มีการใช้ และจัดการกับวัตถุตัวแทน



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

ระเบียบขั้นตอนวิธีที่เสนอ

ระเบียบขั้นตอนวิธี

จากปัญหาและข้อจำกัดของวิธีที่ไม่ใช้วัตถุตัวแทน [6-10] โดยเฉพาะปัญหาการคำนวณระยะทางจำนวนมาก งานวิจัยนี้จึงเสนอให้ การตรวจสอบการเข้าใกล้กันของคู่วัตถุโดยอนุภาคนั้น เป็นการทำงานในขั้นตอนการตัดคู่พื้นที่ แทนที่จะเป็นการตรวจสอบขั้นละเอียดเหมือนในวิธีที่ผ่านมา [6-10] โดยในขั้นละเอียดนั้นจะเป็นการใช้หน่วยประมวลผลกราฟิกทำการตรวจสอบแบบขนานในทุก ๆ คู่พื้นที่ที่ตัดมาพร้อม ๆ กันแทน ดังนั้นจึงต้องมีการกำหนดขอบเขตบริเวณพื้นที่ที่รับผิดชอบของแต่ละอนุภาคขึ้นมาก่อนเพื่อใช้เป็นข้อมูลสำหรับการตัดคู่พื้นที่ โดยทำการแบ่งวัตถุออกเป็นส่วน ๆ

เนื่องจากการตรวจสอบขั้นละเอียดใช้หน่วยประมวลผลกราฟิกในการประมวลผล ดังนั้นจึงต้องมีการโอนถ่ายข้อมูลของรูปทรงสามเหลี่ยมของคู่พื้นที่ที่ตัดมาไปยังหน่วยประมวลผลกราฟิกซึ่งสามารถทำได้โดยเขียนข้อมูลของรูปทรงสามเหลี่ยมเหล่านั้นลงไปในลายฉิว แต่จากการทดลองศึกษาการทำงานของหน่วยประมวลผลกราฟิกทำให้พบว่า การเขียนข้อมูลลงในลายฉิวที่เตรียมไว้ก่อนนั้น จะใช้เวลาน้อยกว่าการสร้างลายฉิวขึ้นมาใหม่แล้วจึงทำการเขียนลงภายหลัง ดังนั้นงานวิจัยนี้ จึงได้ออกแบบให้มีการเตรียมลายฉิวขนาดคงที่ไว้ก่อนแทนที่จะสร้างใหม่ในทุกการทำงาน ซึ่งส่งผลให้การแบ่งวัตถุออกเป็นส่วน ๆ นั้นในแต่ละส่วนหรือแต่ละพื้นที่ต้องมีขนาดของจำนวนรูปทรงสามเหลี่ยมน้อยกว่าหรือ เท่ากับขนาดความจุของลายฉิวที่เตรียมไว้ งานวิจัยนี้จึงแตกต่างกับ วิธีการแบ่งพื้นที่เป็นส่วน ๆ ตามคิสส์ลิ่งมัมเบอร์ [9] เพราะเป็นการแบ่งพื้นที่เพื่อให้สอดคล้องกับการทำงานของหน่วยประมวลผลกราฟิก

เนื่องจากการแบ่งพื้นที่แบ่งโดยอาศัยจำนวนของรูปทรงสามเหลี่ยม ไม่ได้พิจารณาถึงขนาดของพื้นที่ว่าต้องมีขนาดเท่ากัน ดังนั้นจึงทำให้สามารถเกิดกรณีหนึ่งพื้นที่เข้าใกล้กับสองพื้นที่หรือมากกว่านั้นพร้อม ๆ กัน ซึ่งในกรณีนี้หนึ่งอนุภาคต่อหนึ่งพื้นที่จะไม่เพียงพอต่อขั้นตอนการตัดคู่พื้นที่ งานวิจัยนี้จึงได้เสนอให้มีตัวควบคุมอนุภาคประจำอยู่ในแต่ละพื้นที่ ซึ่งมีหน้าที่ทำการตรวจสอบว่าในแต่ละพื้นที่นั้น ควรให้มีจำนวนอนุภาคที่ใช้ในการตรวจสอบการเข้าใกล้กันจำนวนเท่าใดจึงจะเพียงพอต่อการทำงานช่วงเวลานั้น ๆ

ดังนั้นขั้นตอนการทำงานของวิธีที่เสนอนี้ จึงสามารถแบ่งออกได้ดังนี้

3.1 การเตรียมการประมวลผลก่อนตรวจสอบการชนกัน

3.2 การตรวจสอบการชนกันขั้นคัดคู่วัตถุและคัดคู่พื้นที่

3.3 การตรวจสอบการชนกันขั้นละเอียด

เพื่อความเข้าใจของการทำงานในแต่ละขั้นตอน จึงมีการกำหนดนิยามของข้อมูลที่ใช้ในวิธีที่เสนอดังมีรายละเอียดต่าง ๆ ดังต่อไปนี้

นิยาม 3.1 วัตถุ O_i คือ เซตของรูปทรงสามเหลี่ยม โดยรูปทรงสามเหลี่ยม i ประกอบขึ้นมาจากจุดยอด $\{v_1, v_2, v_3\}$ และ ด้าน $\{e_1, e_2, e_3\}$

นิยาม 3.2 วัตถุทั้งหมดในระบบ G เมื่อ $G = \{O_i; i=1, 2, \dots, n\}$ และ n เป็นจำนวนวัตถุในระบบ

นิยาม 3.3 พื้นที่ย่อยของวัตถุ S_j เมื่อ $S_j \subseteq O_i$ โดย $\bigcup_{j=1}^J S_j = O_i$ และ $\bigcap_{j=1}^J S_j = \emptyset$ เมื่อ J คือจำนวนของพื้นที่ย่อยทั้งหมดบนแต่ละ O_i

นิยาม 3.4 อนุภาค p_k คือ วัตถุไม่มีขนาดมีลักษณะประจำตัวดังนี้

- ตำแหน่งของอนุภาค อยู่บนจุดยอด v_i เมื่อ $v_i \in S_j$ โดย p_k สามารถเปลี่ยนเป็นจุดยอดอื่น ๆ ที่มีเส้นเชื่อมกับจุดยอดเดิม เพื่อใช้ในการตรวจสอบการชนกัน
- อายุของอนุภาค คือ สถานะของ p_k โดยหาก p_k มีชีวิตอยู่ค่าอายุจะเท่ากับ 1 แต่ถ้าไม่มีชีวิตค่าอายุจะเท่ากับ 0

นิยาม 3.4 ตัวควบคุมอนุภาค U_j คือ วัตถุบนจุดยอด v_i เมื่อ $v_i \in S_j$ มีหน้าที่เป็นตัวแทนของทุกจุดยอดบน S_j โดยมีลักษณะประจำตัวอยู่สองอย่างคือ ตำแหน่ง u และรัศมีการทำงาน R เพื่อใช้ในการตรวจสอบอย่างหยาบว่า S_j นี้มีโอกาสเกิดการชนกันหรือไม่

กำหนดให้ $\|v_{i1} - v_{i2}\|$ เป็นสัญลักษณ์แทน การคิดระยะทางแบบยูคลิดีเนียน (Euclidean distance) ระหว่างเวกเตอร์ v_{i1} และเวกเตอร์ v_{i2} และให้ u, \bar{x}, v_i เป็นเวกเตอร์ขนาดสามมิติ R เป็นปริมาณสเกลาร์

- ตำแหน่ง u คือ จุดยอดที่อยู่ใกล้ตำแหน่งเฉลี่ย \bar{x} น้อยที่สุด โดย $u = v_i$ ที่ทำให้

$$\min\{\|\bar{x} - v_i\|; v_i \in S_j\} \text{ เมื่อ } \bar{x} = \frac{\sum_{i=1}^n v_i}{n} \text{ โดยที่ } i=1, 2, \dots, n \text{ และ } n \text{ เป็นจำนวนจุดยอด}$$

ทั้งหมดภายในเซต S_j

- รัศมีการทำงาน R คือ ระยะทางมากที่สุดระหว่างตำแหน่งเฉลี่ย \bar{x} กับจุดยอดบน S_j โดย $R = \max\{\|\bar{x} - v_i\|; v_i \in S_j\}$ เมื่อ $i = 1, 2, \dots, n$ และ n เป็นจำนวนจุดยอดทั้งหมดภายในเซต S_j

นิยาม 3.6 คู่วัตถุทั้งหมดในระบบ G' เกิดจากนำสมาชิกของเซต G มาจับคู่กัน โดย $G' = \{O_{ij} \mid \{O_i, O_j\} \in O_{ij}; i = 1, 2, \dots, n; j = 1, 2, \dots, n; i \neq j\}$ เมื่อ n เป็นจำนวนวัตถุทั้งหมด

นิยาม 3.7 คู่ตัวควบคุมอนุภาค U_{ij} เกิดจากการจับคู่ของตัวควบคุม U_i บนวัตถุ O_i กับตัวควบคุม U_j บนวัตถุ O_j ของคู่วัตถุ O_{ij} โดยมีเงื่อนไขในการจับคู่คือ $f(u_i, u_j) > 0$ เมื่อ u_i และ u_j คือ ตำแหน่งของตัวควบคุมอนุภาค U_i และ U_j ตามลำดับ ดังนั้น $U_{ij} = \{U_{ij}^k \mid f(u_i^k, u_j^k) > 0; \{u_i^k, u_j^k\} \in U_{ij}^k; k = 1, 2, \dots, n\}$ เมื่อ n คือจำนวนคู่ที่ผ่านเงื่อนไข $f(u_i, u_j) > 0$ และ

$$f(u_i, u_j) = \begin{cases} 1; \|u_i - u_j\| < R_i + R_j \\ 0; \|u_i - u_j\| \geq R_i + R_j \end{cases}$$

เมื่อ R_i คือ รัศมีการทำงานของตัวควบคุม U_i
 R_j คือ รัศมีการทำงานของตัวควบคุม U_j

นิยาม 3.8 คู่อนุภาค P_{ij} คือสมาชิกเซต U_{ij} ในนิยาม 3.7 ที่แต่ละสมาชิกมีการเปลี่ยนแปลงค่าตำแหน่ง โดยมีเงื่อนไขในการเปลี่ยนแปลงคือ $f_1(p_{ij}) > 0$ โดยก่อนมีการเปลี่ยนแปลงค่าตำแหน่ง $p_{ij} = u_{ij}$ เมื่อ $p_{ij} = \{p_i, p_j\} = \{u_i, u_j\}$ และหลังการเปลี่ยนแปลงค่าตำแหน่ง $p_{ij} = \{p_i^m, p_j^n\}$

$$\text{โดย } f_1(p_{ij}) = \begin{cases} 1; \|p_i - p_j\| > \min\{\|p_i^m - p_j^n\|\} \\ 0; \|p_i - p_j\| \leq \min\{\|p_i^m - p_j^n\|\} \end{cases}$$

เมื่อ $m = 1, 2, \dots, M$ และ M คือจำนวนเพื่อนบ้านของ p_i
 $n = 1, 2, \dots, N$ และ N คือจำนวนเพื่อนบ้านของ p_j

นิยาม 3.9 คู่พื้นที่ที่ย่อยที่มีโอกาสชนกัน S_{ij}^* คือคู่พื้นที่ย่อย S_{ij} ของของแต่สมาชิกจากเซต P_{ij} ที่สมาชิกทุกตัวไม่มีการเปลี่ยนแปลงค่าตำแหน่งและผ่านเงื่อนไข $f_2(p_{ij}) > 0$ โดย $S_{ij} = \{S_i, S_j\}$ เมื่อ $p_i \in S_i$ และ $p_j \in S_j$ เมื่อ $p_{ij} = \{p_i, p_j\}$ และ $p_{ij} \in P_{ij}$ ดังนั้น

$S_y^* = \{S_y^k \mid f_2(p_y^k) > 0; p_y^k \in S_y^k; k = 1, 2, \dots, n\}$ เมื่อ n คือจำนวนคู่วัดที่ผ่านเงื่อนไข $f_2(p_y) > 0$ และ

$$f_2(p_y) = \begin{cases} 1; \|p_i - p_j\| < \text{longestEdge}(p_i, p_j) \\ 0; \|p_i - p_j\| \geq \text{longestEdge}(p_i, p_j) \end{cases}$$

เมื่อ $\text{longestEdge}(p_i, p_j) = \max \{ \max \{ \|p_i - p_m\| \}, \max \{ \|p_j - p_n\| \} \}$

$m = 1, 2, \dots, M$ และ M คือจำนวนเพื่อนบ้านของ p_i

$n = 1, 2, \dots, N$ และ N คือจำนวนเพื่อนบ้านของ p_j

นิยาม 3.10 วัตถุตัวแทนทรงกล่องสี่เหลี่ยม คือ วัตถุสามมิติเซต B โดย $B = \{b_1, b_2\}$ เมื่อ b_1, b_2 เป็นเวกเตอร์สามมิติที่ได้จาก ให้จุดยอด $v_i \in V$ ซึ่งประกอบไปด้วย $\{x_i, y_i, z_i\}$ เมื่อ V คือเซตของจุดยอดของวัตถุหนึ่ง ๆ แล้ว

$b_1 = \{\min\{x_i\}, \min\{y_i\}, \min\{z_i\}\}$ ซึ่งเรียกว่า มุมล่างสุดของของกล่อง

$b_2 = \{\max\{x_i\}, \max\{y_i\}, \max\{z_i\}\}$ ซึ่งเรียกว่า มุมบนสุดของของกล่อง

เมื่อ $i, j, k = 1, 2, \dots, n$ และ n คือขนาดของเซต V

นิยาม 3.11 ระยะทางห่างน้อยที่สุดระหว่างคู่วัด (Minimum separate distance: \dot{D}) คือ ระยะทางที่เป็นตัวกำหนดว่าเกิดการชนกันหรือไม่ โดยหากระยะทางระหว่างคู่วัดที่ได้มาจากจุดยอดกับรูปทรงสามเหลี่ยมมีค่าน้อยกว่าค่า D จะถือว่าคู่วัดนั้น ๆ เป็นหนึ่งในจุดยอดที่เกิดการชนกันของสองวัตถุ

นิยาม 3.12 ช่องประมวลผล คือ ช่องที่ทำหน้าที่ในการประมวลผล งานต่าง ๆ โดยอ่านข้อมูลขาเข้าจากลายผิวที่เตรียมไว้ ผ่านเลขตัวชี้ประจำแต่ละช่องเพื่อใช้ในการเข้าถึงข้อมูล จากนั้นจึงนำข้อมูลที่ได้อ่านประมวลผลแล้วจึงเขียนผลลัพธ์ที่ได้ลงไปนลายผิวอีกแผ่นที่จัดเตรียมไว้ เพื่อใช้เป็นข้อมูลขาออก โดยจะเขียนลงไปนลายผิวข้อมูลขาออกตำแหน่งเดียวกันกับตำแหน่งของช่องประมวลผล

นิยาม 3.13 ลายผิว (Texture) เป็นเซตของข้อมูลประกอบขึ้นจากช่องที่ใช้เก็บข้อมูลเวกเตอร์สี่มิติ $\{d_1, d_2, d_3, d_4\}$ โดยแบ่งออกเป็น 2 ชนิดที่ใช้ในงานวิจัยนี้คือ แบบหนึ่งมิติ $Tex1D$ และแบบสองมิติ $Tex2D$

$$\text{Tex1D} = \{d_i \mid i = 1, 2, \dots, I\}$$

$$\text{Tex2D} = \{d_{ij} \mid i = 1, 2, \dots, I; j = 1, 2, \dots, J\}$$

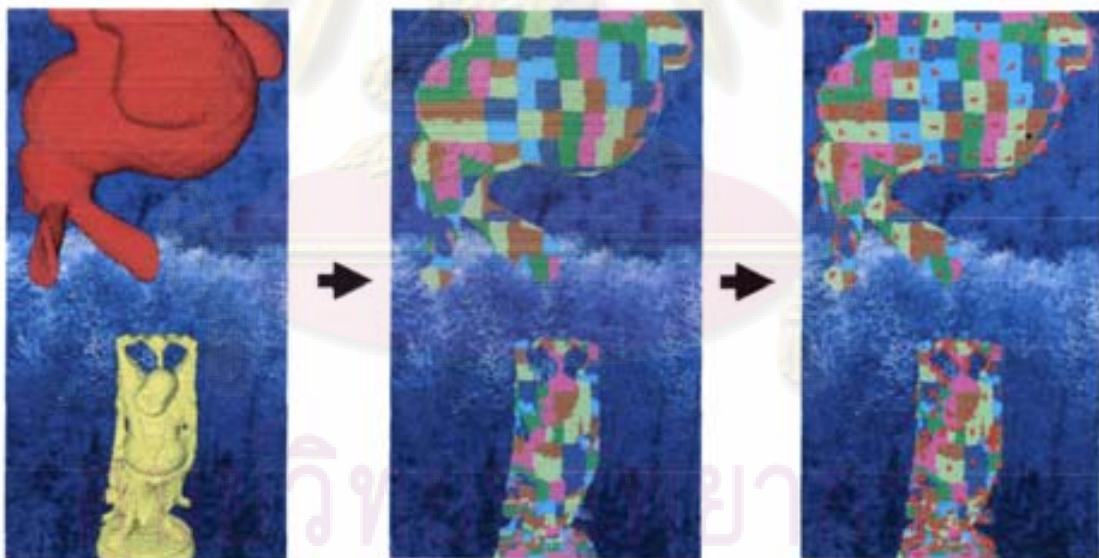
เมื่อ I คือจำนวนช่องที่ใช้เก็บข้อมูลในมิติที่หนึ่งหรือคอลัมน์

J คือจำนวนช่องที่ใช้เก็บข้อมูลในมิติที่สองหรือแถว

โดยลายผิวมันจะมีคุณสมบัติที่สำคัญอย่างหนึ่งคือ อ่านและเขียนในคราวเดียวกันไม่ได้ จึงทำให้ต้องมีการเตรียมว่าลายผิวใดจะเป็นข้อมูลขาเข้าเพื่อใช้อ่าน และลายผิวใดสำหรับข้อมูลขาออกเพื่อใช้เขียน

3.1 การเตรียมการประมวลผลก่อนตรวจสอบการชนกัน

ในขั้นตอนนี้จะทำการแบ่งวัตถุออกเป็นส่วน ๆ โดยในแต่ละส่วนหรือแต่ละพื้นที่นั้นจะประกอบไปด้วยรูปทรงสามเหลี่ยมไม่เกิน 512 รูปทรงสามเหลี่ยม เหตุที่ต้องเป็น 512 รูปทรงสามเหลี่ยมนั้นจะขอยกไปกล่าวในเรื่องการตรวจจบการชนกันชั้นละเอียด ซึ่งจะอธิบายถึงข้อจำกัดของหน่วยประมวลผลกราฟิกเพิ่มเติม

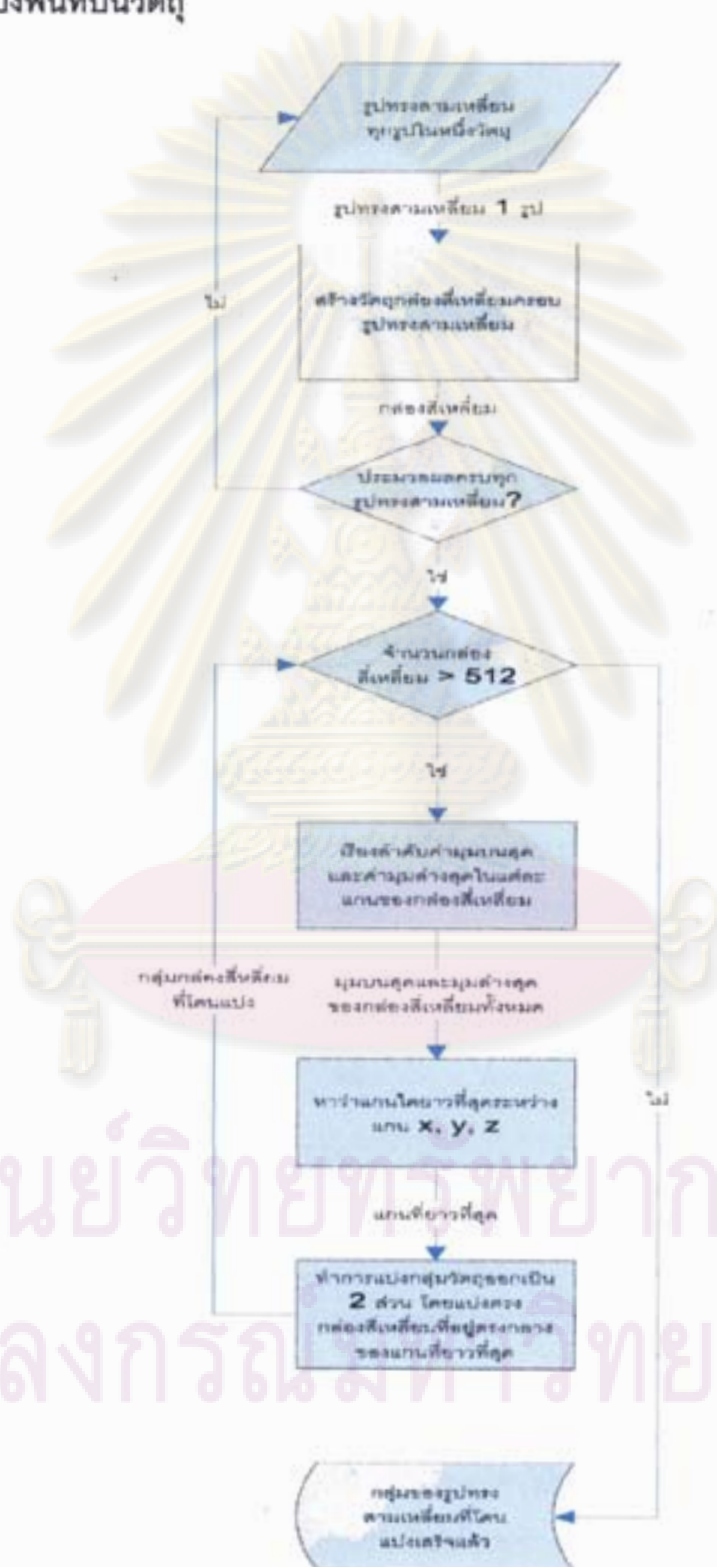


รูปที่ 3-1 ภาพผลลัพธ์หลังผ่านการเตรียมข้อมูลของ 2 วัตถุ

โดยในแต่ละพื้นที่ที่มีการแบ่งออกมานั้น จะมีการกำหนดจุดยอดสำหรับเป็นทิวางตัวควบคุมอนุภาค (Emitter) ดังจุดยอดสีแดงในรูปที่ 3-1 ซึ่งตัวควบคุมอนุภาคนี้อาจใช้ในการตรวจหาพื้นที่ที่ควรมีการปล่อยอนุภาคออกไปค้นหาโอกาสที่จะเกิดการชนกันในพื้นที่นั้น ๆ เนื่องจากหากขาดซึ่งตัวควบคุม จะทำให้มีการปล่อยอนุภาคออกมาหรือน้อยกว่าความต้องการ โดย

รายละเอียดขั้นตอนวิธีการทำงานได้นำเสนอในรูปแบบของโฟลชาร์ท (flow chart) ในรูปที่ 3-2 และรูปที่ 3-3

3.1.1 การแบ่งพื้นที่บนวัตถุ



รูปที่ 3-2 ขั้นตอนการแบ่งวัตถุออกเป็นส่วน ๆ

3.1.2 การกำหนดตำแหน่งและรัศมีการทำงานของตัวควบคุมอนุภาค



รูปที่ 3-3 ขั้นตอนการกำหนดตำแหน่งและรัศมีการทำงานของตัวควบคุมอนุภาค

ศูนย์วิทยพัทยากร จุฬาลงกรณ์มหาวิทยาลัย

ในขั้นตอนนี้การกำหนดระยะรัศมีการทำงานของตัวควบคุมอนุภาคนั้น สามารถนำความยืดหยุ่นของตัววัตถุมาใช้พิจารณารวมด้วยได้ ซึ่งจะช่วยให้สามารถกำหนดรัศมีการทำงานให้มีขนาดเพียงพอต่อการยึดรูปร่างของวัตถุที่อาจเกิดขึ้นได้ระหว่างการทำงาน

3.2 การตรวจจับการชนกันขั้นคัตคู่วัตถุและคู่พื้นที่

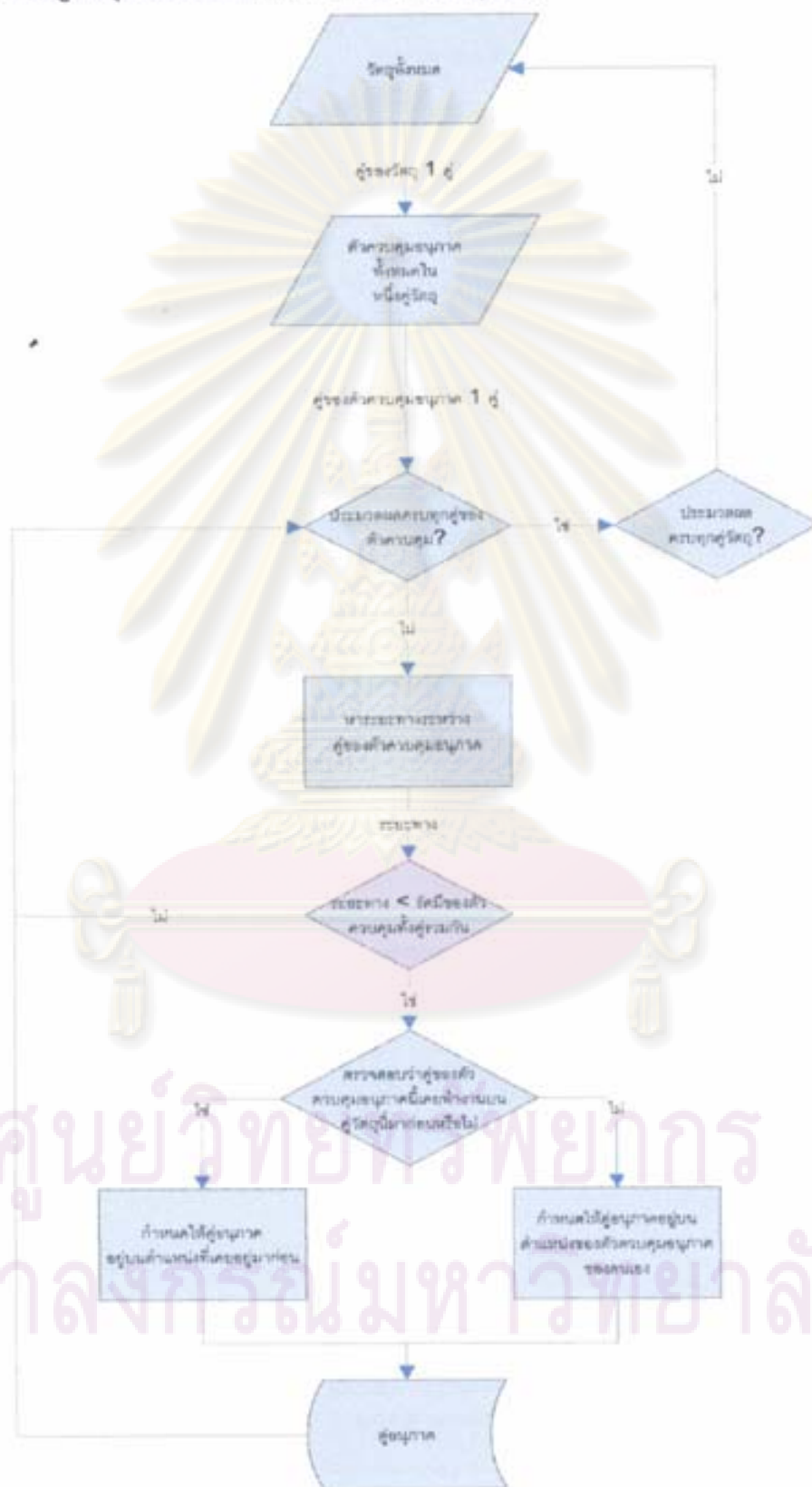
สำหรับการค้นหาคู่พื้นที่ที่มีโอกาสเกิดการชนกันนั้น สิ่งที่ทำเป็นครั้งแรกคือการตรวจสอบแนวกว้างหรือการคัตคู่วัตถุที่มีโอกาสเกิดการชนกัน ซึ่งในงานวิจัยนี้ได้อาศัยตำแหน่งของตัวควบคุมอนุภาคเป็นตัวช่วยในการคัตคู่วัตถุ ซึ่งหลังจากได้คู่วัตถุที่คัตมาแล้วจึงจะปล่อยอนุภาคออกไปเพื่อค้นหาคู่พื้นที่ที่มีโอกาสเกิดการชนกันเฉพาะคู่พื้นที่นั้น ๆ ซึ่งอนุภาคของแต่ละคู่วัตถุจะเป็นอิสระต่อกันไม่มีความเกี่ยวข้องกัน ดังรูปที่ 3-4 รูปล่างซึ่งตัวควบคุมที่มีการปล่อยอนุภาคสีเขียวจะมีสีชมพูและตัวควบคุมอนุภาคที่ไม่ได้ทำงานจะมีสีแดง โดยรายละเอียดขั้นตอนวิธีการทำงานได้นำเสนอในรูปของโฟลชาร์ทในรูปที่ 3-5 รูปที่ 3-6 รูปที่ 3-7 ตามลำดับ



รูปที่ 3-4 รูปบนเป็นตัวอย่างของวัตถุที่ใช้ทดลอง และรูปล่างคือการตรวจหาการชนกันโดยใช้ตัวควบคุมอนุภาคและอนุภาค

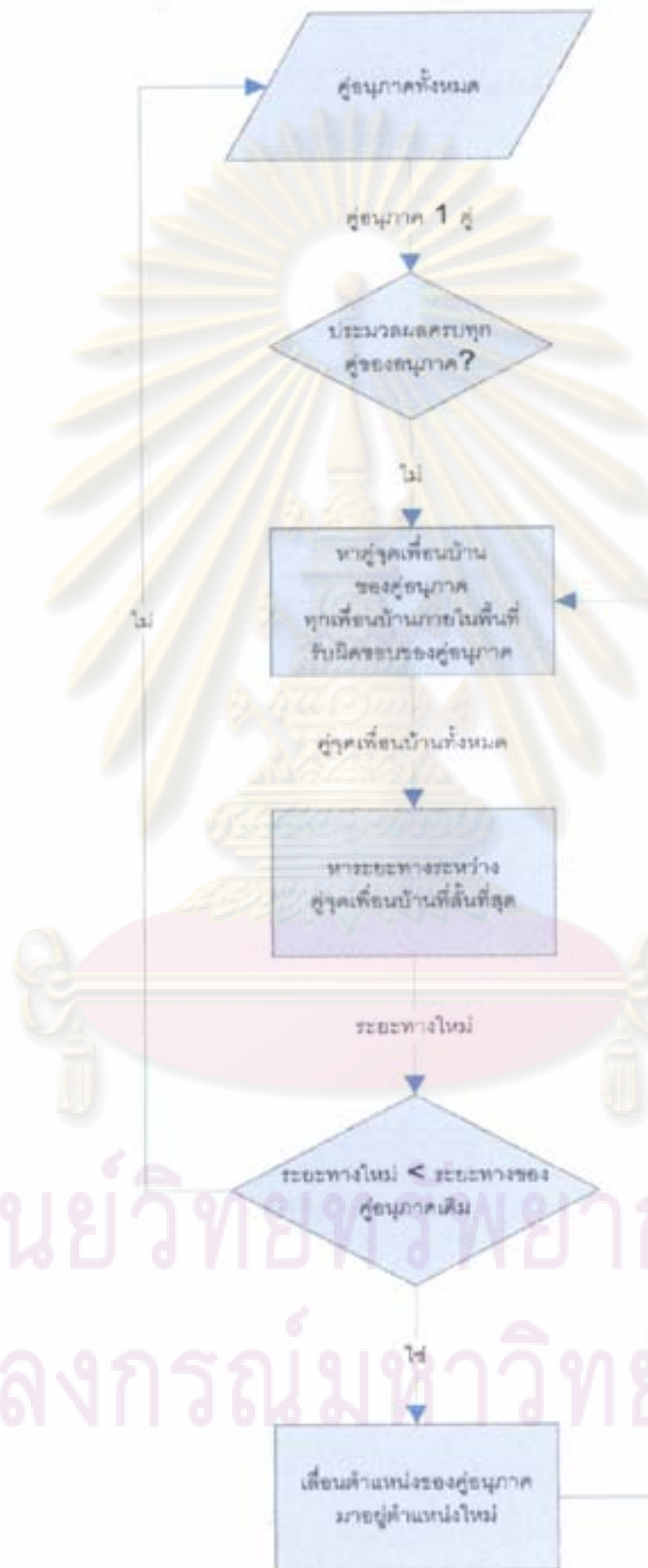
ศูนย์วิจัยทรัพยากรธรณีวิทยา
จุฬาลงกรณ์มหาวิทยาลัย

3.2.1 การคัดเลือกผู้วัดและกำหนดตำแหน่งของอนุภาค



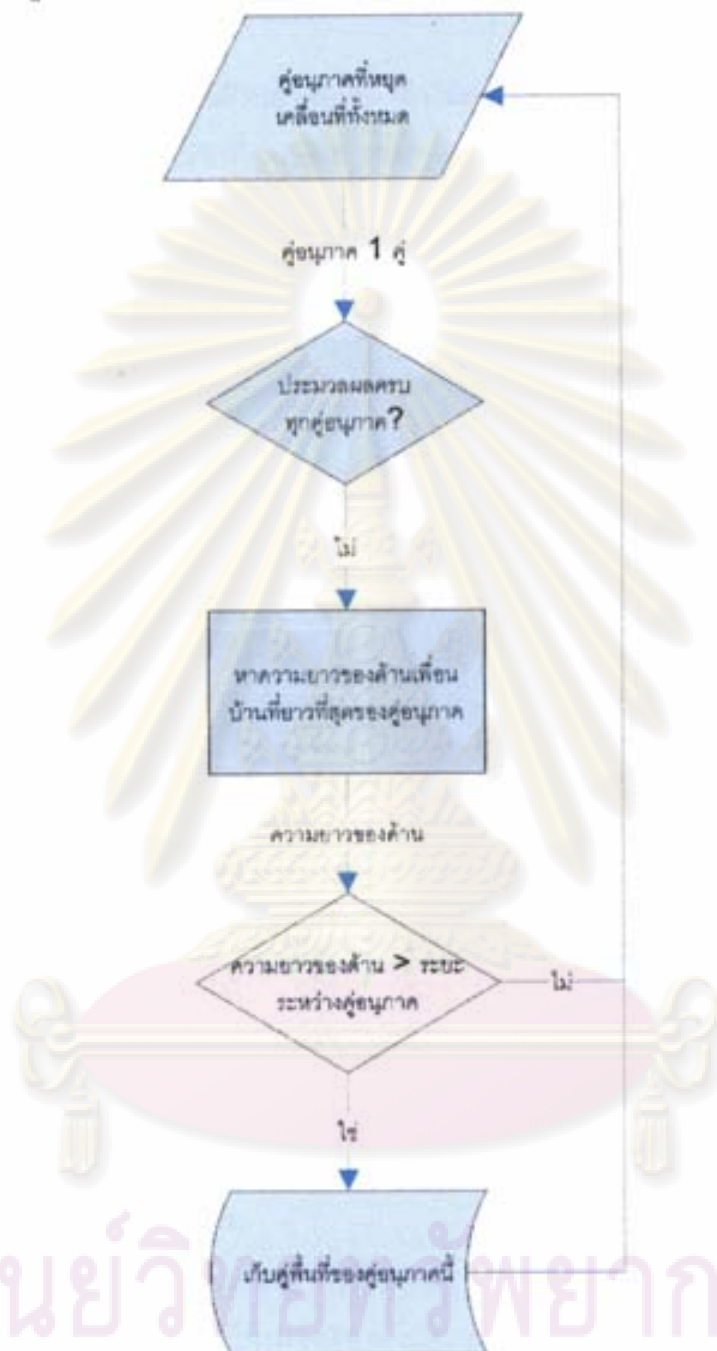
รูปที่ 3-5 ขั้นตอนการกำหนดตำแหน่งของอนุภาค

3.2.2 การเคลื่อนที่ของอนุภาค



รูปที่ 3-6 ขั้นตอนการเคลื่อนที่ของอนุภาค

3.2.3 การเลือกคู่พื้นที่ย่อยที่มีโอกาสเกิดการชนกัน

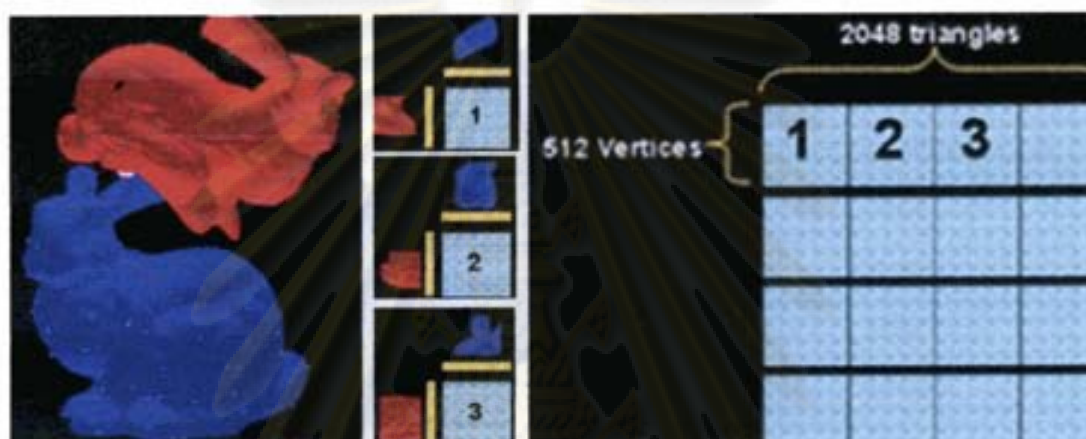


รูปที่ 3-7 ขั้นตอนการเลือกคู่พื้นที่ที่มีโอกาสเกิดการชนกัน

ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.3 การตรวจจับการชนกันชั้นละเอียด

เนื่องมาจากข้อจำกัดทางด้านฮาร์ดแวร์ ทำให้ความละเอียดของจำนวนช่องเพื่อที่จะใช้ในการประมวลผล สามารถมีได้มากที่สุดเพียง 2048×2048 ช่อง นั้นหมายความว่าหากทำการตรวจสอบการชนกันของคู่พื้นที่ที่แต่ละพื้นที่ที่มีจำนวนรูปทรงสามเหลี่ยมไม่เกิน 512 รูปทรงสามเหลี่ยม จะสามารถประมวลผลได้พร้อม ๆ กันสูงสุดเพียง 16 คู่พื้นที่เท่านั้นดังรูปที่ 3-8



รูปที่ 3-8 การประมวลผล 3 คู่พื้นที่ที่มีโอกาสชนกันพร้อม ๆ กันในช่องประมวลผลขนาด 2048×2048 ช่อง

แต่ถ้าหากต้องการให้สามารถประมวลผลคู่พื้นที่พร้อม ๆ กันได้มากกว่านี้ ก็จำเป็นต้องลดจำนวนรูปทรงสามเหลี่ยมต่อพื้นที่ลง แต่พอทำแบบนี้ภาระการทำงานก็จะไปตกอยู่กับหน่วยประมวลผลกลาง เนื่องจากการประมวลผลในส่วนของจำนวนคู่อนุภาคที่เพิ่มขึ้นตามจำนวนพื้นที่ที่มากขึ้น นอกจากนี้หากกำหนดจำนวนรูปทรงสามเหลี่ยมต่อพื้นที่เล็กจนเกินไป ผลลัพธ์ของการชนกันต่อพื้นที่ก็จะเพิ่มจำนวนมากขึ้นตามไปด้วย ทำให้ตอนที่ต้องอ่านข้อมูลกลับมาฝั่งหน่วยประมวลผลกลางไม่สามารถลดขนาดของข้อมูลที่ต้องอ่านกลับมาได้มากนัก และจะส่งผลต่อประสิทธิภาพการทำงานโดยรวม

ซึ่งจากการทดลองเพื่อค้นหาเวลาที่ใช้ในการอ่านข้อมูลกลับมา พบว่าข้อมูลยังมีขนาดเล็ก เวลาที่ใช้ในการอ่านข้อมูลกลับมาก็จะลดลงตามไปด้วยดังตารางที่ 3-1

จำนวนช่องที่ใช้ประมวลผล	เวลาในการอ่านข้อมูลกลับ (ms)
2048×2048	46
1024×1024	12
512×512	3.3
256×256	1.2

128 x 128	0.75
64 x 64	0.5
32 x 32	0.42

ตารางที่ 3-1 ผลการทดลองหาเวลาที่ใช้ในการอ่านข้อมูลขนาดต่าง ๆ จากหน่วยประมวลผลกราฟิกกลับมากสู่นหน่วยประมวลผลกลาง

และจากการทดลองกำหนดจำนวนรูปทรงสามเหลี่ยมต่อพื้นที่ขนาดต่าง ๆ ทำให้พบว่าขนาดข้อมูลที่ต้องอ่านกลับอย่างน้อยที่สุดของแต่ละจำนวนรูปทรงสามเหลี่ยมต่อพื้นที่ มีดังตารางที่ 3-2 ซึ่งจะอธิบายถึงวิธีการลดขนาดข้อมูลก่อนการอ่านกลับในหัวข้อ 3.3.4

จำนวนรูปทรงสามเหลี่ยมต่อพื้นที่	จำนวนคู่พื้นที่ที่ประมวลผลได้	จำนวนช่องข้อมูลที่ต้องอ่านกลับ
2048	1	4096
1024	4	8192
512	16	16384
256	64	32768
128	128	65536

ตารางที่ 3-2 จำนวนช่องข้อมูลที่ต้องอ่านกลับของจำนวนรูปทรงสามเหลี่ยมต่อพื้นที่ขนาดต่าง ๆ

จากผลการทดลองจากตารางที่ 3-1 และตารางที่ 3-2 ทำให้เห็นว่าถ้าให้จำนวนรูปทรงสามเหลี่ยมต่อพื้นที่มีขนาดไม่เกิน 512 แล้ว จะพบว่าเวลาที่ใช้ในการอ่านข้อมูลกลับจะน้อยกว่าหนึ่งมิลลิวินาที และเนื่องจากจุดยอดประสงค์ของการตรวจจบบั๊กการชนของงานวิจัยนี้ จะเน้นไปที่ความละเอียดของวัตถุมากกว่าจำนวนของวัตถุทำให้พื้นที่ผิวสัมผัสมีไม่มากนัก กล่าวคือหากเกิน 16 คู่พื้นที่ก็เกินไม่มาก และ คู่พื้นที่ที่เกินสามารถส่งไปประมวลผลที่หน่วยประมวลผลกลางแทนได้ด้วยความเร็วที่พอยอมรับได้

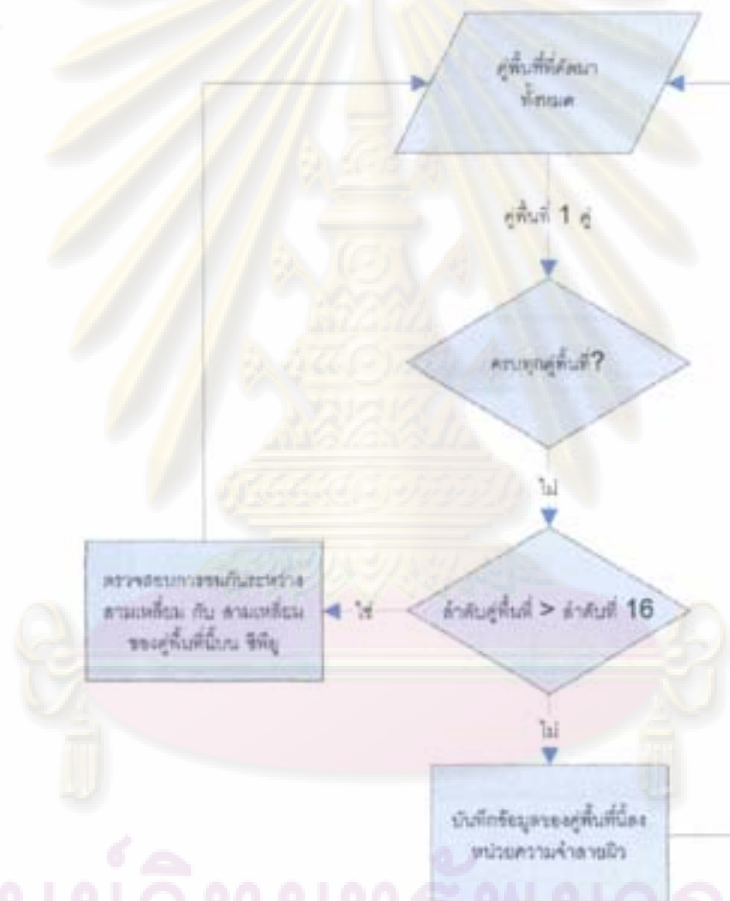
ในขณะที่หากกำหนดให้จำนวนรูปทรงสามเหลี่ยมต่อพื้นที่สูงกว่า 512 เป็น 1024 หรือ 2048 นั้นจำนวนคู่ที่จะประมวลผลจะลดลงไปมาก หากมีคู่ที่เกินมา การส่งไปประมวลผลยังหน่วยประมวลผลกลางจะใช้เวลามากจนยอมรับไม่ได้เนื่องจากคู่พื้นที่ที่มีขนาดใหญ่ หรือถ้าหากกำหนดให้จำนวนรูปทรงสามเหลี่ยมต่อพื้นที่มีขนาดเล็กลงเป็น 256 หรือ 128 ก็จะทำให้จำนวนคู่อนุภาคเพิ่มขึ้นถึง สองและสี่เท่าหากวัตถุที่ใช้มีความละเอียดสูง และเนื่องจากการประมวลผลของการเคลื่อนที่ของแต่ละคู่อนุภาคยังทำอยู่บนหน่วยประมวลผลกลาง จึงทำให้ใช้เวลาในการประมวลเพิ่มขึ้นตามไปด้วย

จากที่กล่าวมาทั้งหมด จึงเป็นสาเหตุที่งานวิจัยนี้เลือกใช้จำนวนรูปทรงสามเหลี่ยมต่อพื้นที่ไม่เกิน 512 รูปทรงสามเหลี่ยม ด้วยเพราะนำไปใช้กับวัตถุที่เน้นความละเอียด ไม่ได้เน้นที่จำนวน และ ข้อจำกัดทางด้านฮาร์ดแวร์ที่จำนวนช่องการประมวลผลควรเป็นเลขที่มาจากการยกกำลังสอง

3.3.1 การเลือกหน่วยประมวลผลให้แก่แต่ละคู่พื้นที่

จากข้อจำกัดของจำนวนคู่ที่สามารถประมวลผลได้พร้อม ๆ กัน ทำให้ต้องมีการเลือกว่าคู่พื้นที่ใดทำงานบนหน่วยประมวลผลกราฟิก และคู่พื้นที่ใดทำงานบนหน่วยประมวลผลกลางดังรูปที่

3-9



รูปที่ 3-9 ขั้นตอนการเลือกหน่วยประมวลผลของแต่ละคู่พื้นที่

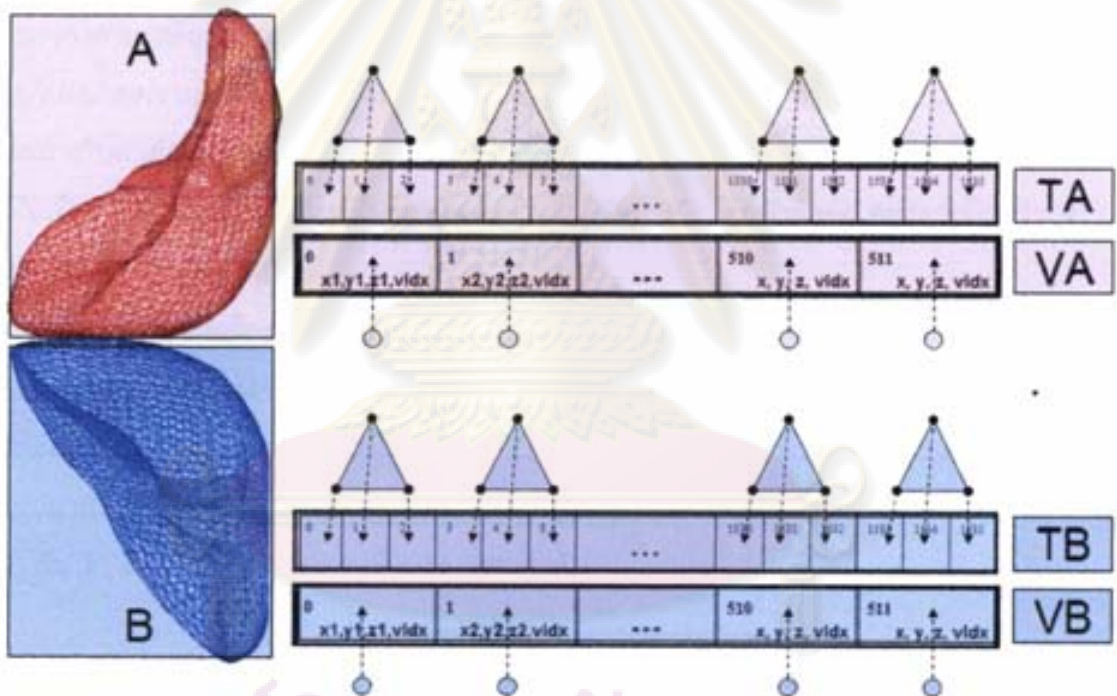
3.3.2 การบันทึกข้อมูลของคู่พื้นที่ลงหน่วยความจำถาวร

เนื่องจากขอบเขตของงานวิจัยนี้จะทำการตรวจจับการชนกันระหว่าง จุดยอดและหน้าของรูปทรงสามเหลี่ยม ข้อมูลที่จำเป็นต้องส่งไปเก็บไว้ที่หน่วยความจำถาวรหนึ่งมิติ จึงแบ่งออกเป็นข้อมูลของรูปทรงสามเหลี่ยมทั้งหมดของแต่ละพื้นที่ และข้อมูลจุดยอดทั้งหมดของแต่ละพื้นที่ โดย

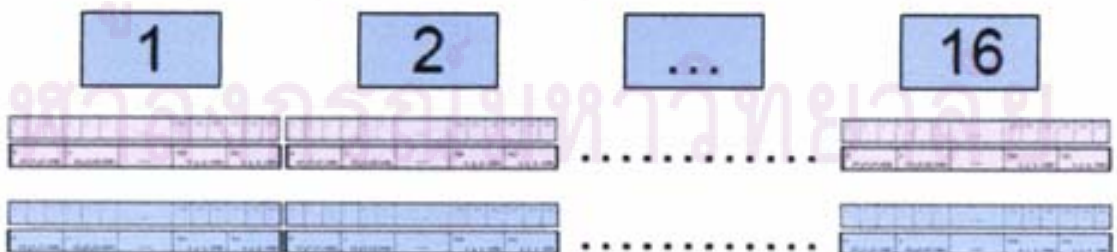
มีการจัดเก็บในลักษณะดังรูปที่ 3-10 โดยตำแหน่งของแต่ละจุดยอดจะโดนจัดเก็บไว้ในเวกเตอร์สี มิติของแต่ละช่อง ซึ่งมีมิติที่สี่นั้นจะเก็บค่าตัวชี้ของจุดยอด (Vidx: Vertex Index)

ในส่วนของรูปทรงสามเหลี่ยมนั้น เนื่องจากหนึ่งช่องของหน่วยความจำลายผิวเก็บข้อมูลที่เป็นเวกเตอร์สีมิติเท่านั้น แต่รูปทรงสามเหลี่ยมมีข้อมูลของจุดยอดที่ต้องใช้ถึงสามจุดยอด ดังนั้นจึงต้องใช้จำนวนช่องเพื่อบันทึกข้อมูลมากกว่าการบันทึกจุดยอดถึงสามเท่า และเช่นเดียวกับการบันทึกจุดยอด มิติที่ สี่ ของแต่ละช่องจะทำการจัดเก็บตัวชี้ของรูปทรงสามเหลี่ยม (Tidx: Triangle Index)

โดยเมื่อมีการบันทึกข้อมูลมากกว่า 1 คู่พื้นที่ หน่วยความจำลายผิวหนึ่งมิติแต่ละลายผิวก็จะมีขนาดยาวมากขึ้นจนสูงสุดอยู่ที่ 16 คู่พื้นที่ดังรูปที่ 3-11



รูปที่ 3-10 การบันทึกข้อมูลของคู่พื้นที่ AB ลงในหน่วยความจำลายผิว TA, TB, VA, VB



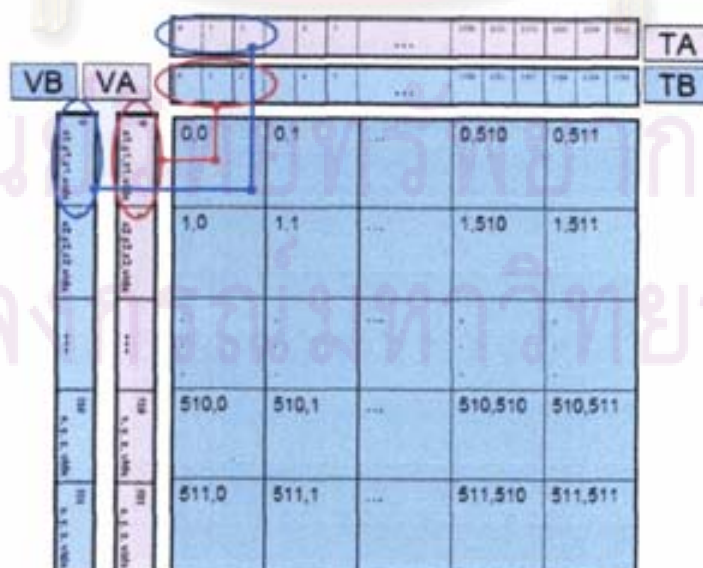
รูปที่ 3-11 หน่วยความจำลายผิวหนึ่งมิติที่บันทึกข้อมูลของ 16 คู่พื้นที่

3.3.3 การเข้าถึงข้อมูลและการตรวจสอบการชนกันในแต่ละช่องการประมวลผล

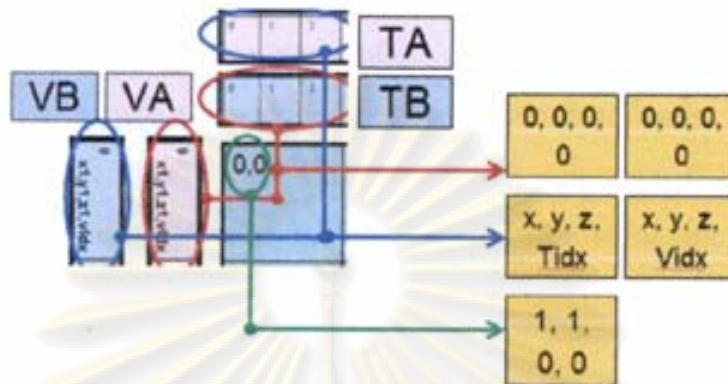
ในแต่ละช่องประมวลผลนั้น จะทำการตรวจสอบว่าช่องนั้น ๆ ต้องเข้าถึงข้อมูลของคู่พื้นที่ขาเข้า (input) ไทในลายผิวที่ได้มากจากขั้นตอนที่ 3.2.2 โดยใช้เลขตัวชี้ที่ประจำอยู่ของแต่ละช่องซึ่งให้ตัวชี้มิติที่หนึ่ง ในการเข้าถึงรูปทรงสามเหลี่ยม และให้ตัวชี้มิติที่สองในการเข้าถึงข้อมูลจุดยอด ดังรูปที่ 3-12 จากนั้นจึงเป็น การตรวจจับการชนกันระหว่างจุดยอดและรูปทรงสามเหลี่ยม (ray-triangle intersection) [4] ซึ่งใช้ค่าระยะทางที่ห่างกันระหว่างคู่จุดของวัตถุที่น้อยที่สุด (Minimum separate distance) D เป็นค่าตัดตัดสินใจว่าชนหรือไม่

จากนั้นจึงนำผลคู่จุดที่ระยะทางน้อยกว่าค่า D ที่ได้ มาบันทึกใหม่ในหน่วยความจำลายผิวเป็นข้อมูลขาออก (output) ทั้งหมดห้าลายผิว โดยสองลายผิวแรกเก็บค่าตำแหน่งของคู่จุดระหว่างสามเหลี่ยมและจุดยอดพร้อมทั้งตัวชี้ของแต่ละรูปทรงสามเหลี่ยมและของแต่ละจุดยอด ที่อยู่ในหน่วยความจำลายผิว T_A และ V_B โดยตั้งชื่อลายผิวใหม่นี้ว่า T_{A2} , V_{B2} และ สองลายผิวต่อมาเก็บค่าตำแหน่งของคู่จุดระหว่างสามเหลี่ยมและจุดยอดพร้อมทั้งตัวชี้ที่ทำให้เป็นเลขติดลบ เพื่อเป็นการบ่งชี้ว่าเป็นผลลัพธ์ที่มาจากลายผิว T_B และ V_A โดยตั้งชื่อทั้ง สองลายผิวใหม่นี้ว่า T_{B2} , V_{A2}

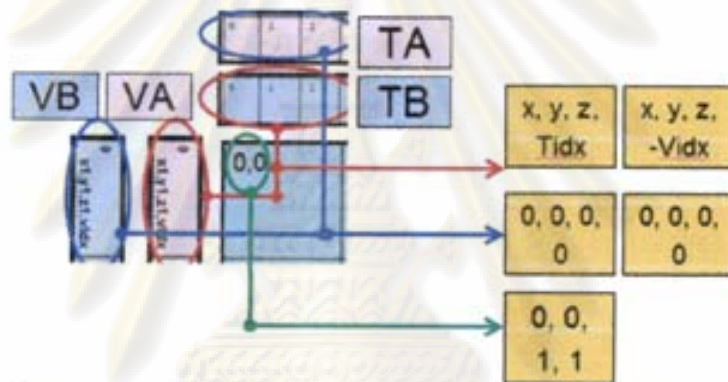
สำหรับลายผิวสุดท้ายตั้งชื่อว่า IDX จะเป็นการเก็บค่าตัวชี้ตำแหน่งของแต่ละช่องประมวลผลที่คู่จุดผ่านค่า D ซึ่งตัวชี้นี้จะบวกค่าเพิ่มอีกหนึ่งเพื่อไม่ให้ซ้ำกับเลขศูนย์ที่จะบันทึกลงไปเมื่อคู่จุดในช่องประมวลผลนั้นไม่ผ่านค่า D โดย สองมิติแรกของเวกเตอร์เก็บค่าตัวชี้ของช่องประมวลผลที่ใช้ลายผิว T_A กับ V_B และ สองมิติสุดท้ายเก็บค่าตัวชี้ของช่องประมวลผลที่ใช้ลายผิว T_B กับ V_A ดังรูปที่ 3-13 และ รูปที่ 3-14 ตามลำดับ



รูปที่ 3-12 การเข้าถึงข้อมูลในหน่วยความจำสายมิวของแต่ละช่องประมวลผลเพื่อใช้ในการตรวจสอบการชนกัน



รูปที่ 3-13 การบันทึกจุดของสายมิว TA และ VB ที่ผ่านค่า 0 ลงหน่วยความจำสายมิวขาออก



รูปที่ 3-14 การบันทึกจุดของสายมิว TB และ VA ที่ผ่านค่า 0 ลงหน่วยความจำสายมิวขาออก

3.3.4 การลดขนาดของข้อมูลก่อนการอ่านกลับ

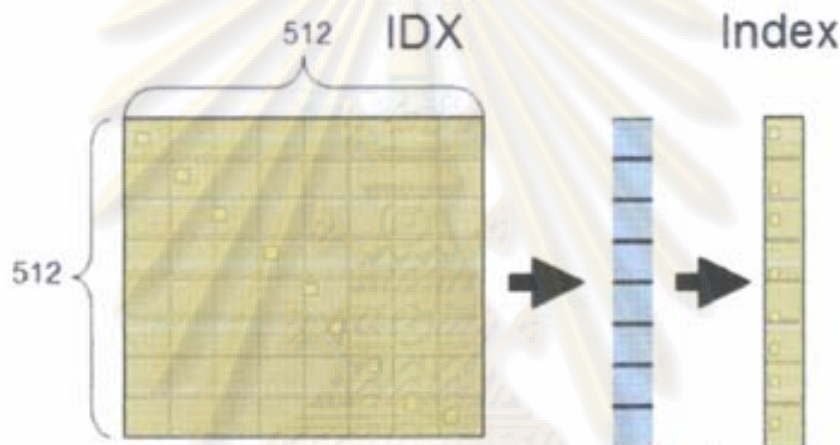
เนื่องจากหนึ่งจุดยอดใด ๆ สามารถจะชนกับหน้ารูปทรงสามเหลี่ยมได้เพียงหน้าเดียว ดังนั้นในแต่ละสายมิวที่ได้มาจากขั้นตอนที่ 3.3.3 นั้น ในแต่ละแถวของแต่ละคู่พื้นที่จึงสามารถมีคำตอบของคู่จุดได้เพียงช่องเดียวดังช่องสีส้มในรูปที่ 3-15

		TA			
		TB			
VB	VA	0,1	0,510	0,511	
0,1	0,510	1,0	1,510	1,511	
0,1	0,511	0,100	0,101	0,1010	0,1011
0,1	0,511	0,110	0,111		0,1111

รูปที่ 3-15 ตัวอย่างผลลัพธ์ของคู่จุดที่ผ่านค่า 0 ในช่องสีส้มของคู่พื้นที่หนึ่ง ๆ

ดังนั้นหากใช้คุณสมบัติดังกล่าวก็จะสามารถลดขนาดของข้อมูลที่จะต้องอ่านกลับลงไปได้มาก กล่าวคือ จากที่ต้องอ่านข้อมูลกลับ 512×512 ช่องในแต่ละคู่พื้นที่ ก็สามารถจะลดเหลือ 512×1 ช่องในแต่ละคู่พื้นที่ของแต่ละลายผิว

ดังนั้นจึงได้กำหนดช่องประมวลผลขนาด 2048×4 ช่องขึ้นมาโดยข้อมูลขาเข้าที่ใช้ในการคำนวณครั้งนี้ก็คือลายผิว IDX ที่เก็บตัวชี้ไปยังช่องประมวลผลที่ผ่านค่า D ของอีก 4 ลายผิว เพื่อให้ได้ข้อมูลขาออกที่มีขนาดเล็กลงเก็บไว้ในลายผิว $Index$ โดยในแต่ละช่องประมวลผลนั้นจะทำการเข้าถึงข้อมูลในแต่ละแถวของคู่พื้นที่ในลายผิว IDX เพื่อที่จะทำการบวกเวกเตอร์ทั้งแถวเข้าด้วยกันดังรูปที่ 3-16



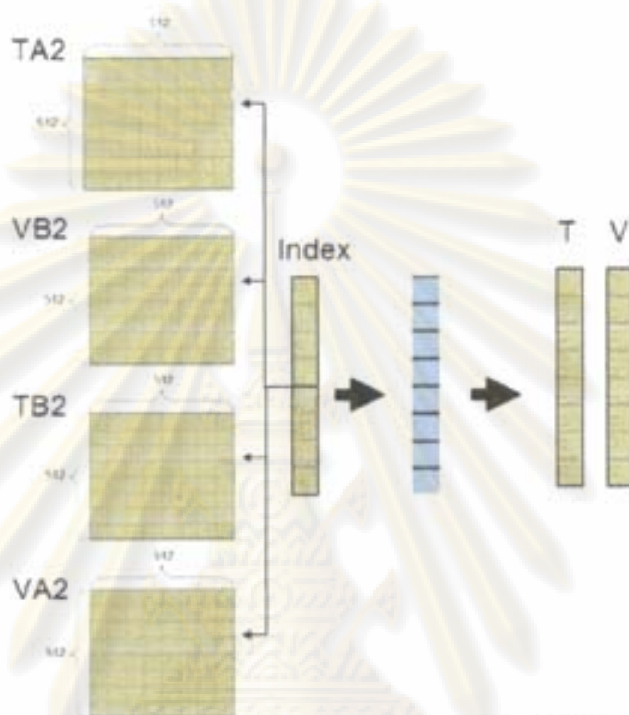
รูปที่ 3-16 การหาตัวชี้ข้อมูลแต่ละแถวของลายผิวที่เก็บตัวชี้

ต่อมาจึงสร้างช่องประมวลผลขนาด 2048×4 ช่องอีกครั้ง โดยข้อมูลขาเข้าเป็นลายผิวที่เก็บตัวชี้ที่ลดขนาดแล้วกับลายผิวที่เก็บจุดทั้ง 4 ลายผิวคือ $T_{A2}, V_{B2}, T_{B2}, V_{A2}$ เพื่อทำการประมวลผลลดขนาดข้อมูลให้เหลือเป็นข้อมูลขาออกขนาด 2048×4 ช่อง จำนวน 2 ลายผิว โดยลายผิวแรกเก็บค่าของจุดยอดที่อยู่บนรูปทรงสามเหลี่ยมตั้งชื่อว่าลายผิว T และลายผิวที่สองเก็บค่าของจุดยอดที่เป็นเป็นคู่กับจุดยอดที่แล้วตั้งชื่อว่าลายผิว V ซึ่งผลลัพธ์ที่ได้เป็นดังตัวอย่างการลดขนาดข้อมูลคู่พื้นที่หนึ่ง ๆ ในรูปที่ 3-17 และภาพผลลัพธ์การลดขนาดข้อมูลจาก 2048×2048 ช่องเหลือ 2048×4 ช่อง จำนวนสองลายผิวในรูปที่ 3-18

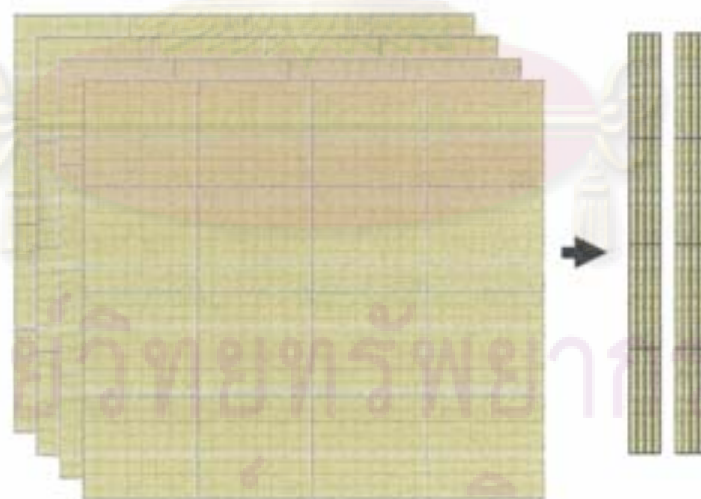
3.3.5 การอ่านข้อมูลกลับและการนำไปใช้

ในขั้นตอนนี้จะเป็นการการอ่านข้อมูลผลลัพธ์จากลายผิวที่ได้มาจากขั้นตอนที่ 3.3.4 เพื่อนำไปเป็นเป็นข้อมูลขาเข้าของขั้นตอนการตอบสนองการชนกัน โดยทำการเช็คค่าผลลัพธ์ที่ตั้งออกมาจากแต่ละช่องของลายผิวทั้งสองลายผิว ซึ่งหากผลลัพธ์การรวมกันของแต่ละมิติของเวกเตอร์ในช่องนั้น ๆ ไม่เท่ากับศูนย์ ก็ให้นำค่าผลลัพธ์ซึ่งเป็นจุดที่ได้มาทำการหาทิศทาง

ระยะทางที่ห่างระหว่างกัน และใช้เลขตัวชี้ที่อยู่ในมิติที่สี่ของเวกเตอร์จากช่องนั้น ๆ เป็นตัวระบุว่าจุดนี้เกิดขึ้นบนรูปทรงสามเหลี่ยมใด ซึ่งทั้งสี่ข้อมูลคือ ทิศทาง ระยะห่าง จุดจุด และ รูปทรงสามเหลี่ยม เป็นข้อมูลขาเข้าของขั้นตอนการตอบสนองของการชนกันนั่นเอง



รูปที่ 3-17 ขั้นตอนการลดขนาดข้อมูลก่อนอ่านกลับในหนึ่งคู่พื้นที่



รูปที่ 3-18 ภาพก่อนหลังการลดขนาดข้อมูลจาก 2048 x 2048 ช่องเหลือ 2048 x 2 ช่องจำนวน 2 ลายสี

การวิเคราะห์ค่าความซับซ้อนของวิธีที่เสนอ

กำหนดให้ m คือ จำนวนของวัตถุ

n คือ จำนวนของรูปทรงสามเหลี่ยมในแต่ละวัตถุ

k คือ จำนวนของตัวควบคุมอนุภาคหรือเท่ากับ $n/512$

m' คือ จำนวนของคู่พื้นที่ที่มีโอกาสเกิดการชนกัน หรือเท่ากับ m^2k^2

ในขั้นตอนการเตรียมการประมวลก่อนตรวจจับการชนกันนั้นจะเห็นได้ว่า จะแบ่งออกเป็นสองช่วงคือช่วงสร้างวัตถุกล่องสี่เหลี่ยมครอบรูปทรงสามเหลี่ยมทุกรูปซึ่งมีค่าความซับซ้อนเป็น $\Theta(mn)$ และช่วงการแบ่งออกเป็นส่วน ๆ ซึ่งจะต้องแบ่งเป็นจำนวน $\log n$ ครั้งในแต่ละ m วัตถุ และในแต่ละครั้งก่อนแบ่งก็จะต้องมีการเรียงลำดับ ค่ามุมบนและมุมล่างสุดของทุกกล่องซึ่งทำให้สามารถคิดค่าความซับซ้อนได้เป็น $\Theta(m \cdot \log n \cdot n \log n)$ เพราะฉะนั้นค่าความซับซ้อนในขั้นตอนนี้จึงเท่ากับ $\Theta(mn \log^2 n)$

ในขั้นตอนการตรวจจับการชนกันในช่วงขั้นตอนการตัดคู่วัตถุจะเป็นการไล่จับคู่เพื่อให้ได้ครบทุกคู่ของวัตถุทั้งหมดจึงมีค่าความซับซ้อนเท่ากับ $\Theta(m^2)$ ซึ่งในแต่ละคู่วัตถุที่ได้ก็จะมีตัวควบคุมอนุภาคอยู่ซึ่งก็จะนำมาไล่จับคู่เพื่อให้ได้คู่ของตัวควบคุมอนุภาคทั้งหมดของคู่วัตถุนั้นๆ จึงได้ค่าความซับซ้อนเป็น $\Theta(m^2k^2)$ จากนั้นจะได้จำนวนคู่อนุภาคทั้งหมดที่ปล่อยออกมาจากตัวควบคุมอนุภาคที่ทำงานซึ่งจำนวนมากที่สุดในทางทฤษฎีก็คือ m^2k^2 คู่อนุภาค ซึ่งในแต่ละคู่อนุภาคก็ต้องมีการเคลื่อนที่ภายในพื้นที่ของตนเองก่อนหยุดนิ่งและจะเคลื่อนที่ไม่เกิน $\log_2 512$ หรือ 9 ครั้งเมื่อแต่ละจุดยอดมีเพื่อนบ้านน้อยที่สุดสองเพื่อนบ้าน ดังนั้นเมื่อคิดค่าความซับซ้อนที่รวมการเคลื่อนที่ของคู่อนุภาคแล้วจึงได้ $O(m^2k^2)$ ซึ่งผลลัพธ์ของขั้นตอนนี้ก็คือจำนวนคู่พื้นที่ที่มีโอกาสเกิดการชนกันทั้งหมด

ในขั้นตอนการตรวจจับการชนกันขั้นละเอียดนั้นจะประกอบไปด้วย ขั้นตอนการเลือกหน่วยประมวลผล ซึ่งถ้าเกินกว่า 16 คู่พื้นที่ คู่พื้นที่ที่เหลือก็ต้องทำการประมวลผลที่หน่วยประมวลกลางซึ่งคิดค่าความซับซ้อนสำหรับคู่ที่เกินมาได้เป็น $\Theta((m'-16) \cdot 512^2)$

สำหรับ 16 คู่พื้นที่ที่ไม่เกินก็จะคิดค่าความซับซ้อนในการส่งข้อมูลไปยังหน่วยประมวลกราฟิกได้เป็น $\Theta(m \cdot 512)$ และค่าความซับซ้อนในการประมวลผลแบบขนานในขั้นตอนการตรวจจับการชนกันรวมทั้งการลดขนาดของข้อมูลในทางทฤษฎีจะได้เป็น $\Theta(1)$ ในขณะที่ขั้นตอนการอ่านข้อมูลกลับและเตรียมข้อมูลให้ขั้นตอนการตอบสนองการชนกันก็จะได้ค่าความซับซ้อนเป็น $\Theta(2048 \cdot 4)$

จากการวิเคราะห์ค่าความซับซ้อนของขั้นตอนการตรวจจับการชนกันทุกขั้นตอนจะเห็นว่า ค่าความซับซ้อนจะขึ้นอยู่กับ จำนวนคู่พื้นที่ที่มีโอกาสเกิดการชนกันหรือเท่ากับ $O(m^2k^2)$ นั้นเอง ซึ่งจะดีกว่าค่าความซับซ้อนที่ไล่ตรวจสอบทีละรูปทรงสามเหลี่ยมไปเรื่อยๆ คือ $\Theta(m^2n^2)$

บทที่ 4

การทดลองและผลการทดลอง

จากที่มาของปัญหาและวัตถุประสงค์จึงต้องทำการวิเคราะห์ผลลัพธ์ที่ได้ จากวิธีที่เสนอซึ่งไม่มีการสร้างและจัดการวัตถุตัวแทนเทียบกับวิธีที่มีการสร้างและจัดการวัตถุตัวแทน โดยได้เลือกวิธีบีวีเอสแบบวิธีเอเอบีบี (AABB) จาก SOFA Framework มาเป็นวิธีที่ใช้ในการทดลอง เปรียบเทียบเทียบกับวิธีที่เสนอ

โดยการทดลองจะแบ่งออกเป็นสองส่วนหลักคือ ส่วนที่วัดประสิทธิภาพด้านความแม่นยำ และส่วนที่วัดประสิทธิภาพด้านความเร็ว โดยมีรายละเอียดต่าง ๆ รวมไปถึงเครื่องมือที่ใช้และการวิเคราะห์ผลการทดลองดังต่อไปนี้

4.1 เครื่องมือที่ใช้และปัจจัยแวดล้อม






การทดลองนี้ได้ทดลองโดยใช้เครื่อง INTEL E8200 2.6 GHz หน่วยความจำ 2048 M และการ์ดแสดงผล NVIDIA Geforce 8800GT 512M โดยพัฒนาบน SOFA Framework และใช้ Visual Studio C++ 2005 เป็นเครื่องมือพัฒนา

ซึ่งการทดลองนี้ในส่วนของการแสดงผลและส่วนของการตอบสนองการชนกัน (collision response) จะใช้เครื่องมือจาก SOFA Framework ในการประมวลผลทั้งหมด จึงทำให้ตัวแปรต้นของการทดลองจะอยู่ในขั้นตอนการตรวจสอบการชนกันเท่านั้น

4.2 วัตถุและจากที่นำมาใช้ในการทดลอง

วัตถุที่นำมาใช้ในการทดลอง จะประกอบไปด้วยวัตถุที่เป็นวัตถุที่นูนออกมา (convex) ความละเอียดต่ำ 2 อย่างคือ Box และ Sphere กับวัตถุที่เป็นวัตถุที่มีการเว้าเข้าไป (concave) โดยแบ่งเป็นความละเอียดต่ำ 4 อย่างคือ Torus Teapot Dragon-low และ Frog และความละเอียดสูง 3 อย่างคือ Bunny Dragon-high และ Buddha โดยมีรูปและรายละเอียดดังแสดงในตารางที่ 4-1

จุฬาลงกรณ์มหาวิทยาลัย

ชื่อโมเดล	จำนวนรูปทรงสามเหลี่ยม
	<p>ชื่อโมเดล: Box</p> <p>จำนวนรูปทรงสามเหลี่ยม: 192</p> <p>จำนวนจุดยอด: 150</p>
	<p>ชื่อโมเดล: Sphere</p> <p>จำนวนรูปทรงสามเหลี่ยม: 1984</p> <p>จำนวนจุดยอด: 994</p>
	<p>ชื่อโมเดล: Teapot</p> <p>จำนวนรูปทรงสามเหลี่ยม: 1056</p> <p>จำนวนจุดยอด: 619</p>
	<p>ชื่อโมเดล: Torus</p> <p>จำนวนรูปทรงสามเหลี่ยม: 1600</p> <p>จำนวนจุดยอด: 800</p>
	<p>ชื่อโมเดล: Dragon-low</p> <p>จำนวนรูปทรงสามเหลี่ยม: 2564</p> <p>จำนวนจุดยอด: 1190</p>
	<p>ชื่อโมเดล: Frog</p> <p>จำนวนรูปทรงสามเหลี่ยม: 4790</p> <p>จำนวนจุดยอด: 2397</p>

	<p>ชื่อโมเดล: Bunny</p> <p>จำนวนรูปทรงสามเหลี่ยม: 69666</p> <p>จำนวนจุดยอด: 34835</p>
	<p>ชื่อโมเดล: Buddha</p> <p>จำนวนรูปทรงสามเหลี่ยม: 100000</p> <p>จำนวนจุดยอด: 49990</p>
	<p>ชื่อโมเดล: Dragon-high</p> <p>จำนวนรูปทรงสามเหลี่ยม: 100000</p> <p>จำนวนจุดยอด: 50000</p>

ตารางที่ 4-1 วัตถุและรายละเอียดของวัตถุที่นำมาใช้ในการทดลอง

จากที่นำมาใช้ในการทดลอง จะแบ่งออกเป็น สามประเภทด้วยกันคือ ฉากที่วัตถุชนกัน แต่ไม่มีการเคลื่อนไหว ฉากที่วัตถุมีการเคลื่อนที่มาชนกันแต่ไม่มีการเปลี่ยนแปลงรูปร่าง และฉากที่วัตถุเคลื่อนที่มาชนกันและมีการเปลี่ยนแปลงรูปร่าง โดยจะเริ่มทดสอบจากฉากประเภทแรกกับวัตถุทุกแบบ เพื่อให้ได้ผลการทดลองนำมาใช้ในการตัดวัตถุที่ไม่จำเป็นต้องนำมาทดลองใช้ในฉากประเภทต่อมา

4.3 การวัดประสิทธิภาพด้านความแม่นยำ

ตัววัดของการทดลองนี้คือจำนวนจุดระหว่างวัตถุที่ใกล้กันเกินกว่าค่าที่กำหนดไว้ โดยจะทำการทดลองกับฉาก สองประเภทคือ ประเภทแรก และประเภทที่สาม ดังต่อไปนี้

4.3.1 ฉากประเภทวัตถุหยุดนิ่ง

ในขั้นตอนนี้จะทำการวัดจำนวนจุดกับฉากประเภทวัตถุหยุดนิ่ง โดยแบ่งออกเป็นสามฉาก ตามประเภทของวัตถุคือ ฉากที่ประกอบไปด้วยวัตถุขนาดความละเอียดต่ำ ฉากที่ประกอบไปด้วย

ด้วยวัตถุที่มีความละเอียดต่ำ และฉากที่ประกอบไปด้วยวัตถุที่มีความละเอียดสูง โดยกำหนดค่าระยะทางที่ห่างกันระหว่างจุดน้อยที่สุด (Minimum separate distance) $D = 0.1$ และ $D = 0.01$ ซึ่งรายละเอียดของแต่ละฉากอยู่ในรูปที่ 4-1 และผลการทดลองดังตารางที่ 4-2

	ฉากที่ 1(จุด)		ฉากที่ 2(จุด)		ฉากที่ 3(จุด)	
	$D = 0.1$	$D = 0.01$	$D = 0.1$	$D = 0.01$	$D = 0.1$	$D = 0.01$
AABB	106	14	25	2	2154	234
Our Method	106	14	25	2	1160	234

ตารางที่ 4-2 ผลการทดลองด้านความแม่นยำกับฉากวัตถุชนิดนี้

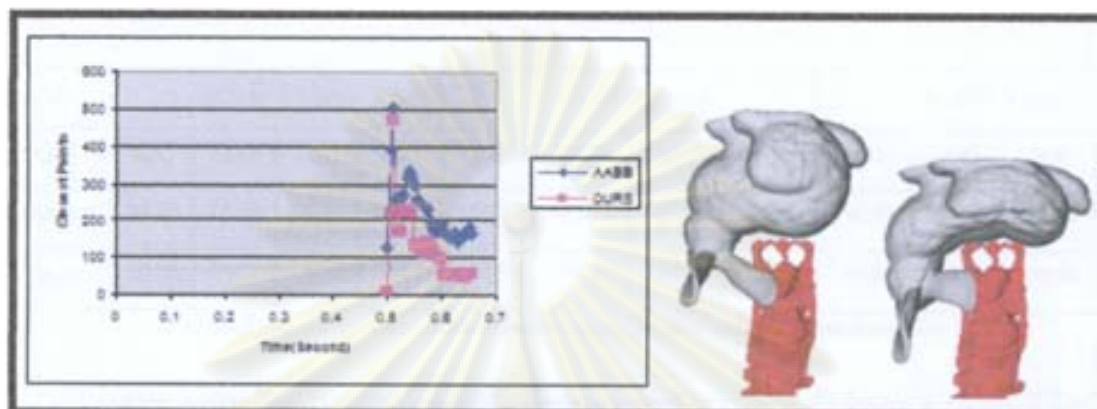


รูปที่ 4-1 รายละเอียดและจำนวนคู่พื้นที่ที่มีโอกาสชนกันของแต่ละฉากที่ใช้ทดลองในการทดลองที่ 3.1 และ 4.1

4.3.2 ฉากประเภทวัตถุเคลื่อนที่และมีการเปลี่ยนแปลงรูปร่าง

ในขั้นตอนนี้จะทำการวัดจำนวนจุด กับฉากประเภทวัตถุที่มีการเคลื่อนที่และมีการเปลี่ยนแปลงรูปร่าง โดยทดลองกับวัตถุที่มีความละเอียดสูงสองชิ้น ซึ่งกำหนดให้ค่าระยะทางที่

ห่างกันระหว่างจุดน้อยที่สุด $D = 0.05$ และ แต่ละเฟรมมีความยาว (Time step) $T = 0.005$ วินาที และเวลาในฉากรวมทุกเฟรมยาว 0.65 วินาที ซึ่งได้ผลการทดลองดังรูปที่ 4-2



รูปที่ 4-2 จำนวนคู่จุดการชนกันในแต่ละเฟรมของวิธีเจเจบีบีและวิธีที่เสนอของจากวัตถุเคลื่อนที่และเปลี่ยนแปลงรูปร่าง

4.4 การวัดประสิทธิภาพด้านความเร็ว

ตัววัดที่ใช้ในการทดลองนี้คือเวลาโดยเฉลี่ยในการค้นหาจุดจุดที่เกิดการชนกันในทุก ๆ 100 เฟรม โดยจะแบ่งออกเป็นจากประเภทต่าง ๆ ดังต่อไปนี้

4.4.1 จากประเภทวัตถุหยุดนิ่ง

ในขั้นตอนนี้จะแบ่งออกเป็นสามฉาก ตามประเภทของวัตถุคือ ฉากที่ประกอบไปด้วยวัตถุ ความละเอียดต่ำ วัตถุที่มีความละเอียดต่ำ และวัตถุที่มีความละเอียดสูง ดังรูปที่ 4-1 ซึ่งได้ผลการทดลองดังตารางที่ 4-3

	ฉากที่ 1 (ms)		ฉากที่ 2 (ms)		ฉากที่ 3 (ms)	
	$D = 0.1$	$D = 0.01$	$D = 0.1$	$D = 0.01$	$D = 0.1$	$D = 0.01$
AABB	5.73	4.51	3.54	3.01	794.01	73.00
Our Method	36.67	35.716	35.388	34.29	48.99	44.14

ตารางที่ 4-3 ผลการทดลองด้านความเร็วกับจากวัตถุหยุดนิ่ง

4.4.2 จากประเภทวัตถุเคลื่อนที่แต่ไม่มีการเปลี่ยนแปลงรูปร่าง

จากการทดลองที่ 4.4.1 จะเห็นได้ว่าวิธีนี้จะมีประสิทธิภาพเมื่อใช้กับวัตถุที่มีความละเอียดสูง ดังนั้นในขั้นตอนนี้จึงเป็นการทดลองกับวัตถุที่มีความละเอียดสูงทั้งหมด 3 ฉาก โดยแบ่งเป็น

ฉากที่หนึ่งให้มีวัตถุความละเอียดสูง สองชิ้น ฉากที่สองให้มีวัตถุความละเอียดสูง สามชิ้น และฉากที่สาม ให้มีวัตถุความละเอียดสูงสี่ชิ้น แต่ละฉากมีความยาว 2 วินาที ให้ระยะทางที่ห่างกันระหว่างคู่จุดน้อยที่สุด $D = 0.05$ และความยาวของแต่ละเฟรม $T = 0.005$ ซึ่งได้ผลการทดลองดัง

ฉาก	ฉากที่ 1 (ms)				ฉากที่ 2 (ms)				ฉากที่ 3 (ms)			
รอบ	100	200	300	400	100	200	300	400	100	200	300	400
AABB	63.30	70.42	59.47	59.31	117.57	120.74	127.25	100.59	146.43	184.16	161.89	115.09
Ours	10.58	14.27	6.38	6.51	23.88	47.25	42.29	5.70	33.90	80.42	59.95	55.29

ตารางที่ 4-4 ผลการทดลองของฉากประเภทวัตถุเคลื่อนที่แต่ไม่มีการเปลี่ยนแปลงรูปร่าง

และสามารถเขียนผลการทดลองให้อยู่รูปของกราฟ รวมถึงจำนวนของคู่พื้นที่ที่ส่งไปประมวลผลชั้นละเอียด ที่หน่วยประมวลผลกราฟิกของแต่ละฉากดังรูปที่ 4-3

4.4.3 ฉากประเภทวัตถุเคลื่อนที่และมีการเปลี่ยนแปลงรูปร่าง

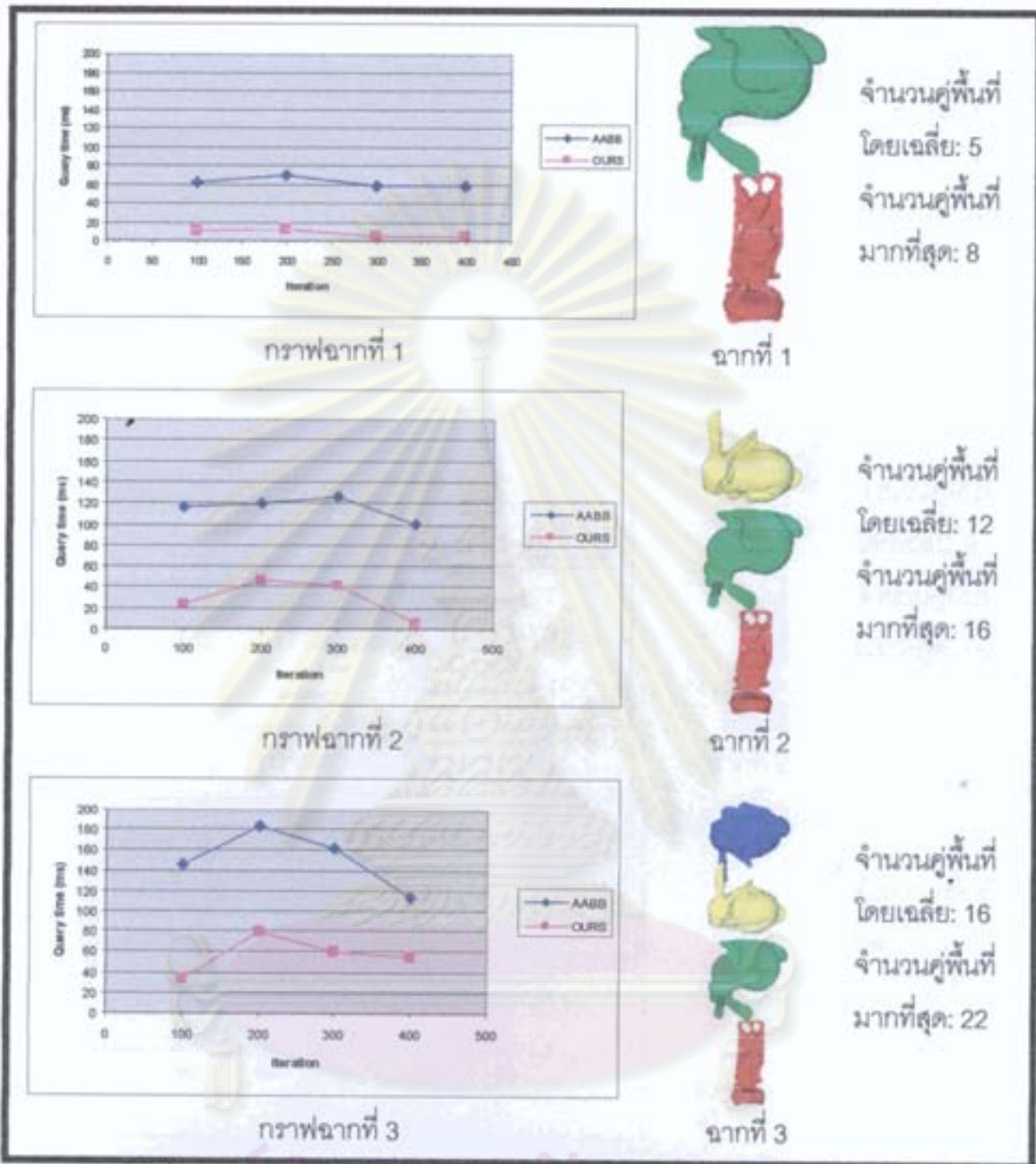
ในขั้นตอนนี้จึงเป็นการทดลองกับวัตถุเข้าความละเอียดสูงทั้งหมด สามฉากโดยแต่ละฉากประกอบไปด้วยวัตถุความละเอียดสูงอย่างละ สองชิ้น ซึ่งมีการวางในรูปแบบที่แตกต่างกัน โดยแต่ละฉากมีความยาว 2 วินาที ซึ่งได้ผลการทดลองดังตารางที่ 4-5

ฉาก	ฉากที่ 1 (ms)				ฉากที่ 2 (ms)				ฉากที่ 3 (ms)			
รอบ	100	200	300	400	100	200	300	400	100	200	300	400
AABB	63.09	103.84	89.16	59.50	73.82	118.22	140.10	101.65	81.60	107.05	99.35	74.93
Ours	10.87	42.12	40.42	7.50	7.15	42.88	44.19	27.33	35.31	70.81	47.51	31.87

ตารางที่ 4-5 ผลการทดลองด้านความเร็วกับวัตถุเคลื่อนที่และเปลี่ยนแปลงรูปร่าง

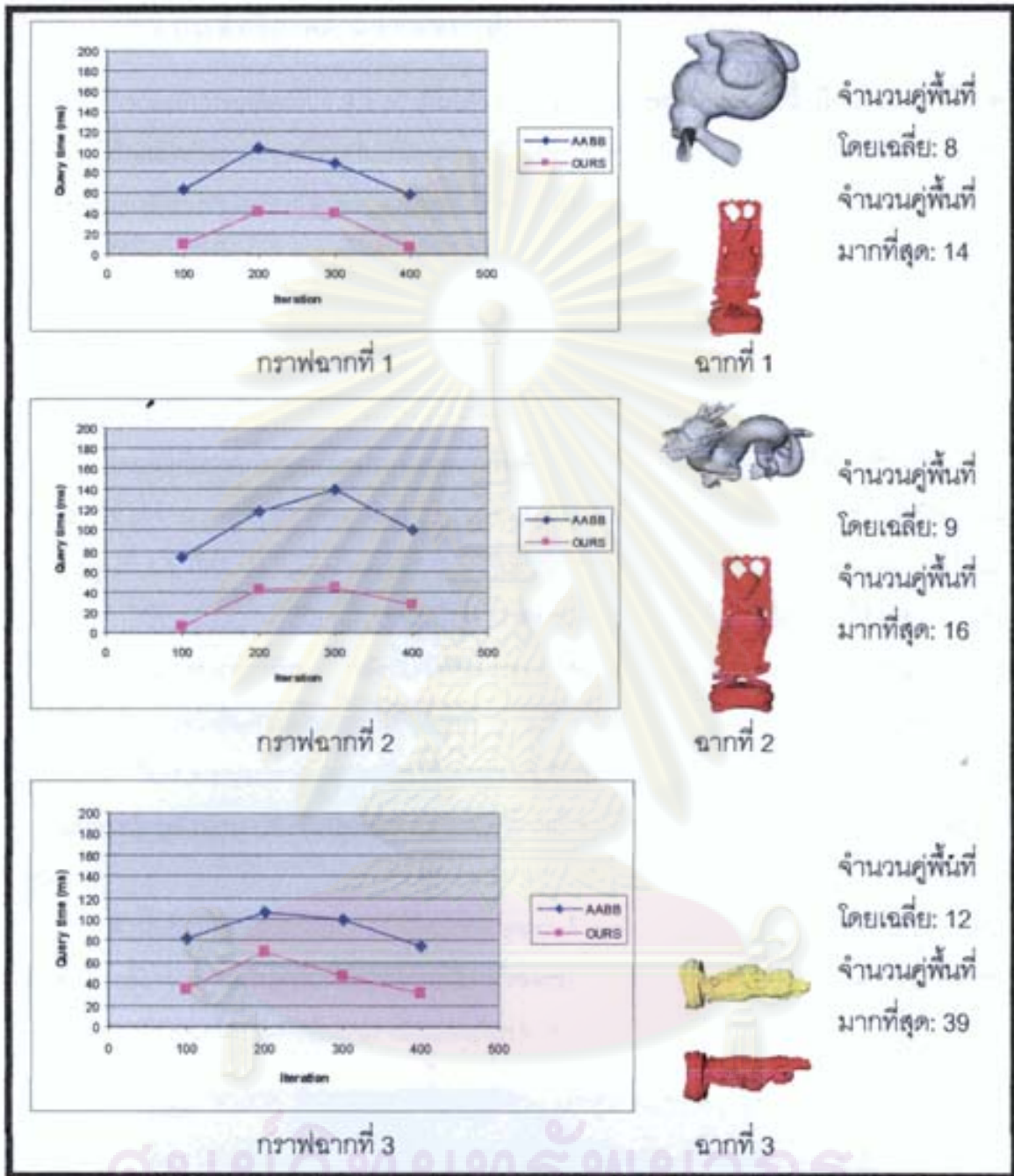
และสามารถเขียนผลการทดลองให้อยู่รูปของกราฟ รวมถึงจำนวนของคู่พื้นที่ที่ส่งไปประมวลผลชั้นละเอียด ที่หน่วยประมวลผลกราฟิกของแต่ละฉากดังรูปที่ 4-4

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4-3 ผลการทดลองในรูปของกราฟในฉากประเภทวัตถุเคลื่อนที่แต่ไม่มีการเปลี่ยนแปลงรูปร่าง

ศูนย์วิจัยโทรพียากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4-4 ผลการทดลองในรูปแบบกราฟของฉากประเภทวัตถุเคลื่อนไหวที่มีการเปลี่ยนแปลงรูปร่าง

4.5 วิเคราะห์ผลการทดลอง

จากผลการทดลองสามารถวิเคราะห์ประสิทธิภาพของวิธีที่เสนอแยกออกเป็นหัวข้อตามการทดลองดังต่อไปนี้

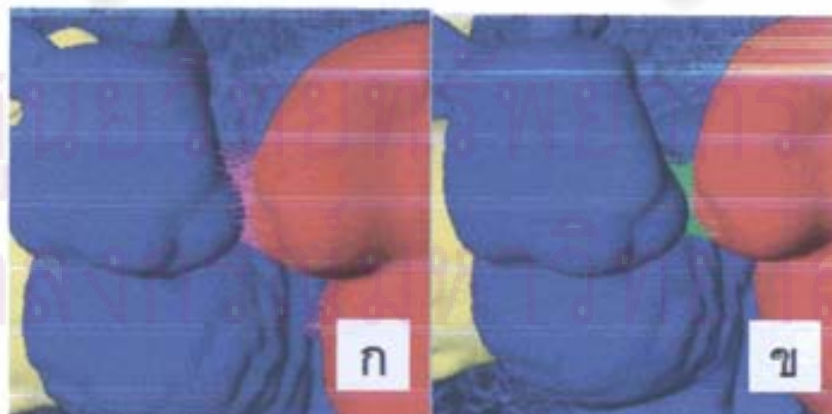
4.5.1 ประสิทธิภาพด้านความแม่นยำ

จากผลการทดลองที่ 4.3.1 จะเห็นได้ว่า ความแม่นยำของวิธีที่เสนอนี้ เมื่อเทียบกับวิธีเอเอบีบีจะ ให้ผลลัพธ์ออกมาเหมือนกันเมื่อทดลองกับวัตถุที่มีความละเอียดไม่สูงและหยาบนิ่ง ถึงแม้ว่าค่าระยะทางที่ห่างกันระหว่างคู่จุดน้อยที่สุด D จะเปลี่ยนไปก็ยังให้ผลลัพธ์แบบเดียวกัน

ในขณะที่เมื่อนำไปทดลองกับวัตถุที่มีความละเอียดสูงและหยาบนิ่ง จะพบว่าผลลัพธ์ที่ได้ นั้นมีขนาดไม่แตกต่างกันเมื่อระยะทางที่ห่างกันระหว่างคู่จุดน้อยที่สุด $D = 0.01$ และต่างกันเมื่อค่า $D = 0.1$ ที่เป็นเช่นนี้ก็เพราะ วิธีที่จะตัดสินใจในการเลือกคัดพื้นที่ที่มีโอกาสเกิดการชนกันของวิธีที่เสนอ กับวิธีเอเอบีบี นั้นมีความแตกต่างกัน

โดยวิธีที่เสนอนั้นจุดยอดประสงค์หลักคือต้องการคัดพื้นที่ที่มีโอกาสเกิดการชนกันออกมาให้ได้จำนวนคู่จุดน้อยที่สุด จึงไม่ได้นำค่า D มาช่วยในการตัดสินใจคัดเลือกพื้นที่ที่ตั้งจะเห็นได้จากรูปที่ 4-1 ที่ค่า D ไม่ส่งผลต่อจำนวนคู่พื้นที่ที่มีโอกาสชนกัน เพราะหากนำค่า D มาใช้ร่วมด้วยจะทำให้จำนวนคู่พื้นที่ที่ผ่านการคัดเลือก มากกว่าจำนวนคู่พื้นที่ที่ประมวลผลได้พร้อม ๆ กัน ในหน่วยประมวลผลกราฟิก ดังนั้นจึงใช้ความยาวของด้านเพื่อนบ้านของคู่อนุภาคของแต่ละคู่พื้นที่ช่วยในการตัดสินใจที่นำเสนอไปในบทที่ 3 แล้วจึงค่อยนำค่า D ไปใช้เป็นตัวคัดคู่จุดในคู่พื้นที่นั้น ๆ ในหน่วยประมวลผลกราฟิก ซึ่งจะต่างกับวิธีเอเอบีบีที่จะนำค่า D ไปบวกเข้าเพื่อขยายขนาดของวัตถุตัวแทน แล้วจึงตัดสินใจคัดว่ามีโอกาสเกิดการชนกันหรือไม่ทั้งตอนคัดคู่พื้นที่และคัดคู่จุด

ดังนั้นวิธีที่เสนอนี้หากค่า D มีขนาดมากกว่าด้านเพื่อนบ้านของคู่พื้นที่ คู่พื้นที่นั้นก็จะโดนคัดทิ้งไป ทั้ง ๆ ที่คู่จุดในคู่พื้นที่ที่คัดทิ้งไปนั้น อาจจะมีระยะระหว่างคู่จุดที่น้อยกว่าค่า D ซึ่งหากเป็นวิธีเอเอบีบีก็จะได้ผลลัพธ์ออกมาทั้งหมดดังรูปที่ 4-5



รูปที่ 4-5 ก) คู่จุดของวิธีเอเอบีบีเมื่อ $D = 0.01$ ข) คู่จุดของวิธีที่เสนอเมื่อ $D = 0.01$

ในขณะที่เดียวกันหากค่า D มีขนาดเล็กกว่าขนาดด้านของเพื่อนบ้านที่กล่าวถึง ก็จะทำให้ผลลัพธ์ของวิธีที่เสนอกับวิธีเอเอบีบี มีผลลัพธ์ที่เท่ากันดังผลการทดลองเมื่อ $D=0.01$ ดังนั้นความแม่นยำของวิธีที่เสนอเมื่อเทียบกับวิธีเอเอบีบี จะมีความแม่นยำเท่ากันก็ต่อเมื่อ ค่า D เข้าใกล้ศูนย์

แต่ในความเป็นจริงเมื่อนำมาทดลองกับกับฉากที่ต้องมีการเคลื่อนไหวในการทดลองที่ 4.3.2 จะไม่สามารถทำได้ เนื่องจากหากค่า D มีค่าเล็กมากก็จะทำให้ค่า D นั้นมีค่าเล็กเกินกว่าระยะทางที่วัตถุที่เคลื่อนที่ไปเมื่อนำเฟรมปัจจุบันเทียบกับเฟรมก่อนหน้า ซึ่งส่งผลให้ไม่พบการชนกันและวัตถุทะลุกันในที่สุด

วิธีแก้ไขก็คือ ลดค่าความยาวของแต่ละเฟรม T ให้น้อยลงมาก ๆ ซึ่งจะทำให้ระยะทางวัตถุของเฟรมก่อนหน้ากับเฟรมปัจจุบันไม่ต่างกันมากกว่าค่า D แต่ก็จะทำให้การทำงานของฉากนั้นนานขึ้นเพราะมีจำนวนเฟรมมากขึ้นเพื่อให้ครบความยาวของฉากที่กำหนด นอกจากนี้วิธีลดค่า T ก็ยังไม่รับรองว่าจะได้ผลในเหตุการณ์ที่วัตถุมีความเร็วสูงมาก ทำให้มีอีกวิธีที่จะแก้ปัญหานี้คือเปลี่ยนไปใช้การตรวจจับการชนกันแบบต่อเนื่อง ซึ่งจะอยู่บนขอบเขตของงานวิจัยนี้

ดังนั้นในการทดลองด้านความแม่นยำ กับวัตถุที่มีการเคลื่อนไหวจึงกำหนดให้ค่า $D=0.05$ และ $T=0.005$ ซึ่งส่งผลให้เกิดกรณีเดียวกันกับวัตถุหยุดนิ่งความละเอียดสูงเมื่อ $D=0.1$ คือ ผลลัพธ์ที่ได้ออกมาน้อยกว่าผลลัพธ์ของวิธีเอเอบีบี แต่เมื่อเปลี่ยนจากดูผลลัพธ์ทางตัวเลข มาสังเกตผลลัพธ์จากภาพ ก็จะได้เห็นว่าทั้งสองวิธีได้ผลลัพธ์ของการเคลื่อนไหวออกมาใกล้เคียงกัน และรูปทรงของกราฟมีรูปร่างที่คล้ายกัน จึงพอกล่าวได้ว่าเมื่อดูผลลัพธ์ในท้ายสุดแล้ววิธีที่เสนอนี้ก็มีความแม่นยำในระดับหนึ่งเมื่ออาศัยการวัดด้วยภาพ

4.5.2 ประสิทธิภาพด้านความเร็ว

จากการทดลอง 4.4.1 ฉากประเภทวัตถุหยุดนิ่ง พบว่าความเร็วของวิธีที่เสนอจะช้ากว่าเมื่อเทียบกับวิธีเอเอบีบี ในฉากที่วัตถุมีความละเอียดต่ำ แต่จะเร็วกว่าในฉากที่วัตถุมีความละเอียดสูง ที่เป็นเช่นนี้เพราะว่า เวลาที่ใช้ในการจัดการกับวัตถุตัวแทนของวิธีเอเอบีบีจะมีน้อยมากหากวัตถุมีความละเอียดต่ำและจะมีสูงมากขึ้นเมื่อวัตถุมีความละเอียดสูง ซึ่งรวมไปถึงเวลาที่ต้องคำนวณหาจุด หากเป็นในวัตถุที่มีความละเอียดต่ำก็ใช้เวลาที่น้อย แต่เมื่อเป็นวัตถุที่มีความละเอียดสูงจำนวนผลลัพธ์ก็จะมีมากทำให้ต้องใช้เวลามากขึ้นตามไปด้วย

ในขณะที่วิธีที่เสนอนั้น ต้องใช้เวลาในการคำนวณ ในส่วนประมวลผลหาจุดในหน่วยประมวลผลกราฟิก และเวลาในการส่งข้อมูลระหว่างหน่วยประมวลผลกราฟิกกับหน่วยประมวลผลกลางซึ่งจะกินเวลามากกว่า เวลาที่ใช้ในการจัดการวัตถุตัวแทนของวัตถุความละเอียดต่ำ

แต่เนื่องจากมีการกำหนดขนาดของคู่พื้นที่และจำนวนของคู่พื้นที่ ที่จะประมวลผลในหน่วยประมวลผลกราฟิกไว้คงที่ ทำให้ไม่ว่าในฉากที่วัตถุจะมีความละเอียดเท่าใดก็ตามเวลาการประมวลที่ใช้ไปเพื่อการทำงานก็จะใช้เวลาใกล้เคียงกันหรือเกือบคงที่ จนกระทั่งจำนวนคู่พื้นที่ที่ใช้ในการประมวลผลมีมากกว่าจำนวนคู่พื้นที่ที่กำหนดไว้

ดังนั้นวิธีที่เสนอนี้จึงเหมาะกับฉากที่มีวัตถุความละเอียดสูง เพราะหากใช้วิธีเอเอบีบีหน่วยประมวลผลกลางก็จะมีส่วนที่ต้องประมวลผลมากขึ้นในการหาคู่จุดทั้งหมด ของวัตถุที่มีความละเอียดสูง จะเห็นได้ชัดเมื่อให้ $D = 0.1$ ในฉากที่สามของการทดลอง 4.4.1 ซึ่งจะทำให้ได้คู่จุดจำนวนสูงขึ้นไปเป็น 10 เท่าจากเมื่อ $D = 0.01$ และต้องใช้เวลาเพิ่มขึ้นเป็น 10 เท่าด้วยเช่นกัน ในขณะที่วิธีที่เสนอนี้ยังคงใช้เวลาเกือบคงที่ในฝั่งของหน่วยประมวลผลกราฟิก และเวลาที่ใช้ในการประมวลทางฝั่งของหน่วยประมวลผลกลางก็เพิ่มขึ้นไม่มาก เพราะใช้ไปกับการคัดคู่พื้นที่และการจัดข้อมูลที่ได้มาจากฝั่งหน่วยประมวลผลกราฟิกเท่านั้น ดังจะเห็นได้จากผลการทดลองในฉากที่ 3 ที่จำนวนคู่จุดที่ได้เพิ่มขึ้นเกือบ 5 เท่าเมื่อเปลี่ยนจาก $D = 0.01$ เป็น $D = 0.1$ แต่เวลาในการประมวลผลเพิ่มขึ้นเพียง 0.1 เท่า เท่านั้น

และจากการที่วิธีเอเอบีบีต้องใช้เวลามากในการจัดการกับวัตถุตัวแทน ในวัตถุความละเอียดสูง ดังนั้นจึงได้ทำการทดลอง 4.4.2 ฉากประเภทวัตถุเคลื่อนที่แต่ไม่มีการเปลี่ยนแปลงรูปร่าง เพื่อศึกษาดูว่าเมื่อวัตถุตัวแทนของวัตถุที่มีความละเอียดสูงมีการเคลื่อนที่จะส่งผลอย่างไรบ้าง

ซึ่งจากการทดลองที่ 4.4.2 พบว่าในฉากที่หนึ่ง นั้นเมื่อถึงรอบการทำงานที่ 200-300 และ 300-400 และฉากที่สอง รอบการทำงานที่ 300-400 วัตถุในฉากจะออกห่างจากกัน ไม่มีการชนกัน วิธีเอเอบีบีต้องใช้เวลามากพอสมควรในการจัดการกับวัตถุตัวแทน ที่มีการเคลื่อนที่แต่ไม่ชนกัน ในขณะที่วิธีที่เสนอไม่มีการประมวลผลในส่วนนี้มีเพียงการประมวลผลเพื่อเช็คการทำงานของตัวควบคุมอนุภาคเท่านั้น ซึ่งจะเห็นได้ชัดว่ามีความเร็วที่สูงกว่า โดยเฉพาะเมื่อวัตถุตัวแทนเพิ่มจำนวนมากขึ้นอย่างในฉากที่สอง

ส่วนในช่วงการประมวลผลเมื่อวัตถุมีการชนกันจะเห็นได้ว่า วิธีที่เสนอต้องมีการเรียกใช้งานหน่วยประมวลผลกราฟิกทำให้ต้องใช้เวลาในการส่งข้อมูลไปกลับระหว่างหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิก รวมถึงเวลาในการประมวลผลหาคู่จุด เวลาที่ใช้ไปจึงมากขึ้นกว่าตอนยังไม่มีการชนกันอย่างเห็นได้ชัด และจากผลการทดลองที่ 4.4.1 และ 4.4.2 จะเห็นได้ว่าเวลาที่ใช้ไปจะเกือบเท่ากันหาก จำนวนคู่พื้นที่ที่คัดเลือกมามีจำนวนใกล้เคียงกันเนื่องด้วยหากจำนวนคู่พื้นที่น้อยเวลาที่ใช้ในการส่งข้อมูลไปกลับระหว่างหน่วยประมวลผลกลางกับหน่วยประมวลผลกราฟิกก็ลดลงตามไปด้วย

นอกจากนี้จำนวนคู่พื้นที่ที่ยังส่งผลต่อความเร็วอย่างเห็นได้ชัดในการทดลองที่ 4.4.2 จากที่สาม เมื่อจำนวนคู่พื้นที่ที่คัดเลือกมา มีจำนวนสูงกว่าที่หน่วยประมวลผลกราฟิกจะรองรับได้ ทำให้คู่ที่เหลือต้องส่งไปทำงานยังฝั่งหน่วยประมวลผลกลาง เวลาที่ใช้จึงเพิ่มมากขึ้นตามจำนวนคู่ที่เกินมาเพราะไม่ได้ทำงานคู่ขนานกันเหมือนทางฝั่งหน่วยประมวลผลกราฟิก แต่อย่างไรก็ตามผลลัพธ์ที่ได้ก็ยังคงเร็วกว่าวิธีเอเอบีบีเนื่องจากจำนวนคู่ที่เกินมาไม่มากนัก

และจากการทดลองที่ 4.4.3 จากประเภทวัตถุเคลื่อนที่และมีการเปลี่ยนแปลงรูปร่าง จะสามารถเขียนผลการทดลองให้อยู่รูปของกราฟ รวมถึงจำนวนของคู่พื้นที่ที่ส่งไปประมวลผลชั้นละเอียด ที่หน่วยประมวลผลกราฟิกของแต่ละฉากดังรูปที่ 4-3 ซึ่งเมื่อนำฉากที่หนึ่ง ของการทดลองนี้ เทียบกับฉากที่หนึ่ง ของการทดลองที่ 4.4.2 ก็จะได้เห็นได้ชัดว่าเมื่อวัตถุมีการเปลี่ยนแปลงรูปร่างไป วิธีเอเอบีบีก็ต้องมีการใช้เวลาในการจัดการวัตถุเพิ่มขึ้นเช่นกัน ในขณะที่วิธีที่เสนอเนื่องจากวัตถุมีความอ่อนสูงมาก ทำให้เมื่อวัตถุนี้ตกลงมายังวัตถุแข็งจึงยุบลงไปตามรูปร่างของวัตถุแข็ง ทำให้จำนวนคู่พื้นที่ที่มีโอกาสเกิดการชนกันสูงขึ้นตามไปด้วย เวลาที่ใช้จึงมากกว่าฉากที่หนึ่งของการทดลองที่ 4.4.2 นอกจากนี้ผลการทดลองจากฉากที่สอง และ สาม จะเห็นได้ชัดว่าปัจจัยสำคัญที่ส่งผลต่อความเร็วของวิธีที่เสนอยังคงเป็นจำนวนคู่พื้นที่ที่มีโอกาสเกิดการชนกันดังจะเห็นได้ในฉากที่สาม ซึ่งจำนวนคู่พื้นที่ที่คัดเลือกมานั้นมีสูงมากในบางช่วงและส่งผลต่อความเร็ว ทำให้ความเร็วของวิธีที่เสนอมีช่องว่างลดลงเมื่อเทียบกับวิธีเอเอบีบี

4.5.3 สรุปผลการทดลอง

จากผลการทดลองทั้งทางด้านความแม่นยำและทางด้านความเร็ว จะเห็นได้ว่า วิธีที่เสนอนี้ให้ประสิทธิภาพที่ดีกว่าวิธีที่ใช้วัตถุตัวแทนอย่างวิธีเอเอบีบีในแง่ของความเร็ว โดยเฉพาะกับวัตถุซึ่งมีความละเอียดสูงแต่ควรมีการชนกันของวัตถุที่ทำให้เกิดคู่พื้นที่ ที่มีโอกาสเกิดการชนกันจำนวนน้อยกว่าหรือเท่ากับ 16 คู่ และจะยิ่งเห็นความแตกต่างได้ชัดเมื่อในคู่พื้นที่นั้น ๆ มีจุดที่เป็นผลลัพธ์อยู่เป็นจำนวนมาก เพราะวิธีที่เสนอจะใช้เวลาเกือบคงที่ในการประมวลผลเนื่องจากเป็นการทำงานแบบขนานผ่านหน่วยประมวลผลกราฟิก ในขณะที่วิธีเอเอบีบีต้องใช้เวลามากขึ้นตามจำนวนผลลัพธ์ที่มากขึ้น

ส่วนในด้านของความแม่นยำ หากใช้กับฉากหรือการจำลองที่ระยะห่างน้อยที่สุดระหว่างจุดของวัตถุเข้าใกล้ศูนย์และวัตถุเคลื่อนที่ไม่เร็วมากแล้ว วิธีนี้จะให้ความแม่นยำเช่นเดียวกับวิธีเอเอบีบี แต่หากวัตถุมีการเคลื่อนที่ที่เร็วระดับหนึ่งทำให้ต้องมีการเพิ่มระยะห่างน้อยที่สุดระหว่างจุดของวัตถุ ผลลัพธ์ที่ได้ก็จะมีจำนวนน้อยกว่าวิธีเอเอบีบี แต่อย่างไรก็ตามจำนวนจุดของผลลัพธ์ที่ได้มากก็เป็นจุดที่อยู่บนคู่พื้นที่ที่ผ่านการคัดเลือก จึงส่งผลให้จำนวนจุดที่อยู่บนคู่พื้นที่นี้ผ่าน

เกณฑ์มากกว่าคู่พื้นที่ที่ไม่ได้ผ่านการคัดเลือก ซึ่งทำให้ผลลัพธ์สุดท้ายเมื่อวัดด้วยภาพจึงมีลักษณะคล้ายและใกล้เคียงกัน จึงพอกกล่าวได้ว่าหากใช้ในงานที่ไม่ต้องการความแม่นยำถึงระดับตัวเลขที่ละเอียดมาก ทั้งยังต้องการความเร็วในการทำงานวิธีที่เสนอนี้จะเหมาะสมอย่างยิ่ง



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สรุปผลการวิจัย และแนวทางการพัฒนาต่อ

สรุปผลการวิจัย

ปัญหาของวิธีการตรวจจับการชนกันโดยอาศัยวัตถุตัวแทนความละเอียดต่ำมาช่วยในการคัดเลือกพื้นที่ที่มีโอกาสเกิดการชนกันของวัตถุจริงความละเอียดสูง คือ ต้องมีการจัดการเพื่อให้วัตถุตัวแทนมีความสอดคล้องกับวัตถุจริงอยู่เสมอ ไม่ว่าจะวัตถุจริงจะมีการเคลื่อนที่หรือมีการเปลี่ยนรูปร่างก็ตาม จึงทำให้การทำงานในส่วนนี้เป็นภาระที่มากขึ้นเรื่อย ๆ เมื่อวัตถุจริงมีความละเอียดสูงและมีจำนวนเพิ่มขึ้นเพราะส่งผลต่อจำนวนของวัตถุตัวแทนโดยตรง

งานวิจัยนี้จึงได้เสนอวิธีที่ไม่ต้องใช้วัตถุตัวแทนในการตรวจหาการชนกัน แต่ใช้ข้อมูลของวัตถุจริงอย่างเดียวในการตรวจจับการชนกัน รวมทั้งแก้ปัญหาที่เกิดขึ้นกับวิธีที่ปรับปรุงมาจากการหาคุณสมบัติใกล้เคียง [6-10] ซึ่งการจะทำแบบนี้ได้จำเป็นต้องมีการประมวลผลแบบคู่ขนาน จึงได้ใช้หน่วยประมวลผลกราฟิกในการประมวลผลลักษณะนี้ แต่ปัญหาที่สำคัญของหน่วยประมวลผลกราฟิกคือ ปัญหาความล่าช้าในการส่งข้อมูลทั้งไปและกลับระหว่างหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิก โดยเฉพาะการส่งข้อมูลกลับจากหน่วยประมวลผลกราฟิกไปยังหน่วยประมวลผลกลาง งานวิจัยนี้จึงมีการตัดข้อมูลบางส่วนที่ไม่เป็นผลลัพธ์ออกไปก่อนส่งกลับไปให้หน่วยประมวลผลกลาง

นอกจากนี้เนื่องจากหน่วยความจำของการ์ดแสดงผลที่หน่วยประมวลผลกราฟิกใช้งานมีจำกัด ทำให้ไม่สามารถรองรับข้อมูลปริมาณที่มากเกินไปได้ งานวิจัยนี้จึงได้เสนอวิธีการคัดเลือกข้อมูลที่จำเป็น คือข้อมูลคู่พื้นที่ที่คาดว่าจะมีผลลัพธ์อยู่เท่านั้น ส่งไปทำการประมวลผลขั้นละเอียดที่หน่วยประมวลผลกราฟิก โดยอาศัยการแบ่งพื้นที่ออกเป็นส่วน ๆ และอาศัยตัวควบคุมอนุภาคประจำแต่ละพื้นที่ควบคุมการปล่อยอนุภาคทำให้ไม่มีอนุภาคมากหรือน้อยจนเกินไป ซึ่งตัวควบคุมอนุภาคนั้นจะมีรัศมีการทำงานขึ้นอยู่กับความยืดหยุ่นของวัตถุ ทำให้ไม่จำเป็นต้องแบ่งพื้นที่ใหม่หากวัตถุมีการยืดรูปร่าง

จากผลการทดลองนำวิธีนี้ไปใช้เทียบกับวิธีที่ใช้วัตถุตัวแทนอย่างวิธีเอเอบีบี จึงสรุปได้ว่าวิธีนี้เหมาะสมอย่างมากกับวัตถุที่มีความละเอียดสูง เพราะให้ความเร็วที่สูงกว่าวิธีเอเอบีบีในการค้นหาผลลัพธ์คู่จุดที่มีโอกาสเกิดการชนกันประมาณ 1.5 - 4 เท่า ขึ้นอยู่กับฉากที่ทดลอง และสามารถให้ความแม่นยำที่สูงเพียงพอใช้ในขั้นตอนการตอบสนองการชนกัน และใกล้เคียงหรือเท่ากับกับวิธีเอเอบีบีเมื่อกำหนดให้ค่าระยะห่างระหว่างคู่จุดของคู่วัตถุเข้าใกล้ศูนย์ นอกจากนี้ยัง

เป็นวิธีที่มีความง่ายไม่ซับซ้อนทั้งในเรื่องโครงสร้างข้อมูล และการค้นหาผลลัพธ์เพราะใช้ข้อมูลเดิมของวัตถุจริง ไม่มีการสร้างวัตถุตัวแทนขึ้นมาซึ่งทำให้ลดความซับซ้อนในการจัดการวัตถุตัวแทนเหล่านี้ และเมื่อนำวิธีที่เสนอนี้มาเทียบกับวิธีเอเอบีบีที่มีการใช้หน่วยประมวลผลกราฟิก [18] จะเห็นได้ว่าเป็นสามารถประมวลผลบนฉากที่มีความละเอียดสูงกว่าและมีวัตถุมากกว่า โดยใช้เวลาที่ใกล้เคียงกัน ทั้งยังให้ผลลัพธ์ที่เพียงพอต่อการตอบสนองการชนกัน คือ คู่จุด ทิศทาง ระยะห่าง และคู่รูปทรงสามเหลี่ยม ในขณะที่วิธีเอเอบีบีที่มีการใช้หน่วยประมวลผลกราฟิก [18] ให้ผลลัพธ์เพียงคู่ของรูปทรงสามเหลี่ยมที่มีการชนกันเท่านั้น

แนวทางในการพัฒนาต่อ

วิธีที่นำเสนอนี้ยังไม่ครอบคลุมถึงการตรวจสอบการชนที่ใช้ได้กับทุกวัตถุ อันเนื่องมาจากสาเหตุที่จะกล่าวถึงต่อไป เพื่อเป็นแนวทางในการพัฒนาต่อทำให้วิธีนี้มีความเหมาะสมและครอบคลุมต่อทุกการใช้งานมากยิ่งขึ้น ดังนี้

1. โครงสร้างการเชื่อมโยงเกิดการเปลี่ยนแปลง

เนื่องจากวิธีนี้ใช้โครงสร้างการเชื่อมโยงของจุดยอดบนวัตถุเป็นทางเดินในการค้นหาจุดยอดชนกัน ดังนั้นถ้าโครงสร้างการเชื่อมโยงมีการเปลี่ยนแปลงไปเช่น การแตกหักของวัตถุ ย่อมจะส่งผลให้ไม่สามารถจะใช้วิธีที่เสนอนี้ได้ ซึ่งแนวทางในการพัฒนาต่อไปคือ ต้องพัฒนาโครงสร้างข้อมูลที่ใช้เก็บโครงสร้างการเชื่อมโยงของจุดยอดที่ได้แบ่งพื้นที่ออกเป็น ส่วน ๆ ให้สามารถมีการแก้ไขได้ทั้ง การเพิ่มจุดยอด การลดจุดยอดและการเปลี่ยนแปลงการเชื่อมโยงโครงสร้างของวัตถุ ซึ่งจะทำให้ได้โครงสร้างการเชื่อมโยงที่พร้อมใช้งานและทันกาลอยู่เสมอในแต่ละพื้นที่

2. มีการชนกันภายในตนเอง

ปัญหาของการชนกันภายในตนเองนั้นหากเป็นการชนกันของคนละพื้นที่ที่แบ่ง ในวัตถุเดียวกัน วิธีที่เสนอสามารถจะนำมาแก้ไขเพื่อให้ตรวจสอบการชนกันภายในได้ โดยการสมมติว่าแต่ละตัวควบคุมอนุภาคล้วนอยู่บนพื้นที่ซึ่งเป็นคนละวัตถุกัน แต่ว่าหากเป็นการชนกันที่เกิดภายในพื้นที่เดียวกันแล้วจะไม่สามารถใช้แนวทางนี้ได้ จึงยังเป็นปัญหาที่น่าสนใจและยังต้องหาแนวทางแก้ไขต่อไป

3. วัตถุมีความเร็วสูง

จากปัญหาที่เกิดขึ้นตามที่ได้กล่าวไว้ในบทที่ 4 การตรวจสอบการชนกันแบบต่อเนื่องจึงจำเป็นอย่างยิ่งโดยเฉพาะวัตถุที่เคลื่อนที่เร็วมาก ซึ่งแนวทางการพัฒนาต่อไปนั้นสามารถจะต่อยอดจากผลลัพธ์เดิมของวิธีที่ได้เสนอไป คือเพิ่มการทำนายและหาสมการการเคลื่อนที่ที่น่าจะเกิดขึ้นระหว่างจุดผลลัพธ์ที่ได้มาขั้นตอนสุดท้าย ซึ่งจะทำได้ทำให้สามารถทราบได้ว่า วัตถุจะเคลื่อนที่ไปชนกัน ณ เวลาเท่าใด ทำให้สามารถป้องกันการทะลุกันของวัตถุที่เคลื่อนที่เร็วมากได้ ซึ่งเป็นส่วนที่การตรวจจับการชนกันแบบไม่ต่อเนื่องขาดหายไป

4. วัตถุที่มีความยืดหยุ่นสูง

ในวิธีที่เสนอนั้นรัศมีของตัวควบคุมอนุภาคได้มีการกำหนดไว้ล่วงหน้า ซึ่งทำให้ผู้ใช้ต้องกำหนดค่าความยืดหยุ่นและค้นหาว่าวัตถุหนึ่งควรยืดได้เท่าไร ทั้งยังเป็นการทำให้ตัวควบคุมอนุภาคต้องทำงานมากโดยไม่จำเป็น เพราะไม่ใช่ทุกครั้งที่วัตถุจะมีการยืดหรือหดแต่รัศมีการทำงานได้กำหนดตายตัวไว้แล้ว ดังนั้นแนวทางการพัฒนาต่อไปคือ จะมีการแก้ไขรัศมีการทำงานของตัวควบคุมอนุภาคโดยเฉพาะกับตัวควบคุมอนุภาคที่มีการทำงานเพราะยอมจะเป็นบริเวณ ที่ทำให้เกิดการยืดหรือหด โดยจะอาศัยระยะทางระหว่างตัวควบคุมอนุภาคเพื่อนบ้านในช่วงเวลา ก่อนหน้า เทียบกับเวลาปัจจุบัน เพื่อหารัศมีการทำงานที่เหมาะสมในแต่ละช่วงเวลาต่อไป

5. การเคลื่อนที่ของอนุภาคจำนวนมาก

จะเห็นได้ว่าเวลาที่ใช้ในปัจจุบันนั้น จะขึ้นอยู่กับเวลาที่ใช้ในการค้นหาพื้นที่เนื่องจากเป็นการทำงานบนหน่วยประมวลผลกลาง เพราะฉะนั้นหากมีพื้นที่มากก็ย่อมต้องใช้เวลาในการค้นหาเพิ่มมากขึ้นเช่นกัน แต่ถ้านักพัฒนาสามารถพัฒนาให้การค้นหานี้มาประมวลผลอยู่ในหน่วยประมวลผลกราฟิกก็จะทำให้เวลาในส่วนนี้คงที่เนื่องด้วยการประมวลผลแบบขนาน เพียงแต่จะแปรผันตามข้อจำกัดของฮาร์ดแวร์ของหน่วยประมวลผลกราฟิกเหมือนเช่นการตรวจจับชั้นละเอียด

6. การแบ่งพื้นที่วัตถุเว้า

ในวิธีที่เสนอนี้ใช้การแบ่งวัตถุโดยอาศัยเพียง ตำแหน่งของรูปทรงสามเหลี่ยมของวัตถุ ไม่ได้พิจารณาถึงการเชื่อมโยงภายใน ดังนั้นเมื่อนำมาใช้กับวัตถุเว้าที่มีความละเอียดต่ำ จึงทำให้เกิดปัญหาพื้นที่เดียวกันแต่ไม่สามารถเชื่อมโยงถึงกันได้ โดยปัญหานี้ส่วนมากจะไม่เกิดขึ้นในวัตถุ ความละเอียดสูงเนื่องจากตำแหน่งของพื้นที่เดียวกัน จุดยอดที่เชื่อมโยงกันมีความใกล้เคียงกันมากกว่าจุดยอดที่ไม่เชื่อมโยงกัน ซึ่งวิธีการแก้ไขสามารถที่จะพัฒนาได้เป็น 2 แนวทางคือ คือ หนึ่ง เลือกใช้วิธีการแบ่งใหม่ที่พิจารณาถึงการเชื่อมโยง สองยังคงใช้วิธีการแบ่งแบบเดิมแต่เพิ่มขั้นตอนให้ตัวควบคุมอนุภาคมีการตรวจก่อนว่าสามารถปล่อยอนุภาคให้เคลื่อนที่ไปครบทุกจุดยอดได้

หรือไม่ ถ้าหากไม่ได้ก็จะสร้างการเชื่อมโยงระหว่างตัวควบคุมอนาคตไปยังจุดยอดตัวแทนของกลุ่มที่ขาดการเชื่อมโยง แต่ไม่ว่าจะแก้ด้วยวิธีใดก็ตาม หากวัตถุประสงค์มีการเปลี่ยนแปลงจุดยอดการเชื่อมโยงก็ยังคงเป็นปัญหาอยู่



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

- [1] Yehy, T. Y., Faloutsosy, P., Patelz, S. J., and Reinmany, G. Parallax: An Architecture for Real-Time Physics. presented at ISCA, 2007.
- [2] Madhav, P., Dinesh, M., and Ming, C.L. Incremental algorithms for collision detection between solid models. in Proceedings of the third ACM symposium on Solid modeling and applications, Salt Lake City, Utah, United States: ACM Press, 1995.
- [3] Nealen, A., Muller, M., Keiser, R., Boxerman, E., and Carlson, M. Physically Based Deformable Models in Computer Graphics. Computer Graphics Forum, vol. 25, pp. 809-836, 2006.
- [4] Christer, E. Real-Time Collision Detection. Morgan Kaufmann, 2005.
- [5] Cameron, S. A Comparison of Two Fast Algorithms for Computing the Distance between Convex Polyhedra. presented at IEEE Trans. Robotics & Automation, 1996.
- [6] Stephane, G., and Debunne, G., Monte-Carlo collision detection. INRIA. 2004.
- [7] Kimmerte, S., Nesme, M., and Faure, F., Hierarchy Accelerated Stochastic Collision Detection. presented at KNF04, 2004.
- [8] Wang, T., Li, W., Wang, Y., Ge, Z., and Han, D. Stochastic collision detection between deformable models using particle swarm optimization algorithm. 12th International Multi-Media Modelling Conference Proceedings. 2006.
- [9] Saenghaengtham, N., and Kanongchaiyos, P. A Collision Detection Algorithm Using Particle Sensor. RAM, 2006.
- [10] Senin, M., Kojekine, N., Savchenko, V., and Hagiwara, I. Particle-based Collision Detection. EUROGRAPHICS, 2003.
- [11] Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., and Purcell, T.J. A Survey of General-Purpose Computation on Graphics Hardware. EUROGRAPHICS, 2005.
- [12] Yehy, T.Y., Faloutsos, P., and Reinman, G. Enabling Real-Time Physics Simulation in Future Interactive Entertainment. Sandbox Symposium, 2006.
- [13] Govindaraju, N.K., Lin, M., and Manocha, D. Fast and Reliable Collision

- Culling using Graphics Hardware. VRST. 2004.
- [14] Govindaraju, N.K, Lin, M., Manocha, D., and Redon, S., CULLIDE: Interactive collision detection between complex models in large environments using graphics hardware. Proceeding of ACM SIGGRAPH / Eurographics Workshop on Graphics Hardware, 2003.
- [15] Knott, D. CInDeR: Collision and Interference Detection in Real Time Using Graphics Hardware. Master's Thesis. Department of Computer Science: THE UNIVERSITY OF BRITISH COLUMBIA, 2003.
- [16] Naga, K.G., Ming, C.L., and Dinesh, M. Quick-CULLIDE: fast inter- and intra-object collision culling using graphics hardware. ACM SIGGRAPH 2005 Courses. Los Angeles, California: ACM Press, 2005.
- [17] GRESS, A., and ZACHMANN, G., Object-space interference detection on programmable graphics hardware. Proceeding of SIAM Conference on Geometric Design and Computing. 2004.
- [18] Xinyu, Z., and Kim, Y.J., Interactive Collision Detection for Deformable Models Using Streaming AABBs. IEEE Transactions on Visualization and Computer Graphics, vol. 13, pp. 318-329, 2007.
- [19] Masayuki, T., Mikio, S., Ishikawajima, H., and Seiichi, K., Rigid body simulation using a particle method. ACM SIGGRAPH 2006 Research posters. Boston, Massachusetts: ACM Press, 2006.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายฐิติ รุ่งเจริญไพศาล เกิดเมื่อวันที่ 6 พฤศจิกายน 2526 ในกรุงเทพมหานคร จบการศึกษาชั้นมัธยมศึกษาที่โรงเรียนสุรวิทยาคาร จ.สุรินทร์ และจบการศึกษาระดับปริญญาบัณฑิต จาก ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยธรรมศาสตร์ เมื่อปีการศึกษา 2548



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย