การแปลงข้อมูลภาพจากรูปแบบราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ สำหรับ GIS

AUTOMATED IMAGE CONVERSION FROM RASTER TO VECTOR FORMAT FOR GIS

T031022

กิ่งกาญจน์ วงศ์วิภาพร

KINGKARN WONGWIPAPORN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2541

ISBN 974-622-173-6

วพ
ก.ค.๑๑
๒๕๔๑
เลขหมู่.....
เลขทะเบียน 31022
วัน, เดือน, ปี 8 ก.ย. 2541

b 10724874
1 1103589

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATED IMAGE CONVERSION FROM RASTER TO VECTOR FORMAT FOR GIS

KINGKARN WONGWIPAPORN

A THESIS SUMMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE PROGRAM IN COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
1998

ISBN 974-622-173-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 1998

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การแปลงข้อมูลภาพจากรูปแบบราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติสำหรับ GIS
นักกศึกษา	นางสาวกิ่งกาญจน์ วงศ์วิภาพร
อาจารย์ผู้ควมคุมวิทยานิพนธ์	ผศ.ดร.บุญวัฒน์ อัตชู
หลักสูตร	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์และเทคโลยีสารสนเทศ
พ.ศ.	2541

บทคัดย่อ

ระบบสารสนเทศทางภูมิศาสตร์ GIS (Geographic Information System) มีการเก็บข้อมูลเพื่อใช้ในการสอบถามและวิเคราะห์ เป็นสองลักษณะ คือ โครงสร้างแบบราสเตอร์ (Raster) และเวคเตอร์ (Vector) โดยข้อมูลเชิงพื้นที่ (Spatial Data) ที่ได้จากภาพถ่ายทางอากาศนั้นมักมีโครงสร้างเป็นแบบราสเตอร์ ที่ไม่ได้เก็บความสัมพันธ์ของส่วนประกอบของภาพในเชิงระยะทางได้อย่างชัดเจน จึงมักจะมีการใช้คนทำการเก็บข้อมูลแบบเวคเตอร์ ซึ่งต้องใช้เวลาและความละเอียดอย่างมาก ดังนั้นวิทยานิพนธ์นี้จึงนำเสนอถึงวิธีการแปลงข้อมูลภาพจากรูปแบบราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติสำหรับ GIS โดยใช้หลักการและทฤษฎีเกี่ยวกับการประมวลผลข้อมูลภาพมาช่วยในการแปลงข้อมูลภาพแบบราสเตอร์ที่มีลักษณะการเก็บเป็นจุดภาพ . (Pixel) เป็นเวคเตอร์ที่มีลักษณะเป็นจุด (Point) เส้น (Line) รูปเหลี่ยม (Polygon) แบบอัตโนมัติ แทนการใช้คนทำการเก็บข้อมูลเวคเตอร์ผ่านเครื่องดิจิไตท์เซอร์ โดยการนำภาพสีหรือขาวดำที่ต้องการแปลงมาผ่านเครื่องสเกนเนอร์ ทำให้ได้ข้อมูลภาพแบบราสเตอร์ และทำการประมวลผลข้อมูลภาพ ตามขั้นตอนที่ได้ออกแบบตามลำดับดังนี้ การหาขอบภาพ (Edge Detection) โดยใช้เทคนิค Range Edge [2] การทำขอบภาพให้บาง (Edge Thinning) โดยใช้เทคนิค Simple Thinning [2] การหาจุดรวมของเส้น (Node) โดยใช้ตัวกรอง การหาส่วนของเส้นตรง (Line Tracking or Line Segment) โดยใช้เทคนิค Chain Link Code และเทคนิคตัวกรองหาทิศทางที่ออกแบบขึ้น การหารูปเหลี่ยมพื้นที่ (Polygon) โดยใช้เทคนิค Labeling ซึ่งทำให้เกิดการแปลงข้อมูลเป็นเวคเตอร์แบบอัตโนมัติ (Automatic Digitizer) ที่สามารถบ่งบอกถึงตำแหน่งของส่วนประกอบของภาพได้อย่างสะดวก รวดเร็วและลดเวลาในการเก็บข้อมูลอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Automated Image Conversion from Raster to Vector Format for GIS
Student	Miss Kingkarn Wongwipaporn
Thesis Advisor	Assitant Prof. Dr. Boonwat Attachoo
Degree	Master of Science program in Computer Science and Information Technology
Year	1998

ABSTRACT

Geographic Information System (GIS) data for requirement of analysis usually collect in two structure; raster and vector. The spatial datal of geographic image is raster format but it does not specify the relation ship of spatial data. So the conversion from raster to vector format for GIS usually use digitizer by manual. This paper presents image processing theory to automatic converting raster image from pixel format to vector format of point , line or polygon. This approach uses scanner to transform color or black and white image to raster image format and process step by step for conversion. Range edge technique and simple thinning are used for edge detection and edge thinning. The filtering method is presented for finding node position of line. Line segment or Line tracking uses Chain Link Code technique and filtering of dimension. Finally, Labeling technique is used for polygon process finding. The experimental result of these processes must show automatic digitizer to vector format that flexible for collecting data and specitfic the position of feature image.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความช่วยเหลือจากบุคคลหลายฝ่าย ผู้วิจัยขอกราบขอบพระคุณไว้ ณ ที่นี้ โดยเฉพาะอย่างยิ่ง ผู้ช่วยศาสตราจารย์ ดร.บุญวัฒน์ อัตชู ที่ได้กรุณาให้ความอนุเคราะห์ในการใช้อุปกรณ์ระบบไมโครคอมพิวเตอร์ และให้คำแนะนำรวมทั้งตรวจสอบความถูกต้องตลอดการดำเนินการวิจัย

กิ่งกาญจน์ วงศ์วิภาพร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... I

บทคัดย่อภาษาอังกฤษ..... II

กิตติกรรมประกาศ..... III

สารบัญ..... IV

สารบัญตาราง..... VII

สารบัญภาพ.....VIII

บทที่ 1. บทนำ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของการวิจัย.....1

1.3 แนวคิดที่ใช้ในการวิจัย.....2

1.4 ขอบเขตของการวิจัย..... 2

1.5 ขั้นตอนการศึกษาและพัฒนา..... 3

1.6 ประโยชน์ที่คาดว่าจะได้รับ.....3

บทที่ 2. ความรู้พื้นฐานเกี่ยวกับระบบสารสนเทศทางภูมิศาสตร์.....4

2.1 ระบบข้อมูล (Information System).....4

2.2 ระบบสารสนเทศทางภูมิศาสตร์.....6

2.3 ขบวนการในการวิเคราะห์ข้อมูลในระบบ GIS.....7

2.4 องค์ประกอบของระบบสารสนเทศทางภูมิศาสตร์.....9

2.4.1 องค์ประกอบของคอมพิวเตอร์ฮาร์ดแวร์.....9

2.4.2 องค์ประกอบของคอมพิวเตอร์ซอฟแวร์.....11

2.4.3 องค์กรในการดำเนินงาน.....11

2.5 ลักษณะของข้อมูลในระบบสารสนเทศทางภูมิศาสตร์.....12

2.5.1 ลักษณะข้อมูลเชิงเฉพาะ.....12

2.5.2 ลักษณะข้อมูลเชิงพื้นที่.....14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3. การบันทึกและจัดเก็บข้อมูลในระบบสารสนเทศทางภูมิศาสตร์.....18
3.1 การจัดเก็บข้อมูลในระบบสารสนเทศทางภูมิศาสตร์.....18
3.2 ลักษณะโครงสร้างและการป้อนข้อมูลในระบบสารสนเทศภูมิศาสตร์.....26
3.2.1 ลักษณะโครงสร้างแบบเวคเตอร์.....26
3.2.2 ลักษณะโครงสร้างแบบราสเตอร์.....31
3.3 การตรวจสอบและแก้ไขข้อมูล.....33
3.4 ข้อดีและข้อเสียของการเก็บข้อมูลลักษณะโครงสร้างราสเตอร์และเวคเตอร์....34
บทที่ 4. การออกแบบขั้นตอนและวิธีการในการแปลงข้อมูลราสเตอร์เป็นเวคเตอร์.....37
4.1 ขั้นตอนในการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์.....37
4.2 การออกแบบขั้นตอนและพัฒนาการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ.....39
4.2.1 ขั้นตอนที่ 1 Scaning การเก็บข้อมูลภาพผ่านเครื่องสแกนเนอร์.....39
4.2.2 ขั้นตอนที่ 2 Skeletonizing การหาโครงร่างที่บางของภาพ.....43
4.2.3 ขั้นตอนที่ 3 Node Improvement การหาจุดรวมของเส้น.....45
4.2.4 ขั้นตอนที่ 4 Line Tracking การหาส่วนของเส้น.....47
4.2.5 ขั้นตอนที่ 5 Segment Merge การหารูปเหลี่ยมพื้นที่.....53
บทที่ 5. ผลลัพธ์จากการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์.....54
บทที่ 6. สรุปวิจารณ์และเสนอแนะ.....65

บรรณานุกรม.....67
ภาคผนวก.....68
ภาคผนวก ก. ภาพแผนที่ประเทศไทยซึ่งเป็นภาพต้นแบบที่ใช้ในการแปลงข้อมูลจากแบบราสเตอร์เป็นเวคเตอร์.....69
ภาคผนวก ข. ภาพแผนที่ภาคเหนือของประเทศไทยที่ได้หลังจากทำการตัดตกแต่งสีแล้ว..71
ภาคผนวก ค. ภาพแผนที่ภาคตะวันออกเฉียงเหนือของประเทศไทยที่ได้หลังจากทำการตัดตกแต่งสีแล้ว.....73
ภาคผนวก ง. ตัวอย่างผลลัพธ์ข้อมูลเวคเตอร์.....75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

ภาคผนวก จ. โปรแกรมต้นฉบับ.....82
ภาคผนวก ฉ. ผลงานที่วิจัยที่ได้รับกวรตีพิมพ์..... 127
ประวัติผู้เขียน.....144

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 ลักษณะของเกณฑ์การวัดในระดับต่าง ๆ 13

3.1 ตารางการเปรียบเทียบคุณสมบัติของโครงสร้างข้อมูลแบบเวคเตอร์และราสเตอร์..... 35

4.1 รายละเอียดข้อมูล Header File 128 Byte.....41

5.1 จำนวนผลลัพธ์ของข้อมูลรูปแบบเวคเตอร์ที่ได้จากแปลงแบบอัตโนมัติ.....58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 ลักษณะข้อมูลของระบบข้อมูล (Information System Taxonomy).....5
2.2 ลักษณะของระบบสารสนเทศภูมิศาสตร์.....6
2.3 การวิเคราะห์ข้อมูลในรูปแบบของ GIS.....8
2.4 ส่วนประกอบของคอมพิวเตอร์ฮาร์แวร์.....10
2.5 รูปแบบของข้อมูลเชิงพื้นที่.....15
2.6 รูปแบบความสัมพันธ์ระหว่างลักษณะข้อมูลเชิงเฉพาะและลักษณะข้อมูลเชิงพื้นที่.....16
3.1 การเก็บข้อมูลแบบ Maual Capture ในระบบเวคเตอร์.....21
3.2 การเก็บข้อมูลแบบ Maual Capture ในระบบราสเตอร์.....23
3.3 ลักษณะของเครื่องมือดิจิไทเซอร์.....24
3.4 ลักษณะของเครื่องมือสแกนเนอร์.....25
3.5 ขบวนการ Generialization ในระดับต่าง ๆ 26
3.6 ลักษณะการป้อนข้อมูลรูปแบบจุด.....27
3.7 ลักษณะการป้อนข้อมูลรูปแบบเส้นที่ใช้ Chain และ Node.....28
3.8 ข้อผิดพลาดในการป้อนข้อมูล.....29
3.9 รูปแบบโครงสร้างข้อมูลมาตราฐานแบบเวคเตอร์.....30
3.10 การป้อนข้อมูลแบบราสเตอร์.....32
4.1 แสดงความสัมพันธ์ของขั้นตอนในการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์.....38
4.2 ตัวอย่างภาพและข้อมูลราสเตอร์ที่ได้จากการนำภาพผ่านเครื่องสแกนเนอร์ ซึ่งมีลักษณะเป็นภาพสี. 40
4.3 การหาขอบภาพโดยการหาความสัมพันธ์รอบจุด.....45
4.4 แสดงความสัมพันธ์ของจุดในตาราง (3x3) ที่ใช้หา (Node).....46
4.5 แสดงความสัมพันธ์ของจุดที่ทำให้เกิดเส้นและทิศทางของเส้น.....48
4.6 ตัวอย่างการหาความสัมพันธ์ของจุดที่ทำให้เกิดเส้นและแนวโน้มของเส้น.....49
4.7 การหาเส้นโดยเริ่มเส้นใหม่เมื่อแนวโน้มทิศทางของเส้นเปลี่ยน.....50
4.8 การหาเส้นโดยเริ่มเส้นใหม่เมื่อแนวโน้มทิศทางของเส้นเปลี่ยน และทิศทางเปลี่ยน 4 ครั้ง....51
4.9 การหาเส้นโดยเริ่มเส้นใหม่เมื่อแนวโน้มทิศทางของเส้นเปลี่ยน และทิศทางเปลี่ยน1ครั้ง.....52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

4.10 แสดงการ Label หาพื้นที่ส่วนของรูปเหลี่ยมและหาเส้นที่เป็นส่วนประกอบของรูปเหลี่ยม..53
5.1 การหาขอบภาพกับระดับความเข้มของแม่สีแต่ละสี และระดับความเข้มขาวดำ โดยวิธี RangeEdge ที่กำหนดค่า Threshold>1 และขนาดตารางที่สนใจเท่ากับ 2x2.....57
5.2 การหาขอบภาพแผนที่ภาคเหนือของประเทศไทย ที่ระดับความเข้มของสีขาวดำ.....58
5.3 การหาขอบภาพแผนที่ภาคตะวันออกเฉียงเหนือของประเทศไทย ที่ระดับความเข้มของสีขาวดำ.....59
5.4 การหา Node และส่วนประกอบของเส้นกับภาพแผนที่ภาคเหนือของประเทศไทย ที่มีระดับความเข้มของสีขาวดำ.....60
5.5 การหา Node และส่วนประกอบของเส้นกับภาพแผนที่ภาคตะวันออกเฉียงเหนือของประเทศไทย ที่มีระดับความเข้มของสีขาวดำ.....61
5.6 ผลลัพธ์ภาพที่ได้จากการทำ Labeling.....62
5.7 ผลลัพธ์ของภาพที่ได้จากการหารูปเหลี่ยมพื้นที่ทางภาคเหนือของประเทศไทย.....62
5.8 ผลลัพธ์ของภาพที่ได้จากการหารูปเหลี่ยมพื้นที่ทางภาคตะวันออกเฉียงเหนือของประเทศไทย.....63
5.9 แสดงตัวอย่างผลลัพธ์ในความสัมพันธ์ของข้อมูลเวคเตอร์ที่ได้จากการแปลงข้อมูลราสเตอร์เป็นเวคเตอร์.....64
6.1 แสดงส่วนของเส้นตรงที่เป็นส่วนประกอบของรูปเหลี่ยม เกินพื้นที่รูปเหลี่ยม.....68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในระบบสารสนเทศทางภูมิศาสตร์ Geographic Information System (GIS) นั้นมักมีการวิเคราะห์ข้อมูลเชิงพื้นที่ (Spatial Data) โดยข้อมูลที่สามารถจัดเก็บผ่านอุปกรณ์คอมพิวเตอร์มาทำการวิเคราะห์จะแบ่งออกเป็น 2 ประเภท คือ ข้อมูลแบบราสเตอร์ (Raster) กับแบบเวคเตอร์ (Vector) โดยรูปแบบข้อมูลที่เป็นแบบราสเตอร์จะมีการจัดเก็บเป็นแบบกริด (Grid) หรือ จุดภาพ (Pixel) ซึ่งไม่ได้เก็บความสัมพันธ์ของระยะทางอย่างชัดเจน เช่นข้อมูลภาพที่ได้จากภาพถ่ายทางอากาศมักจะมีรูปแบบเป็นแบบราสเตอร์ ดังนั้นจึงมักจะต้องมีการทำ Manual Digitize โดยผ่านเครื่อง Digitizer ซึ่งต้องทำการไต่ไปตามเส้น (Tracing) โดยใช้อุปกรณ์ Digitizing table ซึ่งเสียเวลาอีกทั้งยังมีความผิดพลาดได้ และความละเอียดของข้อมูลเวคเตอร์ที่ได้จะมีมากน้อยเพียงใด ขึ้นอยู่กับคนที่ทำการ Tracing ให้เป็นรูปแบบเวคเตอร์เพื่อให้ได้ความสัมพันธ์ของข้อมูลเชิงพื้นที่ เพื่อให้ได้ข้อมูลที่ถูกต้อง ชัดเจน รวดเร็วและใกล้เคียงระยะทางในความเป็นจริงในการวิเคราะห์มากที่สุด

ดังนั้นงานวิทยานิพนธ์นี้จึงได้ศึกษาถึงการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ โดยการเก็บข้อมูลภาพภูมิศาสตร์ผ่านเครื่องสแกนเนอร์ ที่มีลักษณะเป็นแบบราสเตอร์ ซึ่งประกอบด้วยจุดภาพ และในหลาย ๆ จุด รวมกันเป็นความละเอียดของภาพ จากนั้นจะทำการประมวลผลข้อมูลภาพดังกล่าว โดยใช้ทฤษฎีการประมวลผลข้อมูลภาพตามขั้นตอนการแปลงข้อมูลแบบราสเตอร์เป็นเวคเตอร์ที่ได้ศึกษาและออกแบบ จากนั้นก็จะทำการแยกแยะผลลัพธ์ของข้อมูลที่ได้มาเก็บในรูปแบบเวคเตอร์ ซึ่งประกอบด้วย จุด (Point) ส่วนของเส้นตรง (Line Segment) จุดรวมของเส้น (Node) รูปเหลี่ยม (Polygon)

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 ศึกษาถึงขั้นตอนในการบันทึกและเก็บข้อมูลสำหรับ GIS

1.2.2 ศึกษารูปแบบโครงสร้างมาตราฐานแบบเวคเตอร์สำหรับ GIS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

1.2.3 ศึกษาถึงขั้นตอนในการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ

1.2.4 นำทฤษฎีการประมวลผลภาพมาประยุกต์ใช้ในการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์

1.2.5 เปรียบเทียบข้อดีข้อเสียของการเก็บข้อมูลในโครงสร้างแบบราสเตอร์และเวคเตอร์

1.3 แนวคิดที่ใช้ในการวิจัย

เนื่องจากโครงสร้างข้อมูลภาพแบบเวคเตอร์จะสามารถบอกความสัมพันธ์ระหว่างกันและกันได้ เช่น จุดแต่ละจุดจะมีความสัมพันธ์ประกอบกันเป็นเส้น และโครงร่างของเส้นแต่ละเส้นก็สามารถประกอบกันเป็นรูปเหลี่ยม ซึ่งสามารถแสดงความสัมพันธ์ในลักษณะตำแหน่งของข้อมูล แต่ในการเก็บข้อมูลภาพแบบเวคเตอร์นั้นจะต้องใช้เวลามาก และอุปกรณ์คอมพิวเตอร์ที่ใช้ในการเก็บข้อมูลมักจะให้ผลลัพธ์เป็นรูปแบบราสเตอร์ ดังนั้นหากเราสามารถใช้หลักการในการประมวลผลข้อมูลภาพในรูปแบบราสเตอร์ เช่น การกำจัดข้อมูลรบกวนภาพ การหาขอบภาพ การทำขอบภาพให้บาง การหาจุดรวมของเส้น การส่วนประกอบของเส้น การหารูปเหลี่ยมพื้นที่ มาทำการแปลงข้อมูลภาพราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติได้ก็จะทำให้เกิดความสะดวก รวดเร็วในการเก็บยิ่งขึ้น

1.4 ขอบเขตของการวิจัย

จะทำการทดลองเก็บข้อมูลภาพแผนที่ประเทศไทยในรูปแบบราสเตอร์ที่ลักษณะเป็นภาพสีและขาวดำ เพื่อนำมาใช้ในการแปลงให้เป็นรูปแบบเวคเตอร์ โดยในการทดลองแปลงจากราสเตอร์เป็นเวคเตอร์นั้น จะจำกัดในขอบเขตของการแปลง เพื่อให้ได้ข้อมูลที่มีลักษณะเป็นเชิงพื้นที่ (Spatial Data) และเป็นข้อมูลที่มีความสัมพันธ์แบบไม่ต่อเนื่องเท่านั้น ซึ่งจะได้ผลลัพธ์ของข้อมูลรูปแบบเวคเตอร์ที่ประกอบด้วย จุด เส้น จุดรวมของเส้น ส่วนของเส้น และรูปเหลี่ยมพื้นที่ โดยจากผลการทดลองทำให้สามารถทราบว่า พื้นที่ของจังหวัดต่าง ๆ ในภาพเหนือและภาคตะวันออกเฉียงเหนือของประเทศไทยประกอบไปด้วยจุดกี่จุด เส้นกี่เส้น มีจุดรวมของเส้นใดบ้าง และรูปเหลี่ยมพื้นที่ใดอยู่ติดกับพื้นที่ใด ซึ่งสามารถทำการแปลงและเก็บข้อมูลได้รวดเร็วยิ่งขึ้น

1.5 ขั้นตอนการศึกษาและพัฒนา

1.5.1 ศึกษาทฤษฎีและหลักการต่าง ๆ ที่เกี่ยวข้อง ซึ่งเป็นขั้นตอนในการบันทึกและเก็บข้อมูล ที่มีรูปแบบโครงสร้างมาตรฐานแบบเวคเตอร์สำหรับ GIS และขั้นตอนในการแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลภาพ จากราสเตอร์เป็นแบบเวคเตอร์แบบอัตโนมัติ

1.5.2 นำภาพต้นแบบ ซึ่งเป็นภาพแสดงการแบ่งเขตจังหวัดในประเทศไทย ซึ่งมีขนาดของภาพเท่ากับขนาดกระดาษ A4 และลักษณะของภาพเป็นภาพสีและมีตัวอักษรข้อความแสดงชื่อจังหวัดในประเทศไทย มาทำการสแกนโดยผ่านเครื่องสแกนเนอร์

1.5.3 นำ Software เกี่ยวกับการตัดแต่งภาพ เช่น Paint-Brush มาทำการตัดส่วนของภาพที่เป็นส่วนของภาคเหนือและภาคตะวันออกเฉียงเหนือของประเทศไทย พร้อมให้สีของภาพใหม่โดยให้โทนสีที่แตกต่างกันอย่างชัดเจนในแต่ละส่วนของจังหวัด และนำภาพที่ได้ทำการตกแต่งให้ชัดเจนแล้ว ซึ่งมีการเก็บเป็นแบบ .PCX ไฟล์ มาทำการแปลงจากราสเตอร์เป็นเวคเตอร์

1.5.4 ทำการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์โดยการใช้ทฤษฎีในการประมวลผลข้อมูลภาพ การหาขอบภาพ (Edge Detection) [2] การทำขอบภาพให้บาง (Thinning Edge) [2] การหาจุดรวมของเส้น (Node Detection) การหาส่วนของเส้น (Line Tracking) การหาส่วนประกอบของรูปเหลี่ยม (Segment Merge) ซึ่งทำให้เกิดการแปลงเป็นรูปแบบโครงสร้างมาตราฐานเวคเตอร์แบบอัตโนมัติ [4]

1.5.5 ทำการพัฒนาโปรแกรมในการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์แบบอัตโนมัตินั้น จะพัฒนาโดยใช้ภาษา C และ FoxPro ซึ่งในการเก็บข้อมูลผลลัพธ์รูปแบบราสเตอร์เป็นแบบ Text File ส่วนรูปแบบเวคเตอร์มีลักษณะเป็น .DBF ไฟล์ และมีการจัดการแบบ Index ในการอ้างถึงความสัมพันธ์ของข้อมูลในโครงสร้างแบบเวคเตอร์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ทำให้สามารถแปลงข้อมูลภาพที่มีรูปแบบการแบ่งแยกโทนสี ในการแสดงขอบเขตพื้นที่อย่างชัดเจน ที่อยู่ในรูปแบบราสเตอร์เป็นเวคเตอร์ได้

1.6.2 ข้อมูลแบบเวคเตอร์ที่ได้มีลักษณะเป็นโครงสร้างแบบมาตราฐานสำหรับ GIS

1.6.3 รูปแบบข้อมูลเวคเตอร์ที่ได้มีการเก็บความสัมพันธ์ซึ่งกันและกัน ทำให้สามารถนำมาใช้ในการวิเคราะห์ในระบบ GIS ได้อย่างสะดวกรวดเร็วยิ่งขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ความรู้พื้นฐานเกี่ยวกับระบบสารสนเทศทางภูมิศาสตร์

2.1 ระบบข้อมูล (Information System)

ระบบข้อมูล หมายถึง การนำข้อมูลที่เกี่ยวข้องกับทรัพยากรมนุษย์ (Human Resources) ผสมผสานกับข้อมูลทรัพยากรเฉพาะด้านต่างๆ (Technical Resources) แล้วผ่านขบวนการจัดการข้อมูล โดยองค์การซึ่งรับผิดชอบในการผลิตข้อมูล เพื่อสนับสนุนและสนองตอบต่อความต้องการของการจัดการ

ระบบข้อมูลสามารถแบ่งออกได้เป็น 2 ลักษณะ คือ

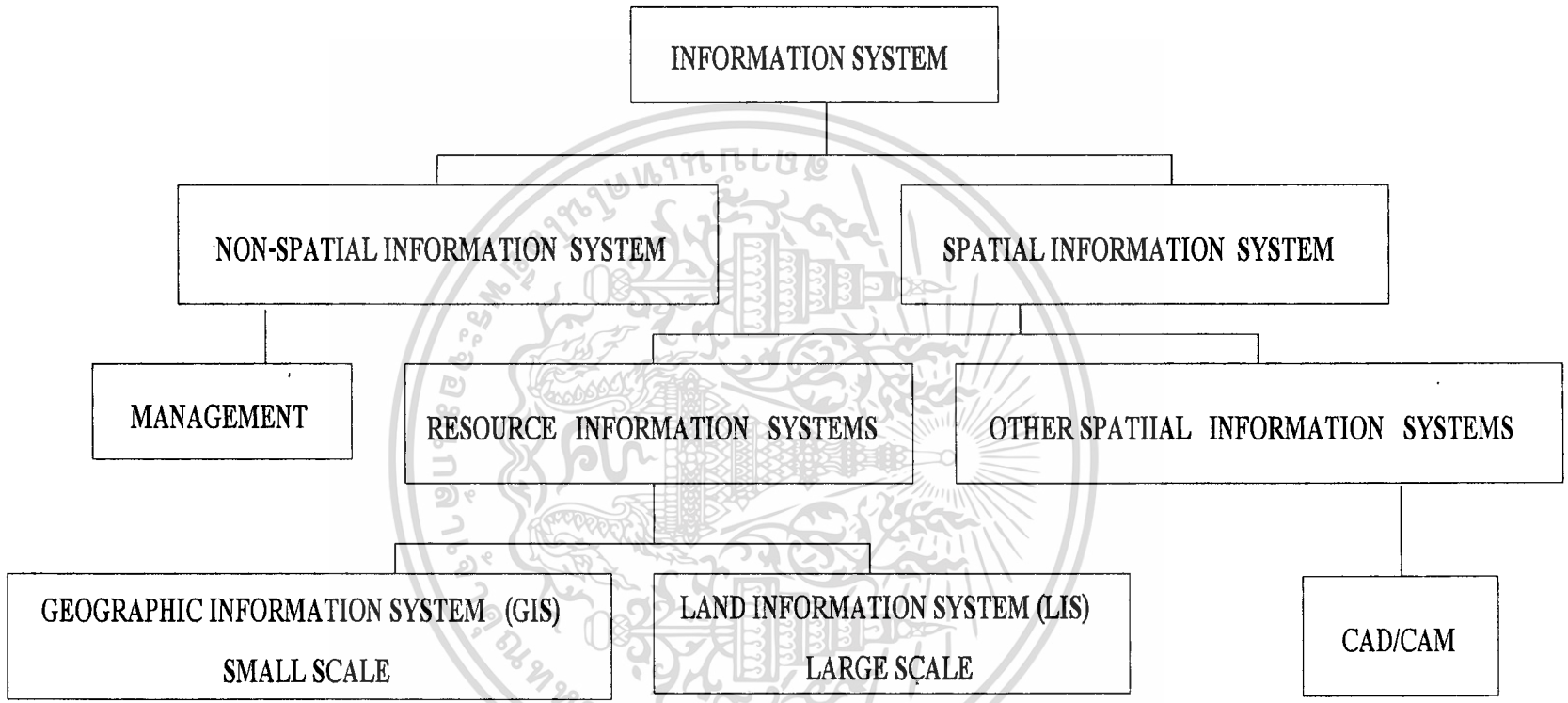
2.1.1 Non - Spatial Information Systems ได้แก่ ระบบข้อมูลในลักษณะของการจัดการในด้านต่าง ๆ

2.1.2 Spatial Information Systems ได้แก่ ระบบข้อมูลด้านทรัพยากร (Resource Information Systems) อันประกอบไปด้วย ระบบสารสนเทศภูมิศาสตร์ (Geographic Information Systems หรือ GIS) และระบบข้อมูลดิน (Land Information Systems หรือ LIS) และระบบข้อมูลอื่น ๆ เช่น ระบบ CAD/CAM เป็นต้น

รายละเอียดต่าง ๆ ของระบบข้อมูล แสดงดังในรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 ลักษณะของระบบข้อมูล (Infomation System Taxonomy)

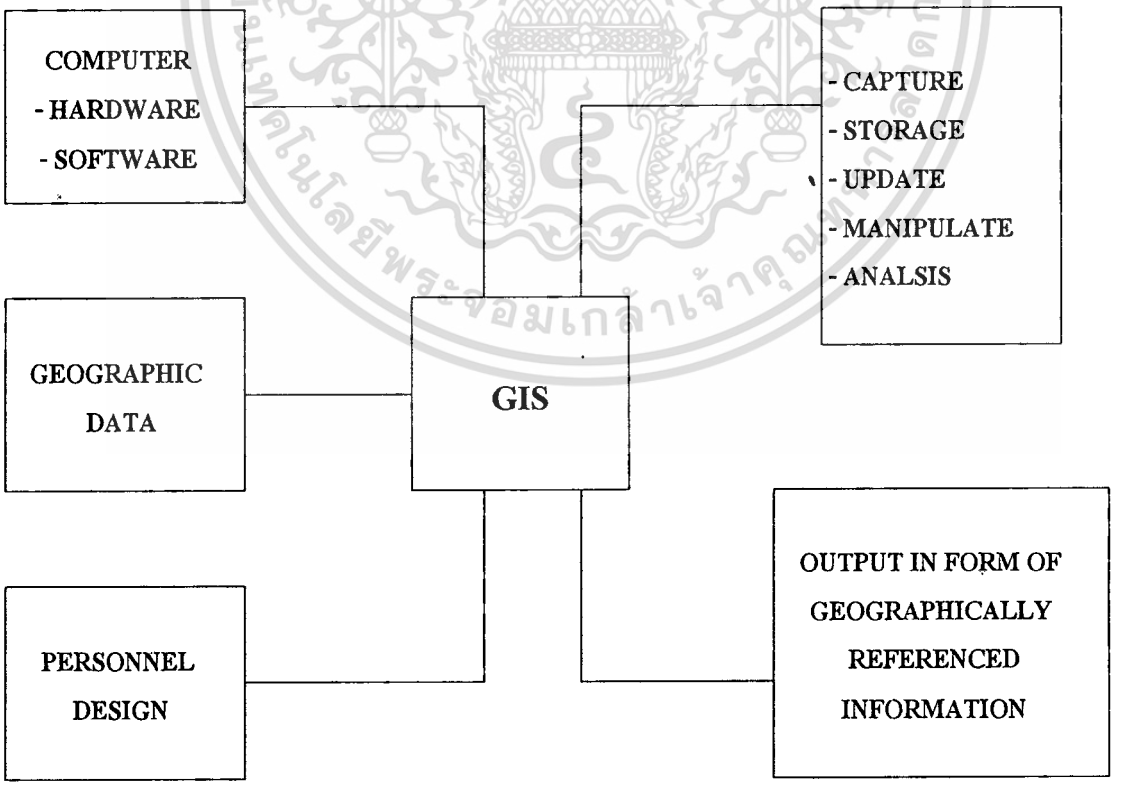


2.2 ระบบสารสนเทศภูมิศาสตร์ (Geographic Information Systems)

ระบบสารสนเทศภูมิศาสตร์ หรือ ระบบ GIS เป็นเครื่องมือที่ใช้ในการวิเคราะห์ข้อมูลเชิงพื้นที่ (Spatial Context) โดยข้อมูลลักษณะต่าง ๆ ในพื้นที่ที่ทำการศึกษา จะถูกนำมาจัดให้อยู่ในรูปแบบที่มีความสัมพันธ์เชื่อมโยงซึ่งกันและกัน ซึ่งจะขึ้นอยู่กับชนิดและรายละเอียดของข้อมูลนั้น ๆ เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดตามต้องการ นอกจากนี้ ยังมีการให้คำจำกัดความของระบบสารสนเทศภูมิศาสตร์ในหลายลักษณะ เช่น

ระบบสารสนเทศภูมิศาสตร์ หมายถึง ขบวนการของการใช้คอมพิวเตอร์ฮาร์ดแวร์ (Hardware) ซอฟท์แวร์ (Software) ข้อมูลทางภูมิศาสตร์ (Geographic Data) และการออกแบบ (Personnel Design) ในการเสริมสร้างประสิทธิภาพของการจัดเก็บข้อมูล การปรับปรุงข้อมูล การคำนวณ และการวิเคราะห์ข้อมูล ให้แสดงผลในรูปของข้อมูลที่สามารถอ้างอิงได้ในทางภูมิศาสตร์ หรือ หมายถึง การใช้สมรรถนะของคอมพิวเตอร์ ในการจัดเก็บ และการใช้ข้อมูลเพื่ออธิบายสภาพต่างๆ บนพื้นผิวโลก โดยอาศัยลักษณะทางภูมิศาสตร์ เป็นตัวเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลต่าง ๆ นั่นเอง ดังแสดงในรูปที่ 2.2

รูปที่ 2.2 ลักษณะของระบบสารสนเทศภูมิศาสตร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ GIS เป็นระบบที่ออกแบบเพื่อแสดงลักษณะของข้อมูลในรูปแบบต่าง ๆ ซึ่งพอสรุปได้ดังนี้ คือ

- Environmental Information ได้แก่ ข้อมูลดิน ธรณีวิทยา แหล่งน้ำ พืชพันธุ์ และสัตว์ป่า เป็นต้น

- Infrastructure Information ได้แก่ อาคารสิ่งปลูกสร้าง สิ่งอำนวยความสะดวก ระบบสื่อสารและขนส่ง เป็นต้น

- Cadrastal Information ได้แก่ การประเมินสิทธิครอบครองกรรมสิทธิ์ และการควบคุมการใช้ที่ดิน ขอบเขตแบ่งแดน เป็นต้น

- Socio-Economic Information ได้แก่ การกระจายตัวของประชากรและสาธารณูปโภคต่าง ๆ เป็นต้น

2.3 ขบวนการในการวิเคราะห์ข้อมูลในระบบ GIS

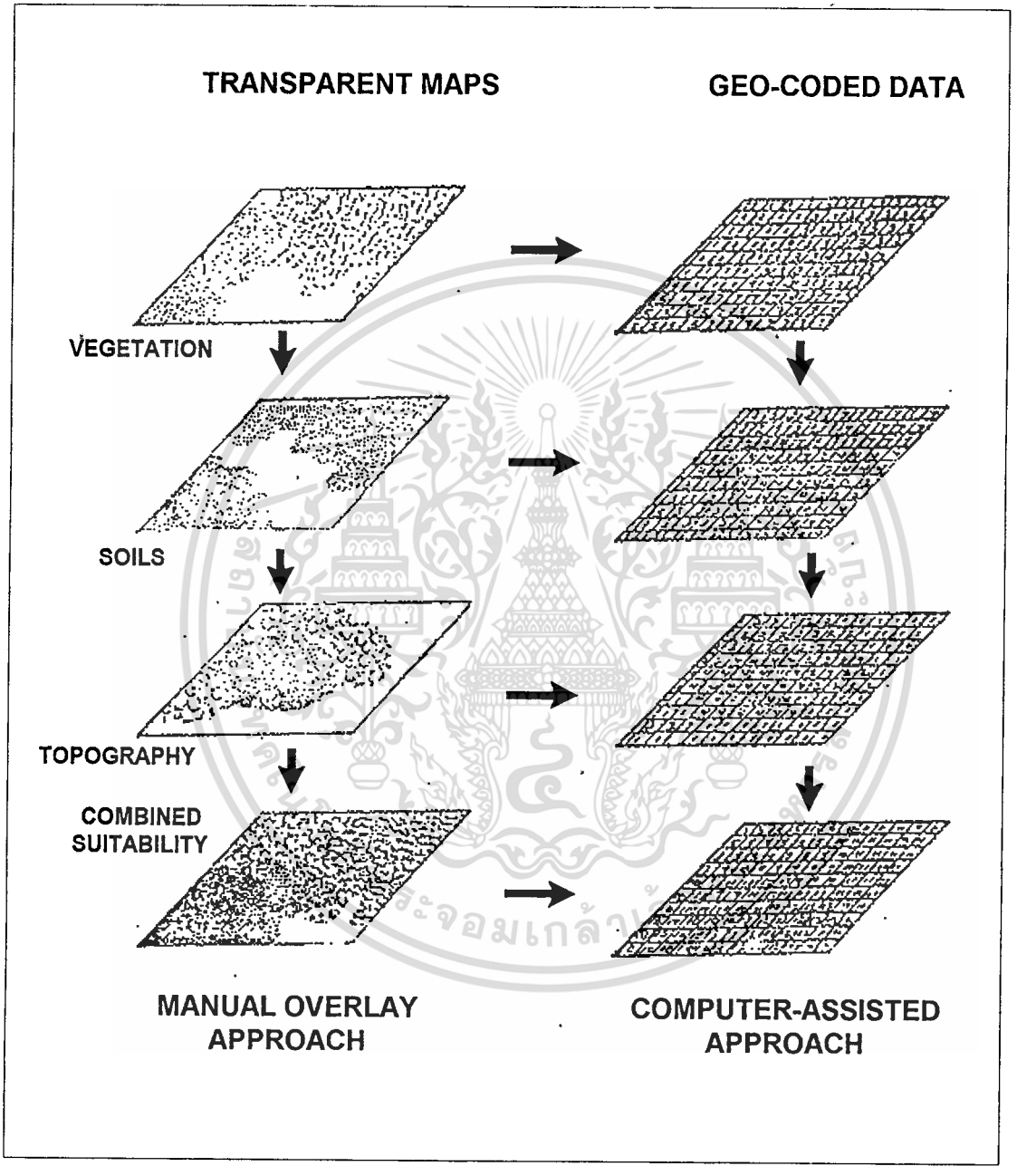
ขบวนการในการวิเคราะห์ข้อมูลในระบบ GIS แบ่งออกเป็น 2 รูปแบบ คือ

2.3.1 Manual Approach เป็นการนำข้อมูลในรูปของแผนที่หรือลายเส้นต่าง ๆ ถ่ายลงบนแผ่นใส แล้วนำมาซ้อนทับกัน (Overlay Techniques) ในแต่ละปัจจัยเพื่อให้ได้ผลลัพธ์ตามต้องการ แต่วิธีการนี้มีข้อจำกัดในเรื่องของจำนวนแผ่นใสที่จะนำมาซ้อนทับกัน ทั้งนี้เนื่องจากความสามารถในการวิเคราะห์ด้วยสายตา (Eyes Interpretation) จะกระทำได้ในจำนวนของแผ่นใสที่ค่อนข้างจำกัด และจำเป็นต้องใช้เนื้อที่และวัสดุในการเก็บข้อมูลค่อนข้างมาก

2.3.2 Computer Assisted Approach เป็นการวิเคราะห์ข้อมูลในรูปของตัวเลข หรือดิจิตอล (Digital) โดยการเปลี่ยนรูปแบบของข้อมูลแผนที่หรือลายเส้นให้อยู่ในรูปของตัวเลข แล้วทำการซ้อนทับกันโดยการนำหลักคณิตศาสตร์และตรรกศาสตร์เข้ามาช่วย วิธีการนี้จะช่วยให้ลดเนื้อที่ในการเก็บข้อมูลลง และสามารถเรียกมาแสดงหรือทำการวิเคราะห์ได้โดยง่าย ตัวอย่างของการวิเคราะห์ข้อมูลในรูปแบบดังกล่าวข้างต้น ดังแสดงในรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 การวิเคราะห์ข้อมูลในรูปแบบของ GIS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 องค์ประกอบของระบบสารสนเทศภูมิศาสตร์ (Components of Geographic Information Systems)

ในที่นี้จะเป็นการอธิบายระบบ GIS ในรูปแบบของ Computer Assisted Approach ซึ่งประกอบไปด้วยส่วนสำคัญ 3 ส่วน คือ

2.4.1 คอมพิวเตอร์ฮาร์ดแวร์ (Computer Hardware)

2.4.2 คอมพิวเตอร์ซอฟท์แวร์ (Computer Software)

2.4.3 องค์กรในการดำเนินงาน (Proper Organizational Context)

2.4.1 คอมพิวเตอร์ฮาร์ดแวร์ (Computer Hardware)

ในส่วนของคอมพิวเตอร์ฮาร์ดแวร์ จะประกอบไปด้วยส่วนต่าง ๆ ดังรูปที่ 2.4 คือ

1. หน่วยประมวลผลกลาง (Central Processing Unit หรือ CPU) ซึ่งจะมีหน่วยควบคุม (Control Unit หรือ CU) ในการจัดลำดับของระบบ และหน่วยคำนวณเปรียบเทียบข้อมูล (Arithmetic - Logic Unit หรือ ALU) โดยใช้หลักคณิตศาสตร์ และตรรกศาสตร์

2. หน่วยจัดเก็บข้อมูลด้วยเครื่องขับดิสค์ (Disk Drive Stroage Unit) โดยปกติเครื่องขับดิสค์จะมีอยู่ 2 แบบ คือ เครื่องขับฮาร์ดดิสค์ (Hard Disk Drive) ซึ่งมีความจุของดิสค์ตั้งแต่ 10 Mb จนถึง 300 Mb เป็นต้น กับเครื่องขับฟลอปปี้ดิสค์ (Floppy Disk Drive) ซึ่งจะมีเครื่องขับดิสค์ขนาด 5.25 นิ้ว มีความจุ 360 Kb หรือ 1.2 Mb และขนาด 3.5 นิ้ว ที่มีความจุ 1.4 Mb เป็นต้น

3. ดิจิไทเซอร์ (Digitizer) เป็นส่วนในการเปลี่ยนรูปแบบข้อมูลจากแผนที่ให้อยู่ในรูปของดิจิตอลจัดส่งไปยังหน่วยประมวลผลกลางและหน่วยจัดเก็บข้อมูล

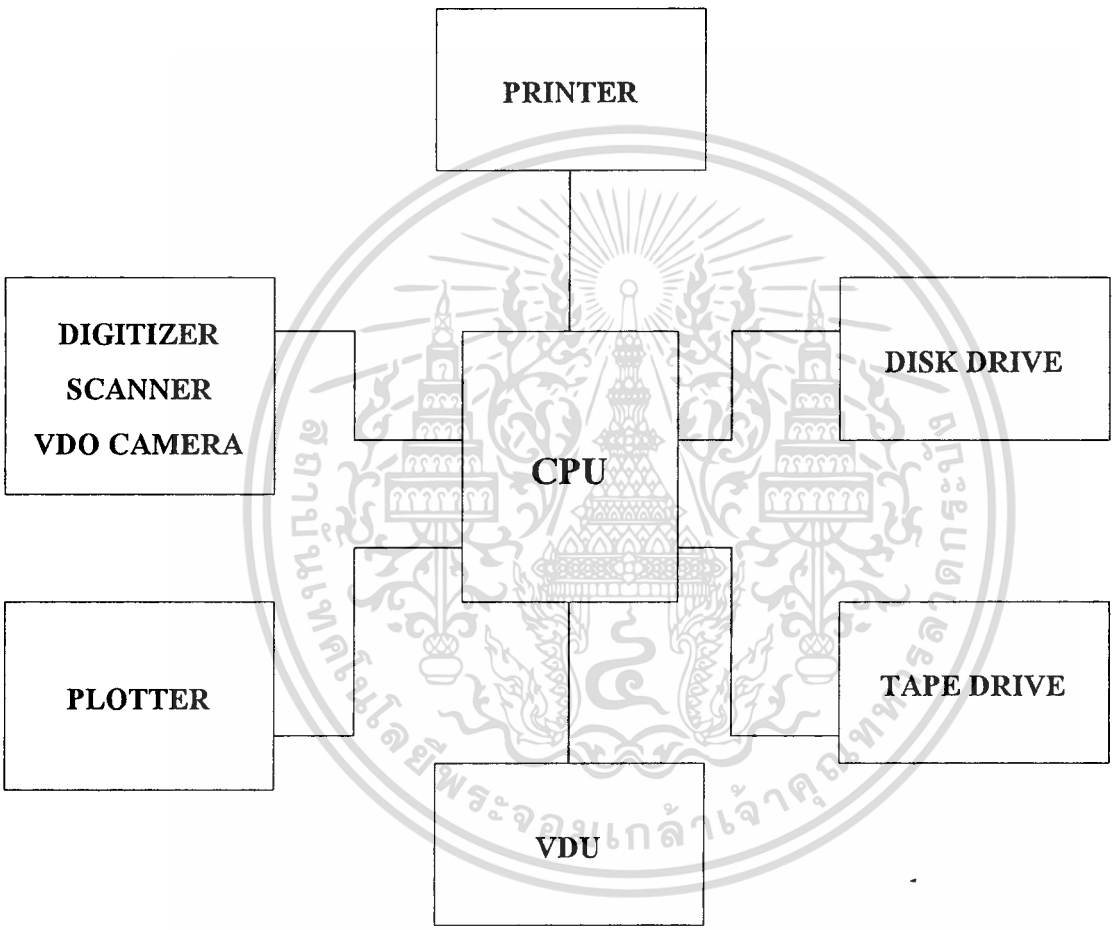
4. พลอตเตอร์ (Plotter) และ พริ้นเตอร์ (Printer) สำหรับแสดงผลโดยพลอตเตอร์จะแสดงข้อมูลที่เป็นลายเส้น ส่วนพริ้นเตอร์จะแสดงข้อมูลตัวหนังสือหรือข้อความต่าง ๆ (Text)

5. เครื่องขับเทป (Tape Drive) จะใช้ในการเก็บรวบรวมข้อมูลลงในเทปแม่เหล็ก (Magnetic Tape) ที่มีความหนาแน่น 1600 BPI (Bits Per Inch) หรือ 6250 BPI

6. หน่วยแสดงผล (Visual Display Unit หรือ Terminal) เป็นส่วนที่ใช้ในการควบคุมคอมพิวเตอร์และอุปกรณ์ประกอบต่าง ๆ (Peripherals) อันได้แก่ พลอตเตอร์ พริ้นเตอร์ ดิจิไทเซอร์ หรือเครื่องมืออื่น ๆ ที่เชื่อมโยงกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4 ส่วนประกอบของคอมพิวเตอร์ฮาร์ดแวร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 คอมพิวเตอร์ซอฟท์แวร์ (Computer Software)

ซอฟท์แวร์ในระบบ GIS จะประกอบด้วยส่วนที่สำคัญ 5 ประการคือ

1. การป้อนข้อมูลและการตรวจสอบข้อมูล (Data Input and Verification) จะเป็นการเปลี่ยนข้อมูลจากแผนที่ต้นแบบ ข้อมูลดาวเทียม ภาพถ่ายทางอากาศ ให้อยู่ในรูปของดิจิตอล โดยมีเครื่องมือที่ใช้ในการนี้ เช่น Terminal หรือ VDU, Digitizer, Scanner เป็นต้น

2. การจัดเก็บข้อมูลและการจัดการฐานข้อมูล (Data Storage and Database Management) เป็นการจัดเก็บข้อมูลทางภูมิศาสตร์เกี่ยวกับ จุด เส้น หรือพื้นที่ (Position Topology, Attribute) ให้มีโครงสร้างที่สามารถจัดเก็บไว้ในคอมพิวเตอร์ และผู้ใช้สามารถเรียกมาใช้ได้สะดวก

3. การคำนวณและการวิเคราะห์ข้อมูล (Data Manipulation and Analysis) ในส่วนนี้จะมีศักยภาพในการคำนวณและวิเคราะห์ข้อมูลหลายรูปแบบ และจะปรับปรุง หรือเปลี่ยนแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม ซึ่งเรียกวิธีการนี้ว่า Data Transformation เพื่อแก้ไขข้อผิดพลาดของข้อมูลนั้น ๆ

4. การรายงานผลข้อมูล (Data Output and Presentation) เป็นวิธีการแสดงผลของข้อมูลที่ได้จากการวิเคราะห์ โดยผลที่ได้จะอยู่ในรูปของแผนที่ ตาราง กราฟ ฯลฯ และจะพิมพ์รายงานผลโดยใช้พลอตเตอร์ หรือ พริ้นเตอร์

5. ความสัมพันธ์กับผู้ใช้ (Interaction with the User) ซอฟท์แวร์ GIS ที่ดีนั้นจะต้องสามารถอำนวยความสะดวกให้กับผู้ใช้ได้เป็นอย่างดี โดยมีการสร้างรายการ (Menu) ต่าง ๆ ที่ไม่ยุ่งยาก เข้าใจได้ง่าย และมีขั้นตอนที่ต่อเนื่องสมบูรณ์

2.4.3 องค์กรในการดำเนินงาน (Proper Organizational Context)

การนำระบบ GIS มาใช้ในงานด้านต่าง ๆ นั้น จำเป็นจะต้องดำเนินการฝึกอบรมบุคลากรให้มีความรู้ความเข้าใจ และมีศักยภาพในการใช้คอมพิวเตอร์ทั้งฮาร์ดแวร์และซอฟท์แวร์ได้เป็นอย่างดี เพื่อให้มีความพร้อมในการที่จะรองรับความก้าวหน้าทางเทคโนโลยีของระบบ GIS โดยมีองค์กรที่มีหน้าที่รับผิดชอบในการฝึกอบรมดังกล่าว นอกจากนี้ยังต้องรับผิดชอบในการพัฒนาระบบ GIS ให้สามารถรองรับและตอบสนองต่อการวางแผนและการจัดการได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ลักษณะของข้อมูลในระบบสารสนเทศภูมิศาสตร์ (Characteristics of GIS Information)

ลักษณะของข้อมูลในระบบ GIS แบ่งออกได้เป็น 2 ลักษณะ คือ

2.5.1 ลักษณะข้อมูลเชิงเฉพาะ (Attribute Characteristics)
2.5.2 ลักษณะข้อมูลเชิงพื้นที่ (Spatial Characteristics)

2.5.1 ลักษณะข้อมูลเชิงเฉพาะ (Attribute Characteristics)

ลักษณะข้อมูลเชิงเฉพาะ หมายถึง ลักษณะประจำตัวหรือลักษณะที่มีความแปรผันในการชี้วัดปรากฏการณ์ต่าง ๆ ตามธรรมชาติ โดยจะระบุถึงสถานที่ที่ทำการศึกษา ในช่วงระยะเวลาหนึ่ง ๆ ลักษณะข้อมูลเชิงเฉพาะ (Attribute) อาจมีลักษณะที่ต่อเนื่องกัน เช่น เส้นชั้นระดับความสูง (Terrain Elevation) หรือเป็นลักษณะที่ไม่ต่อเนื่อง เช่น จำนวนพลเมือง (Number of Inhabitants) และชนิดของสิ่งปกคลุมดิน (Land Cover Types) เป็นต้น ค่าความแปรผันของลักษณะข้อมูลเชิงเฉพาะนี้ จะทำการชี้วัดออกมาในรูปของตัวเลข (Numeric) โดยกำหนดเกณฑ์การวัดออกเป็น 3 ระดับ คือ

1. Nominal Level เป็นระดับที่มีการวัดข้อมูลอย่างหยาบ ๆ โดยจะกำหนดตัวเลขหรือสัญลักษณ์ เพื่อจำแนกลักษณะของสิ่งต่าง ๆ เท่านั้น เช่น การใช้ประโยชน์ที่ดินในพื้นที่หนึ่งจำแนกได้เป็น ป่าไม้ แหล่งน้ำ ทุ่งหญ้า ฯลฯ เป็นต้น ลักษณะเหล่านี้อาจจะแทนค่าโดยตัวเลข เช่น 1 = ป่าไม้, 2 = ทุ่งหญ้า, 3 = แหล่งน้ำ เป็นต้น

2. Ordinal Level หรือ Ranking Level เป็นการเปรียบเทียบลักษณะในแต่ละปัจจัยว่ามีขนาดเล็กกว่า เท่ากัน หรือ ใหญ่กว่า เช่น พื้นที่ป่าไม้มีขนาดใหญ่กว่าพื้นที่ทุ่งหญ้า หรือ 1 > 2 เป็นต้น

3. Interval - Ratio Level เป็นการพิจารณาถึงความสัมพันธ์ในแต่ละปัจจัยของ Ordinal Level ว่ามีความแตกต่างกันมากน้อยเพียงใด เช่น พื้นที่ป่าไม้มีขนาดใหญ่กว่าพื้นที่ทุ่งหญ้า 2 เท่า เป็นต้น

รายละเอียดของเกณฑ์การวัดในระดับต่าง ๆ ดังแสดงในตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ลักษณะของเกณฑ์การวัดในระดับต่าง ๆ

	NOMINAL	ORDINAL	INTERVAL-RATIO
INFORMATION CONTENT	IDENTIFICATION	INDENTIFICATION RANKING	INDENTIFICATION RANKING MEANING OF INTERVAL VALUES
PERMISSIBLE OPERATION	SOME LOGICAL OPERATION	LOGICAL OPERATION	LOGICAL AND ARITHMETICAL OPERATION
RELAVANT STATISTICS	MODE CONTINGENCY COEFFICIENT	MEDIAN PERCENTILES	MEAN, VARIANCE COEFFICIENT OF CORRELATION

2.5.2 ลักษณะข้อมูลเชิงพื้นที่ (Spatial Characteristics)

ลักษณะข้อมูลพื้นที่ จะมีลักษณะ และรูปแบบ (Spatial Features) ต่าง ๆ กัน พอสรุปได้ดังนี้ คือ

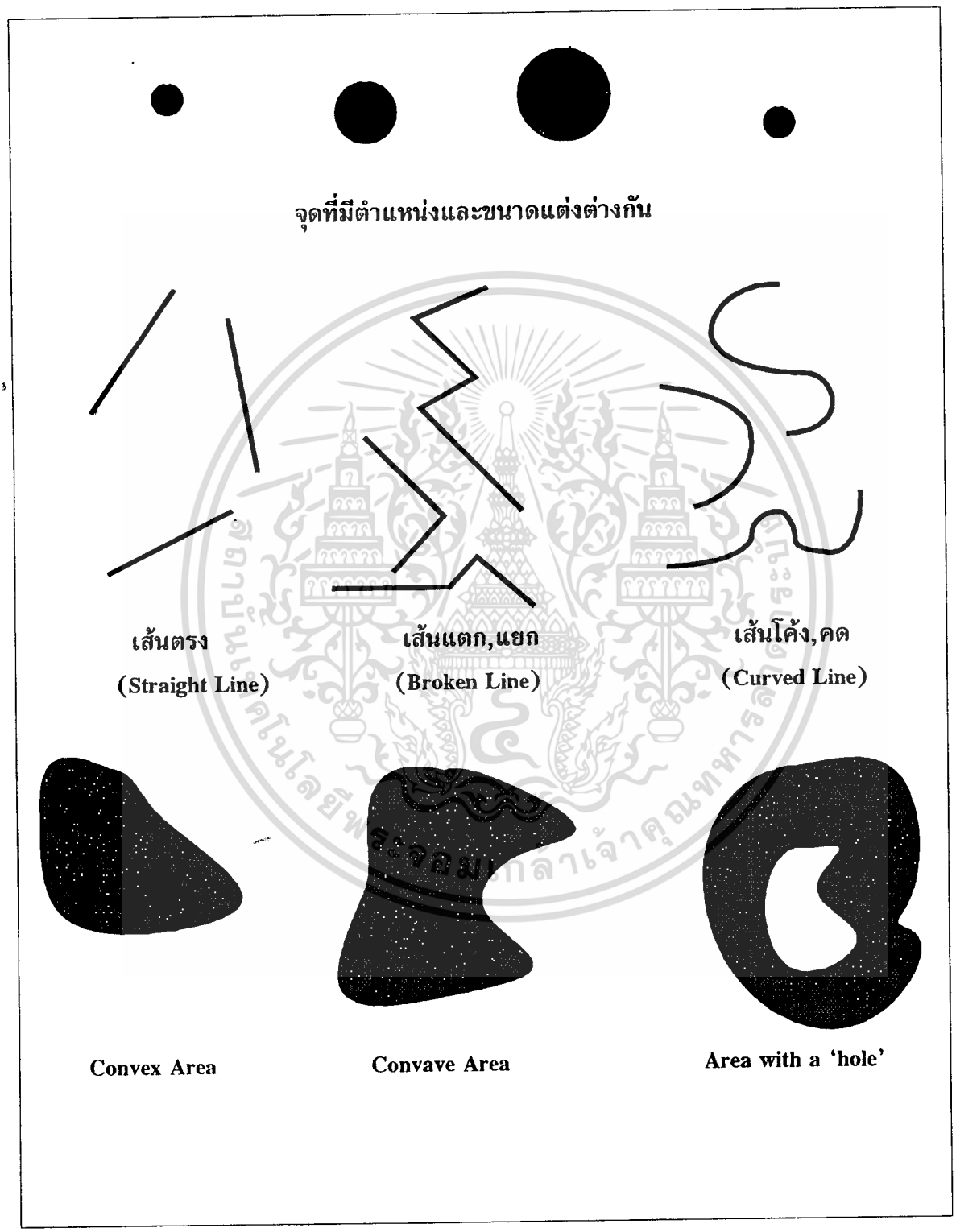
1. รูปแบบของจุด (Point Features) เป็นลักษณะของจุดในตำแหน่งใด ๆ ซึ่งจะสังเกตุได้จากขนาดของจุดนั้น ๆ โดยจะอธิบายถึงตำแหน่งที่ตั้งของข้อมูล เช่น ที่ตั้งของจังหวัด เป็นต้น

2. รูปแบบของเส้น (Linear Features) ประกอบไปด้วยลักษณะของเส้นตรง เส้นหักมุม และเส้นโค้ง ซึ่งรูปร่างของเส้นเหล่านี้จะอธิบายถึงลักษณะต่าง ๆ โดยอาศัยขนาดทั้งความกว้างและความยาว เช่น ถนน หรือแม่น้ำ เป็นต้น และในทางการทำแผนที่รวมทั้งระบบ GIS นั้น รูปแบบของเส้น หมายถึง เส้นหักมุมที่มีความกว้างเฉพาะในความยาวที่กำหนด

3. รูปแบบของพื้นที่ (Areal Features) เป็นลักษณะขอบเขตพื้นที่ที่เรียกว่า โพลีกอน(Polygon) ซึ่งจะประกอบด้วยลักษณะแบบต่าง ๆ คือ Convex, Concave, Area with a Hole ลักษณะเหล่านี้จะอธิบายขอบเขตของข้อมูลต่าง ๆ เช่น ขอบเขตของพื้นที่ป่าไม้ ขอบเขตของจังหวัด ประเทศ เป็นต้น รายละเอียดของรูปแบบข้อมูลเชิงพื้นที่ (Spatial Features) นี้ดังแสดงรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

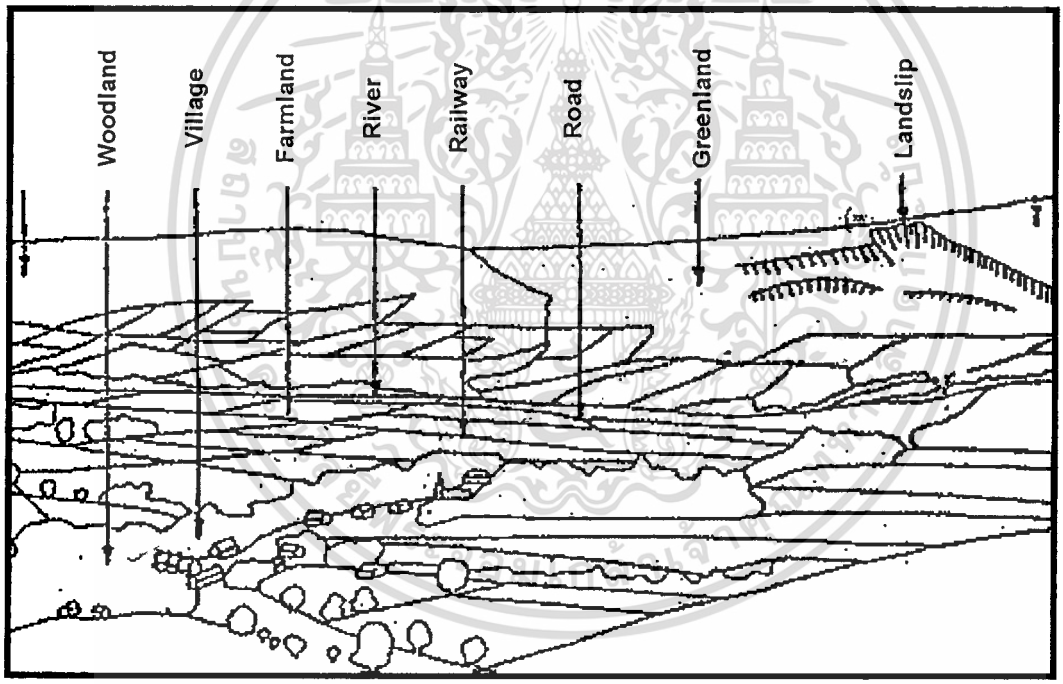
รูปที่ 2.5 แสดงรูปแบบข้อมูลเชิงพื้นที่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะข้อมูล Attribute และ Spatial นี้จะมีความสัมพันธ์ซึ่งกันและกัน โดยความสัมพันธ์ดังกล่าวเป็นไปได้ทั้งในแบบต่อเนื่อง (Continuous) และไม่ต่อเนื่อง (Discrete) ยกตัวอย่างเช่น แผนที่ภูมิประเทศ (Topograghic Map) จะแสดงถึงเส้นระดับความสูงที่มีความสัมพันธ์กันอย่างต่อเนื่อง ในขณะที่จำนวนประชากร ที่อยู่อาศัยในแต่ละชั้นระดับความสูงนั้น จะมีความสัมพันธ์ในลักษณะที่ไม่ต่อเนื่อง โดยจะแปรผันไปตามปัจจัยและสภาพแวดล้อมที่เอื้ออำนวยต่อการดำรงชีวิตเท่านั้น เป็นต้น รูปแบบของความสัมพันธ์ระหว่างลักษณะข้อมูล Attribute และ Spatial ดังแสดงในรูปที่ 2.6

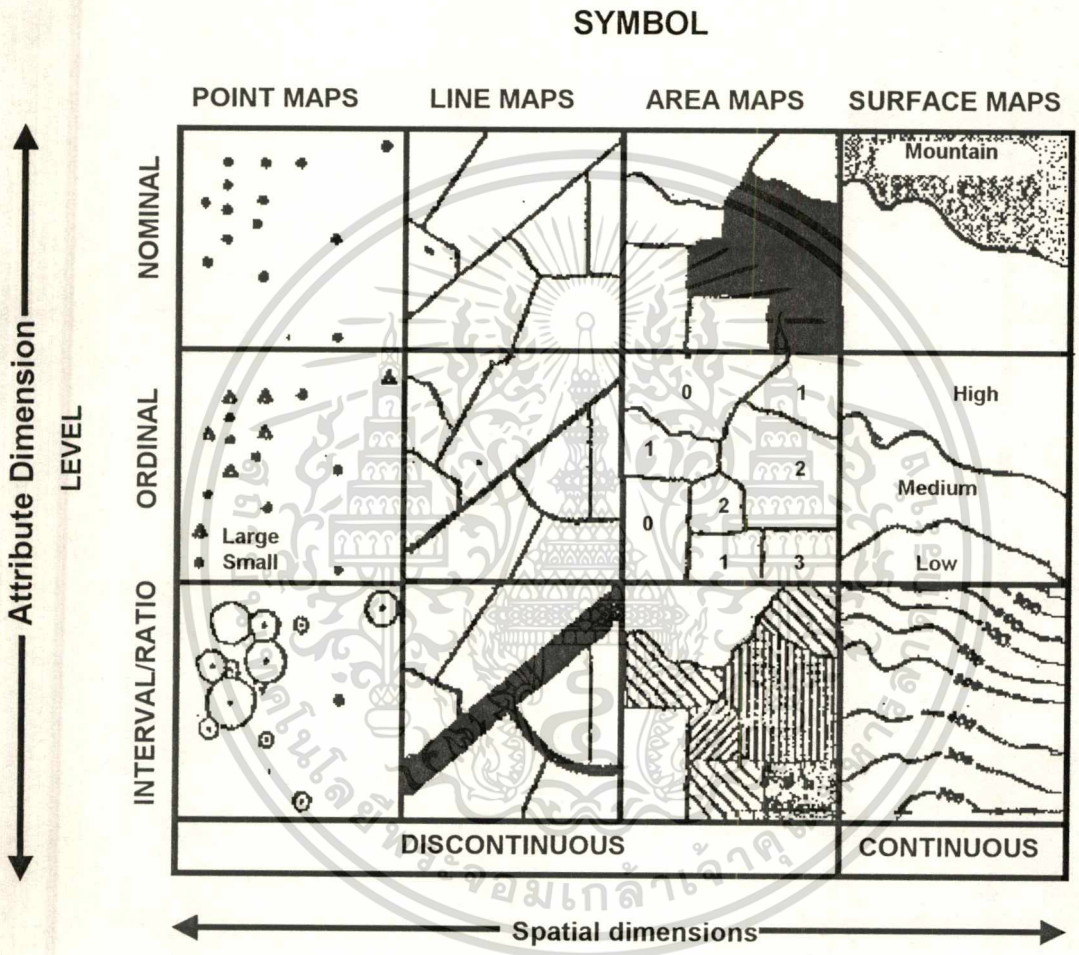
รูปที่ 2.6 รูปแบบความสัมพันธ์ระหว่างลักษณะข้อมูลเชิงเฉพาะและลักษณะข้อมูลเชิงพื้นที่



ข้อมูลในลักษณะต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.6 รูปแบบความสัมพันธ์ระหว่างลักษณะข้อมูลเชิงเฉพาะและลักษณะข้อมูลเชิงพื้นที่(ต่อ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การจัดเก็บข้อมูลในระบบสารสนเทศภูมิศาสตร์
(Data Collection and Stroage)

3.1 การจัดเก็บข้อมูลในระบบสารสนเทศภูมิศาสตร์ (Data Collection and Stroage)

ขบวนการจัดเก็บข้อมูลประกอบไปด้วยขั้นตอนต่าง ๆ คือ

- การจัดเตรียมเอกสาร (Document Preparation)
- การจัดเก็บข้อมูลในระบบตัวเลข (Numerical Capture of Data)
- การจัดเตรียมข้อมูล (Data Pre - Processing)
- การจัดเก็บข้อมูลในรูปแบบ GIS (Data Stroage in GIS Format)

ซึ่งปัจจัยสำคัญที่มีส่วนเกี่ยวข้องในขบวนการดังกล่าวที่นำมาพิจารณาจะประกอบไปด้วย

3.1.1 คุณลักษณะเฉพาะของระบบ GIS (GIS Characteristics)

ลักษณะโครงสร้างของระบบ GIS จะต้องคำนึงถึงสิ่งต่างๆ คือ ข้อมูลที่เป็นประโยชน์ ข้อจำกัดของพื้นที่ที่ทำการศึกษา ขนาด และการกำหนดทิศทางของกริด โดยขนาดของกริดจะขึ้นอยู่กับขนาดของมาตราส่วนนั้น ๆ ในการวิเคราะห์ข้อมูลระดับทวีปหรือระดับชาติขนาดของกริดจะมีขนาดใหญ่กว่าขนาดของกริดที่ใช้ในการวิเคราะห์ข้อมูลระดับภูมิภาคหรือท้องถิ่น นอกจากนี้ขนาดของกริดยังขึ้นอยู่กับความเหมาะสมของคอมพิวเตอร์อีกด้วย การจัดการระบบ GIS จำเป็นที่จะต้องใช้หน่วยความจำในการเก็บรวบรวมข้อมูลที่มีขนาดค่อนข้างใหญ่มาก ซอฟท์แวร์ของระบบ GIS ส่วนใหญ่จึงมักจะมีข้อจำกัดในการจัดขนาดของกริดต่างๆ การเปลี่ยนขนาดของกริดที่มีขนาดเล็กให้มีขนาดใหญ่ขึ้นสามารถที่จะกระทำได้ แต่ในทางกลับกันกริดที่มีขนาดใหญ่จะปรับแก้ให้มีขนาดเล็กลงนั้นกระทำได้ยากมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 แหล่งของข้อมูล (Sources of Information)

ข้อมูลต่าง ๆ ในระบบ GIS สามารถพิจารณาได้ใน 2 ลักษณะคือ ข้อมูลเชิงพื้นที่ (Spatial) และข้อมูลเชิงเฉพาะ (Attribute) ข้อมูลเชิงพื้นที่จะเป็นได้ทั้งรูปแบบของอนาลอก (Analog) หรือ ดิจิตอล (Digital) โดยที่ Analog จะเป็นส่วนประกอบแผนภาพของข้อมูล เช่น แผนที่ รูปภาพ ฯลฯ เป็นต้น ในขณะที่ Digital เป็นข้อมูลที่มีรายละเอียดของลักษณะข้อมูล Spatial ที่อยู่ในรูปของตัวเลข รูปแบบของ Analog จะประกอบด้วยขบวนการจัดเก็บข้อมูลทั้ง 4 ขั้นตอนดังกล่าวข้างต้น คือ การเตรียมเอกสาร การจัดเก็บข้อมูลในระบบตัวเลข การจัดเตรียมข้อมูล และการจัดเก็บข้อมูลในรูปแบบของ GIS แต่ในบางกรณีข้อมูลอาจอยู่ในรูปของตัวเลขของคู่พิกัดอยู่แล้ว ดังนั้น จึงมีเพียงขั้นตอนของการจัดเตรียมข้อมูล และการจัดเก็บข้อมูลในรูปแบบของ GIS เท่านั้น นอกจากนี้ข้อมูลที่อยู่ในรูปของกริดที่เป็นลักษณะของ Matrix ก็สามารถจัดเก็บในรูปแบบ GIS ได้โดยตรง สำหรับข้อมูล Attribute ส่วนใหญ่มักอยู่ในรูปของตัวเลขจึงเป็นการจัดเก็บในคอมพิวเตอร์เพื่อเป็นการจัดเตรียมและจัดเก็บข้อมูลในรูปแบบของ GIS ต่อไป

3.1.3 วิธีการที่ใช้ในการเก็บข้อมูล (Data Capture)

วิธีการที่ใช้ในการป้อนข้อมูลและจัดเก็บข้อมูลนั้นมีเป็นจำนวนมาก ในที่นี้ได้แบ่งวิธีการต่าง ๆ ออกเป็น 3 ประเภท คือ

1. Manual Capture
2. Semi - Automatic Capture
3. Automatic Capture

1. Maunal Capture วิธีการเก็บข้อมูลแบบ Manual Capture ส่วนใหญ่จะเป็นการเก็บข้อมูล Spatial และ Attribute ที่อยู่ในรูปของตัวเลขโดยใช้ Terminal Keyboard ป้อนข้อมูลผ่านไปยังระบบคอมพิวเตอร์โดยตรง วิธีการนี้จะประหยัดในด้านค่าใช้จ่ายแต่ก็อาจจะเกิดปัญหาในส่วนของการป้อนข้อมูลที่มีเป็นจำนวนมาก วิธีการนี้จำเป็นที่จะต้องเปลี่ยนข้อมูลให้อยู่ในรูปของตัวเลขก่อนทำการจัดเก็บไว้ในคอมพิวเตอร์ โดยจะเก็บได้ทั้งในลักษณะของเวคเตอร์และราสเตอร์ ซึ่งพอสรุปได้ดังนี้ คือ

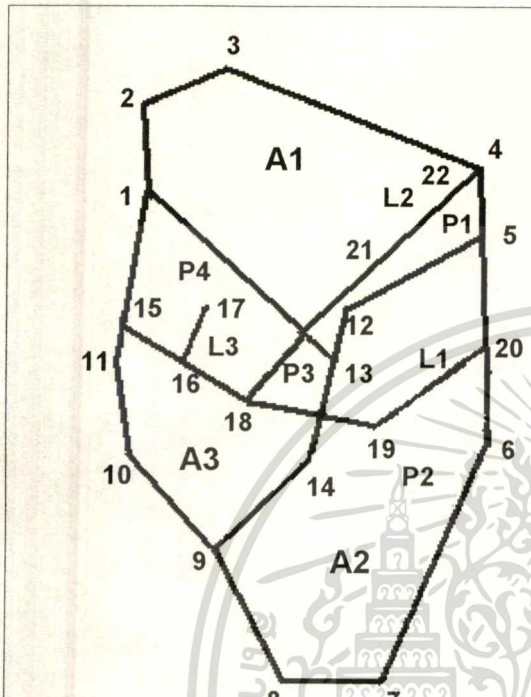
- การเก็บข้อมูลแบบ Manaul Capture ในระบบเวคเตอร์ นั้น แหล่งของข้อมูลส่วนใหญ่จะอยู่ในรูปลักษณะของจุด เส้น หรือ พื้นที่ โดยจะบันทึกข้อมูลคู่พิกัดของจุดยอดที่ประกอบกันขึ้นเป็นรูปลักษณะต่าง ๆ การเก็บข้อมูลที่เป็นข้อมูล Spatial ที่มีการกระจายแบบไม่ต่อเนื่อง (Discreted) จะเป็นการปรับปรุงแผนที่ต้นแบบ เพื่อลดจำนวนจุดยอดที่จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

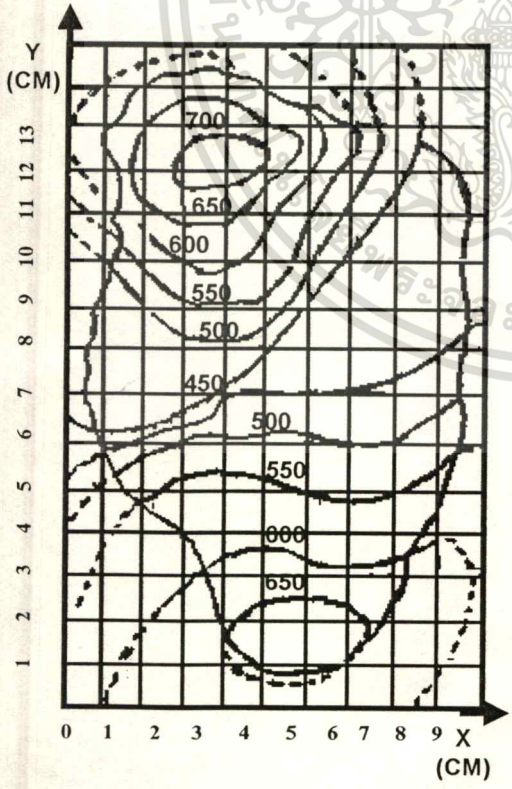
บันทึกให้น้อยลง เรียกวิธีการนี้ว่า "Generalization" ส่วนข้อมูลที่มีการกระจายแบบต่อเนื่อง (Continous) เช่น ระดับความสูงนั้นจะใช้ขบวนการที่เรียกว่า "Point Sampling Procedure" โดยทำการเลือกจุดไปตามเส้นไอโซไลน์ (Isoline) ของแผนที่ต้นแบบ จุดแต่ละจุดจะประกอบด้วยคู่พิกัด (X,Y) และค่าของจุด (Z) นั้น ๆ เช่น ค่าระดับความสูง เป็นต้น การจัดเก็บอาจจะกระทำในรูปของ Triplets โดยการบอกค่าของพิกัด (X,Y) และค่าระดับความสูง (Z) เรียงลำดับตั้งแต่จุดแรกถึงจุดสุดท้าย หรือบอกค่าระดับความสูง (Z) ในแต่ละค่าว่ามีจุดพิกัดใดบ้างเป็นองค์ประกอบ ดังตัวอย่างแสดงในรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 การเก็บข้อมูลแบบ Manual Capture ในระบบเวคเตอร์



FEATIRE	VERTEX#
A1	1,2,3,4,5,12,13
A2	5,6,7,8,9,14,13,12
A3	1,13,14,9,10,11
L1	15,16,18,19,20
L2	18,21,22
L3	16,17
P1	21
P2	19
P3	18
P4	17
.	.
.	.



1. List of Triplets

PL	X	Y	Z
1	6.0	2.4	650
2	9.6	7.0	500
3	10.0	8.7	450
4	4.4	12.0	700
5	3.0	8.4	500
.	.	.	.
.	.	.	.

2. Sequentially ordered List

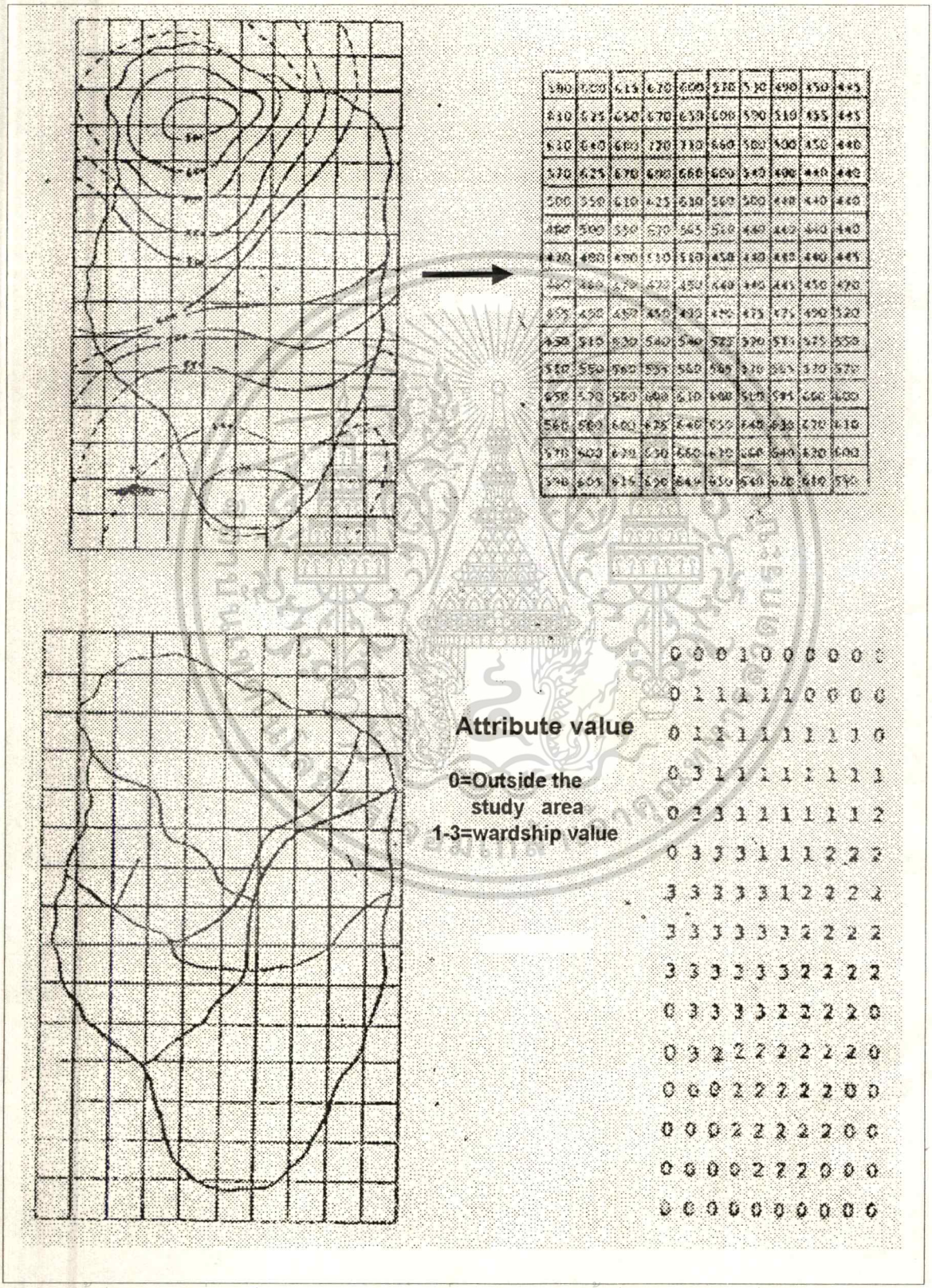
X	Y	Z
450		
	10.0	8.7
	9.2	8.0
	.	.
500		
	3.0	3.4
	.	.
100		
	1.4	12.0
	4.0	12.8
	.	.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การจัดเก็บข้อมูลแบบ Manual Capture ในระบบราสเตอร์ วิธีการนี้กระทำโดยการจัดทำตารางกริด ที่มีขนาดและจำนวนตามต้องการ โดยส่วนใหญ่จะขึ้นอยู่กับการวิเคราะห์ตามมาตราส่วนของแผนที่นั้น ๆ ลงบนแผ่นใส แล้วนำแผ่นใสนั้นมาซ้อนทับกันลงบนแผนที่ต้นแบบบันทึกข้อมูลค่าของลักษณะต่าง ๆ ในแต่ละกริด โดยเปลี่ยนให้อยู่ในรูปของตัวเลข ลักษณะของข้อมูลจะอยู่ในรูปของ Matrix เพื่อทำการจัดเก็บในระบบคอมพิวเตอร์ ตัวอย่างของการจัดเก็บข้อมูลในระบบราสเตอร์นี้ ดังแสดงในรูปที่ 3.2 โดยขนาดของกริดที่กำหนด คือ 1 ตารางกิโลเมตร จำนวนกริดที่ครอบคลุมพื้นที่ตัวอย่าง คือ จำนวน 10 กริด ในแถวแนวนอน และ 15 กริด ในแถวแนวตั้ง (Column) ข้อมูลที่มีการกระจายแบบไม่ต่อเนื่อง เช่น เมือง A, B, C และจะถูกแทนค่าด้วยเลข 1, 2 และ 3 ตามลำดับ และในข้อมูลที่มีการกระจายแบบต่อเนื่อง เช่น ระดับความสูงนั้นในแต่ละกริดจะแทนค่าของแต่ละชั้นความสูง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 การเก็บข้อมูลแบบ Manual Capture ในระบบราสเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

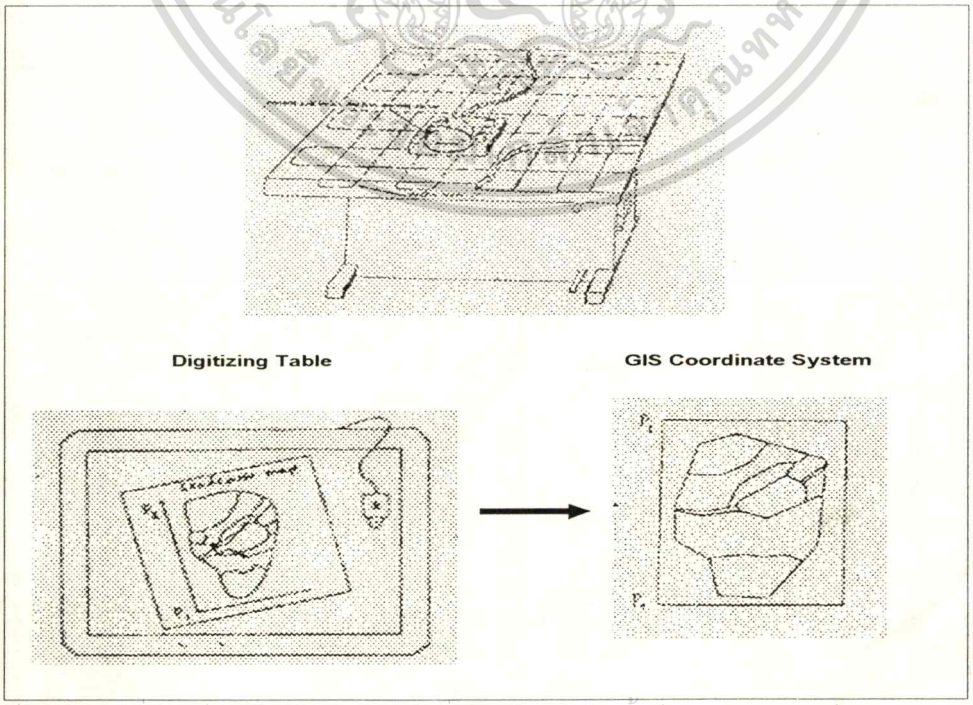
2. Semi - Automatic Capture เนื่องจากการจัดเก็บข้อมูลในระบบ Manual จำเป็นที่จะต้องใช้เวลาค่อนข้างมากในการบันทึกค่าพิกัดลงในกระดาษ แล้วปรับเปลี่ยนให้เก็บลงในคอมพิวเตอร์ ดังนั้นจึงได้มีเครื่องมือช่วยผ่อนแรงในเรื่องดังกล่าว เรียกว่า "Digitizer" ดังแสดงในรูปที่ 3.3 ซึ่งเป็นเครื่องมืออิเลคโทรนิค (Electronic) หรือ อิเลคโทรแมกเนติก (Electromagnetic) ที่ประกอบขึ้นในลักษณะคล้ายกับโต๊ะเขียนแบบ เพื่อให้สามารถวางแผนที่ต้นแบบได้ การบันทึกข้อมูลจุดพิกัดต่าง ๆ ลงในคอมพิวเตอร์ กระทำโดยใช้เครื่องมือที่เรียกว่า Mouse หรือ Puck เคลื่อนที่ไปตามจุดพิกัดของขอบเขตที่ต้องการ ซึ่งมีขั้นตอนพอสรุปได้ดังนี้

- กำหนดมาตราส่วนและจุดควบคุม (Control Point) เพื่อเป็นตัวควบคุมความสัมพันธ์ระหว่างจุดพิกัดต่าง ๆ ส่วนใหญ่จะกำหนดจุดควบคุม 2 จุด โดยจุดแรกจะเป็นจุดกำเนิดซึ่งมีค่า X - Min และ Y - Min และจุดที่สองมีค่า X - Max และ Y - Max หรือเป็นจุดมุมบนซ้าย (Upper Left) และมุมขวาล่าง (Lower Right) ของรูปสี่เหลี่ยมตามลำดับ จุดควบคุมดังกล่าวนี้จะเป็นตัวแปรค่าพิกัดให้อยู่ในรูปพิกัดทางภูมิศาสตร์ต่อไป

- ทำการลาก (Digitize) ขอบเขตลักษณะต่างๆ ในแผนที่ต้นแบบ ซึ่งลักษณะจุด เส้น หรือพื้นที่ นั้นจะถูกเปลี่ยนให้อยู่ในรูปของ Spatial Code และรูปร่างของการลากวาดจะถูกแสดงบนจอกราฟฟิค (Graphic) เพื่อหลีกเลี่ยงข้อผิดพลาดที่จะเกิดขึ้น

- ขั้นตอนสุดท้าย เป็นการปรับปรุงแก้ไขข้อมูลหรือข้อผิดพลาดต่าง ๆ ให้ถูกต้องก่อนการเก็บไว้ในคอมพิวเตอร์ต่อไป

รูปที่ 3.3 ลักษณะของเครื่องมือดิจิไทเซอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

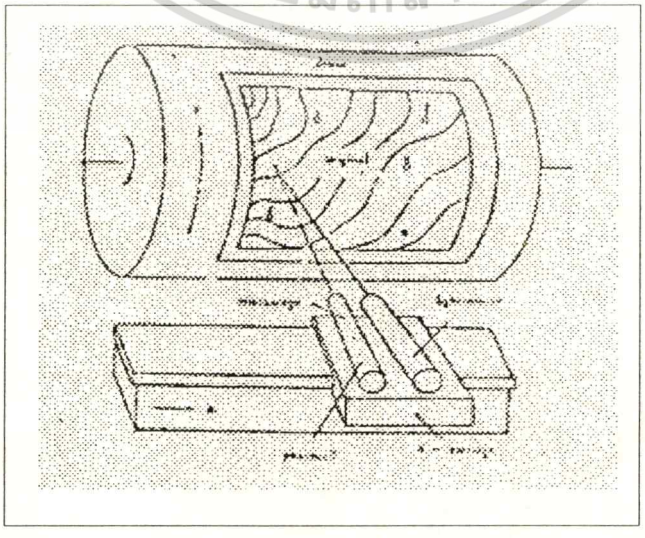
3. Automatic Capture การเก็บข้อมูลชนิดนี้เป็นการเปลี่ยนแปลงข้อมูล Spatial และ Attribute ซึ่งเป็นส่วนของ Analog Image ให้อยู่ในรูปของ Digital Image โดยอัตโนมัติ เครื่องที่ใช้ในการเก็บข้อมูลชนิดนี้แบ่งออกได้ดังนี้คือ

- Vedio Digitizer เป็นเครื่องมือที่ประกอบด้วยกล้องวีดีโอ ต่อเข้ากับ Framegrabber เช่น ไมโครอีเลคทรอนิค (Microelectronic) ซึ่งจะเปลี่ยนข้อมูลภาพที่เป็น Analog ให้อยู่ในรูปของ Digital แบบราสเตอร์ เครื่องมือนี้ส่วนใหญ่จะใช้เพื่อป้อนข้อมูลระยะไกล

- Charge - Coupled Device (CCD) ประกอบไปด้วยอุปกรณ์สารกึ่งตัวนำ (Semiconductor) ที่สามารถแปลงโฟตอน (Photon) ที่ตกลงบนพื้นผิวให้เป็นจำนวนของอิเลคตรอน (Electron) CCD จะบรรจุอยู่ในกล้องขนาด 35 มิลลิเมตรที่มีระบบเลนส์ธรรมดา โดยสามารถผลิตเครื่องมือที่สามารถรับแสงที่มีความไวแสงสูง ๆ ได้ การบันทึกข้อมูลลายเส้นต่าง ๆ กระทำโดยการกรองสัญญาณที่ต่ำหรือสูงกว่าระดับที่ต้องการออกไป

- Scanner เครื่องมือชนิดนี้ได้ถูกพัฒนาโดยชาวอังกฤษ เรียกว่า "Laser Scan" ซึ่งรวมเอา Scanner และ Plotter อันประกอบด้วยระบบออพติคัล (Optical) และ อิเลคโทรนิคเข้าด้วยกัน ทำการลากลายเส้นจากข้อมูลที่อยู่บนแผ่นใส หรือวางลายเส้นของข้อมูลลงไปบนไมโครฟิล์มไดอะโซ (Diazo Microfilm) ภาพของแผนที่ที่ต้องการบันทึกจะปรากฏบนจอภาพจากนั้นกำหนดจุดเริ่มต้นแล้วใช้ลำแสงเลเซอร์ลากลายเส้นไปทีละแถวจนครบทั้งภาพ ทำการเก็บข้อมูลที่ได้ไว้บนคอมพิวเตอร์ วิธีการนี้สามารถที่จะเก็บข้อมูลได้ถูกต้องในด้านของมาตราส่วน และสะดวกต่อการเก็บข้อมูล ที่มีลายเส้นจำนวนมาก เช่น เส้นระดับความสูง (Contour) เป็นต้น ตัวอย่างของเครื่องมือ Scanner ดังแสดงในรูปที่ 3.4

รูปที่ 3.4 ลักษณะของเครื่องมือสแกนเนอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ลักษณะโครงสร้างและการป้อนข้อมูลในระบบสารสนเทศภูมิศาสตร์ (GIS Structure and Data Input)

ลักษณะโครงสร้างของข้อมูลในระบบ GIS นั้น แบ่งออกได้เป็น 2 ลักษณะ คือ

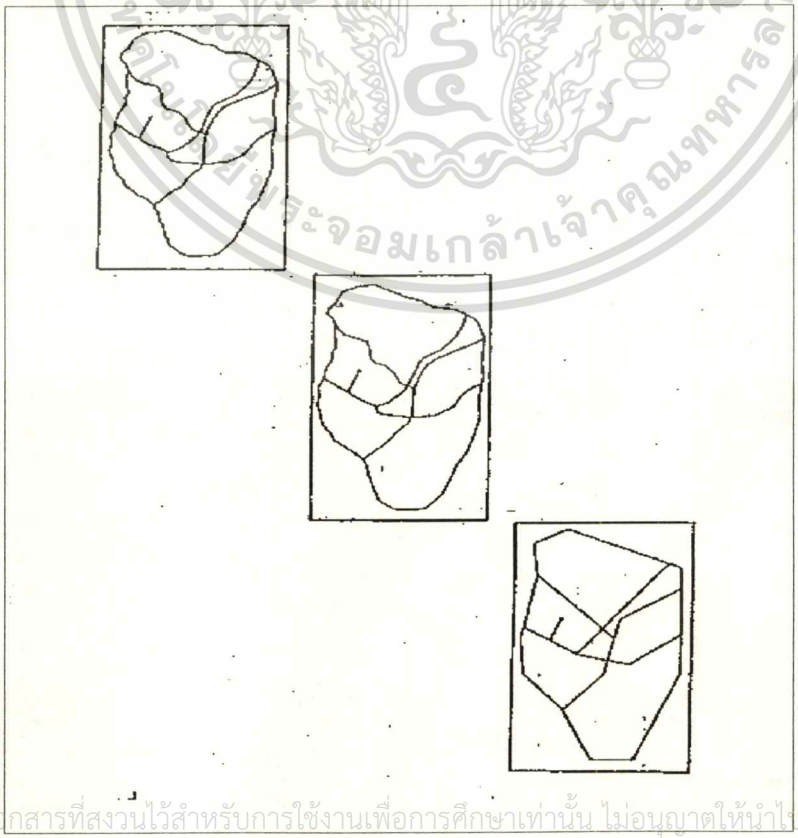
3.2.1 ลักษณะโครงสร้างแบบเวคเตอร์ (Vector Structure)

3.2.2 ลักษณะโครงสร้างแบบราสเตอร์ (Raster Structure)

3.2.1 ลักษณะโครงสร้างแบบเวคเตอร์ (Vector Structure)

ในข้อมูลระบบเวคเตอร์นั้น จะใช้ลักษณะของจุดและเส้น ในการแสดงลักษณะทางภูมิศาสตร์ โดยจุดที่เชื่อมโยงต่อกันด้วยเส้นตรงที่เรียกว่า " อาร์ค" (Arc) เป็นองค์ประกอบที่สำคัญของข้อมูลรูปแบบเส้น (Linear Feature) เช่น ถนน แม่น้ำ เป็นต้น ปลายของอาร์คหลาย ๆ อาร์คที่ต่อกันจนเกิดเป็นขอบเขตนั้นเรียกว่า "โพลีกอน" (Polygon) ขบวนการของข้อมูลแบบเวคเตอร์นี้จะใช้คู่ของพิกัด X และ Y เป็นตัวชี้ตำแหน่ง และลักษณะของสิ่งต่าง ๆ แล้วผ่านขบวนการที่เรียกว่า "Generialization" เพื่อให้ได้รูปร่างลักษณะ มาตราส่วน และรายละเอียดตามต้องการ ดังรูปที่ 3.5

รูปที่ 3.5 ขบวนการ Generialization ในระดับต่าง ๆ

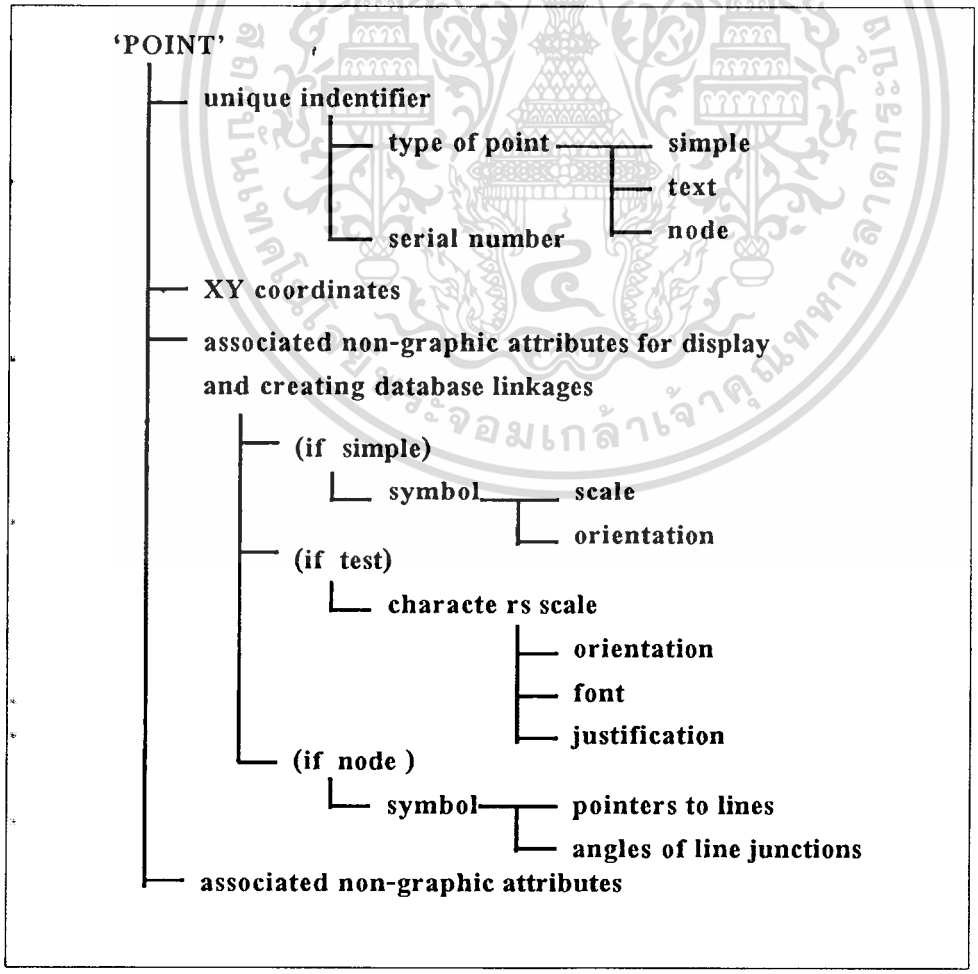


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการป้อนข้อมูลของระบบ GIS ในลักษณะโครงสร้างแบบเวคเตอร์ แบ่งออกเป็น วิธีการในรูปแบบต่าง ๆ ดังนี้ คือ

1. การป้อนข้อมูลที่เป็นจุด (Point Entities) การป้อนข้อมูลที่เป็นจุดจะใช้คู่พิกัด X และ Y เพื่อแสดงตำแหน่งของข้อมูลทางภูมิศาสตร์ หรือลักษณะของภาพต่าง ๆ นอกเหนือจากพิกัด X และ Y แล้ว ก็อาจจะระบุถึงข้อมูลอื่น ๆ ที่ใช้ในการอธิบายความหมาย หรือชนิดของข้อมูลที่เป็นจุดนั้น ๆ เช่น จุด อาจจะเป็นสัญลักษณ์ที่ไม่มีความสัมพันธ์กับข้อมูลอื่น การบันทึกข้อมูลจึงจำเป็นที่จะต้องระวังข้อมูลที่ใช้อธิบายความหมายของจุดและรวมถึงขนาดของข้อมูลจุดนั้น ๆ หรือถ้าจุดนั้นเป็นลักษณะของข้อมูลเกี่ยวกับรายละเอียดต่าง ๆ (Text Entity) การบันทึกข้อมูลจะต้องอธิบายถึงลักษณะที่จะใช้ในการแสดงผล รูปแบบ ตำแหน่ง และมาตราส่วนต่าง ๆ ดังรูปที่ 3.6

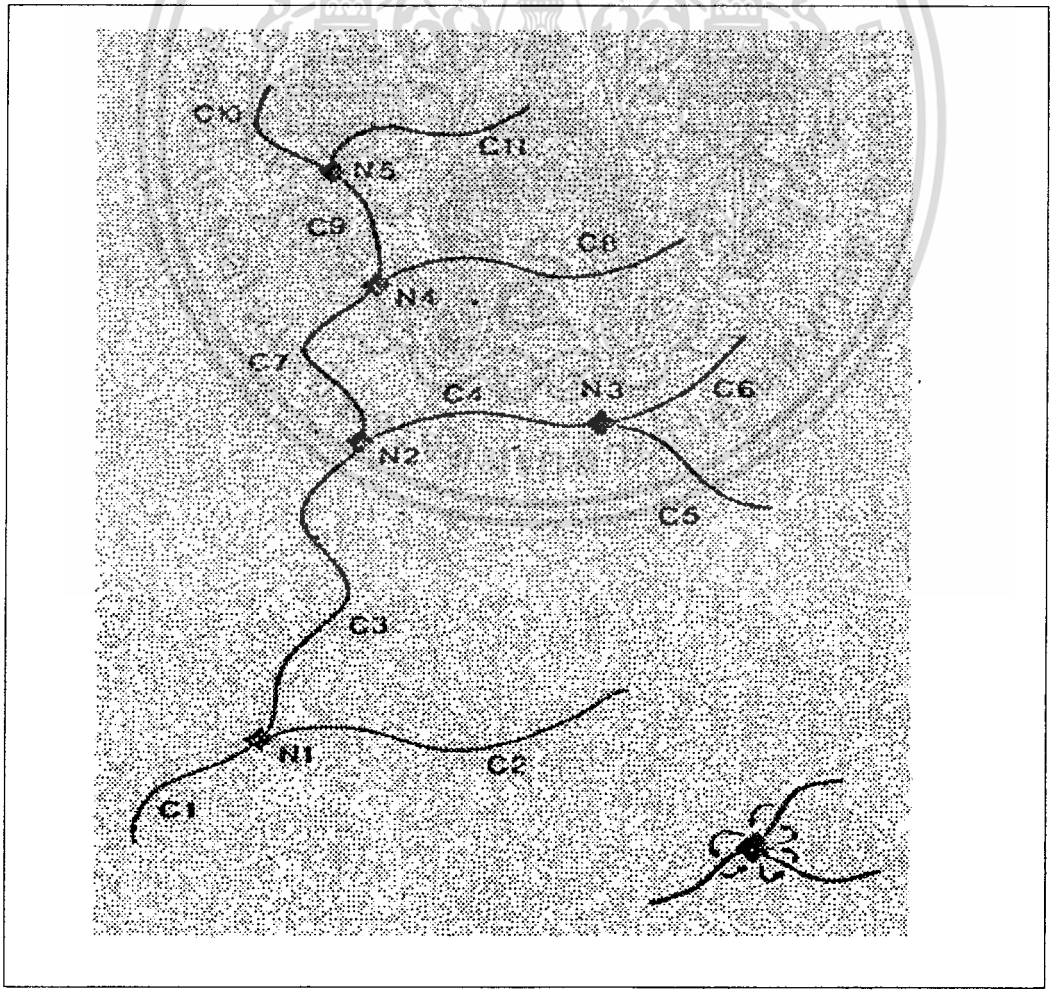
รูปที่ 3.6 ลักษณะการป้อนข้อมูลรูปแบบจุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การป้อนข้อมูลรูปแบบเส้น (Linear Entities) ลักษณะของข้อมูลรูปแบบเส้นนั้น สามารถแบ่งแยกได้ในลักษณะรูปแบบของเส้นที่เกิดจากการประกอบกันของเส้นตรงย่อย ๆ (Segment) ที่มีพิกัดตั้งแต่ 2 พิกัดขึ้นไป ลักษณะของเส้นจะถูกเก็บข้อมูลที่จุดเริ่มต้นและจุดปลายของเส้นเป็นอย่างน้อย รวมถึงข้อมูลที่ใช้อธิบาย หรือแสดงความหมายของสัญลักษณ์นั้น ๆ สำหรับเส้นที่มีลักษณะต่อเนื่องและซับซ้อน จะใช้ลักษณะของคู่พิกัดจำนวนมากในการใช้อธิบาย ซึ่งได้แก่ ลักษณะของอาร์ค และลักษณะลูกโซ่ (Chain or String) ในการป้อนข้อมูลที่เป็นโครงข่ายต่อเนื่อง (Connectivity Network) เช่น ระบบระบายน้ำ หรือระบบขนส่ง เป็นต้น จึงจำเป็นที่จะต้องสร้างตัวเชื่อมหรือตัวชี้ (Pointer) ในโครงสร้างของข้อมูลเพื่อเชื่อม (Chain) ในแขนงต่าง ๆ ซึ่งจะมีจุดที่เรียกว่า "Node" เป็นตัวช่วย โดยที่ Node จะบันทึกข้อมูลขนาดของมุมแต่ละ Chain ที่อยู่ร่วมในแต่ละ Node ดังรูปที่ 3.7

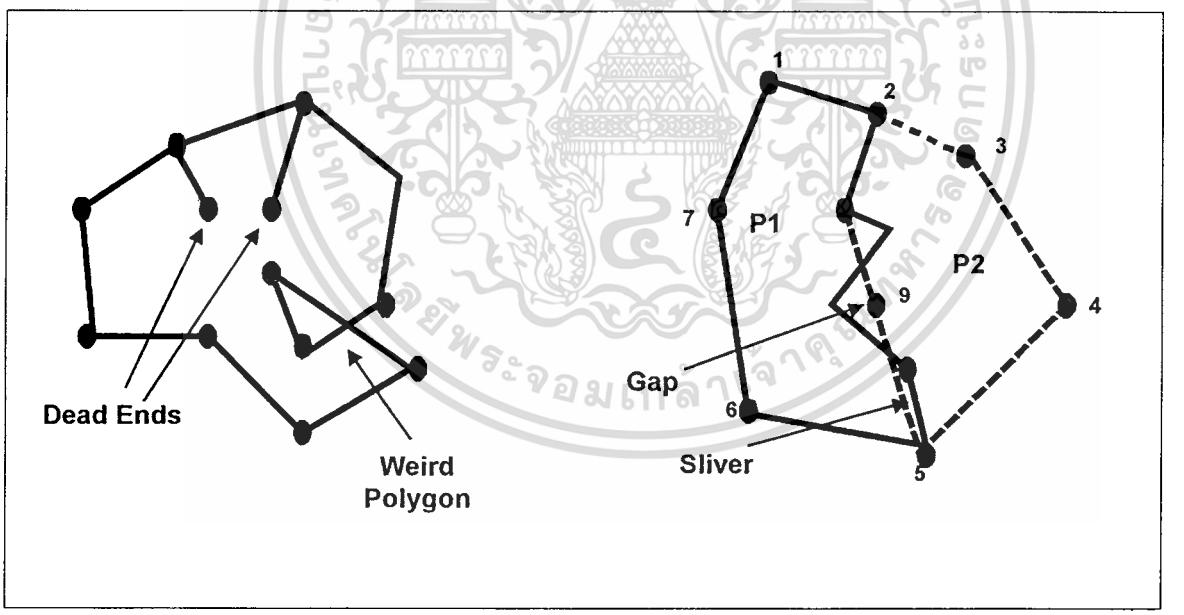
รูปที่ 3.7 ลักษณะการป้อนข้อมูลรูปแบบเส้นที่ใช้ Chain และ Node



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

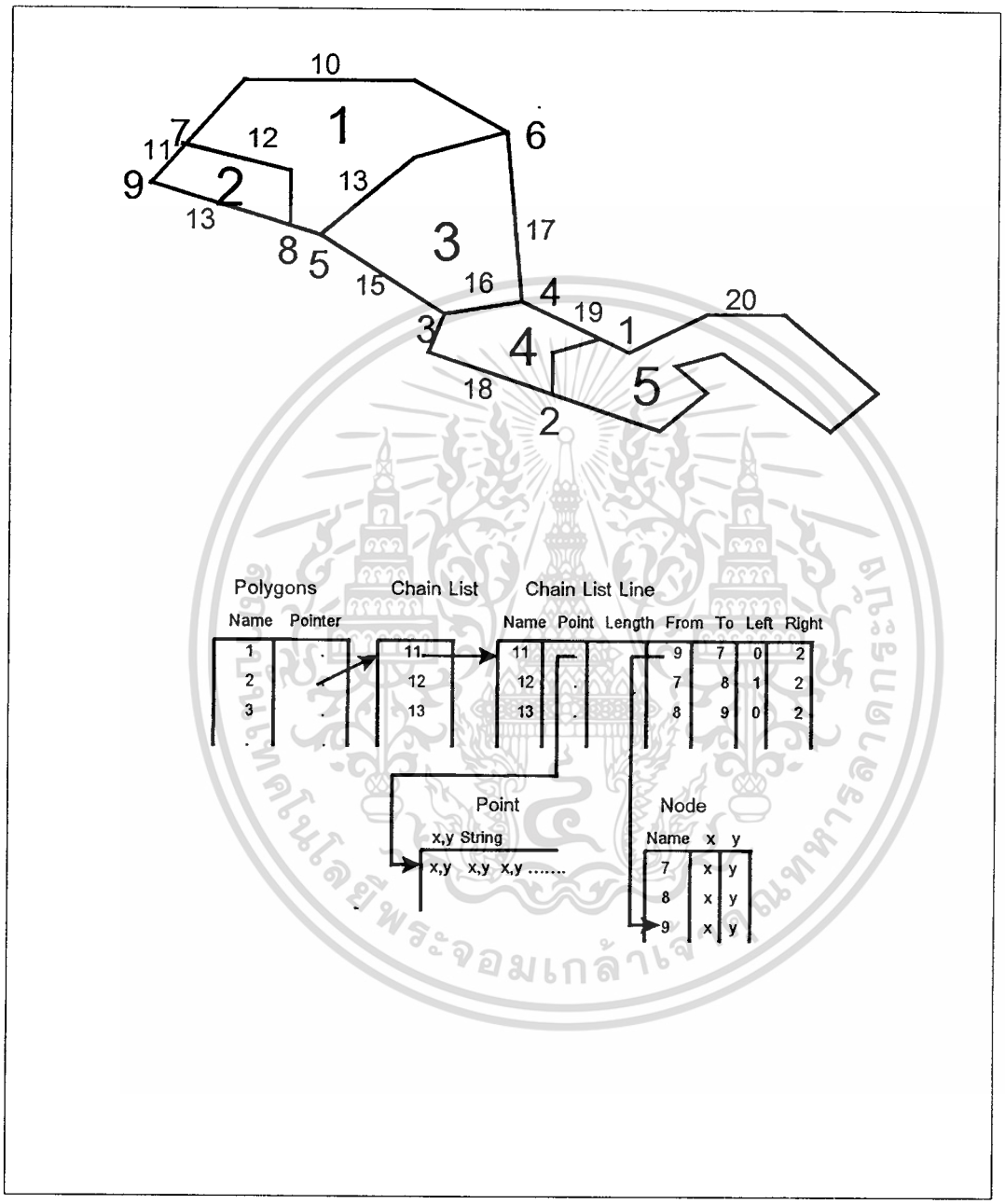
3. การป้อนข้อมูลรูปแบบพื้นที่ (Area Entities) การป้อนข้อมูลรูปแบบพื้นที่ในระบบ GIS นั้นจะพิจารณาในรูปของโพลีกอน เพื่อใช้อธิบายคุณสมบัติทางวิชาวิภาคเฉพาะส่วน (Topological Properties) ของพื้นที่ซึ่งได้แก่ รูปร่าง (Shape) ข้อมูลใกล้เคียง (Neighbour) และระดับชั้นต่าง ๆ (Hierarchy) ในลักษณะที่สามารถแสดงได้ด้วยรูปแบบโครงสร้างมาตราฐานเวคเตอร์ดังรูปที่ 3.9 สำหรับข้อมูลที่มีความซับซ้อนมากขึ้นก็จะใช้ลักษณะ Chain และ Node ในการกำหนดโครงสร้างของข้อมูล สำหรับข้อผิดพลาดที่อาจจะเกิดขึ้นในการเก็บข้อมูลของ Polygon คือ การลาก (Digitize) ขอบเขตที่ติดต่อกันของแต่ละโพลีกอน ซึ่งจำเป็นที่จะต้องทำการลากหรือเก็บข้อมูลซ้ำอาจก่อให้เกิดข้อผิดพลาดที่เรียกว่า "Gap" และ "Sliver" หรือ อาจจะลากขอบเขตของโพลีกอนได้ไม่ครบถ้วนที่เรียกว่า "Dead Ends" และการลากขอบเขตซ้อนตัดกันที่เรียกว่า "Weird Polygon" เป็นต้น ดังแสดงในรูปที่ 3.8 ดังนั้น จึงจำเป็นที่จะต้องใช้ความระมัดระวังในการเก็บข้อมูลพอสมควร เพื่อให้ได้ข้อมูลที่ถูกต้องมากที่สุดเท่าที่จะทำได้

รูปที่ 3.8 ข้อผิดพลาดในการป้อนข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 รูปแบบโครงสร้างข้อมูลมาตราฐานแบบเวคเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ลักษณะโครงสร้างแบบราสเตอร์ (Raster Structure)

ลักษณะโครงสร้างแบบราสเตอร์ จะประกอบด้วยลักษณะของช่องสี่เหลี่ยมที่เรียกว่า "กริด" (Grid Cells or Pixels) โดยส่วนใหญ่แล้วจะเป็นสี่เหลี่ยมจตุรัส หรือสี่เหลี่ยมผืนผ้า ขนาดของกริดขึ้นอยู่กับ ความต้องการของผู้ใช้ หรือข้อจำกัดอยู่ที่รายละเอียด (Resolution) ของข้อมูลนั้น เช่น ข้อมูลดาวเทียม Landsat MSS จะเก็บข้อมูลในลักษณะของราสเตอร์ที่มี Resolution 1 จุดภาพมีขนาด เท่ากับ 80 x 80 เมตร MOS - 1 เท่ากับ 50 x 50 เมตร Landsat TM เท่ากับ 30 x 30 เมตร SPOT XS เท่ากับ 20 x 20 เมตร และ SPOT Panchromatic เท่ากับ 10 x 10 เมตร เป็นต้น นอกจากนี้ขนาดของกริดยังขึ้นอยู่กับขนาดที่เหมาะสมของพื้นที่ที่ศึกษา และระบบที่จะใช้ประมวลผลอีกด้วย ในแต่ละกริดจะบรรจุตัวเลขซึ่งแทนค่า หรือชนิดของข้อมูลที่นำมาทำแผนที่โดยมีลักษณะของแถวแนวนอน (Row) และแถวแนวตั้ง (Column) เป็นตัวกำหนดตำแหน่งและทิศทาง ลักษณะของข้อมูลแบบจุดจะถูกแทนค่าด้วยกริดกริดเดียว ข้อมูลแบบเส้นจะแทนค่าด้วยจำนวนกริดที่อยู่ใกล้เคียงที่อยู่ต่อเนื่องกันตามทิศทางที่กำหนด และข้อมูลแบบพื้นที่ที่จะแทนค่าด้วยความสัมพันธ์และปริมาณการกระจายไปยังกริดใกล้เคียง ลักษณะโครงสร้างแบบราสเตอร์นี้จะง่ายต่อการใช้คอมพิวเตอร์ในการจัดเก็บ การคำนวณ และการแสดงผล

วิธีการป้อนข้อมูลในรูปแบบราสเตอร์ แบ่งออกได้เป็น 3 วิธีคือ

1. Chain Codes
2. Run - Lenght Codes
3. Block Codes

1. Chain Code เป็นการป้อนข้อมูลโดยการกำหนดทิศทาง และจำนวนกริดตามขอบเขตของข้อมูล ซึ่งทิศทางต่าง ๆ จะใช้ตัวเลขในการกำหนด เช่น ทิศตะวันออก = 0 ทิศเหนือ = 1 ทิศใต้ = 3 ทิศตะวันตก = 4 การป้อนข้อมูลจะกระทำในลักษณะตามเข็มนาฬิกา

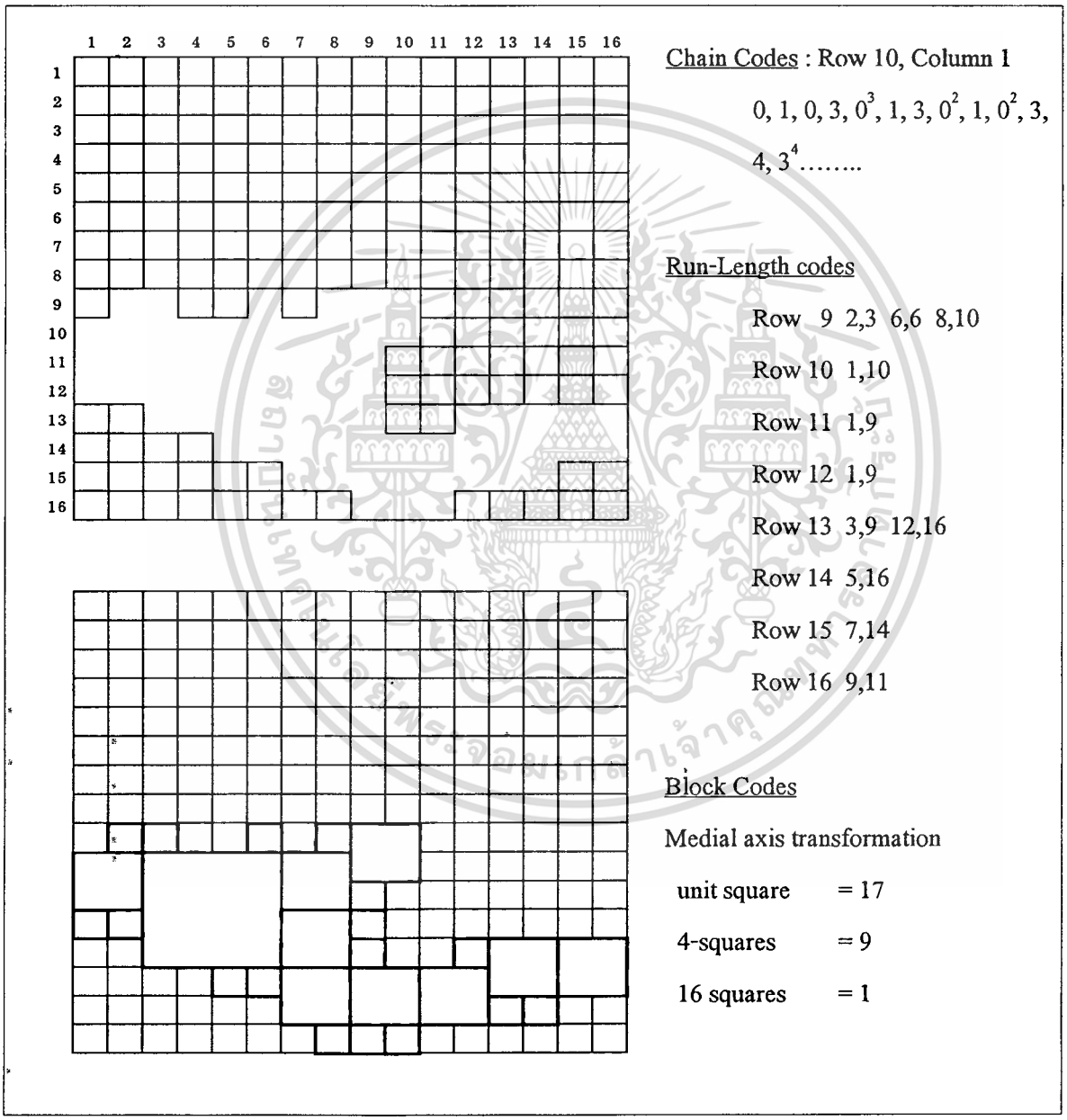
2. Run-Lenght Code เป็นการกำหนดข้อมูลโดยใช้แถวตามแนวนอน (Row) เป็นสิ่งอ้างอิง โดยจะกำหนดจำนวนกริดในแต่ละแถวที่ครอบคลุมขอบเขตของข้อมูล ในทิศทางจากซ้ายไปขวาของกริดจนถึงกริดสุดท้ายของข้อมูลในแต่ละแถวนั้น ๆ วิธีการนี้เหมาะสมที่จะใช้กับเครื่องคอมพิวเตอร์ขนาดเล็กที่มีเนื้อที่ในการเก็บข้อมูลจำกัด โดยสามารถลดจำนวนข้อมูลลงได้มากเมื่อต้องการเก็บไว้ในข้อมูลแบบราสเตอร์ แต่ในทางตรงข้ามก็อาจจำเป็นที่จะต้องเพิ่มความต้องการของขบวนการในการวิเคราะห์ข้อมูลและทำแผนที่ให้มากขึ้น

3. Block Code เป็นการใช้ช่องสี่เหลี่ยมจตุรัสเป็นตัวกำหนดในการป้อนข้อมูล โดยขนาดของช่องสี่เหลี่ยมจตุรัสแต่ละช่องจะประกอบด้วยจำนวนของกริดในลักษณะที่มีข้อจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จะประกอบกันเป็นช่องสี่เหลี่ยม (Block) ที่มีขนาดใหญ่ที่สุด นอกจากนี้ ยังขึ้นกับความเป็นไปได้ในการสร้าง Block ไปตามขอบเขตที่กำหนด เราเรียกลักษณะเช่นนี้ว่า Medial Axis Transformation (MAT) วิธีการนี้เหมาะสมสำหรับการประกอบกันขึ้นเป็นพื้นที่ ในลักษณะที่มีความต่อเนื่องตัวอย่างของวิธีการป้อนข้อมูลแบบราสเตอร์ ดังรูปที่ 3.10

รูปที่ 3.10 การป้อนข้อมูลแบบราสเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

3.3 การตรวจสอบและแก้ไขข้อมูล (Data Verification and Correction)

ข้อผิดพลาดในการป้อนข้อมูลนั้นอาจเกิดขึ้นในลักษณะต่าง ๆ เช่น

1. ข้อมูลเกิดความไม่สมบูรณ์เนื่องจากความบกพร่องในการป้อนข้อมูล เช่น ข้อมูลจุดเส้น หรือกริด ของการป้อนข้อมูลแบบ Semi - Automatic อาจเกิดการ Digitize เกินกว่า 1 ครั้ง ส่วนการป้อนข้อมูลโดยวิธี Scanner ข้อบกพร่องจะอยู่ในรูปของ Gap ระหว่างเส้นซึ่งเกิดจากขบวนการเปลี่ยนแปลงข้อมูลราสเตอร์และเวคเตอร์ที่ไม่สามารถต่อเชื่อมได้ในทุกส่วน

2. ข้อมูลอยู่ในตำแหน่งที่คลาดเคลื่อนจากความเป็นจริง เนื่องจากความไม่ระมัดระวังในการ Digitize ข้อมูล หรืออาจเกิดจากข้อผิดพลาดของฮาร์ดแวร์และซอฟท์แวร์โดยตรง นอกจากนี้มาตราส่วนของข้อมูลอาจเกิดความผิดพลาดติดตามมาได้อีกด้วย

3. ข้อมูลอาจเกิดการบิดเบน (Distortion) เนื่องจากแผนที่ต้นแบบที่ใช้ในการ Digitize มีมาตราส่วนที่ไม่ถูกต้อง เช่น ภาพถ่ายทางอากาศส่วนมากจะมีมาตราส่วนที่ไม่ถูกต้อง เนื่องจากการเอียงของเครื่องบิน สภาพภูมิประเทศที่แตกต่างกัน และความแตกต่างของระยะทางจากเลนส์ถึงส่วนต่าง ๆ ของพื้นที่ นอกจากนี้การบิดเบนอาจเกิดจากการยืดหดตัวของกระดาษที่ใช้ทำแผนที่อีกด้วย

4. ข้อมูลไม่สมบูรณ์เนื่องจากการพิมพ์และการป้อนข้อมูลเกิดการผิดพลาด วิธีการตรวจสอบความผิดพลาดของข้อมูลที่ดีที่สุดก็คือ การให้คอมพิวเตอร์แสดงข้อมูลที่ Digitize ไว้ออกมาตรวจสอบอีกครั้ง โดยอาจจะพิมพ์ลงบนแผ่นใสในมาตราส่วนเดียวกันกับแผนที่ต้นแบบ แล้วนำแผนที่ทั้งสองมาซ้อนทับกันบนโต๊ะแสงเพื่อเปรียบเทียบและหาข้อผิดพลาดที่เกิดขึ้น สำหรับข้อมูลที่เป็น Non - Spatial จะตรวจสอบโดยให้คอมพิวเตอร์พิมพ์ข้อมูลออกมาแล้วตรวจสอบความถูกต้องในแต่ละคอลัมน์ของไฟล์ (File) ต่าง ๆ การตรวจสอบและแก้ไขข้อมูลให้ถูกต้องนั้นจำเป็นที่จะต้องใช้ระยะเวลามากพอสมควร ซึ่งบางครั้งอาจใช้เวลามากกว่าการป้อนข้อมูลเสียอีก การตรวจสอบและแก้ไขข้อมูลนั้นนอกจากกระทำเพื่อการแก้ไขข้อผิดพลาดที่เกิดจากการป้อนข้อมูลแล้ว ยังจำเป็นที่จะต้องตรวจสอบข้อมูลให้มีความทันสมัยอยู่ตลอดเวลา เนื่องจากข้อมูลบางชนิดมักเกิดการเปลี่ยนแปลงอยู่เสมอ เช่น ข้อมูลการใช้ที่ดิน มักเปลี่ยนแปลงเนื่องจากการดัดแปลงสภาพธรรมชาติและการพัฒนาต่าง ๆ เป็นต้น แต่ข้อมูลบางชนิดก็ไม่จำเป็นที่จะต้องแก้ไขมากนัก เนื่องจากข้อมูลมีการเปลี่ยนแปลงเป็นไปอย่างช้า ๆ เช่น ข้อมูลด้านธรณีวิทยา เป็นต้น

การสร้างฐานข้อมูลดิจิตอลนั้นจำเป็นที่ จะต้องใช้ค่าใช้จ่ายและระยะเวลาค่อนข้างมาก การเก็บรวบรวมข้อมูลจึงต้องมีความปลอดภัย เพื่อให้สามารถใช้ประโยชน์ข้อมูลได้อย่างเต็มที่และยาวนาน การเก็บข้อมูลดิจิตอลนั้นส่วนใหญ่จะเก็บไว้ในแผ่นหรือเทปแม่เหล็ก โดยแผ่นแม่เหล็กจะมีขนาด 5.25 นิ้ว หรือ 3.5 นิ้ว ที่มีความหนาแน่นสูง (High Density) ส่วนเทปแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเป็นเทป 9 แทร็ก (Tracks) ที่มีความหนาแน่น 800, 1600 หรือ 6250 BPI (Bits Per Inch) เป็นต้น นอกจากนี้ในปัจจุบันเทคโนโลยีเกี่ยวกับวิดีโอดิสต์ ซึ่งใช้ในการบันทึกดนตรี หรือสัญญาณวิดีโอบนคอมแพคดิสต์ (Compact Disks) นั้น สามารถพัฒนาหน่วยความจำแบบอ่านอย่างเดียว (Read - Only Memories, ROMs) สำหรับคอมพิวเตอร์ได้ โดย CD-ROM (Compact Disk Read - Only Memory) ขนาดเส้นผ่าศูนย์กลาง 12 นิ้ว สามารถเก็บข้อมูลได้ถึง 600 เมกกะไบต์ ดังนั้นข้อมูลในระบบสารสนเทศภูมิศาสตร์สามารถเรียกมาใช้และวิเคราะห์ได้อย่างรวดเร็วมากขึ้น

3.4 ข้อดีและข้อเสียของการเก็บข้อมูลลักษณะโครงสร้างราสเตอร์และเวคเตอร์

ตารางที่ 3.1 แสดงการเปรียบเทียบคุณสมบัติของโครงสร้างข้อมูลแบบเวคเตอร์และราสเตอร์

คุณลักษณะ	โครงสร้างแบบเวคเตอร์	โครงสร้างแบบราสเตอร์
ความง่ายในการเรียนรู้	มีโครงสร้างที่สลับซับซ้อนและยุ่งยากต่อการเรียนรู้	มีโครงสร้างที่ง่ายต่อการเรียนรู้
ความชัดเจนของตำแหน่ง	สามารถระบุตำแหน่งได้แน่นอนชัดเจน	ความชัดเจนของตำแหน่งจะเกิดขึ้นได้จะต้องมีกระบวนการประมวลผลข้อมูล ซึ่งจะต้องใช้เวลาโดยความละเอียดชัดเจนจะถูกจำกัดด้วยจำนวน Pixel ในภาพ
ความชัดเจนของภาพ	จะให้ความชัดเจนในส่วนของภาพที่เป็นจุด เส้นและรูปเหลี่ยมพื้นที่	จะให้ความชัดเจนดี ถ้าเป็นภาพที่แสดงความเป็นจริงของวัตถุจริง เช่น ภาพคน ภาพวิว เพราะแต่ละ Pixel ในภาพที่กระจายอยู่จะมีระดับความเข้มของแต่ละสีที่แตกต่างกัน
ความสามารถในการนำข้อมูลภาพมาซ้อนทับกันเพื่อประโยชน์ในการแสดงผล และวิเคราห์	สามารถทับซ้อนข้อมูลของภาพได้ แต่ถ้าซ้อนหลายชั้นมากเกินไป จะทำให้ยากแก่การแยกแยะข้อมูล	เนื่องจากข้อมูลมีลักษณะการเก็บเป็น Pixel ตามจำนวนของขนาดความละเอียดของภาพ ดังนั้นการซ้อนทับกันหลาย ๆ ชั้นจึงทำให้ไม่ก่อให้เกิดปัญหา เพราะมีข้อมูลทุก Pixel ในการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงการเปรียบเทียบคุณสมบัติของโครงสร้างข้อมูลแบบเวคเตอร์และราสเตอร์(ต่อ)

คุณลักษณะ	โครงสร้างแบบเวคเตอร์	โครงสร้างแบบราสเตอร์
ความง่ายในการวิเคราะห์ข้อมูล	ในการสอบถามข้อมูลเชิงระยะทาง สามารถทำได้รวดเร็ว แต่การวิเคราะห์ข้อมูลมีข้อจำกัดในเรื่องการซ้อนทับของจุดกับเส้น	ในการสอบถามข้อมูลเชิงระยะทางจะช้ากว่าแบบเวคเตอร์ แต่ในการวิเคราะห์ข้อมูลจะทำได้ดีว่าเพราะมีกระบวนการประมวลผลข้อมูลที่ไม่ยุ่งยากนัก เช่น ใช้หลักการตัวกรองข้อมูล (Filter) และมีโมเดลในการประมวลผลข้อมูลมากมาย
โครงสร้างข้อมูล	สลับซับซ้อน	ไม่สลับซับซ้อน
ความต้องการพื้นที่ในการเก็บข้อมูล	ความต้องการพื้นที่ขึ้นอยู่กับจำนวนจุด เส้น และรูปเหลี่ยมที่มีอยู่ภายในภาพ	ความต้องการพื้นที่ในการเก็บขึ้นอยู่กับความละเอียดของภาพ
ความเหมือนจริงของภาพ	จะมีความเหมือนจริงน้อย เนื่องจากการแสดงผลภาพจะมีลักษณะเป็นจุด เส้น หรือรูปเหลี่ยม	จะมีความเหมือนจริงมากเพราะมีระดับความเข้มของสีในแต่ละจุดของภาพที่แสดงถึงความเหมือนจริง
ความสามารถในการนำข้อมูลมาแปลงเป็นลักษณะแบบโครงข่าย	สามารถนำมาใช้ในรูปแบบของโครงข่ายได้ดี กว่าเพราะโครงสร้างมีการเก็บแบบให้ความสัมพันธ์กันระหว่างจุด เส้น รูปเหลี่ยม	ยากในการนำมาใช้ในรูปแบบของโครงข่าย เพราะการเก็บมีลักษณะเป็น Pixel ซึ่งแต่ละ Pixel ไม่ได้แสดงถึงความสัมพันธ์กัน
ค่าใช้จ่ายในการเก็บข้อมูล	การเก็บหรือแสดงผลข้อมูลใช้เวลานาน และอุปกรณ์ในการแสดงผลหรือนำเข้าก็มีราคาแพงด้วย	อุปกรณ์ในการนำเข้าและแสดงผลสามารถหาซื้อได้ง่ายและราคาไม่แพง
การปรับปรุงและแก้ไขข้อมูล	สามารถทำได้ง่ายและรวดเร็ว	จะต้องใช้เวลามากและแก้ไขได้ยาก เพราะการแก้ไขจะต้องทำที่ละ Pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางการเปรียบเทียบคุณสมบัติโครงสร้างข้อมูลแบบราสเตอร์และเวคเตอร์ (ตารางที่ 3.1) แสดงถึงข้อดีและข้อเสียของโครงสร้างทั้ง 2 ประเภท โดยถ้าต้องการข้อมูลที่ได้ภาพที่เหมือนจริงและง่ายในการเก็บก็ควรใช้แบบราสเตอร์ แต่ถ้ามีพื้นที่ในการเก็บข้อมูลน้อยและลักษณะของข้อมูลเป็นลายเส้นก็ควรเก็บแบบเวคเตอร์ ซึ่งโดยทั่วไปความต้องการของการใช้งานข้อมูลในการวิเคราะห์และแสดงผลนั้นจะมีความต้องการเท่า ๆ กัน และถ้าหากสามารถแปลงข้อมูลระหว่างรูปแบบราสเตอร์เป็นแบบเวคเตอร์ได้ หรือแปลงจากแบบเวคเตอร์เป็นแบบราสเตอร์ได้แล้วนั้น ก็จะสามารถทำให้เกิดการประหยัดเวลาในการประมวลผลข้อมูลภาพ พื้นที่ในเก็บข้อมูล และลดค่าใช้จ่ายเกี่ยวกับอุปกรณ์ในการประมวลผล ทั้งผู้ใช้งานก็สามารถเลือกรูปแบบข้อมูลตามความเหมาะสมแก่การใช้งานโดยไม่ต้องคำนึงถึงเงื่อนไขหรือข้อจำกัดในแง่ของ Software และ Hardware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบขั้นตอนและวิธีการในการแปลงข้อมูลราสเตอร์เป็นเวคเตอร์

4.1 ขั้นตอนในการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์

ในการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์นั้นมีขั้นตอนที่เกี่ยวข้อง 9 ขั้นตอน ดังนี้

1. Scanning A การใช้อุปกรณ์คอมพิวเตอร์ในการเก็บข้อมูลภาพเป็นแบบราสเตอร์เข้าเครื่องคอมพิวเตอร์

2. Noise Removal F การขจัดข้อมูลที่รบกวนภาพ

3. Skeletonizing B การหาโครงร่างของภาพที่บาง

4. Node Improvement G การปรับปรุงและหาจุดต่อเชื่อมเพื่อให้เกิดความชัดเจนของโคร่งร่างภาพ

5. Line Tracking C การหาส่วนของเส้น

6. Segment Merge D การหาส่วนประกอบของพื้นที่

7. Topological Reconstruction E การจัดความสัมพันธ์ของส่วนประกอบของภาพ

8. Complex Object Construction And Complex Object Feature Coding H กำหนดวัตถุในภาพที่ซับซ้อน

9. Arc Attribute Feature Coding I การหามุม ลักษณะและทิศทางของเส้น

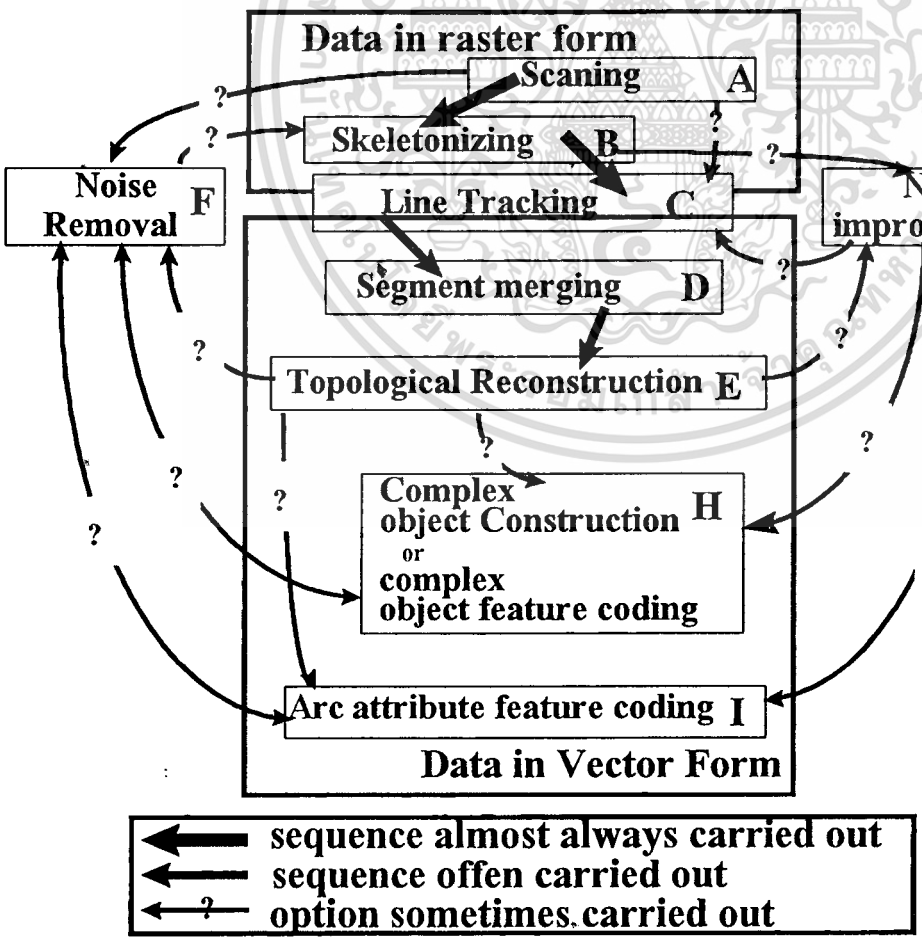
โดยมีความสัมพันธ์ของแต่ละขั้นตอนดังรูปที่ 4.1 ซึ่งจะเห็นได้ว่าการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์ไม่จำเป็นที่จะต้องทำทั้ง 9 ขั้นตอนในบางขั้นตอนอาจไม่จำเป็นต้องทำ เช่นหากภาพที่เราต้องการแปลงเป็นภาพที่ไม่ซับซ้อน ก็ไม่จำเป็นต้องทำการกำหนดวัตถุของภาพที่ซับซ้อน ซึ่งขั้นตอนในการแปลงจะไม่ได้เป็นขั้นตอนที่มาตราฐาน โดยขั้นตอนใดจำเป็นที่จะต้องทำมากน้อยเพียงใดขึ้นอยู่กับภาพต้นแบบที่นำมาแปลงว่ามีความสลับซับซ้อน หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ชัดเจนมากน้อยเพียงใด ดังนั้นในการแปลงข้อมูลสามารถสรุปถึงขั้นตอนที่จำเป็นและไม่จำเป็นได้ดังนี้

- ขั้นตอนที่มักจะต้องทำเสมอมีดังนี้ A B C D หรือ A B C D E
- ขั้นตอน B อาจไม่ต้องทำถ้าภาพต้นแบบมีลักษณะลายเส้นที่ชัดเจนอยู่แล้ว ซึ่งจะทำให้เหลือเพียงขั้นตอน A C D หรือ A C D E
- ขั้นตอน C F G จะสามารถทำได้ทั้งกับรูปแบบข้อมูลที่เป็นราสเตอร์และเวคเตอร์
- ขั้นตอน E F G หากต้องการทำจะต้องคำนึงถึงค่าใช้จ่ายและเวลาในการประมวลผล แต่ถ้าทำจะทำให้ผลลัพธ์ของข้อมูล แบบเวคเตอร์ดูใกล้เคียงกับภาพต้นแบบและชัดเจนมากยิ่งขึ้น
- ขั้นตอน H I จะทำเมื่อภาพมีความซับซ้อนมาก ๆ

รูปที่ 4.1 แสดงความสัมพันธ์ของขั้นตอนในการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์ [4]



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบขั้นตอนและพัฒนาการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ

การออกแบบขั้นตอนแปลงข้อมูลภาพในที่นี้กำหนดให้ภาพที่ผ่านเครื่องสแกนเนอร์มีความคมชัดและมีกลุ่มสีที่ชัดเจน ดังนั้นภาพที่ได้จึงไม่มีความจำเป็นต้องกระทำ Preprocessing โดยขั้นตอนที่ออกแบบมีดังนี้

- ขั้นตอนที่ 1 **Scaning A** การเก็บข้อมูลภาพสีขนาด A4 ผ่านเครื่องสแกนเนอร์ เป็นรูปแบบ .PCX ไฟล์
- ขั้นตอนที่ 2 **Skeletonizing B** การหาโครงร่างของภาพที่บางจะประกอบไปด้วย 2 ขั้นตอน คือ ขั้นตอนที่ 2.1 Edge Detection[2] และ 2.2 Thinning Edge[2]
- ขั้นตอนที่ 3 **Node Improvement G** การหาจุดรวมของเส้นที่ชัดเจน
- ขั้นตอนที่ 4 **Line Tracking C** การหาส่วนของเส้น
- ขั้นตอนที่ 5 **Segment Merge D** การหาส่วนประกอบของพื้นที่รูปเหลี่ยม

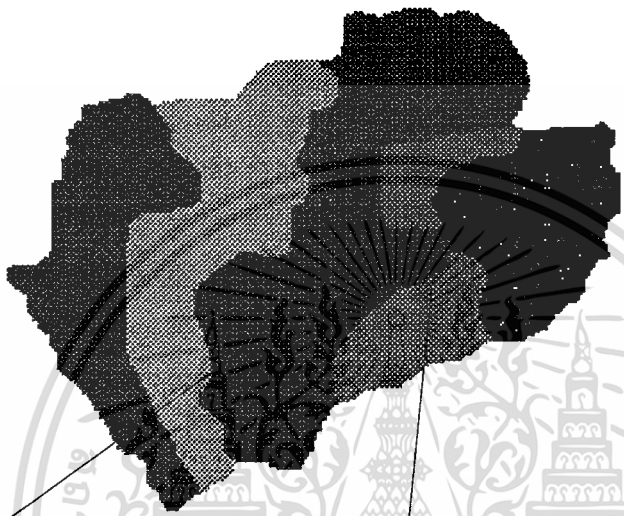
4.2.1 ขั้นตอนที่ 1 Scaning

การสแกนภาพ (Scaning) จะนำภาพต้นแบบ [ภาคผนวก ก] เป็นภาพแผนที่ประเทศไทย มาทำการเก็บข้อมูลภาพแบบราสเตอร์ โดยผ่านอุปกรณ์คอมพิวเตอร์ และทำการตัดตกแต่งโดยใช้ Paint-Brush เพื่อให้ได้ภาพแผนที่ภาคเหนือ [ภาคผนวก ข] และภาคตะวันออกเฉียงเหนือ [ภาคผนวก ค] ที่มีโทนสีที่ชัดเจน ซึ่งจะได้ตัวอย่างข้อมูลภาพที่ตรงตามเงื่อนไขในการทดลองแปลงแบบอัตโนมัติ และได้ข้อมูลราสเตอร์ไฟล์เป็น .PCX ดังรูปที่ 4.2 โดยไฟล์ .PCX จะประกอบด้วย 3 ส่วนดังนี้

- Header 128 Byte
- Bit Map Data (Run-Length Endcoding)
- 256-Color Palette

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 ตัวอย่างภาพและข้อมูลราสเตอร์ที่ได้จากการนำภาพผ่านเครื่องสแกนเนอร์ ซึ่งมีลักษณะเป็นภาพสี และมีการแบ่งแยกโทนสีที่ชัดเจน



Bitmap

x/y	200	201	202	203	204	205	206	207	208	209	210
100	005	005	005	005	005	005	005	005	005	005	005
101	005	005	005	005	005	005	005	005	005	005	005
102	018	018	018	018	005	005	005	005	005	005	005
103	018	018	018	018	005	005	005	005	005	005	005
104	018	018	018	018	018	018	018	005	005	005	005
105	018	018	018	018	018	018	018	018	005	005	005
106	018	018	018	018	018	018	018	018	018	018	018
107	018	018	018	018	018	018	018	018	018	018	018
108	018	018	018	018	018	018	018	018	018	018	018
109	018	018	018	018	018	018	018	018	018	018	018
110	018	018	018	018	018	018	018	018	018	018	018
111	018	018	018	018	018	018	018	018	018	018	018
112	018	018	018	018	018	018	018	018	018	018	018
113	018	018	018	018	018	018	018	018	018	018	018
114	018	018	018	018	018	018	018	018	018	018	018
115	018	018	018	018	018	018	018	018	018	018	018

Color Code

Index	Red	Green	Blue
001	005	010	050
002	030	010	050
003	070	035	045
004	050	078	050
005	002	010	050
006	100	110	099
007	079	210	057
008	110	255	150
009	105	110	050
010	055	210	159
011	066	010	050
012	089	120	050
013	075	130	050
014	095	155	050
015	130	010	050
016	180	015	077
017	195	077	055
018	200	155	255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Header 128 Byte เป็นส่วนหัวของไฟล์ ซึ่งจะเก็บข้อมูลต่าง ๆ เพื่อนำไปใช้ในการ Encoder Data ของข้อมูลภาพ จะมีส่วนประกอบดังตารางที่ 4.1

ตารางที่ 4.1 รายละเอียดข้อมูล Header File 128 Byte

ไบต์ที่	ขนาด	ชื่อข้อมูล	รายละเอียด
0	1	Password	ค่า 'Oah' = ไฟล์ .PCX
1	1	Version	เก็บค่า Version (รุ่น) ของ PC Paint-Bush ดังนี้ 0= Version 2.5 2=Version 2.8 With Palette 3=Version 2.8 Without Palette 4=PC Paintbrush for windows 5= Version 3.0
2	1	Encoding	ค่า=1
3	1	Bits per Pixel	จำนวนของบิตที่ใช้แสดง 1=Pixel จาก 1 Plan (ระนาบสี) 1=EGA,VGA,HERC 4=CGA
4	8	Window-Dimensions	ค่า Integer 4 ค่า (ค่าละ 2 Byte) ให้ค่ามุมบนซ้ายและมุมล่างขวาของภาพ กำหนดให้รูปแบบ X1,Y1,X2,Y2
12	2	Horizontal-Resolution	Dots/inch in X,when printed
14	2	Vertical Resolution	Dots/inch in Y,wher printed
16	48	Header palette	ข้อมูล Color Palette
64	1	Reserved	
65	1	No of Plan	จำนวนของ Planes ที่ใช้แสดงผล
66	1	Byte per Line	จำนวนไบต์ต่อการสแกน 1 บรรทัด
68	2	Header palette interpret	
70	58	Blank to end	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BitMap Data (Run-Length Encoding) จะมีการ Compress Data แบบ Run-Length โดยในที่นี้จะกำหนดให้ จำนวนไบท์ที่เหมือนกันก็จะ Compress โดยจะ Compress ได้มากที่สุด 64 ไบท์ ซึ่งเป็น 6-bit Run-Length Count

Algorithm Decoding สำหรับ 8 bit/pixel/plane

```
Read XMin, YMin, XMax, YMax  from header, byte 4-11;
Computer image size(Number of Pixel) from Header data
        (XMax - XMin+1) * (YMax - YMin+1);
Loop Until Byte_Pixel = Number_of_pixel
        Read bit map data byte from file byte N;
                If  byte N> "Coh"
        Count = byte N XOR "Coh";
        Read bit Map data byte from file byte N+1;
        Loop Until Decode = Count
                Byte_pixel = Byte_Pixel + 1;
        End;
Else  Byte_pixel = Byte_pixel + 1;
End;
```

256-Color Palette ในกรณีที่เป็นโครงสร้างไฟล์ .PCX ที่แสดงผล 256 Color Palette จะมี Index Color อยู่ท้ายไฟล์ จำนวน 768 ไบท์ และจากท้ายไฟล์ขึ้นมาในตำแหน่งที่ 769 จะมีค่า "OC" ซึ่งเป็นตัวเช็คว่าข้อมูลถัดจากนี้ไปจะเป็น 256-Color Palette โดยแต่ละ Color Palette จะประกอบด้วย 3 สี คือ Red, Green, Blue สีละ 1 ไบท์ (768=256x3) ซึ่ง 768 ไบท์จะเรียงเป็น Color Map สำหรับใช้ในการ Set Block Color Register ที่ใช้เป็น Color Map สำหรับแสดงภาพ โดยนำค่า Bit Map Data ที่ Decode แล้ว มาเป็น Index ชี้ตำแหน่งสีที่ต้องการ โดย Index จะต้องทำให้มีค่าตั้งแต่ 0-63 โดยการหาร 4 เนื่องจากข้อมูลในไฟล์เป็น 1 ไบท์ ต่อ 1 สี แต่ใน VGA Color Register Card 1 สี จะแสดงโดยใช้ 6 บิต เท่านั้น

โดยในการเก็บข้อมูลภาพผ่านอุปกรณ์คอมพิวเตอร์ (Scaning) จะได้ข้อมูลภาพแบบราสเตอร์ ณ ระดับความเข้มของสีที่ประกอบด้วยสีแดง เขียว น้ำเงิน ดังตัวอย่างในรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นก็ทำการเลือกค่าความเข้มมาทีละสี พร้อมกับเลือกความเข้มที่เป็นขาวดำ (Gray Level) มาทำการประมวลผล ในขั้นตอนที่ 2 และเปรียบเทียบความสมบูรณ์และชัดเจนของขอบภาพ

4.2.2 ขั้นตอนที่ 2 Skeletonizing

การหาโครงร่างของภาพที่บาง (Skeletonizing) จะประกอบไปด้วย 2 ขั้นตอนด้วยกัน ดังนี้

- **ขั้นตอนที่ 2.1 การหาขอบภาพ** (Edge Detection) ตามทฤษฎีในการประมวลผลหาขอบภาพนั้นมีหลายวิธีด้วยกัน เช่น Prewitt Edge[2], Sobel Edge[2], Kirsh Edge[2], Laplacian Edge [2] และอื่น ๆ อีกมากมาย แต่ในการทดลองนี้จะทำการหาขอบภาพด้วยวิธี Sobel Edge[2] และวิธี Range Edge [2]

Sobel Edge เป็นการหาขอบภาพโดยใช้ตารางตัวกรอง (Mark Operator) ที่มีผลต่อทิศทางในแนวนอนและแนวตั้งดังสมการที่ (1) และ (2) ดังนี้

$$Sx = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & -2 & -1 \end{bmatrix}$$

$$Sy = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$ ..... ข้อมูลความเข้มที่สุ่มมาทีละ 3x3

$$Gx = Sx * I = a+2b+c-g-2h-I \quad \text{..... (1)}$$

$$Gy = Sy * I = a+2d+g-c-2f-I \quad \text{..... (2)}$$

$$A(x,y) = |Gx| + |Gy|$$

I แทนข้อมูลความเข้มของแสงที่สุ่มมาทีละ 3x3

Sx, Sy แทน Mark Operator ในแนวตั้งและแนวนอน

A(x,y) แทนขอบภาพที่เกิดขึ้น

Range Edge เป็นการหาขอบภาพโดยใช้หลักการหาค่าความเข้มที่เหมาะสมในแต่ละจุดของภาพ โดยจะทำการหาผลต่างของค่าความเข้มที่สูงสุดกับความเข้มที่ต่ำ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดของจุดบริเวณรอบ ๆ และจะรอบกว้างเพียงใดขึ้นอยู่กับค่าตัดสินใจ Threshold และผลต่างที่คำนวณได้จะนำมาเป็นค่าความเข้มใหม่ของจุดนั้น ๆ โดยมีสูตรในการคำนวณดังสมการที่ (3)

$$w(k,l)= Max_A\{F(k,l)\} - Min_A\{F(k,l)\} \quad .....(3)$$

k = Row l = Column

$Max_A\{F(k,l)\}$ = ค่าสูงสุดของความเข้มบริเวณจุดรอบ ๆ ที่สนใจ

$Min_A\{F(k,l)\}$ = ค่าต่ำสุดของความเข้มบริเวณจุดรอบ ๆ ที่สนใจ

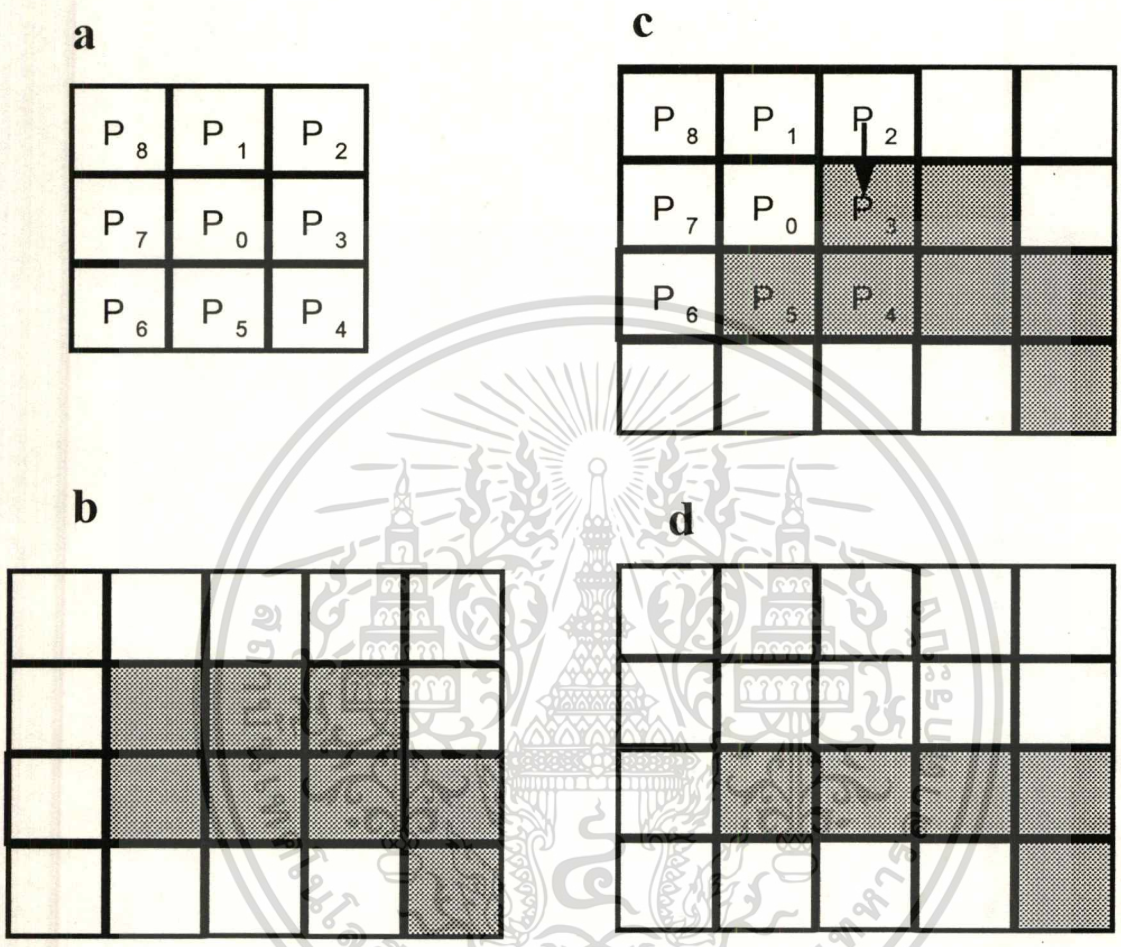
$w(k,l)$ = ค่าระดับความเข้มใหม่แสดงขอบภาพ

หลังจากได้ค่าความเข้มที่เป็นขอบภาพแล้ว จะทำการแปลงข้อมูลภาพให้เป็นข้อมูล Binary คือเก็บเป็นค่าข้อมูล '0' กับ '1' โดยถ้าค่าระดับความเข้มใหม่ >= ค่า Threshold ให้เป็นค่า '1' แต่ถ้าค่าระดับความเข้มใหม่ < ค่า Threshold ให้ค่าเป็น '0' จากนั้นก็จะนำข้อมูล Binary ที่ได้ไปทำการหาขอบภาพที่บาง

- ขั้นตอนที่ 2.2 การหาขอบภาพที่บาง (Thinning Edge) โดยใช้วิธี Simple Thinning [2] ซึ่งจะทำโดยการกำหนดความสัมพันธ์ของจุดในภาพ กับจุดของภาพที่อยู่ข้างเคียง (Neighbouring Pixel) โดยกำหนดความสัมพันธ์ของจุดเป็นตาราง (Window) ขนาด 3x3 ดังรูปที่ 4.3a โดยทั่วไปแล้วจะกำหนดให้วัตถุหรือสิ่งที่สนใจ (Object or Patern) มีค่าระดับความเข้มเป็น '1' ส่วนพื้นที่ที่ไม่สนใจ (Background) มีค่าเป็น '0' จากรูปที่ 4.3a จุด P เป็นจุดใด ๆ ภายในเนื้อวัตถุ ซึ่งนำมาวิเคราะห์หาความสัมพันธ์รอบจุด P_0 จะประกอบด้วยจุด P_8,P_7, P_6, P_5, P_4, P_3, P_2, P_1 ตามลำดับจาก 8->7->6->5->4->3->2->1 โดยการตรวจสอบว่าแต่ละลำดับจาก $P_{8-->1}$ ถ้ามีการเปลี่ยนค่าความเข้มจาก '0' เป็น '1' เพียง 1 ครั้ง ก็จะทำการกำหนดค่าให้จุด P_0 มีค่าเท่ากับ '0' ซึ่งจะสามารถทำให้กำจัดส่วนของขอบภาพที่หนาและขอบภาพที่มีขนาดสั้นและไม่จำเป็นออกพ ดังรูปที่ 4.3b-d และทำให้เกิดขอบภาพที่บางและชัดเจนยิ่งขึ้น จากนั้นจะทำการตรวจสอบข้อมูลภาพ Binary ที่มีขอบภาพที่บางว่าถ้ามีค่าเป็น '1' จะถือว่าเป็นจุด (Point) ซึ่งทำให้สามารถเก็บข้อมูลที่เป็น Point Data ได้ดังรูปที่ 5.9 เพื่อใช้ในการประมวลผลขั้นตอนที่ 3-5 ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.3 การหาขอบภาพโดยการหาความสัมพันธ์รอบจุด

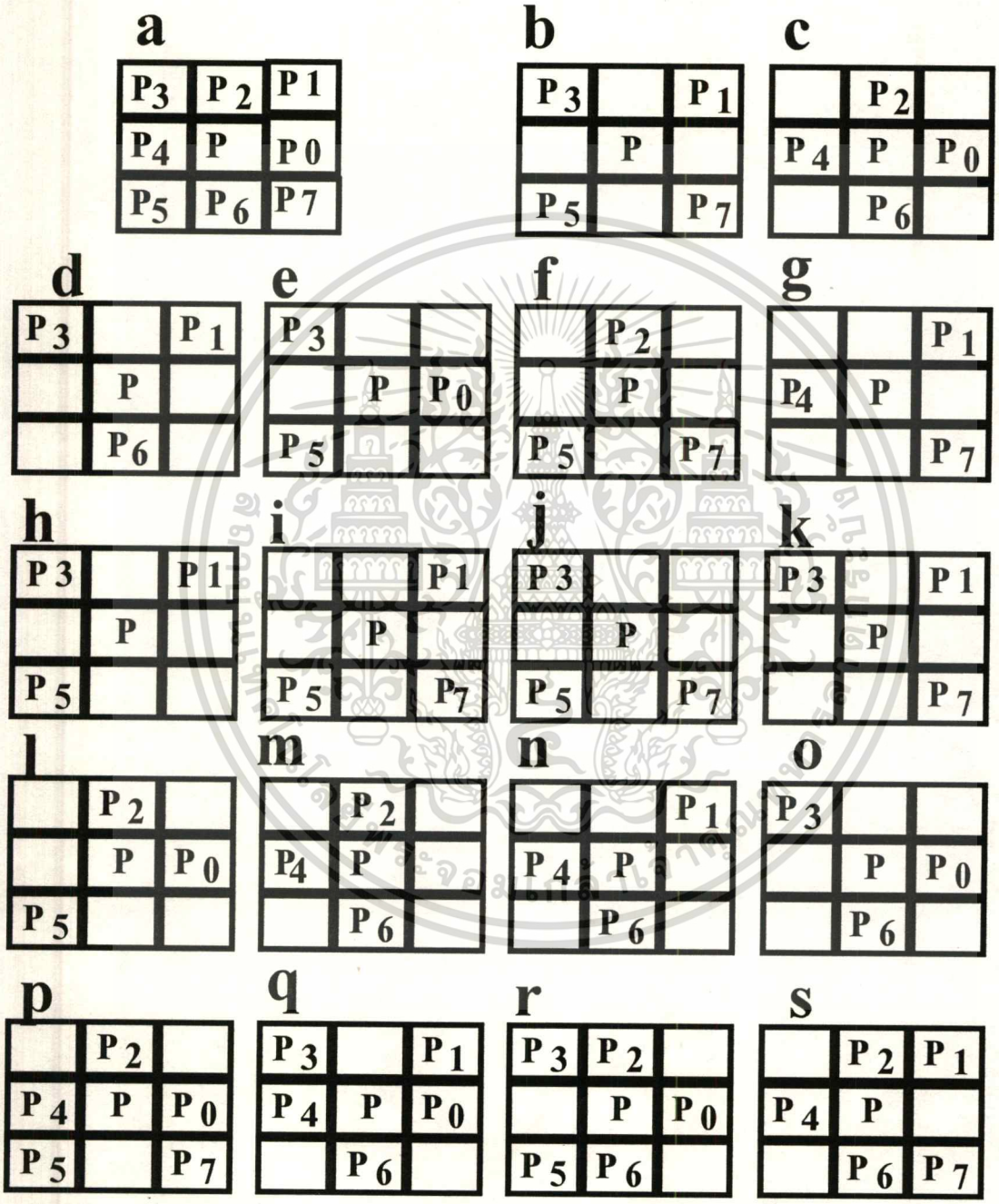


4.2.3 ขั้นตอนที่ 3 Node Improvement

การหาจุดรวมของเส้น (Node) ในการหาจุดรวมของเส้น จะทำโดยการกำหนดความสัมพันธ์ของจุดในภาพ กับจุดของภาพที่อยู่ข้างเคียง (Neighbouring Pixel) โดยกำหนดความสัมพันธ์ของจุดเป็นตาราง (Window) ขนาด 3x3 ของข้อมูลที่ได้ในขั้นตอนที่ 2 และจากรูป 4.4a จุด P เป็นจุดใด ๆ ภายในเนื้อวัตถุ ซึ่งนำมาวิเคราะห์หาความสัมพันธ์รอบจุด P จะประกอบด้วยจุด P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7 และในการหาจุดรวมของเส้น (Node) จะมีตารางที่เป็นตัวกำหนดคุณสมบัติของ Node ดังรูปที่ 4.4b-s โดยการตรวจสอบว่าในแต่ละ P_{0-7} มีค่าเป็น '1' ตามคุณสมบัติที่กำหนดดังรูปที่ 4.4b-s หรือไม่ ถ้าตรงก็จะสามารถระบุได้ว่าเป็นส่วนของ (Node) ซึ่งจะได้ข้อมูล Node Data ดังรูปที่ 5.9 เพื่อใช้เป็นจุดเริ่มต้นของการหาส่วนของเส้นในขั้นตอนที่ 4 ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 แสดงความสัมพันธ์ของจุดในตาราง(3x3) ที่ใช้หา(Node)



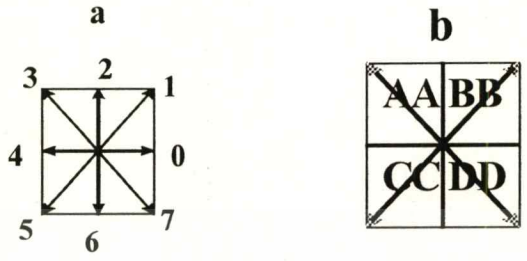
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 ขั้นตอนที่ 4 Line Tracking

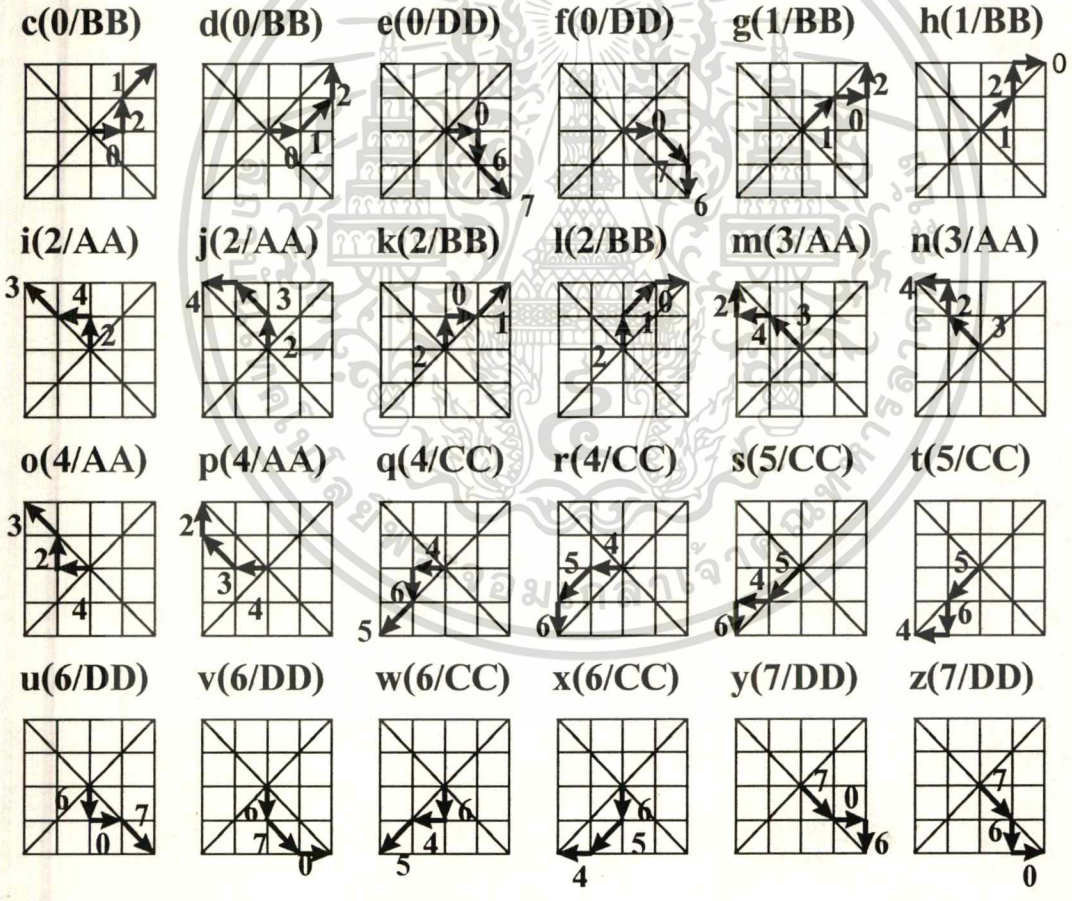
การหาส่วนของเส้นตรง (Line Tracking or Line Segment) จะทำโดยการกำหนดความสัมพันธ์ของจุดในทิศทางต่าง ๆ ที่ต่อเนื่องกันดังรูปที่ 4.5a และแนวโน้มของเส้นดังรูปที่ 4.5b จากข้อมูลที่ได้ในขั้นตอนที่ 2 และ 3 สามารถนำมาหาส่วนของเส้นได้ โดยการนำข้อมูลจุดของภาพที่ได้จากขั้นตอนที่ 2 มาหาความสัมพันธ์ทิศทางของจุด ดังรูปที่ 4.5a และนำข้อมูลจุดรวมของเส้น (Node) ที่ได้จากขั้นตอนที่ 3 เป็นจุดเริ่มต้นในการเริ่มหาส่วนของเส้นในภาพ โดยจะทำการตรวจสอบทิศทางของจุดในทิศทางที่ 0-7 ดังรูปที่ 4.5a พร้อมกับตรวจสอบแนวโน้มของเส้นว่ามีแนวโน้มไปในทิศทาง AA, BB, DD, CC ดังรูปที่ 4.5b โดยโอกาสของทิศทางของจุดและแนวโน้มของเส้นที่เป็นไปได้ ดังรูปที่ 4.5c-z จะถูกทำการตรวจสอบกับทุก ๆ จุดในภาพ และถ้าจุดใดที่ได้ถูกตรวจสอบแล้วว่าเป็นส่วนของเส้นตรงจะถูกแทนค่าด้วย '0' โดยจุดเริ่มต้นของเส้นตรงจะเริ่มที่จุดที่เป็น (Node) หรือจุดที่มีแนวโน้มของเส้นเปลี่ยนทิศทาง และจุดสิ้นสุดของเส้นจะเกิดขึ้นเมื่อเจอ (Node) อื่น หรือเมื่อไม่มีจุดต่อเนื่องในทิศทางที่ 0-7 หรือแนวโน้มของทิศทางของเส้นเปลี่ยนไป โดยรูปที่ 4.6, 4.7, 4.8, 4.9 จะแสดงตัวอย่างในการตรวจสอบทิศทางของจุดและแนวโน้มของเส้นในภาพ ซึ่งจะทำให้ได้ข้อมูล Line in Polygon และ Point Data ในส่วนของ Chain List, Begin Point, End Point, Lenght ดังรูปที่ 5.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 แสดงความสัมพันธ์ของจุดที่ทำให้เกิดเส้นและทิศทางของเส้น



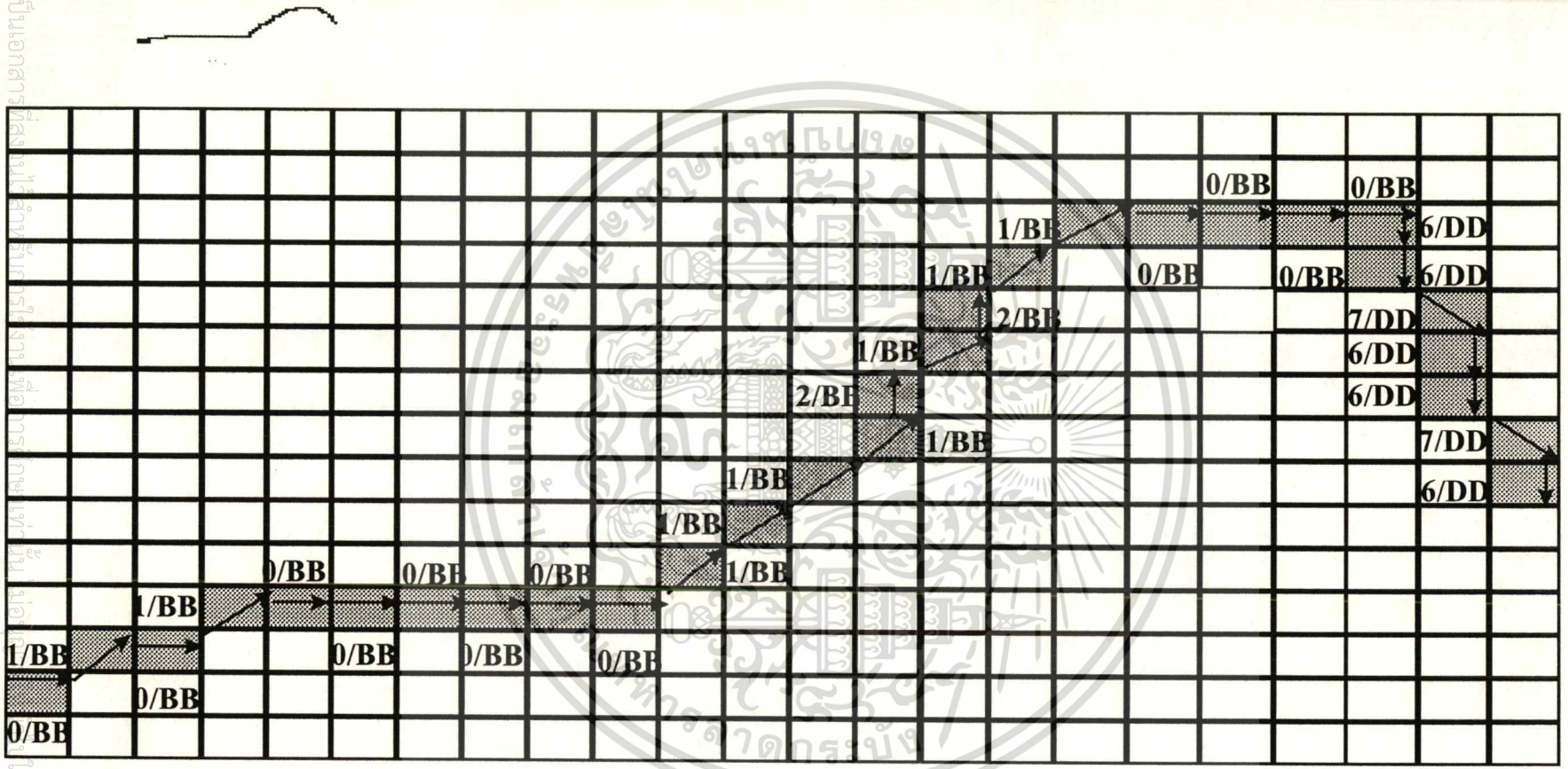
ทิศทางของจุด แนวโน้มทิศทางของเส้น



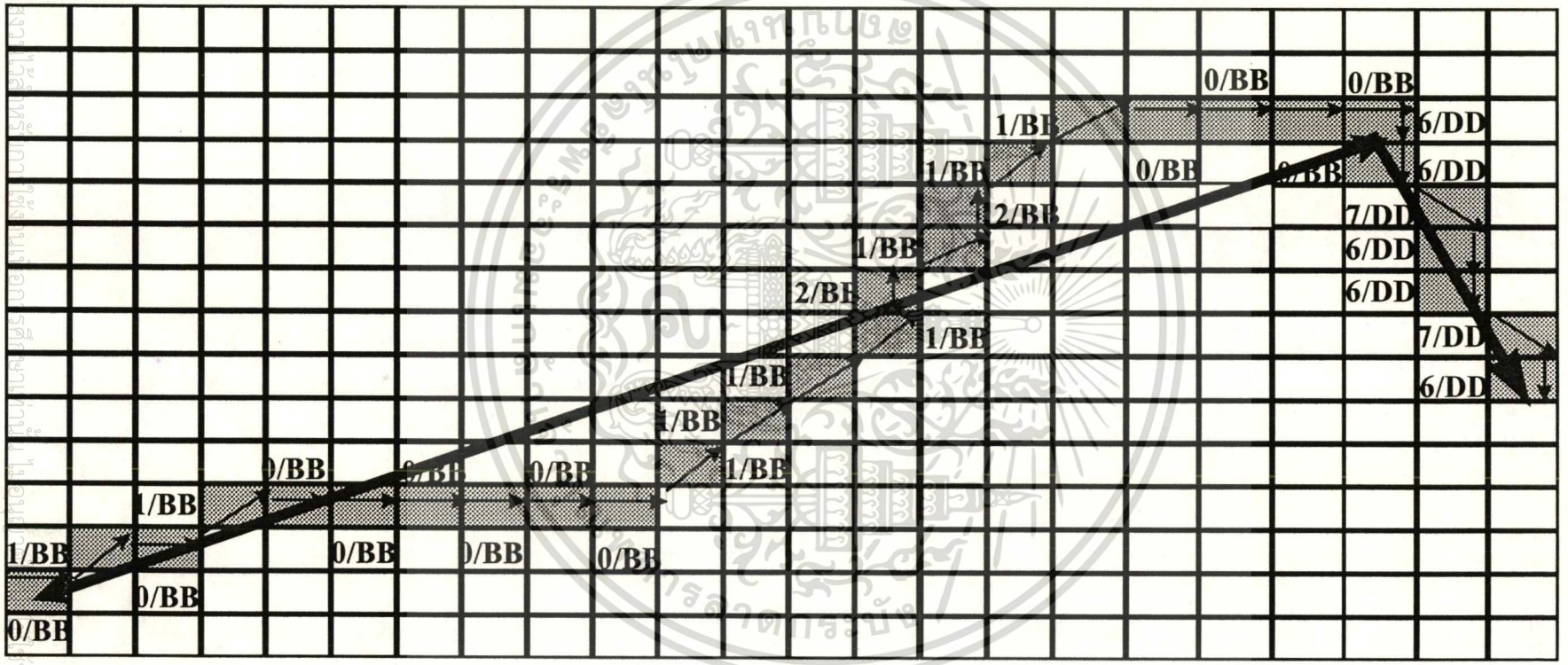
โอกาสที่จะเกิดขึ้นได้ในแนวโน้มทิศทางของเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 ตัวอย่างการหาความสัมพันธ์ของจุดที่ทำให้เกิดเส้นและแนวโน้มทิศทางของเส้น

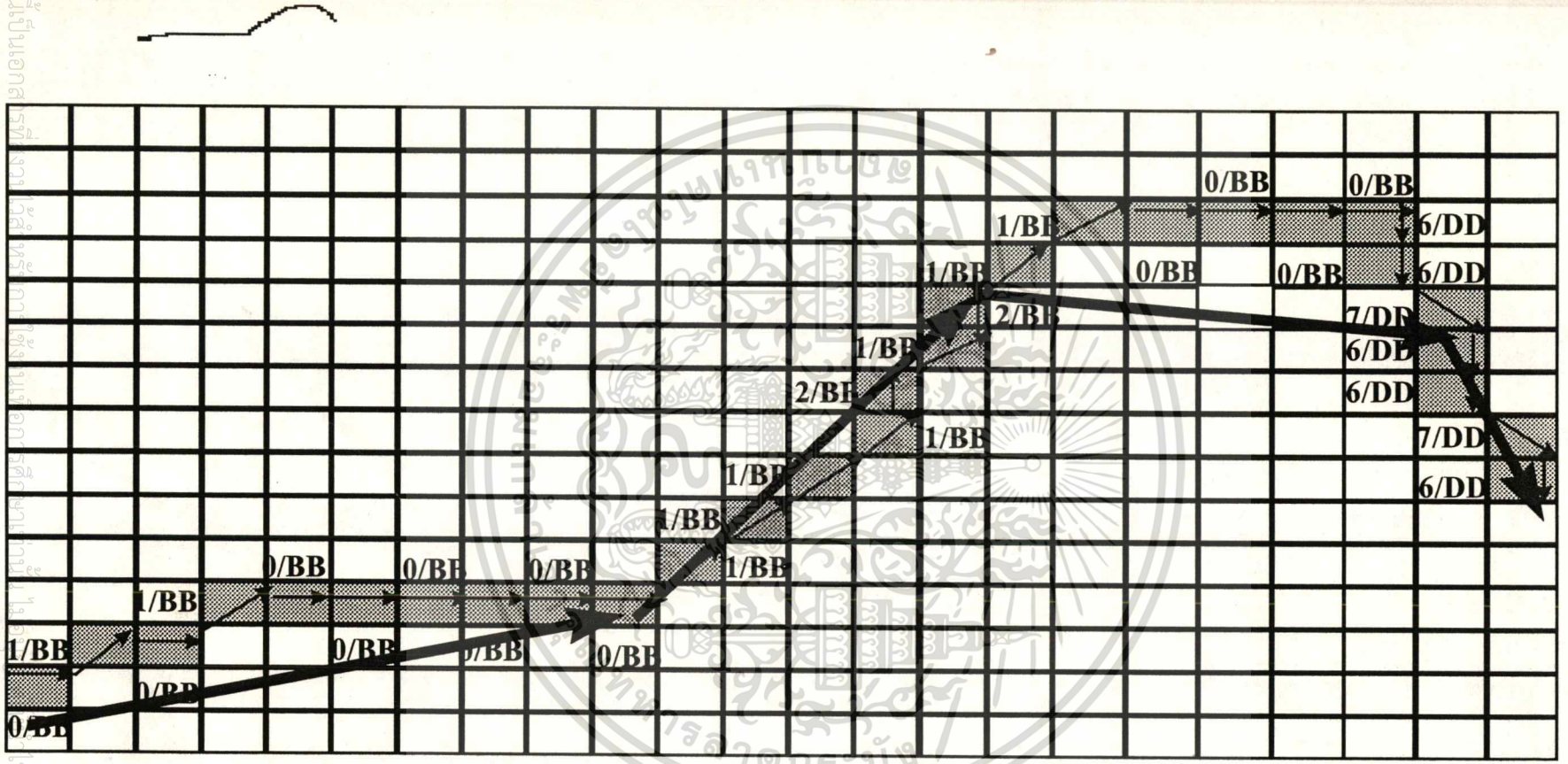


รูปที่ 4.7 การหาเส้น โดยเริ่มเส้นใหม่เมื่อแนวโน้มทิศทางของเส้นเปลี่ยน



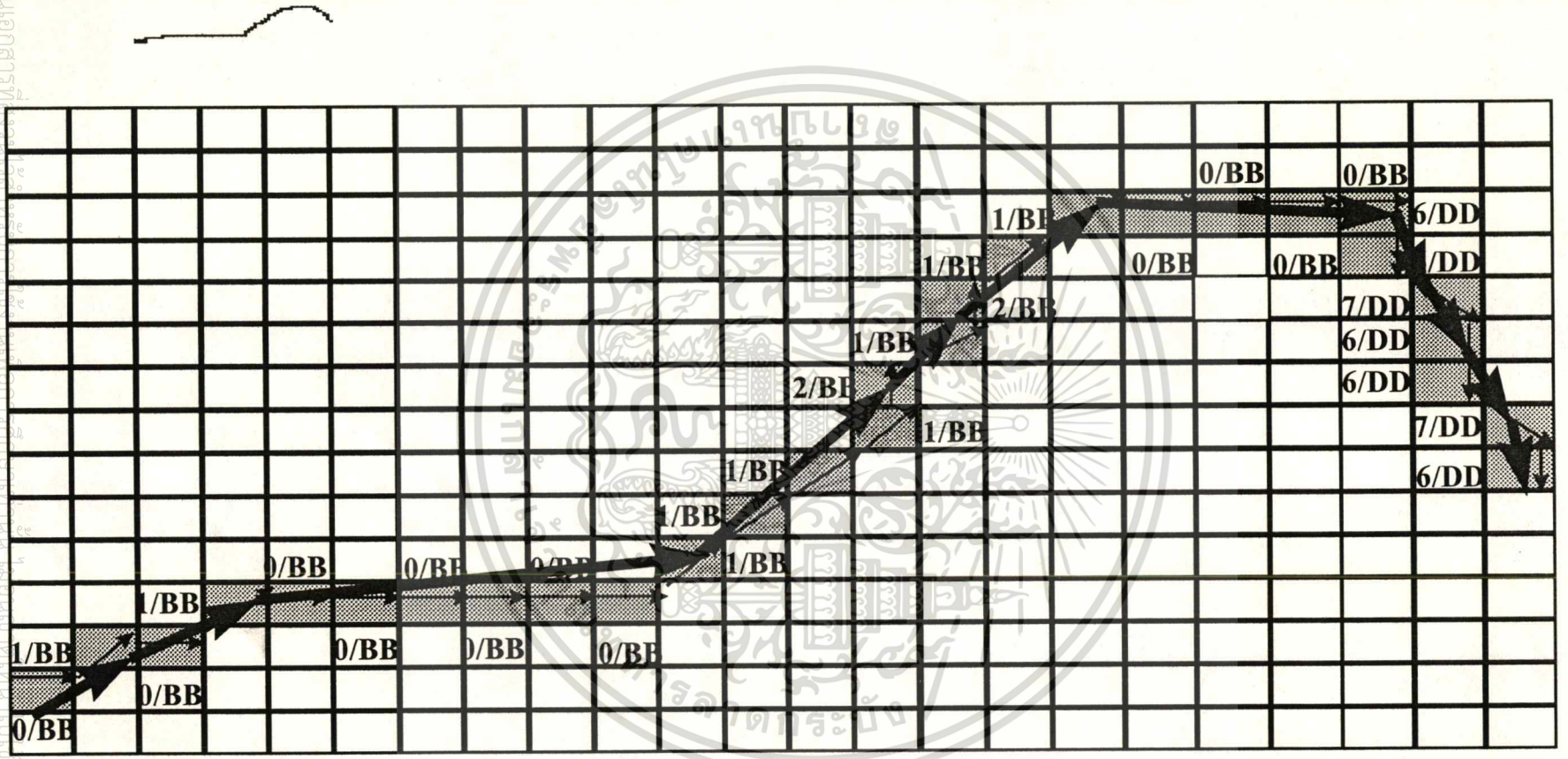
ซึ่งทำให้ส่วนประกอบของภาพแสดงด้วยเส้น 2 เส้น และภาพยังไม่ใกล้เคียงภาพต้นแบบด้านบน

รูปที่ 4.8 การหาเส้น โดยเริ่มเส้นใหม่เมื่อแนวโน้มทิศทางของเส้นเปลี่ยน และทิศทางเปลี่ยน 4 ครั้ง



ซึ่งทำให้ส่วนประกอบของภาพแสดงด้วยเส้น 4 เส้น และภาพยังไม่ใกล้เคียงภาพต้นแบบด้านบน

รูปที่ 4.9 การหาเส้น โดยเริ่มเส้นใหม่เมื่อแนวโน้มทิศทางของเส้นเปลี่ยน และทิศทางเปลี่ยน 1 ครั้ง

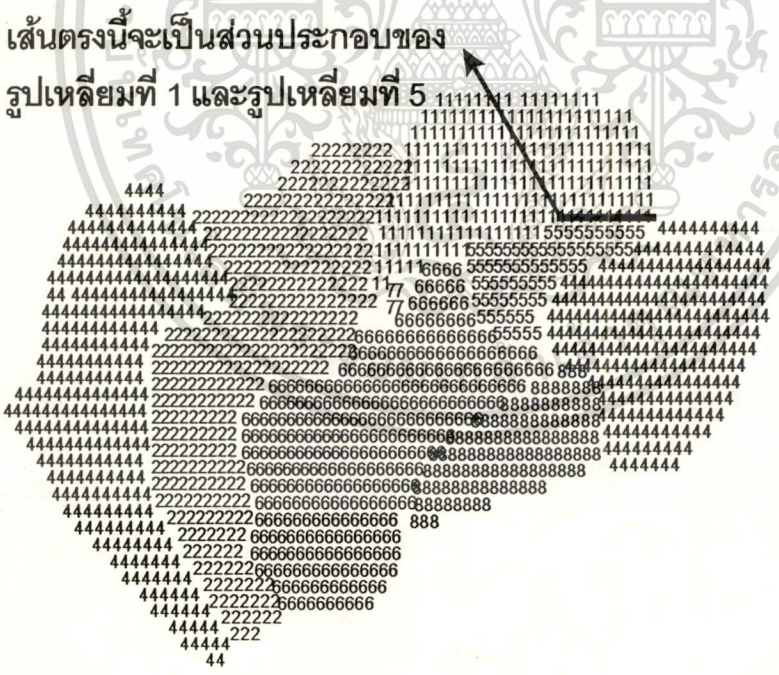


ซึ่งทำให้ส่วนประกอบของภาพแสดงด้วยเส้น 11 เส้น และภาพใกล้เคียงภาพต้นแบบด้านบนมากที่สุด

4.2.5 ขั้นตอนที่ 5 Segment Merge

การหารูปเหลี่ยมพื้นที่ (Polygon) จะทำโดยการกำหนดความสัมพันธ์ของจุดในทิศทางต่าง ๆ เพื่อหาว่าส่วนใดเป็นส่วนของพื้นภาพ และส่วนใดเป็นส่วนของรูปเหลี่ยมพื้นที่ และ เมื่อพบส่วนของรูปเหลี่ยมพื้นที่แล้วก็จะทำการกำหนดลำดับของรูปเหลี่ยมลงในส่วนของพื้นที่รูปเหลี่ยมนั้น (Labeling) ดังรูปที่ 4.10 และเมื่อได้ส่วนที่เป็นพื้นที่ของรูปเหลี่ยมแล้วก็จะหาว่าส่วนของเส้นตรงใดที่เป็นส่วนประกอบของพื้นที่โดยการหาความสัมพันธ์รอบ ๆ จุด ณ บริเวณจุดเริ่มต้นและจุดปลายของเส้น ของข้อมูลที่ได้จากขั้นตอน 4 กับข้อมูลที่ได้จากการ Labeling มาทำการตรวจสอบ และหลังจากหาความสัมพันธ์รอบจุดเริ่มต้น และจุดปลายของเส้นแล้วจะทำให้ทราบว่าเส้นไหนเป็นส่วนประกอบของรูปเหลี่ยม (Polygon) ใด ดังแสดงความสัมพันธ์ในตัวอย่างผลลัพธ์ข้อมูลเวคเตอร์ในรูปที่ 5.9

รูปที่ 4.10 แสดงการ Label หาพื้นที่ส่วนของรูปเหลี่ยมและหาเส้นที่เป็นส่วนประกอบของรูปเหลี่ยม



โดยเมื่อจบขั้นตอนที่ 5 แล้ว จะทำให้ได้ข้อมูลที่มีโครงสร้างแบบมาตราฐานของเวคเตอร์แบบอัตโนมัติ ซึ่งประกอบไปด้วย จุด เส้น รูปเหลี่ยมพื้นที่ โดยข้อมูลที่ได้จะมีความสัมพันธ์ซึ่งกันและกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลลัพธ์จากการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์

จากการทดลองแปลงข้อมูลภาพแผนที่ภาคเหนือและภาคตะวันออกเฉียงเหนือของประเทศไทย จากราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ ซึ่งทำตามขั้นตอนที่ได้ออกแบบ คือ ทำการ Scanning, Skeletoninzing, Node Improvement, Line Tracking, Segment Merge ตามขั้นตอนที่ 1-5 ในรายละเอียดดังบทที่ 4 ทำให้ได้ผลลัพธ์ดังนี้

1. จากการทดลองภาพขอบภาพจากภาพแผนที่ภาคเหนือ [ภาคผนวก ข] และภาพแผนที่ภาคตะวันออกเฉียงเหนือ [ภาคผนวก ค] ของประเทศไทย โดยวิธี Range Edge และกำหนดค่า Threshold>1 และบริเวณความเข้มที่สนใจในตารางขนาด 2x2 ปรากฏว่าขอบภาพที่ได้จากระดับความเข้มของแม่สีแต่ละสีที่แสดงในภาพได้ขอบภาพไม่ครบถ้วน ดังรูปที่ 5.1b, 5.1c, 5.1d เนื่องจากแต่ละจุดของภาพจะมีระดับความเข้มของแม่สีแต่ละสีไม่เท่ากัน บางจุดจะมีค่าความเข้มของสีแดง แต่บางจุดอาจไม่มีค่าความเข้มของสีแดงเลยก็ได้ ดังนั้นจึงทำให้แสดงขอบไม่ครบถ้วน แต่ถ้าเป็นการหาขอบภาพจากสีขาวดำจะทำให้ได้ขอบภาพอย่างครบถ้วน ดังรูปที่ 5.1a

2. ในการหาขอบภาพกับภาพแผนที่ภาคเหนือและภาคตะวันออกเฉียงเหนือของประเทศไทย โดยเลือกค่าความเข้มของสีขาวดำมาใช้ในการประมวลผลภาพเพราะจะแสดงขอบภาพที่ครบถ้วนที่สุด และทดลองหาขอบภาพโดยใช้วิธี Sobel Edge และใช้ค่า Threshold > 32 ซึ่งทำให้ได้ผลลัพธ์ของขอบภาพบางส่วนขาดหายไป และขอบภาพบางส่วนเส้นแตกไม่สม่ำเสมอ ดังรูปที่ 5.2a, 5.3a และเมื่อเปลี่ยนค่า Threshold > 16 ปรากฏว่าขอบภาพดีขึ้น ส่วนของเส้นครบ แต่ยังมีปัญหาขอบภาพบางส่วนหายไป ดังรูปที่ 5.2b, 5.3b ดังนั้นจึงเปลี่ยนค่า Threshold > 1 ซึ่งปรากฏว่าได้เส้นของภาพครบแต่เป็นขอบภาพที่หนามาก ดังรูปที่ 5.2c, 5.3c ต่อมาได้เปลี่ยนวิธีการหาขอบภาพโดยวิธี Range Edge ตามขั้นตอนที่ 2.1 ดังรายละเอียดในบทที่ 4 นั้น โดยเลือกค่า Threhold > 1 และขนาดตารางตรวจสอบเส้นเท่ากับ 2x2 ซึ่งทำให้ได้ขอบภาพที่ชัดเจนและไม่หนาเกินไป ดังรูปที่ 5.2d, 5.3d และเมื่อผ่านการทำขอบภาพให้บางโดยวิธี Simple Thining Edge ตามขั้นตอนที่ 2.2 ดังรายละเอียดในบทที่ 4 นั้น จะทำให้ได้ขอบภาพที่บาง ดังรูปที่ 5.2e, 5.3e

3. นำข้อมูลภาพที่ได้จากการหาขอบภาพกับภาพแผนที่ภาคเหนือและภาคตะวันออกเฉียงเหนือของประเทศไทย ด้วยวิธี Range Edge ที่ค่า Threshold>1 และกำหนดบริเวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเข้มที่สนใจในตารางขนาด 2x2 และทำขอบภาพให้บางด้วยวิธี Simple Thinning Edge แล้วมาทำการหาจุดรวมของเส้น (Node) ตามขั้นตอนที่ 3 ดังรายละเอียดในบทที่ 4 นั้น ซึ่งทำการหาความสัมพันธ์ของจุด และทำให้ได้ผลลัพธ์ของ (Node) ดังรูปที่ 5.4a, 5.5a แล้วจะทำการหาส่วนของเส้นตรง ตามขั้นตอนที่ 4 ดังรายละเอียดในบทที่ 4 นั้น ซึ่งทำให้ได้ผลลัพธ์ ในการหาส่วนของเส้นตรงโดยระบุบเงื่อนไขว่า ถ้าแนวโน้มของเส้นเปลี่ยนจะถือว่าเริ่มเส้นใหม่ ซึ่งจำนวนเส้นที่ได้จะน้อยแต่ภาพที่ได้จะไม่ใกล้เคียงกับภาพต้นแบบ ดังรูปที่ 5.4b, 5.5b และทำการหาส่วนของเส้นตรงใหม่โดยระบุบเงื่อนไขว่าถ้าแนวโน้มของเส้นเปลี่ยนและทิศทางของความต่อเนื่องของเส้นเปลี่ยน 20 ครั้ง ผลลัพธ์แสดงดังรูปที่ 5.4c ถ้าแนวโน้มของเส้นเปลี่ยนและทิศทางของความต่อเนื่องของเส้นเปลี่ยน 10 ครั้ง ผลลัพธ์แสดงดังรูปที่ 5.5c ถ้าแนวโน้มของเส้นเปลี่ยนและทิศทางของความต่อเนื่องของเส้นเปลี่ยน 5 ครั้ง ผลลัพธ์แสดงดังรูปที่ 5.4d, 5.5e ซึ่งทำให้เกิดความชัดเจนของภาพมากยิ่งขึ้น ถ้าแนวโน้มของเส้นเปลี่ยนและทิศทางของความต่อเนื่องของเส้นเปลี่ยน 1 ครั้ง ผลลัพธ์แสดงดังรูปที่ 5.4e, 5.5d ซึ่งทำให้ได้ภาพเหมือนภาพต้นแบบมากที่สุด จากการเปลี่ยนเทียบผลลัพธ์ของภาพในรูปที่ 5.4b-e, 5.5b-e นั้น จะพบว่าภาพที่กำหนดจำนวนครั้งในการเปลี่ยนแนวโน้มของทิศทางของเส้นน้อยเท่าใด ก็จะทำให้ภาพใกล้เคียงภาพต้นแบบมากที่สุด

4. ในการหาส่วนของพื้นที่รูปเหลี่ยมกับภาพแผนที่ภาคเหนือและภาคตะวันออกเฉียงเหนือของประเทศไทย โดยการทำ Labeling ตามขั้นตอนที่ 5 ดังรายละเอียดในบทที่ 4 นั้น จะนำข้อมูลภาพที่ได้จากการหาขอบภาพมาทำการ Labeling ซึ่งทำให้ได้ผลลัพธ์แผนที่ภาคตะวันออกเฉียงเหนือของประเทศไทยดังรูปที่ 5.6a และได้จำนวนรูปเหลี่ยมทั้งหมด 11 รูป และภาคตะวันเหนือของประเทศไทยดังรูปที่ 5.6b และได้จำนวนรูปเหลี่ยมทั้งหมด 8 รูป โดยจะแสดงรูปเหลี่ยมภาคเหนือของประเทศทั้งหมดดังรูปที่ 5.7 และรูปเหลี่ยมภาคตะวันออกเฉียงเหนือของประเทศทั้งหมดดังรูปที่ 5.8

5. จากการทดลองแปลงข้อมูลภาพ จากรูปแบบราสเตอร์เป็นเวคเตอร์ แบบอัตโนมัติตามขั้นตอนที่ 1-5 ดังรายละเอียดในบทที่ 4 โดยในการทดลองแปลงภาพแผนที่ภาคเหนือและภาคตะวันออกเฉียงเหนือของประเทศไทยแบบอัตโนมัตินั้น จะทำการหาขอบภาพจากระดับความเข้มของสีขาวดำด้วยวิธี Range Edge ที่ค่า Threshold>1 และกำหนดบริเวณความเข้มที่สนใจในตารางขนาด 2x2 และทำขอบภาพให้บางด้วยวิธี Simple Thinning Edge และทำการหา Node หา Line Segment หา Polygon โดยการ Labeling ซึ่งทำให้ได้จำนวนผลลัพธ์ของข้อมูลดังตารางที่ 5.1 โดยผลลัพธ์ของข้อมูลที่มีโครงสร้างเป็นมาตรฐานของรูปแบบเวคเตอร์จะประกอบด้วย จุด ส่วนประกอบของเส้น เส้น รูปเหลี่ยม ซึ่งสามารถดูตัวอย่างผลลัพธ์ของข้อมูลแบบเวคเตอร์ในภาคผนวก ง และความสัมพันธ์ของข้อมูลแบบเวคเตอร์ในรูปที่ 5.9

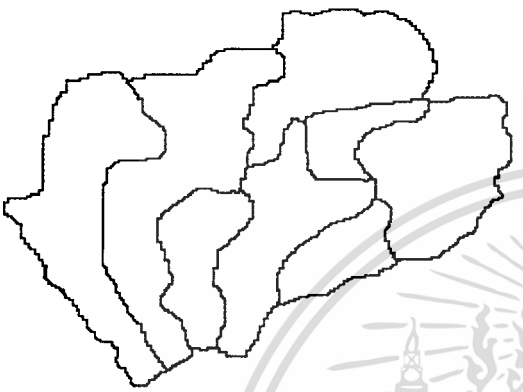
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 จำนวนผลลัพธ์ของข้อมูลรูปแบบเวคเตอร์ที่ได้จากแปลงแบบอัตโนมัติ

ข้อมูลเวคเตอร์	ภาคเหนือของประเทศไทย	ภาคตะวันออกเฉียงเหนือของประเทศไทย
จำนวนจุด	2000	2385
จำนวนเส้น	321	522
จำนวน Node	26	38
จำนวนรูปเหลี่ยม	8	11

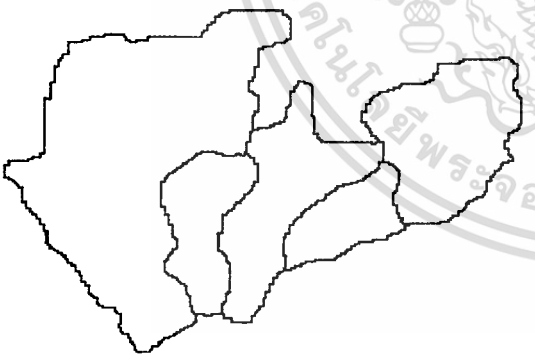
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.1 การหาขอบภาพกับระดับความเข้มของแม่สีแต่ละสี และระดับความเข้มขาวดำ โดยวิธี Range Edge ที่กำหนดค่า Threshold > 1 และขนาดตารางที่สนใจเท่ากับ 2x2

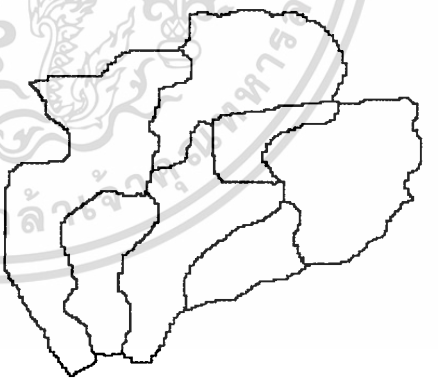


a. ขอบภาพจากระดับความเข้มขาวดำ

b. ขอบภาพจากระดับความเข้มสีแดง



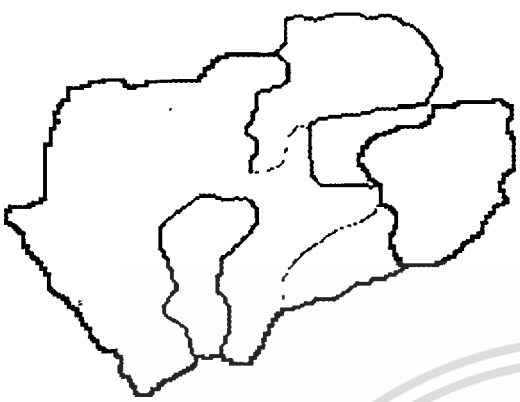
c. ขอบภาพจากระดับความเข้มขาวเขียว



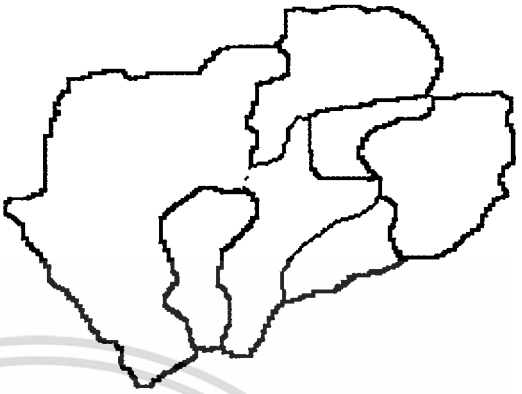
d. ขอบภาพจากระดับความเข้มสีน้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.2 การหาขอบภาพแผนที่ภาคเหนือของประเทศไทย ที่ระดับความเข้มของสีขาวดำ



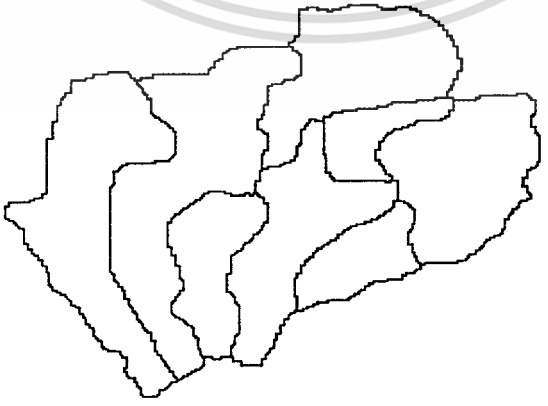
a. ขอบภาพจาก Sobel Edge ค่า Threshold >32



b. ขอบภาพจาก Sobel Edge ค่า Threshold >16

c. ขอบภาพจาก Sobel Edge ค่า Threshold >1

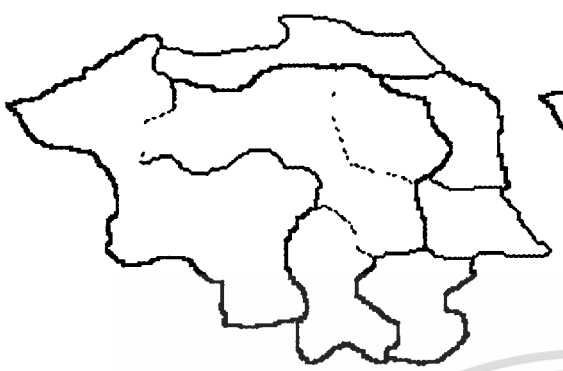
d. ขอบภาพจาก Rangel Edge ค่า Threshold>1 และภายใต้กรอบความเข้มที่ในใจในตาราง ขนาด 2x2



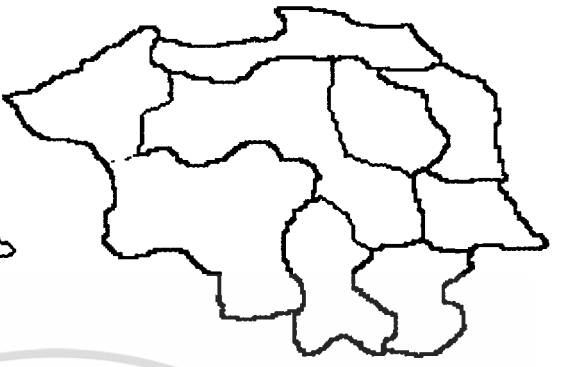
e. ขอบภาพที่บางโดยนำภาพที่ได้จาก การหาขอบภาพ ด้วยวิธี Rangel Edge มาทำ Simple Thinning Edge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

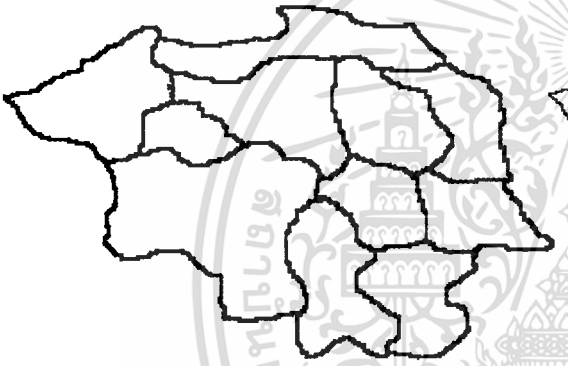
รูปที่ 5.3 การหาขอบภาพแผนที่ภาคตะวันออกเฉียงเหนือของประเทศไทย ที่ระดับความเข้มของสีขาวดำ



a. ขอบภาพจาก Sobel Edge ค่า Threshold>32

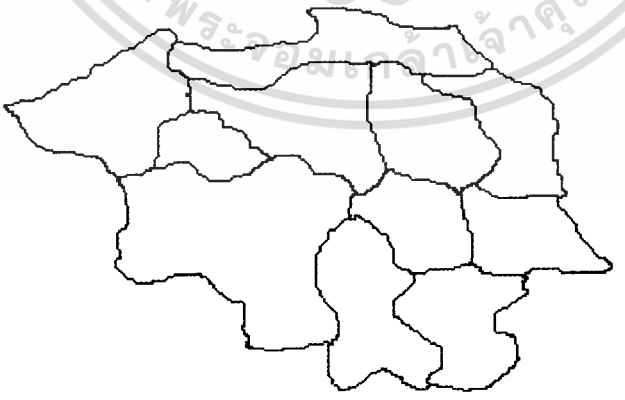


b. ขอบภาพจาก Sobel Edge ค่า Threshold >16



c. ขอบภาพจาก Sobel Edge ค่า Threshold>1

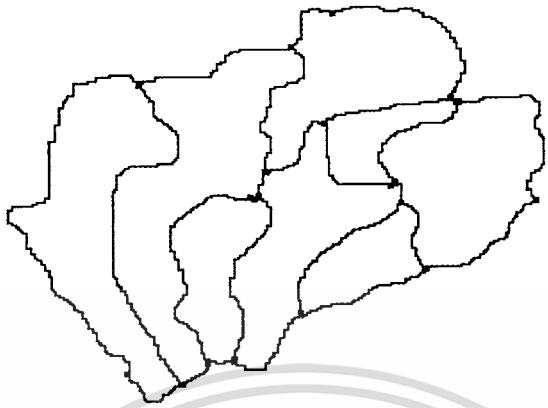
d. ขอบภาพจาก Rangel Edge ค่า Threshold >1 และภายใต้กรอบความเข้มที่ในใจในตาราง ขนาด 2x2



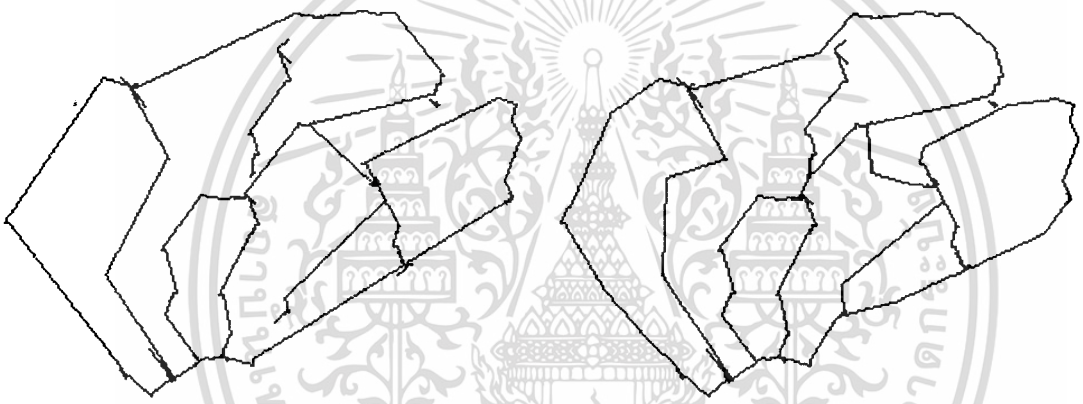
e . ขอบภาพที่บางโดยนำภาพที่ได้จาก การหาขอบภาพ ด้วยวิธี Rangel Edge มาทำ Simple Thinning Edge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.4 การ Node และส่วนประกอบของเส้นกับภาพแผนที่ภาคเหนือของประเทศไทย ที่ระดับความเข้มของสีขาวดำ



a. แสดงผลลัพธ์ที่ได้จากการหา Node



b. เริ่มเส้นใหม่เมื่อแนวโน้มของเส้นเปลี่ยน ได้จำนวนเส้น 60 เส้น

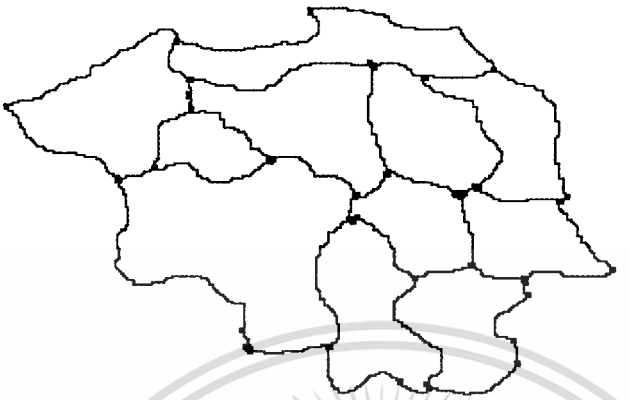
c. เริ่มเส้นใหม่เมื่อแนวโน้มของเส้นเปลี่ยน แสะทิศทางของจุดเปลี่ยน 20 ครั้ง ได้ จำนวนเส้น 125 เส้น

d. เริ่มเส้นใหม่เมื่อแนวโน้มของเส้นเปลี่ยน แสะทิศทางของจุดเปลี่ยน 5 ครั้ง ได้ จำนวนเส้น 227 เส้น

e. เริ่มเส้นใหม่เมื่อแนวโน้มของเส้นเปลี่ยน แสะทิศทางของจุดเปลี่ยน 1 ครั้ง ได้ จำนวนเส้น 321 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

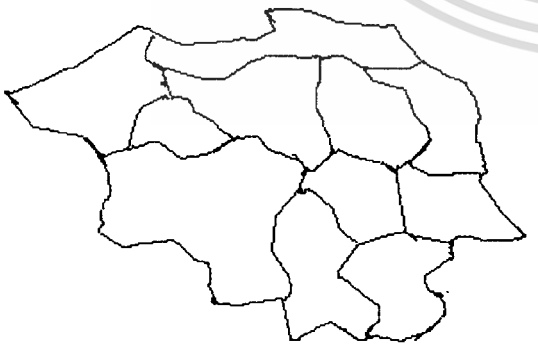
รูปที่ 5.5 การ Node และส่วนประกอบของเส้นกับภาพแผนที่ภาคตะวันออกเฉียงเหนือของประเทศไทย ที่ระดับความเข้มของสีขาวดำ



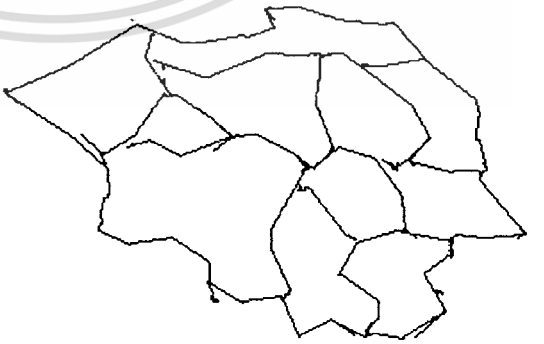
a. แสดงผลลัพธ์ที่ได้จากการหา Node

b. เริ่มเส้นใหม่เมื่อแนวโน้มของเส้นเปลี่ยน ได้จำนวนเส้น 129 เส้น

c. เริ่มเส้นใหม่เมื่อแนวโน้มของเส้นเปลี่ยน และทิศทางของจุดเปลี่ยน 10 ครั้ง ได้ จำนวนเส้น 379 เส้น



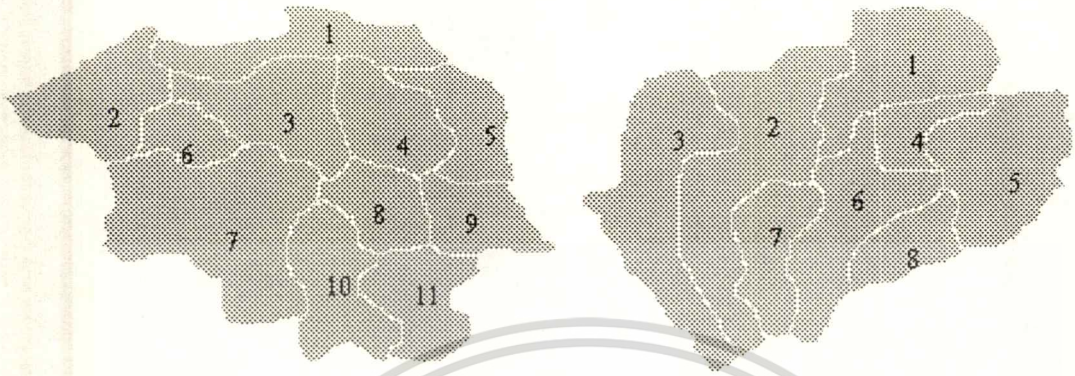
d. เริ่มเส้นใหม่เมื่อแนวโน้มของเส้นเปลี่ยน และทิศทางของจุดเปลี่ยน 1 ครั้ง ได้ จำนวนเส้น 522 เส้น



e. เริ่มเส้นใหม่เมื่อแนวโน้มของเส้นเปลี่ยน และทิศทางของจุดเปลี่ยน 5 ครั้ง ได้ จำนวนเส้น 492 เส้น

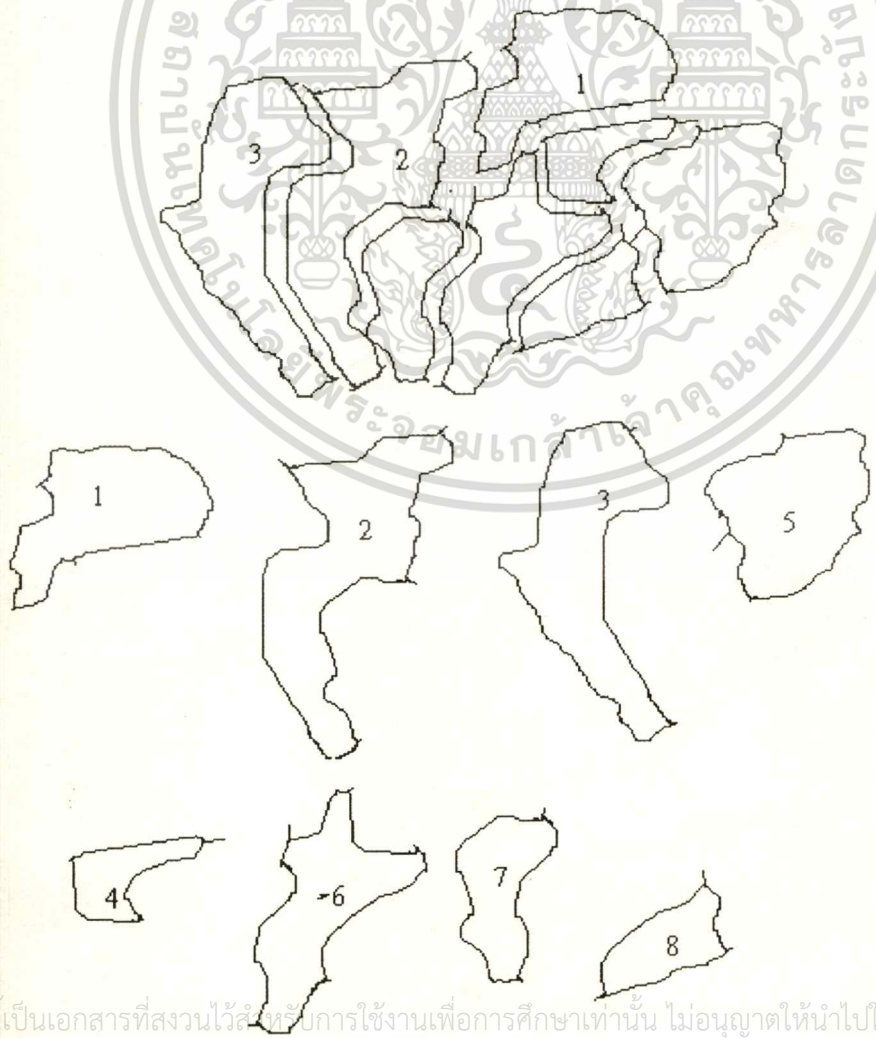
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.6 ผลลัพธ์ของภาพที่ได้จากการทำ Labeling



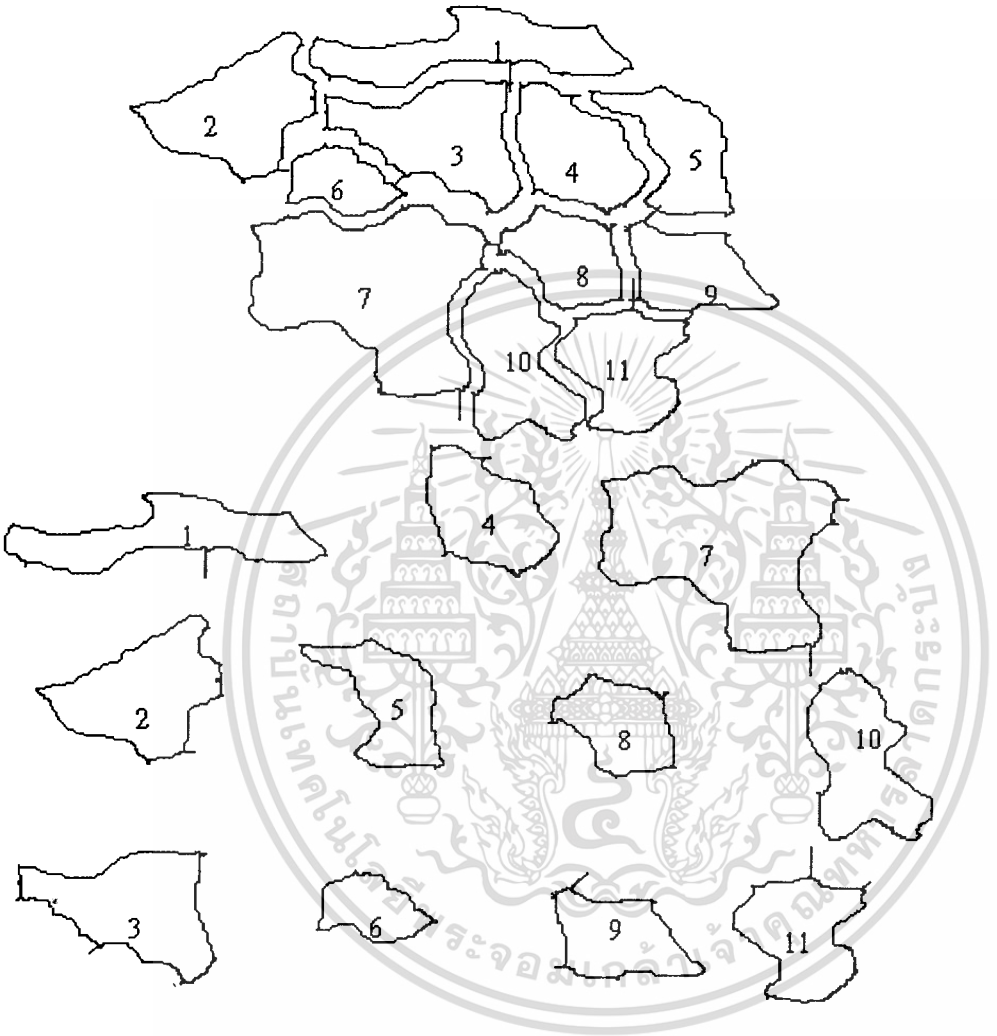
a. แผนที่ภาคเหนือ **b. แผนที่ภาคตะวันออกเฉียงเหนือ**

รูปที่ 5.7 ผลลัพธ์ของภาพที่ได้จากการหารูปเหลี่ยมพื้นที่ทางภาคเหนือของประเทศไทย



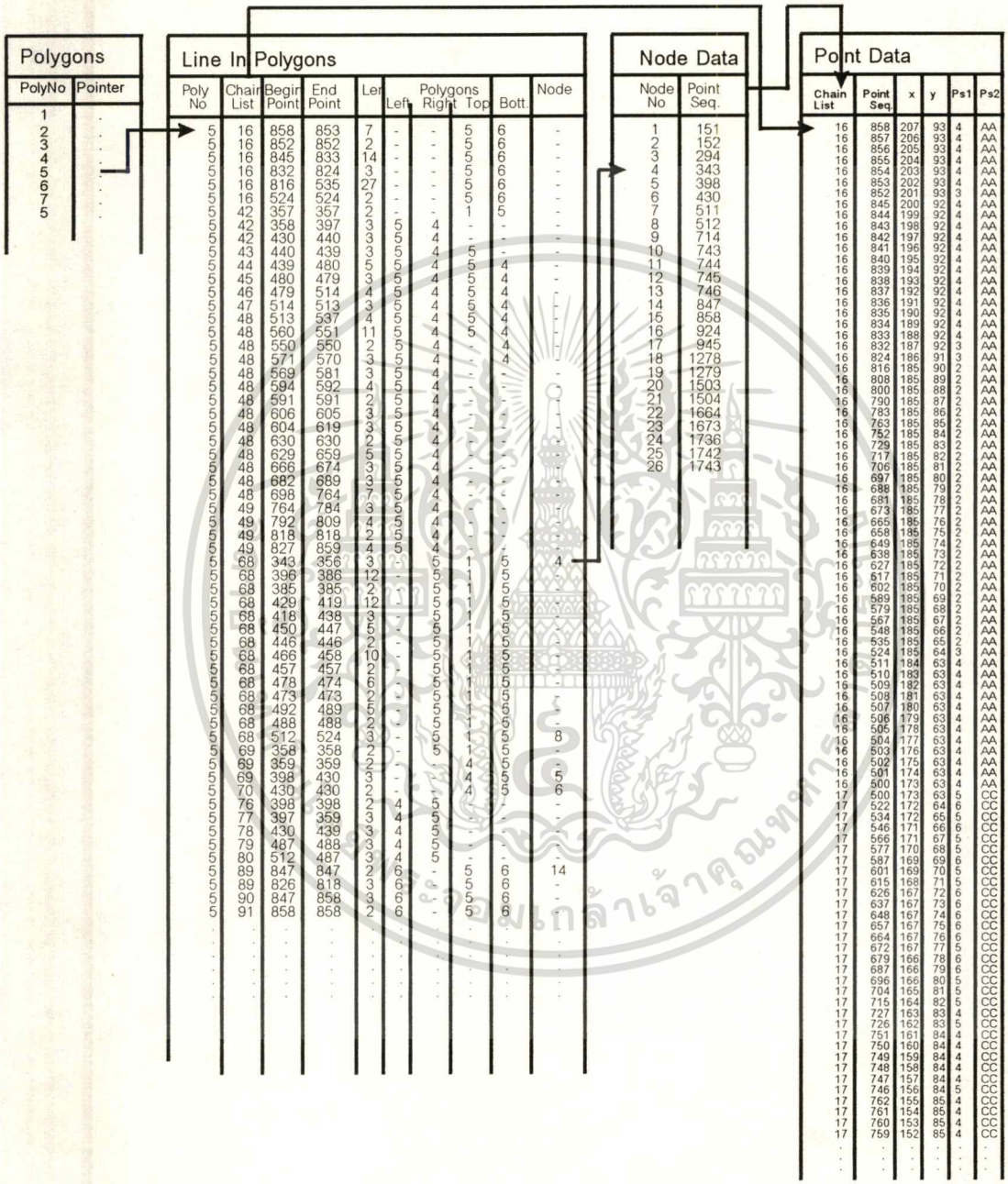
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.8 ผลลัพธ์ของภาพที่ได้จากการหารูปเหลี่ยมพื้นที่ทางภาคตะวันออกเฉียงเหนือของประเทศไทย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.9 แสดงตัวอย่างผลลัพธ์ในความสัมพันธ์ของข้อมูลเวคเตอร์ที่ได้จากการแปลงข้อมูลราสเตอร์เป็นเวคเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

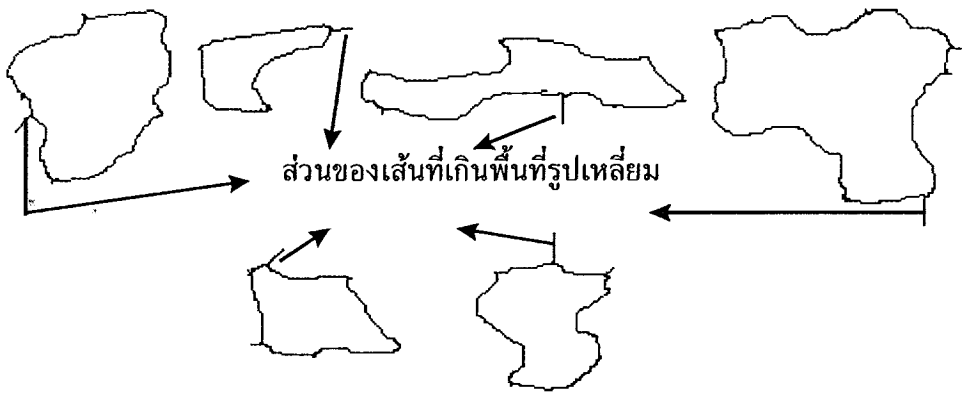
แปลงของทิศทางความต่อเนื่องของจุด ในการกำหนดจุดเริ่มต้นและจุดสิ้นสุดของเส้นให้น้อยที่สุด แต่ก็จะได้จำนวนเส้นของภาพมาก และถ้ากำหนดมากเกินไปก็จะทำให้ภาพผลลัพธ์ที่ได้ในรูปแบบเวคเตอร์ไม่เหมือนภาพต้นแบบ แต่ก็จะได้จำนวนเส้นของภาพน้อยลง

4. การหาเส้นที่เป็นส่วนประกอบของรูปเหลี่ยมหรือวัตถุในภาพ จะต้องทำการ Labeling หาพื้นที่ของแต่ละ Polygon ก่อน จึงจะหาว่าเส้นที่เป็นส่วนประกอบของ Polygon ซึ่งทำให้ต้องเสียเวลาในประมวลผลเพิ่มขึ้น หากสามารถคิดค้นตัวกรองที่กำหนดความสัมพันธ์ของเส้นว่าเส้นใดเป็นส่วนประกอบของรูปเหลี่ยมได้เลย ก็จะทำให้ใช้เวลาในการประมวลผลรวดเร็วยิ่งขึ้น อีกทั้งวิธีการดังกล่าวยังทำให้เกิดปัญหาของส่วนของเส้นตรงที่มีความยาวทำให้เส้นที่เป็นส่วนประกอบของรูปเหลี่ยมเกินออกมาดังรูปที่ 6.1 ซึ่งควรหาแนวทางในพัฒนาและแก้ไขต่อไป

5. จากผลลัพธ์ของข้อมูลเวคเตอร์ที่ได้นั้น จะเห็นได้ว่าในการเก็บเป็น จุด เส้น รูปเหลี่ยม จุดรวมของเส้น นั้นจะมีความสัมพันธ์ซึ่งกันและกัน สำหรับผลลัพธ์ที่เป็นรูปแบบเวคเตอร์ที่ได้จากการแปลงแสดงดังรูปที่ 5.9 และภาคผนวก ง นั้น สามารถนำข้อมูลที่แปลงได้นี้มาจัดเป็นรูปแบบโครงสร้างข้อมูลของ Software ที่มีการเก็บแบบเวคเตอร์ เช่น Auto-CAD หรือ GIS Application Software ซึ่งจะทำให้สะดวกในการประมวลผลข้อมูลภาพยิ่งขึ้น

6. จากภาพที่ใช้ในการทดลองของงานวิจัยนี้เป็นภาพแผนที่ภาคเหนือ และภาคตะวันออกเฉียงเหนือของประเทศไทย ซึ่งมีการแบ่งแยกโทนสีของวัตถุในภาพอย่างชัดเจน ทำให้ไม่จำเป็นต้องทำ Preprocessing และสามารถทำตามขั้นตอนในการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติที่งานวิจัยนี้ได้ออกแบบและพัฒนาขึ้นได้ ซึ่งการแบ่งแยกโทนสีที่ชัดเจนของวัตถุในภาพต้นแบบเป็นข้อจำกัดสำหรับงานวิจัยนี้ แต่ถ้าหากภาพต้นแบบมีการกระจายของกลุ่มของสีมาก เช่นภาพวิว ภาพคน และไม่มีการแบ่งแยกโทนสีที่ชัดเจน มีข้อมูลรบกวนภาพ ก็ควรศึกษาวิธีในการประมวลผลข้อมูลภาพ มาทำการ Preprocessing เพื่อแบ่งแยกโทนสีและหาวัตถุเป้าหมายที่ต้องการ ก่อนทำการแปลงเป็นรูปแบบราสเตอร์เป็นเวคเตอร์ต่อไป

รูปที่ 6.1 แสดงส่วนของเส้นตรงที่เป็นส่วนประกอบของรูปเหลี่ยม เกินพื้นที่รูปเหลี่ยม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปวิจารณ์และเสนอแนะ

จากการออกแบบขั้นตอนในการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ โดยการเก็บข้อมูลภาพแผนที่ประเทศไทย[ภาคผนวก ก] ผ่านเครื่องสแกนเนอร์ และทำการตัดตกแต่งสีให้มีความชัดเจนและความสม่ำเสมอในระดับความเข้มของสี โดยใช้ Paint-Brush ซึ่งทำให้ได้ภาพแผนที่ภาคเหนือ[ภาคผนวก ข] และภาคตะวันออกเฉียงเหนือของประเทศไทย [ภาคผนวก ค] และทำการประมวลผลข้อมูลโดยการหาขอบภาพด้วยวิธี Range Edge[2] และ Sobel Edge[2] การทำขอบภาพให้บางด้วยวิธี SimpleThining[2] หาจุดรวมของเส้น (Node) และหาส่วนของเส้นตรง (Line Segment) ด้วยวิธีหาความสัมพันธ์ของจุดกับจุดรอบ ๆ ตามตารางขนาด (3x3) หารูปเหลี่ยม (Polygon) โดยวิธี Labeling ซึ่งสามารถสรุปผลได้ดังนี้

1. การหาโครงร่างของภาพที่บาง (Skeletonizing) โดยใช้เทคนิคการหาขอบภาพ (Edge Detection) ในการทดลองได้นำระดับความเข้มของสีขาวดำมาทำการหาขอบภาพด้วยวิธี Sobel Edge กับ Range Edge และเปรียบเทียบพบว่า การใช้วิธี Sobel Edge จะต้องสุ่มหาค่า Threshold ที่เหมาะสม เพื่อให้ได้ขอบภาพที่บางและครบถ้วน ทำให้เกิดการประมวลผลหลายครั้งในการสุ่มหาค่า และขอบภาพที่ได้ในกรณีที่ขอบครบถ้วนก็จะยังมีขอบที่หนากว่าวิธี Range Edge แต่ถ้าหากใช้วิธี Range Edge จะสามารถกำหนดขนาดความหนาบางของขอบที่ต้องการได้ โดยวิธีการกำหนดขนาดตารางตัวกรองในการหาขอบภาพกับค่า Threshold ที่ต้องการ เช่น กำหนดค่า Threshold>1 ซึ่งจะทำให้ขอบภาพหนา ดังนั้นจึงกำหนดขนาดขอบด้วยตารางตัวกรองขนาด 2x2 ประกอบด้วย ก็จะทำให้ได้ขอบภาพที่ค่อนข้างบางและครบถ้วน ดังนั้นในการทำขอบภาพให้บางนั้นจึงเพียงแค่ใช้วิธี Simple Thining Edge ซึ่งเป็นวิธีที่ไม่ยุ่งยากซับซ้อนและประมวลผลได้รวดเร็ว

2. การหาจุดรวมของเส้น (Node) นั้นจะมีตัวกรองหาความสัมพันธ์ของจุดในตาราง ซึ่งผลลัพธ์ที่ออกมากับการทดลองภาพแผนที่ภาคเหนือ และภาคตะวันออกเฉียงเหนือของประเทศไทย ปรากฏว่าได้ Node ครบถ้วนซึ่งทำให้สามารถใช้ Node เป็นตัวเริ่มในการหาจุดเริ่มต้นและจุดสิ้นสุดของเส้นต่าง ๆ ในภาพได้ครบถ้วนด้วย

3. การหาส่วนของเส้น (Line Tracking or Line Sgement) โดยการหาความสัมพันธ์ในทิศทางของจุดในตัวกรองตารางที่กำหนด ปรากฏว่า ถ้าต้องการให้ผลลัพธ์ข้อมูลที่เป็นรูปแบบเวคเตอร์ มีความใกล้เคียงกับรูปแบบราสเตอร์มากที่สุด ควรจะกำหนดจำนวนครั้งในการเปลี่ยน

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Star Jeffrey and Estes John , **Geographic Information System an Introduction,** Prentice Hall , 1990.

[2] Pitas Ioannis, **Digital Image Processing Algorithms** , Pentice Hall, 1993.

[3] Drummond Jane , Essen van Rob and Boulerie Pascal , **Some considerations on vcto rizing Algorithms** , ITC Journal , P153-157, 1991-3.

[4] Peuquet J Donna, **A Conceptual Framework and Comparison of Spatial Data Models,** Cartographica , Vol.21, No. 4, P 66-113, 1984.

[5] Heng Zhou, **Integration of Data Format for Vector and Raster Based GIS,** AIT Thesis, No.CS94-18, 1994.

[6] Chang Shi-Kuo, **Principle of Pictorial Information System Design,** Prentice-Hall, 1990

[7] Mask S. Monmonier, **Computer-Assisted Cartography Principles and Prospects,** Prentice-Hall , 1982

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

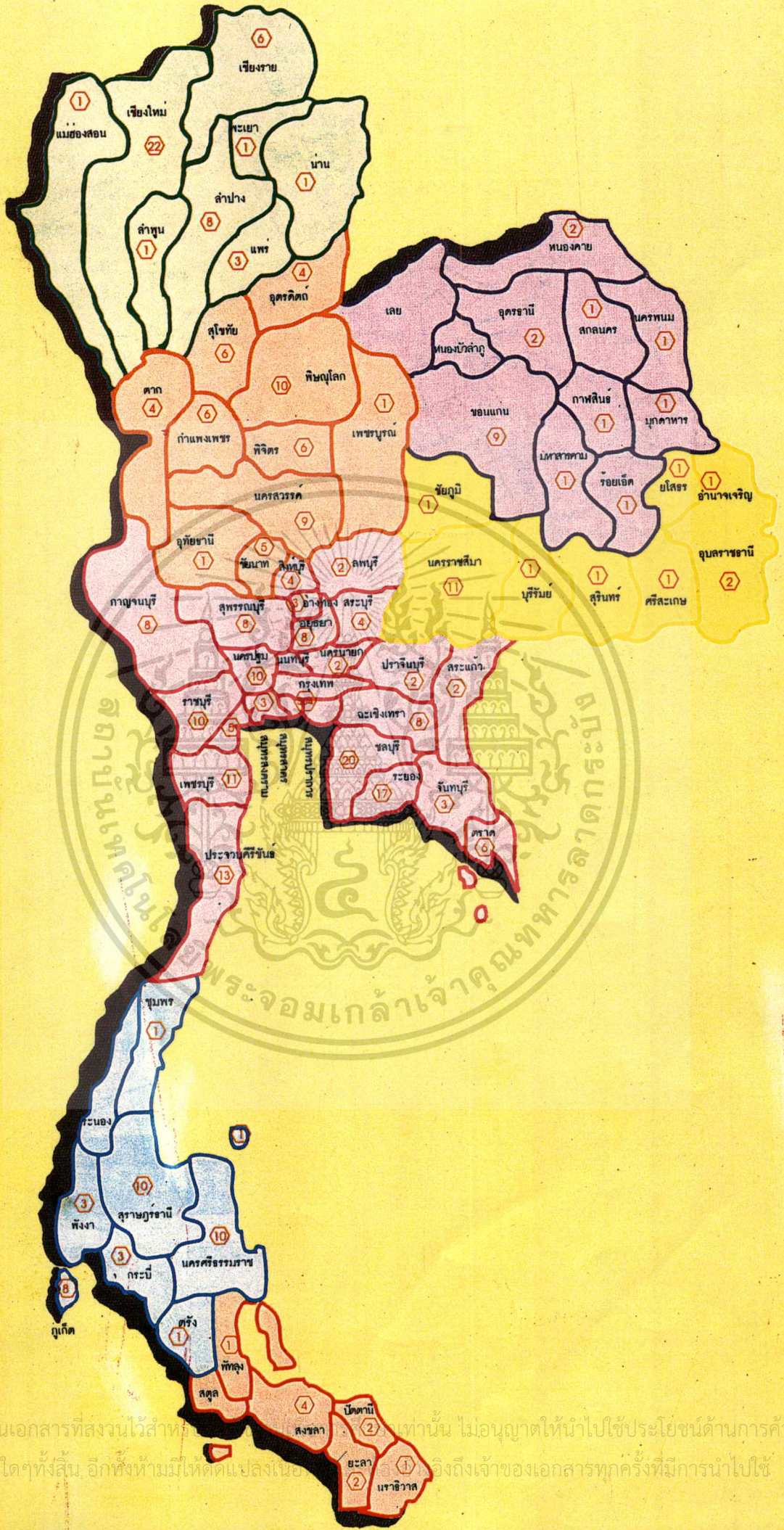
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ภาพแผนที่ประเทศไทยซึ่งเป็นภาพต้นแบบที่ใช้ในการแปลงข้อมูลจากรูปแบบราสเตอร์เป็นเวคเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



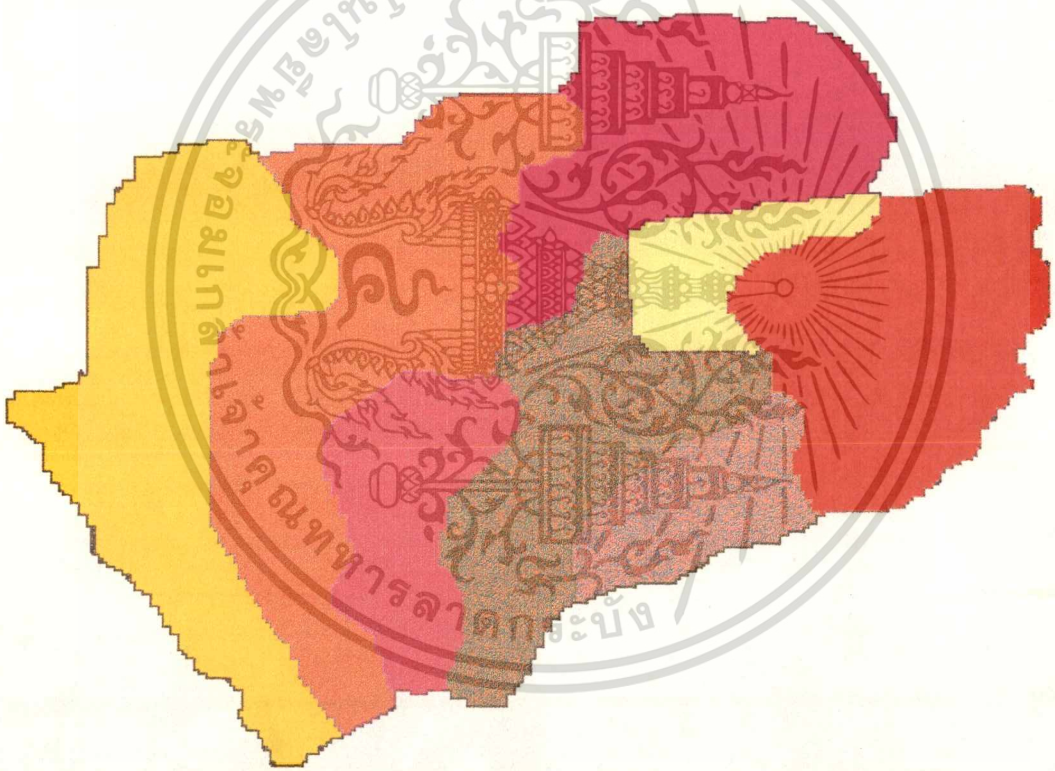
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ภาพแผนที่ภาคเหนือของประเทศไทยที่ได้หลังจากทำการตัดตกแต่งสีแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข



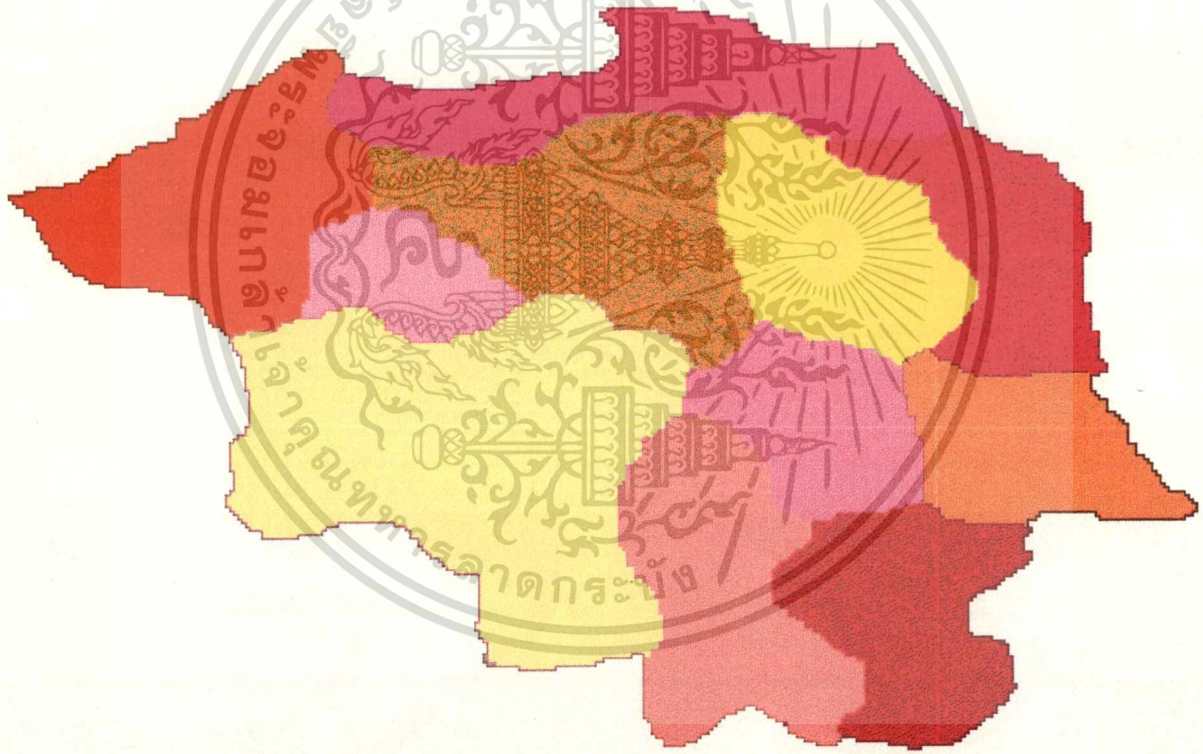
ภาพแผนที่ภาคเหนือของประเทศไทยที่ได้หลังจากทำการตัดตกแต่งสีแล้ว

ภาคผนวก ค

ภาพแผนที่ภาคตะวันออกเฉียงเหนือของประเทศไทยที่ได้หลังจากทำการตัดตกแต่งสีแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค



ภาพแผนที่ภาคตะวันออกเฉียงเหนือของประเทศไทยที่ได้หลังจากทำการตัดตกแต่งสีแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

ตัวอย่างผลลัพธ์ข้อมูลเวคเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POLYGON & CHAIN LIST & LINE DATA

Polygons

PolyNo	Pointer
1	.
2	.
3	.
4	.
5	.
6	.
7	.
5	.

Line In Polygons

Poly No	Chain List	Begin Point	End Point	Len	Polygons Left	Polygons Right	Top	Bott.	Node
5	16	858	853	7	-	-	5	6	-
5	16	852	852	2	-	-	5	6	-
5	16	845	833	14	-	-	5	6	-
5	16	832	824	3	-	-	5	6	-
5	16	816	535	27	-	-	5	6	-
5	16	524	524	2	-	-	5	6	-
5	42	357	357	2	-	-	1	5	-
5	42	358	397	3	5	4	-	-	-
5	42	430	440	3	5	4	-	-	-
5	43	440	439	3	5	4	5	-	-
5	44	439	480	5	5	4	5	4	-
5	45	480	479	3	5	4	5	4	-
5	46	479	514	4	5	4	5	4	-
5	47	514	513	3	5	4	5	4	-
5	48	513	537	4	5	4	5	4	-
5	48	560	551	11	5	4	5	4	-
5	48	550	550	2	5	4	-	4	-
5	48	571	570	3	5	4	-	4	-
5	48	569	581	3	5	4	-	-	-
5	48	594	592	4	5	4	-	-	-

CHAIN LIST & LINE DATA

Line In Polygons

Poly No	Chain List	Begin Point	End Point		Len Left	Polygons Right	Top	Bott.	Node
5	48	591	591	2	5	4	-	-	-
5	48	606	605	3	5	4	-	-	-
5	48	604	619	3	5	4	-	-	-
5	48	630	630	2	5	4	-	-	-
5	48	629	659	5	5	4	-	-	-
5	48	666	674	3	5	4	-	-	-
5	48	682	689	3	5	4	-	-	-
5	48	698	764	7	5	4	-	-	-
5	49	764	784	3	5	4	-	-	-
5	49	792	809	4	5	4	-	-	-
5	49	818	818	2	5	4	-	-	-
5	49	827	859	4	5	4	-	-	-
5	68	343	356	3	-	5	1	5	4
5	68	396	386	12	-	5	1	5	-
5	68	385	385	2	-	5	1	5	-
5	68	429	419	12	-	5	1	5	-
5	68	418	438	3	-	5	1	5	-
5	68	450	447	5	-	5	1	5	-
5	68	446	446	2	-	5	1	5	-

CHAIN LIST & LINE DATA

Line In Polygons

Poly No	Chain List	Begin Point	End Point	Len	Polygons Left	Right	Top	Bott.	Node
5	68	466	458	10	-	5	1	5	-
5	68	457	457	2	-	5	1	5	-
5	68	478	474	6	-	5	1	5	-
5	68	473	473	2	-	5	1	5	-
5	68	492	489	5	-	5	1	5	-
5	68	488	488	2	-	5	1	5	-
5	68	512	524	3	-	5	1	5	8
5	69	358	358	2	-	5	1	5	-
5	69	359	359	2	-	-	4	5	-
5	69	398	430	3	-	-	4	5	5
5	70	430	430	2	-	-	4	5	6
5	76	398	398	2	4	5	-	-	-
5	77	397	359	3	4	5	-	-	-
5	78	430	439	3	4	5	-	-	-
5	79	487	488	3	4	5	-	-	-
5	80	512	487	3	4	5	-	-	-
5	89	847	847	2	6	-	5	6	14
5	89	826	818	3	6	-	5	6	-
5	90	847	858	3	6	-	5	6	-
5	91	858	858	2	6	-	5	6	-
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

NODE DATA

Node Data

Node No	Point Seq.
1	151
2	152
3	294
4	343
5	398
6	430
7	511
8	512
9	714
10	743
11	744
12	745
13	746
14	847
15	858

Node Data

Node No	Point Seq.
16	924
17	945
18	1278
19	1279
20	1503
21	1504
22	1664
23	1673
24	1736
25	1742
26	1743

POINT DATA

Point Data Chain List	Point Seq.	x	y	Ps1	Ps2
16	858	207	93	4	AA
16	857	206	93	4	AA
16	856	205	93	4	AA
16	855	204	93	4	AA
16	854	203	93	4	AA
16	853	202	93	4	AA
16	852	201	93	3	AA
16	845	200	92	4	AA
16	844	199	92	4	AA
16	843	198	92	4	AA
16	842	197	92	4	AA
16	841	196	92	4	AA
16	840	195	92	4	AA
16	839	194	92	4	AA
16	838	193	92	4	AA
16	837	192	92	4	AA
16	836	191	92	4	AA
16	835	190	92	4	AA
16	834	189	92	4	AA
16	833	188	92	4	AA
16	832	187	92	3	AA
16	824	186	91	3	AA

Point Data Chain List	Point Seq.	x	y	Ps1	Ps2
16	816	185	90	2	AA
16	808	185	89	2	AA
16	800	185	88	2	AA
16	790	185	87	2	AA
16	783	185	86	2	AA
16	763	185	85	2	AA
16	752	185	84	2	AA
16	729	185	83	2	AA
16	717	185	82	2	AA
16	706	185	81	2	AA
16	697	185	80	2	AA
16	688	185	79	2	AA
16	681	185	78	2	AA
16	673	185	77	2	AA
16	665	185	76	2	AA
16	658	185	75	2	AA
16	649	185	74	2	AA
16	638	185	73	2	AA
16	627	185	72	2	AA
16	617	185	71	2	AA
16	602	185	70	2	AA

POINT DATA

Point Data Chain List	Point Seq.	x	y	Ps1	Ps2
16	589	185	69	2	AA
16	579	185	68	2	AA
16	567	185	67	2	AA
16	548	185	66	2	AA
16	535	185	65	2	AA
16	524	185	64	3	AA
16	511	184	63	4	AA
16	510	183	63	4	AA
16	509	182	63	4	AA
16	508	181	63	4	AA
16	507	180	63	4	AA
16	506	179	63	4	AA
16	505	178	63	4	AA
16	504	177	63	4	AA
16	503	176	63	4	AA
16	502	175	63	4	AA
16	501	174	63	4	AA
16	500	173	63	4	AA

Point Data Chain List	Point Seq.	x	y	Ps1	Ps2
17	500	173	63	5	CC
17	522	172	64	6	CC
17	534	172	65	5	CC
17	546	171	66	6	CC
17	566	171	67	5	CC
17	577	170	68	5	CC
17	587	169	69	6	CC
17	601	169	70	5	CC
17	615	168	71	5	CC
17	626	167	72	6	CC
17	637	167	73	6	CC
17	648	167	74	6	CC
17	657	167	75	6	CC
17	664	167	76	6	CC
17	672	167	77	5	CC
17	679	166	78	6	CC
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

ภาคผนวก จ

โปรแกรมต้นฉบับ

```
/*-------------------------------------------------------------------------*/
/*  convert file .pcx to .txt (RBG)     CNVPCX.C          */
/*  writen at 03/06/1995                                  */
/*  by kingkarn wongwipaporn                 */
/*-------------------------------------------------------------------------*/
#include <dos.h>
#include <bios.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/* define struct hearder file */
struct {
        unsigned char Password;
        unsigned char Version;
        unsigned char Encoding;
        unsigned char BPP;
        int         WinDemX1;
        int         WinDemY1;
        int         WinDemX2;
        int         WinDemY2;
        int             HoriResolut;
        int             VertResolut;
        unsigned char ColorMap[16][3];
        unsigned char reserve;
        unsigned char NumofPlane;
        int         BytePLine;
        int     PaletteInfo;
      } Hearder;
FILE *fpcx;             /* original file picture          (*.pcx)     */
FILE *fpiztxt,*fpizbyt;     /* picture data 320x200 (64000 dot)     (piz.txt,byt) */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FILE *fspaltxt,*fspalbyt;   /* standard palette at set_mode time              (spal.txt,byt) */
FILE *fpaltxt,*fpalbyt;     /* palette for show picture color 63 level        (pal.txt,byt)  */
FILE *fpalrtxt,*fpalrbyt;   /* palette R for show picture color 63 level      (palr.txt,byt) */
FILE *fpalgtxt,*fpalgbyt;   /* palette G for show picture color 63 level      (palg.txt,byt) */
FILE *fpalbtxt,*fpalbbyt;   /* palette B for show picture color 63 level      (palb.txt,byt) */
FILE *fopaltxt,*fopalbyt;   /* original palette form file *.pcx               (opal.txt,byt) */
FILE *fgpaltxt,*fgpalbyt;   /* gral palette for show picture color 63 level                  */
                            /*          from pal.txt cal.30R+.59G+.11B   (gpal.txt.byt) */
FILE *figpaltxt,*figpalbyt; /* gral palette for show picture color 63 level                  */
                            /*          from interupt vector 10(1017)    (igpal.txt.byt)*/
FILE *figpalrtxt,*figpalrbyt; /* gral R palette for show picture color 63 level              */
                            /*          from interupt vector 10(1017) (igpalr.txt.byt)*/
FILE *figpalgtxt,*figpalgbyt; /* gral G palette for show picture color 63 level              */
                            /*          from interupt vector 10(1017) (igpalg.txt.byt)*/
FILE *figpalbtxt,*figpalbbyt; /* gral B palette for show picture color 63 level              */
                            /*          from interupt vector 10(1017) (igpalg.txt.byt)*/
FILE *fogpaltxt,*fogpalbyt; /* gral palette for show picture color 256 level                 */
                            /*          from opal.txt cal.30R+.59G+.11B  (ogpal.txt,byt)*/
int MAXLINE;
int MAXCOLUMN;
int MaxLoop,linecount;
long int TotalPixel;
unsigned char index;
unsigned char Value,counter;
unsigned int movebyte,ShiftInLine,drope;
unsigned char SRGB[256][3];  /* standard palatte get at set mode        */
unsigned char ORGB[256][3];  /* original palatte get from file *.pcx    */
unsigned char PRGB[256][3];  /* palatte get from file *.pcx  (ORGB>>2)  */
unsigned char GRGB[256][3];  /* gral palatte from cal. from file pal    */
unsigned char IGRGB[256][3]; /* interupt vector gral palatte from PRGB  */
unsigned char OGRGB[256][3]; /* original gral palatte from cal. from opal */
unsigned char PRRGB[256][3]; /* palatte R get from file *.pcx  (ORGB>>2) */
unsigned char PGRGB[256][3]; /* palatte G get from file *.pcx  (ORGB>>2) */
unsigned char PBRGB[256][3]; /* palatte B get from file *.pcx  (ORGB>>2) */
unsigned char IGRGB[256][3]; /* interupt vector gral palatte            */
unsigned char IGRRGB[256][3];/* interupt vector gral palatte from PRRGB  */
unsigned char IGGRGB[256][3];/* interupt vector gral palatte from PGRGB  */
unsigned char IGBRGB[256][3];/* interupt vector gral palatte from PBRGB  */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------------------*/
/*  Open File variables file .PCX                                    */
/*-------------------------------------------------------------------*/
FILE *ofpcx(name)
char *name;
{
  FILE *fppcx;
  strcat(name,".pcx");          /* define file only .pcx */
  fppcx = fopen(name,"rb");    /* open for read binary file */
  if(fppcx==NULL)
    {
      printf("\nError in Opening File -> %s",name);
      exit(0);
    }
  return fppcx;
}
/*-------------------------------------------------------------------*/
/*  Open File variables file *.txt                                   */
/*-------------------------------------------------------------------*/
FILE *oftxt(name)
char *name;
{
  FILE *fptxt;
  fptxt = fopen(name,"wt");    /* open for write text pallett */
  if(fptxt==NULL)
    {
      printf("\nError in Opening File -> %s",name);
      exit(0);
    }
  return fptxt;
}
/*-------------------------------------------------------------------*/
/*  Open File variables file (*.bty)                                 */
/*-------------------------------------------------------------------*/
FILE *ofbyt(name)
char *name;
{
  FILE *fpbyt;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  fpbyt = fopen(name,"wb");    /* open for write byte pallett */
  if(fpbyt==NULL)
    {
     printf("\nError in Opening File -> %s",name);
     exit(0);
    }
  return fpbyt;
}
/*-------------------------------------------------------------------------*/
/* Get current display mode before run program                             */
/*-------------------------------------------------------------------------*/
unsigned char get_BIOS_display_mode()
{
 _AH = 0x0f;
 geninterrupt(0x10);
 return (_AL);
}
/*-------------------------------------------------------------------------*/
/* Set display mode to mode parameter                                      */
/*-------------------------------------------------------------------------*/
void set_dismode(mode)
int mode;
{
  _AH = 0;
  _AL = mode;
  geninterrupt(0x10);
}
/*-------------------------------------------------------------------------*/
/* Set Color Palette to Color Register from selection type color show */
/*-------------------------------------------------------------------------*/
SetColorPalette()
{
 _DX = PRGB;
 _BX = 0;
 _CX = 256;
 _AX = 0x1012;
 geninterrupt(0x10);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------------------------*/
/* Set Color Palette to Color Register from selection type color show */
/*-------------------------------------------------------------------------*/
SetColorPaletteR()
{
  _DX = PRRGB;
  _BX = 0;
  _CX = 256;
  _AX = 0x1012;
  geninterrupt(0x10);
}
/*-------------------------------------------------------------------------*/
/* Set Color Palette to Color Register from selection type color show */
/*-------------------------------------------------------------------------*/
SetColorPaletteG()
{
  _DX = PGRGB;
  _BX = 0;
  _CX = 256;
  _AX = 0x1012;
  geninterrupt(0x10);
}

/*-------------------------------------------------------------------------*/
/* Set Color Palette to Color Register from selection type color show */
/*-------------------------------------------------------------------------*/
SetColorPaletteB()
{
  _DX = PBRGB;
  _BX = 0;
  _CX = 256;
  _AX = 0x1012;
  geninterrupt(0x10);
}
/*************************************************************************/
/* Set Gral color mode */
/*************************************************************************/
set_gralmode()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  _AH = 0x10;
  _AL = 0x1b;
  _BX = 0;
  _CX = 256;
  geninterrupt(0x10);
}
/*-------------------------------------------------------------------------*/
/* Get intial Color Palette from file (spal.byt) 256 color          */
/*-------------------------------------------------------------------------*/
void GetIntColorPalette()
{
   int i;
   _AH = 0x10;
   _AL = 0x17;
   _BX = 0;
   _CX = 256;
   _DX = SRGB;    /* to file spal.byt,txt */
   geninterrupt(0x10);
}
/*-------------------------------------------------------------------------*/
/* picture convert RGB to gral value                                */
/*-------------------------------------------------------------------------*/
unsigned char pc_RGB_gral(unsigned char valcolor, int seqcolor)
{
   div_t divcolor;
   unsigned char newcolor,valuecolor;
   int intvaluecolor;
   valuecolor = valcolor;
intvaluecolor = valuecolor;
if (seqcolor == 0)
        divcolor = div((intvaluecolor * 30), 100);
   if (seqcolor == 1)
        divcolor = div((intvaluecolor * 59), 100);
   if (seqcolor == 2)
        divcolor = div((intvaluecolor * 11), 100);
   if (divcolor.rem >= 50)
        intvaluecolor = divcolor.quot + 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    else intvaluecolor = divcolor.quot;
    newcolor = intvaluecolor;
return (newcolor);
}
/*********************************************************************/
/* Get Color Palette                                                 */
/*********************************************************************/
void GetColorPalette()
{
   unsigned char VGAColorMap[256][3],VGAPltt;
   int i,j,tem;
   fseek(fpcx,-769,SEEK_END);
   if(fread(&VGAPltt,1,1,fpcx)!=1)
     quit_rtn("Error in reading VGAColor ");
   if (VGAPltt==0x0c)            /*check byte no.769 from EOF = 'och'*/
     if(fread(VGAColorMap,768,1,fpcx)!=1)
          quit_rtn("Error in reading VGAColor ");
     else ; /*  ok  */
   else
     quit_rtn("Error in Format read -- 256 color palette");
   for (i=0;i<=255;i++)
     {
     for (j=0;j<=2;j++)
         {
          PRGB[i][j]  = VGAColorMap[i][j] >> 2;       /* to file pal.txt,byt     */
          ORGB[i][j]  = VGAColorMap[i][j];            /* to file opal.txt,byt    */
          OGRGB[i][j] = pc_RGB_gral(ORGB[i][j],j);  /* to file ogpzl.txt,byt */
          GRGB[i][j]  = pc_RGB_gral(PRGB[i][j],j);  /* to file gpal.txt,byt   */
         }
}
}
/*********************************************************************/
/* Get RGB (select for PRRGB,PGRGB,PBRGB)  */
/*********************************************************************/
void GetRGB()
{
   int i,j;
   for (i=0;i<=255;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        PRRGB[i][0] = PRGB[i][0];
        PRRGB[i][1] = 0;
        PRRGB[i][2] = 0;
        /*   */
        PGRGB[i][0] = 0;
        PGRGB[i][1] = PRGB[i][1];
        PGRGB[i][2] = 0;
        /*   */
        PBRGB[i][0] = 0;
        PBRGB[i][1] = 0;
        PBRGB[i][2] = PRGB[i][2];
    }
}
/*-------------------------------------------------------------------------*/
/* Get Gral Color Palette from file (igpal.byt) 256 color          */
/*-------------------------------------------------------------------------*/
void GetGralColor()
{
  _AH = 0x10;
  _AL = 0x17;
  _BX = 0;
  _CX = 256;
  _DX = IGRGB;  /* to file igpal.txt,byt */
  geninterrupt(0x10);
}
/*-------------------------------------------------------------------------*/
/* Get Gral Color Palette from file (igpalr.byt) 256 color          */
/*-------------------------------------------------------------------------*/
void GetGralColorR()
{
  _AH = 0x10;
  _AL = 0x17;
  _BX = 0;
  _CX = 256;
  _DX = IGRRGB;  /* to file igpal.txt,byt */
  geninterrupt(0x10);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------------------*/
/* Get Gral Color Palette from file (igpalg.byt) 256 color          */
/*-------------------------------------------------------------------*/
void GetGralColorG()
{
  _AH = 0x10;
  _AL = 0x17;
  _BX = 0;
  _CX = 256;
  _DX = IGGRGB;  /* to file igpal.txt,byt */
  geninterrupt(0x10);
}
/*-------------------------------------------------------------------*/
/* Get Gral Color Palette from file (igpalb.byt) 256 color          */
/*-------------------------------------------------------------------*/
void GetGralColorB()
{
  _AH = 0x10;
  _AL = 0x17;
  _BX = 0;
  _CX = 256;
  _DX = IGBRGB;  /* to file igpal.txt,byt */
  geninterrupt(0x10);
}
/*******************************************************************/
/* Decode Palette Value & Writ File Pal   (255color)               */
/*******************************************************************/
void WFpal()
{
 int i,j,calOGRGB,calGRGB;
 unsigned char newOGRGB, newGRGB;
   for (i=0;i<=255;i++)
         {
           calOGRGB = OGRGB[i][0] + OGRGB[i][1] + OGRGB[i][2];
           calGRGB  = GRGB[i][0]  + GRGB[i][1]  + GRGB[i][2];
           newOGRGB = calOGRGB;
           newGRGB  = calGRGB;
           OGRGB[i][0] = newOGRGB;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            OGRGB[i][1] = newOGRGB;
            OGRGB[i][2] = newOGRGB;
            GRGB[i][0] = newGRGB;
            GRGB[i][1] = newGRGB;
            GRGB[i][2] = newGRGB;
            fprintf(fpaltxt,"%03d \t %03d \t %03d \t %03d \n",i,PRGB[i][0],PRGB[i][1],PRGB[i][2]);
            fprintf(fspaltxt,"%03d \t %03d \t %03d \t %03d
\n",i,SRGB[i][0],SRGB[i][1],SRGB[i][2]);
            fprintf(fopaltxt,"%03d \t %03d \t %03d \t %03d
\n",i,ORGB[i][0],ORGB[i][1],ORGB[i][2]);
            fprintf(fgpaltxt,"%03d \t %03d \t %03d \t %03d
\n",i,GRGB[i][0],GRGB[i][1],GRGB[i][2]);
            fprintf(figpaltxt,"%03d \t %03d \t %03d \t %03d
\n",i,IGRGB[i][0],IGRGB[i][1],IGRGB[i][2]);
            fprintf(fogpaltxt,"%03d \t %03d \t %03d \t %03d
\n",i,OGRGB[i][0],OGRGB[i][1],OGRGB[i][2]);
          }
     for (i=0;i<=255;i++)
     for (j=0;j<=2;j++)
         {
          fwrite(&PRGB[i][j],1,1,fpalbyt);
          fwrite(&SRGB[i][j],1,1,fspalbyt);
          fwrite(&ORGB[i][j],1,1,fopalbyt);
          fwrite(&GRGB[i][j],1,1,fgpalbyt);
          fwrite(&IGRGB[i][j],1,1,figpalbyt);
          fwrite(&OGRGB[i][j],1,1,fogpalbyt);
         }
}
/*********************************************************************/
/* Decode Palette Value & Writ File Pal   (255color)          */
/*********************************************************************/
void WFpal1()
{
 int i,j;
    for (i=0;i<=255;i++)
          {
           fprintf(fpalrtxt,"%03d \t %03d \t %03d \t %03d
\n",i,PRRGB[i][0],PRRGB[i][1],PRRGB[i][2]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            fprintf(fpalgtxt,"%03d \t %03d \t %03d \t %03d
\n",i,PGRGB[i][0],PGRGB[i][1],PGRGB[i][2]);
            fprintf(fpalbtxt,"%03d \t %03d \t %03d \t %03d
\n",i,PBRGB[i][0],PBRGB[i][1],PBRGB[i][2]);
            fprintf(figpalrtxt,"%03d \t %03d \t %03d \t
03d\n",i,IGRRGB[i][0],IGRRGB[i][1],IGRRGB[i][2]);
            fprintf(figpalgtxt,"%03d \t %03d \t %03d \t
03d\n",i,IGGRGB[i][0],IGGRGB[i][1],IGGRGB[i][2]);
            fprintf(figpalbtxt,"%03d \t %03d \t %03d \t %03d \n",i,IGBRGB[i][0],IGBRGB[i][1],IGBRGB[i][2]);
          }
  for (i=0;i<=255;i++)
    for (j=0;j<=2;j++)
        {
         fwrite(&PRRGB[i][j],1,1,fpalrbyt);
         fwrite(&PGRGB[i][j],1,1,fpalgbyt);
         fwrite(&PBRGB[i][j],1,1,fpalbbyt);
         fwrite(&IGRRGB[i][j],1,1,figpalrbyt);
         fwrite(&IGGRGB[i][j],1,1,figpalgbyt);
         fwrite(&IGBRGB[i][j],1,1,figpalbbyt);
        }
}
/*************************************************************************/
/* Decode Palette Value & Writ File Piz    (piz index table color)   */
/*************************************************************************/
void WFpiz()
{
 int i,j,x,y;
 x=0; y=0; TotalPixel=0;
 fseek(fpcx,128,SEEK_SET);
 fread(&Value,1,1,fpcx);
 while ( !feof(fpcx) )
  {
   if (Value > 0xc0)
     {
        counter = (Value^0xc0);
        fread(&Value,1,1,fpcx);
        for(i=counter;i>0;i--)
         {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(x<320)
          {
              fprintf(fpiztxt,"%03d \t %03d \t %03d \n",x,y,Value);
              fwrite(&Value,1,1,fpizbyt);
              TotalPixel = TotalPixel + 1;
          }
        x++;
        if(x==Hearder.BytePLine)
          {
           x=0;
           y++;
           if(y==(Hearder.WinDemY2+1))
                return;
          }
        } /* for */
    }
   else
    {
       if(x<320)
         {
          fprintf(fpiztxt,"%03d \t %03d \t %03d \n",x,y,Value);
          fwrite(&Value,1,1,fpizbyt);
          TotalPixel = TotalPixel + 1;
         }
       x++;
       if(x==Hearder.BytePLine)
         {
          x=0;
          y++;
          if(y==(Hearder.WinDemY2+1))
            return;
         }
    }  /* end else */
   fread(&Value,1,1,fpcx);
  }
}                          /* End Load_screen */
/*********************************************************************/
/* CLOSE FILE1 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*********************************************************************/
closefile1()
{
   fclose(fpcx);
   fclose(fpaltxt);
   fclose(fspaltxt);
   fclose(fopaltxt);
   fclose(fgpaltxt);
   fclose(figpaltxt);
   fclose(fogpaltxt);
   fclose(fpiztxt);
   fclose(fpalbyt);
   fclose(fspalbyt);
   fclose(fopalbyt);
   fclose(fgpalbyt);
   fclose(figpalbyt);
   fclose(fogpalbyt);
   fclose(fpizbyt);
}
/*********************************************************************/
/* CLOSE FILE2 */
/*********************************************************************/
closefile2()
{
  fclose(fpalrtxt);
  fclose(fpalgtxt);
  fclose(fpalbtxt);
  fclose(figpalrtxt);
  fclose(figpalgtxt);
  fclose(figpalbtxt);
  fclose(fpalrbyt);
  fclose(fpalgbyt);
  fclose(fpalbbyt);
  fclose(figpalrbyt);
  fclose(figpalgbyt);
  fclose(figpalbbyt);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*********************************************************************/
/* Quit function with print message to user                          */
/*********************************************************************/
quit_rtn(massage)
char *massage;
{
   fclose(fpcx);
   fclose(fpaltxt);
   fclose(fpalrtxt);
   fclose(fpalgtxt);
   fclose(fpalbtxt);
   fclose(fspaltxt);
   fclose(fopaltxt);
   fclose(fgpaltxt);
   fclose(figpaltxt);
   fclose(figpalrtxt);
   fclose(figpalgtxt);
   fclose(figpalbtxt);
   fclose(fogpaltxt);
   fclose(fpiztxt);
   fclose(fpalbyt);
   fclose(fpalrbyt);
   fclose(fpalgbyt);
   fclose(fpalbbyt);
   fclose(fspalbyt);
   fclose(fopalbyt);
   fclose(fgpalbyt);
   fclose(figpalbyt);
   fclose(figpalrbyt);
   fclose(figpalgbyt);
   fclose(figpalbbyt);
   fclose(fogpalbyt);
   fclose(fpizbyt);
   printf("\n%s",massage);
   exit(0);
   return(0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*************************************************************************/
/* MAIN PROGRAM                                                          */
/*************************************************************************/
void main()
{
  int i,j,TypeColorShow,FunKey;
  char napcxf[10];
  clrscr();
  window(11,10,80,25);
  cprintf("Entry file name [10 Char]  :\r\n");
  window(41,10,80,25);
  gets(napcxf);
  if (napcxf[0]!='\0')
    {
      fpcx       = ofpcx(napcxf);
      fpaltxt    = oftxt("pal.txt");
      fspaltxt   = oftxt("spal.txt");
      fgpaltxt   = oftxt("gpal.txt");
      fopaltxt   = oftxt("opal.txt");
      fogpaltxt  = oftxt("ogpal.txt");
      figpaltxt  = oftxt("igpal.txt");
      fpiztxt    = oftxt("piz.txt");
      fpalbyt    = ofbyt("pal.byt");
      fopalbyt   = ofbyt("opal.byt");
      fspalbyt   = ofbyt("spal.byt");
      fgpalbyt   = ofbyt("gpal.byt");
      figpalbyt  = ofbyt("igpal.byt");
      fogpalbyt  = ofbyt("ogpal.byt");
      fpizbyt    = ofbyt("piz.byt");
      index      = get_BIOS_display_mode();  /* for get current mode */
      fread(&Hearder,70,1,fpcx);             /*read hearder file */
      if(Hearder.Password != 0x0a)
         quit_rtn("File not .PCX format.");
      if (Hearder.BPP==8)
        {
         set_dismode(0x13);
         printf("please wait");
         GetIntColorPalette();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        GetColorPalette();
        SetColorPalette();
        set_gralmode();
        GetGralColor();
        WFpal();
        WFpiz();
        closefile1();
        fpalrtxt     = oftxt("palr.txt");
        fpalgtxt     = oftxt("palg.txt");
        fpalbtxt     = oftxt("palb.txt");
        figpalrtxt   = oftxt("igpalr.txt");
        figpalgtxt   = oftxt("igpalg.txt");
        figpalbtxt   = oftxt("igpalb.txt");
        fpalrbyt     = ofbyt("palr.byt");
        fpalgbyt     = ofbyt("palg.byt");
        fpalbbyt     = ofbyt("palb.byt");
        figpalrbyt   = ofbyt("igpalr.byt");
        figpalgbyt   = ofbyt("igpalg.byt");
        figpalbbyt   = ofbyt("igpalb.byt");
        GetRGB();
        SetColorPaletteR();
        set_gralmode();
        GetGralColorR();
        SetColorPaletteG();
        set_gralmode();
        GetGralColorG();
        SetColorPaletteB();
        set_gralmode();
        GetGralColorB();
        WFpal1();
        closefile2();
        set_dismode(index);
        quit_rtn("End Program");
    }
else
    {
        printf("OK. END PGM");
        quit_rtn("End Program");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
        }
    else exit(0);
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------------------------*/
/*  detect edge                                                            */
/*  writen at 11/08/1995                                                   */
/*  by kingkarn wongwipaporn                                               */
/*-------------------------------------------------------------------------
*/
/*-------------------------------------------------------------------------
*/
/* declare varialbe for use gobal */
/*-------------------------------------------------------------------------
*/
#define   VALID_MIN         0
#define   VALID_MAX        255
#define   ROW              200
#define   COLUMN           320
#define   MAXROW           200
#define   MAXCOL           320
/*-------------------------------------------------------------------------
*/
#include <alloc.h>
#include <dos.h>
#include <graphics.h>
#include <io.h>
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "utility.c"
float COS[ROW],SIN[ROW];
/*-------------------------------------------------------------------------*/
/*  FUNCTION */
/*-------------------------------------------------------------------------*/
void display_screen();
int  select_menu();
void processloop(int menu_number);
void MASKOPR(char *fin, char *fout, int *next,int levelpal,int gray,int threhold,
             int N1, int M1, int N2, int M2, char typemask);
BYTE SOBELEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                int a6,int a7, int a8);
BYTE PREWITTEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8);
BYTE KIRSCHEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8);
BYTE LAPLACIANEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8);
BYTE POINTDETECT(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8);
BYTE LINEDETECT(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8);
BYTE LEVELEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8, int levelpal);
BYTE HIGHLOWEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8);
void RANGEEDGE(char *fin, char *fout, int *next,int levelpal,int gray,int threhold,
            int N1, int M1, int N2, int M2,char typemask);
void HOUGH(char *fin, char *fout, int *next,int levelpal,int gray,
            int N1, int N2, int n, int m,char typemask);
void EDGFOLLOW(char *fin, char *fout,int *next,int levelpal,int gray,
            int N1, int N2, int n, int m,char typemask);
void HOUGH(char *fin, char *fout, int *next,int levelpal,int gray,
            int N1, int N2, int n, int m,char typemask);
void IHOUGH(char *fin, char *fout, int *next,int levelpal,int gray,
            int N1, int N2, int n, int m,char typemask);
void PrcEDGEFOLLOW(char *fin, char *fout,int ka,int la,int T1,int T, int a);
void cnvidx_col(char *fpiz, char *fpal, char *fcpiz, char modes);
/*-------------------------------------------------------------------------*/
/* MAIN */
/*-------------------------------------------------------------------------*/
void main()
{
 int menu_number;
 menu_number = 0;
 // ----------select menu for calculate edge----------
 while (menu_number != 15)
  {
   display_screen();              /* display screen */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    menu_number = select_menu();   /* select menu# */
    if ((menu_number > 0) && (menu_number < 15))
        {
         processloop(menu_number);
        }
   }
}
/*-------------------------------------------------------------------------
*/
/*  Display_Screen MENU :  */
/*-------------------------------------------------------------------------
*/
void display_screen()
{
  window(1,1,80,25);
  clrscr();
  textcolor(12);
  window(10,2,80,25);
  cprintf(" 8.Show Edge Detection by select piz & pal file..(EDGE.EXE)   \r\n");
  textcolor(15);
  window(15,4,80,25);
  cprintf(" 1. Sobel Edge Detection..... \r\n");
  cprintf(" 2. Prewitt Edge Detection..... \r\n");
  cprintf(" 3. Kirsch Edge Detection..... \r\n");
  cprintf(" 4. Laplacian Edge Detection..... \r\n");
  cprintf(" 5. Range Edge Detection..... \r\n");
  cprintf(" 6. High-Low Pass Edge.Detection..... \r\n");
  cprintf(" 7. ..... \r\n");
  cprintf(" 8. ..... \r\n");
  cprintf(" 9..... \r\n");
  cprintf("10. ..... \r\n");
  cprintf("11. ..... \r\n");
  cprintf("12. Level Change Edge Detection..... \r\n");
  cprintf("13. Point Detection..... \r\n");
  cprintf("14. Line Detection..... \r\n");
  cprintf("15. GOTO MAIN MENU..... \r\n");
  cprintf("\r\n");
  cprintf("   Select number ->:                \r\n");   /* row=23,col=32) */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  window(1,1,80,25);
}
/*-------------------------------------------------------------------------
*/
/*  SELECT_MENU :  select menu number (menu#) */
/*-------------------------------------------------------------------------
*/
int select_menu()
{
 /*... v a r i a b l e ...*/
 int select_number;
 /*.....................*/
 window(36,20,37,20);
 do {
   printf(" ");
   scanf("%2d",&select_number);
 } while(select_number < 1 || select_number > 15);
window(1,1,80,25);
return select_number;
}
/*-------------------------------------------------------------------------*/
/* process loop */
/*-------------------------------------------------------------------------*/
void processloop(int menu_number)
{
 int gray;
 char Fname[20],Fname1[20],Fname2[20],Fname3[20];
 char *fi,*fo,*fp,*fc,*buftext;
 char gmode,glpal,*gavgfil,*ggray,*pbuf;
 unsigned char *thre;
 int  *next, curmode,levelpal,threhold;
  do
    {
      clrscr();
      window(8,5,80,25);
      textcolor(12);
      cprintf(" 8.Show Edge Detection by select piz & pal file..(EDGE.EXE)    \r\n");
      textcolor(15);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    window(5,8,80,25);
    cprintf("FILE NAME INEDEX COLOR PIZ (20 char)____:");
    gets(Fname);
    gets(Fname);
    window(5,9,80,25);
    cprintf("FILE NAME COLOR PIZ (20 char)__________:");
    gets(Fname1);
    window(5,10,80,25);
    cprintf("FILE NAME PAL (20 char)________________:");
    gets(Fname2);
    window(5,11,80,25);
    cprintf("FILE NAME OUTPUT EDGE PIZ (20 char)_____:");
    gets(Fname3);
    window(5,12,80,25);
    cprintf("SELECT MODE PAL (R/G/B)________________:");
    gmode = getche();
    window(5,13,80,25);
    cprintf("SELECT LEVEL PAL (1=2/2=16/3=64/4=256)__:");
    glpal = getchar();
    window(5,14,80,25);
    cprintf("SELECT SHOW GRAY/COLOR (1/0)___________:");
    *ggray = getchar();
    *ggray = getchar();
    gray = atoi(ggray);
    window(5,15,80,25);
    cprintf("ENTRY THREHOLD LEVEL___________________:");
    gets(thre);
    gets(thre);
    threhold = atoi(thre);
if ((Fname[0]  == '\0') || (Fname3[0] == '\0'))
        *next = F10;
    else
      {
        if ((gmode == 'R') || (gmode == 'G') || (gmode == 'B') || (gmode == '\r'))
          {
              if ((glpal == '1') || (glpal == '2') || (glpal == '3') || (glpal == '4'))
                {
                    if ((gray == 0) || (gray == 1))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
{
  curmode = get_BIOS_display_mode();
  levelpal = convert_level_pal(glpal);
  fi  = &Fname[0];
  fo  = &Fname3[0];
  fc  = &Fname1[0];
  if ((Fname2[0] == '\0') && (Fname1[0] == '\0') &&
       (gmode == 'r')) fc = &Fname[0];
      else { fp  = &Fname2[0];
             fc  = &Fname1[0];
             cnvidx_col(fi,fp,fc,gmode); }
  switch(menu_number)
     {
     case 1  : MASKOPR(fc, fo, next, levelpal,gray,threhold,
                       0,0,MAXROW,MAXCOL,'S'); //Sobel
               break;
     case 2  : MASKOPR(fc, fo, next, levelpal,gray,threhold,
                       0,0,MAXROW,MAXCOL,'P'); //Prewill
               break;
     case 3  : MASKOPR(fc, fo, next, levelpal,gray,threhold,
                       0,0,MAXROW,MAXCOL,'K'); //Kirsch
               break;
     case 4  : MASKOPR(fc, fo, next, levelpal,gray,threhold,
                       0,0,MAXROW,MAXCOL,'L'); //Lapacian
               break;
     case 5  : RANGEEDGE(fc, fo, next, levelpal,gray,threhold,
                       0,0,MAXROW,MAXCOL,'R'); //Range
               break;
     case 6  : MASKOPR(fc, fo, next, levelpal,gray,threhold,
                       0,0,MAXROW,MAXCOL,'H'); //High Pass
               break;
               break;
     case 12 : MASKOPR(fc, fo, next, levelpal,gray,threhold,
                       0,0,MAXROW,MAXCOL,'V');
               break;
     case 13 : MASKOPR(fc, fo, next, levelpal,gray,threhold,
                       0,0,MAXROW,MAXCOL,'O');
               break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                case 14 : MASKOPR(fc, fo, next, levelpal,gray,threhold,
                                                  0,0,MAXROW,MAXCOL,'I');
                                          break;
                                default : break;          /* menu# = oth --> select again */
                                }
                            set_dismode(curmode);
                        }
                    }
                }
            } while (*next != F10);
}
/*-------------------------------------------------------------------------*/
/*  .....MASK OPERATION.....        */
/*-------------------------------------------------------------------------*/
void MASKOPR(char *fin, char *fout, int *next,int levelpal,int gray,int threhold,
             int N1, int M1, int N2, int M2, char typemask)

/* Subroutine for Sobel edge detector
  *fin: source buffer
  *fout: destination buffer
  a,b: input and output buffers
  N1=ROW,M1=COLUMN: start coordinates source buffer
  N2=ROW,M2=COLUMN: end  coordinates source buffer */
{
   FILE *fr, *fw1;
   int  i, j, k,l,ii, jj, cn,cg,cc;
   int a0,a1,a2,a3,a4,a5,a6,a7,a8,b0,b1,b2,b3,b4,b5,b6,b7,b8; // input and output buffers
   BYTE  *pictin;       // array save read picture in memory
   BYTE  *pictout;      // array save write picture in memory
   BYTE  c;             // for get and put to file
   unsigned int ar,count;
/*  *****  remart for not run process call **** */
   window(5,19,80,25);
   cprintf("___Please wait a minute____");
    // ----- allocate memory for read file and write file ------
        if( (pictin = (BYTE *) malloc(MAXCOL*MAXROW)) == NULL)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        printf("Not enough memory to allocate buffer of read file");
        exit(1);
    }
    if( (pictout = (BYTE *) malloc(MAXCOL*MAXROW)) == NULL)
    {
        printf("Not enough memory to allocate buffer of write file");
        exit(1);
    }

// ------ read file to malloced array -----
    fr  = openfile(fin, "rb");
    for (j=0; j<MAXROW; j++)
        for (i=0; i<MAXCOL; i++)
        {
            ar = (unsigned) j * MAXCOL + (unsigned) i;
            c  = getc(fr);
            *(pictin + ar) = c;
        }
    fclose(fr);

// ------ process in memory -------
    for(i=N1+1; i<N2-1; i++)     /* ROW */
      for(j=M1+1; j<M2-1; j++)    /* COLUMN */
            {
                                  /*  4  3  2 */
                                  /*  5  0  1 */
                                  /*  6  7  8 */
             a0 = Getc_malloc_array(j  ,i  , pictin);
             a1 = Getc_malloc_array(j+1,i  , pictin);
             a2 = Getc_malloc_array(j+1,i-1, pictin);
             a3 = Getc_malloc_array(j  ,i-1, pictin);
             a4 = Getc_malloc_array(j-1,i-1, pictin);
             a5 = Getc_malloc_array(j-1,i  , pictin);
             a6 = Getc_malloc_array(j-1,i+1, pictin);
             a7 = Getc_malloc_array(j  ,i+1, pictin);
             a8 = Getc_malloc_array(j+1,i+1, pictin);
             if (typemask == 'S') c = SOBELEDGE(a0,a1,a2,a3,a4,a5,a6,a7,a8);
             if (typemask == 'P') c = PREWITTEDGE(a0,a1,a2,a3,a4,a5,a6,a7,a8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (typemask == 'K') c = KIRSCHEDGE(a0,a1,a2,a3,a4,a5,a6,a7,a8);
                if (typemask == 'L') c = LAPLACIANEDGE(a0,a1,a2,a3,a4,a5,a6,a7,a8);
                if (typemask == 'H') c = HIGHLOWEDGE(a0,a1,a2,a3,a4,a5,a6,a7,a8);
                if (typemask == 'O') c = POINTDETECT(a0,a1,a2,a3,a4,a5,a6,a7,a8);
                if (typemask == 'I') c = LINEDETECT(a0,a1,a2,a3,a4,a5,a6,a7,a8);
                if (typemask == 'V') c = LEVELEDGE(a0,a1,a2,a3,a4,a5,a6,a7,a8,levelpal);
                if(c>threhold) c=1;
                 else c=0;
                ar = (unsigned) i * MAXCOL + (unsigned) j;
                *(pictout + ar) = (unsigned char) c;
               if ((c != 0) && (c != 1)) c=0;
               }
        free(pictin);
    // ----- write file from malloced array ------
        fw1 = openfile(fout, "wb");
        for (j=0; j<MAXROW; j++)
           for (i=0; i<MAXCOL; i++)
           {
                ar = (unsigned) j * MAXCOL + (unsigned) i;
                c  = *(pictout + ar);
                fwrite(&c,1,1,fw1);
           }
        fclose(fw1);
        free(pictout);
/*  */ //  *****  end remart for not run process call ****
        open_graphVGAHI(gray);
   do
   {
        picture1(fin,levelpal,0,0);
        setfillstyle(0,0);
        bar(50,205,50+25*8,205+8);
        outtextxy(50, 205, fin); outtextxy(140,205," 16 Level");
        picture1(fout,2,320,0);
        outtextxy(370, 205, fout); outtextxy(480,205," 2 Level");
        picture2(fout,2,321,221);
        outtextxy(370, 425, fout); outtextxy(480,425," INVERSE");
        outtextxy(640-(32*8), 470, "          EDGE        F10=END");
        color_scal(310,225);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        *next = getckey();
    } while (*next != F10);
    closegraph();
}

/*-------------------------------------------------------------------------*/
/*  .....SOBEL EDGE.....        */
/*-------------------------------------------------------------------------*/
BYTE SOBELEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8)
{
    int cc;
    BYTE  c;            // for get and put to file
          /* 4 3 2 */ /* -1  0  1 */ /*  1  2  1 */
          /* 5 0 1 */ /* -2  0  2 */ /*  0  0  0 */
          /* 6 7 8 */ /* -1  0  1 */ /* -1 -2 -1 */
    cc = -1*(int)a4-2*(int)a5-1*(int)a6;
    cc+= +1*(int)a2+2*(int)a1+1*(int)a8;
    c = abs(cc);
    cc = +1*(int)a4+2*(int)a3+1*(int)a2;
    cc+= -1*(int)a6-2*(int)a7-1*(int)a8;
    c+= abs(cc);
    return c;

}
/*-------------------------------------------------------------------------*/
/*  .......PREWITT EDGE.....        */
/*-------------------------------------------------------------------------*/
BYTE PREWITTEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8)
{
    int cc;
    BYTE  c;  // for get and put to file
          /* 4 3 2 */ /* -1  0  1 */ /*  1  1  1 */
          /* 5 0 1 */ /* -1  0  1 */ /*  0  0  0 */
          /* 6 7 8 */ /* -1  0  1 */ /* -1 -1 -1 */
    cc = -1*(int)a4-1*(int)a5-1*(int)a6;
    cc+= +1*(int)a2+1*(int)a1+1*(int)a8;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    c  = abs(cc);
    cc = +1*(int)a4+1*(int)a3+1*(int)a2;
    cc+= -1*(int)a6-1*(int)a7-1*(int)a8;
    c += abs(cc);
    return c;
}
/*-------------------------------------------------------------------------*/
/*  .....KIRSCH EDGE..... */
/*-------------------------------------------------------------------------*/
BYTE KIRSCHEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8)
{
    int cc;
    BYTE  c;          // for get and put to file
                /* 0 degrees*/ /*45 degress*/ /*90 degrees*/ /*135degress*/
    /* 4 3 2 */ /* -1  0  1 */ /*  1  1  1 */ /*  0  1  1 */ /*  1  1  0 */
    /* 5 0 1 */ /* -1  0  1 */ /*  0  0  0 */ /* -1  0  1 */ /*  1  0 -1 */
    /* 6 7 8 */ /* -1  0  1 */ /* -1 -1 -1 */ /* -1 -1  0 */ /*  0 -1 -1 */
    cc = -1*(int)a4-1*(int)a5-1*(int)a6;
    cc+= +1*(int)a2+1*(int)a1+1*(int)a8;
    c  = abs(cc);
    cc = +1*(int)a4+1*(int)a3+1*(int)a2;
    cc+= -1*(int)a6-1*(int)a7-1*(int)a8;
    c += abs(cc);
    cc = +1*(int)a3+1*(int)a2+1*(int)a1;
    cc+= -1*(int)a5-1*(int)a6-1*(int)a7;
    c += abs(cc);
    cc = +1*(int)a3+1*(int)a4+1*(int)a5;
    cc+= -1*(int)a7-1*(int)a8-1*(int)a1;
    c += abs(cc);
    return c;
}
/*-------------------------------------------------------------------------*/
/*  .....LAPLACIAN EDGE..... */
/*-------------------------------------------------------------------------*/
BYTE LAPLACIANEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int cc;
    BYTE  c;            // for get and put to file
    /* 4 3 2 */ /*  0 -1  0 */ /* -1 -1 -1 */
    /* 5 0 1 */ /* -1  4 -1 */ /* -1  8 -1 */
    /* 6 7 8 */ /*  0 -1  0 */ /* -1 -1 -1 */
    cc = +0*(int)a4-1*(int)a3+0*(int)a2;
    cc+= -1*(int)a5+4*(int)a0-1*(int)a1;
    cc+= +0*(int)a6-1*(int)a7+0*(int)a8;
    c = abs(cc);
    cc = -1*(int)a4-1*(int)a3-1*(int)a2;
    cc+= -1*(int)a5+8*(int)a0-1*(int)a1;
    cc+= -1*(int)a6-1*(int)a7-1*(int)a8;
    c+= abs(cc);
//  cc =-4*(int)a0+1*(int)a5+1*(int)a1;
//  cc+= +1*(int)a3+1*(int)a7;
//  c  = abs(cc);
    return c;
}
/*-------------------------------------------------------------------------*/
/*   .....POINT DETECTION..... */
/*-------------------------------------------------------------------------*/
BYTE POINTDETECT(int a0,int a1, int a2,int a3,int a4, int a5,
                 int a6,int a7, int a8)
{
    int cc;
    BYTE  c;            // for get and put to file
          /* 4 3 2 */  /* -1 -1 -1 */
          /* 5 0 1 */  /* -1  8 -1 */
          /* 6 7 8 */  /* -1 -1  1 */
    cc = -1*(int)a4-1*(int)a3-1*(int)a2;
    cc+= -1*(int)a5+8*(int)a0-1*(int)a1;
    cc+= -1*(int)a6-1*(int)a7-1*(int)a8;
    c = abs(cc);
    return c;
}
/*-------------------------------------------------------------------------*/
/*   .....LINE DETECTION..... */
/*-------------------------------------------------------------------------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BYTE LINEDETECT(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8)
{
   int cc;
   BYTE  c;          // for get and put to file
                /* 0 degrees*/ /*45 degress*/ /*90 degrees*/ /*135degress*/
   /* 4 3 2 */ /* -1 -1 -1 */ /* -1 -1  2 */ /* -1  2 -1 */ /*  2 -1 -1 */
   /* 5 0 1 */ /*  2  2  2 */ /* -1  2 -1 */ /* -1  2 -1 */ /* -1  2 -1 */
   /* 6 7 8 */ /* -1 -1 -1 */ /*  2 -1 -1 */ /* -1  2 -1 */ /* -1 -1  2 */
   cc = -1*(int)a4-1*(int)a3-1*(int)a2;
   cc+= +2*(int)a5+2*(int)a0+2*(int)a1;
   cc+= -1*(int)a6-1*(int)a7-1*(int)a8;
   c = abs(cc);
   cc = -1*(int)a4-1*(int)a3+2*(int)a2;
   cc+= -1*(int)a5+2*(int)a0-1*(int)a1;
   cc+= +2*(int)a6-1*(int)a7-1*(int)a8;
   c += abs(cc);
   cc = -1*(int)a4+2*(int)a3-1*(int)a2;
   cc+= -1*(int)a5+2*(int)a0-1*(int)a1;
   cc+= -1*(int)a6+2*(int)a7-1*(int)a8;
   c += abs(cc);
   cc = +2*(int)a4-1*(int)a3-1*(int)a2;
   cc+= -1*(int)a5+2*(int)a0-1*(int)a1;
   cc+= -1*(int)a6-1*(int)a7+2*(int)a8;
   c += abs(cc);
   return c;
}

/*-------------------------------------------------------------------------*/
/*  .....LEVEL CHANGE EDGE DETECTION..... */
/*-------------------------------------------------------------------------*/
BYTE LEVELEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                int a6,int a7, int a8, int levelpal)
{
   int cc;
   BYTE  c;          // for get and put to file
        /* 4 3 2 */
        /* 5 0 1 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            /* 6 7 8 */
  if ((a0 != a1) || (a0 != a2) || (a0 != a3) || (a0 != a4) ||
      ( a0 != a5) || (a0 != a6) || (a0 != a7) || (a0 != a8) )
      {
          cc = levelpal + 1;
          c = abs(cc);
      }
    return c;
}
/*-------------------------------------------------------------------------*/
/*  .....HIGH-LOW PALL FILLTER... */  //not finish
/*-------------------------------------------------------------------------*/
BYTE HIGHLOWEDGE(int a0,int a1, int a2,int a3,int a4, int a5,
                 int a6,int a7, int a8)
{
    int cc;
    BYTE  c;          // for get and put to file
    /* 4 3 2 */ /* 0   0.2  0   */ /* -1 -1 -1 */
    /* 5 0 1 */ /* 0.2  0.2  0.2 */ /* -1  8 -1 */
    /* 6 7 8 */ /* 0   0.2  0   */ /* -1 -1 -1 */
    /* 4 3 2 */ /*  0   -1   0   */ /*  1  1  1 */
    /* 5 0 1 */ /* -1    4  -1   */ /*  1  1  1 */
    /* 6 7 8 */ /*  0   -1   0   */ /*  1  1  1 */
    cc = +0*(int)a4-1*(int)a3+0*(int)a2;
    cc+= -1*(int)a5+4*(int)a0-1*(int)a1;
    cc+= +0*(int)a6-1*(int)a7+0*(int)a8;
    c = abs(cc);
    cc = +1*(int)a4+1*(int)a3+1*(int)a2;
    cc+= +1*(int)a5+1*(int)a0+1*(int)a1;
    cc+= +1*(int)a6+1*(int)a7+1*(int)a8;
    c+= abs(cc);
    return c;
}
/*-------------------------------------------------------------------------*/
/*  .....RANGE EDGE.....*/
/*-------------------------------------------------------------------------*/
void RANGEEDGE(char *fin, char *fout, int *next,int levelpal,int gray,int threhold,
               int N1, int M1, int N2, int M2,char typemask)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Subroutine for Range edge detector
  *fin: source buffer
  *fout: destination buffer
  a,b: input and output buffers
  NW,MW:Window size
  N1=ROW,M1=COLUMN: start coordinates source buffer
  N2=ROW,M2=COLUMN: end   coordinates source buffer */
{
   FILE *fr, *fw1;
   int  i, j, k,l,NW,MW,NW2,MW2,smax,smin,arange;
   int a0,a1,a2,a3,a4,a5,a6,a7,a8,b0,b1,b2,b3,b4,b5,b6,b7,b8; // input and output buffers
   char  rm[1],*cm[1];
   BYTE  *pictin;       // array save read picture in memory
   BYTE  *pictout;      // array save write picture in memory
   BYTE  c;             // for get and put to file
   unsigned int ar,count;
/*  *****  remark for not run process call **** */
   window(5,16,80,25);
   cprintf("ENTRY RANGE ROW MASK (>0 AND <9)________:");
   gets(rm);
   window(5,17,80,25);
   cprintf("ENTRY RANGE COLUMN MASK (>0 AND <9)_____:");
   gets(cm);
   NW = atoi(rm);
   MW = atoi(cm);
   window(5,19,80,25);
   cprintf("___Please wait a minute____");
    // ----- allocate memory for read file and write file ------
        if( (pictin = (BYTE *) malloc(MAXCOL*MAXROW)) == NULL)
        {
           printf("Not enough memory to allocate buffer of read file");
           exit(1);
        }
        if( (pictout = (BYTE *) malloc(MAXCOL*MAXROW)) == NULL)
        {
           printf("Not enough memory to allocate buffer of write file");
           exit(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

// ------ read file to malloced array -----
    fr  = openfile(fin, "rb");
    for (j=0; j<MAXROW; j++)
       for (i=0; i<MAXCOL; i++)
       {
            ar = (unsigned) j * MAXCOL + (unsigned) i;
            c  = getc(fr);
            *(pictin + ar) = c;
       }
    fclose(fr);

// ------ process in memory -------
    NW2=NW/2; MW2=MW/2;
    for(k=N1+NW2; k<N2-NW2; k++)     /* ROW */
      for(l=M1+MW2; l<M2-MW2; l++)     /* COLUMN */
            {
                                  /*  4  3  2 */
                                  /*  5  0  1 */
                                  /*  6  7  8 */
             a0 = Getc_malloc_array(l  ,k  , pictin);
             a1 = Getc_malloc_array(l+1,k  , pictin);
             a2 = Getc_malloc_array(l+1,k-1, pictin);
             a3 = Getc_malloc_array(l  ,k-1, pictin);
             a4 = Getc_malloc_array(l-1,k-1, pictin);
             a5 = Getc_malloc_array(l-1,k  , pictin);
             a6 = Getc_malloc_array(l-1,k+1, pictin);
             a7 = Getc_malloc_array(l  ,k+1, pictin);
             a8 = Getc_malloc_array(l+1,k+1, pictin);
             smax = (int)a0;
             smin = smax;
             for(i=0; i<NW; i++)    //ROW
               for(j=0; j<MW; j++)    //COLUMN
                 {
                     arange = Getc_malloc_array(l+j-MW2,k+i-NW2, pictin);
                     if(smin>arange) smin=arange;
                     if(smax<arange) smax=arange;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                ar = (unsigned) k * MAXCOL + (unsigned) l;
                *(pictout + ar) = (unsigned char) (smax-smin);
               }
        free(pictin);
   // ----- write file from malloced array ------
        fw1 = openfile(fout, "wb");
        for (j=0; j<MAXROW; j++)
           for (i=0; i<MAXCOL; i++)
           {
                ar = (unsigned) j * MAXCOL + (unsigned) i;
                c  = *(pictout + ar);
                if ((c != 0)) c=1;
                putc(c,fw1);
           }
        fclose(fw1);
        free(pictout);
/* */ //  ***** end remart for not run process call ****
        open_graphVGAHI(gray);
   do
   {
        picture1(fin,levelpal,0,0);
        setfillstyle(0,0);
        bar(50,205,50+25*8,205+8);
        outtextxy(50, 205, fin); outtextxy(140,205," 16 Level");
        picture1(fout,2,320,0);
        outtextxy(370, 205, fout); outtextxy(480,205," 2 Level");
        picture2(fout,2,321,221);
        outtextxy(370, 425, fout); outtextxy(480,425," INVERSE");
        outtextxy(640-(32*8), 470, "          EDGE        F10=END");
        color_scal(310,225);
        *next = getckey();
   } while (*next != F10);
   closegraph();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------------------------*/
/*  SHAPE.C                                                                */
/*  shape representation                                                   */
/*  writen at 28/08/1995                                                   */
/*  by kingkarn wongwipaporn                                               */
/*-------------------------------------------------------------------------
*/
/* declare varialbe for use gobal */
/*-------------------------------------------------------------------------
*/
#define  VALID_MIN        0
#define  VALID_MAX        255
#define  ROW              200
#define  COLUMN           320
#define  MAXROW           200
#define  MAXCOL           320
/*-------------------------------------------------------------------------
*/
#include <alloc.h>
#include <dos.h>
#include <graphics.h>
#include <io.h>
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "utility.c"
/*-------------------------------------------------------------------------*/
/*  FUNCTION */
/*-------------------------------------------------------------------------*/
void display_screen();
int  select_menu();
void processloop(int menu_number);
void SIMPLETHIN(char *fi1,char *fo, int *next,int gray,
                int N1, int M1, int N2, int M2);
void TWOPASSTHIN(char *fi,char *fo, int *next,int gray,
                int N1, int M1, int N2, int M2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-------------------------------------------------------------------------*/
/* MAIN */
/*-------------------------------------------------------------------------*/
void main()
{
  int menu_number;
  menu_number = 0;
  // ----------select menu for calculate edge----------
  while (menu_number != 15)
    {
      display_screen();              /* display screen */
      menu_number = select_menu();   /* select menu# */
      if ((menu_number > 0) && (menu_number < 15))
           {
           processloop(menu_number);
           }
    }
}
/*-------------------------------------------------------------------------
*/
/*  Display_Screen MENU :  */
/*-------------------------------------------------------------------------
*/
void display_screen()
{
   window(1,1,80,25);
   clrscr();
   textcolor(12);
   window(1,2,80,25);
   cprintf(" 9.Show Shape Representation by select piz & pal file.....(SHAPE.EXE)\r\n");
   textcolor(15);
   window(15,4,80,25);
   cprintf(" 1. Simple Thinning..... \r\n");
   cprintf(" 2. Two-Pass Thinning..... \r\n");
   cprintf(" 3. ..... ..... \r\n");
   cprintf(" 4. ..... \r\n");
   cprintf(" 5. ..... \r\n");
   cprintf(" 6. ..... \r\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    cprintf(" 7. ..... \r\n");
    cprintf(" 8. ..... \r\n");
    cprintf(" 9. ..... \r\n");
    cprintf("10. ..... \r\n");
    cprintf("11. ..... \r\n");
    cprintf("12. ..... \r\n");
    cprintf("13. ..... \r\n");
    cprintf("14. ..... \r\n");
    cprintf("15. GOTO MAIN MENU..... \r\n");
    cprintf("\r\n");
    cprintf("    Select number ->:                \r\n");    /* row=23,col=32) */
    window(1,1,80,25);
}
/*-------------------------------------------------------------------------
*/
/*  SELECT_MENU :  select menu number (menu#) */
/*-------------------------------------------------------------------------
*/
int select_menu()
{
  /*... v a r i a b l e ...*/
  int select_number;
  /*.....*/
  window(36,20,37,20);
  do {
    printf(" ");
    scanf("%2d",&select_number);
  } while(select_number < 1 || select_number > 15);
 window(1,1,80,25);
 return select_number;
}
/*-------------------------------------------------------------------------*/
/* process loop */
/*-------------------------------------------------------------------------*/
void processloop(int menu_number)
{
  int gray;
  char Fnamei[20],Fnameo[20];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char *fi,*fo;
char *ggray;
int  *next, curmode,levelpal;
 do
  {
    clrscr();
    window(1,5,80,25);
    textcolor(12);
    cprintf(" 9.Show Shape Representation by select piz & pal file.....(SHAPE.EXE)\r\n");
    textcolor(15);
    window(5,8,80,25);
    cprintf("FILE NAME1 BINARY INPUT PICTURE (20 char)__:");
    gets(Fnamei);
    gets(Fnamei);
    window(5,9,80,25);
    cprintf("FILE NAME3 BINARY OUTPUT PICTURE (20 char)_:");
    gets(Fnameo);
    window(5,10,80,25);
    cprintf("SELECT SHOW GRAY/COLOR (1/0)_____:");
    *ggray = getchar();
    gray = atoi(ggray);
    window(5,11,80,25);
    cprintf("___Please wait a minute____");
   if ((Fnamei[0]  == '\0') || (Fnameo[0]  == '\0'))
        *next = F10;
    else
        {
         if ((gray == 0) || (gray == 1))
            {
                curmode = get_BIOS_display_mode();
                fi  = &Fnamei[0];
                fo  = &Fnameo[0];
                switch(menu_number)
                 {                                                    \
                   case 1  : SIMPLETHIN(fi,fo,next,gray,1,1,MAXROW,MAXCOL); //simple
thinning
                             break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    case 2  : TWOPASSTHIN(fi,fo,next,gray,1,1,MAXROW,MAXCOL); //two
pall thinning
                              break;
                    case 4  :
                              break;
                    default : break;          /* menu# = oth --> select again */
                   }
                 set_dismode(curmode);
               }
             else *next = F10;
           }
      } while (*next != F10);
}
/*-------------------------------------------------------------------------*/
/*   .....SIMPLE THINNING.....        */
/*-------------------------------------------------------------------------*/
void SIMPLETHIN(char *fi,char *fo, int *next,int gray,
                int N1, int M1, int N2, int M2)

/* Subroutine for Calcurate Binary picture
   *fi: source buffer
   *fo : destination buffer
   N1=1,M1=1: start coordinates source buffer
   N2=ROW,M2=COLUMN: end  coordinates source buffer */
{
    FILE *fr,*fw;
    int  k,l,i,j,count=0,y[9],trans=0,m,OK=1;
    int av1,av2,a0,a1,a2,a3,a4,a5,a6,a7,a8; // input buffers
    BYTE  *pictin;       // array save read picture in memory1
    BYTE  c;             // for get and put to file
    unsigned int ar;
/*  *****  remart for not run process call **** */
     // ----- allocate memory for read file and write file ------
          if( (pictin = (BYTE *) malloc(MAXCOL*MAXROW)) == NULL)
          {
             printf("Not enough memory to allocate buffer of read file");
             exit(1);
          }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ------ read file to malloced array ----- read 1
    fr  = openfile(fi, "rb");
    for (j=0; j<MAXROW; j++)
       for (i=0; i<MAXCOL; i++)
       {
            ar = (unsigned) j * MAXCOL + (unsigned) i;
            c  = getc(fr);
            *(pictin + ar) = c;
       }
    fclose(fr);
// ------ process in memory -------
    do
     {
      OK=1;
      for (k=N1+1;k<N2-1;k++)
        for (l=M1+1;l<M2-1;l++)
         {
            av1 = Getc_malloc_array(l  ,k  , pictin);
            if (av1==1)
              {  /* count number of 1s in 3 x 3 windows */
               count = 0;
               for (i=-1;i<=1;i++)
                   for (j=-1;j<=1;j++)
                     {
                      av2 = Getc_malloc_array(l+j,k+i, pictin);
                      if (av2==1) count++;
                     }
               if ((count>2) && (count<8))
                   {  /* cunt number of transitions */
                    a0 = Getc_malloc_array(l  ,k  , pictin);  /*  8  1  2 */
                    a1 = Getc_malloc_array(l  ,k-1, pictin);  /*  7  0  3 */
                    a2 = Getc_malloc_array(l+1,k-1, pictin);  /*  6  5  4 */
                    a3 = Getc_malloc_array(l+1,k  , pictin);
                    a4 = Getc_malloc_array(l+1,k+1, pictin);
                    a5 = Getc_malloc_array(l  ,k+1, pictin);
                    a6 = Getc_malloc_array(l-1,k+1, pictin);
                    a7 = Getc_malloc_array(l-1,k  , pictin);
                    a8 = Getc_malloc_array(l-1,k-1, pictin);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    y[0] = a8; y[1] = a1; y[2] = a2;
                    y[3] = a3; y[4] = a4; y[5] = a5;
                    y[6] = a6; y[7] = a7; y[8] = a8;
                    trans=0;
                    for (m=0;m<=7;m++)
                      if ((y[m]==0) && (y[m+1]==1)) trans++;
                      /* if the number of transitions is 1 delete current pixel */
                    if (trans==1)
                        {
                          ar = (unsigned) k * MAXCOL + (unsigned) l;
                          *(pictin + ar) = 0;
                          OK = 0;
                        }
                  }
              }
          } while (OK==0);
  // ----- write file from malloced array ------
      fw = openfile(fo, "wb");
      for (j=0; j<MAXROW; j++)
         for (i=0; i<MAXCOL; i++)
         {
              ar = (unsigned) j * MAXCOL + (unsigned) i;
              c  = *(pictin + ar);
              fwrite(&c,1,1,fw);
         }
      fclose(fw);
      free(pictin);
/*  */ //  ***** end remart for not run process call ****
      open_graphVGAHI(gray);
  do
  {
      picture1(fi,2,0,0);
      setfillstyle(0,0);
      bar(50,205,50+25*8,205+8);
      outtextxy(50, 205, fi); outtextxy(140,205," 2 Level");
      picture1(fo,2,320,0);
      outtextxy(370, 205, fo); outtextxy(480,205," 2 Level");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        picture2(fo,2,321,221);
        outtextxy(370, 425, fo); outtextxy(480,425," INVERSE");
        outtextxy(640-(32*8), 470, "          BINARY      F10=END");
        color_scal(310,225);
        *next = getckey();
    } while (*next != F10);
    closegraph();
}
/*-------------------------------------------------------------------*/
/*  .....TWO PASS THINNING.....  */
/*-------------------------------------------------------------------*/
void TWOPASSTHIN(char *fi,char *fo, int *next,int gray,
                 int N1, int M1, int N2, int M2)

/* Subroutine for Calcurate Binary picture
   *fi: source buffer
   *fo : destination buffer
   N1=1,M1=1: start coordinates source buffer
   N2=ROW,M2=COLUMN: end   coordinates source buffer */
{
    FILE *fr,*fw;
    int  k,l,i,j,count=0,y[9],trans=0,m,OK=1,turn=0;
    int av1,av2,av3,a0,a1,a2,a3,a4,a5,a6,a7,a8; // input buffers
    BYTE  *pictin;       // array save read picture in memory1
    BYTE  *pictout;      // array save read picture in memory2
    BYTE  c;             // for get and put to file
    unsigned int ar;
/*  *****  remart for not run process call **** */
    // ----- allocate memory for read file and write file ------
        if( (pictin = (BYTE *) malloc(MAXCOL*MAXROW)) == NULL)
        {
           printf("Not enough memory to allocate buffer of read file");
           exit(1);
        }
    // ----- allocate memory for read file and write file ------
        if( (pictout = (BYTE *) malloc(MAXCOL*MAXROW)) == NULL)
        {
           printf("Not enough memory to allocate buffer of write file");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        exit(1);
    }
// ------ read file to malloced array ----- read 1
    fr  = openfile(fi, "rb");
    for (j=0; j<MAXROW; j++)
        for (i=0; i<MAXCOL; i++)
        {
            ar = (unsigned) j * MAXCOL + (unsigned) i;
            c  = getc(fr);
            *(pictin + ar) = c;
            *(pictout + ar) = 0;
        }
    fclose(fr);
// ------ process in memory --------
    do
    {
      OK=1; turn=(turn+1)%2;
      /* clear flag buffer */
      for (k=1;k<N2-N1-1;k++)
         for (l=1;l<M2-M1-1;l++)
              { ar = (unsigned) k * MAXCOL + (unsigned) l;
                *(pictout + ar) = 0;
              }
      for (k=N1+1;k<N2-1;k++)
        for (l=M1+1;l<M2-1;l++)
        {  av1 = Getc_malloc_array(l ,k  , pictin);
           if (av1==1)
             {  /* count number of 1s in 3 x 3 windows */
              count = 0;
              for (i=-1;i<=1;i++)
                  for (j=-1;j<=1;j++)
                     {
                      av2 = Getc_malloc_array(l+j,k+i, pictin);
                      if (av2==1) count++;
                     }
              if ((count>2) && (count<8))
                   {  /* cunt number of transitions */
                      a0 = Getc_malloc_array(l ,k  , pictin);  /*  8  1  2 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                a1 = Getc_malloc_array(l  ,k-1, pictin);  /*  7  0  3 */
                a2 = Getc_malloc_array(l+1,k-1, pictin);  /*  6  5  4 */
                a3 = Getc_malloc_array(l+1,k  , pictin);
                a4 = Getc_malloc_array(l+1,k+1, pictin);
                a5 = Getc_malloc_array(l  ,k+1, pictin);
                a6 = Getc_malloc_array(l-1,k+1, pictin);
                a7 = Getc_malloc_array(l-1,k  , pictin);
                a8 = Getc_malloc_array(l-1,k-1, pictin);
                y[0] = a8; y[1] = a1; y[2] = a2;
                y[3] = a3; y[4] = a4; y[5] = a5;
                y[6] = a6; y[7] = a7; y[8] = a8;
                trans=0;
                for (m=0;m<=7;m++)
                  if ((y[m]==0) && (y[m+1]==1)) trans++;
                  /* Flag current pizel too be deleted */
                if (trans==1)
                  if (turn==0 && (y[3]==0 || y[5]==0 || (y[1]==0 && y[7]==0)))
                     { ar = (unsigned) k * MAXCOL + (unsigned) l;
                     *(pictout + ar) = 1;
                     OK = 0;
                     }
                  else
                     if (turn==1 && (y[1]==0 || y[7]==0 || (y[3]==0 && y[5]==0)))
                       { ar = (unsigned) k * MAXCOL + (unsigned) l;
                       *(pictout + ar) = 1;
                       OK = 0;
                       }
              }
          }
    }
/* delete flagged pixels. */
for (k=N1+1;k<N2-1;k++)
    for (l=M1+1;l<M2-1;l++)
    {  ar = (unsigned) k * MAXCOL + (unsigned) l;
         av3  = *(pictout + ar);
         if (av3==1)
           {  ar = (unsigned) k * MAXCOL + (unsigned) l;
                *(pictin + ar) = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                }
            } while (OK==0);
    // ------ write file from malloced array ------
        fw = openfile(fo, "wb");
        for (j=0; j<MAXROW; j++)
            for (i=0; i<MAXCOL; i++)
            {
                ar = (unsigned) j * MAXCOL + (unsigned) i;
                c  = *(pictin + ar);
                if (c!=1) c=0;
                fwrite(&c,1,1,fw);
            }
        fclose(fw);
        free(pictin);
        free(pictout);
/* */ //  *****  end remart for not run process call ****
        open_graphVGAHI(gray);
    do
    {
        picture1(fi,2,0,0);
        setfillstyle(0,0);
        bar(50,205,50+25*8,205+8);
        outtextxy(50, 205, fi); outtextxy(140,205," 2 Level");
        picture1(fo,2,320,0);
        outtextxy(370, 205, fo); outtextxy(480,205," 2 Level");
        picture2(fo,2,321,221);
        outtextxy(370, 425, fo); outtextxy(480,425," INVERSE");
        outtextxy(640-(32*8), 470, "          BINARY      F10=END");
        color_scal(310,225);
        *next = getckey();
    } while (*next != F10);
    closegraph();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ

บทความที่ได้รับการตีพิมพ์ในวารสารสารสนเทศลาดกระบัง (ฉบับพิเศษ)

Ladkrabang Information Journal (Special Issue) ประจำปี 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงข้อมูลภาพจากรูปแบบราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ

Automated Image Conversion From Raster To Vector Format

กิ่งกาญจน์ วงศ์วิภาพร * ดร.บุญวัฒน์ อัตชู **

บทคัดย่อ

บทความวิจัยนี้นำเสนอถึงวิธีการแปลงข้อมูลภาพจากรูปแบบราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ โดยใช้หลักการและทฤษฎีเกี่ยวกับการประมวลผลข้อมูลภาพมาช่วยในการแปลงข้อมูลภาพแบบราสเตอร์ที่มีลักษณะการเก็บเป็นจุด (Pixel) เป็นเวคเตอร์ที่มีลักษณะเป็นจุด (Point) เส้น (Line) รูปเหลี่ยม (Polygon) แบบอัตโนมัติ แทนการใช้คนทำการเก็บข้อมูลเวคเตอร์ผ่านเครื่องดิจิไตท์เซอร์ โดยการนำภาพสีหรือขาวดำที่ต้องการแปลงมาผ่านเครื่องสแกนเนอร์ทำให้ได้ข้อมูลภาพแบบราสเตอร์ และทำการประมวลผลข้อมูลภาพ ตามขั้นตอนที่ออกแบบได้ตามลำดับดังนี้ การหาขอบภาพ (Edge Detection) โดยใช้เทคนิค (Range Edge) [2] การทำขอบภาพให้บาง (Edge Thinning) โดยใช้เทคนิค (Simple Thinning) [2] การหาจุดรวมของเส้น (Node) โดยใช้ตัวกรอง การหาส่วนของเส้นตรง (Line Tracking) or (Line Segment) โดยใช้เทคนิค (Chain Link Code) และเทคนิคตัวกรองหาทิศทางที่ออกแบบขึ้น การหารูปเหลี่ยมพื้นที่ (Polygon) โดยใช้เทคนิค (Labeling) ซึ่งทำให้สามารถเก็บข้อมูลจากผลการทดลอง และพบว่าสามารถแปลงข้อมูลภาพที่มีกลุ่มสี หรือ โทนเดียวกันเป็นรูป Polygon ได้ และเก็บข้อมูลภาพในรูปเวคเตอร์ได้

Abstract

The conversion from raster to vector format usually use digitizer by manual. This paper presents data image processing theory to automatic converting raster image from pixel format to vector format of point , line or polygon. The approach is uses scanner transform color or black and white image to raster image format and process step by step for conversion. Range edge technique and simple thinning are used for edge detection and edge thinning. The filtering method is presented for finding node position of line. Line segment or Line tracking uses Chain Link Code technique and filtering of dimension. Including polygon process finding by labeling technique. The experimental result of these processes can convert color or gray image to polygon in vector format.

---

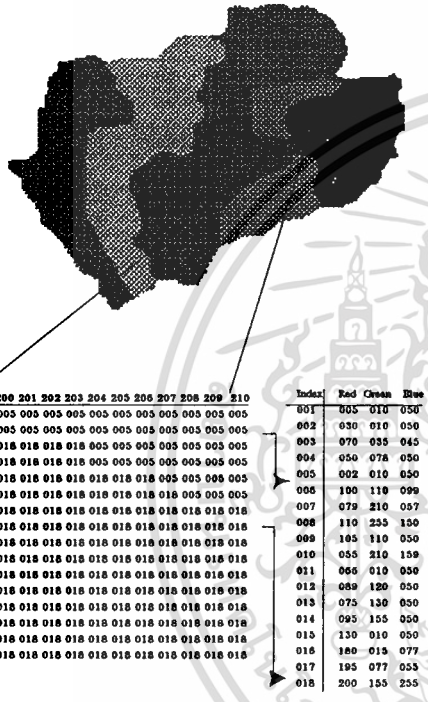
* นักศึกษาปริญญาโท คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

** ผู้ช่วยศาสตราจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บทนำ

ปัจจุบันนี้ในการเก็บข้อมูลภาพเข้าเครื่องคอมพิวเตอร์ โดยผ่านอุปกรณ์คอมพิวเตอร์ เช่น เครื่องสแกนเนอร์ เครื่องดิจิไทท์เซอร์ กล้องถ่ายภาพระยะไกล โดยจะให้ผลลัพธ์ของข้อมูลในการเก็บมีโครงสร้างหลักแบ่งเป็น 2 ประเภท ดังนี้



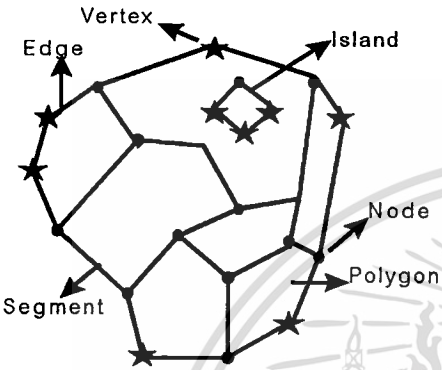
รูปที่ 1 ตัวอย่างภาพและข้อมูลราสเตอร์ที่ได้จากการนำภาพผ่าน เครื่องสแกนเนอร์ ซึ่งมีลักษณะเป็นภาพสี

1.1 ลักษณะโครงสร้างข้อมูลแบบราสเตอร์ (Raster Structure) จะประกอบด้วยลักษณะของช่องสี่เหลี่ยมที่เรียกว่า "กริด" หรือ "จุด" (Grid or Pixels) โดยในแต่ละจุดจะประกอบด้วยตัวเลขซึ่งในการแทนค่าของระดับความเข้มที่ต่างกันในแต่ละส่วนของภาพ และถ้าเป็นภาพสีก็จะมีความเข้มประกอบด้วยระดับความเข้มของแม่สี คือ สีแดง สีเขียว สีน้ำเงิน โดยข้อมูลแบบราสเตอร์จะได้จาก ข้อมูลจากภาพถ่ายดาวเทียม ข้อมูลจากภาพถ่ายทางอากาศ ข้อมูลจากเครื่องสแกนเนอร์ ซึ่งมีลักษณะข้อมูลเป็นแบบอาร์เรย์ 2 มิติ โดยมีลักษณะของแถวแนวนอน (Row) และแถวแนวตั้ง (Column) เป็นตัวกำหนดตำแหน่งและทิศทาง เช่น ภาพที่มีความละเอียด (Resolution) ขนาดแนวนอน (Row) 320 และ แนวตั้ง (Column) 200 ดังนั้นภาพนี้จะมีขนาด 320x200 Pixel โดยประกอบด้วย 64000 Pixel และใช้พื้นที่ในการเก็บ 64 KB เป็นต้น โดยจะแสดงส่วนของเส้นด้วยจำนวนของ Pixel ที่อยู่ใกล้เคียงและอยู่ต่อเนื่องกันตามทิศทางที่กำหนด และจะแสดงส่วนของพื้นที่ด้วยความสัมพันธ์ของปริมาณการกระจาย Pixel ไปยัง Pixel ใกล้เคียง ดังตัวอย่างภาพและข้อมูลราสเตอร์ที่ได้ผ่านเครื่องสแกนเนอร์ในรูปที่ 1

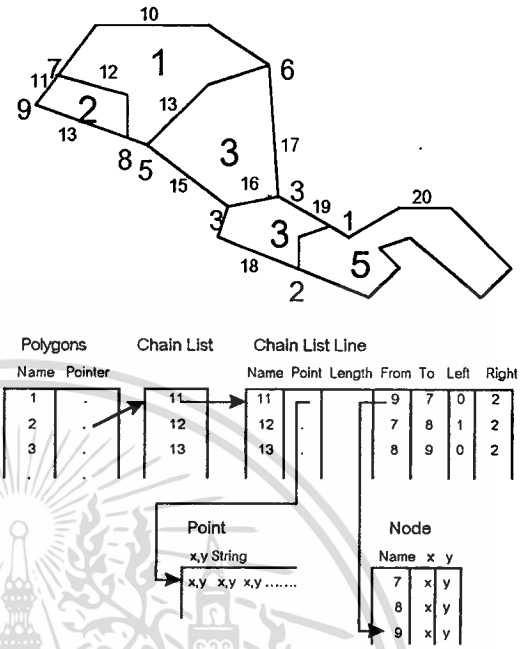
1.2 ลักษณะโครงสร้างแบบเวคเตอร์ (Vector Structure) จะประกอบด้วยลักษณะของจุด (Point) เส้น (Line) อาร์ค (Arc) รูปเหลี่ยมพื้นที่ (Polygon) โดยจุดจะเชื่อมต่อกันเป็นเส้น ซึ่งมักมีลักษณะการเก็บเป็นแบบความสัมพันธ์ต่อเนื่องของจุด (Chain List) และเส้นตรงที่ต่อกันจะทำให้เกิดอาร์ค (Arc) ที่เป็นองค์ประกอบสำคัญของรูปแบบเส้น (Linear Feature) เช่น รูปแบบเส้นแสดงถนน แม่น้ำ เส้นแบ่งเขตแดน เป็นต้น และถ้าปลายของอาร์คหลาย ๆ อาร์ค ที่ต่อกันจนเกิดขอบเขตก็จะเกิดรูปเหลี่ยมพื้นที่ (Polygon) และจุดรวมของเส้น (Node) โดยขบวนการเก็บของข้อมูลแบบเวคเตอร์จะใช้คู่พิกัด X และ Y เป็นตัวชี้ตำแหน่ง ในการเก็บข้อมูลแบบเวคเตอร์เข้าเครื่องคอมพิวเตอร์จะทำโดยผ่านเครื่องดิจิตไทท์เซอร์ (Digitizer) โดยใช้คนเป็นผู้กำหนดค่าพิกัด X,Y พร้อมทั้งระบุว่าเป็นเส้นหรือรูปเหลี่ยมพื้นที่ ผ่านเครื่องดิจิตไทท์เพื่อเก็บข้อมูลเวคเตอร์ในระบบคอมพิวเตอร์ ดังตัวอย่างส่วนประกอบโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเวคเตอร์ในรูปที่ 2 และตัวอย่างการเก็บข้อมูลที่มีความสัมพันธ์ ของโครงสร้างแบบเวคเตอร์ใน รูปที่ 3



รูปที่ 2 ภาพแสดงส่วนประกอบของโครงสร้างข้อมูลแบบเวคเตอร์



รูปที่ 3 ตัวอย่างความสัมพันธ์ของส่วนประกอบข้อมูลในโครงสร้างแบบเวคเตอร์

ตารางที่ 1 แสดงการเปรียบเทียบคุณสมบัติของโครงสร้างข้อมูลแบบเวคเตอร์และราสเตอร์

คุณลักษณะ	โครงสร้างแบบเวคเตอร์	โครงสร้างแบบราสเตอร์
ความง่ายในการเรียนรู้	มีโครงสร้างที่สลับซับซ้อนและยุ่งยากต่อการเรียนรู้	มีโครงสร้างที่ง่ายต่อการเรียนรู้
ความชัดเจนของตำแหน่ง	สามารถระบุตำแหน่งได้แน่นนอนชัดเจน	ความชัดเจนของตำแหน่งจะเกิดขึ้นได้จะต้องมีกระบวนการประมวลผลข้อมูล ซึ่งจะต้องใช้เวลาโดยความละเอียดชัดเจนจะถูกจำกัดด้วยจำนวน Pixel ในภาพ
ความชัดเจนของภาพ	จะให้ความชัดเจนในส่วนของภาพที่เป็นจุด เส้นและรูปเหลี่ยมพื้นที่	จะให้ความชัดเจนดีถ้าเป็นภาพที่แสดงความเป็นจริงของวัตถุจริง เช่นภาพคน ภาพวิว เพราะแต่ละ Pixel ในภาพที่กระจายอยู่จะมีระดับความเข้มของแต่ละสีที่แตกต่างกัน
ความสามารถในการนำข้อมูลภาพมาซ้อนทับกันเพื่อประโยชน์ในการแสดงผลและวิเคราห์	สามารถทับซ้อนข้อมูลของภาพได้ แต่ถ้าซ้อนหลายชั้นมากเกินไป จะทำให้ยากแก่การแยกแยะข้อมูล	เนื่องจากข้อมูลมีลักษณะการเก็บเป็น Pixel ตามจำนวนของขนาดความละเอียดของภาพ ดังนั้นการซ้อนทับกันหลาย ๆ ชั้นจึงทำให้ไม่ก่อให้เกิดปัญหา เพราะมีข้อมูลทุก Pixel ในการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 แสดงการเปรียบเทียบคุณสมบัติของโครงสร้างข้อมูลแบบเวคเตอร์และราสเตอร์ (ต่อ)

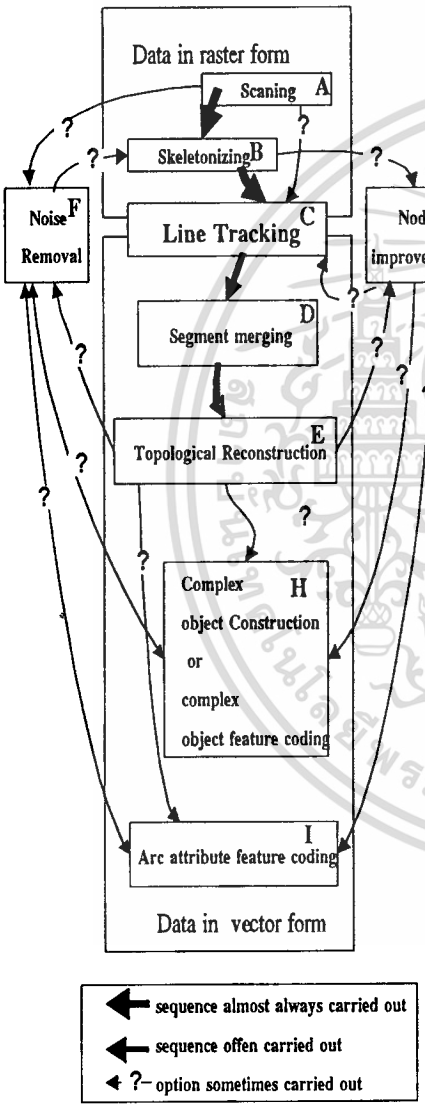
คุณลักษณะ	โครงสร้างแบบเวคเตอร์	โครงสร้างแบบราสเตอร์
ความง่ายในการวิเคราะห์ข้อมูล	ในการสอบถามข้อมูลเชิงระยะทางสามารถทำได้รวดเร็ว แต่การวิเคราะห์ข้อมูลมีข้อจำกัดในเรื่องการซ้อนทับของจุดกับเส้น	ในการสอบถามข้อมูลเชิงระยะทางจะช้ากว่าแบบเวคเตอร์ แต่ในการวิเคราะห์ข้อมูลจะทำได้ดีว่าเพราะมีกระบวนการประมวลผลข้อมูลที่ไม่ยุ่งยากนัก เช่น ใช้หลักการตัวกรองข้อมูล (Filter) และมีโมเดลในการประมวลผลข้อมูลมากมาย
โครงสร้างข้อมูล	สลับซับซ้อน	ไม่สลับซับซ้อน
ความต้องการพื้นที่ในการเก็บข้อมูล	ความต้องการพื้นที่ขึ้นอยู่กับจำนวนจุด เส้น และรูปเหลี่ยมที่มีอยู่ภายในภาพ	ความต้องการพื้นที่ในการเก็บขึ้นอยู่กับความละเอียดของภาพ
ความเหมือนจริงของภาพ	จะมีความเหมือนจริงน้อย เนื่องจากการแสดงผลภาพจะมีลักษณะเป็นจุด เส้น หรือรูปเหลี่ยม	จะมีความเหมือนจริงมากเพราะมีระดับความเข้มของสีในแต่ละจุดของภาพที่แสดงถึงความเหมือนจริง
ความสามารถในการนำข้อมูลมาแปลงเป็นลักษณะแบบโครงข่าย	สามารถนำมาใช้ในรูปแบบของโครงข่ายได้ดี กว่าเพราะโครงสร้างมีการเก็บแบบให้ความสัมพันธ์กันระหว่างจุด เส้น รูปเหลี่ยม	ยากในการนำมาใช้ในรูปแบบของโครงข่าย เพราะการเก็บมีลักษณะเป็น Pixel ซึ่งแต่ละ Pixel ไม่ได้แสดงถึงความสัมพันธ์กัน
ค่าใช้จ่ายในการเก็บข้อมูล	การเก็บหรือแสดงผลข้อมูลใช้เวลานาน และอุปกรณ์ในการแสดงผลหรือนำเข้าก็มีราคาแพงด้วย	อุปกรณ์ในการนำเข้าและแสดงผลสามารถหาซื้อได้ง่ายและราคาไม่แพง
การปรับปรุงและแก้ไขข้อมูล	สามารถทำได้ง่ายและรวดเร็ว	จะต้องใช้เวลามากและแก้ไขได้ยากเพราะการแก้ไขจะต้องทำที่ละ Pixel

จากตารางการเปรียบเทียบคุณสมบัติโครงสร้างข้อมูลแบบราสเตอร์และเวคเตอร์ (ตารางที่ 1) แสดงถึงข้อดีและข้อเสียของโครงสร้างทั้ง 2 ประเภท โดยถ้าต้องการข้อมูลที่ได้ภาพที่เหมือนจริงและง่ายในการเก็บก็ควรใช้แบบราสเตอร์ แต่ถ้ามีพื้นที่ในการเก็บข้อมูลน้อยและลักษณะของข้อมูลเป็นลายเส้นก็ควรเก็บแบบเวคเตอร์ ซึ่งโดยทั่วไปความต้องการของการใช้งานข้อมูลในการวิเคราะห์และแสดงผลนั้นจะมีความต้องการเท่า ๆ กัน และถ้าหากสามารถแปลงข้อมูลระหว่างรูปแบบราสเตอร์เป็นแบบเวคเตอร์ได้หรือแปลงจากแบบเวคเตอร์เป็นแบบราสเตอร์ได้แล้วนั้น ก็จะสามารถทำให้เกิดการประหยัดเวลาในการประมวลผลข้อมูลภาพ พื้นที่ในเก็บข้อมูล และลดค่าใช้จ่ายเกี่ยวกับอุปกรณ์ในการประมวลผล ทั้งผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็สามารถเลือกรูปแบบข้อมูลตามความเหมาะสมแก่การใช้งานโดยไม่ต้องคำนึงถึงเงื่อนไขหรือข้อจำกัดในแง่ของ Software และ Hardware

2. ขั้นตอนในการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์



รูปที่ 4 แสดงความสัมพันธ์ของขั้นตอนในการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์ [4]

ในการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์นั้นมีขั้นตอนที่เกี่ยวข้อง 9 ขั้นตอน ดังนี้

2.1 Scanning A การใช้อุปกรณ์คอมพิวเตอร์ในการเก็บข้อมูลภาพเป็นแบบราสเตอร์เข้าเครื่องคอมพิวเตอร์

2.2 Noise Removal F การขจัดข้อมูลที่รบกวนภาพ

2.3 Skeletonizing B การหาโครงร่างของภาพที่บาง

2.4 Node Improvement G การปรับปรุงและหาจุดต่อเชื่อมเพื่อให้เกิดความชัดเจนของโครงร่างภาพ

2.5 Line Tracking C การหาส่วนของเส้น

2.6 Segment Merge D การหาส่วนที่ประกอบของพื้นที่

2.7 Topological Reconstruction E การจัดความสัมพันธ์ของส่วนประกอบของภาพ

2.8 Complex Object Construction And Complex Object Feature Coding H กำหนดวัตถุในภาพที่ซับซ้อน

2.9 Arc Attribute Feature Coding I การหามุม ลักษณะและทิศทางของเส้น

โดยมีความสัมพันธ์ของแต่ละขั้นตอนดังรูปที่ 4 ซึ่งจะเห็นได้ว่าการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์ไม่จำเป็นที่จะต้องทำทั้ง 9 ขั้นตอนในบางขั้นตอนอาจไม่จำเป็นต้องทำ เช่นหากภาพที่เราต้องการแปลงเป็นภาพที่ไม่ซับซ้อน ก็ไม่จำเป็นต้องทำการกำหนดวัตถุของภาพที่ซับซ้อน ซึ่งขั้นตอนในการแปลงจะไม่ได้เป็นขั้นตอนที่มาตราฐาน โดยขั้นตอนใดจำเป็นที่จะต้องทำมากน้อยเพียงใดขึ้นอยู่กับภาพต้นแบบที่นำมาแปลงว่ามีความสลับซับซ้อน หรือ ชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากน้อยเพียงใด ดังนั้นในการแปลงข้อมูลสามารถสรุปถึงขั้นตอนที่จำเป็นและไม่จำเป็นได้ดังนี้

1. ขั้นตอนที่มักจะต้องทำเสมอมีดังนี้ A B C D หรือ A B C D E

2. ขั้นตอน B สามารถไม่ต้องทำได้ ซึ่งจะทำให้เหลือเพียงขั้นตอน A C D หรือ A C D E

3. ขั้นตอน E F จะสามารถทำได้ทั้งกับรูปแบบข้อมูลที่เป็นราสเตอร์และเวคเตอร์

4. ขั้นตอน E F G หากต้องการทำจะต้องคำนึงถึงค่าใช้จ่ายและเวลาในการประมวลผล แต่ถ้าทำจะทำให้ผลลัพธ์ของข้อมูลแบบเวคเตอร์ดูใกล้เคียงกับภาพต้นแบบ และชัดเจนมากยิ่งขึ้น

5. ขั้นตอน H I จะทำเมื่อภาพมีความซับซ้อนมาก ๆ

3. การออกแบบขั้นตอนและพัฒนาการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์

การออกแบบขั้นตอนแปลงข้อมูลภาพในที่นี้กำหนดให้ภาพที่ผ่านเครื่องสแกนเนอร์มีความคมชัดและมีกลุ่มสีที่ชัดเจน ดังนั้นภาพที่ได้จึงไม่มีความจำเป็นต้องกระทำ Preprocessing โดยขั้นตอนที่ออกแบบมีดังนี้

- **Scaning** A การเก็บข้อมูลภาพสีขนาด 320x200 ผ่านเครื่องสแกนเนอร์ เป็นรูปแบบ .PCX ไฟล์ จะทำตามขั้นตอนที่ 3.1

- **Skeletonizing** B การหาโครงร่างของภาพที่บางจะประกอบไปด้วย 2 ขั้นตอน คือ ขั้นตอนที่ 3.2.1 และ 3.2.2

- **Node Improvement** G การหาจุดรวมของเส้นที่ชัดเจน จะทำตามขั้นตอนที่ 3.3

- **Line Tracking** C การหาส่วนของเส้นจะทำตามขั้นตอนที่ 3.4

- **Segment Merge** D การหาส่วนประกอบของพื้นที่ จะทำตามขั้นตอนที่ 3.5

3.1 การเก็บข้อมูลภาพผ่านเครื่องแกนเนอร์ (Scaning) โดยทำการนำภาพแผนที่ภาคเหนือของประเทศไทยมาสแกนผ่านเครื่องสแกนเนอร์ ซึ่งทำให้ได้ข้อมูลภาพแบบราสเตอร์ ณ ระดับความเข้มของสีที่ประกอบด้วยสีแดง เขียว น้ำเงิน ดังตัวอย่างในภาพที่ 1 จากนั้นก็ทำการเลือกค่าความเข้มของเฉพาะสีแดงมาทำการประมวลผลในขั้นตอนที่ 3.2

3.2 การหาโครงร่างของภาพที่บาง (Skeletonizing) จะประกอบไปด้วย 2 ขั้นตอนด้วยกัน ดังนี้

3.2.1 การหาขอบภาพ (Edge Detection) ตามทฤษฎีในการประมวลผลหาขอบภาพนั้นมีหลายวิธีด้วยกัน เช่น Prewitt Edge, Sobel Edge, Kirsh Edge, Laplacian Edge [2] และอื่น ๆ อีกมากมาย แต่ในการทดลองนี้จะใช้วิธี Range Edge เพราะสามารถกำหนดขนาดของ Edge ที่เหมาะสมและยังกำจัดข้อมูลรบกวนภาพ (Noise) ได้อย่างถูกต้อง โดยไม่ทำให้ขอบภาพหาย ซึ่งใช้หลักการหาค่าความเข้มที่เหมาะสมในแต่ละจุดของภาพ โดยจะทำการหาผลต่างของค่าความเข้มที่สูงสุดกับความเข้มที่ต่ำสุดของจุดบริเวณรอบ ๆ และจะรอบกว้างเพียงใดขึ้นอยู่กับค่าตัดสินใจ Threshold และผลต่างที่คำนวณได้จะนำมาเป็นค่าความเข้มใหม่ของจุดนั้น ๆ โดยมีสูตรในการคำนวณดังสมการที่ 3.2.1

$$w(k,l)= Max_A\{F(k,l)\} - Min_A\{F(k,l)\} \qquad 3.2.1$$

k=Row l=Column

$Max_A\{F(k,l)\}$=ค่าสูงสุดของความเข้มบริเวณจุดรอบ ๆที่สนใจ

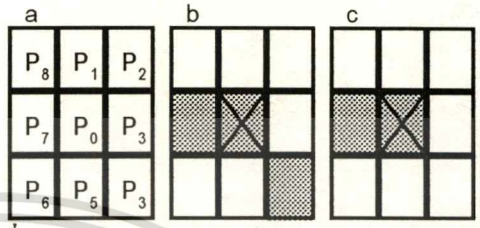
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\mathbf{Min_A}$ **{F(k,l)}=ค่าต่ำสุดของความเข้มบริเวณจุดรอบๆ ที่สนใจ**

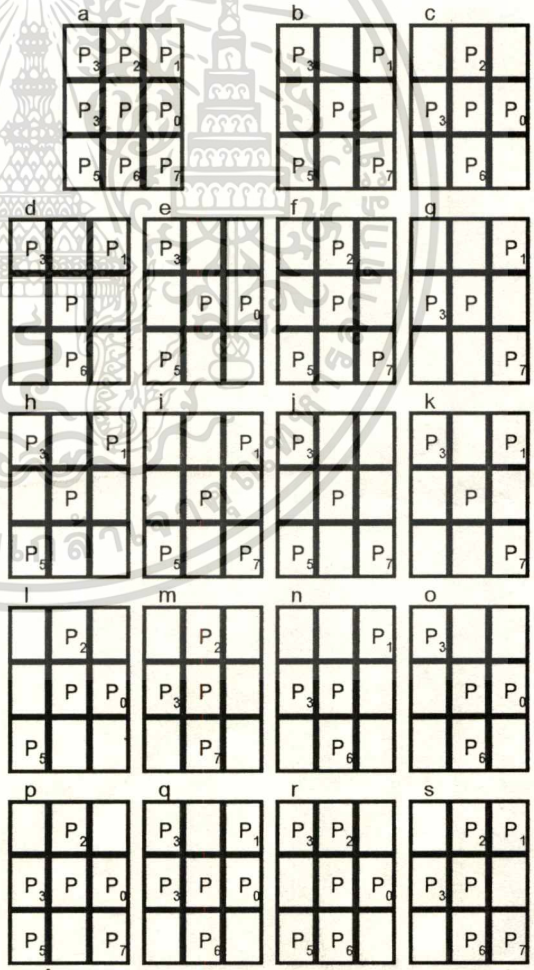
w(k,l) = ค่าระดับความเข้มใหม่ที่คำนวณได้

หลังจากที่ได้ภาพที่มีระดับความเข้มใหม่แล้ว จะทำการแปลงข้อมูลภาพให้เป็น Binary คือเก็บเป็นค่าข้อมูล '0' กับ '1' โดยถ้าค่าระดับความเข้มใหม่ >= 1 ให้ค่าเป็น '1' แต่ถ้าค่าระดับความเข้มใหม่ <1 ให้ค่าเป็น '0' จากนั้นก็จะนำข้อมูล Binary ที่ได้ไปทำการประมวลผลต่อในขั้นตอนที่ 3.2.2

3.2.2 การทำขอบภาพให้บาง (Thinning Edge) เนื่องจากขอบภาพที่ได้จากการทำ Range Edge นั้นเป็นขอบภาพที่ดีและชัดเจนมาแล้ว ดังนั้นในการทำขอบภาพให้บางจึงเพียงแค่ใช้วิธี Simple Thining [5] ซึ่งจะทำโดยการกำหนดความสัมพันธ์ของจุดในภาพ กับจุดของภาพที่อยู่ข้างเคียง (Neighbouring Pixel) โดยกำหนดความสัมพันธ์ของจุดเป็นตาราง (Window) ขนาด 3x3 ดังรูป 5a โดยทั่วไปแล้วจะกำหนดให้วัตถุหรือสิ่งที่สนใจ (Object or Patern) มีค่าระดับความเข้มเป็น '1' ส่วนพื้นที่ที่ไม่สนใจ (Background) มีค่าเป็น '0' จากรูป 5a จุด P เป็นจุดใด ๆ ภายในเนื้อวัตถุ ซึ่งนำมาวิเคราะห์หาความสัมพันธ์รอบจุด P_0 จะประกอบด้วยจุด P_1, P_2, P_3, P_4, P_5, P_6, P_7 ,P_8 ตามลำดับจาก 0->1->2->3->4->5->6->7->8 โดยการตรวจสอบว่าแต่ละลำดับจาก $P_{0->8}$ ถ้ามีการเปลี่ยนค่าความเข้มจาก '0' เป็น '1' เพียง 1 ครั้ง ก็จะทำการกำหนดค่าให้จุด P_0 มีค่าเท่ากับ '0' ซึ่งจะสามารถทำให้กำจัดส่วนของขอบภาพที่หนา ดังรูป 5b และขอบภาพที่มีขนาดสั้นและไม่จำเป็นดังรูป 5c ออกและทำให้เกิดขอบภาพที่บางและชัดเจนยิ่งขึ้น จากนั้นจะทำการตรวจสอบข้อมูลภาพ Binary ที่มีขอบภาพที่บางว่าถ้ามีค่าเป็น '1' จะถือว่าเป็นจุด (Point) ซึ่งทำให้สามารถเก็บข้อมูลที่เป็น Point Data ได้ดังรูปที่ 10 เพื่อใช้ในการประมวลผลขั้นตอนที่ 3.3-3.5 ต่อไป



รูปที่ 5 แสดงความสัมพันธ์ของจุดในตาราง (3x3) ที่ใช้ในการหาขอบภาพที่บาง

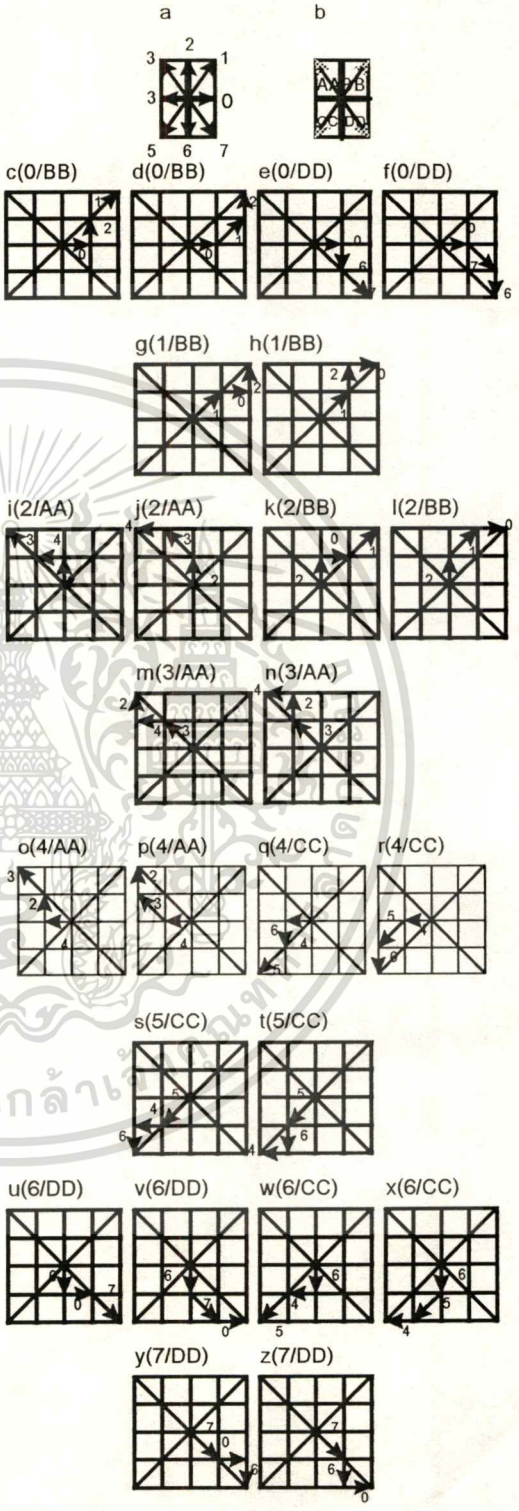


รูปที่ 6 แสดงความสัมพันธ์ของจุดในตาราง(3x3) ที่ใช้หา(Node)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การหาจุดรวมของเส้น (Node) ในการหาจุดรวมของเส้น จะทำโดยการกำหนดความสัมพันธ์ของจุดในภาพ กับจุดของภาพที่อยู่ข้างเคียง (Neighbouring Pixel) โดยกำหนดความสัมพันธ์ของจุดเป็นตาราง (Window) ขนาด 3x3 ของข้อมูลที่ได้ในขั้นตอนที่ 3.3 ดังรูป 6a จากรูป 6a จุด P เป็นจุดใด ๆ ภายในเนื้อวัตถุ ซึ่งนำมาวิเคราะห์หาความสัมพันธ์รอบจุด P จะประกอบด้วยจุด P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7 และในการหาจุดรวมของเส้น (Node) จะมีตารางที่เป็นตัวกำหนดคุณสมบัติของ Node ดังรูป 6b-6s โดยการตรวจสอบว่าในแต่ละ P_{0-7} มีค่าเป็น '1' ตามคุณสมบัติที่กำหนดดังรูป 6b-s หรือไม่ ถ้าตรงก็จะสามารถระบุได้ว่าเป็นส่วนของ (Node) ซึ่งจะได้ข้อมูล Node Data ดังรูปที่ 10 เพื่อใช้เป็นจุดเริ่มต้นของการส่วนของเส้นตรงในขั้นตอนที่ 3.4 ต่อไป

3.4 การหาส่วนของเส้นตรง (Line Tracking) or (Line Segment) จะทำโดยการกำหนดความสัมพันธ์ของจุดในทิศทางต่าง ๆ ที่ต่อเนื่องกันโดยการหาความสัมพันธ์และทิศทางจากตารางขนาด (3x3) ของข้อมูลที่ได้จากขั้นตอนที่ 3.2 ดังรูป 7a และแนวโน้มของทิศทางของเส้นดังรูป 7b โดยในแต่ละจุดที่เป็น (Node) ของข้อมูลที่ได้จากขั้นตอนที่ 3.3 จะถูกทำการตรวจสอบหาเส้นจากทิศทางที่ 0->7 ดังรูป 7a และแนวโน้มของทิศทางว่าเป็น AA,BB,CC,DD ดังรูป 7b จุดใดที่ได้ถูกตรวจสอบแล้วว่าเป็นส่วนของเส้นตรงจะถูกแทนค่าด้วย '0' และจุดเริ่มต้นของเส้นตรงจะเริ่มที่จุดที่เป็น (Node) หรือจุดที่มีแนวโน้มของเส้นเปลี่ยนทิศทาง โดยจุดสิ้นสุดของเส้นจะเกิดขึ้นเมื่อเจอ (Node) อื่น หรือเมื่อไม่มีจุดต่อเนื่องในทิศทางที่ 0-7 หรือแนวโน้มของทิศทางของเส้นเปลี่ยนไป โดยในการตรวจสอบแนวโน้มทิศทางของเส้นของจุดรอบที่ 0-7 จะแสดงดังรูปที่ 7c-7z ซึ่งจะทำให้ได้



รูปที่ 7 แสดงความสัมพันธ์ของจุดที่ทำให้เกิดเส้นและทิศทางของเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล Line in Polygon และ Point Data ในส่วนของ Chain List, Begin Point, End Point, Lenght ดังรูปที่ 10 เพื่อใช้ในการประมวลผลในขั้นตอนที่ 3.5 ต่อไป

3.5 การหารูปเหลี่ยมพื้นที่ (Polygon)

รูปที่ 8 แสดงการ Label หาพื้นที่ส่วนของรูปเหลี่ยม และหาเส้นที่เป็นส่วนประกอบของรูปเหลี่ยม

จะทำโดยการกำหนดความสัมพันธ์ของจุดในทิศทางต่าง ๆ ของข้อมูลที่ได้จากขั้นตอนที่ 3.2 เพื่อหาว่าส่วนใดเป็นส่วนของพื้นภาพ และส่วนใดเป็นส่วนของรูปเหลี่ยมพื้นที่ และเมื่อพบส่วนของรูปเหลี่ยมพื้นที่แล้วก็จะทำการกำหนดลำดับของรูปเหลี่ยมลงในส่วนของพื้นที่รูปเหลี่ยมนั้น (Labeling) ดังรูป 8 และตัวอย่างข้อมูลจำนวนรูปเหลี่ยม (Polygon) ที่ได้ ดังรูปที่ 10 เมื่อได้ส่วนที่เป็นพื้นที่ของรูปเหลี่ยมแล้วก็จะหาว่าส่วนของเส้นตรงใดที่เป็นส่วนประกอบของพื้นที่โดยการหาความสัมพันธ์รอบ ๆ ด้วยตาราง (3x3) ณ บริเวณจุดเริ่มต้นและจุดปลายของเส้น ของข้อมูลที่ได้จากขั้นตอน 3.4 กับข้อมูลที่ได้จากการ Labeling มาทำการตรวจสอบ และหลังจากหาความสัมพันธ์รอบจุดเริ่มต้น และจุดปลายของเส้นแล้วจะทำให้ทราบว่าเส้นไหนเป็นส่วนประกอบของรูปเหลี่ยม (Polygon) ใด ดังตัวอย่างของมูลในรูปที่ 10

4. ผลลัพธ์จากการแปลงข้อมูลจากราสเตอร์เป็นเวคเตอร์

จากการทดลองแปลงข้อมูลภาพแผนที่ภาคเหนือของประเทศไทย จากราสเตอร์เป็นเวคเตอร์ โดยทำตามขั้นตอนที่ได้ออกแบบ คือทำการ Scanning, Skeletoninzing, Node Improvement, Line Tracking, Segment Merge ตามขั้นตอนที่ 3.1-3.5 ทำให้ได้ผลลัพธ์ดังนี้

4.1 ในการหาขอบภาพโดยวิธี Sobel Edge และใช้ค่า Threshold 32 จะทำให้ได้ผลลัพธ์ของภาพดังรูป 9a ซึ่งจะเห็นได้ว่าขอบภาพบางส่วนขาดหายไป และขอบภาพบางส่วนเส้นแตกไม่สม่ำเสมอ และเมื่อเปลี่ยนค่า Threshold เป็น 16 ปรากฏว่าขอบภาพดีขึ้น ส่วนของเส้นครบ แต่ยังมีปัญหาขอบภาพบางส่วนเส้นแตกไม่สม่ำเสมอ ดังรูป 9b ดังนั้นจึงเปลี่ยนค่า Threshold เป็น 1 ซึ่งปรากฏว่าได้เส้นของภาพครบแต่เป็นขอบภาพที่หนามาก ดังรูป 9c ต่อมาได้เปลี่ยนวิธีการหาขอบภาพโดยวิธี Range Edge ตามขั้นตอนที่ 3.2.1 โดยเลือกค่า Threhold เป็น 1 และขนาดตารางตรวจสอบเส้นเท่ากับ 1x1 ซึ่งทำให้ได้ขอบภาพที่ชัดเจนและไม่หนาเกินไปดังรูป 9d และเมื่อผ่านการทำขอบภาพให้บางโดยวิธี Simple Thining Edge ตามขั้นตอนที่ 3.2.2 จะทำให้ได้ขอบภาพที่บางดังรูปที่ 9e

4.2 เมื่อทำการหาจุดรวมของเส้น (Node) ตามขั้นตอนที่ 3.3 โดยหาความสัมพันธ์ของจุดจากรูปที่ 6 ซึ่งทำให้ได้ผลลัพธ์ของ (Node) ดังรูปที่ 9f แล้วจะทำการหาส่วนของเส้นตรง ตามขั้นตอนที่ 3.4 ซึ่งทำให้ได้ผลลัพธ์ดังรูปที่ 9g เป็นการหาส่วนของเส้นตรงโดยระบุเงื่อนไขว่า ถ้าแนวโน้มของเส้นเปลี่ยนจะถือเป็นเส้นใหม่ ซึ่งจำนวนเส้นที่ได้จะน้อยแต่ภาพที่ได้จะไม่ใกล้เคียงกับภาพต้นแบบ ในส่วนของรูปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9h จะแสดงถึงการหาส่วนของเส้นตรงโดยระบุเงื่อนไขว่าถ้าแนวโน้มของเส้นเปลี่ยนและทิศทางของความต่อเนื่องของเส้นเปลี่ยน 15 ครั้ง ในส่วนของรูปที่ 9i จะแสดงถึงการหาส่วนของเส้นตรงโดยระบุเงื่อนไขว่าถ้าแนวโน้มของเส้นเปลี่ยนและทิศทางของความต่อเนื่องของเส้นเปลี่ยน 5 ครั้ง ซึ่งทำให้เกิดความชัดเจนของภาพมากยิ่งขึ้น ในส่วนในรูปที่ 9j จะแสดงถึงการหาส่วนของเส้นตรงโดยระบุเงื่อนไขว่าถ้าแนวโน้มของเส้นเปลี่ยนและทิศทางของความต่อเนื่องของเส้นเปลี่ยน 1 ครั้ง ซึ่งทำให้ได้ภาพเหมือนภาพต้นแบบมากที่สุด จากการเปลี่ยนเทียบผลลัพธ์ของภาพในรูป 9h ,9i,9j นั้น จะพบว่ารูป 9j จะมีจำนวนเส้นมากที่สุด และรูป 9h จะมีจำนวนเส้นน้อยที่สุด

4.3 ในการหาส่วนของพื้นที่รูปเหลี่ยมโดยการทำ Labeling ตามขั้นตอนที่ 3.5 ซึ่งทำให้ได้ผลลัพธ์ดังรูป 9k และได้จำนวนรูปเหลี่ยมทั้งหมด 8 รูป ดังรูปที่ 9m,9n,9o,9p,9q,9r,9s,9t และเมื่อประกอบเป็นภาพแล้วจะได้ดังรูป 9l และในรูปที่ 10 จะแสดงถึงตัวอย่างผลลัพธ์ของข้อมูลภาพที่เป็น จุด (Point) ส่วนประกอบของเส้น (Chain List) เส้น (Line) รูปเหลี่ยม (Polygon) ที่ได้จากรูปที่ 9q

5. บทสรุป

จากการออกแบบขั้นตอนในการแปลงข้อมูลภาพจากราสเตอร์เป็นเวคเตอร์แบบอัตโนมัติ โดยการเก็บข้อมูลภาพที่มีความชัดเจนและความสม่ำเสมอในระดับความเข้มของสี ผ่านเครื่องสแกนเนอร์ และทำการประมวลผลข้อมูลโดยการหาขอบภาพด้วยวิธี Range Edge การทำขอบภาพให้บางด้วยวิธี Simple Thining การหาจุดรวมของเส้น(Node) และหาส่วนของเส้นตรง (Line Tracking) ด้วยวิธีหาความสัมพันธ์ของจุดกับจุดรอบ ๆ ตามตารางขนาด (3x3) การหารูปเหลี่ยม (Polygon) โดยวิธี Labeling ซึ่งสามารถสรุปผลได้ดังนี้

5.1 การหาโครงร่างของภาพที่บาง (Skeletonizing) ที่ใช้เทคนิคการหาขอบภาพ (Edge Detection) โดยในการทดลองได้ทำการเปรียบเทียบการหาขอบภาพด้วยวิธี Sobel Edge กับ Range Edge ซึ่งปรากฏว่าการใช้วิธี Sobel Edge จะต้องสุ่มหาค่า Threshold ที่เหมาะสม เพื่อให้ได้ขอบภาพที่บางและครบถ้วน ทำให้เกิดการประมวลผลหลายครั้งในการสุ่มหาค่า และขอบภาพที่ได้ในกรณีที่ขอบครบถ้วนก็จะยังมีความหนากว่าวิธี Range Edge แต่ถ้าหากใช้วิธี Range Edge จะสามารถกำหนดขนาดความหนาบางของขอบที่ต้องการได้ โดยวิธีการกำหนดขนาดตารางตัวกรองในการหาขอบภาพกับค่า Threshold ที่ต้องการ เช่น กำหนดค่า Threshold>1 ซึ่งจะทำให้ขอบภาพหนา ดังนั้นจึงกำหนดขนาดขอบด้วยตารางตัวกรองขนาด 1x1 ประกอบด้วย ก็จะทำให้ได้ขอบภาพที่ค่อนข้างบางและครบถ้วน ดังนั้นในการทำขอบภาพให้บางนั้นจึงเพียงแค่ใช้วิธี Simple Thining Edge ซึ่งเป็นวิธีที่ไม่ยุ่งยากซับซ้อนและประมวลผลได้รวดเร็ว

5.2 การหาจุดรวมของเส้น (Node) นั้นจะมีตัวกรองหาความสัมพันธ์ของจุดในตาราง ซึ่งผลลัพธ์ที่ออกมากับการทดลองภาพแผนที่ภาคเหนือ ของประเทศไทย ปรากฏว่าได้ Node ครบถ้วนซึ่งทำให้สามารถใช้ Node เป็นตัวเริ่มในการหาจุดเริ่มต้นและจุดสิ้นสุดของเส้นต่าง ๆ ในภาพได้ครบถ้วนด้วย

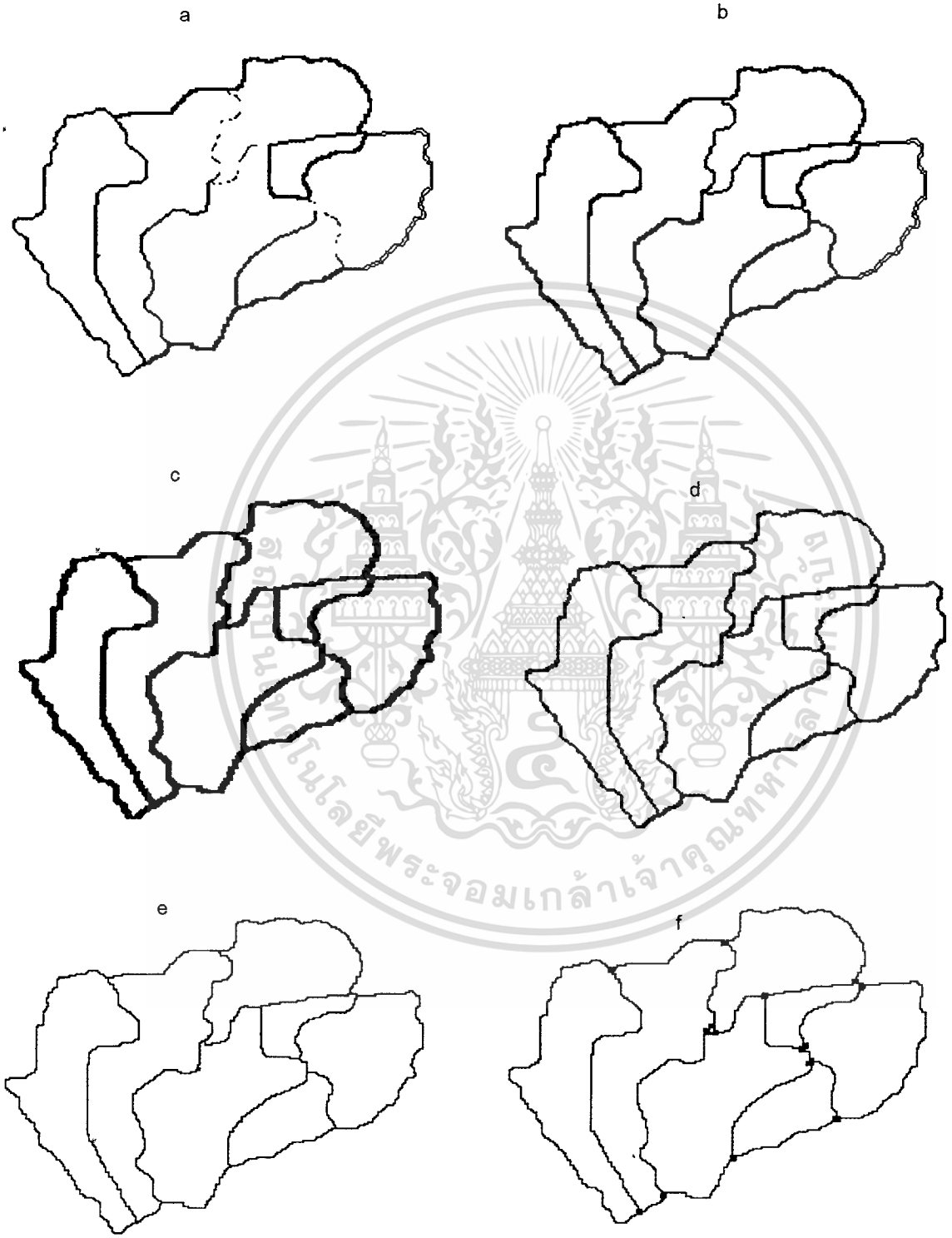
5.3 การหาส่วนของเส้น (Line Tracking) or (Line Sgement) โดยการหาความสัมพันธ์ในทิศทางของจุดในตัวกรองตารางที่กำหนด ปรากฏว่า ถ้าต้องการให้ผลลัพธ์ข้อมูลที่เป็นรูปแบบ Vector มีความใกล้เคียงกับรูปแบบราสเตอร์มากที่สุด ควรจะกำหนดจำนวนครั้งในการเปลี่ยนแปลงของทิศทางความต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องของจุดในการกำหนดจุดเริ่มต้นและจุดสิ้นสุดของเส้นให้น้อยที่สุด แต่ก็จะได้จำนวนเส้นของภาพมากมาย และถ้ากำหนดมากเกินไปก็จะทำให้ภาพผลลัพธ์ที่ได้ในรูปแบบเวคเตอร์ไม่เหมือนภาพต้นแบบ แต่ก็จะได้จำนวนเส้นของภาพน้อยลง

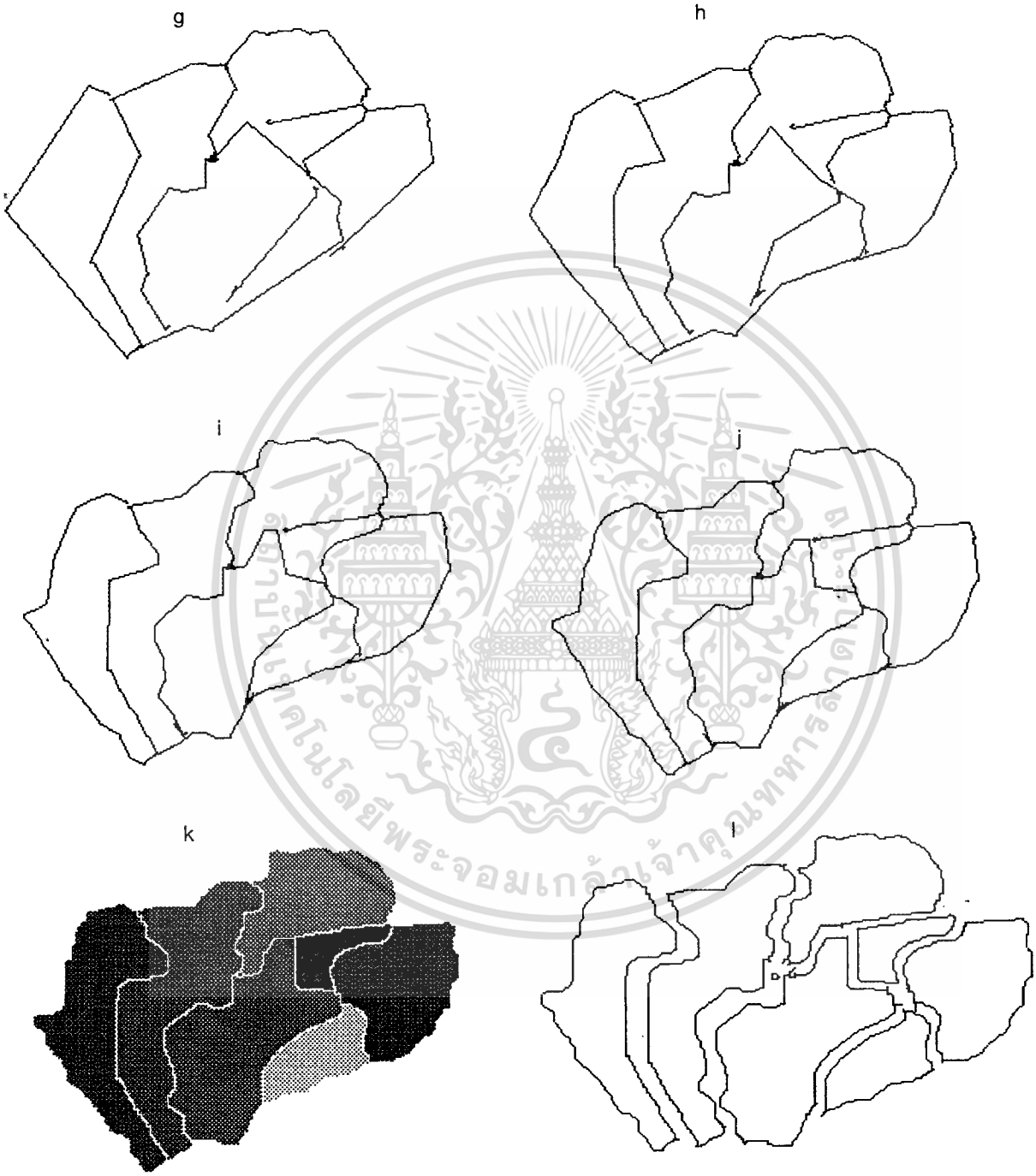
5.4 การหาส่วนของรูปเหลี่ยมหรือวัตถุในภาพ (Segment Merge) โดยการหาส่วนของเส้นที่เป็นส่วนประกอบของวัตถุในภาพเช่น รูปเหลี่ยม (Polygon) จะต้องทำการ Labeling หาพื้นที่ของแต่ละ Polygon ก่อนที่จะหาว่าเส้นที่เป็นส่วนประกอบของ Polygon ซึ่งทำให้ต้องเสียเวลาในประมวลผลเพิ่มขึ้น หากสามารถคิดค้นตัวกรองที่กำหนดความสัมพันธ์ของเส้นว่าเส้นใดเป็นส่วนประกอบของรูปเหลี่ยมได้เลย ก็จะทำให้ใช้เวลาในการประมวลผลรวดเร็วยิ่งขึ้น อีกทั้งวิธีการดังกล่าวยังทำให้เกิดปัญหาของส่วนของเส้นตรงที่มีความยาวทำให้เส้นที่เป็นส่วนประกอบของรูปเหลี่ยมเกินออกมาเช่นรูป 9p และ 9r สำหรับผลลัพธ์ที่เป็นรูปแบบเวคเตอร์ที่ได้จากการแปลงข้อมูลดังรูปที่ 10 จะเห็นได้ว่าในการเก็บเป็น จุด เส้น รูปเหลี่ยม จุดรวมของเส้น นั้นจะมีความสัมพันธ์ซึ่งกันและกัน ซึ่งทำให้สามารถนำข้อมูลที่แปลงได้มาจัดเป็นรูปแบบโครงสร้างข้อมูลของ Software ที่มีการเก็บแบบเวคเตอร์ เช่น Auto-CAD หรือ GIS Application Software ซึ่งจะทำให้สะดวกในการประมวลผลข้อมูลภาพยิ่งขึ้น และเนื่องจากภาพที่ใช้ในการทดลองในบทความนี้เป็นภาพแผนที่ภาคเหนือ ของประเทศไทย ซึ่งมีการแบ่งสีอย่างชัดเจน ทำให้ไม่จำเป็นต้องทำ Preprocessing แต่ในกรณีที่เป็นภาพที่มีการกระจายของกลุ่มของสี เช่นภาพวิว ภาพคน จำเป็นต้องมีการทำ Preprocessing เพื่อหาวัตถุเป้าหมายที่ต้องการเปลี่ยนเป็นข้อมูลที่เป็นรูปแบบเวคเตอร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



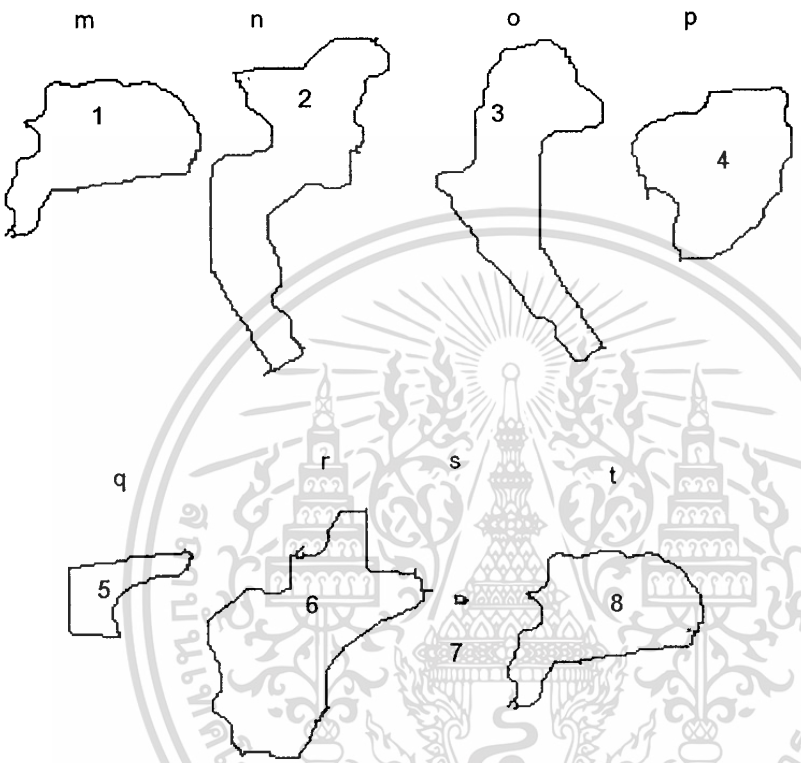
รูปที่ 9 แสดงผลลัพธ์จากการแปลงข้อมูลราสเตอร์เป็นเวคเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



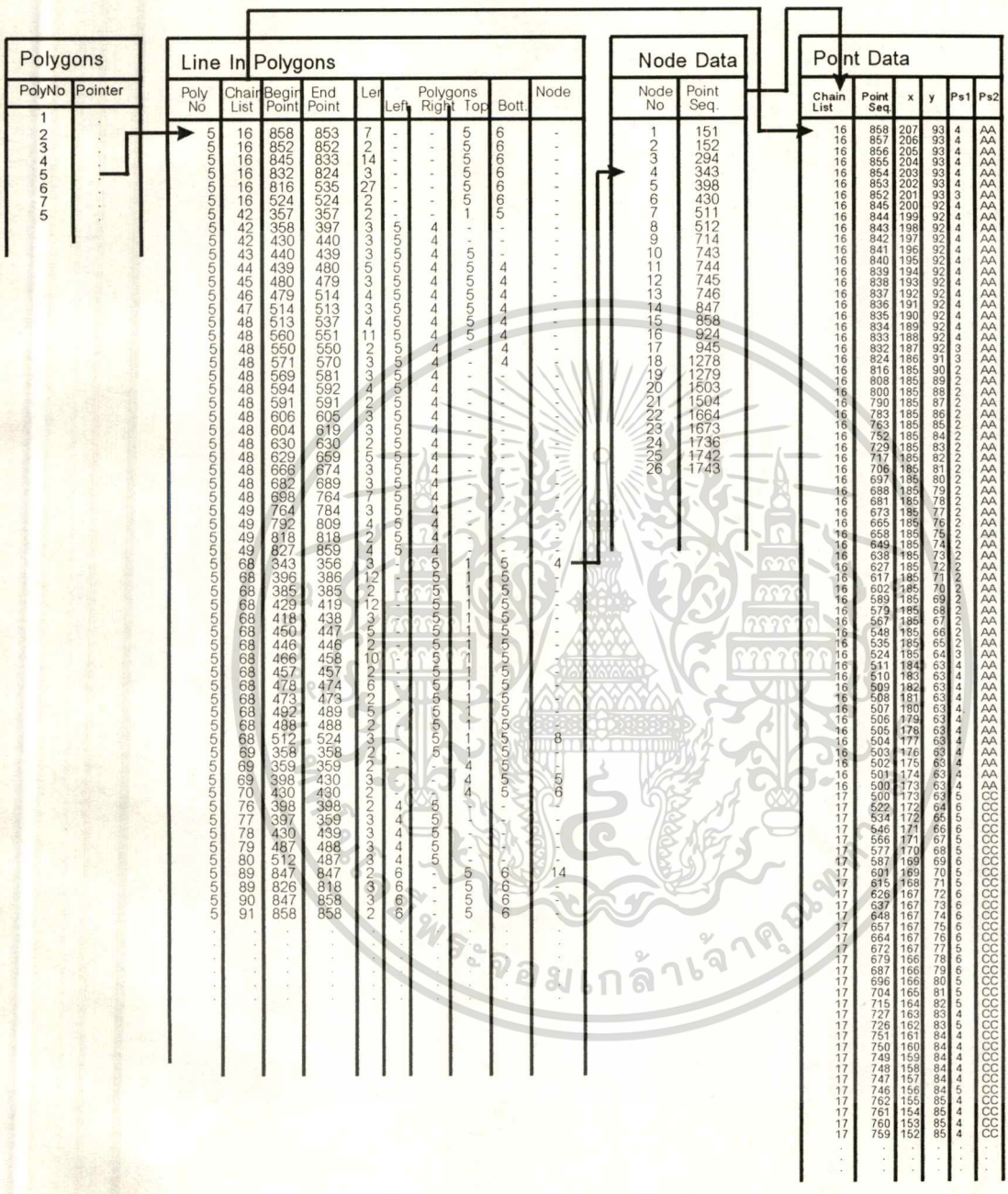
รูปที่ 9 แสดงผลลัพธ์จากการแปลงข้อมูลราสเตอร์เป็นเวคเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9 แสดงผลลัพธ์จากการแปลงข้อมูลราสเตอร์เป็นเวคเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10 แสดงตัวอย่างผลลัพธ์ในความสัมพันธ์ของข้อมูลเวคเตอร์ที่ได้จากการแปลงข้อมูลราสเตอร์เป็นเวคเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. Star Jeffrey and Estes John, **Geographic Information System An Introduction**, Prentice Hall, 1990.

2. Pitas Ioannis,**Digital Image Processing Algorithms**, Pentice Hall,1993.

3. Drummond Jane, Essen van Rob and Boulerie Pascal, **Some considerations on ctorizing Algorithms**,ITC Journal, P153-157, 1991-3.

4. J Peuquet Donna, **A Conceptual Framework And Comparison Of Spatial Data Models**, Cartographica, Vol.21, No. 4, P 66-113, 1984.

5. Heng Zhou, **Integration Of Data Format For Vector And Raster Based GIS**, AIT Thesis, No.CS94-18, 1994.

6. Chang Shi-Kuo, **Principle Of Pictorial Information System Design**, Prentice-Hall International Editions, 1990.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาวกิ่งกาญจน์ วงศ์วิภาพร
วัน เดือน ปีเกิด	5 มกราคม 2511
วุฒิการศึกษาระดับปริญญาตรี	ครุศาสตร์บัณฑิต สาขาคอมพิวเตอร์
สถานที่สำเร็จการศึกษา	วิทยาลัยครูจันทรเกษม
ปีที่สำเร็จการศึกษา	2532
ประสบการณ์การทำงาน	นักเขียนโปรแกรม
	นักวิจัยและพัฒนาระบบ
อาชีพปัจจุบัน	นักวิจัยและพัฒนาระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า