สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตัวตรวจหามอร์โฟโลยี

MORPHOLOGICAL DETECTOR

นายบัณฑิต พญาพรหม
นายวรพจน์ ปวงคำ
นายสิทธิศักดิ์ บุญแปลง

ป๗.
ม 2597
2544
เลขหมู่.....
เลขทะเบียน 45856
วัน, เดือน, ปี 19 ก.พ. 2546
.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม ภาควิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
b11681544

MORPHOLOGICAL DETECTOR

BUNDIT PHAYAPROM

WORRAPOJ POUNGKHUM

SITTISAK BOONPLANG

A THESIS SUBMTTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์	ตัวตรวจหามอร์โฟโลยี	
	MORPHOLOGICAL DETECTOR	
นักศึกษาผู้จัดทำ	นายบัณฑิต พญาพรหม	รหัสประจำตัว 42015393
	นายวรพจน์ ปวงคำ	รหัสประจำตัว 42015398
	นายสิทธิศักดิ์ บุญแปลง	รหัสประจำตัว 42015411
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมการวัดคุม	
ปีการศึกษา	2544	

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
ผศ.สาธิต อินทจักร์	

วัน/เดือน/ปี ที่สอบ วันที่ 19 เดือน มีนาคม พ.ศ. 2545

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ตัวตรวจหามอร์โฟโลยี
	MORPHOLOGICAL DETECTOR
นักศึกษาผู้จัดทำ	นาย บัณฑิต พญาพรหม
	นาย วรพจน์ ปวงคำ
	นาย สิทธิศักดิ์ บุญแปลง
อาจารย์ที่ปรึกษา	ผศ. สาธิต อินทจักร์
ปีการศึกษา	2544

บทคัดย่อ

ในปริญญานิพนธ์นี้จะมานำเสนอตัวตรวจหามอร์โฟโลยี เพื่อนำมาใช้ตรวจหาข้อความ ซึ่งเป็นกระบวนการในการวิเคราะห์เพื่อใช้ดึงข้อความออกมาจากรูปในการตรวจหาข้อความมีขั้นตอนอยู่สี่ขั้นตอน (1) โดยนำรูปภาพที่มีข้อความเป็นส่วนประกอบ มากำจัดสัญญาณรบกวนออกจากรูปภาพก่อนเพื่อทำให้ภาพมีความคมชัดมากขึ้น แล้วนำตัวปฏิบัติการ gradient มาทำการหาขอบภาพอีกทีหนึ่ง รูปภาพที่ได้จะปรากฏขอบของข้อความอย่างชัดเจน จากนั้นนำตัวปฏิบัติการ Thresholding มากระทำกับรูปภาพเพื่อทำให้ภาพเป็นภาพขาวดำ (2) ภาพที่ได้ยังมีสัญญาณรบกวนอยู่ต้องนำมาทำการกำจัดสัญญาณรบกวนออกอีกครั้งหนึ่ง(3)นำภาพผลลัพธ์ที่ได้มาทำการ Labeling คือการกำหนดตำแหน่งข้อความ (4) เมื่อทราบตำแหน่งของข้อความแล้วจะทำการ ดึงข้อความออกมาจากรูปภาพทีละตัวอักษร ภาพผลลัพธ์ที่ได้จะปรากฏแต่เพียงข้อความที่อยู่ในภาพเท่านั้น

สำหรับการทดลองจะใช้ตัวดำเนินการ Erosion และ Dilation ในการทดลองการดึงข้อความนี้ ปรากฏว่าให้ผลลัพธ์อยู่ในเกณฑ์ที่ดี แต่ถ้าพิจารณาลึกลงไปถึงการแยกตัวอักษร (Charecter segmentation) ที่ติดกัน ก็ยังเป็นปัญหาอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Morphology Detector
Authors	Mr.Bundit Phayaprom
	Mr.Worrapoj Poungkhum
	Mr.Sittisak Boonplang
Thesis Adviser	Asst.Prof. Sathit Intrajak
Year	2001

ABSTRACT

This thesis proposes a morphological detector to detect a document image.The document analysis is the early process of the object character recognition.This analysis use to extract a text from a graphic object.The document detection scheme composes of four steps: (1) morphological filtering to remove noise and enhance the edge image, (2) detecting the edge by using morphological gradient; after that, thresholding method is used to segment into a binary image, (3) labeling the objects in the binary image to get the position of each object, (4) finally, text extracting.

This scheme almost uses the fundamental morphological operators that consists of dilation and erosion. In experimentation, our algorithm can extract the objects from the document image sussesfully.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับคำปรึกษาและความอนุเคราะห์จาก-ผศ.สาธิต อินทจักร์ ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูงที่ได้ให้คำแนะนำและคำปรึกษาในการวิจัย และช่วยตรวจสอบแก้ไขในข้อผิดพลาดต่างๆ พร้อมทั้งวิธีแก้ปัญหา

ขอขอบพระคุณภาควิชาวิศวกรรมการวัดคุมทางอุตสาหกรรมที่เอื้อเฟื้อสถานที่จนสามารถทำปริญญานิพนธ์นี้สำเร็จ

ขอขอบคุณ เพื่อนทุกท่าน ซึ่งคอยเป็นกำลังใจที่ดี ในการทำปริญญานิพนธ์ครั้งนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ผู้วิจัยขอบมอบแด่ บิดา มารดา

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญภาพ.....VII

บทที่ 1 บทนำ.....1

1.1 แนวความคิดในการนำเสนอปริญญานิพนธ์.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....2

1.3 ขอบเขตของปริญญานิพนธ์.....2

1.4 รายละเอียดของปริญญานิพนธ์.....2

บทที่ 2 ทฤษฎีมอร์โฟโลยี ที่ใช้ประมวลผลกับรูปขาวดำ.....4

2.1 ทฤษฎีมอร์โฟโลยี.....4

2.2 ไบนารี่มอร์โฟโลยี (Binary Morphology).....4

2.2.1 การอีโรชั่น (Erosion).....6

2.2.2 การไดเรชั่น (Dilation).....8

2.2.3 การโอเพนนิ่ง (Opening).....10

2.2.4 การโคลสซิ่ง (Closing).....11

บทที่ 3 ทฤษฎีมอร์โฟโลยีที่ใช้ประมวลผลกับรูปเกรย์.....12

3.1 ตัวดำเนินการมอร์โฟโลยี.....12

3.1.1 ตัวดำเนินการ Erosion.....13

3.1.2 ตัวดำเนินการ Dilation.....15

3.1.3 ตัวดำเนินการ Opening.....18

3.1.4 ตัวดำเนินการ Closing.....20

3.1.5 ตัวดำเนินการ Gradient.....22

3.1.6 ตัวดำเนินการ Thresholding.....24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 4 การตรวจจับข้อความออกจากรูปภาพ.....26

4.1 Morphological Text Extraction.....26

4.1.1 ขั้นตอนที่ 1.....28

4.1.2 ขั้นตอนที่ 2.....30

4.1.3 ขั้นตอนที่ 3.....31

4.1.4 ขั้นตอนที่ 4.....33

4.2 ตัวอย่างและการพิจารณา.....34

บทที่ 5 ผลการทดลอง.....35

บทที่ 6บทสรุปและวิจารณ์.....41

บรรณานุกรม.....43

ภาคผนวก.....44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 แสดงรูปต้นแบบและ SE.....5
2.2 การกระทำทุกๆ ตำแหน่งในรูปภาพด้วยตัวดำเนินการ Erosion.....7
2.3 ภาพที่ผ่านการ Erosion แล้ว.....8
2.4 การกระทำทุกๆ ตำแหน่งในรูปภาพด้วยตัวดำเนินการ Dilation.....9
2.5 ภาพที่ผ่านการDilation แล้ว.....9
2.6 ภาพที่ผ่านการ Opening แล้ว.....10
2.7 ภาพที่ผ่านการ Closing แล้ว.....11
3.1 แสดงรูปต้นแบบและ SE.....12
3.2 แสดงรูปที่ผ่านการ Erosion แล้ว.....15
3.3 แสดงรูปที่ผ่านการ Dilation แล้ว.....18
3.4 แสดงรูปที่ผ่านการ Opening แล้ว.....20
3.5 แสดงรูปที่ผ่านการ Closing แล้ว.....22
3.6 แสดงรูปที่ผ่านการ Gradient แล้ว.....24
3.7 แสดงรูปที่ผ่านการ Thresholding แล้ว.....25
4.1 รูปต้นแบบที่ใช้ในการวิเคราะห์.....27
4.2 แสดงรูปผลลัพธ์จากการกำจัดสัญญาณรบกวน (Y_{b1}).....28
4.3 แสดงรูปที่ได้จากการหาขอบภาพ (e_5).....29
4.4 แสดงรูปแบบการกระทำของ g_1 และ g_2.....30
4.5 แสดงรูปที่ได้จาการ Threshold.....30
4.6 แสดงรูปผลลัพธ์ที่ได้จากการ Dilation และ Erosion.....31
4.7 แสดงรูปสำเร็จในการดึงข้อความออกจากรูปภาพ.....34
5.1 ภาพต้นแบบและผลลัพธ์.....36
5.2 ภาพต้นแบบและผลลัพธ์.....36
5.3 ภาพต้นแบบและผลลัพธ์.....37
5.4 ภาพต้นแบบและผลลัพธ์.....37
5.5 ภาพต้นแบบและผลลัพธ์.....38
5.6 ภาพต้นแบบและผลลัพธ์.....38
5.7 ภาพต้นแบบและผลลัพธ์.....38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ หน้า

5.8 ภาพต้นแบบและผลลัพธ์ 39

5.9 ภาพต้นแบบและผลลัพธ์ 39

5.10 ภาพต้นแบบและผลลัพธ์ 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 แนวความคิดในการนำเสนอปริญญานิพนธ์

แนวความคิดในการนำเสนอปริญญานิพนธ์ฉบับนี้ มีจุดประสงค์มาจากความต้องการที่จะนำข้อมูลภาพที่มีข้อความประกอบอยู่ มาทำการวิเคราะห์แล้วแยกเฉพาะข้อความออกมา โดยจะเน้นไปที่การวิเคราะห์เอกสารที่มีทั้งตัวอักษรและรูปภาพ โดยใช้ตัวดำเนินการตรวจหามอร์โฟโลยี (Morphology Dectector)

วิธีการที่จะตรวจจับข้อความที่อยู่ในรูปภาพจะประกอบด้วยเทคนิคต่างๆ ที่สามารถนำมาใช้ตรวจจับข้อความออกจากรูปภาพ โดยข้อมูลของแต่ละภาพจะถูกพิจารณาให้เป็นเซต และจะถูกดำเนินการด้วยตัวดำเนินการมอร์โฟโลยีเพื่อหาขอบภาพและกำจัดสัญญาณรบกวน การหาขอบภาพกระทำเพื่อให้เห็นความแตกต่างของวัตถุภายในรูปภาพการกำจัดสัญญาณรบกวนของภาพกระทำเพื่อกำจัดสัญญาณรบกวนที่ไม่พึงประสงค์ที่อยู่ในรูปภาพให้หมดไป การตรวจจับข้อความที่อยู่ในภาพก็จะใช้การวิเคราะห์ดังที่ได้กล่าวไว้ข้างต้นนี้

เทคนิคในการดึงข้อความออกมาจากรูปภาพ ที่จะได้นำเสนอในปริญญานิพนธ์ฉบับนี้มีขั้นตอนในการดึงข้อความออกมาจากรูปภาพอยู่ด้วยกัน 4 ขั้นตอน ซึ่งในแต่ละขั้นตอนที่นำเสนอนี้ล้วนแต่ใช้ตัวดำเนินการมอร์โฟโลยีทั้งสิ้น ซึ่งประกอบด้วยขั้นตอนต่างๆ ดังต่อไปนี้

ขั้นตอนที่ 1 เป็นการปรับปรุงรูปภาพเกรย์ (Gray-scale Image) ที่มีข้อความเป็นส่วนประกอบ มีขั้นตอนย่อยอีก 3 ขั้นตอนคือ ขั้นตอนแรกเป็นการกำจัดสัญญาณรบกวนที่อยู่ในรูปภาพเกรย์ที่มีข้อความอยู่ ซึ่งอาจจะมีสัญญาณรบกวนต้องทำการกำจัดสัญญาณรบกวนออกเสียก่อน ขั้นตอนต่อไปเป็นหาขอบภาพโดยนำเอารูปภาพที่กำจัดสัญญาณรบกวนแล้วนั้น มาทำการหาขอบภาพด้วยวิธีการ gradient ซึ่งเป็นวิธีการตรวจจับความเปลี่ยนแปลงของการดำเนินการมอร์โฟโลยี รูปภาพผลลัพธ์จะปรากฏเป็นรูปขอบของข้อความ ขั้นตอนสุดท้ายเป็นการแปลงภาพจากรูปภาพเกรย์เป็นรูปภาพขาวดำด้วยวิธีการ Threshold เพื่อทำการหาตำแหน่งของข้อความที่อยู่ในรูปภาพในขั้นตอนที่ 4 ต่อไป รูปภาพผลลัพธ์จะเป็นรูปขาวดำที่ปรากฏเป็นรูปขอบของข้อความและรูปขอบของวัตถุอื่นที่เป็นส่วนประกอบของรูปภาพซึ่งเป็นวัตถุที่ไม่ใช้ส่วนประกอบของข้อความ เราจะทำการกำจัดวัตถุที่เราไม่ต้องการออกไปในขั้นตอนต่อไป

ขั้นตอนที่ 2 เป็นการเชื่อมต่อ pixel ของข้อความที่ขาดจากกัน และเป็นการกำจัดสัญญาณรบกวนและวัตถุที่ไม่ใช้ส่วนประกอบของข้อความออกไป โดยใช้ตัวดำเนินการ Dilation ในการเชื่อมต่อ pixel ของข้อความที่ขาดจากกัน โดยเลือก Structuring Element (SE) ที่เหมาะสม เพื่อให้การเชื่อมต่อ pixel มีประสิทธิภาพมากที่สุด จากนั้นนำรูปภาพผลลัพธ์จากการ Dilation มาทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำจัดสัญญาณรบกวนและวัตถุอื่นที่ไม่ใช่ข้อความออกไปด้วยวิธีการ Erosion ซึ่งจะสามารถกำจัดสัญญาณรบกวนออกไปได้มากน้อยเพียงใดขึ้นอยู่กับการเลือกใช้ SE ที่เหมาะสมเช่นกัน รูปภาพผลลัพธ์จะเป็นรูปภาพที่ปรากฏเฉพาะขอบของข้อความและมี pixel ที่เชื่อมต่อกันทั้งหมด

ขั้นตอนที่ 3 เป็นขั้นตอนในการกำหนดค่าของตัวอักษรแต่ละตัวที่อยู่ในรูปภาพโดยวิธีการ Labeling มีวิธีการคือให้ pixel ที่เชื่อมติดกันมีค่าในของ Labeling ที่เท่ากัน

ขั้นตอนที่ 4 เป็นขั้นตอนในการวิเคราะห์ขอบเขตของข้อความโดยหาค่าจุดต่ำสุดและสูงสุดของความกว้างและความยาวของแต่ละตัวอักษร โดยที่แต่ละตัวอักษรจะมีหมายเลขกำกับ เราจะได้ค่าจุดสูงสุดและจุดต่ำสุดของความกว้างและความยาวออกมาเป็นกรอบสี่เหลี่ยม ล้อมตัวอักษรที่ต้องการดึงออกมา แต่ภายในกรอบสี่เหลี่ยมบางกรอบอาจจะยังเป็นวัตถุที่ไม่ใช่ตัวอักษรอยู่ การกำหนดเงื่อนไขในการเป็นตัวอักษรจะได้กล่าวในบทที่ 4 เมื่อได้กรอบสี่เหลี่ยมที่ภายในเป็นตัวอักษรจริงๆ แล้ว จะทำการนำเอาค่าต่ำสุดและสูงสุดนั้น ไปทำการดึงเอาตัวอักษรจากในรูปต้นแบบออกมา ก็จะได้รูปของข้อความที่ดึงออกมาจากรูปภาพ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาการทำงานของตัวดำเนินของ Morphology
2. นำตัวดำเนินการของ Morphology มาใช้ในการตรวจจับข้อความที่อยู่ในรูปภาพ

1.3 ขอบเขตของปริญญานิพนธ์

ส่วนทฤษฎีประกอบด้วย

- ทฤษฎี Morphology ที่ใช้กับภาพแบบ Binary ซึ่งเป็นภาพขาวดำ
- ทฤษฎี Morphology ที่ใช้กับภาพแบบ Gray ซึ่งเป็นภาพสีเทา
- ทฤษฎี Labeling การกำหนดตำแหน่งของตัวอักษรที่อยู่ในรูปภาพ

ในงานวิจัยฉบับนี้ได้จำลองการทำงานของ Morphology ด้วยโปรแกรมภาษา C โดยนำตัวดำเนินการพื้นฐานของมอร์โฟโลยีคือ Erosion, Dilation มาใช้ดำเนินการเพื่อนำไปใช้ในการหาขอบภาพ โดยทำเป็นโปรแกรมสำเร็จรูปที่สามารถตรวจจับข้อความออกมาจากรูปภาพได้

1.4 รายละเอียดของปริญญานิพนธ์

ภายในปริญญานิพนธ์เล่มนี้แบ่งออกเป็น 6 บทด้วยกัน โดยมีรายละเอียดแต่ละบทดังนี้

บทที่ 1 เป็นการกล่าวนำและวัตถุประสงค์ในการทำปริญญานิพนธ์ และขอบเขตของปริญญานิพนธ์

บทที่ 2 เป็นการกล่าวถึงทฤษฎีมอร์โฟโลยีที่ใช้ประมวลผลกับรูปขาวดำ

บทที่ 3 เป็นการกล่าวถึงทฤษฎีมอร์โฟโลยีที่ใช้ประมวลผลกับรูปเกรย์สเกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 เป็นการกล่าวถึงการนำเอาทฤษฎีมอร์โฟโลยีทั้งที่ประมวลผลกับรูปขาวดำและ-ประมวลผลกับรูปภาพเกรย์สเกลมาประยุกต์ใช้งานร่วมกันในการตรวจจับข้อความออกจากรูปภาพ โดยได้นำเสนอเป็นขั้นตอนมีอยู่4 ขั้นตอนและโปรแกรมที่นำมาใช้ในการตรวจจับข้อความออกจากรูปภาพ

บทที่ 5 เป็นการกล่าวถึงผลการทดลองและการใช้งานของโปรแกรมให้มีประสิทธิภาพ

บทที่ 6 เป็นการกล่าวถึงการสรุปผลการวิจัยและข้อเสนอแนะในการพัฒนาให้โปรแกรมมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีมอร์โฟโลยีที่ใช้ประมวลผลกับรูปขาวดำ

2.1 ทฤษฎีมอร์โฟโลยี

ในบทนี้จะเป็นการกล่าวถึงทฤษฎีมอร์โฟโลยีที่นำมาใช้ประมวลผลกับรูปขาวดำ ที่เกี่ยวข้องในการนำไปใช้งานในการวิเคราะห์รูปภาพ มอร์โฟโลยีจัดเป็นคณิตศาสตร์แขนงหนึ่งที่อยู่ในพวกของ พีชคณิต ที่สามารถนำมาใช้ในการประมวลผลภาพได้อย่างมีประสิทธิภาพ เนื่องจากตัวดำเนินการของมันสามารถสร้างขึ้นมาให้เหมาะสมตามลักษณะการใช้งาน และสามารถจำแนกวัตถุที่มีรูปร่างต่างกันในภาพได้อย่างมีเหตุผล ผู้ที่ริเริ่มนำทฤษฎีนี้มาใช้คือ Hadwiger โดยในช่วงแรกๆ จะใช้ประมวลผลกับเซตที่มีค่าเป็น 0 กับ 1 ซึ่งต่อมาได้มีผู้พัฒนาเพื่อนำมาใช้กับการประมวลผลภาพไบนารี และได้พัฒนาเรื่อยมาจนสามารถนำไปใช้กับภาพเกรย์สเกลได้

2.2 ไบนารี่มอร์โฟโลยี (Binary Morphology)

ตัวที่ถูกดำเนินการ (Operand) โดยมอร์โฟโลยีจะมีสองส่วนด้วยกันคือ ส่วนที่เป็นข้อมูลภาพที่จะถูกประมวลผล และส่วนของ Structuring Element(SE) สำหรับภาพไบนารีจะแทนข้อมูลภาพด้วยเซต F และ SE แทนด้วยเซต K ที่มีรูปร่างเฉพาะซึ่งจะเหมือนกับตัวแปรที่จะนำไปดำเนินการกับภาพ ตัวดำเนินการพื้นฐานของมอร์โฟโลยีมีอยู่สองตัวคือ Dilation กับ Erosion โดยตัวดำเนินการ Dilation ของภาพไบนารี F ด้วย K ที่เป็นไบนารีเช่นกัน จะแทนด้วย $F \oplus K$ ที่เป็นสมาชิกทั้งหมดที่ซึ่งมีสมาชิกบางส่วนอยู่ใน K หรืออยู่ใน F ซึ่งแสดงในสมการที่ (1)

$$F \oplus K = \{f + k | \exists k \in K \wedge f \in F\} = \bigcup_{f \in F} K_f \qquad (1)$$

$$F \ominus K = \{x + k | x + k \in F \forall k \in K\} = \bigcap_{k \in K} F_{-k} \qquad (2)$$

ในสมการที่ (2) เป็นตัวดำเนินการ Erosion ของภาพไบนารี F ด้วย K ซึ่งเขียนแทนด้วย $F \ominus K$ ที่ x เป็นสมาชิกทั้งหมดที่ซึ่ง $x + k \in F$ โดยที่ k ทุกตัวอยู่ใน K ผลจากตัวดำเนินการ Dilation และ Erosion เมื่อนำมารวมกับ SE อีกครั้งหนึ่งทำให้เกิดตัวดำเนินการใหม่อีกสองตัวคือ Opening ซึ่งแทนด้วย $F \circ K$ และ Closing แทนด้วย $F \bullet K$ โดยตัวดำเนินการทั้งสองแสดงไว้ในสมการที่ (3) และ (4) ตามลำดับ

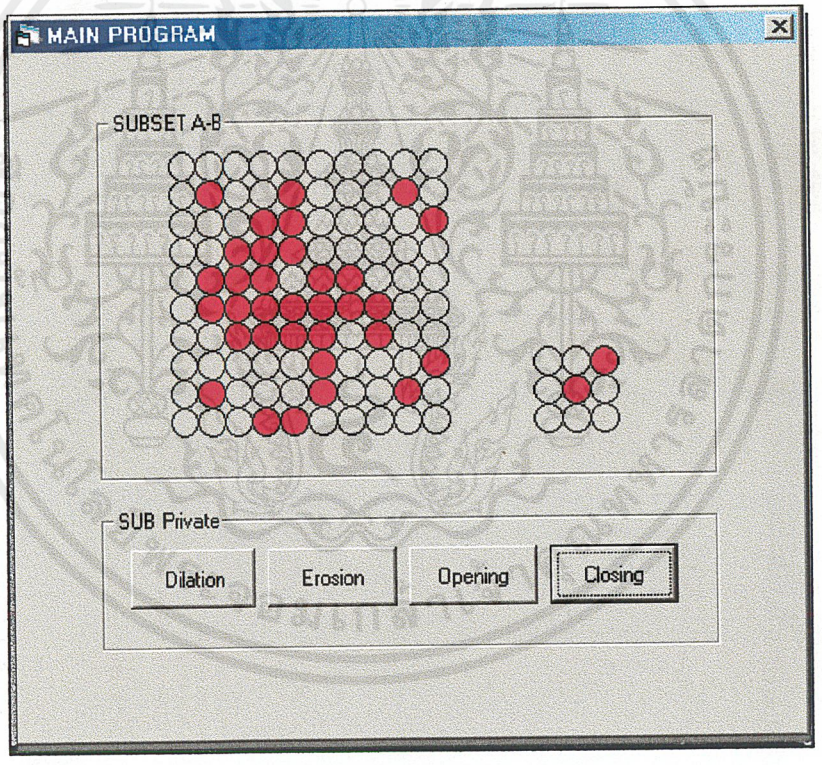
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$F \circ K = (F \ominus K) \oplus K \quad (3)$$

$$F \bullet K = (F \oplus K) \ominus K \quad (4)$$

ในสมการที่ (3) เป็นการนำเอาภาพไบนารี F มากระทำกับ K ด้วยตัวดำเนินการ Erosion เมื่อได้ค่ามาแล้วนำมากระทำกับ K ด้วยตัวดำเนินการ Dilation ในทำนองเดียวกันสมการที่ (4) เป็นการนำเอาภาพไบนารี F มากระทำกับ K ด้วยตัวดำเนินการ Dilation เมื่อได้ค่ามาจึงนำมากระทำกับ K ด้วยตัวดำเนินการ Erosion

จากหลักการและสมการของตัวดำเนินการมอร์โฟโลยีที่ใช้กับภาพขาวดำที่ได้นำเสนอไปแล้วนั้นเป็นการกล่าวถึงฟังก์ชันที่นำมาเป็นเงื่อนไขในการดำเนินการ ต่อไปจะเป็นการจำลองการดำเนินการของตัวดำเนินการมอร์โฟโลยีเพื่อให้ง่ายต่อการทำความเข้าใจได้แสดงภาพต้นแบบและ SE



ภาพที่ 2.1 แสดงภาพต้นแบบและ SE

จากภาพที่ 2.1 เป็นการจำลองเซตของข้อมูลภาพโดยแต่ละจุดแทนด้วยจำนวน 1 Pixel จุดที่มีสีดำมีค่าความเป็นสมาชิกเท่ากับ 1 และจุดที่ว่างเปล่ามีค่าความเป็นสมาชิกเท่ากับ 0 กำหนดให้ข้อมูลภาพต้นแบบมีขนาด 10 X 10 Pixel และ SE มีขนาด 3 X 3 Pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซตของรูปภาพ (F) มีข้อมูลดังนี้

F = { (1,1)(1,4)(1,8)(2,3)(2,4)(2,9)(3,2)(3,3)(3,4)(4,1)(4,2)(4,3)(4,5)(4,6)(5,1)(5,2)(5,3)
(5,4)(5,5)(5,6)(5,7)(6,2)(6,3)(6,4)(6,5)(6,7)(7,5)(7,9)(8,1)(8,5)(8,8)(9,3)(9,4) }

เซตของ SE มีข้อมูลดังนี้

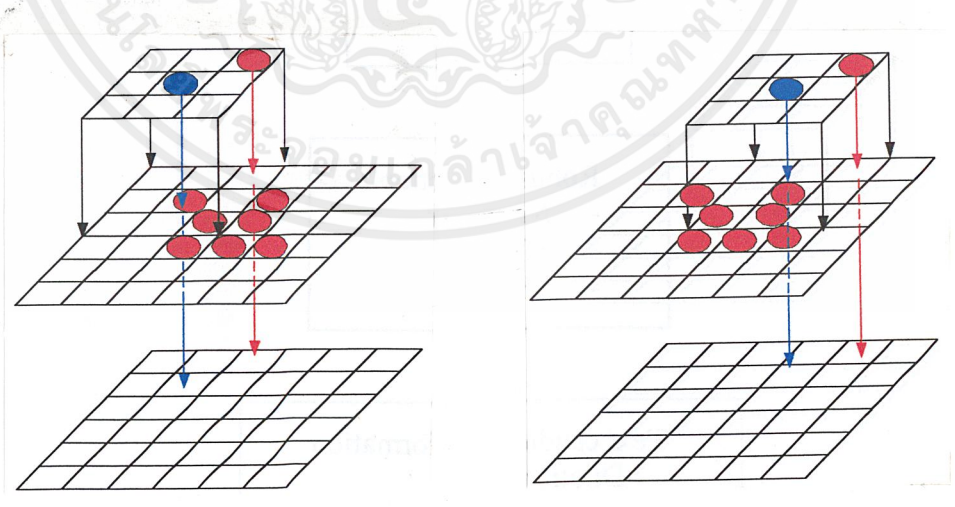
K = { (0,0)(-1,1) } มีจุดออริจิลที่ตำแหน่ง (0,0)

ตัวดำเนินการมอร์โฟโลยีที่ต้องทำการศึกษาได้แก่ Erosion, Dilation, Opening และ Closing ได้แสดงเป็นภาพที่ได้ดำเนินการเรียบร้อยแล้วดังจะได้กล่าวในหัวข้อต่อไปนี้

2.2.1 การอีโรชั่น (Erosion)

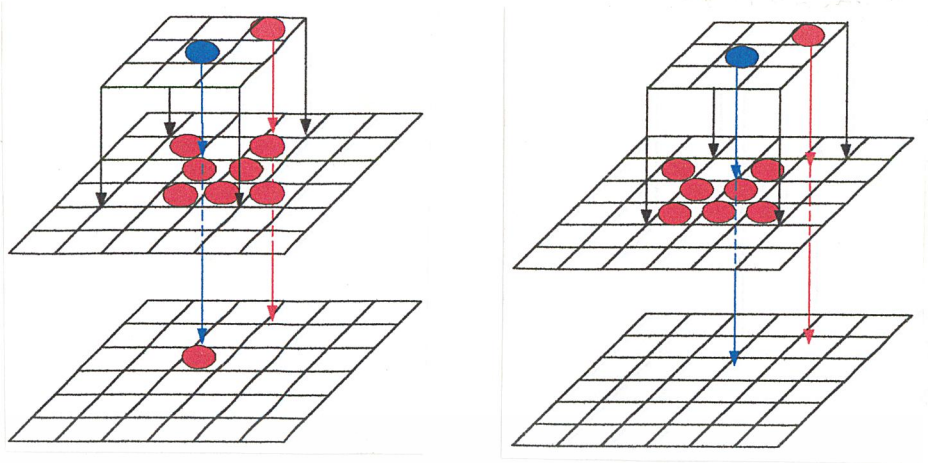
เป็นตัวดำเนินการที่มีผลทำให้จุดที่เป็นสีดำ หรือจุดที่มีค่าความเป็นสมาชิกเท่ากับ 1 ถูกลดทอนเมื่อมีการประมวลผล ลักษณะการลดทอนจะขึ้นอยู่กับการกำหนดค่าสมาชิกและการกำหนดตำแหน่งจุดออริจิลของ SE

จากสมการที่ (2) มีหลักการคือนำ SE เลื่อนไปกระทำทุกๆ ตำแหน่งของรูปภาพ SE จะดำเนินการสับเซตกับข้อมูลของภาพต้นแบบ และทำการเปลี่ยนค่าความเป็นสมาชิกของรูปภาพต้นแบบ ที่ตำแหน่งจุดออริจิลของ SE ทับอยู่ให้มีค่าความเป็นสมาชิกเท่ากับ 1 โดยจะได้แสดงการเลื่อน SE ไปกระทำทุกตำแหน่งของรูปภาพ และรูปภาพผลลัพธ์ ดังต่อไปนี้



(ก) (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

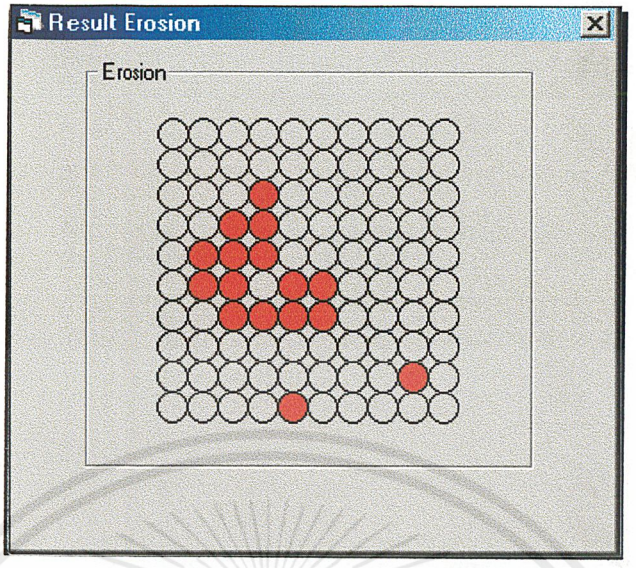


(ค) (ง)

ภาพที่ 2.2 การกระทำทุกๆ ตำแหน่งในรูปภาพด้วยตัวดำเนินการ Erosion

จากภาพที่ 2.2 แสดงให้เห็นถึงการดำเนินการของตัวดำเนินการ Erosion กับภาพต้นแบบ เมื่อให้ SE มีขนาด 3X3 และให้จุดออริจิลของ SE อยู่ที่ตำแหน่งสีน้ำเงิน ให้ภาพด้านบนคือภาพต้นแบบและภาพด้านล่างคือภาพผลลัพธ์ ภาพที่ (ก) เมื่อจุดออริจิลของ SE ซึ่งมีค่าเท่ากับ 1 กระทำกับภาพต้นแบบที่มีค่าเท่ากับ 1 (จุดที่มีสีแดง) เช่นกัน แต่ค่าในรูปต้นแบบที่ SE ทับอยู่นั้นไม่มีค่าเท่ากับ SE ทั้งหมด ก็จะทำให้ภาพผลลัพธ์ถูกลดทอนลงไปและไม่ปรากฏจุดใดๆ จากนั้นทำการเลื่อน SE ไปกระทำกับทุกจุดในรูปภาพต้นแบบ เมื่อจุดในรูปภาพต้นแบบมีค่าเท่ากับ 0 (จุดที่ว่างเปล่า) จะไม่มีการดำเนินการเกิดขึ้น แต่เมื่อเลื่อน SE แล้วเจอจุดที่มีค่าเท่ากับ 1 ในภาพต้นแบบก็จะทำการดำเนินการต่อดังที่ได้กล่าวไปแล้ว และเป็นดังภาพที่ (ข) จะเห็นได้ว่าเมื่อ SE ทับอยู่กับภาพต้นแบบแต่ภาพต้นแบบนั้นมีสมาชิกไม่เหมือนกับ SE ทั้งหมดก็จะทำให้ผลลัพธ์เป็น 0 ต่อไปเมื่อเลื่อน SE ไปกระทำกับภาพต้นแบบที่มีสมาชิกเหมือนกับ SE ทั้งหมดนั้นผลลัพธ์ที่ได้ก็จะเป็น 1 ดังแสดงในรูปที่ (ค) สุดท้ายภาพ (ง) นั้นก็จะเหมือนกับภาพที่ (ก) และ (ข) เมื่อทำการดำเนินการจนจบสิ้นแล้ว ก็จะได้รูปภาพผลลัพธ์ดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.3 ภาพที่ผ่านการ Erosion แล้ว

จากภาพที่ 2.3 เป็นภาพผลลัพธ์ในการดำเนินการ Erosion จะเห็นได้ว่าจำนวนจุดสีดำได้ถูกลดทอนลงตามการกำหนดสมาชิกของ SE ทำให้เซตของผลลัพธ์เป็นดังนี้

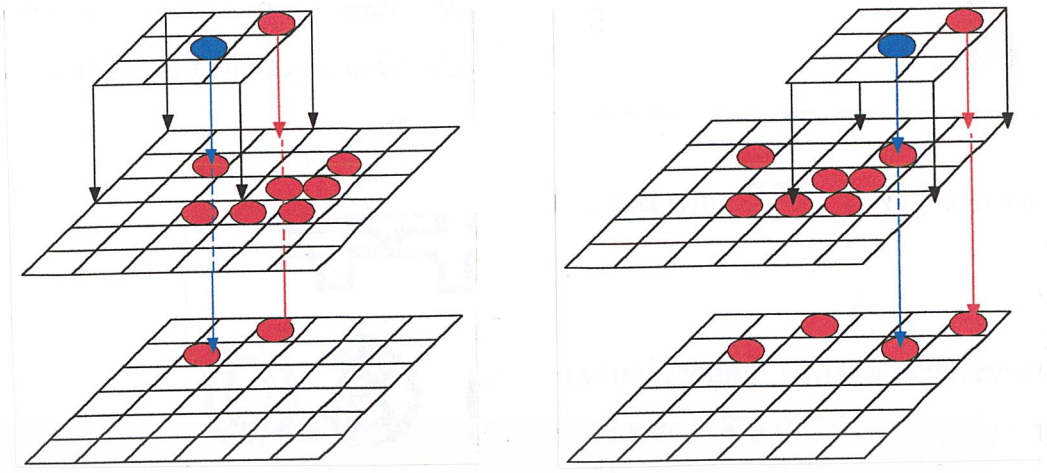
$F \ominus K$ ={(2,3)(3,2)(3,3)(4,1)(4,2)(4,3)(5,1)(5,2)(5,4)(5,5)(6,2)(6,3)(6,4)(6,5)(8,8) (9,4)}

2.2.2 การไดเรชั่น (Dilation)

เป็นการนำ SE ประมวลผลกับรูปภาพ ผลจากการประมวลผลทำให้รูปภาพส่วนที่เป็นสีดำหรือค่าความเป็นสมาชิกเท่ากับ 1 มีการขยายขนาดเพิ่มขึ้น ซึ่งขึ้นอยู่กับการกำหนดค่าสมาชิกและการกำหนดตำแหน่งจุดกำเนิดของ SE

หลักการคือนำเซต K มาทำการรีเฟลคชัน (-K) รีเฟรคชันคือการนำเซตที่ต้องการมาทำการหมุนทั้งแนวแกน x และแกน y หรือหมุนทั้งแถวและหลัก โดยเอาจุด (0,0) เป็นจุดอ้างอิงที่ใช้หมุน แล้วนำผลที่ได้ย้ายไปกระทำทุกๆ ตำแหน่งของรูปภาพ โดยนำมาอินเตอร์เซ็กชั่นกับส่วนของรูปภาพที่ SE ทับอยู่ทุกๆตำแหน่ง นำผลที่ได้ทั้งหมดมาสับเซตกับรูปภาพเดิมอีกครั้งผลลัพธ์ที่ออกมาจะต้องไม่เป็นเซตว่างโดยจะได้แสดงการเลื่อน SE ไปกระทำทุกตำแหน่งของรูปภาพ และรูปภาพผลลัพธ์ ดังต่อไปนี้

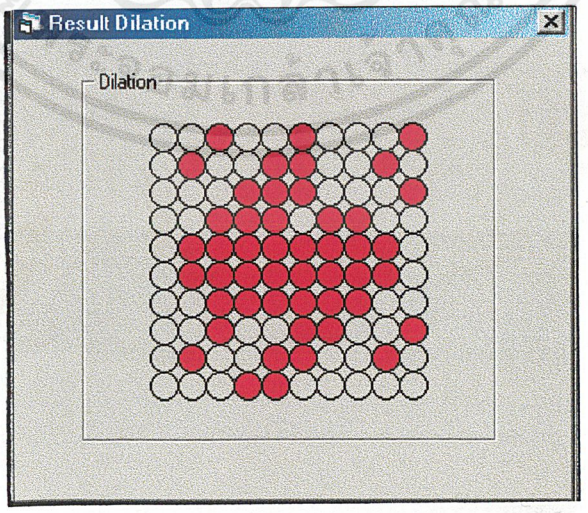
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) (ข)

ภาพที่ 2.4 การกระทำทุกๆ ตำแหน่งในรูปภาพด้วยตัวดำเนินการ Dilation

จากภาพที่ 2.4 แสดงให้เห็นถึงการดำเนินการของตัวดำเนินการ Dilation กับภาพต้นแบบ เมื่อให้ SE มีขนาด 3X3 และให้จุดออริจิลของ SE อยู่ที่ตำแหน่งสีน้ำเงิน ให้ข้อมูลภาพที่อยู่ชั้นกลางคือภาพต้นแบบ และชั้นล่างคือภาพผลลัพธ์ ภาพที่ (ก) เมื่อจุดออริจิลของ SE ซึ่งมีค่าเท่ากับ 1 กระทำกับภาพต้นแบบที่มีค่าเท่ากับ 1 (จุดที่มีสีแดง) เช่นกัน ก็จะทำให้ภาพผลลัพธ์ปรากฏจุดเพิ่มขึ้นตามการกำหนด SE จากนั้นทำการเลื่อน SE ไปกระทำกับทุกจุดในรูปภาพต้นแบบ เมื่อจุดในรูปภาพต้นแบบมีค่าเท่ากับ 0 (จุดที่ว่างเปล่า) จะไม่มีการดำเนินการเกิดขึ้น แต่เมื่อเลื่อน SE แล้วเจอจุดที่มีค่าเท่ากับ 1 ในภาพต้นแบบก็จะทำการดำเนินการต่อ ดังภาพที่ (ข) เมื่อทำการดำเนินการจนจบสิ้นแล้ว ก็จะได้รูปภาพผลลัพธ์ดังรูปที่ 2.5



ภาพที่ 2.5 ภาพที่ผ่านการ Dilation แล้ว

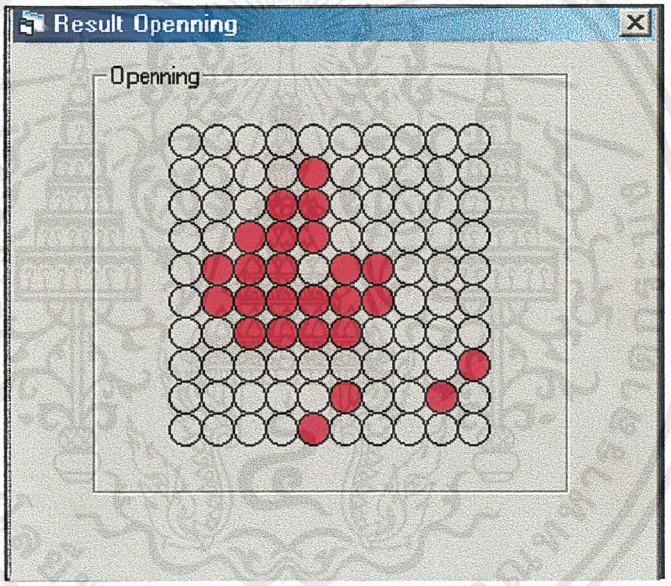
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 2.5 เป็นภาพผลลัพธ์ในการดำเนินการ Dilation จะเห็นได้ว่าจำนวนจุดสีดำได้ถูกเพิ่มขึ้นตามการกำหนดสมาชิกของ SE ทำให้เซตของผลลัพธ์เป็นดังนี้

$$F \oplus K = \{(0,2)(0,5)(0,9)(1,1)(1,4)(1,5)(1,8)(2,3)(2,4)(2,5)(2,9)(3,2)(3,3)(3,4)(3,6)(3,7)$$
$$(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)(4,7)(4,8)(5,1)(5,2)(5,3)(5,4)(5,5)(5,6)(5,7)(5,8)$$
$$(6,2)(6,3)(6,4)(6,5)(6,6)(6,7)(7,2)(7,5)(7,6)(7,9)(8,1)(8,4)(8,5)(8,8)(9,3)(9,4)\}$$

2.2.3 การโอเพนนิ่ง (Opening)

เป็นตัวดำเนินการที่เกิดจากการนำ SE มา Erosion กับรูปภาพแล้วนำผลลัพธ์จากการ Erosion มากระทำ Dilation กับ SE อีกครั้ง แทนด้วยสัญลักษณ์ได้คือ $F \circ K$ ส่วนการดำเนินการนั้นเป็นไปดังที่ได้กล่าวไว้แล้วข้างต้น และภาพผลลัพธ์แสดงได้ดังภาพข้างล่าง



ภาพที่ 2.6 ภาพที่ผ่านการ Opening แล้ว

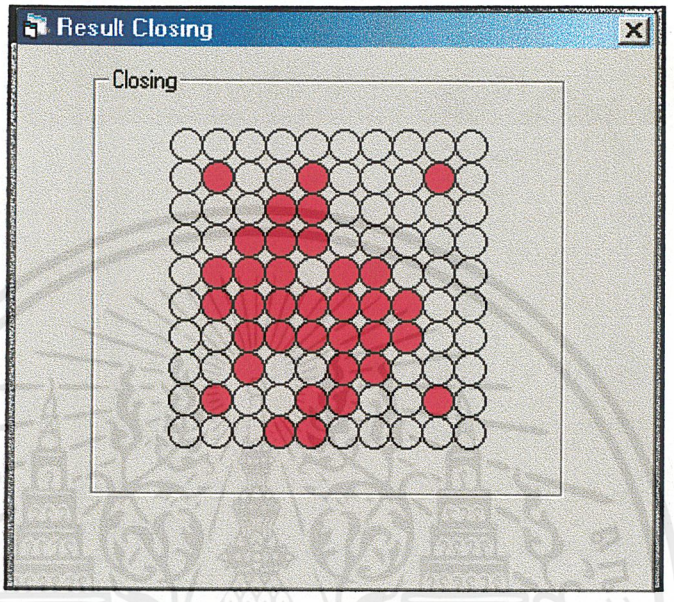
ผลจากการประมวลผลของตัวดำเนินการ Opening ทำให้ส่วนเล็กๆ ของรูปภาพหายไป ดังนั้นจุดเด่นของตัวดำเนินการตัวนี้ นำไปใช้แก้ปัญหาในการกำจัดส่วนที่ไม่ต้องการหรือสัญญาณรบกวน ที่ปนมากับรูปภาพ ซึ่งประสิทธิภาพจะมากน้อยเพียงขึ้นอยู่กับการกำหนดค่าความเป็นสมาชิกที่เหมาะสมและตำแหน่งของจุดกำเนิดของ SE ควบคู่ไปด้วย เซตของผลลัพธ์จะได้ดังนี้

$$F \circ K = \{ (1,4)(2,3)(2,4)(2,5)(3,2)(3,3)(3,4)(4,1)(4,2)(4,3)(4,5)(4,6)(5,1)(5,2)(5,3)(5,4)$$
$$(5,5)(5,6)\ (6,2)(6,3)(6,4)(6,5)(7,9)(8,5)(8,8)(9,4)\ \}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การโคลสซิ่ง (Closing)

เป็นตัวดำเนินการที่เกิดจากการนำSE มากระทำ Dilation แล้วนำผลลัพธ์ที่ได้มากระทำ Erosion กับ SE อีกครั้ง แทนด้วยสัญลักษณ์คือ $F \bullet K$ ส่วนการดำเนินการนั้นจะเป็นไปดังที่ได้กล่าวไปแล้วข้างต้น และภาพผลลัพธ์แสดงได้ดังภาพข้างล่าง



ภาพที่ 2.7 ภาพที่ผ่านการ Closing แล้ว

ผลจากการประมวลผลของตัวดำเนินการ Closing ทำให้บางส่วนของรูปภาพที่ขาดหายกลับสมบูรณ์ใกล้เคียงเหมือนเดิม ซึ่งประสิทธิภาพจะมากน้อยเพียงขึ้นอยู่กับการกำหนดค่าความเป็นสมาชิกที่เหมาะสมและตำแหน่งของจุดกำเนิดของ SE ควบคู่ไปด้วย เซตของผลลัพธ์จะได้ดังนี้

$$F \bullet K = \{(1,1)(1,4)(1,8)(2,3)(2,4)(3,2)(3,3)(3,4)(4,1)(4,2)(4,3)(4,5)(4,6)(5,1)(5,2)(5,3)$$
$$(5,4)(5,5)(5,6)(5,7)(6,2)(6,3)(6,4)(6,5)(6,6)(6,7)(7,2)(7,5)(7,6)(8,1)(8,4)(8,5)$$
$$(8,8)(9,3)(9,4)\}$$

การประมวลผลแต่ละตัวดำเนินการจะมีเอกลักษณ์ในการใช้งานที่แตกต่างกันออกไป แต่มีอย่างหนึ่งที่เหมือนกันคือการกำหนดค่าความเป็นสมาชิก SE ต้องเหมาะสมและมีความสอดคล้องกับรูปภาพเพื่อประสิทธิภาพ จะเห็นว่าในบทนี้จะเป็นการประมวลผลกับรูปภาพที่เป็นสีขาวกับสีดำเท่านั้นส่วนนี้ จะเป็นส่วนของหลักการพื้นฐานที่จะนำไปปรับปรุงพัฒนาไปใช้ประมวลผลกับรูปภาพเกรย์สเกล

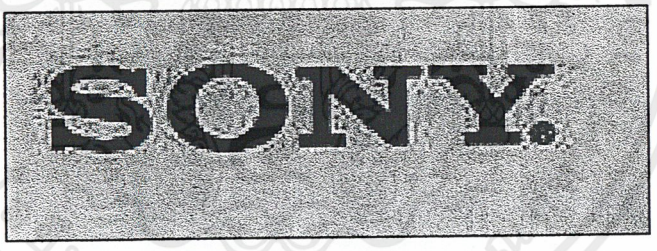
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ทฤษฎีมอร์โฟโลยีที่ใช้ประมวลผลกับภาพเกรย์

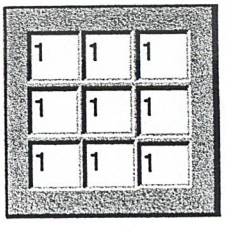
3.1 ตัวดำเนินการมอร์โฟโลยี

เนื่องจากทฤษฎีมอร์โฟโลยีประมวลผลได้ดีกับภาพที่มีสีขาวดำ และต่อมาได้มีการพัฒนาเพื่อนำมาประมวลผลกับภาพเกรย์สเกล ที่เป็นภาพที่มีระดับสีอยู่ในช่วง (0-255) หลักการและทฤษฎีของแต่ละตัวดำเนินการจะเหมือนกันกับตัวดำเนินการมอร์โฟโลยีที่ดำเนินการกับภาพขาวดำที่ได้กล่าวมาแล้วในตอนต้น

เมื่อนำมอร์โฟโลยีมาใช้กับภาพเกรย์สเกล ลักษณะการดำเนินการก็จะต่างกันออกไป คือ จากเซต F ที่เคยใช้แทนภาพไบนารีก็เปลี่ยนมาเป็นฟังก์ชัน $f(x,y)$ ที่แทนภาพเกรย์สเกลและ SE ของเซต K ก็แทนด้วยฟังก์ชัน $k(x,y)$ และที่สำคัญคือตัวดำเนินการยูเนียนใน Dilation ก็ถูกแทนด้วยฟังก์ชันหาค่าสูงสุด Max() ส่วนตัวดำเนินการอินเตอร์เซ็กชันใน Erosion ก็แทนด้วยฟังก์ชันหาค่าต่ำสุด Min() ดังนั้นตัวดำเนินการทั้งสองตัวที่ใช้กับภาพไบนารี สามารถกำหนดให้เหมาะสมกับการประมวลผลภาพเกรย์สเกลได้ ต่อไปจะแสดงการนำตัวดำเนินการมอร์โฟโลยีที่ประมวลผลกับภาพเกรย์ พร้อมกับการกำหนด SE และรูปภาพผลลัพธ์ของการดำเนินการ



กำหนด เซต $f(x,y)$ เป็นเซตของรูปภาพ



เซต $k(x,y)$ เป็นเซตของ SE

ภาพที่ 3.1 แสดงภาพต้นแบบและ SE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

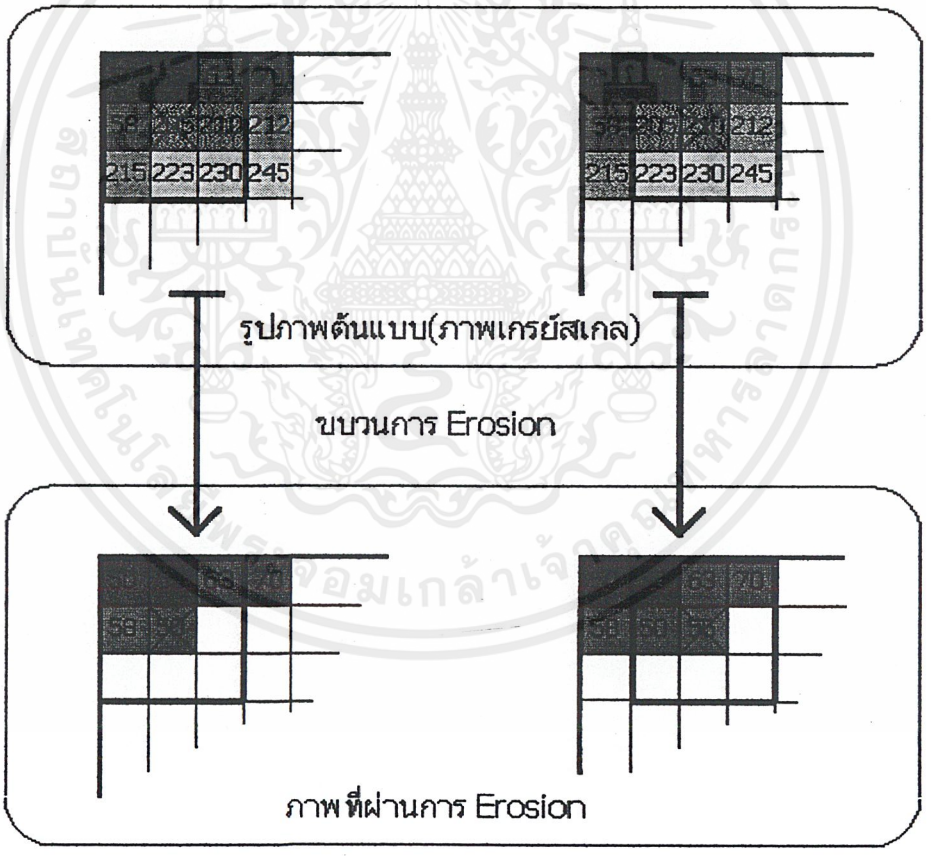
ภาพต้นแบบที่ใช้เป็นภาพเกรย์ที่มีจำนวน Pixel เท่ากับ 180 X 64 Pixel พื้นหลังของภาพมีค่าของระดับสีเท่ากับ 220 และตัวอักษรมีค่าของระดับสีเท่ากับ 18 ส่วน SE ที่ใช้มีขนาด 3 X 3 และมีค่าความเป็นสมาชิกเท่ากับ 1 ทั้งหมด (Simple SE)

ตัวดำเนินการมอร์โฟโลยี

3.1.1 ตัวดำเนินการ Erosion

ตัวดำเนินการ Erosion ระหว่างฟังก์ชัน $f(x,y)$ กับฟังก์ชัน $k(x,y)$ โดยนำ $k(x,y)$ มากระทำกับรูปภาพ ของ เซต $f(x,y)$ โดยเอาจุด Origin ของ $k(x,y)$ มากระทำกับเซต $f(x,y)$ ทีละจุดของภาพ โดยจะนำค่า min รอบๆ จุด Origin ของ $k(x,y)$ มาแสดงผลเมื่อมากระทำทุกจุดของจุดของภาพแล้วก็จะได้ภาพที่ผ่านการ Erosion แล้ว

หลักการและการทดลอง

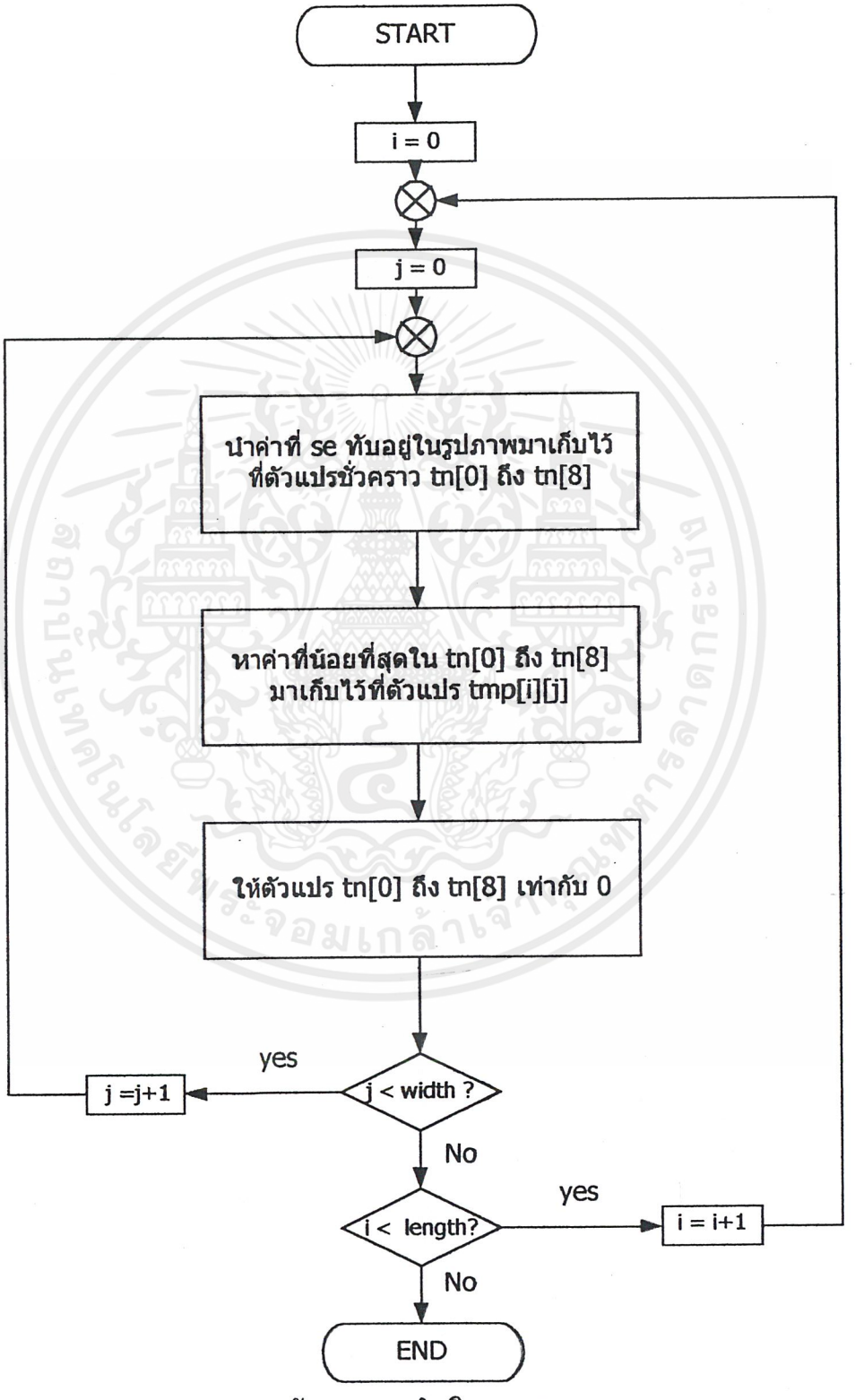


จากภาพเป็นการแสดงขั้นตอนการประมวลผลของตัวดำเนินการ Erosion กล่าวคือเมื่อนำ SE ที่มีขนาด 3 X 3 ที่มีสมาชิกเป็น 1 ทั้งหมด และมีจุดออริจิลที่จุดกึ่งกลางของ SE (แสดงได้ดังจุดกากบาท) แล้วทำการหาค่าที่ต่ำที่สุดที่อยู่ในรูปภาพที่ SE ทับอยู่ นำค่าที่ต่ำที่สุดนั้นมาเป็นค่าผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

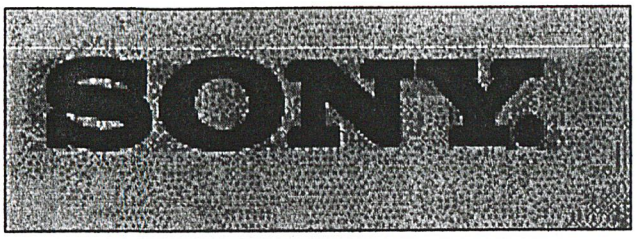
ลัพธ์ของรูปภาพผลลัพธ์ก็จะได้ค่าดังรูป จากนั้นทำการเลื่อน SE ไปกระทำกับทุกจุดในรูปภาพ ก็จะได้รูปภาพผลลัพธ์ของการ Erosion

ต่อไปเป็นการแสดงแผนผังของการตัวดำเนินการ Erosion โดย width คือความกว้างของภาพ ส่วน length คือความสูงของภาพ



แผนผังแสดงการดำเนินการ Erosion

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.2 แสดงภาพที่ผ่านการ Erosion แล้ว

โดยมีรูปแบบสมการดังนี้

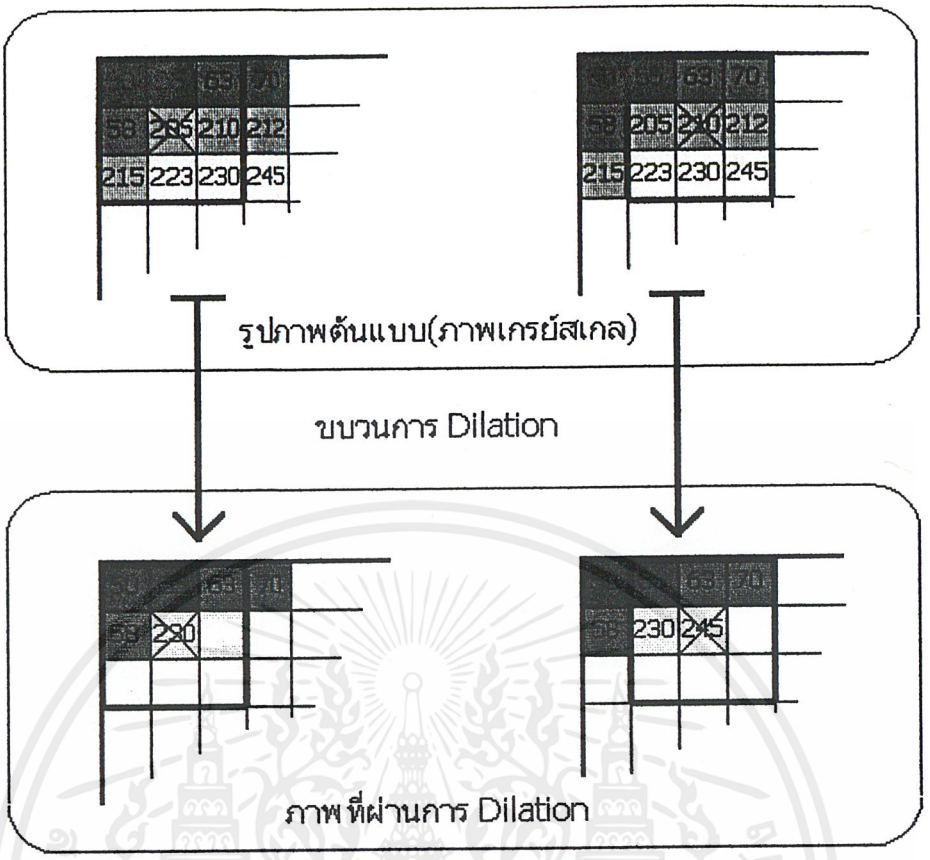
$$(f\Theta k)(x,y) = Min\left(f(x+i,y+j) - k(i,j) \,\middle|\, (i,j) \in K\right) \qquad (5)$$

ส่วนของโปรแกรมที่ออกแบบเพื่อใช้ในการทำงานตามสมการของตัวดำเนินการ Erosion ซึ่งจะอยู่ในส่วนท้ายของปริญญานิพนธ์นี้ โดยมีหลักการคือจะทำการดึงค่าของระดับความเป็นสมาชิกในแต่ละจุดในรูปภาพมาใช้ โดยเก็บไว้ในตัวแปรและนำค่าระดับความเป็นสมาชิกแต่ละจุดมาคูณกับค่า SE ที่กำหนดไว้แต่ละจุดแสดงในแผนผัง แล้วนำค่าที่ได้จากการคำนวณ มาเปรียบเทียบกันเพื่อหาค่าต่ำสุด เก็บไว้ในตัวแปร แล้วส่งค่าที่ได้ไปยังแต่ละจุดของภาพ

3.1.2 ตัวดำเนินการ Dilation

การ Dilation ระหว่างเซต $f(x,y)$ และเซต $k(x,y)$ โดยนำ $k(x,y)$ มากระทำกับรูปภาพของเซต $f(x,y)$ โดยเอาจุด Origin ของ $k(x,y)$ มากระทำกับเซต $f(x,y)$ ที่ละจุดของภาพ โดยจะนำค่า min รอบๆ จุด Origin ของ $k(x,y)$ มาแสดงผลเมื่อมากระทำทุกจุดของจุดของภาพแล้วก็จะได้ภาพที่ผ่านการ Dilation แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

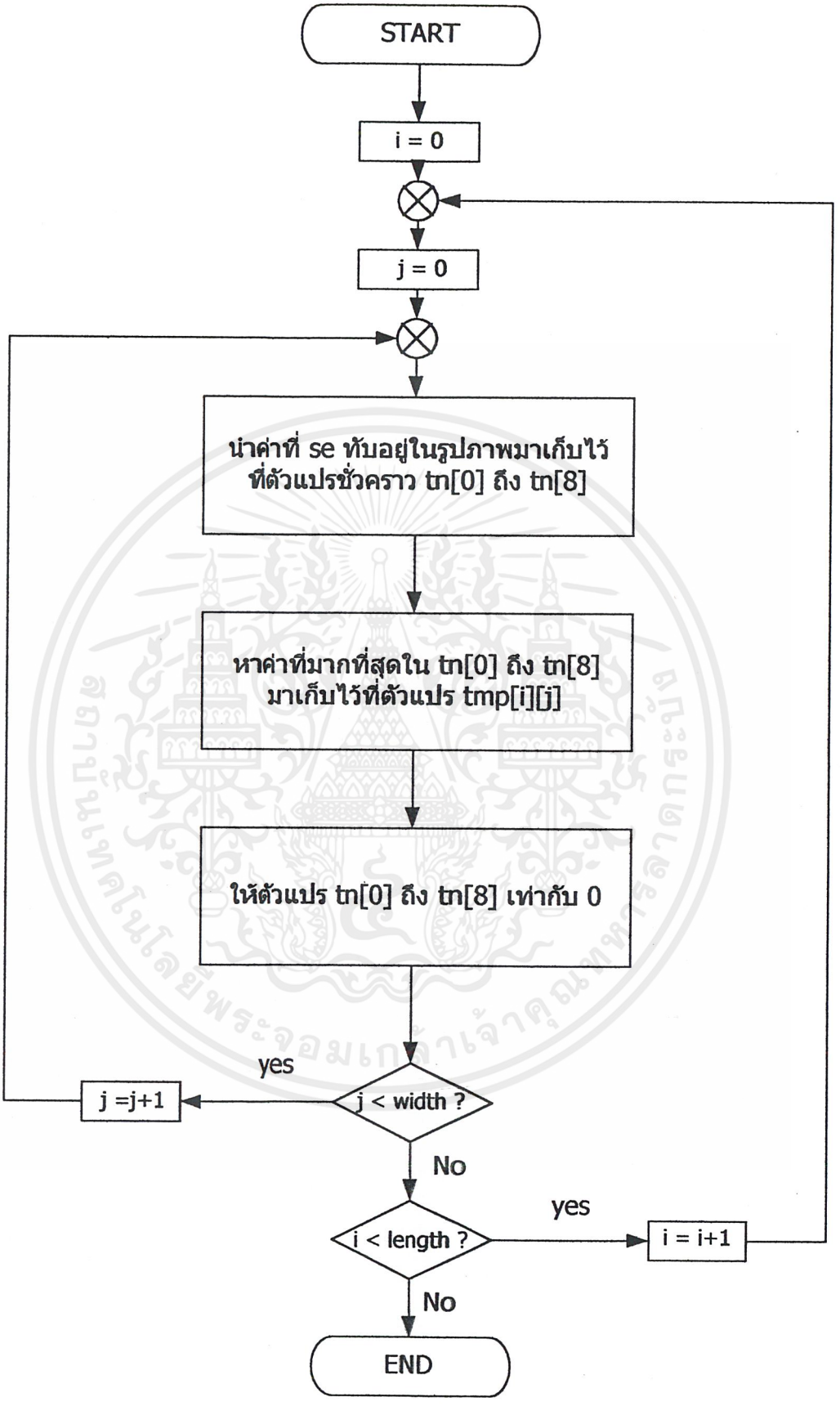


จากภาพเป็นการแสดงขั้นตอนการประมวลผลของตัวดำเนินการ Dilation กล่าวคือเมื่อนำ SE ที่มีขนาด 3 X 3 ที่มีสมาชิกเป็น 1 ทั้งหมด และมีจุดออริจิลที่จุดกึ่งกลางของ SE (แสดงได้ดังจุดกากบาท) แล้วทำการหาค่าที่สูงที่สุดที่อยู่ในรูปภาพที่ SE ทับอยู่ นำค่าที่สูงที่สุดนั้นมาเป็นค่าผลลัพธ์ของรูปภาพผลลัพธ์ก็จะได้ค่าดังรูป จากนั้นทำการเลื่อน SE ไปกระทำกับทุกจุดในรูปภาพ

ต่อไปเป็นการแสดงแผนผังของการตัวดำเนินการ Dilation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

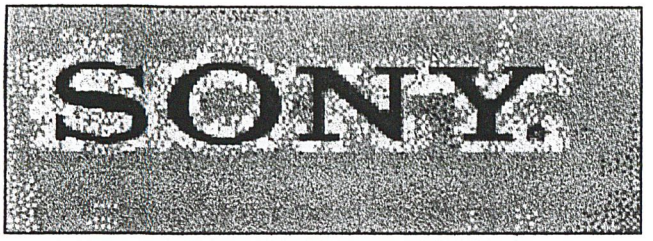
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



แผนผังแสดงการดำเนินการ Dilation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

45856



ภาพที่ 3.3 แสดงรูปที่ผ่านการ Dilation แล้ว

มีรูปแบบสมการดังนี้

$$(f \oplus k)(x,y) = Max\{f(x-i,y-j)+k(x,y) | (i,j) \in K, (x-i,y-j) \in F\} \qquad (6)$$

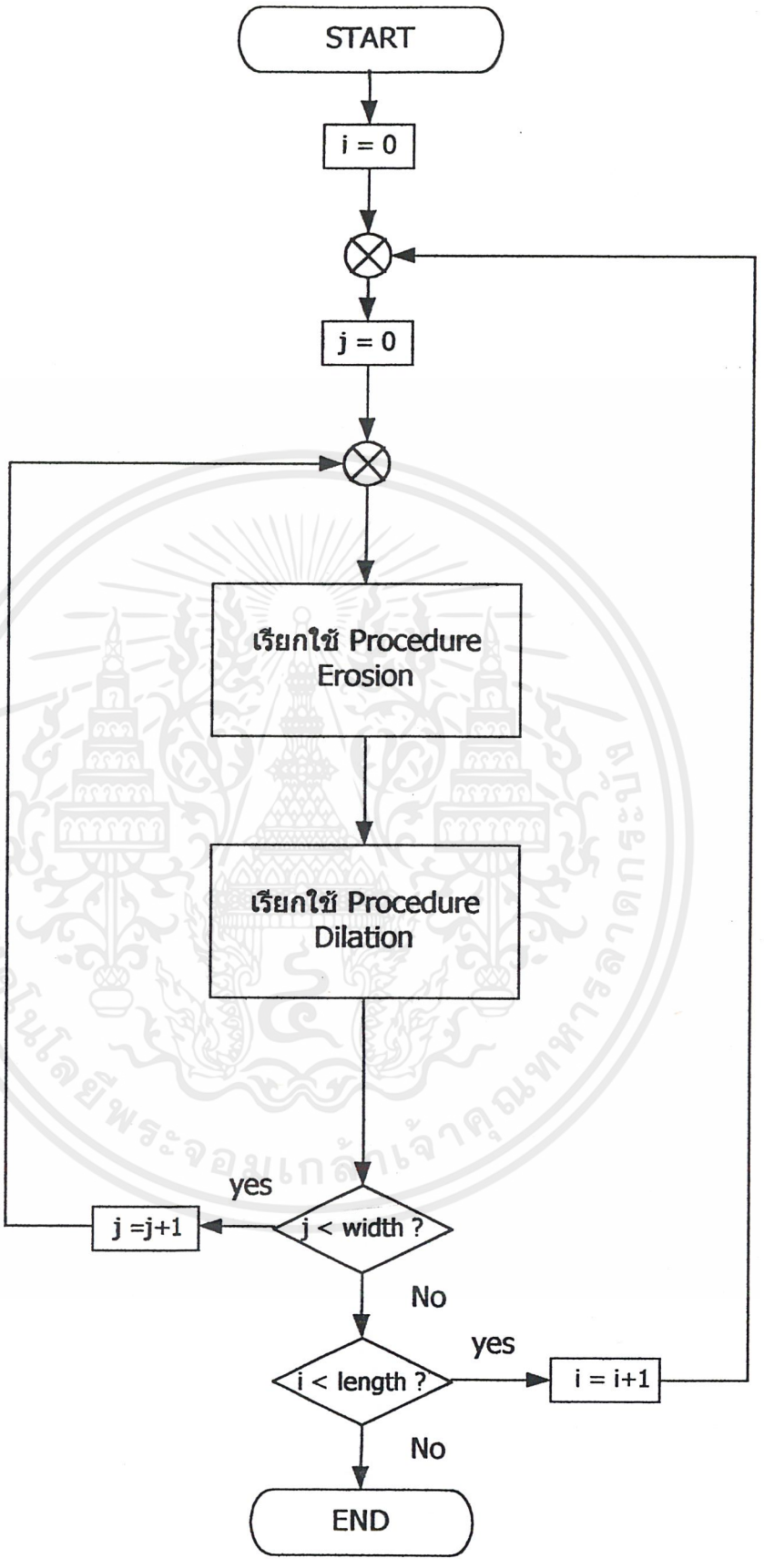
และส่วนของโปรแกรมที่ออกแบบเพื่อใช้ในการทำงานตามสมการของตัวดำเนินการ-Dilation ซึ่งจะอยู่ในส่วนท้ายของปริญญานิพนธ์นี้ โดยมีหลักการคล้ายกับตัวดำเนิน Erosion คือ จะทำการดึงค่าของระดับความเป็นสมาชิกในแต่ละจุดมาใช้ โดยเก็บไว้ในตัวแปรและนำค่าระดับความเป็นสมาชิกแต่ละจุดมาคูณกับค่าของ SE ที่กำหนดไว้แต่ละจุดดังแสดงในแผนผัง นำค่าที่ได้ที่ได้จากการคำนวณ มาเปรียบเทียบกันเพื่อหาค่าค่าสูงสุด เก็บไว้ในตัวแปร แล้วส่งค่าที่ได้ไปยังแต่ละจุดของภาพผลลัพธ์

3.1.3 ตัวดำเนินการ Opening

เป็นตัวดำเนินการที่ได้จากการนำตัวดำเนินการ Erosion และ Dilation มากำหนดเป็นเงื่อนไขเป็นตัวดำเนินการที่ใช้ในการกำจัดสัญญาณรบกวนที่เกิดในภาพได้ ที่มีระดับค่าความเป็นสมาชิกสูงๆ หรือสัญญาณรบกวนที่มีค่าสูงๆ มีรูปแบบของสมการดังนี้

$$(f \circ k)(x,y) = (f \ominus k)(x,y) \oplus k(x,y) \qquad (7)$$

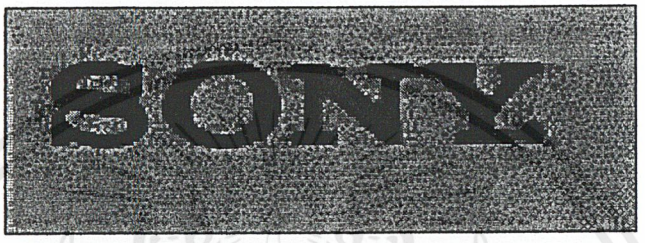
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แผนผังแสดงการดำเนินการ Opening

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรมที่ออกแบบเพื่อใช้ในการทำงานตามสมการของตัวดำเนินการ Opening ซึ่งจะอยู่ในส่วนท้ายของปริญญานิพนธ์นี้ โดยมีหลักการคือนำภาพที่ต้องการจะต้องการทำการ Opening มาทำการ Erosion ก่อนและผลลัพธ์ออกมาจะนำไป Dilation อีกทีหนึ่งจะได้ผลลัพธ์ของการ Openingออกมา การทำงานจะเรียกโปรแกรมส่วนของ Erosion ก่อนแล้วจึงเรียกโปรแกรมส่วนของ Dilation จะทำการดึงค่าของระดับความเป็นสมาชิกในแต่ละจุดมาใช้โดยเก็บไว้ในตัวแปรและนำค่าระดับความเป็นสมาชิกแต่ละจุดมาคูณกับค่าของ SE ที่กำหนดไว้แต่ละจุดและนำค่าที่ได้ที่ได้จากการคำนวณ มาเปรียบเทียบกันเพื่อหาค่าสูงสุด นำค่าที่ได้ส่งไปยังจุดของภาพผลลัพธ์



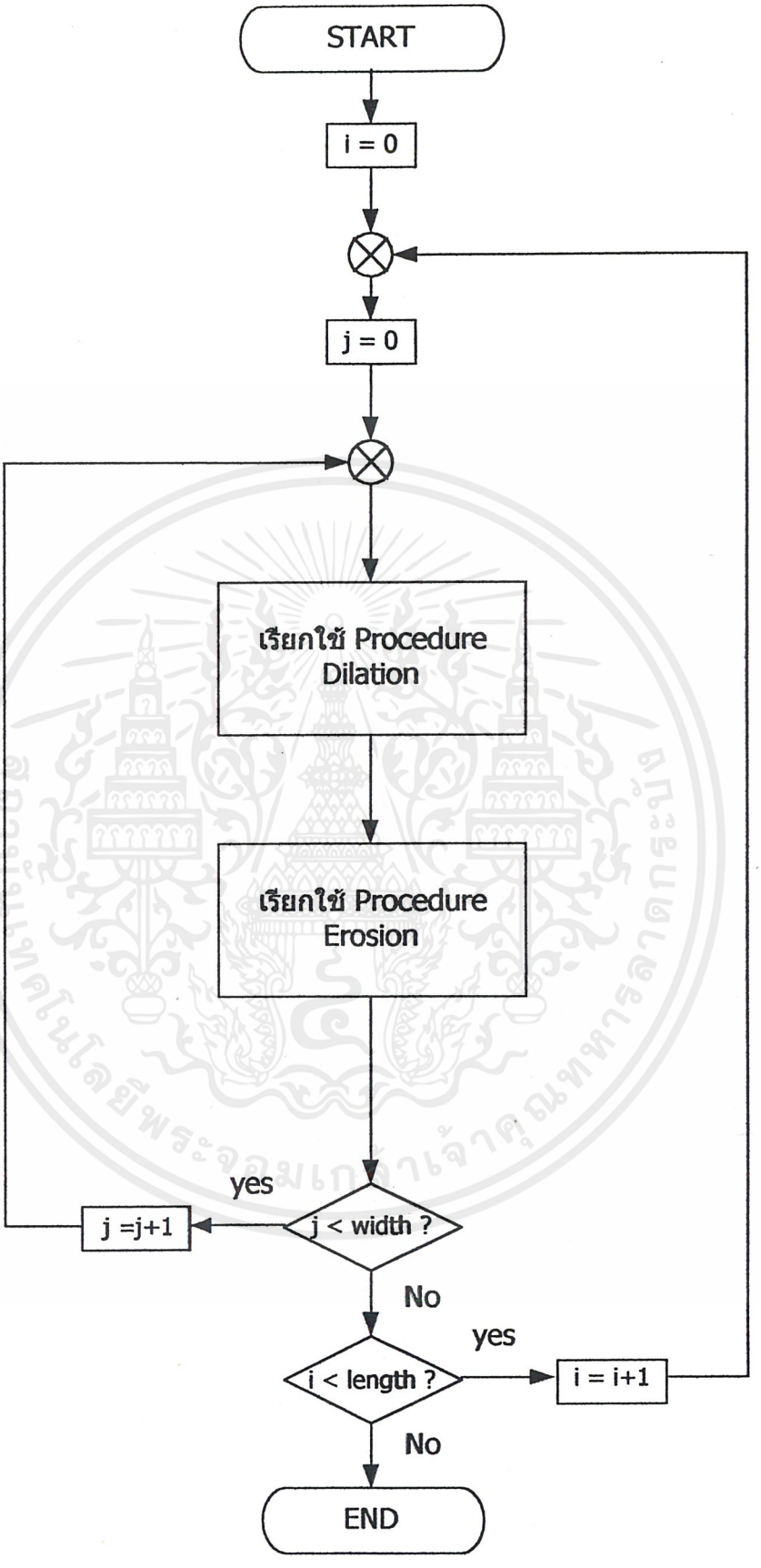
ภาพที่ 3.4 แสดงรูปที่ผ่านการ Opening แล้ว

3.1.4 ตัวดำเนินการ Closing

ตัวดำเนินการซึ่งได้จากการนำตัวดำเนินการ Dilation และ Erosion มากำหนดเป็นเงื่อนไข ตัวดำเนินการนี้ใช้ในการกำจัดสัญญาณรบกวนที่เกิดในภาพได้ ที่มีระดับค่าความเป็นสมาชิกต่ำๆ หรือสัญญาณรบกวนที่มีค่าต่ำๆ มีรูปแบบของสมการดังนี้

$$(f \bullet k)(x,y) = (f \oplus k)(x,y) \Theta k(x,y) \qquad (8)$$

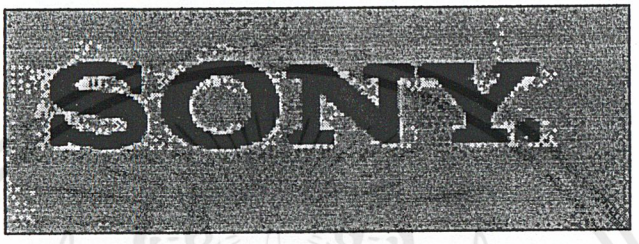
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แผนผังแสดงการดำเนินการ Closing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรมที่ออกแบบเพื่อใช้ในการทำงานตามสมการของตัวดำเนินการ Closing ซึ่งจะอยู่ในส่วนท้ายของปริญญานิพนธ์นี้ โดยมีหลักการคือนำภาพที่ต้องการจะทำการ Closing มาทำการ Dilation ก่อนและผลลัพธ์ออกมาจะนำไป Erosion อีกทีหนึ่งจะได้ผลลัพธ์ของ Closing ออกมา การทำงานจะเรียกโปรแกรมส่วนของ Dilation ก่อนแล้วจึงเรียกโปรแกรมส่วนของ Erosion จะทำการดึงค่าของระดับความเป็นสมาชิกในแต่ละจุดมาใช้ โดยเก็บไว้ในตัวแปรและนำค่าระดับความเป็นสมาชิกแต่ละจุดมาคูณกับค่าของ SE ที่กำหนดไว้แต่ละจุด และนำค่าที่ได้ที่ได้จากการคำนวณ มาเปรียบเทียบกันเพื่อหาค่าค่าต่ำสุด ส่งค่าที่ได้ไปยังแต่ละจุดของภาพผลลัพธ์



ภาพที่ 3.5 แสดงรูปที่ผ่านการ Closing แล้ว

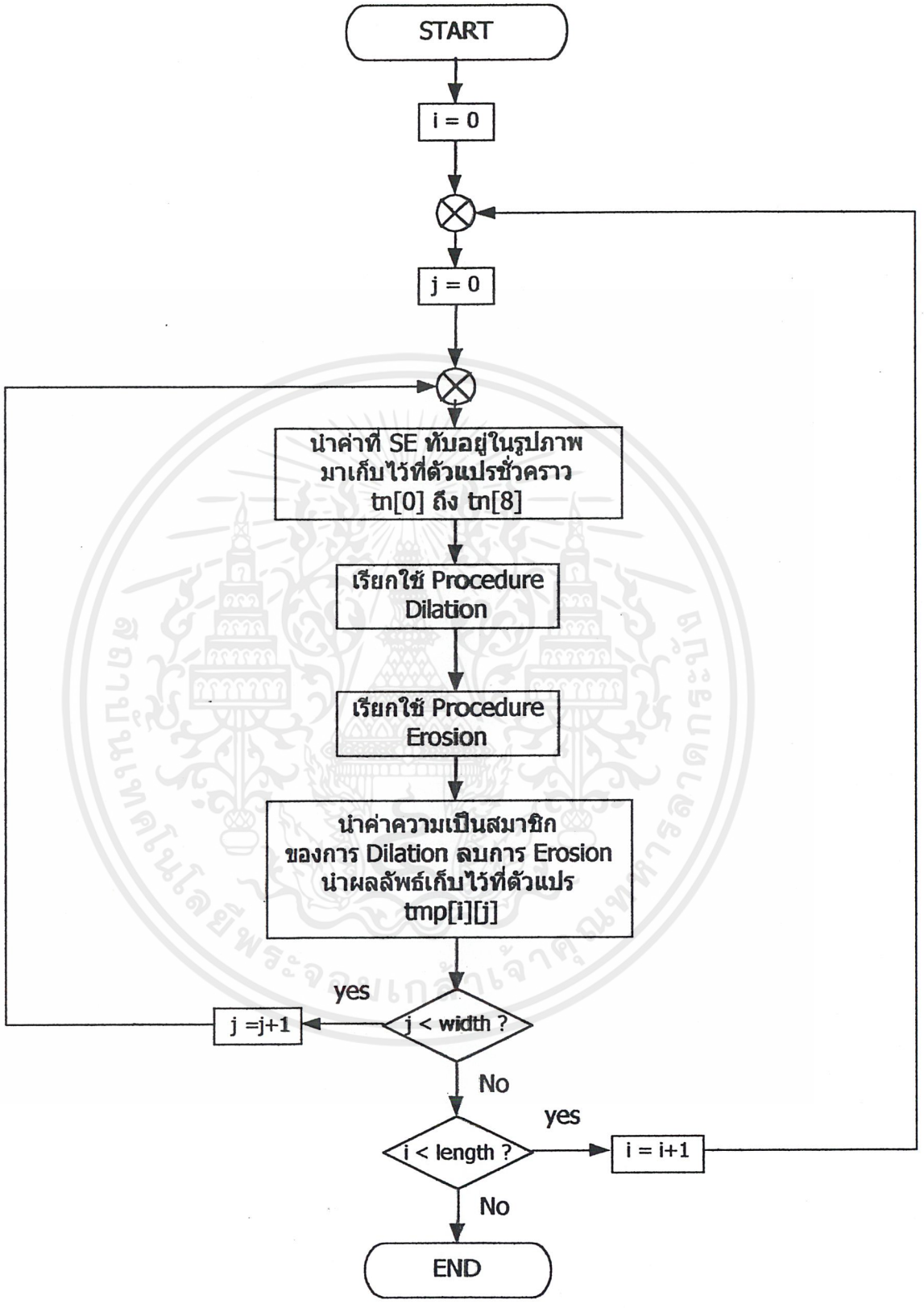
3.1.5 การดำเนินการ Gradient

Gradient เป็นการดำเนินการหาความเปลี่ยนแปลงซึ่งได้จากตัวดำเนินการ Dilation และ Erosion และสามารถนำมาหาขอบภาพของรูปภาพ โดยการดำเนินการคือการนำเอารูปภาพที่ต้องการทำการ Gradient นั้นมาทำการ Dilation ก่อนแล้วนำภาพต้นแบบที่ต้องการทำการ Gradient นั้นมาทำการ Erosion จากนั้นนำเอาภาพผลผลลัพธ์ที่ได้จากการ Dilation และการ Erosion มาลบกันเพื่อให้เห็นการเปลี่ยนแปลงของรูปภาพ ดังนั้นภาพผลลัพธ์จากการ Gradient จะปรากฏเป็นขอบของรูปภาพนั้นเอง การลบกันของภาพ Dilation กับภาพ Erosion ทำได้โดยการนำค่าของจุดแต่ละจุดในรูปมาลบกันแบบจุดต่อจุด โดยนำค่าของภาพการ Dilation เป็นตัวตั้งแล้วนำค่าของภาพการ Erosion มาเป็นตัวลบ จะแสดงได้ดังสมการดังนี้

$$G = (A \oplus S) - (A \ominus S) \qquad (10)$$

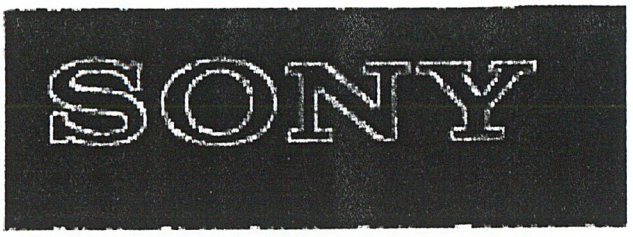
ต่อไปจะเป็นการแสดงแผนผังการทำงานของการดำเนินการ Gradient

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แผนผังแสดงการดำเนินการ Gradient

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.6 แสดงรูปที่ผ่านการ Gradient แล้ว

จากสมการที่ (10) ค่า G คือ ผลลัพธ์ที่ได้จากการ Gradient ได้มาจากการนำเอารูปภาพ A มา Dilation และ Erosion กัน เก็บค่าที่ได้ไว้ในตัวแปร นำค่าจากการ Dilation ลบการ Erosion จะได้ค่า G ออกมา

3.1.6 ตัวดำเนินการ Thresholding

การ Threshold เป็นการทำภาพ Grey ให้เป็นภาพ Binary โดยปกติแล้วภาพ Grey จะมีระดับของความเป็นสมาชิกอยู่ 256 ระดับตั้งแต่ 0-255 โดยที่ค่าระดับความเป็นสมาชิกเท่ากับ 0 จะเป็นสีดำ และจะลดความเข้มของสีลงเรื่อยๆจนเป็นสีขาวซึ่งมีค่าระดับของความเป็นสมาชิกคือ 1 ในรูปภาพแบบ Grey หนึ่งภาพจะประกอบด้วย Pixel หลาย Pixel แต่ละ Pixel จะมีค่าระดับความเป็นสมาชิกที่กำหนดเป็นตัวเลข 0-255 อยู่เราสามารถกำหนดค่าที่จะใช้เปรียบเทียบเช่น 120 ค่าความเป็นสมาชิกที่มากกว่า 120 ในรูปภาพแบบ Gray ให้มีค่าระดับความเป็นสมาชิกเป็น 0 ในรูปภาพแบบไบนารี่และค่าระดับความเป็นสมาชิกที่มีค่าน้อยกว่า 120 ให้มีค่าระดับความเป็นสมาชิกเป็น 1 ในรูปภาพแบบ ไบนารี ซึ่งภาพผลลัพธ์นั้นจะกลายเป็นภาพไบนารี โดยสมการของการ Threshod คือ

$$S = max\left(\left|g \ **e_s\right|, \left|g_2 \ **e_s\right|\right) \qquad (11)$$

$$\gamma = \frac{\sum(e_S \cdot S)}{\sum S} \qquad (12)$$

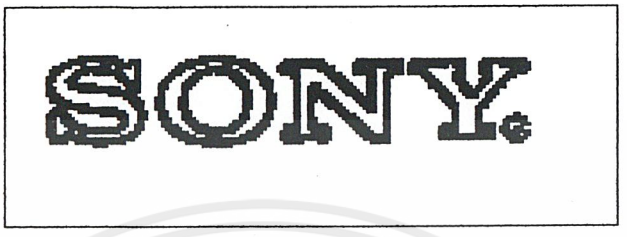
$$e = \begin{cases} 1, if\left(e_s > \gamma\right) \\ 0, otherwise \end{cases} \qquad (13)$$

จากสมการที่ (11) ค่า S คือค่าสูงสุดของการ Convolution ระหว่าง g_1 และ g_2 กับภาพต้นแบบ โดยที่ g_1 มีค่าเท่ากับ $[-1 \quad 0 \quad 1]$ และ g_2 มีค่าเท่ากับ $[-1 \quad 0 \quad 1]^t$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (12) ค่า γ คือค่า Threshold Level เกิดจากผลรวมของรูปต้นแบบ (e_s) คูณกับค่า S หารด้วยผลรวมของค่า S

จากสมการที่ (13) e คือผลลัพธ์ที่ได้จาก Threshold โดยจะเป็น 1 เมื่อ e_s มีค่ามากกว่า γ และเป็น 0 เมื่อมีค่าน้อยกว่า



ภาพที่ 3.7 แสดงรูปที่ผ่านการ Thresholding แล้ว

ทั้งหมดที่ได้กล่าวมาเป็นการนำตัวดำเนินการทั้งหมดมาใช้ในการประมวลผล แต่ละตัวดำเนินการมีเอกลักษณ์ในการใช้งานที่แตกต่างกันไป ประสิทธิภาพที่ได้จะมากน้อยเพียงใดขึ้นอยู่กับการกำหนดค่าความเป็นสมาชิกของ SE ที่แตกต่างไปตามลักษณะโครงสร้างของรูปภาพที่นำมาประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การตรวจจับข้อความออกจากรูปภาพ

ในบทนี้จะกล่าวถึงการนำเอาทฤษฎีมอร์โฟโลยีที่ได้กล่าวไปในบทต้นๆ มาประยุกต์ใช้ในการตรวจจับข้อความออกจากรูปภาพ การตรวจจับข้อความออกมาจากรูปมีประโยชน์หลากหลายในขบวนการทางเอกสาร (Document analysis) การวิเคราะห์ทางเทคนิคของเอกสาร, ตาราง, แผนที่, แผนภูมิ เช่น ข่าวซึ่งจะประกอบด้วยข้อความรูปภาพกราฟฟิก ดังนั้น ตัวอักษรมักจะเป็นส่วนประกอบของรูปภาพ

การตรวจจับข้อความออกมาจากรูปจะดำเนินการบนภาพไบนารี เทคนิคที่ได้แนะนำนี้จะเป็นเทคนิคการวิเคราะห์ โดยจำแนกขอบเขตภาพออกเป็นข้อความ, ตาราง, รูปภาพและรูปเขียนแบบ ข้อจำกัดของเทคนิคนี้คือไม่สามารถกระทำกับรูปภาพที่มีขนาดเล็กที่มีตัวอักษรตัวใหญ่เป็นส่วนประกอบอยู่ในภาพ และรูปภาพที่มีข้อความอยู่ในแนวโค้ง จะใช้ความรู้พื้นฐานนี้ในการกำหนดขอบเขตของข้อความ

คณิตศาสตร์ของ Morphology มีพื้นฐานมาจากเรขาคณิตเชิงการวิเคราะห์ ประกอบไปด้วยฟังก์ชันที่ใช้สำหรับการวิเคราะห์ทางเรขาคณิต เทคนิคการตรวจจับข้อความออกมาจากรูปภาพโดยใช้ Morphology จะมีประสิทธิภาพมากน้อยเพียงใด เช่น มีขนาดของความกว้าง, ความยาว และความหนาตรงตามที่กำหนด โดยในปริญญาฉบับนี้ได้แนะนำวิธีพื้นฐานในการตรวจจับข้อความออกมาจากรูปภาพโดยใช้ตัวดำเนินการคณิตศาสตร์ของ Morphology

ในหัวข้อต่อไปจะเป็นการอธิบายเทคนิคการใช้ตัวปฏิบัติการของ Morphology ในการดึงข้อความออกมาจากรูปภาพ พร้อมตัวอย่างและการเปรียบเทียบระหว่างรูปภาพก่อนและหลังจากการดำเนินการเรียบร้อยแล้ว

4.1 Morphological Text Extraction

วิธีที่ได้เสนอแนะนี้ จะกล่าวถึงในเรื่องความคมชัด สีและขนาดของข้อความที่อยู่ในรูปภาพดังจะได้กล่าวต่อไปนี้

1.รูปภาพที่มีข้อความตัวหนังสือเป็นส่วนประกอบนั้นโดยข้อความนั้นจะมีความแตกต่างกับสีพื้นของภาพ (background)

2. แต่ละตัวอักษรมีขนาดความยาว (L จุดภาพ) มีความกว้าง (W จุดภาพ) และมีความหนา (T pixels) ผลลัพธ์ของความสัมพันธ์ทั้งหมดคือ R (pixels/mm) โดยที่ R คือความละเอียดต่อพื้นที่หนึ่งหน่วยรูปภาพ (Resolusion) ที่ต้องการดึงข้อความออกมานั้นต้องมีขอบเขตขนาดความกว้าง, ความยาว และความหนา ดังต่อไปนี้

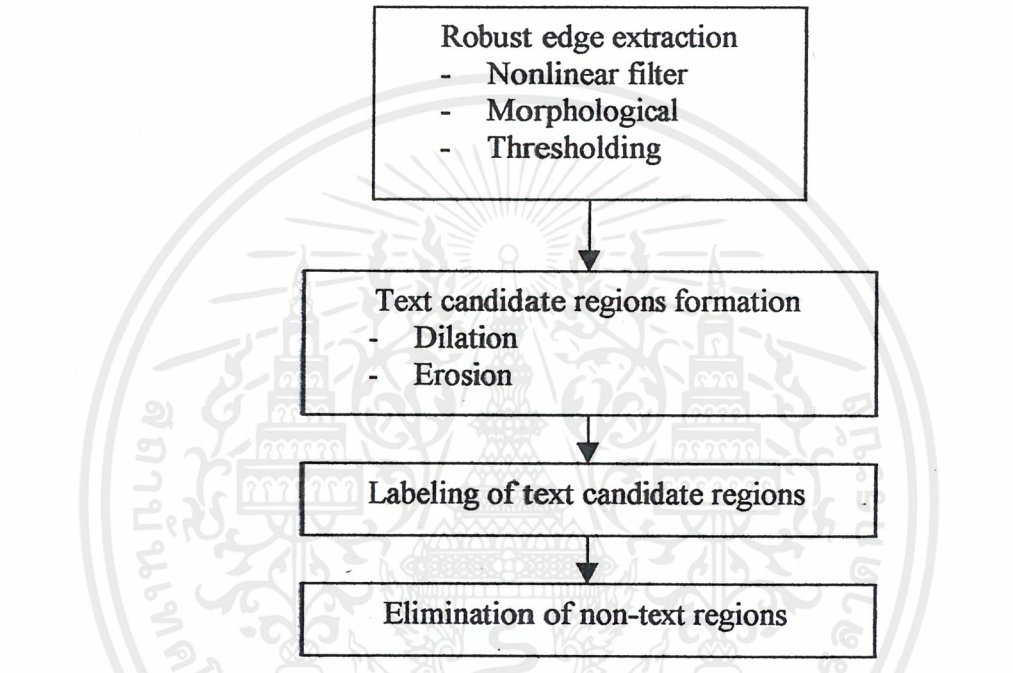
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$1.8R \le L \le 8.4R$ $1.5R \le W \le 7R$ $0.2R \le T \le 2R$

โดยที่ $R \ge 9$ pixels/mm แต่ถ้า R มีค่าน้อยกว่านี้ขบวนการจะไม่สามารถกระทำได้ ถ้า R มีขนาดใหญ่กว่าที่กำหนดขบวนการก็ไม่สามารถกระทำได้เช่นกัน

3. ข้อความประกอบด้วยคำมากกว่า 1 คำ และแต่ละคำจะประกอบด้วยอักษรมากกว่า 3 ตัว ดังนั้นถ้าข้อความประกอบด้วยตัวอักษรน้อยกว่า 3 ก็จะไม่สามารถประมวลผลได้

Flow Diagram ของขั้นตอนต่างๆได้ แสดงในรูปที่ 1 และขั้นตอนจะสรุปได้ดังนี้



ขั้นตอนในการตรวจจับข้อความออกจากรูปภาพแสดงได้ดังนี้

ภาพที่ 4.1 รูปต้นแบบที่ใช้ในการวิเคราะห์

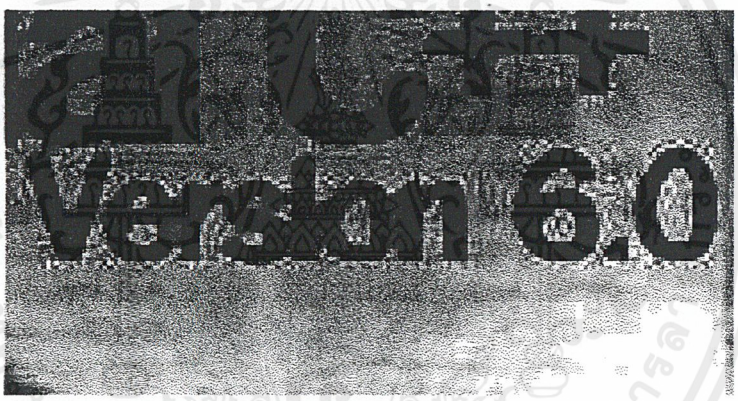
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 ขั้นตอนที่ 1

ขั้นตอนนี้เป็นการปรับปรุงรูปภาพเกรย์ที่มีข้อความเป็นส่วนประกอบ มีขั้นตอนย่อยอีก 3 ขั้นตอนคือ ขั้นตอนแรกเป็นการกำจัดสัญญาณรบกวนที่อยู่ในรูปภาพเกรย์ที่มีข้อความอยู่ ซึ่งอาจจะมีสัญญาณรบกวนชนิด salt and pepper ต้องทำการกำจัดสัญญาณรบกวนออกเสียก่อน โดยใช้ตัวดำเนินการของมอร์โฟโลยีคือ การ open–close และ close–open เป็นตัวกำจัดสัญญาณรบกวนผลลัพธ์ในการกำจัดสัญญาณรบกวนนี้ (Y_{b1}) คือ ค่าเฉลี่ยผลลัพธ์ของการกำจัดสัญญาณ

$$Y_{b1} = \frac{(Y_B)^B + (Y^B)_B}{2} \qquad (15)$$

เมื่อ B คือ structure element แบบง่ายๆ ขนาด 3*3 และ Y_B กับ Y^B กำหนดเป็น opening และ closing ของ Y โดยการกระทำของ B ตามลำดับ



ภาพที่ 4.2 แสดงรูปผลลัพธ์จากการกำจัดสัญญาณรบกวน (Y_{b1})

ขั้นตอนต่อไปเป็นหาขอบภาพโดยนำเอารูปภาพที่กำจัดสัญญาณรบกวนแล้วนั้นมาทำการหาขอบภาพด้วยวิธีการ gradient ซึ่งเป็นวิธีการตรวจจับความเปลี่ยนแปลงของการดำเนินการ มอร์โฟโลยี รูปภาพผลลัพธ์จะปรากฏเป็นรูปขอบของข้อความ

ตัวดำเนินการ Morphological Gradient (MG) จะถูกนำมากระทำใน Y_{b1} และผลลัพธ์ของรูปภาพ คือ

$$e_S = MG(Y_{b1,}B) = \delta_B(Y_{b1}) - \varepsilon_B(Y_{b1}) \qquad (16)$$

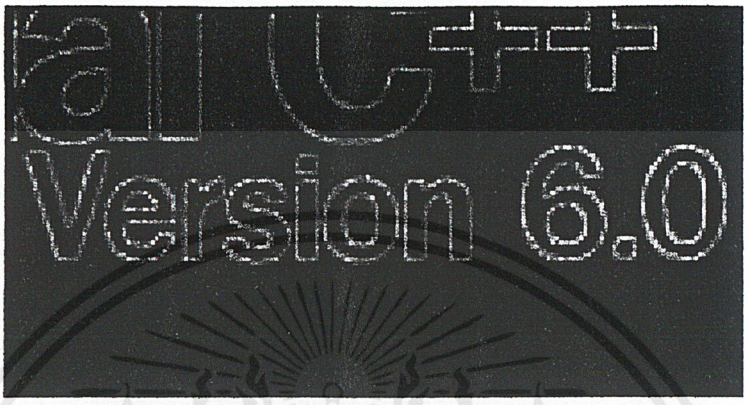
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MG คือ Morphological Gradient

$\delta_B(Y_{b1})$ เป็นการ Dilation รูปภาพด้วย Structure Element B

$\varepsilon_B(Y_{b1})$ เป็นการ Erosion รูปภาพด้วย Structure Element B

e_s คือผลลัพธ์ความต่างระหว่าง Dilation กับ Erosion



ภาพที่ 4.3 แสดงรูปที่ได้จากการหาขอบภาพ (e_S)

ขั้นตอนสุดท้ายเป็นการแปลงภาพจากรูปภาพเกรย์เป็นรูปภาพขาวดำด้วยวิธีการThreshold เพื่อทำการหาตำแหน่งของข้อความที่อยู่ในรูปภาพในขั้นตอนที่ 4 ต่อไป รูปภาพผลลัพธ์จะเป็นรูปขาวดำที่ปรากฏเป็นรูปขอบของข้อความและรูปขอบของวัตถุอื่นที่เป็นส่วนประกอบของรูปภาพซึ่งเป็นวัตถุที่ไม่ใช้ส่วนประกอบของข้อความ เราจะทำการกำจัดวัตถุที่เราไม่ต้องการออกไปในขั้นตอนต่อไป

สมการในการ Thresholding แสดงดังต่อไปนี้

$$\gamma = \frac{\sum(e_S \cdot S)}{\sum S}$$

เมื่อ " ● " คือ การคูณกันแบบ pixel – wise กับ S ซึ่ง S หาได้โดย

$$S = \max(|g_1 ** e_S|, |g_2 ** e_S|)$$

โดย $g_1 = [-1 \quad 0 \quad 1]$, $g_2 = [-1 \quad 0 \quad 1]^t$ และ ** คือ การกระทำกันของ(e_S) แบบสองมิติ การทำการ Thresholding เพื่อให้รูปภาพผลลัพธ์ที่ได้เป็นรูปไบนารี (e)

g_1 และ g_2 สามารถแสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	-1	
-1	0	1
	1	

ภาพที่ 4.4 แสดงรูปแบบการกระทำของ g_1 และ g_2

เมื่อทำการ Convolution เสร็จแล้ว จะได้ค่า γ แล้วนำค่าที่ได้มาพิจาณาตามเงื่อนไขดังนี้

$e =$ 1 เมื่อ $e_s > \gamma$ และ

0 เมื่อเป็น otherwise

เมื่อ e_s เป็นภาพที่ได้จากการหาขอบภาพและ e คือภาพผลลัพธ์จากการ Threshold

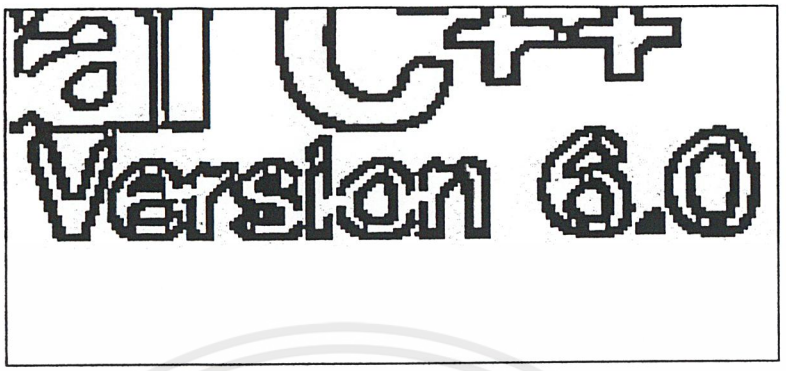
ภาพที่ 4.5 แสดงรูปที่ได้จาการ Threshold

4.1.2 ขั้นตอนที่ 2

เป็นการเชื่อมต่อ pixel ของข้อความที่ขาดจากกัน และเป็นการกำจัดสัญญาณรบกวนและวัตถุที่ไม่ใช่ส่วนประกอบของข้อความออกไป โดยใช้ตัวดำเนินการ Dilation ในการเชื่อมต่อ pixel ของข้อความที่ขาดจากกันโดยเลือก SE ที่เหมาะสม เพื่อให้การเชื่อมต่อ pixel มีประสิทธิภาพมากที่สุด จากนั้นนำรูปภาพผลลัพธ์จากการ Dilation มาทำการกำจัดสัญญาณรบกวนและวัตถุอื่นที่ไม่ใช่ข้อความออกไปด้วยวิธีการ Erosion ซึ่งจะสามารถกำจัดสัญญาณรบกวนออกไปได้มากน้อยเพียงใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นอยู่กับการเลือกใช้ SE ที่เหมาะสมเช่นกัน รูปภาพผลลัพธ์จะเป็นรูปภาพที่ปรากฏเฉพาะขอบของข้อความและมี pixel ที่เชื่อมต่อกันทั้งหมด



ภาพที่ 4.6 แสดงรูปผลลัพธ์ที่ได้จากการ Dilation และ Erosion

4.1.3 ขั้นตอนที่ 3

เป็นขั้นตอนในการกำหนดลำดับตัวอักษร โดยการกำหนดค่าตัวเลขของแต่ละตัวอักษร การกำหนดตัวเลขของตัวอักษรที่อยู่ในรูปภาพ ทำได้โดยวิธี Labeling คือให้ตัวอักษรที่มี pixel ที่เชื่อมติดกันมีค่าในการ Labeling ที่เท่ากัน

โปรแกรมเงื่อนไขที่ใช้ในการ Labeling ได้แสดงเป็น Pseudocode ดังนี้

```
Definitions:  A:i/p N × M binary image , B: o/p N×M labeled image
Algorithm:
        Begin
                % Assign temporary labels to nonzero pixel of the image.
                for i=1 to N ,j=1 to M,do
                  If A(i,j)=1,then sequentially assign unique integer number to B(i,j),
                end do
                set max_temp_label=the maximum assigned integer label.
                % Propagate maximum label per connected component
                do while at least one pixel value of B(i,j) changes
                  %back propagation row-wise
                  for i=N to 2 ,j=1 to M, do
                    if both B(i,j) ane B(i-1,j)> 0 then
                      set both B(i,j) and  B(i-1,j) to maximum B(i,j),B(i-1,j)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            set both B(i,j) and  B(i-1,j) to maximum B(i,j),B(i-1,j)
        end do
        % back propagation column-wise
        for i =1 to N, j=M to 2,do
            if both B(i,j) and B(i,j-1) to maximum(B(i,j),B(i,j-1)
        end do
        % forward propagation row-wise
        for i=1 to N-1, j=1 to M, do
            if both B(i,j) and B(i+1,j) > 0,then
                set both B(i,j) and B(i+1,j) to maximum(B(i,j),B(i+1,j)
        end do
        % forward propagation columm-wise
        for i=1 to N, j=1 to M-1, do
            if both B(i,j) and B(i,j+1) > 0,then
                set both B(i,j) and B(i,j+1) to maximum( B(i,j)B(i,j+1))
        end do
    end wile
    % Sequential relabeling
    for K=1 to max_temp_label.do
        initialize array_label(k)=0
    end do
    max_label=0
    for i=1 to N,j=1 to M.do
        if A (i,j)=1,then
            if array_label(B(i,j))=0,then
                increment max-label
                array_label(B(i,j))=max_label
                B(i,j)=max_label
            else
                B(i,j)=array_label(B(i,j))
            end do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 4.1 การ Labeling

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	1	0	2	2	0
0	0	1	1	1	0	1	1	1	0	2	2	0
0	0	1	1	1	1	1	1	1	0	2	2	0
0	0	1	1	0	1	0	1	1	0	2	2	0
0	0	1	1	0	0	0	1	1	0	2	2	0
0	0	1	1	0	0	0	1	1	0	2	2	0
0	0	1	1	0	0	0	1	1	0	2	2	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

จะเห็นได้ว่าตัวอักษรที่มี Pixel ที่เชื่อมติดกัน จะมีค่าในการ labeling ที่เท่ากัน จากตัวอย่างที่ 4.1 เลข 1 จะแทน Pixel ที่เชื่อมติดกันของตัวอักษร M และเลข 2 จะแทน Pixel ที่เชื่อมติดกันของตัวอักษร I

4.1.4 ขั้นตอนที่ 4

□□ เป็นขั้นตอนในการวิเคราะห์ขอบเขตของข้อความโดยหาค่าจุดต่ำสุดและสูงสุดของความกว้างและความยาวของแต่ละตัวอักษร โดยที่แต่ละตัวอักษรจะมีค่ากำกับ เราจะได้ค่าจุดสูงสุดและจุดต่ำสุดของความกว้างและความยาวออกมาเป็นกรอบสี่เหลี่ยมล้อมตัวอักษรที่ต้องการดึงออกมาแต่ภายในกรอบสี่เหลี่ยมบางกรอบอาจจะยังเป็นวัตถุที่ไม่ใช่ตัวอักษรอยู่ เมื่อได้กรอบสี่เหลี่ยมที่ภายในเป็นตัวอักษรจริงๆ แล้ว จะทำการนำเอาค่าต่ำสุดและสูงสุดนั้น ไปทำการดึงเอาตัวอักษรจากในรูปต้นแบบออกมา ก็จะได้รูปของข้อความที่ดึงออกมาจากรูปภาพ

ข้อความที่เราจะทำการตรวจจับนั้น มักจะเป็นตัวอักษรที่ใหญ่บ้างเล็กบ้างซึ่งไม่เป็นที่น่าพอใจ เช่น ไม่เป็นตัวอักษรที่สามารถกระทำได้โดยใช้ตัวดำเนินการของ Morphology อัตราส่วนของตัวอักษรที่สามารถจะทำการดึงข้อความออกมาได้นั้นต้องมีขนาดดังต่อไปนี้

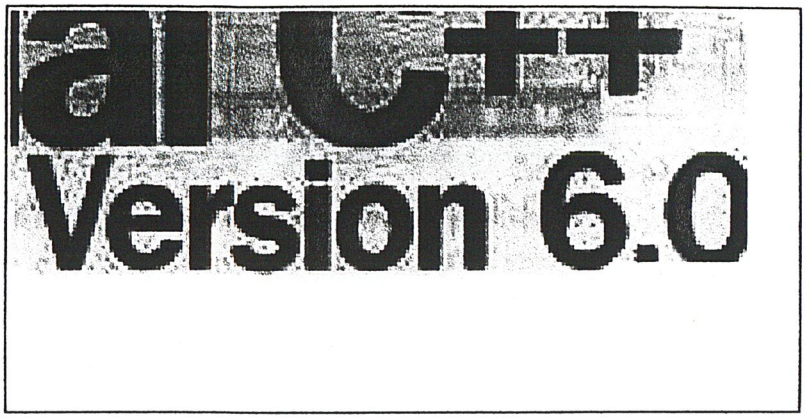
$$W/L, L/W \geq 1/5 \qquad (17)$$

$$W > W_c/8 \qquad (18)$$

$$W < W_c/3 \qquad (19)$$

เมื่อ W แทนด้วยความกว้างเป็น Pixel ของแต่ละกรอบสี่เหลี่ยมที่ล้อมรอบตัวอักษร W_c แทนด้วยความกว้างเป็น Pixel ของรูปภาพต้นแบบ และ L แทนด้วยความยาวเป็น Pixel ของแต่ละกรอบสี่เหลี่ยมที่ล้อมรอบตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.7 แสดงรูปสำเร็จในการดึงข้อความออกจากรูปภาพ

4.2 ตัวอย่างและการพิจารณา

ในหัวข้อที่ได้กล่าวมาแล้วนั้นเป็นขั้นตอนในการตรวจจับข้อความออกมาจากรูปภาพ ซึ่งมีอยู่ด้วย 4 ขั้นตอนหลักที่ใช้กัน แต่ในการตรวจจับข้อความออกจากรูปภาพนั้นยังมีข้อจำกัดอยู่ คือ ข้อความที่อยู่ในรูปภาพที่เล็กเกินไปจะถูกพิจารณาให้เป็นสัญญาณรบกวน และข้อความที่อยู่ในรูปภาพที่ใหญ่เกินไปจะถูกพิจารณาให้เป็น graphics ตามลำดับ แต่ยังมีข้อยกเว้นในการตรวจจับข้อความที่อยู่ในแนวนอนและแนวโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

การทดสอบประสิทธิภาพของตัวตรวจหามอร์โฟโลยี จำเป็นต้องใช้ภาพต้นแบบที่มีขนาดไม่เกิน 200 X 200 Pixel เพราะภาพต้นแบบที่มีขนาดใหญ่กว่าที่กำหนดจะไม่สามารถดึงข้อความออกมาได้ และภาพต้นแบบต้องมีค่าความเข้มของจุดภาพอยู่ในช่วง (0, 255) สำหรับการประเมินผลการทดลองที่ได้จากการทดสอบโปรแกรมนั้น จะประเมินผลตามภาพผลลัพธ์ที่ได้ กล่าวคือ เมื่อเราทราบจำนวนตัวอักษรที่อยู่ในรูปภาพต้นแบบแล้ว เราจะมานับจำนวนตัวอักษรจากภาพผลลัพธ์ที่โปรแกรมสามารถดึงออกมาได้แล้วคำนวณออกมาเป็นเปอร์เซ็นต์ความผิดพลาด ทำให้ทราบประสิทธิภาพของโปรแกรมและตัวตรวจหามอร์โฟโลยีนี้

การใช้งานโปรแกรม

โปรแกรมที่จัดทำขึ้นนี้ใช้ภาษาซีในการเขียนโปรแกรมดังนั้นจึงต้องใช้งานใน MS DOS การเข้าถึง MS DOS มีอยู่ด้วยกันหลายวิธี ดังนี้

1. เข้า Start->Run->พิมพ์ command->ok
2. เข้า Start->Programs->MS DOS Prompt
3. เข้า Start->Shut Down->MS DOS Prompt

เมื่อเข้าถึง MS DOS แล้วสามารถที่จะพิมพ์ข้อความในการใช้งานโปรแกรมได้ดังนี้

C:/Extrac Name.pgm result.pgm

โดยที่ ข้อความ Extrac หมายถึงชื่อของโปรแกรมซึ่งเป็นไฟล์ EXE

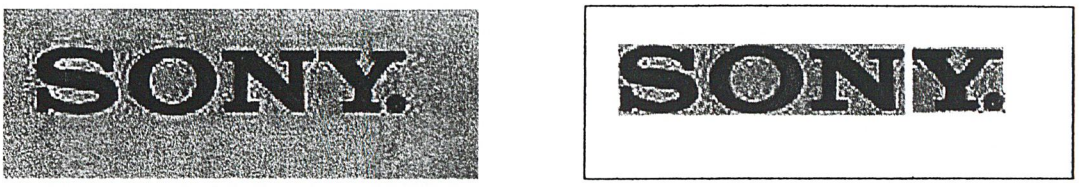
ข้อความ Name.pgm หมายถึงชื่อของรูปภาพที่ต้องการดึงข้อความออกมาต้องเป็นไฟล์ที่มีนามสกุล PGM

ข้อความ result.pgm หมายถึงชื่อของรูปภาพผลลัพธ์จากการดึงข้อความออกมาสามารถที่ตั้งชื่อตามที่ต้องการได้แต่ต้องมีนามสกุล PGM เท่านั้น

จากการทดลองโปรแกรมโดยใช้รูปภาพต้นแบบที่มีข้อความเป็นส่วนประกอบจำนวน 20 รูปและได้ผลลัพธ์เป็นที่ไม่น่าพอใจคือมีประสิทธิภาพในการดึงข้อความออกมาจากรูปภาพได้ประมาณ 20% และจะมีข้อจำกัดอยู่คือไม่สามารถที่จะทำการดึงข้อความที่มีตัวอักษรเล็กกว่า 1/8 ของความกว้างของรูปภาพต้นแบบได้ และในกรณีที่วัตถุที่มีเงื่อนไขตรงตามการกำหนดการเป็นตัวอักษร ผลที่ได้ก็จะมีความผิดพลาดเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างผลการทดลองการใช้งานของโปรแกรมแสดงได้ดังนี้



(ก) (ข)

ภาพที่ 5.1 ภาพต้นแบบและภาพผลลัพธ์ของข้อความ "SONY"

ภาพที่ 5.1 (ก) เป็นภาพที่มีโทนสีที่เหมาะสม จะสามารถดึงข้อความออกมาจากภาพได้เป็นอย่างดีเพราะว่าสีของข้อความ(Text Color) ซึ่งมีสีที่เข้ม และสีของพื้น (Blackground Color) ซึ่งมีสีที่อ่อนนั้นมีความแตกต่างกันอย่างชัดเจน จึงสามารถใช้ตัวตรวจหามอร์โฟโลยีดึงข้อความออกมาได้ดังภาพที่ 5.1 (ข)

ผลที่ได้จะเห็นได้ว่า จะประกอบไปด้วยข้อความ 2 ชุดคือ "SON" และ "Y" แต่ถ้าเป็นไปตามเงื่อนไขจะสามารถแยกได้ 4 ชุดคือ "S" , "O", "N", "Y" เหตุที่แยกได้เพียง 2 ชุดเนื่องจากในขั้นตอนการ Thersholding มี pixel คั้นระหว่างตัว "S" กับตัว "O" และ "O" กับ "N" ทำให้ได้ผลลัพธ์จัดชุดตัวอักษรอยู่ในกลุ่มเดียวกัน ทางแก้ไขปัญหาคือทำการเปลี่ยน SE เพื่อให้จุด pixel ที่เชื่อมต่อระหว่างตัวอักษรแต่ละตัวหายไป

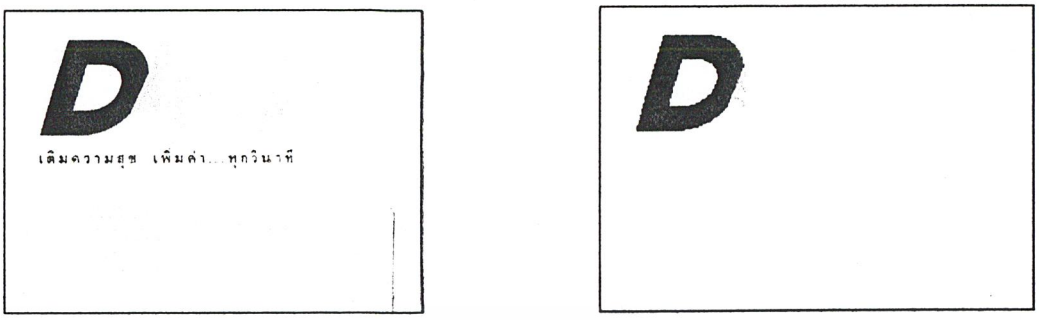
(ก) (ข)

ภาพที่ 5.2 ภาพต้นแบบและภาพผลลัพธ์ของข้อความ "แนวหน้า"

ภาพที่ 5.2 (ก) สีของตัวอักษร และสีพื้นมีความแตกต่างกันอย่างชัดเจน จึงสามารถดึงข้อความออกมาจากรูปภาพได้ดังภาพที่ 5.2 (ข) เพราะสีของข้อความมีสีอ่อน ซึ่งมีความแตกต่างกับสีพื้น ซึ่งมีสีเข้มอย่างชัดเจน ตัวตรวจหามอร์โฟโลยีจึงดึงข้อความออกมาได้

จากผลที่ได้จะเห็นได้ว่าตัวอักษรคำว่า "แนวหน้า" ติดกันเกินไปทำให้ไม่สามารถตัวอักษรเป็นชุดได้ ทำให้เห็นเพียง 1 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) (ข)

ภาพที่ 5.3 ภาพต้นแบบและภาพผลลัพธ์ของข้อความ “D4U”

ในรูปที่ 5.3 (ก) มีโทนสีสว่างมาก จะเห็นได้ว่าสีของข้อความเกือบจะเป็นสีเดียวกับสีพื้น ดังนั้นตัวตรวจหามอร์โฟโลยี จึงไม่สามารถดึงข้อความออกมาจากรูปภาพได้อย่างมีประสิทธิภาพ ดังภาพที่ 5.3 (ข) ซึ่งเป็นข้อจำกัดอย่างหนึ่งในการใช้โปรแกรมในการดึงข้อความออกมา คือไม่สามารถดึงข้อความที่มีสีของข้อความใกล้เคียงกับสีพื้นได้

ผลที่ได้จะเห็นได้ว่าข้อความ “4U” ซึ่งมีโทนสีที่ใกล้เคียงกับสีพื้นทำให้ไม่สามารถตรวจจับได้ ส่วนข้อความว่า “เติมความสุข เพิ่มค่า...ทุกวินาที” ก็ไม่สามารถตรวจจับได้เนื่องจากมีขนาดเล็กเกินกว่าขอบเขตที่กำหนด

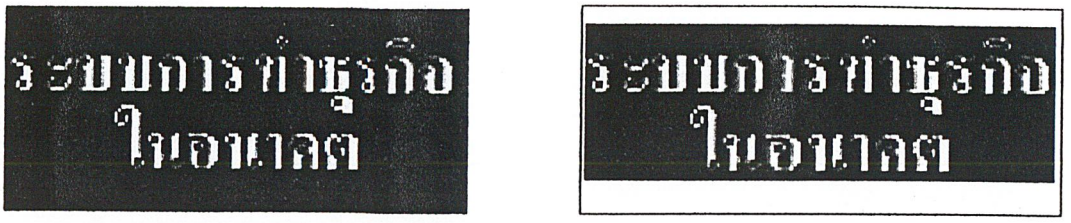
(ก) (ข)

ภาพที่ 5.4 ภาพต้นแบบและภาพผลลัพธ์ของข้อความ “msn”

ภาพที่ 5.4 (ก) มีโทนสีสว่าง แต่ตัวตรวจหามอร์โฟโลยีไม่สามารถที่จะดึงข้อความได้อย่างมีประสิทธิภาพ เนื่องจากมีวัตถุที่มีขนาดใหญ่เป็นส่วนประกอบในภาพซึ่งเป็นข้อจำกัดอีกอย่างหนึ่งของตัวตรวจหามอร์โฟโลยี ที่มีวัตถุขนาดใหญ่เป็นส่วนประกอบอยู่ในรูปภาพ

ผลที่ได้จะเห็นได้ว่า ตัวอักษร “msn” สามารถตรวจจับได้ แต่สาเหตุที่มีแถบสีดำติดมาด้วยเนื่องจากตัวดำเนินการมอร์โฟโลยีเห็นเป็นชุดเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) (ข)

ภาพที่ 5.5 ภาพต้นแบบภาพผลลัพธ์ของข้อความ "ระบบการทำธุรกิจในอนาคต"

ภาพที่ 5.5 (ก) สีของตัวอักษรและสีพื้นมีความแตกต่างกันอย่างชัดเจน ตัวตรวจหามอร์โฟโลยีจึงทำการดึงข้อความออกมาได้ แต่จะสังเกตเห็นได้ว่า ข้อความในบรรทัดล่างนั้นไม่ได้ทำการดึงแต่ข้อความแต่จะมีภาพของพื้นมาด้วย ทั้งนี้เนื่องจากว่าในขั้นตอนของการ Gradient นั้น ตัวอักษรอยู่ใกล้กันมากดังนั้นในขั้นตอนนี้ตัวอักษรมีจุดที่ติดต่อกัน จึงทำการดึงข้อความออกมาได้เป็นขอบเขตทั้งหมดจึงดึงออกมาได้ดังภาพที่ 5.5 (ข)

(ก) (ข)

ภาพที่ 5.6 ภาพต้นแบบและภาพผลลัพธ์ของข้อความ "SPACE COWBOYS"

ภาพที่ 5.6 (ก) เป็นภาพที่สีของข้อความนั้นแตกต่างจากสีพื้น แต่จะสังเกตเห็นได้ว่าทางด้านมุมบนด้านขวามีวัตถุที่มีสีใกล้เคียงกับสีของข้อความ ตัวตรวจหามอร์โฟโลยีจึงไม่สามารถดึงเฉพาะข้อความเท่านั้นแต่จะดึงเอาวัตถุนั้นออกมาด้วย และเป็นข้อจำกัดของตัวตรวจหามอร์โฟโลยีนี้คือจะดึงเอาวัตถุที่มีสีใกล้เคียงกับสีของข้อความออกมาด้วย

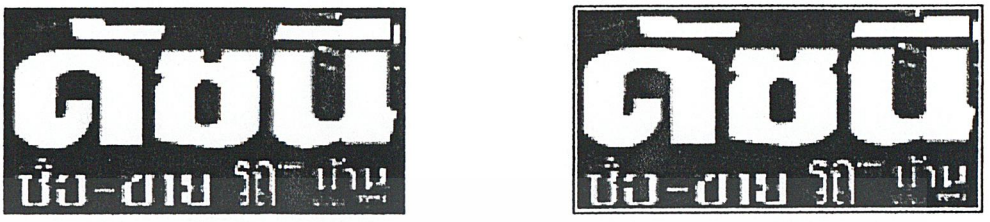
(ก) (ข)

ภาพที่ 5.7 ภาพต้นแบบและภาพผลลัพธ์ของข้อความ " EthaiCD.com"

ภาพที่ 5.7 ภาพมีสีของข้อความมีความแตกต่างจากสีพื้นอย่างชัดเจน แต่จะมีวัตถุวงกลมที่มีสีใกล้เคียงกับสีของข้อความ ตัวตรวจหามอร์โฟโลยีจึงดึงวัตถุนั้นออกมาด้วย ดังภาพที่ 5.7 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าภาพผลลัพธ์จะมีวัตถุนั้นมาด้วย สาเหตุเนื่องมาจากการ Thersholding และการเลือก SE มากระทำกับรูปภาพ ถ้าต้องการให้รูปเป็นไปตามข้อกำหนด สามารถเปลี่ยน SE ได้เพื่อให้ได้ภาพที่ดีที่สุดตรงตามข้อกำหนด



(ก) (ข)

ภาพที่ 5.8 ภาพต้นแบบและภาพผลลัพธ์ของข้อความ " ดัชนี"

ภาพที่ 5.8 จะเห็นได้ว่าข้อความมีขนาดใหญ่มากและมีตัวอักษรอยู่เต็มรูปภาพดังนั้นตัวตรวจหามอร์โฟโลยีจับข้อความออกมาทั้งหมด ดังภาพที่ 5.8 (ข) ซึ่งจะเป็นข้อจำกัดอย่างหนึ่ง คือ รูปภาพไม่ควรจะมีข้อความที่ใหญ่เกินไป

(ก) (ข)

ภาพที่ 5.9 ภาพต้นแบบและภาพผลลัพธ์ของข้อความ "ถ้าคุณมีความตั้งใจจริง"

ภาพที่ 5.9 (ก) จะเห็นได้ว่าภาพต้นแบบมีวัตถุที่ไม่พึงประสงค์อยู่ที่พื้นหลัง (Background) ของข้อความเต็มไปหมด ตัวตรวจหามอร์โฟโลยีจึงไม่สามารถดึงข้อความออกมาได้อย่างมีประสิทธิภาพได้ ดังภาพผลลัพธ์ที่ 5.9 (ข) คือตัวตรวจหามอร์โฟโลยีจะดึงวัตถุเหล่านั้นออกมาด้วย สาเหตุเนื่องมาจากการทำ Thersholding ไม่ดีนัก

(ก) (ข)

ภาพที่ 5.10 ภาพต้นแบบและภาพผลลัพธ์ของข้อความ "INFO" และ "ฐานข้อมูลหน่วยงานรัฐ"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 5.10 (ก) จะเห็นได้ว่าในภาพต้นแบบมีภาพของสัญลักษณ์ทางการค้าปรากฏอยู่ด้วย ตัวตรวจหามอร์โฟโลยีจึงดึงเอาทั้งข้อความและภาพสัญลักษณ์นั้นออกมาด้วยดังภาพที่ 5.10 (ข) เป็นข้อจำกัดของตัวตรวจหามอร์โฟโลยีที่จะดึงเอาวัตถุที่มีขอบเขตที่สามารถเป็นข้อความออกมาด้วยนั้นเอง

ผลที่ได้จะเห็นว่าสามารถแบ่งตัวอักษรออกเป็น 2 ชุดใหญ่ โดยชุดแรกที่มีข้อความว่า "GINFO" ส่วนชุดที่สอง มีข้อความว่า "ฐานข้อมูลหน่วยงานรัฐ" เหตุที่ไม่เห็นความแตกต่างของภาพเนื่องจากสีพื้นของภาพต้นแบบมีโทรสีใกล้เคียงสีขาว ผลที่ได้จึงออกมาใกล้เคียงกัน

จากภาพผลการทดลอง ในการใช้ตัวตรวจหามอร์โฟโลยีทำการดึงข้อความ ออกมาจากรูปภาพที่ได้จัดทำขึ้นนี้ผลลัพธ์ที่ได้เป็นที่น่าพอใจ คือสามารถดึงข้อความออกมาจากรูปภาพได้ดังภาพที่แสดงไว้ด้านบน แต่โปรแกรมที่ได้จัดทำขึ้นนี้ยังมีข้อจำกัดอยู่คือไม่สามารถดึงข้อความได้อย่างมีประสิทธิภาพได้ถ้าภาพต้นแบบนั้นมีวัตถุที่มีขนาดตรงตามเงื่อนไขในการกำหนดตัวอักษรดังแสดงในภาพที่ 5.10 และไม่สามารถดึงข้อความที่สีของตัวอักษรและสีพื้นใกล้เคียงกันได้ดังแสดงในภาพที่ 5.3 และภาพที่มีวัตถุไม่พึงประสงค์อยู่ภายในรูปภาพดังภาพที่5.9 ซึ่งต้องทำการพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
บทสรุปและวิจารณ์

การตรวจหามอร์โฟโลยีเป็นการตรวจหาและดึงข้อความออกมาจากรูปภาพโดยมีขั้นตอนในการดึงข้อความอยู่ 4 ขั้นตอนซึ่งขั้นตอนทั้งหมดที่ได้นำเสนอนี้ได้ใช้ตัวดำเนินการมอร์โฟโลยี ที่สามารถดึงข้อความออกมาจากรูปภาพได้

ตัวดำเนินการ Dilation และ Erosion เป็นตัวดำเนินการพื้นฐานของมอร์โฟโลยีที่ต้องทำการศึกษาเป็นอันดับแรกซึ่งสามารถใช้งานได้ดีกับภาพไบนารี่ ขึ้นอยู่กับกำหนด SE แต่ความยุ่งยากก็จะตามมาในการคำนวณหาค่าความเป็นสมาชิกที่เหมาะสม ตัวดำเนินการพื้นฐานของมอร์โฟโลยีทั้งสองตัวนี้ยังนำไปประยุกต์ใช้งานร่วมกัน เพื่อสร้างตัวดำเนินการใหม่ให้เกิดขึ้นคือตัวดำเนินการ Opening เป็นตัวดำเนินการที่เกิดจากการทำ Erosion แล้วผลลัพธ์ที่ได้มาทำการ Dilation อีกครั้งหนึ่งใช้ในการกำจัดสัญญาณรบกวนที่เกิดจากสัญญาณรบกวนที่เป็นโทนสีขาว ส่วนตัวดำเนินการ Closing เกิดจากการทำ Dilation แล้วนำผลลัพธ์ที่ได้มาทำการ Erosion อีกครั้ง ซึ่งเป็นตัวดำเนินการที่ใช้ในการกำจัดสัญญาณรบกวนที่เป็นโทนสีดำ ดังนั้นตัวดำเนินการ Opening และ Closing เป็นตัวดำเนินการของมอร์โฟโลยีที่เกิดจากการประยุกต์ใช้งาน

ส่วนการหาขอบภาพสามารถใช้ตัวดำเนินการของมอร์โฟโลยีคือตัวดำเนินการ Gradient ซึ่งมีโครงสร้างของตัวดำเนินการที่เกิดจาก การนำตัวดำเนินการพื้นฐานของมอร์โฟลยี Dilation และ Erosion มาหาค่าผลต่างของความเป็นสมาชิกในแต่ละการดำเนินการ และตัวดำเนินการ Threshold เป็นการแปลงรูปภาพเกรย์เป็นรูปภาพขาวดำ ในขั้นตอนต่อไปเป็นการนำวิธีการ Labeling ซึ่งเป็นขั้นตอนในการกำหนดลำดับตัวอักษร โดยการกำหนดค่าตัวเลขของแต่ละตัวอักษร

จากการตรวจหาข้อความโดยใช้มอร์โฟโลยีในปริญญานิพนธ์ฉบับนี้ สามารถทำการดึงข้อความจากรูปภาพได้อย่างมีประสิทธิภาพแต่ยังมีข้อจำกัดอยู่คือยังไม่สามารถดึงข้อความที่มีขนาดใหญ่เกินกว่า 1/3 ของความกว้างของภาพต้นแบบ และข้อความที่มีขนาดเล็กกว่า 1/8 ของความกว้างของภาพต้นแบบก็ไม่สามารถดึงออกมาได้เช่นกัน และรูปภาพที่สีของข้อความ (Text Color)กับสีของพื้น (Background Color) ที่มีความใกล้เคียงกันก็ไม่สามารถ

จากภาพผลการทดลอง ในการใช้ตัวตรวจหามอร์โฟโลยี ทำการดึงข้อความออกมาจากรูปภาพที่ได้จัดทำขึ้นนี้ผลลัพธ์ที่ได้ไม่เป็นที่น่าพอใจ คือสามารถดึงข้อความออกมาจากรูปภาพได้เพียง 20% เนื่องมาจากสาเหตุต่างๆดังต่อไปนี้

1. ขั้นตอนการ Threshold ยังไม่ได้ผลตามทฤษฎีทำให้ผลที่ออกมาไม่ดีเท่าที่ควร
2. Edge dectection โดยขอบของวัตถุที่ได้แม้วัตถุชิ้นเดียวกันก็ยังมีส่วนที่ไม่ต่อเนื่องกันทั้งหมด
3. การทำ Program Labeling ยังไม่สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสนอแนะ

ในขั้นตอนในการกำจัดสัญญาณรบกวน สามารถเลือก SE ที่เหมาะสมกับรูปภาพและมีประสิทธิภาพที่ดีที่สุดได้ และในขั้นตอนที่ 2 ต้องทำการเลือก SE แบบไบนารี ที่สามารถเชื่อม pixel และกำจัดสัญญาณรบกวนได้อย่างมีประสิทธิภาพดีที่สุด และขั้นตอนสุดท้ายเป็นการวิเคราะห์ขอบเขตของตัวอักษร สามารถสร้างเงื่อนไขในการวิเคราะห์ใหม่ได้เพื่อให้การดึงข้อความจากรูปให้มีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

สาธิต อินทจักร์, 2538, การหาขอบภาพหลายระดับด้วยมอร์โฟโลยี, วารสารการประชุมทางวิชาการ ครั้งที่ 37,หน้าที่ 32-37.

สาธิต อินทจักร์, 2540, การแสดงรูปภาพของไฟล์บิตแมป, วารสารอินเตอร์เน็ตอินทราเน็ต ฉบับที่ 13, หน้า 39-56.

ชายแดน สุทเธนทร์, อภิโชค เทียนวิไลรัตน์, อภิชาติ พรหมทัศน์, 2542, การประมวลผลภาพขั้นพื้นฐานด้วยฟัซซี่มอร์โฟโลยี, ปริญญานิพนธ์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

สาธิต อินทจักร์ และ กิตติ ไพฑูรย์วัฒนกิจ, 2540, การวิเคราะภาพลักษณ์เอกสารภาษาไทยด้วยทฤษฎีกราฟ, วิศวกรรมสาร "ฉบับวิจัยและพัฒนา", หน้าที่ 50-60, ปีที่ 8 ฉบับที่ 2, 2540.

Serra.J, 1982, Image Analysis and Mathematical morphology, vol.1, Academic press, London.

Robert M. Haralick, Linda G. Shapiro,1992, Computer and robot vision (vol.1), Addison-Wesley.

Yassin M.Y. Hasan and Lina J. Karam, 2000, Morphology Text Extraction from Images, IEEE Transaction on image Processing, Vol.9, No. 11, Novenber 2000.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมตัวตรวจหามอร์โฟโลยี

```
#define MORPHOLOGY_INCLUDE
#include "stdio.h"
#include "math.h"
#include "malloc.h"
#define MLIB
#define SQRT2 1.414213562
#ifndef PI
#define PI 3.1415926535
#endif
#define BLACK 0
#define WHITE 1

/* The image header data structure  */
struct header {
        int nr, nc;   /* Rows and columns in the image */
        int oi, oj;   /* Origin */
};

/* The IMAGE data structure  */
struct image {
        struct header *info;    /* Pointer to header */
        unsigned char **data;   /* Pixel values */
};
struct se_struct {
        unsigned char **data;
        int nr, nc, oi, oj;
};
typedef struct image * IMAGE;
typedef struct se_struct * SE;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#if defined(MLIB)
int   PGM_ORIGIN_COL=0, PGM_ORIGIN_ROW=0;
#else
extern int PGM_ORIGIN_COL, PGM_ORIGIN_ROW;
#endif
#if defined(MLIB)
int   PBM_SE_ORIGIN_COL=0, PBM_SE_ORIGIN_ROW=0;
#else
extern int PBM_SE_ORIGIN_COL, PBM_SE_ORIGIN_ROW;
#endif

int get_se (char *filename, SE *p);
int print_se (SE p);
int range (IMAGE im, int i, int j);
void dil_apply (IMAGE im, SE p, int ii, int jj, IMAGE res);
int bin_dilate (IMAGE im, SE p);
void erode_apply (IMAGE im, SE p, int ii, int jj, IMAGE res);
int bin_erode (IMAGE im, SE p);
IMAGE newimage (int nr, int nc);
void freeimage (IMAGE im);
int read_pbm (char *fn, IMAGE *im);
void pbm_param (char *s);
void write_pbm (char *filename, IMAGE image);
void pbm_getln (FILE *f, char *b);
void get_num_pbm (FILE *f, char *b, int *bi, int *res);
int read_pgm (char *fn, IMAGE *im);
void pgm_param (char *s);
void write_pgm (char *filename, IMAGE image);
void pgm_getln (FILE *f, char *b);
void get_num_pgm (FILE *f, char *b, int *bi, int *res);
int gray_erode (IMAGE im);
int gray_dilate (IMAGE im);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int gray_erode1 (IMAGE im1);
int gray_filter (IMAGE im,IMAGE im1);
int gray_gradient (IMAGE im,IMAGE im1);
int gray_threshold (IMAGE im);
int text_label (IMAGE im);
int region_analysis (IMAGE im,IMAGE im1);
/*************************************************/
int range (IMAGE im, int i, int j)
{
        if ((i<0) || (i>=im->info->nr)) return 0;
        if ((j<0) || (j>=im->info->nc)) return 0;
        return 1;
}

/* GET_SE - Read a structuring element from a file  */
int get_se (char *filename, SE *p)
{
        SE q;
        IMAGE seim;

        *p = (SE)0;

/* Read the PBM format structuring element file */
        if (read_pbm (filename, &seim) == 0) return 0;

/* Allocate a structure for the SE */
        q = (SE)malloc (sizeof (struct se_struct));
        if (q == 0)
        {
        printf ("Can't allocate structuring element in GET_SE.\n");
         return 0;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        q->nr = seim->info->nr;        q->nc = seim->info->nc;
        q->oi = PBM_SE_ORIGIN_ROW;     q->oj = PBM_SE_ORIGIN_COL;
        q->data = seim->data;

        free (seim->info);
        seim->data = 0;
        free (seim);
        *p = q;

        return 1;
}

/* PRINT_SE - Print a structuring element to the screen */

int print_se (SE p)
{
        int i,j;

        printf ("\n==========\n");
        if (p == NULL)
         printf (" Structuring element is NULL.\n");
        else
        {
         printf ("Structuring element: %dx%d origin at (%d,%d)\n",
                 p->nr, p->nc, p->oi, p->oj);
         for (i=0; i<p->nr; i++)
         {
          printf ("      ");
          for (j=0; j<p->nc; j++)
           printf ("%4d ", p->data[i][j]);
          printf ("\n");
         }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        printf ("\n=============================================================\n");
        return 0;
}

/*  Apply a dilation step on one pixel of IM, reult to RES  */

void dil_apply (IMAGE im, SE p, int ii, int jj, IMAGE res)
{
        int i,j, is,js, ie, je, k;

/* Find start and end pixel in IM */
        is = ii - p->oi;        js = jj - p->oj;
        ie = is + p->nr;        je = js + p->nc;

/* Place SE over the image from (is,js) to (ie,je). Set pixels in RES
  if the corresponding SE pixel is 1; else do nothing.          */
        for (i=is; i<ie; i++)
          for (j=js; j<je; j++)
          {
           if (range(im,i,j))
           {
            k = p->data[i-is][j-js];
            if (k>=0) res->data[i][j] |= k;
           }
          }
}

/* Dilate the given image using the given structuring element */

int bin_dilate (IMAGE im, SE p)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    IMAGE tmp;
    int i,j;

/* Source image empty? */
    if (im==0)
    {
     printf ("Bad image in BIN_DILATE\n");
     return 0;
    }

/* Create a result image */
    tmp = newimage (im->info->nr, im->info->nc);
    if (tmp == 0)
    {
     printf ("Out of memory in BIN_DILATE.\n");
     return 0;
    }
    for (i=0; i<tmp->info->nr; i++)
     for (j=0; j<tmp->info->nc; j++)
      tmp->data[i][j] = 0;

/* Apply the SE to each black pixel of the input */
    for (i=0; i<im->info->nr; i++)
     for (j=0; j<im->info->nc; j++)
      if (im->data[i][j] == WHITE)
       dil_apply (im, p, i, j, tmp);

/* Copy result over the input */
    for (i=0; i<im->info->nr; i++)
     for (j=0; j<im->info->nc; j++)
      im->data[i][j] = tmp->data[i][j];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Free the result image - it was a temp */
        freeimage (tmp);
        return 1;
}

/* Apply a erosion step on one pixel of IM, reult to RES */

void erode_apply (IMAGE im, SE p, int ii, int jj, IMAGE res)
{
        int i,j, is,js, ie, je, k, r;

/* Find start and end pixel in IM */
        is = ii - p->oi;      js = jj - p->oj;
        ie = is + p->nr;      je = js + p->nc;

/* Place SE over the image from (is,js) to (ie,je). Set pixels in RES
  if the corresponding pixels in the image agree.     */
        r = 1;
        for (i=is; i<ie; i++)
        {
          for (j=js; j<je; j++)
          {
            if (range(im,i,j))
            {
              k = p->data[i-is][j-js];
              if ((k == 1) && (im->data[i][j]==0)) r = 0;
            } else if (p->data[i-is][j-js] != 0) r = 0;
          }
          printf ("\n");
        }
        res->data[ii][jj] = r;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Erode the image IM using the structuring element P  */

int bin_erode (IMAGE im, SE p)
{
        IMAGE tmp;
        int i,j;

/* Source image empty? */
        if (im==0)
        {
         printf ("Bad image in BIN_ERODE\n");
         return 0;
        }

/* Create a result image */
        tmp = newimage (im->info->nr, im->info->nc);
        if (tmp == 0)
        {
         printf ("Out of memory in BIN_ERODE.\n");
         return 0;
        }
        for (i=0; i<tmp->info->nr; i++)
         for (j=0; j<tmp->info->nc; j++)
           tmp->data[i][j] = 0;

 /* Apply the SE to each black pixel of the input */
         for (i=0; i<im->info->nr; i++)
          for (j=0; j<im->info->nc; j++)
             erode_apply (im, p, i, j, tmp);

 /* Copy result over the input */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for (i=0; i<im->info->nr; i++)
         for (j=0; j<im->info->nc; j++)
          im->data[i][j] = tmp->data[i][j];

/* Free the result image - it was a temp */
        freeimage (tmp);
        return 1;
}
/* Erosion the given image using the simple structuring element */
int gray_erode (IMAGE im)
{
        IMAGE tmp;
        int i,j,k,l;
        int tm[8];
/* Source image empty? */
        if (im==0)
        {
         printf ("Bad image in GRAY_EROSION\n");
         return 0;
        }

/* Create a result image */
        tmp = newimage (im->info->nr, im->info->nc);
        if (tmp == 0)
        {
         printf ("Out of memory in GRAY_ERSION.\n");
         return 0;
        }

        for (i=0; i<tmp->info->nr; i++)
         for (j=0; j<tmp->info->nc; j++)
          tmp->data[i][j] = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for (i=0; i<=8; i++)
          tm[i] = 0;

          l=0;
/* Apply the SE to each black pixel of the input */
        for (i=1; i<im->info->nr; i++)
         for (j=1; j<im->info->nc; j++)
          {
          tm[0] = im->data[i-1][j-1];
          tm[1] = im->data[i-1][j];
          tm[2] = im->data[i-1][j+1];
          tm[3] = im->data[i][j-1];
          tm[4] = im->data[i][j];
          tm[5] = im->data[i][j+1];
          tm[6] = im->data[i+1][j-1];
          tm[7] = im->data[i+1][j];
          tm[8] = im->data[i+1][j+1];

           k=255;
          for (l=0; l<=7; l++)
          {
                if (k > tm[l])
                {
                  k = tm[l];
                }
           }
           tmp->data[i][j] = k;
          }

/* Copy result over the input */
        for (i=1; i<im->info->nr-1; i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            for (j=1; j<im->info->nc-1; j++)
             im->data[i][j] = tmp->data[i][j];

/* Free the result image - it was a temp */
        freeimage (tmp);
        return 1;
}
int gray_erode1 (IMAGE im)
{
        IMAGE tmp,im1;
        int i,j,k,l;
        int tm[8];
/* Source image empty? */
        if (im==0)
        {
         printf ("Bad image in GRAY_EROSION\n");
         return 0;
        }

/* Create a result image */
        tmp = newimage (im->info->nr, im->info->nc);
        if (tmp == 0)
        {
         printf ("Out of memory in GRAY_EROSION.\n");
         return 0;
        }

        for (i=0; i<tmp->info->nr; i++)
         for (j=0; j<tmp->info->nc; j++)
           tmp->data[i][j] = 0;

        for (i=0; i<=8; i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    tm[i] = 0;

    l=0;

  for (i=0; i<im->info->nr; i++)
   for (j=0; j<im->info->nc; j++)
    {
     tm[0] = im->data[i-1][j-1];
     tm[1] = im->data[i-1][j];
     tm[2] = im->data[i-1][j+1];
     tm[3] = im->data[i][j-1];
     tm[4] = im->data[i][j];
     tm[5] = im->data[i][j+1];
     tm[6] = im->data[i+1][j-1];
     tm[7] = im->data[i+1][j];
     tm[8] = im->data[i+1][j+1];

     k=255;
     for (l=0; l<=7; l++)
     {
          if (k > tm[l])
          {
            k = tm[l];
          }
      }
      tmp->data[i][j] = k;
   }

/* Copy result over the input */
        for (i=0; i<im1->info->nr; i++)
         for (j=0; j<im1->info->nc; j++)
          im1->data[i][j] = tmp->data[i][j];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Free the result image - it was a temp */
        freeimage (tmp);
        return 1;
}
/* Dilate the given image using the simple structuring element */
int gray_dilate (IMAGE im)
{
        IMAGE tmp;
        int i,j,k,l;
        int tm[8];
/* Source image empty? */
        if (im==0)
        {
         printf ("Bad image in GRAY_DILATE\n");
         return 0;
        }

/* Create a result image */
        tmp = newimage (im->info->nr, im->info->nc);
        if (tmp == 0)
        {
         printf ("Out of memory in GRAY_DILATE.\n");
         return 0;
        }

        for (i=0; i<tmp->info->nr; i++)
         for (j=0; j<tmp->info->nc; j++)
           tmp->data[i][j] = 0;

        for (i=0; i<=8; i++)
           tm[i] = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        l=0;
/* Apply the SE to each black pixel of the input */
        for (i=1; i<im->info->nr-1; i++)
         for (j=1; j<im->info->nc-1; j++)
          {
          tm[0] = im->data[i-1][j-1];
          tm[1] = im->data[i-1][j];
          tm[2] = im->data[i-1][j+1];
          tm[3] = im->data[i][j-1];
          tm[4] = im->data[i][j];
          tm[5] = im->data[i][j+1];
          tm[6] = im->data[i+1][j-1];
          tm[7] = im->data[i+1][j];
          tm[8] = im->data[i+1][j+1];

          k=0;
          for (l=0; l<=8; l++)
          {
              if (k < tm[l])
              {
                k = tm[l];
              }
           }
           tmp->data[i][j] = k;
          }

/* Copy result over the input */
        for (i=1; i<im->info->nr-1; i++)
         for (j=1; j<im->info->nc-1; j++)
          im->data[i][j] = tmp->data[i][j];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Free the result image - it was a temp */
        freeimage (tmp);
        return 1;
}
/*----------------------------------------*/
int gray_filter(IMAGE im,IMAGE im1)
{

        int i,j,k;

/* Source image empty? */
        if ((im&&im1)==0)
        {
         printf ("Bad image in GRAY_FILTER\n");
         return 0;
        }
 /* ---------------------------------*/
         /*      CLOSE - OPEN        */
 /*---------------------------------*/
        gray_dilate(im);
        gray_erode(im);
        gray_erode(im);
        gray_dilate(im);
 /* ---------------------------------*/
         /*      OPEN - CLOSE        */
 /*---------------------------------*/
        gray_erode(im1);
        gray_dilate(im1);
        gray_dilate(im1);
        gray_erode(im1);

          k=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Apply all pixel of the input */
        for (i=1; i<im->info->nr-1; i++)
         for (j=1; j<im->info->nc-1; j++)
           {
            k = (im->data[i][j] + im1->data[i][j]) /2;
            im->data[i][j] = k;
            im1->data[i][j] = k;
           }

        return 1;
}
/*-----------------------------------------------*/
int gray_gradient(IMAGE im,IMAGE im1)
{

        int i,j,k;

/* Source image empty? */
        if (im==0)
        {
         printf ("Bad image in GRAY_GRADINT\n");
         return 0;
        }
/* Perform the dilation */
        gray_dilate(im);
/* Perform the erosion */
        gray_erode1(im);

          k=0;
/* Apply all pixel of the input */
        for (i=0; i<im->info->nr; i++)
         for (j=0; j<im->info->nc; j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
             k = im->data[i][j]-im1->data[i][j] ;
             im->data[i][j] = k;
            }

        return 1;
}
/*-----------------------------------------*/
int gray_threshold(IMAGE im)
{
        int i,j,s,a,b,g,grammar;
        int tn[3];
        float value,sum;

/* Source image empty? */
        if (im==0 )
        {
         printf ("Bad image in GRAY_THRESHOLD\n");
         return 0;
        }
         for (i=0; i<=3; i++)
           tn[i] = 0;

           sum=0;
           value=0;
/* Apply all pixel of the input */
        for (i=1; i<im->info->nr; i++)
         for (j=1; j<im->info->nc; j++)
           {
            tn[0] = im->data[i-1][j];
            tn[1] = im->data[i+1][j];
            tn[2] = im->data[i][j-1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                tn[3] = im->data[i][j+1];
                s=0;
                a=0;
                b=0;
                a = (tn[0] * -1)+tn[1];
                b = (tn[2] * -1)+tn[3];
               if (a > b)
                s = a ;
               else s = b;

                for (g=0; g<=3; g++)
                 { tn[g] = 0;}
                 value=value+(im->data[i][j]*s);
                 sum=sum+s;
               }
                grammar = value/sum;

/* perform threshold */
          for (i=0; i<im->info->nr; i++)
           for (j=0; j<im->info->nc; j++)
           {
            if (im->data[i][j] > grammar)
                    im->data[i][j] = 1;
            else im->data[i][j] = 0;
           }
          for (j=0; j<im->info->nc; j++)
            {
            im->data[0][j] = 0;
            im->data[1][j] = 0;
            im->data[im->info->nr-1][j] = 0;
            im->data[im->info->nr-2][j] = 0;
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for (i=0; i<im->info->nr; i++)
          {
          im->data[i][0] = 0;
          im->data[i][1] = 0;
          im->data[i][im->info->nc-1] = 0;
          im->data[i][im->info->nc-2] = 0;
          }

        return 1;
}
/*-----------------------------------------*/
int text_label (IMAGE im)
{
        int i,j,k,label[8000];
        IMAGE tmp;
        int max;
/* Source image empty? */
        if (im==0)
        {
         printf ("Bad image in TEXT_LABELING\n");
         return 0;
        }
/* Create a result image */
        tmp = newimage (im->info->nr, im->info->nc);
        if (tmp == 0)
        {
         printf ("Out of memory in TEXT_LABELING.\n");
         return 0;
        }

        for (i=0; i<im->info->nr; i++)
         for (j=0; j<im->info->nc; j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            tmp->data[i][j] = 0;


/*Assign temporary labels to nonzero of the image  */
              max=0;
           for (i=0; i<im->info->nr; i++)
            for (j=0; j<im->info->nc; j++)
              if ( im->data[i][j] == 1)
                    {
                      max = max+1;
                      tmp->data[i][j] = max;
                      }

/*Propagate maximum label per connected component  */
   for (k=0; k<=10; k++)
    {
           for (i=(im->info->nr-1); i>=1; i--)
            for (j=0; j<im->info->nc; j++)
             if ((tmp->data[i][j] > 0) && (tmp->data[i-1][j] > 0))
             {
                 if (tmp->data[i][j] > tmp->data[i-1][j])
                    tmp->data[i-1][j] = tmp->data[i][j];
                 else tmp->data[i][j] = tmp->data[i-1][j];
             }

           for (i=0; i<im->info->nr; i++)
            for (j=(im->info->nc-1); j>=1; j--)
             if ((tmp->data[i][j] > 0) && (tmp->data[i][j-1] > 0))
             {
                 if (tmp->data[i][j] > tmp->data[i][j-1])
                    tmp->data[i][j-1] = tmp->data[i][j];
                 else tmp->data[i][j] = tmp->data[i][j-1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }

      for (i=0; i<(im->info->nr-1); i++)
       for (j=0; j<im->info->nc; j++)
        if ((tmp->data[i][j] > 0 ) && (tmp->data[i+1][j] > 0))
        {
            if (tmp->data[i][j] > tmp->data[i+1][j])
               tmp->data[i+1][j] = tmp->data[i][j];
            else tmp->data[i][j] = tmp->data[i+1][j];
        }

      for (i=0; i<im->info->nr; i++)
       for (j=0; j<(im->info->nc-1); j++)
        if ((tmp->data[i][j] > 0) && (tmp->data[i][j+1] > 0))
        {
            if (tmp->data[i][j] > tmp->data[i][j+1])
               tmp->data[i][j+1] = tmp->data[i][j];
            else tmp->data[i][j] = tmp->data[i][j+1];
        }
    }

/* Sequential relabeling */
      for (k=0; k<=max; k++)
        label[k] = 0;

        max = 0;
      for (i=0; i<im->info->nr; i++)
       for (j=0; j<im->info->nc; j++)
            if (im->data[i][j] == 1)
             {
             if (label[tmp->data[i][j]] == 0)
             {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                max = max+1;
                label[tmp->data[i][j]] = max;
                tmp->data[i][j] = max;
                }
               else tmp->data[i][j] = label[tmp->data[i][j]];
              }

/* Copy result over the input */
        for (i=0; i<im->info->nr; i++)
         for (j=0; j<im->info->nc; j++)
           im->data[i][j] = tmp->data[i][j];


/* Free the result image - it was a temp */
        freeimage(tmp);
        return 1;
}
/* ************************************************************ */
int region_analysis (IMAGE im,IMAGE im1)
{
        int p,k,s,i,j,max,t,x,y,w,z,a[200],b[200],c[200],d[200];
        int width[200],length[200],value[200];

/* Source image empty? */
        if ((im==0)&&(im1==0))
        {
         printf ("Bad image in REGEION ANALYSIS.\n");
         return 0;
        }

        max=0;
 /*  Fine max value */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for (i=0; i<im->info->nr; i++)
 for (j=0; j<im->info->nc; j++)
      if ( im->data[i][j] > max)
        max = im->data[i][j];

for (t=1; t<=max; t++)
  {
      a[t] = 0;
      b[t] = 0;
      c[t] = 0;
      d[t] = 0;
      width[t] = 0;
      length[t] = 0;
  }
      x=im->info->nr; y=im->info->nc;
      w=0; z=0;

/* fine coordinate of each text */
      for (t=1; t<=max; t++)
      {
          x=im->info->nr; y=im->info->nc;
          w=0; z=0;
          for (i=0; i<im->info->nr; i++)
           for (j=0; j<im->info->nc; j++)
           {
            if ( im->data[i][j] == t )
              {
              if ( x >= i)
                   a[t] = x = i;
              if ( y >= j)
                   b[t] = y = j;
              if ( w < i)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    c[t] = w = i;
                if ( z < j)
                    d[t] = z = j;
                }
              }
            }
/* condition of text analysis */
            p = 0;
           for (t=1; t<=max; t++)
           {
            width[t] = (c[t]+1)-a[t];
            length[t]= (d[t]+1)-b[t];
           if (width[t] > 0 && length[t] > 0)
            if (width[t]/length[t] >= 1/5 || length[t]/width[t] >= 1/5)
            if ((width[t] && length[t]) <= im->info->nr/3)
            if (width[t]  > im->info->nr/8)
              {
              p = p+1;
              value[p] = t;
              }
           }

           for (t=1; t<=p; t++)
            {
             a[t] = a[value[t]];
             b[t] = b[value[t]];
             c[t] = c[value[t]];
             d[t] = d[value[t]];
            }

 /* set image to white image   */
          for (i=0; i<im->info->nr; i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            for (j=0; j<im->info->nc; j++)
              im->data[i][j] = 255;

/* Copy only Text from original image */

            for (t=1; t<=p; t++)
              for (k=a[t]; k<=c[t]; k++)
              for (s=b[t]; s<=d[t]; s++)
                    im->data[k][s] = im1->data[k][s];

            return 1;
}

/* ========================================================== */
int read_pbm (char *fn, IMAGE *im)
{
            int i,j,k,n,m,bi, b;
            unsigned char ucval;
            int val;
            char buf1[256];
            IMAGE image;

            FILE *f;

            f = fopen (fn, "r");
            if (f==NULL)
            {
             printf ("Can't open the PBM file named '%s'\n", fn);
             return 0;
            }

            pbm_getln (f, buf1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (buf1[0] == 'P')
        {
         switch (buf1[1])
         {
case '1':     k=1; break;
case '2':     k=2; break;
case '3':     k=3; break;
case '4':     k=4; break;
case '5':     k=5; break;
case '6':     k=6; break;
default:      printf ("Not a PBM/PGM/PPM file.\n");
                  return 0;
          }
        }
        bi = 2;

        get_num_pbm (f, buf1, &bi, &m);       /* Number of columns */
        get_num_pbm (f, buf1, &bi, &n);       /* Number of rows */
        if (k!=1 && k!=4) get_num_pbm (f, buf1, &bi, &b);     /* Max value */
        else b = 1;
/* Allocate the image */
        if (k==3 || k==6)     /* Colour */
        {
                printf ("Colour - later\n\n");
                exit (1);
        } else
        {
         *im = (IMAGE)newimage (n, m);
         for (i=0; i<n; i++)
          for (j=0; j<m; j++)
           if (k<3)
           {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                fscanf (f, "%d", &val);
                (*im)->data[i][j] = (unsigned char)val;

            } else {
                fscanf (f, "%c", &ucval);
                (*im)->data[i][j] = ucval;

            }
        }
        return 1;
}

void get_num_pbm (FILE *f, char *b, int *bi, int *res)
{
        int i;
        char str[80];

        while (b[*bi]==' ' || b[*bi]=='\t' || b[*bi]=='\n')
        {
         if (b[*bi] == '\n')
         {
          pbm_getln (f, b);
          *bi = 0;
         } else
         *bi += 1;
        }
        i = 0;
        while (b[*bi]>='0' && b[*bi]<='9')
         str[i++] = b[(*bi)++];
        str[i] = '\0';
        sscanf (str, "%d", res);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

/* Get the next non-comment line from the PBM file f into the
  buffer b. Look for 'pragmas' - commands hidden in the comments */

void pbm_getln (FILE *f, char *b)
{
        int i;
        char c;

/* Read the next significant line (non-comment) from f into buffer b */
        do
        {

/* Read the next line */
          i = 0;
          do
          {
           fscanf (f, "%c", &c);
           b[i++] = c;
           if (c == '\n') b[i] = '\0';
          } while (c != '\n');

/* If a comment, look for a special parameter */
           if (b[0] == '#') pbm_param (b);
        } while (b[0]=='\n' || b[0] == '#');
}

/*     Look for a parameter hidden in a comment       */
void pbm_param (char *s)
{
        int i,j;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        char key[24];

/* Extract the key word */
        for (i=0; i<23; i++)
        {
         j = i;
         if (s[i+1] == ' ' || s[i+1] == '\n') break;
         key[i] = s[i+1];

        }
        key[j] = '\0';

/* Convert to lower case */
        for (i=0; i<j; i++)
         if ( (key[i]>='A') && (key[i]<='Z') )
              key[i] = (char) ( (int)key[i] - (int)'A' + (int)'a' );

/* Which key word is it? */
        if (strcmp(key, "origin") == 0)        /* ORIGIN key word */
        {
         sscanf (&(s[j+1]), "%d %d",
           &PBM_SE_ORIGIN_COL, &PBM_SE_ORIGIN_ROW);

         return;
        }
}
/*-------------------------------------------------*/
int read_pgm (char *fn, IMAGE *im)
{
        int i,j,k,n,m,bi, b;
        unsigned char ucval;
        int val;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char buf1[256];
IMAGE image;

FILE *f;

f = fopen (fn, "r");
if (f==NULL)
{
 printf ("Can't open the PGM file named '%s'\n", fn);
 return 0;
}

pgm_getln (f, buf1);
if (buf1[0] == 'P')
{
 switch (buf1[1])
 {
case '1':    k=1; break;
case '2':    k=2; break;
case '3':    k=3; break;
case '4':    k=4; break;
case '5':    k=5; break;
case '6':    k=6; break;
default:     printf ("Not a PBM/PGM/PPM file.\n");
             return 0;
  }
 }
 bi = 2;

 get_num_pgm (f, buf1, &bi, &m);      /* Number of columns */
 get_num_pgm (f, buf1, &bi, &n);      /* Number of rows */
 if (k!=1 && k!=4) get_num_pgm (f, buf1, &bi, &b);     /* Max value */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else b = 1;


/* Allocate the image */
        if (k==3 || k==6)      /* Colour */
        {
                printf ("Colour - later\n\n");
                exit (1);
        } else
        {
         *im = (IMAGE)newimage (n, m);
         for (i=0; i<n; i++)
           for (j=0; j<m; j++)
            if (k<3)
            {
                fscanf (f, "%d", &val);
                (*im)->data[i][j] = (unsigned char)val;

            } else {
                fscanf (f, "%c", &ucval);
                (*im)->data[i][j] = ucval;

            }
        }

        return 1;
}

void write_pgm (char *filename, IMAGE image)
{
        FILE *f;
        int i,j,k;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        f = fopen (filename, "w");
        fprintf (f,"P2\n#CREATOR:XV Version 3.00 REV:3/30/93\n%d %d\n%d\n", image->
info->nc, image->info->nr,255);
        k=0;
        for (i=0; i<image->info->nr; i++)
         for (j=0; j<image->info->nc; j++)
         {
                fprintf (f, "%d ", image->data[i][j]);
                k++;
                if (k > 30)
                {
                 fprintf (f, "\n");
                 k = 0;
                }
         }
        fprintf (f, "\n");
        fclose (f);
}

void get_num_pgm (FILE *f, char *b, int *bi, int *res)
{
        int i;
        char str[80];

        while (b[*bi]==' ' || b[*bi]=='\t' || b[*bi]=='\n')
        {
         if (b[*bi] == '\n')
         {
          pgm_getln (f, b);
          *bi = 0;
         } else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        *bi += 1;
      }
      i = 0;
      while (b[*bi]>='0' && b[*bi]<='9')
       str[i++] = b[(*bi)++];
      str[i] = '\0';
      sscanf (str, "%d", res);

}

void pgm_getln (FILE *f, char *b)
{
      int i;
      char c;

/* Read the next significant line (non-comment) from f into buffer b */
      do
      {

/* Read the next line */
       i = 0;
       do
       {
        fscanf (f, "%c", &c);
        b[i++] = c;
        if (c == '\n') b[i] = '\0';
       } while (c != '\n');

/* If a comment, look for a special parameter */
       if (b[0] == '#') pgm_param (b);
      } while (b[0]=='\n' || b[0] == '#');
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Look for a parameter hidden in a comment  */
void pgm_param (char *s)
{
        int i,j;
        char key[24];

/* Extract the key word */
        for (i=0; i<23; i++)
        {
         j = i;
         if (s[i+1] == ' ' || s[i+1] == '\n') break;
         key[i] = s[i+1];

        }
        key[j] = '\0';

/* Convert to lower case */
        for (i=0; i<j; i++)
         if ( (key[i]>='A') && (key[i]<='Z') )
                key[i] = (char) ( (int)key[i] - (int)'A' + (int)'a' );

/* Which key word is it? */
        if (strcmp(key, "origin") == 0)       /* ORIGIN key word */
        {
         sscanf (&(s[j+1]), "%d %d",
           &PGM_ORIGIN_COL, &PGM_ORIGIN_ROW);

         return;
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
struct image  *newimage (int nr, int nc)
{
        struct image  *x;           /* New image */
        unsigned char *ptr;         /* new pixel array */
        int i;

        if (nr < 0 || nc < 0) {
                printf ("Error: Bad image size (%d,%d)\n", nr, nc);
                return 0;
        }

/*      Allocate the image structure   */
        x = (struct image  *) malloc( sizeof (struct image) );
        if (!x) {
                printf ("Out of storage in NEWIMAGE.\n");
                return 0;
        }

/*      Allocate and initialize the header     */

        x->info = (struct header *)malloc( sizeof(struct header) );
        if (!(x->info)) {
                printf ("Out of storage in NEWIMAGE.\n");
                return 0;
        }
        x->info->nr = nr;       x->info->nc = nc;

/*      Allocate the pixel array       */

        x->data = (unsigned char **)malloc(sizeof(unsigned char *)*nr); /* Pointers to rows */
        if (!(x->data)) {
                printf ("Out of storage in NEWIMAGE.\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                return 0;
        }

        for (i=0; i<nr; i++) {
         ptr = (unsigned char *)malloc(nc); /* Allocate one row  */
         if (!ptr) {
                printf ("Out of storage in NEWIMAGE.\n");
                return 0;
         } else x->data[i] = ptr;
        }
        return x;
}

void freeimage (struct image  *z)
{
/*    Free the storage associated with the image Z    */
        int i;

        if (z != 0) {
          for (i=0; i<z->info->nr; i++)
            free (z->data[i]);
          free (z->info);
          free (z->data);
          free (z);
        }
}
/*=============================================================*/
void main (int argc, char *argv[])
{
        IMAGE im,im1;
        SE p;
        int k;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        argv[3]="se4145.pbm";
        argv[4]="se4146.pbm";

        if (argc < 2)
        {
         printf ("Can't perform extraction");
         exit(1);
        }
    printf ("\nPerform an extraction on image '%s'.\n",argv[1]);
/* Read the input image */
        if (read_pgm (argv[1], &im) == 0) exit(2);
        if (read_pgm (argv[1], &im1) == 0) exit(3);
/* Perform filter image */
    gray_filter(im,im1);
    freeimage(im1);
/* Perform gradient image */
    gray_gradient(im,im1);
/* Perform the gray threshold */
    gray_threshold(im);
/* Read the structuring element */
    k = get_se (argv[3], &p);
        if (!k)
        {
        printf ("Bad structuring element, file name '%s'.\n", argv[3]);
         exit (4);
        }
     printf ("Structuring element is:\n");
     print_se (p);
/* Perform the dilation */
        bin_dilate (im, p);
/* Read the structuring element */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        k = get_se (argv[4], &p);
        if (!k)
        {
        printf ("Bad structuring element, file name '%s'.\n", argv[4]);
         exit (5);
        }
        printf ("Structuring element is:\n");
        print_se (p);
/* Perform the erosion */
/*      bin_erode (im, p);  */
/* Perform the text_label */
        text_label(im);
/* Read original image */
        if (read_pgm (argv[1], &im1) == 0) exit(6);
/* Perform the region analysis */
        region_analysis(im,im1);
/* Write the result to the specified file */
        write_pgm (argv[2], im);
        freeimage(im);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้