

การแบ่งส่วนภาพโดยทฤษฎีกราฟของภาพจากการต่อแบบ  
หลายความละเอียดผ่านการรักษาขอบแบบปรับเรียบ

SEGMENTATION VIA GRAPH THEORY ON THE EDGE PRESERVING  
SMOOTHING OF MULTIREOLUTION MOSAICED IMAGE

พรพจน์ ไพธพงษ์วิวัฒน์  
PHORNPHOT PHOPHONGVIWAT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2545

ISBN 974-954-666-0

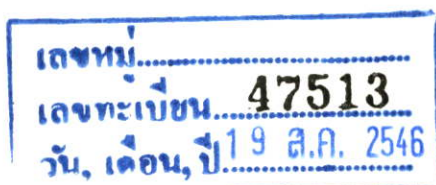
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การแบ่งส่วนภาพโดยทฤษฎีกราฟของภาพจากการต่อแบบหลายความละเอียด  
ที่ผ่านการรักษาขอบแบบปรับเรียบ

SEGMENTATION VIA GRAPH THEORY ON THE EDGE PRESERVING  
SMOOTHING OF MULTIREOLUTION MOSAICED IMAGE

พรพจน์ โพธิ์พงษ์วิวัฒน์

PHORNPOT PHOPHONGVIWAT



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2545

ISBN 974-954-666-0

**SEGMENTATION VIA GRAPH THEORY ON THE EDGE PRESERVING  
SMOOTHING OF MULTIREOLUTION MOSAICED IMAGE**

**PHORNPHOT PHOPHONGVIWAT**

**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 974-954-666-0**

**COPYRIGHT 2002**

**SCHOOL OF GRADUATE STUDIES**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

หัวข้อวิทยานิพนธ์	การแบ่งส่วนภาพโดยทฤษฎีกราฟของภาพจากการต่อแบบหลาย ความละเอียดที่ผ่านการรักษาขอบแบบปรับเรียบ
นักศึกษา	นาย พรพจน์ โพธิ์พงษ์วิวัฒน์
รหัสประจำตัว	40061006
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2545
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. ฟุ่กคี่ ชีวสุวิทย์

### บทคัดย่อ

การต่อภาพถ่ายดาวเทียมมีประโยชน์และจำเป็นอย่างมากในการวิเคราะห์บริเวณที่มี  
เมฆหมอกปกคลุมตัวอย่างเช่นการติดตามไฟป่า การร่อนภาพถ่ายจากดาวเทียมเพียงดวงเดียว  
อาจจะไม่สามารถนำมาใช้ได้ทันเหตุการณ์ ดังนั้นภาพถ่ายความละเอียดต่างกันที่ได้จากดาวเทียม  
ดวงอื่น ซึ่งถ่ายได้ในเวลาใกล้เคียงกันสามารถนำมาทดแทนพื้นที่ดังกล่าวได้ โดยการรวมภาพ  
หลายความละเอียดที่ได้มาจากดาวเทียมหลายดวงเข้าด้วยกัน ซึ่งจำเป็นต้องมีการชักตัวอย่างใหม่  
ให้ได้ความละเอียดเท่ากันก่อนต่อเข้าด้วยกัน ภาพที่ถูกต่อจะนำมาทำการปรับปรุงให้ขอบเขต  
ของพื้นที่ย่อยส่วนต่างๆ ชัดเจน และมีความเป็นเนื้อเดียวกันด้วยการรักษาขอบแบบปรับเรียบ  
ซึ่งมีการนำเสนอเทรรมเพลทแบบใหม่ ที่จะทำให้สามารถรักษาขอบเขตที่ผอมบางขนาดหนึ่งจุดภาพได้  
พร้อมกับกำจัดสัญญาณรบกวนไปได้ด้วยในตัว แล้วจึงนำไปแบ่งส่วนด้วยทฤษฎีกราฟที่มีจุดเด่น  
คือสามารถกำหนดจำนวนขอบเขตที่ต้องการ ทำให้ได้ภาพซึ่งมีขอบเขตที่ชัดเจนตามจำนวนที่  
ต้องการ ง่ายต่อการนำภาพไปวิเคราะห์เพื่อแปลความหมายจากภาพต่อไป

<b>Thesis Title</b>	Segmentation via Graph Theory on the Edge Preserving Smoothing Multiresolution Mosaiced Image
<b>Student</b>	Mr. Phornphot Phophongviwat
<b>Student ID.</b>	40061006
<b>Degree</b>	Master of Engineerring
<b>Programme</b>	Electrical Engineering
<b>Year</b>	2002
<b>Thesis Advisor</b>	Assoc. Prof. Dr. Fusak Cheevasuvit

#### **ABSTRACT**

Mosaicing of satellite image are very necessary and usefully for the cloud cover problem. For some task, such as for forest fire tracking, the complete scene image are required, but sometime the other orbits of the same satellite is too late for the use. Hence, the images with different resolutions from another satellite which acquired in contiguous time will be replaced the cloud covered regions. To merge the different solution images together, the resampling process must be applied. The obtained mosaiced image is already removed the noise pixels without blurring edges and also unified the common regions in the connecting boundaries. To detect the boundaries of the interested region, the graph theory will be applied onto the smoothed mosaiced image for obtaining accurate region boundaries.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ดี ด้วยคำแนะนำและคำปรึกษาจาก รศ.ดร. พุศศักดิ์ ชิวสุวิทย์ อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ ผศ. สักกริยา ชิตวงศ์ ที่คอยให้คำปรึกษาและชี้แนะ ในขั้นตอนการพัฒนาโปรแกรม จนทำให้ได้โปรแกรมเพื่อใช้งานในขั้นตอนต่างๆ

ขอขอบคุณเพื่อนๆ พี่ๆ และน้องๆ ทุกคนที่มีส่วนช่วยเหลือในการค้นหารูปภาพ โปรแกรมประยุกต์ทางคณิตศาสตร์ เอกสารอ้างอิงต่างๆ ที่เป็นประโยชน์ได้นำมาใช้งานและไม่ได้ถูกใช้งาน วิจัยณ์-ดิชมผลการทดลอง จนมีข้อมูลมากมายในการเรียบเรียงวิทยานิพนธ์

ท้ายที่สุดนี้ขอกราบขอบพระคุณคุณพ่อและคุณแม่ที่คอยกระตุ้นเตือน ให้ผู้เขียนมีสมาธิ มีกำลังใจ จนสามารถเขียนวิทยานิพนธ์ฉบับนี้ให้สำเร็จลุล่วงได้ด้วยดี

พรพจน์ โพธิ์พงษ์วิวัฒน์

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	V
สารบัญภาพ.....	VI
บทที่ 1 บทนำ.....	1
1.1 คำนำ.....	1
1.2 วัตถุประสงค์ของวิทยานิพนธ์.....	1
1.3 ขอบเขตการวิจัย.....	2
บทที่ 2 การต่อภาพแบบหลายความละเอียด.....	3
2.1 คำนำ.....	3
2.2 การแก้ความผิดพลาดรูปทรงเรขาคณิต.....	4
2.3 การชักตัวอย่างข้อมูลและการแทรกค่าข้อมูล.....	11
2.4 การปรับปรุงฮิสโตแกรมของภาพ.....	14
2.5 การปรับความสว่าง.....	16
2.6 การต่อภาพแบบสหสัมพันธ์.....	18
2.8 สรุปผลการทดลอง.....	21
บทที่ 3 การรักษาขอบแบบปรับเรียบ.....	22
3.1 คำนำ.....	22
3.2 เทรนเพลทที่ใช้ในการปรับเรียบ.....	22
3.3 กระบวนการรักษาขอบแบบปรับเรียบ.....	25
3.4 ผลการทดลอง.....	27
3.5 สรุปผลการทดลอง.....	28
บทที่ 4 การแบ่งส่วนภาพโดยทฤษฎีกราฟ.....	29
4.1 คำนำ.....	29
4.2 การจัดข้อมูลภาพให้อยู่ในลักษณะกราฟ.....	29

## สารบัญ (ต่อ)

	หน้า
4.3 การหาซื้อทดสอบที่สเปนนิ่งทรี.....	31
4.4 การเข้ารหัสภาพเซกเมนต์ชั้น.....	35
4.5 ผลการทดลอง.....	38
4.6 สรุปผลการทดลอง.....	42
<b>บทที่ 5</b> สรุปผลการวิจัยและข้อเสนอแนะ.....	43
5.1 สรุปผลการวิจัย.....	43
5.2 ปัญหาที่เกิดขึ้นและข้อเสนอแนะ.....	44
เอกสารอ้างอิง.....	45
ภาคผนวก ก ผลงานที่ได้รับการตีพิมพ์.....	46
ภาคผนวก ข โปรแกรมการรักษาขอบแบบปรับเรียบ.....	52
ประวัติผู้เขียน.....	63

# สารบัญตาราง

ตารางที่	หน้า
2.1 พิกัดของภาพที่จะแก้ความผิดพลาดเชิงเรขาคณิตและภาพอ้างอิงหน่วยเป็นพิกเซล.....	10
3.1 สถิติการถูกเลือกใช้งานของเทรมเพลทแบบต่างๆ.....	28

# สารบัญรูป

รูปที่	หน้า
2.1 จำลองภาพแม่น้ำบนพิกัดจริงเทียบกับภาพแม่น้ำที่ได้จากดาวเทียม.....	3
2.2 การเลือกจุดควบคุมบนพื้นดินก่อนการแก้ความเพี้ยนเชิงเรขาคณิต.....	5
2.3 ภาพจากดาวเทียม JERS-1 ความละเอียด 18 เมตร × 24 เมตร ขนาด 600 × 600 จุดภาพ.....	8
2.4 ภาพจากดาวเทียม MOS-1 ความละเอียด 50 × 50 เมตร ขนาด 256 × 256 จุดภาพ.....	9
2.5 ลักษณะการชักตัวอย่างและแทรกค่าข้อมูล.....	11
2.6 ผลลัพธ์ของการชักตัวอย่างและแทรกค่าข้อมูลของภาพที่ผ่านการแก้ความเพี้ยน เชิงเรขาคณิตของภาพในรูปที่ 2.3 จำนวนจุดภาพเปลี่ยนเป็น 228 × 285 จุดภาพ.....	14
2.7 การยัดฮิสโตแกรมด้วยสมการเชิงเส้น.....	15
2.8 ผลลัพธ์จากการกระจายฮิสโตแกรมของภาพในรูปที่ 2.6.....	16
2.9 ผลลัพธ์ของการปรับความสว่างของภาพในรูปที่ 2.8 ให้ใกล้เคียงกับรูปที่ 2.4.....	18
2.10 การหาดำแหน่งซ้อนทับของภาพ.....	19
2.11 ที่มาของข้อมูลจุดภาพใหม่.....	20
2.12 ผลลัพธ์ของการต่อภาพรูปที่ 2.9 เข้ากับรูปที่ 2.4.....	21
3.1 รูปแบบเทรมเพลททั้ง 5 รูปแบบของ Tomata.....	23
3.2 รูปแบบเทรมเพลททั้ง 9 รูปแบบของ Nagao.....	23
3.3 รูปแบบเทรมเพลทที่ออกแบบขึ้นใหม่.....	24
3.4 แผนภูมิการรักษารูปแบบเปรียบเทียบ.....	26
3.5 ภาพต้นแบบก่อนการเปรียบเทียบ.....	27
3.6 ภาพหลังจากผ่านการรักษารูปแบบเปรียบเทียบ.....	28
4.1 การจัดข้อมูลภาพให้อยู่ในลักษณะกราฟ.....	30
4.2 ค่าจุดยอดและน้ำหนักเส้นเชื่อมของกราฟที่สมบูรณ์ของภาพขนาด 4 × 4 จุดภาพ.....	30
4.3 ขั้นตอนการหาซื้อตัดเดสท์สแพนนิ่งทรีของกราฟ.....	33
4.4 แผนภูมิการหาซื้อตัดเดสท์สแพนนิ่งทรีของกราฟ.....	34
4.5 ทิศทางและค่ารหัสแบบ 4 ทิศทางและ 8 ทิศทาง.....	35
4.6 ตัวอย่างรหัสลูกโซ่ของเซกเมนต์แบบ 4 ทิศทาง.....	36
4.7 ตัวอย่างรหัสลูกโซ่ของเซกเมนต์แบบ 8 ทิศทาง.....	36
4.8 ข้อจำกัดของการใช้รหัสแบบ 8 ทิศทาง.....	37
4.9 ผลของการเข้ารหัสขอบแบบ 4 ทิศทาง เมื่อแบ่งภาพออกเป็น 1000 ส่วน ของภาพที่ไม่ผ่านการรักษารูปแบบ.....	38

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.10 ผลของการเข้ารหัสขอบแบบ 4 ทิศทาง เมื่อแบ่งภาพออกเป็น 500 ส่วน ของภาพที่ไม่ผ่านการรักษาขอบ.....	38
4.11 ผลการแบ่งส่วนภาพด้วยทฤษฎีกราฟ เมื่อแบ่งภาพออกเป็น 1000 ส่วน ของภาพที่ไม่ผ่านการรักษาขอบ.....	39
4.12 ผลการแบ่งส่วนภาพด้วยทฤษฎีกราฟ เมื่อแบ่งภาพออกเป็น 500 ส่วน ของภาพ ที่ไม่ผ่านการรักษาขอบ.....	39
4.13 ผลของการเข้ารหัสขอบแบบ 4 ทิศทาง เมื่อแบ่งภาพออกเป็น 1000 ส่วน ของภาพที่ผ่านการรักษาขอบแบบปรับเรียบ.....	40
4.14 ผลของการเข้ารหัสขอบแบบ 4 ทิศทาง เมื่อแบ่งภาพออกเป็น 500 ส่วน ของภาพที่ผ่านการรักษาขอบแบบปรับเรียบ.....	40
4.15 ผลการแบ่งส่วนภาพด้วยทฤษฎีกราฟ เมื่อแบ่งภาพออกเป็น 1000 ส่วน ของภาพที่ผ่านการรักษาขอบแบบปรับเรียบ.....	41
4.16 ผลการแบ่งส่วนภาพด้วยทฤษฎีกราฟ เมื่อแบ่งภาพออกเป็น 500 ส่วน ของภาพที่ผ่านการรักษาขอบแบบปรับเรียบ.....	41

# บทที่ 1

## บทนำ

### 1.1 คำนำ

ปัจจุบันเทคโนโลยีแขนงต่างๆ มีพัฒนาการไปอย่างรวดเร็ว โดยเฉพาะอุปกรณ์สารกึ่งตัวนำ และอิเล็กทรอนิกส์ ส่งผลให้ชิ้นส่วนของเครื่องมือเครื่องใช้ต่างๆ มีขนาดเล็กกลง แต่มีขีดความสามารถในการทำงานที่รวดเร็ว มีสมรรถนะในทุกด้านดีกว่าเก่า แต่ก็ส่งผลถึงราคาที่สูงมากขึ้นด้วย ดังนั้นอุปกรณ์เครื่องมือราคาแพงต่างๆ จึงควรได้ถูกใช้งานอย่างคุ้มค่าที่สุด

หลายหน่วยงานในหลายประเทศได้ส่งดาวเทียมขึ้นสู่วงโคจร เพื่อใช้สำรวจทรัพยากรธรรมชาติต่างๆ ตรวจสอบความเปลี่ยนแปลงของสภาพภูมิประเทศ ป่าไม้ ชายฝั่งทะเล รวมถึงใช้ในการสื่อสาร ดาวเทียมแต่ละดวงที่ถูกส่งขึ้นไปนั้นใช้ขยับในการสร้างจำนวนมหาศาล บางดวงมีอายุการใช้งานที่ถูกกำหนดโดยอุปกรณ์ที่นำมาประกอบ ดังนั้นข้อมูลทุกๆ ข้อมูลที่ดาวเทียมจะสามารถอำนวยความสะดวกได้ ควรจะถูกนำมาใช้งานให้เกิดประโยชน์สูงสุด

ท่ามกลางความแปรปรวนของสภาพลมฟ้าอากาศในปัจจุบัน ส่งผลให้ภาพถ่ายดาวเทียมจำนวนมาก ไม่สามารถนำมาใช้งานได้ทันที เนื่องจากมักจะมีสัญญาณรบกวนเกิดขึ้นในภาพ โดยเฉพาะประเทศที่อยู่ในแนวเส้นศูนย์สูตร ทั้งนี้เนื่องจากในภูมิประเทศดังกล่าวมักจะมีเมฆหมอกปกคลุมเป็นกลุ่มๆ อยู่ตลอดทั้งปี เมื่อเกิดวิกฤตการณ์ทางธรรมชาติที่ต้องได้รับการวิเคราะห์ปัญหาในทันที เช่นการเกิดพายุ อุทกภัย หรือการเกิดไฟป่า ภาพถ่ายดาวเทียมจากดาวเทียมหนึ่งดวงที่ถ่ายได้ในช่วงเวลาหนึ่ง อาจถูกเมฆปกคลุมบางส่วนของบริเวณที่ต้องการพิจารณา การรูดภาพถ่ายจากดาวเทียมดวงเดิมก็อาจไม่ทันต่อการแก้ไขสถานการณ์อันเลวร้ายที่เกิดขึ้นได้ ดังนั้นภาพถ่ายจากดาวเทียมดวงอื่นที่ถ่ายได้ในบริเวณและเวลาใกล้เคียงกันอาจนำมาช่วยในการวิเคราะห์ปัญหาได้ แต่ภาพถ่ายที่ได้จากดาวเทียมแต่ละดวงนั้นอาจมีความละเอียดของภาพไม่เท่ากัน ซึ่งขึ้นอยู่กับคุณสมบัติของดาวเทียมแต่ละดวงเอง ทำให้การวิเคราะห์ปัญหาจากภาพโดยเฉพาะอย่างยิ่ง การคำนวณหาพื้นที่ของบริเวณที่เป็นจุดสนใจ อาจทำได้ยากเนื่องจากบริเวณที่เป็นจุดสนใจอยู่ในพื้นที่คาบเกี่ยวกันระหว่างสองภาพที่มีความละเอียดต่างกัันนั่นเอง

### 1.2 วัตถุประสงค์ของวิทยานิพนธ์

เพื่อให้สามารถนำภาพทุกภาพที่ได้จากดาวเทียมทุกดวงมาใช้ให้เกิดประโยชน์สูงสุดในวิทยานิพนธ์ฉบับนี้จึงได้นำเสนอ การต่อภาพจากดาวเทียมสองดวงที่มีรายละเอียดต่างกัน แสดงลำดับขั้นตอนในการเตรียมภาพก่อนการต่อและนำมาปรับปรุงคุณภาพของภาพภายหลังการต่อด้วยการรักษาขอบแบบปรับเรียบ ซึ่งมีการนำเสนอหน้าปก (Template) แบบใหม่ เพื่อคงไว้ซึ่งขอบเขตของส่วนย่อยของภาพที่แคบขนาด 1 จุดภาพ ก่อนที่จะนำไปแบ่งส่วนให้ได้จำนวนพื้นที่ย่อยตามต้องการด้วยจุดเด่นของทฤษฎีกราฟ จนได้ภาพที่มีขอบเขตครอบคลุมบริเวณที่ต้องการพิจารณา สะดวกในการนำไปวิเคราะห์และตีความหมายจากภาพต่อไป

### 1.3 ขอบเขตการวิจัย

การวิจัยนี้ต้องการเน้นให้เห็นถึงหลักการของขั้นตอนการดำเนินงานต่างๆ อย่างชัดเจน ดังนั้นจึงจะใช้ภาพตัวอย่างที่ครอบคลุมบริเวณพื้นที่เดียวกันที่ได้จากดาวเทียมสองดวงคือ

MOS-I : ดาวเทียมสำรวจด้านสมุทรศาสตร์ของประเทศญี่ปุ่น โคจรเหนือพื้นโลก 909 กิโลเมตร และโคจรผ่านที่เดิมทุกๆ 17 วัน มีความละเอียดต่อจุดภาพขนาด 8 บิตที่ 50 เมตร × 50 เมตร [5]

JERS-1 : ดาวเทียมสำรวจทรัพยากรแหล่งแร่-พลังงานและสภาพแวดล้อมของประเทศญี่ปุ่น โคจรเหนือพื้นโลก 568 กิโลเมตร และโคจรผ่านที่เดิมทุกๆ 44 วัน มีความละเอียดต่อจุดภาพขนาด 8 บิตที่ 18 เมตร × 24 เมตร [3,4]

รายละเอียดการวิจัยของวิทยานิพนธ์ฉบับนี้จะแบ่งย่อยออกเป็น 5 บท มีรายละเอียดของเนื้อหาในแต่ละบทดังนี้

บทที่ 1 บทนำ

บทที่ 2 แสดงการต่อภาพที่มีความละเอียดต่างกัน รูปแบบข้อมูลภาพถ่ายดาวเทียมที่จะนำมาคือ เทคนิคการแก้ไขความผิดพลาดรูปทรงเรขาคณิตของภาพ การชักตัวอย่างข้อมูลภาพก่อนต่อภาพ การกำจัดรอยตะเข็บที่เกิดจากการต่อและการเตรียมภาพเพื่อนำไปใช้ในขั้นตอนของบทถัดไป

บทที่ 3 เป็นการปรับปรุงรายละเอียดของขอบพื้นที่ย่อยในภาพ รวมถึงการรักษาความเป็นเนื้อเดียวกันภายในพื้นที่ย่อยส่วนต่างๆ แสดงถึงขั้นตอนการปรับเรียบที่พัฒนาขึ้นจากการกรองแบบมัธยฐาน (Median Filtering) โดยได้นำเสนอหน้ากาก (Template) แบบใหม่ที่จะทำให้ยังคงรักษาขอบของภาพที่มีขนาดน้อยกว่า 3 จุดภาพได้ และสามารถลดสัญญาณรบกวนออกไปด้วยในกระบวนการเดียวกัน

บทที่ 4 การแบ่งส่วนภาพ เพื่อแสดงให้เห็นถึงขอบเขตที่ต้องการ การสร้างขอบเขตของพื้นที่ย่อย และนำมาทาบบเข้ากับภาพเดิม จนได้ภาพที่มีขอบเขตตามต้องการ ภาพผลลัพธ์ที่ได้จะส่งผลให้การจำแนกข้อมูลภาพและการแปลความหมายจากภาพมีความแม่นยำและสะดวกขึ้นกว่าเดิม

บทที่ 5 บทสรุป จุดเด่นของผลการวิจัย และข้อเสนอแนะ

ส่วนท้ายเล่มเป็นภาคผนวกแสดงรายละเอียดต้นแบบ โปรแกรมที่สร้างขึ้นในขั้นตอนการวิจัย พัฒนาขึ้นจากภาษาซี สำหรับผู้ที่สนใจจะนำไปพัฒนาให้เหมาะสมกับการประยุกต์ใช้งานต่อไป

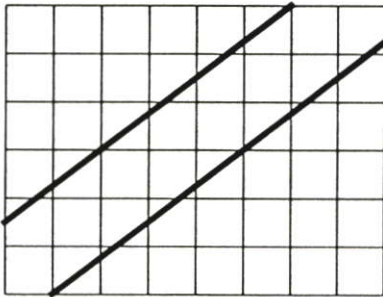
## บทที่ 2

# การต่อภาพแบบหลายความละเอียด

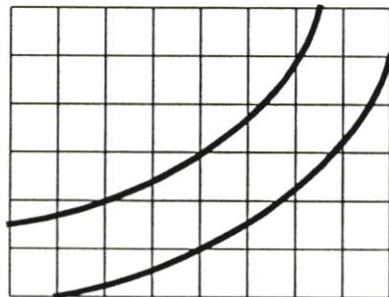
### 2.1 คำนำ

การต่อภาพมักกระทำกับภาพที่มีความละเอียดเท่ากันและมีรูปแบบของข้อมูลเหมือนกัน (ถ้าเป็นไฟล์ในคอมพิวเตอร์ก็คือมีนามสกุลเดียวกันนั่นเอง) โดยสามารถนำข้อมูลของรูปที่จะทำการต่อมาเรียงต่อกันด้วยโปรแกรมจัดการรูปภาพทั่วไป เช่น Adobe Photoshop เป็นต้น หากภาพที่ต้องการต่อมีความเข้มสีที่ใกล้เคียงกันหรือภาพมีที่มาเดียวกัน และเป็นภาพที่อยู่ในบริเวณต่อเนื่องสัมพันธ์กันอยู่แล้ว การต่อภาพโดยใช้โปรแกรมสำเร็จรูปก็สามารถให้ภาพผลลัพธ์ที่มีความกลมกลืนดูสมบูรณได้ในระดับหนึ่ง เนื่องจากโปรแกรมสมัยใหม่ต่างบรรจุเครื่องมือในการปรับแต่งภาพไว้แล้วมากมายพอสมควร

ข้อมูลดิบของภาพถ่ายดาวเทียมที่ได้จากดาวเทียมแต่ละดวงในขั้นแรกนั้น จะเป็นข้อมูลที่ดาวเทียมทำการเก็บค่าที่ได้รับจากอุปกรณ์ตรวจวัดของตัวดาวเทียมเอง ขณะเคลื่อนที่ไปตามแนวโคจรและมุมที่ดาวเทียมแต่ละดวงทำกับพื้นโลก รวมถึงการทรงตัวของตัวดาวเทียมเองขณะบันทึกค่าข้อมูลภาพ ทำให้ภาพที่ได้มีรูปร่างไปจากสัดส่วนจริงหากเทียบกับพิกัดบนพื้นดิน ซึ่งต่อไปจะเรียกว่าความผิดพลาดเชิงเรขาคณิต (Geometric Distortion) ตัวอย่างจำลองให้เห็น ได้ดังรูปที่ 2.1



ก. พิกัดจริงบนพื้นดิน



ข. ภาพที่ได้จากดาวเทียม

รูปที่ 2.1 จำลองภาพแม่น้ำบนพิกัดจริงเทียบกับภาพแม่น้ำที่ได้จากดาวเทียม

ข้อมูลดิบของภาพที่ได้จากดาวเทียมแต่ละดวงต่างก็มีความผิดพลาดเชิงเรขาคณิต และมีความสว่างของภาพต่างกันไป ดังนั้นโปรแกรมสำเร็จรูปที่มีใช้กันทั่วไปนั้น ไม่อาจรองรับกับความเพี้ยนและความเข้มสีของแต่ละภาพได้ จำเป็นที่จะต้องศึกษาหาวิธีแก้ไขและเขียนโปรแกรมขึ้นมารองรับสำหรับแต่ละภาพ ให้ได้ภาพที่ถูกต้องตามพิกัดจริง เพื่อนำไปใช้ในงานวิเคราะห์และแปลความหมายจากภาพต่อไป

## 2.2 การแก้ไขความผิดพลาดเชิงเรขาคณิต (Geometric Correction) [1]

ความผิดพลาดเชิงเรขาคณิต จะเกิดขึ้นกับภาพถ่ายดาวเทียมทุกภาพ การแก้ไขความผิดพลาดของภาพให้ถูกต้อง สามารถทำได้โดยการกำหนดแบบจำลองของจุดสำคัญต่างๆ ในภาพขึ้นมาโดยประมาณตามพิกัดจริง แล้วใช้แบบจำลองนี้ในการดึงข้อมูลภาพทั้งภาพให้เข้าสู่พิกัดของแบบจำลองเพื่อสร้างภาพใหม่ขึ้นมา วิธีการนี้เป็นวิธีการพื้นฐานที่จะสามารถแก้ไขความผิดพลาดเชิงเรขาคณิตที่เกิดจากการหมุนรอบตัวเองของโลกและความไม่สม่ำเสมอของวงโคจรของดาวเทียมให้ลดน้อยลงได้ แต่หากภาพที่ต้องการแก้ไขมีขนาดใหญ่และมีความผิดพลาดเชิงเรขาคณิตมาก การดึงข้อมูลภาพจำเป็นต้องหาความสัมพันธ์ทางคณิตศาสตร์ขึ้นมารองรับ ทั้งนี้เพื่อให้สะดวกสำหรับการเขียนโปรแกรมควบคุมคอมพิวเตอร์ให้สามารถรองรับกับการแก้ไขความเพี้ยนของแต่ละรูปได้อย่างรวดเร็ว

การแก้ไขความผิดพลาดเชิงเรขาคณิตโดยใช้แบบจำลองตามพิกัดจริง จะส่งผลให้ภาพที่ได้มีความถูกต้องใกล้เคียงพิกัดจริงบนพื้นดินที่สุด แต่เนื่องจากภาพถ่ายดาวเทียมมักมีขนาดใหญ่และครอบคลุมพื้นที่บริเวณกว้าง บางครั้งครอบคลุมผืนป่าที่ยังไม่เคยได้รับการสำรวจเพื่อสร้างแผนที่พิกัดจริงบนพื้นดิน หรือแผนที่พิกัดจริงบนพื้นดินอาจถูกสร้างขึ้นเป็นเวลานานมาแล้ว อาจมีการเปลี่ยนแปลงไปตามกาลเวลาของสภาพภูมิศาสตร์ เช่น การพังทลายของตลิ่งริมแม่น้ำ การบุกรุกแนวชายฝั่งทะเล กระบวนการแก้ไขความผิดพลาดเชิงเรขาคณิตของภาพถ่ายดาวเทียม โดยไม่มีภาพอ้างอิง หรืออ้างอิงจากภาพที่อาจเกิดความผิดพลาดขึ้นแล้วย่อมไม่ส่งผลดี เนื่องจากอาจสร้างความผิดพลาดมากขึ้นอีก

ดังนั้นในวิทยานิพนธ์นี้จะใช้ภาพถ่ายอย่างสองภาพ ทำการแก้ไขความผิดพลาดเชิงเรขาคณิตโดยใช้ภาพใดภาพหนึ่งเป็นภาพอ้างอิง และใช้อีกภาพเป็นภาพที่จะถูกปรับแก้ความเพี้ยน (ตัวอย่างจะได้แสดงให้เห็นในผลการทดลอง) โดยภาพอ้างอิงจะถูกใช้เพื่อเปรียบเทียบพิกัดที่เป็นจุดอ้างอิงบนพื้นดินเท่านั้น อีกภาพหนึ่งจะเป็นภาพที่ถูกนำมาผ่านกระบวนการแก้ไขความผิดพลาดเชิงเรขาคณิต เพื่อสร้างภาพใหม่ที่มีพิกัดความเพี้ยนเท่าเทียมกับภาพอ้างอิง โดยให้ความสัมพันธ์ระหว่างพิกัดของข้อมูลภาพใหม่กับพิกัดของจุดข้อมูลภาพเก่าเป็นดังนี้

$$u = f(x, y) \quad (2.1)$$

$$v = g(x, y) \quad (2.2)$$

เมื่อ  $(x, y)$  คือจุดพิกัดของข้อมูลภาพก่อนการปรับแก้

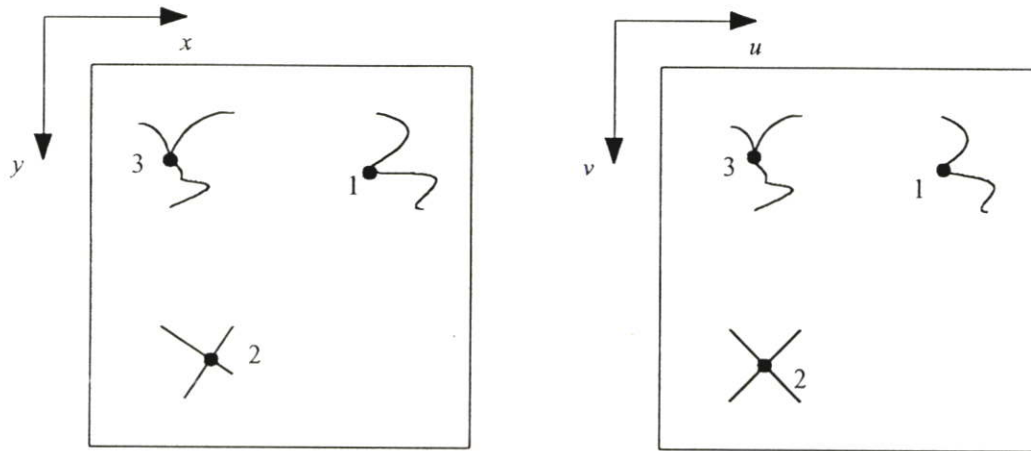
$(u, v)$  คือจุดพิกัดของข้อมูลภาพหลังการปรับแก้

$f(x, y)$  คือฟังก์ชันความสัมพันธ์ระหว่างจุดอ้างอิงของภาพอ้างอิงและภาพที่จะแก้

$g(x, y)$  คือฟังก์ชันความสัมพันธ์ระหว่างจุดอ้างอิงของภาพอ้างอิงและภาพที่จะแก้

การหาจุดอ้างอิงบนพื้นดินระหว่างภาพสองภาพ สามารถทำได้โดยการเลือกลักษณะเด่นที่เห็นได้ชัดเจนจากทั้งสองภาพ เช่น จุดตัดของถนน จุดแยกของแม่น้ำ มุมของแปลงเกษตรกรรม หรือลักษณะ

เด่นอื่นๆ ภายในภาพ โดยเรียกจุดพิกัดเหล่านี้ว่า จุดควบคุมบนพื้นดิน (Ground Control Point) ตัวอย่าง แสดงให้เห็น ได้ดังรูปที่ 2.2



ก. พิกัดของภาพเดิม

ข. พิกัดของภาพใหม่

รูปที่ 2.2 การเลือกจุดควบคุมบนพื้นดินก่อนการแก้ความเพี้ยนเชิงเรขาคณิต

จากรูปที่ 2.2 จะเห็นได้ว่า การเปลี่ยนแปลงพิกัดของจุดควบคุมบนพื้นดินไม่ได้เป็นไปแบบเชิงเส้น เนื่องจากมีความเพี้ยนเกิดขึ้นทั้งในแนวเส้นแวง (Longitude) และแนวเส้นรุ้ง (Latitude) ดังนั้น จึงต้องใช้สมการที่ตัวแปรต้น-แปรตาม-ตัวแปรสองตัว มาใช้ในการแสดงความสัมพันธ์ ในที่นี้จะใช้สมการพหุนามอันดับที่สอง แสดงได้ดังสมการที่ 2.3 และ 2.4 ตามลำดับ มาใช้ในการแก้ปัญหา

$$u(x, y) = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2 \quad (2.3)$$

$$v(x, y) = b_0 + b_1x + b_2y + b_3xy + b_4x^2 + b_5y^2 \quad (2.4)$$

เมื่อ  $(x, y)$  คือจุดพิกัดของข้อมูลภาพก่อนการปรับแก้

$(u, v)$  คือจุดพิกัดของข้อมูลภาพหลังการปรับแก้

จากสมการที่ 2.3 มีค่าคงที่สัมประสิทธิ์ไม่ทราบค่า คือ  $a_0 - a_5$  ที่จำเป็นต้องหาค่าหากต้องการนำสมการไปใช้งาน ในที่นี้จะใช้สมการถดถอยแบบหลายเชิง (Multiple Regression) โดยการเขียนสมการของความผิดพลาด  $R$  ของข้อมูลทั้งหมด  $L$  ข้อมูล ที่เบี่ยงเบนไปจากฟังก์ชัน  $u$  ดังนี้

$$r_i = u_i - f(x_i, y_i) \quad , i = 1, 2, \dots, L \quad (2.5)$$

เมื่อ  $r_i$  คือค่าเบี่ยงเบนของข้อมูลแต่ละจุด

$L$  คือจำนวนจุดข้อมูล

จากสมการที่ 2.5 เมื่อใช้วิธีกำลังสองน้อยสุด (Least Square) จะสามารถเขียนสมการของค่าความเบี่ยงเบนทั้งหมดของทุกจุดข้อมูลได้เป็น

$$R = \sum_{i=1}^L [r_i]^2 \quad (2.6)$$

จากสมการ 2.3 สมการของค่าเบี่ยงเบน  $R$  สามารถกระจายได้เป็น

$$R = \sum_{i=0}^L \left[ u_i - (a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2) \right]^2 \quad (2.7)$$

เมื่อต้องการให้ค่าเบี่ยงเบน  $R$  มีค่าน้อยที่สุด จึงให้อนุพันธ์ย่อยของ  $R$  เทียบกับแต่ละสัมประสิทธิ์  $a_0 - a_5$  ที่ต้องการ มีค่าเท่ากับ 0 จะได้

$$\frac{\partial R}{\partial a_k} = 0, \quad k = 0, 1, 2, \dots, N \quad (2.8)$$

เมื่อ  $N$  คือลำดับสูงสุดของสัมประสิทธิ์

จากสมการที่ 2.8 สามารถเขียนได้เป็น  $N+1$  สมการ นั่นคือ

$$\begin{aligned} \sum_{i=0}^L a_0 &+ \sum_{i=0}^L a_1 x_i &+ \sum_{i=0}^L a_2 y_i &+ \sum_{i=0}^L a_3 x_i y_i &+ \sum_{i=0}^L a_4 x_i^2 &+ \sum_{i=0}^L a_5 y_i^2 &= \sum_{i=0}^L u_i \\ \sum_{i=0}^L a_0 x_i &+ \sum_{i=0}^L a_1 x_i^2 &+ \sum_{i=0}^L a_2 x_i y_i &+ \sum_{i=0}^L a_3 x_i^2 y_i &+ \sum_{i=0}^L a_4 x_i^3 &+ \sum_{i=0}^L a_5 x_i y_i^2 &= \sum_{i=0}^L u_i x_i \\ \sum_{i=0}^L a_0 y_i &+ \sum_{i=0}^L a_1 x_i y_i &+ \sum_{i=0}^L a_2 y_i^2 &+ \sum_{i=0}^L a_3 x_i y_i^2 &+ \sum_{i=0}^L a_4 x_i^2 y_i &+ \sum_{i=0}^L a_5 y_i^3 &= \sum_{i=0}^L u_i y_i \\ \sum_{i=0}^L a_0 x_i y_i &+ \sum_{i=0}^L a_1 x_i^2 y_i &+ \sum_{i=0}^L a_2 x_i y_i^2 &+ \sum_{i=0}^L a_3 x_i^2 y_i^2 &+ \sum_{i=0}^L a_4 x_i^3 y_i &+ \sum_{i=0}^L a_5 x_i y_i^3 &= \sum_{i=0}^L u_i x_i y_i \\ \sum_{i=0}^L a_0 x_i^2 &+ \sum_{i=0}^L a_1 x_i^3 &+ \sum_{i=0}^L a_2 x_i^2 y_i &+ \sum_{i=0}^L a_3 x_i^3 y_i &+ \sum_{i=0}^L a_4 x_i^4 &+ \sum_{i=0}^L a_5 x_i^2 y_i^2 &= \sum_{i=0}^L u_i x_i^2 \\ \sum_{i=0}^L a_0 y_i^2 &+ \sum_{i=0}^L a_1 x_i y_i^2 &+ \sum_{i=0}^L a_2 y_i^3 &+ \sum_{i=0}^L a_3 x_i y_i^3 &+ \sum_{i=0}^L a_4 x_i^2 y_i^2 &+ \sum_{i=0}^L a_5 y_i^4 &= \sum_{i=0}^L u_i y_i^2 \end{aligned} \quad (2.9)$$

สมการที่ 2.9 สามารถจัดรูปใหม่ ให้อยู่ในรูปเมทริกซ์ได้เป็น

$$\begin{bmatrix} L & \sum_{i=0}^L x_i & \sum_{i=0}^L y_i & \sum_{i=0}^L x_i y_i & \sum_{i=0}^L x_i^2 & \sum_{i=0}^L y_i^2 \\ \sum_{i=0}^L x_i & \sum_{i=0}^L x_i^2 & \sum_{i=0}^L x_i y_i & \sum_{i=0}^L x_i^2 y_i & \sum_{i=0}^L x_i^3 & \sum_{i=0}^L x_i y_i^2 \\ \sum_{i=0}^L y_i & \sum_{i=0}^L x_i y_i & \sum_{i=0}^L y_i^2 & \sum_{i=0}^L x_i y_i^2 & \sum_{i=0}^L x_i^2 y_i & \sum_{i=0}^L y_i^3 \\ \sum_{i=0}^L x_i y_i & \sum_{i=0}^L x_i^2 y_i & \sum_{i=0}^L x_i y_i^2 & \sum_{i=0}^L x_i^2 y_i^2 & \sum_{i=0}^L x_i^3 y_i & \sum_{i=0}^L x_i y_i^3 \\ \sum_{i=0}^L x_i^2 & \sum_{i=0}^L x_i^3 & \sum_{i=0}^L x_i^2 y_i & \sum_{i=0}^L x_i^3 y_i & \sum_{i=0}^L x_i^4 & \sum_{i=0}^L x_i^2 y_i^2 \\ \sum_{i=0}^L y_i^2 & \sum_{i=0}^L x_i y_i^2 & \sum_{i=0}^L y_i^3 & \sum_{i=0}^L x_i y_i^3 & \sum_{i=0}^L x_i^2 y_i^2 & \sum_{i=0}^L y_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^L u_i \\ \sum_{i=0}^L u_i x_i \\ \sum_{i=0}^L u_i y_i \\ \sum_{i=0}^L u_i x_i y_i \\ \sum_{i=0}^L u_i x_i^2 \\ \sum_{i=0}^L u_i y_i^2 \end{bmatrix} \quad (2.10)$$

ด้วยวิธีการเดียวกันเมื่อกระทำกับสมการที่ 2.4 จะทำให้ได้ระบบสมการดังนี้

$$\begin{bmatrix} L & \sum_{i=0}^L x_i & \sum_{i=0}^L y_i & \sum_{i=0}^L x_i y_i & \sum_{i=0}^L x_i^2 & \sum_{i=0}^L y_i^2 \\ \sum_{i=0}^L x_i & \sum_{i=0}^L x_i^2 & \sum_{i=0}^L x_i y_i & \sum_{i=0}^L x_i^2 y_i & \sum_{i=0}^L x_i^3 & \sum_{i=0}^L x_i y_i^2 \\ \sum_{i=0}^L y_i & \sum_{i=0}^L x_i y_i & \sum_{i=0}^L y_i^2 & \sum_{i=0}^L x_i y_i^2 & \sum_{i=0}^L x_i^2 y_i & \sum_{i=0}^L y_i^3 \\ \sum_{i=0}^L x_i y_i & \sum_{i=0}^L x_i^2 y_i & \sum_{i=0}^L x_i y_i^2 & \sum_{i=0}^L x_i^2 y_i^2 & \sum_{i=0}^L x_i^3 y_i & \sum_{i=0}^L x_i y_i^3 \\ \sum_{i=0}^L x_i^2 & \sum_{i=0}^L x_i^3 & \sum_{i=0}^L x_i^2 y_i & \sum_{i=0}^L x_i^3 y_i & \sum_{i=0}^L x_i^4 & \sum_{i=0}^L x_i^2 y_i^2 \\ \sum_{i=0}^L y_i^2 & \sum_{i=0}^L x_i y_i^2 & \sum_{i=0}^L y_i^3 & \sum_{i=0}^L x_i y_i^3 & \sum_{i=0}^L x_i^2 y_i^2 & \sum_{i=0}^L y_i^4 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^L v_i \\ \sum_{i=0}^L v_i x_i \\ \sum_{i=0}^L v_i y_i \\ \sum_{i=0}^L v_i x_i y_i \\ \sum_{i=0}^L v_i x_i^2 \\ \sum_{i=0}^L v_i y_i^2 \end{bmatrix} \quad (2.11)$$

จากระบบสมการที่สร้างขึ้นสามารถนำมาแก้สมการหาค่าสัมประสิทธิ์  $a_0 - a_5$  และ  $b_0 - b_5$  นำไปแทนลงในสมการ 2.3 และ 2.4 ตามลำดับ ทำให้สามารถหาค่าของพิกัด  $u, v$  ที่สัมพันธ์กับพิกัดของภาพอ้างอิง  $x, y$  เพื่อนำมาสร้างภาพใหม่ ที่จะมีพิกัดความเพี้ยนเท่าเทียมกัน



รูปที่ 2.3 ภาพจากดาวเทียม JERS-1 ความละเอียด 18 เมตร × 24 เมตร ขนาด 600 × 600  
จุดภาพ



รูปที่ 2.4 ภาพจากดาวเทียม MOS-I ความละเอียด 50×50 เมตร ขนาด 256×256 จุดภาพ

ตารางที่ 2.1 พิกัดของภาพที่จะแก้ความผิดพลาดเชิงเรขาคณิตและภาพอ้างอิงหน่วยเป็นพิกเซล

ลำดับที่	JERS-I		MOS-I	
	uu	vv	xx	yy
1	196	207	74	100
2	28	82	11	39
3	279	461	109	223
4	363	521	141	251
5	548	176	204	80
6	514	191	191	88
7	550	136	204	61
8	555	39	204	14
9	453	464	172	222
10	499	476	190	228
11	528	119	195	53
12	466	206	173	96

ข้อมูลที่แสดงในตารางที่ 2.1 ถูกใช้แทนลงในระบบสมการที่ 2.10 และ 2.11 ทำการแก้ระบบสมการดังกล่าว จะได้สัมประสิทธิ์ทั้งหมดดังนี้

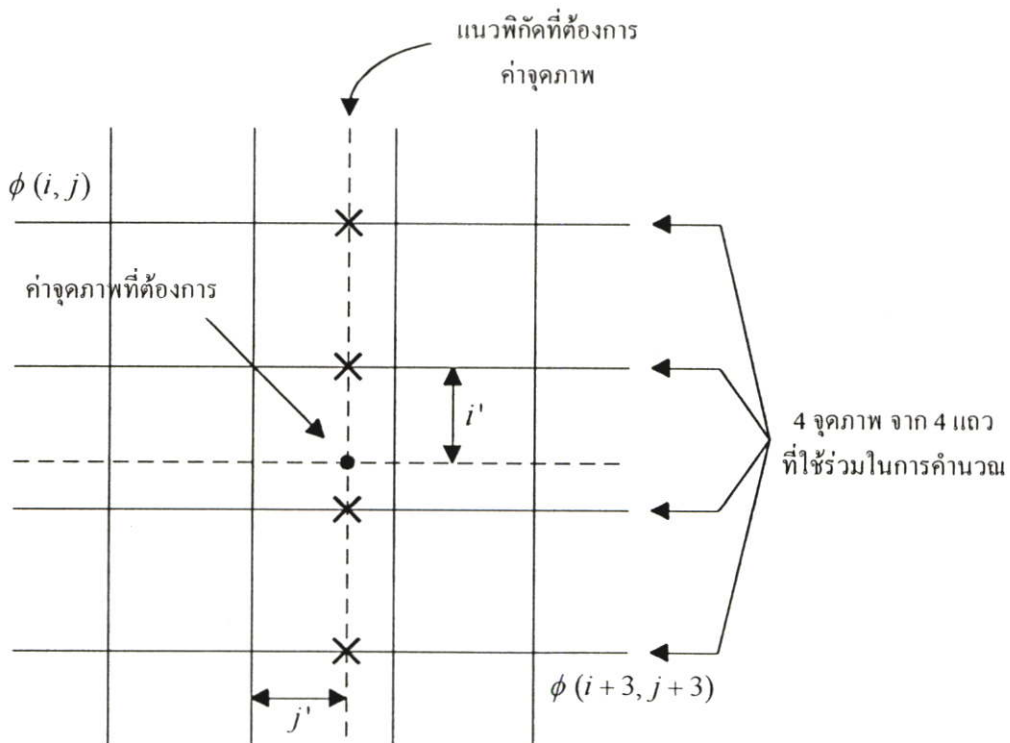
$$\begin{aligned}
 a_0 &= -1.3703471996 & b_0 &= -0.4991797543 \\
 a_1 &= 0.3474587717 & b_1 &= 0.0038828122 \\
 a_2 &= 0.0337184847 & b_2 &= 0.4820606746 \\
 a_3 &= -0.0000268223 & b_3 &= 0.0000003022 \\
 a_4 &= 0.0000386804 & b_4 &= -0.0000216280 \\
 a_5 &= -0.0000066253 & b_5 &= 0.0000078957
 \end{aligned} \tag{2.12}$$

จากข้อมูลสมการที่ 2.12 ทำให้สัมประสิทธิ์ในสมการที่ 2.3 และ 2.4 สมบูรณ์ สามารถนำสมการทั้งสองไปใช้สร้างภาพใหม่ที่มีพิกัดความเพี้ยนเชิงเรขาคณิตเท่าเทียมกัน แต่ข้อมูลภาพที่ได้ยังไม่สามารถนำมาแสดงผลได้ ทั้งนี้เนื่องจากพิกัดที่ได้ใหม่ ไม่ได้เป็นค่าจำนวนเต็ม ดังนั้นจึงต้องนำข้อมูลภาพที่ได้ไปผ่านกระบวนการชักรัดตัวอย่างและแทรกค่าข้อมูลต่อไป

### 2.3 การซ้กดตัวอย่างข้อมูลและการแทรกค่าข้อมูล [1,9]

จากการปรับแก้ความเพี้ยนเชิงเรขาคณิตในหัวข้อที่ผ่านมา จะต้องมีการหาพิกัดที่สอดคล้องหรือสัมพันธ์กันในแต่ละตำแหน่งของข้อมูลภาพ ซึ่งผลที่ได้ทำให้ข้อมูลจุดภาพจำนวนมากมีพิกัดของข้อมูลอยู่ในตำแหน่งที่เป็นทศนิยม ขณะที่หลายตำแหน่งที่เป็นจำนวนเต็มกลับไม่มีข้อมูลจุดภาพอยู่ และเป็นปัญหาหากจะนำข้อมูลดังกล่าวแสดงผลออกทางจอภาพ ดังนั้นข้อมูลของภาพภายหลังจากการปรับแก้ความเพี้ยนเชิงเรขาคณิต จึงจำเป็นต้องนำมาซ้กดตัวอย่างใหม่ (Resampling) เพื่อปรับพิกัดของข้อมูลภาพให้อยู่ในตำแหน่งที่เป็นจำนวนเต็ม

การแก้ไขความเพี้ยนเชิงเรขาคณิตในหัวข้อที่ผ่านมา กระทำกับภาพเพียงภาพเดียว ดังนั้นจะมีเพียงภาพเดียวเช่นกันที่จะต้องนำมาทำการซ้กดตัวอย่างและแทรกค่าข้อมูลใหม่ เนื่องจากอีกภาพหนึ่งนั้นมีพิกัดเป็นจำนวนเต็มไม่ได้ผ่านกระบวนการใดๆ ดังนั้นจึงไม่เป็นปัญหาสำหรับการแสดงผลเมื่อใช้ภาพที่มีความละเอียดน้อยกว่าเป็นภาพอ้างอิง ภาพที่มีรายละเอียดมากกว่าจะถูกซ้กดข้อมูลภาพออกมาวางในพิกัดใหม่ที่ขยายกว่า ดังนั้นอาจมีข้อมูลภาพส่วนหนึ่งถูกไม่ถูกเลือกมา ข้อมูลภาพที่ถูกเลือกไว้ เมื่อนำมาผ่านกระบวนการซ้กดตัวอย่าง เพื่อปรับพิกัดให้เป็นจำนวนเต็มแล้ว ก็พร้อมที่จะนำไปต่อภาพได้ทันที แต่หากใช้ภาพที่มีรายละเอียดมากกว่าเป็นภาพอ้างอิง ภาพที่มีรายละเอียดน้อยกว่าจะมีข้อมูลไม่เพียงพอที่จะซ้กดตัวอย่างนำไปวางในพิกัดใหม่ได้ครบถ้วนทุกจุดภาพ ดังนั้นจึงต้องมีการแทรกค่าข้อมูลที่ขาดหายไป เพื่อให้ได้ข้อมูลที่สมบูรณ์ครบทุกจุดภาพ



รูปที่ 2.5 ลักษณะการซ้กดตัวอย่างและแทรกค่าข้อมูล

การชักตัวอย่างและแทรกค่าข้อมูลในที่นี้ จะใช้ข้อมูลจุดภาพเวกเตอร์ตำแหน่งที่ต้องการพิจารณาจำนวน 16 จุดภาพ แสดงดังรูปที่ 2.3 ซึ่งประกอบไปด้วยจุดภาพ 4 แถวในแนวนอน เมื่อให้ค่าจุดภาพในแต่ละแถวเป็น

$$\begin{array}{l}
 \text{แถวที่ 1} \quad \phi(i, j) \quad \phi(i, j+1) \quad \phi(i, j+2) \quad \phi(i, j+3) \\
 \text{แถวที่ 2} \quad \phi(i+1, j) \quad \phi(i+1, j+1) \quad \phi(i+1, j+2) \quad \phi(i+1, j+3) \\
 \text{แถวที่ 3} \quad \phi(i+2, j) \quad \phi(i+2, j+1) \quad \phi(i+2, j+2) \quad \phi(i+2, j+3) \\
 \text{แถวที่ 4} \quad \phi(i+3, j) \quad \phi(i+3, j+1) \quad \phi(i+3, j+2) \quad \phi(i+3, j+3) \quad (2.12)
 \end{array}$$

ในแต่ละแถวจะมีข้อมูลจุดภาพ 4 จุด ซึ่งเราจะสร้างฟังก์ชันที่ผ่านจุดข้อมูลทั้งสี่ด้วยเส้นโค้งกำลังสาม (Cubic Spline) เพื่อหาค่าในตำแหน่ง  $\phi(i, j')$ ,  $\phi(i+1, j')$ ,  $\phi(i+2, j')$  และ  $\phi(i+3, j')$  ตามลำดับ โดยจะประกอบไปด้วย 4 ฟังก์ชัน จาก 4 แถว ดังนี้

- ค่าข้อมูลจุดภาพในแถวที่ 1 จะทำให้ได้ฟังก์ชันเพื่อหาค่า  $\phi(i, j')$
- ค่าข้อมูลจุดภาพในแถวที่ 2 จะทำให้ได้ฟังก์ชันเพื่อหาค่า  $\phi(i+1, j')$
- ค่าข้อมูลจุดภาพในแถวที่ 3 จะทำให้ได้ฟังก์ชันเพื่อหาค่า  $\phi(i+2, j')$
- ค่าข้อมูลจุดภาพในแถวที่ 4 จะทำให้ได้ฟังก์ชันเพื่อหาค่า  $\phi(i+3, j')$  (2.13)

ข้อมูลผลลัพธ์ที่ได้จากสมการ (2.12) เป็นข้อมูลที่อยู่ในแนวพิกัดที่ต้องการทราบค่าจุดภาพ  $\phi(i', j')$  ดังนั้นข้อมูล 4 ข้อมูลที่ได้จะนำมาใช้สร้างฟังก์ชัน เพื่อหาค่า  $\phi(i', j')$  ซึ่งมีพิกัดตามต้องการ

การสร้างฟังก์ชันพหุนามกำลังสามเพื่อประมาณค่าในช่วงที่ต้องการ จะใช้วิธีการของลากรองจ์ (Lagrange Interpolation) [9] เนื่องจากจุดเด่นคือไม่ต้องมีการหาค่าอนุพันธ์ในขั้นตอนของการสร้างฟังก์ชัน และฟังก์ชันที่ได้ผ่านทุกจุดข้อมูล ทำให้ไม่ต้องหาค่าสหสัมพันธ์ (Correlation) เพื่อตรวจสอบความเหมาะสมของฟังก์ชันซ้ำอีกเป็นการลดความวุ่นวายในการพัฒนาโปรแกรมลงไปได้ การประมาณค่าในช่วงโดยพหุนามลากรองจ์อันดับที่  $n$  สามารถสร้างได้จากความสัมพันธ์ดังนี้

$$f_n(x) = \sum_{i=0}^n L_i(x) f(x_i) \quad (2.14)$$

เมื่อ

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \quad (2.15)$$

ในที่นี้ต้องการสร้างสมการอันดับที่สามดังนั้นจะกระจายสมการที่ 2.14 ได้เป็น

$$f_3(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) + L_3(x)f(x_3) \quad (2.16)$$

และกระจายสมการที่ 2.15 ได้เป็น

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} \cdot \frac{x - x_3}{x_0 - x_3}$$

$$L_1(x) = \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2} \cdot \frac{x - x_3}{x_1 - x_3}$$

$$L_2(x) = \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1} \cdot \frac{x - x_3}{x_2 - x_3}$$

$$L_3(x) = \frac{x - x_0}{x_3 - x_0} \cdot \frac{x - x_1}{x_3 - x_1} \cdot \frac{x - x_2}{x_3 - x_2} \quad (2.17)$$

กำหนดให้

$f(x_i)$  คือค่าความเข้มระดับสีเทาที่จุด  $x_i$

$x_i$  คือพิกัดจุดภาพ

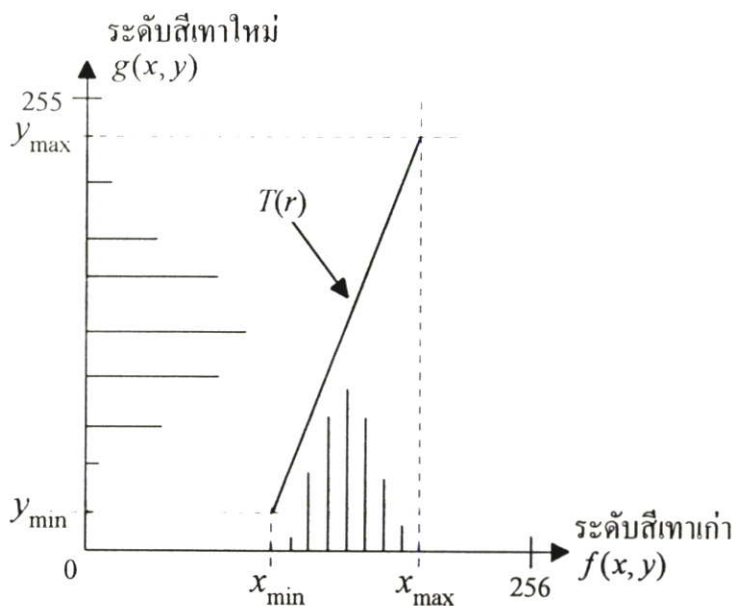
สมการที่ 2.16 และ 2.17 จะทำให้สามารถแก้สมการที่ 2.13 และ 2.14 เพื่อหาค่าความเข้มระดับสีเทาในพิกัดตำแหน่งใดๆ ที่ไม่มีข้อมูล เมื่อกระทำกับทุกพิกัดที่ข้อมูลขาดหายไปก็จะทำให้ได้ข้อมูลจุดภาพครบทุกพิกัด สามารถแสดงผลออกจอภาพได้และพร้อมที่จะนำไปใช้งานต่อไป



รูปที่ 2.6 ผลลัพธ์ของการชักตัวอย่างและแทรกค่าข้อมูลของภาพที่ผ่านการแก้ความเพี้ยนเชิงเรขาคณิตของภาพในรูปที่ 2.3 จำนวนจุดภาพเปลี่ยนเป็น  $228 \times 285$  จุดภาพ

## 2.4 การปรับปรุงฮิสโตแกรมของภาพ

ภาพที่มีความละเอียดและความเพี้ยนเชิงเรขาคณิตที่เท่าเทียมกันและถูกชักตัวอย่างแทรกค่าข้อมูลปรับพิกัดเรียบร้อยแล้วคือภาพที่พร้อมจะนำไปต่อกันเป็นภาพใหม่ แต่ภาพถ่ายดาวเทียมส่วนใหญ่ก็มีความเข้มของระดับสีเทาใกล้เคียงกันทั้งภาพ ทั้งนี้เนื่องจากการตอบสนองของวัตถุต่างๆ บนพื้นโลกที่มีต่ออุปกรณ์วัด (Sensor) ของดาวเทียม ที่ให้ค่าความแตกต่างกับอุปกรณ์ตรวจวัดใกล้เคียงกัน ทำให้พิจารณาเห็นรายละเอียดในบริเวณต่างๆ ภายในภาพได้ไม่ดี จึงทำการปรับปรุงคุณภาพของภาพเสียก่อน การใช้โปรแกรมสำเร็จรูปที่ใช้สำหรับจัดการรูปภาพทั่วไปก็ไม่สามารถนำมาจัดการกับภาพถ่ายดาวเทียมได้ เนื่องจากภาพถ่ายดาวเทียมมักมีบริเวณที่มีค่าระดับสีเทาที่สว่างมากหรือมืดมากติดต่อกันใกล้เคียงในบริเวณเดียวกัน ดังเหตุผลที่ได้กล่าวข้างต้น เมื่อทำการปรับปรุงภาพด้วยโปรแกรมสำเร็จรูปจะทำให้เกิดความสว่างอิ่มตัว (Brightness Saturation) กล่าวคือบริเวณที่สว่างก็จะสว่างมากขึ้น บริเวณที่มืดก็จะมีมากยิ่งขึ้น ส่งผลเสียต่อการวิเคราะห์ข้อมูลภาพถ่ายดาวเทียม ดังนั้นจะอาศัยคุณสมบัติของสมการเชิงเส้นมาช่วยในการขยายย่านฮิสโตแกรม



รูปที่ 2.7 การยืดฮิสโตแกรมด้วยสมการเชิงเส้น

การขยายฮิสโตแกรมแบบเชิงเส้นเป็นการแปลงอัตราความแตกต่างของภาพ ให้ช่วงระดับสีเทาที่บริเวณที่ต้องการ กระจายออกไปยังบริเวณที่กำหนดอย่างสม่ำเสมอ จากรูปที่ 2.7 สามารถสร้างสมการแสดงความสัมพันธ์ในการขยายฮิสโตแกรมได้ดังนี้

$$g(x,y) = \left[ \frac{(f(x,y) - x_{\min})(y_{\max} - y_{\min})}{(x_{\max} - x_{\min})} \right] + y_{\min} \quad (2.18)$$

เมื่อ	$g(x,y)$	คือ ค่าระดับสีเทาใหม่
	$f(x,y)$	คือ ค่าระดับสีเทาเดิม
	$x_{\min}$	คือ ค่าระดับสีเทาค่าสุดของฮิสโตแกรมเดิม
	$x_{\max}$	คือ ค่าระดับสีเทาสูงสุดของฮิสโตแกรมเดิม
	$y_{\min}$	คือ ค่าระดับสีเทาค่าสุดของฮิสโตแกรมใหม่
	$y_{\max}$	คือ ค่าระดับสีเทาสูงสุดของฮิสโตแกรมใหม่

ค่าของ  $x_{\min}$  ถึง  $x_{\max}$  นั้น คือค่าคงที่ซึ่งจะต้องกำหนดให้ครอบคลุมช่วงของฮิสโตแกรมเดิม เพื่อให้ข้อมูลภาพเก่าทุกข้อมูลถูกนำมาใช้ในการคำนวณด้วย หากเลือกค่าไม่เหมาะสมอาจส่งผลให้ภาพผลลัพธ์ที่ได้ผิดเพี้ยนไป ค่าของ  $y_{\min}$  ถึง  $y_{\max}$  คือช่วงของฮิสโตแกรมของภาพผลลัพธ์ที่ต้องการ ควรให้มีช่วงที่กว้างครอบคลุมช่วงของ  $x_{\min}$  ถึง  $x_{\max}$  ด้วย ทั้งนี้เพื่อให้ภาพผลลัพธ์มีการกระจายของฮิสโตแกรมมากกว่าเดิม ซึ่งจะส่งผลให้ข้อมูลของภาพผลลัพธ์มีการกระจายต่อเนื่องออกไป ภาพที่ได้จึงมี

ความต่อเนื่องของค่าระดับสีเทา ไม่สว่างจ้าหรือมืดจัดเกินไป



รูปที่ 2.8 ผลลัพธ์จากการกระจายฮิสโตแกรมของภาพในรูปที่ 2.6

## 2.5 การปรับความสว่าง (Brightness Adjustment)

ความสว่างของภาพที่มาจากดาวเทียมคนละดวงกัน ย่อมมีความสว่างของภาพต่างกัน หากนำมาต่อกันจะเกิดความไม่ต่อเนื่องของความสว่างทำให้สังเกตเห็นรอยต่อชัดเจน การปรับความสว่างของภาพสองภาพให้ใกล้เคียงกันจะช่วยลดรอยตะเข็บที่เกิดจากการต่อให้ลดน้อยลง ก่อนจะนำไปต่อและกำจัดรอยตะเข็บต่อไป การปรับความสว่างของภาพในที่นี้ จะทำการปรับปรุงเพียงภาพเดียว ให้มีความสว่างใกล้เคียงกับอีกภาพ วิธีการปรับค่าความสว่างมีรายละเอียดต่างๆ ดังนี้

คำนวณหาค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐานของข้อมูลภาพทั้งสองจาก

$$m_R = \frac{1}{T} \sum_{i=1}^T g_R(i) \quad , \quad \sigma_R = \sqrt{\frac{1}{T} \sum_{i=1}^T [g_R(i) - m_R]^2} \quad (2.19)$$

$$m_a = \frac{1}{T} \sum_{i=1}^T g_a(i) \quad , \quad \sigma_a = \sqrt{\frac{1}{T} \sum_{i=1}^T [g_a(i) - m_a]^2} \quad (2.20)$$

นำค่าที่ได้จากสมการที่ 2.19 และ 2.20 มาแทนลงในสมการ 2.21

$$y = \left[ \frac{\sigma_R}{\sigma_a} \right] x + m_R - \left[ \frac{\sigma_R}{\sigma_a} \right] m_a \quad (2.21)$$

เมื่อ	$R$	หมายถึงภาพอ้างอิง
	$a$	หมายถึงภาพใดๆ ที่ต้องการปรับความสว่าง
	$g_R$	หมายถึงค่าความสว่างของข้อมูลภาพอ้างอิง
	$g_a$	หมายถึงค่าความสว่างของข้อมูลภาพที่จะปรับ
	$m_R$	หมายถึงค่าเฉลี่ยของข้อมูลภาพอ้างอิง
	$\sigma_R$	หมายถึงค่าเบี่ยงเบนมาตรฐานของข้อมูลภาพอ้างอิง
	$m_a$	หมายถึงค่าเฉลี่ยของข้อมูลภาพที่จะปรับ
	$\sigma_a$	หมายถึงค่าเบี่ยงเบนมาตรฐานของข้อมูลภาพที่จะปรับ
	$x$	หมายถึงค่าระดับสีเทาเดิม
	$y$	หมายถึงค่าระดับสีเทาใหม่

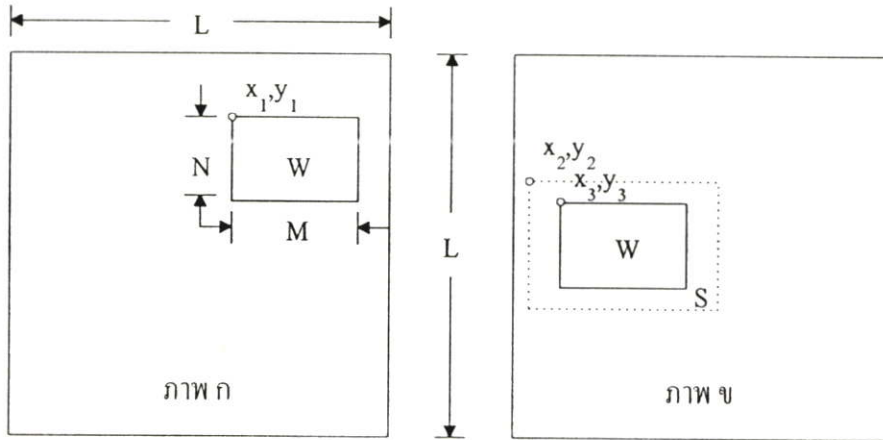


รูปที่ 2.9 ผลลัพธ์ของการปรับความสว่างของภาพในรูปที่ 2.8 ให้ใกล้เคียงกับรูปที่ 2.4

## 2.6 การต่อภาพแบบสหสัมพันธ์ [12]

ในการต่อภาพดาวเทียมทั้งสองภาพเข้าด้วยกันจำเป็นต้องหาค่าที่แสดงความสัมพันธ์ของข้อมูลภาพในบริเวณที่ซ้อนทับ เพื่อให้ได้บริเวณที่ต่อกันแล้วกลมกลืนได้ดี โดยดูจากลักษณะที่สอดคล้องหรือเหมือนกันของภาพทั้งสอง ซึ่งในการสังเกตด้วยตาเปล่าถ้าบริเวณที่ต้องการหามีลักษณะคล้ายกับบริเวณอื่นๆ ในภาพที่ไม่ต้องการก็อาจจะทำให้เกิดการผิดพลาดได้ง่าย ดังนั้นเพื่อให้ได้บริเวณข้อมูลภาพที่สัมพันธ์กันมากที่สุดในการต่อข้อมูลภาพจึงได้อาศัยการคำนวณทางคณิตศาสตร์เข้ามาช่วย โดยที่จะดูจากค่าตัวเลขที่แสดงความสัมพันธ์ระหว่างภาพต้นฉบับและภาพที่ต้องการจะนำมาต่อ นั่น ซึ่งเรียกว่า ค่าสหสัมพันธ์ของข้อมูลภาพ (Image correlation) ลักษณะของสมการที่จะได้นำเสนอต่อไปเป็นการหาค่าสหสัมพันธ์ของข้อมูลภาพโดยการอาศัยการเปรียบเทียบบริเวณที่ต้องการของข้อมูลภาพทั้งสองแบบ กล่าวคือพิกัดภาพต่อพิกัดภาพหรือทีละพิกัดภาพ โดยจะกำหนดให้บริเวณที่ต้องการจะสัมพันธ์มีลักษณะเป็นหน้ากาก(Mask) มีขนาดตามต้องการ และเคลื่อนที่ไปบนข้อมูลภาพต้นฉบับจนกระทั่งพบบริเวณที่สัมพันธ์กันก็จะแสดงค่าความสัมพันธ์ออกมาเป็นตัวเลข ค่าของตัวเลขที่มากจะแสดงถึงความสัมพันธ์ของภาพมีลักษณะที่ใกล้เคียงหรือเหมือนกันมากที่สุด แต่ทั้งนี้จะไม่เกิน "1"

การหาพิกัดที่ซ้อนทับกันของข้อมูลภาพที่ต้องการ โดยทั่วไปข้อมูลภาพที่จะนำมาต่อเข้าด้วยกันนั้นจะต้องมีพื้นที่บางส่วนของที่เหลื่อมทับกันระหว่างข้อมูลภาพทั้งสองไม่น้อยกว่า 20 เปอร์เซ็นต์ หรือ 1 ใน 5 ส่วน ดังนั้นการกำหนดรอยต่อก็จะหาได้จากบริเวณที่พิกัดของข้อมูลภาพมีการซ้อนทับกัน ซึ่งในการต่อข้อมูลภาพนั้นจะไม่คำนึงว่าค่าระดับสีเทาของข้อมูลภาพทั้งสองจะแตกต่างกันหรือเหมือนกันเพราะว่ามีซอฟต์แวร์ที่ต้องนำมาคิดแต่ในการพิจารณาจะคำนึงถึงเพียงลักษณะทางภูมิประเทศของภาพเท่านั้น



รูปที่ 2.10 การหาค่าแห่งซ้อนทับของภาพ

จากรูปที่ 2.10(ก) เลือกภาพย่อย  $W$  ณ ตำแหน่งเริ่มต้น  $(x_1, y_1)$  ขนาด  $M \times N$  จุดภาพ จากภาพใหญ่ขนาด  $L \times L$  จุดภาพ โดยเลือกจากตำแหน่งส่วนที่มีลักษณะเด่นชัด เช่นแม่น้ำ ทางแยก ถนน เป็นต้น การหาค่าแห่งซ้อนทับที่เหมาะสมของภาพสามารถคำนวณได้ดังสมการ

$$E(i, j) = \frac{\sum_{n=1}^N \sum_{m=1}^M \{w(n, m) - \bar{w}\} \{s(i+n-1, j+m-1) - \bar{s}\}}{\sqrt{\sum_{n=1}^N \sum_{m=1}^M \{w(n, m) - \bar{w}\}^2} \sqrt{\sum_{n=1}^N \sum_{m=1}^M \{s(i+n-1, j+m-1) - \bar{s}\}^2}} \quad (2.22)$$

เมื่อ

$w(n, m)$  หมายถึงข้อมูลภาพย่อยจุดที่พิจารณา

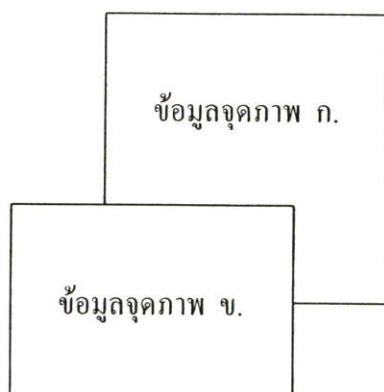
$\bar{w} = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N w(m, n)$  หมายถึงค่าเฉลี่ยข้อมูลภาพย่อย

$s(i+n-1, j+m-1)$  หมายถึงข้อมูลภาพหลักจุดที่พิจารณา

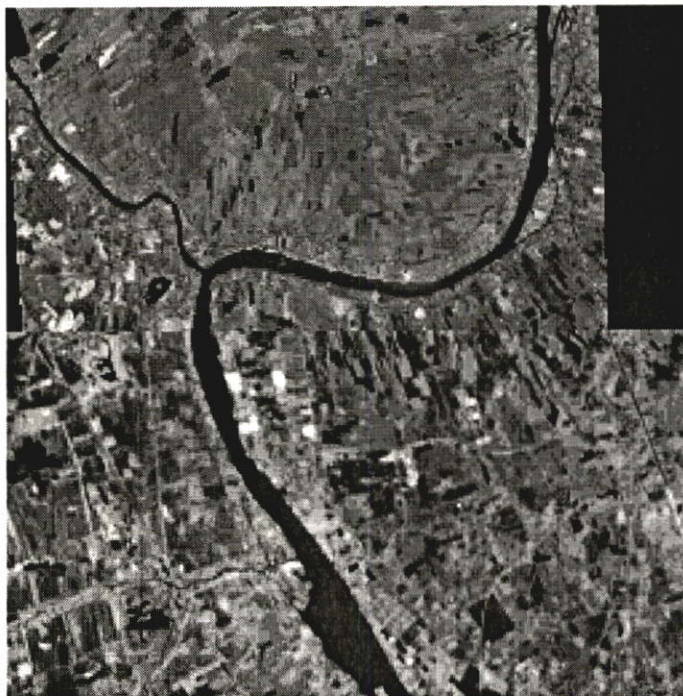
$$\bar{s} = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N s(i+n-1, j+m-1) \quad \text{หมายถึงค่าเฉลี่ยข้อมูลภาพหลัก}$$

ในการคำนวณค่า  $E(i, j)$  มีค่าเท่ากับ 1 หมายความว่า ข้อมูลภาพที่นำมาเปรียบเทียบมีความคล้ายกันทุกประการ แต่ถ้า  $E(i, j)$  มีค่าเข้าใกล้ 1 ก็หมายความว่า ข้อมูลภาพที่นำมาเปรียบเทียบคล้ายกันมากๆ ทั้งนี้การตัดสินใจได้ว่าข้อมูลภาพจะมีความคล้ายกันมากแค่ไหน ขึ้นอยู่กับค่า  $E(i, j)$  ว่าเข้าใกล้ 1 มากเพียงใด หากนำภาพที่เป็นภาพเดียวกันมาเปรียบเทียบกันเองจะทำให้ได้ค่า  $E(i, j)$  เป็น 1 นั่นคือมีความคล้ายกันมากที่สุด

เมื่อสามารถคำนวณหาตำแหน่งซ้อนทับที่เหมาะสมได้แล้ว การต่อภาพก็คือการนำเอาข้อมูลจุดภาพของภาพสองภาพมาเรียงให้ต่อเนื่องกันไปนั่นเอง โดยอาจได้ภาพใหม่ที่มีขนาดใหญ่กว่าเดิมรูปที่ 2.12 แสดงลักษณะของข้อมูลของภาพภายหลังการต่อภาพ



รูปที่ 2.11 ที่มาของข้อมูลจุดภาพใหม่



รูปที่ 2.12 ผลลัพธ์ของการต่อภาพรูปที่ 2.9 เข้ากับรูปที่ 2.4

## 2.7 สรุปผลการทดลอง

การต่อภาพแบบหลายความละเอียดประกอบไปด้วยหลายขั้นตอน ยิ่งยากกว่าการต่อภาพแบบปกติมากมาย ทั้งนี้เนื่องจากภาพที่จะนำมาต่อมีคุณสมบัติหลายอย่างที่แตกต่างกันมาก ดังนั้นจึงจำเป็นต้องมีการปรับปรุงก่อนต่อภาพ เพื่อให้มีคุณสมบัติใกล้เคียงกันดังต่อไปนี้

- 2.7.1 มีความเพี้ยนเชิงเรขาคณิตใกล้เคียงกัน เมื่อต่อแล้วจะได้ภาพที่มีสัดส่วนเดียวกัน
- 2.7.2 มีความสว่างใกล้เคียงกัน เพื่อให้ลดความแตกต่างบริเวณรอยต่อ

เมื่อภาพมีคุณสมบัติใกล้เคียงกันแล้ว การต่อภาพก็จะสามารถทำได้คล้ายกับการต่อภาพทั่วไป ในกระบวนการต่อภาพก่อให้เกิดค่าตัวแปรขึ้นมากมาย และยากแก่การถ่ายโอนเพื่อให้โปรแกรมสามารถทำงานให้เสร็จได้ในขั้นตอนเดียว ทั้งนี้เนื่องจากยังมีค่าตัวแปรบางค่าที่จำเป็นต้องให้ผู้ใช้โปรแกรมเป็นผู้ป้อน เช่น จุดพิกัดอ้างอิง (ซึ่งเป็นตัวแสดงความสัมพันธ์ระหว่างสองภาพ) ที่จะต้องกำหนดใหม่ทุกครั้งตามแต่คู่ภาพที่ต้องการต่อ ซึ่งยังคงทำให้การใช้งานค่อนข้างยุ่งยาก ภายหลังขั้นตอนการชักตัวอย่างและแทรกค่าข้อมูล ข้อมูลภาพใหม่ที่ได้อาจจะผิดเพี้ยนไปจากค่าเดิมบ้าง แต่ในที่นี้ถือว่าค่าผิดพลาดที่เกิดขึ้นยอมรับได้ เมื่อเทียบกับผลการตอบสนองของตัวตรวจวัด (Sensor) ของดาวเทียม

## บทที่ 3

### การรักษาขอบแบบปรับเรียบ

#### 3.1 บทนำ

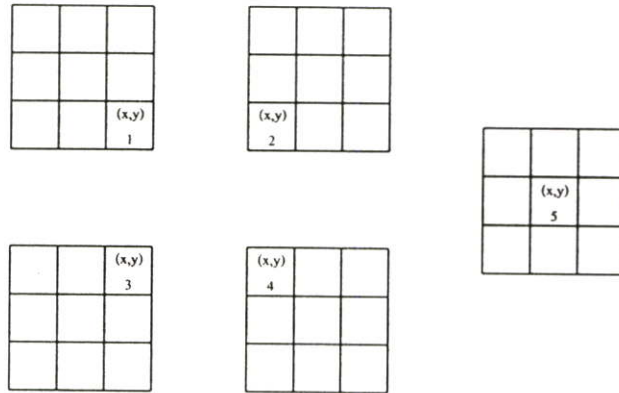
ภาพถ่ายดาวเทียมประกอบไปด้วยพื้นที่ย่อยมากมาย ภาพที่จัดเก็บได้อาจมีพื้นที่ย่อยที่เป็นสัญญาณรบกวนแบบสุ่มเกิดขึ้นกระจายอยู่ในภาพด้วย หากนำภาพที่ยังไม่ได้ทำการกำจัดสัญญาณรบกวนเหล่านี้ไปใช้งาน จะทำให้การวิเคราะห์ภาพถ่ายเป็นไปด้วยความยุ่งยากมากและเป็นเหตุให้ความน่าเชื่อถือจากการวิเคราะห์ภาพลดน้อยลงด้วย การกำจัดสัญญาณรบกวนออกจากภาพได้ถูกคิดค้นขึ้นมากมายหลายวิธี การปรับเรียบด้วยค่าเฉลี่ย (Mean value smoothing) ถือเป็นวิธีพื้นฐานที่นิยมใช้ในการกำจัดสัญญาณรบกวนแบบสุ่มออกจากภาพ เนื่องจากมีขั้นตอนการประมวลผลที่ไม่ยุ่งยากและใช้เวลาในการคำนวณน้อย การวางหน้ากาก (Template) จัตุรัสขนาด  $3 \times 3$  จุดภาพ หรือ  $5 \times 5$  จุดภาพ ลงไปเพื่อคำนวณค่าจุดภาพจุดใหม่ ทำให้สัญญาณรบกวนถูกกำจัดไป ขณะที่ขอบของพื้นที่ย่อยต่างๆ ถูกทำลายลงไปด้วย ในวิทยานิพนธ์นี้จะได้นำเสนอหน้ากากแบบใหม่ ที่จะสามารถกำจัดสัญญาณรบกวนแบบสุ่มออกไปโดยที่ยังคงรักษาขอบของพื้นที่ย่อยต่างๆ เอาไว้ได้

#### 3.2 เทรมเพลท (Template) ที่ใช้ในการปรับเรียบ [2,7]

ขบวนการปรับเรียบถูกนำมาประยุกต์ใช้เพื่อกำจัดสัญญาณรบกวนที่แปลกปลอมเข้ามาอยู่ในภาพ การปรับเรียบภาพโดยใช้เทมเพลท สามารถกำจัดสัญญาณรบกวนออกไปได้ แต่การปรับเรียบเมื่อถูกใช้ในภาพที่มีความแตกต่างของค่าระดับสีเทาในภาพมากๆ จะส่งผลให้ภาพผลลัพธ์ขาดความคมชัดลงไปด้วย เนื่องจากค่าระดับสีเทาของจุดภาพต่างๆ ถูกเฉลี่ยเปลี่ยนค่าไปให้ใกล้เคียงกับจุดข้างเคียง ขอบเขตของพื้นที่ย่อยต่างๆ อาจถูกทำลายลงไป ส่งผลถึงการแบ่งส่วนภาพที่จะกระทำได้ผลแย่ลงไปด้วย ในวิทยานิพนธ์นี้จะได้นำเสนอเทมเพลทแบบใหม่ที่พัฒนาขึ้นจากหลักการปรับเรียบภาพของ Tomata โดยมีการเลือกขอบเขตบริเวณที่กำหนดไว้ล่วงหน้า (ซึ่งก็คือลักษณะเทมเพลทที่จะได้เตรียมไว้นั่นเอง) เพื่อคว่าจุดภาพที่พิจารณา  $(x, y)$  ไปอยู่ในที่ใดแล้วให้ความเป็นเอกพันธ์สูงที่สุดโดยใช้ค่าความเบี่ยงเบนมาตรฐานเป็นเกณฑ์ในการตัดสินใจ เมื่อพบพื้นที่ที่ให้ค่าความเบี่ยงเบนมาตรฐานต่ำที่สุด ก็จะนำเอาค่าเฉลี่ยของระดับสีเทาจุดภาพต่างๆ ในพื้นที่ดังกล่าวแทนลงในค่าระดับสีเทาของจุดภาพตำแหน่ง  $(x, y)$  สร้างเป็นภาพใหม่ การปรับเรียบภาพนี้จะทำตลอดทั้งภาพและมีการทำซ้ำกับทุกๆ จุดภาพด้วยหน้ากากที่เตรียมไว้ทุกๆ แบบ จนกระทั่งไม่เกิดการเปลี่ยนแปลงค่าใดๆ ของจุดภาพ

เทมเพลทของ Tomata เป็นเทมเพลทแบบสี่เหลี่ยมจัตุรัสขนาดใดๆ โดยให้จุดพิจารณา  $(x, y)$  อยู่ในตำแหน่งมุมทั้งสี่ของเทมเพลทและอยู่กึ่งกลางเทมเพลท รวมทั้งหมด 5 รูปแบบแสดงได้ดังรูปที่ 3.1 ปัญหาที่เกิดขึ้นในวิธีของ Tomata คือ ถ้าจุดภาพที่พิจารณาอยู่ในบริเวณขอบมุมของพื้นที่ย่อยและกำลังกับพื้นที่ย่อยข้างเคียง เช่นพื้นที่ที่มีลักษณะเป็นรูปสี่เหลี่ยม จุดภาพดังกล่าว

จะถูกเฉลี่ยค่าไปกับเทรมเพลทแบบต่างๆ ส่งผลให้พื้นที่ที่ย่อยดังกล่าวถูกทำลายขอบเขตลงไป ทั้งนี้เนื่องจากกระบวนการทำซ้ำจนไม่เกิดการเปลี่ยนแปลงค่าของจุดใดๆ ในภาพ ในขั้นตอนการปรับเรียบ ซึ่งต่อมาได้มีการพัฒนาเทรมเพลทแบบใหม่โดย Nagao เพื่อปรับปรุงวิธีการปรับเรียบให้รองรับกับพื้นที่ที่มีรูปทรงเป็นมุมแหลมได้ โดยเทรมเพลทของ Nagao มีอยู่ 9 รูปแบบ แต่ละรูปแบบของเทรมเพลทจะมีจุดภาพที่พิจารณาปรากฏอยู่ในตำแหน่งต่างๆ ของเทรมเพลทด้วย เทรมเพลทของ Nagao แสดงได้ดังรูปที่ 3.2



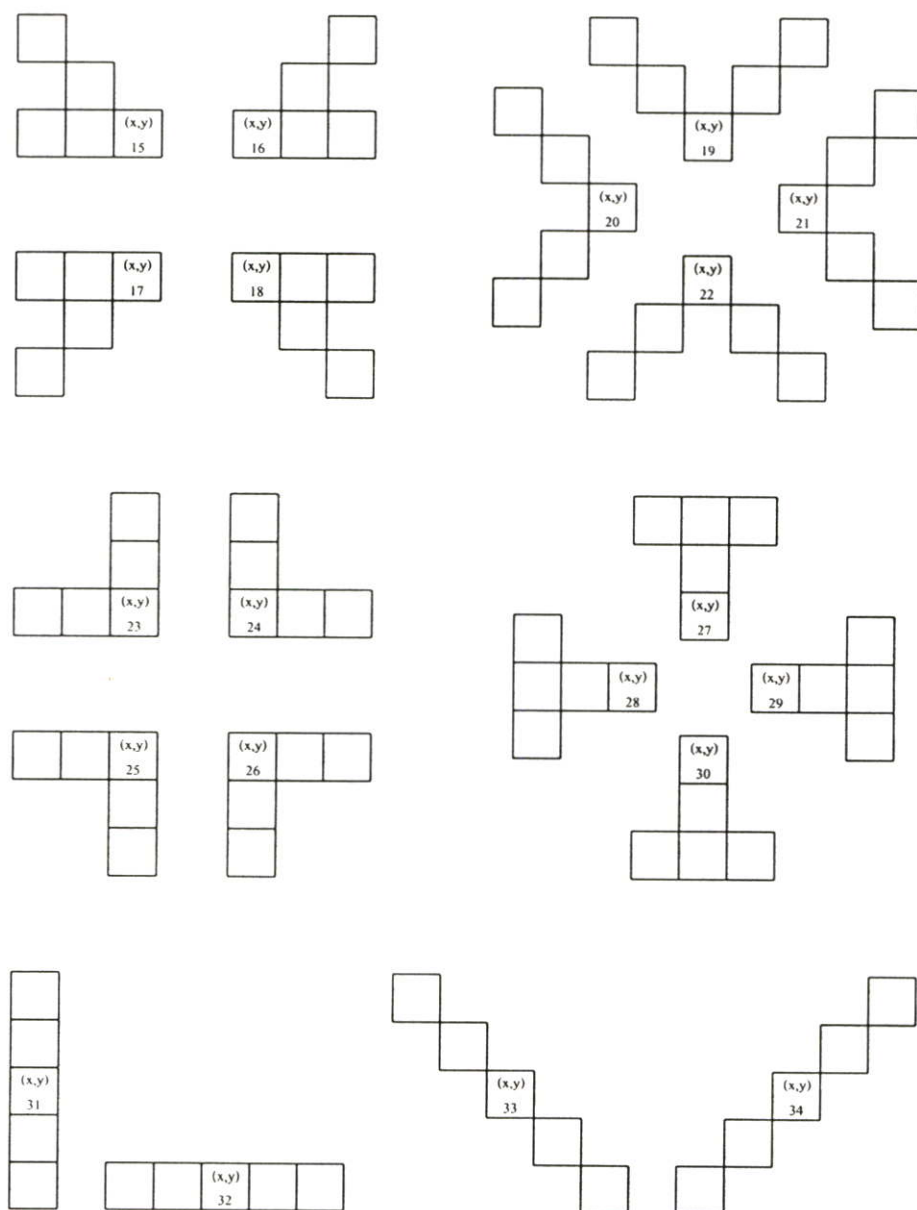
รูปที่ 3.1 รูปแบบเทรมเพลททั้ง 5 รูปแบบของ Tomata



รูปที่ 3.2 รูปแบบเทรมเพลททั้ง 9 รูปแบบของ Nagao

เทรมเพลทของ Nagao ซึ่งมีถึง 9 รูปแบบ จะทำให้ใช้เวลาในการประมวลผลสูง เนื่องจากมีขนาดใหญ่กว่า การคำนวณต้องทำแบบวนซ้ำทุกจุดภาพและทุกเทรมเพลท แต่ก็ส่งผลให้ภาพผลลัพธ์ที่ได้มีความคมชัดของขอบภาพเพิ่มขึ้นกว่าเทรมเพลทแบบ Tomata อย่างไรก็ตามถ้าหากพื้นที่ที่สนใจมีขนาดเล็กกว่าเทรมเพลทที่ใช้ในการปรับเรียบ ก็ยังคงทำให้พื้นที่นั้นถูกกลืนหายไปเช่นกัน เนื่องจากถูกยุบไปรวมกับพื้นที่ข้างเคียงที่มีขอบเขตใหญ่กว่า

จากแนวคิดของ Nagao นำมาออกแบบเทรมเพลทใหม่ เพื่อให้รองรับกับพื้นที่ย่อยที่มีขนาดเล็ก หรือผอมบางเป็นรูปทรงต่างๆ โดยรูปแบบของหน้ากาจะไม่ไปทำลายขอบเขตที่มีขนาดเล็ก แต่มีความสามารถในการกำจัดสัญญาณรบกวนได้ เทรมเพลทที่ออกแบบใหม่แสดงได้ดังรูปที่ 3.3



รูปที่ 3.3 รูปแบบเทรมเพลทที่ออกแบบขึ้นใหม่

### 3.3 กระบวนการรักษาขอบแบบปรับเรียบ

การรักษาขอบแบบปรับเรียบประกอบไปด้วยขั้นตอนหลายขั้นตอนในการคำนวณเพื่อหาค่าความเข้มของจุดภาพแต่ละจุดภาพใหม่ทั้งภาพ การคำนวณโดยแยกคำนวณตามแต่ละรูปแบบของเทรมเพลทแล้วจึงทำซ้ำกับเทรมเพลทรูปแบบอื่นๆจะทำให้ไม่ได้ค่าของจุดภาพที่เหมาะสมกับแต่ละขอบเขตพื้นที่ย่อย

เพื่อให้ค่าจุดภาพใหม่ที่ได้มีความเหมาะสมที่สุดและเป็นการลดขั้นตอนและจำนวนของข้อมูลในการจัดการ โปรแกรม ดังนั้นในวิทยานิพนธ์นี้ จึงได้ทำการคำนวณหาจุดภาพใหม่โดยใช้เทรมเพลทแบบของ Tomata ,แบบของ Nagao และแบบที่ได้ออกแบบขึ้นใหม่รวม 34 รูปแบบ ในการตรวจหาความเหมาะสมของขอบเขตพื้นที่ย่อยสำหรับแต่ละจุดภาพ ให้สำเร็จในขั้นตอนเดียว โดยสามารถอธิบายเป็นขั้นตอนตามลำดับดังนี้

#### 3.3.1 คำนวณค่าเฉลี่ยของข้อมูลทุกจุดภาพ ในทุกๆ เทรมเพลท จากสมการ

$$m(k) = \frac{1}{N_k} \sum_{i,j \in S_k} f(i,j) \quad ; k = 1,2,3,\dots,M \quad (3.1)$$

เมื่อ  $M$  หมายถึงจำนวนเทรมเพลท  
 $S_k$  หมายถึงสับเซตของจุดภาพในเทรมเพลทที่  $k^{\text{th}}$   
 $N_k$  หมายถึงจำนวนจุดภาพในเทรมเพลทที่  $k^{\text{th}}$   
 $f(i,j)$  หมายถึงค่าระดับสีเทาของจุดภาพที่  $(i,j)$

#### 3.3.2 คำนวณค่าเบี่ยงเบนมาตรฐานของแต่ละเทรมเพลทที่พิจารณาจากจุดภาพ $f(x,y)$ โดย

$$V(k) = \frac{1}{N_k} \sum_{i,j \in S_k} [f(i,j) - f(x,y)]^2 \quad ; k = 1,2,3,\dots,M \quad (3.2)$$

3.3.3 ตรวจหาค่าเบี่ยงเบนมาตรฐานต่ำสุดที่ได้จากเทรมเพลททั้ง 34 รูปแบบ

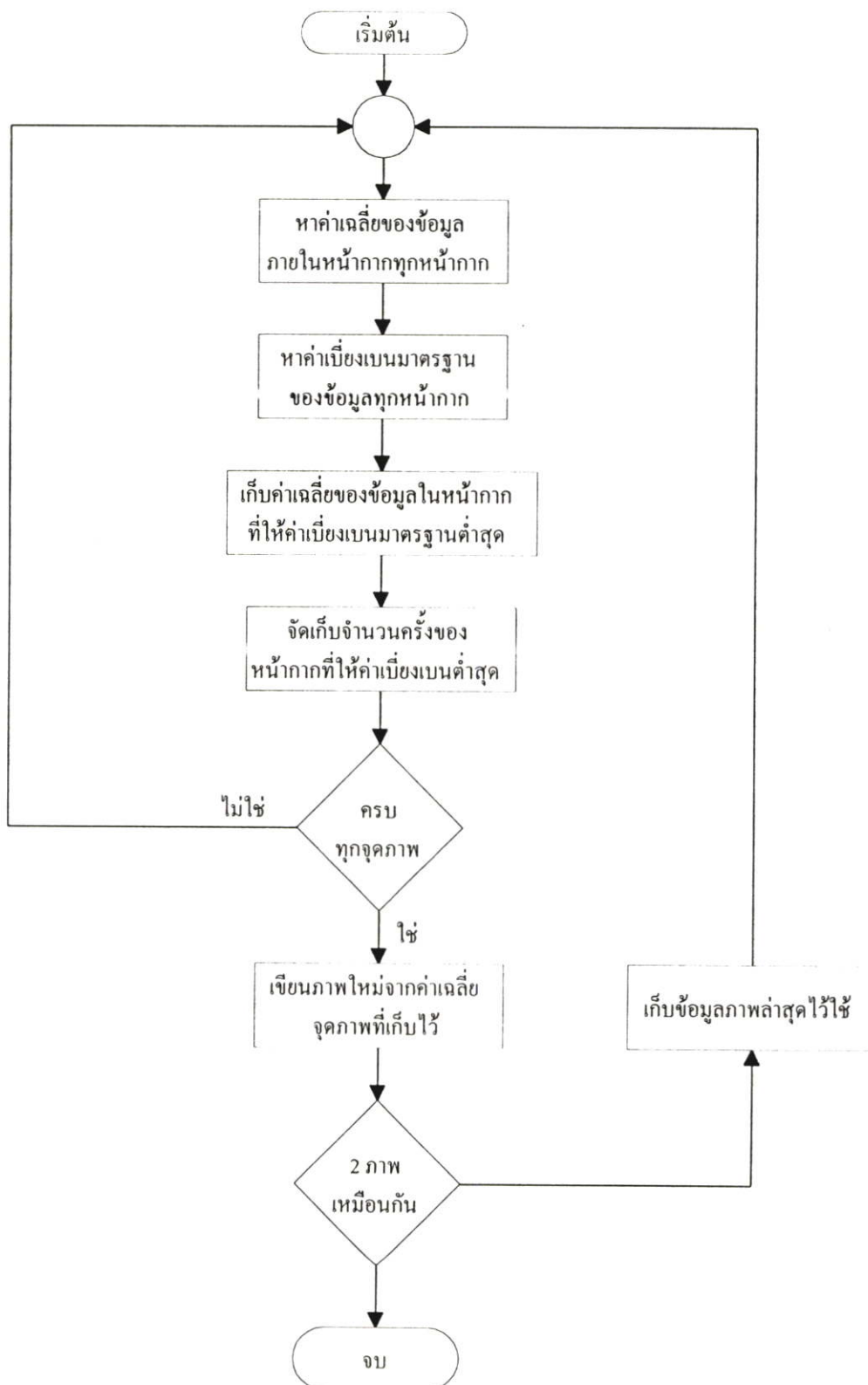
3.3.4 จัดเก็บค่าเฉลี่ยของเทรมเพลทที่ให้ค่าเบี่ยงเบนมาตรฐานต่ำที่สุดลงในจุดภาพ  $(x,y)$

3.3.5 จัดเก็บจำนวนครั้งของเทรมเพลทที่ให้ค่าเบี่ยงเบนมาตรฐานต่ำที่สุด

3.3.6 ทำซ้ำตั้งแต่ขั้นตอนที่ 3.3.1 ถึง 3.3.5 จนครบทุกจุดภาพ

3.3.7 ทำซ้ำขั้นตอนที่ 3.3.1 ถึง 3.3.6 จนค่าระดับสีเทาของทุกๆ จุดภาพไม่เปลี่ยนแปลง

3.3.8 ข้อมูลจุดภาพที่จัดเก็บไว้ในข้อ 3.3.4 ชุดท้ายสุด คือภาพที่ได้จากการรักษาขอบแบบปรับเรียบ



รูปที่ 3.4 แผนภูมิการรักษารูปแบบปรับเรียบ

### 3.4 ผลการทดลอง



รูปที่ 3.5 ภาพต้นแบบก่อนการปรับเรียบ



รูปที่ 3.6 ภาพหลังจากผ่านการรักษาขอบแบบปรับเรียบ

ตารางที่ 3.1 สถิติการถูกเลือกใช้งานของเทรมเพลทแบบต่างๆ

ลำดับที่	จำนวนครั้ง	ลำดับที่	จำนวนครั้ง	ลำดับที่	จำนวนครั้ง	ลำดับที่	จำนวนครั้ง
1	12114	9	3662	18	4663	26	4292
2	3706	10	3463	19	2020	27	6310
3	3029	11	2991	20	2042	28	5027
4	2239	12	3065	21	2010	29	6314
5	1963	13	3316	22	1920	30	4850
6	3721	15	4903	23	4424	31	5446
7	3628	16	4730	24	4562	32	3757
8	3468	17	4692	25	4414	33	2309
						34	2022

### 3.5 สรุปผลการทดลอง

รูปที่ 3.5 แสดงให้เห็นว่าขอบเขตพื้นที่ย่อยต่างๆ ได้ถูกปรับปรุงให้มีค่าระดับสีเทาใกล้เคียงกัน เมื่อเปรียบเทียบกับรูปที่ 3.4 ซึ่งจะส่งผลคือการแบ่งส่วนภาพ เนื่องจากมีขอบเขตของพื้นที่ย่อยที่ชัดเจนกว่า ลดความแตกต่างของค่าระดับสีของพื้นที่เล็กๆ ที่เรียงติดต่อกัน อันเกิดจากการเฉลี่ยค่าด้วยเทรมเพลทที่เหมาะสมที่สุดที่ได้เตรียมไว้

ตารางที่ 3.1 แสดงให้เห็นถึงสถิติการเลือกใช้เทรมเพลทในขั้นตอนการปรับเรียบ เทรมเพลทที่ 14 ไม่ถูกใช้งานเนื่องจากซ้ำกับเทรมเพลทที่ 5 ตารางที่ 3.1 ได้จากการกำหนดให้โปรแกรมทำงานเพียง 5 รอบ เพื่อสังเกตการถูกเลือกใช้งานของเทรมเพลท หากมีบางเทรมเพลทถูกเลือกใช้งานน้อยมาก แสดงว่าเทรมเพลทนั้นๆ ไม่เหมาะสมที่จะใช้งานสำหรับภาพที่กำลังกระทำอยู่ จึงไม่ควรให้โปรแกรมทำงานจนเสร็จ เนื่องจากการเสียเวลาไปโดยใช่เหตุ หากเทรมเพลทที่ไม่ถูกเลือกใช้งานมีหลายเทรมเพลทแสดงว่าเทรมเพลทส่วนใหญ่ไม่เหมาะสมกับขอบเขตของพื้นที่ย่อยของภาพเลย จำเป็นต้องออกแบบใหม่ให้เหมาะสมกับแต่ละภาพต่อไป

การออกแบบเทรมเพลทให้เล็ก แต่มีหลากหลายรูปแบบ จะให้ผลของการปรับเรียบที่ดีกว่า เนื่องจากการกระจายของรูปแบบสำหรับใช้เปรียบเทียบหาขอบเขตที่เหมาะสมมากกว่า อย่างไรก็ตามการมีจำนวนเทรมเพลทหลายๆ นั้น จะส่งผลถึงเวลาที่ใช้ในการคำนวณโดยตรง ดังนั้นจึงขึ้นอยู่กับคุณภาพของภาพผลลัพธ์ที่ต้องการว่ายอมรับได้เพียงใด

## บทที่ 4

# การแบ่งส่วนภาพโดยทฤษฎีกราฟ

### 4.1 คำนำ

ภาพถ่ายขาวเทียมนั้นเป็นภาพที่ถูกจัดเก็บไว้ในลักษณะของข้อมูลดิจิทัล การแสดงผลจะต้องแสดงออกทางจอทีวี หรือจอมอนิเตอร์ โดยอยู่ในรูปของอาร์เรย์สองมิติ  $(x, y)$  ซึ่งจะเป็นการกำหนดจำนวนเส้นและจำนวนจุดต่อเส้นของภาพ เช่นภาพมีความละเอียดขนาด 256 เส้น และเส้นละ 256 จุด จะเรียกว่าภาพมีความละเอียด  $256 \times 256$  จุดภาพ โดยที่จุดภาพแต่ละจุดสามารถแสดงความแตกต่างของระดับสีเทา (gray level) ได้เท่ากับขนาดของจำนวนบิตข้อมูลที่ใช้เก็บข้อมูลหนึ่งจุดภาพ ในภาพถ่ายขาวเทียมนักกำหนดให้มีความแตกต่างของค่าระดับสีเทาในหนึ่งจุดภาพเป็น 256 ระดับ สำหรับภาพ 1 แบนด์ ซึ่งถือว่าเพียงพอในการแสดงภาพ เพราะตาของมนุษย์ไม่สามารถที่จะแยกความแตกต่างของระดับสีเทาได้ถึง 256 ระดับ

การแบ่งส่วนภาพกระทำเพื่อแยกหรือกลั่นกรองส่วนที่สำคัญของภาพออกมาให้เห็นเด่นชัดขึ้น โดยการรวมส่วนของภาพที่มีค่าระดับความเข้มใกล้เคียงกันหรือเท่ากันให้เป็นส่วนเดียวกัน และแยกส่วนที่มีค่าความเข้มที่ต่างกันมากออกเป็นอีกส่วนหนึ่ง ซึ่งค่าระดับความแตกต่างที่มีค่าน้อยนั้นมนุษย์ไม่สามารถจะแยกแยะได้ แต่เมื่อผ่านกระบวนการแบ่งส่วนภาพแล้ว จะทำให้สามารถแยกแยะด้วยสายตาได้ง่ายขึ้น

ทฤษฎีกราฟถูกใช้อย่างกว้างขวางในการเข้ารหัสเพื่อบีบอัดข้อมูลภาพ ในที่นี้จะประยุกต์ใช้ทฤษฎีกราฟมาจัดการแบ่งส่วนภาพ เนื่องจากเป็นวิธีที่มีการนำเอาข้อมูลส่วนใหญ่ของภาพมาใช้เป็นเกณฑ์ในการตัดสินใจแยกหรือรวมแต่ละส่วนย่อยๆ ดังนั้นการแบ่งส่วนภาพโดยทฤษฎีกราฟจึงสามารถให้ภาพที่มีความกลมกลืนกันในแต่ละพื้นที่ย่อย ตามลักษณะโครงสร้างของภาพเดิม และยังมีคุณสมบัติเด่นคือสามารถที่จะกำหนดจำนวนพื้นที่ย่อยได้ตามต้องการ

### 4.2 การจัดข้อมูลภาพให้อยู่ในลักษณะกราฟ [13,14]

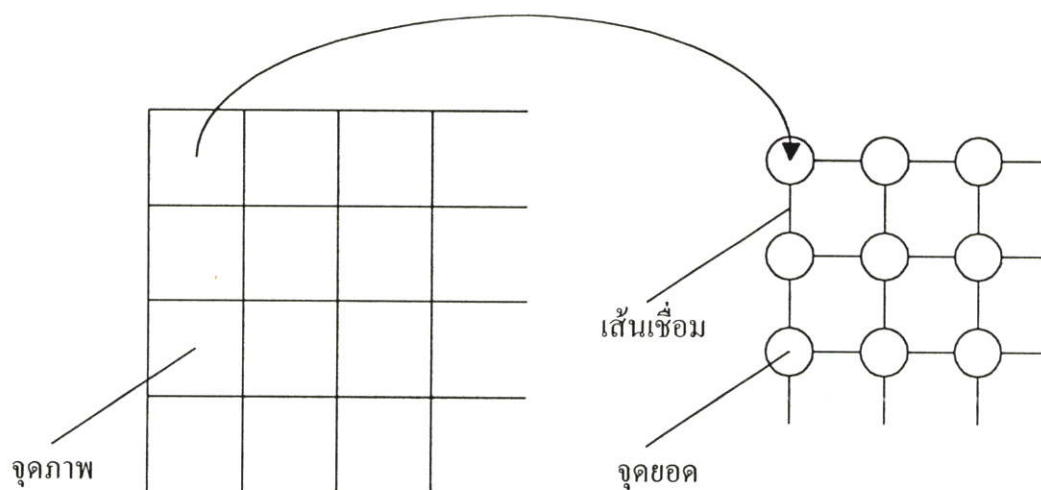
ความเข้มของข้อมูลภาพแต่ละจุดภาพจะถูกแปลงไปเป็นค่าจุดยอดของกราฟแต่ละจุดด้วยภาพที่มีความละเอียด  $256 \times 256$  จุดภาพ จะถูกแปลงเป็นกราฟที่มีจุดยอด  $256 \times 256$  ด้วยเช่นกัน

เมื่อให้ความเข้มหรือค่าระดับสีเทาของจุดภาพที่ตำแหน่ง  $(x, y)$  คือ  $f(x, y)$  แล้วค่าน้ำหนักจุดยอดของกราฟ :  $V_i$  จะเป็น

$$V_i = f(x, y) \tag{4.1}$$

โดยที่  $x, y$  ถูกแปลงไปเป็น  $i$  ในลักษณะจุดต่อจุด (one-to-one mapping) จากนั้นทำการหาค่าน้ำหนักตัวเชื่อมของจุดยอดต่างๆ ในที่นี้จะใช้การเชื่อมต่อแบบ 4 ทิศทาง กระทำกับจุดยอดของ

กราฟเฉพาะจุดยอดที่อยู่ใกล้กันที่สุดเท่านั้น เพื่อลดขั้นตอนและขนาดของข้อมูลที่จะเกิดขึ้น ขณะประมวลผล รูปที่ 4.1 แสดงการแปลงข้อมูลภาพไปเป็นกราฟ



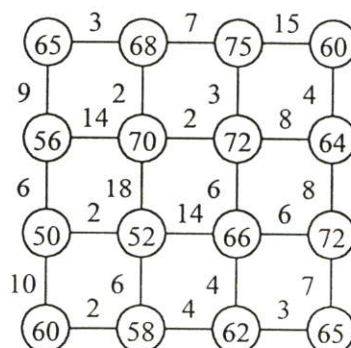
รูปที่ 4.1 การจัดข้อมูลภาพให้อยู่ในลักษณะกราฟ

ค่าน้ำหนักเส้นเชื่อมจะหาได้จากค่าสัมบูรณ์ ของค่าความแตกต่างระหว่างจุดยอดของกราฟที่อยู่ประชิดติดกัน ซึ่งเป็นการวัดความเหมือนหรือความใกล้เคียงของความเข้มระหว่างจุดภาพคู่ที่เส้นเชื่อมต่อยุ่กันเอง ให้  $e_{i,j}$  เป็นความเข้มของเส้นเชื่อม จะได้ว่า

$$e_{i,j} = |v_i - v_j| \quad (4.2)$$

จากสมการที่ 4.1 และ 4.2 สามารถนำมาสร้างกราฟที่มีค่าจุดยอดและค่าน้ำหนักเส้นเชื่อม แสดงเป็นตัวอย่างได้ดังรูปที่ 4.2

65	68	75	60
56	70	72	64
50	52	66	72
60	58	62	65



รูปที่ 4.2 ค่าจุดยอดและน้ำหนักเส้นเชื่อมของกราฟที่สมบูรณ์ของภาพขนาด  $4 \times 4$  จุดภาพ

### 4.3 การหาซ็อดเตสท์สแพนนิ่งทรี [11]

ซ็อดเตสท์สแพนนิ่งทรี (Shortest Spanning Tree : SST) เป็นการเปลี่ยนเส้นเชื่อมเส้นที่มีน้ำหนักน้อยที่สุดให้เป็นเส้นเชื่อมที่มีค่าน้ำหนักเชื่อมต่อเป็นศูนย์และทำการเฉลี่ยค่าน้ำหนักจุดยอดคู่ที่ถูกเปลี่ยนเส้นเชื่อมให้มีค่าเท่ากัน กระทำการเปลี่ยนเส้นเชื่อมเส้นที่ตรวจพบว่ามีค่าน้ำหนักเชื่อม น้อยที่สุดเส้นต่อไปและเฉลี่ยค่าน้ำหนักจุดยอดอีก จนกระทั่งค่าน้ำหนักจุดยอดของทั้งภาพมีค่าเท่ากัน เส้นเชื่อมที่ยังคงอยู่จะเรียกว่า ซ็อดเตสท์สแพนนิ่งทรี

รายละเอียดของวิธีการหาซ็อดเตสท์สแพนนิ่งทรี มีดังต่อไปนี้

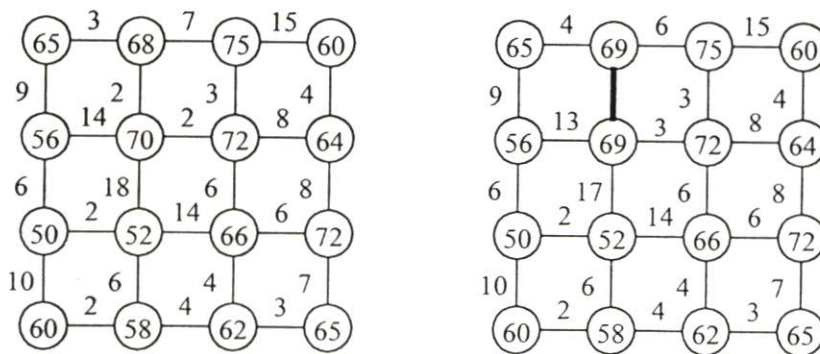
- 4.3.1 จัดเรียงลำดับค่าน้ำหนักเส้นเชื่อมของกราฟจากค่าต่ำไปหาค่าสูง
- 4.3.2 ตรวจสอบหาเส้นเชื่อมที่มีค่าน้ำหนักน้อยที่สุด
- 4.3.3 เปลี่ยนค่าน้ำหนักเส้นเชื่อมจุดยอดที่ตรวจพบให้เป็นศูนย์ รวมกันขึ้นเป็นสแพนนิ่งทรี
- 4.3.4 เฉลี่ยค่าน้ำหนักของกลุ่มจุดยอดระหว่างเส้นเชื่อม ตามความสัมพันธ์

$$Vertex_{mean} = \frac{\sum_{i=1}^N x_i}{N} \tag{4.3}$$

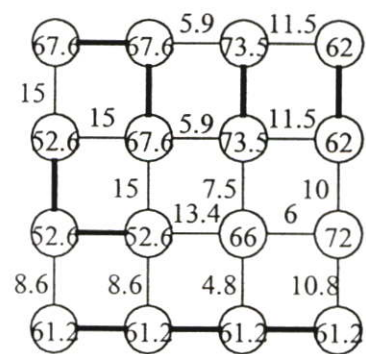
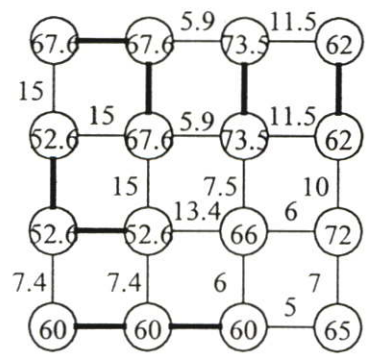
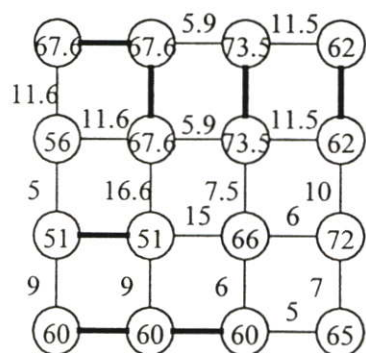
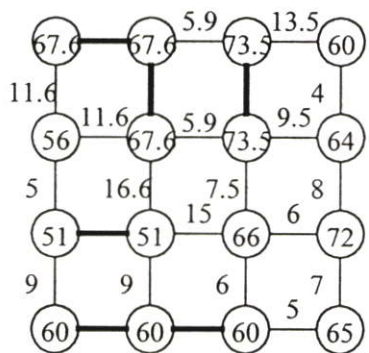
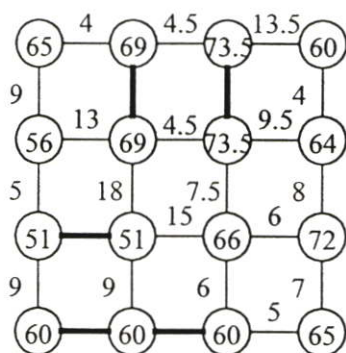
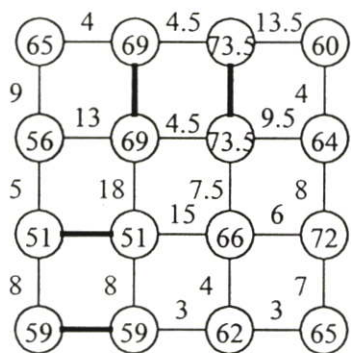
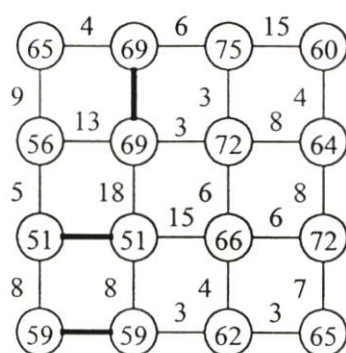
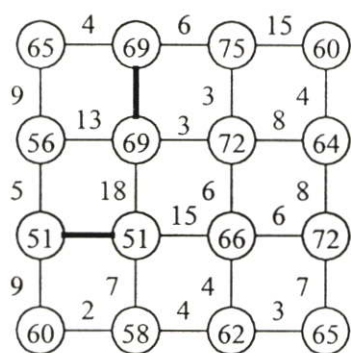
เมื่อ  $N$  คือ จำนวนจุดยอดที่ถูกปรับค่าน้ำหนักเส้นเชื่อม  
 $x_i$  คือ ค่าน้ำหนักจุดยอดของกราฟแต่ละจุดที่ถูกปรับค่าน้ำหนักจุดยอด

- 4.3.5 เมื่อเชื่อมเส้นและเฉลี่ยค่าน้ำหนักจุดยอดแล้วเกิดวนรอบให้ตัดเส้นเชื่อมที่ทำให้เกิดวนรอบ
- 4.3.6 ทำซ้ำตั้งแต่ขั้นตอนที่ 4.3.1 จนกระทั่งค่าน้ำหนักจุดยอดมีเพียงค่าเดียว

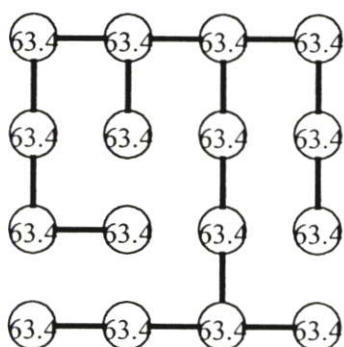
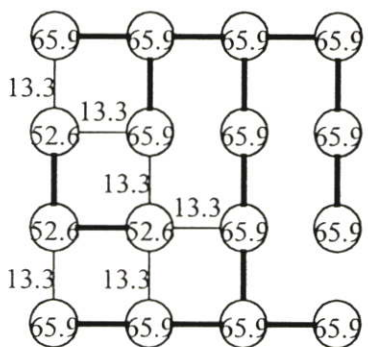
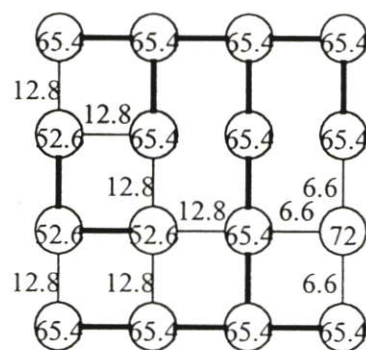
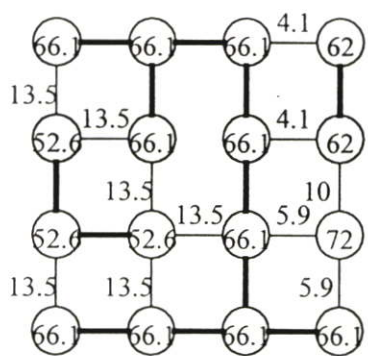
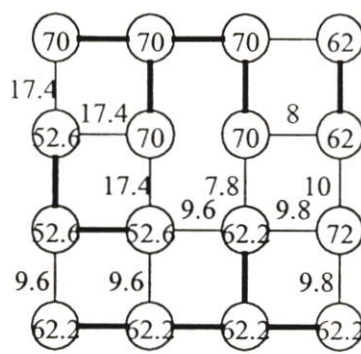
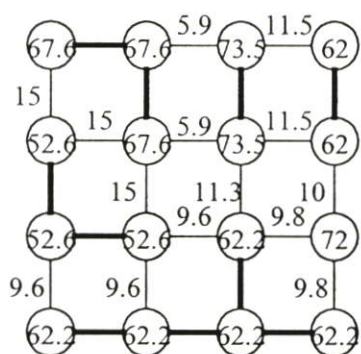
ขั้นตอนของวิธีการหาซ็อดเตสท์สแพนนิ่งทรีแสดงเป็นแผนภูมิการทำงานได้ดังรูปที่ 4.4 และรูปที่ 4.3 แสดงรายละเอียดการสร้างทรีของรูปที่ 4.2 จนได้เป็นซ็อดเตสท์สแพนนิ่งทรีตามลำดับ



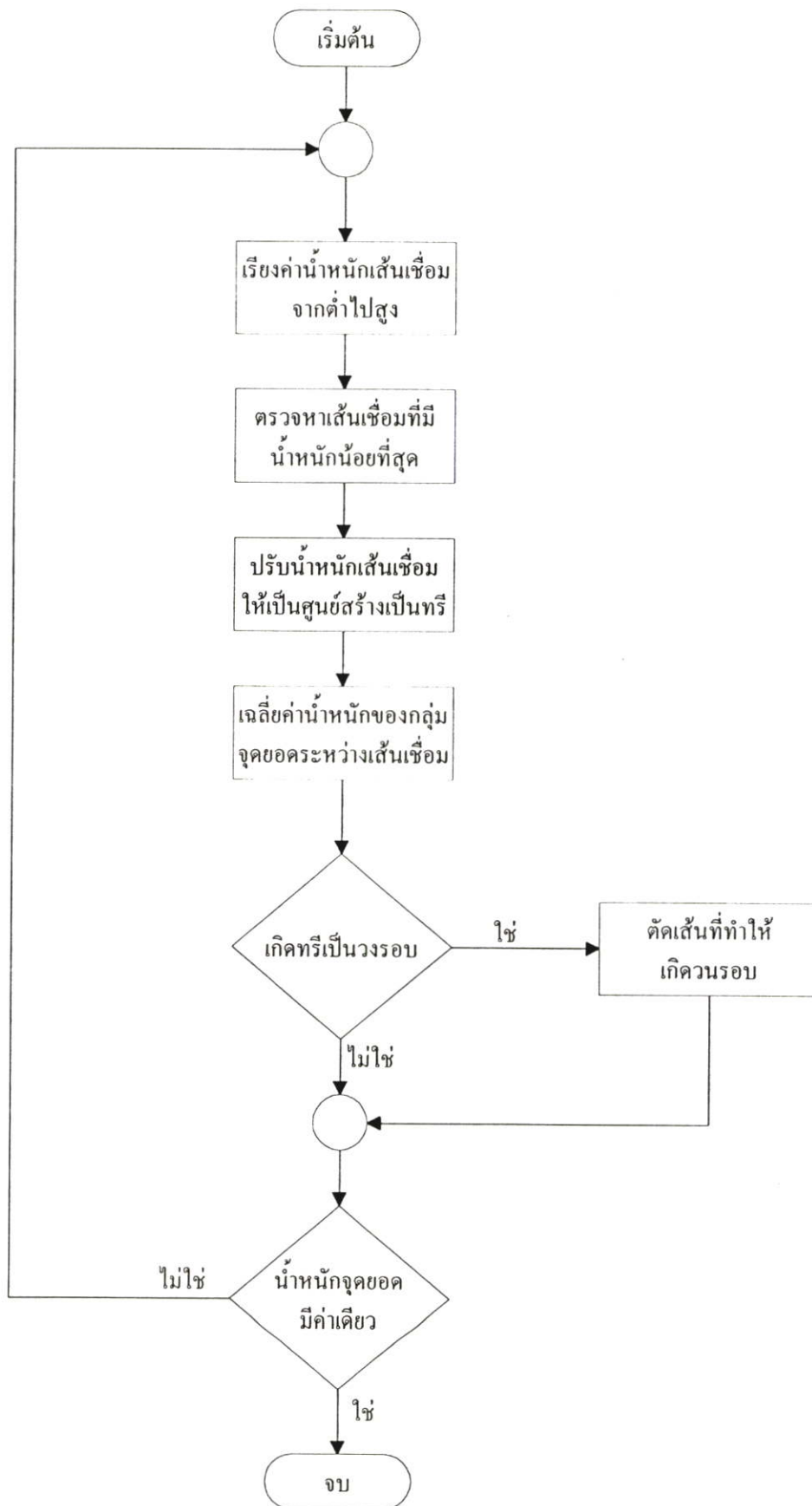
รูปที่ 4.3 ขั้นตอนการหาซ็อดเตสท์สแพนนิ่งทรีของกราฟ



รูปที่ 4.3 (ต่อ)



รูปที่ 4.3 (ต่อ)



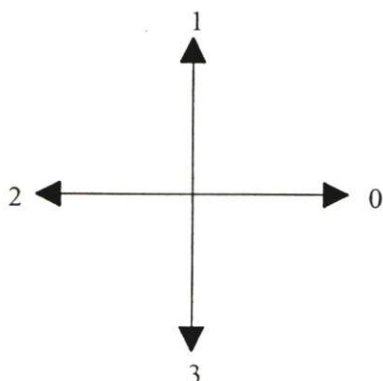
รูปที่ 4.4 แผนภูมิการหาซ้อดเตสท์สแพนนิ่งทรีของกราฟ

#### 4.4 การเข้ารหัสภาพเซกเมนต์ชัน (Segmentation image coding) [10]

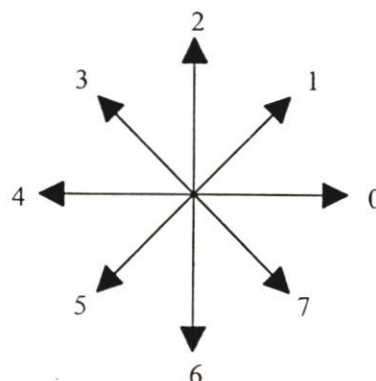
ส่วนสำคัญในการหาเส้นแสดงขอบเขตของภาพแต่ละส่วน จะอาศัยการเข้ารหัสเพื่อเก็บรายละเอียดลักษณะการเชื่อมต่อ (connectivity) ของขอบของจุดภาพ โดยสามารถแยกพิจารณาได้เป็น 2 รูปแบบดังนี้

แบบที่ 1 เป็นแบบ 4-connectivity ซึ่งเราจะพิจารณาในรูปของทิศทางได้ 4 ทิศทาง สามารถแทนค่าตัวเลขให้แก่รหัสทิศทางได้เป็น 0, 1, 2 และ 3 ตามลำดับ ดังรูปที่ 4.5 (ก)

แบบที่ 2 เป็นแบบ 8-connectivity ซึ่งเราจะพิจารณาในรูปของทิศทางได้ 8 ทิศทาง สามารถแทนค่าตัวเลขให้แก่รหัสทิศทางได้เป็น 0, 1, 2, 3, 4, 5, 6 และ 7 ตามลำดับ ดังรูปที่ 4.5 (ข)



(ก) แบบ 4-connectivity



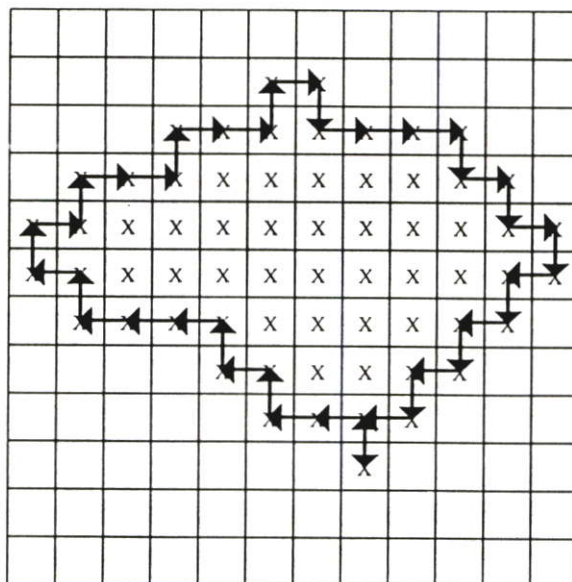
(ข) แบบ 8-connectivity

รูปที่ 4.5 ทิศทางและค่ารหัสแบบ 4 ทิศทางและ 8 ทิศทาง

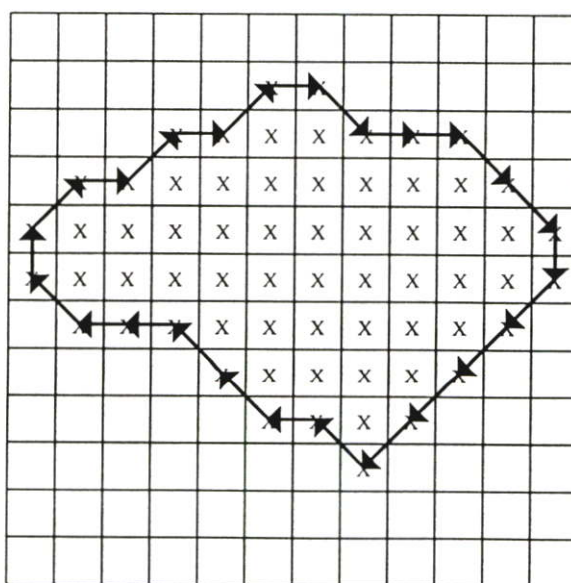
รูปแบบของการหาขอบทั้งแบบ 4 ทิศทาง และ 8 ทิศทาง เมื่อนำไปใช้ในการหาขอบของภาพ จะทำให้ได้จำนวนรหัสที่แตกต่างกันกล่าวคือ แบบ 4 ทิศทาง จะมีจำนวนรหัสมากกว่า ดังตัวอย่างในรูปที่ 4.6 และ 4.7

จากรูปที่ 4.6 และ 4.7 จะเห็นได้ว่าจำนวนรหัสของการตามขอบแบบ 8 ทิศทาง จะมีจำนวนน้อยกว่าการใช้รหัสแบบ 4 ทิศทาง และถึงแม้ว่าค่าของรหัสแบบ 8 ทิศทาง สามารถแทนด้วยค่าทางดิจิทัลขนาด 3 บิต ในขณะที่แบบ 4 ทิศทาง แทนด้วยค่าขนาด 2 บิต แต่เมื่อรวมรหัสทั้งหมด การติดตามขอบแบบ 8 ทิศทาง ก็ยังมีจำนวนรหัสน้อยกว่าแบบ 4 ทิศทาง แต่ในวิทยานิพนธ์นี้เลือกใช้การต่อเนื่องของการหาขอบแบบ 4 ทิศทางในการเข้ารหัสขอบของเซกเมนต์ เนื่องจากว่าภาพที่ได้จากการทำเซกเมนต์ชันบางภาพอาจให้ภาพที่มีลักษณะไม่สามารถใช้การต่อเนื่องแบบ 8 ทิศทางได้ ดังตัวอย่างในรูปที่ 4.8 จากรูปจะเห็นว่าในกรณีที่ภาพมีส่วนที่ยื่นออกมามีลักษณะเป็นจุดภาพที่มี

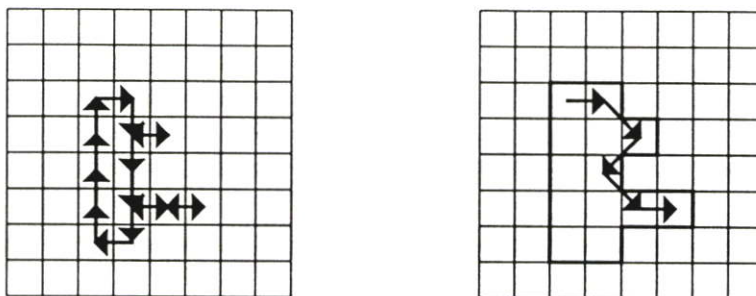
ความหนาเพียง 1 จุดภาพ การใช้การต่อเนื่องแบบ 8 ทิศทาง ไม่สามารถที่จะเข้ารหัสให้ครบรอบได้ (Loop) ในขณะที่รหัสแบบ 4 ทิศทางสามารถทำได้ และเมื่อรหัสลูกโซ่ของการติดตามขอบแบบ 4 ทิศทางหนึ่งตัวสามารถใช้หน่วยความจำในการเก็บเพียง 2 บิต ดังนั้นหน่วยความจำขนาด 1 ไบท์ จึงสามารถเก็บค่าของรหัสขอบได้ 4 ตัว ทำให้จัดการกับหน่วยความจำได้ง่ายกว่า ส่วนที่สองของรหัส คือค่าจุดเริ่มต้นของขอบที่เริ่มเข้ารหัส ซึ่งค่านี้จะนำไปใช้เป็นตัวกำหนดจุดเริ่มต้นของขอบ ในขั้นตอนของการสร้างขอบภาพต่อไป



รูปที่ 4.6 ตัวอย่างรหัสลูกโซ่ของเซกเมนต์แบบ 4 ทิศทาง



รูปที่ 4.7 ตัวอย่างรหัสลูกโซ่ของเซกเมนต์แบบ 8 ทิศทาง



รูปที่ 4.8 ข้อจำกัดของการใช้รหัสแบบ 8 ทิศทาง

การเข้ารหัสขอบจะเริ่มต้นจากจุดซ้ายบนสุดของเซกเมนต์ แล้วเคลื่อนที่ไปทิศทางตามเข็มนาฬิกาจนกว่าจะมารอบพบกับจุดเริ่มต้นใหม่ โดยจัดเก็บด้วยข้อมูลดิจิทัลขนาด 2 บิต ตามข้อกำหนดดังนี้

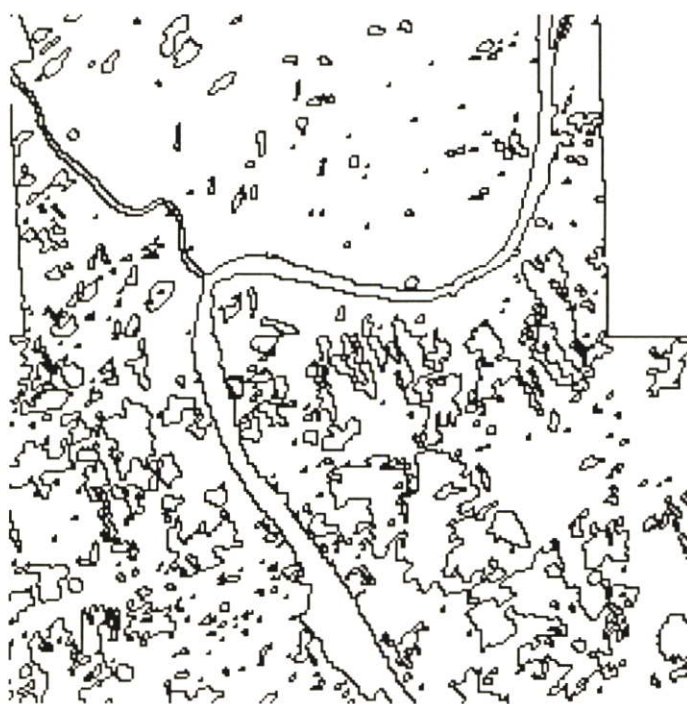
- 00 เคลื่อนที่ไปทางทิศตะวันออก
- 01 เคลื่อนที่ไปทางทิศเหนือ
- 10 เคลื่อนที่ไปทางทิศตะวันตก
- 11 เคลื่อนที่ไปทางทิศใต้

จากรูปที่ 4.8 เมื่อใช้การเข้ารหัสแบบ 4 ทิศทาง จะได้ค่ารหัสลูกโซ่ของรหัสขอบแสดงเป็นข้อมูลเลขฐานสิบดังนี้ 00303230032232211111

## 4.5 ผลการทดลอง



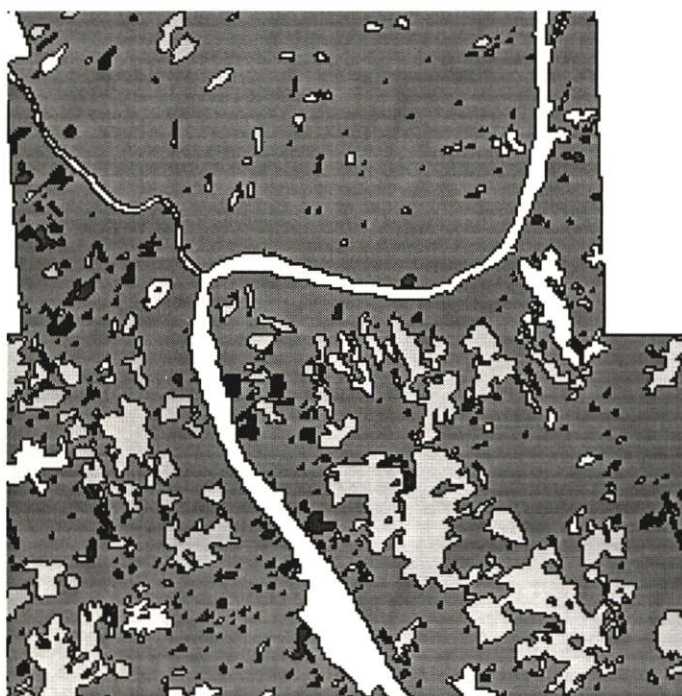
รูปที่ 4.9 ผลของการเข้ารหัสขอบแบบ 4 ทิศทาง เมื่อแบ่งภาพออกเป็น 1000 ส่วน ของภาพที่ไม่ผ่านการรักษาขอบ



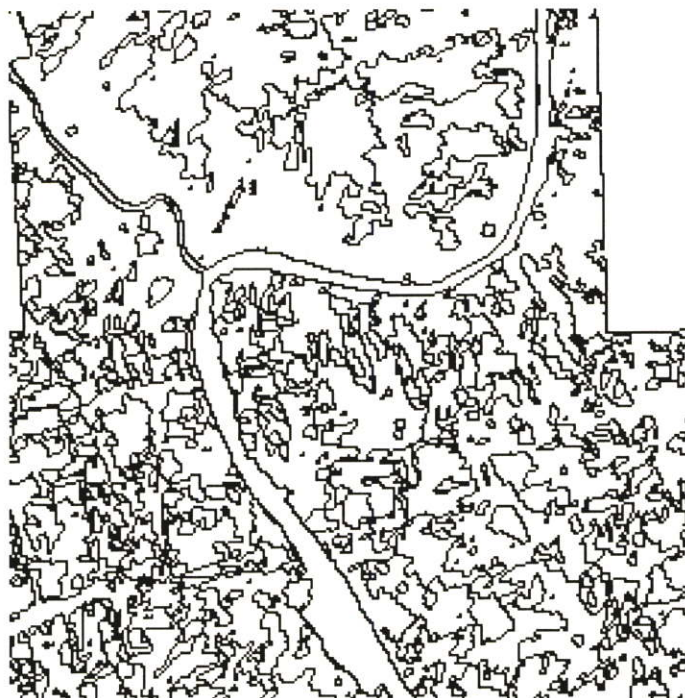
รูปที่ 4.10 ผลของการเข้ารหัสขอบแบบ 4 ทิศทาง เมื่อแบ่งภาพออกเป็น 500 ส่วน ของภาพที่ไม่ผ่านการรักษาขอบ



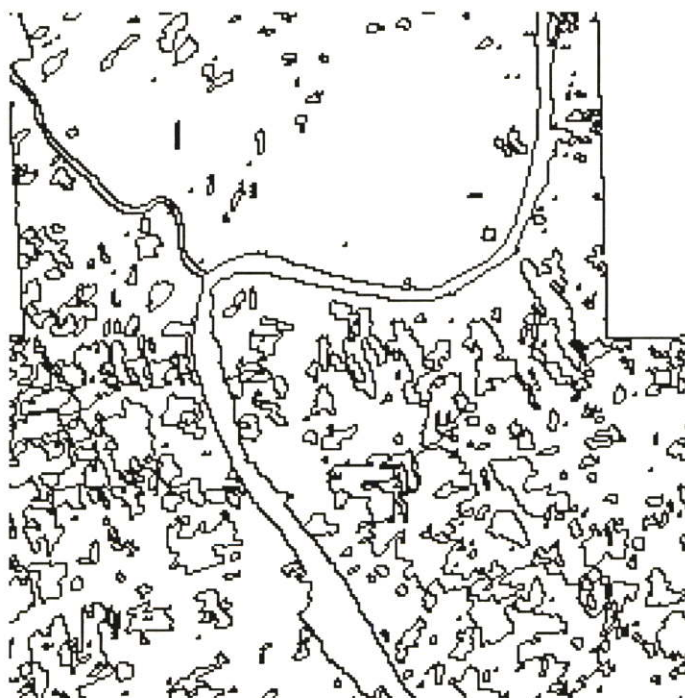
รูปที่ 4.11 ผลการแบ่งส่วนภาพด้วยทฤษฎีกราฟ เมื่อแบ่งภาพออกเป็น 1000 ส่วน ของภาพที่ไม่ผ่านการรักษาขอบ



รูปที่ 4.12 ผลการแบ่งส่วนภาพด้วยทฤษฎีกราฟ เมื่อแบ่งภาพออกเป็น 500 ส่วน ของภาพที่ไม่ผ่านการรักษาขอบ



รูปที่ 4.13 ผลของการเข้ารหัสขอบแบบ 4 ทิศทาง เมื่อแบ่งภาพออกเป็น 1000 ส่วน ของภาพ ที่ผ่านการรักษาขอบแบบปรับเรียบ



รูปที่ 4.14 ผลของการเข้ารหัสขอบแบบ 4 ทิศทาง เมื่อแบ่งภาพออกเป็น 500 ส่วน ของภาพ ที่ผ่านการรักษาขอบแบบปรับเรียบ



รูปที่ 4.15 ผลการแบ่งส่วนภาพด้วยทฤษฎีกราฟ เมื่อแบ่งภาพออกเป็น 1000 ส่วน ของภาพ ที่ผ่านการรักษาขอบแบบปรับเรียบ



รูปที่ 4.16 ผลการแบ่งส่วนภาพด้วยทฤษฎีกราฟ เมื่อแบ่งภาพออกเป็น 500 ส่วน ของภาพ ที่ผ่านการรักษาขอบแบบปรับเรียบ

#### 4.6 สรุปผลการทดลอง

การแบ่งส่วนภาพด้วยทฤษฎีกราฟมีข้อเด่นคือ สามารถกำหนดจำนวนพื้นที่ย่อยที่ต้องการได้ และในขั้นตอนการคำนวณมิได้ใช้สมการทางคณิตศาสตร์ที่ยุ่งยากซับซ้อนแต่อย่างใด เพียงแต่ต้องทำการจัดเก็บข้อมูลในขั้นตอนหาช้อคเดสท์สแพนนิ่งทรีเป็นลำดับให้ดี เพราะเป็นหัวใจในการแบ่งส่วนภาพด้วยวิธีนี้ การตัดและเฉลี่ยค่าเส้นเชื่อมเป็นเพียงการบวกและหาร ทำให้ได้ผลลัพธ์อย่างรวดเร็ว ผลการทดลองแสดงการแบ่งส่วนภาพของภาพจากการต่อแบบหลายความละเอียดที่ยังไม่ได้ผ่านการรักษาขอบและที่ผ่านการรักษาขอบแล้ว โดยแยกออกเป็น 1000 และ 500 ส่วน ตามลำดับภาพที่ผ่านการรักษาขอบแบบปรับเรียบแล้ว จะสามารถแบ่งส่วนของภาพได้ละเอียดกว่าดังจะเปรียบเทียบให้เห็นได้จากรูปที่ 4.9 และรูปที่ 4.14 ผลการแบ่งส่วนภาพเป็น 500 ส่วน ของภาพที่ผ่านการรักษาขอบแบบปรับเรียบแล้ว (รูปที่ 4.14) ให้รายละเอียดพื้นที่ย่อยได้เทียบเคียงกับผลการแบ่งส่วนภาพที่ไม่ผ่านการรักษาขอบ (รูปที่ 4.9) ที่ต้องแบ่งถึง 1000 ส่วน

## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

ในการประมวลผลภาพถ่ายดาวเทียม หน่วยความจำเป็นสิ่งจำเป็นสำหรับการจองพื้นที่เพื่อใช้ในการจัดการกับภาพที่มีขนาดใหญ่ หากยังมีขั้นตอนในการจัดการหลายขั้นตอน ก็ยังทำให้พื้นที่ที่ทั้งที่ใช้เก็บข้อมูลภาพและจัดเก็บข้อมูลต่างๆ มากมายยิ่งขึ้นไปด้วย ต้องระมัดระวังในการจัดการให้ดี เนื่องจากหากเกิดความผิดพลาดใดในขั้นตอนการทำงาน ยากที่จะตรวจสอบจากภาพผลลัพธ์ได้ ในวิทยานิพนธ์ฉบับนี้จึงได้แบ่งส่วนการทำงานในแต่ละขั้นตอนไว้ ทั้งนี้เพื่อให้ง่ายแก่การตรวจสอบผลการคำนวณต่างๆ และสะดวกในการจัดการหน่วยความจำ

การต่อภาพแบบหลายความละเอียด โดยเฉพาะหากภาพมีคุณสมบัติแตกต่างกันมาก จำเป็นต้องมีการปรับปรุงคุณภาพของภาพทั้งสองให้ใกล้เคียงกันก่อน มิฉะนั้นจะทำให้ภาพผลลัพธ์ภายหลังการต่อขาดความกลมกลืนยากแก่การแปลความหมายจากภาพ การแก้ความเพี้ยนเชิงเรขาคณิตในวิทยานิพนธ์ฉบับนี้เป็นการปรับปรุงสัดส่วนของภาพสองภาพให้มีสัดส่วนที่เหมือนกัน ซึ่งยังมีใช้สัดส่วนจริงเมื่อเทียบกับพิกัดภาคพื้นดิน อย่างไรก็ตามหลักการที่นำเสนอสามารถนำไปประยุกต์ใช้ได้กับคู่ภาพใดๆ ที่ต้องการปรับปรุงสัดส่วนของภาพให้เหมือนกัน

การรักษาขอบแบบปรับเรียบเป็นการปรับปรุงคุณภาพอย่างหนึ่งของภาพภายหลังการต่อที่จะช่วยให้ขอบเขตพื้นที่ย่อยต่างๆ ภายในภาพมีความกลมกลืน รวมถึงช่วยลดสัญญาณรบกวนลงไปได้ แต่ยังคงรักษาขอบเขตของพื้นที่ย่อยต่างๆ เอาไว้ได้ การสร้างภาพใหม่ด้วยการแทนค่าเฉลี่ยของข้อมูลที่ถูกละทิ้งบริเวณด้วยเทรมเพลทรูปแบบต่างๆ จะทำให้ได้ภาพผลลัพธ์ที่มีขอบเขตชัดเจนกว่าเดิมและเป็นผลดีต่อการแบ่งส่วนภาพเพื่อแปลความหมายต่อไป รูปแบบเทรมเพลทที่พัฒนาขึ้นใหม่จากแนวคิดของ Tomata และ Nagao นั้น [2,7] มีรูปแบบจากการพิจารณาขอบเขตย่อยของภาพที่จะทำการปรับปรุง หากจะนำไปใช้เพื่อพัฒนาภาพใดๆ ควรจะพิจารณาแนวทางขอบเขตของภาพนั้นๆ ด้วย ทั้งนี้เพื่อให้ได้เทรมเพลทที่เหมาะสมกับภาพผลลัพธ์ที่ต้องการปรับปรุง ขนาดของเทรมเพลทที่ใช้จะส่งผลต่อความเร็วในการคำนวณอย่างมาก และจะส่งผลให้ภาพผลลัพธ์มีคุณภาพด้อยลงกว่าเดิมหากใช้อย่างขาดความระมัดระวัง ดังนั้นจึงควรตรวจสอบการถูกเลือกใช้ของเทรมเพลทแต่ละรูปแบบด้วย ทั้งนี้เพื่อให้ประหยัดเวลาในการคำนวณและเพิ่มความมั่นใจว่าจะได้ภาพผลลัพธ์ที่มีคุณภาพดีกว่าเดิม

การแบ่งส่วนภาพโดยทฤษฎีกราฟ เป็นการอาศัยความแตกต่างของค่าระดับสีเทาของจุดภาพในบริเวณที่ติดต่อกันมาใช้ในการแบ่งส่วน โดยทำการจัดเก็บข้อมูลค่าความแตกต่างระหว่างจุดภาพเอาไว้ทั้งภาพแล้วทำการยุบรวมจุดภาพที่มีค่าความแตกต่างของค่าระดับสีเทาน้อยที่สุดเข้าไว้ด้วยกัน จัดเก็บลำดับที่ในการยุบรวมจุดภาพเอาไว้ การแบ่งส่วนก็คือการตัดแนวที่ได้ทำการเก็บข้อมูลเอาไว้ล่วงหน้าแล้ว ดังนั้นจึงสามารถที่จะกำหนดได้ว่าจะแบ่งส่วนของภาพออกเป็นกี่ส่วน ในขั้นตอน

การคำนวณไม่มีการใช้คณิตศาสตร์ชั้นสูง ดังนั้นจึงสามารถใช้ได้ดีกับเครื่องคอมพิวเตอร์ที่สมรรถนะไม่สูงมากนัก และใช้เวลาไม่นาน การแบ่งส่วนภาพโดยทฤษฎีกราฟสามารถใช้ได้ดีกับภาพใดๆ ที่ยังไม่ได้มีการปรับปรุง แต่การใช้กับภาพที่ได้ผ่านการปรับปรุงคุณภาพแล้ว จะส่งผลให้ได้ภาพผลลัพธ์ที่มีรายละเอียดของพื้นย่อยดีกว่าดังจะเห็นได้จากผลการทดลองในบทที่ 4

## 5.2 ปัญหาที่เกิดขึ้นและข้อเสนอแนะ

ในขั้นตอนการต่อภาพของวิทยานิพนธ์นี้ต้องมีการหาจุดพิกัดอ้างอิงระหว่างภาพสองภาพเอง ปริมาณค่าลำดับของจุดอ้างอิงจะเป็นข้อกำหนดสำคัญที่จะทำให้ได้ฟังก์ชันที่ให้ความสัมพันธ์ได้ดีกว่า ทั้งนี้ทั้งนั้นขึ้นอยู่กับผลลัพธ์ที่ต้องการด้วยว่ายอมรับได้เพียงใด

ในการรักษาขอบแบบปรับเรียบนั้นยังมีข้อจำกัดคือต้องมีการปรับปรุงเทมเพลตที่ใช้ในการคำนวณ ไม่สามารถปล่อยให้โปรแกรมทำงานแบบอัตโนมัติได้ เนื่องจากในแต่ละรอบการทำงานใช้เวลานานมาก หากปล่อยให้ทำงานจนเสร็จสิ้นกระบวนการจะเสียเวลาอย่างมาก จึงต้องตรวจสอบการสุ่มใช้เทมเพลตแต่ละแบบด้วย

ขั้นตอนต่างๆ จนได้มาซึ่งภาพที่ถูกแบ่งส่วนแล้วนั้น ประกอบไปด้วยโปรแกรมที่ได้พัฒนาขึ้นหลายโปรแกรม เมื่อจะใช้งานในแต่ละขั้นตอนยังต้องมีการป้อนค่าตัวแปรเข้า-ออก และเนื่องจากโปรแกรมทำงานบนดอส (DOS) ทำให้ค่อนข้างลำบากในการโต้ตอบกับผู้ใช้งาน แต่ก็มีข้อดีคือตัวโปรแกรมที่สามารถทำงานได้นั้น มีขนาดเล็กและไม่ต้องพึ่งพาระบบปฏิบัติการมาก แต่การที่มีหลายขั้นตอนก็ทำให้มีไฟล์ที่เกิดขึ้นในขั้นตอนต่างๆ มากมาย หากไม่ได้ทำการบันทึกหรือจดจำให้ดีแล้วอาจจะทำให้นำไฟล์ที่เกิดขึ้นไปใช้ผิดขั้นตอนหรือผิดไฟล์ได้

## เอกสารอ้างอิง

- [1] J.A. Richards, **Remote Sensing Digital Image Analysis ,An Introduction**. 2nd Ed. Germany :Springer-Verlag.1986
- [2] R.C. Gonzalez, R.E. Woods, **Digital Image Processing**. USA : Addison-Wesley Publishing Company. Inc.
- [3] [http://www.eoc.nasda.go.jp/guide/satellite/satdata/jers\\_e.html](http://www.eoc.nasda.go.jp/guide/satellite/satdata/jers_e.html)
- [4] [http://spaceboy.nasda.go.jp/Db/Kaihatu/Eisei/Eisei\\_E/Jers\\_i\\_e.html](http://spaceboy.nasda.go.jp/Db/Kaihatu/Eisei/Eisei_E/Jers_i_e.html)
- [5] <http://www.eorc.nasda.go.jp/EORC/Gallery/>
- [6] ปราโมทย์ เดชะอำไพ. ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม . สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย . 2538
- [7] F. Tomata and S. Tsuji, "Extraction of multiple regions smoothing in selected neighborhoods," IEEE Trans. Systems, Man and Cybernetics SMC-7, pp. 107-109, 1977.
- [8] S. Murai and M. Akiyama, "Digital mosaic of colour aerial photographs," 14 th Congress, International society for photogrammetry, Hamerg, pp. 426-429, 1980
- [9] S.C. Chapra and R.P. Canale, **Numerical Methods for Engineers**. Third Edition
- [10] สุชาติ นิยมประดิษฐ์. "การลดข้อมูลภาพโดยการเข้ารหัสพื้นที่ของภาพ." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2534.
- [11] J.B. Jun Kruskal, "On the shortest spanning subtree of a graph and the travelling salesman problem," Proc. Am. Math. Soc., Vol. 7 , pp. 48-50 , 1956.
- [12] สักกรียา ชิตวงค์. "การกำจัดรอยตะเข็บออกจากการต่อภาพสี." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2536.
- [13] O.J. Morris, M.de J. Lee and A.G. Constantinides, "Graph theory for image analysis : an approach based on the shorest spanning tree," Proc. IEE, Vol. 133, pt. F, No. 2 , pp.146-152, 1986
- [14] O.J. Morris, M.de J. Lee and A.G. Constantinides, "A unified method for segmentation and edge detection using graph theory," Proc. ICASSP, pp. 2051-2054, Tokyo, Japan, 1986.

ภาคผนวก ก  
ผลงานที่ได้รับการตีพิมพ์

# Segmentation via graph theory on the edge preserving smoothing of multiresolution mosaiced image

S. Chitwong, F. Cheevasuvit, K. Dejhan, S. Mitatha, and P. Phopongviwat  
 Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang  
 Bangkok 10520, Thailand  
 Phone: 662-3267346, Fax: 662-3267347, E-mail : kcsakrey@kmitl.ac.th

## ABSTRACT

Mosaicing of satellite image are very necessary and usefully for the cloud cover problem. For some task, such as for forest fire tracking, the complete scene image are required. But sometime the other orbits of the same satellite is too late for the use. Hence, the images with different resolutions from another satellite which acquired in contiguous time will be replaced the cloud covered regions. To merge the different solution images together, the resampling process must be applied. The obtained mosaicing image will be smoothed image in order to remove the noise pixels without blurring edges and also to unify the common regions in the connecting boundaries. To detect the boundaries of the interested region, the graph theory will be applied onto the smoothed mosaiced image for obtaining accurate region boundaries.

## 1. INTRODUCTION

Since the cloud cover problem always exists in the equator countries. Therefore, some acquired satellite images will lose a huge of land used information due to the cover of cloud. This will be caused the decreasing of efficiency is the utilization of satellite images especially for the forest fire tracking. Sometimes, it is too late to waiting the next orbits of the same satellite. Thus, the images from another satellite with different resolutions are required to inspect the region covered by cloud. For better follow up the area of interest, these image are merged together to obtain a mosaicing image. Consequently, the resampling process is applied in order to push the images in the same resolution. Before resampling the mosaiced image for detecting the boundaries of the interested region, the smoothing process is applied. Two objectives of smoothing process have been established. First is to remove the noise pixels while preserving edges. The second is to unify the connecting regions in different resolution images from mosaiced image. Finally, the graph theory will be used to detect the boundaries of the homogeneous region. The details of image mosaicing, edge preserving smoothing and segmenting will be described in the following subheadings.

## 2. MULTIREOLUTION IMAGE MOSAICING

Mosaiced image can be obtained from the same resolution images. But, the aim of this paper is to merge the different resolution images for forming a mosaiced image. However, the overlap regions must be presented in each other. The process of multiresolution image mosaicing will be consisted of the following steps.

1) Geometric registration. The corresponding of ground control points in each image of the overlap region have been selected. The geometric registration can be performed by using the method of curve fitting with a second order polynomial. Then the Bihaminic spline interpolation from MATH LAB Version 4.2 is applied in order to obtain the pixel's coordinate in integer form.

2) Brightness adjustment. Normally, the brightness over the overlap region in each image is quite different from one another. Hence, the brightness adjustment will be used for smoothing the brightness of connecting regions. The brightness adjustment function is expressed by

$$y = \left[ \frac{\sigma_R}{\sigma_a} \right] x + m_R - \left[ \frac{\sigma_R}{\sigma_a} \right] m_a \quad (1)$$

The subscript  $R$  refers to a reference image, while  $a$  referes to any image which need to change the brightness. From the equation (1),  $y$  is the new gray value a pixel which has  $x$  as old gray value of any image. Where  $m_R$  and  $\sigma_R$  are stand

for mean and standard deviation of reference image, while  $m_a$  and  $\sigma_a$  represent the mean and standard deviation of any image.

3) Seam removal. Since the mosaiced image still has a slight discontinuity of contrast over the overlap region. Therefore, a linear weighting function from [1] is applied to scale the brightness in the overlap region as shown in Fig. 1. The function can be expressed by

$$N(i, j) = \begin{cases} O_A(i, j) & \text{for } m_1 \leq i < m_2 \\ f_a(i)O_A(i, j) + f_b(i)O_B(i, j) & \text{for } m_2 \leq i \leq m_3 \\ O_B(i, j) & \text{for } m_3 \leq i \leq m_4 \end{cases} \quad (2)$$

where  $O_A(i, j)$  and  $O_B(i, j)$  are the old brightness of pixel in image A and B respectively. For  $m_1$  to  $m_4$  are the coordinates of pixel in the vertical direction.

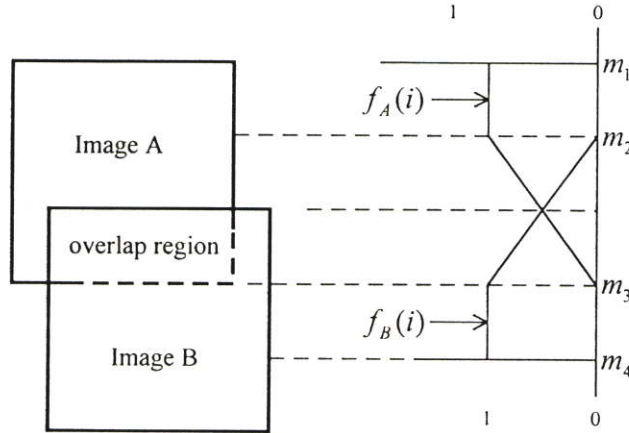


Fig. 1 Linear weight function for seam removal

From the mentioned steps, the resulting of masaiced image can be represented in Fig. 2.



(a) Image from satellite JERS-1

with resolution of  $18m \times 24m$

(b) Image from satellite MOS-1

with resolution of  $50m \times 50m$   
(selected as reference image)

(c) Masaiced image

Fig. 2 Multiresolution mosaiced image.

### 3. EDGE PRESERVING SMOOTHING

Generally, the segmented image is consisted of many small regions which may be effected by noise pixels. To eliminate these noise pixels, here we proposed to use the edge preserving smoothing process. The process tries to remove the noise

which the edges of regions still maintain. The masks for edge preserving smoothing process have been research by [2,3]. However their masks as shown in Fig. 3(a) and (b) can not preserve the thin regions with the width less than 3 pixels. Therefore we propose another masks as shown in Fig. 3(c) for maintaining the thin regions in the smoothing process. The edge preserving smoothing process can be elucidated as the following steps.

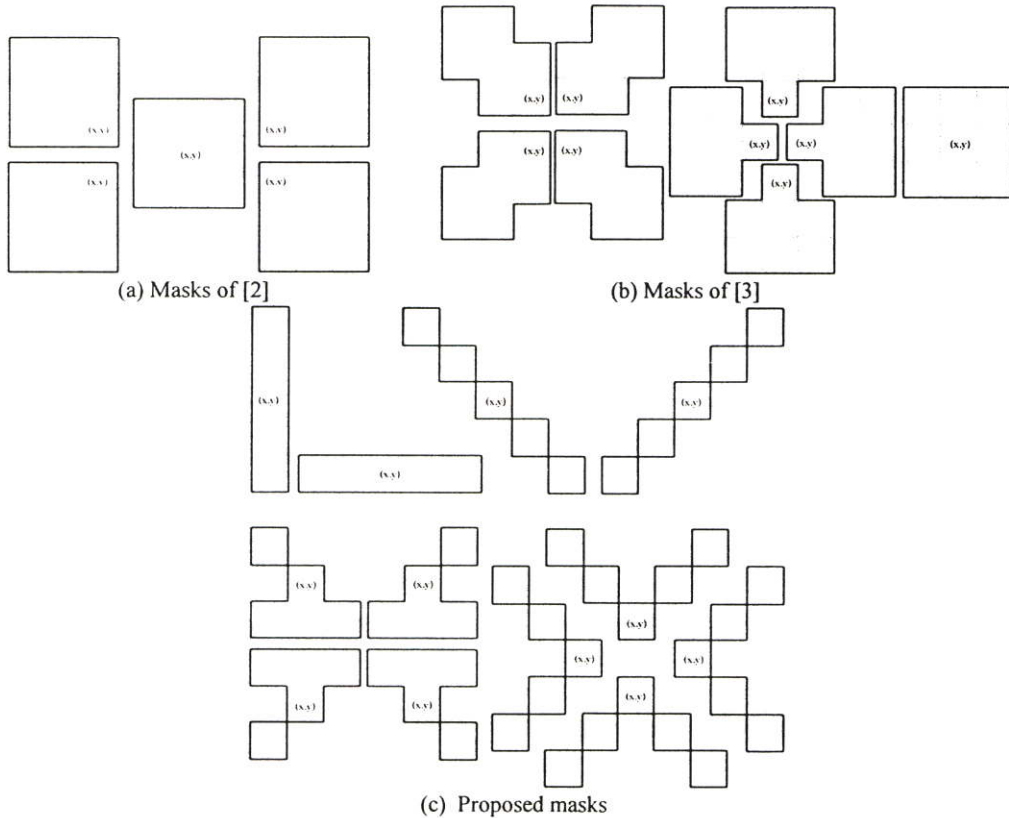


Fig. 3 the masks for edge preserving smoothing process

1) Calculate the mean value of all pixels in each mask by

$$m(k) = \frac{1}{N_k} \sum_{i,j \in S_k} f(i,j) \quad ; k = 1,2,3,\dots,M \quad (3)$$

$M$  is the number of templates,  $S_k$  is the subset of pixels in the  $k^{\text{th}}$  mask.  $N_k$  is the number of pixels in  $k^{\text{th}}$  mask and  $f(i,j)$  is the gray value of pixel  $(i,j)$ .

2) Calculate the variance of each mask with respect to consider pixel  $f(x,y)$  by

$$V(k) = \frac{1}{N_k} \sum_{i,j \in S_k} [f(i,j) - f(x,y)]^2 \quad ; k = 1,2,3,\dots,M \quad (4)$$

3) Seek the minimum variance mask.

4) Assign the mean value of the selected mask to the pixel  $(x,y)$ .

5) Repeat step 1) to 4) to all pixels of the image.

6) Iterate the process from step 1) to 5) until the gray level of all pixels is unchanged.

By cooperating the proposed masks with the mask of [2] and [3] and applied the smoothing process onto the image of Fig. 2 (c). We can obtain edge preserving smoothing image as shown in Fig. 4.

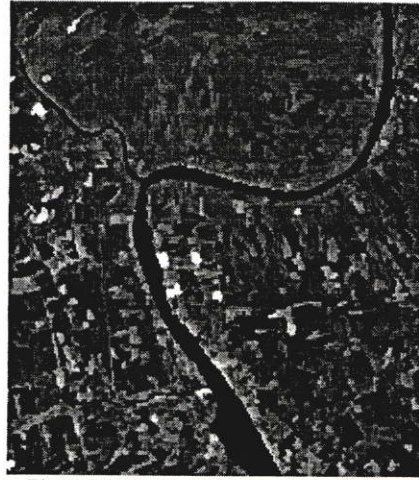


Fig. 4 Edge preserving smoothing image

#### 4. IMAGE SEGMENTATION BASED ON GRAPH THEORY

Graph theory are widely used in data compression [4], scene analysis [5] and etc. Here, we propose to use the graph theory for image segmentation, since it can provide the high accurate boundaries of segmented homogeneous regions. The algorithm of image segmentation based on graph theory is

1) Mapping an image onto a graph. Each gray value of pixel will be mapped onto the corresponding vertex of graph as shown in Fig. 5. Each vertex of the graph will be connected to the adjacent vertices in 4 perpendicular direction by the links.

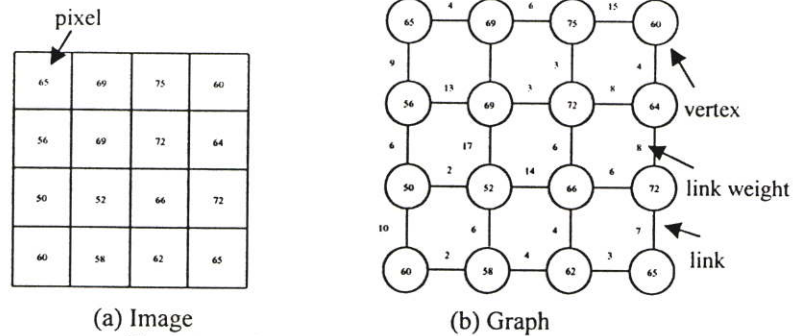


Fig. 5 Mapping an image onto a graph

2) Calculate the link weights. The weight of each link in the graph is obtained by the absolute different of vertex weight from the two connected vertices. This value will be used to measure the resemblance vertices.

3) Remove the link and merge two vertices together. The links of the graph will be removed sequentially from minimum link weight to the maximum link weight. Each time of link removal, the connected vertices will be merged together. The common weight of merged vertices will be replaced by their average vertex weights. Then the link weights around the merged vertices must be recalculated.

4) Assign the number of segmented regions. In order to obtain  $M + 1$  segmented regions, we have to assign the value  $M$  to the process for cutting the lastest  $M$  links.

#### 5. EXPERIMENTAL RESULTS

By applying the graph theory onto the edge preserving smoothing image of Fig. 4, the segmented image will be resulted as demonstrated in Fig. 6 with 1000 regions.

#### 6. CONCLUSION

This paper has been described a method for segmenting the multispectral mosaiced image which obtained by edge preserving smoothing process. The result of segmented image shows clearly that the thin regions can be maintained and the

boundary of each region quite accurate. Therefore, the problem of cloud cover can be conquered and the tracking regions can be followed up.



Fig. 6 Segmented image

## 7. REFERENCES

1. S. Murai and M. Akiyama, "Digital mosaic of colour aerial photographs," 14<sup>th</sup> Congress, International society for photogrammetry, Hamerg, pp. 426-429, 1980.
2. F. Tomita and S. Tsuji, "Extraction of multiple regions by smoothing in selected neighbourhood," IEEE Trans. System Man and Cybernetics SMC-7, pp. 107-109, 1977.
3. M. Nagao and T. Matsuyama, "Edge preserving smoothing," Computer Graphics and Image Processing, vol.9, pp. 374-407, 1979.
4. P.W. Fung, K.K. Ly, and Y. Attikiouzed, "Automatic segmentation of biomedical image," Proc ICASSP, pp. 882-885, 1988.
5. P. Narendra and M. Goldberg, "A graph-theoretic approach to image segmentation," Proc. of IEEE Computer Society conference on paterm recognition and image processing. Pp. 247-256, 1977.
6. K.S. Fu and J.K. Mui, "A survey on Image Segmentation," Pattern recognition, vol.13, no.1, pp. 3-16, 1981.
7. S.L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure," Proc. 2<sup>nd</sup> Int. Joint Conf. on Pattern Recognition, pp. 424-433, Aug. 1974.
8. F. Cheevasuvit, H. Maitre and D. Vidal-Madjar, "A robust method for picture segmentation based on a split-and-merge procedure," Computer Vision, Graphics and image processing, vol. 34, pp. 268-281, 1986.
9. J.B. Jun Kruskal, "On the shortest spanning subtree of a graph and the travelling salesman problem," Proc. Am. Math. Soc., vol. 7, pp. 48-50, 1956.
10. D. Cheriton and R.E. Tarjan, "Finding minimum spanning trees," SIAM J. Comput., vol. 5, pp. 724-742, 1967.
11. O.J. Morris, M.de J. Lee, and A.G. Constantinides, "Graph theory for image analysis : an approach based on the shortest spanning tree," Proc. IEE, vol. 133, pt. F, no. 2, pp.146-152, 1986.
12. O.J. Morris, M.de J.Lee, and A.G. Constantinides, "A unified method for segmentation and edge detection using graph theory," Proc. ICASSP, pp. 2051-2054, Tokyo, Japan, 1986.

ภาคผนวก ข  
โปรแกรมที่ใช้ในการรักษาออบแบบปรับเรียบ

```

/*-----*/
/* PROGRAM      : PNAGA.CPP      */
/* DESCRIPTION   : Edge proserving smoothing by modified Nagao's alg.  */
/* Modify by    : Phot           */
/* Date        : 1 March 2002   */
/* REFERENCE    : Segmentation on the edge preserving smoothing image */
/* COMPILER     : WC 10.0       */
/* PRESENT AT   : SPIE 2000 San Diego Ca USA      */
/*-----*/

```

```

#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <math.h>

```

```

#define uc unsigned char
#define ul unsigned long

```

```

////////////////////////////////////

```

```

int TP=3;
int LP=10;
char name[20];
char oname[20];
long xs=256;
long ys=256;

```

```

////////////////////////////////////

```

```

long imgsize;
uc *img,*out;
void merror(char *txt);

```

```

int a[[5][5]= {

```

```

{{1,1,1,0,0},//0-0 //1
  {1,1,1,0,0},
  {1,1,1,0,0},
  {0,0,0,0,0},
  {0,0,0,0,0}},
{{0,0,1,1,1},//0-1 //2
  {0,0,1,1,1},
  {0,0,1,1,1},
  {0,0,0,0,0},
  {0,0,0,0,0}},
  {{0,0,0,0,0},//0-3 //3
    {0,0,0,0,0},
    {1,1,1,0,0},
    {1,1,1,0,0},
    {1,1,1,0,0}},
  {{0,0,0,0,0},//0-2 //4
    {0,0,0,0,0},
    {0,0,1,1,1},
    {0,0,1,1,1},
    {0,0,1,1,1}},
  {{0,0,0,0,0},//0-4 //5
    {0,1,1,1,0},
    {0,1,1,1,0},
    {0,1,1,1,0},
    {0,0,0,0,0}},

  {{1,1,0,0,0},//1-0 //6
    {1,1,1,0,0},
    {0,1,1,0,0},
    {0,0,0,0,0},
    {0,0,0,0,0}},
  {{0,0,0,1,1},//1-1 //7
    {0,0,1,1,1},
    {0,0,1,1,0},
    {0,0,0,0,0},
    {0,0,0,0,0}},

```

```

{{0,0,0,0,0},//1-3 //8

```

```

    {0,0,0,0,0},

```

```

    {0,1,1,0,0},

```

```

    {1,1,1,0,0},

```

```

    {1,1,0,0,0}},

```

```

{{0,0,0,0,0},//1-2 //9

```

```

    {0,0,0,0,0},

```

```

    {0,0,1,1,0},

```

```

    {0,0,1,1,1},

```

```

    {0,0,0,1,1}},

```

```

{{0,1,1,1,0},//2-0 //10

```

```

    {0,1,1,1,0},

```

```

    {0,0,1,0,0},

```

```

    {0,0,0,0,0},

```

```

    {0,0,0,0,0}},

```

```

{{0,0,0,0,0},//2-3 //11

```

```

    {1,1,0,0,0},

```

```

    {1,1,1,0,0},

```

```

    {1,1,0,0,0},

```

```

    {0,0,0,0,0}},

```

```

{{0,0,0,0,0},//2-1 //12

```

```

    {0,0,0,1,1},

```

```

    {0,0,1,1,1},

```

```

    {0,0,0,1,1},

```

```

    {0,0,0,0,0}},

```

```

{{0,0,0,0,0},//2-2 //13

```

```

    {0,0,0,0,0},

```

```

    {0,0,1,0,0},

```

```

    {0,1,1,1,0},

```

```

    {0,1,1,1,0}},

```

```

//modify tp

```

```

{{1,0,0,0,0},//5-0 //14

```

```

    {0,1,0,0,0},

```

```

    {1,1,1,0,0},

```

$\{0,0,0,0,0\},$   
 $\{0,0,0,0,0\},$   
 $\{\{0,0,0,0,1\},//5-1 \quad //15$   
 $\{0,0,0,1,0\},$   
 $\{0,0,1,1,1\},$   
 $\{0,0,0,0,0\},$   
 $\{0,0,0,0,0\},$   
 $\{\{0,0,0,0,0\},//5-3 \quad //16$   
 $\{0,0,0,0,0\},$   
 $\{1,1,1,0,0\},$   
 $\{0,1,0,0,0\},$   
 $\{1,0,0,0,0\},$   
 $\{\{0,0,0,0,0\},//5-2 \quad //17$   
 $\{0,0,0,0,0\},$   
 $\{0,0,1,1,1\},$   
 $\{0,0,0,1,0\},$   
 $\{0,0,0,0,1\},$   
  
 $\{\{1,0,0,0,1\},//6-1 \quad //18$   
 $\{0,1,0,1,0\},$   
 $\{0,0,1,0,0\},$   
 $\{0,0,0,0,0\},$   
 $\{0,0,0,0,0\},$   
 $\{\{1,0,0,0,0\},//6-0 \quad //19$   
 $\{0,1,0,0,0\},$   
 $\{0,0,1,0,0\},$   
 $\{0,1,0,0,0\},$   
 $\{1,0,0,0,0\},$   
  
 $\{\{0,0,0,0,1\},//6-2 \quad //20$   
 $\{0,0,0,1,0\},$   
 $\{0,0,1,0,0\},$   
 $\{0,0,0,1,0\},$   
 $\{0,0,0,0,1\},$   
 $\{\{0,0,0,0,0\},//6-3 \quad //21$   
 $\{0,0,0,0,0\},$

$\{0,0,1,0,0\},$   
 $\{0,1,0,1,0\},$   
 $\{1,0,0,0,1\}\},$

$\{\{0,0,1,0,0\},//6-1 \quad //22$

$\{0,0,1,0,0\},$   
 $\{1,1,1,0,0\},$   
 $\{0,0,0,0,0\},$   
 $\{0,0,0,0,0\}\},$

$\{\{0,0,1,0,0\},//6-0 \quad //23$

$\{0,0,1,0,0\},$   
 $\{0,0,1,1,1\},$   
 $\{0,0,0,0,0\},$   
 $\{0,0,0,0,0\}\},$

$\{\{0,0,0,0,0\},//6-2 \quad //24$

$\{0,0,0,0,0\},$   
 $\{1,1,1,0,0\},$   
 $\{0,0,1,0,0\},$   
 $\{0,0,1,0,0\}\},$

$\{\{0,0,0,0,0\},//6-3 \quad //25$

$\{0,0,0,0,0\},$   
 $\{0,0,1,1,1\},$   
 $\{0,0,1,0,0\},$   
 $\{0,0,1,0,0\}\},$

$\{\{0,1,1,1,0\},//6-1 \quad //26$

$\{0,0,1,0,0\},$   
 $\{0,0,1,0,0\},$   
 $\{0,0,0,0,0\},$   
 $\{0,0,0,0,0\}\},$

$\{\{0,0,0,0,0\},//6-0 \quad //27$

$\{1,0,0,0,0\},$   
 $\{1,1,1,0,0\},$   
 $\{1,0,0,0,0\},$   
 $\{0,0,0,0,0\}\},$

$\{\{0,0,0,0,0\},//6-2 \quad //28$

```

        {0,0,0,0,0},
        {0,0,1,0,0},
        {0,0,1,0,0},
        {0,1,1,1,0}},
    {{0,0,0,0,0},//6-3 //29
        {0,0,0,0,1},
        {0,0,1,1,1},
        {0,0,0,0,1},
        {0,0,0,0,0}},
    {{0,0,1,0,0},//3-0 //30
        {0,0,1,0,0},
        {0,0,1,0,0},
        {0,0,1,0,0},
        {0,0,1,0,0}},
    {{0,0,0,0,0},//3-1 //31
        {0,0,0,0,0},
        {1,1,1,1,1},
        {0,0,0,0,0},
        {0,0,0,0,0}},
    {{1,0,0,0,0},//4-0 //32
        {0,1,0,0,0},
        {0,0,1,0,0},
        {0,0,0,1,0},
        {0,0,0,0,1}},
    {{0,0,0,0,1},//4-1 //33
        {0,0,0,1,0},
        {0,0,1,0,0},
        {0,1,0,0,0},
        {1,0,0,0,0}},
};

```

```

void merror(char *txt)
{
    printf("\n%s\n",txt);
    exit(1);
}

```

```

void main(int argc,char *argv[])
{

int x,y,i,j,k;
FILE *fh;
float vari,mean;
int tpcount[33],tpnum,meani;
float sumx,sumxx;
int loop;

if(argc<7)
    merror("pnaga iname.img oname.img xsize ysize NUMTP LOOP");

sprintf(name,"%s",argv[1]);
sprintf(oname,"%s",argv[2]);
xs=atol(argv[3]);
ys=atol(argv[4]);
TP=atoi(argv[5]);
LP=atoi(argv[6]);

fprintf(stdout,"\ninpname=%s",name);
fprintf(stdout,"\noutname=%s",oname);
fprintf(stdout,"\nxsize=%ld",xs);
fprintf(stdout,"\nysize=%ld",ys);
fprintf(stdout,"\nNUM TEMPLATE=%d",TP);
fprintf(stdout,"\nNUM LOOP  =%d\n",LP);
// getch();

//ALLOCATE
imgsize=xs*ys;
if((img = (uc*)calloc((size_t)imgsize,sizeof(char))) == NULL)
    merror("alloc [uc *img] error");
if((out = (uc*)calloc((size_t)imgsize,sizeof(char))) == NULL)
    merror("alloc [uc *out] error");

```

```

//READ INPUT
if((fh = fopen(name,"rb")) == NULL)
    merror("open file error");

for(y=0;y<ys;y++)
{
    fread(img+(size_t)(y*xs),sizeof(char),(size_t)xs,fh);
}
fclose(fh);

////////////////////////////////////
//PROCESS
for(k=0;k<33;k++)
    tpcount[k]=0;

for(loop=0;loop<LP;loop++)
{
    for(y=2;y<ys-2;y++)
        for(x=2;x<xs-2;x++)
        {
            float minvari=9999999.0;
            tpnum=0;
            meani=0;
            //*****
            // Calculate mean and variance
            //*****
            for(k=0;k<TP;k++)
            {
                int temp,cmask;
                sumx=0.0;
                sumxx=0.0;
                cmask=0;
                // calculate mean value
                for(j=0;j<5;j++)
                    for(i=0;i<5;i++)
                        {

```

```

// int temp;
if(a[k][j][i]==1)
{
    temp=a[k][j][i]*(*(img+(y+j-2)*(size_t)xs+x+i-2));
    sumx=sumx+(float)temp;
    cmask=cmask+1;
// sumxx=sumxx+temp*(float)temp;
}
}
mean=sumx/(float)cmask;
// calculate variance value
cmask=0;
for(j=0;j<5;j++)
for(i=0;i<5;i++)
{
    int temp;
    if(a[k][j][i]==1)
    {
        temp=*(img+(y+j-2)*(size_t)xs+x+i-2)-(int)mean;
        // sumxx+=pow(temp,2.0);
        sumxx=sumxx+(temp*temp);
        cmask=cmask+1;
    }
}

sumx=sumxx/(float)cmask;
vari=sqrt(sumx);
if(vari<minvari)
{
    minvari=vari;
    tpnum=k;
    meani=(int)mean;
    if(meani>255) meani=255;
    else if(meani<0) meani=0;
}
}

```

```

//*****
tpcount[tpnum]++;
*(out+y*(size_t)xs+x)=meani;
}
for(y=2;y<ys-2;y++)
for(x=2;x<xs-2;x++)
{
int temp;
temp=*(out+y*(size_t)xs+x);
*(img+y*(size_t)xs+x)=temp;
}
// if(kbhit()) goto pass;
}
// pass:

////////////////////////////////////
//WRITE RESULT

if((fh = fopen(oname,"wb")) == NULL)
merror("open file error");

for(y=0;y<ys;y++)
{
fwrite(out+(size_t)(y*xs),sizeof(char),(size_t)xs,fh);
}
fclose(fh);

//fprintf("\nloop=%d\n",loop);

for(k=0;k<TP;k++)
fprintf(stdout,"tpcount[%d]=%d",k,tpcount[k]);

free(img);
free(out);
}

```

## ประวัติผู้เขียน

นายพรพจน์ โปธิ์พงษ์วิวัฒน์ เกิดเมื่อวันที่ 30 สิงหาคม 2515 ที่อำเภอท่าม่วง จังหวัดกาญจนบุรี จบการศึกษาระดับปริญญาตรีในหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมระบบวัดคุม จากมหาวิทยาลัยเทคโนโลยีมหานคร ในปีการศึกษา 2539 และเข้าศึกษาต่อที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปี พ.ศ. 2540 เข้าทำงานเป็นอาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร ถึงปัจจุบัน