การศึกษาการประมวลผลแบบขนานบนระบบคลัสเตอร์ในการสร้างภาพเชิงปริมาตร

A STUDY OF PARALLEL VOLUME RENDERING ON CLUSTER SYSTEM

นนท์ บัณฑิตวงษ์
NONT BANDITWONG

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2545

ISBN 974-324-124-8

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การศึกษาการประมวลผลแบบขนานบนระบบคลัสเตอร์ ในการสร้างภาพเชิงปริมาตร

A STUDY OF PARALLEL VOLUME RENDERING ON CLUSTER SYSTEM

นนท์ บัณฑิตวงษ์

NONT BANDITWONG

เลขหมู่.....
เลขทะเบียน 45649
วัน, เดือน, ปี 12 ก.พ. 2546

วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พศ. 2545

ISBN 974-324-124-8

A STUDY OF PARALLEL VOLUME RENDERING ON CLUSTER SYSTEM

NONT BANDITWONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2002

ISBN 974-324-124-8

COPYRIGHT 2002

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การศึกษาการประมวลผลแบบขนานบนระบบคลัสเตอร์ในการสร้างภาพเชิงปริมาตร
นักศึกษา	นายนนท์ บัณฑิตวงษ์
รหัสประจำตัว	42061168
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2545
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.มนัส สังวรศิลป์

บทคัดย่อ

เนื้อหาของวิทยานิพนธ์เล่มนี้ นำเสนอวิธีการเพิ่มความเร็วในการสร้างภาพเชิงปริมาตร (Volume Rendering) ทางการแพทย์ด้วยการกระจายกลุ่มงานย่อยไปประมวลผลอย่างพร้อมๆกันบนเครื่องคอมพิวเตอร์แต่ละเครื่องบนระบบคลัสเตอร์ (Cluster) การกำหนดงาน การรับค่าสีของจุดภาพที่ประมวลผลเสร็จ และการแสดงผลภาพจะถูกทำจากโพรเซสหนึ่งที่ทำหน้าที่เสมือนเป็นผู้จัดการ(Manager) และโพรเซสอื่นๆซึ่งถูกกำหนดเป็นเสมือนคนงาน (Worker) จะหน้าที่ประมวลผลเพื่อสร้างค่าสีของจุดภาพ (Pixel)

ในวิทยานิพนธ์ฉบับนี้ได้ทำการปรับปรุงวิธีการกำหนดงานเพื่อให้แต่ละโพรเซสได้รับปริมาณงานที่เหมาะสมไปประมวลผลด้วยวิธีการกระจายงานแบบ Work Pool โดยที่แต่ละโพรเซสยังคงสามารถประมวลผลให้เสร็จในเวลาที่ใกล้เคียงกันถึงแม้ว่าปริมาณงานในเครื่องคอมพิวเตอร์แต่ละเครื่องบนระบบคลัสเตอร์จะไม่เท่ากันก็ตาม จากการทดสอบเพื่อสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์ ซึ่งทำการกระจายพิกัดจุดเริ่มต้นและขนาดของแต่ละบล๊อคในภาพไปประมวลผลบนโพรเซสต่างๆภายในระบบคลัสเตอร์นี้สามารถลดเวลาในการสร้างภาพเชิงปริมาตรได้กว่า 88 เปอร์เซ็นต์

Thesis Title	A Study of Parallel Volume Rendering on Cluster System
Student	Mr.Nont Banditwong
Student ID.	42061168
Degree	Master of Enginering
Programme	Electrical Engineering
Year	2002
Thesis Advisor	Assoc.Prof.Dr.Manas Sangworasil

ABSTRACT

This thesis exploits a concept of parallel programming method to speed-up a medical volume rendering process, by distributing work for processing concurrently on each computer in clustering system. The system mainly comprises of manager and workers. Manager schedules works for each worker; in this case building a group of pixel values. The finished work from each worker is send back to the manager. The resulting pixel is then displayed.

The thesis improves a scheduling process in the clustering system by distributing an appropriate work to each worker using a work-pool scheduling scheme. In this scheme, though the work load for each process in a clustering system is uneven, the variation of the processing time for each workers is insignificant. Our purposed system workers successfully to reduce the rendering time of up to 88 percent.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดีเนื่องมากจากความช่วยเหลือจากบุคคลหลายๆ ฝ่ายด้วยกัน

ขอบคุณ คุณพ่อ คุณแม่ ที่ให้ความสนับสนุนด้านทุนการศึกษาตลอดจนคำแนะนำและกำลังใจในการทำวิทยานิพนธ์ฉบับนี้ด้วยดีเสมอมา

ขอขอบคุณ รศ.ดร.มนัส สังวรศิลป์ อาจารย์ที่ปรึกษาผู้ซึ่งคอยให้คำปรึกษา และให้ความสนับสนุนทางด้านความรู้และแนวทางในการทำวิทยานิพนธ์ ตลอดระยะเวลาการศึกษา รวมถึงเอื้อเฟื้อสนับสนุนเครื่องมือ สถานที่ที่ใช้ในการทำวิจัยและที่พักอาศัยอย่างพร้อมเพรียง ผู้เขียนรู้สึกซาบซึ้งในความเมตตาของท่านเป็นอย่างสูง

ขอขอบคุณ ผศ.ดร.บำรุง สมสวัสดิ์ในเรื่องข้อมูลและคำปรึกษาเกี่ยวกับระบบคลัสเตอร์และให้ความอนุเคราะห์ในการใช้ระบบคลัสเตอร์ของภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่นเพื่อทดสอบโปรแกรมแบบขนาน

ขอบคุณ คุณวรเทพ ไพบูลย์รัตนากร ที่ให้คำแนะนำเกี่ยวกับการสร้างภาพสามมิติทางการแพทย์ทั้งสองแบบ

ขอขอบคุณเพื่อนๆในระบบอินเตอร์เน็ต รวมไปถึง Usenet News Group ต่างๆที่ช่วยเหลือในด้านเอกสาร ข้อคิดเห็น และช่วยแก้ไขปัญหาที่ผู้เขียนไม่สามารถแก้ได้ด้วยตัวเอง ตลอดจนซอร์ฟแวร์ฟรีจาก MPICH และ GNU/Linux ที่ทำให้เกิดวิทยานิพนธ์ฉบับนี้ขึ้นมาได้

ขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่สนับสนุนระบบอินเตอร์เน็ตความเร็วสูงเพื่อให้สามารถค้นหาข้อมูลจากระบบอินเทอร์เน็ตได้อย่างมีประสิทธิภาพ

และสุดท้ายขอบคุณ เพื่อนๆ พี่ๆ น้องๆ ในห้องวิจัยที่ได้แบ่งปันความรู้หลายๆอย่างตลอดจนช่วยเหลือในการจัดทำวิทยานิพนธ์ฉบับนี้

นนท์ บัณฑิตวงษ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VIII

สารบัญรูป.....IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....2

1.3 สมมติฐานของการศึกษา.....2

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในงานวิจัย.....2

1.5 ขอบเขตการวิจัย.....3

บทที่ 2 หลักการสร้างภาพสามมิติเบื้องต้น.....5

2.1 บทนำ.....5

2.2 ภาพสามมิติทางการแพทย์.....5

2.3 ระบบพิกัด และการแปลงเรขาคณิต.....6

2.4 การส่องสว่าง และ การให้แสงเงา.....12

2.4.1 เกรเดียนต์.....12

2.4.1.1 การหาค่าเกรเดียนด้วยวิธีตัวประมาณค่าเกรเดียนต์แบบเซ็นทรัลดิฟเฟอเรนซ์.....13

2.4.1.2 การใช้ค่าเกรเดียนต์ในขบวนการสร้างภาพเชิงปริมาตร.....14

2.4.2 แสงแวดล้อม.....14

2.4.3 การสะท้อนแบบกระจาย.....15

2.4.4 การสะท้อนแบบกระจก.....16

2.5 สรุป.....18

สารบัญ (ต่อ)

หน้า

บทที่ 3 การสร้างภาพเชิงปริมาตร.....19

3.1 บทนำ.....19

3.2 การเตรียมข้อมูลภาพ.....20

3.3 การฉายแสง.....21

3.3.1 การอินเทอโพเลทแบบเชิงเส้น.....23

3.3.2 คิวบิกคอนโวลูชั่นอินเทอโพเลชั่น.....25

3.4 การแบ่งกลุ่มข้อมูล.....26

3.4.1 ฮีสโตแกรม.....27

3.4.2 ฟังก์ชันถ่ายโอน.....27

3.4.3 การให้สี.....30

3.5 การประกอบภาพ.....30

3.5.1 การประกอบภาพแบบหน้าไปหลัง.....32

3.5.2 การประกอบภาพแบบหลังไปหน้า.....34

3.6 สรุป.....35

บทที่ 4 การประมวลผลแบบขนานและโครงสร้างของระบบคลัสเตอร์.....36

4.1 บทนำ.....36

4.2 การประมวลผลแบบขนาน.....36

4.2.1 การประยุกต์ใช้การประมวลผลแบบขนาน.....37

4.3 การจำแนกประเภทของคอมพิวเตอร์.....39

4.3.1 จำแนกตามงานที่ประยุกต์ใช้.....41

4.3.2 จำแนกตามลักษณะของเครื่องในระบบ.....41

4.4 องค์ประกอบและการสร้างระบบคลัสเตอร์.....42

4.4.1 ระบบเครือข่ายความเร็วสูง.....42

4.4.2 โปรแกรมบริการการสื่อสาร.....44

4.4.3 ระบบปฏิบัติการ.....45

4.4.4 คลัสเตอร์มิดเดิ้ลแวร์.....46

4.4.5 โปรแกรมสำหรับสร้างสภาวะแวดล้อมของการโปรแกรมแบบขนาน.....46

4.5 สรุป.....47

สารบัญ (ต่อ)

หน้า

บทที่ 5 โปรแกรมแบบขนาน.....48

5.1 บทนำ.....48

5.2 รูปแบบของการพัฒนาโปรแกรมแบบขนาน.....48

5.2.1 การสร้างโปรแกรมแบบขนานโดยอัตโนมัติ.....49

5.2.2 การสร้างโปรแกรมแบบขนานโดยใช้คลังของชุดคำสั่งแบบขนาน.....49

5.2.3 การสร้างโปรแกรมแบบขนานด้วยตัวเอง.....50

5.3 วิธีการออกแบบอัลกอริทึมของโปรแกรมแบบขนาน.....50

5.3.1 ขั้นตอนการแบ่งงาน.....51

5.3.2 ขั้นตอนออกแบบการสื่อสาร.....52

5.3.3 ขั้นตอนรวมกลุ่มงาน.....54

5.3.4 ขั้นตอนกำหนดงาน.....55

5.4 การสื่อสารระหว่างงานด้วยวิธีส่งข้อความ.....55

5.4.1 การสื่อสารแบบจุดต่อจุด.....59

5.4.2 สื่อการเป็นกลุ่ม.....59

5.5 การสร้างภาพเชิงปริมาตรทางการแพทย์บนระบบคลัสเตอร์.....61

5.5.1 การแบ่งงานในการสร้างภาพเชิงปริมาตร.....62

5.5.2 การรวมกลุ่มงาน.....62

5.5.3 การสื่อสารระหว่างงานในการสร้างภาพเชิงปริมาตร.....63

5.6 สรุป.....65

บทที่ 6 การทดลองและผลการทดลอง.....66

6.1 บทนำ.....66

6.2 การวัดประสิทธิภาพของการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์.....66

6.3 ผลของการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์.....69

6.4 ผลของการจำลองการกำหนดงานให้แก่เครื่องในระบบคลัสเตอร์ด้วยวิธีการที่นำเสนอ.....74

สารบัญ (ต่อ)

หน้า

บทที่ 7 สรุปผลและแนวทางการพัฒนา.....75

7.1 สรุปและวิจารณ์ผลการทดลอง.....75

7.2 แนวทางการพัฒนา.....76

เอกสารอ้างอิง.....77

ภาคผนวก ผลงานวิจัยที่ได้รับการตีพิมพ์.....79

ประวัติผู้เขียน.....80

สารบัญตาราง

หน้า

6.1 ผลของเวลาที่ใช้ในการสร้างภาพเชิงปริมาตรของข้อมูลขนาด 587x341x500 จุดภาพ.....70

สารบัญรูป

รูปที่ หน้า

2.1 ระบบพิกัดมือขวา.....6

2.2 การอ้างอิงตำแหน่งของวัตถุในระบบภาพสองมิติ.....6

2.3 การเลื่อนวัตถุในสามมิติ.....9

2.4 การเปลี่ยนขนาดของวัตถุในสามมิติ.....10

2.5 การหมุนวัตถุรอบแกน X ในสามมิติ.....10

2.6 การหมุนวัตถุรอบแกน Y ในสามมิติ.....11

2.7 การหมุนวัตถุรอบแกน Z ในสามมิติ.....12

2.8 แสดงการหาค่าเวกเตอร์ปกติจากเกรเดียนต์ของพื้นผิว.....13

2.9 ผลของแสงแวดล้อมที่มีต่อวัตถุ ซึ่งมีค่า ka ต่างๆกัน.....15

2.10 แสดงการตกกระทบของแสงในพื้นผิวที่ตั้งฉากกับแนวลำแสง เทียบกับพื้นผิว
ที่ไม่ตั้งฉากกับแนวลำแสง.....15

2.11 ผลของการสะท้อนแสงแบบกระจายร่วมกับผลของแสงแวดล้อมที่มีต่อวัตถุที่มีค่า kd ต่างๆกัน
และ แสงแวดล้อมมีค่า ka=0.3.....16

2.12 องค์ประกอบการสะท้อนแบบกระจก.....16

2.13 ผลของการสะท้อนแสงแบบกระจกร่วมกับผลของการสะท้อนแบบกระจาย และแสงแวดล้อม
ที่มีต่อวัตถุ.....17

3.1 ขั้นตอนในการสร้างภาพเชิงปริมาตร.....19

3.2 ปริมาตรข้อมูลภาพ.....20

3.3 ขั้นตอนการเตรียมข้อมูลภาพ.....20

3.4 การฉายแสงเข้าไปในปริมาตรข้อมูลภาพ.....22

3.5 การฉายแสงในสองมิติโดยแปลงตำแหน่งของระนาบการมอง.....22

3.6 ความสัมพันธ์แบบเชิงเส้นของข้อมูล.....23

3.7 การอินเทอโพเลทในสามมิติ.....24

3.8 ระยะทางระหว่างจุดสี่จุดที่ใช้ในการอินเทอโพเลท.....25

3.9 คิวบิกคอนโวลูชั่นในสองสิติ.....26

3.10 ภาพซึ่งเกิดจากการขยายภาพสิบเท่าโดยใช้อินเทอโพเลชั่นแบบต่างๆ.....26

3.11 ฮีสโตแกรมของข้อมูลภาพทางการแพทย์ที่ถ่ายด้วยเครื่อง CT.....27

3.12 ฟังก์ชันถ่ายโอนของ Levoy.....29

สารบัญรูป(ต่อ)

หน้า

3.13 การแยกแยะชนิดของอวัยวะซึ่งมีขอบเขตผสมกัน.....29
3.14 ฟังก์ชันถ่ายโอนของข้อมูลที่มีมากกว่าหนึ่งวัตถุ.....29
3.15 การเดินทางของแสงจาก a ไป b โดยมีทิศทาง ds.....31
3.16 ลักษณะและทิศทางของการประกอบภาพแบบหน้าไปหลัง.....33
3.17 ลักษณะและทิศทางของการประกอบภาพแบบหน้าไปหลัง.....34
4.1 ความสัมพันธ์ระหว่างวิทยาศาสตร์เชิงทฤษฎี การทดลอง และเชิงคำนวณ.....38
4.2 แผนผังการจำแนกประเภทของระบบคอมพิวเตอร์.....40
4.3 โครงสร้างของระบบคลัสเตอร์.....42
4.4 การเชื่อมต่อเครื่องในระบบคลัสเตอร์เข้ากับระบบเครือข่ายส่วนตัว (Private Network).....43
4.5 การเชื่อมต่อระบบเครือข่ายแบบกับศูนย์รวมเซอร์ฟเวอร์.....44
4.6 ระบบคลัสเตอร์ที่การสื่อสารกันภายในระบบจะถูกทำผ่านระบบเครือข่ายความเร็วสูง.....44
5.1 ขั้นตอนการทำงานของการสร้างโปรแกรมแบบขนานโดยอัตโนมัติ.....49
5.2 ขั้นตอนการสร้างโปรแกรมแบบขนานโดยใช้คลังของชุดคำสั่งแบบขนาน.....50
5.3 ขั้นตอนการสร้างโปรแกรมแบบขนานด้วยตัวเอง.....50
5.4 การแบ่งขอบเขตข้อมูลของปัญหา.....51
5.5 การแบ่งขอบเขตหน้าที่ของปัญหา.....51
5.6 ลักษณะการสื่อสารในการคำนวณไฟไนต์ดิฟเฟอเรนต์ในสองมิติ.....52
5.7 การสื่อสารเพื่อรวมค่าจากงานที่เป็นผู้จัดการ.....53
5.8 ตาราง (Grid) ซึ่งสร้างจากวิธีวิเคราะห์ไฟไนต์อีลีเมนต์.....53
5.9 การรวมกลุ่มงานในการประมวลผลแบบขนานเพื่อลดเวลาในการสื่อสาร.....54
5.10 แต่ละเครื่องในระบบคลัสเตอร์มีหน่วยความจำของตัวเอง.....56
5.11 ลักษณะรหัสต้นฉบับซึ่งมีโครงสร้างแบบโปรแกรมเดี่ยวหลายข้อมูล.....57
5.12 การจัดกลุ่มงานใน MPI.....58
5.13 การส่งและรับข้อมูลของ MPI.....58
5.14 ลักษณะการทำงานของ MPI_Bcast.....60
5.15 ลักษณะการทำงานของ MPI_Scatter.....60
5.16 ลักษณะการทำงานของ MPI_Gather.....61
5.17 ลักษณะการทำงานของ MPI_Reduce.....61
5.18 การแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อยตามแนวลำแสง.....62

สารบัญรูป(ต่อ)

หน้า

5.19โครงสร้างข้อมูลในการสื่อสารของการสร้างภาพเชิงปริมาตร.....63
5.20 การกำหนดงานแบบศูนย์รวมงาน (Work Pool).....63
5.21 ขั้นตอนการทำงานของโปรแกรมในส่วนผู้จัดการ.....64
5.22 ขั้นตอนการทำงานของโปรแกรมในส่วนคนงาน (Worker).....65
6.1 กราฟในอุดมคติของเวลาที่ใช้ในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์.....68
6.2 กราฟในอุดมคติของอัตราการเพิ่มความเร็วในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์.....68
6.3 กราฟในอุดมคติของประสิทธิภาพในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์.....69
6.4 กราฟของเวลาที่ใช้ในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ของข้อมูลภาพขนาด 587x341x500 จุดภาพ.....71
6.5 กราฟของประสิทธิภาพในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ของข้อมูลภาพขนาด 587x341x500 จุดภาพ.....71
6.6 กราฟของประสิทธิภาพในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ของข้อมูลภาพขนาด 587x341x500 จุดภาพ.....72
6.7 ผลของการสร้างภาพของปริมาตรภาพขนาด 587x341x500 จุดภาพ.....73
6.8 ผลการจำลองการกำหนดงานบนระบบคลัสเตอร์.....74

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ภาพทางการแพทย์ (Medical Image) มีไว้เพื่อช่วยแพทย์ในการวินิจฉัยโรค ซึ่งในอดีตภาพทางการแพทย์จะถูกสร้างเป็นสองมิติ อวัยวะทั้งหมดจะถูกฉาย (Project) ลงมาซ้อนทับกันบนระนาบสองมิติ ผู้ที่สามารถเข้าใจภาพถ่ายและสามารถวินิจฉัยโรคจากภาพเหล่านี้ได้จะต้องเป็นผู้ที่มีความเชี่ยวชาญในการอ่านภาพทางการแพทย์นี้เช่นรังสีแพทย์ ในปัจจุบันเทคโนโลยีการสร้างภาพทางการแพทย์มีความก้าวหน้าไปมากโดยสามารถสร้างภาพถ่ายของอวัยวะภายในร่างกายที่เป็นสามมิติได้ ทำให้ผู้ที่ไม่ใช่ผู้เชี่ยวชาญเฉพาะด้านสามารถเข้าใจความหมายของภาพทางการแพทย์ได้อย่างไม่ยากนัก และช่วยให้รังสีแพทย์สามารถสื่อสารกับแพทย์แขนงอื่นเกี่ยวกับการรักษาโรคของผู้ป่วยได้ง่ายมากขึ้น ภาพสามมิตินี้สามารถสร้างจากเครื่องถ่ายภาพทางการแพทย์เช่น เครื่อง CT (Computed Tomography) และ MRI (Magnetic Resonance Imaging) เป็นต้น

วิธีการสร้างภาพสามมิติทางการแพทย์อาจแบ่งเป็นสองวิธีใหญ่ๆคือ การสร้างภาพเชิงพื้นผิว และการสร้างภาพเชิงปริมาตร การสร้างภาพเชิงพื้นผิวมีข้อดีคือสามารถสร้างภาพได้รวดเร็ว แต่ไม่สามารถสร้างภาพของอวัยวะบางอย่างที่มีขอบเขตไม่แน่นอนเช่นเนื้อเยื่อได้อย่างถูกต้อง ภาพเชิงปริมาตรถูกสร้างขึ้นมาเพื่อแก้ไขข้อจำกัดของภาพเชิงพื้นผิวโดยสามารถแสดงภาพของอวัยวะที่มีขอบเขตไม่แน่นอนได้ แต่การสร้างภาพเชิงปริมาตรมีข้อเสียคือใช้เวลาในการสร้างภาพแต่ละมุมมองมาก การเพิ่มความเร็วในการสร้างภาพเชิงปริมาตรทางการแพทย์อาจทำได้สามวิธี ซึ่งในแต่ละวิธีก็มีข้อดีข้อเสียแตกต่างกัน

1. ใช้ฮาร์ดแวร์ (Hardware) ที่มีความเร็วสูงในการประมวลผลเพื่อสร้างภาพ
2. ปรับปรุงอัลกอริทึมของโปรแกรมที่ใช้ในการสร้างภาพ [21]
3. นำโปรแกรมที่ใช้สร้างภาพไปประมวลผลบนระบบคอมพิวเตอร์แบบขนาน

วิธีแรกอาจทำได้ง่ายในปัจจุบันเนื่องจากอุปกรณ์คอมพิวเตอร์มีราคาถูกลงมาก แต่ไม่สามารถรองรับงานที่อาจมีมากขึ้นในอนาคตได้ วิธีที่สองเป็นวิธีที่ดีที่สุดแต่อาจทำได้ยาก โดยปัจจุบันยังมีงานวิจัยที่ออกมาอย่างต่อเนื่อง[21] และวิธีสุดท้ายเป็นวิธีที่กำลังได้รับความสนใจเป็นอย่างมากในปัจจุบันเนื่องจากสามารถใช้ร่วมกับวิธีที่หนึ่งและสอง เพื่อเพิ่มความเร็วในการประมวลผลให้มากขึ้นไปอีกได้

ระบบการประมวลผลแบบขนานในปัจจุบันมีหลายแบบ และแบบที่กำลังได้รับความสนใจเป็นอย่างมากเนื่องจากสร้างได้ง่าย ราคาถูก ดูแลรักษาง่าย และเพิ่มจำนวนหน่วยการประมวลผลได้ง่ายโดยเสียค่าใช้จ่ายน้อย คือระบบคลัสเตอร์ (Cluster) [1] ระบบคลัสเตอร์สามารถใช้งานจากที่ใดก็ได้ในระบบเครือข่าย (Network System) ซึ่งอาจตั้งอยู่ที่ศูนย์คอมพิวเตอร์ของหน่วยงาน นอกจากระบบคลัสเตอร์จะสามารถช่วยเพิ่มความเร็วในการประมวลผลภาพเชิงปริมาตรได้แล้วยังสามารถใช้ร่วมกันกับงานประมวลผลอื่นๆที่ใช้เวลาในการประมวลผลมากได้อย่างสะดวก เช่นงานสร้างแบบจำลองเพื่อสร้างเซลล์แสงอาทิตย์เป็นต้น [20]

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อพัฒนาให้การสร้างภาพเชิงปริมาตรทางการแพทย์ในแต่ละมุมมองทำได้รวดเร็วขึ้นด้วยการประมวลผลบนระบบคลัสเตอร์
2. เพื่อแนะนำการสร้างและใช้งานระบบคลัสเตอร์อย่างง่าย โดยใช้ทรัพยากรที่มีอยู่ในห้องปฏิบัติการ
3. เพื่อที่จะสามารถนำหลักการที่ได้นำเสนอไปประยุกต์ใช้กับการประมวลผลอื่นๆที่ต้องใช้เวลานานได้ต่อไป

1.3 สมมติฐานของการศึกษา

อัลกอริทึม (Algorithm) ของการสร้างภาพเชิงปริมาตรสามารถแบ่งออกเป็นงานย่อยๆได้ ซึ่งสามารถกระจายให้แต่ละโพรเซส (Process) ในระบบคลัสเตอร์ประมวลผลไปพร้อมๆกันได้ ดังนั้นเวลาที่ใช้ในการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์นี้ควรจะลดน้อยลง

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในงานวิจัย

ข้อได้เปรียบของอัลกอริทึมในการประมวลผลภาพส่วนใหญ่คือสามารถแบ่งงานออกเป็นส่วนย่อยๆได้ง่าย การสร้างภาพเชิงปริมาตรเป็นอีกงานหนึ่งที่สามารถแบ่งงานออกเป็นส่วนย่อยได้และสามารถประยุกต์ใช้กับอัลกอริทึมของการสร้างภาพเชิงปริมาตรแบบต่างๆได้

ในงานวิจัยนี้ทำการแบ่งการคำนวณที่ใช้ในการสร้างภาพออกเป็นส่วนย่อยแล้วทำการกระจายไปให้เครื่องต่างๆในระบบคลัสเตอร์ประมวลผลอย่างพร้อมๆกัน จากนั้นส่งกลุ่มของจุดภาพที่สร้างเสร็จไปแสดงผลในเครื่องที่กำหนดไว้ในระบบ ในระบบคลัสเตอร์ที่เป็นเนื้อผสม (Heterogeneous) กล่าวคือแต่ละเครื่องในระบบคลัสเตอร์อาจมีส่วนประกอบที่แตกต่างกันเช่นความแตกต่างของหน่วยประมวลผลกลาง (CPU) ความแตกต่างของหน่วยความจำ ความแตกต่างของฮาร์ดดิสก์ (Hard disk) ความแตกต่างของระบบปฏิบัติการ หรือแม้กระทั่งความแตกต่างของปริมาณ

งานในแต่ละเครื่องมีผลทำให้ความสามารถของการประมวลผลของแต่ละเครื่องแตกต่างกัน ดังนั้น ถ้าแบ่งงานขนาดเท่าๆกันให้แต่ละโพรเซสของเครื่องประมวลผล แต่ละโพรเซสอาจทำงานเสร็จไม่พร้อมกันโดยบางโพรเซสอาจว่างงานในขณะที่บางโพสเซสทำงานตลอดเวลาซึ่งจะทำให้การประมวลผลไม่ได้ผลที่เร็วที่สุดที่ระบบสามารถทำได้ ดังนั้นเทคนิคการกระจายงานที่ดีจะทำให้ทุกโพสเซสทำงานอย่างต่อเนื่องและเสร็จงานในเวลาที่ใกล้เคียงกัน

ในงานวิจัยนี้ใช้วิธีการกำหนดงานแบบ ศูนย์รวมงาน (Work Pool) ซึ่งอาศัยหลักการที่ว่าให้โพรเซสที่ว่างงานไปรับงานจากศูนย์รวมงานแทนที่จะกำหนดงานให้แก่แต่ละโพรเซสล่วงหน้า ซึ่งวิธีนี้จะทำให้โพรเซสที่ทำงานอยู่บนเครื่องที่มีความสามารถในการประมวลผลต่ำซึ่งจะทำงานเสร็จได้ช้ามารับงานได้น้อยครั้งกว่าโพรเซสที่ทำงานอยู่บนเครื่องที่มีความสามารถในการประมวลผลมากกว่าซึ่งจะมารับงานมาประมวลผลบ่อยครั้งกว่า ทำให้สามารถหลีกเลี่ยงเหตุการณ์ที่งานหลายๆงานไปค้างอยู่บนโพรเซสที่ประมวลผลได้ช้า

1.5 ขอบเขตการวิจัย

วิทยานิพนธ์ฉบับนี้มุ่งเน้นที่การเพิ่มความเร็วของการสร้างภาพเชิงปริมาตรทางการแพทย์ โดยกระจายงานย่อยๆไปทำการประมวลผลบนแต่ละเครื่องในระบบคลัสเตอร์ โดยรายละเอียดต่างๆภายในวิทยานิพนธ์ฉบับนี้ได้จัดแบ่งเนื้อหาออกเป็น 7 บทซึ่งแต่ละบทมีหัวข้อและเนื้อหาดังต่อไปนี้

บทที่ 1 บทนำ

อธิบายถึงความเป็นมาและความสำคัญของปัญหาตลอดจนวัตถุประสงค์และขอบเขตของวิทยานิพนธ์

บทที่ 2 หลักการสร้างภาพสามมิติเบื้องต้น

อธิบายถึงกลักการและทฤษฎีที่เกี่ยวข้องกับการสร้างภาพสามมิติ ซึ่งประกอบด้วยระบบพิกัด ระบบเมตริกและการแปลงเรขาคณิต การส่องสว่างและการให้แสงเงา การประมาณค่าเวกเตอร์ปกติของพื้นผิวด้วยค่าเกรเดียน ผลของแสงแวดล้อม และการจำลองการสะท้อนของแสงบนวัตถุ

บทที่ 3 การสร้างภาพเชิงปริมาตร

อธิบายขั้นตอนและวิธีการสร้างภาพเชิงปริมาตรซึ่งประกอบด้วยการเตรียมข้อมูลภาพ การฉายแสงเข้าในปริมาตรข้อมูลภาพ การอินเทอโพเลทแบบต่างๆ การแบ่งกลุ่มข้อมูลและการประกอบภาพ

บทที่ 4 การประมวลผลแบบขนานและโครงสร้างของระบบคลัสเตอร์

อธิบายถึงที่มาของการประมวลผลแบบขนาน การประยุกต์ใช้การประมวลผลแบบขนานในงานต่างๆ ประเภทของระบบคอมพิวเตอร์และระบบการประมวลผลแบบขนาน ตลอดจนโครงสร้างของระบบคลัสเตอร์

บทที่ 5 การโปรแกรมแบบขนาน

กล่าวถึงลักษณะของโปรแกรมแบบขนาน รูปแบบและวิธีการสร้างโปรแกรมแบบขนานซึ่งประกอบด้วยการแบ่งงาน ขั้นตอนออกแบบการสื่อสาร ขั้นตอนรวมกลุ่มงาน และขั้นตอนการกำหนดงานเป็นขั้นตอนสุดท้าย

นอกจากนั้นยังได้กล่าวถึงลักษณะการทำงานของคลังคำสั่ง(Library) MPI เพื่อใช้ในการส่งข้อความระหว่างโพรเซสในระบบคลัสเตอร์ และส่วนสุดท้ายเป็นวิธีการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์

บทที่ 6 การทดลองและผลการทดลอง

ในการทดลองได้กล่าวถึงวิธีการวัดประสิทธิภาพของโปรแกรมแบบขนาน วิธีการทดลองและผลการทดลอง

บทที่ 7 สรุปผลและแนวทางการพัฒนา

กล่าวถึงบทสรุปของวิทยานิพนธ์และแนวทางการวิจัยต่อไป

บทที่ 2

หลักการสร้างภาพสามมิติเบื้องต้น

2.1 บทนำ

ในบทนี้จะเป็นการอธิบายถึงหลักการและทฤษฎีที่เกี่ยวข้องกับการสร้างภาพสามมิติโดย ส่วนแรกจะกล่าวถึงลักษณะของการสร้างภาพสามมิติทางการแพทย์อย่างคร่าวๆ ระบบพิกัดและการแปลงเรขาคณิตในสามมิติซึ่งใช้สำหรับอ้างอิงวัตถุต่างๆในระบบ จากนั้นจะกล่าวถึงการฉายแสงซึ่งใช้สำหรับสร้างจุดภาพภาพในระนาบการมอง และจะกล่าวถึงวิธีการให้แสงเงาเพื่อให้ภาพที่ได้ดูสมจริงมากขึ้นเป็นลำดับต่อไป

2.2 ภาพสามมิติทางการแพทย์

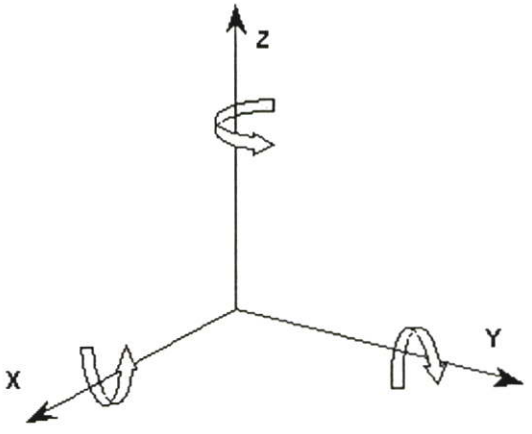
การวินิจฉัยโรคหลายโรคในปัจจุบันสามารถทำได้โดยอาศัยภาพถ่ายอวัยวะภายในร่างกาย ซึ่งทำได้โดยใช้เครื่อง X-Ray Computed Tomography (CT), Angiography, Positron Emission Tomography (PET), หรือ Magnetic Resonance Imaging (MRI) เครื่องถ่ายภาพทางการแพทย์เหล่านี้สามารถช่วยแพทย์ในการวินิจฉัยโรคให้มีความถูกต้องมากยิ่งขึ้น รวมไปถึงการติดตามผลของการรักษาให้เป็นไปอย่างมีประสิทธิภาพ

การทำความเข้าใจภาพทางการแพทย์นั้นมีความซับซ้อน ผู้ที่จะสามารถเข้าใจภาพเหล่านี้ได้ต้องมีความรู้เกี่ยวกับโครงสร้าง และความสัมพันธ์ของอวัยวะภายในร่างกายเป็นอย่างดี เนื่องจากโรคหลายโรคจะทำให้โครงสร้างและหน้าที่การทำงานของอวัยวะภายในร่างกายเปลี่ยนแปลงไป การดูภาพของภาคตัดขวางจากเครื่องถ่ายภาพทางการแพทย์อาจไม่เพียงพอที่จะวินิจฉัยโรค การนำภาพเหล่านั้นกลับมาสร้างใหม่ (Reconstruct) ให้เป็นภาพสามมิติจะช่วยให้การวินิจฉัยโรคทำได้ง่ายและถูกต้องมากยิ่งขึ้น ในปัจจุบันเทคนิคการสร้างภาพสามมิติทางการแพทย์อาจทำได้สองวิธีคือ การสร้างภาพเชิงพื้นผิว (Surface Rendering) [2] และการสร้างภาพเชิงปริมาตร (Volume Rendering) [3]

การสร้างภาพเชิงพื้นผิวมีข้อดีคือสามารถสร้างภาพได้อย่างรวดเร็ว แต่ไม่สามารถแสดงภาพของอวัยวะที่มีขอบเขตไม่แน่นอนเช่นเนื้อเยื่อได้ ภาพเชิงปริมาตรถูกสร้างขึ้นมาเพื่อแก้ไขข้อจำกัดของการสร้างภาพเชิงพื้นผิว ซึ่งสามารถแสดงภาพของอวัยวะที่มีขอบเขตไม่แน่นอนได้ดี แต่มีข้อเสียคือใช้เวลาสร้างภาพในแต่ละมุมมองค่อนข้างมาก

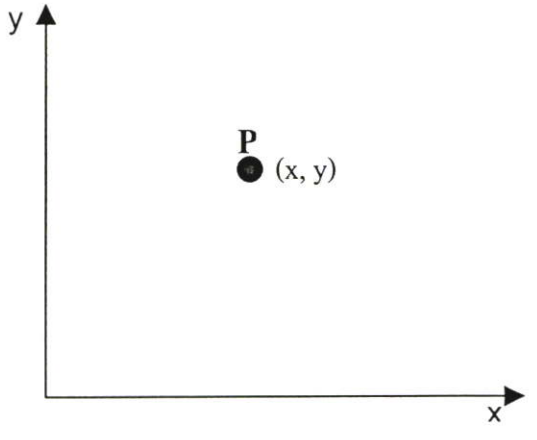
2.3 ระบบพิกัด และการแปลงเรขาคณิต (Coordinate System and Transformation)

ในการจำลองโครงสร้างของวัตถุหรือสร้างภาพเสมือนจริงในคอมพิวเตอร์จำเป็นต้องมีระบบพิกัดที่ใช้สำหรับอ้างอิงตำแหน่งและจัดการกับวัตถุในระบบ เช่น การเลื่อน (Transformation) การหมุน (Rotation) และการเปลี่ยนขนาด (Scaling) วัตถุเป็นต้น ระบบพิกัดที่ใช้อ้างอิงตำแหน่งของวัตถุในงานด้านคอมพิวเตอร์กราฟิกมีสองแบบคือระบบมือขวาและระบบมือซ้าย ระบบทั้งสองมีความแตกต่างกันในการอ้างอิงทิศทางการหมุนของแกน ซึ่งทำให้สมการที่ใช้ในการแปลงของสองระบบแตกต่างกันไปด้วย ในงานวิจัยนี้ใช้ระบบพิกัดมือขวาดังแสดงในรูปที่ 2.1



รูปที่ 2.1 ระบบพิกัดมือขวา

พิจารณาการอ้างอิงตำแหน่งของวัตถุในสองมิติ โดยกำหนดพิกัดของวัตถุในแกน X และ Y ตามลำดับ ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 การอ้างอิงตำแหน่งของวัตถุในระบบภาพสองมิติ

การเลื่อนวัตถุสามารถทำได้โดยใช้วิธีการบวกระยะที่ต้องการเลื่อนออกไปดังสมการ 2.1

$$P' = T + P \tag{2.1}$$

P' คือ ตำแหน่งวัตถุที่ต้องการเลื่อนไป

T คือ ระยะทางของการเลื่อนตามแนวแกน x และ y ตามลำดับ

P คือ ตำแหน่งตั้งต้นของวัตถุ

เช่น การเลื่อนจุดที่พิกัด (x, y) ไปตามแนวแกน x เป็นระยะทางสิบหน่วยจะมีรูปแบบดังสมการ 2.2 และ 2.3

$$x' = 10 + x \tag{2.2}$$

$$y' = 0 + y \tag{2.3}$$

การเปลี่ยนขนาดและการหมุนจะมีรูปแบบสมการที่แตกต่างออกไปจากการเลื่อนวัตถุ โดยจะใช้วิธีการคูณแทนการบวก ดังแสดงในสมการ 2.4 และ 2.5 ตามลำดับ

$$P' = S.P \tag{2.4}$$

$$P' = R.P \tag{2.5}$$

สมการ 2.1 2.4 และ 2.5 สามารถเขียนให้อยู่ในรูปของเมตริกได้ดังแสดงในสมการ 2.6 2.7 และ 2.8 ตามลำดับ [4] โดยที่ S คือ เมตริกของการเปลี่ยนขนาด (Scaling Matrix) T คือเมตริกของการเปลี่ยนตำแหน่ง (Transformation Matrix) และ R คือเมตริกของการหมุน (Rotation Matrix)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.6}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.7}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.8}$$

จากสมการทั้งสามจะเห็นได้ว่าไม่สามารถรวมเมตริกของการเปลี่ยนตำแหน่ง T เมตริกของการเปลี่ยนขนาด S และ และเมตริกของการหมุน R ให้อยู่ในรูปแบบทั่วไปได้เนื่องจากเมตริกของการ

เปลี่ยนตำแหน่งมีมิติและวิธีการ (operation) ที่แตกต่างออกไปจากพวก ซึ่งไม่สะดวกเมื่อนำไปประยุกต์ใช้งาน

ระบบพิกัดโฮโมจีเนียส (Homogeneous Coordinate System) [4] ถูกสร้างขึ้นมาเพื่อทำให้สามารถรวมเมตริกของการแปลงทั้งสามให้อยู่ในรูปแบบเดียวกันได้ โดยสมการในระบบพิกัดโฮโมจีเนียสนี้จะอยู่ในรูปของการคูณทั้งหมด และจะเพิ่มพิกัดที่สามให้กับจุด (x, y) คือ (x, y, W) อาจกล่าวได้ว่าพิกัดสองพิกัดในระบบพิกัดโฮโมจีเนียส (x', y', W') และ (x, y, W) เป็นจุดเดียวกัน ถ้าพิกัดทั้งสองเป็นจำนวนเท่าซึ่งกันและกันเช่น (3, 6, 3) และ (6, 12, 6) ซึ่งสามารถแสดงว่าพิกัดสองเป็นจุดเดียวกันได้ด้วยการทำนอร์มอลไลส์ (Normalize) พิกัดทั้งสองดังสมการ 2.9

$$\left(\frac{x}{W}, \frac{y}{W}, \frac{W}{W}\right) = \left(\frac{x'}{W'}, \frac{y'}{W'}, \frac{W'}{W'}\right) \tag{2.9}$$

เมตริกของการเลื่อน การเปลี่ยนขนาด และการหมุนในระบบพิกัดโฮโมจีเนียสเป็นดังสมการที่ 2.10 2.11 และ 2.12 ตามลำดับ

$$\mathrm{T} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

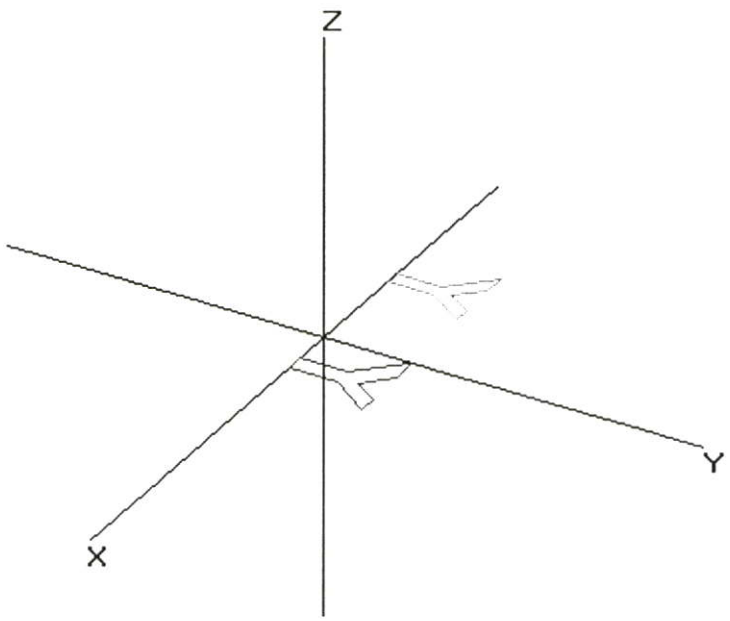
$$\mathrm{S} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_Y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

$$\mathrm{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

ในระบบภาพสามมิติ การอ้างอิงตำแหน่งของวัตถุจะทำโดยใช้พิกัดสามตัวคือ X, Y และ Z ซึ่งเมตริกของการแปลงในสามมิติจะมีลักษณะคล้ายกับเมตริกของการแปลงในสองมิติ เพียงแต่เพิ่มองค์ประกอบในแนวแกน Z เข้าไป และมิติของเมตริกจะเปลี่ยนเป็น 4x4 ซึ่งเมตริกของการเลื่อนตำแหน่ง การเปลี่ยนขนาด และการหมุนในแนวแกน X, Y และ Z แสดงดังสมการ 2.13 2.14 2.15 2.16 และ 2.17 ตามลำดับ

$$\begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.13}$$

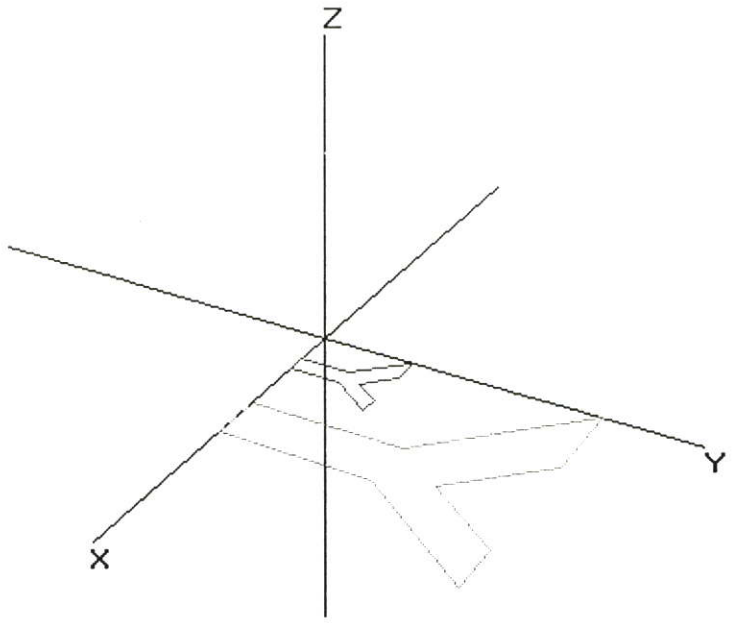
ตัวอย่างของการเลื่อนวัตถุในสามมิติแสดงดังรูปที่ 2.3



รูปที่ 2.3 การเลื่อนวัตถุในสามมิติ

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

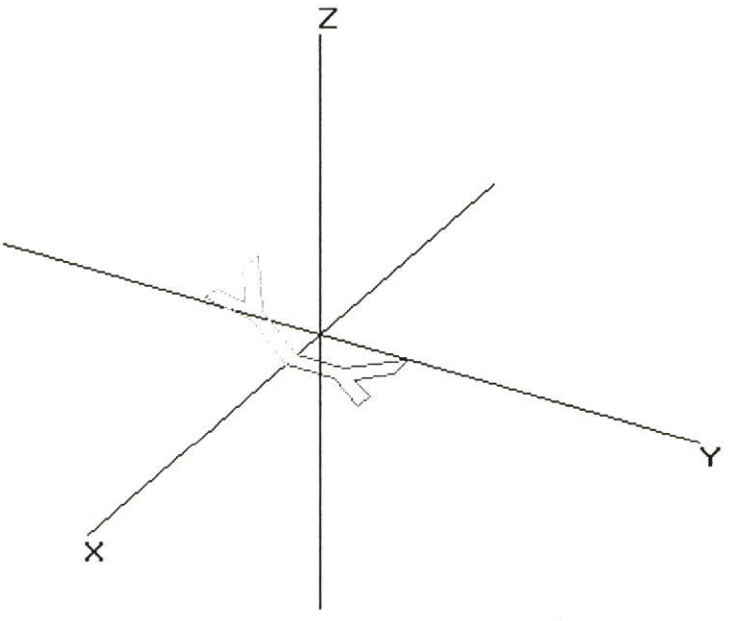
ตัวอย่างของการเปลี่ยนขนาดของวัตถุในสามมิติแสดงดังรูปที่ 2.4



รูปที่ 2.4 การเปลี่ยนขนาดของวัตถุในสามมิติ

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.15}$$

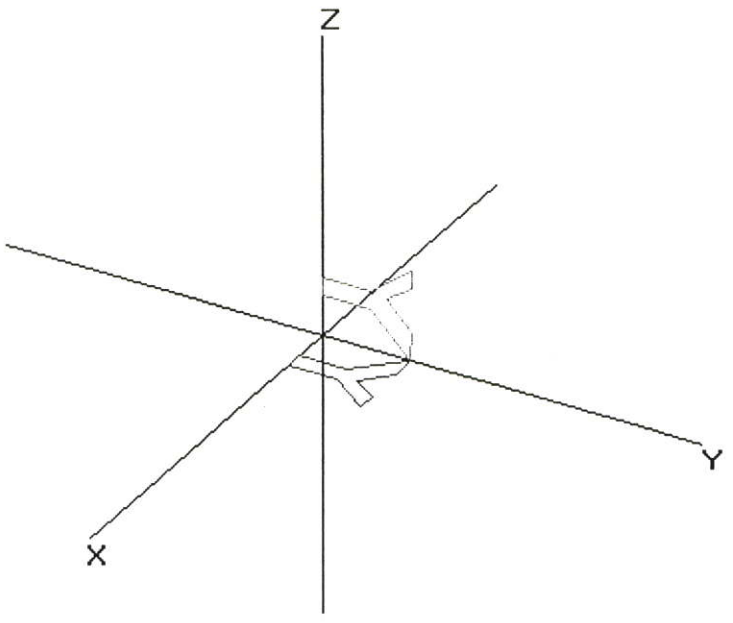
ตัวอย่างของการหมุนวัตถุรอบแกน X ในสามมิติ แสดงดังรูปที่ 2.5



รูปที่ 2.5 การหมุนวัตถุรอบแกน X ในสามมิติ

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

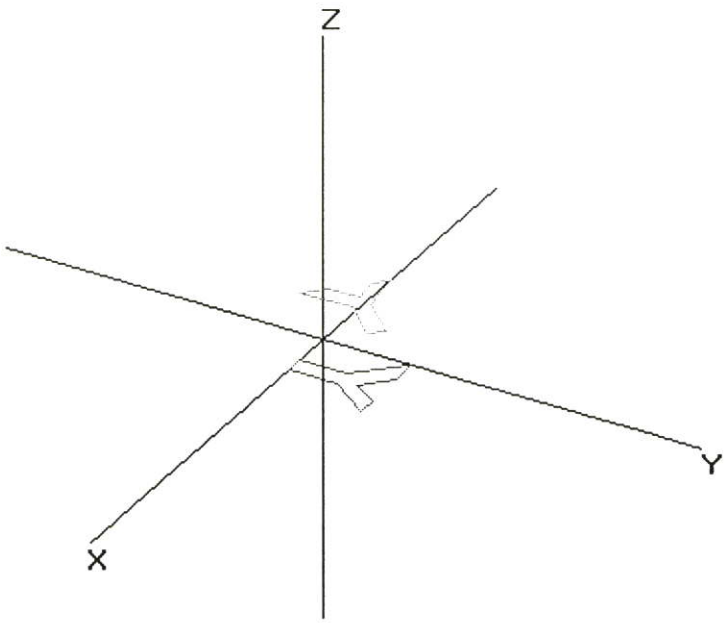
ตัวอย่างของการหมุนวัตถุรอบแกน Y ในสามมิติ แสดงดังรูปที่ 2.6



รูปที่ 2.6 การหมุนวัตถุรอบแกน Y ในสามมิติ

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.17}$$

ตัวอย่างของการหมุนวัตถุรอบแกน Z ในสามมิติ แสดงดังรูปที่ 2.7



รูปที่ 2.7 การหมุนวัตถุรอบแกน Z ในสามมิติ

2.4 การส่องสว่าง และ การให้แสงเงา (Illumination and Shading)

การส่องสว่างและการให้แสงเงาในคอมพิวเตอร์กราฟิกจะถูกทำเพื่อให้ภาพที่ได้มีความสมจริงมากยิ่งขึ้น โดยอาศัยสมการคณิตศาสตร์มาจำลองผลของแสงในธรรมชาติที่มีต่อวัตถุ ซึ่งประกอบด้วย การเกิดเงา การสะท้อน การกระเจิง และการถูกดูดกลืนของแสงเมื่อตกกระทบวัตถุ ผลของการให้แสงเงาในแต่ละวัตถุจะแตกต่างกันขึ้นอยู่กับทิศทางของแสง มุมระหว่างผู้สังเกต (Viewer) กับแสง และ คุณสมบัติของพื้นผิวของวัตถุเป็นต้น

ในการสร้างภาพเชิงปริมาตร ความสมจริงของภาพที่ได้อาจไม่ใช่สิ่งที่สำคัญมากที่สุด การให้แสงเงาจะถูกทำให้สามารถมองเห็นโครงสร้างของวัตถุได้ดีขึ้นซึ่งจะทำให้ผู้สังเกตเข้าใจความหมายของภาพได้ง่าย นอกจากนั้นการใช้แบบจำลองของการส่องสว่างและให้แสงเงาที่มีความสมจริงมากๆจะต้องใช้เวลาในการประมวลผลมากตามไปด้วย องค์ประกอบที่ใช้ในการคำนวณการส่องสว่างและให้แสงเงามีดังนี้

2.4.1 เกรเดียนต์ (Gradient)

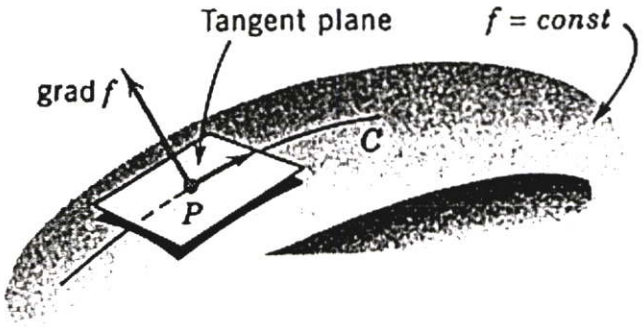
ในคอมพิวเตอร์กราฟิก การให้แสงเงาจำเป็นที่จะต้องรู้ค่าเวกเตอร์ปกติ (Normal Vector) ของพื้นผิวที่แสงส่องไปถึง การหาค่าเวกเตอร์ปกติสามารถทำได้โดยหาค่าเกรเดียนต์ของพื้นผิว ซึ่งเกรเดียนต์เป็นตัวดำเนินการ (Operator) ทางคณิตศาสตร์ชนิดหนึ่ง มีรูปแบบดังสมการ 2.19 โดยที่ $f(x, y, z)$ เป็นสเกลลาฟังก์ชันและสามารถหาอนุพันธ์ได้ (Differentiable)

$$grad\ f = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} + \frac{\partial f}{\partial z} \tag{2.19}$$

เกรเดียนต์จะบอกถึงอัตราเร็วของการเปลี่ยนแปลงของฟังก์ชัน $f(x, y, z)$ ในกรณีที่ฟังก์ชันเป็นพื้นผิวซึ่ง

$$f(x, y, z) = c = \text{ค่าคงที่} \tag{2.20}$$

จากดังรูปที่ 2.8 ถ้าเกรเดียนต์ของ f ที่จุด P ไม่เป็นศูนย์แล้ว ค่าเกรเดียนต์นี้จะเป็นค่าเวกเตอร์ปกติของพื้นผิวที่จุด P นั้นๆ ถ้าเกรเดียนต์เป็นศูนย์จะหมายความว่าไม่มีความเปลี่ยนแปลงในพื้นผิวนั้นๆ



รูปที่ 2.8 แสดงการหาค่าเวกเตอร์ปกติจากเกรเดียนต์ของพื้นผิว

ในกรณีของข้อมูลเชิงปริมาตร เกรเดียนต์จะบอกถึงอัตราเร็วในการเปลี่ยนแปลงของข้อมูลในปริมาตร ถ้าขนาดของเกรเดียนมีค่าเป็นศูนย์หมายความว่าไม่มีการเปลี่ยนแปลงในค่าของจุดภาพ การหาค่าเกรเดียนสามารถทำได้หลายวิธีซึ่งแต่ละวิธีก็จะมีความถูกต้องและความซับซ้อนของการคำนวณแตกต่างกันไป [5]

2.4.1.1 การหาค่าเกรเดียนด้วยตัวประมาณค่าเกรเดียนต์แบบเซ็นทรัลดิฟเฟอเรนซ์ (The Central Difference Gradient Estimator)

การประมาณค่าเกรเดียนต์ด้วยวิธีนี้สามารถทำได้ง่าย และรวดเร็ว แต่ก็จะให้คุณภาพของภาพด้อยกว่าวิธีอื่น เนื่องจากว่าในการคำนวณหาค่าเกรเดียนต์ด้วยวิธีนี้ใช้ข้อมูลในปริมาตรข้อมูลภาพเพียงหกจุดมาทำการคำนวณ ในวิทยานิพนธ์ฉบับนี้เลือกใช้การหาค่าเกรเดียนต์ด้วยวิธีนี้ เนื่องจากสามารถทำได้ง่ายและรวดเร็ว

ตัวประมาณค่าเกรเดียนต์แบบเซ็นทรัลดิฟเฟอเรนซ์มีสมการดังนี้

$$D_x = f(x-1, y, z) - f(x+1, y, z) \quad (2.21)$$

$$D_y = f(x, y-1, z) - f(x, y+1, z) \quad (2.22)$$

$$D_z = f(x, y, z-1) - f(x, y, z+1) \quad (2.23)$$

$f(x, y, z)$ คือค่าความสว่างของจุดภาพที่ตำแหน่ง (x, y, z) ในปริมาตรข้อมูล ส่วน D_x, D_y และ D_z คือองค์ประกอบของเกรเดียนต์ในสามมิติดังแสดงในสมการ 2.24

$$D = [D_x \; D_y \; D_z] \quad (2.24)$$

การหาค่าเกรเดียนต์ด้วยวิธีอื่นสามารถดูรายละเอียดได้จาก [5]

2.4.1.2 การใช้ค่าเกรเดียนต์ในขบวนการสร้างภาพเชิงปริมาตร

ค่าเกรเดียนต์จะถูกใช้ในสองขั้นตอนของการสร้างภาพเชิงปริมาตรคือขบวนการให้แสงเงาและขบวนการแยกแยะชนิดของข้อมูลภาพซึ่งจะถูกกล่าวถึงในบทต่อไป ในคอมพิวเตอร์กราฟิกเวกเตอร์ปกติจะถูกใช้ในขั้นตอนการให้แสงเงาแก่รูปทรงเรขาคณิตซึ่งเป็นพื้นผิวของวัตถุ แต่ในภาพเชิงปริมาตร ข้อมูลภาพจะมีขอบเขตไม่แน่นอนทำให้ไม่สามารถทราบได้ว่าที่ใดในข้อมูลเป็นพื้นผิว ซึ่งทำให้ไม่สามารถหาเวกเตอร์ปกติของพื้นผิวได้ ดังนั้นค่าเกรเดียนต์จึงถูกนำมาใช้แทนเวกเตอร์ปกติในกระบวนการส่องสว่างและให้แสงเงา

2.4.2 แสงแวดล้อม (Ambient Light)

องค์ประกอบนี้เป็นส่วนที่ง่ายที่สุดของการให้แสง แสงแวดล้อมจะมีค่าเท่ากันในทุกทิศทางไม่ขึ้นอยู่กับทิศทางของแหล่งกำเนิดแสง แต่การใช้องค์ประกอบของแสงแวดล้อมอย่างเดียวในการให้แสงจะทำให้ได้ภาพไม่สมจริง เนื่องจากแสงที่เกิดขึ้นที่วัตถุจะมีความสว่างเท่ากันทั้งหมด ตัวอย่างขององค์ประกอบนี้ในธรรมชาติคือแสงจากดวงอาทิตย์เป็นต้น องค์ประกอบของแสงแบบนี้มีรูปแบบดังสมการ 2.25 [4]

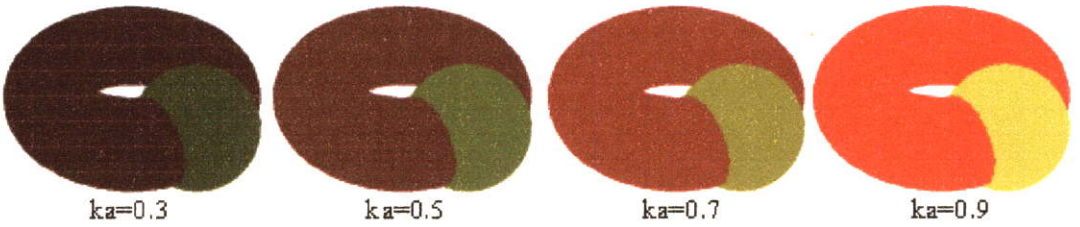
$$I = C_a k_a \quad (2.25)$$

I คือ ความเข้มของแสงที่ผ่านการให้แสงแวดล้อมแล้ว

C_a คือ สีของแสงแวดล้อม

k_a คือ สัมประสิทธิ์การสะท้อนแสงแวดล้อมของวัตถุ (Ambient-Reflection Coefficient)

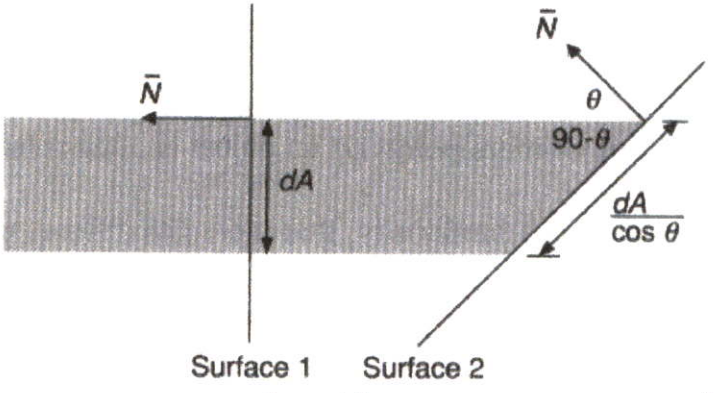
ผลของแสงแวดล้อมที่มีต่อวัตถุ แสดงดังรูปที่ 2.9



รูปที่ 2.9 ผลของแสงแวดล้อมที่มีต่อวัตถุ ซึ่งมีค่า ka ต่างๆกัน

2.4.3 การสะท้อนแบบกระจาย (Diffuse Reflection)

เนื่องจากแสงแวดล้อมเพียงอย่างเดียวจะทำให้ภาพที่ได้ไม่สมจริง การเพิ่มแหล่งกำเนิดแสงแบบจุด (Point Light Source) เข้าไปในระบบจะช่วยแก้ปัญหานี้ได้ แนวลำแสงของแหล่งกำเนิดแสงแบบนี้จะกระจายออกไปทุกทิศทุกทางอย่างสม่ำเสมอ พื้นผิวที่ตั้งฉากกับแหล่งกำเนิดแสงจะได้รับพลังงานแสงในปริมาณมากที่สุด ส่วนอื่นๆที่ไม่ตั้งฉากกับแหล่งกำเนิดแสงจะได้รับพลังงานที่น้อยกว่า พิจารณารูปที่ 2.10 แสงในปริมาณเท่าๆกันตกกระทบลงบนพื้นผิวสองพื้นผิว พื้นผิวแรก (Surface 1) วางตัวในแนวตั้งฉากกับลำแสง ส่วนพื้นผิวที่สอง (Surface 2) วางตัวทำมุม $90-\theta$ กับแนวลำแสง



รูปที่ 2.10 แสดงการตกกระทบของแสงในพื้นผิวที่ตั้งฉากกับแนวลำแสง เทียบกับพื้นผิวที่ไม่ตั้งฉากกับแนวลำแสง

ถ้าแสงที่ฉายสู่พื้นผิวทั้งสองมีพลังงานเท่ากัน(P) อัตราส่วนของพลังงานของแสงที่ตกกระทบต่อพื้นผิวแรกคือ $\frac{P}{dA}$ ส่วนพื้นผิวที่สองคือ $\frac{P.\cos\theta}{dA}$ และในกรณีนี้ θ มีค่าตั้งแต่ $0^\circ \le \theta < 90^\circ$ ซึ่งมีค่ามากที่สุดเท่ากับหนึ่ง ดังนั้นพลังงานของแสงต่อหนึ่งหน่วยพื้นที่ของพื้นผิวที่หนึ่งจะมีค่ามากกว่าพื้นผิวที่สองเสมอ ซึ่งแสดงว่าแสงจะสว่างที่สุดในพื้นผิวที่ตั้งฉากกับแนวลำแสง

สมการของการสะท้อนแบบกระจายคือ

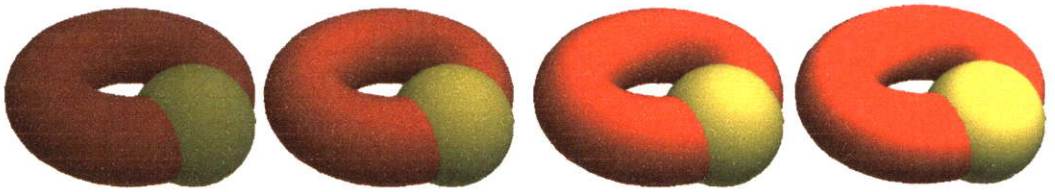
$$I = C_p k_d \cos\theta \qquad (2.26)$$

I คือ ความเข้มของแสงสะท้อนแบบกระจาย

C_p คือ สีของแหล่งกำเนิดแสง

k_d คือ สัมประสิทธิ์การสะท้อนแสงแบบกระจายของวัตถุ (Diffuse-Reflection Coefficient)

ผลของการสะท้อนแสงแบบกระจายร่วมกับผลของแสงแวดล้อมที่มีต่อวัตถุ แสดงดังรูปที่ 2.11

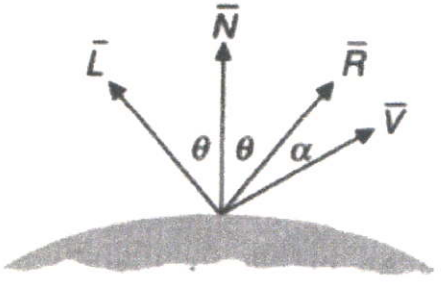


k_d=0.3 k_d=0.5 k_d=0.7 k_d=0.9

รูปที่ 2.11 ผลของการสะท้อนแสงแบบกระจายร่วมกับผลของแสงแวดล้อมที่มีต่อวัตถุที่มีค่า kd ต่างๆกัน และ แสงแวดล้อมมีค่า ka=0.3

2.4.4 การสะท้อนแบบกระจก (Specular Reflection)

การสะท้อนแบบนี้พบได้ในวัตถุที่เป็นมันวาวเช่น โลหะ กระจก หรือวัตถุที่มีผิวมัน สีของแสงที่สะท้อนออกมาจะเป็นจุดสีสว่าง ไม่ใช่สีของวัตถุเหมือนในกรณีของการสะท้อนแบบกระจาย สำหรับวัตถุที่สามารถสะท้อนแสงได้อย่างสมบูรณ์ เช่นกระจกเงา ค่าสัมประสิทธิ์การสะท้อนแสงของวัตถุ (n) ควรจะเข้าใกล้อนันต์ องค์ประกอบการสะท้อนแบบกระจกแสดงดังรูปที่ 2.12



รูปที่ 2.12 องค์ประกอบการสะท้อนแบบกระจก

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

สมการของการสะท้อนแบบกระจกคือ

$$I = C_p k_s O_s (\bar{R}.\bar{V})^n \tag{2.27}$$

I คือความสว่างของแสงที่เกิดจากการสะท้อนแบบกระจก

C_p คือสีของแหล่งกำเนิดแสง

k_s คือสัมประสิทธิ์การสะท้อนแสงแบบกระจก (Specular-Reflection Coefficient)

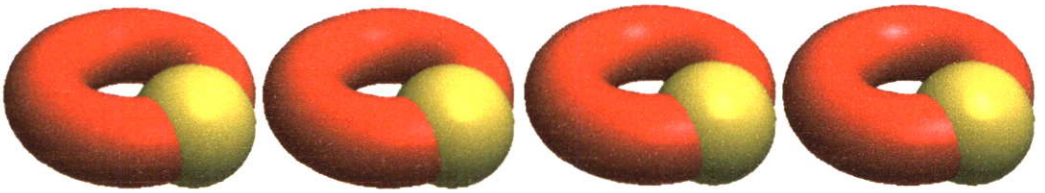
O_s คือสีของแสงสะท้อน

$\bar{V}$ คือเวกเตอร์ที่ชี้จากพื้นผิวไปที่ผู้สังเกต

$\bar{R}$ คือเวกเตอร์ปกติของการสะท้อน

n คือสัมประสิทธิ์การสะท้อนแสงของวัตถุ

ผลของการสะท้อนแสงแบบกระจกร่วมกับผลของการสะท้อนแบบกระจาย และแสงแวดล้อมที่มีต่อวัตถุ แสดงดังรูปที่ 2.13



k_s=0.3 k_s=0.5 k_s=0.7 k_s=0.9

รูปที่ 2.13 ผลของการสะท้อนแสงแบบกระจกร่วมกับผลของการสะท้อนแบบกระจาย และแสงแวดล้อมที่มีต่อวัตถุ

แบบจำลองการส่องสว่างซึ่งประกอบกันขึ้นจากแสงแวดล้อม การสะท้อนแบบกระจาย และการสะท้อนแบบกระจกสามารถเขียนดังสมการ 2.28

$$I = C_a k_a + C_p \left[k_d (\bar{N}.\bar{L}) + k_s O_s (\bar{R}.\bar{V})^n \right] \tag{2.28}$$

I คือความสว่างของแสงซึ่งประกอบกันขึ้นจากแสงแวดล้อม การสะท้อนแบบกระจาย และการสะท้อนแบบกระจก

C_a คือ สีของแสงแวดล้อม

C_p คือสีของแหล่งกำเนิดแสง

O_s คือสีของแสงสะท้อน

k_a คือ สัมประสิทธิ์การสะท้อนแสงแวดล้อมของวัตถุ (Ambient-Reflection Coefficient)

k_d คือ สัมประสิทธิ์การสะท้อนแสงแบบกระจายของวัตถุ (Diffuse-Reflection Coefficient)

k_s คือสัมประสิทธิ์การสะท้อนแสงแบบกระจก (Specular-Reflection Coefficient)

$\overline{N}$ คือเวกเตอร์ปกติของพื้นผิว

$\overline{L}$ คือเวกเตอร์ระหว่างพื้นผิวไปสู่แหล่งกำเนิดแสง

$\overline{V}$ คือเวกเตอร์ที่ชี้จากพื้นผิวไปที่ผู้สังเกต

$\overline{R}$ คือเวกเตอร์ปกติของการสะท้อน

n คือสัมประสิทธิ์การสะท้อนแสงของวัตถุ

โดยสมการนี้จะถูกกระทำกับแต่ละองค์ประกอบ แดง เขียว และน้ำเงินของสีเพื่อจำลองผลของแสดงที่ตกกระทบลงบนวัตถุ

2.5 สรุป

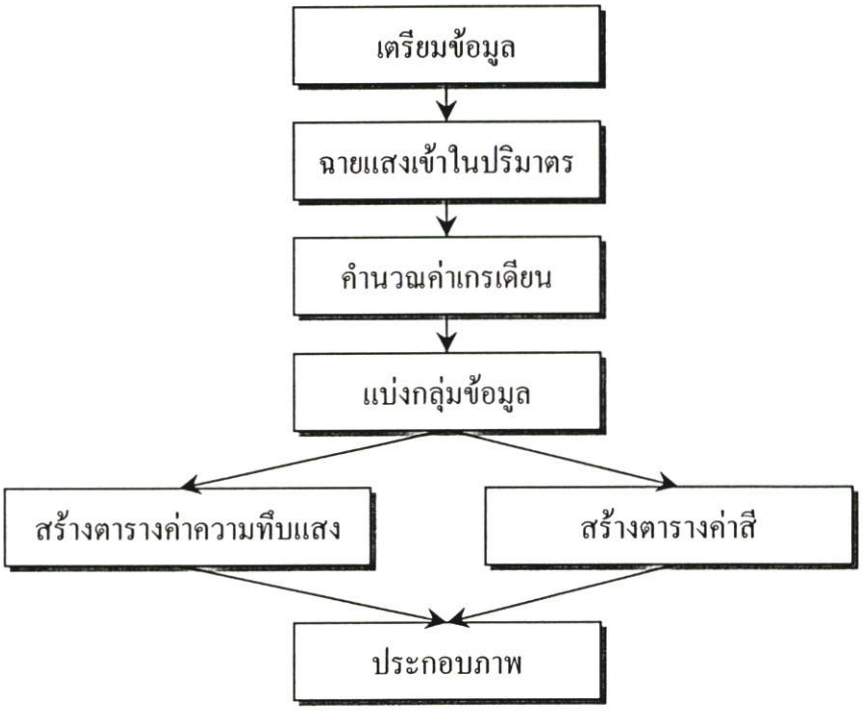
ในบทนี้ได้อธิบายถึงหลักการต่างๆที่จำเป็นต้องใช้ในการสร้างภาพเชิงปริมาตรซึ่งประกอบด้วย ระบบพิกัดและการแปลงเรขาคณิตเพื่อใช้สำหรับเลื่อน หมุน หรือ ย่อและขยายวัตถุให้อยู่ในตำแหน่งที่สามารถเห็นวัตถุที่ต้องการได้ชัดเจน การส่องสว่างและให้แสงเงาจะถูกทำเพื่อให้เห็นวัตถุที่ต้องการ ได้ชัดเจนมากยิ่งขึ้นโดยแบบจำลองการส่องสว่างอาจประกอบด้วยแสงแวดล้อม แสงสะท้อนแบบกระจาย และแสงสะท้อนแบบกระจก ในบทต่อไปจะกล่าวถึงขั้นตอนในการสร้างภาพเชิงปริมาตรอย่างละเอียดซึ่งอาศัยองค์ประกอบที่ได้หาไว้แล้วในบทนี้ช่วยในการสร้างภาพเชิงปริมาตร

บทที่ 3

การสร้างภาพเชิงปริมาตร

3.1 บทนำ

ในบทนี้จะกล่าวถึงขั้นตอนในการสร้างภาพเชิงปริมาตร ซึ่งสามารถแบ่งง่ายๆเป็นห้าขั้นตอน โดยประกอบด้วยการเตรียมข้อมูลภาพ การฉายแสงเข้าในปริมาตร การคำนวณค่าเกรเดียน การแบ่งกลุ่มข้อมูล และการประกอบภาพดังแสดงในรูปที่ 3.1

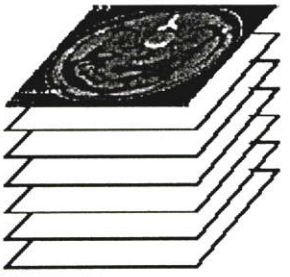


รูปที่ 3.1 ขั้นตอนในการสร้างภาพเชิงปริมาตร

การเตรียมข้อมูลภาพจะถูกทำเพื่อนำภาพจากข้อมูลต้นฉบับมาจัดเรียงเป็นปริมาตร การฉายแสงจะถูกทำเพื่อหาตำแหน่งในปริมาตรข้อมูลภาพเพื่อทำการประมวลผลต่อไป การแบ่งกลุ่มข้อมูลจะทำเพื่อกำหนดค่าสีและความทึบแสงให้กับข้อมูล และการประกอบภาพจะทำเพื่อประกอบค่าสีและค่าความทึบแสงที่ได้เป็นจุดภาพในระนาบการมองเป็นขั้นตอนสุดท้าย

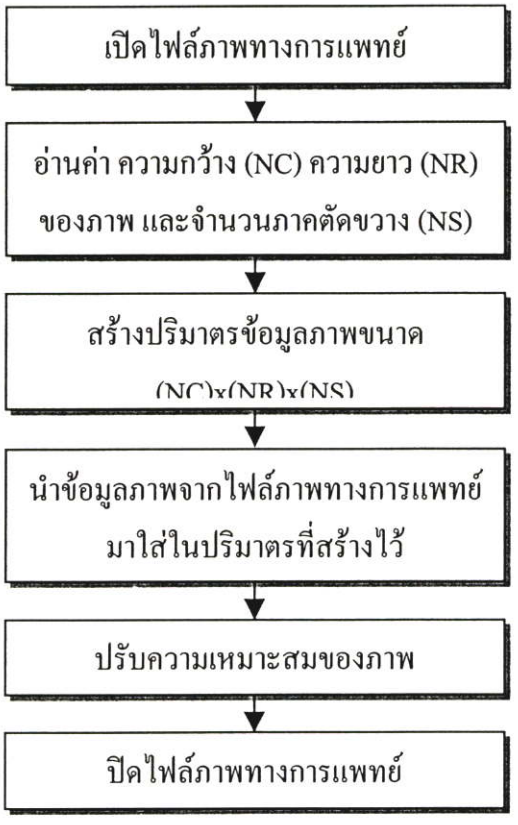
3.2 การเตรียมข้อมูลภาพ

การเตรียมข้อมูลภาพจะถูกทำเป็นขั้นตอนแรกเพื่อนำข้อมูลภาพจากไฟล์ภาพทางการแพทย์มาจัดเรียงเป็นปริมาตรของข้อมูลภาพ (ดังรูปที่ 3.2) เช่นข้อมูลภาพจากไฟล์ภาพแบบอิมมาตรอน (Imatron)[7] และไฟล์ภาพแบบไดคอม (DICOM)[7] เป็นต้นแต่เนื่องจากว่าในวิทยานิพนธ์ฉบับนี้ต้องการข้อมูลภาพทางการแพทย์ที่มีจำนวนข้อมูลมากเป็นพิเศษจึงใช้ภาพจาก The Visible Human Project [6] ซึ่งเป็นไฟล์ที่เก็บแบบเจเปก (JPEG) โดยแต่ละภาคตัดขวางจะมีขนาด 587x341พิกเซล



รูปที่ 3.2 ปริมาตรข้อมูลภาพ

การการเตรียมข้อมูลภาพสามารถทำได้ตามขั้นตอนในรูปที่ 3.3



รูปที่ 3.3 ขั้นตอนการเตรียมข้อมูลภาพ

การนำข้อมูลภาพทางการแพทย์ออกมาใช้อาจทำได้ยากเนื่องจากรูปแบบการเก็บข้อมูลมีความซับซ้อนและไม่เป็นที่เปิดเผยอย่างเป็นทางการเนื่องจากเหตุผลทางการค้า วิธีการเก็บข้อมูลในไฟล์แบบไดคอม และอิมมาตรอนสามารถศึกษาได้จาก Medical Image Format FAQ [7]

3.3 การฉายแสง (Ray Casting)

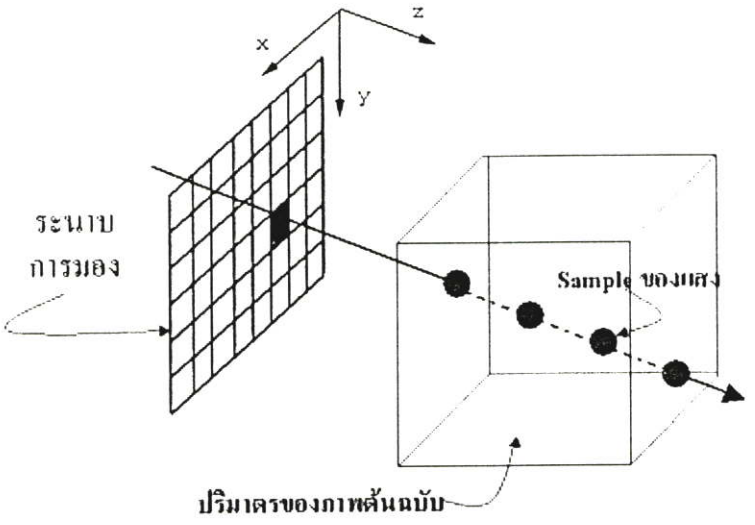
ภาพสามมิติจะปรากฏที่ระนาบการมองซึ่งประกอบไปด้วยจุดภาพ (Pixel) หลายๆจุด จุดภาพเหล่านี้จะถูกให้ค่าสีด้วยกระบวนการฉายแสง ซึ่งในกระบวนการนี้แสงจะถูกฉายจากจุดภาพในระนาบการมองไปสู่ปริมาตรข้อมูลภาพแล้วผ่านการประมวลผลเพื่อหาค่าสีมาแสดงในจุดภาพนั้นๆ ดังแสดงในรูปที่ 3.4 ก่อนที่แสงจะถูกฉายอาจต้องมีการหมุนปริมาตรข้อมูลภาพเพื่อให้เห็นส่วนของวัตถุในมุมมองที่ต้องการ ปกติปริมาตรข้อมูลภาพมักมีขนาดใหญ่ซึ่งอาจประกอบไปด้วยข้อมูลภาพมากกว่าหนึ่งล้านข้อมูล การแปลงตำแหน่งของข้อมูลภาพโดยตรงอาจใช้เวลานานมาก เนื่องจากต้องทำการแปลงตำแหน่งของข้อมูลภาพทุกจุด วิธีการแปลงตำแหน่งแบบนี้ว่าการแปลงตำแหน่งแบบไปหน้า (Forward Transformation) ในทางกลับกันแทนที่จะแปลงตำแหน่งของข้อมูลภาพเพื่อให้เห็นวัตถุในมุมมองที่ต้องการ สามารถแปลงตำแหน่งของระนาบการมองในทิศทางตรงข้ามกับการแปลงตำแหน่งของข้อมูลภาพโดยที่ตำแหน่งของข้อมูลภาพยังคงเดิม เรียกวิธีการแปลงแบบนี้ว่าการแปลงตำแหน่งแบบกลับหลัง (Inverse Transformation) ดังรูปที่ 3.5

ตำแหน่งของจุดภาพในระนาบการมอง (x, y, z) จะถูกเปลี่ยนไปเป็นตำแหน่งที่แท้จริงของข้อมูลภาพในปริมาตร (x', y', z') โดยอาศัยเมตริกการแปลงผกผัน (Inverse Transform Matrix) ดังสมการ 3.1

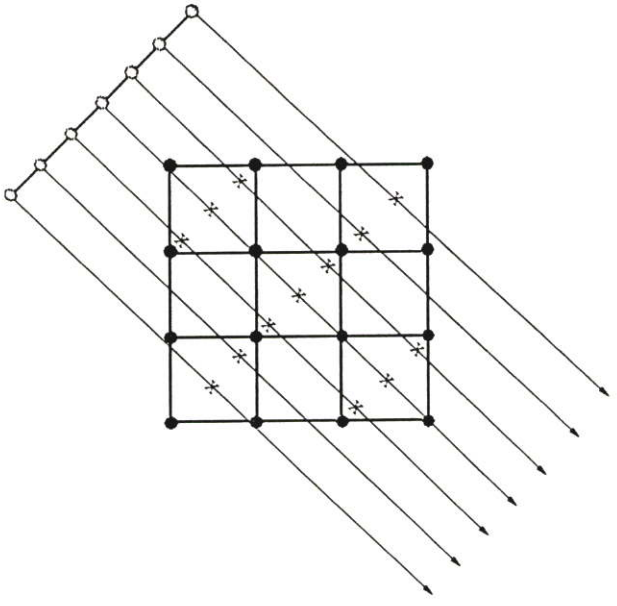
$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = M^{-1} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.1}$$

M^{-1} คือ เมตริกการแปลงผกผัน

จากรูปที่ 3.5 ตำแหน่งของจุดแสง (Sample Point) ที่ถูกเลือกอาจไม่ตรงกับตำแหน่งของข้อมูลภาพ (Voxel) การหาค่าความสว่างของข้อมูลในแนวลำแสงสามารถทำได้จากกระบวนการอินเทอโพเลชั่น(Interpolation) [5] จากค่าของข้อมูลรอบๆข้อมูลที่ถูกเลือกนั้นซึ่งในวิทยานิพนธ์ฉบับนี้ใช้วิธีอินเทอโพเลชั่นแบบเชิงเส้น (Linear Interpolation) ในการหาค่าของจุดแสง



รูปที่ 3.4 การฉายแสงเข้าไปในปริมาตรข้อมูลภาพ



● ข้อมูลภาพ

○ จุดภาพบนระนาบการมอง

x ข้อมูลที่ถูกเลือก

⟶ แนวลำแสง

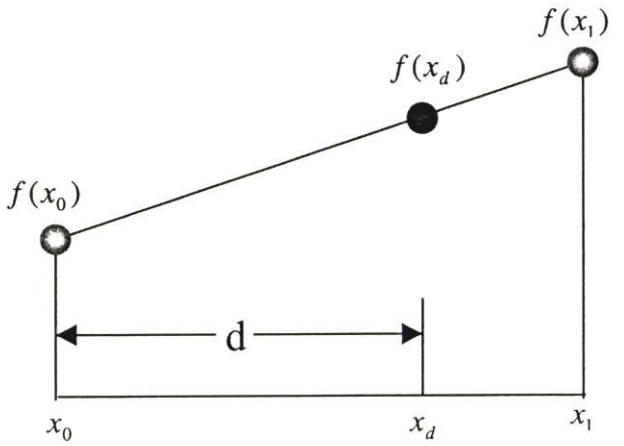
รูปที่ 3.5 การฉายแสงในสองมิติโดยแปลงตำแหน่งของระนาบการมอง

3.3.1 การอินเทอโพเลชั่นแบบเชิงเส้น (Linear Interpolation)

การอินเทอโพเลชั่นแบบเชิงเส้นสามารถทำได้ง่ายและรวดเร็วกว่าการอินเทอโพเลชั่นแบบอื่นๆ โดยการอินเทอโพเลชั่นแบบเชิงเส้นนี้อาศัยสมการเชิงเส้นมาทำการคำนวณค่าที่ต้องการอิน-เทอโพเลทซึ่งแสดงดังสมการ 3.2

$$y = mx + c \tag{3.2}$$

สมมติว่าจุดสองจุด $f(x_0)$ และ $f(x_1)$ มีความสัมพันธ์แบบเชิงเส้น ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 ความสัมพันธ์แบบเชิงเส้นของข้อมูล

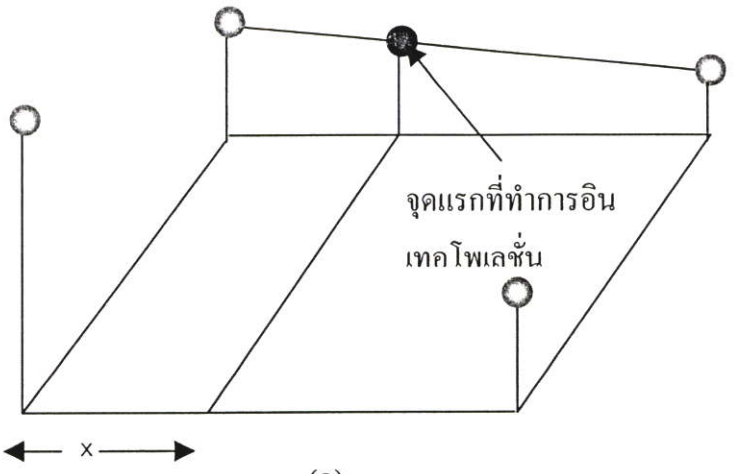
จากรูปที่ 3.6 สามารถเขียนสมการความสัมพันธ์แบบเชิงเส้นของ ข้อมูล $f(x_0)$ และ $f(x_1)$ ได้ดังสมการ 3.3

$$f(x_1) = \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) x_1 + f(x_0) \tag{3.3}$$

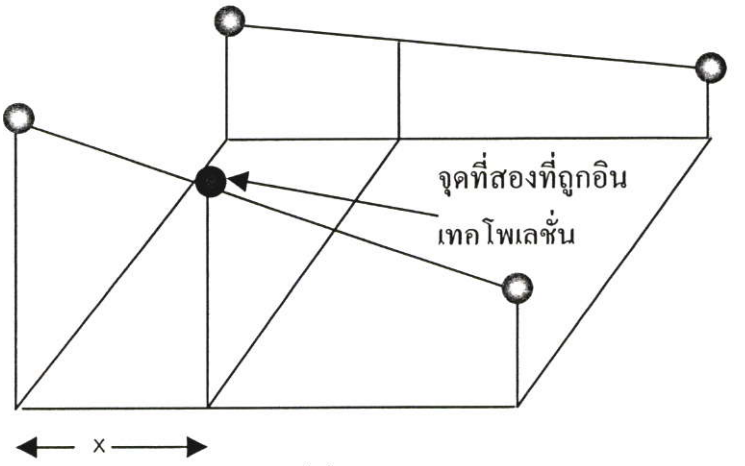
ค่าของข้อมูลที่ตำแหน่ง x_d สามารถหาได้จากสมการ 3.4

$$f(x_d) = \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) (x_d - x_0) + f(x_0) \tag{3.4}$$

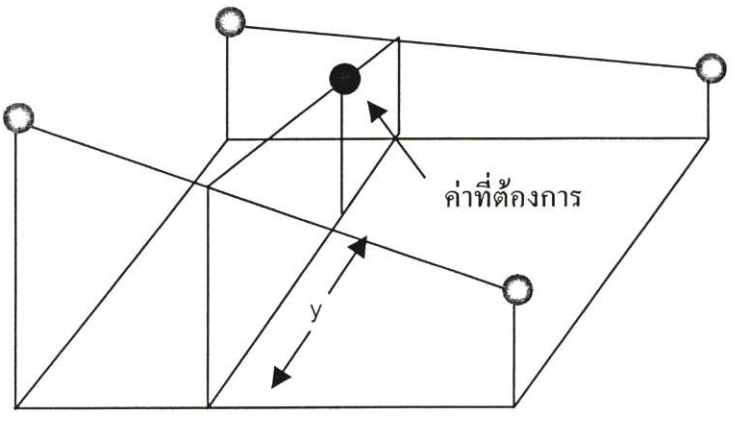
สำหรับข้อมูลในสามมิติ การอินเทอโพเลชั่นจะต้องทำสามครั้งคือแกน x สองครั้งและในแนวแกน y หนึ่งครั้ง ซึ่งอาจเรียกการอินเทอโพเลชั่นแบบนี้ว่าไตรลิเนียร์อินเทอโพเลชั่น (Trilinear interpolation) ดังแสดงในรูปที่ 3.7



(ก)



(ข)



(ค)

รูปที่ 3.7 การอินเทอโพเลชั่นในสามมิติ

(ก) ค่าที่ถูกอินเทอโพเลชั่นในแนวแกน x เป็นจุดแรก

(ข) ค่าที่ถูกอินเทอโพเลชั่นในแนวแกน x เป็นจุดที่สอง

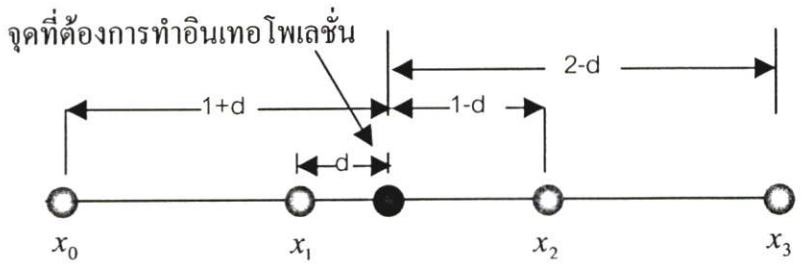
(ค) ค่าที่ถูกอินเทอโพเลชั่นในแนวแกน y

3.3.2 คิวบิกคอนโวลูชั่นอินเทอโพเลชั่น (Cubic Convolation Interpolation)

การอินเทอโพเลชั่นด้วยคิวบิกคอนโวลูชั่นจะให้คุณภาพของภาพที่ดีกว่าการอินเทอโพเลชั่นแบบเชิงเส้น โดยลักษณะของภาพที่ได้จะมีความคมชัดมากกว่า ถ้าพิจารณาในหนึ่งมิติคิวบิกคอนโวลูชั่นจะใช้ค่าสี่ค่าในการคำนวณแต่ในการอินเทอโพเลชั่นแบบเชิงเส้นจะใช้เพียงสองค่า ข้อกำหนดของคิวบิกคอนโวลูชั่นแสดงดังสมการ 3.5 [4]

$$f(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 \le |x| \le 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \le |x| \le 2 \\ 0 & 2 \le |x| \end{cases} \tag{3.5}$$

ปกติค่า a จะอยู่ในช่วง –3 ถึง 0 ถ้าค่า a เข้าใกล้ 0 จะทำให้ได้ภาพที่มีความเบลอมากขึ้น ถ้า a มีค่าเข้าใกล้ –3 จะทำให้ได้รูปที่มีลักษณะคมขึ้น ในกรณีทั่วๆไปใช้ค่า a เท่ากับ -0.5 [5]



รูปที่ 3.8 ระยะทางระหว่างจุดสี่จุดที่ใช้ในการอินเทอโพเลชั่น

จากรูปที่ 3.8 เป็นระยะทางระหว่างจุดสี่จุดที่ใกล้ที่สุดซึ่งใช้ในการอินเทอโพเลชั่น ถ้าหากรู้ระยะทางนี้จะทำให้การคำนวณง่ายลง กำหนดให้สมการ 3.6 เป็นสมการของคิวบิกคอนโวลูชั่น

$$f(d) = c_0.p_0 + c_1.p_1 + c_2.p_2 + c_2.p_3 \tag{3.6}$$

ค่าของจุดภาพที่ตำแหน่ง x_0 x_1 x_2 และ x_3 คือ p_0 p_1 p_2 และ p_3 ตามลำดับ ถ้าหากทราบสมการของคอนโวลูชันฟังก์ชันก็จะสามารถหาค่าสัมประสิทธิ์ของสมการคอนโวลูชัน c_0 c_1 c_2 และ c_3 ได้ดังนี้

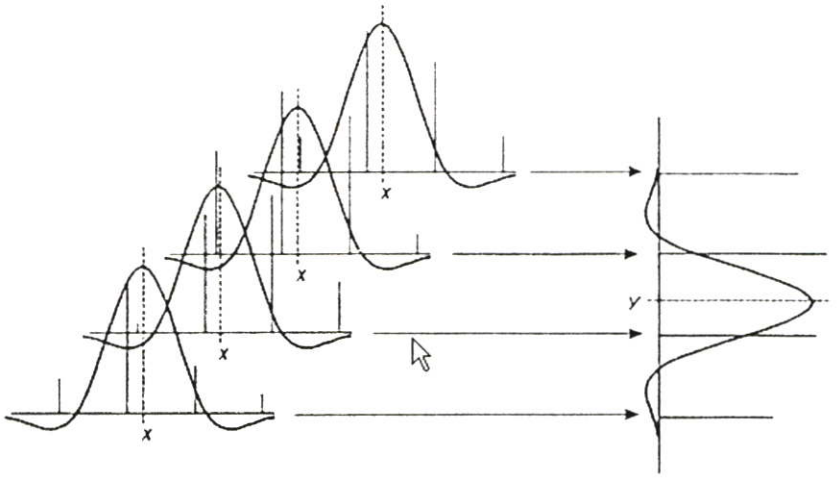
$$c_0 = a(1+d)^3 - 5a(1+d)^2 + 8a(1+d) - 4a = ad^3 - 2ad^2 + ad \tag{3.7}$$

$$c_1 = (a+2)d^3 - (a+3)d^2 + 1 \tag{3.8}$$

$$c_2 = -(a+2)d^3 + (2a+3)d^2 - ad \tag{3.9}$$

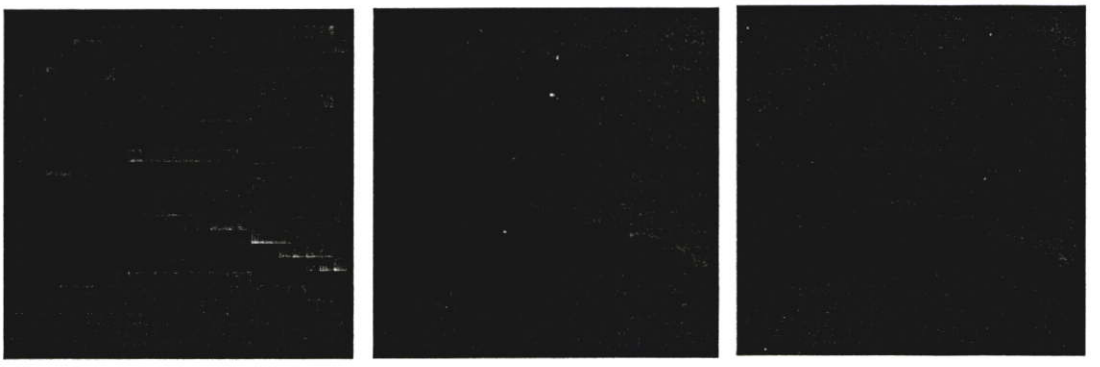
$$c_3 = -ad^3 + ad^2 \tag{3.10}$$

คิวบิกคอนโวลูชั่นในหนึ่งมิติสามารถขยายเป็นสองและสามมิติได้เช่นเดียวกับอินเทอโพเลชั่นแบบเชิงเส้นดังรูปที่ 3.9



รูปที่ 3.9 คิวบิกคอนโวลูชั่นในสองสิติ

ตัวอย่างของรูปซึ่งถูกอินเทอโพเลชั่นแบบต่างๆแสดงดังรูปที่ 3.10



(ก) (ข) (ค)

รูปที่ 3.10 ภาพซึ่งเกิดจากการขยายภาพสิบเท่าโดยใช้อินเทอโพเลชั่นแบบต่างๆ

(ก) อินเทอโพเลชันแบบเพื่อนบ้านที่ใกล้ที่สุด (Nearest Neighbor Interpolation)

(ข) อินเทอโพเลชันแบบเชิงเส้น (Linear Interpolation)

(ค) คิวบิกอินเทอโพเลชัน (Cubic Interpolation)

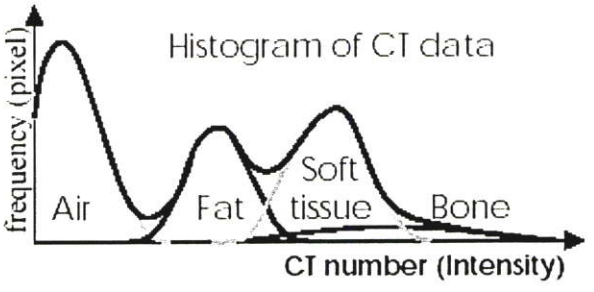
3.4 การแบ่งกลุ่มข้อมูล (Classification)

การแบ่งกลุ่มข้อมูลจะถูกทำเพื่อให้สามารถเห็นโครงสร้างของอวัยวะได้โดยไม่จำเป็นต้องทราบโครงสร้างที่แน่นอนของอวัยวะ โดยโครงสร้างนี้จะมีลักษณะกึ่งโปร่งใสผู้สังเกตสามารถมองโครงสร้างภายในอวัยวะได้เกือบทั้งหมด ซึ่งต่างจากการสร้างภาพเชิงพื้นผิวที่ผู้สังเกตสามารถ

มองเห็นพื้นผิวภายนอกของอวัยวะเท่านั้น การแบ่งกลุ่มข้อมูลในการสร้างภาพเชิงปริมาตรจะทำโดยการกำหนดค่าความทึบแสง (Opacity) ให้กับแต่ละจุดภาพ (Voxel) ในปริมาตรของข้อมูลภาพ โดยค่าความทึบแสงจะบอกถึงความทึบหรือความสามารถในการดูดกลืนแสงของจุดภาพ ความทึบแสงในการสร้างภาพเชิงปริมาตรจะมีค่าอยู่ในช่วง 0 ถึง 1 จุดภาพที่มีค่าเข้าใกล้ 1 จะเป็นจุดภาพที่มีลักษณะทึบแสงส่วนจุดภาพที่มีค่าความทึบแสงใกล้ 0 จะเป็นจุดภาพที่มีความโปร่งแสงมาก โครงสร้างที่ต้องการเห็นจะถูกกำหนดให้มีค่าความทึบแสงมาก ส่วนโครงสร้างที่มีความสำคัญน้อยหรือไม่ต้องการเห็นจะถูกกำหนดความทึบแสงให้มีค่าน้อย การดูว่าข้อมูลที่นำมาสร้างภาพมีวัตถุอยู่ในภาพที่ความสว่างใดบ้างสามารถดูได้จากฮีสโตแกรมของภาพภาคตัดขวางของอวัยวะภายในร่างกาย

3.4.1 ฮีสโตแกรม (Histograms)

ฮีสโตรแกรมของภาพจะบอกถึงความถี่ของความสว่างที่พบในจุดภาพ ค่าของฮีสโต-แกรมจะถูกพล็อตเป็นกราฟสองมิติโดยข้อมูลในแนวแกน x จะเป็นค่าความสว่าง และข้อมูลในแกน y จะเป็นความถี่ของข้อมูล ดังแสดงในรูปที่ 3.11



รูปที่ 3.11 ฮีสโตแกรมของข้อมูลภาพทางการแพทย์ที่ถ่ายด้วยเครื่อง CT

ฮีสโตแกรมในการสร้างภาพเชิงปริมาตรจะช่วยให้สามารถกำหนดฟังก์ชันการถ่ายโอน(Transfer Function)ได้ง่ายขึ้น ข้อมูลที่มีค่าของฮีสโตแกรมที่มีลักษณะเป็นยอดแหลม(Peak) มักจะเป็นวัตถุในข้อมูลภาพเช่นที่ความสว่างน้อยที่สุดมักจะเป็นอากาศ ส่วนที่ความสว่างมากที่สุดมักจะเป็นกระดูก

3.4.2 ฟังก์ชันถ่ายโอน (Transfer Function)

ฟังก์ชันถ่ายโอนมีไว้เพื่อกำหนดค่าความทึบแสงให้กับจุดภาพโดยตัวแปรของฟังก์ชันอาจเป็นค่าความสว่าง หรืออาจเป็นทั้งความสว่างและขนาดของเกรเดียนหรืออาจมีตัวแปรอื่นเข้ามาร่วมด้วยก็ได้ ดังแสดงในสมการ 3.11

$$\alpha_i = O(I_i, |\nabla_i|, ..., ..., ...) \tag{3.11}$$

O คือฟังก์ชันการถ่ายโอนของค่าความทึบแสง

I คือค่าความสว่างของจุดภาพ

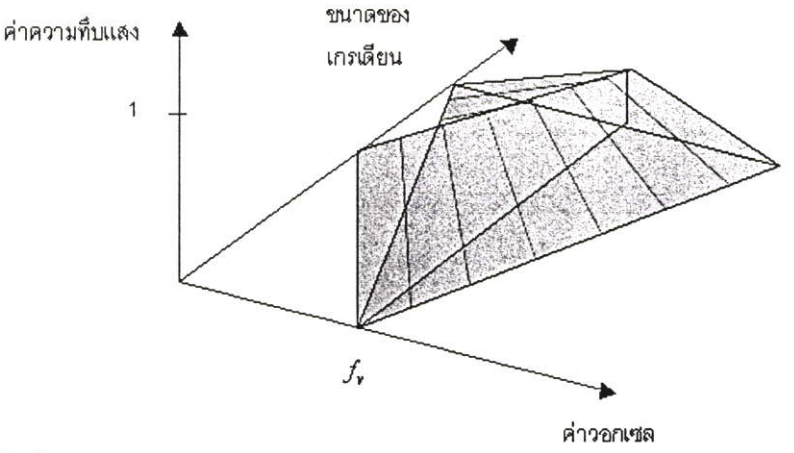
$|\nabla|$ คือขนาดของเกรเดียนของจุดภาพ

ในกรณีของภาพที่ถ่ายจากเครื่อง CT ค่าความสว่างของจุดภาพในแต่ละอวัยวะจะแตกต่างกันอย่างชัดเจน การใช้ค่าความสว่างของจุดภาพเพียงอย่างเดียวเพียงพอในการแยกแยะอวัยวะต่างๆออกจากกัน แต่ในกรณีของภาพที่ถ่ายจากเครื่อง MRI อวัยวะที่ต่างกันอาจมีค่าความสว่างเท่ากันได้ การใช้ค่าความสว่างเพียงอย่างเดียวในการแยกแยะภาพของอวัยวะซึ่งถ่ายจากเครื่อง MRI อาจไม่เพียงพอดังนั้นจึงต้องมีการพิจารณาค่าอื่นๆมาร่วมด้วย เช่นขนาดของเกรเดียนเป็นต้น เกรเดียนจะบอกถึงความเร็วในการเปลี่ยนแปลงของข้อมูล ในบริเวณรอยต่อของอวัยวะสองชนิดจะมีความเร็วในการเปลี่ยนแปลงของค่าความสว่างอย่างคงที่ซึ่งการใช้ตัวดำเนินการเกรเดียน (Gradient Operator) กระทำกับข้อมูลในปริมาตรสามารถใช้หาขอบเขตของอวัยวะได้ Levoy [3] ได้นำเสนอฟังก์ชันการถ่ายโอนซึ่งสามารถใช้ในการสร้างภาพเชิงปริมาตรได้ดีโดยใช้ค่าความสว่างและขนาดของเกรเดียน ฟังก์ชันถ่ายโอนของ Levoy มีลักษณะดังสมการ 3.12

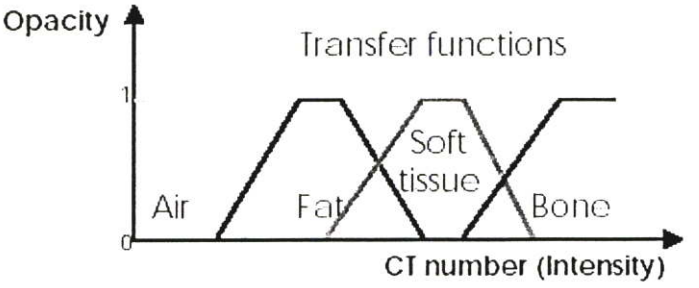
$$
\begin{aligned}
&\alpha_i(r, f_v) = 1 - \frac{1}{r|\nabla_i|}|f_v - I_i| && \text{ถ้า } |\nabla_i| > 0 \text{ และ } I_i - r|\nabla_i| \le f_v \le I_i + r|\nabla_i| \\
&\alpha_i(r, f_v) = 1 && \text{ถ้า} |\nabla_i| = 0 \\
&\alpha_i(r, f_v) = 0 && \text{ค่าอื่นๆ}
\end{aligned}
\qquad (3.12)
$$

ค่าความสว่างของจุดภาพที่ต้องการให้ปรากฏคือ f_v ซึ่งมีค่าความทึบแสงมากที่สุดส่วนค่าความสว่างที่ไม่ใช่ f_v และที่ขนาดของเกรเดียนค่าต่างๆจะมีค่าความทึบแสงลดลงโดยรูปของฟังก์ชันถ่ายโอนจะมีลักษณะคล้ายเต็นท์ (Tent) ดังแสดงในรูปที่ 3.12 ในกรณีที่มีหลายวัตถุอยู่ในข้อมูลและขอบเขตของวัตถุบางอย่างอาจมีการผสมกันเช่นไขมันกับเนื้อเยื่อโดยไม่สามารถกำหนดได้แน่นอนว่าวัตถุเหล่านั้นเป็นอวัยวะประเภทใด การกำหนดค่าของความสว่างที่ใช้ในฟังก์ชันส่งผ่านจะกำหนดเป็นเซ็ทของข้อมูลที่เป็นลักษณะฟัซซี่เซ็ท(Fuzzy Set) [8] ซึ่งแสดงดังรูปที่ 3.13

การจะแยกวัตถุซึ่งขอบเขตบางส่วนผสมกันต้องใช้ฟังก์ชันส่งผ่านที่มีการกำหนดค่า f_v ตามค่าความสว่างของวัตถุซึ่งไม่ผสมกับวัตถุอื่น ในส่วนของวัตถุที่ผสมกันจะกำหนดให้ค่าของความทึบแสงมีค่าน้อยกว่าค่าความทึบแสงของวัตถุนั้นเพียงอย่างเดียว ซึ่งในกรณีของวัตถุที่มีขอบเขตบางส่วนผสมกันนี้ Levoy ได้นำเสนอฟังก์ชันถ่ายโอนดังสมการ 3.13



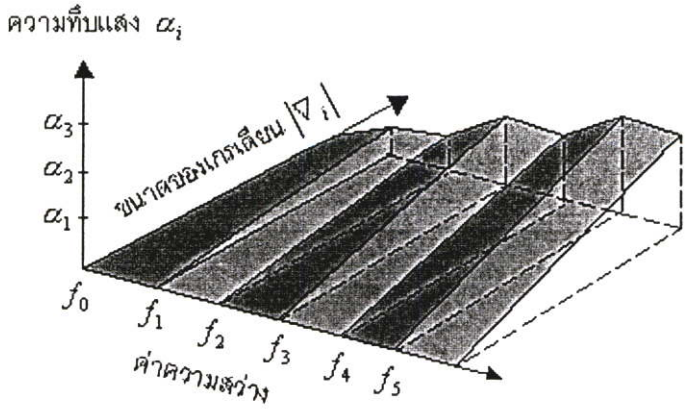
รูปที่ 3.12 ฟังก์ชันถ่ายโอนของ Levoy



รูปที่ 3.13 การแยกแยะชนิดของอวัยวะซึ่งมีขอบเขตผสมกัน

$$\alpha_i = \begin{cases} \dfrac{|\nabla_i|}{f_{n+1}-f_n}[\alpha_n(f_{n+1}-f_i)+\alpha_{n+1}(f_i-f_n)] & ;n=0,2,4,... \\ \alpha_n|\nabla_i| & ;n=1,3,5,... \end{cases} \tag{3.13}$$

ฟังก์ชันถ่ายโอนของสมการ 3.8 จะมีลักษณะดังรูปที่ 3.14



รูปที่ 3.14 ฟังก์ชันถ่ายโอนของข้อมูลที่มีมากกว่าหนึ่งวัตถุ

3.4.3 การให้สี

การให้สีแก่กลุ่มของข้อมูลทำให้สามารถแยกแยะกลุ่มข้อมูลได้ง่ายขึ้นโดยข้อมูลที่อยู่ในกลุ่มที่ต่างกันจะถูกให้สีต่างกัน ส่วนข้อมูลสองกลุ่มที่มีขอบเขตปนกันค่าสีจะของข้อมูลในส่วนนี้จะเป็นการอินเทอร์โพเลชั่นแบบเชิงเส้นของข้อมูลสองกลุ่มนั้น ฟังก์ชันถ่ายโอนของการให้สีจะถูกทำในแต่ละค่าสี แดง เขียว น้ำเงิน ดังสมการ 3.14 3.15 และ 3.16 ตามลำดับ

$$R_i = T_r(I_i, |\nabla_i|, ..) \tag{3.14}$$

$$G_i = T_g(I_i, |\nabla_i|, ..) \tag{3.15}$$

$$B_i = T_b(I_i, |\nabla_i|, ..) \tag{3.16}$$

T_r, T_g, T_b เป็นฟังก์ชันถ่ายโอนของค่าสี แดง เขียว และ น้ำเงินตามลำดับ
ค่าสีของจุดภาพนี้จะถูกนำไปให้แสงเงาอีกครั้งในกระบวนการให้แสงเงาเพื่อให้ได้สีที่สมจริงมากขึ้น

3.5 การประกอบภาพ (Image Compositing)

การประกอบภาพเป็นขั้นตอนสุดท้ายของการสร้างภาพเชิงปริมาตร โดยจะเป็นการรวมคุณสมบัติของจุดแสงที่ได้ถูกคำนวณไว้ก่อนหน้านี้มาสร้างเป็นจุดภาพบนระนาบการมอง การฉายแสงจะสร้างภาพโดยการใช้ข้อมูลทั้งปริมาตรข้อมูลมาทำการคำนวณโดยอาศัยสมมติฐานที่ว่าข้อมูลในปริมาตรมีคุณสมบัติ กระเจิง (Scatter) กีดขวาง (Occlude) สร้าง (Generate) และสะท้อน (Reflect) แสง ภาพในระนาบการมองจะเกิดจากผลของแสงที่ส่องลงบนข้อมูลในปริมาตรรวมกันตลอดหนึ่งแนวลำแสงโดยสามารถเขียนเป็นสมการทางคณิตศาสตร์ได้ดังนี้

$$I(a,b) = \int_a^b g(s) e^{-\int_a^s \tau(x)dx} ds \tag{3.17}$$

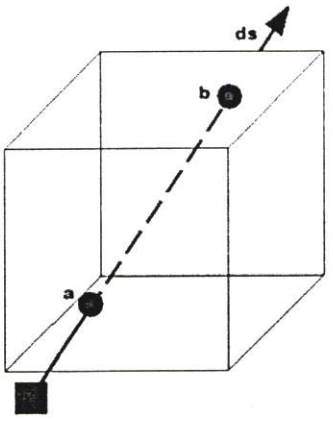
$I(a,b)$ เป็นความเข้มแสงที่ระนาบการมอง

ds เป็นทิศทางของแสงซึ่งเดินทางจากจุด a ไปที่จุด b (ดังรูปที่ 3.15)

g(s) เป็นคุณสมบัติเฉพาะของจุดแสงในแนวลำแสง

$\tau(x)$ เป็นสัมประสิทธิ์ซึ่งบอกถึงความทึบแสงของจุดภาพโดยจะเป็นอัตราส่วนของการลดลงของความสว่างของแสงต่อหนึ่งหน่วยความยาวเนื่องจากการถูกดูดกลืนและการ

กระจายแสง g(s) และ $\tau(x)$ จะถูกใช้เพื่อเปลี่ยนค่าความสว่างของข้อมูลไปเป็นค่าสีและค่าความทึบแสงตามลำดับหรืออาจเรียกตัวปรทั้งสองว่าฟังก์ชันถ่ายโอนซึ่งถูกกล่าวถึงก่อนหน้านี้ในหัวข้อ 3.4.2



รูปที่ 3.15 การเดินทางของแสงจาก a ไป b โดยมีทิศทาง ds

การอินทิเกรทในคอมพิวเตอร์นั้นต้องเป็นลักษณะของการอินทิเกรทแบบไม่ต่อเนื่อง (Discrete Integration) โดยการประมาณค่าของการอินทิเกรทแบบต่อเนื่องในรูปแบบของการอินทิเกรทแบบไม่ต่อเนื่องสามารถทำได้โดยใช้ Riemann sum ดังสมการ 3.18

$$\int_0^d h(x)dx \approx \sum_{i=0}^{n} h(x_i)\Delta x \tag{3.18}$$

ถ้า Δx มีค่าน้อยๆความถูกต้องของการอินทิเกรทก็จะมีมากขึ้น

สมมติว่าค่าความสว่างและค่าความโปร่งแสงของจุดแสงในแนวลำแสงได้ถูกคำนวณไว้แล้ว สมการของการประกอบภาพจะเป็นดังนี้

$$I(a,b) = \sum_{i=0}^{n} I_i \prod_{j=0}^{i-1} T_j \tag{3.19}$$

I_i เป็นความสวางของจุดแสงที่ตำแหน่ง i ในแนวลำแสง

T_j เป็นความโปร่งแสงของจุดแสงที่ตำแหน่ง j ในแนวลำแสงโดยจะมีค่าอยู่ระหว่าง 0 และ 1 ถ้าความโปร่งมีค่ามากหมายความว่าแสงสามารถผ่านจุดแสงนี้ได้มาก

ค่าความโปร่งแสงสามารถเขียนอยู่ในรูปของค่าความทึบแสงได้ดังสมการ 3.20

$$\alpha = 1 - T \tag{3.20}$$

การประกอบภาพสามารถทำได้สองวิธีคือการประกอบภาพแสงแบบหน้าไปหลัง (Front-To-Back Compositing) และการการประกอบภาพแบบหลังไปหน้า (Back-To-Front Compositing) ซึ่งทั้งสองวิธีก็มีข้อดีและข้อเสียดังจะกล่าวในหัวข้อต่อไป

3.5.1 การประกอบภาพแบบหน้าไปหลัง (Front-To-Back Compositing)

ถ้าหากใช้ความทึบแสงแทนความโปร่งแสงสมการ 3.19 สามารถเขียนใหม่ได้เป็น

$$I(a,b) = \sum_{i=0}^{n} I_i \prod_{j=0}^{i-1} (1 - \alpha_j) \tag{3.21}$$

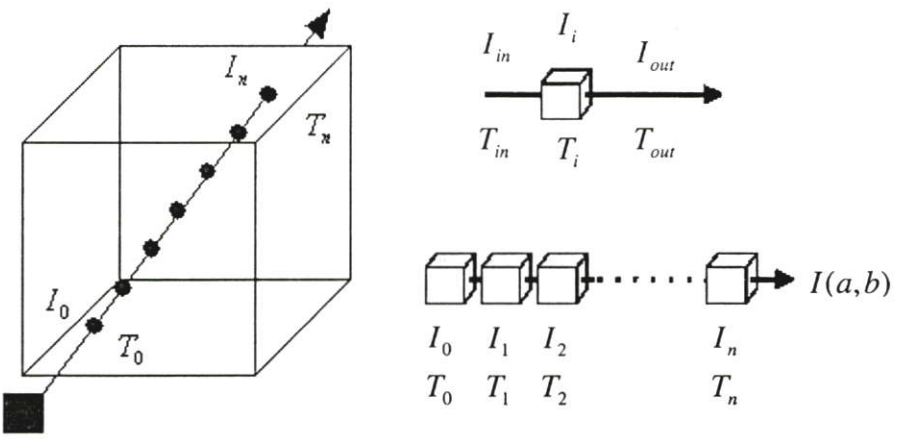
จากสมการ 3.21 ความสว่าง $I(a,b)$ เกิดจากการสะสมของ I_i คูณกับทุกค่าความโปร่งแสง $(1-\alpha_j)$ ของจุดแสงที่มีตำแหน่งน้อยกว่า I ค่า I_i เป็นคุณสมบัติเฉพาะตัวของจุดแสงโดยเกิดจากค่าสีคูณกับค่าความทึบแสงของจุดแสงนั้นๆดังสมการ

$$I_i = C_i * \alpha_i \tag{3.22}$$

ลักษณะและทิศทางของการประกอบภาพแบบหน้าไปหลังแสดงดังรูปที่ 3.16

สมการ 3.21 สามารถเขียนแบบกระจายได้ดังนี้

$$\begin{aligned} \sum_{i=0}^{n} I_i \prod_{j=0}^{i-1} (1 - \alpha_j) &= I_0 + I_1(1-\alpha_0) + I_2(1-\alpha_0)(1-\alpha_1) + \\ &\quad \ldots + I_n(1-\alpha_0)\ldots(1-\alpha_{n-1}) \\ &= C_0\alpha_0 + C_1\alpha_1(1-\alpha_0) + C_2\alpha_2(1-\alpha_0)(1-\alpha_1) + \\ &\quad \ldots + C_n\alpha_n(1-\alpha_0)\ldots(1-\alpha_n) \end{aligned} \tag{3.23}$$



รูปที่ 3.16 ลักษณะและทิศทางของการประกอบภาพแบบหน้าไปหลัง

สมการการประกอบภาพแบบหน้าไปหลังที่ตำแหน่งใดๆในแนวลำแสงสามารถเขียนได้เป็น

$$\begin{aligned} T_{out} &= T_{in}T_i \\ I_{out} &= I_{in} + T_{in}I_i \end{aligned} \tag{3.24}$$

I_{out} และ T_{out} เป็นความสว่างและความโปร่งแสงสะสมที่ทะลุผ่านจุดแสงในตำแหน่ง i

I_{in} และ T_{in} เป็นความสว่างและความโปร่งแสงสะสมที่เข้ามาสู่จุดแสงในตำแหน่ง i

I_i และ T_i เป็นความสว่างและความโปร่งแสงในจุดแสงที่กำลังพิจารณา

จากสมการ 3.24 สามารถเขียนการประกอบภาพแบบหน้าไปหลังในรูปแบบของภาษา C ได้ดังนี้

```
Trans = 1.0;
Inten = I[0];

for(i = 1;  i <= n;  i++)
{
        Trans = Trans * T[i-1];
        Inten = Inten + Trans * I[i];
}
```

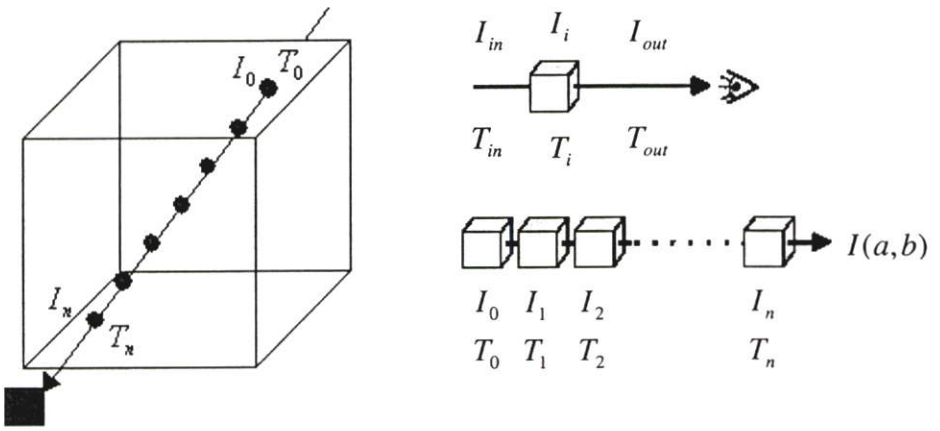
ในการประกอบภาพแบบหน้าไปหลังนี้จะต้องทำการคำนวณค่าของความโปร่งแสงสะสมก่อน จากนั้นนำค่าความโปร่งแสงสะสมไปคำนวณหาค่าความสว่างสะสมอีกครั้งหนึ่ง การที่ต้องคำนวณหาความโปร่งแสงสะสมก่อนเป็นข้อเสียของการประกอบภาพแบบหน้าไปหลังเพราะใช้การคำนวณมากกว่าการประกอบภาพแบบหลังไปหน้าซึ่งใช้การคำนวณหาค่าความสว่างสะสมในขั้นตอนเดียว แต่การประกอบภาพแบบหน้าไปหลังมีข้อดีคือสามารถหยุดการประกอบภาพในแนวลำแสงได้เมื่อค่าความโปร่งแสงสะสมมีค่าเป็นศูนย์หรือเข้าใกล้ศูนย์ ถ้าจุดแสงใดมีค่าของความโปร่งแสงเป็นศูนย์จะหมายความว่าจุดแสงนั้นมีความทึบแสงสูงซึ่งตามหลักของความเป็นจริงแล้วจุดแสงใดก็ตามที่อยู่หลังจุดแสงนี้จะไม่สามารถมองเห็นได้ โดยจากหลักความจริงนี้ถ้าความโปร่งแสงสะสมมีค่าเป็นศูนย์ก็สามารถถือว่าเสร็จสิ้นการประกอบภาพในแนวลำแสงได้

3.5.2 การประกอบภาพแบบหลังไปหน้า (Back-To-Front Compositing)

ในวิทยานิพนธ์ฉบับนี้เลือกใช้การประกอบภาพแบบหลังไปหน้าเนื่องจากใช้การคำนวณน้อยกว่าการประกอบภาพแบบหน้าไปหลัง [9] สมการของการประกอบภาพแบบหลังไปหน้าคือ

$$I(a,b) = \sum_{i=0}^{n} I_i \prod_{j=i+1}^{n} (1-\alpha_j) \tag{3.25}$$

ลักษณะและทิศทางของการประกอบภาพแบบหลังไปหน้าแสดงดังรูปที่ 3.17



รูปที่ 3.17 ลักษณะและทิศทางของการประกอบภาพแบบหน้าไปหลัง

สมการการประกอบภาพแบบหลังไปหน้าที่ตำแหน่งใดๆในแนวลำแสงสามารถเขียนได้เป็น

$$I_{out} = I_{in}T_i + I_i \tag{3.26}$$

I_{out} และ T_{out} เป็นความสว่างและความโปร่งแสงสะสมที่ทะลุผ่านจุดแสงในตำแหน่ง i
I_{in} และ T_{in} เป็นความสว่างและความโปร่งแสงสะสมที่เข้ามาสู่จุดแสงในตำแหน่ง i
I_i และ T_i เป็นความสว่างและความโปร่งแสงในจุดแสงที่กำลังพิจารณา

จากสมการ 3.26 สามารถเขียนการประกอบภาพแบบหลังไปหน้าในรูปแบบของภาษา C ได้ดังนี้

```
Inten = I[0];
for(i = 1; i <= n; i++)
{
        Inten = Inten * T[i] + I[i];
}
```

3.6 สรุป

ในบทนี้ได้กล่าวถึงหลักการสร้างภาพเชิงปริมาตรซึ่งประกอบด้วยการเตรียมข้อมูลภาพ การฉายแสงเข้าสู่ปริมาตรข้อมูลภาพ การหาค่าเกรเดียนต์ การแบ่งกลุ่มข้อมูล และการประกอบภาพ ซึ่งทั้งหมดเป็นกระบวนการที่ต้องทำเพื่อใช้ในการสร้างภาพเชิงปริมาตร ในบทต่อไปจะเป็นการกล่าวถึงโครงสร้างของระบบคลัสเตอร์ซึ่งเป็นระบบคอมพิวเตอร์แบบขนานสมัยใหม่ซึ่งสามารถสร้างได้ง่ายและมีประสิทธิภาพสูง และสามารถประยุกต์ใช้กับงานได้หลายรูปแบบ

บทที่ 4

การประมวลผลแบบขนานและโครงสร้างของระบบคลัสเตอร์

4.1 บทนำ

แนวความคิดของระบบคลัสเตอร์เกิดขึ้นในปี ค.ศ. 1960 โดยบริษัท IBM ต้องการเชื่อมต่อระบบเมนเฟรม (Mainframe) ขนาดใหญ่เข้าด้วยกันเพื่อใช้ในเชิงการค้า [10] แต่ในตอนนั้นระบบคลัสเตอร์ยังไม่เป็นที่รู้จักอย่างแพร่หลายเนื่องจากข้อจำกัดทางด้านราคาและความแพร่หลายของฮาร์ดแวร์ (Hardware) นอกจากนั้นยังขาดโปรแกรม (Tools) มาตรฐานสำหรับการเชื่อมต่อเครื่องในระบบเข้าด้วยกัน จนในช่วงกลางปี 1980 เป็นช่วงที่เทคโนโลยีระบบคลัสเตอร์สามารถเกิดขึ้นได้เนื่องจากสามารถสร้างไมโครโพรเซสเซอร์ที่มีประสิทธิภาพได้ และเกิดระบบเครือข่ายความเร็วสูงขึ้น ซึ่งเครื่องคอมพิวเตอร์ในระบบเครือข่ายสามารถสื่อสารกันได้ภายในเวลาไม่เกินหนึ่งมิลลิวินาที นอกจากนั้นความต้องการความเร็วในการประมวลผลงานทางวิทยาศาสตร์และวิศวกรรมศาสตร์ซึ่งมีมากขึ้นอย่างต่อเนื่องก็มีส่วนผลักดันให้เกิดเทคโนโลยีคลัสเตอร์ขึ้น

ระบบคลัสเตอร์ได้เกิดขึ้นอย่างเป็นรูปธรรมในปี 1994 เมื่อ Thomas Sterling และ Don Becker จาก CESDIS (The Center of Excellence in Space Data and Information Sciences) ต้องการที่จะวิเคราะห์ข้อมูลจากอวกาศที่มีจำนวนมากและซับซ้อนซึ่งต้องใช้เครื่องคอมพิวเตอร์ที่สามารถประมวลผลได้เร็วมากๆและในขณะนั้นมีเพียงเครื่องซุปเปอร์คอมพิวเตอร์ซึ่งมีราคาแพงมากแต่เนื่องจากงบประมาณที่มีอยู่จำกัดทำให้ Thomas และ Don มีแนวความคิดที่จะใช้ระบบคลัสเตอร์มาทำการประมวล โดยได้สร้างระบบคลัสเตอร์จากเครื่องคอมพิวเตอร์ซึ่งใช้ชิพ Intel DX4 เป็นหน่วยประมวลผลกลาง (CPU) และส่วนประกอบอื่นๆซึ่งสามารถหาได้ง่ายจากท้องตลาดและได้ตั้งชื่อระบบคลัสเตอร์นี้ว่า Beowulf [11]

4.2 การประมวลผลแบบขนาน

การประมวลผลแบบขนานเป็นวิธีการในการแบ่งปัญหาขนาดใหญ่ออกเป็นปัญหาย่อยๆแล้วทำการแก้ปัญหาเหล่านั้นไปพร้อมๆกัน[10] ในปัจจุบันระบบประมวลผลแบบขนานถูกใช้อย่างแพร่หลายทั้งในการคำนวณทางวิทยาศาสตร์และการประยุกต์ใช้กับงานทั่วๆไป จากความต้องการความสามารถในการประมวลผลที่มากขึ้นนี้ทำให้มีความต้องการระบบคอมพิวเตอร์ที่มีความเร็วสูงขึ้น ราคาถูกลง และมีการพัฒนาอย่างต่อเนื่อง

การพัฒนาระบบคอมพิวเตอร์แบบขนานถูกแบ่งออกเป็นสองกลุ่มคือ

1. ระบบคอมพิวเตอร์ขนานแบบหลายหน่วยประมวลผล (Massively Parallel Processors (MPPs)) ซึ่งเป็นระบบที่มีความสามารถสูงที่สุดในปัจจุบันโดยระบบคอมพิวเตอร์นี้อาจประกอบด้วยหน่วยประมวลผลกลางตั้งแต่หนึ่งร้อยถึงหนึ่งพันหน่วยซึ่งหน่วยประมวลผลกลางเชื่อมต่อกันผ่านระบบบัส (BUS) ปัจจุบันระบบ MPP ถูกใช้ในการแก้ปัญหาที่ต้องการความเร็วในการประมวลผลมากๆหลายอย่าง เช่น การสร้างแบบจำลองสภาพภูมิอากาศของโลก และ การออกแบบยาเป็นต้น

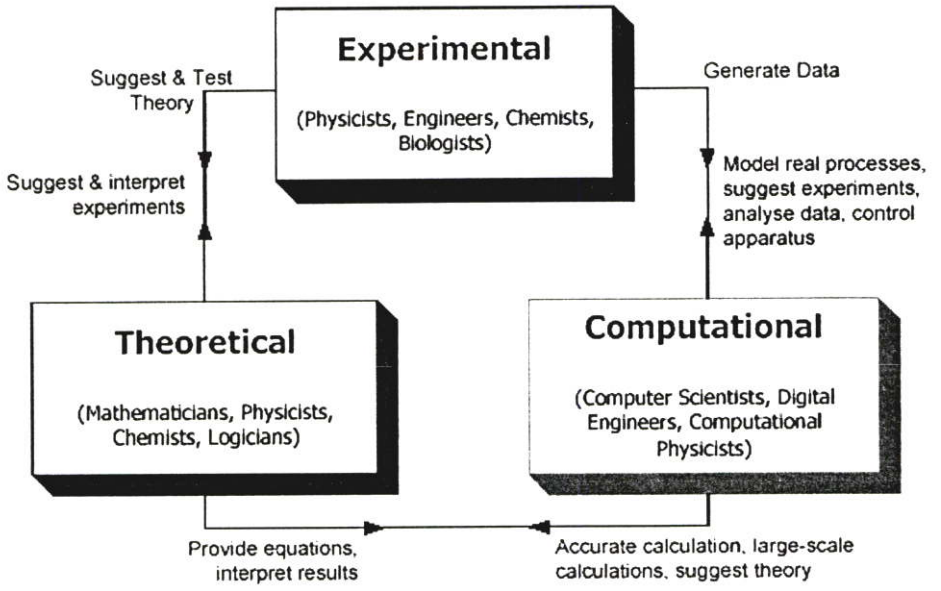
2. ระบบคอมพิวเตอร์แบบกระจาย (Distributed Computing) ซึ่งเป็นกระบวนการที่เครื่องคอมพิวเตอร์หลายๆเครื่องเชื่อมต่อกันผ่านระบบเครือข่ายเพื่อแก้ปัญหาขนาดใหญ่โดยประสิทธิภาพของระบบคอมพิวเตอร์แบบนี้อาจจะเทียบเท่ากับหรือมากกว่าระบบซุปเปอร์คอมพิวเตอร์ขนาดใหญ่ได้ ระบบระบบคอมพิวเตอร์ขนานแบบหลายหน่วยประมวลผลที่ได้กล่าวถึงในตอนแรกสามารถนำมาเชื่อมต่อกันผ่านทางระบบเครือข่ายเพื่อใช้ในลักษณะของระบบระบบคอมพิวเตอร์แบบกระจายเพื่อทำให้มีประสิทธิภาพในการคำนวณมากขึ้น

4.2.1 การประยุกต์ใช้การประมวลผลแบบขนาน

ในอดีตการแก้ปัญหาทางวิทยาศาสตร์มีสาขาวิชาที่สัมพันธ์กันสองสาขาคือวิทยาศาสตร์เชิงทฤษฎี (Theoretical Science) และ วิทยาศาสตร์เชิงการทดลอง (Experimental Science) นักวิทยาศาสตร์เชิงทฤษฎีจะใช้ทฤษฎีทางวิทยาศาสตร์สร้างแบบจำลองทางคณิตศาสตร์และสมมุติฐานของปัญหาจากนั้น นักวิทยาศาสตร์เชิงการทดลองนำแบบจำลองและสมมติฐานของปัญหาไปสร้างการทดลองเพื่อพิสูจน์แบบจำลองและสมมติฐาน ถ้าหากการทดลองเป็นไปตามสมมติฐานและแบบจำลองถูกต้องแสดงว่าสามารถแก้ปัญหานั้นๆได้ แต่ถ้าหากการทดลองล้มเหลวนักวิทยาศาสตร์เชิงทฤษฎีจะนำแบบจำลองและสมมติฐานกลับไปแก้ไขและนำกลับมาทดลองอีกครั้ง ปัญหาในปัจจุบันมีความซับซ้อนมาก งานหลายอย่างไม่สามารถเสี่ยงต่อการล้มเหลวได้เนื่องจากข้อจำกัดทางด้านงบประมาณ เวลา และความปลอดภัย เมื่อระบบคอมพิวเตอร์ถูกพัฒนาขึ้นมาจึงเกิดสาขาวิชาวิทยาศาสตร์เชิงคำนวณ (Computational Science) ขึ้นมาเพื่อช่วยในการแก้ปัญหาทางวิทยาศาสตร์ที่ซับซ้อนให้เป็นไปอย่างมีประสิทธิภาพ ความสัมพันธ์ของวิทยาศาสตร์ทั้งสามสาขานี้แสดงดังรูปที่ 4.1 เมื่อนักวิทยาศาสตร์เชิงทฤษฎีพัฒนาแบบจำลองทางคณิตศาสตร์ของปัญหาขึ้นมาจะส่งแบบจำลองนั้นให้ให้แก่นักวิทยาศาสตร์เชิงคำนวณเพื่อสร้างโปรแกรมเพื่อแก้ปัญหานั้นแบบเชิงตัวเลข (Numerically) ส่วนนักวิทยาศาสตร์เชิงการทดลองจะให้คำแนะนำเกี่ยวกับคุณสมบัติในความเป็นจริงต่างๆที่ใช้ในการแก้ปัญหาแก่นักวิยาศาสตร์เชิงคำนวณเพื่อใช้เป็นข้อมูลในการแก้ปัญหา

ข้อดีของการใช้คอมพิวเตอร์แก้ปัญหาทางวิทยาศาสตร์มีดังนี้

1. ใช้งบประมาณน้อยกว่า และทำได้รวดเร็วกว่าการทดลองในห้องปฏิบัติการ
2. คอมพิวเตอร์สามารถแก้ปัญหาได้หลายเงื่อนไขกว่าการใช้เฉพาะเครื่องมือที่มีอยู่ในห้องปฏิบัติการ
3. ข้อจำกัดของการแก้ปัญหาด้วยการคำนวณมีเพียงแค่ความเร็วและจำนวนหน่วยความจำของคอมพิวเตอร์ แต่การทดลองในห้องปฏิบัติการจะมีข้อจำกัดในทางปฏิบัติมากกว่า



รูปที่ 4.1 ความสัมพันธ์ระหว่างวิทยาศาสตร์เชิงทฤษฎี การทดลอง และเชิงคำนวณ

การประยุกต์ใช้ระบบการประมวลผลแบบขนานสามารถแบ่งออกได้เป็นห้ากลุ่มใหญ่ๆคือ

4.2.1.1 แบบจำลองของการทำนายและการจำลองแบบ (Preditive Modelling and Simulations)

แบบจำลองหลายมิติ (Multidimensional Modeling) ของอากาศ สิ่งแวดล้อมของโลก อวกาศและเศรษฐกิจโลก การสร้างแบบจำลองเพื่อทำนายเหตุการณ์เหล่านี้ถูกทำโดยการทดลองด้วยคอมพิวเตอร์ซิมมูเลชันซึ่งการคำนวณต้องการระบบที่มีความเร็วในการประมวลผลอย่างต่ำ 1 GFLOPS (Giga-floting point operation per second) หรือหนึ่งพันล้านการปฏิบัติงานแบบทศนิยมต่อหนึ่งวินาที

4.2.1.2 การออกแบบทางวิศวกรรมและระบบอัตโนมัติ (Engineering Design and Automation)

ระบบคอมพิวเตอร์แบบขนานถูกใช้ในการวิเคราะห์และออกแบบทางวิศวกรรมเช่น การวิเคราะห์ไฟไนต์อีลีเมนต์ (Finite-Element Analysis) พลศาสตร์เชิงคำนวณ (Computational Aerodynamic) เป็นต้น

4.2.1.3 การค้นหาแหล่งพลังงาน (Energy Resources Exploration)

คอมพิวเตอร์มีบทบาทสำคัญในการค้นหาน้ำมันและก๊าซธรรมชาติทั้งในด้านการขุดหาและการจัดการควบคุมระบบ

4.2.1.4 ด้านการแพทย์ การทหาร และการวิจัยพื้นฐาน (Medical, Military and Basic Research)

ในด้านการแพทย์ คอมพิวเตอร์ความเร็วสูงจะถูกใช้ในการสร้างภาพทางการแพทย์ การออกแบบหัวใจเทียม การวินิจฉัยโรคตับ และการศึกษาทางด้านพันธุวิศวกรรม สำหรับด้านการทหารคอมพิวเตอร์ความเร็วสูงจะถูกใช้ในการออกแบบอาวุธ และการทำสงครามด้วยอุปกรณ์อิเล็กทรอนิกส์

4.2.1.5 การสร้างภาพ (Visualization)

เทคนิคซึ่งใช้ในอุตสาหกรรมหนังเช่น The Matrix และ Toy Story ซึ่งการคำนวณสำหรับสร้างภาพต้องใช้คอมพิวเตอร์ความเร็วสูงสำหรับสร้างภาพเสมือนจริงจำนวนมาก ในหนังทั้งสองเรื่องใช้ระบบคลัสเตอร์ในการประมวลผล นอกจากนั้นการสร้างภาพเสมือนจริงจากข้อมูลทางวิทยาศาสตร์ก็เป็นอีกงานหนึ่งซึ่งต้องใช้คอมพิวเตอร์ความเร็วสูงในการประมวลผลเพื่อสร้างภาพ

4.3 การจำแนกประเภทของคอมพิวเตอร์

คอมพิวเตอร์สามารถแบ่งออกได้เป็นสี่ประเภทโดยจำแนกตามจำนวนของชุดคำสั่งและข้อมูลได้ดังนี้

- SISD : ชุดคำสั่งเดียว ข้อมูลเดี่ยว (Single Instruction Stream / Single Data Stream)
- SIMD: ชุดคำสั่งเดี่ยว หลายข้อมูล (Single Instruction Stream / Multiple Data Stream)

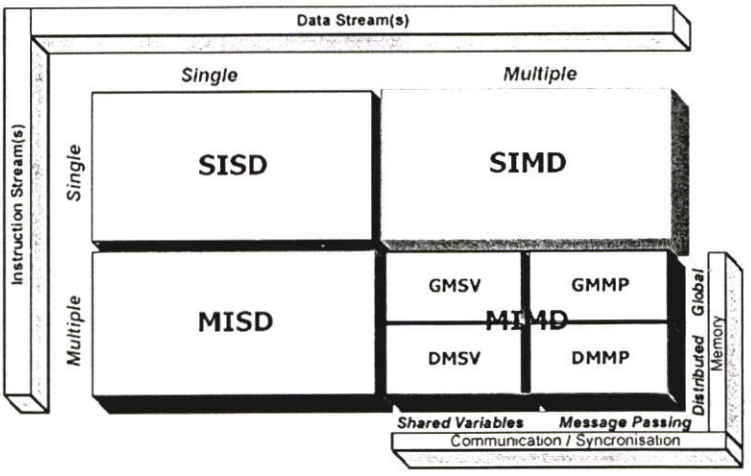
- MISD: หลายชุดคำสั่ง ข้อมูลเดี่ยว (Multiple Instruction Stream / Single Data Stream) ยังไม่มีระบบแบบนี้เกิดขึ้นจริง
- MIMD: หลายชุดคำสั่ง หลายข้อมูล (Multiple Instruction Stream / Multiple Data Stream)

ตัวอย่างของระบบชุดคำสั่งเดี่ยวข้อมูลเดี่ยวคือเครื่องคอมพิวเตอร์ส่วนบุคล (Personal Computer) ซึ่งชุดคำสั่งจะถูกประมวลผลตามลำดับแต่อาจจะมีการซ้อนทับ (Overlap) ของชุดคำสั่งในขั้นตอนประมวลผลได้โดยใช้หลักการของท่อลำเลียง (Pipelining) ระบบคอมพิวเตอร์แบบขนานจะจัดอยู่ในพวกระบบหลายชุดคำสั่งหลายข้อมูลซึ่งอาจเป็นระบบหลายหน่วยประมวลผล (Multiprocessor) และระบบหลายเครื่องคอมพิวเตอร์ (Multiple Computer)

ระบบคอมพิวเตอร์แบบหลายชุดคำสั่งหลายข้อมูลนี้สามารถแบ่งย่อยได้สี่แบบคือ

- GMSV: หน่วยความจำรวม ตัวแปรร่วม(Global Memory / Shared Variables)
- GMMP: หน่วยความจำรวม ส่งข้อความ(Global Memory / Message Passing) ยังไม่มีระบบแบบนี้เกิดขึ้นจริง
- DMSV: หน่วยความจำแบบกระจาย ตัวแปรร่วม(Distributed Memory / Share Variables)
- DMMP: หน่วยความจำแบบกระจาย ส่งข้อความ(Distributed memory / Message Passing)

แผนผังการจำแนกประเภทของระบบคอมพิวเตอร์แสดงดังรูปที่ 4.2



รูปที่ 4.2 แผนผังการจำแนกประเภทของระบบคอมพิวเตอร์

ระบบคอมพิวเตอร์แบบขนานที่ใช้ในวิทยานิพนธ์ฉบับนี้เป็นระบบแบบ DMMP ซึ่งเป็นระบบที่ใช้เครื่องคอมพิวเตอร์มาทำการเชื่อมต่อกันผ่านระบบเครือข่ายความเร็วสูงและทำงานร่วมกันเสมือนเป็นระบบเดียว [1] โดยระบบชนิดนี้เรียกว่าระบบคลัสเตอร์

โดยทั่วๆไปแล้วสามารถจำแนกชนิดของระบบคลัสเตอร์ได้หลายวิธีเช่น [10]

4.3.1 จำแนกตามงานที่ประยุกต์ใช้ (Application Target)

- ระบบคลัสเตอร์แบบประสิทธิภาพสูง (High Performance (HP) Clusters)
- ระบบคลัสเตอร์แบบความคงทนสูง (High Availability (HA) Clusters)

ระบบคลัสเตอร์แบบประสิทธิภาพสูงนิยมใช้สำหรับงานทางวิทยาศาสตร์ซึ่งต้องการความเร็วในการประมาลผลมากๆ ส่วนระบบคลัสเตอร์แบบความคงทนสูงจะนิยมใช้ในงานควบคุมอุปกรณ์ที่ต้องการความแน่นอนสูง หรืองานทางธุรกิจที่ระบบคอมพิวเตอร์ต้องใช้ตลอดเวลาเช่น เครื่องบริการเว็บ (Web Server) เช่นเว็บสำหรับสั่งสินค้าของ Amazon.com ถ้าหากเครื่องบริการเว็บใช้งานไม่ได้ช่วงเวลานานๆเนื่องจากระบบคอมพิวเตอร์เสียก็จะทำให้เสียรายได้และเกิดความเบื่อหน่ายแก่ผู้ที่ต้องการสั่งสินค้าซึ่งอาจเปลี่ยนใจไปใช้บริการของที่อื่น ระบบคลัสเตอร์แบบความคงทนสูงอาจทำงานได้ตลอดทั้งปีโดยที่ระบบไม่มีการปิดทำงาน

4.3.2 จำแนกตามลักษณะของเครื่องในระบบ

ระบบคลัสเตอร์เกิดจากการเชื่อมต่อเครื่องคอมพิวเตอร์หลายๆเครื่องเข้าด้วยกันโดยเครื่องคอมพิวเตอร์เหล่านี้อาจจะมีลักษณะเหมือนกันทั้งหมดหรือไม่เหมือนกันเลยก็ได้ การจำแนกนี้สามารถแบ่งระบบคลัสเตอร์ออกเป็นสองพวกคือ

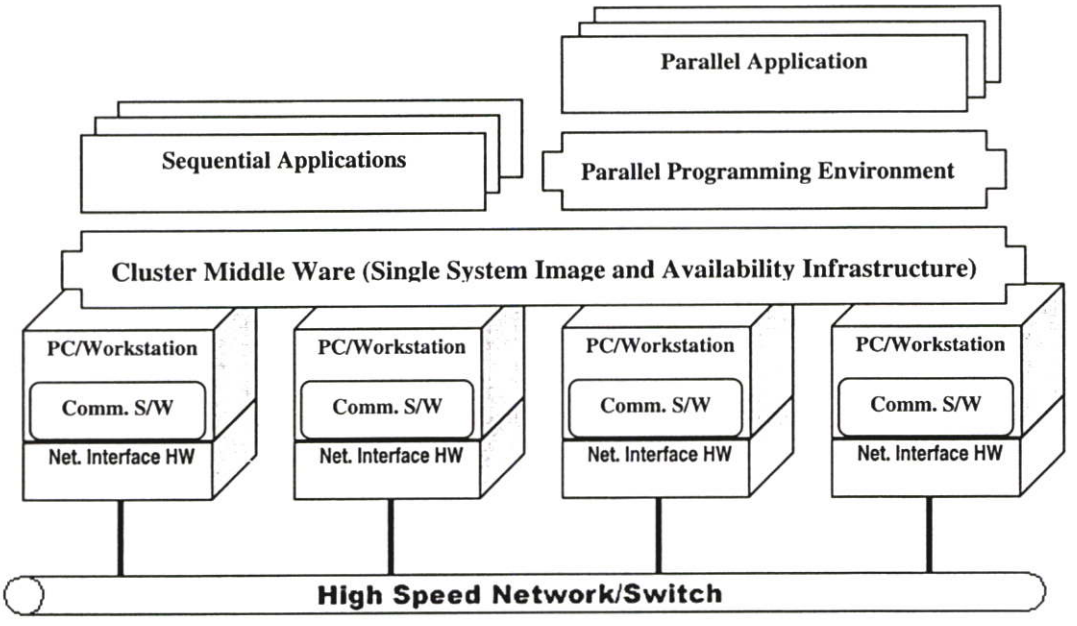
- ระบบคลัสเตอร์แบบเนื้อเดียว (Homogeneous Cluster) ระบบคลัสเตอร์แบบนี้เป็นระบบที่มีองค์ประกอบทางด้านฮาร์ดแวร์และซอร์ฟแวร์ที่ใช้ในแต่ละเครื่องเหมือนกันทั้งหมดได้แก่ หน่วยประมวลผลกลาง ชนิดและขนาดของหน่วยความจำ ชนิดและขนาดของฮาร์ดดิสก์ และชนิดของระบบปฏิบัติการเป็นต้น โดยระบบคลัสเตอร์นี้เป็นแบบที่นิยมสร้างกันเนื่องจากการบริหารจัดการระบบสามารถทำได้อย่างสะดวก นอกจากนั้นการเขียนโปรแกรมเพื่อทำการประมวลผลบนระบบคลัสเตอร์สามารถเขียนในครั้งเดียวแล้วทำงานได้กับทุกๆเครื่องในระบบ

- ระบบคลัสเตอร์แบบเนื้อผสม (Heterogeneous Cluster) ระบบคลัสเตอร์แบบนี้เป็นระบบที่มีความยืดหยุ่นสูงโดยสามารถสร้างจากเครื่องแบบใดก็ได้ที่สนับสนุนการประมวลผลแบบขนาน แต่ปัญหาของระบบนี้คือการสร้างโปรแกรมสำหรับประมวลผลแบบนขนานจะมีความยุ่งยากและซับซ้อนมากขึ้นเนื่องจากต้องทำการสร้างโปรแกรมที่สามารถประมวลผลเฉพาะกับเครื่อง

หนึ่งๆ เช่นเครื่องที่ใช้ระบบปฏิบัติการลีนุกส์ (Linux) และระบบที่ใช้ไมโครซอฟส์วินโดว์ (Microsoft Windows) เป็นระบบปฏิบัติการ โปรแกรมที่สร้างขึ้นมาอาจไม่สามารถใช้ด้วยกันได้ วิธีแก้ปัญหาอย่างหนึ่งคือสร้างโปรแกรมโดยใช้ตัวแปรภาษาที่สนับสนุนมาตรฐานเช่น ANSI C เป็นต้น หรืออาจสร้างโปรแกรมที่สามารถบนประมวลผลได้บนทุกระบบโดยที่ใช้รหัสต้นฉบับ (Source Code) ตัวเดียวกัน เช่นภาษาจาวา (Java) เป็นต้น

4.4 องค์ประกอบและการสร้างระบบคลัสเตอร์

องค์ประกอบของระบบคลัสเตอร์แสดงดังรูปที่ 4.3



รูปที่ 4.3 องค์ประกอบของระบบคลัสเตอร์

รายละเอียดของแต่ละองค์ประกอบมีดังนี้ [10]

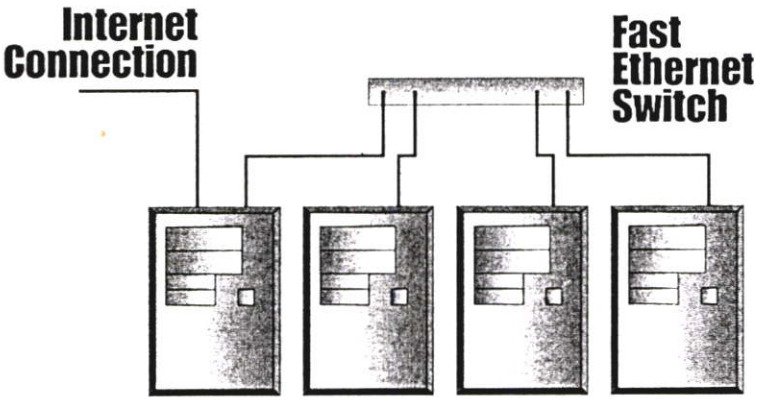
4.4.1 ระบบเครือข่ายความเร็วสูง (High Speed Network)

การสื่อสารระหว่างแต่ละเครื่องในระบบคลัสเตอร์จะทำผ่านระบบเครือข่ายความเร็วสูง ซึ่งอุปกรณ์เครือข่ายแต่ละชนิดจะมีความเร็วและราคาแตกต่างกันไป ตัวอย่างอุปกรณ์เครือข่ายที่สามารถใช้สร้างระบบคลัสเตอร์ได้มีดังนี้

- Ethernet Fast Ethernet และ Gigabit Ethernet ในปัจจุบันอุปกรณ์ Ethernet และ Fast Ethernet สามารถหาซื้อได้ง่ายและราคาถูกในวิทยานิพนธ์ฉบับนี้ใช้ Fast Ethernet ซึ่งมีความเร็วในการส่งผ่านข้อมูลประมาณ 100 ล้านบิตต่อวินาที ร่วมกับ Switching Hub

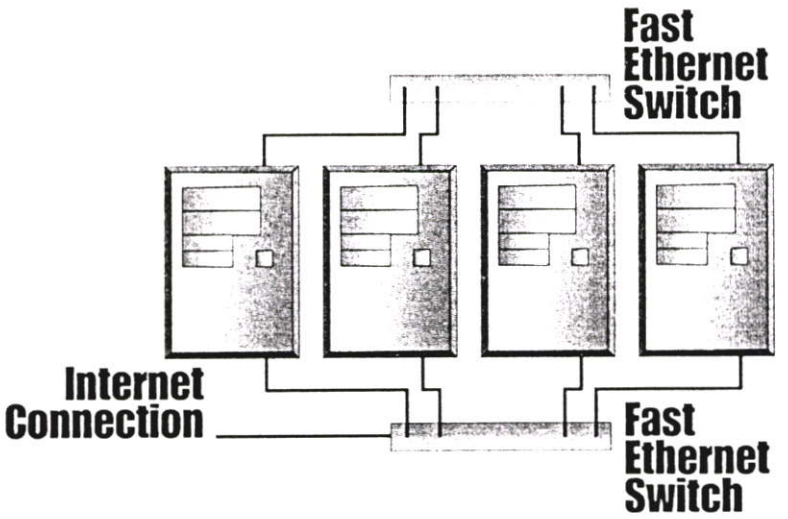
- Myrinet มีความเร็วในการส่งผ่านข้อมูล 1.28 พันล้านบิตต่อวินาที (Gigabit per second) แต่มีราคาแพงกว่า Fast Ethernet มาก ในระบบคลัสเตอร์ที่ต้องการความเร็วในการส่งข้อมูลสูงมากๆมักใช้ระบบเครือข่ายนี้ในการสื่อสารระหว่างเครื่อง

การเชื่อมต่อแต่ละเครื่องเข้ากับระบบเครือข่ายอาจทำได้ดังรูปที่ 4.4 โดยมีเครื่องเดียวที่เชื่อมต่อกับระบบภายนอกซึ่งอาจเป็นระบบอินเตอร์เน็ตก็ได้ ส่วนเครื่องที่เหลือต่อเข้ากับ Hub หรือ Switching ธรรมดา ลักษณะการเชื่อมต่อแบบรูปที่ 4.4 มีข้อดีคือมีความปลอดภัยสูง ผู้ใช้จะต้องผ่านเครื่องที่ต่อกับเครือข่ายภายนอกก่อนถึงจะสามารถเข้าถึงเครื่องที่อยู่ในเครือข่ายภายในได้

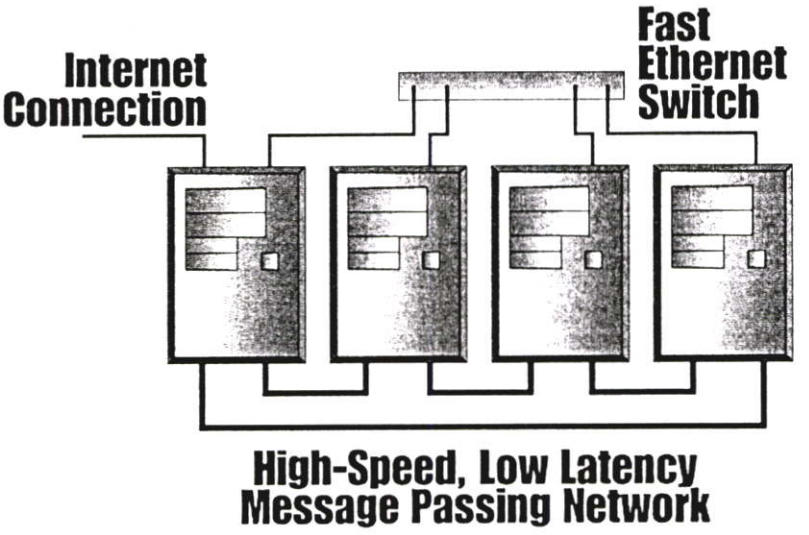


รูปที่ 4.4 การเชื่อมต่อเครื่องในระบบคลัสเตอร์เข้ากับระบบเครือข่ายส่วนตัว (Private Network)

ถึงแม้ว่าระบบในรูปที่ 4.4 จะมีความปลอดภัยสูงแต่ในแง่การประยุกต์ใช้กับงานบางอย่างเช่นงานที่ต้องการเข้าถึงแต่ละเครื่องโดยตรงอาจทำได้ไม่สะดวกเช่นงานเกี่ยวกับศูนย์รวมเซอร์ฟเวอร์ (Server Farm) ในการทำอีเมลล์เซอร์ฟเวอร์ที่มีความคงทนสูง (High Availability Email Server) หรือ เวปเซอร์ฟเวอร์ที่มีความคงทนสูง (High Availability Web Server) สามารถทำการเชื่อมต่อแต่ละเครื่องเข้ากับระบบเครือข่ายได้ดังรูปที่ 4.5 ซึ่งแต่ละเครื่องเชื่อมต่อกันผ่านระบบเครือข่ายในขณะที่ทุกๆเครื่องก็เชื่อมต่อกับระบบภายนอกด้วยเช่นกัน



รูปที่ 4.5 การเชื่อมต่อระบบเครือข่ายแบบศูนย์รวมเซอร์ฟเวอร์



รูปที่ 4.6 ลักษณะการเชื่อมต่อของเครื่องในระบบคลัสเตอร์ที่การสื่อสารกันภายในระบบจะถูกทำผ่านระบบเครือข่ายความเร็วสูง

ระบบคลัสเตอร์ที่ต้องการส่งข้อมูลระหว่างเครื่องด้วยความเร็วสูงอาจต้องมีระบบเครือข่ายพิเศษซึ่งความเร็วสูงกว่าระบบเครือข่ายปกติ ส่วนข้อมูลที่ส่งไปภายนอกระบบจะทำผ่านระบบเครือข่ายปกติซึ่งแสดงดังรูปที่ 4.6

4.4.2 โปรแกรมบริการการสื่อสาร (Communication Software)

เนื่องจากในวิทยานิพนธ์ฉบับนี้ใช้ MPI เป็นโปรแกรมสำหรับสร้างสภาวะแวดล้อมของการโปรแกรมแบบขนาน (Parallel Programming Environment) ซึ่งจะกล่าวถึงในหัวข้อที่ 4.4.5 โดย

MPI [12] สนับสนุนโปรแกรมบริการการสื่อสารสองตัวคือ RSH (Remote Shell) และ SSH (Secure Shell) โปรแกรมทั้งสองตัวนี้ทำหน้าที่จัดการไฟล์และประมวลผลโปรแกรมระหว่างเครื่องได้ Remote Shell จะถูกติดตั้งมาให้ในระบบปฏิบัติการแบบยูนิกส์ (UNIX) ส่วนมาก แต่มีข้อบกพร่องในเรื่องระบบรักษาความปลอดภัย ในระบบปฏิบัติการแบบยูนิกส์ยุกต์ใหม่มักไม่แนะนำให้ใช้ Remote Shell เช่นในระบบปฏิบัติการลีนุกส์ (LINUX) ส่วนมากจะมี Secure Shell ติดตั้งมาให้ซึ่งมีความปลอดภัยมากกว่า Remote Shell เนื่องจากมีการเข้ารหัสข้อมูลก่อนทำการส่งข้อมูล ถ้าระบบคลัสเตอร์ไม่ได้เชื่อมต่อกับระบบอินเตอร์เน็ตก็เป็นไปได้ว่า Remote Shell สามารถใช้ได้อย่างปลอดภัย

4.4.3 ระบบปฏิบัติการ (Operating System)

ระบบปฏิบัติการที่สามารถใช้ระบบเครือข่ายได้ส่วนมากสามารถนำมาสร้างเป็นระบบคลัสเตอร์ได้ เช่นไมโครซอร์ฟวินโดว์ เอ็นที (Microsoft Windows NT) ไมโครซอร์ฟวินโดว์สองพัน (Microsoft Windows 2000) ลีนุกส์ (Linux) ซันโซลาริส (Sun Solaris) เป็นต้น ในงานวิจัยนี้แนะนำให้ใช้ระบบคลัสเตอร์ที่มีระบบปฏิบัติการเป็นลีนุกส์เนื่องจาก

- เป็นระบบปฏิบัติการที่แจกให้ใช้ฟรี
- สามารถทำเป็นระบบคลัสเตอร์ได้จากซอฟส์แวร์และไลบรารี่ที่มีแจกให้ใช้ฟรีบนระบบอินเทอร์เน็ตโดยที่ไม่ต้องเสี่ยค่าใช้จ่ายในส่วนของซอร์ฟแวร์เพิ่มเติมในการสร้างระบบคลัสเตอร์
- มีประสิทธิภาพและความแน่นอนสูง
- มีการพัฒนาอย่างต่อเนื่อง
- มีความยืดหยุ่นในการปรับเปลี่ยนระบบสูง
- โปรแกรมที่พัฒนาบนระบบลีนุกส์สามารถย้ายไปทำงานบนระบบยูนิกส์ตระกูลอื่นได้ง่าย
- ระบบคลัสเตอร์ส่วนมากใช้ลีนุกส์เป็นระบบปฏิบัติการ ซึ่งเมื่อมีปัญหาสามารถขอคำปรึกษาจากผู้ดูแลระบบเหล่านั้นได้
- โปรแกรมที่เกี่ยวกันระบบคลัสเตอร์ส่วนมากพัฒนาบนระบบลีนุกส์
- สามารถแก้ไขแก่น (Kernel) ของระบบปฏิบัติการเพื่อปรับปรุงโครงสร้างของระบบปฏิบัติการได้ และสามารถพัฒนาระบบคลัสเตอร์โดยเพิ่มระบบการทำงานเข้าไปในแก่นของระบบปฏิบัติการโดยตรงได้

ระบบปฏิบัติการอื่นๆที่เป็นระบบยูนิกส์สามารถใช้ร่วมกับระบบลีนุกส์เพื่อทำเป็นระบบคลัสเตอร์ได้ง่าย เนื่องจากมีหลักการออกแบบและโครงสร้างของระบบปฏิบัติการใกล้เคียงกัน

4.4.4 คลัสเตอร์มิดเดิ้ลแวร์ (Cluster Middle Ware)

คลัสเตอร์มิดเดิ้ลแวร์เป็นโปรแกรมที่ทำให้ผู้ใช้ระบบคลัสเตอร์รู้สึกว่ากำลังใช้ระบบเดี่ยวเช่นเดียวกับที่ใช้เครื่องคอมพิวเตอร์ส่วนบุคคลอยู่ โปรแกรมส่วนคลัสเตอร์มิดเดิ้ลแวร์นี้จะต้องทำที่ระดับแก่น (Kernel) ของระบบปฏิบัติการเพื่อที่จะสามารถจัดการถ่ายโอนโพรเซส (Process) ที่จะเกิดขึ้นไปสู่เครื่องอื่นๆในระบบได้ ตัวอย่างของคลัสเตอร์มิดเดิ้ลแวร์นี้คือ

- BPROC (Beowulf-distributed process space) ซึ่งเป็นโปรแกรมส่วนที่เพิ่มเข้าไปในแก่นของระบบปฏิบัติการลีนุกส์ โดยพัฒนาจากผู้สร้างระบบ Beowulf ในโครงการ CESDIS ของ NASA [11]
- MOSIX เป็นโปรแกรมส่วนที่เพิ่มเข้าไปในส่วนของแก่นของระบบปฏิบัติการเช่นเดียวกับ BPROC ถูกพัฒนาโดยมหาวิทยาลัย Hebrew University of Jerusalem [13]
- Solaris-MC เป็นระบบปฏิบัติการที่สนับสนุนการกระจายงานในระบบคลัสเตอร์ถูกพัฒนาโดย Sun Labs

เนื่องจากว่าระบบคลัสเตอร์ที่ใช้ในวิทยานิพนธ์ฉบับนี้เป็นระบบคลัสเตอร์ที่สร้างขึ้นเฉพาะงาน การกระจายงานอาจไม่จำเป็นต้องใช้คลัสเตอร์มิดเดิ้ลแวร์ แต่การกระจายงานจะทำในชั้นของสภาวะแวดล้อมของการโปรแกรมแบบขนานเช่น MPI หรือ PVM

4.4.5 โปรแกรมสำหรับสร้างสภาวะแวดล้อมของการโปรแกรมแบบขนาน (Parallel Programming Environment)

โปรแกรมสำหรับสร้างสภาวะแวดล้อมของการโปรแกรมแบบขนานมีหน้าที่ทำให้โปรแกรมแบบขนานที่ประมวลผลผ่านสภาวะแวดล้อมนี้สามารถกระจายงานได้อย่างอัตโนมัติตามที่ผู้สร้างโปรแกรมได้ออกแบบไว้ สภาวะแวดล้อมของการโปรแกรมแบบขนานแบ่งออกเป็นสองส่วนคือ

4.4.5.1 ส่วนของโปรแกรมสร้างสภาวะแวดล้อมแบบขนาน

ส่วนของโปรแกรมสภาวะแวดล้อมนี้ทำหน้าที่ในการเชื่อมต่อระหว่างเครื่องในระบบคลัสเตอร์ เมื่อทำการประมวลผลโปรแกรมแบบขนานที่ออกแบบโดยใช้คลังชุดคำสั่ง (Library) ของการโปรแกรมแบบขนานนี้ สภาวะแวดล้อมแบบขนานจะทำการส่งงานที่ผู้สร้างโปรแกรมกำหนด ไปประมวลผลบนเครื่องที่เลือกไว้โดยอัตโนมัติ หรือถ้าในโปรแกรมที่สร้างขึ้นไม่ได้ระบุ

ชื่อของเครื่องที่ต้องการให้ประมวลผล สภาวะแวดล้อมแบบขนานนี้จะทำการกระจายงานให้โดยอัตโนมัติคล้ายกับหน้าที่ของคลัสเตอร์มิดเดิ้ลแวร์ แต่ส่วนของสภาวะแวดล้อมแบบขนานนี้มีความยืดหยุ่นและประสิทธิภาพมากกว่า แต่ก็สร้างได้ยากกว่าเนื่องจากขึ้นอยู่กับโปรแกรมประยุกต์ที่สร้างขึ้น

4.4.5.2 ส่วนคลังชุดคำสั่ง (Library) ของการโปรแกรมแบบขนาน

โปรแกรมแบบขนานจำเป็นที่ต้องทำการสื่อสารผ่านระบบเครือข่าย ทางเลือกหนึ่งที่สะดวกรวดเร็วและมีประสิทธิภาพคือใช้คลังชุดคำสั่งแบบขนานที่มีผู้ที่พัฒนาไว้แล้ว การกระจายงานอาจประกอบด้วยการกระจายให้เครื่องทุกเครื่องในระบบหรือกระจายสู่บางเครื่อง หรือกระจายสู่กลุ่มของเครื่องที่มีระบบปฏิบัติการเดียวกันหรือกลุ่มของเครื่องที่ได้ออกแบบไว้ก่อน นอกจากนั้นคลังชุดคำสั่งยังได้จัดเตรียมวิธีการส่งข้อมูลชนิดต่างๆเช่น จำนวนเต็ม ทศนิยม หรือแม้แต่ข้อมูลที่มีความสลับซับซ้อนมากๆเช่นข้อมูลในหน่วยความจำที่ไม่อยู่ติดเป็นกลุ่มเดียวเป็นต้น

โปรแกรมในส่วนของสภาวะแวดล้อมของการโปรแกรมแบบขนานที่นิยมใช้ในงานวิจัยมีสองตัวคือ

- Parallel Virtual Machine (PVM) [14] ถูกพัฒนาโดย Oak Ridge National Laboratory มีลักษณะเป็นเครื่องเสมือน (Virtual Machine) เพื่อใช้สำหรับประมวลผลโปรแกรมแบบขนาน เริ่มแรกผู้ใช้ต้องกำการเพิ่มเครื่องในระบบคลัสเตอร์เข้าไปในส่วนที่เป็นเครื่องเสมือนจากนั้นเมื่อทำการประมวลผลโปรแกรมส่วนที่เป็นเครื่องเสมือนจะทำการกระจายงานไปทำบนเครื่องที่ต้องการตามที่ได้ออกแบบไว้ในโปรแกรม

- Message Passing Interface (MPI) [12] เป็นมาตรฐานของการประมวลผลแบบขนานในระบบคลัสเตอร์ ซึ่งสนับสนุนภาษา C และ Fortran ในวิทยานิพนธ์ฉบับนี้ใช้ MPICH ซึ่งเป็นคลังชุดคำสั่งของการส่งข้อมูลที่สร้างตามมาตรฐาน MPI ลักษณะของการใช้คลังชุดคำสั่งและการโปรแกรมแบบขนานในลักษณะต่างๆจะกล่าวถึงในบทต่อไป

4.5 สรุป

ในบทนี้ได้กล่าวถึงการจำแนกระบบคอมพิวเตอร์ การประยุกต์ใช้การประมวลผลแบบขนาน และองค์ประกอบที่จำเป็นในการสร้างระบบคลัสเตอร์ ซึงประกอบด้วยระบบเครือข่ายความเร็วสูงและการเชื่อมต่อแบบต่างๆ โปรแกรมสำหรับการสื่อสาร ระบบปฏิบัติการ คลัสเตอร์มิดเดิ้ลแวร์ และโปรแกรมสร้างสภาวะแวดล้อมของการโปรแกรมแบบขนาน

ในบทต่อไปจะกล่าวถึงการเขียนโปรแกรมแบบขนานลักษณะต่างๆ การสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์เป็นลำดับต่อไป

บทที่ 5

โปรแกรมแบบขนาน

5.1 บทนำ

การประมวลผลแบบขนานนอกจากต้องมีระบบคอมพิวเตอร์แบบขนานแล้วยังจำเป็นต้องมีโปรแกรมแบบขนานเพื่อใช้สำหรับประมวลผลบนระบบคอมพิวเตอร์แบบขนาน ลักษณะของโปรแกรมแบบขนานจะขึ้นอยู่กับลักษณะของระบบคอมพิวเตอร์แบบขนานโดยโปรแกรมแบบขนานสำหรับระบบคอมพิวเตอร์ขนานแบบหนึ่งๆอาจไม่สามารถทำงานได้บนระบบคอมพิวเตอร์ขนานแบบอื่น

การสร้างโปรแกรมแบบขนานจะมีความยากง่ายแตกต่างกันไปขึ้นอยู่กับการประยุกต์ใช้งาน เครื่องมือที่ใช้ในการสร้างโปรแกรมแบบขนานและชนิดของระบบคอมพิวเตอร์แบบขนาน นอกจากนั้นประสิทธิภาพของโปรแกรมแบบขนานของงานหนึ่งๆอาจไม่เท่ากันเมื่อสร้างโปรแกรมแบบขนานนี้ด้วยวิธีที่ต่างกัน

การสร้างโปรแกรมแบบขนานมีสองวิธี

1. การสร้างโปรแกรมแบบขนานโดยนัย (Implicit Parallelism) วิธีนี้สามารถทำได้ง่ายโดยใช้ภาษาแบบขนานหรือตัวแปลภาษา (Compiler) แบบขนาน แต่วิธีนี้ผู้สร้างโปรแกรม (Programmer)ไม่สามารถควบคุมการจัดการ การแบ่งส่วนคำนวณ และการแบ่งข้อมูลได้ ซึ่งโดยทั่วไปแล้วจะมีประสิทธิภาพต่ำกว่าการสร้างโปรแกรมแบบขนานโดยตรง (Explicit Parallelism)

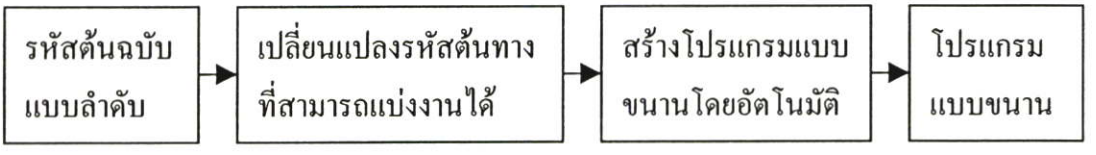
2. การสร้างโปรแกรมแบบขนานโดยตรง วิธีนี้ผู้สร้างโปรแกรมต้องทำการจัดแบ่งการทำงานเองทั้งหมด ซึ่งรูปแบบ และวิธีการแบ่งงานในแต่ละการประยุกต์ใช้จะไม่เหมือนกัน ซึ่งจะทำให้มีความยากง่ายแตกต่างกันด้วย แต่การใช้วิธีการแบ่งงานที่เหมาะสมกับการประยุกต์ใช้จะทำให้ได้ประสิทธิภาพในการประมวลผลสูงกว่าการสร้างโปรแกรมแบบขนานโดยนัย

5.2 รูปแบบของการพัฒนาโปรแกรมแบบขนาน

ก่อนที่จะสร้างโปรแกรมแบบขนานส่วนมากผู้สร้างโปรแกรมมักจะสร้างโปรแกรมในแบบลำดับ (Sequential Program) ซึ่งเป็นโปรแกรมที่ใช้ประมวลผลในเครื่องคอมพิวเตอร์แบบธรรมดาให้สามารถทำงานได้อย่างถูกต้องก่อน จากนั้นจะพัฒนาโปรแกรมแบบลำดับนี้ไปเป็นโปรแกรมแบบนานได้จากวิธีเหล่านี้

5.2.1 การสร้างโปรแกรมแบบขนานโดยอัตโนมัติ (Automatic Parallelism)

การสร้างโปรแกรมแบบขนานด้วยวิธีนี้เป็นวิธีที่ง่ายที่สุด แต่ก็จะมีประสิทธิภาพต่ำที่สุด โดยหน้าที่ของการสร้างโปรแกรมแบบขนานจะเป็นหน้าที่ของตัวแปรภาษาซึ่งตัวแปรภาษาจะตรวจสอบรหัสต้นฉบับ (Source Code) ซึ่งอาจประกอบด้วยส่วนของรหัสที่มีการวนซ้ำ และการประมวลผลกับข้อมูลที่ซ้ำๆกัน โดยตัวแปรภาษานี้จะทำการเปลี่ยนรหัสต้นฉบับนี้ไปเป็นรหัสที่สนับสนุนการประมวลผลแบบขนานจากนั้นใช้ตัวแปรภาษาเปลี่ยนรหัสที่สนับสนุนการประมวลผลแบบขนานนี้ไปเป็นโปรแกรมแบบขนาน ข้อจำกัดของการสร้างโปรแกรมแบบขนานด้วยวิธีนี้ขึ้นอยู่กับเทคโนโลยีที่ตัวแปรภาษาใช้ในการเปลี่ยนรหัสต้นฉบับเป็นรหัสที่สนับสนุนการประมวลผลแบบขนาน ขั้นตอนการทำงานของการสร้างโปรแกรมแบบขนานโดยอัตโนมัติซึ่งแสดงดังรูปที่ 5.1



รูปที่ 5.1 ขั้นตอนการทำงานของการสร้างโปรแกรมแบบขนานโดยอัตโนมัติ

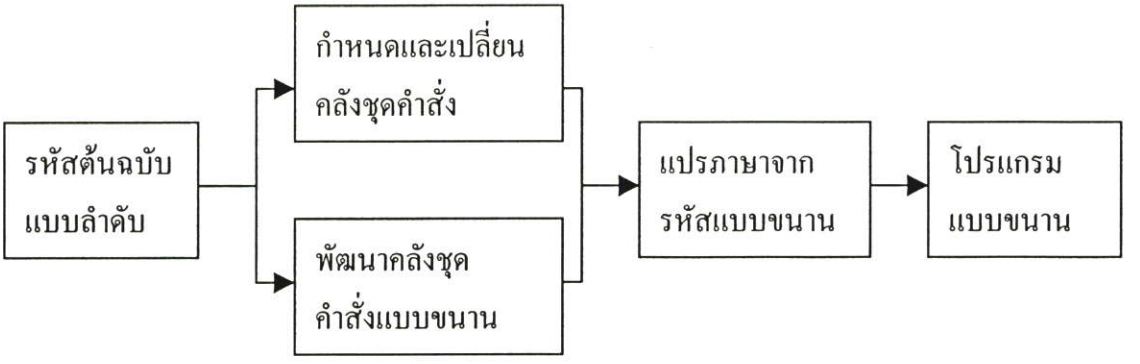
5.2.2 การสร้างโปรแกรมแบบขนานโดยใช้คลังของชุดคำสั่งแบบขนาน (Parallel Library)

วิธีนี้เป็นวิธีที่มีประสิทธิภาพดีกว่าวิธีแรก โดยใช้คลังของชุดคำสั่งแบบขนานที่พัฒนาไว้แล้วเช่น คลังชุดคำสั่งของการแปลงอนุกรมฟูเรียร์(Fourier) แบบขนาน และการคูณเมตริกแบบขนานเป็นต้น โดยการเปลี่ยนชุดคำสั่งจากรหัสต้นฉบับเป็นรหัสที่สนับสนุนการประมวลผลแบบขนานจะถูกทำโดยผู้สร้างโปรแกรมแบบขนาน จากนั้นใช้ตัวแปรภาษาทำการเปลี่ยนรหัสที่สนับสนุนการประมวลผลแบบขนานไปเป็นโปรแกรมแบบขนาน ขั้นตอนการสร้างโปรแกรมแบบขนานโดยใช้คลังของชุดคำสั่งแบบขนานแสดงดังรูปที่5.2 การใช้คลังของชุดคำสั่งแบบขนานสามารถสร้างโปรแกรมแบบขนานในสองรูปแบบ

- ใช้ควบคุมโครงสร้างของโปรแกรมแบบขนานในการกระจายงานไปตามส่วนต่างๆ ของหน่วยการประมวลผล (Processing Elements)
- ใช้ในการทดแทนชุดคำสั่งต้นฉบับที่สามารถเปลี่ยนการทำงานเป็นแบบขนานได้โดยใช้คลังชุดคำสั่งที่สนับสนุนการทำงานแบบขนานที่พัฒนาขึ้น

ในวิทยานิพนธ์ฉบับนี้ใช้วิธีการสร้างโปรแกรมแบบขนานโดยใช้คลังของชุดคำสั่งแบบขนาน ซึ่งใช้เพื่อควบคุมโครงสร้างของโปรแกรมแบบขนานในการกระจายงานไปตามโพรเซส (Processes)

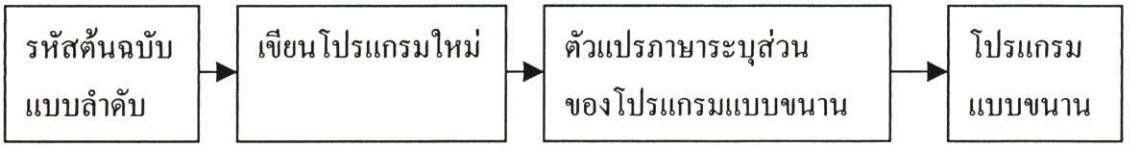
ต่างๆที่อยู่ในหน่วยการประมวลผล (Processors) แต่ไม่ได้ใช้คลังชุดคำสั่งในการเปลี่ยนรหัสต้นฉบับเป็นรหัสแบบขนาน



รูปที่ 5.2 ขั้นตอนการสร้างโปรแกรมแบบขนานโดยใช้คลังของชุดคำสั่งแบบขนาน

5.2.3 การสร้างโปรแกรมแบบขนานด้วยตัวเอง (Major Recoding)

การสร้างโปรแกรมแบบขนานด้วยวิธีนี้เป็นแบบที่ยืดหยุ่นมากที่สุด ผู้สร้างโปรแกรมสามารถเลือกตัวแปรภาษาใดก็ได้ที่ต้องการ รวมถึงสามารถเลือกรูปแบบและโปรแกรมที่ใช้ในการสื่อสาร แต่วิธีนี้นับเป็นวิธีที่ยากและเสียเวลามากที่สุด ขั้นตอนการสร้างโปรแกรมแบบขนานด้วยตัวเองแสดงดังรูปที่ 5.3



รูปที่ 5.3 ขั้นตอนการสร้างโปรแกรมแบบขนานด้วยตัวเอง

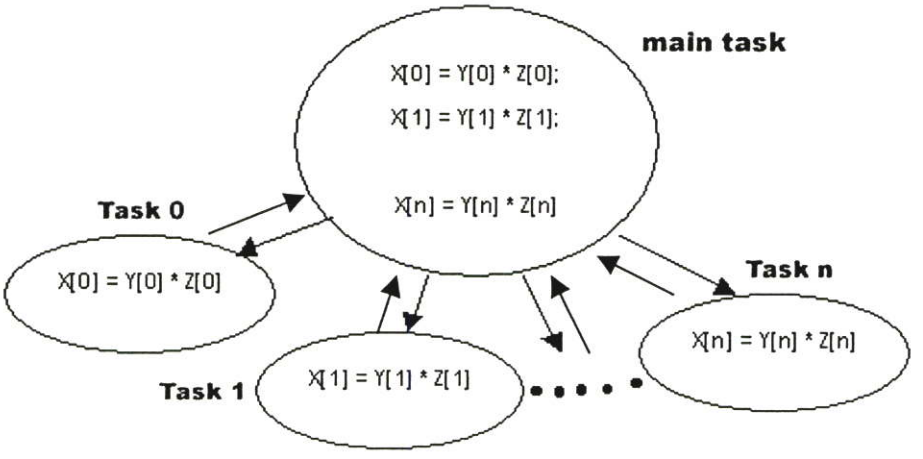
5.3 วิธีการออกแบบอัลกอริทึมของโปรแกรมแบบขนาน (Methodical Design of Parallel Algorithms)

การออกแบบอัลกอริทึมของโปรแกรมแบบขนานไม่มีข้อกำหนดตายตัว Ian Foster [15] ได้แนะนำวิธีการออกแบบอัลกอริทึมแบบขนานของโปรแกรมไว้สี่ขั้นตอน คือ

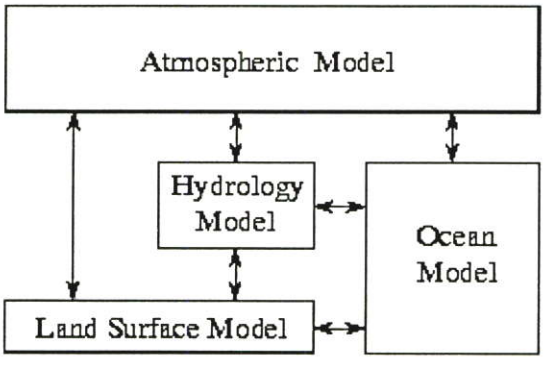
- ขั้นตอนการแบ่งงาน (Partitioning)
- ขั้นตอนออกแบบการสื่อสาร (Communication)
- ขั้นตอนการรวมกลุ่มงาน (Agglomeration)
- ขั้นตอนกำหนดงาน (Mapping)

5.3.1 ขั้นตอนการแบ่งงาน

การแบ่งงานเป็นหัวใจของการประมวลผลแบบขนาน ซึ่งเป็นการแยกการคำนวณออกเป็นส่วยย่อย การแบ่งงานสามารถทำได้สองแบบคือแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อย (Domain Decomposition) แบบที่สองคือแบ่งปัญหาตามหน้าที่ของการคำนวณออกเป็นส่วนย่อย (Functional Decomposition) ตัวอย่างของการแบ่งข้อมูลของปัญหาแสดงดังรูปที่ 5.4 ซึ่งเป็นการแบ่งข้อมูลในการคำนวณออกเป็นส่วนๆ ส่วนการแบ่งงานตามหน้าที่ของปัญหาแสดงดังรูปที่ 5.5 ซึ่งเป็นตัวอย่างของการสร้างแบบจำลองของสภาพภูมิอากาศ ซึ่งประกอบด้วยงานย่อยๆคือการสร้างแบบจำลองของทะเล พื้นดิน และก๊าซ โดยทั่วไปแล้วการแบ่งข้อมูลจะมีความยืดหยุ่นและสามารถทำได้ง่ายกว่าการแบ่งงานตามหน้าที่



รูปที่ 5.4 การแบ่งขอบเขตข้อมูลของปัญหา



รูปที่ 5.5 การแบ่งขอบเขตหน้าที่ของปัญหา

5.3.2 ขั้นตอนออกแบบการสื่อสาร

ขั้นตอนการสื่อสารจะถูกทำโดยพิจารณาการสื่อสารระหว่างงานที่แบ่งออกไป ในงานบางอย่าง งานย่อยที่ถูกแบ่งไม่สามารถประมวลผลได้ในเวลาเดียวกันเนื่องจากต้องอาศัยข้อมูลจาก

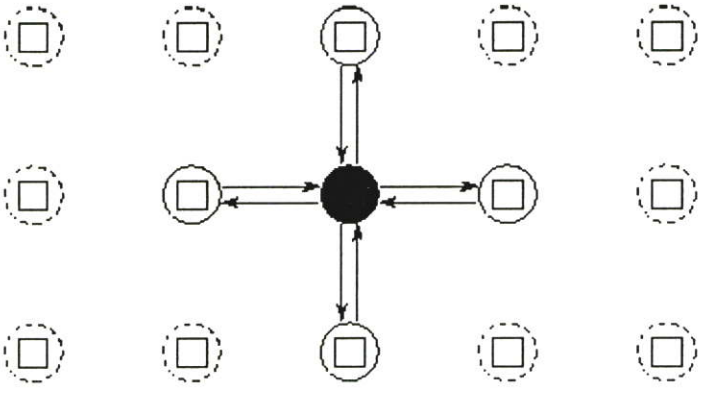
งานย่อยอื่นเพื่อใช้ในการประมวลผลดังนั้นจึงต้องมีการสื่อสารระหว่างงานย่อย การเลือกวิธีการสื่อสารที่เหมาะสมจะทำให้การปะมวลผลมีประสิทธิภาพมากยิ่งขึ้น การออกแบบขั้นตอนการสื่อสารจะแบ่งเป็นสองขั้นตอน ขั้นตอนแรกออกแบบโครงสร้างของการเชื่อมโยงว่างานใดต้องการข้อมูลและงานใดทำหน้าที่ให้ข้อมูล ขั้นตอนที่สองคือกำหนดโครงสร้างของข้อความ (Message) ที่ใช้ในการส่งและรับของแต่ละการเชื่อมโยง ลักษณะของการสื่อสารอาจแบ่งได้เป็นสี่แบบดังนี้

5.3.2.1 การสื่อสารระยะใกล้ และการสื่อสารแบบคลอบคลุม (Local and Global Communication)

การสื่อสารระยะใกล้คือการสื่อสารที่เกิดขึ้นระหว่างงานจำนวนน้อย เช่นการวิเคราะห์เชิงเลข (Numerical Analysis) ในวิธีไฟไนต์ดิฟเฟอเรนต์ของ Jacobi ซึ่งการหาค่า $X_{i,j}^{(t+1)}$ ทำได้จากสมการ 5.1 โดยใช้ข้อมูลห้าจุดในการคำนวณ

$$X_{i,j}^{(t+1)} = \frac{4X_{i,j}^{(t)} + X_{i-1,j}^{(t)} + X_{i+1,j}^{(t)} + X_{i,j-1}^{(t)} + X_{i,j+1}^{(t)}}{8} \tag{5.1}$$

งานแต่ละงานทำหน้าที่หาค่าของ $X_{i,j}^{(t+1)}$ ที่ตำแหน่งต่างๆกัน และในขณะเดียวกันก็ทำหน้าที่ส่งค่าของตัวเองให้กับงานอื่นด้วยดังรูปที่ 5.6

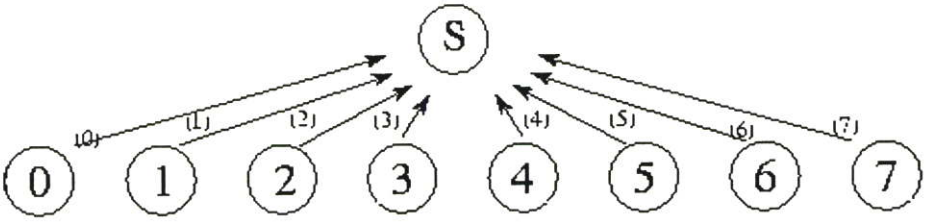


รูปที่ 5.6 ลักษณะการสื่อสารในการคำนวณไฟไนต์ดิฟเฟอเรนต์ในสองมิติ

การสื่อสารแบบคลอบคลุมจะเป็นการสื่อสารในลักษณะที่งานๆหนึ่งต้องติดต่อกับงานหลายๆงานเช่นการคำนวณหาค่าผลรวมดังสมการ 5.2

$$S = \sum_{i=0}^{N-1} X_i \tag{5.2}$$

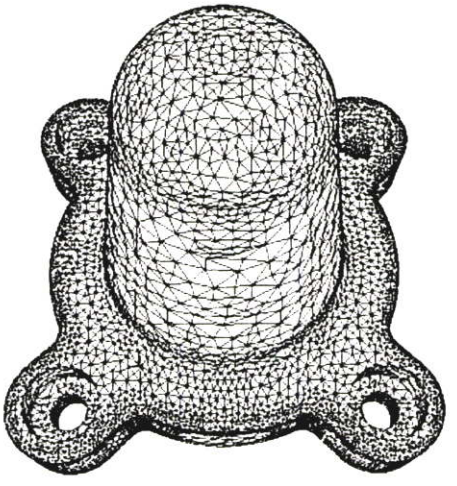
การคำนวณแบบขนานจะเกิดจากงานที่เป็นผู้จัดการ(Manager) จะรับค่า X_i จากงานที่เป็นคนงาน (Worker) เพื่อทำการหาค่า S โดยการสื่อสารจะมีลักษณะดังรูปที่ 5.7



รูปที่ 5.7 การสื่อสารเพื่อรวมค่าจากงานที่เป็นผู้จัดการ

5.3.2.2 การสื่อสารแบบไม่มีโครงสร้างและการสื่อสารแบบเปลี่ยนแปลงได้ (Unstructured and Dynamic Communication)

ในหัวข้อที่ผ่านมาเป็นการสื่อสารแบบคงที่และมีโครงสร้างซึ่งไม่มีการเปลี่ยนแปลงรูปแบบการสื่อสารตามเวลาเช่นการสื่อสารในรูปแบบตารางและแบบโครงสร้างต้นไม้ การสื่อสารแบบไม่มีโครงสร้างและสามารถเปลี่ยนแปลงได้จะใช้ในปัญหาเช่นการวิเคราะห์ไฟไนต์อีลีเมนต์ ซึ่งแสดงดังรูปที่ 5.8



รูปที่ 5.8 ตาราง (Grid) ซึ่งสร้างจากวิธีวิเคราะห์ไฟไนต์อีลีเมนต์

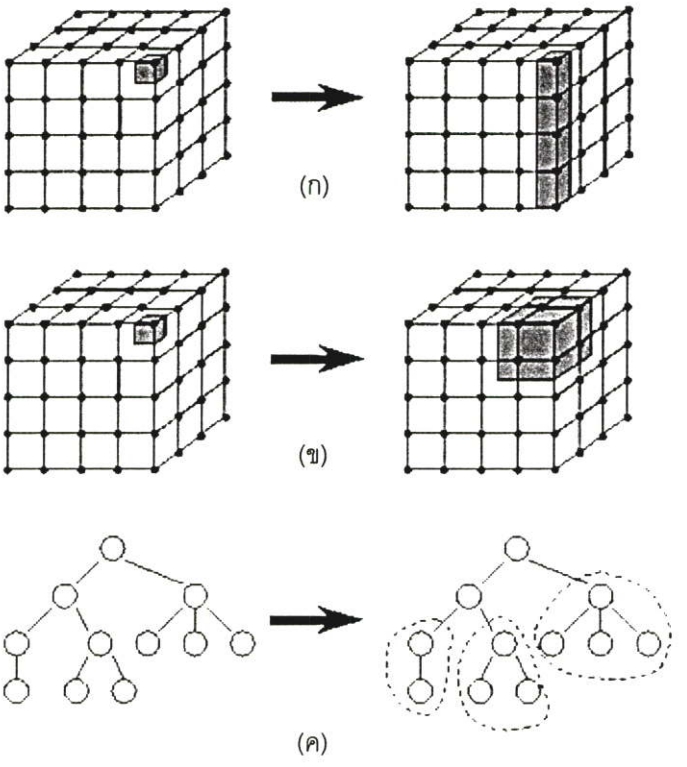
รูปร่างของตารางที่สร้างจากวิธีไฟไนต์อีลีเมนต์นี้จะไม่มีรูปแบบที่แน่นอนทำให้ไม่สามารถออกแบบการสื่อสารไว้ก่อนได้ วิธีแบ่งงานอาจทำได้โดยกำหนดให้แต่ละจุดต่อ (Vertex) ของเส้นตารางเป็นงานหนึ่งงาน และการสื่อสารจะทำตามจุดต่อตารางที่แบ่งไว้

5.3.2.3 การสื่อสารแบบประสานกันและการสื่อสารแบบไม่ประสานกัน (Synchronous and Asynchronous Communication)

การสื่อสารแบบประสานกัน ผู้ส่งและผู้รับรู้ว่าจะเกิดการสื่อสารกันก่อนจะเกิดการสื่อสารกันจริง ส่วนการสื่อสารแบบไม่ประสานกันผู้ส่งจะไม่รู้ว่าเมื่อใดผู้ระบบจะต้องการข้อมูล ผู้รับข้อมูลจะต้องทำการร้องขอข้อมูลโดยตรงจากผู้ส่งข้อมูลเมื่อต้องการ

5.3.3 ขั้นตอนรวมกลุ่มงาน

ขั้นตอนรวมกลุ่มงานจะถูกทำเพื่อลดเวลาที่ใช้ในการสื่อสารระหว่างงานลงและทำให้หน่วยประมวลผล (Processors) สามารถประมวลผลงานได้มากขึ้น



รูปที่ 5.9 การรวมกลุ่มงานในการประมวลผลแบบขนานเพื่อลดเวลาในการสื่อสาร

การรวมกลุ่มงานอาจทำได้หลายลักษณะขึ้นอยู่กับประเภทของงานเช่นจากรูปที่ 5.9 ก. เป็นการรวมกลุ่มงานจากงานในตารางสามมิติเป็นงานในตารางสองมิติ ข.เป็นการรวมกลุ่มงานจากงานในตารางสามมิติเป็นงานที่มีกลุ่มใหญ่ขึ้นในสามมิติ ค. เป็นการรวมกลุ่มงานในรูปแบบแผนภูมิต้นไม้เป็นต้น

5.3.4 ขั้นตอนกำหนดงาน

การกำหนดงานจะถูกทำเป็นขั้นตอนสุดท้ายในการออกแบบอัลกอริทึมแบบขนาน ในคอมพิวเตอร์แบบขนาน การกำหนดงานจะต่างจากระบบคอมพิวเตอร์ที่มีหน่วยประมวลผลเดี่ยว (Uniprocessor) โดยระบบปฏิบัติการจะเป็นผู้กำหนดและจัดลำดับงาน (Scheduling) ให้กับหน่วยประมวลผลเพื่อทำการประมวลผล ในคอมพิวเตอร์แบบขนานการกำหนดงานอาจทำโดยอัตโนมัติจากระบบปฏิบัติการแบบขนานหรืออาจกำหนดจากผู้สร้างโปรแกรมโดยตรง การกำหนดงานเป็นขั้นตอนที่ซับซ้อนซึ่งวิธีการกำหนดงานที่ดีจะทำให้ใช้เวลาในการประมวลผลงานน้อยและมีความยืดหยุ่นในการใช้งาน หลักการในการกำหนดงานที่ดีมีสองข้อคือ

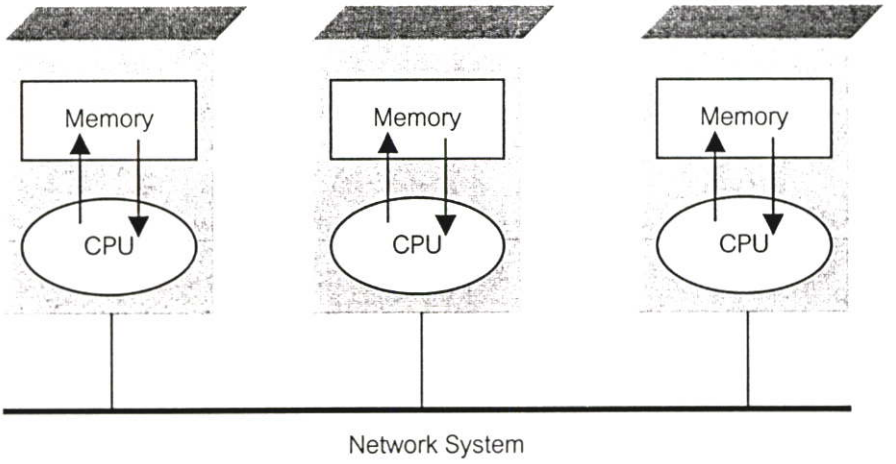
- กำหนดงานที่สามารถประมวลผลพร้อมๆกันได้ให้ทำงานในหน่วยประมวลผลต่างกัน
- กำหนดงานที่ต้องมีการสื่อสารกันบ่อยให้อยู่ในหน่วยประมวลผลเดียวกัน

การกำหนดงานอาจไม่มีรูปแบบที่แน่นอนขึ้นอยู่กับประเภทของงานและลักษณะคอมพิวเตอร์แบบขนาน ในระบบที่งานย่อยมีปริมาณงานไม่เท่ากันหรือระบบคอมพิวเตอร์แบบขนานเป็นแบบเนื้อผสม (Heterogeneous) ซึ่งความสามารถในการประมวลผลของแต่ละหน่วยประมวลผลอาจไม่เท่ากันจะทำให้วิธีการกำหนดงานมีความยากและซับซ้อนขึ้นไปอีก

5.4 การสื่อสารระหว่างงานด้วยวิธีส่งข้อความ (Message Passing)

จากบทที่4 ได้กล่าวถึงโครงสร้างของคอมพิวเตอร์แบบขนานซึ่งระบบคลัสเตอร์เป็นระบบคอมพิวเตอร์แบบขนานที่สื่อสารระหว่างงานด้วยการส่งข้อความ และแต่ละเครื่องมีหน่วยความจำเป็นของตัวเอง (DMMP) ซึ่งแสดงดังรูปที่5.10 วิธีการส่งข้อความในระบบแบบหลายคอมพิวเตอร์ (Multicomputer) สามารถทำได้โดย

- ออกแบบตัวแปรภาษาแบบพิเศษซึ่งสนับสนุนการประมวลผลแบบขนาน
- เพิ่มไวยากรณ์ (Syntax) ในตัวแปรภาษาที่มีอยู่แล้วให้สนับสนุนการสื่อสารแบบส่งข้อความ
- ใช้ตัวแปรภาษาที่มีอยู่แล้วร่วมกับคลังชุดคำสั่ง(Library) ซึ่งสนับสนุนการสื่อสารแบบส่งข้อความ



รูปที่ 5.10 แต่ละเครื่องในระบบคลัสเตอร์มีหน่วยความจำของตัวเอง

ตัวแปรภาษาที่สนับสนุนการสื่อสารแบบส่งข้อความได้แก่ OCCAM ซึ่งออกแบบมาเพื่อใช้กับหน่วยประมวลผลที่เรียกว่า Transputer ส่วนตัวแปรภาษาที่มีการเพิ่มไวยากรณ์ของการสื่อสารด้วยการส่งข้อความได้แก่ CC+ และ Fortran M ในวิทยานิพนธ์ฉบับนี้ใช้คลังชุดคำสั่งของการสื่อสารแบบส่งข้อความร่วมกับตัวแปรภาษาที่มีอยู่แล้วโดยคลังชุดคำสั่งนี้มีชื่อว่า MPI (Message Passing Interface) [12] [16] [17] คลังชุดคำสั่งอีกตัวหนึ่งที่นิยมใช้กันคือ PVM (Parallel Virtual Machine) [14]

ก่อนที่ระบบคอมพิวเตอร์จะทำการประมวลผลงานจะต้องมีการสร้างโพรเซสขึ้น โดยโพรเซสจะเกิดจากระบบปฏิบัติการ การสร้างโพรเซสในระบบคลัสเตอร์จะถูกทำโดยระบบปฏิบัติการของแต่ละหน่วยประมวลผลเช่นเดียวกันกับระบบคอมพิวเตอร์อื่นๆ การควบคุมการสร้างโพรเซสจะทำจากเครื่องที่ถูกกำหนดเป็นผู้จัดการ(Manager) หรือเครื่องที่ทำการสั่งประมวลผลโปรแกรม โดยทั่วไปการสร้างงานในระบบคลัสเตอร์ที่ใช้คลังชุดคำสั่งในการสื่อสารระหว่างงานจะมีสองแบบคือ

- การสร้างโพรเซสแบบกำหนดไว้ก่อน (Static Process Creation) วิธีการสร้างโพรเซสแบบนี้จะใช้ในคลังชุดคำสั่ง MPI ซึ่งการสร้างโพรเซสจะเกิดตอนเริ่มสั่งให้ประมวลผลโปรแกรม
- การสร้างโพรเซสแบบกำหนดในขณะโปรแกรมทำงาน (Dynamic Process Creation) ซึ่งการสร้างโพรเซสในลักษณะนี้จะเกิดเมื่อใดก็ได้เมื่อโพรเซสที่เป็นผู้จัดการยังทำงานอยู่ การสร้างโพรเซสลักษณะนี้จะใช้ในคลังชุดคำสั่ง PVM

ใน MPI จะใช้แบบจำลองโปรแกรมแบบโปรแกรมเดี่ยวหลายข้อมูล (Single Program Multiple Data) หรือ SPMD ซึ่งทุกๆโพรเซสที่ถูกสร้างในระบบคลัสเตอร์จะประมวลผลโปรแกรมเดียวกันแต่จะสามารถควบคุมให้โพรเซสที่อยู่ในหน่วยประมวลผลที่ต่างกันสามารถประมวลผลในส่วนของโปรแกรมที่ต่างกันได้ ดังแสดงในรูปที่ 5.11

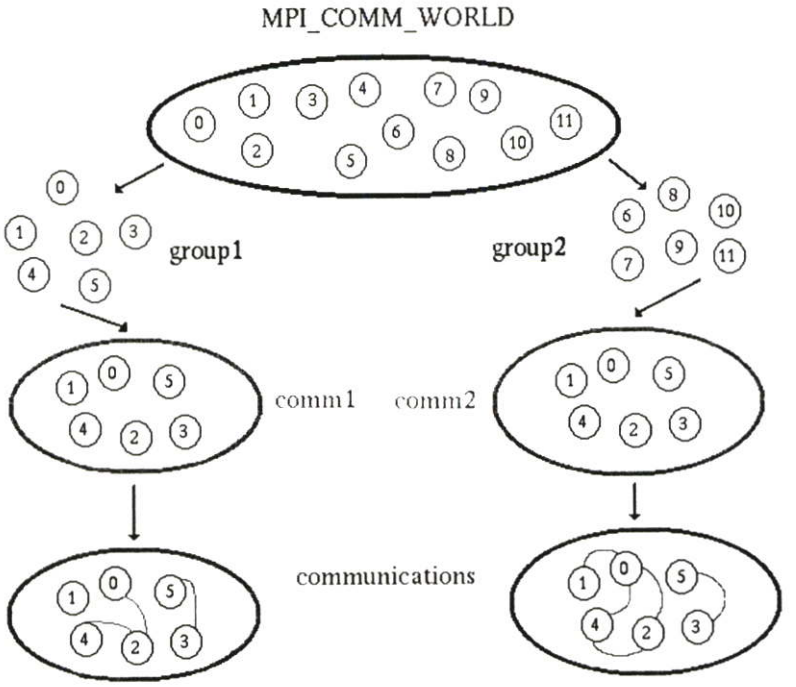
```
int main(int argc, char *argv[])
{
        if(tid == 0){
        ... Manager Code ...
        }else{
        ... Worker Code ...
        }
}
```

รูปที่ 5.11 ลักษณะรหัสต้นฉบับซึ่งมีโครงสร้างแบบโปรแกรมเดี่ยวหลายข้อมูล

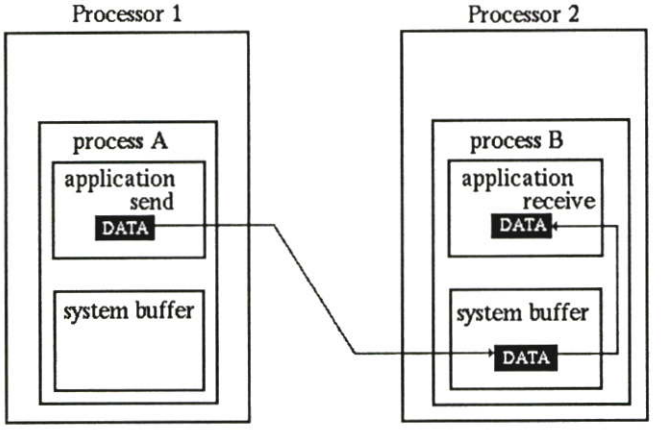
จากรูปที่ 5.11 เป็นรหัสต้นฉบับของโปรแกรมแบบขนาน ค่า tid ที่ได้จากคลังชุดคำสั่ง MPI จะบอกว่าโปรแกรมกำลังประมวลผลอยู่บนโพรเซสใด ถ้า tid มีค่าเท่ากับศูนย์แสดงว่าโปรแกรมกำลังถูกประมวลผลอยู่บนโพรเซสที่ทำหน้าที่เป็นผู้จัดการ ส่วนโพรเซสที่มีค่า tid อื่นๆจะเป็นโพรเซสที่มีหน้าที่เป็นคนงาน จากคุณสมบัติของค่า tid ดังกล่าวนี้จะทำให้สามารถใส่รหัสต้นฉบับของโปรแกรมแบบขนานสำหรับงานต่างๆในระบบคลัสเตอร์ไว้ในโปรแกรมเดียวได้โดยใช้ประโยคเงื่อนไข if else เป็นตัวแบ่งแยก

งานใน MPI สามารถจัดรวมกลุ่มได้ด้วยสิ่งที่เรียกว่า COMMUNICATOR งานทั้งหมดจะถูกระบุด้วยตัวแปร MPI_COMM_WORLD ผู้สร้างโปรแกรมสามารถกำหนดกลุ่มของงานเพิ่มเติมเพื่อให้สามารถจัดการงานเหล่านั้นได้ง่ายขึ้นดังแสดงในรูปที่ 5.12

จากหัวข้อที่ 5.3.2 ซึ่งเป็นการออกแบบลักษณะการสื่อสารระหว่างงานในระบบคลัสเตอร์ MPI ได้จัดเตรียมคลังชุดคำสั่งสำหรับการส่งและรับข้อมูลลักษณะต่างๆ [18] ซึ่งสามารถประยุกต์ใช้ในโปรแกรมแบบขนานที่ออกแบบไว้ ลักษณะของการส่งและรับข้อมูลของ MPI เป็นดังรูปที่ 5.13 ซึ่งเป็นการส่งข้อมูลจากโพรเซส A ของหน่วยประมวลผล 1 ไปสู่โพรเซส B ในหน่วยประมวลผล 2 การส่งข้อมูลจะส่งโดยตรงจากตัวแปรของโปรแกรมส่วนการรับข้อมูลจะทำผ่านตัวเก็บข้อมูลของระบบ (Buffer) ก่อนแล้วจึงคัดลอกสู่ตัวแปรของโปรแกรมอีกครั้งหนึ่งเพื่อทำให้สามารถสื่อสารแบบไม่ประสานกันได้ (Asynchronous) การสื่อสารของ MPI สามารถแบ่งได้เป็นการสื่อสารแบบจุดต่อจุด (Point-Point Communication) และแบบสื่อการเป็นกลุ่ม (Collective Communication)



รูปที่ 5.12 การจัดกลุ่มงานใน MPI



รูปที่ 5.13 การส่งและรับข้อมูลของ MPI

5.4.1 การสื่อสารแบบจุดต่อจุด (Point-Point Communication)

การสื่อสารแบบจุดต่อจุดนี้เป็นพื้นฐานของระบบสื่อสารแบบส่งข้อความ ซึ่งข้อความจะถูกส่งระหว่างโพรเซสสองโพรเซสโดยด้านหนึ่งเป็นผู้ส่งและอีกด้านหนึ่งเป็นผู้รับ คลังชุดคำสั่งที่สนับสนุนการส่งข้อความลักษณะนี้แบ่งเป็นสองกลุ่มคือการส่งและรับข้อความแบบมีการยืนยันและ การส่งและรับข้อความแบบไม่มีการยืนยัน

5.4.1.1 การส่งและรับข้อความแบบมีการยืนยัน (Blocking Send and Blocking Receive)

การส่งแบบนี้จะเป็นการส่งข้อมูลที่มีความแน่นอน ขั้นตอนการส่งจะสิ้นสุดและคืนการทำงานให้กับโปรแกรมต่อเมื่อผู้รับได้รับข้อมูลเรียบร้อยแล้ว ข้อควรระวังในการใช้การส่งข้อความด้วยวิธีนี้คือการเกิดเหตุการณ์ล็อกตาย (Dead Lock) ซึ่งเป็นเหตุการณ์ที่งานรอการส่งข้อมูลตลอดไปโดยไม่มีการคืนการทำงานให้โปรแกรม ทำให้ไม่สามารถทำงานเสร็จได้ คำสั่งของ MPI ในกลุ่มคำสั่งนี้ได้แก่ MPI_SEND และ MPI_RECV

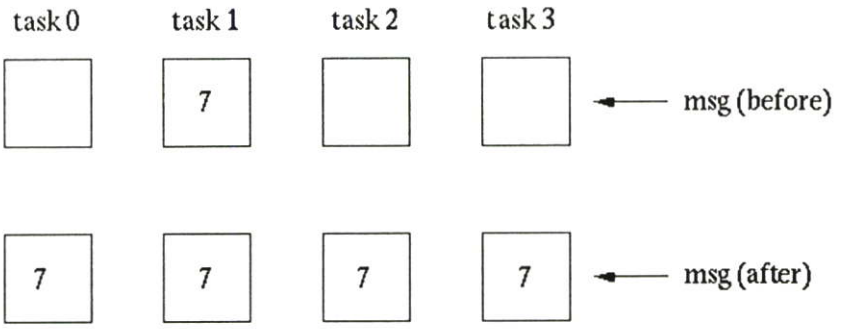
5.4.1.2 การส่งและรับข้อความแบบไม่มีการยืนยัน (Non-Blocking Send and Non-Blocking Receive)

การส่งและรับข้อความแบบไม่มีการยืนยันจะถูกทำเพื่อปรับปรุงประสิทธิภาพของการส่งข้อความ การส่งข้อความแบบไม่มีการยืนยันนี้จะไม่รอให้ฝ่ายรับข้อความได้รับข้อความก่อนเหมือนการส่งข้อมูลแบบมีการยืนยันโดยจะคืนการทำงานให้โปรแกรมทันที การส่งข้อความลักษณะนี้มีความเสี่ยงในความผิดพลาดเมื่อมีการเขียนทับตัวแปรของงานที่ต้องการส่งข้อความและรับข้อความก่อนที่ข้อความจะถูกส่งไปจริง MPI จึงมีคำสั่งที่ตรวจสอบความสมบูรณ์ของการส่งและรับด้วยคำสั่ง MPI_WAIT และ MPI_TEST [17] เพื่อตรวจสอบสถานะการส่งและรับข้อความก่อนที่จะมีการเขียนข้อมูลทับข้อมูลเก่า ตัวอย่างคำสั่งของการส่งและรับข้อความแบบไม่มีการยืนยันคือ MPI_ISEND และ MPI_IRECV

5.4.2 สื่อการเป็นกลุ่ม (Collective Communication)

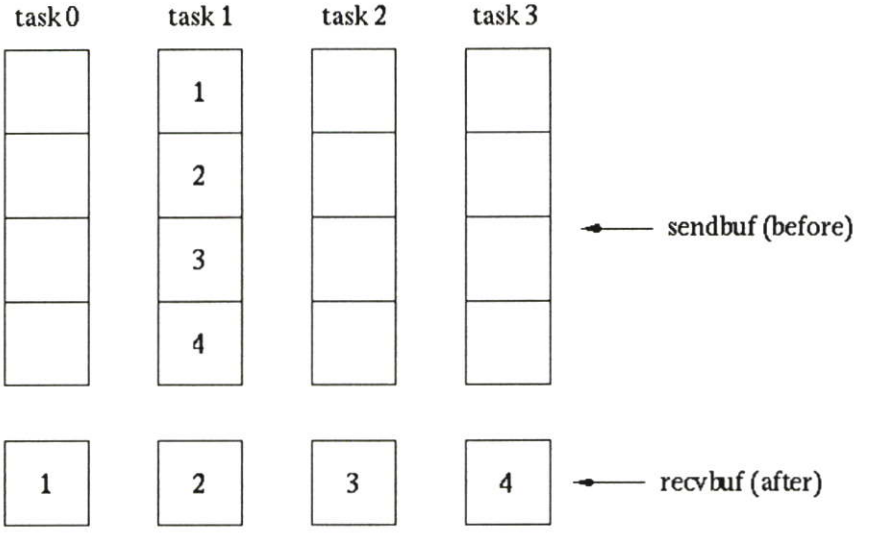
การสื่อสารแบบเป็นกลุ่มจะเป็นการสื่อสารของทุกโพรเซสในกลุ่ม ตัวอย่างคำสั่งของการสื่อการแบบเป็นกลุ่มคือ

- MPI_Bcast เป็นคำสั่งในการกระจายค่าในตัวแปร (msg) ไปให้ทุกๆโพรเซสในกลุ่ม ดังแสงในรูปที่ 5.14



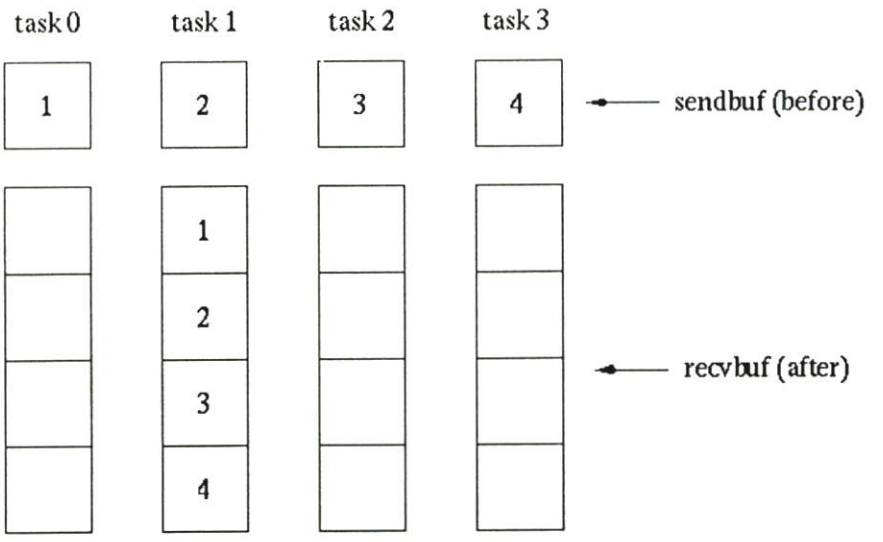
รูปที่ 5.14 ลักษณะการทำงานของ MPI_Bcast

- MPI_Scatter เป็นคำสั่งในการกระจายค่าในอะเรย์ (Array) สำหรับส่งข้อมูล (sendbuf) ไปสู่ตัวแปรของทุกๆโพรเซสในกลุ่ม (recvbuf) ตามลำดับตำแหน่งของอะเรย์ที่ใช้ส่งนั้น ดังแสดงในรูปที่ 5.15



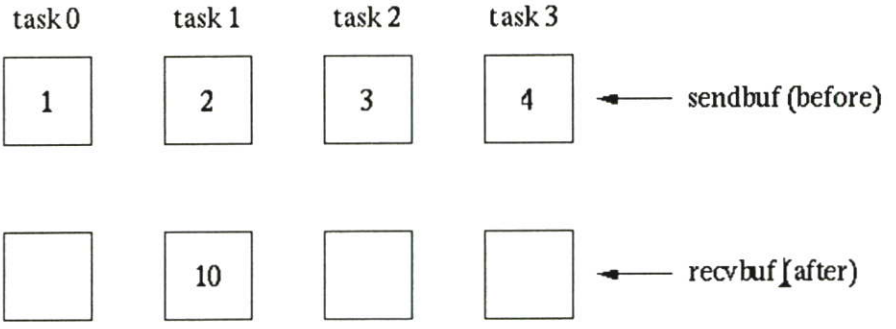
รูปที่ 5.15 ลักษณะการทำงานของ MPI_Scatter

- MPI_Gather เป็นการเก็บรวบรวมค่าในตัวแปร (sendbuf) ของโพรเซสในกลุ่มมาใส่ในอะเรย์ (recvbuf) ของโพรเซสที่เรียกใช้คำสั่งตามลำดับงานดังแสดงในรูปที่ 5.16



รูปที่ 5.16 ลักษณะการทำงานของ MPI_Gather

- MPI_Reduce เป็นการรวมค่าที่เก็บอยู่ในตัวแปร (sendbuf) ของทุกๆโพรเซสมาเก็บในตัวแปร (recvbuf) ของโพรเซสที่เรียกใช้คำสั่งนี้ดังแสดงในรูปที่ 5.17



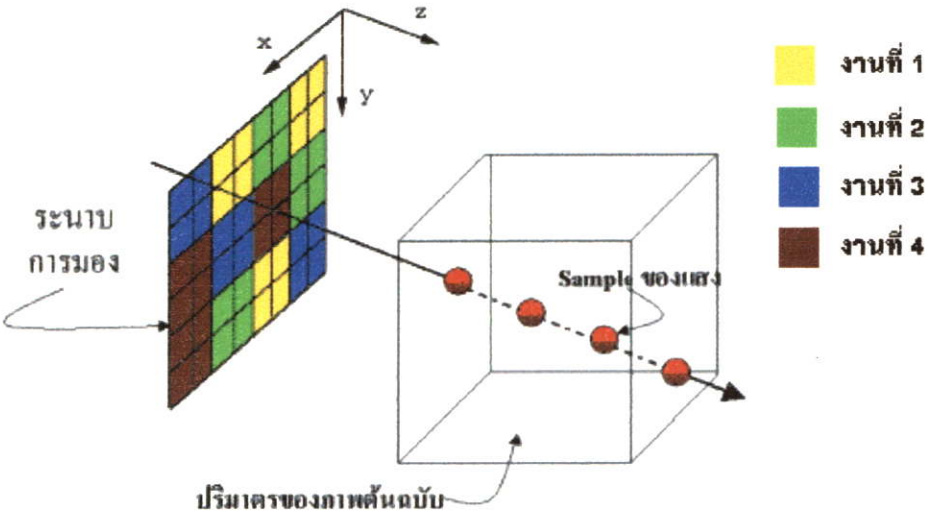
รูปที่ 5.17 ลักษณะการทำงานของ MPI_Reduce

5.5 การสร้างภาพเชิงปริมาตรทางการแพทย์บนระบบคลัสเตอร์

ในวิทยานิพนธ์ฉบับนี้เป็นการพัฒนาการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์เพื่อเพิ่มความเร็วในการสร้างภาพของแต่ละมุมมองให้เร็วขึ้น ในวิธีการที่นำเสนอใช้หลักการที่ว่ากระบวนการสร้างภาพเชิงปริมาตรสามารถแบ่งการทำงานออกได้เป็นกลุ่มๆซึ่งไม่ขึ้นต่อกัน และไม่มีความจำเป็นต้องมีการสื่อสารระหว่างงานย่อย (Embarrassingly Parallel Computations) ซึ่งนับเป็นข้อได้เปรียบของงานประมวลผลภาพ และงานด้านคอมพิวเตอร์กราฟิกส่วนใหญ่ งานประเภทอื่นอาจต้องใช้วิธีการแบ่งงานที่ต่างออกไป [19]

5.5.1 การแบ่งงานในการสร้างภาพเชิงปริมาตร

การแบ่งงานในการสร้างภาพเชิงปริมาตรใช้วิธีแบ่งข้อมูลของปัญหาออกเป็นส่วยย่อย (Domain Decomposition) โดยหน่วยของข้อมูลที่เล็กที่สุดคือจุดภาพ (Voxel) ตามแนวลำแสงดังแสดงในรูปที่ 5.18 การแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อยมีข้อดีคือสามารถทำงานได้ไม่ว่าจะมีจำนวนโพรเซสอยู่ในระบบคลัสเตอร์เท่าใดก็ตาม (Scalable) แต่ต้องไม่มากกว่าจำนวนกลุ่มข้อมูลที่แบ่งไว้โดยสามารถแบ่งได้ละเอียดที่สุดถึงหนึ่งแนวลำแสงต่อหนึ่งงาน ปัญหาที่พบคือถ้าหากกำหนดให้ข้อมูลของปริมาตรภาพไปที่แต่ละโพรเซสในระบบคลัสเตอร์ เมื่อต้องการหมุนภาพจะต้องทำการกระจายชุดข้อมูลที่แบ่งนั้นไปให้แต่ละโพรเซสในระบบใหม่ ซึ่งข้อมูลภาพอาจมีขนาดใหญ่มากทำให้สินเปลืองเวลาในการส่งข้อมูล วิธีการแก้ไขคือให้แต่ละโพรเซสทำการอ่านข้อมูลภาพทั้งปริมาตรไปเก็บไว้ในหน่วยความจำของตัวเอง เมื่อมีการหมุนภาพก็เพียงทำการหมุนตามขั้นตอนของการสร้างภาพเชิงปริมาตรดังที่กล่าวมาแล้วในบทที่ 3 แต่ข้อเสียของวิธีนี้คือทำให้สิ้นเปลืองหน่วยความจำ



ภาพที่ 5.18 การแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อยตามแนวลำแสง

5.5.2 การรวมกลุ่มงาน

การรวมกลุ่มงานของการสร้างภาพเชิงปริมาตรนี้สามารถทำได้ง่ายโดยให้แนวลำแสงที่อยู่ติดกันรวมเป็นกลุ่มงานเดียวกัน การกำหนดให้งานหนึ่งงานเป็นการสร้างภาพในแนวหนึ่งลำแสงอาจเป็นงานที่ต้องอาศัยการคำนวณน้อยเกินไปและทำให้มีการสื่อสารมากเกินความจำเป็นซึ่งอาจทำให้เวลาที่ใช้ในการสร้างภาพส่วนใหญ่เสียไปกับการสื่อสารในขณะที่แต่ละโพรเซสใช้เวลาในการคำนวณของแต่ละงานน้อยมาก ในวิทยานิพนธ์ฉบับนี้ใช้การรวมกลุ่มข้อมูลขนาด 16x16 32x32 และ 64x64 แนวลำแสง

5.5.3 การสื่อสารระหว่างงานในการสร้างภาพเชิงปริมาตร

เนื่องจากงานที่ถูกแบ่งในการสร้างภาพเชิงปริมาตรไม่จำเป็นต้องมีการสื่อสารระหว่างกลุ่มงานด้วยกัน การออกแบบการสื่อสารจึงไม่มีความยุ่งยาก ซึ่งในวิทยานิพนธ์ฉบับนี้ใช้หลักการของศูนย์รวมงาน (Work Pool) [19] ซึ่งเป็นการสื่อสารแบบสองทิศทางระหว่างโพรเซสที่เป็นผู้จัดการและโพรเซสที่เป็นคนงานโดยมีลักษณะดังรูปที่ 5.20 โครงสร้างของข้อมูลที่ใช้ส่งประกอบด้วยตำแหน่งของจุดเริ่มของบล็อคตามแนวแกน X และ Y ส่วนโครงสร้างของข้อมูลที่ใช้รับประกอบด้วยตำแหน่งของจุดเริ่มของบล็อคที่ประมวลผลเสร็จและค่าสีของข้อมูลภาพภายในบล็อค ดังแสดงในรูปที่ 5.19

```
typedef struct{
   int x;  /* x origin*/
   int y; /* y origin */
} manager_block;
```

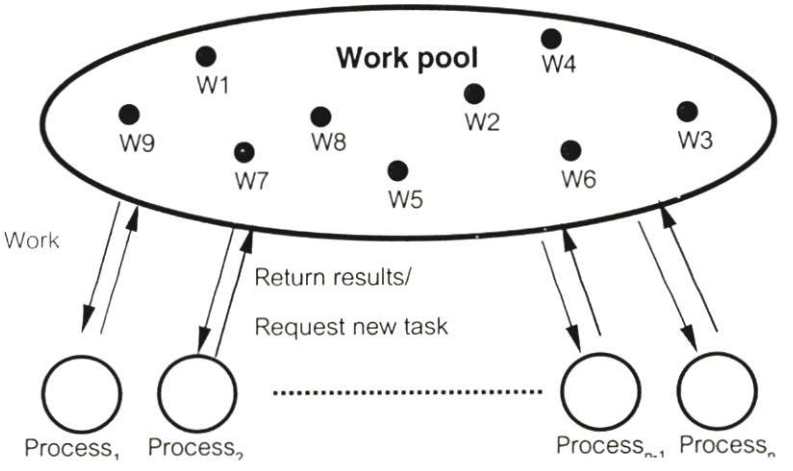
(ก)

```
typedef struct {
   int x;  /* x origin */
   int y; /* y origin */
   /* image data */
   unsigned char data[blocksize][blocksizex3];
}worker_block;
```

(ข)

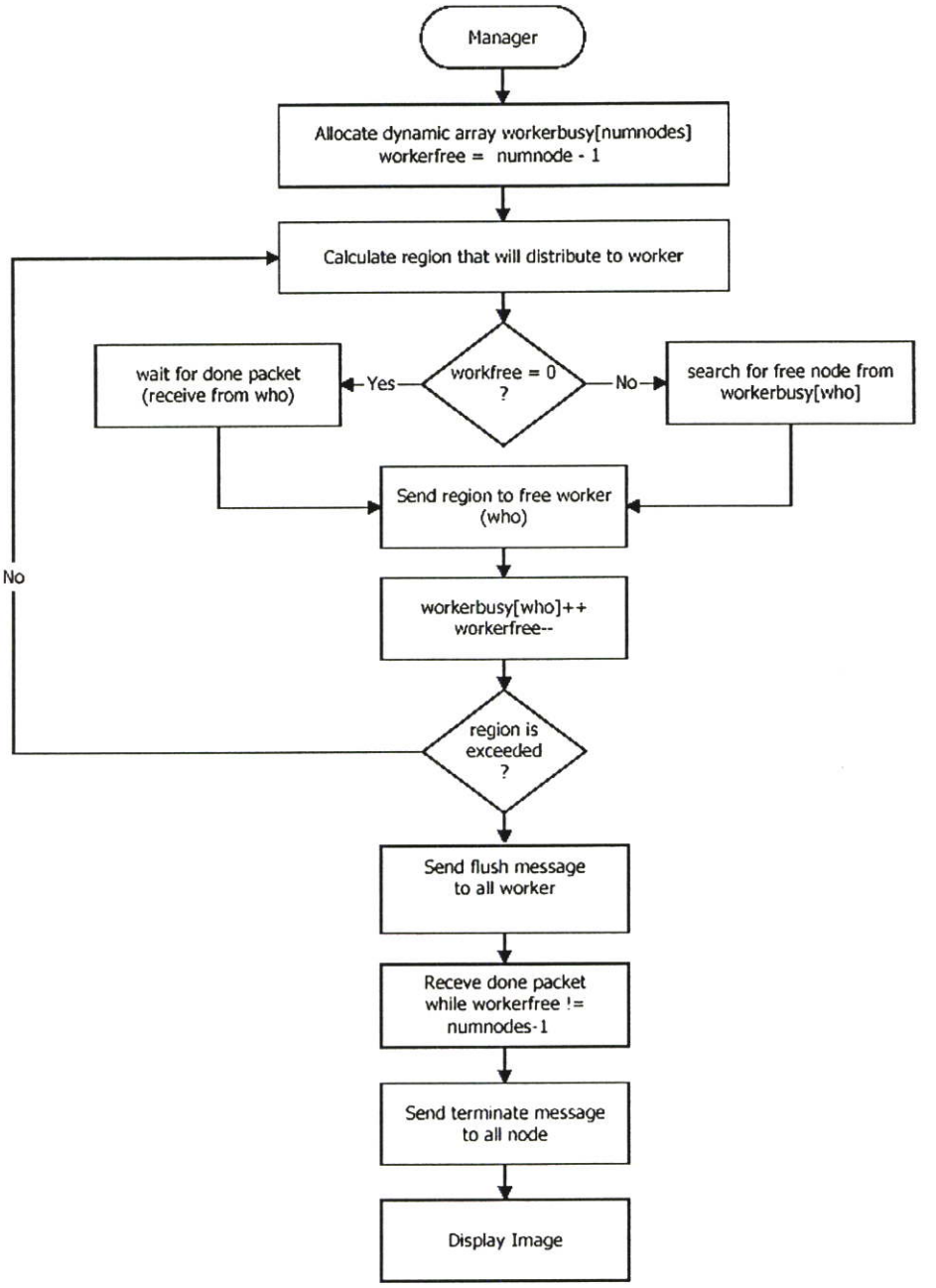
รูปที่ 5.19 โครงสร้างข้อมูลในการสื่อสารของการสร้างภาพเชิงปริมาตร
(ก) โครงสร้างข้อมูลที่ใช้ในการส่งไปสู่ของโพรเซสที่เป็นคนงาน (Worker)
(ข) โครงสร้างข้อมูลที่รับจากโพรเซสที่เป็นคนงาน

5.6.4 การกำหนดงานในการสร้างภาพเชิงปริมาตร

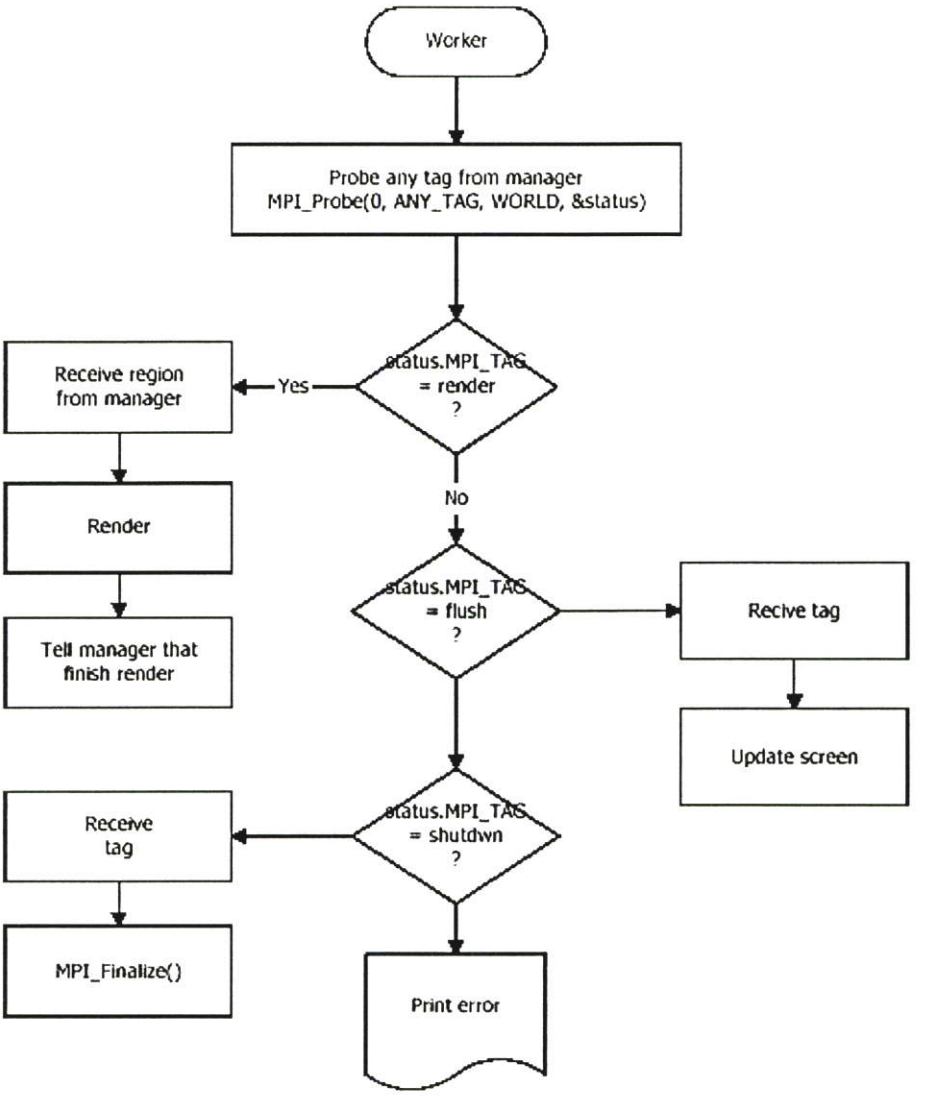


รูปที่ 5.20 การกำหนดงานแบบศูนย์รวมงาน (Work Pool)

การกำหนดงานแบบศูนย์รวมงานอาศัยหลักการที่ว่าให้โพรเซสที่สร้างขึ้นในระบบคลัสเตอร์ไปรับงานที่ต้องประมวลผลจากศูนย์รวมงาน แทนที่จะให้ผู้สร้างโปรแกรมเป็นผู้กำหนดงานที่ต้องทำให้แก่โพรเซสโดยตรงซึ่งทำให้ขาดความยืดหยุ่นในกรณีที่ต้องการให้มีจำนวนโพรเซสในระบบมากขึ้นหรือต้องการเปลี่ยนแปลงขนาดของกลุ่มงาน ขั้นตอนการทำงานของโปรแกรมในส่วนผู้จัดการ (Manager) แสดงดังรูปที่ 5.21 ส่วนขั้นตอนการทำงานของโปรแกรมในส่วนคนงาน (Worker) แสดงดังรูปที่ 5.22



รูปที่ 5.21 ขั้นตอนการทำงานของโปรแกรมในส่วนผู้จัดการ



รูปที่ 5.22 ขั้นตอนการทำงานของโปรแกรมในส่วนคนงาน (Worker)

5.6 สรุป

ในบทนี้ได้กล่าวถึงหลักการสร้างโปรแกรมแบบขนานซึ่งลักษณะของโปรแกรมอาจแตกต่างกันขึ้นอยู่กับงานที่นำไปประยุกต์ใช้ ขั้นตอนการออกแบบโปรแกรมแบบขนานสามารถทำได้โดยเริ่มจากขั้นตอนการแบ่งงาน การออกแบบลักษณะของการสื่อสาร การรวมกลุ่มงาน และสุดท้ายการกำหนดงาน นอกจากนั้นในบทนี้ยังได้อธิบายถึงหลักการสร้างโปรแกรมการสร้างภาพเชิงปริมาตรแบบขนานบนระบบคลัสเตอร์ซึ่งใช้หลักการของศูนย์รวมงาน ในบทต่อไปจะกล่าวถึงวิธีการทดลองและผลการทดลองเป็นลำดับต่อไป

บทที่ 6

การทดลองและผลการทดลอง

6.1 บทนำ

ในบทนี้จะกล่าวถึงการทดลองและผลการทดลองของการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์โดยแบ่งการทดลองออกเป็นสองส่วน ส่วนแรกวัดเวลาที่ใช้ในการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์ของภาควิชาฟิสิกส์มหาวิทยาลัยขอนแก่น และในส่วนที่สองจะเป็นการจำลองเหตุการณ์ในการกำหนดงานบนระบบคลัสเตอร์ของวิธีการที่นำเสนอในวิทยานิพนธ์ฉบับนี้ ที่แต่ละเครื่องในระบบมีความสามารถในการประมวลผลไม่เท่ากัน ซึ่งโพรเซสที่ทำงานอยู่บนเครื่องที่มีความสามารถในการประมวลผลต่ำควรจะได้งานไปทำน้อยกว่าโพรเซสที่ทำงานอยู่บนเครื่องที่มีความสามารถในการประมวลผลสูงกว่า

6.2 การวัดประสิทธิภาพของการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์

การวัดเวลาที่ใช้ในการสร้างภาพเชิงปริมาตรในวิทยานิพนธ์ฉบับนี้จะทำการวัดเวลาที่ใช้ในการสร้างภาพจริงๆเท่านั้นซึ่งจะไม่รวมไปถึงเวลาที่ใช้ในการอ่านภาพมาเก็บไว้ในหน่วยความจำและการตั้งค่าระบบ (Initialization) ของ MPI ซึ่งกระบวนการทั้งสองจะถูกทำครั้งเดียวในการสร้างภาพ เมื่อต้องทำการสร้างภาพในมุมมองใหม่จะไม่ต้องทำกระบวนการทั้งสองอีก แนวโน้มของเวลาที่ใช้ในการสร้างภาพด้วยวิธีการที่นำเสนอในวิทยานิพนธ์นี้เป็นดังสมการ 6.1

$$T = \frac{100}{p} \tag{6.1}$$

เมื่อ T คือเปอร์เซ็นต์ของเวลาที่ใช้ในการสร้างภาพ

p คือจำนวนเครื่องที่ใช้ประมวลผลในระบบคลัสเตอร์

เวลาที่ใช้ในการประมวลผลจะมีลักษณะตามสมการ 6.2 [16]

$$T_p = t_{comp} + t_{comm} \tag{6.2}$$

เมื่อ T_p คือเวลาที่ใช้ในการสร้างภาพ

t_{comp} คือเวลาที่ใช้ในการประมวลผล

t_{comm} คือเวลาที่ใช้ในการสื่อสารระหว่างโพรเซส

ระบบคลัสเตอร์เป็นระบบที่การสื่อสารระหว่างโพรเซสจะทำผ่านระบบเครือข่ายซึ่งความเร็วของการสื่อสารจะขึ้นอยู่กับชนิดและคุณภาพของอุปกรณ์ระบบเครือข่าย อุปกรณ์เครือข่ายที่ดีควรจะทำให้ t_{comm} มีค่าน้อยๆ เพราะเมื่อเกิดการสื่อสารระหว่างหลายๆโพรเซสในระบบคลัสเตอร์ t_{comm} จะมีค่ามากขึ้นตามจำนวนของการสื่อสารเหล่านั้นด้วยซึ่งจะส่งผลให้เวลาที่ใช้ในการสร้างภาพมากขึ้น กราฟในอุดมคติของเวลาที่ใช้ในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ซึ่งให้ $t_{comm} = 0$ แสดงดังรูปที่ 6.1

นอกจากการวัดเวลาที่ใช้ในการประมวลผลแล้วยังสามารถวัดประสิทธิภาพได้จากค่าอัตราการเพิ่มของความเร็ว (Speedup) ของการสร้างภาพได้จากสมการ 6.3

$$S_p = \frac{T_1}{T_p} \tag{6.3}$$

เมื่อ S_p คืออัตราการเพิ่มของความเร็วที่ใช้ในการสร้างภาพ

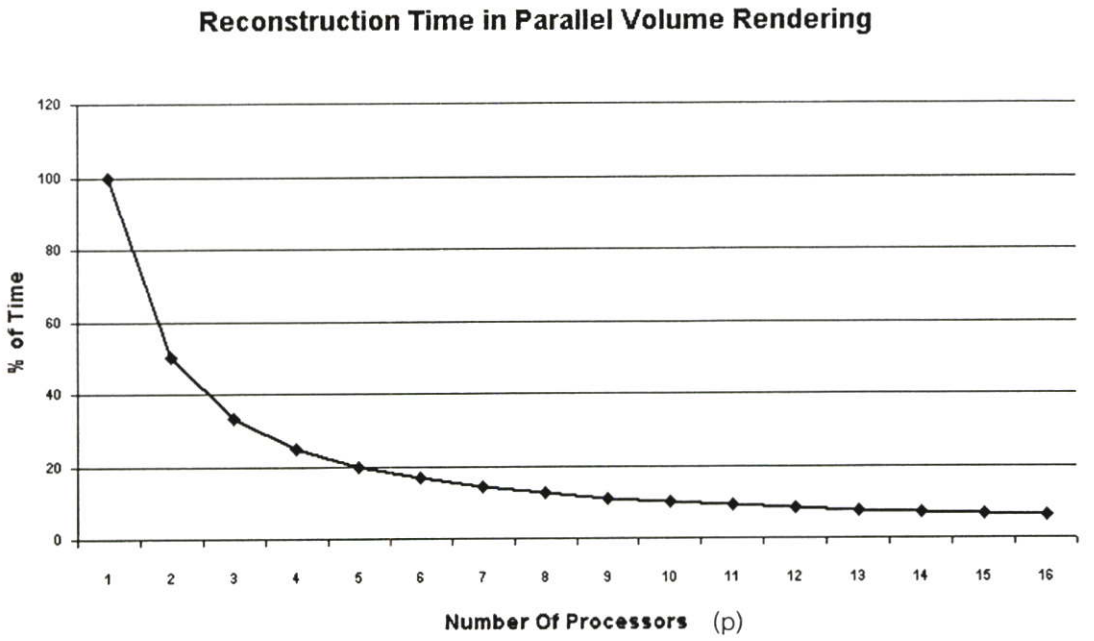
T_1 เวลาที่ใช้ในการสร้างภาพบนเครื่องหนึ่งเครื่องด้วยอัลกอริทึมแบบลำดับ

T_p เวลาที่ใช้ในการสร้างภาพบนเครื่องจำนวน p เครื่องด้วยอัลกอริทึมแบบขนานบนระบบคลัสเตอร์

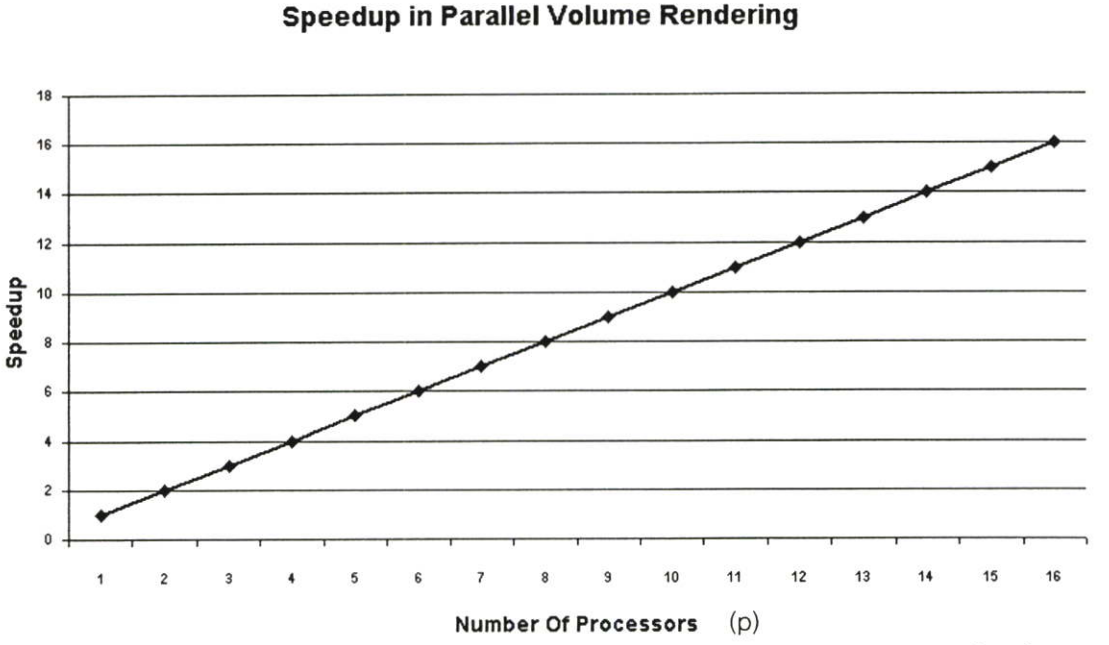
กราฟในอุดมคติของอัตราการเพิ่มความเร็วในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ แสดงดังรูปที่ 6.2 และค่าสุดท้ายสำหรับการวัดประสิทธิภาพคือค่าประสิทธิภาพ (Efficiency) ของการสร้างภาพได้ดังสมการ 6.4

$$E_p = \frac{S_p}{p} = \frac{T_1}{pT_p} = 1 \tag{6.4}$$

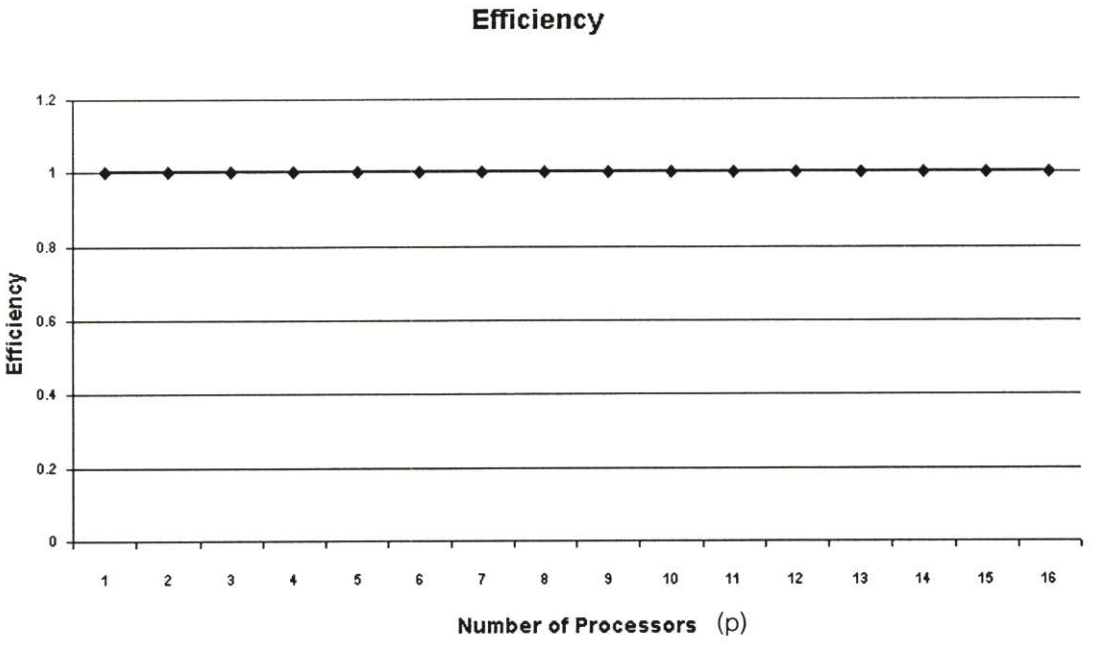
กราฟในอุดมคติของประสิทธิภาพในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ แสดงดังรูปที่ 6.3



รูปที่ 6.1 กราฟในอุดมคติของเวลาที่ใช้ในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์



รูปที่ 6.2 กราฟในอุดมคติของอัตราการเพิ่มความเร็วในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์



รูปที่ 6.3 กราฟในอุดมคติของประสิทธิภาพในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์

6.3 ผลของการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์

การทดลองส่วนแรกเป็นการสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์ของภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่นซึ่งเป็นระบบคลัสเตอร์ที่สร้างจากเครื่องคอมพิวเตอร์ส่วนบุคคล จำนวน 16 เครื่องโดยในแต่ละเครื่องประกอบด้วยหน่วยประมวลผลกลาง รุ่น P-III 550 MHz. หน่วยความจำ 128 MB. และใช้ระบบปฏิบัติการ GNU/Linux Slackware ซึ่งใช้ Kernel รุ่น 2.4.18 และ GNU C Compiler รุ่น 2.91.66

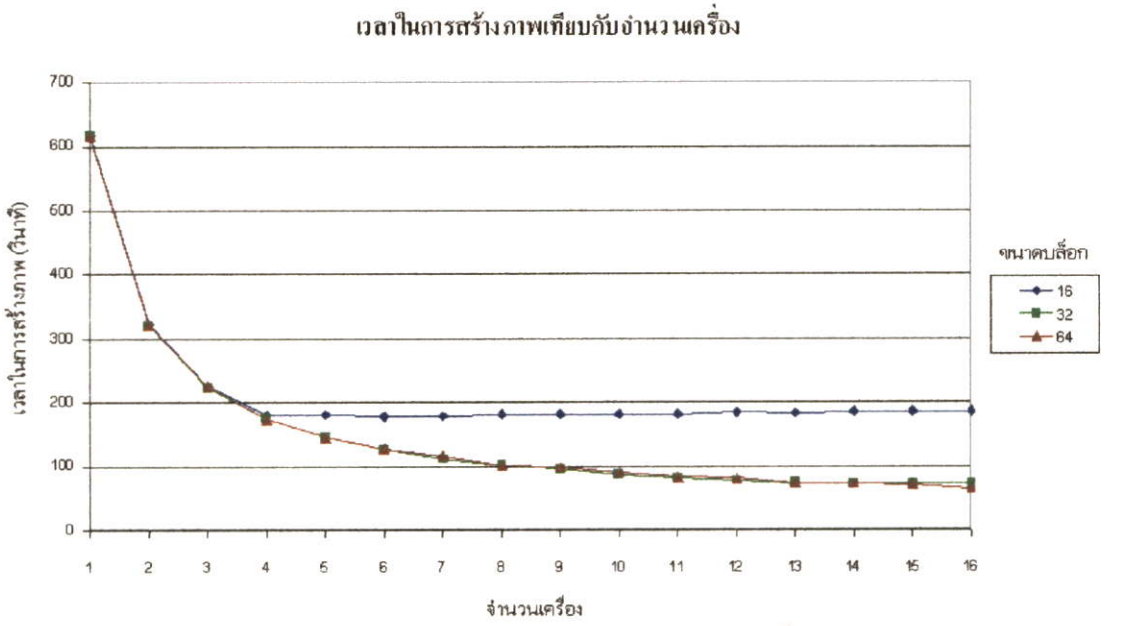
ภาพที่ใช้ในการทดลองเป็นภาพถ่ายร่างกายมนุษย์ที่ได้จากเครื่อง CT ของโครงการ Visible Human Project [6] ซึ่งมีขนาด 587x341x500 จุดภาพ ผลของเวลาที่ใช้ในการสร้างภาพแสดงดังตารางที่ 6.1 กราฟของเวลาที่ใช้ในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ของข้อมูลภาพขนาด 587x341x500 แสดงดังรูปที่ 6.4จากกราฟในรูปที่ 6.4 จะเห็นได้ว่าเมื่อกำหนดให้บล็อกของกลุ่มงานย่อยมีขนาด 16x16 พิกเซลจะใช้เวลาในการสร้างภาพมากกว่าการให้บล็อกของกลุ่มงานย่อยที่มีขนาดใหญ่กว่า 32x32 พิกเซลขึ้นไป ซึ่งเมื่อใช้จำนวนเครื่องมากกว่าสี่เครื่องในการประมวลผลจะทำให้เวลาที่ใช้ในการสร้างภาพเพิ่มมากขึ้น เนื่องจากเมื่อจำนวนบล็อกเล็กลงจะทำให้มีจำนวนงานที่ต้องทำการประมวลผลมากขึ้น ซึ่งจะส่งผลให้เกิดการสื่อสารในระบบมากขึ้นทำให้ต้องใช้เวลาในการสื่อสารมากขึ้นตามไปด้วย เวลาที่ใช้ในการประมวลผลจะเกิดจากเวลาที่ใช้ในการคำนวณเพื่อสร้างพิกเซลของภาพหนึ่งบล็อกค่อนข้างน้อยเมื่อเทียบกับเวลาที่ใช้ในการส่งข้อมูลของแต่ละบล็อก ส่วนบล็อกย่อยของงานที่มีขนาดใหญ่กว่า

32x32 พิกเซล เวลาที่ใช้ในการสร้างภาพจะไม่ต่างกันมากนัก เนื่องจากมีจำนวนงานไม่มาก และเวลาส่วนใหญ่ที่ใช้ในการประมวลผลถูกใช้ไปในการสร้างจุดพิกเซลของภาพ ไม่ใช่เวลาที่ใช้ในการส่งข้อมูล

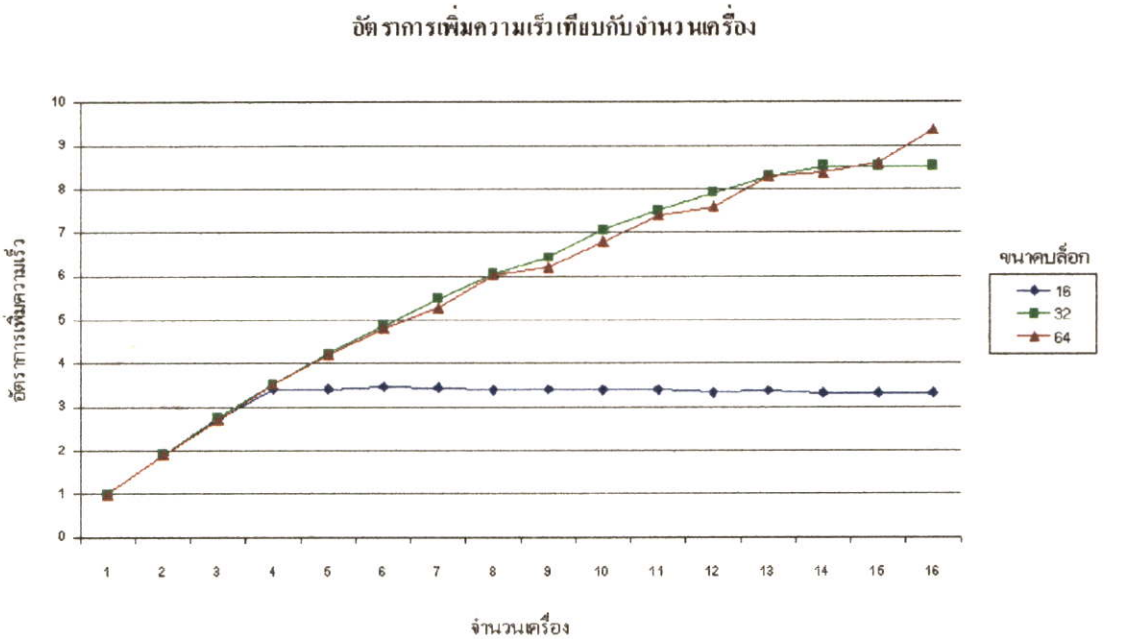
จากข้อมูลในตารางที่ 6.1 สามารถสรุปได้ว่า เมื่อให้บล็อกข้อมูลย่อยมีขนาด 16x16 พิกเซล เมื่อใช้เครื่องในการประมวลผลมากกว่า 4 เครื่องเวลาที่ใช้ในการประมวลผลไม่ได้ลดลง ซึ่งสำหรับ บล็อกข้อมูลขนาด 16x16 พิกเซลนี้ สามารถลดเวลาในการสร้างภาพลงได้ ประมาณ 70% ส่วนบล็อกข้อมูลย่อยขนาด 32x32 และ 64x64 พิกเซลสามารถลดเวลาในการประมวลผลได้ใกล้เคียงกันคือ 88% ที่จำนวนเครื่อง 16 เครื่อง

ตารางที่ 6.1 ผลของเวลาที่ใช้ในการสร้างภาพเชิงปริมาตรของข้อมูลขนาด 587x341x500 จุดภาพ

จำนวนเครื่องที่ใช้	เวลาที่ใช้ในการสร้างภาพ (วินาที)		
	กลุ่มงานขนาด16x16จุด	กลุ่มงานขนาด32x32จุด	กลุ่มงานขนาด64x64จุด
1	618.07	617.27	616.43
2	323.11	321.73	321.90
3	226.09	223.57	227.44
4	181.21	175.13	174.48
5	180.91	145.77	146.34
6	177.91	126.30	128.00
7	178.93	112.45	116.74
8	181.88	101.86	102.17
9	181.75	95.83	99.33
10	181.85	87.33	90.67
11	182.03	82.14	83.33
12	185.06	77.91	81.14
13	183.3	74.40	74.38
14	185.9	72.44	73.60
15	185.95	72.42	71.67
16	186.03	72.42	65.71



รูปที่ 6.4 กราฟของเวลาที่ใช้ในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ของข้อมูลภาพขนาด 587x341x500 จุดภาพ

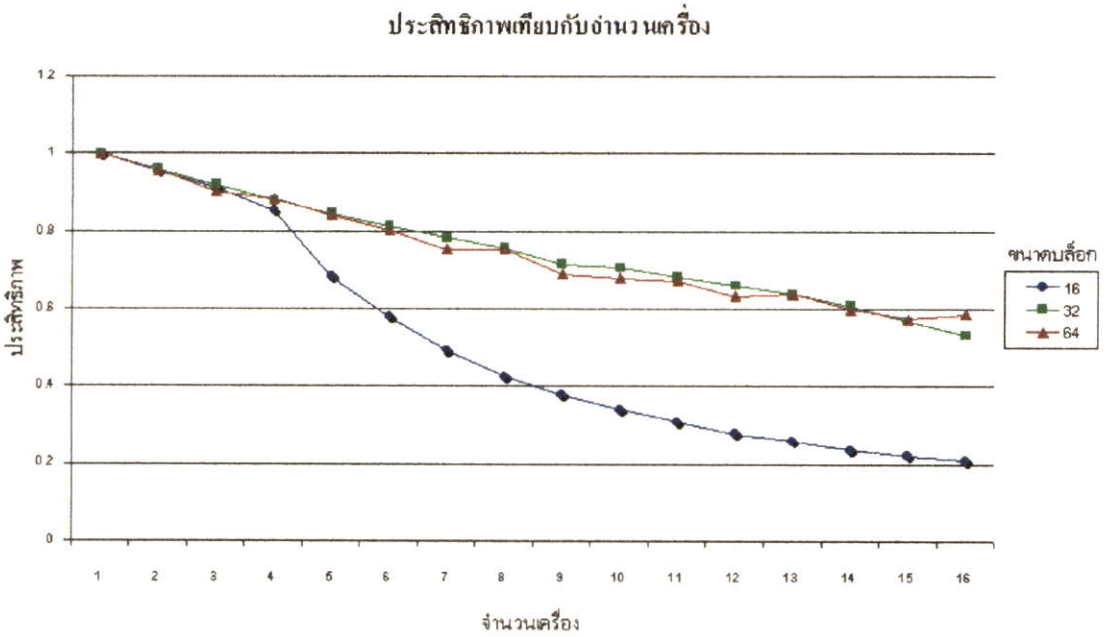


รูปที่ 6.5 กราฟของประสิทธิภาพในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ของข้อมูลภาพขนาด 587x341x500 จุดภาพ

กราฟของอัตราการเพิ่มความเร็วในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ของข้อมูลภาพขนาด 587x341x500 จุดภาพแสดงดังรูปที่ 6.5 จากกราฟจะเห็นได้ชัดว่าเมื่อสร้างภาพโดยใช้บล็อกข้อมูลย่อยขนาด 16x16 พิกเซล และใช้เครื่องมาก

กว่าสี่เครื่องความเร็วของการสร้างภาพไม่ได้เร็วขึ้นซึ่งเนื่องมาจากเหตุผลที่กล่าวมาแล้วในตอนต้น ส่วนบล็อกข้อมูลขนาดใหญ่กว่า เช่น 32x32 และ 64x64 พิกเซลเมื่อจำนวนเครื่องที่ใช้ในการประมวลผลไม่มากกราฟของอัตราการเพิ่มความเร็วจะใกล้เคียงกับกราฟในอุดมคติ แต่เมื่อจำนวนเครื่องในระบบมากขึ้นจะทำให้อัตราการเพิ่มความเร็วลดลงเนื่องจากเสียเวลาในการสื่อสารมากขึ้น

กราฟของประสิทธิภาพในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบ คลัสเตอร์ของข้อมูลภาพขนาด 587x341x500 จุดภาพแสดงดังรูปที่ 6.6 จากกราฟจะเห็นประสิทธิภาพของการประมวลผลเมื่อลดลงเมื่อใช้จำนวนเครื่องในระบบมากขึ้นได้อย่างชัดเจน



รูปที่ 6.6 กราฟของประสิทธิภาพในการสร้างภาพเทียบกับจำนวนเครื่องที่ใช้ในการประมวลผลบนระบบคลัสเตอร์ของข้อมูลภาพขนาด 587x341x500 จุดภาพ

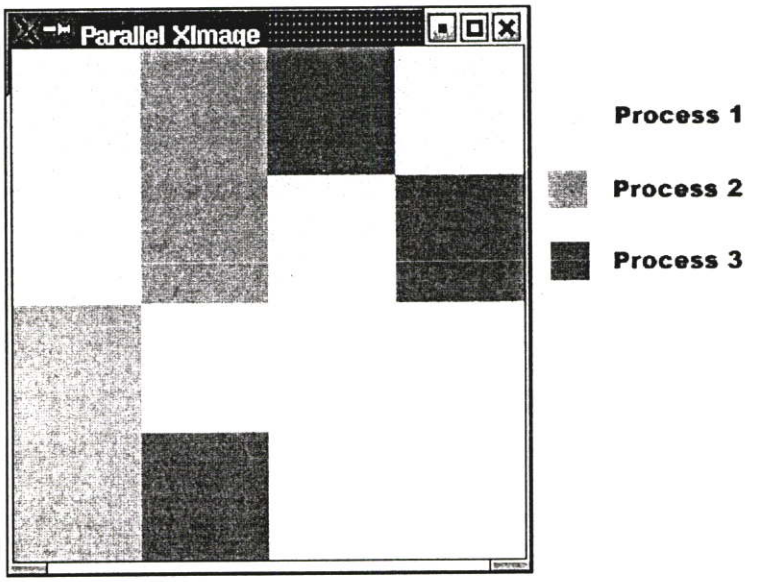
ผลของการสร้างภาพของปริมาตรภาพขนาด 587x341x500 จุดภาพแสดงดังรูปที่ 6.11

รูปที่ 6.7 ผลของการสร้างภาพของปริมาตรภาพขนาด 587x341x500 จุดภาพ

6.4 ผลของการจำลองการกำหนดงานให้แก่เครื่องในระบบคลัสเตอร์ด้วยวิธีการที่นำเสนอ

การกำหนดงานในระบบคลัสเตอร์เป็นกระบวนการหนึ่งที่สลับซับซ้อน เนื่องจากเป็นการยากที่จะคำนวณความสามารถในการประมวลผลของงานที่จะกำหนดให้เพื่อที่จะหาโพรเซสที่เหมาะสมเพื่อจะกำหนดงานให้ การกำหนดงานอย่างเหมาะสมจะช่วยให้ระบบคลัสเตอร์สามารถประมวลผลได้อย่างมีประสิทธิภาพ งานจะไม่ไปค้างอยู่บนเครื่องที่มีความสามารถในการประมวลผลต่ำในขณะที่อาจมีบางเครื่องในระบบว่างงาน ในวิธีการที่นำเสนอให้โพรเซสที่ว่างงานมารับงานจากศูนย์รวมงานไปทำเมื่อว่างงาน แทนที่จะกำหนดงานให้แต่ละโพรเซสตั้งแต่แรก

การจำลองการกำหนดงานสามารถทำได้โดยการเขียนโปรแกรมกำหนดให้แต่ละโพรเซสทำการประมวลผลไม่เท่ากัน โดยโพรเซสที่ 1 (Process 1) จะทำงานน้อยที่สุดซึ่งคล้ายกับว่าเครื่องที่โพรเซสที่ 1 ทำงานอยู่มีความสามารถในการประมวลผลมากที่สุด ซึ่งจะทำการวนซ้ำเป็นจำนวน 8.0×10^{7} รอบ โพรเซสที่ 2 ทำการวนซ้ำจำนวน 1.6×10^{8} รอบ และโพรเซสที่ 3 (Process 3) เสมือนว่าเป็นโพรเซสที่ทำงานอยู่บนเครื่องที่มีความสามารถในการประมวลผลต่ำที่สุดโดยวนซ้ำ 2.4×10^{8} รอบ เมื่อแต่ละโพรเซสทำงานเสร็จจะทำการแสดงสีประจำตัวของโพรเซสออกมาแสดงดังจอภาพ ซึ่งได้ผลการทดลองดังรูปที่ 6.8



รูปที่ 6.8 ผลการจำลองการกำหนดงานบนระบบคลัสเตอร์

จากรูปที่ 6.8 บล็อคสีต่างๆเป็นตัวแทนของงานที่ถูกประมวลผลบนโพรเซสต่างๆกันโดยโพรเซสที่ 1 เป็นโพรเซสที่สามารถทำงานได้เร็วที่สุดดังนั้นควรจะได้รับงานไปทำมากที่สุดซึ่งควรจะมีบล็อกสีของโพรเซสที่ 1 เกิดขึ้นมากที่สุด โพรเซสที่สองควรจะได้รับงานน้อยกว่าโพรเซสที่ 1 ก็ควรจะมีบล็อกสีของโพรเซสที่ 2 น้อยกว่าโพรเซสที่ 1 และโพรเซสที่ 3 ควรจะได้รับงานไปทำน้อยที่สุดซึ่งควรจะเกิดบล็อกสีของโพรเซสที่ 3 น้อยที่สุด

จากผลการทดลองมีบล็อกสีที่แสดงถึงการถูกประมวลผลบนโพรเซส 1 จำนวน 9 บล็อก บล็อกสีที่แสดงถึงการถูกประมวลผลบนโพรเซสที่ 2 จำนวน 4 บล็อก และ โพรเซสที่ 3 จำนวน 3 บล็อก ซึ่งเป็นไปตามที่คาดหวังไว้

บทที่ 7

สรุปผลและแนวทางการพัฒนา

7.1 สรุปและวิจารณ์ผลการทดลอง

ระบบคลัสเตอร์จัดเป็นระบบคอมพิวเตอร์ขนานแบบใหม่ซึ่งกำลังมีการพัฒนาเพื่อนำไปประมวลผลงานที่ต้องใช้การคำนวณมากๆเช่น การทำนายสภาพภูมิอากาศ การออกแบบทางวิศวกรรม เป็นต้น และจากวิทยานิพนธ์ฉบับนี้ได้ประยุกต์ใช้ระบบคลัสเตอร์เพื่อพัฒนาโปรแกรมแบบขนานในการสร้างภาพเชิงปริมาตร เพื่อเพิ่มความเร็วในการสร้างภาพของแต่ละมุมมองได้

หลักการของการประมวลผลแบบขนานคือ ทำการแบ่งปัญหาออกเป็นส่วนย่อยแล้วทำการแก้ปัญหาย่อยนั้นไปพร้อมๆกัน ซึ่งวิธีการแบ่งปัญหาออกเป็นส่วนย่อยสามารถทำได้สองวิธีคือ การแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อย (Domain Decomposition) และการแบ่งหน้าที่ของปัญหาออกเป็นส่วนย่อย (Functional Decomposition) ซึ่งในวิทยานิพนธ์ฉบับนี้ใช้การแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อย ข้อมูลย่อยของการสร้างภาพเชิงปริมาตรคือข้อมูลของปริมาตรภาพตามแนวลำแสงซึ่งถูกฉายในกระบวนการฉายแสง การแบ่งงานสามารถทำได้ละเอียดที่สุดคือแบ่งหนึ่งงานต่อการฉายแสงหนึ่งลำแสง การแบ่งงานที่ละเอียดมากเกินไปอาจทำให้เกิดการสื่อสารในระบบคลัสเตอร์ที่มากเกินจำเป็นซึ่งอาจทำให้ประสิทธิภาพในการประมวลผลลดลง เพราะเวลาในการทำงานส่วนใหญ่ถูกใช้ไปกับการสื่อสาร ซึ่งการสื่อสารในระบบคลัสเตอร์มักเป็นปัญหาคอขวด (Bottle Neck) ของระบบ เปรียบเทียบกับระบบคอมพิวเตอร์ขนานแบบหลายหน่วยประมวลผล (Multi-Processors) การสื่อสารระหว่างหน่วยประมวลผลจะทำผ่านระบบบัส (Bus) ซึ่งมีความเร็วในการสื่อสารมากกว่าระบบเครือข่ายหลายเท่าตัว ดังนั้นการออกแบบระบบคลัสเตอร์สำหรับงานที่ต้องมีการสื่อสารกันระหว่างโพรเซสมากๆ จะต้องเลือกใช้ระบบเครือข่ายที่มีความเร็วสูงเช่น Myrinet หรือ Gigabit Network เป็นต้น หรืออาจจัดให้โพรเซสที่ต้องสื่อสารกันบ่อยๆทำงานอยู่ในหน่วยประมวลผลเดียวกัน ส่วนโพรเซสที่สื่อสารกันน้อยให้อยู่ในหน่วยประมวลผล ต่างกัน ซึ่งในวิทยานิพนธ์ฉบับนี้ได้ทำการรวมกลุ่มงานของการฉายแสงออกเป็นกลุ่มซึ่งจากการทดลองได้ลองแบ่งกลุ่มขนาด 16x16 แนวลำแสง 32x32 แนวลำแสง และ 64x64 แนวลำแสงซึ่งพบว่าถ้าแบ่งงานเล็กกว่า 32x32 แนวลำแสงจะทำให้ประสิทธิภาพในการประมวลผลลดลง และจากผลการทดลองพบว่าสามารถลดเวลาในการสร้างภาพได้มากถึง 88 เปอร์เซ็นต์

การกำหนดงานในระบบคลัสเตอร์จะถูกทำเพื่อกำหนดให้งานย่อยที่ทำการแบ่งไว้ไปทำงานบนโพรเซสที่กำหนด การกำหนดงานที่เหมาะสมจะช่วยให้ระบบสามารถประมวลผลได้อย่างมีประสิทธิภาพและทำงานเสร็จภายในเวลาที่ระบบสามารถทำได้ขณะนั้นแต่การกำหนดงานที่

ไม่เหมาะสมอาจทำให้งานเสร็จช้า และใช้ทรัพยากรของระบบอย่างไม่คุ้มค่า เช่นการกำหนดงานหลายๆงานไปทำการประมวลผลบนโพรเซสที่ทำงานอยู่บนเครื่องมีความสามารถในการประมวลผลต่ำ ในขณะที่บางโพรเซสที่ทำงานอยู่บนเครื่องที่มีความสามารถในการประมวลผลสูงกว่าได้รับงานไปทำน้อย การแก้ปัญหานี้สามารถทำได้ด้วยการสมดุลโหลด (Load Balancing) ซึ่งเป็นปัญหาที่แก้ไขได้ยากอย่างหนึ่งในการประมวลผลแบบขนาน ในวิทยานิพนธ์ฉบับนี้ใช้วิธีกำหนดงานแบบศูนย์รวมงาน (Work Pool) ซึ่งอาศัยหลักการที่ว่าให้โพรเซสที่ว่างงานไปรับงานจากศูนย์รวมงานเองแทนที่จะกำหนดงานให้แก้โพรเซสตั้งแต่แรก

7.2 แนวทางการพัฒนา

ถึงแม้ว่าการประมวลผลแบบขนานบนระบบคลัสเตอร์เป็นวิธีเพิ่มความเร็วของการประมวลผลที่มีประสิทธิภาพมากวิธีหนึ่ง แต่ระบบคลัสเตอร์ก็ยังมีราคาแพงมากเมื่อเทียบกับการประมวลผลบนเครื่องคอมพิวเตอร์หนึ่งเครื่อง การประมวลผลงานที่ไม่ได้มีการปรับปรุงอัลกอริทึมที่ดีบนระบบคลัสเตอร์ อาจให้ความเร็วในการประมวลผลพอๆกับการประมวลผลที่มีการปรับปรุงอัลกอริทึมการทำงานมาอย่างดีบนเครื่องคอมพิวเตอร์หนึ่งเครื่อง การใช้อัลกอริทึมที่ดีร่วมกับการประมวลผลแบบขนานบนระบบคลัสเตอร์จะช่วยเพิ่มประสิทธิภาพในการประมวลผลให้มากขึ้นไปอีก เช่นนำวิธีการสร้างภาพเชิงปริมาตรทางการแพทย์โดยใช้การแปลงระยะทาง [21] มาปรับปรุงเป็นอัลกอริทึมแบบขนานเพื่อสร้างภาพเชิงปริมาตรบนระบบคลัสเตอร์ น่าจะทำให้สามารถสร้างภาพเชิงปริมาตรได้เร็วขึ้น

เอกสารอ้างอิง

[1] Gregory F. Pfister. In Search of Clusters. 2nd Edition. NJ. Prentice Hall PTR. 1998.

[2] Lorensen W. E., Cline H. E. "Marching Cubes : A High Resolution 3D Surface Construction Algorithm." Computer Graphics (Proc. SIGGRAPH)., vol. 21, No.4, July 1987, pp. 163-169.

[3] Levoy M. "Display of Surface from Volume Data." IEEE Computer Graphics & Applications., vol.8, No.3, May 1998, pp. 29-37

[4] Foley D. James. et. al. Computer Graphics Principles and Practice. 2nd Edition in C. Addison-Wesley. 1996.

[5] Lichtenbelt B. et. al. Introduction to Volume Rendering. Hewlett-Packard Press. 1998.

[6] National Library of Medicine, "The Visible Human Project" [Online] Available: http://www.nlm.nih.gov/research/visible/visible_human.html

[7] David A. Clunie. "Medical Image Format FAQ." [Online] Available: http://www.dclunie.com/medical-image-faq/html/

[8] ประเสริฐ อัศวรุ่งสกุล "การแยกแยะภาพสามมิติของ MRI ส่วนสมองโดยฟัซซี่ซีมีนแบบประมาณค่า" วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2544.

[9] วรเทพ ไพบูลย์รัตนากร และคณะ "อัลกอริธึมสำหรับสร้างภาพเชิงปริมาตรทางการแพทย์" การประชุมวิชาการทางวิศวกรรมไฟฟ้าครั้งที่ 22 มหาวิทยาลัยเกษตรศาสตร์ 2542 หน้า 369-372

[10] Buyya R. High Performance Cluster Computing: Architectures and Systems, Volume 1. Prentice Hall PTR. 1999

[11] Beowulf Project. "Beowulf Cluster." [Online] Available: http://www.beowulf.org

[12] MPICH Project. "MPICH-A Portable Implementation of MPI." [Online] Available: http://www-unix.mcs.anl.gov/mpi/mpich/

[13] MOSIX Project. "MOSIX." [Online] Available: http://www.mosix.cs.huji.ac.il

[14] The Parallel Virtual Machine (PVM). "Parallel Virtual Machine." [Online] Available: http://www.epm.ornl.gov/pvm/pvm_home.html

[15] I. Foster. "Designing and Building Parallel Program." [Online] Available: http://www-unix.mcs.anl.gov/dbpp/

[16] Gropp W. et. al. Using MPI : Portable Parallel Programming with the Message-Passing Interface. The MIT Press. 1994.

[17] Gropp W. et. al. MPI : The Complete Reference. MIT Press. 1994.

[18] MPI Tutorial. "Message Passing Interface (MPI)." [Online] Available: http://www.llnl.gov/computing/tutorials/workshops/workshop/mpi/MAIN.html

[19] Wilkinson B., Allen M. Parallel Programming: Technique and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 1999.

[20] บำรุง สมสวัสดิ์ และคณะ 2545 "แบบจำลองการทำงานเซลล์สุริยะแบบคลับแซนวิชใน 2 มิติ (ระยะ 1B)" รายงานโครงการวิจัย สำนักงานกองทุนสนับสนุนการวิจัย.

[21] วรเทพ ไพบูลย์รัตนากร "การเพิ่มความเร็วให้การสร้างภาพเชิงปริมาตรทางการแพทย์โดยใช้การแปลงระยะทาง." วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2544.

ภาคผนวก

ผลงานวิจัยที่ได้รับการตีพิมพ์

1. วรเทพ ไพบูลย์รัตนากร นนท์ บัณฑิตวงษ์ สุรพันธุ์ เอื้อไพบูลย์ และ มนัส สังวรศิลป์ "การเพิ่มความเร็วให้การสร้างภาพเชิงปริมาตรทางการแพทย์ด้วยการแปลงระยะทาง" การประชุมวิชาการทางไฟฟ้าครั้งที่ 23 มหาวิทยาลัยเชียงใหม่ พ.ศ. 2543 หน้า 449-452
2. Paiboonratanakorn, W., Banditwong, N., Sangworasilp, M., Kusirivanitchakorn, P., "Three Planes Contour Technique for 3-D Medical Image Rendering" Proc. IEEE-ISPACS, Phuket, Thailand, Dec 1999,pp.681-684.
3. นนท์ บัณฑิตวงษ์ "การเร่งความเร็วในการสร้างภาพสามมิติทางการแพทย์โดยใช้ PC-CLUSTER" การประชุมเสนอผลงานวิจัยระดับบัณฑิตศึกษาของประเทศไทย ครั้งที่ 2 คณะวิทยาศาสตร์ มหาวิทยาลัยมหิดล พ.ศ. 2544 หน้า 79
4. นนท์ บัณฑิตวงษ์ เชษฐ์ ทัดทอง และ มนัส สังวรศิลป์ "เทคนิคการสร้างภาพสามมิติทางการแพทย์ของอวัยวะภายในร่างกายจากข้อมูลที่มีความสว่างมากที่สุด" การประชุมวิชาการทางไฟฟ้าครั้งที่ 24 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ. 2544 หน้า 1318-1322

ประวัติผู้เขียน

นายนนท์ บัณฑิตวงษ์ เกิดเมื่อวันที่ 10 พฤศจิกายน 2518 ที่จังหวัดขอนแก่น สำเร็จการศึกษาปริญญาตรีทางด้าน วิทยาศาสตร์ สาขา ฟิสิกส์ จากมหาวิทยาลัยขอนแก่น ปีการศึกษา 2540 มกราคม ปี 2544 ถึง มีนาคม 2545 ทำงานที่บริษัทพรีเมียร์ ทีดีโอจำกัด ในตำแหน่ง นักวิจัย สิงหาคม ปี 2545 ถึง ปัจจุบัน ทำงานที่บริษัทอินเทอร์เน็ตประเทศไทย จำกัด มหาชน ในตำแหน่ง โปรแกรมเมอร์