

การจำลองต้นามิแบบ 3 มิติ



นายรัฐศรัณย์ จิรชนาวัดน์

ศูนย์วิทยทรัพยากร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคณนา ภาควิชาคณิตศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2551

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

3-D TSUNAMI SIMULATION



Mr. Ratsaran Jiratanawat

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

Department of Mathematics

Faculty of Science

Chulalongkorn University

Academic Year 2008

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

การจำลองสีนามิแบบ 3 มิติ

โดย

นายรัฐศรัณย์ จิรชนาวัฒน์

สาขาวิชา

วิทยาการคณนา

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ผู้ช่วยศาสตราจารย์ ดร.รัชลิดา ลิปิกรณ์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

รองศาสตราจารย์ ดร.จักษ์ อัสวานันท์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นำวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง  
ของการศึกษาตามหลักสูตรปริญญาโทบริหารธุรกิจ

..... คณบดีคณะวิทยาศาสตร์  
(ศาสตราจารย์ ดร. สุพจน์ หารหนองบัว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ  
(รองศาสตราจารย์ สุชาดา ศิริพันธ์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก  
(ผู้ช่วยศาสตราจารย์ ดร.รัชลิดา ลิปิกรณ์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม  
(รองศาสตราจารย์ ดร.จักษ์ อัสวานันท์)

..... กรรมการ  
(อาจารย์ ดร.ศุภกานต์ พิมพ์เรศ)

ศูนย์วิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

รัฐสรณ์ย์ จิรชนาวัดน์ : การจำลองสึนามิแบบ 3 มิติ. (3-D TSUNAMI SIMULATION) อ. ที่  
 ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.รัชลิดา ลิปิกรณ์ , อ. ที่ปรึกษาวิทยานิพนธ์ร่วม : รศ.ดร.จักษ์  
 อิศวานันท์, 82 หน้า.

เพื่อให้เข้าใจการเกิดสึนามิ และเพิ่มความรู้ในการศึกษากระบวนการเกิดสึนามิ  
 วิทยานิพนธ์นี้นำเสนอการจำลองภาพเคลื่อนไหว 3 มิติ ของการเกิดสึนามิ โดยใช้ข้อมูลเชิงตัวเลขที่ได้  
 จากการจำลองการเกิดสึนามิเมื่อวันที่ 26 ธันวาคม 2547 เวลา 0:58:53 (GTM) ในมหาสมุทรอินเดีย โดย  
 ใช้ Boussinesq model

ปัญหาหลักในการสร้างภาพเคลื่อนไหว 3 มิติ มาจากเวลาที่ใช้ในการประมวลผลจาก  
 ข้อมูลเชิงตัวเลขที่มีขนาดใหญ่ต้องใช้เวลานาน ทำให้เห็นภาพกระตุกไม่ต่อเนื่อง เพื่อให้แบบจำลองสึนามิ  
 มีความเสมือนจริงมากยิ่งขึ้น จึงมีการเพิ่มการให้แสงและเงาเข้าไปในกระบวนการจำลองสึนามิ และ  
 เพื่อให้เห็นมุมมองการเกิดสึนามิได้ดียิ่งขึ้น จึงได้เพิ่มฟังก์ชันต่าง ๆ อาทิ การย่อ-ขยายภาพ การหยุดภาพ  
 การเคลื่อนย้ายภาพ และการหมุนภาพ นอกจากนี้วิทยานิพนธ์นี้ได้นำเสนอเทคนิคที่สามารถลดเวลาใน  
 การประมวลผลและลดพื้นที่ในการจัดเก็บข้อมูล โดยทำการแปลงข้อมูลจากข้อมูลจำนวนจริง ไปเป็น  
 ข้อมูลเลขลักษณะฐานสองในรูปแบบไบต์อาร์เรย์ เพื่อให้ทำการประมวลผลได้เร็วขึ้น รวมทั้งใช้การ  
 ประมาณค่าในช่วงเสมือนพหุนาม เพื่อเพิ่มความละเอียดของภาพในส่วนที่ต้องการแสดง และเพิ่มเฟรม  
 ภาพเพื่อให้ภาพเคลื่อนไหวมีความต่อเนื่องมากยิ่งขึ้น

ศูนย์วิทยทรัพยากร

จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา คณิตศาสตร์

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2551

ลายมือชื่อนิสิต.....

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม.....

รัฐสรณ์ย์ จิรชนาวัดน์

รัชลิดา ลิปิกรณ์

จักษ์ อิศวานันท์

## 4872438023 : MAJOR COMPUTATIONAL SCIENCE

KEY WORD: 3D RENDERING / TSUNAMI / SIMULATION

RATSARAN JIRATANAWAT : 3-D TSUNAMI SIMULATION. THESIS PRINCIPAL  
ADVISOR : ASST. PROF. RAJALIDA LIPIKORN, Ph.D., THESIS CO-ADVISOR : ASSOC.  
PROF. JACK ASAVANANT, Ph.D, 82 pp.

In order to understand the occurrence of tsunami and to enrich one's knowledge on tsunami, this thesis presents three-dimensional tsunami simulation using tsunami data on Dec 26, 2004 which were generated by Boussinesq model.

The main problem in three-dimensional tsunami simulation is the computational complexity during the processing of huge numerical data that causes the simulation to flicker. Illumination and shadow are included in the rendering process in order to make the simulation more realistic and for better viewing, zooming, pause-play, move, and rotation are also available. Moreover, this thesis introduces a technique that can significantly reduce the computational time and storage space by converting floating numbers to byte array and also uses spline interpolation to increase resolution and extra in between movie frames to smooth the simulation.

ศูนย์วิทยทรัพยากร

จุฬาลงกรณ์มหาวิทยาลัย

Department : **Mathematics**

Field of study : **Computational Science**

Academic year : **2008**

Student's signature : .....

Principal advisor's signature : .....

Co-advisor's signature : .....

## กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. รัชติดา ลิปิกรณ์ อาจารย์ที่  
ปรึกษาวิทยานิพนธ์หลัก ที่ท่านได้กรุณาให้ความรู้ คำแนะนำ และคำปรึกษาต่าง ๆ ที่ทำให้  
วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี

ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.จักษ์ อิศวานนท์ อาจารย์ที่ปรึกษา  
วิทยานิพนธ์ร่วม ที่ท่านได้กรุณาช่วยให้คำปรึกษาและแนะนำที่ดี ทำให้วิทยานิพนธ์นี้สมบูรณ์ยิ่งขึ้น

ขอกราบขอบพระคุณ รองศาสตราจารย์ สุชาดา ศิริพันธุ์ ประธานกรรมการ และ  
อาจารย์ ดร.ศุภกานต์ พิมลเรศ กรรมการ ที่ท่านช่วยเหลือ รวมถึงคำแนะนำในการทำงานวิจัยนี้  
ซึ่งทำให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณครอบครัวที่คอยให้กำลังใจ และ คุณณรงค์ฤทธิ์ แก้วบรร จักร  
สำหรับข้อมูลที่น่าสนใจมาใช้ในการงานวิจัยนี้ ตลอดจนพี่ ๆ เพื่อน ๆ คณาทุกคนที่ให้คำปรึกษา และเป็น  
กำลังใจให้ด้วยดีมาโดยตลอด

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	ง
คัดย่อภาษาอังกฤษ .....	จ
กิตติกรรมประกาศ .....	ฉ
สารบัญ .....	ช
สารบัญตาราง .....	ฌ
สารบัญภาพ .....	ฎ
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 วัตถุประสงค์ของการวิจัย .....	2
1.3 ขอบเขตของการวิจัย .....	3
1.4 ระเบียบวิธีวิจัย .....	3
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง .....	5
2.1 กล่าวนำ .....	5
2.2 งานวิจัยที่เกี่ยวข้อง .....	6
2.3 ทฤษฎี .....	8
2.3.1 การให้แสงและเงา .....	8
2.3.1.1 โมเดลการส่องสว่างวงกว้าง .....	9
2.3.1.2 การสะท้อนและการหักเห .....	13
2.3.1.3 เงา .....	14
2.3.1.4 วิธีการให้แสงและเงาโพลีกอน .....	15
2.3.1.5 ขั้นตอนวิธีซีบ์เฟอ์ .....	17
2.3.1.6 สายท่อการให้แสงและเงา .....	21
2.3.1.7 การสุ่มตัวอย่างสำหรับการสร้างภาพสามมิติ .....	21
2.3.1.8 ความคิดเห็นของภาพที่พบในการสร้างภาพสามมิติด้วยคอมพิวเตอร์ ...	23
2.3.2 ข้อมูลและการประมวลผล .....	24
2.3.2.1 การแทนข้อมูลในคอมพิวเตอร์ .....	25
2.3.2.2 การเข้ารหัสตัวเลข .....	25

2.3.2.3 ตัวดำเนินการข้อมูลเชิงระดับบิต.....	26
2.3.2.4 บิตฟิลด์.....	28
2.3.2.5 การประมาณค่าในช่วงเสมือนพหุนาม.....	28
บทที่ 3 วิธีดำเนินการวิจัย.....	32
3.1 วิธีดำเนินการส่วนที่1 .....	33
3.1.1 ข้อมูลที่นำมาจำลอง .....	33
3.1.2 การรวมข้อมูลส่วนพื้นน้ำและพื้นดิน.....	38
3.1.3 การเพิ่มความละเอียดของภาพ .....	40
3.1.4 การเพิ่มเฟรมภาพ.....	42
3.1.5 การแปลงรหัส.....	44
3.2 วิธีดำเนินการส่วนที่2 .....	46
3.2.1 การถอดรหัส.....	46
3.2.2 การให้แสงและเงา.....	48
บทที่ 4 ผลการประมวลผล.....	54
4.1 ผลจากการประมวลผล.....	54
4.2 วิเคราะห์ผลการทดลอง.....	55
4.2.1 ผลการเพิ่มจุดที่ทำการคำนวณ .....	57
4.2.2 ผลการเพิ่มเฟรมภาพ .....	59
4.2.3 ผลการเปรียบเทียบภาพเมื่อปรับลดสเกล.....	59
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ.....	63
5.1 สรุปผลการวิจัย.....	63
5.2 อภิปรายผลการวิจัย.....	64
5.3 ข้อเสนอแนะ .....	64
รายการอ้างอิง .....	65
ภาคผนวก ก. ....	69
ภาคผนวก ข. ....	76
ภาคผนวก ค. ....	80
ประวัติผู้เขียนวิทยานิพนธ์.....	82



## สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงตัวดำเนินการสำหรับการคำนวณข้อมูลในรูปแบบบิต.....	27
ตารางที่ 2.2 แสดงตัวอย่างของการดำเนินการ AND และ OR .....	27
ตารางที่ 3.1 รายละเอียดข้อมูลที่นำมาจำลอง .....	40
ตารางที่ 3.2 เปรียบเทียบขนาดข้อมูลระหว่างไฟล์ข้อมูลอักษรกับไฟล์ข้อมูลเลขลักษณะฐานสอง..	45
ตารางที่ 3.3 แสดงการ AND บิต จากรูปคำสั่งที่ 3.18 .....	47
ตารางที่ 3.4 แสดงการ OR บิต จากรูปคำสั่งที่ 3.19 .....	48
ตารางที่ 4.1 แสดงเวลาในการอ่านข้อมูลมาจัดเก็บในบัฟเฟอร์ก่อนทำการแปลงข้อมูล .....	54
ตารางที่ 4.2 แสดงเวลาในการอ่านข้อมูลมาจัดเก็บในบัฟเฟอร์หลังทำการแปลงข้อมูล.....	55
ตารางที่ 4.3 เปรียบเทียบเวลาในการอ่านข้อมูลมาจัดเก็บในบัฟเฟอร์.....	55
ตารางที่ 4.4 แสดงอัตราความเร็วเฟรมต่อวินาทีในการให้แสงและเงา.....	55
ตารางที่ 4.5 เปรียบเทียบเวลาอัตราความเร็วการประมวลผลเฟรมต่อวินาที .....	56
ตารางที่ 4.6 เปรียบเทียบความละเอียดของข้อมูลก่อนและหลังทำการเพิ่มจุดที่ใช้ในการคำนวณ..	57
ตารางที่ 4.7 เปรียบเทียบจำนวนเฟรมก่อนและหลังทำการเพิ่มจุดที่ใช้ในการคำนวณ.....	59
ตารางที่ 4.8 แสดงอัตราความเร็วเฟรมต่อวินาทีในการให้แสงและเงาหลังเพิ่มข้อมูลและเฟรม .....	59

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญภาพ

หน้า

รูปที่ 2.1 แสดงภาพการจำลองในรูปแบบสองมิติ.....	7
รูปที่ 2.2 แสดงค่าต่างๆ ที่ใช้ในการคำนวณหาปริมาณแสงที่ได้จากแหล่งกำเนิดแสง ชนิดจุดกำเนิดแสง .....	9
รูปที่ 2.3 มุมตกกระทบ $\theta$ ของแสงระหว่างเวกเตอร์ $L$ กับเวกเตอร์บรรทัดฐาน $N$ ที่พื้นผิว .....	11
รูปที่ 2.4 แสดงการสะท้อนของแสงที่เป็นไปตามกฎของแสง .....	13
รูปที่ 2.5 แสดงการหักเหของแสงเมื่อมีการเดินทางผ่านตัวกลางที่มีความหนาแน่นไม่เท่ากัน .....	14
รูปที่ 2.6 แสดงภาพตัวอย่างของ Gouraud สำหรับโพลีกอน 1 อัน .....	16
รูปที่ 2.7 จาก $(x,y)$ บนเส้นตรวจสอบ ตำแหน่งถัดไปมีพิกัด $(x+1,y)$ และตำแหน่งลงมามีพิกัด $(x,y-1)$ .....	19
รูปที่ 2.8 แสดงกรณีตำแหน่งของการสุ่มตัวอย่างนั้นไม่อยู่บน โพลีกอน.....	22
รูปที่ 2.9 แสดงกรณีที่โพลีกอนมีขนาดเล็กกว่าขนาดจุดภาพแต่บังเอิญอยู่ตรงตำแหน่ง การสุ่มตัวอย่างพอดี .....	22
รูปที่ 2.10 แสดงการเกิดบ็อปป์ง .....	23
รูปที่ 2.11 แสดงการเกิดรอยหยักที่เกิดขึ้นที่ขอบของวัตถุ.....	24
รูปที่ 2.12 แสดงตำแหน่งบิตที่เก็บค่าตัวเลขจุดทศนิยม .....	26
รูปที่ 3.1 แผนผังการทำงานขั้นตอนดำเนินการ.....	32
รูปที่ 3.2 แผนผังการทำงานในขั้นดำเนินการส่วนที่ 1.....	33
รูปที่ 3.3 ภาพข้อมูลที่นำมาจำลองบริเวณมหาสมุทรอินเดีย .....	34
รูปที่ 3.4 ภาพข้อมูลที่นำมาจำลองบริเวณชายฝั่งภาคใต้ของประเทศไทย.....	34
รูปที่ 3.5 ภาพข้อมูลที่นำมาจำลองบริเวณจังหวัดพังงา .....	35
รูปที่ 3.6 ภาพตัวอย่างข้อมูลส่วนพื้นน้ำที่นำมาจำลองบางส่วน.....	36
รูปที่ 3.7 ภาพตัวอย่างข้อมูลส่วนพื้นดินที่นำมาจำลองบางส่วน .....	37
รูปที่ 3.8 ภาพตัวอย่างข้อมูลที่ทำการรวมกันระหว่างพื้นดินและน้ำบางส่วน .....	39
รูปที่ 3.9 การเพิ่มข้อมูลใหม่จาก $3 \times 3$ จุดที่ใช้คำนวณ เป็น $5 \times 5$ จุดที่ใช้คำนวณ.....	40
รูปที่ 3.10 แสดงเฟรมภาพที่มีความละเอียดข้อมูล $2 \times 2$ .....	41
รูปที่ 3.11 แสดงการประมาณค่าตามแกน $x$ .....	41
รูปที่ 3.12 แสดงการประมาณค่าตามแกน $y$ .....	42

รูปที่ 3.13 แสดงภาพแต่ละเฟรมมีความละเอียด 2X2 .....	42
รูปที่ 3.14 แสดงภาพเฟรมใหม่ที่ได้จากการประมาณค่าในช่วง.....	43
รูปที่ 3.15 แสดงการแทรกเฟรมใหม่เข้าไประหว่างเฟรมที่ทำการประมาณค่า.....	43
รูปที่ 3.16 คำสั่งในการขยับบิตเพื่อเก็บข้อมูลตัวเลขทศนิยมได้ครบ 32 บิต .....	45
รูปที่ 3.17 แผนผังการทำงานขั้นตอนดำเนินการส่วนที่ 2.....	46
รูปที่ 3.18 ตัวอย่างคำสั่งในการขยับบิตและ AND บิตกับช่วงข้อมูลที่ต้องการ .....	47
รูปที่ 3.19 ตัวอย่างคำสั่งในการOR บิต .....	47
รูปที่ 3.20 แสดงระนาบเรขาคณิต.....	49
รูปที่ 3.21 แสดงการจัดเก็บค่าเป็นอาร์เรย์ 1 มิติ.....	50
รูปที่ 3.22 แสดงตำแหน่งภาพเรขาคณิตจากจุดอ้างอิง .....	50
รูปที่ 3.23 แสดงภาพตัวอย่างจุดยอด.....	51
รูปที่ 3.24 แสดงภาพพื้นผิวที่สร้างจากจุดอ้างอิง .....	51
รูปที่ 3.25 แสดงตัวอย่างในการใส่สีให้แต่ละระดับความสูงของพื้นดิน.....	53
รูปที่ 4.1 เปรียบเทียบภาพระหว่างภาพก่อนและหลังทำการเพิ่มจำนวนจุดที่ทำการคำนวณ บริเวณจังหวัดพังงา .....	58
รูปที่ 4.2 แสดงการขยายภาพบริเวณจังหวัดพังงาของข้อมูลทำการเพิ่มความละเอียด เป็น 289X289 .....	58
รูปที่ 4.3 แสดงภาพบริเวณมหาสมุทรอินเดียและบริเวณชายฝั่งทะเลภาคใต้ของประเทศไทย โดยไม่ได้ปรับสเกลของความสูงของพื้นดิน .....	60
รูปที่ 4.4 แสดงการปรับลดสเกลของความสูงของพื้นดิน100 เท่า.....	61
รูปที่ 4.5 แสดงการปรับลดสเกลของความสูงของพื้นดิน300 เท่า.....	62
รูปที่ ข.1 แสดงเฟรมภาพสูงสุดของคลื่นสึนามิบริเวณชายฝั่งภาคใต้ของประเทศไทย.....	78
รูปที่ ข.2 แสดงเฟรมภาพสูงสุดของคลื่นสึนามิบริเวณจังหวัดพังงา.....	79
รูปที่ ค.1 แสดงเฟรมภาพที่ 10 .....	80
รูปที่ ค.2 แสดงเฟรมภาพที่ 30 .....	80
รูปที่ ค.3 แสดงเฟรมภาพที่ 50 .....	80
รูปที่ ค.4 แสดงเฟรมภาพที่ 70 .....	80

รูปที่ ค.5 แสดงเฟรมภาพที่ 90 .....	81
รูปที่ ค.6 แสดงเฟรมภาพที่ 110 .....	81
รูปที่ ค.7 แสดงเฟรมภาพที่ 130 .....	81
รูปที่ ค.8 แสดงเฟรมภาพที่ 150 .....	81



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

สึนามิ (Tsunami) คือ คลื่นหรือกลุ่มคลื่นที่มีจุดกำเนิดอยู่ในเขตทะเลลึก ซึ่งมักปรากฏหลังแผ่นดินไหวขนาดใหญ่ แผ่นดินไหวใต้ทะเล ภูเขาไฟระเบิด ดินถล่ม แผ่นดินทรุด หรือ อุกกาบาตขนาดใหญ่ตกลงในทะเล คลื่นสึนามิสามารถเข้าทำลายพื้นที่ชายฝั่ง ทำให้เกิดการสูญเสียทั้งชีวิตและทรัพย์สินได้

คลื่นจะมีพฤติกรรมเป็นคลื่นน้ำตื้น (shallow-water waves) เมื่ออัตราส่วนระหว่างความลึกของน้ำและขนาดของคลื่นนั้นมีค่าต่ำ ดังนั้น เนื่องจากมีขนาดของคลื่นที่สูงมาก คลื่นสึนามิจึงมีคุณสมบัติเป็นคลื่นน้ำตื้นแม้อยู่ในทะเลลึกก็ตาม คลื่นน้ำตื้นนั้นมีความเร็วโดยประมาณเท่ากับรากที่สองของผลคูณระหว่างความเร่งจากสนามแรงโน้มถ่วง ( $9.8$  เมตร/วินาที<sup>2</sup>) และความลึกของน้ำ ตัวอย่างเช่น ในมหาสมุทรแปซิฟิกที่มีความลึกประมาณ  $4,000$  เมตร คลื่นจะเคลื่อนที่ด้วยความเร็วประมาณ  $200$  เมตรต่อวินาที หรือ  $720$  กิโลเมตรต่อชั่วโมง ส่วนที่ชายฝั่งที่มีความลึก  $40$  เมตร คลื่นจะมีความเร็วช้าลงเหลือ  $20$  เมตรต่อวินาที หรือ  $72$  กิโลเมตรต่อชั่วโมง

คลื่นสึนามิแตกต่างจากคลื่นน้ำธรรมดาอย่างมาก ตัวคลื่นนั้นสามารถเดินทางได้เป็นระยะทางไกล โดยไม่สูญเสียพลังงาน และสามารถเข้าทำลายชายฝั่งที่อยู่ห่างไกลจากจุดกำเนิดหลายพันกิโลเมตรได้ โดยทั่วไปแล้วคลื่นสึนามิซึ่งเป็นคลื่นในน้ำ จะเดินทางได้ช้ากว่าการสั่นสะเทือนของแผ่นดินไหวที่เป็นคลื่นที่เดินทางในพื้นดิน ดังนั้น คลื่นอาจเข้ากระทบฝั่งภายหลังจากที่ผู้คนบริเวณนั้นรู้สึกว่าจะเกิดแผ่นดินไหวเป็นเวลาหลายชั่วโมง

คลื่นโดยทั่วไปจะมีคุณสมบัติสำคัญที่วัดได้อยู่สองประการคือ คาบ ซึ่งจะเป็นเวลาระหว่างลูกคลื่นสองลูก และความยาวคลื่น ซึ่งเป็นระยะห่างระหว่างลูกคลื่นสองลูก ในทะเลเปิด คลื่นสึนามิมีคาบที่นานมาก โดยเริ่มจากไม่กี่นาทีไปจนเป็นชั่วโมง ในขณะที่เดียวกันก็มีความยาวคลื่นที่ยาวมาก โดยอาจยาวถึงหลายร้อยกิโลเมตร ในขณะที่คลื่นทั่วไปที่เกิดจากลมที่ชายฝั่งนั้นมีคาบประมาณ  $10$  วินาที และมีความยาวคลื่นประมาณ  $150$  เมตรเท่านั้น ความสูงของคลื่นในทะเลเปิดมักน้อยกว่าหนึ่งเมตร ซึ่งทำให้ไม่เป็นที่สังเกตของผู้คนบนเรือ คลื่นสึนามิจะเคลื่อนที่ด้วยความเร็วตั้งแต่  $500$  ถึง  $1,000$  กิโลเมตรต่อชั่วโมง อย่างไรก็ตาม เมื่อเข้าสู่ชายฝั่งที่มีความลึกลดลง คลื่นจะมีความเร็วลดลงและเริ่มก่อตัวเป็นคลื่นสูง โดยอาจมีความสูงมากกว่า  $30$  เมตร

เมื่อวันที่ 26 ธันวาคม 2547 เวลา 0:58:53 (GTM) ได้เกิดแผ่นดินไหวครั้งใหญ่ด้านชายฝั่งตะวันตก ทางตอนเหนือของเกาะสุมาตรา ประเทศอินโดนีเซีย มีความรุนแรงขนาด 9.1-9.3 ริกเตอร์ จนเป็นสาเหตุที่ทำให้เกิดคลื่นสึนามิในมหาสมุทรอินเดีย ทำให้ประเทศไทยได้รับความเสียหายจากการเกิดคลื่นสึนามิครั้งนี้อย่างมาก

งานวิจัยที่ผ่านมา ณรงค์ฤทธิ์ แก้วบรรจกร [1] ได้ทำการจำลองเชิงตัวเลขโดยใช้แบบจำลอง GEOWAVE ซึ่งประกอบด้วย 2 ส่วน คือ ส่วนของการเกิดแผ่นดินไหว และส่วนของการแผ่ตัวของคลื่น ในส่วนของการจำลองเชิงตัวเลขในส่วนของการเกิดแผ่นดินไหวใช้วิธีการของ Okada[2] และผลที่ได้จะนำไปใช้เป็นตัวเริ่มต้นของการคำนวณการแผ่ตัวของคลื่นที่สอดคล้องกับสมการ Fully Nonlinear Boussinesq อันดับสอง(ภาคผนวก ก.) ในการตรวจสอบความถูกต้องของแหล่งกำเนิดสึนามิ ใช้ข้อมูลระดับน้ำที่วัดได้ในช่วงเวลาที่เกิดคลื่นสึนามิจริง จากสถานีวัดระดับน้ำขึ้น-น้ำลงตามแนวชายฝั่งมหาสมุทรอินเดีย และข้อมูลที่ถูกบันทึกโดย Jason1 altimeter [3] นอกจากนี้แล้วได้เปรียบเทียบความสูงของคลื่นตามแนวชายฝั่งที่ได้จากการจำลองเชิงตัวเลขกับข้อมูลที่ได้จากการสังเกตตามแนวชายฝั่งทะเลอันดามันประเทศไทย เป็นกรณีศึกษาด้วย

## 1.2 วัตถุประสงค์ของการวิจัย

เพื่อจำลองการเกิดสึนามิในรูปแบบสามมิติ จากข้อมูลเชิงตัวเลขขนาดใหญ่ที่ใกล้เคียงกับการเกิดสึนามิ เมื่อวันที่ 26 ธันวาคม 2547 เนื่องจากได้มีการนำข้อมูลที่ได้จากการจำลองเชิงตัวเลขไปตรวจสอบความถูกต้อง โดยเปรียบเทียบข้อมูลระดับน้ำทะเลจากสถานีวัดระดับน้ำขึ้น-น้ำลงตามแนวชายฝั่งมหาสมุทรอินเดีย และข้อมูลที่ถูกบันทึกโดย Jason1 altimeter [3] เพื่อแสดงลักษณะของสึนามิที่เกิดขึ้น

วิทยานิพนธ์ฉบับนี้ นำเสนอการจำลองสึนามิจากข้อมูลเชิงตัวเลขซึ่งเป็นงานของ ณรงค์ฤทธิ์[1] มาสร้างภาพเคลื่อนไหวสามมิติ โดยมีการให้แสงและเงา เพื่อดูลักษณะการเกิดของคลื่น ได้ชัดเจนในหลายมุมมอง ทั้งยังสามารถย่อ-ขยาย หรือหยุดภาพเคลื่อนไหว เพื่อทำการศึกษาลักษณะของคลื่น ณ ตำแหน่งหรือเวลาที่ต้องการ

ปัญหาของงานวิจัยนี้ คือเวลาในการประมวลผลข้อมูลเชิงตัวเลขที่มีขนาดใหญ่ เมื่อนำมาจำลองสึนามิสามมิติ จำเป็นต้องใช้เวลานานมาก เนื่องจากการอ่านข้อมูลจากแฟ้มมาเก็บไว้ในบัฟเฟอร์ ซึ่งเป็นอุปสรรคในการจำลอง ดังนั้นงานวิจัยนี้จึงได้นำเสนอเทคนิคที่ช่วยลดเวลาในการอ่านข้อมูลและการประมวลผล โดยทำการแปลงข้อมูลเลขจำนวนจริง เป็นข้อมูลเลขลักษณะฐานสองก่อนแล้วค่อยนำมาเข้ากระบวนการให้แสงและเงา และได้ทำการเพิ่มความละเอียดของภาพในส่วน

ที่ต้องการแสดง และการเฟรมภาพเพื่อให้ภาพเคลื่อนไหวมีความต่อเนื่องมากยิ่งขึ้น โดยใช้ การประมาณค่าในช่วงเสมือนพหุนาม (Spline Interpolation)

### 1.3 ขอบเขตของการวิจัย

1. จำลองบริเวณมหาสมุทรอินเดีย พิกัด : ลองติจูดที่  $72^{\circ}$  E ถึง  $102^{\circ}$  E และ ละติจูดที่  $13^{\circ}$  S ถึง  $23.5^{\circ}$  N ความละเอียดของภาพ  $1793 \times 2193$  จุดที่ทำการคำนวณ (grid point)
2. จำลองบริเวณชายฝั่งทะเลภาคใต้ของประเทศไทย พิกัด : ลองติจูดที่  $97.977303^{\circ}$  E ถึง  $100.32409^{\circ}$  E และละติจูดที่  $6.07916^{\circ}$  N ถึง  $10.875^{\circ}$  N ความละเอียดของภาพ  $560 \times 1152$  จุดที่ทำการคำนวณ
3. จำลองบริเวณจังหวัดพังงาของประเทศไทย พิกัด : ลองติจูดที่  $97.99829^{\circ}$  E ถึง  $98.60283^{\circ}$  E และละติจูดที่  $8.6^{\circ}$  N ถึง  $9.2^{\circ}$  N ความละเอียดของภาพ  $145 \times 145$  จุดที่ทำการคำนวณ
4. โปรแกรมแสดงฟังก์ชัน การหมุนภาพ (rotation) การหยุดภาพ (pause) การเดินภาพ(play) การเคลื่อนย้ายภาพ (move) และการขยาย-ย่อภาพ (zoom in-zoom out)
5. มีการปรับสเกลของความสูงของพื้นดิน เนื่องจากหน้าจอมีขนาดจำกัด และทำให้ภาพดูต่างกันมากเมื่อเปรียบเทียบกับความสูงของน้ำถ้าใช้สเกลจริง
6. การให้สี แสง และเงา แบ่งตามลำดับความสูง

### 1.4 ระเบียบวิธีวิจัย

การจำลองการเกิดคลื่นสึนามิ เมื่อวันที่ 26 ธันวาคม 2547 ส่วนมากจะทำการสร้างภาพจากการกำหนดความสูงของคลื่นขึ้นมาสร้างเป็นภาพเคลื่อนไหวสองมิติหรือสามมิติ ณรงค์ฤทธิ์[1] ได้ทำการจำลองเชิงตัวเลขแล้วทำการเปรียบเทียบจากข้อมูลจริงที่วัดได้จากระดับน้ำที่วัดได้ในช่วงเวลาที่เกิดคลื่นสึนามิจริงจากสถานีวัดระดับน้ำขึ้น-น้ำลงตามแนวชายฝั่งมหาสมุทรอินเดีย และข้อมูลที่ถูกรับบันทึกโดย Jason1 altimeter [3] ซึ่งได้ข้อมูลที่ขนาดใหญ่มา

การศึกษาลักษณะโครงสร้างของข้อมูลที่ได้จากการจำลองเชิงตัวเลข นำมาสร้างภาพสามมิติ โดยการอ่านข้อมูลจากแฟ้มข้อมูลมาจัดเก็บในบัฟเฟอร์ ก่อนนำมาทำการให้แสงและเงาแบบสามมิติ (3D rendering) แต่เนื่องจากข้อมูลเชิงตัวเลขที่นำมาสร้างภาพนั้นมีขนาดใหญ่มาก จะทำให้เกิดปัญหาในการอ่านข้อมูลมาจัดเก็บในบัฟเฟอร์ใช้เวลานานมาก

ดังนั้นในงานวิจัยนี้จึงได้ทำการแปลงข้อมูลเชิงตัวเลขเป็นเลขลักษณะฐานสอง จัดเก็บในแฟ้มข้อมูลใหม่ ก่อนที่จะอ่านข้อมูลจากแฟ้มข้อมูลใหม่มาจัดเก็บในบัฟเฟอร์ แล้วนำมาทำการให้แสงและเงาแบบสามมิติ ขณะเดียวกันเมื่อทำการขยายภาพเข้าไป (zoom-in) จะทำการตัดข้อมูลส่วนที่ต้องการแสดง ทำให้ความละเอียดของภาพนั้น น้อยเกินไป จึงได้ทำการเพิ่มความละเอียดของข้อมูล และเพิ่มจำนวนเฟรมภาพเพื่อให้ภาพมีความต่อเนื่องมากขึ้น โดยใช้การประมาณค่าในช่วงเสมือนพหุนาม

ผลที่ได้ คือ ภาพการเกิดสึนามิวันที่ 26 ธันวาคม 2547 ในรูปแบบสามมิติที่มีฟังก์ชันในการแสดงผลต่างๆ อาทิ การหยุดภาพ การเดินภาพ การหมุนภาพ การขยาย-ย่อภาพ

โดยในบทที่ 2 จะอธิบายที่มาและความสำคัญ ความหมาย แนวคิดและทฤษฎี ในกระบวนการสร้างภาพสามมิติ โดยหัวข้อหลัก คือการให้แสงและเงา และการดำเนินการระดับบิต รวมถึงการประมาณค่าในช่วงเสมือนพหุนาม เพื่อใช้ในการเพิ่มข้อมูลให้มีความละเอียดมากยิ่งขึ้น ในส่วนที่เราต้องการแสดงผลภาพ ซึ่งมีความสำคัญในงานวิจัยนี้

บทที่ 3 อธิบายวิธีดำเนินการวิจัย ซึ่งจะแบ่งเป็น 2 ส่วนหลักๆ คือ ส่วนที่ 1 ทำการแปลงข้อมูลเชิงตัวเลขที่ได้จากผลลัพธ์เป็นเลขลักษณะฐานสอง รวมถึงประมาณค่าในช่วงเสมือนพหุนาม เพื่อเพิ่มข้อมูลของภาพและเพิ่มจำนวนเฟรมภาพบริเวณที่เราต้องการ และส่วนที่ 2 นำข้อมูลที่ได้ทำการแปลงข้อมูลแล้ว มาสร้างภาพสามมิติ โดยใช้กระบวนการให้แสงและเงา ตลอดจนอธิบายฟังก์ชันที่ใช้ในการแสดงผลต่างๆ

บทที่ 4 อธิบายผลการประมวลผล และวิเคราะห์ผลที่ได้ เมื่อการเปรียบเทียบผลที่ได้กับการประมวลแบบทั่วไป ตลอดจนแสดงผลจากการเพิ่มความละเอียดและเฟรมภาพ โดยผลการทดลองจะเลือกบริเวณจังหวัดพังงาเป็นกรณีศึกษา รวมถึงการปรับสเกลความสูงของพื้นดิน เพื่อให้ได้ภาพที่เหมาะสม

บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ



## บทที่ 2

### เอกสารและงานวิจัยที่เกี่ยวข้อง

#### 2.1 กล่าวนำ

ในบทนี้จะกล่าวถึง พื้นฐานของการสร้างภาพสามมิติ ซึ่งประกอบด้วยที่มาและความสำคัญ ความหมาย ทฤษฎีต่างๆที่ใช้ในงานวิจัยและระเบียบขั้นตอนวิธีในการให้แสงและเงา ในทางคอมพิวเตอร์กราฟิก สักระบวนการสร้างภาพจากแบบจำลองเริ่มเป็นประเด็นสำคัญในการศึกษาดังแต่ยุค 1970 เนื่องจากความซับซ้อนของเรขาคณิตคอมพิวเตอร์ โดยกระบวนการนี้มีความสำคัญในแง่ต่าง ๆ เช่น คอมพิวเตอร์เกม การจำลอง เทคนิคพิเศษทางภาพยนตร์หรือโทรทัศน์ และการออกแบบการสร้างภาพมโนทัศน์ ส่วนซอฟต์แวร์สำเร็จรูปบ้างก็รวมเข้ากระบวนการนี้กับซอฟต์แวร์สำหรับสร้างแบบจำลอง และสร้าง ภาพเคลื่อนไหว บ้างก็แยกเป็นซอฟต์แวร์เฉพาะต่างหาก อีกทั้งยังสามารถหาได้ในรูปแบบของซอฟต์แวร์ไม่จำกัดลิขสิทธิ์ ซึ่งขั้นตอนของการสร้างตัวเรนเดอร์ หรือตัวสร้างภาพจากแบบจำลองนั้นอาศัยการรวมศาสตร์ต่าง ๆ หลายแขนง เช่น ฟิสิกส์ของแสง การรับรู้ด้านการมองเห็น คณิตศาสตร์ และวิศวกรรมซอฟต์แวร์ กรณีของกราฟิกส์สามมิติ การสร้างภาพจากแบบจำลองนั้นเป็นกระบวนการที่ทำได้ช้า และกินเวลาในการคำนวณมาก (เช่นขั้นตอนการสร้างภาพยนตร์) ซึ่งสามารถใช้กราฟิกส์ฮาร์ดแวร์เร่งความเร็วการประมวลผลสามมิติแบบทันทีกาลเข้าช่วยได้

การจำลองคลื่นมีหลายลักษณะส่วนใหญ่จะใช้สมการน้ำตื้น (Shallow-Water Waves)[5][6] มาจำลองคลื่น เพื่อให้เห็นภาพการเคลื่อนไหวของคลื่นในลักษณะต่างๆ

การเกิดสึนามิ เมื่อวันที่ 26 ธันวาคม 2547 เวลา 0:58:53 (GTM) เกิดจากการที่แผ่นดินไหวครั้งใหญ่ด้านชายฝั่งตะวันตก ทางตอนเหนือของเกาะสุมาตรา ประเทศอินโดนีเซีย มีความรุนแรงขนาด 9.1-9.3 ริกเตอร์ จนเป็นสาเหตุที่ทำให้เกิดสึนามิในมหาสมุทรอินเดีย เพื่อให้เห็นกระบวนการเกิดสึนามิในครั้งนี้ จึงได้ทำการจำลองภาพในลักษณะสามมิติ เนื่องจากงานก่อนหน้านั้น[1] ทำการแสดงในลักษณะสองมิติ โดยเราจะทำการจำลองโดยใช้ข้อมูลที่ได้จาก แบบจำลอง GEOWAVE ซึ่งประกอบด้วย 2 ส่วน คือ ส่วนของการเกิดแผ่นดินไหว และส่วนของการแผ่ตัวของคลื่น การจำลองเชิงตัวเลขในส่วนของแผ่นดินไหวใช้วิธีการของ Okada[2] และผลที่ได้จะนำไปใช้เป็นค่าเริ่มต้นของการคำนวณการแผ่ตัวของคลื่นที่สอดคล้องกับสมการ Fully Nonlinear Boussinesq อันดับสอง

## 2.2 งานวิจัยที่เกี่ยวข้อง

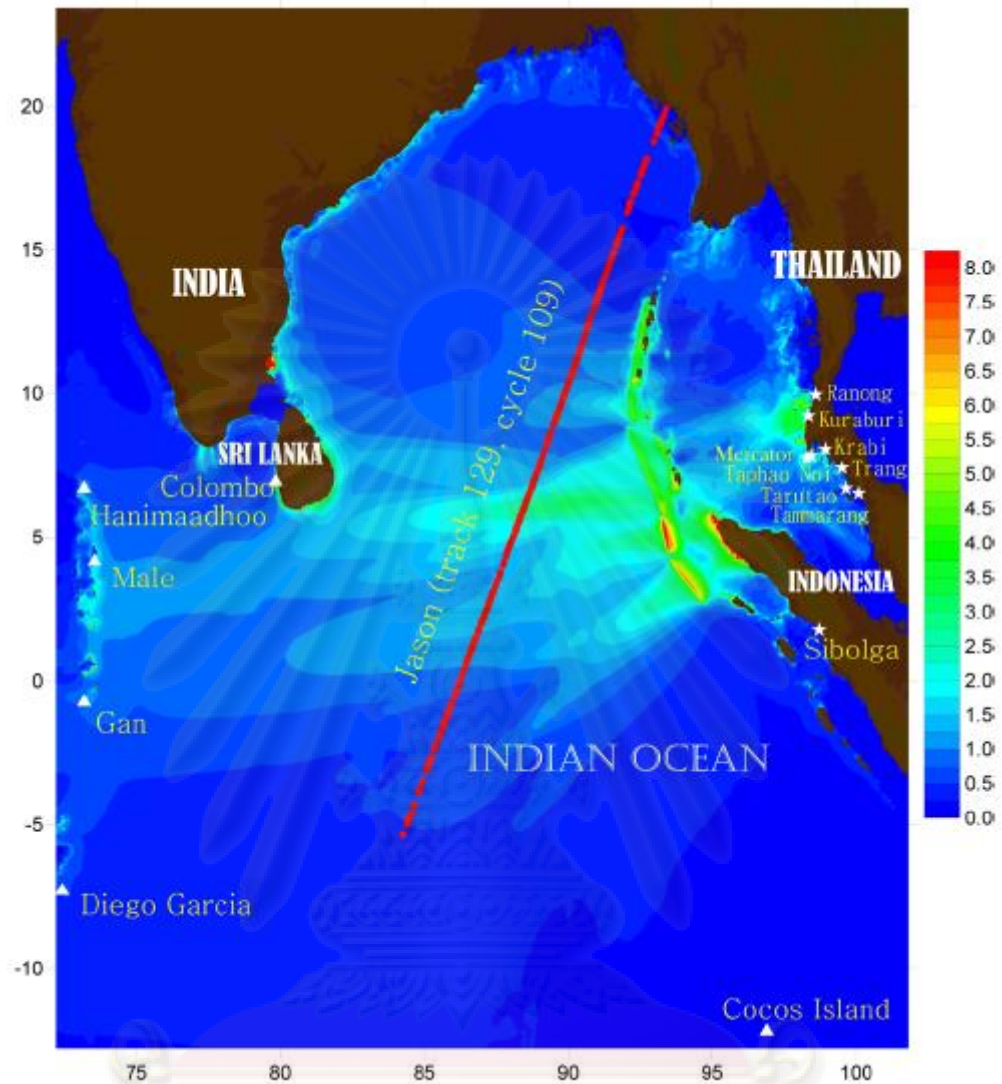
ในกระบวนการสร้างภาพสามมิติด้วยคอมพิวเตอร์ ( digital image synthesis) นั้นสามารถแบ่งเป็นกระบวนการหลักๆ ตามลักษณะของงานได้ 2 ประเภท คือการสร้างแบบจำลอง (modeling) และการให้แสงและเงา (Rendering)

Michael Kass และ Gavin Miller [4] เสนอแบบจำลองของน้ำ โดยในการสร้างพื้นผิวน้ำ อธิบายด้วย height-field ซึ่งคือการสร้างภาพสองมิติของความสูง(2D grid of heights) เพื่อจะทำให้การคำนวณได้รวดเร็วกว่าการใช้ภาพสามมิติโดยตรง( full three dimensional model) โดยส่วนใหญ่การสร้างภาพสามมิติ จะใช้เทคนิคการสร้างภาพสองมิติของความสูงนี้ ซึ่งเป็นวิธีที่ใช้กันอย่างแพร่หลาย

งานวิจัยที่เกี่ยวข้องกับการสร้างภาพคลื่นสามมิติ [5][6][7] โดยสร้างแบบจำลองจากสมการคลื่น รวมถึงเทคนิคต่างๆที่นำมาใช้ในการดำเนินการสร้างภาพสามมิติ(Implementation) และงานวิจัยที่เกี่ยวข้องกับการให้แสงและเงา[8][9][10] ในรูปแบบต่างๆ

Gower[3] ได้ทำการจำลองภาพคลื่นสึนามิเมื่อ วันที่ 26 ธันวาคม 2547 แสดงในรูปแบบสองมิติ โดยใช้สีในการแสดงความสูงของระดับน้ำทะเลที่แตกต่างกันออกไป

ในงานวิจัยของเราจะทำสร้างภาพเคลื่อนไหวสามมิติของคลื่นสึนามิ โดยการให้แสงและเงา จากการจำลองเชิงตัวเลข ของ ณรงค์ฤทธิ์ [1] ซึ่งใช้แบบจำลอง GEOWAVE ซึ่งประกอบด้วย 2 ส่วน คือ ส่วนของการเกิดแผ่นดินไหว และส่วนของการแผ่ตัวของคลื่น การจำลองเชิงตัวเลขในส่วนของการเกิดแผ่นดินไหวใช้วิธีการของ Okada[2] และผลที่ได้จะนำไปใช้เป็นค่าเริ่มต้นของการคำนวณการแผ่ตัวของคลื่นที่สอดคล้องกับสมการ Fully Nonlinear Boussinesq อันดับสอง (ภาคผนวก ก.) ในการตรวจสอบความถูกต้องของแหล่งกำเนิดสึนามิ ใช้ข้อมูลระดับน้ำที่วัดได้ในช่วงเวลาที่เกิดคลื่นสึนามิจริง จากสถานีวัดระดับน้ำขึ้น - น้ำลงตามแนวชายฝั่งมหาสมุทรอินเดีย นอกจากนั้นแล้วได้เปรียบเทียบความสูงของคลื่นตามแนวชายฝั่งที่ได้จากการจำลองเชิงตัวเลขกับข้อมูลที่ได้จากการสังเกตตามแนวชายฝั่งทะเลอันดามันประเทศไทย เป็นกรณีศึกษาด้วย มาทำการแปลงข้อมูลเป็นเลขลักษณะฐานสองก่อน เพื่อลดเวลาในการอ่านข้อมูลจากแฟ้มข้อมูล ช่วยให้การประมวลผลได้เร็วขึ้น และได้ทำการประมาณค่าในช่วงเสมือนพหุนามในช่วงที่เราต้องการ เพื่อเพิ่มความละเอียดของภาพและเพิ่มจำนวนเฟรมภาพ



รูปที่ 2.1 แสดงภาพการจำลองในรูปแบบสองมิติ  
(ที่มาจาก : Jason1 altimeter track 129, cycle 109)

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

## 2.3 ทฤษฎี

### 2.3.1 การให้แสงและเงา (Rendering)

การให้แสงและเงา (rendering) [11][12][13] คือการประมวลผลเพื่อสร้างภาพสองมิติจากข้อมูลรายละเอียดฉาก (scene description) ที่อยู่ในพีคัดสามมิติ โดยภาพที่ได้จากการให้แสงและเงานี้จะเป็นภาพดิจิทัล (digital image) แบบภาพแรสเตอร์ (raster image) ซึ่งมีความละเอียดของภาพ (resolution) และ ความลึกของสี (color depth) ที่คงที่ค่าหนึ่ง

โดยที่ความละเอียดของภาพ นั้นเป็นค่าที่บอกถึงขนาดของภาพว่ามีขนาดใหญ่หรือเล็กมากน้อยเพียงใด ส่วนความลึกของสีนั้นก็คือความละเอียดของภาพเช่นเดียวกัน แต่เป็นความละเอียดของภาพในเรื่องของสี โดยจะวัดจากขนาดข้อมูลที่ใช้เก็บค่าสีมีหน่วยเป็นบิต (bit) ข้อมูลสีดังกล่าวสามารถที่จะมีได้หลายช่องสัญญาณ ตั้งแต่ 1 ช่องสัญญาณ ขึ้นไป สำหรับมาตรฐานในปัจจุบันจะประกอบไปด้วย 3 ช่องสัญญาณ ได้แก่ R, G และ B ตามลำดับ โดย R คือสีแดง G คือสีเขียว และ B คือสีฟ้า

ดังนั้นจุดมุ่งหมายของการให้แสงและเงา ก็คือการคำนวณเพื่อให้ได้มาซึ่งข้อมูลที่ประกอบกันเป็นภาพหนึ่งๆ นั้นเอง เพียงแต่แตกต่างจากการสร้างภาพแบบอื่น ตรงที่เป็นการสร้างจากข้อมูลของวัตถุในพีคัดสามมิติที่สร้างโดยใช้คอมพิวเตอร์ ซึ่งการให้แสงและเงาเองนั้นมีการบวนการสลับซับซ้อนมาก แต่มีหลักการโดยรวมที่เหมือนกับการสร้างภาพดิจิทัลอื่นๆ

ขั้นตอนสำคัญของการ ให้แสงและเงา จะทำการคำนวณค่าสี และความสว่าง ณ พีคัดต่าง ๆ บนแบบจำลอง โดยใช้หลักการคำนวณหาจุดตกกระทบของแสงจากแหล่งกำเนิด หลังจากนั้นจึงเข้าสู่กระบวนการสร้างให้เป็นจุดภาพ หรือแรสเตอร์ไรเซชัน (Rasterization) ซึ่งจะเป็นการฉาย (Project) แบบจำลองลงบนระนาบสองมิติ ก่อนส่งค่าไปแสดงบนจอภาพ กระบวนการนี้เป็นขั้นตอนสุดท้ายในสายทอกราฟิกส์ หรือกราฟิกส์ไปป์ไลน์ (Graphics pipeline) โดยจะให้ผลลัพธ์เป็นการแสดงรูปหรือการเคลื่อนไหวที่สร้างขึ้น

การส่องสว่างวงกว้าง (Global illumination) เป็นชื่อเรียกรวมต่างๆ ไปของขั้นตอนวิธีที่ใช้ในการให้แสงและเงาในคอมพิวเตอร์กราฟิกส์สามมิติเพื่อเพิ่มความสมจริงของแสงเงาให้กับแบบ ขั้นตอนวิธีเหล่านี้จะไม่มองแค่แสงจะต้องเดินทางมาจากต้นกำเนิดแสงโดยตรง แต่จะมองว่าแสงซึ่งมาจากต้นกำเนิดนั้นๆ อาจสะท้อนมาจากพื้นผิวของวัตถุอื่นๆ ที่อยู่ในาฉากด้วย

ในทางทฤษฎีการสะท้อน การหักเหและเงา ล้วนแล้วแต่เป็นตัวอย่างของการส่องสว่างวงกว้าง เพราะในการจำลองวัตถุหนึ่งๆ จะมีผลต่อการให้แสงและเงาของวัตถุอื่นๆ ไม่ใช่เฉพาะผลจากการกระทบของแสงที่มาจากต้นกำเนิดแสงโดยตรงเท่านั้น อย่างไรก็ตามในทางปฏิบัติ

จะนับเพียงการจำลองการสะท้อนแสงแบบฟังก์กระจายและการสะท้อนหรือหักเหของแสงจากพื้นผิวโค้งเท่านั้น

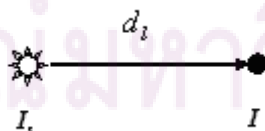
เรามักพบว่าภาพที่ถูกให้แสงและเงาด้วยกระบวนการแบบการส่องสว่างวงกว้าง มีความสมจริงมากกว่าภาพที่ให้แสงและเงาแบบคำนวณจากต้นกำเนิดแสงโดยตรงเท่านั้น อย่างไรก็ตามการคำนวณแสงในลักษณะนี้จะใช้กำลังในการประมวลผลสูงมาก และใช้เวลานานกว่าวิธีอื่น วิธีการหนึ่งซึ่งเป็นที่นิยมในการคำนวณการส่องสว่างวงกว้าง คือการคำนวณแล้วจัดเก็บผลที่ได้ลงในตัววัตถุ(เช่นในรูปของพอยต์คลาวด์) วิธีนี้เรียกว่า เรดิโอซิตี (radiosity) ซึ่งผลลัพธ์นี้สามารถนำไปใช้สร้างภาพจากมุมมองต่างๆ โดยจะถูกใช้เป็นแนวทางที่จะคำนวณแสงในฉากคร่าวๆเป็นการลดความซับซ้อนในการคำนวณแสงจริงๆซึ่งใช้เวลามาก

### 2.3.1.1 โมเดลการส่องสว่างวงกว้าง(Illumination models)

ในการสร้างภาพสามมิติด้วยคอมพิวเตอร์โดยใช้กระบวนการให้แสงและเงานั้นจะใช้หลักการที่จำลองมาจากปรากฏการณ์ที่เกิดขึ้นจริงในธรรมชาติ อันได้แก่ คุณสมบัติของแสงและคุณสมบัติของพื้นผิววัตถุ อย่างไรก็ตามคุณสมบัติในธรรมชาติเหล่านี้มักจะมีผลซับซ้อนมากเกินไปที่จะจำลองในการสร้างภาพด้วยคอมพิวเตอร์ได้ทั้งหมด แหล่งกำเนิดแสงเราไม่ได้สมมติวิธีที่ง่ายแต่เราเลือกการให้แสงตามธรรมชาติ ที่เกิดจากดวงอาทิตย์ซึ่งอยู่ระยะไกลทำให้แสงเป็นวงกว้าง ดังนั้นเราจะเลือกให้แสงแบบแสงล้อมรอบ(Ambient light) โดยแสงล้อมรอบได้พัฒนามาจากจุดกำเนิดแสง(Point light source)

จุดกำเนิดแสง (Point light source)

เป็นแหล่งกำเนิดแสงที่พื้นฐานที่สุด โดยเป็นแหล่งกำเนิดแสงที่ไม่มีขนาด กล่าวคือเป็นจุดใดๆ เพียงจุดเดียวในพิกัดสามมิติที่มีทิศทางของแสงกระจายออกไปโดยรอบ ตัวอย่างดังรูปที่ 2.2 ที่แหล่งกำเนิดแสงเป็นแบบจุดกำเนิดแสงที่มีความเข้ม



รูปที่ 2.2 แสดงค่าต่างๆ ที่ใช้ในการคำนวณหาปริมาณแสงที่ได้จากแหล่งกำเนิดแสงชนิดจุดกำเนิดแสง

เราสามารถที่จะทำการหาความเข้มของแสง ที่ตำแหน่งใดได้จากสมการ(2.1)

$$I = \frac{1}{d_l} I_s \quad (2.1)$$

โดยที่

$I$	ความเข้มของแสงของแหล่งกำเนิดแสงที่จุดใดๆ ห่างออกไปเป็นระยะ $L$
$I_s$	ความเข้มของแสง ที่เราทำการกำหนดให้กับแหล่งกำเนิดแสง
$d_l$	คือระยะทางจากแหล่งกำเนิดแสงถึงจุดที่ต้องการทราบความเข้มของแสง

จะเห็นว่าจุดกำเนิดแสงมีค่าอัตราส่วนการลดทอน( attenuation factor) ( $f$ ) เท่ากับ  $1/d_l^2$  ซึ่งเป็นไปตามกฎการส่องสว่างที่มีลักษณะเป็นผกผันกำลังสอง จะทำให้วัดความสว่างมากเมื่ออยู่ใกล้แหล่งกำเนิดแสง และมีความสว่างลดลงอย่างรวดเร็วเมื่ออยู่ห่างออกไประยะหนึ่ง และความสว่างจะค่อยๆ ลดลงทีละน้อยเมื่ออยู่ห่างออกไป ดังนั้นจึงจึงมีการคิดค้นลักษณะการเสื่อมลดของแสงที่มีความเหมาะสมมากกว่า เช่น การเสื่อมลดที่มีลักษณะเป็นเชิงเส้น หรือแบบอื่นๆ ที่สลับซับซ้อนกว่า ตามสมการ 2.2

$$f = \frac{1}{a_0 + a_1 d_l + a_2 d_l^2} \quad (2.2)$$

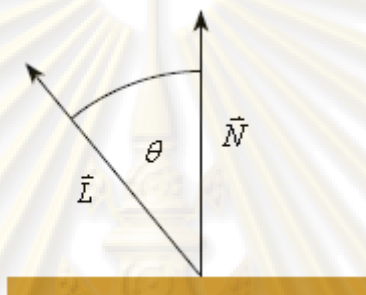
เราไม่สามารถคำนวณสมการ2.2 ที่อนันต์ได้ เพราะระยะทางไปจุดกำเนิดแสงไม่ชัดเจน ดังนั้นทุกๆจุดในฉากจึงมีระยะทางเท่ากัน จึงปรับทั้งจุดกำเนิดที่ไกลและใกล้ เราสามารถกำหนดฟังก์ชันการเสื่อมลดของแสง ดังสมการ(2.3)[13]

$$f = \begin{cases} 1.0 & , \text{จุดกำเนิดแสงที่ระยะอนันต์} \\ \frac{1}{a_0 + a_1 d_l + a_2 d_l^2} & , \text{จุดกำเนิดแสงที่ระยะใดๆ} \end{cases} \quad (2.3)$$

โดยที่ค่า  $a$  คือค่าคงที่ใดๆ ที่ถูกกำหนดขึ้นเพื่อให้เกิดลักษณะของการเสื่อมลดที่แตกต่างกันออกไป

การสะท้อนพื้นผิวแบบกระจาย (diffuse reflection) ในอุดมคติจะสะท้อนแสงตามกฎโคไซน์ของแลมเบิร์ต (Lambert's cosine law) ที่ว่าพื้นผิวของวัตถุสะท้อนแสงออกไปรอบตัวมันในปริมาณที่เท่ากันทุกทิศทาง จากเหตุผลดังกล่าวนี้เองจึงทำให้ ปริมาณแสงที่สะท้อนจะไม่ขึ้นกับทิศทางการมอง แต่จะขึ้นอยู่กับมุมของการตกกระทบของแสงเท่านั้น

เนื่องจากมุมตกกระทบนี้ขึ้นอยู่กับเวกเตอร์บรรทัดฐาน (normal vector) หรือ  $\vec{N}$  และทิศทางของแสง  $\vec{L}$  แสดงดังรูปที่ 2.3 ดังนั้นเราจะได้โมเดลการส่องสว่างวงกว้าง (illumination model) ดังสมการ(2.2)



รูปที่ 2.3 แสดงมุมตกกระทบ  $\theta$  ของแสงระหว่างเวกเตอร์  $\vec{L}$  กับเวกเตอร์บรรทัดฐาน  $\vec{N}$  ที่พื้นผิว

$$I_{l,diff} = k_d I_l (\vec{N} \cdot \vec{L}) \quad (2.4)$$

โดยที่

$I_{l,diff}$  คือ การสะท้อนพื้นผิวแบบกระจาย

$k_d$  คือ ค่าคงตัวที่ใช้กำหนดคุณสมบัติการสะท้อนแสงแบบกระจายให้กับพื้นผิวของวัตถุ

$I_l$  คือ ความเข้มของแหล่งกำเนิดแสงแบบจุด

$\vec{N} \cdot \vec{L} = \cos\theta$ , เมื่อ  $\theta$  คือ มุมที่แสงตกกระทบกับพื้นผิวของวัตถุ

ศูนย์วิทยุทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

### แสงล้อมรอบ (Ambient light)

แสงล้อมรอบ เป็นแหล่งกำเนิดแสง ที่ถูกสร้างขึ้น ซึ่งมีลักษณะเป็นแสงที่เป็นวงกว้างจากระยะไกล เพื่อทดแทนปริมาณแสงที่ได้จากการส่องสว่างวงกว้าง เนื่องจากการคำนวณหาปริมาณแสงที่ได้จากการส่องสว่างวงกว้าง นั้นใช้เวลามาก ดังนั้นจึงได้มีการนำวิธีอื่นที่พอจะทดแทนได้มาใช้ โดยวิธีการที่ดีที่สุดในแง่ของความเร็วในการคำนวณนั่นก็คือการเพิ่มปริมาณแสงให้กับทุกๆ จุดบนพื้นผิวของวัตถุด้วยปริมาณแสงที่เท่ากัน ซึ่งปริมาณแสงที่ได้ก่อนข้างจะมีความแตกต่างจากปริมาณแสงที่ได้จากการส่องสว่างวงกว้างที่แท้จริงอยู่มาก

แพ็กเกจกราฟฟิกส์ส่วนมากจะแนะนำ ค่าสัมประสิทธิ์การสะท้อนของแสงล้อมรอบ (ambient reflection coefficient)  $k_a$  ซึ่งสามารถกำหนดค่าความเข้มของแสงล้อมรอบ (ambient-light intensity)  $I_a$  ในแต่ละพื้นผิวได้ ดังสมการ(2.5)

$$I = k_a I_a \quad (2.5)$$

โดยที่

$I$  คือ ความเข้มของแสงของแหล่งกำเนิดแสงที่จุดใดๆ

$I_a$  คือ ความเข้มของแสงล้อมรอบ

$k_a$  คือ ค่าสัมประสิทธิ์ของการสะท้อนแสงล้อมรอบ ตั้งแต่ช่วง 0 ถึง 1

ในทางเพิ่มเติม ถ้าหากใช้ร่วมกันกับแหล่งกำเนิดแสงอื่นๆ แล้วก็จะให้ผลที่ดีกว่าการใช้แหล่งกำเนิดแสงตัวใดตัวหนึ่งเพียงลำพัง อย่างเช่นเราสามารถรวมแสงล้อมรอบกับจุดกำเนิดแสงมาคำนวณ เพื่อแสดงการสะท้อนพื้นผิวแบบกระจายทั้งหมดที่ตำแหน่งของพื้นผิว เพื่อให้ง่ายเราก็จะเพิ่มพารามิเตอร์สำหรับปรับผลกระทบของแสงในแบบจำลองของเรา สามารถเขียนสมการการสะท้อนพื้นผิวแบบกระจายทั้งหมด สำหรับจุดกำเนิดแสงจุดเดียว ดังสมการ(2.6)

$$I = \begin{cases} k_a I_a + k_d I_l (\vec{N} \cdot \vec{L}), & \text{ถ้า } (\vec{N} \cdot \vec{L}) > 0 \\ k_a I_a, & \text{ถ้า } (\vec{N} \cdot \vec{L}) \leq 0 \end{cases} \quad (2.6)$$

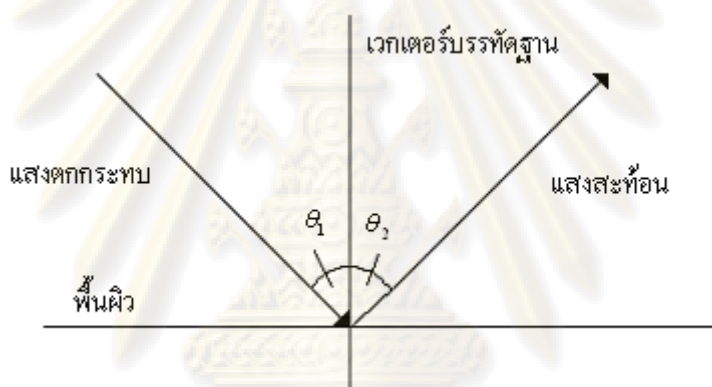
เมื่อทั้ง  $k_a$  และ  $k_d$  ขึ้นอยู่กับคุณสมบัติพื้นผิวของวัตถุ และค่าอยู่ในช่วง 0 ถึง 1.0



### 2.3.1.2 การสะท้อนและการหักเห (Reflection & Refraction)

เมื่อแสงตกกระทบบนวัตถุที่มีมันวาว เช่น กระจกเงา ก็จะทำให้เกิดการสะท้อนแสง และเมื่อแสงส่องผ่านวัตถุที่โปร่งใสในมุมที่ไม่ตั้งฉากก็จะมีหักเหเกิดขึ้น

- การสะท้อน จะเกิดขึ้นเมื่อแสงตกกระทบบนวัตถุที่มีความเรียบและสามารถสะท้อนแสงได้ โดยที่มุมตกกระทบบนของแสงและมุมที่แสงสะท้อนออกไปนั้นจะมีขนาดมุมที่เท่ากัน ตามกฎการสะท้อน โดยมุมทั้งสองดังกล่าวนี้เป็นการวัดมุมจากเวกเตอร์บรรทัดฐานไปยังเวกเตอร์ที่เป็นทิศทางของแสงที่ตกกระทบบนและจากเวกเตอร์บรรทัดฐานไปยังเวกเตอร์ที่เป็นทิศทางของแสงที่สะท้อนออกไป ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 แสดงการสะท้อนของแสงที่เป็นไปตามกฎของแสงโดยที่  $\theta_1 = \theta_2$

- การหักเห จะเกิดขึ้นเมื่อแสงเดินทางผ่านรอยต่อระหว่างวัตถุที่มีความหนาแน่นแตกต่างกัน จะทำให้ทิศทางของแสงเบี่ยงเบนไปจากแนวทางเดิม ถ้าหากทิศทางการตกกระทบบนนั้นไม่ตั้งฉากกับพื้นผิวของวัตถุ ดังแสดงในรูปที่ 2.5

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.5 แสดงการหักเหของแสงเมื่อมีการเดินทางผ่านตัวกลางที่มีความหนาแน่นไม่เท่ากัน

โดยมีความสัมพันธ์ต่างๆ เป็นไปตามกฎของ Snell [7][10] ที่ว่ามุมหักเหจะขึ้นอยู่กับมุมตกกระทบและดัชนีของการหักเหของวัตถุนั้น ดังสมการ (2.7)

$$n_2 \sin \theta_2 = n_1 \sin \theta_1 \quad (2.7)$$

โดยที่

$\theta_1$	คือ มุมที่แสงตกกระทบทำกับทิศทางที่ตั้งฉากกับรอยต่อระหว่างตัวกลางทั้งสองชนิด
$\theta_2$	คือ มุมที่แสงหักเหทำกับทิศทางที่ตั้งฉากกับรอยต่อระหว่างตัวกลางทั้งสองชนิด
$n_1$	ค่า IOR ของตัวกลางหรือวัตถุด้านที่แสงตกกระทบ
$n_2$	ค่า IOR ของตัวกลางหรือวัตถุด้านที่แสงเกิดการหักเห

### 2.3.1.3 เงา (Shadow)

ในการสร้างภาพสามมิติด้วยคอมพิวเตอร์นั้น แหล่งกำเนิดแสงจะสร้างแสงที่สามารถทะลุทะลวงวัตถุได้ เนื่องจากในขั้นตอนการคำนวณความสว่าง (illumination) นั้นเราไม่ได้คำนึงถึงว่ามีวัตถุมาบังแหล่งกำเนิดแสงหรือไม่และบังอย่างไร

ในธรรมชาติเงาไม่มีตัวตนแต่เงาเป็นคำที่ใช้เรียกบริเวณที่มีแสงในจุดนั้นๆ น้อยกว่าตำแหน่งอื่นๆ เนื่องจากมีวัตถุอื่นบังแหล่งกำเนิดแสงที่ส่องมายังตัวมัน เมื่อมีวัตถุอยู่หนึ่งอันและมีแหล่งกำเนิดแสงเพียงหนึ่งแหล่งจะทำให้เกิดเงาขึ้น 2 ชนิดคือ

- เงามืด(umbra) คือบริเวณที่ไม่ได้รับแสงจากแหล่งกำเนิดแสงโดยตรงเลย
- เงามัว(penumbra) เกิดจากบริเวณนั้นได้รับแสงไม่เต็มที่ เนื่องจากแหล่งกำเนิดแสง

ถูกบดบังบางส่วน

วิธีการสร้างเงานั้นจะขึ้นอยู่กับ ขั้นตอนวิธีการให้แสงและเงา ที่ใช้ด้วยว่า เอื้ออำนวยต่อการสร้างเงาด้วยแบบต่างๆ มากน้อยเพียงใด เพราะการสร้างเงานั้นจะอยู่ในขั้นตอน การให้แสงและเงา

### 2.3.1.4 วิธีการให้แสงและเงาโพลีกอน(Polygon rendering methods)

โมเดลการส่องสว่างวงกว้าง[8][9][13] เป็นการพิจารณาเพื่อคำนวณค่าระดับความสว่างที่ตำแหน่งใดตำแหน่งหนึ่งบนพื้นผิวเท่านั้น วิธีการดังกล่าวยังไม่ใช้วิธีการสร้างภาพทั้งหมด แต่เป็นกระบวนการย่อยส่วนหนึ่งของกระบวนการให้แสงและเงา

#### *Constant-Intensity Surface Rendering*

เป็นวิธีการที่ง่ายและเร็วที่สุดสำหรับการให้แสงและเงาพื้นผิวของโพลีกอน เนื่องจากทำการคำนวณเพียงครั้งเดียวเพื่อให้ได้ค่าระดับความสว่างทั้งหมดทั่วทั้งโพลีกอนชิ้นนั้นๆ โดยการคำนวณจาก surface normal vector ของแต่ละโพลีกอนซึ่งมีเพียงค่าเดียว เนื่องจากในการให้แสงและเงา นั้นจะใช้โพลีกอนที่มีรูปร่างสามเหลี่ยมเพียงอย่างเดียวจึงทำให้เป็นระนาบเพียงระนาบเดียวและมี surface normal vector ที่เท่ากันในทุกตำแหน่งบนโพลีกอนสามเหลี่ยมชิ้นนั้นๆ ซึ่งเราจะใช้ normal vector ดังกล่าวในการคำนวณเพื่อหาค่าระดับความสว่างจากนั้นเราก็จะเติมค่าระดับความสว่าง ที่ได้จากการคำนวณในแต่ละช่องสัญญาณของ RGB นี้ลงบนพื้นที่โพลีกอนชิ้นนั้นๆ

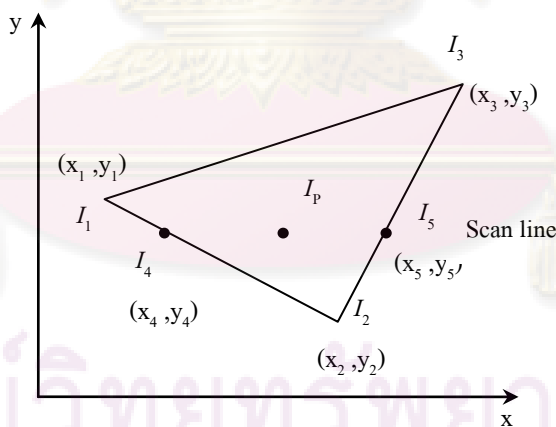
#### *Gouraud Surface Rendering*

พื้นผิวที่โค้งจะมีค่าระดับความสว่าง ของแต่ละพิกเซลต่างกันเนื่องจากความโค้งของผิวที่หักเหไปในทิศทางที่ไม่เท่ากัน รวมถึงสาเหตุอื่นๆ เช่น การที่แสงตกลงบนแต่ละจุดบนโ

ลิกอนด้วยค่าระดับความสว่างที่ไม่เท่ากัน เนื่องมาจากการเชื่อมลดตามคุณลักษณะของ แหล่งกำเนิดแสง ซึ่งจะเห็นว่าเราไม่สามารถทำการคำนวณค่าระดับความสว่างเพียงครั้งเดียวเพื่อให้เกิด ค่าระดับความสว่างที่แตกต่างกันทั่วทั้งโพลีกอนได้ เราจึงต้องคำนวณค่าระดับความสว่างของ พิกเซลบนโพลีกอนนั้นๆ ทุกพิกเซลจึงจะได้ค่าระดับความสว่างที่แตกต่างกัน

ในการคำนวณจริงๆ นั้นไม่สามารถทราบข้อมูลที่จำเป็น เช่น เวกเตอร์บรรทัดฐาน หรือข้อมูลอื่นๆ ที่ใช้ในการคำนวณในตำแหน่งอื่นๆ ที่นอกเหนือจากตำแหน่งจุดยอด (vertex) ของ โพลีกอนได้ เนื่องจากสำหรับโพลีกอนข้อมูลทุกอย่าง จะสามารถทราบได้จากตำแหน่งที่เป็นจุด ยอดของโพลีกอนเท่านั้น โดยภายในพื้นที่ที่วงรอบปิดของโพลีกอนนั้นเราไม่สามารถที่จะทราบได้ เลยว่าแท้จริงแล้ว เวกเตอร์บรรทัดฐานมีทิศทางไปในทิศทางใด มีลักษณะการสะท้อนแสงเป็นแบบ ไค ซึ่งปัญหาดังกล่าวสามารถที่จะแก้ได้ โดยใช้การประมาณค่าในช่วงจากตำแหน่งที่ทราบค่า จำเป็นในการคำนวณเหล่านั้น ซึ่งตำแหน่งดังกล่าวก็คือที่ จุดยอดของโพลีกอน

Gouraud[9][11][13] ได้เสนอให้ทำการเฉลี่ยค่าความสว่างที่ตำแหน่งจุดยอดของ โพลีกอน เพื่อทำการหาค่าความสว่างของตำแหน่งอื่นๆ ภายในโพลีกอน กล่าวโดยสรุปก็คือ จะทำ การคำนวณความสว่างด้วย อินลูมิเนชัน โมเดลที่ตำแหน่งของจุดยอดเพียงอย่างเดียว จากนั้นจึงทำ การเฉลี่ยค่าความสว่าง ดังกล่าวเพื่อเป็นความสว่างของตำแหน่งอื่นๆที่เหลือ



รูปที่ 2.6 แสดงภาพตัวอย่างของ Gouraud สำหรับโพลีกอน 1 อัน

จากรูปที่ 2.6 สมมุติว่า  $I_1$ ,  $I_2$  และ  $I_3$  เป็นค่าความสว่าง ที่ได้จากการคำนวณโดยใช้ อินลูมิเนชัน โมเดลใดๆ ที่มุมทั้งสามของโพลีกอนตามลำดับ โดยมี  $I_4$  และ  $I_5$  เป็นค่าความสว่าง บน ด้านของโพลีกอนที่ได้โดยการเฉลี่ยค่าจาก  $I_1$ ,  $I_2$  และ  $I_3$  ตามแนวขอบของโพลีกอนและมี  $I_p$  เป็นค่า ความสว่างที่จุดใดๆ บนโพลีกอนที่เราต้องการหา เมื่อเราพิจารณาที่เส้นตรวจสอบ (Scan line) นั้นๆ

จะตัดขอบของโพลีกอนจำนวนสองด้านทำให้เกิด  $I_4$  และ  $I_5$  เนื่องจากเส้นตรวจสอบ สามารถตัดขอบของโพลีกอนได้เพียงหนึ่งคู่เท่านั้น ซึ่งกรณีนี้เราสามารถที่จะทำการหา  $I_p$  ได้จาก

$$I_4 = I_1 + I_2 \frac{(y_4 - y_1)}{(y_2 - y_1)}$$

$$I_5 = I_2 + I_3 \frac{(y_5 - y_2)}{(y_3 - y_2)}$$

$$I_p = I_4 + I_5 \frac{(x - x_4)}{(x_5 - x_4)}$$

สาเหตุที่ใช้เส้นลากผ่านในแนวนอนนั้นเนื่องจากการ shading ใน image space ตามทิศทางการวางตัวของเส้นตรวจสอบ หลักการโดยสรุปก็คือเมื่อเส้นตรวจสอบลากผ่านโพลีกอนใดๆ ก็จะตัดขอบของโพลีกอนสองตำแหน่ง โดยที่ตำแหน่งตัดกันทั้งสองจุดนี้เราจะทำการประมาณค่าในช่วง ค่าความสว่างจากตำแหน่งจุดยอด ทั้งสองปลายของขอบโพลีกอนนั้นๆ เมื่อได้ค่าสี่ที่ตำแหน่งเส้นตรวจสอบ ตัดขอบของโพลีกอนทั้งสองแล้วทำการประมาณค่าในช่วง ค่าสีบนเส้นตรวจสอบ ซึ่งทำให้เราได้ค่าสีของโพลีกอนที่ตำแหน่งเส้นตรวจสอบทั้งหมดทุกพิกเซลในโพลีกอนนั้นๆ

### 2.3.1.5 ขั้นตอนวิธีซีบัฟเฟอร์ (Z-Buffer algorithm)

การพิจารณาพื้นผิวที่เห็น (visible surfaces) ได้ของขั้นตอนวิธีซีบัฟเฟอร์ [11][13] จะใช้การเปรียบเทียบค่าความลึก (Z-value) ซึ่งเป็นค่าที่วัดระยะจากตำแหน่งของกล้องไปยังจุดบนพื้นผิวที่กำลังต้องการพิจารณา ดังนั้นเราจึงต้องมีบัฟเฟอร์ (buffer) เพื่อทำหน้าที่เก็บค่าความลึกเหล่านี้ชั่วคราวเรียกว่า ซีบัฟเฟอร์ (Z-Buffer) ดังนั้นการให้แสงและเงา ด้วยวิธีนี้จึงประกอบไปด้วย บัฟเฟอร์ 2 ส่วน คือ เฟรมบัฟเฟอร์ (frame buffer) และ ซีบัฟเฟอร์

- เฟรมบัฟเฟอร์ ใช้เก็บข้อมูลระดับความสว่างของภาพซึ่งจะประกอบด้วย 3 ช่องสัญญาณ ได้แก่ R, G และ B หรืออีกนัยหนึ่ง คือ สีแดง สีเขียว และสีน้ำเงิน โดยที่เราสามารถกำหนดได้ว่าจะให้แต่ละช่องสัญญาณมีความละเอียดมากน้อยเพียงใด โดยมาตรฐานทั่วไปซึ่งไม่ต้องการความถูกต้องมากนักจะแบ่งเป็น 256 ระดับในแต่ละช่องสัญญาณหรือมีขนาดข้อมูลเท่ากับ 8 บิตในแต่ละช่องสัญญาณ

- ซีบัฟเฟอร์ มีลักษณะที่คล้ายกับเฟรมบัฟเฟอร์ โดยจะมีความละเอียด ที่เท่ากันกับเฟรมบัฟเฟอร์ เพียงแต่ไม่ได้ใช้เก็บข้อมูลสี แต่เป็นบัฟเฟอร์ที่ใช้สำหรับเก็บค่าความลึกของแต่ละจุดภาพ เป็นการชั่วคราวเพื่อประโยชน์ในการพิจารณาพื้นผิวที่เห็นได้ โดยที่ซีบัฟเฟอร์ นี้จะมีเพียงช่องสัญญาณเดียวสำหรับแต่ละจุดภาพ และจะใช้ข้อมูลชนิดเลขทศนิยม (floating point) ซึ่งมีขนาดเท่ากับ 32 หรือ 64 บิต การเก็บข้อมูลซีบัฟเฟอร์ โดยความละเอียดที่สูงนี้จะทำให้เราสามารถตรวจสอบค่าความลึก ได้อย่างละเอียดและถูกต้องมากยิ่งขึ้น

### ขั้นตอนของซีบัฟเฟอร์ (Z-Buffer algorithm) [13]

1. กำหนดค่าเริ่มต้น ซีบัฟเฟอร์ และเฟรมบัฟเฟอร์ ทุกๆตำแหน่งบัฟเฟอร์(x,y)

$$\text{depthBuff}(x,y) = 1.0 \quad , \quad \text{frameBuff}(x,y) = \text{backgndColor}$$

2. ดำเนินการสร้างแต่ละโพลีกอนในฉาก(scene) 1 ครั้ง

- สำหรับแต่ละตำแหน่งจุดภาพ (x,y) ของโพลีกอน คำนวณค่าความลึก z (ถ้าไม่ทราบค่ามาก่อน)
- ถ้า  $z < \text{depthBuff}(x,y)$  แล้วคำนวณสีพื้นผิวที่ตำแหน่งนั้นและเซตค่า

$$\text{depthBuff}(x,y) = z \quad , \quad \text{frameBuff}(x,y) = \text{surfColor}(x,y)$$

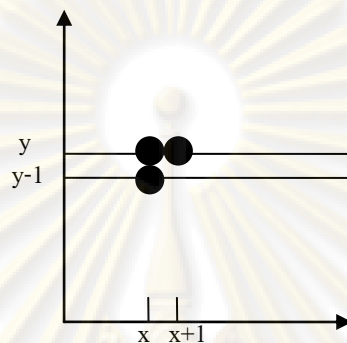
ภายหลังพื้นผิวจะถูกดำเนินการทั้งหมด ซึ่งซีบัฟเฟอร์จะบรรจุค่าความลึก สำหรับพื้นผิวที่มองเห็นได้ และเฟรมบัฟเฟอร์จะบรรจุค่าสีสำหรับพื้นผิวเหล่านั้น

กระบวนการแต่ละขั้นตอนของซีบัฟเฟอร์ ได้สรุปไว้ใน ขั้นตอนวิธี นี้ สมมติว่าค่าความลึก(depth values) เป็นช่วงปกติ 0.0 ถึง 1.0 ด้วยระนาบภาพ( view plane) ที่ค่าความความลึก  $\text{depth} = 0$

ให้ค่าความลึกสำหรับตำแหน่งจุดยอด(vertex positions) ของโพลีกอนในฉาก ซึ่งเราสามารถคำนวณความลึกที่ทุกๆจุดอื่นบนระนาบแสดงโพลีกอน ที่ตำแหน่งพื้นผิว(x,y) ค่าความลึกคำนวณจากสมการระนาบนี้

$$z = \frac{-Ax - By - D}{C} \quad (2.8)$$

สำหรับเส้นตรวจสอบในรูปที่ 2. 7 ในแนวนอนตำแหน่งจุด  $x$  ที่อยู่ติดกันจะต่างกัน  $\pm 1$  และในแนวตั้งตำแหน่งจุด  $y$  ที่อยู่ติดกันจะต่างกัน  $\pm 1$  ถ้าความลึกของตำแหน่ง  $(x,y)$  คือ  $z$  แล้วความลึก  $z'$  ของตำแหน่งถัดไป คือ  $(x+1,y)$  ไปตามเส้นตรวจสอบ จากสมการ(2.8)



รูปที่ 2.7 จาก  $(x,y)$  บนเส้นตรวจสอบ ตำแหน่งถัดไปมีพิกัด  $(x+1,y)$  และตำแหน่งลงมามีพิกัด  $(x,y-1)$

$$z' = \frac{-A(x+1) - By - D}{C} \quad (2.9)$$

หรือ

$$z' = z - \frac{A}{C} \quad (2.10)$$

$-A/C$  คือ ค่าคงที่สำหรับแต่ละพื้นผิว ดังนั้น ถ้าความลึก ถัดไปตามเส้นตรวจสอบ จะถูกหาจากค่าก่อนหน้านั้นเพียงครั้งเดียว

ประมวลผลตำแหน่งจุดภาพจากซ้ายไปขวาไปตามเส้นตรวจสอบ ซึ่งเราเริ่มโดยการคำนวณค่าความลึกบนเส้นขอบทางซ้ายของโพลีกอนที่อยู่บนเส้นตรวจสอบ สำหรับแต่ละตำแหน่งถัดไปตามเส้นตรวจสอบ เราคำนวณค่าความลึกจากสมการ(2.10)

เราสามารถดำเนินการ ขั้นตอนวิธีซีพเพอร์ โดยเริ่มที่ตำแหน่งบนสุดของโพลีกอน แล้วเราสามารถคำนวณซ้ำที่ค่าพิกัด  $x$  ลดลง ทางขอบซ้ายของโพลีกอน ค่า  $x$  สำหรับตำแหน่งเริ่มต้นบนแต่ละเส้นตรวจสอบ สามารถคำนวณจากค่า  $x$  เริ่มต้นบนเส้นตรวจสอบอันก่อนหน้านั้นได้

$$x' = x - \frac{1}{m}$$

เมื่อ  $m$  คือความชันของขอบ ค่าความลึกลดลงที่ขอบได้คำนวณซ้ำ

$$z' = z + \frac{A/m + B}{C} \quad (2.11)$$

ถ้าเราประมวลผลลงมาตั้งฉากกับขอบ แล้วค่าความชันเป็นค่าอนันต์ ทำให้การคำนวณซ้ำลดลงไป เป็น

$$z' = z + \frac{B}{C}$$

การสร้างพื้นผิวของโพลีกอน ขั้นตอนวิธีซีบัพเฟอร์ เป็นวิธีการที่ง่ายมากในการ ดำเนินการและไม่ต้องทำการเรียงลำดับพื้นผิวมณฑก แต่ข้อเสียอย่างหนึ่งของขั้นตอนวิธีซีบัพเฟอร์ คือการที่ต้องใช้หน่วยความจำขนาดใหญ่สำหรับใช้เป็นเฟรมบัพเฟอร์ และ ซีบัพเฟอร์ โดยเฉพาะ ซี บัพเฟอร์ ที่มีขนาดอย่างน้อย 32 บิต(หรือ 4 ไบต์) สำหรับแต่ละช่องสัญญาณ

ลำดับ ขั้นตอนวิธีซีบัพเฟอร์ ในการคำนวณโดยใช้ขั้นตอนของ Foley[14] เป็น แนวทางในการศึกษา ดังต่อไปนี้

1. สร้างโพลีกอนสามมิติ โดยกำหนดตำแหน่งฉาก , โครงสร้าง, การ เปลี่ยนแปลง และแสง
2. เราต้องการใส่สีตำแหน่งซีบัพเฟอร์ แต่ละ โพลีกอนให้กลายเป็นภาพวัตถุ 2 มิติ ในระนาบ 3 มิติ เปลี่ยนแนวระนาบในมุมมองของกล้องเป็นการ เปลี่ยนรูป เริ่ม ต้นสำหรับแต่ละวัตถุ และแสงบนฉาก ซึ่งเป็น แสง ที่ สามารถมองเห็นได้ และเติมบัพเฟอร์ด้วยจุดที่เปลี่ยนแปลง
3. ถ้าโพลีกอนสามมิติ ได้ทำการเปลี่ยนแปลงทั้งหมด แล้ว จะทำการ เปลี่ยน ภาพของโพลีกอนสองมิติไปเป็นบัพเฟอร์ นั่นคือ โพลีกอน สองมิติ จะอ้าง ถึงแหล่งกำเนิดของวัตถุ 3 มิติ
4. ตรวจสอบเพิ่มเติมการคำนวณความเข้มข้นของแหล่งกำเนิดวัตถุโดยใช้ แสง และให้ค่าในแต่ละจุด

For each polygon {

for each pixel (x,y) inside the polygon projection area {

if (Z\_polygon\_pixel(x,y) > depth\_buffer(x,y) ) {

depth\_buffer(x,y) = Z\_polygon\_pixel(x,y);

color\_buffer(x,y) = polygon color at (x,y)

}

}

}



### 2.3.1.6 สายท่อการให้แสงและเงา(Rendering pipeline)

ในขั้นตอนทั้งหมดของการให้แสงและเงา เมื่อขั้นตอนหลักๆ รวมถึงกระบวนการย่อยอื่นๆ เหล่านี้เมื่อนำมาเชื่อมต่อกันก็จะทำให้ได้กระบวนการทั้งหมดของการให้แสงและเงา หรือสายท่อการให้แสงและเงา ที่สมบูรณ์ในที่สุด

ในกรณีที่เป็นงานสร้างภาพเคลื่อนไหว ซึ่งเป็นการกำหนดการเคลื่อนที่ ไม่ว่าจะด้วยวิธีใดจะต้องแปลงการกำหนดการเคลื่อนที่ทุกรูปแบบให้เป็นตำแหน่งของวัตถุที่นิ่งสนิท ณ เวลาต่าง ๆ เนื่องจากเราไม่สามารถการให้แสงและเงาวัตถุที่เคลื่อนที่ได้ จากนั้นจึงส่งข้อมูลของวัตถุที่ไม่มีการเคลื่อนที่นี้ให้กับตัวทำแสงและเงา(renderer) เพื่อทำการให้แสงและเงาที่ละภาพจนครบทุกตำแหน่งเวลาตามที่เรต้องการจนเกิดเป็นภาพเคลื่อนไหวในที่สุด

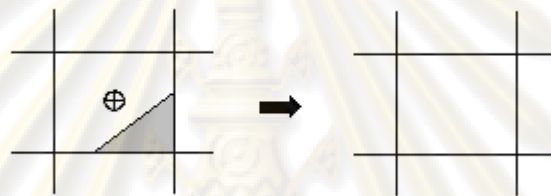
ขั้นตอนการแปลงข้อมูลรายละเอียดฉาก(scene description) ทั้งในส่วนของโครงสร้างโมเดลและภาพเคลื่อนไหว ให้อยู่ในสภาพที่พร้อมจะทำการให้แสงและเงา นี้เป็นกระบวนการแรกของสายท่อการให้แสงและเงา เรียกว่า การแหว่ผ่านฐานข้อมูล(database traversal) ซึ่งไม่ว่าเราจะทำการให้แสงและเงาด้วยวิธีการใดๆ ก็จะต้องผ่านขั้นตอนนี้ก่อนเสมอ ส่วนขั้นตอนต่อไปนี้อาจจะมีความแตกต่างกันออกไป ขึ้นอยู่กับวิธีการให้แสงและเงา

### 2.3.1.7 การสุ่มตัวอย่างสำหรับการสร้างภาพสามมิติ

ในการสร้างภาพสามมิติด้วยคอมพิวเตอร์นั้นเราไม่สามารถที่จะทำให้อยู่ในรูปของสัญญาณต่อเนื่อง(continuous signal)ได้ เนื่องจากความเป็นดิจิทัลของคอมพิวเตอร์จึงทำให้เราต้องใช้สัญญาณแบบวิยุต(discrete signal)แทน ดังนั้นกระบวนการหลักที่สำคัญของการให้แสงและเงาก็คือการสุ่มตัวอย่าง(sampling) ข้อมูลที่ได้จากการสร้างแบบจำลอง เพื่อสร้างสัญญาณแบบวิยุต ซึ่งก็คือ รูปภาพที่เป็นภาพดิจิทัล

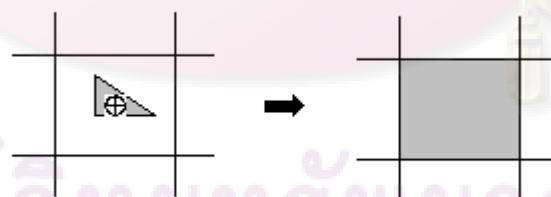
ภาพดิจิทัล มีลักษณะเป็นอาร์เรย์ของจุดภาพที่วางตัวอยู่ในแนวตั้งและแนวนอน โดยแต่ละจุดภาพก็จะมีค่าระดับความสว่างหรือค่าสีเป็นของตัวเองเพียงหนึ่งค่า ดังนั้นค่าระดับความสว่างของจุดภาพที่ประกอบขึ้นเป็นภาพๆ หนึ่งจึงมีลักษณะเป็นสัญญาณแบบวิยุต ที่อยู่ในโดเมนพื้นที่(spatial domain) เราจะต้องทำการหาตัวอย่าง (sample) ซึ่งจะมาเป็นค่าระดับความสว่างในแต่ละจุดภาพ โดยใช้ข้อมูลที่ได้จากการสร้างแบบจำลอง ซึ่งเราได้ทำการกำหนดไว้ว่ามีวัตถุใดบ้าง แต่ละวัตถุมีรูปร่างและพื้นผิวเป็นอย่างไร รับแสงจากแหล่งกำเนิดแสงมากน้อยเพียงใดเป็นต้น

- การสุ่มตัวอย่างแบบจุด คือการสุ่มตัวอย่างที่ทำการเลือกเพียงจุดใดจุดหนึ่งขึ้นมาให้เป็นตัวแทนของพื้นที่ซึ่งอยู่ภายใต้จุดภาพนั้นทั้งหมด ในความเป็นจริงนั้นเราควรใช้ข้อมูลทั้งหมด ซึ่งอยู่ในพื้นที่ภายใต้จุดภาพนั้นๆ เพื่อคำนวณหาค่าระดับความสว่างของจุดภาพ แต่แนวคิดดังกล่าวนั้นแทบเป็นไปได้ไม่ได้เนื่องจากใช้การคำนวณที่ซับซ้อนเป็นอย่างมาก ดังนั้นเราจึงต้องใช้วิธีการสุ่มตำแหน่งบนพื้นที่ภายใต้จุดภาพนั้นๆ ขึ้นมาเพียงจุดเดียวเพื่อคำนวณหาค่าความสว่างของจุดภาพ อย่างไรก็ตามแต่ละตำแหน่งบนจุดภาพใดๆ นั้นอาจจะมีค่าระดับความสว่างที่แตกต่างกันออกไปมากมาย ดังนั้นการเลือกขึ้นมาเพียงตัวอย่างเดียวจะทำให้เกิดความผิดเพี้ยนของภาพ (aliasing) ขึ้นได้



รูปที่ 2.8 แสดงกรณีที่ตำแหน่งของการสุ่มตัวอย่างนั้น ไม่อยู่บนพื้นที่ของโพลีกอน

รูปที่ 2.8 แสดงกรณีที่ตำแหน่งของการสุ่มตัวอย่างนั้น ไม่อยู่บนพื้นที่ของโพลีกอน ซึ่งทำให้ค่าระดับความสว่างของโพลีกอนที่อยู่ภายใต้จุดภาพนั้นๆ ไม่มีผลใดๆ ต่อค่าระดับความสว่างของจุดภาพทั้งสิ้น



รูปที่ 2.9 เป็นกรณีที่โพลีกอนมีขนาดเล็กกว่าขนาดพิกเซล

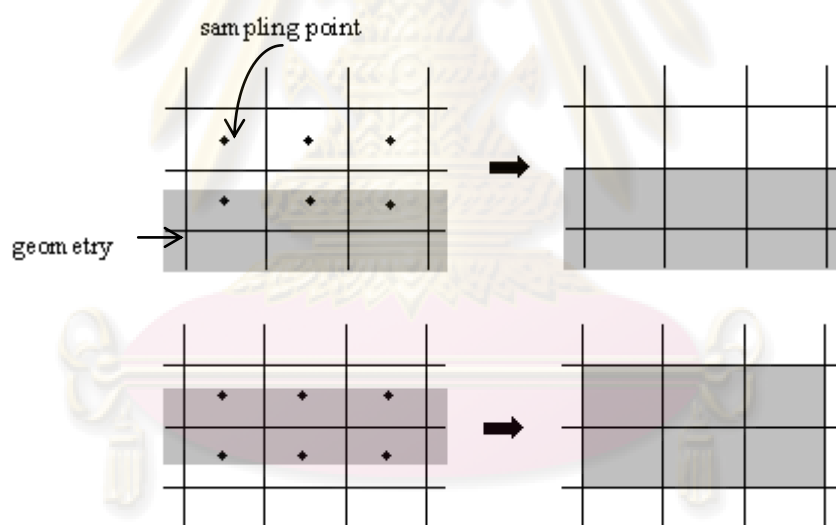
แต่บังเอิญอยู่ตรงตำแหน่งการสุ่มตัวอย่างพอดี

รูปที่ 2.9 เป็นกรณีที่โพลีกอนมีขนาดเล็กกว่าขนาดของจุดภาพมากแต่บังเอิญอยู่ตรงตำแหน่งที่ทำการสุ่มตัวอย่างพอดี ดังนั้นค่าระดับความสว่างของจุดภาพทั้งหมดจึงได้จากค่าระดับความสว่างของโพลีกอนชิ้นเล็กๆ นี้

ถึงแม้ว่าการสุ่มตัวอย่างแบบจุด มีโอกาสที่จะทำให้เกิดความผิดเพี้ยนของภาพได้ง่ายแต่ก็เป็นที่ยอมรับเนื่องจากใช้หลักการคำนวณที่ไม่ซับซ้อน รวดเร็วและใช้ได้ดีในเกือบทุกกรณี

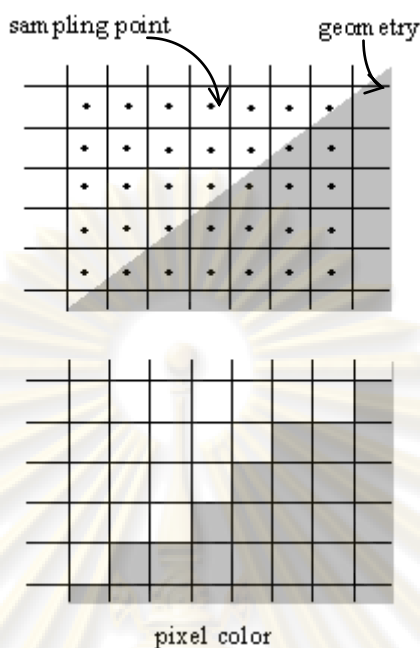
### 2.3.1.8 ความผิดเพี้ยนของภาพที่พบในการสร้างภาพสามมิติด้วยคอมพิวเตอร์

- การเกิดป๊อปปีง(Popping)[11][12] คือ การเกิดความผิดเพี้ยนของภาพซึ่งเป็นผลมาจากการใช้การสุ่มตัวอย่างแบบจุด กล่าวคือถ้าหากเราทำการสุ่มตัวอย่างลงบนพื้นที่ 1 จุดภาพใดๆ แล้วไม่มีวัตถุปรากฏอยู่ก็จะทำให้สีของจุดภาพนั้นๆ เป็นสีของพื้นหลัง( background) แต่ถ้าหากมีวัตถุอยู่ก็จะทำให้สีของจุดภาพนั้นๆ เป็นสีของวัตถุทันที ดังรูปที่ 2.10 ดังนั้นถ้าหากเราทำการเคลื่อนตำแหน่งของวัตถุไปเรื่อยๆ เราจะสังเกตเห็นว่าที่ขอบของวัตถุนั้นจะเกิดการกระพริบอันเนื่องจากการเปลี่ยนแปลงค่าสีของจุดภาพอย่างกะทันหัน



รูปที่ 2.10 แสดงการเกิดป๊อปปีง

- รอยหยัก(Jagged edge) ก็เป็นผลมาจากการเกิดป๊อปปีง โดยจะเกิดกับขอบของวัตถุที่อยู่ในแนวเอียงเมื่อเทียบกับแนวการวางตัวของตำแหน่งที่เราทำการสุ่มตัวอย่าง ทำให้เรามองเห็นเป็นขอบหยักเกิดขึ้น เราสามารถที่จะเห็นขอบหยักนี้ได้อย่างชัดเจนเนื่องจากค่าจุดภาพที่เป็นขอบของวัตถุและค่าสีของจุดภาพข้างเคียงนั้นมีการเปลี่ยนแปลงแบบก้าวกระโดด ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 แสดงการเกิดรอยหยักที่เกิดขึ้นที่ขอบของวัตถุ

ทั้งการเกิดป้อปปีงและรอยหยัก นั้นนอกจากจะเกิดที่ขอบของวัตถุแล้วยังสามารถที่จะเกิดกับพื้นผิวของวัตถุได้อีกด้วย ถ้าหากไม่มีวิธีการที่ดีพอที่จะกำจัดความผิดเพี้ยนของภาพ

### 2.3.2 ข้อมูล และการประมวลผล

ความจริงแล้วคอมพิวเตอร์ไม่สามารถเข้าใจสิ่งใดๆได้เลย สิ่งที่คอมพิวเตอร์ทำได้คือ ทำตามคำสั่งที่ได้ถูกโปรแกรมไว้ คอมพิวเตอร์ทำงานโดยอาศัยความแตกต่างของสถานะทางกายภาพ 2 สถานะ ที่เกิดจากกระแสไฟฟ้า ขั้วแม่เหล็ก หรือการสะท้อนของแสงเท่านั้น อาจกล่าวได้ว่าสิ่งที่คอมพิวเตอร์เข้าใจได้ คือ สถานะ การเปิด หรือ การปิดของสวิตช์ เท่านั้น ฉะนั้นคำว่า ข้อมูลจึงหมายถึงสถานะเปิด /ปิด (0/1) ที่ได้ถูกจัดกลุ่ม คอมพิวเตอร์จะนำข้อมูลดังกล่าว มาจัดรูปแบบใหม่ให้ได้เป็นสิ่งที่เรียกว่า สารสนเทศ ซึ่งจะเป็นผลที่ได้จากการประมวลข้อมูลโดยโปรแกรม เพื่อสามารถสื่อสารความหมายบางอย่าง ได้ดีขึ้นมากกว่า ข้อมูลที่ยังไม่ได้ผ่านการประมวลผล เช่น ข้อมูลเป็นตัวอักษรบนหน้ากระดาษ แต่ถ้าได้ประมวลผลให้เป็นสารสนเทศ ในรูปของรายงาน ตาราง หรือกราฟ ก็จะสามารถสื่อสารความหมายบางอย่าง ได้ดีขึ้น ทั้งนี้ขึ้นอยู่กับผู้ใช้

### 2.3.2.1 การแทนข้อมูลในคอมพิวเตอร์

สถานะของสวิตช์อิเล็กทรอนิกส์เล็กๆ ภายในคอมพิวเตอร์นั้นจะถูกแทนด้วยตัวเลข เนื่องจากสวิตช์มีแค่ 2 สถานะ คือ ปิดและเปิด ตัวเลขที่ใช้แทนสถานะของสวิตช์ก็จะมี 2 ตัว เช่นเดียวกัน คือ 0 และ 1 ระบบตัวเลขที่มีการใช้ตัวเลขเพียง 2 ตัวนี้ เรียกว่า ระบบเลขลักษณะฐานสอง(Binary system) ถ้าต้องการแทนตัวเลขที่มีค่ามากกว่า 1 ก็จะต้องใช้ตัวเลข 2 ตัว

สำหรับข้อมูลประเภทอื่นๆ ไม่ว่าจะเป็นข้อมูลตัวอักษร ข้อมูลรูปภาพ หรือแม้กระทั่งคำสั่งในโปรแกรมคอมพิวเตอร์ ก็จะถูกแทนด้วยเลข 0 และ 1 นี้ เช่นเดียวกัน ตัวอักษรที่เราเห็นบนจอภาพคอมพิวเตอร์นั้นเป็นเพียงวิธีการที่คอมพิวเตอร์ใช้สัญลักษณ์แทนตัวเลขเท่านั้น

เมื่อพูดถึงข้อมูลคอมพิวเตอร์แล้ว สวิตช์แต่ละตัวที่ใช้แทนสถานะเปิด /ปิด 1 ตัวนั้นเรียกว่า บิต (bit) ดังนั้น บิตจึงเป็นหน่วยที่เล็กที่สุดของข้อมูลในคอมพิวเตอร์ หน่วยที่ใหญ่ขึ้นของบิต คือ กลุ่มของบิตจำนวน 8 บิต เรียกว่าไบต์ (byte) และด้วยจำนวน 8 บิตหรือ 1 ไบต์นี้เองที่ทำให้คอมพิวเตอร์สามารถแทนค่าของตัวเลขได้ 256 ค่าด้วยกัน คือ 0 ถึง 255 (ซึ่งก็คือ 0 ถึง  $2^8$ ) และตัวจำนวนบิตขนาด 8 บิตนี้สามารถใช้แทนตัวอักษรต่างๆ ที่ใช้ในชีวิตประจำวันได้ค่อนข้างครบถ้วน

### 2.3.2.2 การเข้ารหัสตัวเลข

ตัวเลขในคอมพิวเตอร์สามารถแทนได้หลายวิธี ขึ้นอยู่กับรูปแบบตัวเลขที่ต้องการใช้งาน ตัวเลขจำนวนเต็มสามารถแทนได้ด้วยเลขลักษณะฐานสองโดยตรง โดยกำหนดจำนวนบิตที่ใช้ โดยปกติหน่วยประมวลผลกลาง(Central Processing Unit, CPU) จะออกแบบให้มีจำนวนบิตในตัว หน่วยประมวลผลกลางเป็น 8 บิต หรือ 16 บิต และออกแบบวงจรการคำนวณให้ใช้กับตัวเลขแบบ 8 บิต หรือ 16 บิต ซึ่งจะคำนวณได้เร็ว สำหรับเลขจำนวนเต็มเล็กๆ ก็แทนด้วยเลข 8 บิต (0-256) ถ้ามากกว่านั้นก็ใช้เลข 16 บิต (0-65536) ในกรณีที่ต้องการตัวเลขลบ เช่น -1 วิธีหนึ่งคือใช้บิตหน้า เป็น บิตเครื่องหมาย เช่น สำหรับ 8 บิต ให้บิตที่ 8 แทนเครื่องหมายโดยให้ 0 เป็นบวก และ 1 เป็นลบ ที่เหลือ 7 บิต จะได้เลข 0 – 127 (จะได้ เลข ศูนย์ 2 ตัว เป็น ศูนย์บวก และ ศูนย์ลบ ซึ่งเป็นเลขลักษณะฐานสองที่ไม่เท่ากัน) ตัวเลขลบอาจจะใช้วิธีที่เรียกว่า two complement แทนก็ได้

ตัวเลขในระบบฐาน 10 อาจแทนด้วยเลขฐาน 2 ในแต่ละหลัก เช่น 12 แทนด้วยเลขฐาน 2 ขนาด 4 บิตสองตัว คือ 0001 0010 ก็ได้ วิธีการนี้ เรียก Binary Coded Decimal ที่ใช้ 4

บิตต่อ 1 ตัวเพราะต้องการแทนตัวเลข 0-9 และทั้งนี้การบวก ลบ เลข ต้องใช้การคำนวณพิเศษกว่าบวก ลบ เลขลักษณะฐานสอง โดยตรง เนื่องจากการทศข้ามหลัก ไม่เหมือนกัน

ตัวเลขจุดทศนิยม มีการแทนค่าได้หลายวิธีเช่นกัน วิธีหนึ่งเรียกว่า IEEE floating point (IEEE754) เป็นมาตรฐานที่นิยมโดยจะอยู่ในรูปเลขลักษณะฐานสอง จำนวน 32 บิต ซึ่งมีส่วนประกอบสามส่วน คือบิตที่ 31 เป็นส่วนของเครื่องหมาย( Sign) บิตที่ 30-23 เป็นส่วนจำนวนกำลังของสิบ(Exponent) จะถูกเก็บลงในหน่วยความจำ 8 บิต (โดยค่า 127 หมายถึง 0 ดังนั้น 8 บิต จะแทนค่าได้จาก -127 ถึง 128) ส่วนสุดท้าย บิตที่ 22 ถึงบิตที่ 0 เป็นส่วนของความละเอียดของตัวเลขนั้น(Mantissa) ดังแสดงในรูปที่ 2.12 [15]

31	30	...	23	22	...	0
Sign Bit	Exponent Bits			Mantissa Bits		

รูปที่ 2.12 แสดงตำแหน่งบิตที่เก็บค่าตัวเลขจุดทศนิยม

### 2.3.2.3 ตัวดำเนินการข้อมูลเชิงบิต

ตัวดำเนินการประเภทนี้มีแนวคิดมาจากการคำนวณข้อมูลที่เกี่ยวข้องกับบิต เพื่อให้ผลลัพธ์ในการคำนวณออกมาเป็นข้อมูลบิต (บิตเป็นหน่วยที่เล็กที่สุดของข้อมูลในคอมพิวเตอร์ บิต) ส่วนใหญ่แล้วตัวดำเนินการประเภทนี้นิยมใช้กับการคำนวณที่สลับซับซ้อน หรือการคำนวณเชิงคณิตศาสตร์ขั้นสูง

การจัดเก็บข้อมูลประเภทต่างๆ จะเก็บในรูปแบบของบิต และการเก็บข้อมูลจะเป็นเครื่องหมาย กล่าวคือบิตทางซ้ายสุดของข้อมูลจะใช้เก็บเครื่องหมาย ตารางที่ 2.1[16] แสดงตัวอย่างตัวดำเนินการระดับบิต

ตารางที่ 2.1 แสดงตัวดำเนินการสำหรับการคำนวณข้อมูลในรูปแบบบิต

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์ของ a (เมื่อ a = 1101; b = 1001)
~	NOT (1เป็น0; 0เป็น1)	~a	0010
&	AND	a & b	1001
	OR	a   b	1101
<<	ขยับบิตทางซ้าย	a << 2	0100
>>	ขยับบิตทางขวา	a >> 2	0011
>>>	ขยับบิตทางขวา (สำหรับข้อมูล unsign)	a >>> 2	0011

คำสั่ง NOT ใช้สำหรับกลับบิต เมื่อข้อมูลเดิมในตำแหน่งบิตใดๆเป็น 0 จะถูกกลับเป็น 1 และเมื่อเป็น 1 จะถูกกลับเป็น 0

คำสั่ง AND ใช้สำหรับดำเนินการจากข้อมูล 2 จำนวน โดยอาศัยข้อพิจารณาดังตารางที่ 2.2

คำสั่ง OR ใช้สำหรับดำเนินการจากข้อมูล 2 จำนวน โดยอาศัยข้อพิจารณาดังตารางที่ 2.2

ตารางที่ 2.2 แสดงตัวอย่างของการดำเนินการ AND และ OR

a	b	a&b	a b
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

คำสั่งขยับบิต(shift) ใช้สำหรับดำเนินการกับข้อมูลหนึ่งตัว โดยที่สามารถทำให้มีการขยับของบิตไปตามทิศทางที่กำหนดเช่นทางซ้าย หรือทางขวา การใช้คำสั่งบิตต้องมีจำนวนบอกด้วยว่าขยับไปเป็นเท่าไร ดังตัวอย่างในตารางที่ 2.1

การขยับบิตไปทางขวาจะทำให้บิตที่ถูกขยับออกนอกกรอบสูญหายไป ส่วนบิตที่จะเข้ามาใส่ใหม่ทางซ้ายนั้นขึ้นอยู่กับเครื่องหมายของค่าเดิมนั้นว่าเป็นบวก หรือลบ ในกรณีของค่าบวก ก็จะเป็นที่แน่นอนว่าจะนำค่า 0 มาใส่ทางซ้าย แต่ถ้าเป็นค่าลบแล้ว แต่ละเครื่องจะทำงานไม่เหมือนกัน

### 2.3.2.4 บิตฟิลด์

ลักษณะการใช้งานที่จะทำให้ประโยชน์มากอีกอย่างหนึ่งก็ได้แก่ การใช้คุณสมบัติของการปฏิบัติการบนบิต [17] สร้างวิธีการเก็บข้อมูลแบบสวางเนื้อที่ (packed data) วิธีการเก็บข้อมูลแบบสวางเนื้อที่นี้ยึดถือหลักการที่ว่าเราจะใช้เนื้อที่ข้อมูลเท่าที่จำเป็นจริงๆ เท่านั้น ตัวอย่างที่เคยเห็นกันมาแล้วก็ได้แก่ การระบุชนิดข้อมูลเป็น int (แทนที่จะเป็น long) ถ้าเราทราบว่าข้อมูลเป็นตัวเลขจำนวนเต็มที่มีขนาดไม่ใหญ่นัก เป็นต้น ส่วนการใช้การปฏิบัติการบนบิตมาช่วยที่จะเห็นบ่อยๆ ได้แก่ การเซตค่าตรรกะ ว่าเป็นค่าจริงหรือเท็จซึ่งถ้าเราระบุชนิดของตรรกะ นี้ให้เป็น char ก็ต้องใช้เนื้อที่ถึง 16 บิตและใช้เนื้อที่ 32 บิต ถ้าเราระบุชนิดเป็น int ซึ่งที่จริงแล้วเนื้อที่เพียง 1 บิต (คือ 0 เป็นค่าเท็จ และ 1 เป็นค่าจริง) ก็มีความเพียงพอแล้วในการเก็บค่าตรรกะ ซึ่งก็จะเห็นว่าเราสามารถสวางเนื้อที่ในการเก็บข้อมูลไปได้เป็นอย่างมากทีเดียว โดยเฉพาะถ้ามีข้อมูลประเภทนี้หลาย ๆ ตัว

### 2.3.2.5 การประมาณค่าในช่วงเสมือนพหุนาม (Spline Interpolation)

ฟังก์ชัน  $S(x)$  เรียกว่า ฟังก์ชันเสมือนพหุนามดีกรี  $k$  ถ้า

- โดเมนของ  $S$  อยู่ในช่วง  $[a, b]$
- $S, S', \dots, S^{(k-1)}$  ต่อเนื่องในช่วง  $[a, b]$ .
- ที่จุด  $x_i$  ซึ่ง  $a = x_0 < x_1 < \dots < x_n = b$  และ  $S$  เป็นพหุนามกำลังที่เป็นไปได้มากที่สุด  $k$  บนแต่ละช่วง  $[x_i, x_{i+1}]$

พูดง่าย ๆ ว่าฟังก์ชันเสมือนพหุนาม เป็น piecewise polynomial functions [18][20] ซึ่งเชื่อว่าอนุพันธ์จะต่อเนื่องและอยู่ในช่วงที่แน่นอน

เมื่อ  $k = 1$  เรียกว่า แบบเชิงเส้น (linear splines)

เมื่อ  $k = 2$  เรียกว่า แบบพหุนามดีกรี 2 (quadratic splines) และ

เมื่อ  $k = 3$  เรียกว่า แบบพหุนามดีกรี 3 (cubic splines)



### แบบเชิงเส้น(Linear Spline)

เราต้องการฟังก์ชันเชิงเส้น  $S(x)$  ซึ่ง  $S(x_i) = y_i$  สำหรับ  $0 \leq i \leq n$   
ให้

$$S(x) = \begin{cases} S_0(x), & x_0 \leq x \leq x_1 \\ S_1(x), & x_1 \leq x \leq x_2 \\ \vdots & \vdots \\ S_{n-1}(x), & x_{n-1} \leq x \leq x_n \end{cases}$$

โดยแต่ละ  $S_i(x)$  เป็นเชิงเส้น

พิจารณาฟังก์ชันเชิงเส้น  $S_i(x)$  เส้นที่ผ่านจุด  $(x_i, y_i)$  และ  $(x_{i+1}, y_{i+1})$  ความชันของ  $S_i(x)$  คือ

$$m_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad (2.10)$$

เราสามารถกล่าวได้ว่า เส้นที่ผ่านจุด  $(x_i, y_i)$  และ  $(x, S_i(x))$  สำหรับทุกๆ  $x$  ในช่วง  $[x_i, x_{i+1}]$  ดังนั้น สำหรับทุกๆ  $x$  ในช่วงนี้ด้วย

$$m_i = \frac{S_i(x) - y_i}{x - x_i}$$

โดยให้

$$S_i(x) = y_i + m_i(x - x_i) \quad (2.11)$$

วิธีการของการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น[20]

1. คำนวณค่า  $m_i$  สำหรับ  $0 \leq i \leq n-1$  โดยใช้ (2.10)
2. สำหรับทุกๆ  $x \in [x_0, x_n]$  หาช่วงที่ถูกต้องของ  $x$  และหาค่าเสมือนพหุนามโดยใช้ (2.11)

จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนวิธี[20]

(กำหนดค่า  $x_i, y_i, m_i$  สำหรับช่วง  $0 \leq x \leq n-1$  และกำหนดค่า  $x$ )

```

for i = 0 to n-1
    if  $x \leq x_{i+1}$ 
        break;
    end
end
 $S(x) = y_i + m_i(x - x_i)$ 

```

บันทึก

- เมื่อ  $x < x_0$  ให้  $S(x) = y_0 + m_0(x - x_0)$   
เมื่อ  $x > x_n$ , ให้  $S(x) = y_{n-1} + m_{n-1}(x - x_{n-1})$   
(เราใช้ค่าแรกและค่าสุดท้ายเพื่อหาค่า)
- เปรียบเทียบกับการประมาณในช่วงเชิงพหุนาม การประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้นจะง่ายและประหยัดเวลามากกว่า
- ถ้าต้องการประสิทธิภาพที่มากที่สุด สามารถใช้การค้นหาแบบไบนารี (binary search) เพื่อหาช่วงที่ต้องการ

### แบบพหุนามดีกรีสอง(Quadratic Spline)

พหุนามดีกรีสอง[19] จะต้องมีอนุพันธ์ อันดับหนึ่งต่อเนื่องที่ จุดเชื่อม (knot หรือ interior point) ระหว่างสองช่วงใดๆ

พหุนามดีกรีสองที่ใช้ประมาณค่าในแต่ละช่วงข้อมูลมีรูปแบบดังนี้

$f(x) = ax^2 + bx + c$  นั่นคือฟังก์ชันกำลังสองที่ประมาณค่าฟังก์ชันของ  $x \in [x_{i-1}, x_i]$  สำหรับช่วงที่  $i$  ใดๆ อยู่ในรูป

$$f(x) = a_i x^2 + b_i x + c_i \quad \text{เมื่อ } i = 1, 2, \dots, n$$

จากรูปแบบทั่วไปนี้จะมีตัวแปรที่ไม่ทราบค่าสามตัวด้วยกันคือ  $a_i, b_i$  และ  $c_i$  สำหรับในแต่ละช่วง จากข้อมูล  $n+1$  จุด มีทั้งหมด  $n$  ช่วง ดังนั้นเรามีตัวไม่ทราบค่าทั้งหมด  $3n$  ตัว หมายความว่าเราต้อง การ  $3n$  สมการ เพื่อตัวแปรแก้สมการหาค่าทั้ง  $3n$  ตัว ดังนั้นเงื่อนไขสร้างขึ้นมาเพื่อให้ได้จำนวนสมการตาม ที่ต้องการคือ

1. สองฟังก์ชันที่อยู่ติดกันมีค่าเท่ากันที่จุดเชื่อมภายใน (interior points)

จุดเชื่อมภายในมีทั้งหมด  $n-1$  จุด  $(x_1, x_2, \dots, x_{n-1})$  จะได้ว่า

สำหรับ  $i = 2, 3, \dots, n$

$$a_{i-1}x_{i-1}^2 + b_{i-1}x_{i-1} + c_{i-1} = f(x_{i-1})$$

$$a_i x_{i-1}^2 + b_i x_{i-1} + c_i = f(x_{i-1})$$

จากเงื่อนไขนี้เราจะได้ทั้งหมด  $2(n-1) = 2n-2$  สมการด้วยกัน

เช่น ที่  $i=2$  หรือที่  $x_1$  ซึ่งเป็นจุดเชื่อมระหว่างช่วงที่หนึ่ง  $[x_0, x_1]$  และช่วงที่สอง  $[x_1, x_2]$

$$\text{ช่วงที่ 1 จะได้สมการ } f(x_1) = a_1 x_1^2 + b_1 x_1 + c_1$$

$$\text{ช่วงที่ 2 จะได้สมการ } f(x_1) = a_2 x_1^2 + b_2 x_1 + c_2$$

2. ฟังก์ชันที่หนึ่งผ่านจุด  $(x_0, f(x_0))$  และฟังก์ชันที่  $n$  ผ่านจุด  $(x_n, f(x_n))$  เราจะได้ 2 สมการด้วยกัน

$$f(x_0) = a_1 x_0^2 + b_1 x_0 + c_1$$

$$f(x_n) = a_n x_n^2 + b_n x_n + c_n$$

3. อนุพันธ์อันดับหนึ่งของฟังก์ชันที่อยู่ติดกันมีค่าเท่ากันที่จุดเชื่อม เราจะได้  $n-1$  สมการด้วยกัน

จากฟังก์ชันกำลังสอง  $f(x) = ax^2 + bx + c$  จะมีอนุพันธ์อันดับหนึ่งอยู่ใน

รูป  $f'(x) = 2ax + b$  จุดเชื่อมภายในมีทั้งหมด  $n-1$  จุด  $(x_1, x_2, \dots, x_{n-1})$  จะได้ว่า

สำหรับ  $i = 2, 3, \dots, n$

$$2a_{i-1}x_{i-1} + b_{i-1} = 2a_i x_{i-1} + b_i$$

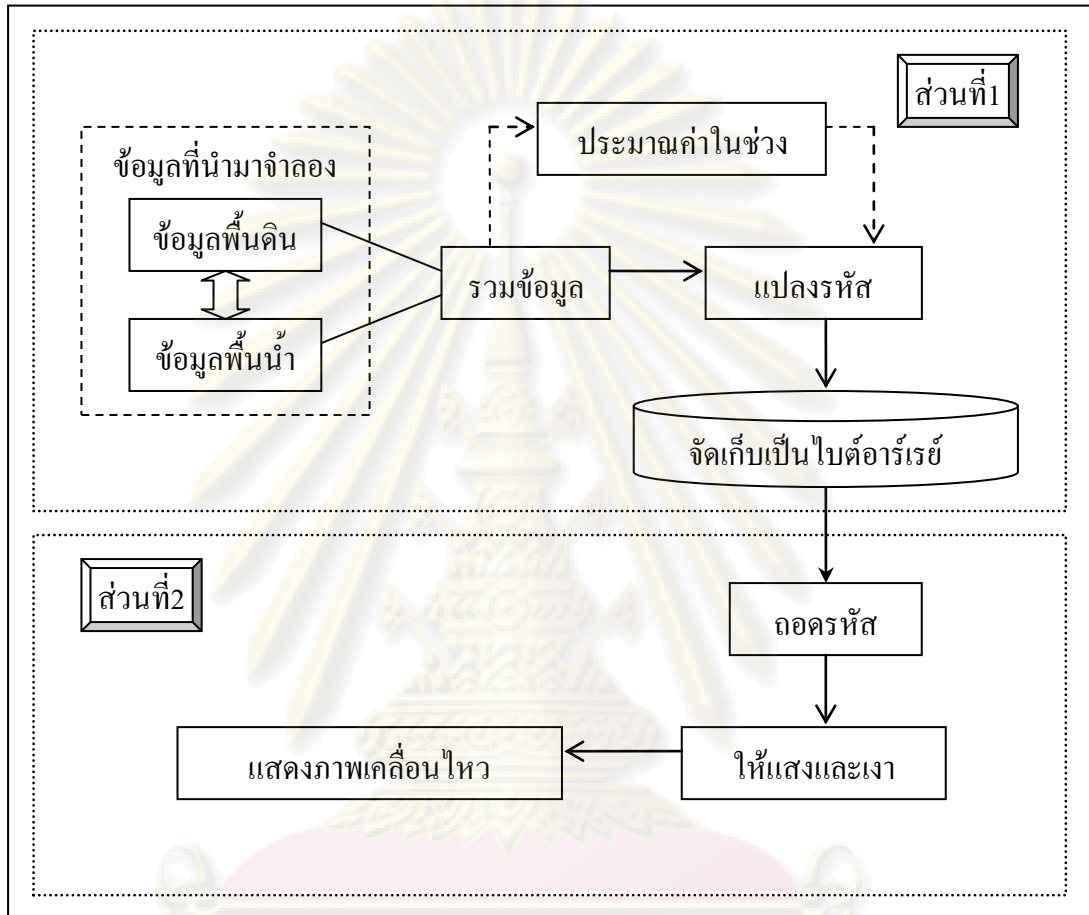
4. อนุพันธ์อันดับสองของฟังก์ชันแรกเท่ากับศูนย์ (Natural Quadratic Spline) จะได้  $a_1 = 0$

$$f''(x) = 2a_1 = 0$$

หลังจากที่ได้  $3n$  สมการ สำหรับ  $3n$  ตัวแปรแล้วนำไปแก้ระบบสมการโดยใช้เมตริกซ์

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

### บทที่ 3 วิธีดำเนินการวิจัย



รูปที่ 3.1 แผนผังการทำงานขั้นตอนดำเนินการ

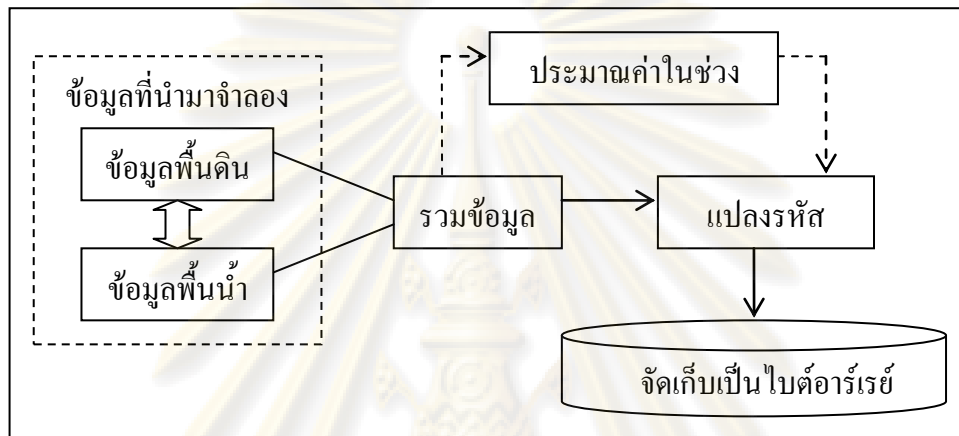
รูปที่ 3.1 แสดงขั้นตอนดำเนินการทั้งหมด ซึ่งเพื่อให้การประมวลผลที่เร็วจึงแยกการทำงานหลักๆ ออกเป็น 2 ส่วน เพราะทั้ง 2 ส่วนสามารถแยกการทำงานได้ไม่จำเป็นต้องทำงานทั้งหมดพร้อมกัน ถ้าเราไม่แยกการทำงานจะทำให้ใช้เวลานานมากในการประมวลผลแต่ละรอบการทำงาน

ซึ่งในส่วนที่หนึ่งนั้น เราจะทำการแปลงข้อมูลแล้วจัดเก็บไว้อีกเพิ่มข้อมูลหนึ่งเพื่อให้นำข้อมูลที่แปลงได้ รวมถึงการประมาณค่าในช่วงเสมือนพหุนามในส่วนที่เราต้องการเพิ่มข้อมูล ในส่วนที่สองจะนำข้อมูลที่ได้จากส่วนที่หนึ่งที่ได้ทำการแปลงข้อมูลแล้วนำไปสร้างภาพ

สามมิติโดยใช้ขั้นตอนวิธีซีบีพีเฟอร์เป็นหลักและใช้เทคนิคในการให้แสงและเงา และเพิ่มฟังก์ชันการแสดงผล[15][21][22][23][24][25] ซึ่งอธิบายรายละเอียดทั้ง 2 ส่วนดังต่อไปนี้

### 3.1 วิธีดำเนินการส่วนที่ 1

เป็นการวิเคราะห์ข้อมูลที่ได้จากณรงก์ฤทธิ์[1] และทำการแปลงข้อมูล ดังรูปที่ 3.2



รูปที่ 3.2 แผนผังการทำงานในขั้นตอนดำเนินการส่วนที่ 1

#### 3.1.1 ข้อมูลที่นำมาจำลอง

แบบจำลองที่ใช้ศึกษาพฤติกรรมของคลื่นสึนามิที่ได้รับการยอมรับ และใช้กันอย่างแพร่หลาย คือ Boussinesq model เป็นแบบจำลองที่อธิบายพฤติกรรมของคลื่นจากสมการอนุกรมมวลและโมเมนต์ โดยทำการพิจารณาโมเมนต์ใน 2 มิติ และใช้ระเบียบวิธีผลต่างสี่เหลี่ยมในการหาคำตอบของสมการ ซึ่งในขั้นตอนของการหาคำตอบโดยใช้วิธีการทำซ้ำ จะต้องทำการคำนวณซ้ำหลายๆ รอบจนกว่าคำตอบจะลู่เข้า และนอกจากนั้นแบบจำลองนี้ยังได้เพิ่มพจน์ที่ได้จากการทดลองซึ่งได้แก่ bottom friction , breaking wave , subgrid turbulent mixing และ sponge layer เข้ามาเพื่อให้ลักษณะของคลื่นที่ได้ใกล้เคียงกับคลื่นที่เกิดขึ้นจริงมากที่สุด

ข้อมูลที่ได้มีขนาดใหญ่ เพื่อสร้างให้ใกล้เคียงกับคลื่นที่เกิดขึ้นจริงในการสร้างภาพเคลื่อนไหวสามมิติ แต่จุดนี้เองที่เป็นปัญหาในการประมวลผลซึ่งทำให้ต้องแปลงข้อมูลก่อนนำไปทำการให้แสงและเงา ซึ่งจะอธิบายในหัวข้อ 3.1.3

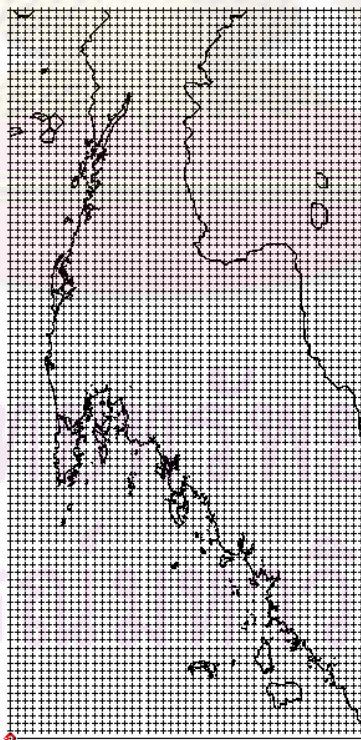
ข้อมูลเชิงตัวเลขที่ได้นำมาจำลอง แบ่งเป็น 3 บริเวณ คือ

1. บริเวณมหาสมุทรอินเดีย ดังแสดงในรูปที่ 3.3



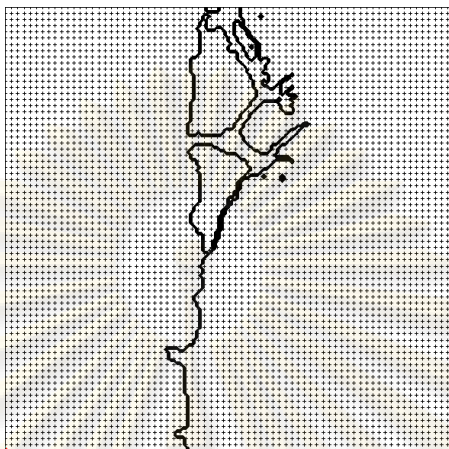
รูปที่ 3.3 ภาพข้อมูลที่นำมาจำลองบริเวณมหาสมุทรอินเดีย

2. บริเวณชายฝั่งภาคใต้ของประเทศไทย ดังแสดงในรูปที่ 3.4



รูปที่ 3.4 ภาพข้อมูลที่นำมาจำลองบริเวณชายฝั่งภาคใต้ของประเทศไทย

### 3. บริเวณจังหวัดพังงาของประเทศไทย ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 ภาพข้อมูลที่นำมาจำลองบริเวณจังหวัดพังงาของประเทศไทย

และข้อมูลเชิงตัวเลขที่ได้นำมาจำลอง แบ่งเป็น 2 ส่วน คือ

#### 1. ข้อมูลส่วนพื้นน้ำ

ข้อมูลส่วนนี้จะเป็นความสูงของระดับน้ำทะเล ดังนั้นส่วนที่เป็นพื้นดินจะถูกตั้งค่าให้เป็นค่าคงที่ติดลบที่ไม่ซ้ำกับความลึกของพื้นดินใต้มหาสมุทรที่อยู่ ข้อมูลส่วนพื้นดิน จึงได้ทำการกำหนดค่าพื้นดินเป็นค่าคงที่ -9999 ที่เราเลือกที่จะกำหนด เป็นค่าคงที่ -9999 นั้น เนื่องจากเราสามารถใช้อุปกรณ์หมายลบแบ่งระหว่างความสูงของพื้นน้ำกับพื้นดิน และที่กำหนดค่าเป็น 9999 เพื่อไม่ให้ค่าไปใกล้เคียงพื้นดินใต้มหาสมุทร นั่นหมายความว่าสามารถกำหนดเป็นค่าคงที่ลบใดๆ ที่มากพอที่จะไม่ใกล้เคียงกับค่าจริง เพื่อประโยชน์ในการแบ่งส่วนน้ำและพื้นดิน ตัวอย่างข้อมูลบางส่วนแสดงดังรูปที่ 3.6

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

ส่วนที่ 1	1793	2191		
ส่วนที่ 2	72.0	102.000000		
	-13.000000	23.500000		
ส่วนที่ 3	0.000725	0.000726	1.304550	5.23983
	7.017017	-9999	-9999	-9999
	-9999	-9999	-9999	-9999
	...			

รูปที่ 3.6 ภาพตัวอย่างข้อมูลส่วนพื้นน้ำที่นำมาจำลองบางส่วน

### ข้อมูลที่ได้แต่ละส่วนจากรูปที่ 3.6

ส่วนที่ 1 : จำนวนความละเอียดของภาพ เช่น 1793      2191

นั่นก็คือจำนวนจุดที่ใช้ในการคำนวณ 1793 x 2191 จุด

ส่วนที่ 2 : ค่าตำแหน่งของภาพ เช่น 72.000000      102.000000

-13.000000      23.500000

นั่นก็คือตำแหน่งของภาพละติจูดตั้งแต่ช่วง 72.000000 ถึง 102.000000 และลองจิจูดตั้งแต่ช่วง -13.000000 ถึง 23.500000

ส่วนที่ 3 : เป็นส่วนของความสูงของระดับน้ำทะเลเมื่อเทียบกับระดับน้ำทะเลเฉลี่ย และความสูงของพื้นดิน ซึ่งส่วนนี้จะมีค่าคงที่ หน่วยเป็นเมตร เช่น 0.000725      0.000726      1.304550  
5.23983      7.017017      -9999      -9999      -9999      ...

ดังเช่น ตำแหน่งที่ [0,0] เป็นตำแหน่งของน้ำที่มีความสูง 0.000725 เมตร ตำแหน่งที่ [0,1] เป็นตำแหน่งของน้ำที่มีความสูง 0.000726 เมตร ... ที่ [0,4] เป็นตำแหน่งของน้ำที่มีความสูง 7.017017 เมตร และตำแหน่งที่เป็นค่าคงที่ -9999 เป็นส่วนของพื้นดิน เป็นต้น



## 2. ข้อมูลส่วนพื้นดิน

ข้อมูลจะเป็นความสูงของพื้นดิน โดยค่าลบเป็นส่วน ความลึกของ พื้นดินที่อยู่ใต้ มหาสมุทร ตัวอย่างข้อมูลบางส่วนแสดงดังรูปที่ 3. 7 ซึ่งจะเห็นได้ว่าความลึกของ พื้นดินที่อยู่ใต้ มหาสมุทรจะไม่ซ้ำ กับค่า -9999 ที่เรากำหนดไว้ข้อมูลส่วนพื้นน้ำนั่นเอง

ส่วนที่ 1	1793	2191		
ส่วนที่ 2	72.0	102.000000		
	-13.000000	23.500000		
ส่วนที่ 3	-4362.169920	-4332.522792	-4362.169920	-2.003334
	-1.098233	2.432111	2.523321	3.340228
	5.002211	10.52090	100.302855	2111.491201
	...			

รูปที่ 3.7 ภาพตัวอย่างข้อมูลส่วนพื้นดินที่นำมาจำลองบางส่วน

### ข้อมูลที่ได้แต่ละส่วนจากรูปที่ 3.7

ส่วนที่ 1 : จำนวนความละเอียดของภาพ เช่น 1793 2191

นั่นก็คือจำนวนจุดที่ใช้ในการคำนวณ  $1793 \times 2191$

ส่วนที่ 2 : ค่าตำแหน่งของภาพ เช่น 72.000000 102.000000

-13.000000 23.500000

นั่นก็คือตำแหน่งของภาพละติจูดตั้งแต่ช่วง 72.000000 ถึง 102.000000 และลองจิจูดที่ตั้งตั้งแต่ช่วง -13.000000 ถึง 23.500000

ส่วนที่ 3 : เป็นส่วนของความสูงของพื้นดิน ซึ่งส่วนที่มีค่าลบจะเป็นพื้นดินใต้ทะเล หน่วยเป็นเมตร  
 เช่น -4362.169920      -4332.522792      -4362.169920      -2.003334      -1.098233  
 2.432111      2.523321      3.340228      5.002211      10.52090      100.302855  
 2111.491201

ดังเช่น ตำแหน่งที่ [0,0] เป็นตำแหน่งของพื้นดินใต้ทะเลลึกลงไป 4362.16992 เมตร ตำแหน่งที่ [0,1] เป็นตำแหน่งของพื้นดินใต้ทะเลลึกลงไป 4332.522792 เมตร ... ตำแหน่งที่ [0,4] เป็นตำแหน่งของพื้นดินใต้ทะเลลึกลงไป 1.098233 เมตร ตำแหน่งที่ [0,5] เป็นตำแหน่งของพื้นดินที่มีความสูง 2.432111 เมตร เป็นต้น

### 3.1.2 การรวมข้อมูลส่วนพื้นน้ำและพื้นดินไว้เพิ่มเติมด้วยกัน

นำข้อมูลส่วนพื้นน้ำและพื้นดิน ณ ตำแหน่งเดียวกันมาเปรียบเทียบกัน ถ้าข้อมูลส่วนพื้นน้ำไม่ใช่ค่าคงที่ -9999 จะทำการเก็บข้อมูลส่วนพื้นน้ำและเปลี่ยนข้อมูลให้เป็นค่าลบ และถ้าข้อมูลส่วนพื้นน้ำเป็นค่าคงที่ -9999 จะทำการเก็บข้อมูลส่วนพื้นดินโดยที่ไม่ต้องเปลี่ยนข้อมูลให้เป็นค่าลบ เพื่อที่เราจะได้ใช้เครื่องหมายลบเป็นตัวแบ่งค่าระหว่างส่วนของความสูงระดับน้ำทะเลกับความสูงของพื้นดิน ซึ่งเราจะให้ค่าที่ติดลบเป็นความสูงส่วนพื้นน้ำ และค่าที่ไม่ติดลบเป็นส่วนพื้นดิน

ตัวอย่างการรวมข้อมูลส่วนพื้นน้ำและพื้นดินบางส่วนระหว่างข้อมูลรูปที่ 3.6 กับข้อมูลรูปที่ 3.7 เช่น

ณ ตำแหน่งที่ [0,0] ในส่วนพื้นน้ำมีค่าเท่ากับ 0.000725 และส่วนพื้นดินมีค่าเท่ากับ -4362.169920 ซึ่งจะเห็นว่าค่าส่วนพื้นน้ำไม่ใช่ค่าคงที่ -9999 ดังนั้น เราจะทำการเก็บข้อมูลส่วนพื้นน้ำและทำการเปลี่ยนข้อมูลให้มีค่าลบ นั่นคือเก็บค่าเป็น -0.000725

ขณะเดียวกันเปรียบเทียบ ณ ตำแหน่งที่ [0,5] ในส่วนพื้นน้ำมีค่าเท่ากับ -9999 และส่วนพื้นดินมีค่าเท่ากับ 2.432111 ซึ่งจะเห็นว่าค่าส่วนพื้นน้ำเป็นค่าคงที่ -9999 ดังนั้น เราจะทำการเก็บข้อมูลส่วนพื้นดินโดยที่ไม่ต้องทำการเปลี่ยนข้อมูลให้มีค่าลบ นั่นคือเก็บค่าเป็น 2.432111

เมื่อรวมข้อมูลเสร็จจะได้ข้อมูลแสดงดังรูปที่ 3. 8 และรายละเอียดของข้อมูลทั้ง 3 บริเวณ แสดงให้เห็นในตารางที่ 3.1

ส่วนที่ 1	1793	2191		
ส่วนที่ 2	72.0	102.000000		
	-13.000000	23.500000		
ส่วนที่ 3	-0.000725	-0.000726	-1.304550	-5.23983
	-7.017017	2.432111	2.523321	3.340228
	5.002211	10.52090	100.302855	2111.491201
	...			

รูปที่ 3.8 ภาพตัวอย่างข้อมูลที่ทำกรรวมกันระหว่างพื้นดินและน้ำบางส่วน

#### ข้อมูลที่ได้แต่ละส่วนจากรูปที่ 3.8

ส่วนที่ 1 : จำนวนความละเอียดของภาพ เช่น 1793                      2191

นั่นก็คือจำนวนจุดที่ใช้ในการคำนวณ 1793 x 2191

ส่วนที่ 2 : ค่าตำแหน่งของภาพ เช่น 72.000000                      102.000000

-13.000000                      23.500000

นั่นก็คือตำแหน่งของภาพละติจูดตั้งแต่ช่วง 72.000000 ถึง 102.000000 และลองจิจูดที่ตั้งแต่ช่วง -13.000000 ถึง 23.500000

ส่วนที่ 3 : เป็นส่วนของความสูงของพื้นดิน ซึ่งส่วนที่มีค่าลบจะเป็นพื้นดินใต้ทะเล หน่วยเป็นเมตร

เช่น -0.000725                      -0.000726                      -1.304550                      -5.23983                      -7.017017                      2.432111

2.523321                      3.340228                      5.002211                      10.52090                      100.302855                      2111.491201

...

ดังเช่น ตำแหน่งที่ [0,0] เป็นตำแหน่งของพื้นน้ำที่มีความสูง 0.000725 เมตร ตำแหน่งที่ [0,1] เป็นตำแหน่งของพื้นน้ำที่มีความสูง 0.000726 เมตร ตำแหน่งที่ [0,4] เป็นตำแหน่งของพื้นน้ำที่มีความสูง 7.017017 เมตร ตำแหน่งที่ [0,5] เป็นตำแหน่งของพื้นดินที่มีความสูง 2.432111 เมตร เป็นต้น

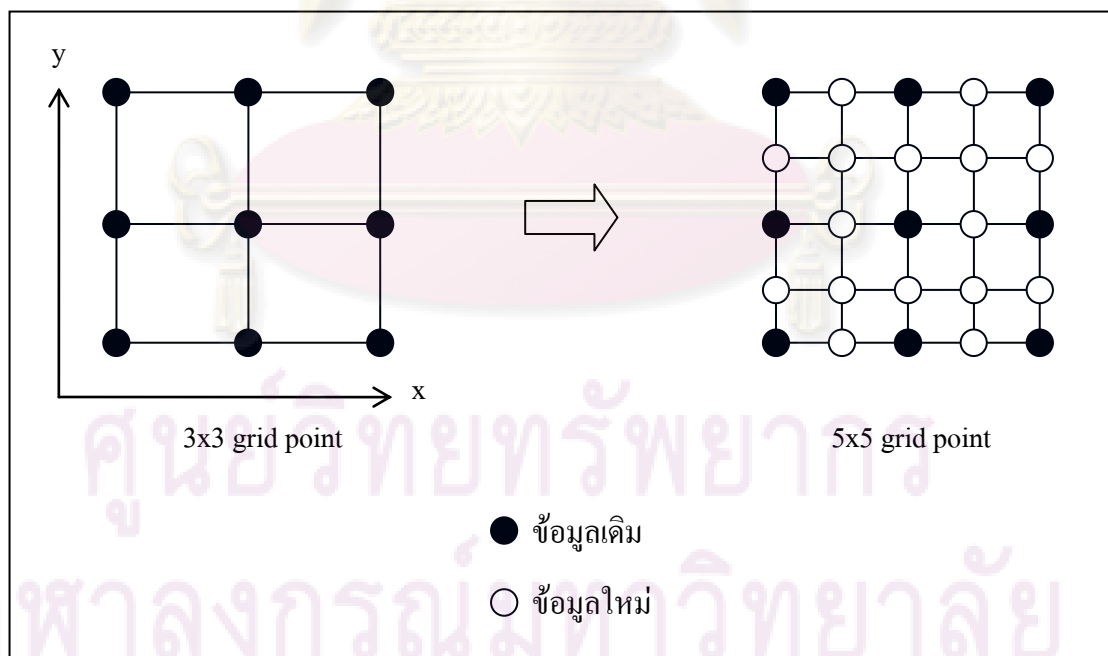
## รายละเอียดข้อมูล

ตารางที่ 3.1 รายละเอียดข้อมูลที่นำมาจำลอง

บริเวณ	ความละเอียดของภาพ ในแต่ละเพิ่มข้อมูล	จำนวน เพิ่มข้อมูล	ขนาดเพิ่มข้อมูล ทั้งหมด
มหาสมุทรอินเดีย	1793x2191	150	8.36 GB
ชายฝั่งภาคใต้ของประเทศไทย	560x1152	300	2.74 GB
จังหวัดพังงา	145x145	220	34.4 MB

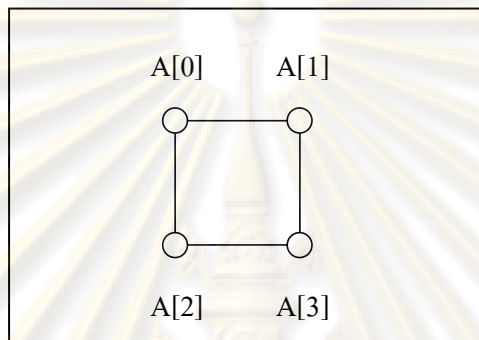
### 3.1.3 การเพิ่มความละเอียดของภาพ

ในการเพิ่มข้อมูล เราจะทำการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น (อธิบายในบทที่ 2) ระหว่างข้อมูลที่ติดกัน โดยการประมาณค่าในแนวนอน (แกน  $x$ ) และแนวตั้ง (แกน  $y$ ) ดังรูปที่ 3.9



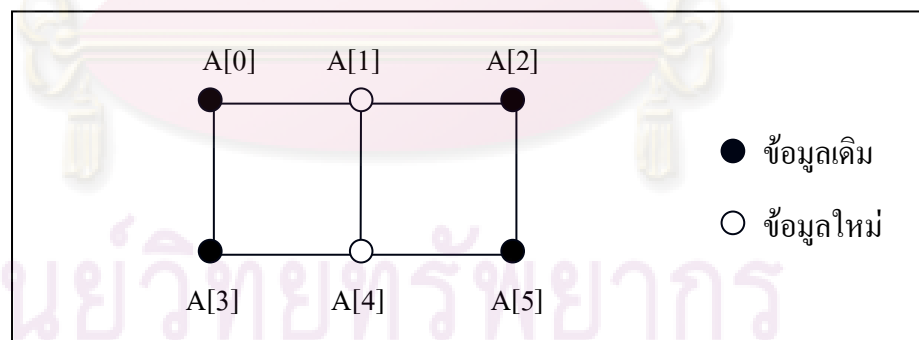
รูปที่ 3.9 การเพิ่มข้อมูลใหม่จาก 3x3 จุดที่ใช้จำนวน เป็น 5x5 จุดที่ใช้จำนวน

เราจะทำการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้นระหว่างจุด 2 จุด ที่ติดกัน เพื่อให้ได้จุดใหม่ ระหว่างจุด 2 จุดนั้นๆ จากนั้นจะทำการเลื่อนตำแหน่งของ array ใหม่ ดังนั้นถ้ามีข้อมูล  $N \times N$  เมื่อทำการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้นโดยประมาณค่าระหว่าง 2 จุดที่ติดกันเพียง 1 ค่า จะได้ข้อมูลเป็น  $(2N-1) \times (2N-1)$  แต่ถ้าต้องการประมาณค่าระหว่าง 2 จุดที่ติดกัน 2 ค่า จะได้ข้อมูลเป็น  $(3N-1) \times (3N-1)$  เป็นต้น ขึ้นอยู่ว่าต้องการประมาณค่า เพื่อเพิ่มข้อมูลมากเพียงใด



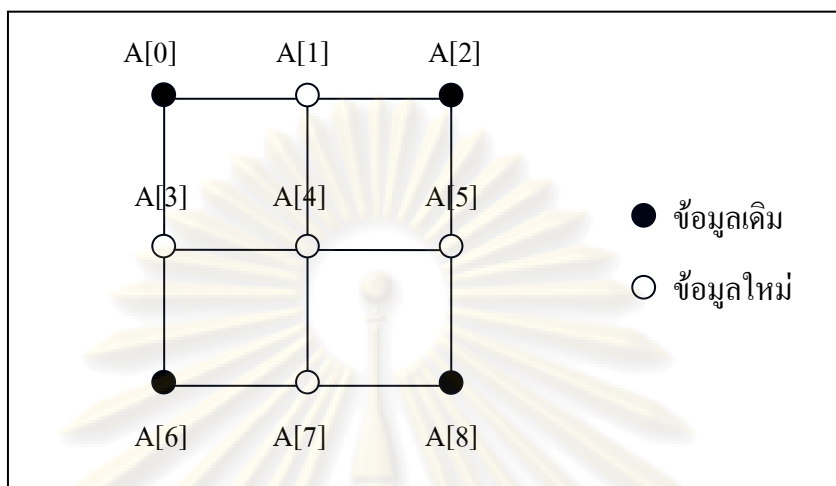
รูปที่ 3.10 แสดงเฟรมภาพที่มีความละเอียดข้อมูล  $2 \times 2$

จากรูปที่ 3.10 จะทำการประมาณในช่วงเสมือนพหุนามแบบเชิงเส้น ข้อมูลระหว่าง 2 จุด ที่ติดกัน 1 ค่า โดยทำการประมาณค่า ตามแกน x ก่อน นั่นคือ ประมาณค่าระหว่าง  $A[0]$  กับ  $A[1]$  และ ประมาณค่าระหว่าง  $A[2]$  กับ  $A[3]$  ซึ่งจะได้ค่าใหม่ 2 ค่า จากนั้นจะแทรกข้อมูลใหม่ไประหว่างช่วงที่ประมาณได้ และทำการเลื่อนตำแหน่งของ array ใหม่ แสดงดังรูปที่ 3.11



รูปที่ 3.11 แสดงการประมาณค่าตามแกน x

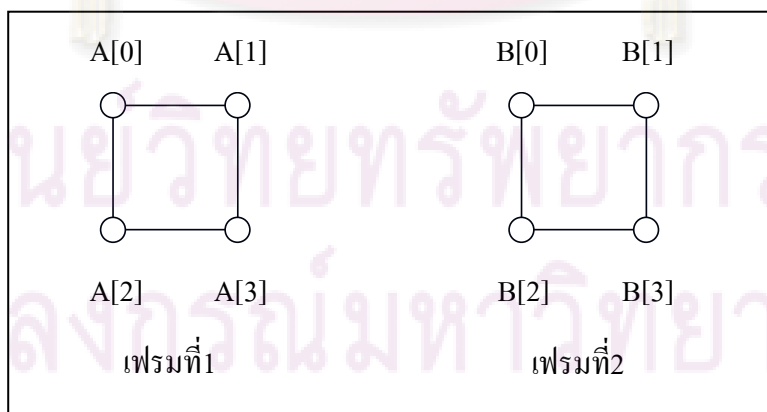
จากรูปที่ 3.11 ทำการ ประมาณค่าตามแกน y คือ ประมาณค่าระหว่าง  $A[0]$  กับ  $A[3]$  ประมาณค่าระหว่าง  $A[1]$  กับ  $A[4]$  และประมาณค่าระหว่าง  $A[2]$  กับ  $A[5]$  ซึ่งจะได้ค่าใหม่ 3 ค่า จากนั้นจะแทรกข้อมูลใหม่ไประหว่างช่วงที่เราประมาณได้ และทำการเลื่อนตำแหน่งของ array ใหม่อีกครั้ง ซึ่งจะเพิ่มความละเอียดของเฟรมภาพเป็น  $3 \times 3$  แสดงดังรูปที่ 3.12



รูปที่ 3.12 แสดงการประมาณค่าตามแกน y

### 3.1.4 การเพิ่มเฟรมภาพ

ทำการเปรียบเทียบข้อมูลระหว่างเฟรมที่ติดกัน เพื่อได้เฟรมใหม่ที่อยู่ระหว่าง 2 เฟรมนั้นๆ โดยเปรียบเทียบข้อมูลตำแหน่งเดียวของแต่ละเฟรม และทำการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น แล้วจัดเก็บในเฟรมใหม่ ก่อนที่จะแทรกระหว่างเฟรมที่เปรียบเทียบกัน ดังรูปที่ 3.13 แต่ละเฟรมมีข้อมูล  $2 \times 2$  เมื่อทำการเพิ่มเฟรมระหว่างเฟรมที่ 1 กับเฟรมที่ 2 เพื่อที่ได้เฟรมระหว่างเฟรม ที่ 1 และเฟรมที่ 2 โดยทำการประมาณค่าข้อมูลระหว่าง 2 จุด ณ ตำแหน่งเดียวกัน

รูปที่ 3.13 แสดงภาพแต่ละเฟรมมีความละเอียด  $2 \times 2$

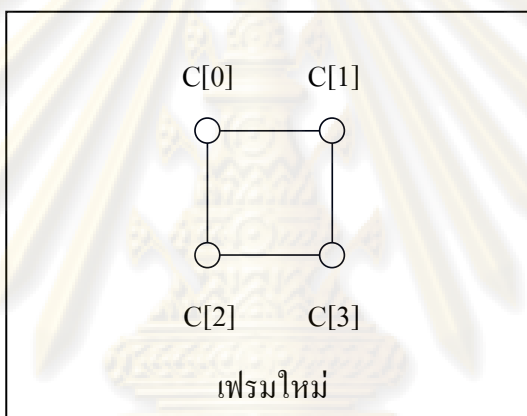
ประมาณค่าตำแหน่งที่ 0 นั่นคือ ค่า  $A[0]$  กับ  $B[0]$  จัดเก็บค่าที่ประมาณค่าได้ ไว้  
เฟรมใหม่ ณ ตำแหน่งที่ 0 นั่นคือ ค่า  $C[0]$

ประมาณค่าตำแหน่งที่ 1 นั่นคือ ค่า  $A[1]$  กับ  $B[1]$  จัดเก็บค่าที่ประมาณค่าได้ ไว้  
เฟรมใหม่ ณ ตำแหน่งที่ 1 นั่นคือ ค่า  $C[1]$

ประมาณค่าตำแหน่งที่ 2 นั่นคือ ค่า  $A[2]$  กับ  $B[2]$  จัดเก็บค่าที่ประมาณค่าได้ ไว้  
เฟรมใหม่ ณ ตำแหน่งที่ 2 นั่นคือ ค่า  $C[2]$

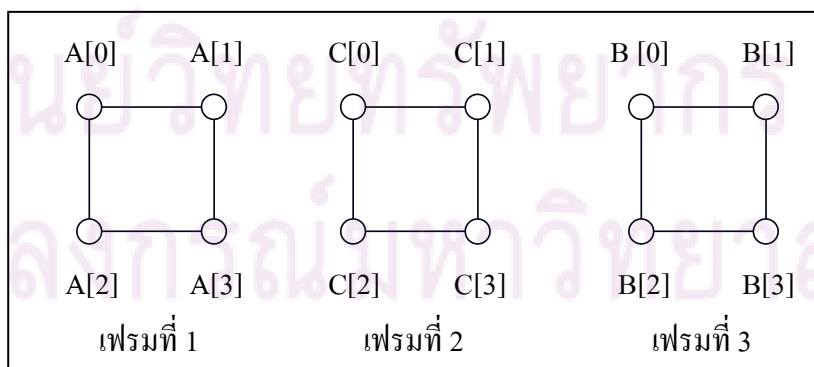
ประมาณค่าตำแหน่งที่ 3 นั่นคือ ค่า  $A[3]$  กับ  $B[3]$  จัดเก็บค่าที่ประมาณค่าได้ ไว้  
เฟรมใหม่ ณ ตำแหน่งที่ 3 นั่นคือ ค่า  $C[3]$

ซึ่งจะได้เฟรมภาพใหม่ แสดงดังรูปที่ 3.14



รูปที่ 3.14 แสดงภาพเฟรมใหม่ที่ได้จากการประมาณค่าในช่วง

จากนั้นทำการแทรกเฟรมใหม่เข้าไประหว่างเฟรมที่ทำการประมาณค่า แล้วทำการ  
เลื่อนเฟรม แสดงดังรูปที่ 3.15



รูปที่ 3.15 แสดงการแทรกเฟรมใหม่เข้าไประหว่างเฟรมที่ทำการประมาณค่า

### 3.1.5 การแปลงรหัส (Encoding)

เนื่องจากข้อมูลเชิงตัวเลขที่ได้มานั้นมีขนาดใหญ่ เพราะต้องการข้อมูลที่มีความละเอียด เพื่อให้ได้ภาพจำลองที่เสมือนจริง อีกทั้งข้อมูลเชิงตัวเลขที่ได้มานั้นมีการจัดเก็บเป็นแบบอักษร(Text File) เมื่อนำมาสร้างภาพเคลื่อนไหว 3 มิติ ปัญหาที่เกิดขึ้นตามมา คือทำให้การประมวลผลใช้เวลามากกว่าที่จำเป็น เนื่องจากคอมพิวเตอร์ไม่สามารถคำนวณตัวเลขและคำสั่งต่างๆ ได้โดยตรง จึงต้องทำการแปลงข้อมูลที่ได้รับมาเป็นภาษาเครื่องก่อน จนกระทั่งเสร็จการประมวลผลจึงแปลงภาษาเครื่องให้เป็นเลขฐานสิบออกมาอีกที ดังนั้นถ้าเราทำการแปลงข้อมูลให้อยู่ในรูปแบบบิต จะเป็นการประหยัดพื้นที่และในขณะเดียวกันจะลดเวลาในการบันทึกและอ่าน เนื่องจากบิตเป็นลักษณะข้อมูลที่เครื่องคอมพิวเตอร์ทำงานด้วยจริงๆ จึงช่วยลดขั้นตอนประมวลผลทำให้ประหยัดเวลาในการทำงาน

เนื่องจากข้อมูลที่ได้จากการจัดเก็บข้อมูลแบบอักษรนั้นจะจัดเก็บเป็นรหัสแอสกี มีขนาด 32 บิต เพราะฉะนั้นต้องใช้พื้นที่ในการจัดเก็บ 1 ตัวอักษร คือ 7 บิต ดังนั้นถ้าเพิ่มข้อมูลอักษรที่มีขนาดใหญ่มากจะต้องใช้พื้นที่ในการจัดเก็บ และในการอ่านข้อมูลจำเป็นต้องใช้เวลามากไปด้วย

เพื่อที่จะช่วยให้การประมวลผลนั้นเร็วขึ้น จึงได้ทำการแปลงข้อมูลจากเพิ่มข้อมูลตัวเลขจำนวนจริงที่ถูกเก็บเป็นแบบอักษร ไปเป็นเพิ่มข้อมูลเลขลักษณะฐานสองก่อน ทำให้ใช้พื้นที่น้อยลง ดังนั้นเวลาที่ใช้ในการดึงข้อมูลจึงทำได้เร็วขึ้น ส่งผลให้การประมวลผลนั้นเร็วขึ้นตามไปด้วยแล้วจึงนำข้อมูลเลขลักษณะฐานสองที่ได้ไปใช้ในการสร้างภาพสามมิติต่อไป

ขั้นตอนการแปลงเพิ่มข้อมูลตัวเลขที่ถูกเก็บเป็นแบบอักษร ไปเป็นเพิ่มข้อมูลเลขลักษณะฐานสอง

1. อ่านข้อมูลจากเพิ่มทีละ 1 ข้อมูล ซึ่งจะได้ข้อมูลตัวเลขทศนิยม 1 ตัว เช่น 0.000725 เป็นต้น โดยข้อมูลเลขทศนิยม 1 ตัวนั้นมีขนาด 32 บิต
2. แปลงค่าตัวเลขทศนิยม( 32บิต) ที่ได้จากข้อที่ 1 เป็นไบต์อาร์เรย์ โดยที่ 1 ไบต์อาร์เรย์จัดเก็บข้อมูล 8 บิต ดังนั้นจะได้ 4 ไบต์อาร์เรย์ เพื่อที่จะได้ข้อมูลตามตำแหน่งที่เราต้องการ จึงได้ทำการขยับบิต ดังตัวอย่างที่ 1



ตัวอย่างที่ 1 สมมุติข้อมูลเลขทศนิยม 32 บิตที่อ่านได้จากแฟ้ม คือ

$X_0X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}X_{12}X_{13}X_{14}X_{15}X_{16}X_{17}X_{18}X_{19}X_{20}X_{21}X_{22}X_{23}X_{24}X_{25}X_{26}X_{27}X_{28}X_{29}X_{30}X_{31}$

ทำการขยับเพื่อที่จะได้ข้อมูลตามตำแหน่งที่เราต้องการ ดังคำสั่งในรูปที่ 3.16 (การดำเนินการระดับบิตได้อธิบายไว้ในบทที่ 2)

```
b[0] = (byte) (bits>>24); //ขยับบิตขวา 24
b[1] = (byte) (bits>>16); //ขยับบิตขวา 16
b[2] = (byte) (bits>>8); //ขยับบิตขวา 8
b[3] = (byte) (bits);
```

รูปที่ 3.16 คำสั่งในการขยับบิตขวาเพื่อเก็บข้อมูลตัวเลขทศนิยมได้ครบ 32 บิต

ดังนั้น ไบต์อาร์เรย์ทั้ง 4 ที่ได้ คือ

ไบต์อาร์เรย์ที่ 1 =  $X_0X_1X_2X_3X_4X_5X_6X_7$

ไบต์อาร์เรย์ที่ 2 =  $X_8X_9X_{10}X_{11}X_{12}X_{13}X_{14}X_{15}$

ไบต์อาร์เรย์ที่ 3 =  $X_{16}X_{17}X_{18}X_{19}X_{20}X_{21}X_{22}X_{23}$

ไบต์อาร์เรย์ที่ 4 =  $X_{24}X_{25}X_{26}X_{27}X_{28}X_{29}X_{30}X_{31}$

- จัดเก็บข้อมูลในรูปแบบไบต์อาร์เรย์ ลงแฟ้มข้อมูลใหม่ เช่น จากแฟ้มข้อมูลเดิมชื่อ 000.grd จัดเก็บลงแฟ้มข้อมูลใหม่ชื่อ b000.grd
- ทำการบีบอัดแฟ้มให้อยู่ในรูปแบบ ZIP

หลังจากแปลงเป็นไบต์อาร์เรย์แล้วทำการบีบอัดข้อมูล จะเห็นได้ว่าขนาดของแฟ้มข้อมูลเล็กกว่าเดิมดังตารางที่ 3.2

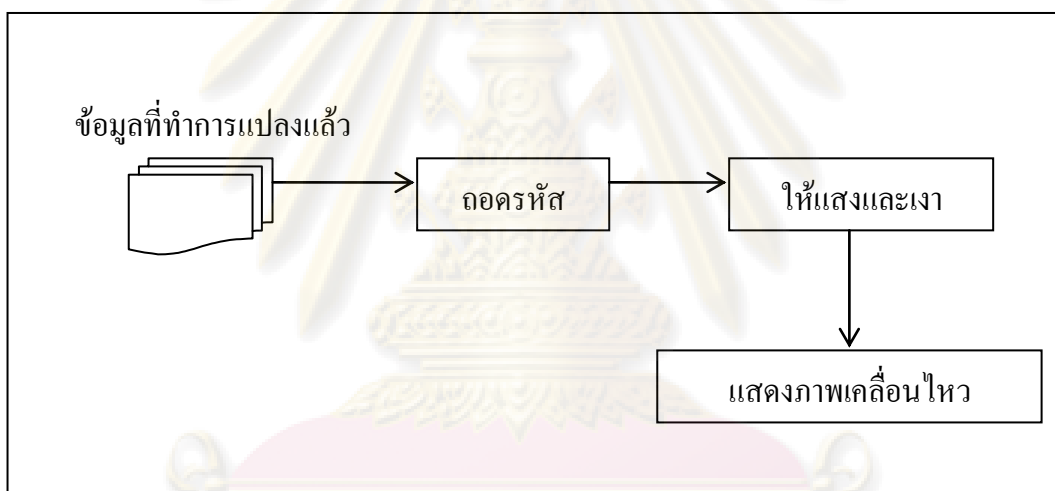
ตารางที่ 3.2 เปรียบเทียบขนาดข้อมูลระหว่างแฟ้มข้อมูลอักษรกับแฟ้มข้อมูลเลขลักษณะฐานสอง

บริเวณ	ขนาดแฟ้มข้อมูลก่อนแปลง	ขนาดแฟ้มข้อมูลหลังแปลง
มหาสมุทรอินเดีย	8.36 GB	1.86 GB
ชายฝั่งภาคใต้ของประเทศไทย	2.74 GB	628 MB
จังหวัดพังงา	34.4 MB	16.4 MB

เมื่อทำการแปลงข้อมูลเป็นแฟ้มข้อมูลเลขลักษณะฐานสองและทำการบีบอัดเสร็จ จะนำข้อมูลในส่วนนี้ไปทำการสร้างภาพสามมิติในขั้นตอนการส่วนที่ 2 ต่อไป

### 3.2 วิธีดำเนินการส่วนที่ 2

เป็นขั้นตอนการให้แสงและเงา โดยนำข้อมูลที่ทำการแปลงรหัสแล้ว ในขั้นตอนดำเนินการส่วนที่ 1 มาจำลองสีนามิในรูปแบบสามมิติ แสดงผลงานได้ดังรูปที่ 3. 17 ซึ่งการทำงานในส่วนนี้จะเริ่มตั้งแต่การถอดรหัสดจากแฟ้มข้อมูลเลขลักษณะฐานสอง ตลอดจนแสดงเป็นภาพในแต่ละเฟรม เมื่อนำภาพในแต่ละเฟรมมาแสดงต่อเนื่องกัน จึงได้เป็นภาพเคลื่อนไหว



รูปที่ 3.17 แผนผังการทำงานขั้นตอนดำเนินการส่วนที่ 2

#### 3.2.1 การถอดรหัส(Decoding)

อ่านข้อมูลจากแฟ้มข้อมูลที่ทำการบีบอัดขึ้นมาเป็นเลขลักษณะฐานสอง แล้วทำการแปลงข้อมูลจากข้อมูลเลขลักษณะฐานสองที่อ่านได้ไปเป็นข้อมูลเลขจำนวนจริงเพื่อนำไปดำเนินการสร้างภาพต่อไป

ขั้นตอนการแปลงข้อมูลเลขลักษณะฐานสอง ไปเป็นข้อมูลเลขทศนิยม

1. อ่านค่าที่ละ 4 ไบต์อาร์เรย์
2. แต่ละอาร์เรย์ทำการขยับบิต และand(&) 32 บิต เพื่อให้ข้อมูลเป็น 32 บิต โดยแต่ละช่วงที่เราต้องการจะตั้งค่าเป็นค่า 11111111 ของแต่ละช่วง ดังตัวอย่างคำสั่งดังรูปที่ 3.18 ซึ่งเมื่อนำมาทำการ and บิต แล้วข้อมูลที่เราได้อ่านได้ยังคงอยู่ในช่วงที่เราต้องการ อธิบายได้ดังตารางที่ 3.3 และจัดเก็บไว้ในอีกตัวแปรหนึ่ง (การดำเนินการระดับบิตได้อธิบายไว้ในบทที่ 2)

```
b1 = (byteBuffer[0] << 24) & 0xFF000000;
b2 = (byteBuffer[1] << 16) & 0x00FF0000;
b3 = (byteBuffer[2] << 8) & 0x0000FF00;
b4 = byteBuffer[3] & 0x000000FF;
```

รูปที่ 3.18 คำสั่งในการขยับบิตซ้ายและดำเนินการand บิตกับช่วงข้อมูลที่ต้องการ เพื่อให้ได้ 32 บิต

ตารางที่ 3.3 แสดงการ and บิต จากรูปคำสั่งที่ 3.18

	ขยับบิตค่าที่อ่านได้	ช่วงที่ต้องการเก็บ	ไบต์อาร์เรย์ที่ทำการ & ช่วงที่ต้องการเก็บ
b1	byteBuffer[0] << 24	11111111 00000000 00000000 00000000	byteBuffer[0] 00000000 00000000 00000000
b2	byteBuffer[1] << 16	00000000 11111111 00000000 00000000	00000000 byteBuffer[1] 00000000 00000000
b3	byteBuffer[2] << 8	00000000 00000000 11111111 00000000	00000000 00000000 byteBuffer[2] 00000000
b4	byteBuffer[3]	00000000 00000000 00000000 11111111	00000000 00000000 00000000 byteBuffer[3]

3. ใช้ตัวดำเนินการ or บิต เพื่อรวมให้เป็นข้อมูลเลขลักษณะฐานสอง 32 บิต ตามที่ต้องการ จัดเก็บไว้ในอีกตัวแปรหนึ่ง ดังตัวอย่างคำสั่งในรูปที่ 3.19 และอธิบายไว้ในตาราง 3.4

```
bits = b1 | b2 | b3 | b4;
```

รูปที่ 3.19 คำสั่งในการ or บิต ซึ่งเป็นค่าเลขทศนิยมที่ต้องการ 32 บิต

ตารางที่ 3.4 แสดงการ or บิต จากรูปคำสั่งที่ 3.19

Bit ที่	31-24	23-16	15-8	7-0
b1	byteBuffer[0]	00000000	00000000	00000000
b2	00000000	byteBuffer[1]	00000000	00000000
b3	00000000	00000000	byteBuffer[2]	00000000
b4	00000000	00000000	00000000	byteBuffer[3]
b1 b2 b3 b4	byteBuffer[0]	byteBuffer[1]	byteBuffer[2]	byteBuffer[3]

4. ค่าที่ได้อ่านจะถูกเก็บไว้ในบัพเฟอร์ เพื่อนำไปสร้างภาพต่อไป

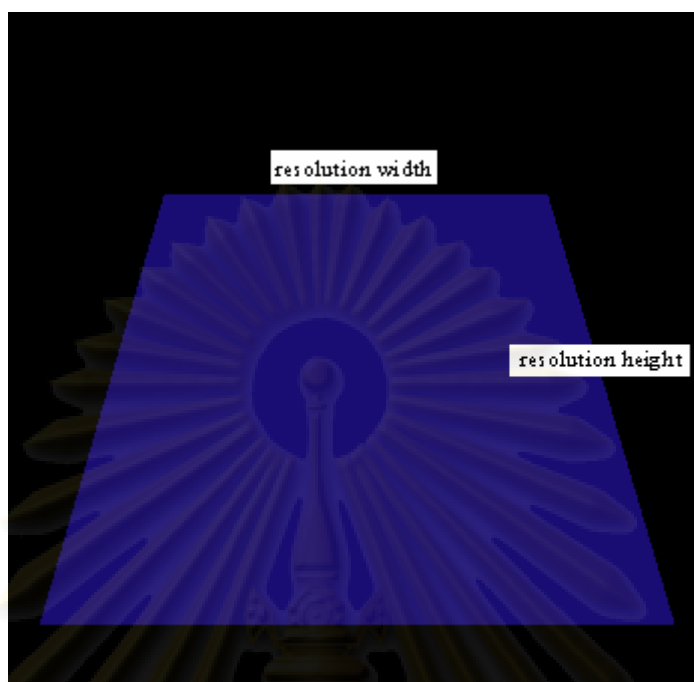
### 3.2.2 การให้แสงและเงา(Rendering)

ส่วนนี้เป็นส่วนที่สำคัญที่สุดในการจำลองภาพสามมิติ โดยเราจะใช้ขั้นตอนวิธีซีบัพเฟอร์ในการสร้างพื้นผิว โดยการนำข้อมูลความสูงของแต่ละจุดจัดเก็บ อยู่ในบัพเฟอร์ ในหัวข้อ 3.2.1 เพื่อนำมาสร้างภาพ

#### ขั้นตอนการให้แสงและเงา

- เราจะเริ่มจากการสร้างระนาบเรขาคณิต (geometry plane) โดยใช้ความละเอียดที่อ่านได้ คือ ค่าความละเอียดของภาพทางกว้าง(resolution width) และค่าความละเอียดของภาพทางสูง(resolution height) มาสร้างได้ดังรูปที่ 3.20

ศูนย์วิจัยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.20 แสดงระนาบเรขาคณิต

- หลังจากนั้น จะจัดการค่าที่ได้จากการแปลงข้อมูลแล้วในบัพเฟอร์ โดยจะจัดเก็บเป็นอาร์เรย์ 1 มิติ เพื่อใช้สำหรับการอ้างอิงค่าที่จะนำไปสร้างภาพ 3 มิติ

จากรูปแบบอาร์เรย์สามมิติ :  $A[x1][y1][z1]$  ,  $A[x2][y2][z2]$  , ...,  $A[xN][yN][zN]$

โดยที่ ค่า  $x$  คือ ค่าของตำแหน่งคอลัมน์

ค่า  $y$  คือ ค่าของตำแหน่งแถว

ค่า  $z$  คือ ค่าความสูงของน้ำทะเล หรือพื้นดิน

ค่า  $N$  คือ จำนวนจุดที่ใช้ในการคำนวณ (grid point)

ซึ่งสามารถอธิบายเป็นจุดที่เก็บค่าของอาร์เรย์ 1 มิติ ได้ดังรูปที่ 3.21

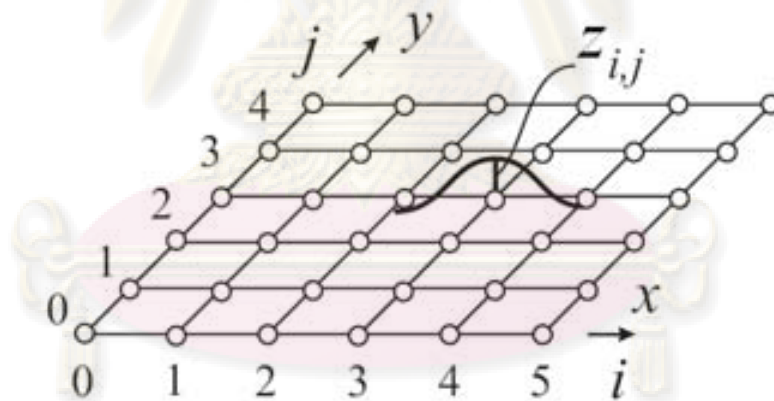
ศูนย์วิจัยทรัพยากรทางทะเล  
จุฬาลงกรณ์มหาวิทยาลัย

$p[0]$	$p[1]$	$p[2]$	...	$p[w-1]$
$p[w]$	$p[w+1]$	$p[w+2]$	...	$p[(w*2)-1]$
.	.	.	...	.
.	.	.	...	.
$p[w*(h-1)]$	$p[w*(h-1)+1]$	$p[w*(h-1)+2]$	...	$p[(w*h)-1]$

$w = \text{resolution width}$   
 $h = \text{resolution height}$

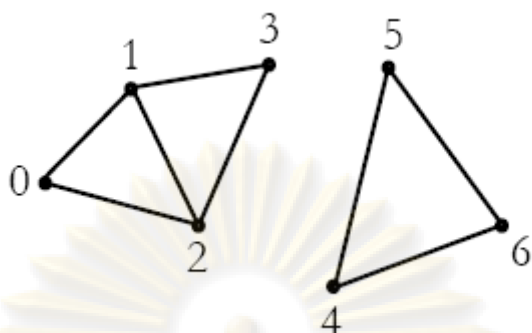
รูปที่ 3.21 แสดงการจัดเก็บค่าเป็นอาร์เรย์ 1 มิติ

ดังนั้นจากรูปที่ 3.21 จะได้ว่า  $p[0] = [x1]$  คือ ค่าตำแหน่งคอลัมน์ ,  $p[1] = [y1]$  คือ ค่าตำแหน่งแถว ,  $p[2] = [z1]$  คือ ค่าความสูงของน้ำทะเล และพื้นดิน เป็นต้น ดังนั้น แต่ละจุดภาพจะมีข้อมูลของความสูงของระดับน้ำทะเลหรือความสูงของพื้นดินที่แตกต่างกันออกไป ซึ่งเป็นวิธีที่นิยมใช้ในการสร้างภาพ[4][6] แสดงดังรูปที่ 3.22



รูปที่ 3.22 แสดงตำแหน่งภาพเรขาคณิตจากจุดอ้างอิง

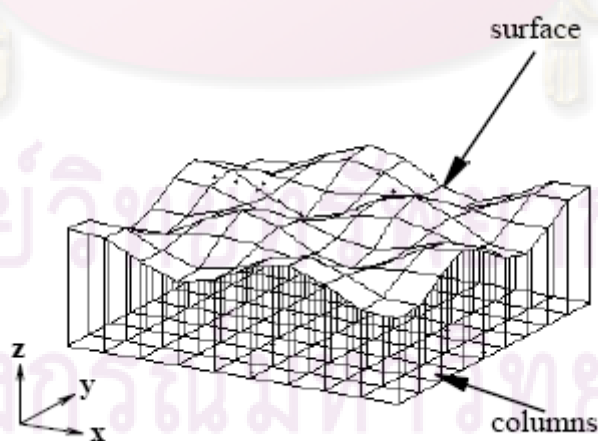
- จากนั้นจะทำการสร้างภาพเรขาคณิตจากข้อมูลที่เก็บอยู่ในบัพเฟอร์ ในการคำนวณจะอ้างอิงจากจุดอ้างอิง ซึ่งแต่ละจุดภาพจะมีข้อมูลของความสูงของระดับน้ำทะเลหรือความสูงของพื้นดินที่แตกต่างกันออกไป



รูปที่ 3.23 แสดงภาพตัวอย่างจุดยอด

โดยใช้วิธีการTriangleStripArray [19][23] คือการจัดเตรียมจุดยอด เพื่อสร้างรูปสามเหลี่ยม โดยมีอาร์เรย์ของ stripVertexCounts ซึ่งเป็นกลุ่มของจุดยอดที่กลายเป็น strip โดยแต่ละ strip นิยามให้เป็น ชั้นของรูปสามเหลี่ยม ดังนั้นจะต้องมีอย่างน้อย 3 จุดยอด ภายใน ชั้นของรูปสามเหลี่ยม แต่ละจุดยอดจะเริ่มต้นจากจุดยอดที่สามใน strip นั้น นิยามรูปสามเหลี่ยมโดยใช้จุดนั้น และอีก 2 จุดก่อนหน้า การกำหนดตำแหน่งของรูปสามเหลี่ยม ดังต่อไปนี้

จุดยอด 0 , 1 และ 2 กำหนดให้กับรูปสามเหลี่ยมแรก รูปสามเหลี่ยมถัดไปกำหนดโดยจุดยอด 1 , 2 และ 3 ในการกำหนดตำแหน่ง ทำโดยการย้อนกลับลำดับจุดยอดของรูปสามเหลี่ยมก่อนหน้าสำหรับ 2 จุด ทำการต่อกับรูปสามเหลี่ยมปัจจุบัน ดังนั้น เราจะได้ลำดับ 2 , 1 , 3 รูปสามเหลี่ยมถัดไปจะมีจุดยอด 2 , 3 , 4 ถัดไปเป็น 4 , 3 , 5 แต่ละ strip เริ่มการจัดเรียงตัวเองจากรูปสามเหลี่ยมแรก



รูปที่ 3.24 แสดงภาพพื้นผิวที่สร้างจากจุดอ้างอิง

• หลังจากที่ได้ทำภาพเรขาคณิตจากความสูงของแต่ละตำแหน่งแล้วดังรูปที่ 3.24 จะทำการใส่สีของพื้นดินและพื้นน้ำ ตามสีที่ได้ทำการกำหนด และจัดเก็บไว้ในบัฟเฟอร์ สีที่ใช้จะเกิดจากการผสมของแม่สีแสงมาผสมเพื่อให้เกิดเป็นสีตัวอื่นๆ ซึ่งแม่สีแสงนั้นจะประกอบด้วย สีแดง สีเขียว และสีน้ำเงิน (Red Green Blue, RGB) โดยมีอัตราส่วนความเข้มของแม่สีแต่ละตัวเป็นตัวแปรสำคัญที่ทำให้ปรากฏสีสันต่างๆ ที่หลากหลายในธรรมชาติ ระดับความอ่อนแก่ของแม่สีแต่ละตัวมีค่าความอ่อนแก่เป็นตัวเลขในช่วงระหว่าง 0 ถึง 1.0

โดยเราจะทำการ กำหนดสีส่วนของพื้นดินเป็นโทนสี น้ำตาล ตามระดับความสูงของพื้นดิน ถ้าระดับความสูงมากขึ้น สีจะอ่อนลง ดังตัวอย่างสมการต่อไปนี้

$$\text{RED} = \frac{Z}{\text{Max}} ; \quad \text{GREEN} = \frac{Z}{\text{Max}} ; \quad \text{BLUE} = 0 ;$$

ซึ่ง RED คือ สีแดง  
 GREEN คือ สีเขียว  
 BLUE คือ สีน้ำเงิน  
 Z คือ ค่าความสูงของพื้นดิน แต่ละจุดภาพ  
 Max คือ ค่าสูงสุดความสูงของพื้นดิน

โดยจะเอาค่าความสูงแต่ละจุดภาพมาหารด้วยค่าสูงสุดความสูงของพื้นดิน เพื่อให้ค่าสีนั้นอยู่ในช่วง 0 ถึง 1.0 นั้นเอง

และกำหนดสีส่วนของพื้นน้ำ เป็นโทนสีน้ำเงิน ตามระดับความลึกของน้ำ ถ้ามีความลึกมาก ท้องน้ำจะเป็นสีดำ ดังตัวอย่างสมการต่อไปนี้

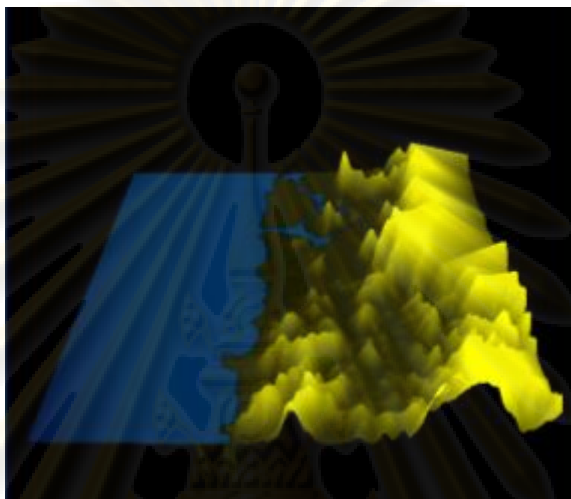
$$\text{RED} = 0 ; \quad \text{GREEN} = -0.2*Z + 0.2 ; \quad \text{BLUE} = -0.2*Z + 0.4 ;$$

ซึ่ง RED คือ สีแดง  
 GREEN คือ สีเขียว  
 BLUE คือ สีน้ำเงิน  
 Z คือ ค่าความสูงของพื้นน้ำ แต่ละจุดภาพ

ซึ่งรายละเอียดการกำหนดสีของภาพทั้งหมดแสดงใน (ภาคผนวก ข.)



- นอกจากนั้น ได้ทำการกำหนดแสง เนื่องจากการแสดงภาพเป็นลักษณะภาพมูมกว้างจึงเลือกใช้แสงล้อมรอบ นั่นก็คือการเพิ่มปริมาณแสงให้กับทุกๆ จุดบนพื้นผิวของวัตถุด้วยปริมาณแสงที่เท่ากัน แสดงดังรูปที่ 3.25



รูปที่ 3.25 แสดงตัวอย่างในการใส่สีและแสงให้แต่ละระดับความสูงของพื้นดิน

เพื่อให้สามารถเห็นมุมมองที่ชัดเจน ยังเพิ่มฟังก์ชันการทำงานสำหรับแสดงแบบจำลองสีนามิ โดยใช้เมาท์เป็นตัวควบคุมภาพ ดังนี้

- ใช้ปุ่มซ้ายของเมาท์ค้างในการเปลี่ยนมุมมองของภาพ (360 องศา)
- ใช้ปุ่มกลางทำการซูมภาพเข้าและออก
- ใช้ปุ่มขวาของเมาท์ค้างในการเคลื่อนย้ายภาพ (ขึ้น-ลง , ซ้าย-ขวา)

เมื่อทำการให้แสงและเงาในแต่ละเฟรมเสร็จสิ้น จะทำการแสดงภาพแต่ละเฟรมต่อเนื่องกัน ก็จะเห็นเป็นภาพเคลื่อนไหวในที่สุด

ศูนย์วิจัยคอมพิวเตอร์  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 4

### การประมวลผลข้อมูล

#### 4.1 ผลจากการประมวลผล

ผลการจำลองสีนามีสามมิติโดยใช้เครื่องคอมพิวเตอร์ Note Book , Intel(R) Celeron(R)M CPU 420@1.6GHz , 512MB RAM ในการรันโปรแกรมทั้งหมด

ในงานวิจัยนี้เนื่องจากข้อมูลในส่วนนี้ยังไม่ได้มีคนนำมาจำลอง จึงได้ทำการทดลองใน 2 รูปแบบเพื่อเปรียบเทียบกัน นั่นคือรูปแบบที่ใช้โดยทั่วไปที่จัดเก็บในรูปแบบตัวอักษร และรูปแบบเลขล็กษณฐานสองที่เราได้ทำการแปลง ซึ่งถ้าทำการจำลองสีนามิโดยใช้ข้อมูลจริงที่ไม่ได้ทำการแปลงค่า จะใช้เวลานานมากในการอ่านข้อมูลมาเก็บในบัฟเฟอร์ก่อนที่จะทำการให้แสงและเงา แสดงได้ดังตารางที่ 4.1 ส่วนการสร้างภาพจะใช้วิธีการทั่วไปโดยใช้ ขั้นตอนวิธีซีบัฟเฟอร์เป็นหลัก

ตารางที่ 4.1 แสดงเวลาในการอ่านข้อมูลมาจัดเก็บในบัฟเฟอร์ก่อนทำการแปลงข้อมูล

บริเวณ	จำนวนจุดที่ทำการคำนวณ	จำนวนเฟรม	เวลาในการอ่านข้อมูล
มหาสมุทรอินเดีย	1793x2191	150	1062.328 วินาที
ชายฝั่งภาคใต้ของประเทศไทย	560x1152	300	351.172 วินาที
จังหวัดพังงา	145x145	220	8.282 วินาที

จากตารางที่4.1 ด้วยวิธีการอ่านข้อมูลรูปแบบที่ใช้โดยทั่วไปที่จัดเก็บในรูปแบบตัวอักษร จะเห็นได้ว่าบริเวณมหาสมุทรอินเดียต้องใช้เวลา 1062.328 วินาที หรือประมาณ 17 นาที และบริเวณชายฝั่งภาคใต้ของประเทศไทยต้องใช้เวลา 351.172 วินาที หรือประมาณ 6 นาที ซึ่งใช้เวลาในการอ่านข้อมูลมาก ทำให้เป็นอุปสรรคในการให้แสงและเงา ยิ่งขนาดข้อมูลเพิ่มมากขึ้น ก็จะทำให้ใช้เวลาในการอ่านข้อมูลมากขึ้น

จึงได้ทำการทดลองแปลงข้อมูลให้อยู่ในรูปแบบเลขล็กษณฐานสอง ก่อนนำมาจำลองภาพ เคลื่อนไหว สามมิติ (ดังที่อธิบายในบทที่ 3) เพื่อลดเวลาในการอ่านข้อมูลมาเก็บในบัฟเฟอร์ให้น้อยลง ทำให้เวลาอ่านข้อมูลไม่เป็นอุปสรรคในการให้แสงและเงา ดังแสดงในตารางที่

ตารางที่ 4.2 แสดงเวลาในการอ่านข้อมูลมาจัดเก็บในบัฟเฟอร์หลังทำการแปลงข้อมูล

บริเวณ	จำนวนจุดที่ทำการ คำนวณ	จำนวนเฟรม	เวลาในการอ่านข้อมูล
มหาสมุทรอินเดีย	1793x2191	150	3.625 วินาที
ชายฝั่งภาคใต้ของประเทศไทย	560x1152	300	2.344 วินาที
จังหวัดพังงา	145x145	220	0.875 วินาที

จากตารางที่ 4.2 ด้วยข้อมูลในรูปแบบเลขลักษณะฐานสอง จะเห็นได้ว่าบริเวณมหาสมุทรอินเดียใช้เวลาในการอ่านข้อมูลมาจัดเก็บในบัฟเฟอร์เพียง 3.625 วินาที บริเวณชายฝั่งภาคใต้ของประเทศไทยใช้เวลาเพียง 2.344 วินาที และบริเวณจังหวัดพังงาใช้เวลาเพียง 0.875 วินาที เท่านั้น

เมื่อนำเวลาทั้ง 2 รูปแบบมาเปรียบเทียบกัน นั่นคือ ข้อมูล ที่จัดเก็บในรูปแบบตัวอักษรแบบทั่วไป กับรูปแบบเลขลักษณะฐานสองที่เราได้ทำการแปลง แสดงดังตารางที่ 4.3

ตารางที่ 4.3 เปรียบเทียบเวลาในการอ่านข้อมูลมาจัดเก็บในบัฟเฟอร์

บริเวณ	รูปแบบทั่วไป	รูปแบบเลขลักษณะฐานสอง
มหาสมุทรอินเดีย	1062.328 วินาที	3.625 วินาที
ชายฝั่งภาคใต้ของประเทศไทย	351.172 วินาที	2.344 วินาที
จังหวัดพังงา	8.282 วินาที	0.875 วินาที

จากตารางที่ 4.3 จะเห็นได้ว่าเราสามารถลดเวลาในการอ่านข้อมูลมาจัดเก็บในบัฟเฟอร์ ได้มากขึ้นเลยทีเดียว

หลังจากนั้นเมื่อนำข้อมูลที่ทำการแปลงข้อมูลแล้วมาทำการให้แสงและเงา เวลาที่ใช้ในการแสดงภาพเคลื่อนไหวสามมิติ แสดงดังตารางที่ 4.4 ซึ่งได้ผลอัตราความเร็วในการแสดงภาพเป็นเฟรมต่อวินาที

ตารางที่ 4.4 แสดงอัตราความเร็วเฟรมต่อวินาทีในการให้แสงและเงา

บริเวณ	จำนวนจุดที่ทำการ คำนวณ	จำนวนเฟรม	อัตราความเร็ว(FPS)
มหาสมุทรอินเดีย	1793x2191	150	1.43
ชายฝั่งภาคใต้ของประเทศไทย	560x1152	300	5.52
จังหวัดพังงา	145x145	220	10.67

จากตารางที่ 4.4 แสดงอัตราความเร็วการประมวลผลเฟรมต่อวินาที ในการให้แสงและเงาของข้อมูลที่ทำกรแปลงเป็นเลขลัษณฐานสอง บริเวณมหาสมุทรอินเดีย บริเวณชายฝั่งภาคใต้ของประเทศไทย และบริเวณจังหวัดพังงา โดยใช้ข้อมูล จากแฟ้มที่ได้ทำการรวมพื้นดินกับพื้นน้ำ (ดังอธิบายใน บทที่ 3)

เมื่อเราได้ทำการเปรียบเทียบอัตราความเร็วการประมวลผลเฟรมต่อวินาทีในการให้แสงและเงา ระหว่างข้อมูลที่ไม่ได้ทำการแปลง กับข้อมูลที่ทำกรแปลงเป็นเลขลัษณฐานสอง จะได้ผลดังตารางที่ 4.5

ตารางที่ 4.5 เปรียบเทียบเวลาอัตราความเร็วการประมวลผลเฟรมต่อวินาที

บริเวณ	รูปแบบทั่วไป	รูปแบบเลขลัษณฐานสอง
มหาสมุทรอินเดีย	0.14	1.43
ชายฝั่งภาคใต้ของประเทศไทย	0.85	5.52
จังหวัดพังงา	7.58	10.67

จากตารางที่ 4.5 จะเห็นได้ว่าอัตราความเร็วในการให้แสงและเงา ในรูปแบบเลขลัษณฐานสอง ช่วยลดเวลาในการประมวลผล ทำให้สามารถแสดงภาพเคลื่อนไหวสามมิติได้โดยไม่เกิดการกระตุก

#### 4.2 วิเคราะห์ผลการทดลอง

สังเกตจากผลการทดลอง จะเห็นได้ว่าบริเวณจังหวัดพังงานั้นมีความละเอียดของภาพน้อย ทำให้เมื่อเวลาทำการขยายภาพ(zoom-in) จะทำให้ภาพแตกไม่สามารถดูได้ จึงได้ทำการทดลองต่อไป โดยทำการเพิ่มความละเอียดของภาพ โดยวิธีการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น(อธิบายในบทที่2) เนื่องจากที่เลือกใช้วิธีการแบบเชิงเส้นนี้ เพราะเป็นวิธีที่ง่าย และจะได้ข้อมูลที่อยู่ในช่วงเป็นที่น่าสนใจได้ เราสามารถเลือกที่จะใช้วิธีการประมาณค่าในช่วงเสมือนพหุนาม(Spline Interpotation) วิธีการอื่น ได้ อย่างเช่น แบบพหุนามดีกรี2 (Quadratic Spline) หรือแบบพหุนามดีกรี3(Cubic Spline) ซึ่งใช้ต้องเวลาประมวลผลที่มากขึ้น

เพื่อเพิ่มความ เรียบของภาพจึงได้ทำการทดลองต่อไป โดยทำการเพิ่มเฟรมภาพ เพื่อให้ได้เฟรมภาพที่มากขึ้น จะทำให้ภาพมีความต่อเนื่องมากขึ้น โดยบริเวณที่ทำการศึกษา เราจะเลือกบริเวณจังหวัดพังงา เนื่องจากบริเวณนี้มีข้อมูลน้อย และเพื่อที่จะ ได้ดูการเกิดคลื่นที่ดียิ่งขึ้น โดยทำการเพิ่มเฟรมภาพ ด้วยวิธีการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น(ดังที่อธิบายใน บทที่2) เช่นเดียวกับการเพิ่มข้อมูล

#### 4.2.1 ผลการเพิ่มจุดที่ทำการคำนวณ

การเพิ่มความละเอียดของภาพบริเวณจังหวัดพังงา จะทำการเพิ่มในแนวนอน(แกน x) และแนวตั้ง(แกน y) จากความละเอียด ข้อมูล  $N \times N$  เมื่อทำการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น เราจะได้ข้อมูลเป็น  $(2N-1) \times (2N-1)$  (อธิบายในบทที่ 3) เมื่อนำมาเปรียบเทียบ ความละเอียดก่อนและหลังทำการเพิ่มจุดที่ใช้ในการคำนวณ แสดงดังตารางที่ 4.6

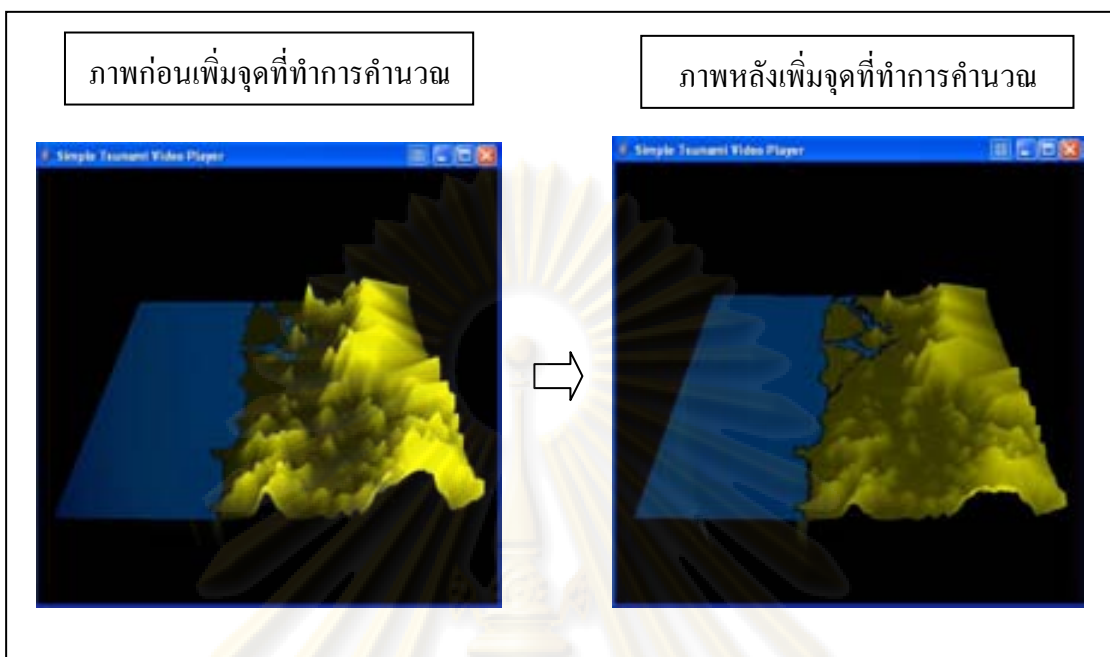
ตารางที่ 4.6 เปรียบเทียบความละเอียดของข้อมูลก่อนและหลังทำการเพิ่มจุดที่ใช้ในการคำนวณ

บริเวณ	ความละเอียดก่อน	ความละเอียดหลัง
จังหวัดพังงา	145x145	289x289

จากตารางที่ 4.6 จะเห็นได้ว่าบริเวณจังหวัดได้เพิ่มความละเอียดของแต่ละเฟรม ภาพจาก 145 x145 จุดที่ใช้คำนวณ เป็น 289x289 จุดที่ใช้คำนวณ

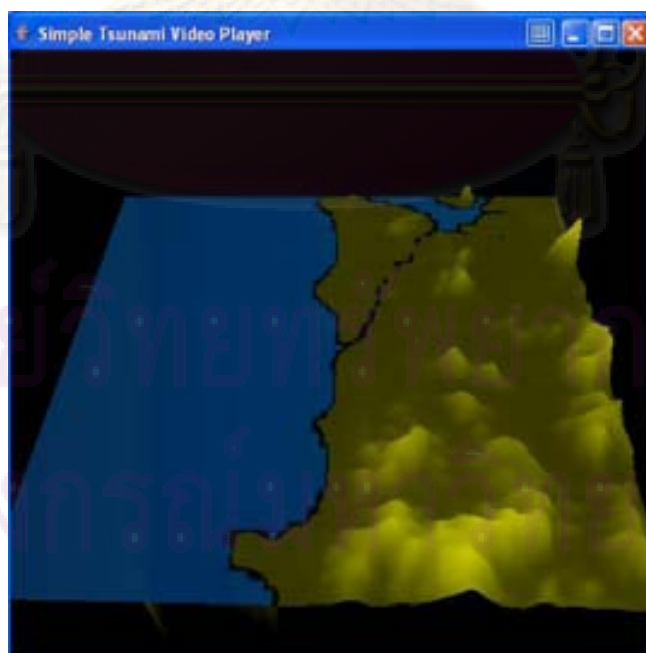
โดยภาพที่ทำการจำลองหลังจากทำการเพิ่มจำนวนจุดที่ทำการคำนวณ เมื่อมา เปรียบเทียบกับภาพที่ทำการจำลองโดยที่ไม่ได้เพิ่มจำนวนจุดที่ทำการคำนวณ ดังรูปที่ 4.1

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.1 เปรียบเทียบภาพระหว่างภาพก่อนและหลังทำการเพิ่มจำนวนจุดที่ทำการคำนวณ บริเวณจังหวัดพังงา

หลังจากทำการ เพิ่มความละเอียดของภาพบริเวณจังหวัดพังงา แล้ว ได้ทำการ ทดสอบการขยายภาพ(zoom-in) บริเวณนี้ เข้าไปดังรูปที่ 4.2 จะเห็นได้ว่าภาพนั้นสามารถดูได้



รูปที่ 4.2 แสดงการขยายภาพบริเวณจังหวัดพังงาของข้อมูลที่ทำการเพิ่มความละเอียดเป็น289x289

#### 4.2.2 ผลการเพิ่มเฟรมภาพ

การเพิ่มเฟรมภาพบริเวณจังหวัดพังงา จะทำการเพิ่มเฟรมภาพจากจำนวน  $N$  เมื่อทำการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น เราจะได้เฟรมภาพเป็น  $2N-1$  (อธิบายในบทที่ 3) เมื่อนำมาเปรียบเทียบจำนวนเฟรมภาพก่อนและหลังทำการเพิ่มเฟรมภาพ แสดงดังตารางที่ 4.7

ตารางที่ 4.7 เปรียบเทียบจำนวนเฟรมก่อนและหลังทำการเพิ่มจุดที่ใช้ในการคำนวณ

บริเวณ	จำนวนเฟรมก่อน	จำนวนเฟรมหลัง
จังหวัดพังงา	220	439

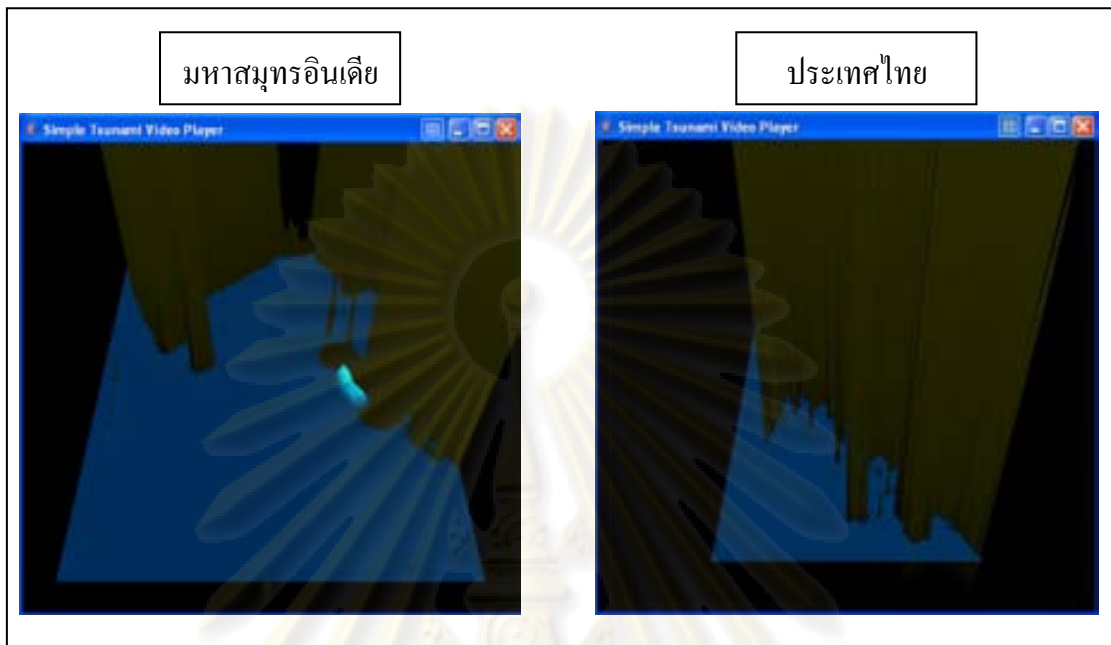
เมื่อเราทำการเพิ่มความละเอียดของข้อมูล และเพิ่มเฟรมภาพบริเวณจังหวัดพังงา แล้ว เราได้ทำการแปลงข้อมูลที่ได้ใหม่นั้นเป็นข้อมูลเลขถัษณฐานสอง แล้วทำการสร้างภาพสามมิติโดยทำการให้แสงและเงา เวลาที่ใช้ในการแสดงภาพเคลื่อนไหวสามมิติ แสดงดังตารางที่ 4.8 ซึ่งได้ผลอัตราความเร็วในการแสดงภาพเป็นเฟรมต่อวินาที

ตารางที่ 4.8 แสดงอัตราความเร็วเฟรมต่อวินาทีในการให้แสงและเงาหลังเพิ่มข้อมูลและเฟรมภาพ

บริเวณ	จำนวนจุดที่ทำการคำนวณ	จำนวนเฟรม	อัตราความเร็ว(FPS)
จังหวัดพังงา	289x289	439	7.24

#### 4.2.3 ผลการเปรียบเทียบภาพเมื่อปรับลดสเกล

เมื่อทำการวิเคราะห์ภาพที่แสดงทั้งส่วนของพื้นดินและน้ำ จะเห็นว่าปัญหาในการแสดงภาพ จำลองจะเกิดขึ้นตรงส่วนความสูงของพื้นดินตามความจริง ซึ่งจะสามารถสังเกตเห็นได้จากภาพมหาสมุทรอินเดีย บริเวณชายฝั่งทะเลภาคใต้ของประเทศไทย และจังหวัดพังงา ว่ามีความแตกต่างระหว่างความสูงของน้ำทะเลและพื้นดิน เช่น ความสูงของคลื่นที่สูงสุดมีความสูงแค่ 7.02 เมตร ขณะที่ความสูงของพื้นดินที่สูงสุดมีความสูงถึง 3047.50 เมตร ดังรูปที่ 4.3 โดยแสดงเพียง 2 บริเวณ คือบริเวณมหาสมุทรอินเดีย และบริเวณชายฝั่งทะเลภาคใต้ของประเทศไทย

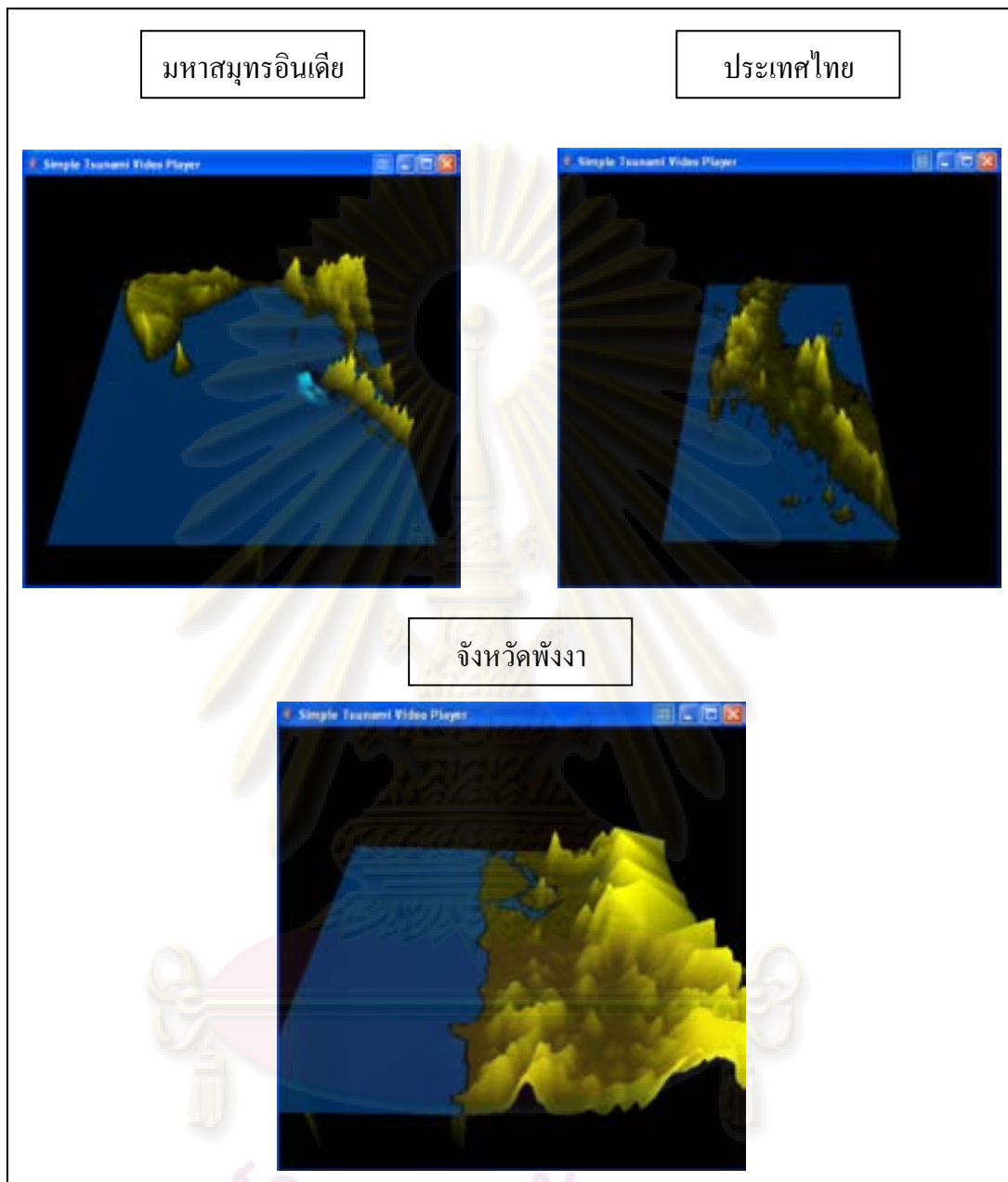


รูปที่ 4.3 แสดงภาพบริเวณมหาสมุทรอินเดียและบริเวณชายฝั่งทะเลภาคใต้ของประเทศไทย โดยไม่ได้ปรับลดสเกลของความสูงของพื้นดิน

จึงทำให้จำเป็นที่จะต้องปรับสเกลความสูงของพื้นดิน เพื่อให้ภาพที่แสดงนั้นเหมาะสม ทั้ง 3 บริเวณศึกษา โดยที่รูปที่ 4.4 ได้ทำการปรับลดความสูงของส่วนพื้นดิน 100 เท่า รูปที่ 4.5 ปรับลดความสูงของส่วนพื้นดิน 300 เท่า

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

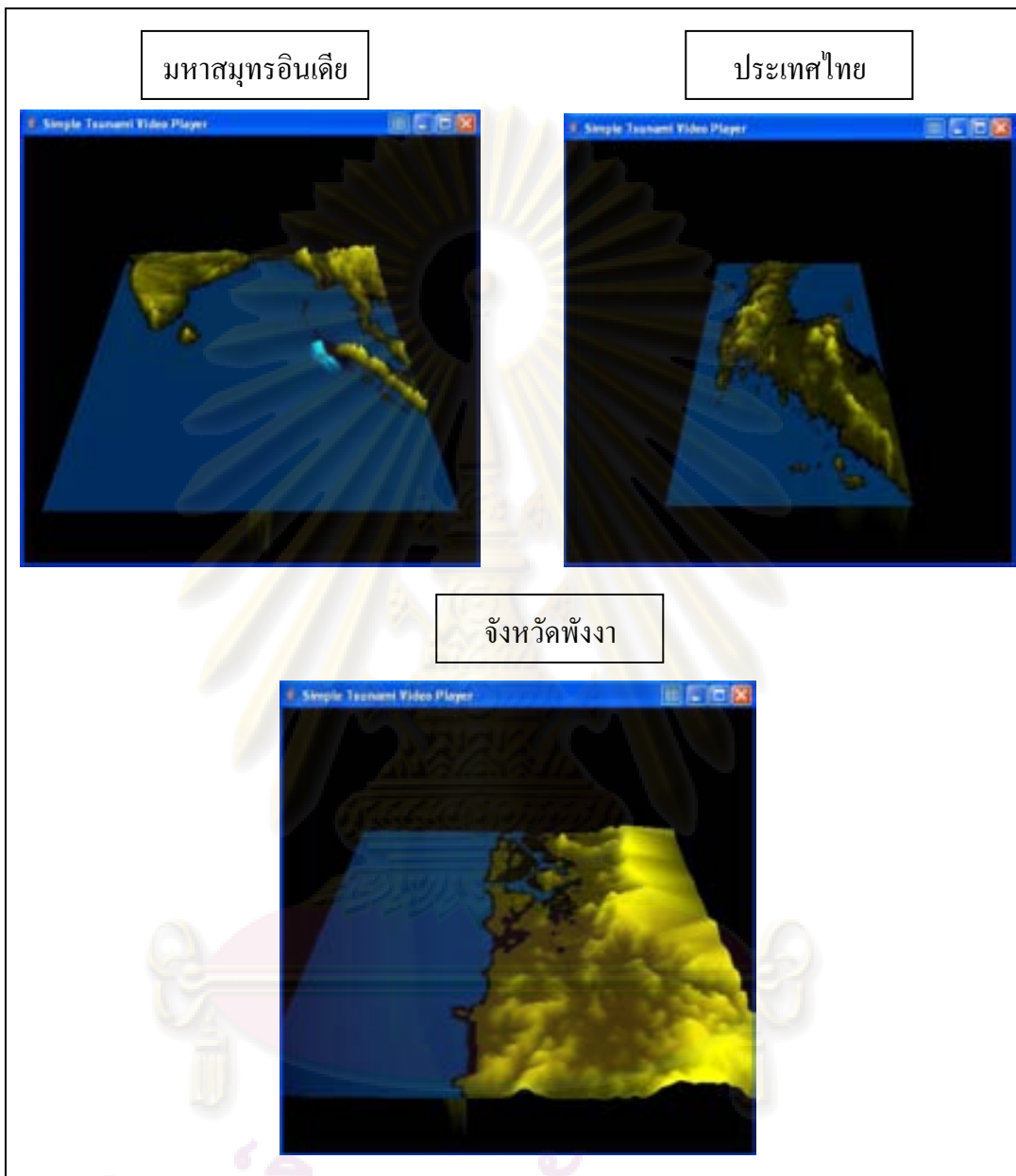




ศูนย์วิทยทรัพยากร

รูปที่ 4.4 แสดงการปรับลดสเกลของความสูงของพื้นดิน 100 เท่า

จุฬาลงกรณ์มหาวิทยาลัย



ศูนย์วิทยทรัพยากร

รูปที่ 4.5 แสดงการปรับลดสเกลของความสูงของพื้นดิน 300 เท่า

จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 5

### สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

เมื่อสังเกตจากผลการทดลองจะเห็นได้ว่าการแปลงข้อมูลจากข้อมูลเลขจำนวนจริงที่จัดเก็บเป็นข้อมูลตัวอักษร มาเป็นข้อมูลเลขลักษณะฐานสองนั้น ทำให้ช่วยประหยัดเวลาในการประมวลผลของการสร้างภาพสามมิติที่มีข้อมูลขนาดใหญ่ โดยที่เราไม่จำเป็นต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพในการคำนวณสูง

การสร้างภาพสามมิติและการให้แสงและเงา เราจะใช้วิธีการทั่วไปโดยใช้ขั้นตอนวิธีซีพเพอร์เป็นหลัก แต่เนื่องจากความสูงของพื้นดินและพื้นน้ำมีความแตกต่างกันมาก จึงทำให้การสร้างภาพจำเป็นต้องปรับข้อมูลความสูงของพื้นดินเพื่อให้เห็นภาพในมุมมองต่างๆ ได้ชัดเจนขึ้น

แต่เนื่องจากข้อมูลที่ได้จากการวิเคราะห์เชิงตัวเลข ยังมีความละเอียดไม่เพียงพอในการจำลองภาพเคลื่อนไหวสามมิติ จึงได้เพิ่มความละเอียดของภาพ โดยวิธีการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น เหตุที่เลือกใช้วิธีการแบบเชิงเส้นนี้ เพราะเป็นวิธีที่ง่ายและสามารถคำนวณหาข้อมูลที่อยู่ในช่วงได้ถูกต้อง ถ้าเราเลือกที่จะใช้วิธีการประมาณค่าในช่วงเสมือนพหุนามวิธีการอื่น อย่างเช่น แบบพหุนามดีกรี 2 หรือแบบพหุนามดีกรี 3 ค่าที่ได้ จะไม่ต่างกันมาก แต่เวลาที่ใช้ในการประมวลผลจะเพิ่มขึ้นตามลำดับ

นอกจากนี้ ได้มีการเพิ่มเฟรมภาพ ให้มากขึ้น เพื่อให้ภาพมีความต่อเนื่องมากขึ้น โดยทำการเพิ่มเฟรมภาพ ด้วยวิธีการประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น เช่นเดียวกับการเพิ่มข้อมูล

งานวิจัยนี้ได้จำลองสีนามิสามมิติ ที่ทำงานกับคอมพิวเตอร์พีซี ให้แสดงภาพเคลื่อนไหวสามมิติได้ แต่เนื่องจากการคำนวณต้องใช้เวลามาก ผู้วิจัยจึงได้หาแนวทางที่จะทำให้เวลาของการทำงานกับเครื่องคอมพิวเตอร์พีซีใช้เวลาอันน้อยลง โดยที่ไม่จำเป็นต้องใช้คอมพิวเตอร์ระดับสูงในการประมวลผล ด้วยการแปลงข้อมูลให้เป็นข้อมูลเลขลักษณะฐานสองก่อน เพื่อลดเวลาในการอ่านข้อมูล เวลาในการประมวลผล และสแกนพื้นที่ในการจัดเก็บ และใช้แบบจำลองทางคณิตศาสตร์ในการให้แสงและเงาที่เหมาะสมกับสภาพทางภูมิศาสตร์ อีกทั้งเพิ่มฟังก์ชันการใช้งาน เพื่อให้เห็นมุมมองต่างๆ ได้ดียิ่งขึ้น รวมถึงการใช้การประมาณค่าในช่วงเสมือนพหุนามแบบเชิงเส้น เพื่อเพิ่มความละเอียดของภาพในบริเวณที่ต้องการแสดง และเพิ่มจำนวนเฟรมภาพเพื่อให้ภาพเกิดความต่อเนื่องมากยิ่งขึ้น โดยแสดงบริเวณจังหวัดพังงาเป็นกรณีศึกษา

## 5.2 อภิปรายผลการวิจัย

การสร้างภาพสามมิติด้วยกระบวนการวิธีนำเสนอ ช่วยให้การประมวลผลเร็วขึ้น และสามารถจำลองภาพสามมิติได้อย่างต่อเนื่อง โดยไม่จำเป็นต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูง อีกทั้งยังสามารถเพิ่มความละเอียดของภาพ เพื่อให้ได้ภาพที่คมชัดและเรียบไม่ขาดช่วง

ผลในการแสดงภาพจะมีความเสมือนจริงได้นั้น จะต้องอาศัยความรู้หลายด้านทั้งด้านศิลปะ ด้านฟิสิกส์ที่เกี่ยวข้องกับธรรมชาติของแสงและสี ซึ่งงานวิจัยนี้ในส่วนของ การกำหนดสีพื้นดินจะใช้โทนสีเขียวเป็นหลัก โดยที่ถ้าสีอ่อนลงคือความสูงที่เพิ่มขึ้น และส่วนของพื้นน้ำจะใช้สีน้ำเงิน โดยที่ท้องน้ำจะให้สีดำ ยิ่งมีความลึกมากจะมีสีเข้ม เพื่อให้เห็นว่าน้ำนั้นมีความลึกและความสูงที่ชัดเจนยิ่งขึ้น

ในการให้แสงและเงา เนื่องจากภาพที่แสดงผลนั้นเป็นภาพมุมมองกว้างของมหาสมุทรอินเดีย จึงใช้แสงแบบล้อมรอบในการให้แสง เพราะ เป็นการเพิ่มปริมาณแสงให้กับทุกๆ จุดบนพื้นผิวของวัตถุด้วยปริมาณแสงที่เท่ากัน

การปรับความสูงในส่วนของพื้นดินนั้นมีความจำเป็น เนื่องจากข้อมูลความสูงส่วนพื้นดินและพื้นน้ำนั้นต่างกันมาก ยกตัวอย่างเช่น ข้อมูลบริเวณชายฝั่งภาคใต้ของประเทศไทย มีความสูงของคลื่นที่สูงที่สุดเท่ากับ 6.99 เมตร ในขณะที่ ความสูงของพื้นดินที่สูงที่สุดเท่ากับ 2632.28 เมตร และที่เลือกเฉพาะ ปรับสเกลส่วนของพื้นดิน เนื่องจาก ถ้าเราปรับความสูงของคลื่นด้วยอาจจะทำให้เห็นการเกิดคลื่นไม่ชัดเจน แต่จะให้ปรับลดสเกลเท่าไรจึงจะเหมาะสมขึ้นอยู่กับความเสมือนจริง ซึ่งงานวิจัยนี้บริเวณมหาสมุทรอินเดียและ บริเวณชายฝั่งภาคใต้ของประเทศไทย ได้ปรับลดสเกลของภาพในส่วนพื้นดิน 300 เท่า และบริเวณจังหวัดพังงา ได้ปรับลด สเกลของภาพในส่วนพื้นดิน 100 เท่า

## 5.3 ข้อเสนอแนะ

ควรปรับปรุงการให้แสงและเงาที่เสมือนจริงมากขึ้น โดยกำหนดแหล่งกำเนิดแสงให้ตรงกับเวลาที่เกิดสึนามิจริง ซึ่งสามารถทำได้โดยการใช้ สมการคณิตศาสตร์ประกอบกับสมการฟิสิกส์ที่เกี่ยวข้องกับธรรมชาติของแสง ในการกำหนดแสง แต่การคำนวณจะค่อนข้างซับซ้อน เพราะแหล่งกำเนิดแสงนั้นเคลื่อนที่อยู่ตลอดเวลา

## รายการอ้างอิง

- [1] ณรงค์ฤทธิ์ แก้วบรรจกร. การจำลองเชิงตัวเลขของคลื่นสึนามิในมหาสมุทรอินเดียเมื่อวันที่ 26 ธันวาคม 2547. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต สาขาวิชาวิทยาการคณนา ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2548.
- [2] Okada, Y. Surface deformation due to shear and tensile faults in a half-space. Bull. Seis. Soc. Am. (1985) : 1135-1154.
- [3] Gower, J. Jason 1 detects the December 26, 2004 tsunami. Eos Trans. AGU. (2005)
- [4] Kass, M., and Miller, G. Rapid, Stable Fluid Dynamics for Computer Graphics. Siggraph (1990) : 49-57.
- [5] Hagen, T.R., Hjelmervik, J.M., Lie, K.A., Natvig, J.R., and Henriksen, M.O. Visual simulation of shallow-water waves. Simulation Modelling Practice and Theory. (2005) : 716-726.
- [6] Nishidate, Y. and Nikishkov, G.P. Fast Water Animation Using the Wave Equation with Damping[online]. 2005. Available from : [http://www.uaizu.ac.jp/~niki/papers/2005\\_Fast\\_water\\_animation.pdf](http://www.uaizu.ac.jp/~niki/papers/2005_Fast_water_animation.pdf) [cited 24 Jan 2008].
- [7] Noe, Karsten. Implementing Rapid, Stable Fluid Dynamics on the GPU[online]. 2004. Available from : [http://projects.n-o-e.dk/GPU\\_water\\_simulation/gpu-water.pdf](http://projects.n-o-e.dk/GPU_water_simulation/gpu-water.pdf) [cited 21 May 2007].
- [8] Engel, K., Kraus, M., and Ertl, T. High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading. ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware. (2001) : 9-16.
- [9] Ebert, D., and Rheingans, P. Volume Illustration : Non-Photorealistic Rendering of Volume Models. Proceedings of IEEE Visualization'00. (October 2000) : 195-202.
- [10] Salama, C.R., Scheuering, M., Soza, G., and Greiner, G. Fast Volumetric Deformation On General Purpose Hardware. ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware. (2001) : 17-23.
- [11] พรพล สาครินทร์ และ กฤษณา แก้วมณี . 3D Graphics Rendering. กรุงเทพมหานคร : ชัคเชส มีเดีย , 2544.

- [12] ปุณณรัตน์ พิชญ์ไพพบูลย์ . Computer Graphics : สำหรับนักออกแบบ. กรุงเทพมหานคร : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย , 2542.
- [13] Hearn, D., and Baker, M.P. Computer graphics with OpenGL. Third edition. United States of America : Pearson Education , 2004.
- [14] Foley, J.D., Andries, V.D., Steven, F.K., and John, H.F. Computer Graphics Principles and Practice. Second edition. Reading : Addison-Wesley , 1990.
- [15] วีระศักดิ์ ชิงถาวร. Java programming. เล่มที่1. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น , 2549.
- [16] รุ่งโรจน์ โพนคำ . กะเทาะเปลือกจาวา : Object-Oriented Programming with Java Language. กรุงเทพมหานคร : จิรวัดน์ เอ็กซ์เพรส .
- [17] ดวงแก้ว สวามิภักดิ์. การโปรแกรมภาษา C. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น , 2539.
- [18] Leader, J.J. Numerical Analysis and Scientific Computation. United States of America : Pearson Education , 2004.
- [19] Pokorny, C.K. Computer Graphics: An Object-Oriented Approach To The Art And Science. Wilsonville : Franklin, Beedle & Associate Incorporated., 1994.
- [20] Titley, D. Linear Spline Interpolation[online]. Available from : [http://www.cs.mcgill.ca/~dtitle/cs350/notes/20\\_linsplin.pdf](http://www.cs.mcgill.ca/~dtitle/cs350/notes/20_linsplin.pdf). [cited 28 October 2008].
- [21] วีระศักดิ์ ชิงถาวร. Java programming. เล่มที่2. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น , 2549.
- [22] วรเศรษฐ สุวรรณิก และ ทศพล ธนะทิพานนท์. เขียนโปรแกรม Java เบื้องต้น. กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2549.
- [23] Thomas, W.C. An introduction to object-oriented programming with Java. Third edition. United States of America : The McGraw-Hill Companies , 2004.
- [24] Hopkins, G. The joy of java 3d[online]. 2001. Available from : <http://www.java3d.org/tutorial.html>. [cited 18 May 2007].
- [25] Sun microsystem. Java 3D Api specification [online]. Sun microsystem, 1999. Available from : <http://java.sun.com/products/java-media/3D/forDevelopers/j3dguide/j3dTOC.doc.html>. [cited 18 May 2007].
- [26] Chen, Q., Kirby, J.T., Dalrymple, R.A., Kennedy, A.B., and Chawla, A. Boussinesq modeling of wave transformation, breaking, and run-up. II: 2D. J. Wtrwy, Port, Coast, and Oc. Engrg, ASCE (2000) : 48-56.

[27] Watts, P., Grilli, S.T., Kirby, J.T., Fryer, G.J. and Tappin, D. Landslide tsunami case studies using a Boussinesq model and a fully nonlinear tsunami generation model. Nat. Hazards and Earth Sci. Systems (2003) : 391-402.

[28] Day, S.J., Watts, P., Grilli, S.T., and Kirby, J.T. Mechanical models of the 1975 Kalapana, Hawaii earthquake and tsunami. Marine Geology (2005) : 59-92.



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ก.

สมการ Boussinesq , Bottom Friction, Wave Breaking,

Moving Shorelines และ Subgrid Turbulence

ในการศึกษานี้ เราพิจารณาอาณาเขตของคลื่น 3 มิติ กับ ระดับผิวน้ำ  $\eta(x, y, t)$  ที่เวลา  $t$  แปรตาม ตัวแปรความลึกของน้ำ  $h(x, y)$  ระบบพิกัด  $(x, y, z)$  ถูก adopted ด้วย  $z$  ซึ่งวัดได้เมื่อมันสูงขึ้นจากระดับน้ำคงที่ ของไหล ถูกสมมติว่าเป็น inviscid และไม่สามารถนับได้ และการไหลถูกสมมติว่าเป็น irrotational สมการไม่เชิงเส้นของ Boussinesq กำหนดโดย:

$$\begin{aligned} \eta_t + \nabla \cdot \left\{ (h + \eta) \left[ \mathbf{u}_\alpha + \left( z_\alpha + \frac{1}{2}(h - \eta) \right) \nabla (\nabla \cdot (h\mathbf{u}_\alpha)) \right. \right. \\ \left. \left. + \left( \frac{1}{2} z_\alpha^2 - \frac{1}{6} (h^2 - h\eta + \eta^2) \right) \nabla (\nabla \cdot \mathbf{u}_\alpha) \right] \right\} = 0 \end{aligned} \quad (ก.1)$$

$$\begin{aligned} \mathbf{u}_{\alpha t} + (\mathbf{u}_\alpha \cdot \nabla) \mathbf{u}_\alpha + g \nabla \eta + z_\alpha \left\{ \frac{1}{2} z_\alpha \nabla (\nabla \cdot \mathbf{u}_{\alpha t}) + \nabla (\nabla \cdot (h\mathbf{u}_{\alpha t})) \right\} \\ + \nabla \left\{ \frac{1}{2} (z_\alpha^2 - \eta^2) (\mathbf{u}_\alpha \cdot \nabla) (\nabla \cdot \mathbf{u}_\alpha) + \frac{1}{2} [\nabla \cdot (h\mathbf{u}_\alpha) + \eta \nabla \cdot \mathbf{u}_\alpha]^2 \right\} \\ + \nabla \left\{ (z_\alpha - \eta) (\mathbf{u}_\alpha \cdot \nabla) (\nabla \cdot (h\mathbf{u}_\alpha)) - \eta \left[ \frac{1}{2} \eta \nabla \cdot \mathbf{u}_{\alpha t} + \nabla \cdot (h\mathbf{u}_{\alpha t}) \right] \right\} = 0 \end{aligned} \quad (ก.2)$$

เมื่อ  $\eta$  คือ การยกพื้นผิวให้สูง  $h$  คือ ความลึกของน้ำที่คงที่  $\mathbf{u}_\alpha$  คือเวกเตอร์ความเร็วในแนวนอนที่ความลึกน้ำ  $z = z_\alpha = -0.531h$ ,  $\nabla = (\partial/\partial x, \partial/\partial y)$  คือ ตัวควบคุมความลาดชันในแนวนอน  $g$  คือความเร่ง และตัวห้อย  $t$  แทนอนุพันธ์ย่อยตามเวลา

เพื่อที่จะพัฒนาตัวแบบสำหรับการประยุกต์ที่ใช้ได้จริง ผลทางกายภาพของ

damping ที่เกี่ยวกับการเสียดสี และ wave breaking, boundary absorbtion, และ moving shoreline ถูก

รวมเข้ากับ the model scheme สมการ (ก.1) และ (ก.2) ถูกดัดแปลงขยายเป็น

$$\eta_t = E(\eta, u, v) + \gamma E_2(\eta, u, v) \quad (\text{ก.3})$$

$$\begin{aligned} [U(u)]_t &= F(\eta, u, v) + [F_1(v)]_t + \gamma [F_2(\eta, u, v) + F'(\eta, u_t, v_t)] \\ &\quad + F_b + F_{br} + F_{bs} + F_{sp} \end{aligned} \quad (\text{ก.4})$$

$$\begin{aligned} [V(v)]_t &= G(\eta, u, v) + [G_1(u)]_t + \gamma [G_2(\eta, u, v) + G'(\eta, u_t, v_t)] \\ &\quad + G_b + G_{br} + G_{bs} + G_{sp} \end{aligned} \quad (\text{ก.5})$$

เมื่อ  $u$  และ  $v$  คือ ความเร็วในแนวนอนตามทิศทางของ  $x$  และ  $y$  ที่ความลึก  $z = z_\alpha = -0.531h$ , นั่นคือ  $(u, v) = u_\alpha$  และ  $\gamma$  คือ พารามิเตอร์ที่ควบคุมให้เราเลือกระหว่างกรณี fully nonlinear Boussinesq ( $\gamma=1$ ) or weakly nonlinear Boussinesq ( $\gamma=0$ ) ปริมาณ  $U, V, E, F, G, F_1, G_1, F', G', E_2, F_2$  และ  $G_2$  เป็นฟังก์ชันของ  $\eta, u, v, u_t$  หรือ  $v_t$  ซึ่งนิยามดังนี้

$$U = u + h [b_1 h u_{xx} + b_2 (hu)_{xx}] \quad (\text{ก.6})$$

$$V = v + h [b_1 h v_{yy} + b_2 (hv)_{yy}] \quad (\text{ก.7})$$

$$\begin{aligned} E &= -\frac{1}{\kappa} [(\Lambda u)_x + (\Lambda v)_y] \\ &\quad - \left\{ a_1 h^3 (u_{xx} + v_{xy}) + a_2 h^2 [(hu)_{xx} + (hv)_{xy}] \right\}_x \\ &\quad - \left\{ a_1 h^3 (u_{xy} + v_{yy}) + a_2 h^2 [(hu)_{xy} + (hv)_{yy}] \right\}_y \end{aligned} \quad (\text{ก.8})$$

$$F = -g\eta_x - (uu_x + vv_y) \quad (\text{ก.9})$$

$$G = -g\eta_y - (uv_x + vv_y) \quad (\text{ก.10})$$

$$F_1 = -h [b_1 h v_{xy} + b_2 (hv)_{xy}] \quad (\text{ก.11})$$

$$G_1 = -h \left[ b_1 h u_{xy} + b_2 (hu)_{xy} \right] \quad (\text{ก.12})$$

$$F^t = \left\{ \frac{1}{2} \eta^2 \left[ (u_t)_x + (v_t)_y \right] + \eta \left[ [h(u_t)]_x + [h(v_t)]_y \right] \right\}_x \quad (\text{ก.13})$$

$$G^t = \left\{ \frac{1}{2} \eta^2 \left[ (u_t)_x + (v_t)_y \right] + \eta \left[ [h(u_t)]_x + [h(v_t)]_y \right] \right\}_y \quad (\text{ก.14})$$

$$\begin{aligned} E_2 = & - \left\{ \left[ a_1 h^2 \eta + \frac{1}{6} \eta (h^2 - \eta^2) \right] (u_{xx} + v_{yy}) \right\}_x \\ & - \left\{ \left[ a_2 h \eta - \frac{1}{2} \eta (h + \eta) \right] \left[ (hu)_{xx} + (hv)_{yy} \right] \right\}_x \\ & - \left\{ \left[ a_1 h^2 \eta + \frac{1}{6} \eta (h^2 - \eta^2) \right] (u_{xy} + v_{yy}) \right\}_y \\ & - \left\{ \left[ a_2 h \eta - \frac{1}{2} \eta (h + \eta) \right] \left[ (hu)_{xy} + (hv)_{yy} \right] \right\}_y \end{aligned} \quad (\text{ก.15})$$

$$\begin{aligned} F_2 = & - \left\{ \frac{1}{2} (z_\alpha^2 - \eta^2) \left[ u (u_x + v_y)_x + v (u_x + v_y)_y \right] \right\}_x \\ & - \left\{ (z_\alpha - \eta) \left[ u \left[ (hu)_x + (hv)_y \right]_x + v \left[ (hu)_x + (hv)_y \right]_y \right] \right\}_x \\ & - \frac{1}{2} \left\{ \left[ (hu)_x + (hv)_y + \eta (u_x + v_y) \right]^2 \right\}_x \end{aligned} \quad (\text{ก.16})$$

$$\begin{aligned} G_2 = & - \left\{ \frac{1}{2} (z_\alpha^2 - \eta^2) \left[ u (u_x + v_y)_x + v (u_x + v_y)_y \right] \right\}_y \\ & - \left\{ (z_\alpha - \eta) \left[ u \left[ (hu)_x + (hv)_y \right]_x + v \left[ (hu)_x + (hv)_y \right]_y \right] \right\}_y \\ & - \frac{1}{2} \left\{ \left[ (hu)_x + (hv)_y + \eta (u_x + v_y) \right]^2 \right\}_y \end{aligned} \quad (\text{ก.17})$$

เมื่อ  $a_1, a_2, b_1, b_2$  เป็นค่าคงที่ซึ่งมีความสัมพันธ์กับ the dimensionless referenced water depth  $\beta = z_\alpha/h = -0.531$  โดย

$$a_1 = \frac{1}{2}\beta^2 - \frac{1}{6}, \quad a_2 = \beta + \frac{1}{2}, \quad b_1 = \frac{1}{2}\beta^2, \quad b_2 = \beta \quad (\text{ก.18})$$

ตัวประกอบ  $\Lambda$  และ  $\kappa$  ใน (ก.8) ถูกจัดเตรียมไว้สำหรับการหา moving shoreline

ในการศึกษานี้ Boussinesq model ถูกดัดแปลงด้วยการขยายไปปกคลุม bottom friction, wave breaking, moving shorelines and subgrid turbulence

The bottom friction ถูกกำหนดโดย

$$(F_b, G_b) = \frac{K}{h+\eta} \mathbf{u}_\alpha |\mathbf{u}_\alpha| \quad (\text{ก.19})$$

เมื่อ  $K = 1 \times 10^{-5}$

Wave breaking ในน้ำตื้น ถูกนิยามโดย

$$F_{br} = \frac{1}{h+\eta} \left[ v \left( (h+\eta) u_{\alpha x} \right)_x + \frac{1}{2} \left( v \left( (h+\eta) u_{\alpha y} + (h+\eta) v_{\alpha x} \right) \right)_y \right] \quad (\text{ก.20})$$

$$G_{br} = \frac{1}{h+\eta} \left[ \frac{1}{2} \left( v \left( (h+\eta) v_{\alpha x} + (h+\eta) u_{\alpha y} \right) \right)_x + \left( v \left( (h+\eta) v_{\alpha y} \right) \right)_y \right] \quad (\text{ก.21})$$

เมื่อตัวห้อย  $x$  และ  $y$  แทนอนุพันธ์  $x$  และ  $y$  ตามตำแหน่ง และ  $v$  คือ กระแสน้ำวน ซึ่งความหนืดจำกัด บนหน้าแรกของ the breaking wave ซึ่งถูกนิยาม ดังนี้

$$v = B\delta^2 |(h+\eta) \nabla \cdot \mathbf{M}| \quad (\text{ก.22})$$

เมื่อ  $\delta$  คือสัมประสิทธิ์ความยาว ซึ่งจากการทดลองได้  $\delta=1.2$  ปริมาณ  $B$  ซึ่งควบคุมการสลายตัวของพลังงาน ถูกกำหนดโดย

$$B = \begin{cases} 1, & \eta_t \geq 2\eta_t^* \\ \frac{\eta_t}{\eta_t^*} - 1, & \eta_t^* < \eta_t \leq 2\eta_t^* \\ 0, & \eta_t \leq \eta_t^* \end{cases} \quad (\text{ก.23})$$

การเริ่ม และการหยุดชะงักของ wave breaking ใช้พารามิเตอร์  $\eta_t^*$ , ซึ่งถูกนิยามดังนี้

$$\eta_t^* = \begin{cases} \eta_t^{(F)}, & t \geq T^* \\ \eta_t^{(I)} + \frac{t-t_0}{T^*}(\eta_t^{(F)} - \eta_t^{(I)}), & 0 \leq t-t_0 < T^* \end{cases} \quad (\text{ก.24})$$

เมื่อ  $T^*$  คือเวลาที่เปลี่ยนไป,  $t_0$  คือ เวลาซึ่ง the wave breaking เกิดขึ้น และ  $t-t_0$  คือเวลาของเหตุการณ์ breaking ค่าของ  $\eta_t^{(I)}$  แปรปรวนอยู่ระหว่าง  $0.35\sqrt{gh}$  และ  $0.65\sqrt{gh}$ , ขณะที่ค่าของ  $\eta_t^{(F)}$  และ  $T^*$  คือ  $0.15\sqrt{gh}$  และ  $5\sqrt{h/g}$  ตามลำดับ การคิดค้นและการพิสูจน์ the breaking model ได้อธิบายไว้ใน Kennedy[24]

ตัวประกอบ  $\Lambda$  และ  $\kappa$  ใน (ก.8) ถูก แนะนำให้รู้จักกัน โดย Kennedy[24]และ Chen[24] เพื่อทำให้เกิด a porous beach method ใช้เก็บ the subaerial portion of the model grid computationally active และเพื่อให้การคำนวณ runup ง่ายขึ้น บนชายฝั่งต่างๆ กำหนดโดย

$$\kappa = \begin{cases} \delta + (1-\delta)e^{\lambda\left(\frac{\eta-z}{h_0}\right)}, & \eta \leq z^* \\ 1, & \eta > z^* \end{cases} \quad (\text{ก.25})$$

และ

$$\Lambda = \begin{cases} \delta(\eta + h_0) + \frac{(1-\delta)h_0}{\lambda} (e^{\lambda \frac{(\eta-z^*)}{h_0}} - e^{-\lambda \frac{(h_0+z^*)}{h_0}}), & \eta \leq z^* \\ (\eta - z^*) + \delta(z^* + h_0) + \frac{(1-\delta)h_0}{\lambda} (1 - e^{-\lambda \frac{(h_0+z^*)}{h_0}}), & \eta > z^* \end{cases} \quad (\text{ก.26})$$

เมื่อ  $h_0$  แทนความลึกของหาดทรายที่มีรูพรุนน้ำซึมผ่านได้ ซึ่งจะต้องลึกกว่า ความลึกของคลื่นที่มากที่สุดขณะคำนวณ ทางเลือกของ  $z^*$  ได้อธิบายไว้ใน Kennedy [24] และกำหนดโดย

$$z^* = \frac{z^s}{1-\delta} + h_0 \left( \frac{\delta}{1-\delta} + \frac{1}{\lambda} \right) \quad (\text{ก.27})$$

เมื่อ  $z^s$  คือ การยกสูงของพื้นดินใต้ทะเล เมื่อ  $\delta = 0.08$  และ  $\lambda = 25$  พื้นฐานการศึกษาจำนวนเหตุการณ์ tsunami runup เป็นไปตาม Watts [25] และ Day [27]

The Smagorinsky type subgrid model เพื่ออธิบายผลที่เกิดจากกระแสวนความหนืด บนการไหลได้พื้นโลก

$$F_{bs} = \frac{1}{h+\eta} \left[ \left( v_s \left( (h+\eta) u_\alpha \right)_x \right)_x + \frac{1}{2} \left( v_s \left( \left( (h+\eta) u_\alpha \right)_y + \left( (h+\eta) v_\alpha \right)_x \right) \right)_y \right] \quad (\text{ก.28})$$

$$G_{br} = \frac{1}{h+\eta} \left[ \frac{1}{2} \left( v_s \left( \left( (h+\eta) v_\alpha \right)_x + \left( (h+\eta) u_\alpha \right)_y \right) \right)_x + \left( v_s \left( (h+\eta) v_\alpha \right)_y \right)_y \right] \quad (\text{ก.29})$$

เมื่อ  $v_s$  คือกระแสวนความหนืด เนื่องจาก the subgrid turbulence

$$v_s = c_m \Delta x \Delta y \left[ (U_x)^2 + (V_y)^2 + \frac{1}{2} (U_y + V_x)^2 \right]^{\frac{1}{2}} \quad (\text{ก.30})$$

เมื่อ  $U$  และ  $V$  คือความหนืด ประกอบกับเวลาเฉลี่ยของบริเวณที่เกิดขึ้นอยู่ข้าง  
 ใต้  $\Delta x$  และ  $\Delta y$  คือ ระยะของ  $x$  และ  $y$  ตามลำดับ และ  $c_m$  คือ ค่าสัมประสิทธิ์ the default value  
 of 0.2 แนวทางของการจำลอง บริเวณที่เกิดขึ้นอยู่ข้างใต้ ได้รับจากการเฉลี่ย ความเร็วชั่วขณะ  
 มากกว่า จุดของคลื่นที่สูงที่สุด 2 จุด และ  $v_\alpha$  คือ updated accordingly



ศูนย์วิทยทรัพยากร  
 จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข.

### การกำหนดสีของภาพ

การกำหนดสีส่วนของพื้นดินเป็น โทนสีน้ำตาล ตามระดับความสูงของพื้นดิน ถ้าระดับความสูงมากขึ้น สีจะอ่อนลง ดังตัวอย่างสมการต่อไปนี้

$$\text{RED} = \frac{Z}{\text{Max}} ; \quad \text{GREEN} = \frac{Z}{\text{Max}} ; \quad \text{BLUE} = 0 ;$$

ซึ่ง RED คือ สีแดง  
 GREEN คือ สีเขียว  
 BLUE คือ สีน้ำเงิน  
 Z คือ ค่าความสูงของพื้นดิน แต่ละจุดภาพ  
 Max คือ ค่าสูงสุดความสูงของพื้นดิน

โดยเราจะเอาค่าความสูงแต่ละจุดภาพมาหารด้วยค่าสูงสุดความสูงของพื้นดิน เพื่อให้ค่าสีนั้นอยู่ในช่วง 0 ถึง 1.0 นั้นเอง

```
// Updating Land Color
```

```
// color float array format : [r1][g1][b1][r2][g2][b2]...[rN][gN][bN]
```

```
for (int i=0; i < landColor.length; i+=3){           // for each set of color
    landColor[i] = rawCoord[i+1]/(max);           // RED
    landColor[i+1] = rawCoord[i+1]/(max);         // GREEN
    landColor[i+2] = 0;                           // BLUE
}
```

ศูนย์วิทยทรัพยากร  
 จุฬาลงกรณ์มหาวิทยาลัย



และการกำหนดสีส่วนของพื้นน้ำ เป็นโทนสีน้ำเงิน ตามระดับความลึกของน้ำ ถ้า น้ำมีความลึกมาก ท้องน้ำจะเป็นสีดำ ดังตัวอย่างสมการต่อไปนี้

$$\text{RED} = 0; \quad \text{GREEN} = -0.2*Z + 0.2; \quad \text{BLUE} = -0.2*Z + 0.4;$$

ซึ่ง RED คือ สีแดง  
GREEN คือ สีเขียว  
BLUE คือ สีน้ำเงิน  
Z คือ ค่าความสูงของพื้นน้ำ แต่ละจุดภาพ

// Updating Water Color

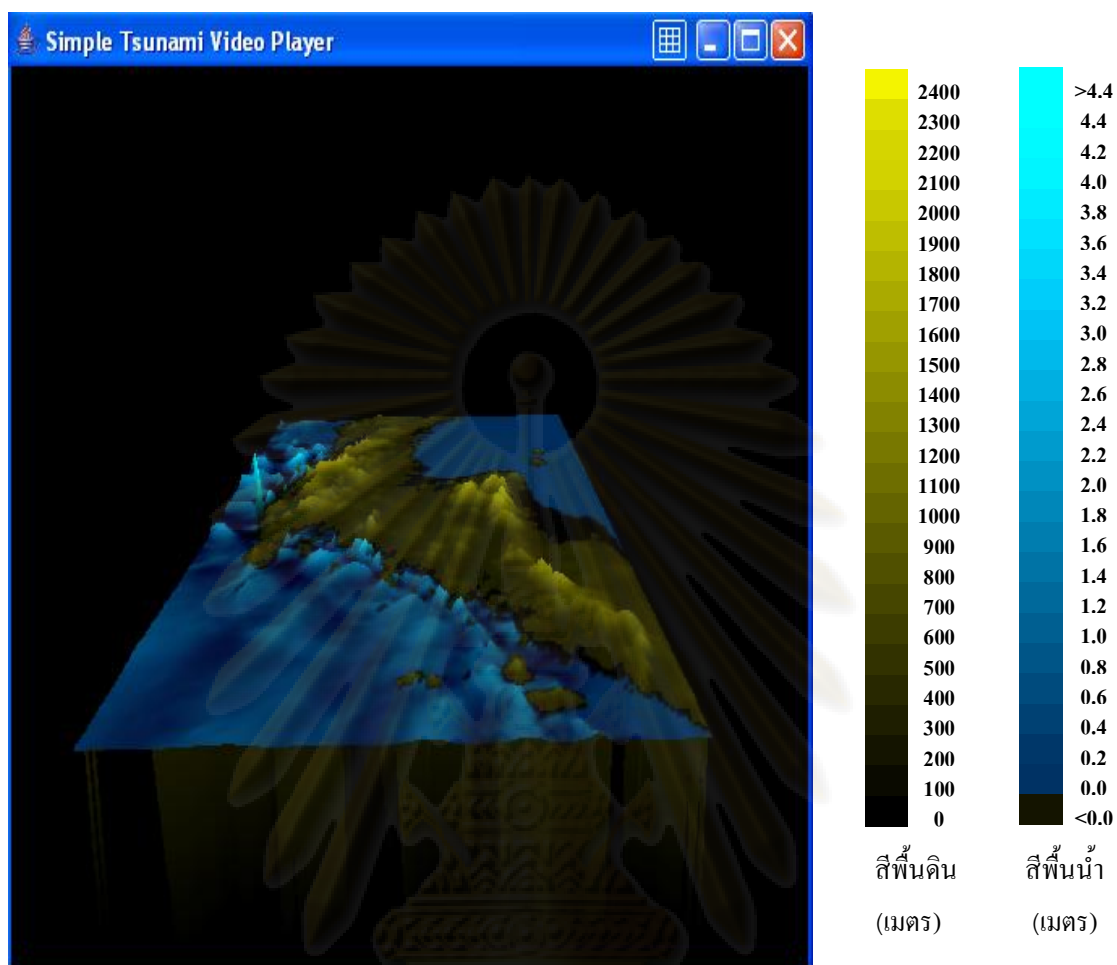
// color float array format : [r1][g1][b1][r2][g2][b2]...[rN][gN][bN]

```
for (int i=0; i < waterColor.length; i+=3){ // for each set of color
    waterColor[i] = 0; // RED
    waterColor[i+1] = waterContrast*(rawCoord[i+1])+0.2f; // GREEN
    waterColor[i+2] = waterContrast*(rawCoord[i+1]) +0.4f; // BLUE
}
```

การกำหนดสีให้กับระดับความสูงของพื้นดินและพื้นน้ำ แสดงดัง รูปที่ ข.1 ซึ่ง แสดงเฟรมภาพสูงสุดของคลื่นสึนามิบริเวณชายฝั่งภาคใต้ของประเทศไทย ซึ่งค่าสูงสุดของน้ำอยู่ที่ 6.99 เมตร ในขณะที่ความสูงของพื้นดินที่สูงที่สุดเท่ากับ 2632 .28 เมตร ซึ่งในส่วนพื้นดินนี้ได้ทำการปรับลดสเกลภาพลง 300 เท่า จึงทำให้เห็นว่าตัวคลื่นมีความสูงมากกว่าพื้นดินดังรูปที่ ข.1

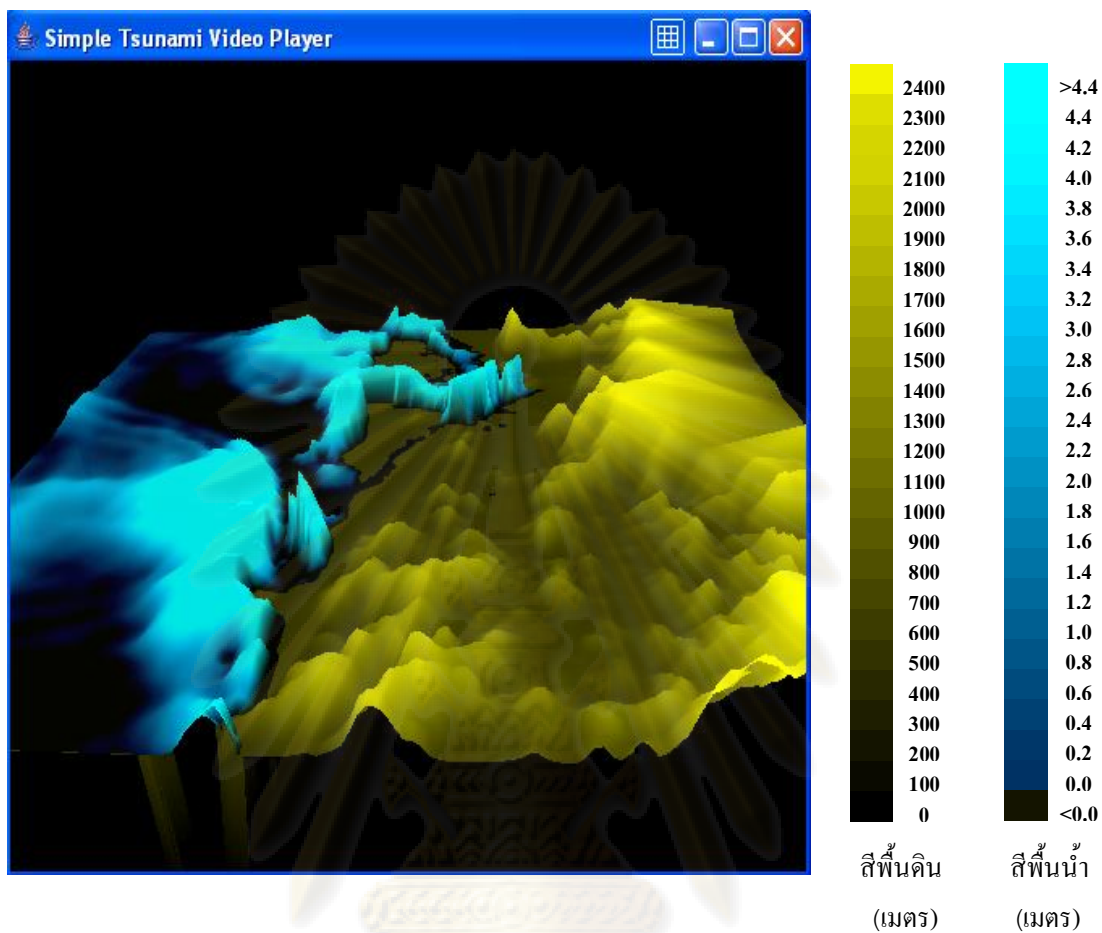
ทำนองเดียวกันรูปที่ ข.2 แสดงเฟรมภาพสูงสุดของคลื่นสึนามิบริเวณ จังหวัดพังงา ซึ่งค่าสูงสุดของน้ำอยู่ที่ 3.88 เมตร ในขณะที่ความสูงของพื้นดินที่สูงที่สุดเท่ากับ 1038.81เมตร ซึ่ง ในส่วนพื้นดินนี้ได้ทำการปรับลดสเกลภาพลง 100 เท่า จึงทำให้เห็นว่าตัวคลื่นมีความสูงมากกว่า พอๆกับพื้นดินดังรูปที่ ข.2

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ข.1 แสดงเฟรมภาพสูงสุดของคลื่นสึนามิบริเวณชายฝั่งภาคใต้ของประเทศไทย  
(พื้นดินปรับลดสเกล300เท่า)

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ข.2 แสดงเฟรมภาพสูงสุดของคลื่นสึนามิบริเวณจังหวัดพังงา

(พื้นดินปรับลดสเกล100เท่า)

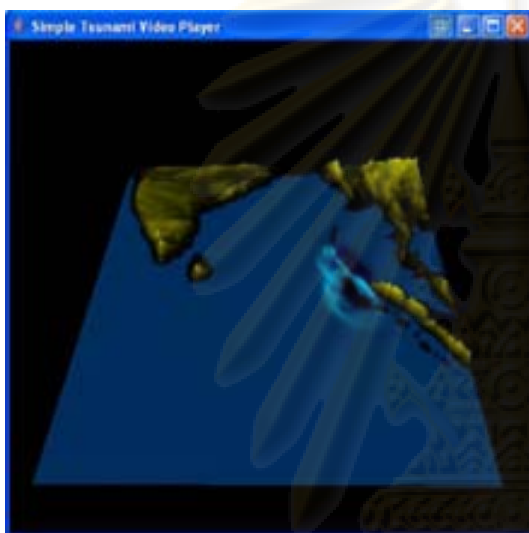
ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค.

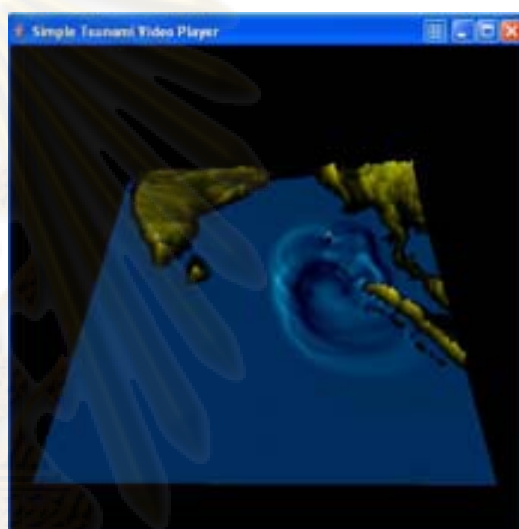
### การแสดงผลเฟรมภาพ

ผลการจำลองสึนามิสามมิติโดยใช้เครื่องคอมพิวเตอร์ Note Book, Intel(R) Celeron(R)M CPU 420@1.6GHz , 512MB RAM ในการรันโปรแกรมทั้งหมด

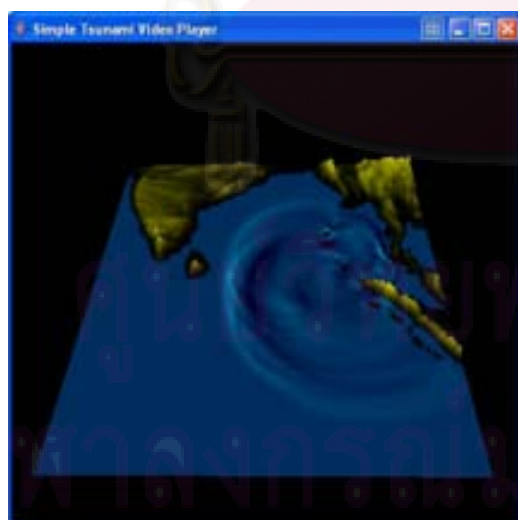
แสดงผลการจำลองสึนามิสามมิติบริเวณมหาสมุทรอินเดียเป็นเฟรมภาพ ณ เวลาต่าง ๆ แสดงดังรูปที่ ค.1-ค.8 โดยทำการปรับลดสเกลภาพส่วนพื้นดินลง 300 เท่า



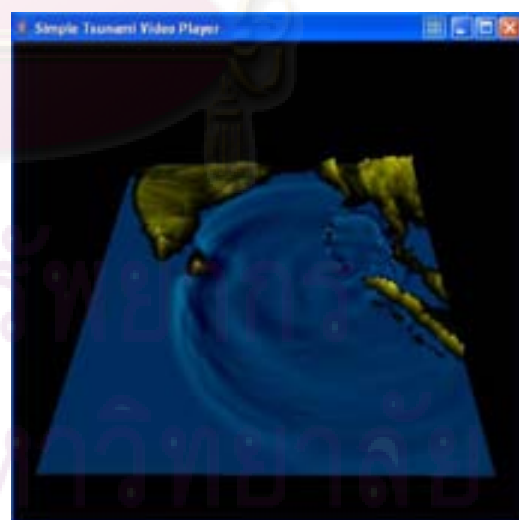
รูปที่ ค.1 แสดงเฟรมภาพที่ 10



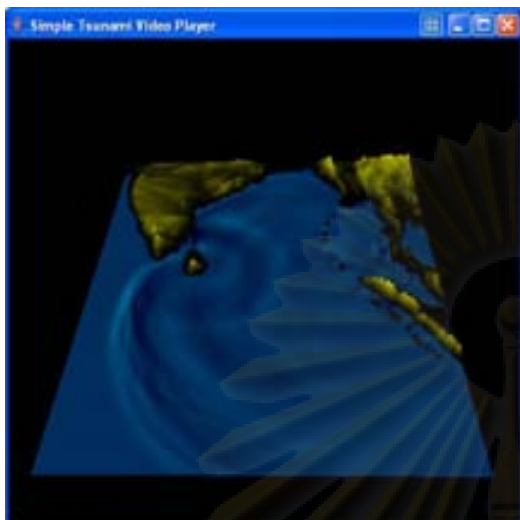
รูปที่ ค.2 แสดงเฟรมภาพที่ 30



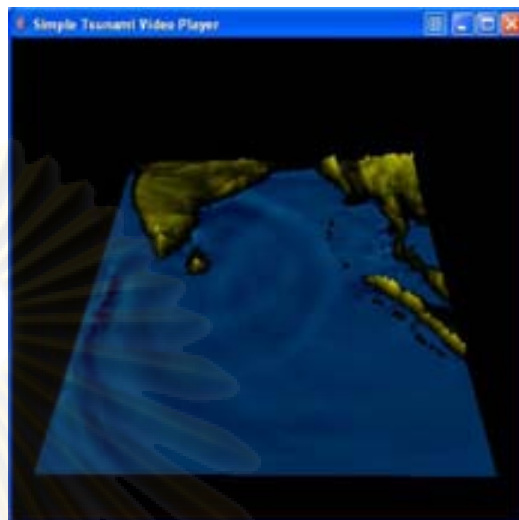
รูปที่ ค.3 แสดงเฟรมภาพที่ 50



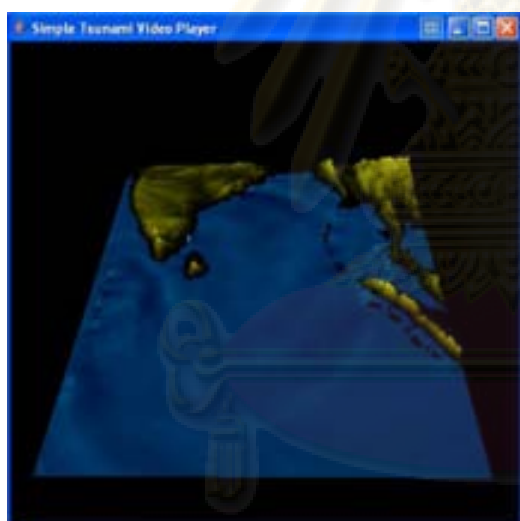
รูปที่ ค.4 แสดงเฟรมภาพที่ 70



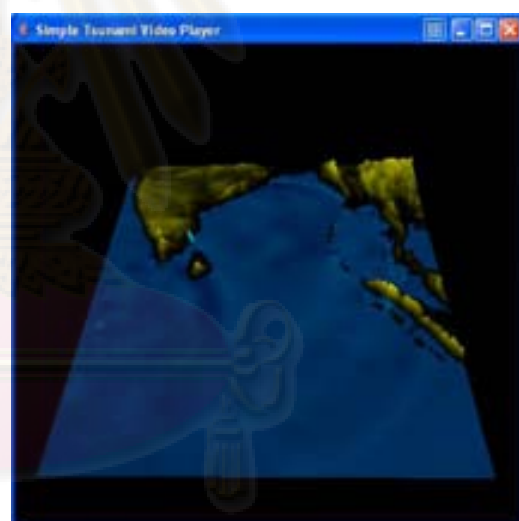
รูปที่ ค.5 แสดงเฟรมภาพที่ 90



รูปที่ ค.6 แสดงเฟรมภาพที่ 110



รูปที่ ค.7 แสดงเฟรมภาพที่ 130



รูปที่ ค.8 แสดงเฟรมภาพที่ 150

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

## ประวัติผู้เขียนวิทยานิพนธ์

นายรัฐสรณ์ย์ จิรชนาวัดน์ เกิดเมื่อวันที่ 18 มกราคม พุทธศักราช 2525 สำเร็จ การศึกษาระดับปริญญาวิทยาศาสตรบัณฑิต สาขาคณิตศาสตร์ จากภาควิชาคณิตศาสตร์ คณะ วิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ เมื่อปีการศึกษา 2548 และเข้าศึกษาต่อในหลักสูตรปริญญา มหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2548



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย