เทคนิคการหาขอบภาพโดยใช้หน้าต่าง

IMAGE EDGE FINDING WITH WINDOWS

ไพศาล สุธีบรรเจิด

PHAISARN SUTHEBANJARD

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2548

ISBN 974-13-1995-8

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เทคนิคการหาขอบภาพโดยใช้หน้าต่าง

IMAGE EDGE FINDING WITH WINDOWS

T060906

ไพศาล สุธีบรรเจิด

PHAISARN SUTHEEBANJARD

เลขหมู่.....
เลขทะเบียน 60906
วัน,เดือน,ปี 6 ก.ค. 2549

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2548

ISBN 974-15-1995-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMAGE EDGE FINDING WITH WINDOWS

PHAISARN SUTHEEBANJARD

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2005

ISBN 974-15-1995-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2005

SCHOOL OF GRADUATE STUDIES

KINGMONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	เทคนิคการหาขอบภาพโดยใช้หน้าต่าง
นักศึกษา	นาย ไพศาล สุธีบรรเจิด
รหัสประจำตัว	45066047
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2548
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. อาริต ธรรมโน
อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม	รศ. ดร. วิเชียร เปรมชัยสวัสดิ์

บทคัดย่อ

ระบบแปลงไฟล์ภาพเอกสารเป็นไฟล์ข้อความอัตโนมัติ (OCR System) มีกระบวนการที่สำคัญคือ การแบ่งส่วนประกอบภาพเอกสาร ดังนั้นวิทยานิพนธ์ฉบับนี้จึงได้นำเสนออัลกอริทึมที่ใช้ในการแบ่งส่วนประกอบภาพเอกสาร ซึ่งวิธีที่นำเสนอนี้สามารถทำการประมวลผลได้อย่างรวดเร็วด้วยการใช้หน้าต่างในการตรวจสอบหาจุดดำ และเดินรอบขอบของกลุ่มดำ เพื่อนำขอบของตัวอักษรมาสร้างรูปขนาดย่อ (Optimum picture) โดยนำหน้าต่าง 12x12 พิกเซล ของขนาดจริงมาใช้กำหนด 1 พิกเซลใน Optimum picture ขอบเขตของบล็อกตัวอักษรหาได้จาก Optimum picture ด้วยวิธี Dilation1PX ผลการทดลองแสดงให้เห็นถึงวิธีที่นำเสนอสามารถทำการแบ่งส่วนประกอบภาพเอกสารได้รวดเร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Image Edge Finding with Windows
Student	Mr. Phaisarn Sutheebanjard
Student ID.	45066047
Degree	Master of Science
Programme	Information Technology
Year	2005
Thesis Advisor	Assoc. Prof. Dr. Arit Thanmano
Thesis Co-Advisor	Assoc. Prof. Dr. Wichian Premchaiswadi

ABSTRACT

Document page segmentation is one of the most important processes in converting image in to alphabet in an OCR system. The process is necessary for an automatic OCR system. The paper proposes an algorithm to speed up the processing time of document page segmentation. Window is used to find black pixel and contour border. Then, the optimum picture is created from these borders of characters. The 12x12 pixels window of original picture is defined as 1 pixel in optimum picture. The border of each character block is extracted from optimum picture with Dilation1PX technique. The experimental results show that the propose scheme can speed up the processing time of the document page segmentation significantly.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เพราะได้รับโอกาส คำแนะนำและปรึกษา จากอาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร. วิเชียร เปรมชัยสวัสดิ์ และ รศ. ดร. อาริต ธรรมโน ซึ่งได้ให้คำปรึกษาในการแก้ไขปัญหาต่างๆ และชี้แนวทางในการทำวิทยานิพนธ์ตลอดมา

ขอบคุณอาจารย์ทุกท่านในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้สั่งสอนวิชาความรู้ต่างๆ ซึ่งเป็นรากฐานที่ทำให้งานวิทยานิพนธ์นี้สำเร็จด้วยดี โดยเฉพาะอาจารย์ธนารัตน์ ชลิดาพงศ์ ที่ได้สอนความรู้เกี่ยวกับ Digital Image Processing ซึ่งจำเป็นมากสำหรับการทำวิทยานิพนธ์นี้

ขอบคุณเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศทุกท่าน ที่ได้อำนวยความสะดวกด้านเอกสารและติดต่อประสานงานต่างๆ ซึ่งทำให้การทำวิทยานิพนธ์นี้ ดำเนินงานไปได้โดยสะดวก

ขอบคุณบิดา-มารดา ที่สนับสนุนในการศึกษาหาวิชาความรู้ และเป็นกำลังใจให้สามารถทำงานวิทยานิพนธ์นี้จนสำเร็จ

สุดท้ายนี้ ขอบคุณทุกท่านที่ไม่ได้กล่าวถึงข้างต้น ที่เคยช่วยเหลือกัน

ไพศาล สุธีบรรเจิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมา และความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 3

1.3 สมมุติฐานทางการศึกษา 3

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในงานวิจัย 3

1.5 ขอบเขตของการศึกษา 4

1.6 ขั้นตอนการศึกษา 5

1.7 คำจำกัดความที่ใช้ในการศึกษา 5

บทที่ 2 ทฤษฎีและแนวคิดที่เกี่ยวข้อง 6

2.1 คุณลักษณะทั่วไปของภาพเอกสาร 6

2.2 ขั้นตอนการประมวลผลภาพเอกสาร 8

2.3 ความรู้พื้นฐานเกี่ยวกับข้อมูลภาพอิเล็คทรอนิกส์ 9

2.4 วิธีการที่นำมาใช้ในการแบ่งส่วนประกอบภาพเอกสาร 18

2.5 การแบ่งส่วนประกอบภาพเอกสาร 19

บทที่ 3 การหาจุดเริ่มต้นของการประมวลผลด้วยการใช้หน้าต่าง 22

3.1 แนวคิดในการตรวจสอบหาจุดเริ่มต้นของการประมวลผลด้วยการใช้หน้าต่าง.. 22

3.2 เปรียบเทียบการหาจุดดำจุดแรกด้วยการตรวจสอบทุกจุด และการใช้หน้าต่างแบบต่างๆ 24

3.3 เปรียบเทียบการหาจุดดำจุดแรกด้วยการตรวจสอบทุกจุด และการใช้หน้าต่างแบบ I 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.4 ความสัมพันธ์ของการขยายขอบภาพให้ใหญ่ขึ้น กับการหาจุดดำจุดแรก ด้วยการตรวจสอบทุกจุด และการใช้หน้าต่างแบบ I 30

บทที่ 4 แนวทางที่ใช้ในงานวิจัย 33

4.1 แนวคิดการแบ่งส่วนประกอบภาพเอกสาร 33

4.2 การหาขอบเขตของตัวอักษร และรูป 34

4.3 การนำ Optimum Picture มาใช้ในการหาขอบเขตของบล็อกตัวอักษร 35

4.4 การตัดแบ่งคอลัมน์ 39

4.5 การรวมบล็อกที่ Overlap กัน 39

4.6 การหาระดับบรรทัด 40

บทที่ 5 ผลการทดลอง 42

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ 44

6.1 สรุปผลการวิจัย 44

6.2 ข้อเสนอแนะ 46

เอกสารอ้างอิง 47

ภาคผนวก ก บทความที่ตีพิมพ์ในวารสาร 48

ภาคผนวก ข อัลกอริทึมและขั้นตอนการทำงาน 55

ภาคผนวก ค ภาพเอกสารที่ใช้ในการทดลอง 59

ประวัติผู้เขียน 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 คุณสมบัติต่างๆ ของค่าที่เลือกมาเป็นเธรดโซลด์ 13

3.1 คุณสมบัติของหน้าต่างแบบ A-K 24

3.2 เปรียบเทียบเวลาการทำงาน วิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ A-K 25

3.3 เปรียบเทียบเวลาการทำงาน วิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ I 28

3.4 เวลาของการทดลองการขยายขอบภาพ 31

5.1 ผลการทดลองด้วยวิธี Dilation1PX 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
1.1 วิธีการตรวจสอบหาจุดดำภายในหน้ากากของวิธี Page Segmentation and Classification Based on Pattern-list Analysis	2
1.2 ระยะห่างระหว่างตัวอักษร สระบน สระล่าง	2
2.1 การจัดเก็บส่วนประกอบที่เป็นตัวอักษร	7
2.2 การจัดเก็บขอบเขตส่วนประกอบที่เป็นรูปภาพ	8
2.3 ระบบแกนและพิกัด	9
2.4 แสดงฮิสโตแกรมของรูปเกรย์สเกล	10
2.5 แสดงฮิสโตแกรมของจุดดำ	11
2.6 แสดงการทำฮิสโตแกรมด้วยขอบของส่วนประกอบย่อย	12
2.7 ค่าต่างๆที่นำมาใช้เป็นค่าเธรดโชลด์	13
2.8 แปลงภาพเกรย์สเกล เป็นภาพไบนารี่	15
2.9 จุดที่มีความสัมพันธ์แบบ N4(p), ND(p) และ N8(p)	16
2.10 การ Label	16
2.11 การเดินรอบขอบด้วย Chain Codes	17
2.12 การทำ Dilation ด้วย Mask ขนาด 3x3	18
2.13 เปรียบเทียบการทำ Dilation ด้วย Mask ขนาดต่างๆ	19
3.1 หน้าต่างแบบ A-K	23
3.2 รูปที่ใช้ในการทดลองจับเวลา	26
3.3 สระไม้เอก มีขนาด 2x9 พิกเซล และ 2x7 พิกเซล ทำให้หน้าต่าง A–F ทำงานผิดพลาดได้	27
3.4 กราฟแสดงเวลาการทำงานของวิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ I	30
3.5 ตัวอย่างภาพที่ใช้ทดสอบ	31
3.6 กราฟเวลาการทำงานในช่วงที่ 1	32
4.1 รูปขนาดย่อ (Optimum Picture)	33
4.2 ขอบเขตของรูปที่หาได้โดยการใช้หน้าต่างแบบ I	34
4.3 Mask 3x3 พิกเซล ใช้ในการทำ Dilation	35
4.4 Optimum Picture ที่ใช้ในการหาขอบเขตของบล็อกตัวอักษรด้วยวิธี Dilation1PX	35
4.5 ความสามารถในการแยกบล็อกออกจากกันของ วิธี Dilation1PX	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.6 ความสามารถในการแยกบล็อกออกจากกันของ หน้าต่าง 32x32 พิกเซล	36
4.7 ตัวอย่างการวางหน้าต่างที่สามารถแยกคอลัมน์ได้ และไม่ได้	37
4.8 ตัวอย่างวิธีต่างๆที่สามารถแยกคอลัมน์ได้ และไม่ได้	38
4.9 ทั้ง 2 วิธีไม่สามารถแยกบล็อกออกจากกันได้เมื่อบล็อกใกล้กันมากๆ	38
4.10 ทำฮิสโตแกรมด้วยขอบของตัวอักษร	39
4.11 การรวมบล็อกที่ Overlap กัน	40
4.12 เปรียบเทียบการทำฮิสโตแกรม	41
6.1 เอกสารที่มีการจัดวางคอลัมน์อย่างสลับซับซ้อน	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การจัดเก็บข้อมูล วิธีที่นิยมใช้กันมากคือ การเก็บข้อมูลลงบนกระดาษ แต่เนื่องจากปริมาณของข้อมูลที่เพิ่มจำนวนมากขึ้นอย่างรวดเร็ว ทำให้เกิดปัญหาต่างๆตามมามากมาย เช่น การจัดเก็บและค้นคืนข้อมูลทำได้ยากและใช้เวลามาก, ข้อมูลไม่สามารถใช้ร่วมกันได้ เช่น หนังสือ 1 เล่ม สามารถอ่านได้ครั้งละ 1 คน, ข้อมูลไม่สามารถใช้ได้ตลอดเวลา เช่น ห้องสมุดจะมีเวลาเปิด-ปิดทำการ, เปลืองพื้นที่และกำลังคนในการจัดเก็บและดูแลรักษาข้อมูล

จากปัญหาต่างๆของการจัดเก็บข้อมูลด้วยกระดาษ ทำให้ปัจจุบันเริ่มมีความนิยมในการจัดเก็บข้อมูลบนสื่ออิเล็คทรอนิกส์ เพราะการจัดเก็บข้อมูลบนสื่ออิเล็คทรอนิกส์ สามารถทำการจัดเก็บและค้นคืนข้อมูลได้อย่างสะดวกรวดเร็ว, ข้อมูลสามารถใช้ร่วมกันได้, ลดพื้นที่ในการจัดเก็บข้อมูล, การทำสำรองข้อมูล (Backup) ทำได้สะดวก

ขั้นตอนการนำข้อมูลที่จัดเก็บบนกระดาษ มาจัดเก็บบนสื่ออิเล็คทรอนิกส์ เริ่มจากการสแกนเอกสารเพื่อให้ได้ไฟล์ภาพเอกสารไบนารี่ แล้วนำไฟล์ภาพเอกสารไบนารี่ที่ได้ มาผ่านกระบวนการแปลงไฟล์ภาพเอกสารไบนารี่เป็นไฟล์ข้อความ เพื่อทำการจัดเก็บต่อไป โดยกระบวนการแปลงไฟล์ภาพเอกสารไบนารี่เป็นไฟล์ข้อความนี้ แบ่งการทำงานได้เป็น 2 ขั้นตอน คือ

1. การแบ่งส่วนประกอบภาพเอกสารไบนารี่ (Document Page Segmentation)
2. การแปลงภาพอักษรเป็นข้อความ (Optical Character Recognition - OCR)

กระบวนการแปลงไฟล์ภาพเอกสารไบนารี่เป็นไฟล์ข้อความนี้ ทำให้การทำสำเนาข้อมูลที่จัดเก็บในกระดาษ มาจัดเก็บลงบนสื่ออิเล็คทรอนิกส์ทำได้สะดวกและรวดเร็วขึ้น แต่ถ้าข้อมูลมีปริมาณมาก กระบวนการนี้ก็จะใช้เวลาในการทำงานมาก ดังนั้นจึงมีผู้เสนออัลกอริทึมที่ทำให้กระบวนการนี้ทำงานได้รวดเร็วขึ้น เช่น Page Segmentation and Classification Based on Pattern-list Analysis[4] วิธีการนี้ใช้หน้ากาก 3x3 แล้วตรวจสอบหาจุดดำภายในหน้ากากในทิศทาง right-left และ up-down ดังรูปที่ 1.1 โดยถ้ามีจุดดำอย่างน้อย 1 จุดใน 9 จุด ก็จะถือว่าพบข้อมูล จากนั้นหาขอบขวา และขอบล่างของกลุ่มจุดดำนี้

วิธีที่ได้กล่าวถึงนี้ ทำการตรวจสอบหาจุดดำด้วยการตรวจสอบครั้งละ 3x3 พิกเซล ถ้าพบจุดดำอย่างน้อย 1 จุด จะถือว่ามีข้อมูล แต่ถ้าภายใน 3x3 พิกเซล ไม่มีจุดดำ วิธีนี้ก็ต้องตรวจสอบทุกจุดภายในหน้ากาก 3x3 พิกเซลนี้ ดังนั้นในขั้นตอนการตรวจสอบหาจุดดำจุดแรก ขณะที่ยังไม่เจอจุดดำ วิธีนี้ต้องทำการตรวจสอบทุกจุด ดังนั้นจึงไม่ต่างกับวิธีการตรวจสอบทุกจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	4	1
8	5	2
9	6	3

(ก)

7	8	9
4	5	6
1	2	3

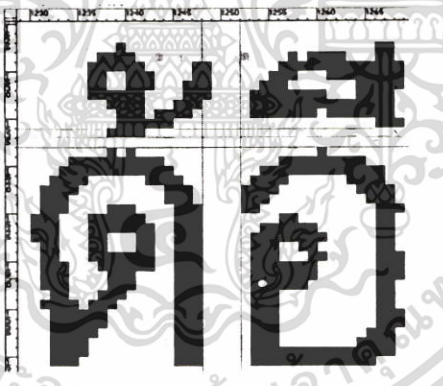
(ข)

รูปที่ 1.1 วิธีการตรวจสอบหาจุดดำภายในหน้ากากของวิธี Page Segmentation and Classification Based on Pattern-list Analysis[4]

(ก) ใช้ในการตรวจสอบหาจุดดำจุดแรก และขอบทางขวา

(ข) ใช้ในการตรวจสอบหาขอบทางด้านล่าง

แต่เมื่อเจอจุดดำจุดแรกแล้ว วิธีนี้จะทำการหาขอบทางขวา และขอบล่างของตัวอักษรโดยการตรวจสอบหาขอบสามารถทำได้อย่างรวดเร็วเพราะไม่ต้องตรวจสอบทุกจุด ดังนั้นจึงหาขอบเขตแบบสี่เหลี่ยมของตัวอักษรได้รวดเร็วกว่า วิธีการเดินรอบขอบด้วย Chain Codes ครั้งละ 1 พิกเซล แต่วิธีนี้ไม่สามารถนำมาใช้กับภาพอักษรภาษาไทยได้ เนื่องจากภาษาไทยไม่ได้มีเพียงตัวอักษรแต่มีสระบน สระล่างด้วย ซึ่งสระบน สระล่างนี้ บางครั้งมีระยะห่างกับตัวอักษรเพียง 1-2 พิกเซลเท่านั้น ดังรูปที่ 1.2 จึงทำให้ตัวอักษรที่อยู่ใกล้กัน รวมกันเป็นขอบเขตเดียวกันได้



(ก)

(ข)

รูปที่ 1.2 ระยะห่างระหว่างตัวอักษร สระบน สระล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ ยังมีผู้นำเสนอวิธีการแบ่งส่วนประกอบภาพเอกสารอย่างรวดเร็วด้วยการใช้ window map[3] คือเป็นการลดขนาดของภาพเอกสารลง แล้วนำภาพเอกสารที่เล็กลงนี้มาใช้ในการหาขอบเขตแบบสี่เหลี่ยมของบล็อกตัวอักษร ก็จะทำให้สามารถหาขอบเขตแบบสี่เหลี่ยมของบล็อกตัวอักษรได้อย่างรวดเร็วเพราะเป็นการกระทำกับจำนวนข้อมูลที่น้อยลงมาก

วิธีการดังกล่าวนี้ ได้ทำการลดขนาดของภาพลง ด้วยการใช้ Feature Extraction เช่น Black/White-ratio คือการนำจุดดำภายใน window มาเทียบอัตราส่วนกับจำนวนจุดทั้งหมดใน window วิธีการนี้จึงจำเป็นต้องตรวจสอบทุกจุดภายในหน้าต่าง ซึ่งทำให้ต้องประมวลผลนาน

ดังนั้น วิทยานิพนธ์นี้จึงนำเสนอวิธีการตรวจสอบหาจุดดำอย่างถูกต้องและรวดเร็วด้วยการใช้หน้าต่าง แล้วตรวจสอบเพียงบางตำแหน่งภายในหน้าต่าง ทำให้สามารถตรวจสอบหาจุดดำได้อย่างถูกต้องและรวดเร็ว และนำจุดดำที่ได้มาสร้างเป็นรูปขนาดย่อ (Optimum Picture) ซึ่งมีขนาดเล็กกว่ารูปต้นฉบับมาก จากนั้นจะนำ Optimum Picture มาใช้ในการหาขอบเขตของบล็อกตัวอักษรแบบรูปร่างไม่แน่นอน ทำให้การแบ่งส่วนประกอบภาพเอกสารประมวลผลได้อย่างรวดเร็ว

1.2 ความมุ่งหมายและวัตถุประสงค์ ของการศึกษา

1. เพื่อพัฒนาอัลกอริทึมที่สามารถแบ่งส่วนประกอบภาพเอกสารไบนารี่ ได้อย่างถูกต้องและรวดเร็ว โดยแบ่งส่วนประกอบออกเป็นส่วนประกอบที่เป็นรูปภาพ และส่วนประกอบที่เป็นกลุ่มของตัวอักษร (Block)

2. สามารถจัดเก็บขอบเขตของรูปภาพ และบล็อกของตัวอักษร ในลักษณะที่ไม่เป็นสี่เหลี่ยมได้ ทำให้ข้อมูลที่ได้มีความถูกต้อง

1.3 สมมุติฐานทางการศึกษา

วิทยานิพนธ์นี้ นำเสนอวิธีการแบ่งส่วนประกอบภาพเอกสารไบนารี่ โดยผลการทดลอง ที่ได้ทำกับภาพเอกสารภาษาไทย ภาษาอังกฤษ ในรูปแบบวารสารทางวิชาการ และนิตยสาร ผลที่ได้จากการทดลองเป็นขอบเขตของส่วนประกอบ ที่ถูกแบ่งประเภทเป็นรูปภาพ หรือบล็อกของตัวอักษร โดยกระบวนการทำงานทั้งหมดสามารถประมวลผลได้รวดเร็วกว่าวิธีที่มีอยู่เดิม

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในงานวิจัย

การแบ่งส่วนประกอบภาพเอกสารไบนารี่ อันดับแรกคือทำการตรวจสอบหาจุดเริ่มต้นในการเริ่มประมวลผลโดยการหาจุดดำจุดแรกก่อน โดยทั่วไปจะใช้วิธีการตรวจสอบทุกจุดด้วยการสแกนแบบราสเตอร์ (Raster Scan) คือการตรวจสอบทีละจุดจากซ้ายไปขวา จากบนลงล่าง ซึ่งวิธีนี้ต้องทำการตรวจสอบทุกจุดจนกว่าจะเจอจุดดำ จึงทำให้ใช้เวลาในการประมวลผลมาก ดังนั้นงานวิจัยนี้ จึงนำเสนอวิธีการใช้หน้าต่างขนาด 12x12พิกเซล แล้วตรวจสอบเพียงบางตำแหน่งใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง ทำให้การตรวจสอบหาจุดดำจุดแรกเป็นไปอย่างรวดเร็ว เมื่อได้จุดเริ่มต้นของข้อมูลแล้ว จากนั้นจะทำการกำหนดขอบเขตของส่วนประกอบ ซึ่งมีหลายวิธีการด้วยกัน ดังนี้

1. วิธีการแบบบนลงล่าง (Top-Down Approach) เช่น วิธี Recursive X-Y Cut [2]

2. วิธีแบบล่างขึ้นบน (Bottom-Up Approach) เช่น การรวมส่วนประกอบเล็กที่มีความสัมพันธ์กันมาเป็นส่วนประกอบที่ใหญ่ขึ้น [4]

3. วิธีการแบบผสม (Hybrid Approach) เช่น A Fast and Efficient Method for Document Segmentation for OCR [1] เป็นการใช้หน้าต่างขนาด 32x32 พิกเซลในการกำหนดขอบเขตของกลุ่มตัวอักษร และรูปภาพ

แนวคิดที่ใช้ในการวิจัยคือ การแบ่งส่วนประกอบของเอกสารนั้นปัจจุบันมีวิธีการที่ดีที่สามารถแบ่งส่วนประกอบของเอกสารไบนารี่ได้อย่างถูกต้อง แต่วิธีการต่าง ๆ ดังกล่าวใช้เวลาในการประมวลผลมาก จึงได้เกิดแนวความคิดในการที่จะทำให้การแบ่งส่วนประกอบของเอกสารเป็นไปอย่างรวดเร็วขึ้น โดยการนำหน้าต่างขนาด 12x12 พิกเซล มาใช้ตรวจสอบหาจุดดำ เพื่อนำไปสร้างรูปขนาดย่อ (Optimum Picture) ที่มีขนาดเล็กว่ารูปต้นฉบับประมาณ 144 เท่า จากนั้นนำรูปขนาดย่อนี้ มาใช้ในการตรวจสอบหาขอบเขตของบล็อกตัวอักษร ซึ่งจะทำให้การประมวลผลใช้เวลาลดน้อยลง

1.5 ขอบเขตของการศึกษา

1. ศึกษาและเปรียบเทียบอัลกอริทึม ที่ใช้ในการแบ่งส่วนประกอบภาพเอกสารไบนารี่ (Document Page Segmentation) หลาย ๆ วิธีการ และได้นำวิธีการ A Fast and Efficient Method for Document Segmentation for OCR[1] มาเปรียบเทียบกับวิธี Dilation1PX ซึ่งเป็นวิธีการที่ผู้ทดลองนำมาจัดทำเป็นวิทยานิพนธ์

2. คิดค้น พัฒนา และทดลองอัลกอริทึม ในการแบ่งส่วนประกอบภาพเอกสารไบนารี่ ให้มีความถูกต้อง และรวดเร็วขึ้น

3. ภาพเอกสารไบนารี่ที่ใช้ในการทดลอง มีลักษณะดังนี้
   - ภาพเอกสารภาษาไทย และภาษาอังกฤษ
   - ภาพเอกสารไม่เอียง และไม่มีกรอบ
   - ตัวอักษรในภาพเอกสารมีขนาดไม่เล็กกว่า 14 และไม่ใหญ่กว่า 24 และทำการสแกนด้วยความละเอียด 300 จุดต่อนิ้ว (300dpi) ซึ่งจะทำให้ได้ตัวอักษรที่มีขนาดความกว้างและความสูงไม่น้อยกว่า 5 พิกเซล และไม่เกิน 80 พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ขั้นตอนการศึกษา

1. ศึกษาบทความที่เกี่ยวข้องกับการแบ่งส่วนประกอบภาพเอกสารไบนารี่ เพื่อก่อให้เกิดแนวทางในการคิดค้น และพัฒนาวิธีการแบ่งส่วนประกอบภาพเอกสารไบนารี่ ได้อย่างถูกต้องและรวดเร็วขึ้น

2. นำวิธีที่ศึกษามาทำการทดลองกับภาพเอกสารไบนารี่ เพื่อให้เข้าใจในวิธีการทำงานให้มากขึ้น

3. วิเคราะห์ข้อดี และข้อเสียของงานวิจัยที่ได้ทำการศึกษามา

4. ออกแบบ และพัฒนาอัลกอริทึม ที่ใช้ในการแบ่งส่วนประกอบภาพเอกสารไบนารี่

5. ทำการทดลอง วิเคราะห์ผลการทดลอง และสรุปผลการทดลอง

6. จัดทำวิทยานิพนธ์

1.7 คำจำกัดความที่ใช้ในการศึกษา

หน้าต่าง 32x32 คือวิธีการที่ใช้ในการเดินรอบขอบของกลุ่มข้อมูล เพื่อหาขอบเขตของกลุ่มข้อมูล (ตัวอักษรและรูปภาพ) ด้วยวิธีการเดินแบบ Chain Codes แล้วตรวจสอบภายในหน้าต่าง ถ้าพบจุดดำภายในหน้าต่างมากกว่า 10 จุด ให้กำหนดว่าหน้าต่างนี้มีข้อมูล

หน้าต่าง 12x12 คือวิธีการที่ใช้ในการตรวจสอบหาจุดดำ โดยการตรวจสอบในตำแหน่งที่ได้กำหนดไว้ โดยไม่ต้องตรวจสอบทุกจุด ถ้าพบจุดดำอย่างน้อย 1 จุด ให้กำหนดว่าหน้าต่างนี้มีข้อมูล

บล็อกตัวอักษร (Block) คือกลุ่มข้อมูลตัวอักษรที่ประกอบไปด้วยตัวอักษรหลายๆตัวรวมกัน ใน 1 บล็อกจะมีเพียง 1 คอลัมน์เท่านั้น และใน 1 คอลัมน์อาจมีอยู่ 1 ย่อหน้า หรือมีหลายๆ ย่อหน้าได้

Optimum Picture คือรูปขนาดเล็ก ที่ทำการย่อส่วนจากรูปต้นฉบับลงประมาณ 144 เท่า เพื่อใช้ในการประมวลผลที่เกี่ยวกับโครงสร้างของเอกสาร ซึ่งการทำงานกับรูปที่เล็กลงนี้ จะทำให้การประมวลผลเป็นไปอย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและแนวคิดที่เกี่ยวข้อง

2.1 คุณลักษณะทั่วไปของภาพเอกสาร

2.1.1 รูปแบบการจัดเก็บภาพเอกสาร สามารถแบ่งได้เป็น 3 ประเภทดังนี้

1. ภาพขาวดำ หรือเรียกอีกอย่างว่าภาพไบนารี (Binary) เป็นการเก็บภาพแบบง่ายที่สุด คือ ในการเก็บภาพ 1 พิกเซล จะใช้พื้นที่ในการเก็บ 1 บิต ข้อมูลที่เก็บจะเป็น "0" หรือ "1" ทำให้ภาพที่ได้มีเพียงสีขาวหรือดำ เท่านั้น

2. ภาพเกรย์สเกล (Gray Scale) เป็นภาพที่สามารถแสดงสีดำ แล้วไล่ระดับสีไปเป็นสีเทาจนกระทั่งเป็นสีขาว ในการเก็บภาพ 1 พิกเซล ถ้าใช้พื้นที่ในการเก็บ 8 บิต ภาพ Gray Scale นี้จะมีระดับขาวดำได้ถึง 256 ระดับ สามารถคิดได้จากสูตร 2^n เมื่อ n คือจำนวนบิตที่ใช้เก็บข้อมูล

3. ภาพสี โดยทั่วไปภาพ 1 พิกเซล ใช้พื้นที่ในการเก็บ 8 บิต, 16 บิต หรือ 32 บิต ขึ้นอยู่กับรายละเอียดของภาพ เช่นภาพที่มีความละเอียดสีไม่มากนัก ใช้ 8 บิตในการเก็บก็เพียงพอ (เช่น ".gif" สามารถแสดงได้ 256 เฉดสี) แต่ถ้าต้องการภาพเหมือนจริง ก็ควรใช้ 32 บิตในการจัดเก็บ ซึ่งจะทำให้ได้สีมากถึงสี่พันล้านสี

2.1.2 ส่วนประกอบของภาพเอกสาร ภาพเอกสารโดยทั่วไป ไม่ว่าจะเป็นภาพเอกสารประเภทแบบเรียน, วารสาร, บทความวิชาการ, หนังสือพิมพ์, แผนที่ ฯลฯ จะประกอบไปด้วย

1. ข้อความ (ตัวอักษรและตัวเลข) ที่ปรากฏในส่วนต่างๆของภาพเอกสาร ไม่ว่าจะเป็น หัวเรื่อง ชื่อเรื่อง เนื้อหา และเลขหน้า
2. ตาราง
3. สมการคณิตศาสตร์ และสูตรทางวิทยาศาสตร์ต่างๆ
4. รูปภาพ (สี, เกรย์สเกล, ไบนารี่)
5. กราฟิกต่างๆ (ภาพลายเส้น, กราฟแบบต่างๆ, ตราสัญลักษณ์)

ตัวอักษรในส่วนของเนื้อหา โดยทั่วไปมักจะมีขนาดไม่ใหญ่มากนัก (การสแกนที่ความละเอียด 300 จุดต่อนิ้ว ตัวอักษรมีขนาดไม่เกิน 80 พิกเซล) ซึ่งต่างกับขนาดของรูปภาพ ซึ่งส่วนมากจะมีขนาดใหญ่กว่า 80 พิกเซล ทำให้สามารถนำขนาดของกลุ่มดำ มาใช้เป็นเครื่องมือในการกำหนดว่ากลุ่มดำนี้เป็นตัวอักษร หรือกลุ่มดำนี้เป็นรูปภาพได้

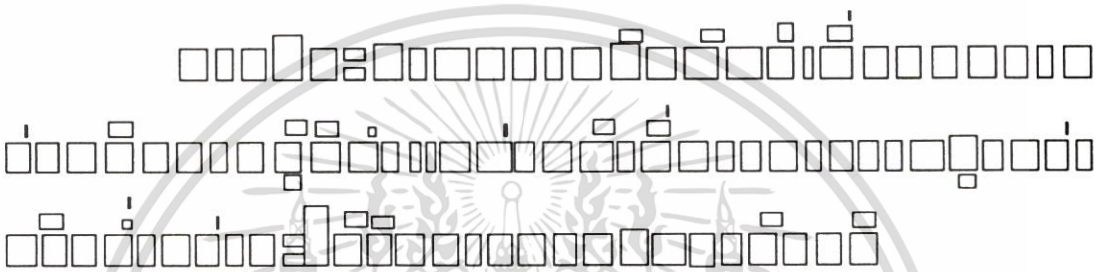
2.1.3 การจัดเก็บส่วนประกอบของภาพเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การจัดเก็บส่วนประกอบของภาพเอกสารที่เป็นตัวอักษรทำได้โดย การเก็บขอบเขตแบบสี่เหลี่ยม (บน, ล่าง, ซ้าย, ขวา) ของตัวอักษร ดังในรูปที่ 2.1

การประสานความสัมพันธ์เพื่อขอความร่วมมือจากผู้มีตำแหน่งหน้าที่ทางการงานสูงกว่าหรือต่ำกว่าจะได้รับการตอบสนองด้วยดี

(ก)



(ข)

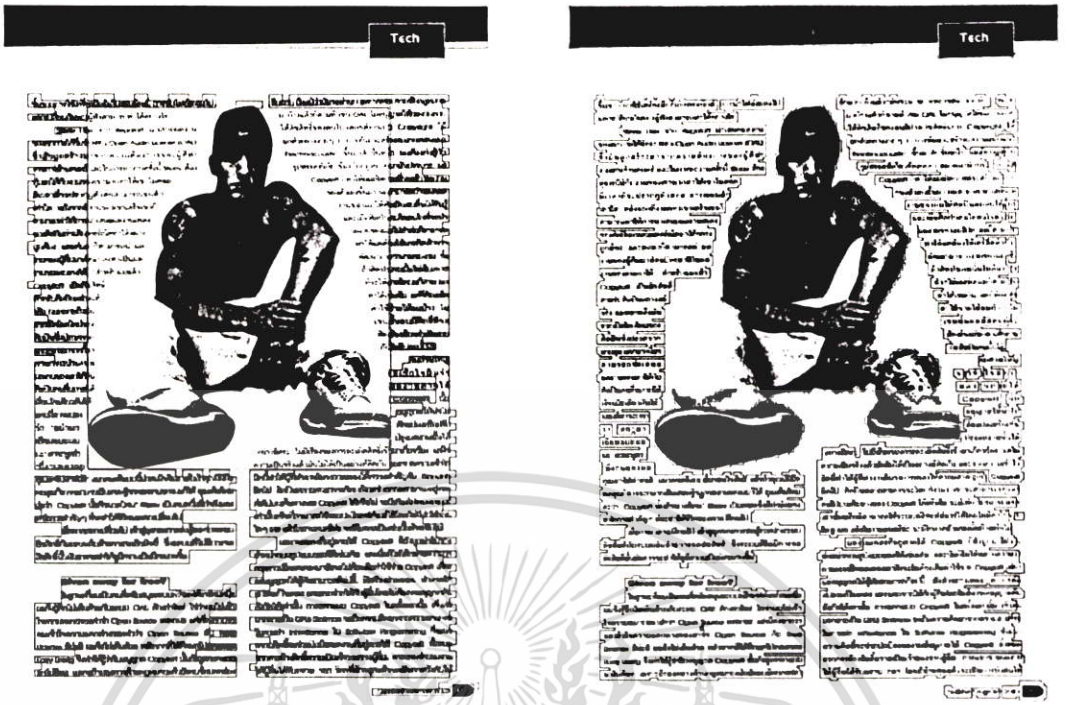
รูปที่ 2.1 การจัดเก็บส่วนประกอบที่เป็นตัวอักษร (ก) รูปต้นฉบับ (ข) การเก็บขอบเขตของตัวอักษร (L, T, R, B)

2. การจัดเก็บส่วนประกอบของภาพเอกสารที่เป็นรูปภาพทำได้โดย การเก็บขอบเขตแบบสี่เหลี่ยม และการเก็บขอบเขตแบบรูปร่างไม่แน่นอน เช่นในรูปที่ 2.2 (ก) เป็นการเก็บขอบเขตของรูปภาพแบบสี่เหลี่ยม ข้อดีของวิธีนี้คือ การจัดเก็บและนำมาใช้สามารถทำได้ง่าย แต่วิธีนี้มีข้อเสียคือ ใช้อธิบายขอบเขตของรูปภาพได้อย่างหยาบๆเท่านั้น และในรูปที่ 2.2 (ข) เป็นการเก็บขอบเขตของรูปภาพแบบรูปร่างไม่แน่นอน ข้อดีของวิธีนี้คือ สามารถใช้อธิบายขอบเขตของรูปภาพได้ละเอียดกว่า การจัดเก็บแบบสี่เหลี่ยม แต่วิธีนี้มีข้อเสียคือ การจัดเก็บและนำมาใช้ทำได้ยาก

(ก)

(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ค) (ง)

รูปที่ 2.2 การจัดเก็บขอบเขตส่วนประกอบที่เป็นรูปภาพ
(ก), (ค) การเก็บขอบเขตของรูปภาพแบบสี่เหลี่ยม
(ข), (ง) การเก็บขอบเขตของรูปภาพแบบรูปร่างไม่แน่นอน

จากรูปที่ 2.2 (ค) เป็นการเก็บขอบเขตของรูปภาพแบบสี่เหลี่ยม จะเห็นได้ว่าขอบเขตของรูปภาพ ได้ครอบคลุมไปถึงตัวอักษร ทำให้ข้อมูลตัวอักษรที่อยู่ในขอบเขตของรูปไม่ได้รับการจัดเก็บ และรูปที่ 2.2 (ง) เป็นการเก็บขอบเขตของรูปภาพแบบรูปร่างไม่แน่นอน การจัดเก็บข้อมูลรูปจะจัดเก็บเฉพาะขอบเขตของรูปเท่านั้น ทำให้ข้อมูลตัวอักษร สามารถจัดเก็บได้อย่างครบถ้วน

2.2 ขั้นตอนการประมวลผลภาพเอกสาร

1. การทำภาพ (Image Acquisition) เป็นการนำภาพที่เก็บในสื่อที่ไม่ใช่ดิจิตอล เช่น ภาพถ่าย, หนังสือ มาเก็บให้อยู่ในรูปแบบดิจิตอลโดยใช้อุปกรณ์ต่างๆ เช่น สแกนเนอร์ (Scanner), กล้องถ่ายรูปดิจิตอล (Digital Camera)

2. การปรับปรุงรูป (Preprocessing) เป็นขั้นตอนการปรับปรุงรายละเอียดของภาพให้มีความชัดเจนยิ่งขึ้น เช่น การแก้ไขความเข้มของแสง (Enhancing Contrast) การปรับความเอียงของภาพ (Skew Correction) และการตัด-ขยายกรอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การแปลงภาพสี และภาพเกรย์สเกล (Gray Scale) ให้เป็นภาพไบนารี่ โดยใช้ค่ากลาง (Threshold) ในการกำหนดว่าจุดภาพใดเป็นวัตถุ หรือจุดภาพใดเป็นพื้นหลัง ขั้นตอนนี้เป็นขั้นตอนที่สำคัญเพราะการเลือกค่าเธรดโชลด์ที่ไม่ดี จะทำให้สูญเสียข้อมูลที่สำคัญที่จะนำไปประมวลผลในขั้นตอนต่อไปได้

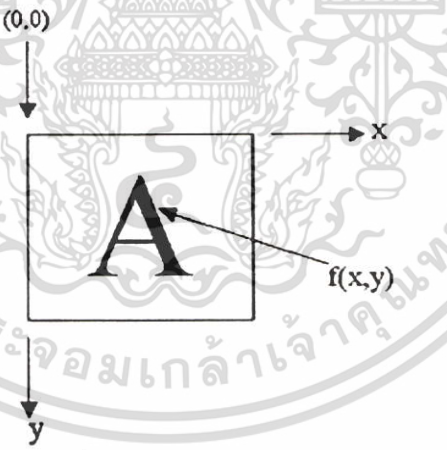
4. การแบ่งส่วนประกอบภาพเอกสาร (Document Page Segmentation) ขั้นตอนนี้ใช้ในการแบ่งส่วนประกอบภาพเอกสารออกเป็นส่วนๆ และระบุได้ว่าส่วนไหนเป็นรูปภาพ และส่วนไหนเป็นตัวอักษร

5. การจดจำและการแปล (Recognition and Interpretation) เป็นขั้นตอนที่ใช้ในการกำหนดความหมาย (Meaning) ให้กับวัตถุนั้นๆ ด้วยกฎและข้อมูลที่ได้กำหนดไว้ในฐานความรู้

2.3 ความรู้พื้นฐานเกี่ยวกับข้อมูลภาพอิเล็คทรอนิกส์

2.3.1 การแสดงภาพดิจิตอลและการอ้างอิงตำแหน่งในภาพ

ภาพดิจิตอลโดยทั่วไปจะแสดงผลแบบสองมิติ การอ้างอิงตำแหน่งในภาพสามารถทำได้โดยใช้ฟังก์ชั่น $f(x,y)$ โดย x คือตำแหน่งในแนวคอลัมน์ และ y คือตำแหน่งในแนวแถว ดังในรูปที่ 2.3 แสดงถึงระบบแกนและพิกัด ที่ใช้ในการวิจัยนี้



รูปที่ 2.3 ระบบแกนและพิกัด

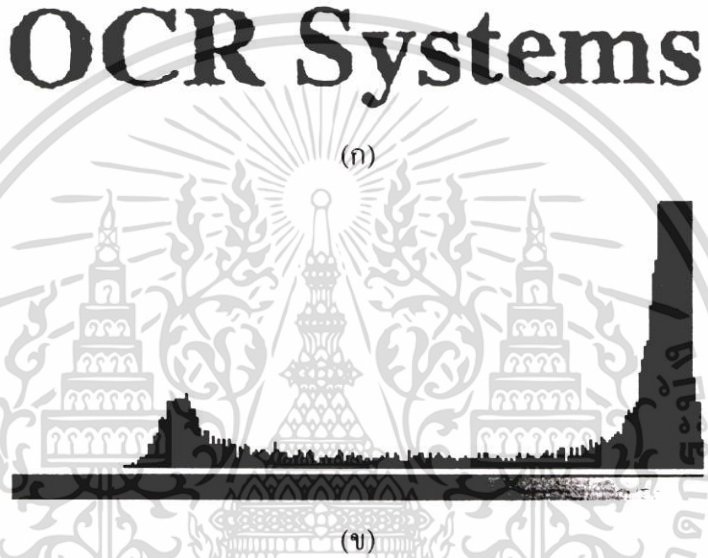
จากรูปที่ 2.3 การจัดเก็บข้อมูลภายในรูป เป็นการจัดเรียงในรูปแบบ อาร์เรย์ NxM เมตริกซ์ ซึ่ง N คือ จำนวนแถว และ M คือ จำนวนคอลัมน์ ดังแสดงในสมการที่ 2.1

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 กราฟแสดงความถี่ (ฮิสโตแกรม) ของรูปดิจิตอล

2.3.2.1 กราฟแสดงความถี่ของค่าระดับสีเทา (Gray Level) คือกราฟที่แสดงความถี่หรือความหนาแน่นของค่าระดับสีเทา (Gray Level) ที่อยู่ในช่วง [0, L-1] ของรูปนั้นๆ เมื่อ L คือค่าระดับสีเทาสูงสุด เช่นจุดภาพ 1 จุดใช้พื้นที่ในการจัดเก็บ 8 บิต ดังนั้นค่า L คือ $2^8 = 256$ ดังนั้นค่าระดับสีเทาจะอยู่ในช่วง [0, 255] สามารถแสดงได้โดยใช้กราฟแบบแท่ง ดังรูปที่ 2.4 (ข) โดยมีแกน x แทนค่าระดับสีเทาในช่วง [0, L-1] ส่วนแกน y แทนค่าความถี่ หรือความหนาแน่นของค่าระดับสีเทา



(ก)

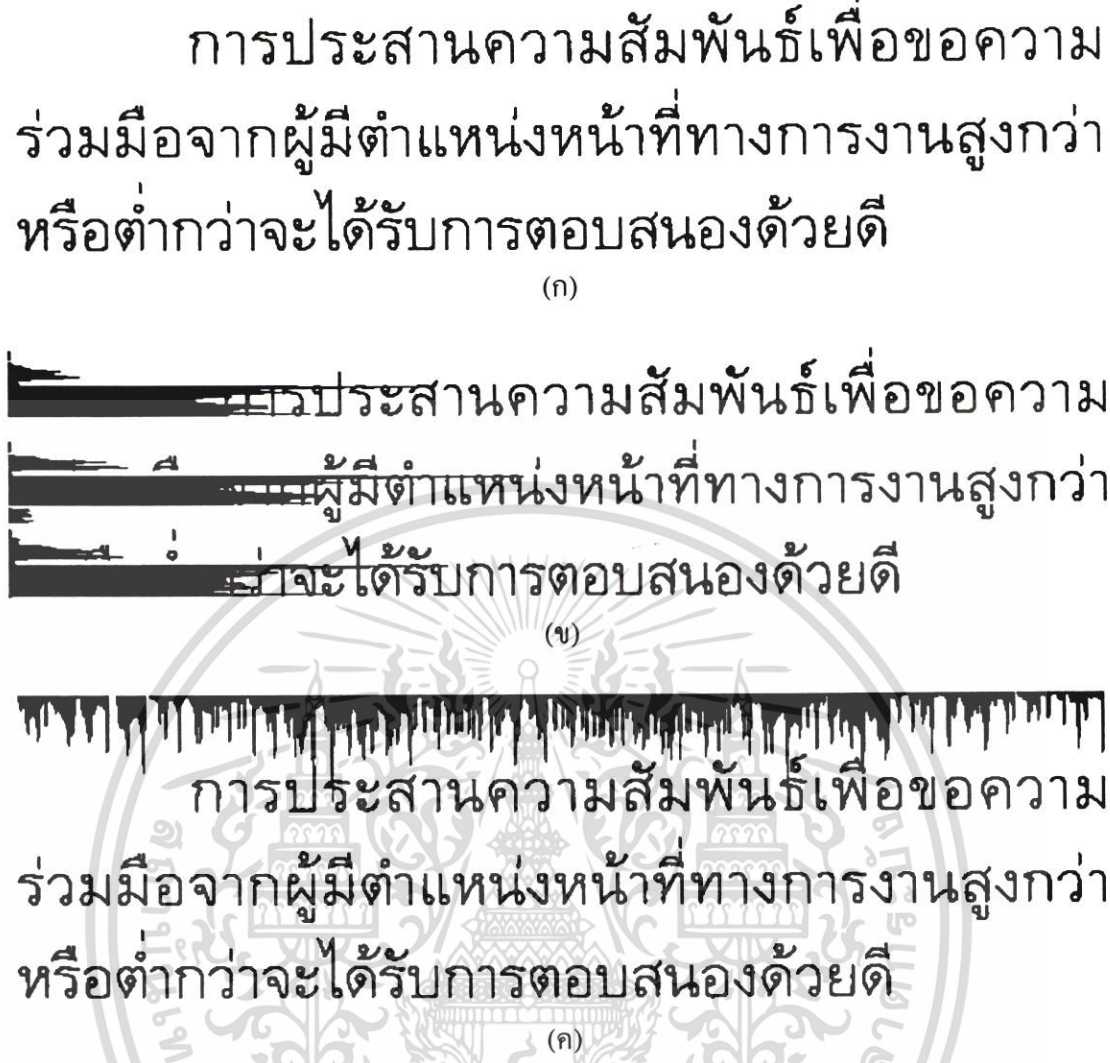
(ข)

รูปที่ 2.4 แสดงฮิสโตแกรมของรูปเกรย์สเกล (L=256) (ก) รูปต้นฉบับ (ข) ฮิสโตแกรม ที่อยู่ในช่วง [0-255]

การกระจายตัวของกราฟฮิสโตแกรม สามารถบอกลักษณะของรูปได้ว่า เป็นรูปที่มืดหรือเป็นรูปที่สว่าง เช่นรูปที่ 2.4 (ข) จะเห็นได้ว่าความหนาแน่นอยู่ทางด้านขวาของกราฟ ทำให้ทราบได้ว่ารูปที่ใช้ทำฮิสโตแกรมนี้ มีลักษณะที่สว่าง ส่วนรูปที่มีลักษณะมืด ความหนาแน่นของกราฟจะกระจายตัวอยู่ทางด้านซ้าย

2.3.2.2 กราฟแสดงความถี่ของจุดดำ มีลักษณะการทำงานเหมือนกับการเอียงถาดลูกแก้วไปข้างใดข้างหนึ่ง โดยรูปที่ 2.5 (ข) เป็นการทำฮิสโตแกรมของจุดดำ ในแนวนอน ซึ่งสามารถนำมาประยุกต์ใช้ในการหาระดับบรรทัดของตัวอักษรได้ และรูปที่ 2.5 (ค) เป็นการทำฮิสโตแกรมของจุดดำ ในแนวตั้ง ซึ่งสามารถนำมาประยุกต์ใช้ในการตัดแบ่งคอลัมน์ได้

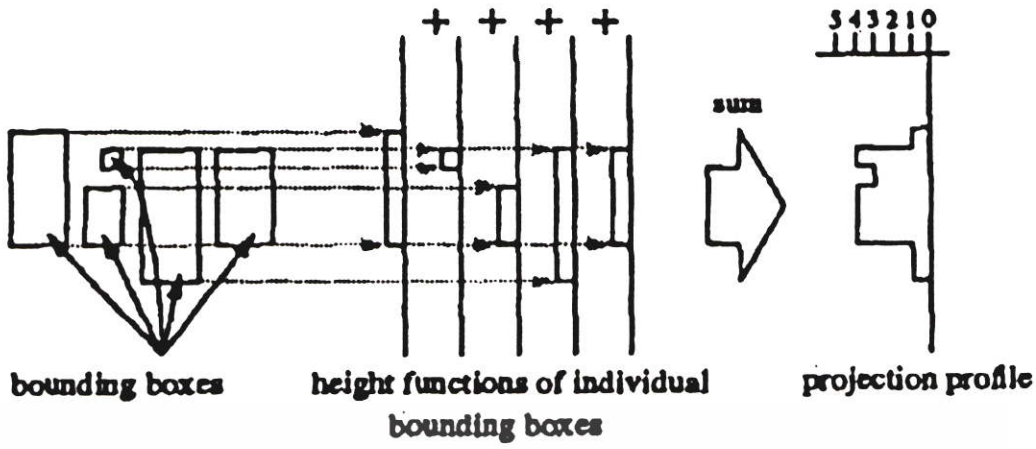
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงฮิสโตแกรมของจุดดำ (ก) รูปต้นฉบับ
(ข) ฮิสโตแกรมของจุดดำในแนวนอน
(ค) ฮิสโตแกรมของจุดดำในแนวตั้ง

2.3.2.3 กราฟแสดงความถี่ด้วยขอบของส่วนประกอบย่อย[2, 5] การทำงานจะเปลี่ยนจากการใช้จุดดำในการทำฮิสโตแกรมมาเป็นการใช้ขอบของส่วนประกอยย่อย ดังรูปที่ 2.6 ซึ่งในกรณีที่เรารู้ขอบเขตของส่วนประกอบย่อยแล้ว การทำฮิสโตแกรมด้วยวิธีนี้ จะเป็นไปอย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)

การประสานความสัมพันธ์เพื่อขอความร่วมมือจากผู้มีตำแหน่งหน้าที่ทางการงานสูงกว่าหรือต่ำกว่าจะได้รับการตอบสนองด้วยดี

(ข)

รูปที่ 2.6 แสดงการทำฮิสโตแกรมด้วยขอบของส่วนประกอบย่อย
(ก) ขั้นตอนการทำฮิตโตแกรมด้วยขอบของส่วนประกอบย่อย[2]
(ข) ตัวอย่างการทำฮิตโตแกรมด้วยขอบของส่วนประกอบย่อย

2.3.3 การแปลงรูปสี และรูปเกรย์สเกลเป็นรูปไบนารี่

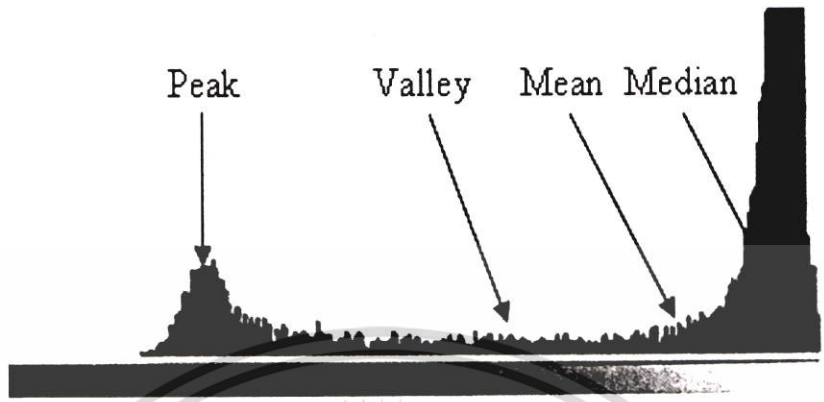
ก่อนจะนำรูปสี หรือรูปเกรย์สเกล (Gray Scale) มาเข้ากระบวนการแบ่งส่วนประกอบภาพเอกสาร จำเป็นต้องทำการแปลงรูปดังกล่าวให้เป็นรูปไบนารี่ (Binary) ก่อน โดยใช้เทคนิคการทำเธรดโชลด์ (Threshold Technique) ซึ่งหลักการนี้จะทำการแบ่งกลุ่มของจุดภายในรูปออกเป็น 2 กลุ่มอย่างชัดเจน คือกลุ่มของวัตถุ (Object) และกลุ่มของส่วนที่เป็นพื้นหลัง โดยทำการตรวจสอบแต่ละจุดภายในรูปโดยใช้ $f(x,y)$ เป็นตัวเปรียบเทียบ ถ้าค่า $f(x,y)$ น้อยกว่าค่าเธรดโชลด์ $f(x,y) < Thr$ ถือว่าเป็นจุดของวัตถุ แต่ถ้าค่า $f(x,y)$ มากกว่าหรือเท่ากับค่าเธรดโชลด์ $f(x,y) \geq Thr$ ถือว่าเป็นจุดของพื้นหลัง ดังนั้นการแปลงเป็นรูปไบนารี่สามารถนิยามด้วยสมการไดดังสมการที่ 2.2

$$f_{thr}(x,y) = \begin{cases} 1 & if \quad f(x,y) < Thr \\ 0 & if \quad f(x,y) \geq Thr \end{cases} \tag{2.2}$$

เมื่อ 1 คือจุดที่เป็นวัตถุ และ 0 คือจุดที่เป็นพื้นหลังของรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.1 การกำหนดค่าเธรดโชลด์ สามารถทำได้หลายวิธี เช่น การใช้ค่ากลาง (Median), การใช้ค่าเฉลี่ย (Mean) หรือใช้ Histogram เข้าช่วย ดังรูปที่ 2.7



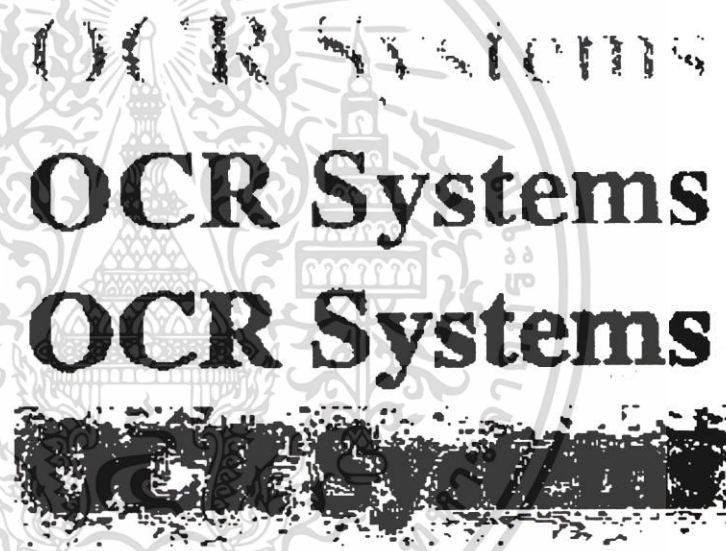
(ก)

(ข) Peak : Threshold=62

(ค) Valley : Threshold=162

(ง) Mean : Threshold=213

(จ) Median : Threshold=244



รูปที่ 2.7 ค่าต่างๆที่นำมาใช้เป็นค่าเธรดโชลด์

ตารางที่ 2.1 คุณสมบัติต่างๆ ของค่าที่เลือกมาเป็นเธรดโชลด์

	Value	Count	Count Below	Count Above	%In Range	%Below	%Above
Peak	62	102	1017	20925	0.463	4.614	94.924
Valley	162	21	3988	18035	0.095	18.091	81.814
Mean	213	27	5035	16982	0.122	22.84	77.037
Median	244	1126	10867	10051	5.108	49.297	45.595

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 (ก) แสดงฮิสโตแกรมที่ได้จากรูปที่ 2.4 (ก) โดยในรูปที่ 2.7 (ก) ได้แสดงตำแหน่งที่เป็นค่า Mean, Median, Peak, Valley และคุณสมบัติของค่าต่างๆเหล่านี้ได้แสดงในตารางที่ 2.1 โดย แต่ละคอลัมน์มีความหมายดังนี้

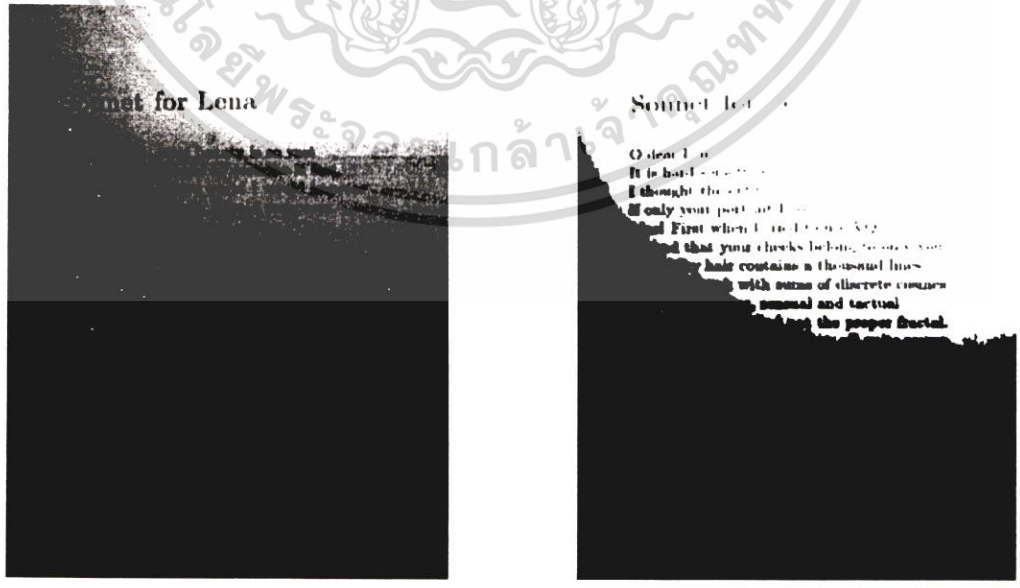
- **Value** หมายถึง ค่าที่นำมาใช้เป็นเธรดโชลด์ ซึ่งจะอยู่ในช่วง [0-255]
- **Count** หมายถึง จำนวนจุดในรูปที่มีค่าเท่ากับ Value
- **Count Below** หมายถึง จำนวนจุดในรูปที่มีค่าน้อยกว่าค่า Value
- **Count Above** หมายถึง จำนวนจุดในรูปที่มีค่ามากกว่าค่า Value
- **%In Range** หมายถึง จำนวนจุดในรูปที่มีค่าเท่ากับ Value คิดเป็นเปอร์เซ็นต์
- **%Below** หมายถึง จำนวนจุดในรูปที่มีค่าน้อยกว่าค่า Value คิดเป็นเปอร์เซ็นต์
- **%Above** หมายถึง จำนวนจุดในรูปที่มีค่ามากกว่าค่า Value คิดเป็นเปอร์เซ็นต์

โดยรูปที่ 2.4 (ก) มีจำนวนจุดในรูปทั้งสิ้น 22044 จุด

2.3.3.2 การแบ่งประเภทของการทำเธลดโชลด์ สามารถแบ่งได้เป็น 2 ประเภทคือ

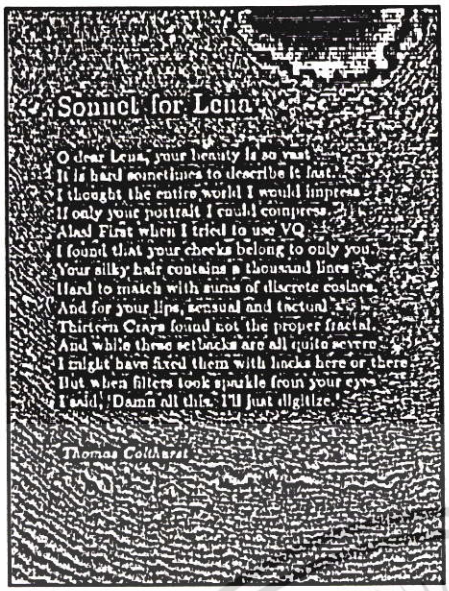
1. การทำเธรดโชลด์แบบครอบคลุม (Global Thresholding) คือการใช้ค่าเธรดโชลด์ค่าเดียวในการทำเธรดโชลด์กับแต่ละจุดภาพทั่วทั้งภาพ เหมาะกับภาพที่มีจุดภาพของส่วนที่เป็นวัตถุกับจุดภาพของส่วนพื้นหลัง ที่แต่ละส่วนมีระดับความเข้มสม่ำเสมอกัน และระดับความเข้มระหว่างวัตถุและพื้นหลังแตกต่างกันอย่างชัดเจนทั่วทั้งภาพ

2. การทำเธรดโชลด์แบบปรับค่า (Adaptive Thresholding) ค่าเธรดโชลด์มีการปรับเปลี่ยนค่าตามตำแหน่งจุดภาพ เหมาะกับภาพที่มีระดับความเข้มไม่สม่ำเสมอในส่วนของพื้นหลัง เช่นในรูปที่ 2.8 (ก)

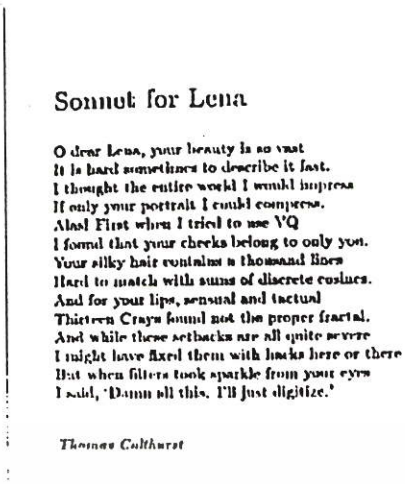


(ก) (ข)

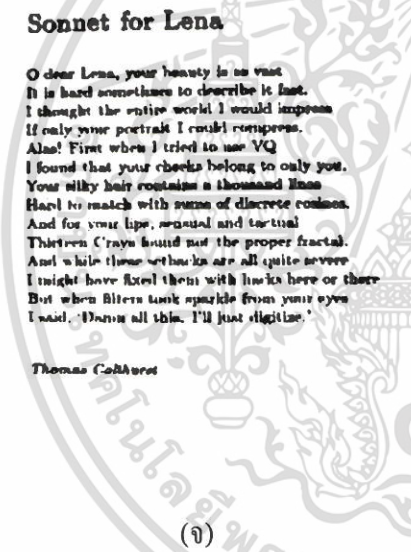
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



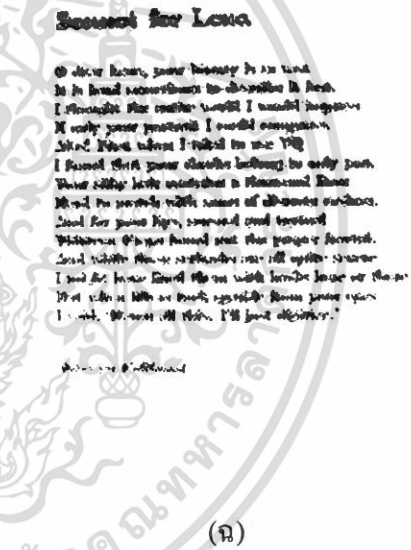
(ค)



(ง)



(จ)



(ฉ)

รูปที่ 2.8 แปลงภาพเกรย์สเกล เป็นภาพไบนารี่ [6]

(ก) รูปต้นฉบับเป็นรูปเกรย์สเกลที่มีระดับความเข้มของพื้นหลังภายในรูปต่างกัน

(ข) Global Thresholding : ค่า Mean จากทั้งรูป นำมาเป็นค่า Threshold

(ค) Adaptive Thresholding : ค่า Mean จาก 7×7 Neighborhood นำมาเป็นค่า Threshold

(ง) Adaptive Thresholding : ค่า Mean จาก 7×7 Neighborhood ลบด้วย 7 นำมาเป็นค่า Threshold

(จ) Adaptive Thresholding : ค่า Mean จาก 75×75 Neighborhood ลบด้วย 10 นำมาเป็นค่า Threshold

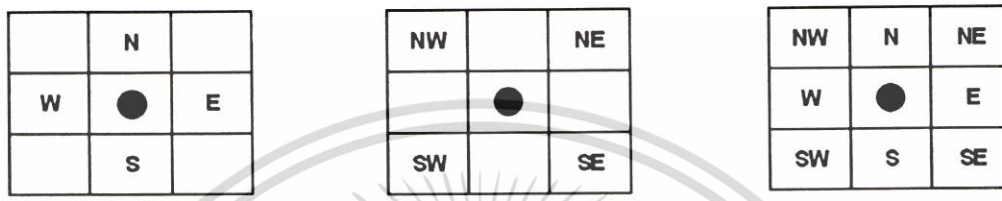
(ฉ) Adaptive Thresholding : ค่า Median จาก 7×7 Neighborhood ลบด้วย 4 นำมาเป็นค่า Threshold

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 ความสัมพันธ์ระหว่างจุด (Relationships between pixels)

ความสัมพันธ์ระหว่างจุดที่อยู่ใกล้ชิดกันโดยกำหนดให้จุด p อยู่ที่พิกัด (x, y) จะมีจุดที่อยู่ใกล้ในแนวตั้งและแนวนอน ได้แก่ (x+1, y), (x-1, y), (x, y+1), (x, y-1) เรียกว่าส่วนที่ติดกัน 4 ทิศของ p (4-neighbors) หรือ $N_4(p)$

นอกเหนือจากนี้ยังมีอีก 4 จุด คือ (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) เรียกว่า $N_D(p)$ และเมื่อรวม $N_4(p)$ และ $N_D(p)$ เรียกว่า $N_8(p)$



$N_4(p)$ $N_D(p)$ $N_8(p)$

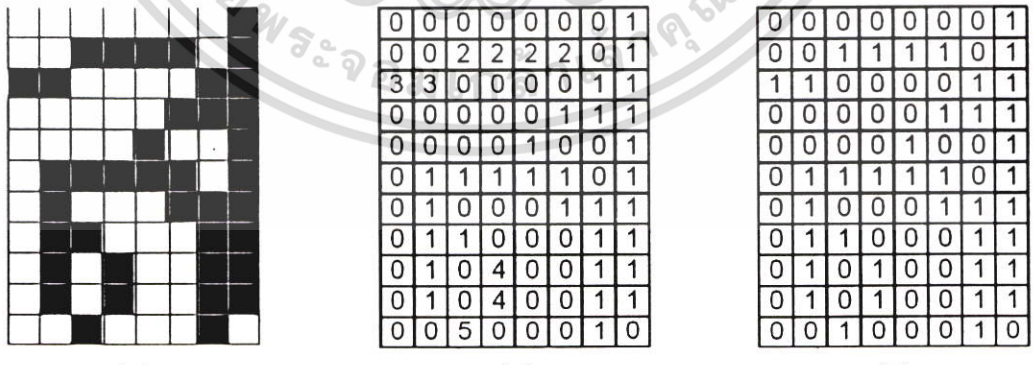
รูปที่ 2.9 จุดที่มีความสัมพันธ์แบบ $N_4(p)$, $N_D(p)$ และ $N_8(p)$

2.3.5 การเชื่อมต่อกันระหว่างจุด (Labeling)

การหาขอบเขต และส่วนประกอบของวัตถุ สามารถใช้วิธีการเชื่อมต่อกันระหว่างจุด ในการค้นหาได้ ซึ่งการเชื่อมต่อกันระหว่างจุดแบ่งได้เป็น 2 ชนิดคือ

1. ส่วนติดกัน 4 ส่วน (4-Neighborhood Connected Component) ของจุด p และ q เมื่อ p อยู่ใน $N_4(p)$

2. ส่วนติดกัน 8 ส่วน (8-Neighborhood Connected Component) ของจุด p และ q เมื่อ p อยู่ใน $N_8(p)$



(ก)

0	0	0	0	0	0	0	1
0	0	2	2	2	2	0	1
3	3	0	0	0	0	1	1
0	0	0	0	0	1	1	1
0	0	0	0	1	0	0	1
0	1	1	1	1	1	0	1
0	1	0	0	0	1	1	1
0	1	1	0	0	0	1	1
0	1	0	4	0	0	1	1
0	1	0	4	0	0	1	1
0	0	5	0	0	0	1	0

(ข)

0	0	0	0	0	0	0	1
0	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1
0	0	0	0	0	1	1	1
0	0	0	0	1	0	0	1
0	1	1	1	1	1	0	1
0	1	0	0	0	1	1	1
0	1	1	0	0	0	1	1
0	1	0	1	0	0	1	1
0	1	0	1	0	0	1	1
0	0	1	0	0	0	1	0

(ค)

รูปที่ 2.10 การ Label, (ก) รูปต้นฉบับ

(ข) การ Label ด้วยวิธี 4-Neighborhood Connected Component

(ค) การ Label ด้วยวิธี 8-Neighborhood Connected Component

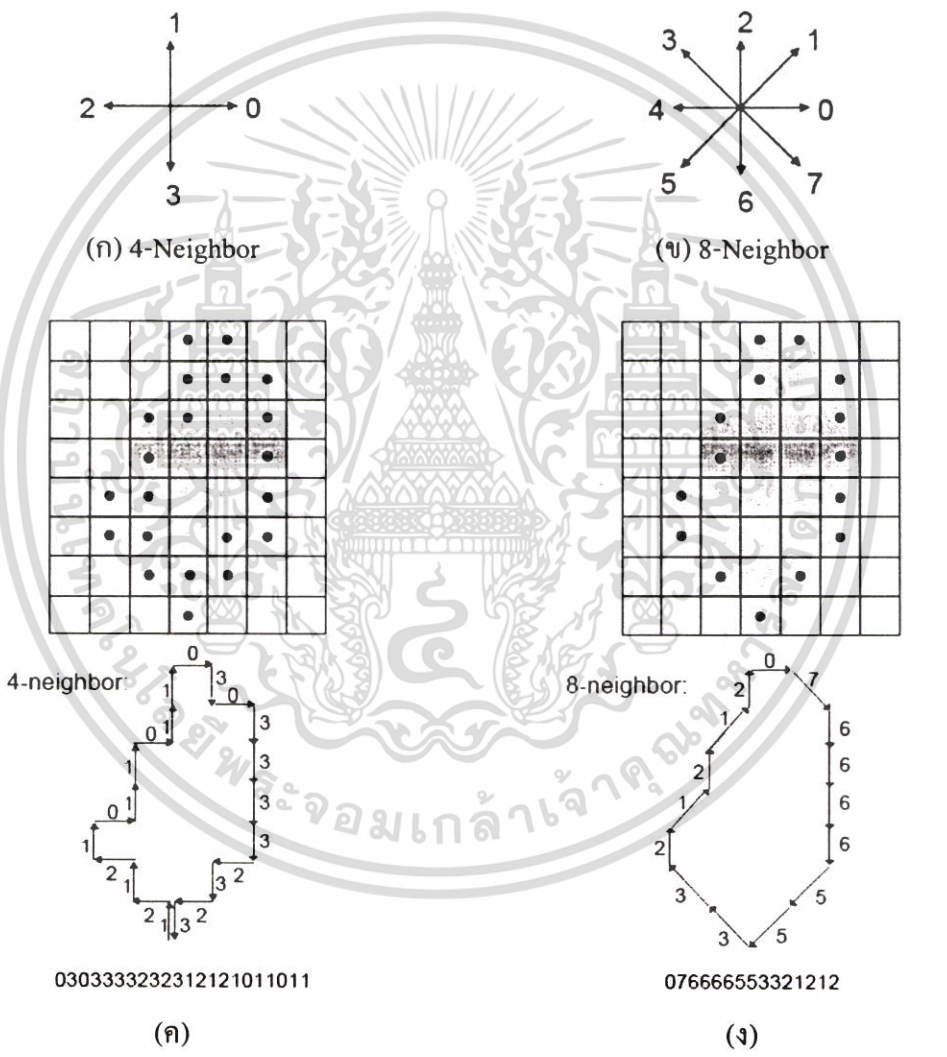
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รูปที่ 2.10 (ข) Label ด้วยวิธี 4-Neighborhood Connected Component แบ่งได้เป็น 5 วัตถุ และรูปที่ 2.10 (ค) Label ด้วยวิธี 8-Neighborhood Connected Component แบ่งได้เป็น 1 วัตถุ

2.3.6 Chain Codes

Chain Codes ใช้ในการเดินรอบขอบ (Contouring) ของส่วนประกอบที่ติดต่อกัน เพื่อหาขอบเขตและรูปร่างของส่วนประกอบ ผลลัพธ์ที่ได้คือ ขอบเขตของส่วนประกอบ ซึ่งสามารถมีรูปร่างใดๆก็ได้ ไม่จำเป็นว่าต้องเป็นสี่เหลี่ยม รูปร่างของขอบเขตที่ได้ขึ้นอยู่กับข้อมูล โดย Chain Codes โซ่มีทั้งแบบ 4 ทิศทาง และ 8 ทิศทาง ดังรูปที่ 2.11



รูปที่ 2.11 การเดินรอบขอบด้วย Chain Codes[7]

(ก) Chain Codes 4 ทิศทาง, (ข) Chain Codes 8 ทิศทาง

(ค) การเดินรอบขอบด้วย Chain Codes 4 ทิศทาง

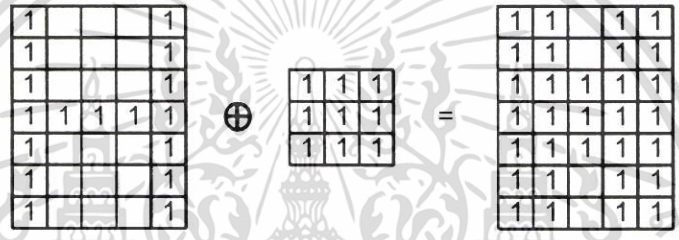
(ง) การเดินรอบขอบด้วย Chain Codes 8 ทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
60906

ข้อควรระวังในการใช้ Chain Codes คือ ตำแหน่งเริ่มต้นในการเดินรอบขอบวัตถุควรอยู่ที่ขอบของวัตถุ เพราะหากตำแหน่งเริ่มต้นอยู่ภายในของวัตถุแล้ว จะทำให้ Chain Codes ไม่สามารถทำการเดินรอบขอบของวัตถุได้

2.4 วิธีการที่นำมาใช้ในการแบ่งส่วนประกอบภาพเอกสาร

2.4.1 การทำ Dilation เป็นการขยายจุดภาพให้ใหญ่ขึ้น เช่นในรูปที่ 2.12 เป็นการขยายจุดภาพทั้งในแนวตั้งและในแนวนอน ด้วย Mask ขนาด 3x3 ซึ่งสามารถนำมาประยุกต์ใช้ในการเชื่อมขอบเขตของตัวอักษรที่อยู่ใกล้กันเข้าด้วยกัน ทั้งในแนวตั้งและในแนวนอน เพื่อรวมเป็นขอบเขตระดับบล็อกของตัวอักษรได้



รูปที่ 2.12 การทำ Dilation ด้วย Mask ขนาด 3x3

การประสานความสัมพันธ์เพื่อขอความร่วมมือจากผู้มีตำแหน่งหน้าที่ทางการงานสูงกว่าหรือต่ำกว่าจะได้รับการตอบสนองด้วยดี

(ก)

การประสานความสัมพันธ์เพื่อขอความร่วมมือจากผู้มีตำแหน่งหน้าที่ทางการงานสูงกว่าหรือต่ำกว่าจะได้รับการตอบสนองด้วยดี

(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ค)

(ง)

รูปที่ 2.13 เปรียบเทียบการทำ Dilation ด้วย Mask ขนาดต่างๆ, (ก) รูปต้นฉบับ
(ข) รูปที่ผ่านการทำ Dilation ด้วย Mask ขนาด 3x3
(ค) รูปที่ผ่านการทำ Dilation ด้วย Mask ขนาด 11x11
(ง) รูปที่ผ่านการทำ Dilation ด้วย Mask ขนาด 35x35

รูปที่ 2.13 (ง) เป็นตัวอย่างการทำ Dilation ด้วย Mask ขนาด 35x35 ซึ่งทำให้ตัวอักษรที่อยู่ใกล้กันรวมกันเป็นบล็อกเดียวกัน

2.4.2 วิธีการตัดซ้ำแนวแกน X-Y (Recursive Projection X-Y Cut) วิธีการนี้มีพื้นฐานมาจากการฮิตโตแกรมจุดดำในแนวตั้งและแนวนอน โดยจะทำการฮิตโตแกรมภาพในแนวตั้ง หรือแนวนอน แล้วตรวจสอบว่าสามารถตัดแบ่งในแนวที่กำลังทำการตรวจสอบได้หรือไม่ ถ้าได้ให้ทำการตัดแบ่ง แล้วตรวจสอบในแนวตั้งฉากกับแนวที่กำลังตรวจสอบอยู่ สลับกันเรื่อยไปจนไม่สามารถตัดแบ่งได้อีก

2.5 การแบ่งส่วนประกอบภาพเอกสาร

การแบ่งส่วนประกอบภาพเอกสาร เป็นการแบ่งภาพเอกสารออกเป็นส่วนๆ โดยการแบ่งจะพิจารณาจากข้อมูลที่อยู่ในภาพเอกสาร ซึ่งข้อมูลในแต่ละส่วนย่อยที่แบ่งได้นี้ ต้องเป็นข้อมูลชนิดเดียวกัน เช่น ส่วนย่อยที่มีข้อมูลเป็นชนิดข้อความ ส่วนย่อยที่มีข้อมูลเป็นชนิดรูปภาพ ซึ่งวิธีการที่ใช้แบ่งส่วนประกอบภาพเอกสารนี้มีด้วยกันหลายวิธี เช่น วิธีการตัดซ้ำตามแนวแกน (Recursive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Projection X-Y Cut), วิธีการเดินรอบขอบ (Contouring) ฯลฯ ซึ่งวิธีต่างๆนี้ สามารถแบ่งได้เป็น 3 ประเภทคือ วิธีการแบบบนลงล่าง (Top-Down Approach)[2], วิธีการแบบล่างขึ้นบน (Bottom-Up Approach)[4, 5] และวิธีการแบบผสม (Mixed Approach)[1] โดยแต่ละวิธีจะมีจุดเด่นต่างกันไป อย่างเช่นวิธีการแบบบนลงล่างเป็นวิธีที่สามารถทำงานได้อย่างรวดเร็ว วิธีการแบบล่างขึ้นบนเป็นวิธีการที่สามารถใช้ได้กับเอกสารที่มีรูปแบบที่หลากหลายได้ แต่จะใช้เวลาในการประมวลผลนาน วิธีการแบบผสมเป็นการนำเอาการทำงานของวิธีการแบบบนลงล่างและวิธีการแบบล่างขึ้นบนมาใช้ร่วมกัน เพื่อให้การทำงานเป็นไปอย่างรวดเร็วด้วยคุณสมบัติของวิธีการแบบบนลงล่าง และสามารถทำงานได้อย่างถูกต้องด้วยคุณสมบัติของวิธีการแบบล่างขึ้นบน

การแบ่งส่วนประกอบภาพเอกสารแบ่งได้เป็น 3 ประเภทดังนี้

2.5.1 วิธีการแบบบนลงล่าง (Top-Down Approach) เช่นวิธีการตัดซ้ำแนวแกน X-Y (Recursive Projection X-Y Cut) วิธีการนี้ในเบื้องต้นจะทำการแบ่งภาพอย่างหยาบๆก่อน โดยแบ่งภาพเอกสารให้เป็นภาพเอกสารส่วนใหญ่ๆก่อน จากนั้นจึงนำภาพเอกสารแต่ละส่วนใหญ่มาแบ่งให้เป็นส่วนย่อยๆ และทำแบบนี้จนกระทั่งข้อมูลภาพถูกแบ่งเป็นส่วนย่อยๆที่ไม่สามารถแบ่งได้อีก ข้อดีของวิธีการแบบบนลงล่างนี้คือ การแบ่งภาพเอกสารให้เป็นส่วนใหญ่ๆ เนื่องจากเป็นการแบ่งแบบหยาบๆ ทำให้การประมวลผลทำได้รวดเร็ว แต่วิธีการแบบบนลงล่างนี้มีข้อเสียคือ เมื่อแบ่งภาพเอกสารละเอียดลงเรื่อยๆ การประมวลผลจะใช้เวลามากขึ้น

2.5.2 วิธีการแบบล่างขึ้นบน (Bottom-Up Approach) วิธีการนี้เริ่มจากการวิเคราะห์ส่วนประกอบย่อย แล้วนำส่วนประกอบย่อยที่ได้มาประกอบกันเป็นบรรทัด, เป็นบล็อก โดยพิจารณาจากความใกล้ชิด และขนาดของแต่ละส่วนประกอบ ทำให้วิธีการนี้สามารถทำงานได้อย่างถูกต้อง ถึงแม้ว่าภาพเอกสารจะมีรูปแบบที่หลากหลายนอกจากนั้น วิธีการล่างขึ้นบนนี้ส่วนมากยังคงทำงานได้อย่างถูกต้องกับเอกสารที่มีความเอียงและเอกสารที่มีรูปแบบไม่แน่นอน เทคนิคในการใช้วิธีการแบบล่างขึ้นบนมีอยู่มากมาย เช่นในบางครั้งต้องการให้ข้อมูลที่อยู่ห่างไกลกันไม่มากนักเป็นข้อมูลชนิดเดียวกันอย่างเช่น ตัวอักษรที่อยู่ใกล้กันจะทำการจัดให้อยู่ในบล็อกเดียวกัน ก็สามารถทำได้ด้วยการใช้วิธีการทำภาพเบลอเข้าช่วยได้ หรือจะใช้วิธีการวิเคราะห์หาบล็อกสี่เหลี่ยมโดยการใช้หน้าต่างเล็กๆ จำนวนมากที่อยู่ในขอบเขตใหญ่ช่วยในการเชื่อมต่อระหว่างบล็อก วิธีการนี้สามารถลดเวลาที่ใช้ในการประมวลผลลงได้อย่างมาก [3]

2.5.3 วิธีการแบบผสม (Hybrid Approach) เป็นการนำวิธีการแบบบนลงล่างมาใช้ร่วมกับวิธีการแบบล่างขึ้นบน เช่น อัลกอริทึมของ B. Kruatrachue และ P.Suthaphan [1] ซึ่งทำงานดังนี้

1. Block Extraction (top-down) โดยใช้กรอบหน้าต่างขนาด 32x32 พิกเซลเดินรอบขอบของวัตถุ เพื่อใช้ในการแบ่งวัตถุออกเป็นส่วนประกอบใหญ่ การใช้หน้าต่างขนาด 32x32 พิกเซล ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเดินรอบขอบของวัตถุทำให้สามารถหาขอบเขตของส่วนประกอบใหญ่ได้อย่างรวดเร็ว แต่ส่วนประกอบใหญ่ที่ได้นี้ ยังไม่สามารถนำไปใช้งานได้ จำเป็นจะต้องมีการนำส่วนประกอบใหญ่นี้ไปประมวลผลต่อ เพื่อให้ภายในแต่ละส่วนประกอบใหญ่มีข้อมูลที่เป็นชนิดเดียวกัน

2. Multi-Column Block Detection and Segmentation (bottom-up) ใช้จำแนกว่าส่วนประกอบย่อยในส่วนประกอบใหญ่นี้ เป็นตัวอักษร หรือเป็นรูปภาพ โดยกำหนดจากขอบเขตของส่วนประกอบย่อย คือถ้าส่วนประกอบย่อยนี้มีขนาดไม่ถึง 100 พิกเซลถือว่าส่วนประกอบย่อยนี้เป็นตัวอักษร แต่ถ้าส่วนประกอบย่อยนี้มีขนาดมากกว่า 100 พิกเซลจะถือว่าส่วนประกอบย่อยนี้เป็นส่วนของรูปภาพ จากนั้นนำส่วนประกอบย่อยที่เป็นตัวอักษรมารวมกันเป็นขอบเขตที่ใหญ่ขึ้น เช่น เป็นบรรทัด, เป็นบล็อก และทำการแยกส่วนประกอบย่อยที่เป็นรูปภาพออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

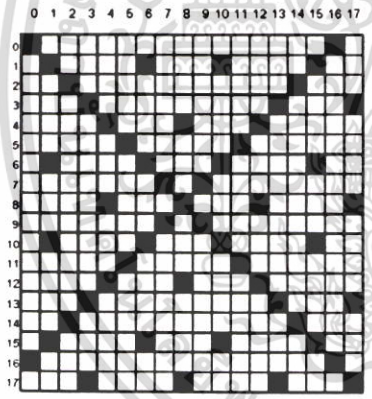
บทที่ 3
การหาจุดเริ่มต้นของการประมวลผลด้วยการใช้หน้าต่าง

3.1 แนวคิดในการตรวจสอบหาจุดเริ่มต้นของการประมวลผลด้วยการใช้หน้าต่าง

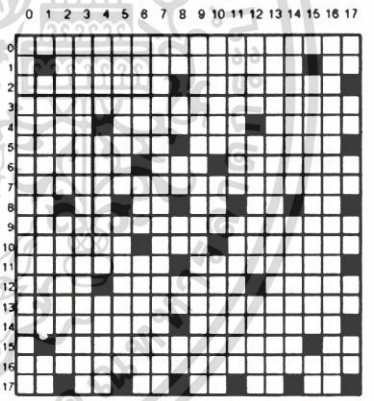
การแบ่งส่วนประกอบภาพเอกสารไบนารี่ อันดับแรกต้องทำการตรวจสอบหาจุดเริ่มต้นของการประมวลผล ซึ่งก็คือการตรวจสอบหาจุดดำจุดแรก โดยทั่วไปจะใช้การตรวจสอบทุกจุดแบบราสเตอร์สแกน (Raster Scan) คือการสแกนจากซ้ายไปขวา จากบนลงล่าง จนกระทั่งพบจุดดำจุดแรก จะส่งพิกัดจุดดำที่ได้ให้ขั้นตอนต่อไปนำไปประมวลผล ซึ่งวิธีนี้จะต้องทำการตรวจสอบทุกจุด ทำให้ต้องใช้เวลาในการประมวลผลนาน

ดังนั้นในวิทยานิพนธ์นี้ จึงนำเสนอวิธีใช้หน้าต่างในการตรวจสอบหาจุดดำจุดแรก โดยการวางหน้าต่าง แล้วตรวจสอบเพียงบางตำแหน่งในหน้าต่าง ทำให้สามารถตรวจสอบหาจุดดำจุดแรกได้อย่างรวดเร็วกว่าวิธีตรวจสอบทุกจุด

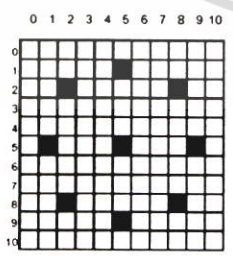
วิทยานิพนธ์นี้ ได้นำเสนอหน้าต่างที่ได้ออกแบบไว้ ทั้งสิ้น 11 แบบ ดังรูปที่ 3.1



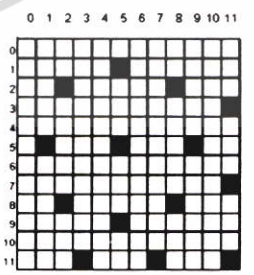
หน้าต่างแบบ A (18x18พิกเซล)



หน้าต่างแบบ B (18x18พิกเซล)

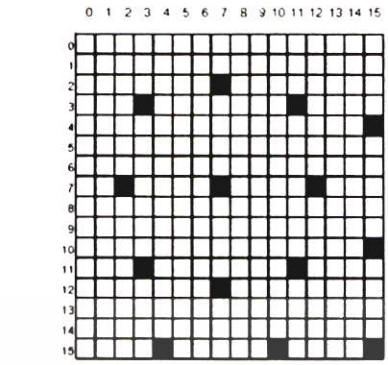


หน้าต่างแบบ C (11x11พิกเซล)

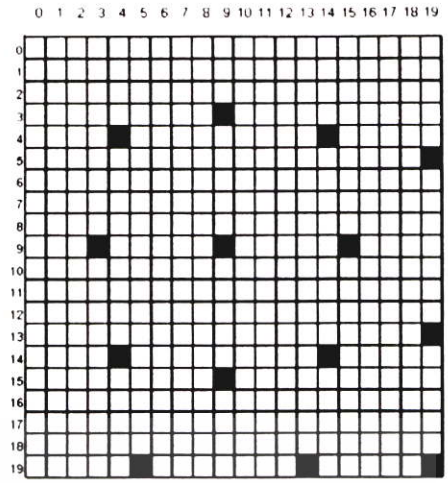


หน้าต่างแบบ D (12x12พิกเซล)

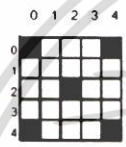
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



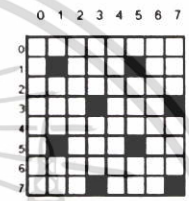
หน้าต่างแบบ E (16x16พิกเซล)



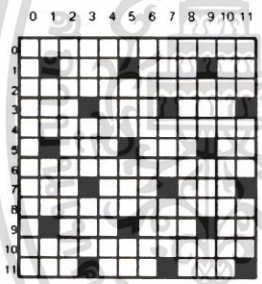
หน้าต่างแบบ F (20x20พิกเซล)



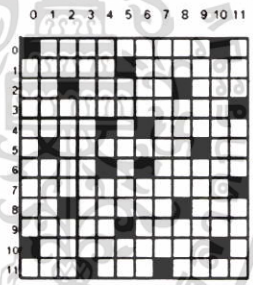
หน้าต่างแบบ G (5x5พิกเซล)



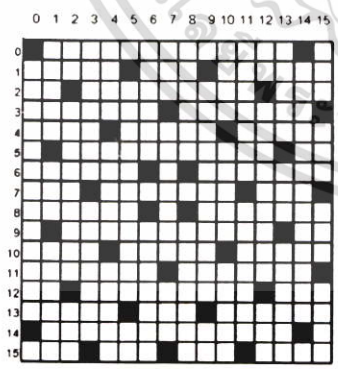
หน้าต่างแบบ H (8x8พิกเซล)



หน้าต่างแบบ I (12x12พิกเซล)



หน้าต่างแบบ J (12x12พิกเซล)



หน้าต่างแบบ K (16x16พิกเซล)

รูปที่ 3.1 หน้าต่างแบบ A-K

รูปที่ 3.1 แสดงขนาดของหน้าต่าง และตำแหน่งภายในหน้าต่าง ที่ใช้ในการตรวจสอบหาจุดดำของหน้าต่างแบบ A-K สิ่งที่น่าสนใจคือ ตำแหน่งที่ใช้ในการตรวจสอบ เพราะมีผลโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับความถูกต้อง เช่นหน้าต่างแบบ A มีตำแหน่งที่ใช้ในการตรวจสอบอยู่เป็นจำนวนมาก และมีการกระจายตัวครอบคลุมพื้นที่ของหน้าต่างได้ดีกว่า เมื่อเทียบกับหน้าต่างแบบ F ที่มีตำแหน่งที่ใช้ในการตรวจสอบที่น้อยกว่า ถึงแม้จะมีการกระจายตัวที่สมดุลย์ แต่ว่าตำแหน่งที่ใช้ตรวจสอบอยู่ห่างกันมากทำให้อาจตรวจจับตัวอักษรบางตัวไม่ได้

ดังนั้น การตรวจสอบหาจุดดำด้วยการใช้หน้าต่างแบบ A จะมีประสิทธิภาพทางด้านความถูกต้องมากกว่าการใช้หน้าต่างแบบ F

ตารางที่ 3.1 คุณสมบัติของหน้าต่างแบบ A-K

แบบหน้าต่าง	ขนาดหน้าต่าง	จุดทั้งหมด	จุดที่ใช้ตรวจสอบ	%จุดตรวจสอบ ต่อ 1 หน้าต่าง
A	18x18	324	52	16.049
B	18x18	324	32	9.877
C	11x11	121	9	7.438
D	12x12	144	14	9.72
E	16x16	256	14	5.469
F	20x20	400	14	3.5
G	5x5	25	5	20
H	8x8	64	8	12.5
I	12x12	144	18	12.5
J	12x12	144	20	13.889
K	16x16	256	34	13.281

ตารางที่ 3.1 คุณสมบัติของหน้าต่างแบบ A-K สิ่งที่น่าสนใจคือ ขนาดของหน้าต่าง และเปอร์เซ็นต์ของตำแหน่งที่ใช้ในการตรวจสอบ เพราะมีผลโดยตรงกับความเร็วในการทำงาน เช่น หน้าต่างแบบ A มีจำนวนจุดที่ใช้ตรวจสอบ 16.049 เปอร์เซ็นต์ ซึ่งมากกว่าหน้าต่างแบบ F ที่มีจำนวนจุดที่ใช้ตรวจสอบเพียง 3.5 เปอร์เซ็นต์ ดังนั้น การใช้หน้าต่างแบบ F จึงสามารถทำงานได้รวดเร็วกว่าการใช้หน้าต่างแบบ A

3.2 เปรียบเทียบการหาจุดดำจุดแรกด้วยการตรวจสอบทุกจุด และการใช้หน้าต่างแบบต่างๆ

หัวข้อนี้เป็นการเปรียบเทียบการหาจุดดำจุดแรกด้วยการตรวจสอบทุกจุด และการใช้หน้าต่างแบบต่างๆ เพื่อเปรียบเทียบหาวิธีที่ดีที่สุด ในการตรวจหาจุดดำจุดแรกได้อย่างถูกต้องและรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองได้ทำการจับเวลาโดยแบ่งเป็น 2 ช่วง เวลาในช่วงที่ 1 คือ ขั้นตอนการทำ Block Extraction ด้วยวิธีการหาจุดเริ่มต้นของการประมวลผลด้วยวิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ A-K และเวลาในช่วงที่ 2 คือ ขั้นตอนการทำ Multi-Column Block Detection and Segmentation

การทดลองทำบนเครื่อง Sempron 2400 แรม 960 MB ภายใต้ระบบปฏิบัติการ WindowXP SP2 ใช้โปรแกรมภาษา Visual Basic.Net 2003 เอกสารขนาด A4 สแกนที่ความละเอียด 300 dpi ภาพเอกสารไม่มีกรอบ ภาพเอกสารที่ใช้ในการจับเวลาแสดงในรูปที่ 3.2 (ก) เวลาที่ใช้ในการทดลองแสดงในตารางที่ 3.2

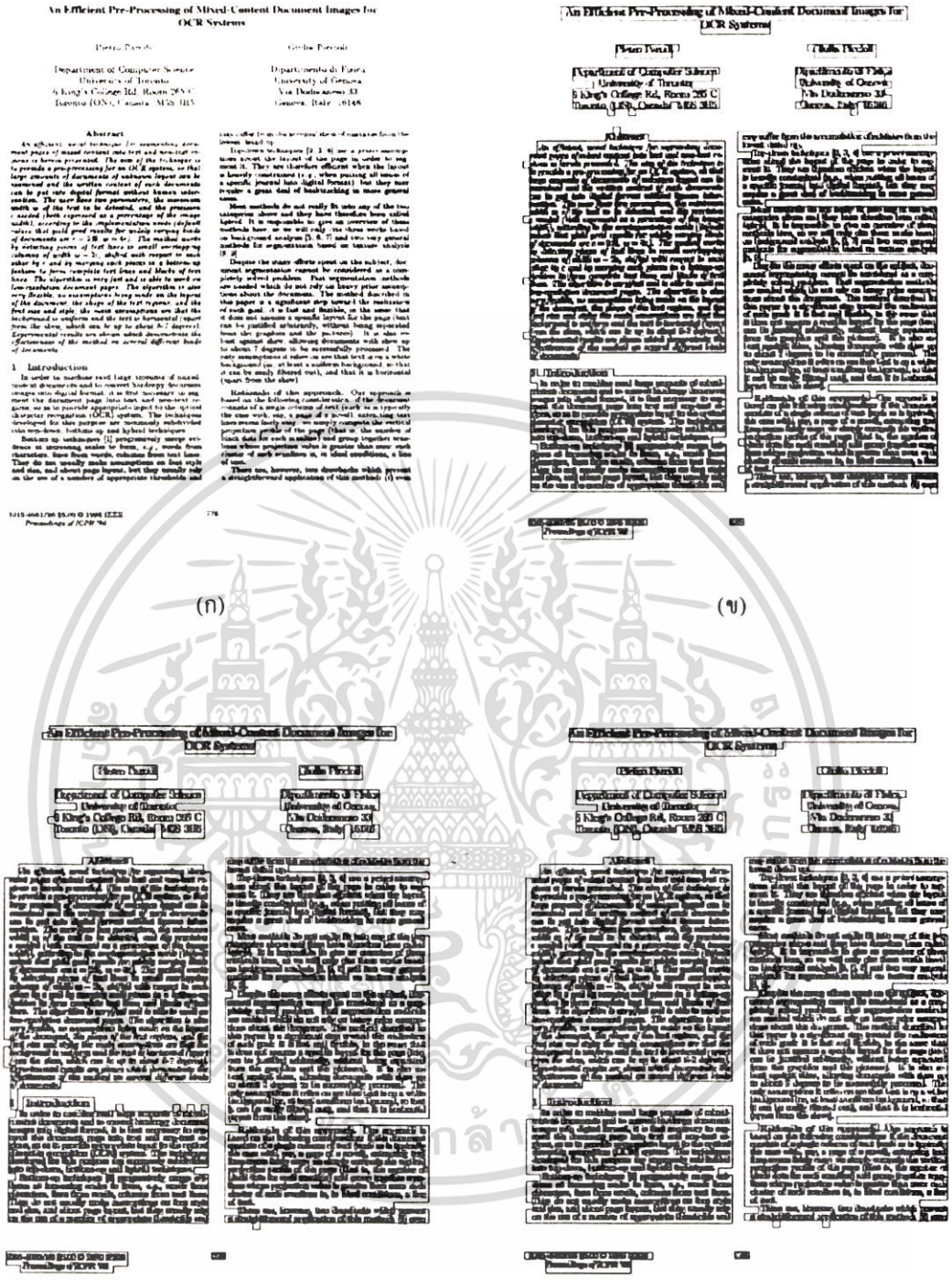
ตารางที่ 3.2 เปรียบเทียบเวลาการทำงาน วิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ A-K

วิธี	%จุดตรวจสอบ ต่อ 1 หน้าต่าง	เวลาในช่วงที่1 (วินาที)	เวลาในช่วงที่2 (วินาที)	เวลารวม (วินาที)
ตรวจสอบทุกจุด	100	18.906	16.937	35.843
หน้าต่างแบบ A	16.049	9.218	15.859	25.78
หน้าต่างแบบ B	9.877	9.171	16.0	25.171
หน้าต่างแบบ C	7.438	8.843	16.375	25.218
หน้าต่างแบบ D	9.72	9.31	16.218	25.250
หน้าต่างแบบ E	5.469	8.859	16.93	24.953
หน้าต่างแบบ F	3.5	8.515	15.921	24.437
หน้าต่างแบบ G	20	10.250	16.546	26.796
หน้าต่างแบบ H	12.5	9.703	16.718	26.421
หน้าต่างแบบ I	12.5	9.640	16.859	26.5
หน้าต่างแบบ J	13.889	9.812	17.109	26.921
หน้าต่างแบบ K	13.281	9.843	17.62	26.906

ตารางที่ 3.2 แสดงเวลาการทำงานในขั้นตอน Block Extraction (เวลาในช่วงที่1) โดยใช้การหาจุดเริ่มต้นของบล็อกด้วยหน้าต่างแบบต่างๆ หน้าต่างทั้ง 11 แบบใช้เวลา 8.515 – 10.25 วินาที ซึ่งรวดเร็วกว่าการหาจุดเริ่มต้นของบล็อกด้วยการตรวจสอบทุกจุด ซึ่งใช้เวลา 18.906 วินาที

ส่วนเวลาการทำงานในขั้นตอน Multi-Column Block Detection and Segmentation (เวลาในช่วงที่2) แต่ละวิธีใช้เวลาใกล้เคียงกันเพราะในขั้นตอนนี้ ใช้อัลกอริทึมเดียวกันในการทำงานส่วนเวลาที่แตกต่างกันเล็กน้อยนั้น เกิดจากตำแหน่งเริ่มต้นของข้อมูลที่แต่ละวิธีพบตำแหน่งเริ่มต้นต่างกันไปทำให้การวางหน้าต่าง 32x32 ต่างกัน เช่นในรูปที่ 3.2 (ข), (ค), (ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



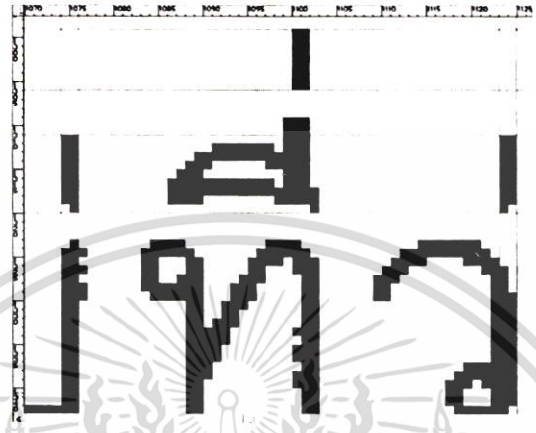
(ก) (ข)

(ค) (ง)

รูปที่ 3.2 รูปที่ใช้ในการทดลองจับเวลา, (ก) รูปต้นฉบับขนาด 2300x3000พิกเซล
(ข) รูปที่ทำ Block Extraction โดยการหาตำแหน่งเริ่มต้นด้วยหน้าต่างแบบ C
(ค) รูปที่ทำ Block Extraction โดยการหาตำแหน่งเริ่มต้นด้วยหน้าต่างแบบ H
(ง) รูปที่ทำ Block Extraction โดยการหาตำแหน่งเริ่มต้นด้วยการตรวจสอบทุกจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการนำหน้าต่างทั้ง 11 แบบ มาทดสอบกับภาพอักษรภาษาไทย พบว่าหน้าต่างแบบ G–K สามารถทำงานได้อย่างถูกต้อง ส่วนหน้าต่างแบบ A–F ยังมีการทำงานที่ผิดพลาดอยู่บ้าง คือ ไม่สามารถตรวจจับอักษรบางตัวได้ เนื่องจากช่องว่างภายในหน้าต่างที่ไม่ได้ทำการตรวจสอบ มีขนาดใหญ่กว่าตัวอักษรบางตัว ดังรูปที่ 3.3



รูปที่ 3.3 สระไม้เอก มีขนาด 2x9 พิกเซล และ 2x7 พิกเซล ทำให้หน้าต่าง A–F ทำงานผิดพลาดได้

ผลการทดลองแสดงให้เห็นว่า การตรวจสอบหาจุดดำจุดแรกด้วยการใช้หน้าต่างทำได้รวดเร็วกว่าการตรวจสอบทุกจุด และในด้านความถูกต้องหน้าต่างแบบ G-K สามารถตรวจจับตัวอักษร และรูปภาพได้ทั้งหมด และเมื่อนำหน้าต่างแบบ H, I มาวางเรียงกับแบบ Raster Scan จะพบว่าหน้าต่างแบบ H, I มีช่องว่างระหว่างตำแหน่งที่ใช้ในการตรวจสอบน้อยที่สุดเพียง 3x3 พิกเซลเท่านั้น

จึงสามารถสรุปได้ว่า การหาจุดเริ่มต้นของการประมวลผลด้วยหน้าต่างแบบ H, I สามารถทำงานได้อย่างถูกต้องและมีประสิทธิภาพทางด้านความรวดเร็ว มากกว่าการตรวจสอบทุกจุด

3.3 เปรียบเทียบการหาจุดดำจุดแรกด้วยการตรวจสอบทุกจุด และการใช้หน้าต่างแบบ I

การศึกษาในหัวข้อ 3.2 ทำให้ทราบได้ว่า การหาจุดดำจุดแรกด้วยวิธีใช้หน้าต่างแบบ H, I มีความเหมาะสมที่สุดทั้งในด้านของการออกแบบ คือแต่ละจุดตรวจสอบมีการกระจายตัวอย่างสมดุลย์ และในด้านของการทำงานคือมีความถูกต้อง และความรวดเร็วในการทำงาน

ในหัวข้อนี้ จึงทำการทดลองการตรวจสอบหาจุดดำจุดแรก ด้วยวิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ I (เพราะหน้าต่างแบบ I มีขนาดหน้าต่างเหมาะที่จะนำไปใช้ในวิธี Dilation1PX มากกว่าหน้าต่างแบบ H) เพื่อเปรียบเทียบความถูกต้อง และความรวดเร็วในการทำงาน โดยได้นำภาพเอกสารไบนารี่จำนวน 17 ตัวอย่าง มาทดลองจับเวลาที่ใช้ในการประมวลผล และตรวจสอบความถูกต้อง โดยนับจำนวนบล็อกที่โปรแกรมคำนวณได้ นำมาเปรียบเทียบกับของจริงที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองได้ทำการจับเวลาโดยแบ่งเป็น 2 ช่วง เวลาในช่วงที่ 1 คือขั้นตอนการทำ Block Extraction ด้วยวิธีการหาจุดเริ่มต้นของการประมวลผลด้วยวิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ I และเวลาในช่วงที่ 2 คือ ขั้นตอนการทำ Multi-Column Block Detection and Segmentation

การทดลองทำบนเครื่อง Sempron 2400 แรม 960 MB ภายใต้ระบบปฏิบัติการ WindowXP SP2 ใช้โปรแกรมภาษา Visual Basic.Net 2003 เอกสารขนาด A4 สแกนที่ความละเอียด 300 dpi ภาพเอกสารไม่มีกรอบ เวลาที่ใช้ในการทดลองแสดงในตารางที่ 3.3

ตารางที่ 3.3 เปรียบเทียบเวลาการทำงานวิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ I

ตัวอย่างที่	ขนาดภาพ (พิกเซล)	เวลาในช่วงที่ 1 (วินาที)		เวลาในช่วงที่ 2 (วินาที)	เวลารวม (วินาที)
1	2400x3300	ตรวจสอบทุกจุด	16.953	10.875	27.828
		หน้าต่างแบบ I	7.953	9.796	17.750
2	2400x3300	ตรวจสอบทุกจุด	19.296	16.171	35.468
		หน้าต่างแบบ I	10.296	15.31	25.328
3	2400x3300	ตรวจสอบทุกจุด	17.937	12.953	30.890
		หน้าต่างแบบ I	9.437	12.937	22.375
4	2400x3300	ตรวจสอบทุกจุด	17.875	14.109	31.984
		หน้าต่างแบบ I	9.375	14.203	23.578
5	2800x3500	ตรวจสอบทุกจุด	22.718	16.78	38.796
		หน้าต่างแบบ I	12.796	18.93	30.890
6	2500x3350	ตรวจสอบทุกจุด	18.734	11.843	30.578
		หน้าต่างแบบ I	10.859	12.500	23.359
7	2400x3300	ตรวจสอบทุกจุด	14.984	6.515	21.500
		หน้าต่างแบบ I	6.171	6.609	12.781
8	2700x3700	ตรวจสอบทุกจุด	28.859	19.0	47.859
		หน้าต่างแบบ I	12.421	19.828	32.250
9	2400x3300	ตรวจสอบทุกจุด	17.890	13.734	31.625
		หน้าต่างแบบ I	8.984	12.421	21.406
10	2400x3300	ตรวจสอบทุกจุด	16.859	33.203	50.62
		หน้าต่างแบบ I	7.937	37.62	45.0

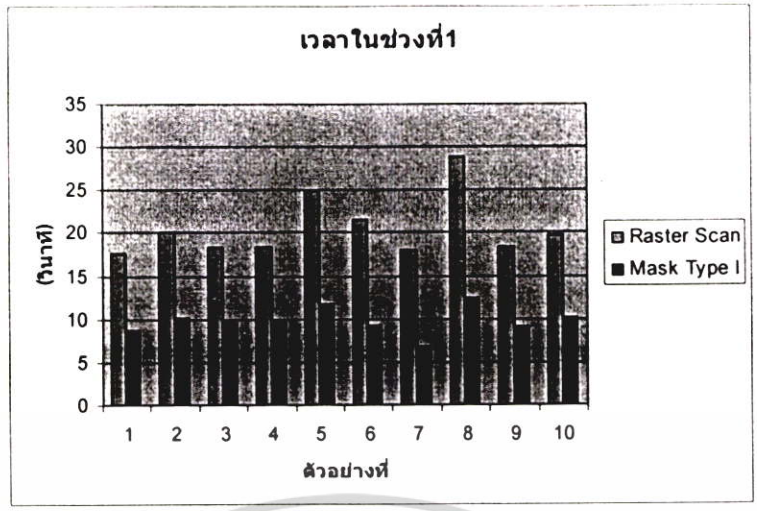
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 (ต่อ)

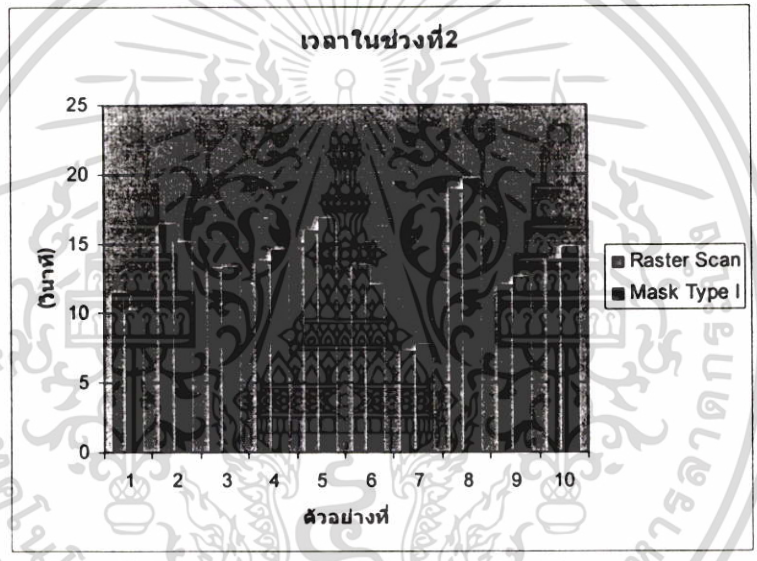
ตัวอย่างที่	ขนาดภาพ (พิกเซล)	เวลาในช่วงที่ 1 (วินาที)		เวลาในช่วงที่ 2 (วินาที)	เวลารวม (วินาที)
11	2700x3500	ตรวจสอบทุกจุด	24.593	15.62	39.656
		หน้าต่างแบบ I	10.0	13.234	23.234
12	2750x3300	ตรวจสอบทุกจุด	20.78	13.875	33.953
		หน้าต่างแบบ I	11.359	13.578	24.937
13	2800x3300	ตรวจสอบทุกจุด	26.468	17.843	44.312
		หน้าต่างแบบ I	12.31	17.265	29.296
14	2400x3300	ตรวจสอบทุกจุด	17.531	56.375	13.906
		หน้าต่างแบบ I	9.859	59.218	9.78
15	2300x3000	ตรวจสอบทุกจุด	15.500	15.359	30.859
		หน้าต่างแบบ I	7.750	15.93	22.843
16	2400x3300	ตรวจสอบทุกจุด	19.796	15.140	34.937
		หน้าต่างแบบ I	10.515	15.500	26.15
17	2400x3300	ตรวจสอบทุกจุด	16.796	9.437	26.234
		หน้าต่างแบบ I	8.0	11.312	19.312

ผลการทดลอง การตรวจสอบหาจุดดำจุดแรก ด้วยวิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ I สามารถทำงานได้อย่างถูกต้อง แต่วิธีใช้หน้าต่างแบบ I สามารถทำงานได้อย่างรวดเร็วกว่าวิธีตรวจสอบทุกจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูปที่ 3.4 กราฟแสดงเวลาการทำงานของวิธีตรวจสอบทุกจุดและวิธีใช้หน้าต่างแบบ I จากตัวอย่างที่ 1-10 ในตารางที่ 3.3

3.4 ความสัมพันธ์ของการขยายขอบภาพให้ใหญ่ขึ้น กับการหาจุดดำจุดแรกด้วยการตรวจสอบทุกจุด และการใช้หน้าต่างแบบ I

กระบวนการแบ่งส่วนประกอบภาพเอกสารไบนารี่ การทำงานในขั้นตอนที่ 1 หรือ Block Extraction ประกอบไปด้วย 2 ขั้นตอนย่อย คือ

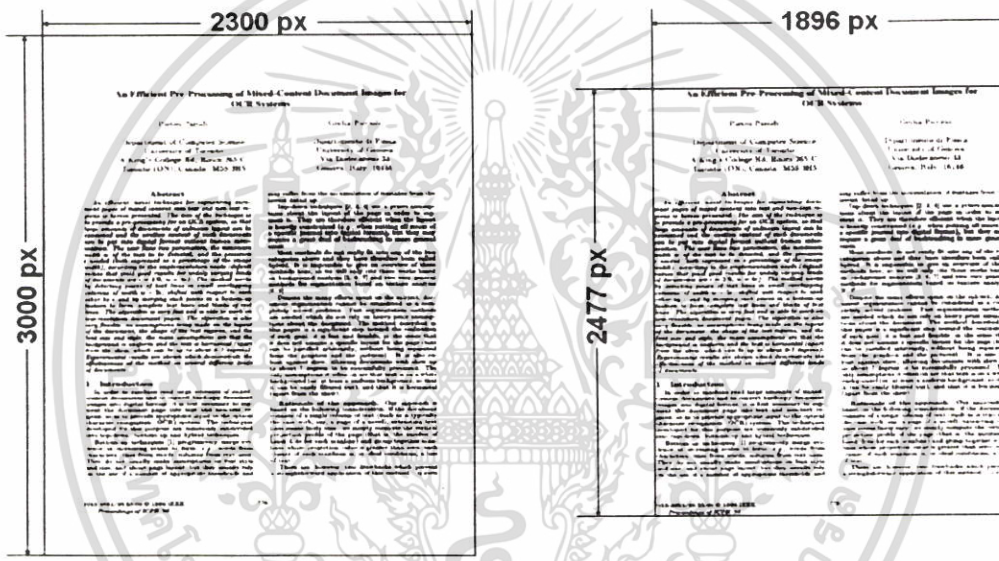
1. การหาจุดเริ่มต้นของการประมวลผล คือการหาจุดดำจุดแรก (ด้วยการตรวจสอบทุกจุด หรือหน้าต่างแบบ I)

2. การเดินรอบขอบของบล็อกด้วยหน้าต่าง 32x32 พิกเซล เพื่อหาขอบเขตของบล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น เพื่อที่จะเปรียบเทียบความรวดเร็วในการตรวจสอบหาจุดดำจุดแรก ด้วยวิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ I ให้เห็นชัดเจนยิ่งขึ้น จึงได้ทำการขยายขอบของภาพทดสอบให้ใหญ่ขึ้น เพื่อให้การตรวจสอบหาจุดดำจุดแรกต้องตรวจสอบจำนวนจุดที่มากขึ้น ดังนั้นเวลาที่ได้จากการทดลองที่เพิ่มขึ้นจะเป็นเวลาที่ใช้ในการตรวจสอบหาจุดดำจุดแรก แต่เวลาที่ใช้ในการเดินรอบขอบของบล็อกด้วยหน้าต่าง 32x32 พิกเซล ไม่ควรจะเพิ่มขึ้น

การทดลอง ทำบนเครื่อง Sempron 2400 แรม 960 MB ภายใต้ระบบปฏิบัติการ WindowXP SP2 ใช้โปรแกรมภาษา Visual Basic.Net 2003 เอกสารขนาด A4 สแกนที่ความละเอียด 300 dpi ภาพเอกสารไม่มีกรอบ ภาพเอกสารที่ใช้ในการจับเวลาแสดงในรูปที่ 3.5 เวลาที่ใช้ในการทดลองแสดงในตารางที่ 3.4



(ก) (ข)

รูปที่ 3.5 ตัวอย่างภาพที่ใช้ทดสอบ, (ก) ขนาดภาพ 2300 x 3000พิกเซล
(ข) ขนาดกรอบตัวอักษรโดยประมาณ 1896 x 2477พิกเซล

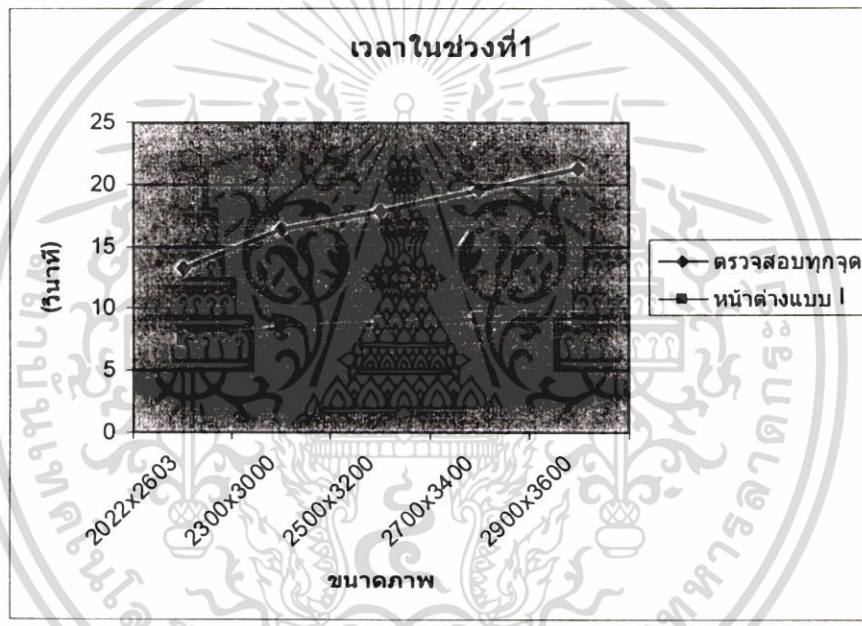
ตารางที่ 3.4 เวลาของการทดลองการขยายขอบภาพ

ขนาดภาพ	ขนาดกรอบตัวอักษร (พิกเซล)	เวลาในช่วงที่1 (วินาที)		เวลาในช่วงที่2 (วินาที)	เวลารวม (วินาที)
2022x2603	1896x2477	ตรวจสอบทุกจุด	13.234	15.265	28.50
		หน้าต่างแบบ I	7.515	15.0	22.515
2300x3000	1896x2477	ตรวจสอบทุกจุด	16.375	16.281	32.656
		หน้าต่างแบบ I	8.500	16.937	25.437

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 (ต่อ)

ขนาดภาพ	ขนาดกรอบตัวอักษร (พิกเซล)	เวลาในช่วงที่1 (วินาที)		เวลาในช่วงที่2 (วินาที)	เวลารวม (วินาที)
2500x3200	1896x2477	ตรวจสอบทุกจุด	17.843	16.953	34.796
		หน้าต่างแบบ I	8.796	17.687	26.484
2700x3400	1896x2477	ตรวจสอบทุกจุด	19.531	15.515	35.46
		หน้าต่างแบบ I	9.218	17.93	26.312
2900x3600	1896x2477	ตรวจสอบทุกจุด	21.296	16.328	37.625
		หน้าต่างแบบ I	9.234	17.000	26.234



รูปที่ 3.6 กราฟเวลาการทำงานในช่วงที่ 1 (จากตารางที่3.5)

การเปลี่ยนแปลงขนาดของขอบภาพ มีผลโดยตรงกับเวลาที่ใช้ในการตรวจสอบหาจุดคำจุดแรก ผลการทดลองในตารางที่ 3.4 จะเห็นได้ว่า วิธีใช้หน้าต่างแบบ I เวลาเพิ่มขึ้นน้อยมาก ต่างกับวิธีตรวจสอบทุกจุด ที่เวลาเพิ่มขึ้นอย่างมาก

จากการตรวจสอบความถูกต้องของผลลัพธ์ที่ได้จากวิธีตรวจสอบทุกจุด และวิธีใช้หน้าต่างแบบ I มีประสิทธิภาพทางด้านความถูกต้องเหมือนกัน แต่วิธีใช้หน้าต่างแบบ I มีประสิทธิภาพทางด้านความรวดเร็วในการประมวลผลมากกว่าวิธีการตรวจสอบทุกจุด

จึงสามารถสรุปได้ว่า การหาจุดเริ่มต้นของการประมวลผล ควรใช้หน้าต่างแบบ I ซึ่งสามารถทำงานได้อย่างถูกต้อง และรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
แนวทางที่ใช้ในงานวิจัย

4.1 แนวคิดการแบ่งส่วนประกอบภาพเอกสาร

จากการศึกษาวิธีการแบ่งส่วนประกอบภาพเอกสาร ด้วยอัลกอริทึมของ Kruatrachue, B. และ Suthaphan, P. [1] พบว่าการแบ่งส่วนประกอบภาพเอกสารสามารถทำได้อย่างถูกต้อง และรวดเร็ว แต่มีบางขั้นตอนที่สามารถปรับปรุงให้ทำงานได้เร็วขึ้นอีก เช่น

1. ขั้นตอน Block Extraction การตรวจสอบหาจุดดำจุดแรกด้วยการตรวจสอบทุกจุดแบบ Raster Scan เนื่องจากเป็นการตรวจสอบทุกจุดจึงทำให้ช้า

2. ขั้นตอน Block Extraction เมื่อเจอจุดดำจุดแรกแล้ว ต่อไปคือการหาขอบเขตของบล็อกด้วยการใช้หน้าต่าง 32x32พิกเซล เดินรอบขอบของบล็อก ซึ่งการใช้หน้าต่าง 32x32 พิกเซล จะทำการตรวจสอบภายในหน้าต่างว่ามีจุดดำมากกว่า 10 จุดหรือไม่ ถ้ามากกว่าถือว่าเป็นหน้าต่างที่มีข้อมูล แต่ถ้าไม่ใช่ถือว่าเป็นหน้าต่างที่ไม่มีข้อมูล ดังนั้นในการเดินรอบขอบด้วยหน้าต่าง 32x32 พิกเซลนี้ เมื่อหน้าต่างเดินไปในตำแหน่งที่ไม่มีข้อมูลหน้าต่าง 32x32 พิกเซล ต้องตรวจสอบเป็นจำนวนถึง 32x32=1024 ครั้ง และในการเดินรอบขอบของบล็อก โอกาสที่จะเจอหน้าต่างที่ไม่มีข้อมูล มีโอกาสเป็นไปได้มากทีเดียว

จากการทำงานทั้ง 2 ขั้นตอนที่กล่าวมา จึงเกิดแนวความคิดที่จะตรวจสอบหาจุดดำด้วยการใช้หน้าต่าง แล้วนำกลุ่มจุดดำที่เป็นตัวอักษรมาสร้างรูปขนาดย่อ (Optimum Picture) ที่มีขนาดเล็กกว่ารูปต้นฉบับประมาณ 144 เท่า แต่ยังคงโครงสร้างของรูปต้นฉบับไว้อย่างถูกต้อง แล้วกระทำการใดๆกับ Optimum Picture นี้แทน ก็จะทำให้การทำงานทำได้รวดเร็วขึ้น

(ก) ขนาด 2400x3300 พิกเซล

(ข) ขนาด 200x275 พิกเซล

รูปที่ 4.1 รูปขนาดย่อ (Optimum Picture), (ก) รูปต้นฉบับ, (ข) ตัวอย่าง Optimum Picture

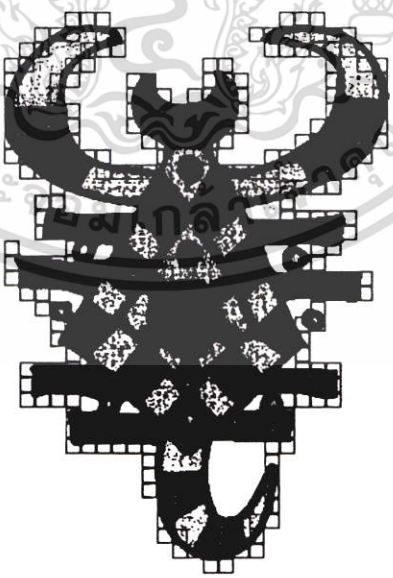
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 (ข) เป็นตัวอย่าง Optimum Picture มีขนาด 200x275 พิกเซล ซึ่งเล็กกว่ารูปที่ 4.1 (ก) ที่เป็นรูปต้นที่มีขนาด 2400x3300 พิกเซล ซึ่งขนาดที่เล็กของ Optimum Picture นี้ ทำให้ Optimum Picture สามารถเก็บโครงสร้างของภาพเอกสารไว้ได้ แต่ไม่สามารถเก็บรายละเอียดของภาพเอกสารได้ ดังนั้น Optimum Picture จึงสามารถนำมาใช้ได้กับ วิธีที่เกี่ยวกับโครงสร้างของภาพเอกสารเท่านั้น

4.2 การหาขอบเขตของตัวอักษร และรูป

การหาขอบเขตของตัวอักษร และรูป ที่ใช้ในวิทยานิพนธ์นี้ เริ่มจากการตรวจสอบหาจุดดำจุดแรกแบบ Raster Scan ด้วยการใช้หน้าต่างแบบ I เมื่อตรวจพบจุดดำแล้ว ให้ทำการตรวจสอบหาจุดดำจุดแรกภายในหน้าต่าง แล้วเดินรอบขอบของจุดดำด้วย Chain Codes 1 พิกเซล เพื่อนำขอบเขตของกลุ่มดำมาใช้ในการกำหนดว่า กลุ่มดำที่พบนี้เป็นตัวอักษร รูป หรือเป็นเพียงสัญญาณรบกวน (Noise)

1. กลุ่มดำที่พบเป็นรูป ถ้ากลุ่มดำที่พบมีขนาดกว้างหรือยาวมากกว่า 80 พิกเซล ให้เดินรอบกลุ่มดำด้วยหน้าต่างแบบ I เพื่อหาขอบเขตและรูปร่างของรูป ดูในรูปที่ 4.2
2. กลุ่มดำเป็นตัวอักษร ถ้ากลุ่มดำที่พบไม่ใช่รูปภาพ และมีขนาดกว้างหรือยาวมากกว่า 4 พิกเซล ให้เก็บตัวอักษรนี้ไว้
3. กลุ่มดำนี้เป็นสัญญาณรบกวน ถ้ากลุ่มดำที่พบไม่ใช่รูปและไม่ใช่ตัวอักษร ให้ลบสัญญาณรบกวนนี้ออกจากภาพต้นฉบับ



รูปที่ 4.2 ขอบเขตของรูปที่หาได้โดยการใช้หน้าต่างแบบ I

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การนำ Optimum Picture มาใช้ในการหาขอบเขตของบล็อกตัวอักษร

4.3.1 การสร้าง Optimum Picture

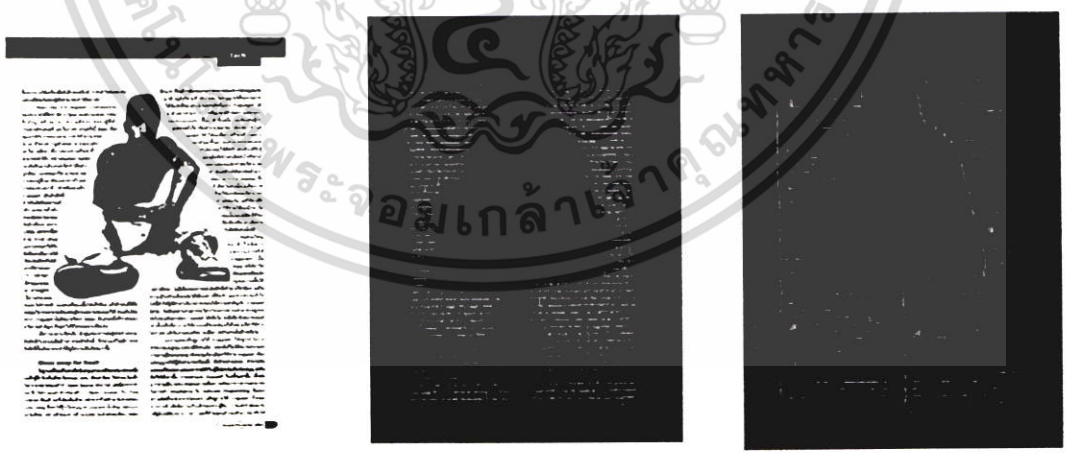
นำขอบเขตของตัวอักษรมาวาดลง Optimum Picture โดยกำหนดขนาด 12x12 พิกเซลของขนาดจริง นำมาย่อขนาดลงเหลือเพียง 1 พิกเซลใน Optimum Picture ทำให้ Optimum Picture มีความกว้าง และความสูง เล็กลงกว่ารูปต้นฉบับประมาณด้านละ 12 เท่า จึงทำให้ Optimum Picture มีขนาดเล็กลงจากรูปต้นฉบับประมาณ 144 เท่า

4.3.2 การหาขอบเขตของบล็อกตัวอักษรจาก Optimum Picture ด้วยวิธี Dilation1PX

วิธีการนี้ เหมือนกับการเบลอภาพทั้งในแนวตั้ง และแนวนอน ทำให้ตัวอักษรที่อยู่ใกล้ๆกัน ถูกรวมเข้าด้วยกัน การเบลอภาพทำได้โดยใช้ Mask 3x3 พิกเซล ดังรูปที่ 4.3 ซึ่งใน Optimum Picture 1 พิกเซล แทนข้อมูลในรูปต้นฉบับขนาด 12x12 พิกเซล ดังนั้น การทำ Dilation ด้วย Mask 3x3 พิกเซล จึงเหมือนกับการเบลอภาพด้านละ 12 พิกเซล จึงทำให้ตัวอักษรที่อยู่ใกล้ๆกัน รวมกันเป็นบล็อก เช่นรูปที่ 4.4 จากนั้นทำการเดินรอบขอบกลุ่มตัวอักษรด้วย Chain Code 1 พิกเซล ก็จะทำให้ได้ขอบเขตของบล็อกตัวอักษร

1	1	1
1	1	1
1	1	1

รูปที่ 4.3 Mask 3x3 พิกเซล ใช้ในการทำ Dilation



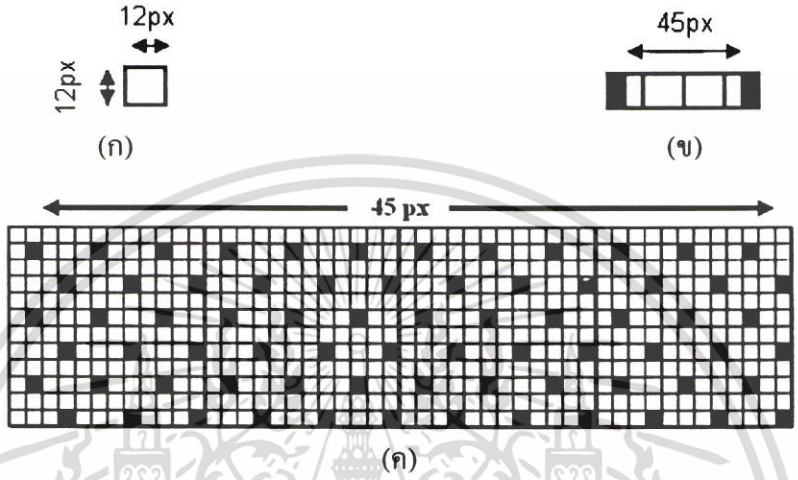
(ก) (ข) (ค)

รูปที่ 4.4 Optimum Picture ที่ใช้ในการหาขอบเขตของบล็อกตัวอักษรด้วยวิธี Dilation1PX
(ก) รูปต้นฉบับ, (ข) Optimum Picture ที่เกิดจากการวาดขอบของตัวอักษร
(ค) นำ Optimum Picture ในรูป (ข) มาทำ Dilation 3x3พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 วิเคราะห์บล็อกตัวอักษรที่ได้จากวิธี Dilation1PX

การหาขอบเขตของบล็อกตัวอักษรด้วยวิธี Dilation1PX สามารถทำการหาขอบเขตของบล็อกตัวอักษรได้ดีถ้าระยะห่างระหว่าง 2 คอลัมน์ มีมากกว่า 45 พิกเซล แต่ถ้า ระยะห่างระหว่าง 2 คอลัมน์ มีน้อยกว่า 45 พิกเซล วิธี Dilation1PX ก็มีโอกาสทำงานผิดพลาดโดยรวมทั้ง 2 คอลัมน์นี้เข้าเป็นบล็อกเดียวกัน ดังรูปที่ 4.5

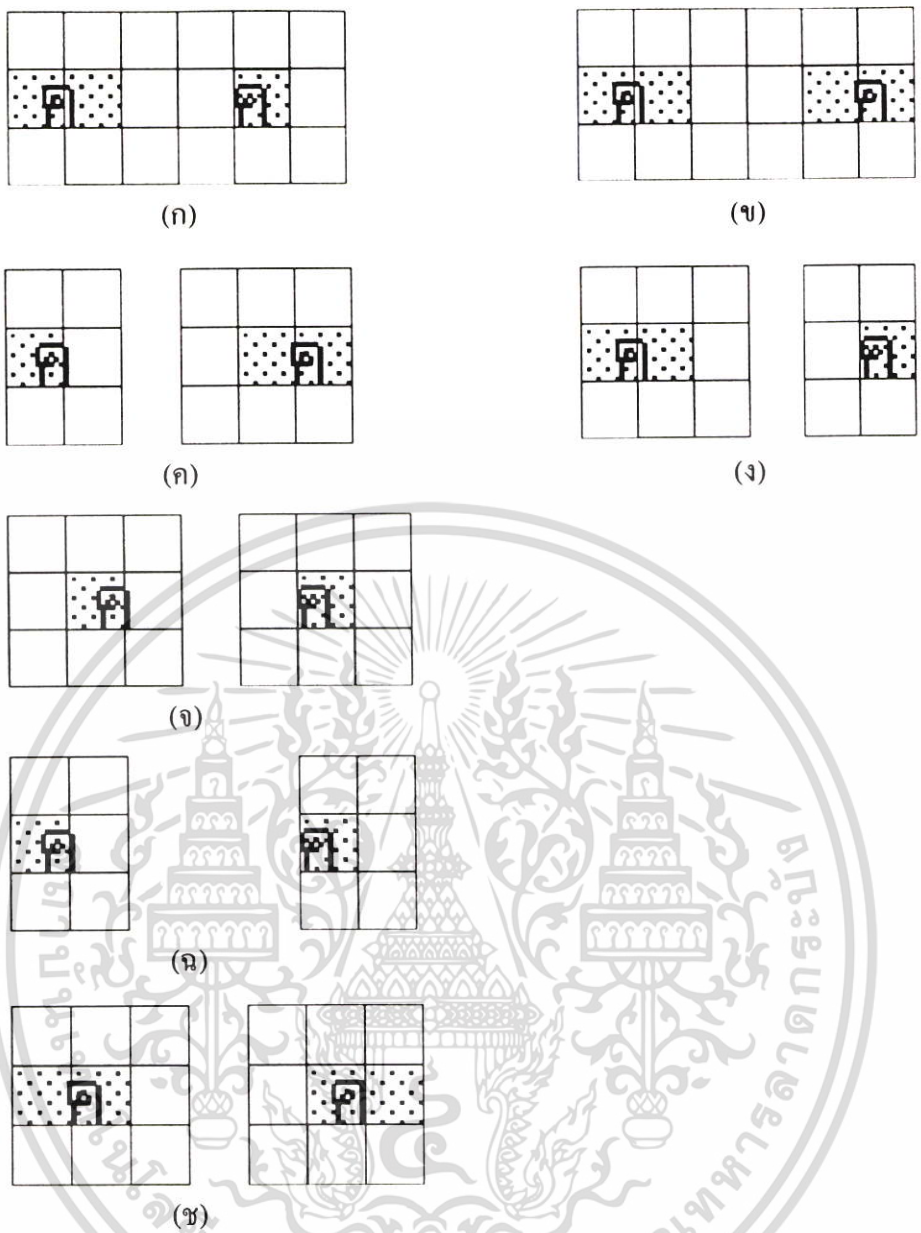


รูปที่ 4.5 ความสามารถในการแยกบล็อกออกจากกันของ วิธี Dilation1PX
(ก) หน้าต่าง 12x12 พิกเซล, (ข) ระยะห่างระหว่างบล็อกน้อยกว่าหรือเท่ากับ 45 พิกเซล วิธี Dilation1PX มีโอกาสที่จะไม่สามารถแยกบล็อกออกจากกันได้
(ค) ขยายรูป (ข) ให้เห็นถึงจุดตรวจสอบของหน้าต่างแบบ I

รูปที่ 4.5 แสดงให้เห็นถึงโอกาสการทำงานผิดพลาดของวิธี Dilation1PX ซึ่งจะเห็นได้ว่ามีโอกาสทำงานผิดพลาดได้เมื่อ 2 คอลัมน์ ห่างกันน้อยกว่า 45 พิกเซล แต่ก็ยังมีโอกาศทำงานผิดพลาดน้อยกว่าวิธีใช้หน้าต่าง 32x32 พิกเซล[1] (ดูรูปที่ 4.6 ประกอบ)

รูปที่ 4.6 ความสามารถในการแยกบล็อกออกจากกันของหน้าต่าง 32x32 พิกเซล
(ก) หน้าต่าง 32x32 พิกเซล, (ข) ระยะห่างระหว่างบล็อกน้อยกว่าหรือเท่ากับ 62 พิกเซล วิธีการใช้หน้าต่าง 32x32พิกเซล มีโอกาสที่จะไม่สามารถแยกบล็อกออกจากกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ตัวอย่างการวางหน้าต่างที่สามารถแยกคอลัมน์ได้ และไม่ได้

(ก) ระหว่างคอลัมน์ห่างกัน 34 พิกเซล แยกคอลัมน์ไม่ได้

(ข) ระหว่างคอลัมน์ห่างกัน 45 พิกเซล แยกคอลัมน์ไม่ได้

(ค) ระหว่างคอลัมน์ห่างกัน 46 พิกเซล แยกคอลัมน์ได้

(ง) ระหว่างคอลัมน์ห่างกัน 46 พิกเซล แยกคอลัมน์ได้

(จ) ระหว่างคอลัมน์ห่างกัน 35 พิกเซล แยกคอลัมน์ได้

(ฉ), (ช) ระหว่างคอลัมน์ห่างกัน 47 พิกเซล แยกคอลัมน์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.7 สามารถสรุปได้เป็น 3 กรณีคือ

1. ถ้าระยะห่างระหว่างคอลัมน์ น้อยกว่า 35 พิกเซล ไม่สามารถแยกคอลัมน์ได้

2. ถ้าระยะห่างระหว่างคอลัมน์ มากกว่าหรือเท่ากับ 35 พิกเซล แต่น้อยกว่าหรือเท่ากับ 45 พิกเซล มีโอกาสที่จะแยกคอลัมน์ได้ และไม่ได้ เช่นรูป (ข) ไม่สามารถแยกคอลัมน์ออกจากกันได้ แต่รูป (จ) สามารถแยกคอลัมน์ออกจากกันได้

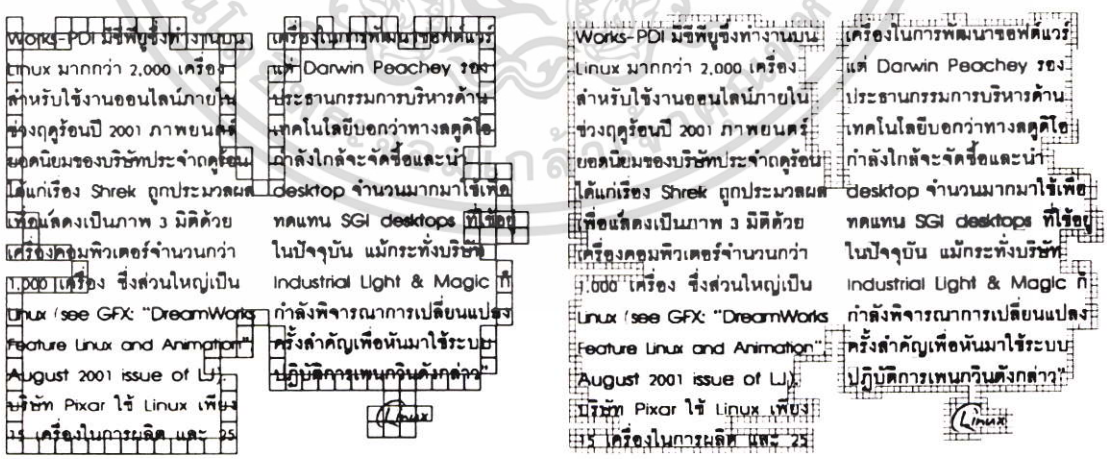
3. ถ้าระยะห่างระหว่างคอลัมน์ มากกว่า 45 พิกเซล สามารถแยกคอลัมน์ได้

(ก) (ข)

รูปที่ 4.8 ตัวอย่างวิธีต่างๆที่สามารถแยกคอลัมน์ได้ และไม่ได้

(ก) วิธีการใช้หน้าต่าง 32x32 พิกเซล ไม่สามารถแยกคอลัมน์ได้

(ข) วิธี Dilation1PX สามารถแยกคอลัมน์ได้



(ก) (ข)

รูปที่ 4.9 ทั้ง 2 วิธีไม่สามารถแยกบล็อกออกจากกันได้เมื่อบล็อกใกล้กันมากๆ

(ก) วิธีใช้หน้าต่าง 32x32 พิกเซล ไม่สามารถแยกคอลัมน์ได้

(ข) วิธี Dilation1PX ไม่สามารถแยกคอลัมน์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.9 ทั้ง 2 คอลัมน์อยู่ใกล้กันมากๆ ทำให้ทั้ง 2 วิธี รวม 2 คอลัมน์นี้ไว้เป็นบล็อกเดียวกัน ซึ่งในกรณีนี้จำเป็นต้องมีการตัดแบ่งคอลัมน์ บล็อกที่ได้ถึงจะถูกต้อง

4.4 การตัดแบ่งคอลัมน์

ในขั้นตอน การนำ Optimum Picture มาใช้ในการหาขอบเขตของบล็อกตัวอักษร เมื่อได้บล็อกของตัวอักษรแล้ว จะนำบล็อกของตัวอักษรมาตรวจสอบว่าภายในบล็อกตัวอักษรนี้ ประกอบไปด้วยกี่คอลัมน์ ถ้ามีเพียง 1 คอลัมน์ ก็ไม่ต้องทำอะไร แต่ถ้ามีมากกว่า 1 คอลัมน์ ก็จะทำขั้นตอนการตัดแบ่งคอลัมน์

การตัดแบ่งคอลัมน์ที่ใช้ในวิทยานิพนธ์ ใช้วิธีการนำขอบของตัวอักษรภายในบล็อกตัวอักษร มาทำฮิสโตแกรมในแนวตั้ง ดังรูปที่ 4.10 ซึ่งจะทำให้การทำฮิสโตแกรมทำได้อย่างรวดเร็ว เพราะเป็นการนำขอบเขตของตัวอักษรที่มีอยู่แล้วมาใช้ ไม่ต้องฮิสโตแกรมตรวจสอบทุกจุด เมื่อได้ฮิสโตแกรมมาแล้ว ให้นำฮิสโตแกรมมาทำการตรวจสอบว่ามีช่องว่างที่ไม่มีตัวอักษรอยู่เลย เป็นขนาดกว้างมากกว่า หรือเท่ากับค่าเทรคโชลด์ที่กำหนดหรือไม่ ถ้าใช่ ก็จะทำการตัดแบ่งบล็อกที่ตำแหน่งนี้

แต่ถ้าคุณต้องการตรวจสอบวินโดวส์
เอ็กซ์พี และออฟฟิศเอ็กซ์พีด้วยตัวคุณเอง
ง่ายๆ ก็สามารถทำเองหรืออัตโนมัติได้โดย

รูปที่ 4.10 ทำฮิสโตแกรมด้วยขอบของตัวอักษร

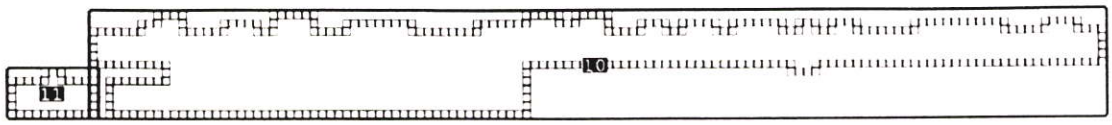
4.5 การรวมบล็อกที่ Overlap กัน

นำบล็อกของตัวอักษรทั้งหมดมาทำการตรวจสอบว่า มีขอบเขตแบบสี่เหลี่ยมของบล็อกตัวอักษรคาบเกี่ยวกัน (Overlap) อยู่หรือไม่ ถ้ามีให้รวมบล็อกทั้ง 2 เข้าเป็นบล็อกเดียวกัน ดังรูปที่ 4.11

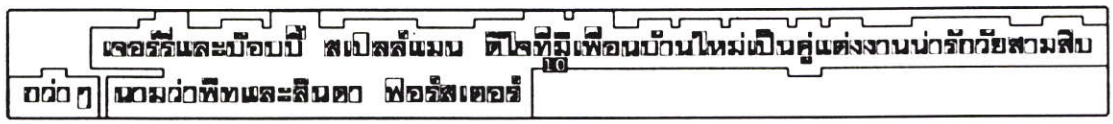
เจอร์รีและบ๊อบบี้ สเปลล์แมน ดีใจที่มีเพื่อนบ้านใหม่เป็นคู่แต่งงานน่ารักวัยสามสิบกว่าๆ นามว่าพีทและลินดา ฟอร์สเตอร์

(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข)



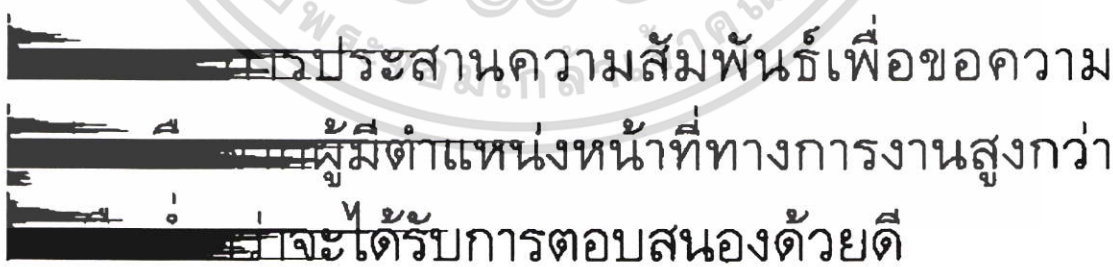
(ค)

รูปที่ 4.11 การรวมบล็อกที่ Overlap กัน (ก) รูปต้นฉบับ
(ข) บล็อกที่ Overlap กัน, (ค) บล็อกที่ Overlap กันถูกรวมเป็น Block เดียวกัน

ถ้าส่งบล็อกในรูปที่ 4.11 (ข) เข้าโปรแกรม OCR ผลลัพธ์ที่ได้คือ "เจอร์รี่และบ็อบบี่ สเปลล์แมน ดีใจที่มีเพื่อนบ้านใหม่เป็นคู่แต่งงานน่ารักวัยสามสิบ นามว่าพีทและลินดา ฟอร์สเตอร์ กว่าๆ" ซึ่งจะทำให้ความหมายผิดไป แต่ถ้าส่งบล็อกในรูปที่ 4.11 (ค) เข้าโปรแกรม OCR ผลลัพธ์ที่ได้คือ "เจอร์รี่และบ็อบบี่ สเปลล์แมน ดีใจที่มีเพื่อนบ้านใหม่เป็นคู่แต่งงานน่ารักวัยสามสิบกว่าๆ นามว่าพีทและลินดา ฟอร์สเตอร์" ดังนั้นจึงมีความจำเป็นที่จะต้องรวมบล็อกที่ Overlap กัน เพื่อความหมายที่ถูกต้อง

4.6 การหาระดับบรรทัด

การหาระดับบรรทัด ใช้วิธีการฮิสโตแกรมด้วยขอบของตัวอักษร ซึ่งจะทำให้สามารถทำฮิสโตแกรมได้อย่างรวดเร็ว แล้วนำฮิสโตแกรมที่ได้มาทำการตรวจสอบหาช่องว่างที่ไม่มีตัวอักษร เพื่อนำมาเป็นช่องว่างระหว่างบรรทัด ก็จะทำให้สามารถหาตัวอักษรระดับบรรทัดได้



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประสานความสัมพันธ์เพื่อขอความร่วมมือจากผู้มีตำแหน่งหน้าที่ทางการงานสูงกว่าหรือต่ำกว่าจะได้รับการตอบสนองด้วยดี

(ข)

รูปที่ 4.12 เปรียบเทียบการทำฮิสโตแกรม

(ก) ฮิสโตแกรมทีละจุด, (ข) ฮิสโตแกรมด้วยขอบของตัวอักษร

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

งานวิจัยนี้ ได้พัฒนาโปรแกรมโดยใช้ Visual Basic.Net 2003 ทำการประมวลผลบนเครื่อง Sempron 2400 แรม 960 MB ภายใต้ระบบปฏิบัติการ WindowXP SP2 และทำการทดลองกับภาพเอกสารไบนารี่ที่ผ่านการสแกนที่ความละเอียด 300 dpi ภาพเอกสารไม่มีกรอบ ภาพเอกสารที่ใช้ในการจับเวลาจำนวน 17 รูป ดังแสดงในภาคผนวก ง รูปที่ ง.1 – รูปที่ ง.17

นำผลการทดลองมาตรวจสอบความถูกต้องของภาพเอกสาร โดยนับจำนวนบล็อกที่โปรแกรมคำนวณได้ นำมาเปรียบเทียบกับของจริงที่ถูกต้อง พบว่าภาพเอกสารทั้งหมด ได้ผลลัพธ์ที่ถูกต้อง

ผลการทดลองที่ได้จากการจับเวลาการทำงาน ตามตารางที่ 5.1 โดยแบ่งเวลาเป็น 2 ช่วงคือ

1. เวลาในช่วงที่ 1 เป็นเวลาในการหาขอบเขตของตัวอักษร และขอบเขตของรูปภาพ
2. เวลาในช่วงที่ 2 เป็นเวลาที่ใช้ในการหาขอบเขตของบล็อกตัวอักษร

ตารางที่ 5.1 ผลการทดลองด้วยวิธี Dilation1PX

ตัวอย่างที่	ขนาดภาพ (พิกเซล)	เวลาในช่วงที่1 (วินาที)	เวลาในช่วงที่2 (วินาที)	เวลารวม (วินาที)
1	2400x3300	6.812	0.765	7.577
2	2400x3300	7.437	1.500	8.937
3	2400x3300	7.500	1.0	8.5
4	2400x3300	6.750	1.328	8.078
5	2800x3500	7.515	1.671	9.186
6	2500x3350	5.937	1.125	7.062
7	2400x3300	3.687	1.125	4.812
8	2700x3700	8.906	1.531	10.437
9	2400x3300	5.625	1.218	6.843
10	2400x3300	4.875	1.93	6.805
11	2700x3500	6.562	1.187	7.749
12	2750x3300	6.828	1.375	8.203

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 (ต่อ)

ตัวอย่างที่	ขนาดภาพ (พิกเซล)	เวลาในช่วงที่1 (วินาที)	เวลาในช่วงที่2 (วินาที)	เวลารวม (วินาที)
13	2800x3300	8.15	1.546	9.696
14	2400x3300	5.171	1.109	6.280
15	2300x3000	6.93	0.984	7.914
16	2400x3300	6.515	1.390	7.905
17	2400x3300	4.796	1.125	5.921

ผลการทดลองจากตารางที่ 5.1 สรุปได้ดังนี้

1. การทำงานในขั้นตอนที่ 1 ใช้หน้าต่างแล้วตรวจสอบเพียงบางตำแหน่งในหน้าต่าง ทำให้การประมวลผลเป็นไปอย่างรวดเร็ว และเมื่อหน้าต่างตรวจสอบเจอจุดดำแล้ว จะทำการหาขอบเขตแบบสี่เหลี่ยมของกลุ่มดำที่พบด้วย Chain Code 1 พิกเซล ซึ่งใช้เวลาในการประมวลผลนาน ดังนั้นเวลาการทำงานในช่วงที่ 1 จึงไม่เร็วมากนัก

2. การทำงานในขั้นตอนที่ 2 เป็นการทำงานกับ Optimum Picture ซึ่งมีขนาดที่เล็กกว่าต้นฉบับ 144 เท่า ดังนั้นจึงทำให้การทำงานเป็นไปอย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

6.1.1 สรุปผลการทำงานด้วยวิธี Dilation1PX

บทความวิจัยนี้ได้นำเสนอวิธีการแบ่งส่วนประกอบภาพเอกสารไบนารี่ ที่สามารถทำงานได้อย่างถูกต้องและรวดเร็ว ด้วยการใช้หน้าต่างในการตรวจสอบหาจุดดำ แล้วนำตำแหน่งของกลุ่มจุดดำที่เป็นตัวอักษรมาสร้าง Optimum Picture จากนั้นเดินรอบขอบกลุ่มจุดดำใน Optimum Picture เพื่อหาขอบเขตของบล็อกตัวอักษร โดยเรียกขั้นตอนทั้งหมดนี้ว่าวิธี Dilation1PX

การตรวจสอบหาจุดดำด้วยการใช้หน้าต่างแบบ I สามารถตรวจสอบได้อย่างถูกต้อง เพราะช่องว่างที่ไม่ได้รับการตรวจสอบภายในหน้าต่างแบบ I มีขนาดใหญ่ที่สุดเพียง 3x3 พิกเซล แต่ตัวอักษรโดยทั่วไปมีขนาดใหญ่กว่านี้ ดังนั้นจึงทำให้หน้าต่างแบบ I สามารถตรวจสอบหาจุดดำของตัวอักษรได้อย่างถูกต้อง และการตรวจสอบหาจุดดำจะทำการตรวจสอบเพียงบางตำแหน่งในหน้าต่าง ไม่ได้ทำการตรวจสอบทุกตำแหน่ง จึงทำให้การใช้หน้าต่างแบบ I สามารถทำงานได้อย่างรวดเร็ว

การหาขอบเขตของบล็อกตัวอักษรด้วย Optimum Picture สามารถทำงานได้อย่างรวดเร็ว เนื่องจาก Optimum Picture มีขนาดเล็กกว่ารูปต้นฉบับถึง 144 เท่า ด้วยขนาดที่เล็กนี่เอง จึงทำให้การทำงานต่างๆกับ Optimum Picture สามารถทำงานได้อย่างรวดเร็ว

แต่เนื่องจาก Optimum Picture ถูกย่อขนาดลงจากขนาดจริงถึง 144 เท่า จึงทำให้ Optimum Picture สามารถใช้อธิบายถึงโครงสร้างของภาพเอกสารได้เท่านั้น เช่น การหาขอบเขตของบล็อกตัวอักษร และการหาขอบเขตของรูป แต่ไม่สามารถบอกรายละเอียดภายในภาพเอกสารได้ เช่น ขอบเขตของตัวอักษร (ในงานวิจัยนี้ ขอบเขตของตัวอักษรหาได้ตั้งแต่ขั้นตอนแรก)

การหาขอบเขตของบล็อกตัวอักษรใน Optimum Picture ด้วยวิธี Dilation1PX มีความสามารถในการหาขอบเขตของบล็อกตัวอักษรได้ดีกว่า วิธีการใช้หน้าต่าง 32x32 พิกเซล คือมีโอกาสในการรวม 2 คอลัมน์มาเก็บเป็นบล็อกเดียวกันเกิดขึ้นได้น้อยกว่า ซึ่งข้อดีของการเก็บขอบเขตของบล็อกได้อย่างถูกต้อง คือไม่ต้องเสียเวลาในการตัดแบ่งคอลัมน์ ก็จะทำให้เวลาการทำงานโดยรวมของขั้นตอนการหาขอบเขตของบล็อกทำงานได้รวดเร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.2 สรุปขั้นตอนการทำงานของวิธี Dilation1PX และวิธีที่นำเสนอใน [1]

การแบ่งส่วนประกอบภาพเอกสารไบนารี่ด้วยวิธี Dilation1PX และวิธีที่นำเสนอใน [1] สามารถทำงานได้อย่างถูกต้องเมื่อทดลองกับภาพเอกสารภาษาไทย และภาษาอังกฤษ ในรูปแบบวารสารทางวิชาการ และนิตยสาร แต่ถ้าเป็นภาพเอกสารที่มีโครงสร้างสลับซับซ้อน เช่น หนังสือพิมพ์ วิธี Dilation1PX ไม่สามารถทำงานได้อย่างถูกต้อง แต่วิธี [1] สามารถทำงานได้อย่างถูกต้อง (ดูรายละเอียดในหัวข้อที่ 6.2) ดังนั้น ในหัวข้อนี้จึงทำการสรุปการทำงานในแต่ละขั้นตอนของวิธี Dilation1PX และวิธีที่นำเสนอใน [1]

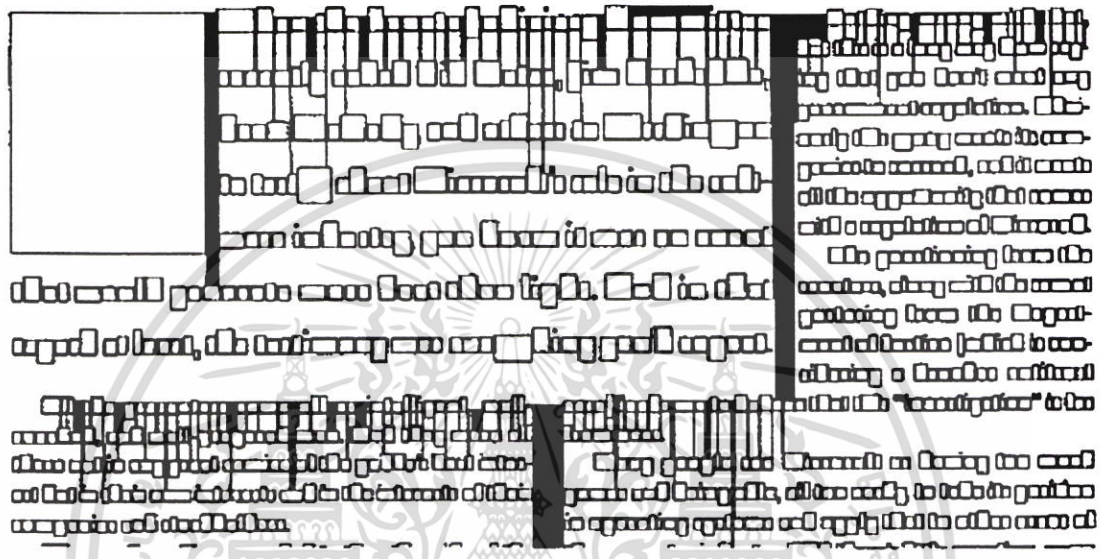
ตารางที่ 6.1 สรุปขั้นตอนการทำงานของวิธี Dilation1PX และวิธี [1]

ขั้นตอน	วิธี [1]	วิธี Dilation1PX
การตรวจสอบหาจุดดำจุดแรก (ตัวอักษรหรือรูปภาพ)	ตรวจสอบทีละจุดแบบ Raster Scan (จากซ้ายไปขวา จากบนลงล่าง)	ตรวจสอบอย่างรวดเร็วด้วยหน้าต่างแบบ I ครั้งละ 1 หน้าต่างแบบ Raster Scan (จากซ้ายไปขวา จากบนลงล่าง)
การหาขอบเขตของกลุ่มจุดดำ	ใช้หน้าต่าง 32x32 พิกเซล เดินรอบขอบ ในรูปต้นฉบับ	เดินรอบขอบทีละ 1 พิกเซล ในรูป Optimum Picture ที่ผ่านการทำ Dilation ด้วย Mask ขนาด 3x3
การจัดเก็บตัวอักษร	เก็บขอบเขต บน, ล่าง, ซ้าย, ขวา ด้วยการเดินรอบขอบทีละ 1 พิกเซล	เก็บขอบเขต บน, ล่าง, ซ้าย, ขวา ด้วยการเดินรอบขอบทีละ 1 พิกเซล
การจัดเก็บบล็อกตัวอักษร	เก็บขอบเขตแบบรูปร่างไม่แน่นอน ด้วยการเดินรอบขอบทีละ 1 หน้าต่างขนาด 32x32 พิกเซล	เก็บขอบเขตแบบรูปร่างไม่แน่นอน ด้วยการเดินรอบขอบทีละ 1 พิกเซล ใน Optimum Picture
การจัดเก็บรูปภาพ	เก็บขอบเขต บน, ล่าง, ซ้าย, ขวา ด้วยการเดินรอบขอบทีละ 1 พิกเซล	เก็บขอบเขตแบบรูปร่างไม่แน่นอน ด้วยการเดินรอบขอบทีละ 1 หน้าต่างด้วยหน้าต่างแบบ I
การตัดคอลัมน์	ตรวจสอบหาช่องว่างระหว่างกลุ่มข้อความที่สามารถตัดทะลุได้อย่างน้อย 3 บรรทัด ทำให้สามารถตัดแบ่งคอลัมน์ที่มีความซับซ้อนได้ดี	ตรวจสอบหาช่องว่างระหว่างกลุ่มข้อความที่สามารถตัดทะลุจนสุดกลุ่มข้อมูล ทำให้สามารถตัดแบ่งคอลัมน์ที่ไม่ซับซ้อนมากได้ (ดูรายละเอียดในหัวข้อที่ 6.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ข้อเสนอแนะ

การตัดแบ่งคอลัมน์ในภาพเอกสารที่มีการจัดวางคอลัมน์อย่างสลับซับซ้อน เช่นรูปที่ 6.1 วิธี [1] จะจัดเก็บคอลัมน์ทั้งหมดนี้เป็นบล็อกเดียวกัน (เนื่องจากระยะห่างระหว่างคอลัมน์น้อยมาก) จากนั้น จะทำการตัดแบ่งคอลัมน์ เพื่อให้ 1 บล็อกข้อมูลมีเพียง 1 คอลัมน์เท่านั้น ผลลัพธ์ที่ได้ก็จะมีความถูกต้อง



รูปที่ 6.1 เอกสารที่มีการจัดวางคอลัมน์อย่างสลับซับซ้อน

วิธีที่ผู้วิจัยนำเสนอ ในขั้นแรกจะจัดเก็บคอลัมน์ทั้งหมดนี้เป็นบล็อกเดียวกัน (เนื่องจากระยะห่างระหว่างคอลัมน์น้อยมาก) จากนั้นจะทำการตัดแบ่งคอลัมน์ แต่จะไม่สามารถทำการตัดแบ่งคอลัมน์ในรูปที่ 6.1 ได้ เพราะวิธีที่นำเสนอนี้ ใช้วิธีนำขอบเขตของตัวอักษรทั้งหมดภายในบล็อกมาทำ Histogram ในแนวตั้ง เพื่อหาช่องว่างที่เกิดขึ้น ซึ่งก็คือการตัดทะลุในแนวตั้ง แต่รูปที่ 6.1 ไม่สามารถตัดทะลุได้

ผู้วิจัยจึงนำเสนอแนวทางในการตัดคอลัมน์ในกรณีที่เอกสารมีการจัดวางคอลัมน์อย่างสลับซับซ้อน โดยการนำวิธี [1] มาทำการประยุกต์ใช้ คือทำการตรวจสอบว่ามีช่องว่างที่สามารถตัดทะลุไปได้เกิน 3 บรรทัดหรือไม่ ถ้าใช่ให้ทำการตัดแบ่งคอลัมน์ที่ตำแหน่งนั้น แต่การตรวจสอบหาตำแหน่งดังกล่าวแทนที่จะทำกับรูปต้นฉบับ ให้ทำกับ Optimum Picture (ที่ยังไม่ได้ทำ Dilation ด้วย Mask ขนาด 3x3) ซึ่งจะทำให้สามารถตัดแบ่งคอลัมน์ได้อย่างถูกต้องและรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] B. Kruatrachue, P. Suthaphan, "A Fast and Efficient Method for Document Segmentation for OCR," Electrical and Electronic Technology, 2001. Vol. 1, 19-22 Aug. 2001 pp. 381-383

[2] Jaekyu Ha, Robert M. Haralick, Ihsin T. Phillips, "Recursive X-Y Cut using Bounding Boxes of Connected Components," Document Analysis and Recognition, 1995. Proceedings of the Third International Conference Volume2, 14-16 Aug. 1995 pp 952-955 vol.2

[3] Sauvola J, Pietikainen M. "Page segmentation and classification using fast feature extraction and connectivity analysis," Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on Volume 2, 14-16 Aug. 1995 pp 1127-1131 vol.2

[4] Jiajun Wang, Yanling Li, Xianwu Huang, Zhenya He, "Page Segmentation and Classification Based on Pattern-list Analysis," Image Segmentation, ISIMP, 2004. pp 735-738

[5] Jaekyu Ha, Robert M. Haralick, Ihsin T. Phillips. "Document Page Decomposition by the Bounding-Box Projection Technique," Document Analysis and Recognition, 1995. Proceedings of the Third International Conference on Volume 2, 14-16 Aug. 1995 pp 1119-1122 vol.2

[6] R. Fisher, S. Perkins, A. Walker and E. Wolfart. "Adaptive Thresholding." [Online]. http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm. 2003

[7] Dr. Hagit Hel-Or. "Image Processing Lectures 13." [Online]. http://cs.haifa.ac.il/hagit/courses /ip/Lectures/Ip13_Binaryx4.pdf. 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
บทความที่ตีพิมพ์ในวารสาร

บทความที่ตีพิมพ์ในวารสาร มี 1 บทความคือ

[1] ไพศาล สุธีบรรเจิด, วิเชียร เปรมชัยสวัสดิ์, และ นุชรี เปรมชัยสวัสดิ์ "วิธีการหากลุ่มข้อมูลของเอกสารอย่างรวดเร็ว." การประชุมวิชาการทางวิศวกรรมไฟฟ้า, ครั้งที่ 26, 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการหากลุ่มข้อมูลของเอกสารอย่างรวดเร็ว

A Fast Block Extraction Method for Document Segmentation

ไพศาล สุธีบรรเจิด*, วิเชียร เปรมชัยสวัสดิ์**, และ นุชรี เปรมชัยสวัสดิ์***
*คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ถนนฉลองกรุง เขตลาดกระบัง กรุงเทพ 10520 โทร. 0-2860-8886 E-mail: mr_phaisarn@yahoo.com
**ผู้ช่วยอธิการบดี มหาวิทยาลัยสยาม
235 ถ. เพชรเกษม เขตภาษีเจริญ กรุงเทพ 10163 wichian@siam.edu
***คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยธุรกิจบัณฑิตย์
ถนนประชาชื่น หลักสี่ กรุงเทพ 10210 nucharee@dpu.ac.th

บทคัดย่อ

ปัจจุบันมีวิธีที่ใช้ในการแบ่งโครงสร้างของเอกสารอย่างมีประสิทธิภาพและรวดเร็ว เช่น การทำ Block Extraction โดยใช้กรอบหน้าต่างขนาด 32*32 พิกเซล[1] วิธีนี้จะหาจุดดำจุดแรกเพื่อใช้ในการเริ่มต้นวางกรอบหน้าต่างขนาด 32*32 พิกเซล จากนั้นจะใช้กรอบหน้าต่างนี้เดินรอบขอบของกลุ่มข้อมูล ก็จะสามารถแบ่งเอกสารออกเป็นกลุ่มข้อมูลได้อย่างรวดเร็ว แต่วิธีที่กล่าวถึงนี้ไม่ได้อธิบายในส่วนของ การหาจุดดำจุดแรก ซึ่งใช้เป็นจุดเริ่มต้นในการประมวลผล ในบทความนี้จึงได้นำเสนอวิธีที่ใช้ในการหาจุดดำจุดแรกอย่างมีประสิทธิภาพ และรวดเร็วด้วยการใช้หน้าต่าง แล้วตรวจสอบเพียงบางจุดในหน้าต่าง ไม่ต้องตรวจสอบทุกจุด ทำให้สามารถทำ Block Extraction ได้รวดเร็วขึ้น ในบทความนี้ได้นำเสนอหน้าต่างที่ออกแบบขึ้นทั้งหมด 6 แบบ นำมาเปรียบเทียบ และเลือกหน้าต่างที่เหมาะสมที่สุดในการนำไปใช้ จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถทำงานได้เร็วขึ้น

คำสำคัญ: การแบ่งโครงสร้างของเอกสาร, หน้าต่าง

Abstract

In the document segmentation process, the first black pixel must be found and used as the starting point of the process. This paper presents a new method for finding the starting point in an efficient way. To speed up the process. Six types of windows are proposed and their efficiency is also compared. The experimental results show that the proposed method can speed up the process for finding the starting point significantly.

Key words: document segmentation, window

1. คำนำ

การทำ Document Segmentation มีอัลกอริทึมที่ใช้ในการทำอยู่หลายอัลกอริทึม โดยจุดเด่นของแต่ละอัลกอริทึมจะอยู่ที่ประสิทธิภาพในด้านความถูกต้องแม่นยำ และความรวดเร็วในการคำนวณ โดยอัลกอริทึมต่างๆเหล่านี้สามารถแบ่งได้เป็น 3 ประเภท ได้แก่

1.1 วิธีการบนลงล่าง (Top-Down Approach) [2]

วิธีการนี้ในเบื้องต้นจะทำการแบ่งภาพอย่างหยาบๆก่อน โดยแบ่งภาพเอสารให้เป็นกลุ่มข้อมูลที่มีขอบเขตใหญ่ก่อน จากนั้นจึงนำแต่ละกลุ่มข้อมูลที่มีขอบเขตใหญ่ มาแบ่งอย่างละเอียดอีกที ให้เป็นกลุ่มข้อมูลที่มีขอบเขตเล็กลง

1.2 วิธีการล่างขึ้นบน (Bottom-Up Approach) [3]

เป็นการวิเคราะห์ส่วนประกอบย่อย แล้วนำส่วนประกอบย่อยที่ได้มาประกอบกันเป็นบรรทัด และกลุ่มข้อมูล(block) โดยพิจารณาจากความใกล้ชิด และขนาดของแต่ละส่วนประกอบ

1.3 วิธีการแบบผสม (Mixed Approach) [4-5]

เป็นการนำวิธี top-down มาใช้ร่วมกับวิธี bottom-up เช่น อัลกอริทึมของ **B. Kruatrachue และ P. Suthaphan [1]** ซึ่งทำงานดังนี้

1. Block Extraction (top-down) [1] โดยใช้กรอบหน้าต่างขนาด 32*32 พิกเซลเดินรอบขอบของกลุ่มข้อมูลเพื่อใช้ในการแบ่งข้อมูลออกเป็นส่วนประกอบใหญ่
2. Multi-Column Block Detection and Segmentation (Bottom-up) [1] เดินรอบขอบทีละ 1 พิกเซล เพื่อหาขอบเขตย่อย แล้วนำขอบเขตย่อยมาหาแนวบรรทัด เพื่อใช้ในการตัดแบ่งคอลัมน์

จุดเด่นของวิธีนี้คือ การใช้กรอบหน้าต่างขนาด 32*32 พิกเซล [1] ในการเดินรอบขอบของกลุ่มข้อมูล โดยไม่ต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบทุกจุด ทำให้การประมวลผลเป็นไปอย่างรวดเร็ว แต่วิธีนี้ยังไม่สมบูรณ์ เพราะไม่ได้กล่าวถึงขั้นตอนการหาจุดคำจุดแรก ที่ใช้เป็นจุดเริ่มต้นในการวางกรอบหน้าต่างขนาด 32*32 พิกเซล

บทความนี้จึงนำเสนอวิธีการหาจุดคำจุดแรก เพื่อใช้เป็นจุดเริ่มต้นในการวางกรอบหน้าต่างขนาด 32*32 พิกเซล อย่างมีประสิทธิภาพ และรวดเร็ว ด้วยการใช้หน้าต่างแล้วตรวจสอบเพียงบางตำแหน่งในหน้าต่าง

ในบทความนี้ได้พัฒนาและออกแบบหน้าต่างทั้งหมด 6 แบบ แล้วทำการทดลอง และนำผลมาเปรียบเทียบหาขนาด และรูปแบบหน้าต่างที่ให้ผลการทำงานดีที่สุด โดยวัดจากประสิทธิภาพในด้านความถูกต้อง และความรวดเร็วในการคำนวณ

2. การแบ่งส่วนประกอบของภาพเอกสาร

การแบ่งส่วนประกอบของภาพเอกสาร คือการแบ่งขอบเขตของภาพออกเป็นส่วนๆ โดยในแต่ละส่วนจะมีรายละเอียดที่เป็นชนิดเดียวกัน เช่น ขอบเขตข้อมูลที่เป็นข้อความ, ขอบเขตข้อมูลที่เป็นภาพ ซึ่งแบ่งขั้นตอนการทำงานได้เป็น 2 ขั้นตอนดังนี้

2.1 Block Extraction (top-down)

การทำ Block Extraction คือการแบ่งส่วนประกอบของเอกสารออกเป็นส่วนใหญ่ๆ เหมือนกับการใช้สายตามองภาพในระยะไกลแล้วแบ่งภาพออกเป็นส่วนๆ ดังในรูปที่2.

การทำ Block Extraction สามารถแบ่งการทำงานออกได้เป็น 2 ขั้นตอน ดังนี้

รูปที่ 1. ภาพต้นฉบับ

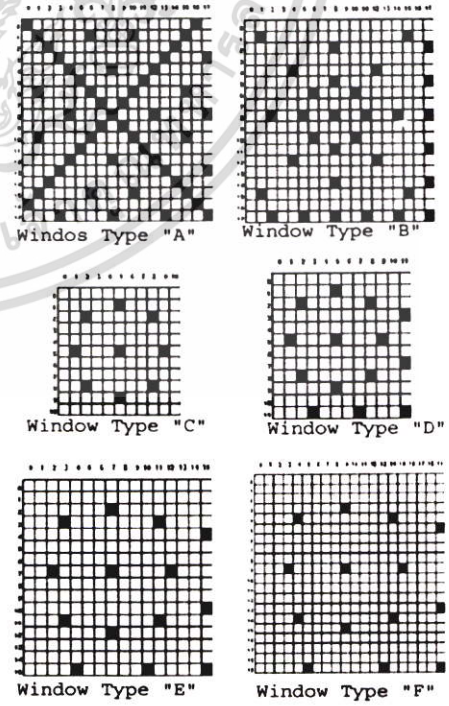
รูปที่ 2. ภาพที่ผ่านขั้นตอนการทำ Block Extraction

2.1.1 การหาจุดเริ่มต้นของกลุ่มข้อมูล

ในขั้นตอนนี้ จะทำการค้นหา และกำหนดตำแหน่งเริ่มต้นในการวางกรอบหน้าต่างอันแรก (กรอบหน้าต่างขนาด 32*32พิกเซล) ของขั้นตอนการหาขอบเขตของกลุ่มข้อมูล

การหาตำแหน่งเริ่มต้นของกลุ่มข้อมูล ก็คือการตรวจสอบหาจุดคำจุดแรก ซึ่งถ้าสามารถตรวจสอบหาจุดคำจุดแรกได้อย่างรวดเร็ว ก็จะสามารถทำขั้นตอน Block Extraction ได้อย่างรวดเร็วตามไปด้วย ในบทความนี้จึงได้เสนอวิธีในการหาจุดคำจุดแรกอย่างรวดเร็วและมีประสิทธิภาพ ด้วยการใช้หน้าต่าง ในการทำงานแล้วตรวจสอบเพียงบางตำแหน่งในหน้าต่าง ทำให้การประมวลผลเป็นไปได้อย่างรวดเร็ว

ในบทความนี้ ได้นำเสนอหน้าต่างที่ได้พัฒนา และออกแบบทั้งหมด 6 แบบ ดังแสดงในรูปที่3.



รูปที่ 3. หน้าต่างแบบ "A" – "F"

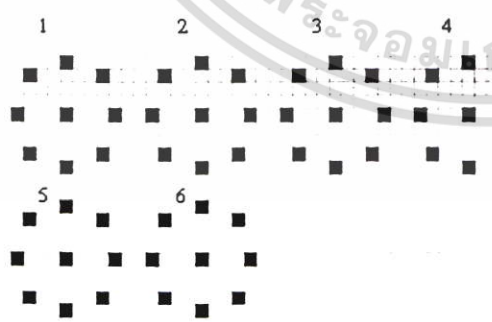
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างแต่ละแบบที่ได้พัฒนาและออกแบบมา มีข้อแตกต่างกันดังนี้

- ขนาดหน้าต่าง
- จำนวนจุดที่ใช้ตรวจสอบ (จุดในรูปคือตำแหน่งที่ใช้ตรวจสอบ)
- การกระจายตำแหน่งของจุดที่ใช้ตรวจสอบ

แนวคิดที่ใช้ในการออกแบบหน้าต่าง คือ หน้าต่างต้องมีขนาดใหญ่ แต่จำนวนจุดที่ใช้ในการตรวจสอบต้องมีจำนวนน้อย (เพื่อให้การตรวจสอบหน้าต่างเป็นไปอย่างรวดเร็ว) และการกระจายตำแหน่งจุดที่ใช้ตรวจสอบต้องเป็นไปอย่างสมดุลย์ เพื่อให้แต่ละจุดครอบคลุมพื้นที่ให้มากที่สุด

การหาจุดเริ่มต้นของกลุ่มข้อมูล(การหาจุดดำจุดแรก) ทำได้โดยนำหน้าต่างแรกวางที่ตำแหน่งมุมบนซ้ายของภาพเอกสาร แล้วตรวจสอบภายในหน้าต่างว่ามีจุดดำหรือไม่ โดยตำแหน่งที่ทำการตรวจสอบหาจุดดำขึ้นอยู่กับชนิดของหน้าต่างที่นำมาใช้ (ดูรูปที่3. ประกอบ) ถ้าตรวจพบจุดดำภายในหน้าต่าง จะตรวจสอบว่าจุดดำที่พบเป็นสัญญาณรบกวน (noise) หรือไม่ โดยการเดินรอบขอบ ณ. ตำแหน่งที่พบจุดดำ โดยเดินทีละ 1 พิกเซล ถ้าความกว้าง หรือความยาวที่ได้มากกว่า 5 พิกเซล ให้ถือว่าสิ่งที่อยู่ภายในหน้าต่างนี้คือข้อมูลให้จบการทำงานในขั้นตอนนี้ และส่งตำแหน่งมุมบนซ้ายของหน้าต่างนี้ ให้เป็นจุดเริ่มต้นของขั้นตอน "การหาขอบเขตของกลุ่มข้อมูล" ต่อไป แต่ถ้าความกว้างและความยาวน้อยกว่า 5 พิกเซล ถือว่ากลุ่มจุดสีดำที่พบเป็นสัญญาณรบกวน ให้ลบสัญญาณรบกวนนี้ออกจากภาพเอกสาร แล้วตรวจสอบตำแหน่งต่อไปภายในหน้าต่าง ถ้าตรวจสอบทุกตำแหน่งในหน้าต่างแล้ว ยังไม่พบจุดดำ ให้ถือว่าภายในหน้าต่างนี้ไม่มีข้อมูล จากนั้นให้วางหน้าต่างต่อไป การวางหน้าต่างจะวางจากซ้ายไปขวา และจากบนลงล่างดังในรูปที่ 4.



รูปที่ 4. ลำดับการวางหน้าต่างในขั้นตอน การหาจุดเริ่มต้นของกลุ่มข้อมูล

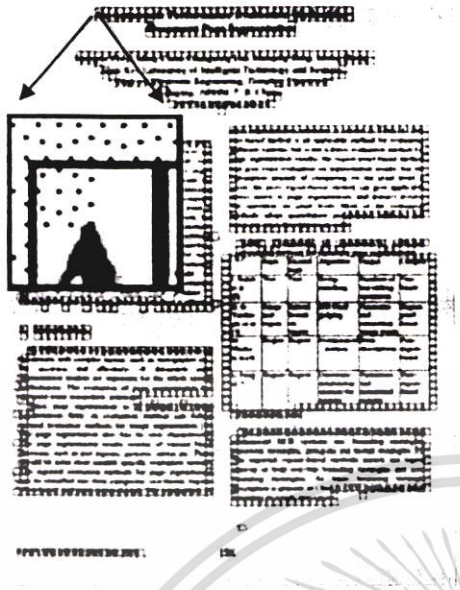
2.1.2 การหาขอบเขตของกลุ่มข้อมูล

เมื่อได้ตำแหน่งเริ่มต้นของกลุ่มข้อมูลแล้ว จะทำการวางกรอบหน้าต่างขนาด 32*32 พิกเซล รอบขอบของกลุ่มข้อมูลตามทิศทางรหัสแนว (Chain Codes) จนกระทั่ง หน้าต่างกลับมาที่ตำแหน่งเริ่มต้นจึงหยุด โดยกำหนดว่า ในกรอบหน้าต่างขนาด 32*32 พิกเซลนี้ ต้องมีพิกเซลดำอยู่อย่างน้อย 10 พิกเซล จึงจะถือว่าเป็นหน้าต่างที่มีข้อมูล การเดินรอบขอบข้อมูลด้วยกรอบหน้าต่างนี้ ทำให้รู้ขอบเขตของกลุ่มข้อมูลได้ ซึ่งขอบเขตของกลุ่มข้อมูลที่ได้นี้ สามารถมีรูปร่างต่างๆกันได้ขึ้นอยู่กับข้อมูลที่มี ไม่จำเป็นว่าจะต้องเป็นสี่เหลี่ยม จากนั้นกลับไปทำขั้นตอนที่ 2.1.1 เพื่อหาจุดเริ่มต้นของกลุ่มข้อมูลอื่นๆ ต่อไป

หลังจากผ่านขั้นตอน Block Extraction แล้ว ข้อมูลที่ได้จะถูกแบ่งออกเป็นกลุ่มข้อมูล(block) ซึ่งในแต่ละกลุ่มข้อมูลอาจมีมากกว่า 1 คอลัมน์ได้ เนื่องจากระยะห่างระหว่างคอลัมน์น้อยกว่า 32 พิกเซล ซึ่งต้องมีการแยกคอลัมน์ออกจากกันโดยใช้ Multi-Column Block Detection and Segmentation ซึ่งจะกล่าวถึงในขั้นตอนต่อไป

รูปที่ 5. ภาพที่ผ่านขั้นตอนการทำ Block Extraction โดยการหาตำแหน่งเริ่มต้นของกลุ่มข้อมูลด้วยการสแกนแบบ raster

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6. ภาพที่ผ่านขั้นตอนการทำ Block Extraction โดยการหาตำแหน่งเริ่มต้นของกลุ่มข้อมูลด้วยการใช้หน้าต่าง

รูปที่5-6 เปรียบเทียบ การหาตำแหน่งเริ่มต้นของกลุ่มข้อมูลด้วยวิธีการสแกนแบบ raster ดังรูปที่5. และการหาตำแหน่งเริ่มต้นของกลุ่มข้อมูลด้วยการใช้หน้าต่างดังรูปที่6. โดยจุดสีแดงในรูปแสดงตำแหน่งที่ใช้ในการตรวจสอบหาจุดดำจุดแรก ซึ่งสามารถเห็นได้ว่าจุดสีแดงในรูปที่6. น้อยกว่าจุดสีแดงในรูปที่5. ซึ่งหมายถึงว่า รูปที่6. สามารถประมวลผลได้รวดเร็วกว่ารูปที่5. เป็นอย่างมาก (เส้นสีน้ำเงินในรูปคือ กรอบหน้าต่างขนาด 32*32 พิกเซล)

2.2 Multi-Column Block Detection and Segmentation (bottom - up)

จากขั้นตอน Block Extraction ที่ผ่านมา เป็นไปได้ที่ใน 1 กลุ่มข้อมูล (block) จะมีมากกว่า 1 คอลัมน์ จึงใช้วิธี bottom-up ในการแบ่งคอลัมน์ให้ 1 กลุ่มข้อมูลมีเพียง 1 คอลัมน์ โดยแยกข้อมูลออกเป็นส่วนย่อยที่สุดด้วยการเดินรอบขอบทีละ 1 พิกเซล จะทำให้สามารถหากรอบสี่เหลี่ยม รอบส่วนประกอบย่อยได้ (bounding box) ถ้ากรอบสี่เหลี่ยมของส่วนประกอบย่อยที่ได้ มีความกว้าง หรือความสูงมากกว่าค่าที่กำหนดไว้ ก็จะกำหนดให้เป็นข้อมูลชนิดรูปภาพ แต่ถ้าความกว้าง และความสูงน้อยกว่าค่าที่กำหนดไว้ ก็จะให้เป็นข้อมูลชนิดตัวอักษร จากนั้นนำกรอบสี่เหลี่ยมของส่วนประกอบย่อยมาหาแนวระดับบรรทัด จากนั้นวิเคราะห์การตัดแบ่งคอลัมน์ ด้วยการฉายเงาภาพจุดขาวที่บรรทัดแรกของกลุ่มข้อมูล การฉายเงาภาพจุดขาวจะทำเฉพาะจุดขาวที่อยู่ระหว่าง กรอบสี่เหลี่ยมของส่วนประกอบย่อย ตามแนวตั้ง ถ้าความกว้าง และความสูงของการฉายเงาภาพจุดขาวนี้มีค่ามากกว่าค่าที่กำหนดไว้ จะกำหนดให้เป็นแนวแบ่งคอลัมน์ แนวแบ่งคอลัมน์สิ้นสุดที่แถว (y)ใด ก็จะเริ่มต้นทำการฉายเงาภาพจุดขาวที่แถวนั้นต่อ จนกระทั่งสิ้นสุดภาพเอกสาร หรือจบเมื่อไม่สามารถตัดแบ่งคอลัมน์ได้อีก

3. ผลการทดลอง

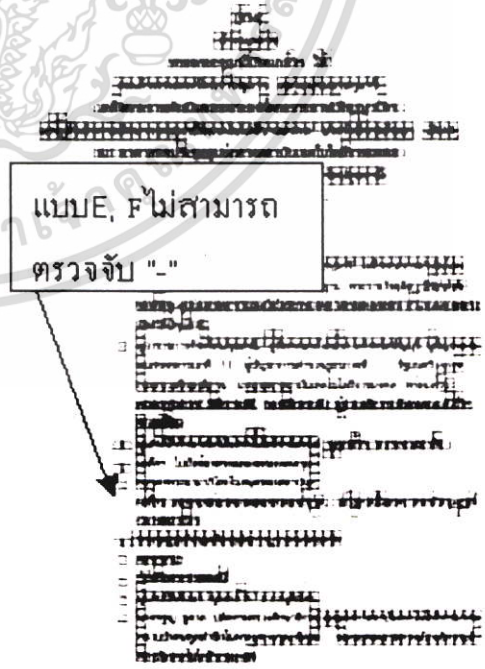
จากหน้าต่างที่ได้พัฒนา และออกแบบมาทั้ง 6 แบบ จะเลือกหน้าต่างที่ดีที่สุด มาใช้หาจุดเริ่มต้นของกลุ่มข้อมูล โดยเลือกจากเงื่อนไขในด้านของประสิทธิภาพความถูกต้อง และเวลาที่ใช้ในการคำนวณ

3.1 ความถูกต้อง

นำหน้าต่างทั้ง 6 แบบ มาทดสอบกับภาพอักษรภาษาไทยและภาษาอังกฤษ หน้าต่างแบบ "A"–"D" สามารถทำงานได้อย่างถูกต้อง ส่วนหน้าต่างแบบ "E"–"F" ยังมีการทำงานที่ผิดพลาดอยู่บ้าง คือ ไม่สามารถตรวจจับสัญลักษณ์บางตัวได้ ดังในรูปที่7.

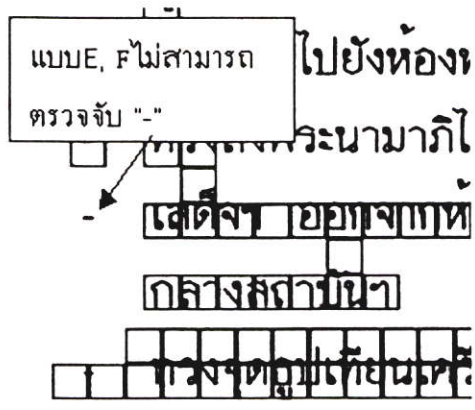
3.2 เวลาที่ใช้ในการคำนวณ

นำหน้าต่างทั้ง 6 แบบ มาเปรียบเทียบเวลาการทำงาน เพื่อแสดงให้เห็นถึงประสิทธิภาพในการคำนวณที่รวดเร็วของหน้าต่างแต่ละแบบ โดยการทดลองนี้ จับเวลาเฉพาะขั้นตอนการทำ Block Extraction เท่านั้น เพื่อให้สามารถเห็นถึงเวลาที่แตกต่าง ของหน้าต่างแต่ละแบบได้อย่างชัดเจน



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข)

รูปที่ 7. (ก) แสดงความผิดพลาดจากการใช้หน้าต่าง "E", "F"
(ข) จากรูป (ก) ขยายภาพให้ดูได้ชัดเจนขึ้น

การทดลองนี้ ทำบนเครื่อง Pentium4 1.6 GHz แรม 256 MB ภายใต้ระบบปฏิบัติการ WindowXP ใช้โปรแกรมภาษา Visual Basic.Net 2002 เอกสารขนาด A4 สแกนที่ความละเอียด 300dpi ภาพเอกสารไม่มีกรอบ ภาพเอกสารที่ใช้ในการจับเวลาแสดงในรูปที่9. เวลาที่ใช้ในการทดลองแสดงในตารางที่1.

ตารางที่ 1. เปรียบเทียบเวลาการทำงานของหน้าต่างแบบต่างๆ

แบบหน้าต่าง	ขนาดหน้าต่าง	จุดทั้งหมด	จุดตรวจสอบ	จุดที่ใช้ตรวจสอบคิดเป็น %	เวลา (sec)
A	18x18	324	52	16.049	7.801
B	18x18	324	32	9.877	7.39
C	11x11	121	9	7.438	7.37
D	12x12	144	14	9.72	7.43
E	16x16	256	14	5.469	7.06
F	20x20	400	14	3.5	6.77

จากตารางที่1 แสดงเวลาการทำงานในขั้นตอน Block Extraction โดยใช้การหาจุดเริ่มต้นของกลุ่มข้อมูล ด้วยหน้าต่างแบบต่างๆ หน้าต่างทั้ง 6 แบบใช้เวลาประมาณ 6 - 8 วินาที ซึ่งรวดเร็วกว่าการหาจุดเริ่มต้นของกลุ่มข้อมูลด้วยการสแกนแบบ raster ซึ่งใช้เวลา 24.986 วินาที

4. สรุปผลการทดลอง

ในบทความนี้นำเสนอ วิธีการหาจุดเริ่มต้นของกลุ่มข้อมูล อย่างมีประสิทธิภาพ และรวดเร็วด้วยหน้าต่างแบบ "C" ซึ่งเหมาะกับงานคุณภาพ ที่ต้องการความถูกต้องสูง และการประมวลผลที่รวดเร็ว

แต่ในกรณีที่ต้องการนำอัลกอริทึมนี้ ไปใช้กับภาพเอกสารที่สแกนด้วยความละเอียดที่มากกว่า 300dpi (มีผลทำให้ตัวอักษรในภาพเอกสารมีขนาดใหญ่ขึ้น) หรือนำไปใช้กับภาพเอกสารที่ตัวอักษรมีขนาดใหญ่ ขอแนะนำหน้าต่างแบบ "E" (ซึ่งสามารถทำงานได้อย่างถูกต้องเมื่อตัวอักษรมีขนาดใหญ่ขึ้น และยังทำงานได้รวดเร็วกว่าหน้าต่างแบบ "C") แต่ถ้าตัวอักษรมีขนาดใหญ่มาก ขอแนะนำหน้าต่างแบบ "F" หรือจะสร้างหน้าต่างขึ้นมาใหม่โดยนำหน้าต่างแบบ "F" มาทำการขยายขนาด (ซึ่งจริงๆแล้วหน้าต่างแบบ "F" ก็คือการนำหน้าต่างแบบ "E" มาขยายขนาดนั่นเอง)

รูปที่ 8. ตัวอย่างภาพที่ผ่านขั้นตอน Block Extraction

รูปที่ 9. ตัวอย่างภาพที่ผ่านขั้นตอน Block Extraction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ อาจารย์ วิเชียร เปรมชัยสวัสดิ์ ที่ให้คำแนะนำปรึกษาจนบทความนี้สำเร็จลุล่วงไปด้วยดี

ขอขอบคุณ อาจารย์ ธนารัตน์ ชลิดาพงศ์ ที่ได้สั่งสอนความรู้พื้นฐานเกี่ยวกับ Digital Image Processing

เอกสารอ้างอิง

[1] B. Kruatrachue, P. Suthaphan, "A Fast and Efficient Method for Document Segmentation for OCR," *Electrical and Electronic Technology, 2001*. Vol. 1, 19-22 Aug. 2001 pp. 381 -383

[2] T. Akiyama and N. Hagita, "Automated entry system for printed documents," *Pattern Recognition* 23, 1990, pp. 1141-1154.

[3] K. C. Fan, C. H. Liu, and Y. K. Wang, "Segmentation and classification of mixed text/graphics/image documents," *Pattern Recognition* Letters 15, 1994, pp. 1201-1209.

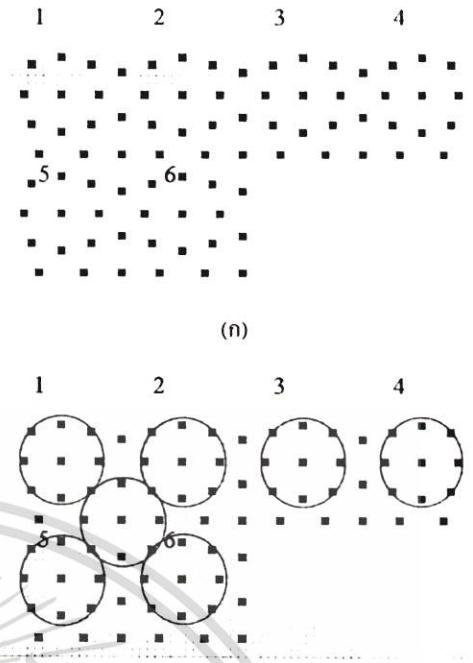
[4] S. N. Srihari, T. Hong, and G. Srikantan, "Machine printed Japanese document recognition," *Pattern recognition* 30, 1997, pp. 1301-1313.

[5] D. Wang, S.N. Srihari. "Classification of Newspaper Image Blocks using Texture Analysis," Computer Vision, Graphics, and Image Processing, Vol. 47, 1989, pp. 327-352.

ภาคผนวก แนวคิดในการออกแบบหน้าต่าง

ในขั้นตอนการออกแบบหน้าต่างได้ค้นพบว่า หน้าต่างแบบ "C" มีความสมมาตร เหมาะที่จะนำมาใช้ในการตรวจหาจุดคำจุดแรก แต่การนำหน้าต่างมาใช้ โดยการวางหน้าต่างจากซ้ายไปขวา และจากบนลงล่างดังรูปที่4. ทำให้เกิดข้อจำกัด คือ ไม่สามารถขยายขนาดของหน้าต่างให้มีขนาดใหญ่ขึ้นได้ เพราะการขยายขนาดหน้าต่างจะทำให้ช่องว่างระหว่างหน้าต่างขนาดใหญ่ขึ้นตามไปด้วย ซึ่งพื้นที่บริเวณนี้จะไม่ได้รับการตรวจหาจุดคำ

ทางออกของปัญหานี้คือต้องวางหน้าต่างแบบ "C" เยื้องกัน ซึ่งช่องว่างที่เกิดขึ้นจะมีขนาดเล็กกว่าการวางเรียงแบบเดิม จึงได้มีการพัฒนาหน้าต่างแบบ "D" แล้วทำการขยายขนาดเป็นหน้าต่างแบบ "E" และหน้าต่างแบบ "F" ตามลำดับ



รูปที่ 10. ลำดับการวางหน้าต่างแบบ "E"

รูปที่10. (ก) เป็นการวางหน้าต่างแบบ "E" จำนวน 6 ครั้ง (ข) เน้นให้เห็นถึงหน้าต่างแบบ "C" ที่ซ่อนอยู่ภายในหน้าต่างแบบ "E"

นาย ไพศาล สุธีบรรเจิด
คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
สนใจงานในด้าน Digital Image Processing

รศ.ดร. วิเชียร เปรมชัยสวัสดิ์
จบปริญญาเอกจาก Waseda University, Tokyo ประเทศญี่ปุ่น

รศ.ดร.นุชรี เปรมชัยสวัสดิ์
จบปริญญาเอกจาก Waseda University, Tokyo ประเทศญี่ปุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
อัลกอริทึมและขั้นตอนการทำงาน

ข.1 กระบวนการแบ่งส่วนประกอบภาพเอกสารไบนารี่ด้วยวิธี Dilation1PX

1. หาขอบเขตของตัวอักษรและขอบเขตของรูปภาพ
   1.1 เก็บตำแหน่งของตัวอักษร โดยเก็บขอบเขต บน, ล่าง, ซ้าย, ขวา ของตัวอักษร
   1.2 หาขอบเขตแบบรูปร่างไม่แน่นอนของรูป ด้วยการใช้หน้าต่างแบบ I เดินรอบรูป
2. หาขอบเขตของบล็อกตัวอักษรจาก Optimum Picture
   2.1 สร้าง Optimum Picture จากขอบเขตของตัวอักษร
   2.2 หาขอบเขตของบล็อกตัวอักษร
   2.3 ตัดแบ่งคอลัมน์
   2.4 นำตัวอักษรจากขั้นตอนที่ 1 เก็บเข้าในบล็อกตัวอักษร
   2.5 รวมบล็อกที่ Overlap กัน
   2.6 หาระดับบรรทัดของตัวอักษร

ข.1.1 การหาขอบเขตของตัวอักษรและขอบเขตของรูปภาพ

1. ใช้หน้าต่างแบบ I ตรวจสอบหาจุดดำ โดยทำการวางหน้าต่างแบบ I แบบ Raster scan

2. เมื่อเจอจุดดำภายในหน้าต่าง ให้ทำการหาจุดดำจุดแรกภายในหน้าต่างด้วยการตรวจสอบทีละจุดแบบ Raster Scan

3. เมื่อเจอจุดดำจุดแรกภายในหน้าต่างแล้ว ให้เดินรอบขอบกลุ่มจุดดำทีละ 1จุด เพื่อหาขนาดและขอบเขตของกลุ่มจุดดำนี้

4. ตรวจสอบขนาดของกลุ่มจุดดำที่ได้ ว่ากลุ่มจุดดำนี้เป็น รูปภาพ, ตัวอักษร, หรือสัญญาณรบกวน โดยที่

   - ถ้าความกว้างหรือความยาวมากกว่า 80 จุด กำหนดให้กลุ่มจุดดำที่พบเป็นรูปภาพ
   - ถ้าความกว้างหรือความยาวมากกว่า 4 จุด และไม่ใช่รูปภาพกำหนดให้กลุ่มจุดดำที่พบเป็นตัวอักษร
   - ไม่ใช่ตัวอักษรและรูปภาพ กำหนดให้เป็นสัญญาณรบกวน

5. ถ้าเป็นรูปภาพ

   - เดินรอบขอบของกลุ่มดำด้วยหน้าต่างแบบ I เพื่อเก็บตำแหน่งขอบของรูปภาพ ดังรูปที่ ข.3
   - ลบส่วนที่เป็นรูปภาพออกจากภาพต้นฉบับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ถ้าเป็นตัวอักษร

- เก็บขอบเขต (บน, ล่าง, ซ้าย, ขวา) ของตัวอักษร
- ลบตัวอักษรออกจากรูปต้นฉบับ

7. ถ้าเป็นสัญญาณรบกวน ให้ลบออกจากรูปต้นฉบับ

ผลที่ได้จากขั้นตอนที่ 1

- ได้ขอบเขตแบบรูปร่างไม่แน่นอนของรูปภาพ โดยเก็บตำแหน่งหน้าต่างที่เป็นขอบของรูป
- ได้ขอบเขตแบบสี่เหลี่ยมของตัวอักษรแต่ละตัว (L, T, R, B)

ข.1.2 การหาขอบเขตของบล็อกตัวอักษรจาก Optimum Picture

1. นำขอบเขตแบบสี่เหลี่ยมของตัวอักษรมาวาดลง Optimum Picture โดยกำหนดขนาด 12x12 พิกเซลของขนาดจริง นำมาย่อขนาดลงเหลือเพียง 1 พิกเซลใน Optimum Picture ดังรูปที่ ข.6 (ข)

2. นำ Optimum Picture มาทำการ Dilation ด้วย Mask ขนาด 3x3 พิกเซล ดังรูปที่ ข.6 (ค)

3. ตรวจสอบหาตัวอักษรใน Optimum Picture โดยการตรวจสอบทุกจุด แบบ Raster Scan เพื่อหาจุดเริ่มต้นในการประมวลผล

4. เมื่อพบตัวอักษรให้เดินรอบขอบของกลุ่มตัวอักษรด้วย Chain Code 1 พิกเซล เพื่อเก็บตำแหน่งขอบของบล็อกตัวอักษร ดังรูปที่ ข.7

5. นำตัวอักษรที่ได้จากขั้นตอนที่ 1 มาตรวจสอบว่ามีขอบเขตอยู่ในบล็อกนี้หรือไม่ ถ้าใช่เก็บตัวอักษรเข้าบล็อกนี้

6. นำขอบเขตของตัวอักษรทุกตัวในบล็อกนี้ มาทำฮิสโตแกรมแนวตั้ง จากนั้นตรวจสอบว่า มีช่องว่างขนาดใหญ่พอที่จะทำการตัดคอลัมน์ได้หรือไม่ ถ้าสามารถตัดแบ่งคอลัมน์ได้ ให้ตัดแบ่งคอลัมน์ที่ตำแหน่งนี้

7. ถ้ามีการตัดคอลัมน์ จะทำการหาขอบเขตแบบสี่เหลี่ยมของบล็อก จากนั้นตรวจสอบว่าตำแหน่งหน้าต่างแรกที่เจอ อยู่ทางซ้ายหรือทางขวาของตำแหน่งที่จะตัดคอลัมน์ ถ้าอยู่ทางซ้ายให้กำหนดขอบเขตแบบสี่เหลี่ยมของบล็อกทางด้านขวาใหม่ด้วยตำแหน่งที่จะตัดคอลัมน์ แต่ถ้าอยู่ทางขวาให้กำหนดขอบเขตแบบสี่เหลี่ยมของบล็อกทางด้านซ้ายใหม่ด้วยตำแหน่งที่จะตัดคอลัมน์ แล้วใช้ขอบเขตแบบสี่เหลี่ยมของบล็อกนี้ มาเป็นขอบเขตในการรอบกลุ่มดำด้วย Chain Code 1พิกเซลอีกครั้ง โดยเดินอยู่ภายในขอบเขตแบบสี่เหลี่ยมของบล็อกนี้เท่านั้น

8. นำตัวอักษรมาจัดเข้าบล็อกตัวอักษรใหม่อีกครั้ง

9. ลบส่วนที่เป็นตัวอักษรภายในบล็อกตัวอักษรที่ได้นี้ ออกจาก Optimum Picture โดยการ Label ภายในขอบของบล็อก ให้เป็นตัวอักษรให้หมด จากนั้นลบส่วนที่เป็นตัวอักษรภายในขอบเขตแบบสี่เหลี่ยมของบล็อกนี้ใน Optimum Picture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. ตรวจสอบว่าบล็อกไหนที่มีขอบเขตแบบสี่เหลี่ยม (L, T, R, B) คาบเกี่ยวกันอยู่ให้รวมเป็นบล็อกเดียวกัน

11. นำตำแหน่งของจุดรอบๆบล็อก มาคูณด้วย 12x12 (ขนาดของหน้าต่างแบบ I) เพื่อจัดเก็บตำแหน่งของขอบเขตแบบรูปร่างไม่แน่นอนของบล็อกในขนาดจริง

12. ทำการฮิสโตแกรมในแนวนอน เพื่อหาระดับบรรทัดของตัวอักษร

ผลที่ได้จากขั้นตอนที่ 2

- ได้ขอบเขตแบบรูปร่างไม่แน่นอนของบล็อกตัวอักษร และเก็บตัวอักษรไว้ภายในบล็อก

ข.2 กระบวนการแบ่งส่วนประกอบภาพเอกสารไบนารี่ด้วยวิธีที่นำเสนอใน [1]

1. การแบ่งส่วนประกอบใหญ่
2. การแบ่งส่วนประกอบย่อยภายในส่วนประกอบใหญ่
3. การแบ่งคอลัมน์และปรับปรุงส่วนประกอบใหญ่

ข.2.1 การแบ่งส่วนประกอบใหญ่

1. ตรวจสอบหาจุดดำจุดแรกด้วยการสแกนแบบราสเตอร์ (Raster Scan)

2. เมื่อเจอจุดดำให้ทำการตรวจสอบด้วยหน้าต่าง 32x32 พิกเซล ว่าภายในหน้าต่างมีจุดดำเกินกว่า 10 จุดหรือไม่ ถ้าใช่ให้ถือว่าเป็นหน้าต่างที่มีข้อมูล

3. ถ้าเป็นหน้าต่างที่มีข้อมูล ให้ใช้หน้าต่าง 32x32 พิกเซลเดินรอบขอบของกลุ่มดำนี้ เพื่อหาขอบเขตของกลุ่มดำที่พบ

4. ลบจุดดำภายในขอบเขตใหญ่ที่ได้

ผลที่ได้จากขั้นตอนที่ 1

- ได้ขอบเขตแบบรูปร่างไม่แน่นอนของกลุ่มดำ (ที่ไม่ได้แยกว่าเป็นตัวอักษรหรือรูปภาพ)

ข.2.2 การแบ่งส่วนประกอบย่อยในส่วนประกอบใหญ่

1. ตรวจสอบหาส่วนประกอบย่อยภายในส่วนประกอบใหญ่ ด้วยการเดินรอบขอบทีละ 1 พิกเซล

2. ตรวจสอบขนาดของกลุ่มจุดดำที่ได้ ว่ากลุ่มจุดดำนี้เป็น รูปภาพ, ตัวอักษร, หรือสัญญาณรบกวน โดยที่

- ถ้าความกว้างหรือความยาวมากกว่า 100 จุด กำหนดให้กลุ่มจุดดำที่พบเป็นรูปภาพ
- ถ้าความกว้างหรือความยาวมากกว่า 4 จุด และไม่ใช่รูปภาพกำหนดให้กลุ่มจุดดำที่พบเป็นตัวอักษร
- ไม่ใช่ตัวอักษรและรูปภาพ กำหนดให้เป็นสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ในกรณีขอบเขตที่ได้เป็นขอบเขตของตัวอักษร ถ้าเป็นขอบเขตของตัวอักษรตัวแรกให้นำไปหาตัวอักษรระดับบรรทัด โดยการสแกนหาจุดดำ 5 แนวทิศทางจากด้านล่างของขอบเขตตัวอักษรลงไปเป็นระยะทางเท่ากับความสูงของตัวอักษร แล้วนำขอบเขตของตัวอักษรที่สูงที่สุดมาเป็นตัวอักษรระดับบรรทัด จากนั้นทำการหากลุ่มตัวอักษรระดับบรรทัด

- สแกนจุดดำ 5 แนว ทิศทางจากด้านขวาของขอบเขตตัวอักษรไปเป็นระยะทางเท่ากับความกว้างของขอบเขตตัวอักษร
- สแกนจุดดำ 5 แนว ทิศทางจากด้านบนของขอบเขตตัวอักษรไปเป็นระยะทางเท่ากับครึ่งหนึ่งของความสูงของขอบเขตตัวอักษร
- สแกนจุดดำ 5 แนว ทิศทางจากด้านล่างของขอบเขตตัวอักษรไปเป็นระยะทางเท่ากับครึ่งหนึ่งของความสูงของขอบเขตตัวอักษร
- สแกนจุดดำ 5 แนว ทิศทางจากด้านซ้ายของขอบเขตตัวอักษรไปเป็นระยะทางเท่ากับความกว้างของขอบเขตตัวอักษร

5. ในกรณีขอบเขตที่ได้เป็นขอบเขตของรูปภาพ ให้นำขอบเขตของรูปภาพแบบสี่เหลี่ยมมาลบออกจากขอบเขตใหญ่

ผลที่ได้จากขั้นตอนที่ 2

- ได้ขอบเขตแบบรูปร่างไม่แน่นอนของบล็อกตัวอักษร โดยเก็บตัวอักษรไว้ภายในบล็อก และได้ขอบเขตแบบสี่เหลี่ยมของรูปภาพ

ข.2.3 การแบ่งส่วนประกอบย่อยในส่วนประกอบใหญ่

1. การตัดแบ่งคอลัมน์ในส่วนประกอบใหญ่ โดยนำกลุ่มข้อมูลระดับบรรทัดมาทำการจัดเรียงบรรทัดจากน้อยไปมาก เพื่อหาขอบเขตของระดับบรรทัดบน แล้วทำการเปรียบเทียบกับบรรทัดถัดไปว่าช่องว่างระหว่างกลุ่มมีความกว้างไม่น้อยกว่าขอบเขต 1 ตัวอักษรสามารถผ่านทะลุไปยังสุดขอบเขตของส่วนประกอบใหญ่ได้หรือไม่

- ถ้าช่องว่างนั้นสามารถทะลุไปเกิน 3 บรรทัดข้อความ หรือจนกระทั่งสุดขอบล่างของส่วนประกอบใหญ่ ให้ตัดแบ่งคอลัมน์ที่ตำแหน่งดังกล่าว
- ถ้าช่องว่างนั้นสามารถทะลุไปบรรทัดข้อความที่อยู่ข้างล่าง ให้บรรทัดที่อยู่ข้างล่างนั้นเป็นบรรทัดเริ่มในการตรวจสอบต่อไป

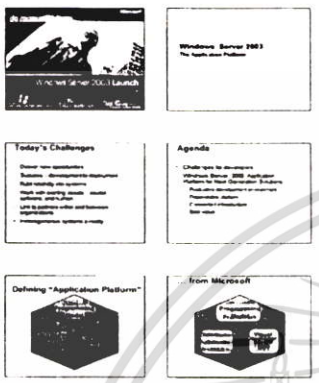
ผลที่ได้จากขั้นตอนที่ 3

- ได้ขอบเขตแบบรูปร่างไม่แน่นอนของบล็อกตัวอักษร ที่ภายในแต่ละบล็อกจะมีเพียง 1 คอลัมน์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค
ภาพเอกสารที่ใช้ในการทดสอบ

ภาพเอกสารทั้งหมดสแกนที่ความละเอียด 300จุดต่อนิ้ว ขนาดฟอนต์ประมาณ 16

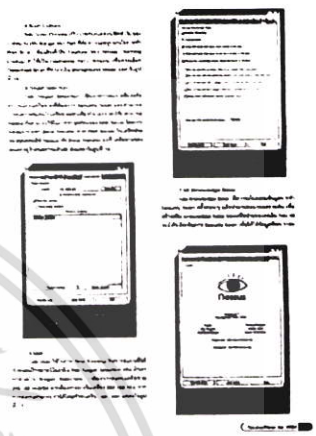


รูปที่ ค.1

ขนาด 2400 x 3300พิกเซล

รูปที่ ค.2

ขนาด 2400 x 3300พิกเซล



รูปที่ ค.3

ขนาด 2400 x 3300พิกเซล

รูปที่ ค.4

ขนาด 2400 x 3300พิกเซล

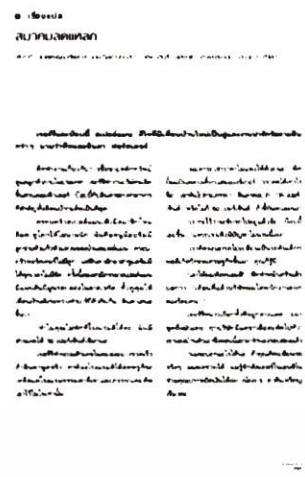
รูปที่ ค.5

ขนาด 2800 x 3500พิกเซล

รูปที่ ค.6

ขนาด 2500 x 3350พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.7

ขนาด 2400 x 3300พิกเซล

รูปที่ ค.8

ขนาด 2700 x 3700พิกเซล

รูปที่ ค.9

ขนาด 2400 x 3300พิกเซล

รูปที่ ค.10

ขนาด 2400 x 3300พิกเซล

รูปที่ ค.11

ขนาด 2700 x 3500พิกเซล

รูปที่ ค.12

ขนาด 2750 x 3300พิกเซล

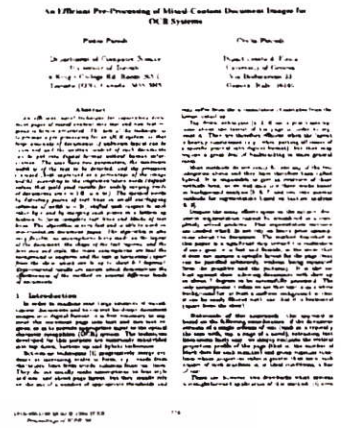
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค.13

ขนาด 2800 x 3300พิกเซล

รูปที่ ค.14

ขนาด 2400 x 3300พิกเซล



รูปที่ ค.15

ขนาด 2300 x 3000พิกเซล

รูปที่ ค.16

ขนาด 2400 x 3300พิกเซล

รูปที่ ค.17

ขนาด 2400 x 3300พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นาย ไพศาล สุธีบรรเจิด
วัน เดือน ปี เกิด	10 สิงหาคม พ.ศ.2522 ที่จังหวัดเพชรบูรณ์
ที่อยู่	97 ตรอกวัดราชวรินทร์ 1 ซ.สมเด็จพระเจ้าตากสิน21 แขวงบุคคโล เขตธนบุรี กรุงเทพ 10600 อีเมลล์ mr.phaisarn@gmail.com
ประวัติการศึกษา	วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ศูนย์กลางสถาบันเทคโนโลยีราชมงคล ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้