สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร และการรู้จำตัวอักษร

THE DOCUMENT ANALYSIS AND IDENTIFICATION AND CHARACTER RECOGNITION

หนังสืออ้างอิง
ห้ามนำออกนอกห้องสมุด

นายสมศักดิ์ วลัยรัชต์

MR. SOMSAK WALAIRACHT

T 021210

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2537

ISBN 974-621-133-1

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่
เลขทะเบียน 21210
วัน, เดือน, ปี - 3 ส.ค. 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE DOCUMENT ANALYSIS AND IDENTIFICATION AND CHARACTER RECOGNITION

SOMSAK WALAIRACHT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

GRADUATE SCHOOL

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1994

ISBN 974-621-133-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารและการรู้จำตัวอักษร
THE DOCUMENT ANALYSIS AND INDENTIFICATION AND CHARACTER RECOGNITION

ชื่อนักศึกษา นายสมศักดิ์ วลัยรัชต์ รหัสประจำตัว 32620021

หลักสูตร วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมคอมพิวเตอร์

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.ชม กิ้มปาน

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.ชม	กิ้มปาน	
รศ.ดร.ฟูศักดิ์	ชีวสุวิทย์	
ผศ.ดร.รัตติกร	วรากูลศิริพันธุ์	
ผศ.ดร.บุญวัฒน์	อัตชู	
ผศ.ดร.กิตติ	ไพฑูรย์วัฒนกิจ	

ค่าระดับคะแนนที่เป็นเอกฉันท์จากคณะกรรมการสอบ GOOD

วัน/เดือน/ปี ที่สอบ 23 มิถุนายน 2537 เวลา 10.00 น.

สถานที่สอบ คณะวิศวกรรมศาสตร์

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.นพิล ตังเจริญพร)
คณบดีบัณฑิตวิทยาลัย

วันที่ 8 เดือน กรกฎาคม พ.ศ. 2537

หมายเหตุ การวัดผลวิทยานิพนธ์ให้ใช้ค่าระดับคะแนนดังนี้

ค่าระดับคะแนน	ผลการศึกษา
O	Outstanding (ดีเยี่ยม)
G	Good (ดี)
P	Pass (ผ่าน)
F	Fail (ไม่ผ่าน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร และการรู้จำตัวอักษร
นักศึกษา	นายสมศักดิ์ วลัยรัชต์
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร. ชม กิ้มปาน
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2537

บทคัดย่อ

งานวิจัยในหัวข้อการรู้จำตัวอักษร (Character recognition) กำลังเป็นหัวข้อที่ได้รับความสนใจอย่างมาก และมีการศึกษาวิจัยเพื่อนำเสนอเทคนิคและวิธีการใหม่ๆ กันอยู่ตลอดเวลา ทั้งนี้ก็เพราะว่างานวิจัยนี้สามารถนำไปประยุกต์ใช้กับการทำงานในชีวิตประจำวันของเรา และก่อให้เกิดคุณประโยชน์อย่างมาก โดยอาศัยระบบการรู้จำตัวอักษรนี้ เครื่องคอมพิวเตอร์จะสามารถวิเคราะห์และรู้จำภาพตัวอักษรที่ปรากฏอยู่บนหน้าเอกสารได้ ซึ่งจะช่วยให้เราสามารถจัดเก็บข้อมูลต่างๆ บนหน้าเอกสารที่มีปริมาณมากๆ ไว้ในเครื่องคอมพิวเตอร์ได้อย่างง่ายดาย ซึ่งแตกต่างจากเดิมที่การจัดเก็บเอกสารทำได้ด้วยความยากลำบาก โดยการพิมพ์ป้อนผ่านทางแป้นพิมพ์เพียงอย่างเดียวเท่านั้น

ผลงานวิจัยที่ผ่านๆ มาในอดีตต่างมุ่งเน้นและให้ความสำคัญกับการค้นคว้าหาวิธีการหรือเทคนิคใหม่ๆ สำหรับการรู้จำตัวอักษรเพียงอย่างเดียว จนทำให้ละเลยไม่ได้ให้ความสำคัญกับส่วนของการจัดเตรียมข้อมูล ซึ่งมีความสำคัญในการทำงานไม่น้อยไปกว่ากันเลย ถ้าจะพิจารณาถึงลักษณะของข้อมูลของระบบแล้ว เราพบว่าข้อมูลในหนึ่งหน้าเอกสารแท้ที่จริงแล้ว ไม่ได้มีเพียงข้อความที่เป็นตัวอักษรพิมพ์อยู่อย่างเดียว แต่อาจมีรูปภาพหรือสัญญลักษณ์อื่นพิมพ์รวมอยู่บนหนึ่งหน้ากระดาษด้วย ดังนั้นการเพิ่มเติมขั้นตอนของการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารให้กับระบบการรู้จำตัวอักษร โดยสามารถกำหนดส่วนที่เป็นรูปภาพซึ่งจะต้องแยกไปจัดเก็บไว้ต่างหาก และส่วนที่เป็นข้อความที่จะต้องส่งผ่านต่อไปเพื่อทำการรู้จำตัวอักษร จะทำให้การทำงานของระบบสมบูรณ์ถูกต้อง และมีประสิทธิภาพมากยิ่งขึ้น

ดังนั้นวิทยานิพนธ์ฉบับนี้จึงขอนำเสนอระบบการรู้จำตัวอักษรที่แตกต่างไปจากงานวิจัยอื่นๆ โดยได้เพิ่มเติมส่วนของการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร ที่ใช้เทคนิคการค้นหาขอบเขตโดยอาศัยทฤษฎีเซลลูล่าออโตมาตา (Zone detection using Cellular Automata) ซึ่งสามารถกำหนดขอบเขตของส่วนประกอบของหน้าเอกสารทั้งหมดได้อย่างถูกต้อง และสำหรับในส่วนของการรู้จำตัวอักษรได้เลือกใช้วิธีการพิจารณาคุณสมบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางโทโพโลยีของลายเส้นของตัวอักษรในแต่ละพื้นที่ของภาพตัวอักษรที่ถูกแบ่งออกเป็นส่วนๆ ทั้งหมด 8 ส่วน ณ. ตำแหน่งจุดศูนย์ถ่วงของภาพตัวอักษร และอาศัยทฤษฎีฟีเจอร์คอนเซนเทรชั่น (Feature concentration) กำหนดรหัสให้กับจุดภาพอ้างอิง ณ. ตำแหน่งจุดศูนย์ถ่วงของลายเส้นตัวอักษรที่ปรากฏในแต่ละพื้นที่ ได้เป็นรหัสซึ่งแทนด้วยเลขฐาน 16 จำนวน 8 รหัส รวมกับรหัสของภาพตัวอักษรทั้งภาพ ได้รหัสแทนตัวอักษรตัวหนึ่งๆ ทั้งหมด 9 รหัส เรียกว่าค่า Q-code และเก็บรหัสทั้งหมดไว้เป็นพจนานุกรมสำหรับการรู้จำตัวอักษรต่อไป

ผลการทดลองในแต่ละขั้นตอนของการทำงานปรากฏผลโดยในส่วนของการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร เมื่อทำการทดลองกับตัวอย่างภาพหน้าเอกสารขนาดกระดาษ A4 สามารถวิเคราะห์และกำหนดขอบเขตต่างๆ ได้อย่างถูกต้องโดยใช้เวลาในการประมวลผลภาพหน้าเอกสารหนึ่งๆ เฉลี่ย 1.5 นาที และในส่วนของการรู้จำตัวอักษร สามารถวิเคราะห์และรู้จำตัวอักษรได้ถูกต้องเฉลี่ย 89% และใช้เวลาในการรู้จำตัวอักษรแต่ละตัวน้อยกว่า 1 วินาที โดยทำการทดลองบนเครื่องไมโครคอมพิวเตอร์ที่มีหน่วยประมวลผลกลางชนิด 80386 ทำงานที่ความเร็วสัญญาณนาฬิกา 25 MHz

การทำงานในแต่ละขั้นตอน ซึ่งประกอบด้วยทฤษฎี วิธีการ การทดลอง และผลการทดลองได้กล่าวไว้อย่างชัดเจนโดยละเอียดในวิทยานิพนธ์ฉบับนี้แล้ว

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	The Document Analysis and Identification and Character Recognition
Student	Mr. Somsak Walairacht
Thesis Advisor	Assoc.Prof.Dr. Chom Kimpan
Level of Study	Master of Engineering in Electrical Engineering
Department	Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang
Year of Academic	1994

Abstract

There is a lot of interests in the field of character recognition. It conquers computer data entry efficiency by training a computer to understand and recognize the image of character instead of retyping the character again.

Many algorithms and heuristic were proposed. Most of them do not describe more about the preprocessing process, the picture and text image have to be separated before the recognition. This thesis will improve this problem by introducing a technique of the preprocessing process.

The process consists of the document analysis and character recognition. First, the process will analyze the whole image of the document to identify its contents. Algorithm used is called Zone Detection by Using Cellular Automata. Picture image will be separated and store as graphical image file. For text characters, each individual character has to be segmented from sentences before passing to the recognition routine. Character segmentation uses the technique of edge detection by contouring along edge of the character. For the recognition, each image of the character is separated into 8 regions at its point of center of gravity. Then, by topology properties of image of character in each region, the feature code is calculated which is represented as 9 hexadecimal numbers. Therefore, each character has its own unique feature codes storing for the recognition. Result of the recognition process, an ascii text file is created. Saving a lot of manpower, the document is then computerizely stored and its contents can be retrieved for further works.

The experimentation is carried out using microcomputer 80386 clock speed 25 MHz and document is A4 in size containing picture images and text characters. For the process of document analysis and identification, the time taken is approximately about 1.5 minutes. For the recognition, the accuracy is about 89 % and takes less than 1 second for each character. Detail of the experiment is described clearly in this thesis.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอกราบขอบพระคุณคุณพ่อ และคุณแม่ของข้าพเจ้าที่ให้การสนับสนุนด้านการศึกษา ดูแลและเอาใจใส่ข้าพเจ้ามาโดยตลอด

ขอขอบพระคุณ รองศาสตราจารย์ ดร. ชม กิ้มปาน ที่ได้ให้ความอนุเคราะห์ ทั้งทางด้านความรู้ คำปรึกษา และแนะนำการจัดทำวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วงด้วยดี

ขอขอบพระคุณเหล่าคณาจารย์และเจ้าหน้าที่ภาควิชาวิศวกรรมคอมพิวเตอร์ และสำนักวิจัยและบริการคอมพิวเตอร์ ในความช่วยเหลือทุกๆ ด้าน

ขอบคุณเพื่อนๆ และน้องๆ ทุกคนที่ช่วยให้การทำงานครั้งนี้สำเร็จลุล่วงไปได้

และสุดท้ายนี้ขอขอบพระคุณ

รองศาสตราจารย์ ดร. ชม กิ้มปาน

รองศาสตราจารย์ ดร. ฟูศักดิ์ ชีวสุวิทย์

ผู้ช่วยศาสตราจารย์ ดร. รัตติกร วรากูลศิริพันธุ์

ผู้ช่วยศาสตราจารย์ ดร. บุญวัฒน์ อัตชู

ผู้ช่วยศาสตราจารย์ ดร. กิตติ ไพฑูรย์วัฒนกิจ

ที่ให้ความกรุณาเป็นกรรมการสอบวิทยานิพนธ์ของข้าพเจ้าในครั้งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	III
กิติกรรมประกาศ	IV
สารบัญ	V
สารบัญภาพประกอบ	VIII
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของงานวิจัยในสาขานี้	1
1.2 ผลงานวิจัยที่ผ่านมา	2
1.3 แนวทางของวิทยานิพนธ์	4
บทที่ 2 อุปกรณ์ที่ใช้ในการเก็บข้อมูลและลักษณะของข้อมูล	8
2.1 เครื่องตรวจกวาดภาพ	8
2.1.1 ลักษณะและการทำงานของเครื่องตรวจกวาดภาพ	9
2.1.2 เครื่องตรวจกวาดภาพที่ใช้ในงานวิจัย	11
2.2 ลักษณะของข้อมูล	12
2.2.1 ลักษณะของข้อมูลภาพกราฟิก	12
2.2.2 การเปลี่ยนข้อมูลบิตแม็พเป็นข้อมูลแบบไบนารี่	14
2.3 ตัวอย่างข้อมูลที่ใช้ในงานวิจัย	17
บทที่ 3 การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร	19
3.1 บทนำ	19
3.2 ทฤษฎีที่นำมาใช้	19
3.2.1 ทฤษฎีเซลลูล่าออโตมาตา	19
3.2.2 เทคนิคการติดตามรอยขอบของภาพ	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.3 การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร	22
3.3.1 การกำหนดขอบเขต	22
3.3.2 การแยกและคัดลอกขอบเขต	31
3.4 การจัดเก็บส่วนประกอบของหน้าเอกสาร	34
บทที่ 4 การรู้จำตัวอักษร	35
4.1 บทนำ	35
4.2 ทฤษฎีที่นำมาใช้	36
4.2.1 ทฤษฎีฟีเจอร์คอนเซนเทรชั่น	36
4.2.2 การคำนวณหาจุดศูนย์ถ่วงของภาพ	39
4.2.3 การคำนวณหาความกว้างและความสูงของตัวอักษร	41
4.3 การแยกภาพตัวอักษรออกจากภาพประโยค	44
4.4 การพิจารณาหาลักษณะเด่นของภาพตัวอักษร	46
4.4.1 การปรับแต่งตำแหน่งของจุดอ้างอิง	46
4.4.2. การพิจารณาคุณสมบัติทางโทโพโลยี่ของลายเส้นของภาพตัวอักษร	48
4.5 การจัดเรียงรูปประโยคหลังการวิเคราะห์	51
บทที่ 5 โปรแกรมระบบและผลการทดลอง	52
5.1 ขั้นตอนการทำงานของระบบ	52
5.1.1 โปรแกรมสำเร็จรูปสำหรับเก็บภาพจากเครื่องตรวจกวาดภาพ	53
5.1.2 โปรแกรมระบบที่พัฒนาขึ้น	54
5.1.2.1 อุปกรณ์ที่ต้องการสำหรับการทำงานของระบบ	54
5.1.2.2 ขั้นตอนการทำงานของโปรแกรมที่พัฒนาขึ้น	54
5.2 การทดสอบการทำงานของโปรแกรม	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 6 บทสรุป	68
6.1 สรุปการทำงานของระบบ	69
6.1.1 ข้อมูลอินพุทกับการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร	69
6.1.2 ข้อมูลอินพุทกับการรู้จำตัวอักษร	69
6.2 ข้อจำกัดของการทำงานของระบบ	70
6.2.1 ข้อจำกัดในส่วนของการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร	70
6.2.2 ข้อจำกัดในส่วนของการรู้จำตัวอักษร	71
6.2 ข้อเสนอแนะสำหรับแนวทางการพัฒนาระบบในอนาคต	71
เอกสารอ้างอิง	72
ภาคผนวก ก แฟ้มข้อมูลกราฟิกรูปแบบ TIFF Version 5.0	74
ภาคผนวก ข ตารางรหัสแทนอักขระ	81
ภาคผนวก ค รูปแบบตัวอักษรและค่า Q-code	82
ภาคผนวก ง โปรแกรมการทำงานของระบบ	96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพประกอบ

หน้า

รูปที่ 2.1 แสดงตัวอย่างภาพเครื่องตรวจกวาดภาพ 8

รูปที่ 2.2 แสดงหลักการทำงานของเครื่องตรวจกวาดภาพ 10

รูปที่ 2.3 แสดงภาพเครื่องตรวจกวาดภาพที่ใช้ในงานวิจัยนี้ 11

รูปที่ 2.4 แสดงตัวอย่างของ Bitmap pattern 12

รูปที่ 2.5 แสดง Bitmap pattern ในรูปของตัวเลข 0 กับ 1 13

รูปที่ 2.6 แสดง Bitmap data 13

รูปที่ 2.7 แสดงโครงสร้าง Bitmap ความกว้างเป็นผลคูณของ 8 14

รูปที่ 2.8 แสดงโครงสร้าง Bitmap ความกว้างไม่เป็นผลคูณของ 8 15

รูปที่ 2.9 แสดงตัวอย่างภาพตัวอักษรในลักษณะไบนารี่ 16

รูปที่ 2.10 แสดงตัวอย่างของภาพหน้าเอกสาร 17

รูปที่ 2.11 แสดงตัวอย่างของชุดตัวอักษรที่ใช้ในงานวิจัย 18

รูปที่ 3.1 แสดงลักษณะการติดตามรอยขอบของภาพ 21

รูปที่ 3.2 แสดงลักษณะของจุดตาย 23

รูปที่ 3.3 แสดงลักษณะของจุดเป็น 24

รูปที่ 3.4 แสดงลักษณะของจุดเกิด 24

รูปที่ 3.5 แสดงตัวอย่างของจุดตายที่มีค่าเป็น 1 ถูกเปลี่ยนให้มีค่าเป็น 0 25

รูปที่ 3.6 แสดงตัวอย่างของจุดเป็นที่มีค่าเป็น 1 ยังคงมีค่าเป็น 1 ต่อไป 25

รูปที่ 3.7 แสดงตัวอย่างของจุดเกิดที่เดิมมีค่าเป็น 0 ถูกเปลี่ยนให้มีค่าเป็น 1 26

รูปที่ 3.8 แสดงตัวอย่างข้อมูลภาพในขั้นตอนการกำหนดขอบเขต 27

รูปที่ 3.9 แสดงตัวอย่างข้อมูลภาพในขั้นตอนการกำหนดขอบเขต 28

รูปที่ 3.10 แสดงตัวอย่างข้อมูลภาพในขั้นตอนการกำหนดขอบเขต 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพประกอบ (ต่อ)

หน้า

รูปที่ 3.11 แสดงตัวอย่างข้อมูลภาพในขั้นตอนการกำหนดขอบเขต 30

รูปที่ 3.12 แสดงตัวอย่างพิกัดของขอบเขต 32

รูปที่ 3.13 แสดงตัวอย่างของขอบเขตต่างๆ ที่หาได้ของภาพข้อมูลตัวอย่าง 33

รูปที่ 4.1 แสดงการกำหนดทิศทางทั้ง 4 ทิศ 36

รูปที่ 4.2 แสดงการพิจารณาลักษณะของจุดขาว 37

รูปที่ 4.3 แสดงค่า Q-code ณ.ตำแหน่งจุดศูนย์ถ่วงของภาพตัวอักษร ก 38

รูปที่ 4.4 แสดงตำแหน่งจุดศูนย์ถ่วงของภาพตัวอักษร 39

รูปที่ 4.5 แสดงพื้นที่ของตัวอักษรที่ถูกแบ่งออก 40

รูปที่ 4.6 แสดงภาพตัวอักษรและความกว้างและความสูงของตัวอักษร 42

รูปที่ 4.7 แสดงผังการทำงานของการหาความกว้างและความสูงของภาพตัวอักษร 43

รูปที่ 4.8 แสดงการแยกภาพตัวอักษรออกจากภาพประโยค 44

รูปที่ 4.9 แสดงจุดอ้างอิงตำแหน่งของภาพตัวอักษร 45

รูปที่ 4.10 แสดงจุดอ้างอิงที่ปรับแต่งจากจุดศูนย์ถ่วงในพื้นที่ 2 ของตัวอักษร ก 46

รูปที่ 4.11 แสดงจุดอ้างอิงที่ถูกปรับแต่งในพื้นที่ 3 ของตัวอักษร ง 47

รูปที่ 4.12 แสดงจุดอ้างอิงที่ถูกปรับแต่งของภาพตัวอักษร ข 47

รูปที่ 4.13 แสดงรหัส Q-code ของภาพตัวอักษร ก 48

รูปที่ 5.1 แสดงการทำงานของระบบ 52

รูปที่ 5.2 แสดงภาพโปรแกรม PhotoStyler 53

รูปที่ 5.3 แสดงภาพหน้าจอของโปรแกรมที่พัฒนาขึ้น 55

รูปที่ 5.4 แสดงวินโดว์ของการเปิดแฟ้มข้อมูล 56

รูปที่ 5.5 แสดงตัวอย่างภาพหน้าเอกสาร 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพประกอบ (ต่อ)

หน้า

รูปที่ 5.6 แสดงวินโดว์ของการติดตั้งค่า Coarseness 58

รูปที่ 5.7 แสดงจำนวนของส่วนประกอบต่างๆ บนหน้าเอกสารที่คำนวณได้ 59

รูปที่ 5.8 แสดงวินโดว์ของการจัดเก็บภาพส่วนประกอบของหน้าเอกสาร 60

รูปที่ 5.9 แสดงภาพการทำงานในส่วนการรู้จำตัวอักษร 61

รูปที่ 5.10 แสดงวินโดว์การสิ้นสุดการทำงานของโปรแกรม 62

รูปที่ 5.11 แสดงตัวอย่างหน้าเอกสารที่ใช้ในการทดสอบ 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงค่าของตัวแปรที่ใช้ในการกำหนดขอบเขต 31

ตารางที่ 4.1 แสดงสัญญลักษณ์ของ Q-code 37

ตารางที่ 4.2 แสดงตัวอย่างค่า Q-code 49

ตารางที่ 4.3 แสดงตัวอย่างค่า Q-code 50

ตารางที่ 5.1 แสดงเวลาที่ใช้ในการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร 64

ตารางที่ 5.2 แสดงผลของการวิเคราะห์ชุดตัวอักษรต้นฉบับภาษาไทย 65

ตารางที่ 5.3 แสดงผลของการวิเคราะห์ชุดตัวอักษรต้นฉบับภาษาอังกฤษ 65

ตารางที่ 5.4 แสดงผลการรู้จำตัวอักษรชุดทดสอบตัวอักษรไทย 66

ตารางที่ 5.5 แสดงผลการรู้จำตัวอักษรชุดทดสอบตัวอักษรอังกฤษ 66

ตารางที่ 5.6 แสดงผลการรู้จำตัวอักษรชุดทดสอบตัวเลขไทยและอารบิค 67

ตารางที่ 5.7 แสดงผลการรู้จำตัวอักษรชุดทดสอบตัวเลขไทยและอารบิค 67

ตารางที่ 5.8 แสดงผลสรุปการรู้จำตัวอักษร 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของงานวิจัยในสาขานี้

จากความก้าวหน้าในระบบการประมวลผลด้วยเครื่องคอมพิวเตอร์ การประยุกต์ใช้งานเครื่องคอมพิวเตอร์เป็นไปอย่างกว้างขวางในวงการต่างๆ เกือบจะทุกสาขาอาชีพ แต่ไม่ว่าเครื่องคอมพิวเตอร์จะถูกนำไปใช้สำหรับงานในลักษณะใดก็ตาม วิธีหนึ่งที่ผู้ใช้จะสามารถติดต่อกับเครื่องคอมพิวเตอร์เพื่อทำการป้อนข้อมูลหรือคำสั่งต่างๆ ให้กับเครื่องคอมพิวเตอร์ก็คือ การใช้แป้นพิมพ์หรือคีย์บอร์ด (Keyboard) การป้อนข้อมูลผ่านทางแป้นพิมพ์นี้เป็นวิธีการดังเดิมที่ใช้กันมาตั้งแต่เริ่มแรกที่ทำการคิดค้นประดิษฐ์เครื่องคอมพิวเตอร์ขึ้นใช้งาน และก็ยังคงใช้ต่อเนื่องกันมาจนกระทั่งปัจจุบัน โดยทั่วๆ ไปแล้ววิธีการนี้กล่าวได้ว่าจะเป็นวิธีที่ง่ายและสะดวกสบายสำหรับผู้ใช้ที่มีความชำนาญ หรือเป็นผู้ที่สามารถพิมพ์ดีดได้คล่องแคล่ว แต่สำหรับผู้ใช้ที่ยังขาดความชำนาญโอกาสผิดพลาดในการใช้งานแป้นพิมพ์ย่อมเกิดขึ้นได้ง่ายและเสียเวลาไปไม่น้อยเนื่องจากขาดความคุ้นเคย แต่ปัญหาที่สำคัญที่สุดนั้นคือ กรณีที่ข้อมูลที่จะต้องจัดเก็บมีปริมาณมากๆ จำเป็นต้องจัดหาบุคคลากรและเครื่องไม้เครื่องมือเพิ่มขึ้น การแก้ปัญหาโดยวิธีนี้เป็นการแก้ปัญหาแบบผิดๆ ซึ่งจะเป็นการสิ้นเปลืองทั้งงบประมาณและบุคคลากรเป็นอันมาก การแก้ปัญหาที่ถูกต้องคือการนำเทคโนโลยีใหม่ๆ มาทดแทน การใช้เครื่องอ่านตัวอักษรด้วยแสง ก็เป็นวิธีหนึ่งที่น่าสนใจซึ่งจะนำมากล่าวในที่นี้

ในปัจจุบันวิวัฒนาการทางด้านเทคโนโลยีคอมพิวเตอร์ได้เจริญรุดหน้าไปอย่างรวดเร็ว เครื่องคอมพิวเตอร์มีประสิทธิภาพในการทำงานสูงขึ้น และมีอุปกรณ์สนับสนุนการทำงานอื่นๆ มีมากมาย อุปกรณ์สนับสนุนการทำงานของเครื่องคอมพิวเตอร์ชนิดหนึ่งที่จะกล่าวถึงก็คือ เครื่องอ่านตัวอักษรด้วยแสง (Optical Character Reader) การทำงานของเครื่องอาศัยการฉายแสงลงบนแผ่นกระดาษแล้วนำปริมาณแสงที่สะท้อนกลับมาเปลี่ยนไปเป็นสัญญาณไฟฟ้า ซึ่งสัญญาณไฟฟ้าที่เป็นตัวแทนของตัวอักษรจะถูกนำไปเปลี่ยนเป็นรหัสข้อมูลทางดิจิตอลที่จะถูกนำไปเก็บไว้ในหน่วยความจำของเครื่องคอมพิวเตอร์ เพื่อส่งต่อไปให้เครื่องคอมพิวเตอร์ทำการวิเคราะห์และรู้จำต่อไปว่าตัวอักษรนั้นเป็นตัวอักษรอะไร

จากนี้ไปปัญหาของการจัดเก็บข้อมูลจำนวนมากก็สามารถแก้ไขได้โดยอาศัยเครื่องอ่านตัวอักษรด้วยแสงนี้ อ่านตัวอักษรแล้วนำไปเก็บไว้ในเครื่องคอมพิวเตอร์ทดแทนการใช้แป้นพิมพ์ และไม่ว่าปริมาณของข้อมูลจะมีมากน้อยเพียงใด ก็สามารถจัดเก็บได้เสร็จสิ้นอย่างรวดเร็ว โดยผู้ใช้ไม่จำเป็นต้องเป็นผู้ที่มีความชำนาญในการใช้แป้นพิมพ์อีกต่อไป ซึ่งจะช่วยประหยัดทั้งงบประมาณ เวลา และบุคคลากรเป็นจำนวนมาก

ในประเทศไทยคอมพิวเตอร์ได้เข้ามามีบทบาทอย่างมากในเกือบจะทุกวงการไม่ว่าจะเป็นในภาครัฐหรือภาคเอกชนเช่นเดียวกับในหลายๆ ประเทศทั่วโลก มีการพัฒนาระบบคอมพิวเตอร์ในหลายๆ ด้านให้เหมาะสมกับการใช้งานในประเทศ ไม่ว่าจะเป็นการ การพัฒนาซอฟแวร์ขึ้นใช้เอง หรือจะเป็นการพัฒนาทางด้านฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และซอฟแวร์สำหรับระบบภาษาไทย เมื่อกล่าวถึงระบบการรู้จำตัวอักษรที่มีการศึกษาวิจัยกันในต่างประเทศ พบว่ายังไม่สามารถนำวิธีการเดียวกันมาใช้กับตัวอักษรภาษาไทยได้ในทันที เป็นเหตุให้มีการศึกษาวิจัยให้เครื่องคอมพิวเตอร์สามารถที่จะรับรู้และเข้าใจตัวอักษรภาษาไทยเพิ่มขึ้นด้วย เพื่อการใช้งานเครื่องคอมพิวเตอร์ในประเทศจะเป็นไปอย่างเต็มประสิทธิภาพ รวมถึงสามารถแก้ไขปัญหาและอุปสรรคในการทำงานดังที่ได้กล่าวมาแล้วข้างต้น การศึกษาค้นคว้าวิจัยเป็นไปอย่างกว้างขว้าง มีการนำเสนอวิธีการหลายๆ วิธี ได้ผลลัพธ์และประสิทธิภาพการทำงานที่แตกต่างกันไปตามสภาวะแวดล้อมของการทำงาน ซึ่งงานวิจัยที่ผ่านๆ มาทั้งหลายนี้ล้วนแล้วแต่เป็นประโยชน์และเป็นรากฐานที่สำคัญในการพัฒนาระบบต่อไปในอนาคต สำหรับวิทยานิพนธ์นี้ก็จะเป็นอีกวิธีหนึ่งที่ได้ทำการวิจัยขึ้นมา

1.2 ผลงานวิจัยที่ผ่านมา

ที่ผ่านมาได้มีนักวิจัยผู้ทรงคุณวุฒิหลายท่านทำการศึกษาค้นคว้าในเรื่องการรู้จำตัวอักษรทั้งภาษาไทยและภาษาอังกฤษไว้มากมาย ซึ่งพอจะยกตัวอย่างและสรุปวิธีการทำงานพร้อมผลการทดลองมาพอสังเขปได้ดังต่อไปนี้

1.2.1 ผลงานวิจัยของรศ.ดร.ชม กิ้มปาน เรื่อง "การจดจำรูปแบบตัวพิมพ์อักษรภาษาไทย"[1] โดยนำเสนอวิธีการไว้ 2 วิธีคือ

1.2.1.1 การรู้จำแบบซ้อนทับ[2] (Matching method) เป็นวิธีการที่ได้แนวคิดจากรูปร่างลักษณะของตัวอักษรภาษาไทยที่มีรูปร่างลักษณะการเขียนที่คล้ายๆกันอยู่หลายตัว ซึ่งสามารถนำมาจัดเป็นกลุ่มๆ อย่างคร่าวๆ ได้ ในส่วนของการรู้จำได้แบ่งการทำงานออกเป็น 2 ขั้นตอนใหญ่ๆ คือขั้นตอนของการแบ่งหยาบ ที่นำเทคนิคการทำเบลอมาใช้เพื่อที่จะกำจัดลายละเอียดปลีกย่อยเกินไปออกเสีย แล้วนำทฤษฎีคณิตศาสตร์ชั้นสูงของการกระจายแบบคาร์ฮูแนนโลบของไอเกนเวคเตอร์ที่มีค่าไอเกนสูงสุดมาคำนวณหารูปแบบมาตราฐานสำหรับตัวอักษรในแต่ละกลุ่มย่อย แล้วในขั้นตอนของการแบ่งอย่างละเอียดจะเป็นการซ้อนทับตัวอักษรที่ต้องการรู้จำกับรูปแบบมาตราฐาน ส่วนผลของการรู้จำให้ผลลัพธ์เป็นที่น่าพอใจ ความถูกต้องสูงกว่า 90 %

1.2.1.2 วิธีการวิเคราะห์โครงสร้าง[5] (Structural analysis method) วิธีการนี้ได้นำเอาคุณสมบัติทางโทโพโลยี่มาช่วยในการพิจารณา ภาพตัวอักษรที่ต้องการรู้จำจะถูกทำให้บาง แล้วจึงถูกแบ่งกลุ่มออกแบบหยาบด้วยฟังก์ชั่นการตัดสินใจเชิงเส้นโดยพิจารณาจากความกว้างและความสูงของตัวอักษร ในส่วนของการแบ่งละเอียดเป็นการคำนวณหาตัวเลขแสดงความต่อเนื่อง (Connected numbers) ของจุดโครงร่างตัวอักษร จากนั้นคุณสมบัติทางโทโพโลยี่จะถูกนำมาใช้แบ่งแยกความแตกต่างของตัวอักษร ผลของการรู้จำที่ได้เป็นที่น่าพอใจเช่นเดียวกับวิธีแรก แต่จากขบวนการการทำให้บางอาจทำให้โครงร่างของตัวอักษรขาดหายไปบ้าง ซึ่งจะทำให้ผลของการรู้จำผิดพลาดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากงานวิจัยทั้งสองผลงานของ รศ.ดร. ชม กิ้มปาน ทำการประมวลผลโดยเครื่องคอมพิวเตอร์ขนาดใหญ่ระดับเมนเฟรมที่มีประสิทธิภาพการทำงานสูง แต่ว่าขั้นตอนการทำงานกับเครื่องค่อนข้างยุ่งยากซับซ้อน ไม่สะดวกและคล่องตัวเหมือนเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) จึงยังไม่ได้มีการนำมาใช้งานอย่างจริงจัง

1.2.2 ผลงานวิจัยของ นายประสาร ดังติสานนท์ เรื่อง "การจดจำรูปแบบตัวอักษรคัดลายมือภาษาไทยโดยวิธีแยกลักษณะเด่น"[6] ซึ่งพิจารณาถึงโครงสร้างของรูปแบบตัวอักษรทั้งลายเส้นและพื้นเบื้องหลังของลายเส้นตัวอักษรควบคู่กันไป ขั้นตอนทำงานแบ่งออกเป็น 3 ขั้นตอนใหญ่คือ ขั้นตอนแรกเป็นการเปลี่ยนลักษณะลายเส้นและลักษณะลายพื้นเบื้องหลังลายเส้นไปเป็นรหัสเบื้องต้น (Initial feature extraction) ขั้นตอนที่สองเป็นการลดทอนรหัสเบื้องต้นที่ซ้ำซ้อน (Unification) ขั้นตอนที่สามเป็นการรวมรหัสลายเส้นและลายพื้นเข้าด้วยกันเพื่อให้เป็นคุณสมบัติของรูปแบบตัวอักษร (Concentration) ผลการรู้จำให้ความถูกต้องสูงเกินกว่า 90 % แต่ว่าเวลาที่ใช้ในการรู้จำตัวอักษรแต่ละตัวต้องใช้เวลาประมาณ 3 นาที ซึ่งใช้เวลานานเกินไป จึงไม่สามารถนำออกมาใช้งานได้จริง

1.2.3 ผลงานวิจัยของ นายสุรพันธุ์ เอื้อไพบูลย์ เรื่อง "การจดจำตัวอักษรลายมือเขียนภาษาไทยโดยการพิจารณาหัวของตัวอักษร"[7] ทำการเลือกพิจารณาส่วนหัวของตัวอักษรภาษาไทยที่มีลักษณะเป็นวงกลมเล็กๆ ขั้นตอนการทำงานแบ่งออกเป็น 4 ขั้นตอนคือ ขั้นตอนที่หนึ่งเป็นการเตรียมภาพตัวอักษรแต่ละตัวไว้ 2 ชุด โดยชุดหนึ่งจะเป็นภาพปกติ และอีกชุดหนึ่งจะเป็นภาพตัวอักษรที่ผ่านการทำให้บางแล้ว ขั้นตอนที่สอง จะทำการค้นหาส่วนหัวจากตัวอักษรปกติ และคำนวณหาคุณสมบัติทางโทโพโลยี่จากตัวอักษรที่ถูกทำให้บาง ในขั้นตอนที่สามเป็นการแบ่งตัวอักษรออกเป็นกลุ่มโดยพิจารณาจำนวนหัวของตัวอักษร และในขั้นตอนสุดท้าย จะเป็นการวิเคราะห์ตัวอักษรด้วยเทคนิคพิเศษ 4 เทคนิคได้แก่ 1.Subhead Region 2.Feature Code 3.Head Style และ 4.Width per Height ratio การทำงานของระบบใช้กล้องเก็บภาพ CCD เป็นอุปกรณ์ป้อนข้อมูล (Input device) ซึ่งการใช้งานคงยุ่งยากพอสมควร และต้องพึ่งผู้ที่มีความชำนาญเป็นผู้ปรับแต่งความคมชัดในการถ่ายเก็บภาพ และในส่วนของอัลกอริทึมที่ใช้ เป็นการพิจารณาเฉพาะส่วนหัวของตัวอักษรซึ่งค่อนข้างจะเป็นการพิจารณาเฉพาะเจาะจงเกินไป หากข้อมูลเป็นภาพตัวอักษรที่เขียนแบบไม่มีหัว หรือลายเส้นของหัวขาดไม่สมบูรณ์หรือไม่เป็นวงปิด หรือไม่ว่าหัวมีลักษณะทึบตัน ผลของการรู้จำย่อมผิดพลาดอย่างแน่นอน แต่อย่างไรก็ตาม หากทุกอย่างเป็นไปตามเงื่อนไขที่ผู้ทำวิจัยกำหนดไว้งานวิจัยนี้สามารถรู้จำตัวอักษรลายมือเขียนให้ผลของการรู้จำที่ดีพอสมควร

1.2.4 ผลงานวิจัยของ นายชาย เกษมอมรกุล เรื่อง "การออกแบบพจนานุกรมสำหรับการเรียนรู้อักษรคัดลายมือไทย-อังกฤษอัตโนมัติบนเครื่องไมโครคอมพิวเตอร์"[8] เป็นการพัฒนาปรับปรุงผลงานวิจัยของนายประสาน ดังติสานนท์ มีการเพิ่มเติมส่วนของการรู้จำตัวอักษรโดยค้นหาคุณสมบัติเด่นของตัวอักษรจากพจนานุกรมของการรู้จำที่จัดเก็บโดยพิจารณาจากความถี่ของการใช้งานของอักษรแต่ละตัว ทำให้การค้นหาทำได้อย่างรวดเร็ว นอกจากนั้นตัวระบบเองยังสามารถที่จะรู้จำตัวอักษรที่ไม่มีข้อมูลปรากฏในพจนานุกรมได้โดยอัตโนมัติโดยอาจให้ผู้ใช้ช่วยตัดสินใจบ้าง ส่วนของพจนานุกรมสามารถเพิ่มเติมตัวเองได้โดยอัตโนมัติเช่นกัน ประสิทธิภาพการรู้จำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกปรับปรุงให้ดีขึ้นมาก โดยเฉพาะอย่างยิ่งในเรื่องของเวลาที่ใช้ในการรู้จำ แต่ในการเขียนลายมือคัดเพื่อเป็นข้อมูลในการวิเคราะห์ ยังจำกัดขนาดตัวอักษร และจำนวนบรรทัด จึงยังไม่เหมาะที่จะนำไปใช้งานจริง

1.2.5 ผลงานวิจัยของนายสุรสิทธิ์ ราตรี เรื่อง "การรู้จำอักษรตัวพิมพ์ภาษาไทยโดยวิธีการค้นหาลักษณะโครงสร้างของลายเส้น"[9] ได้นำเสนอวิธีการค้นหาลักษณะโครงสร้างลายเส้นซึ่งเป็นลักษณะเด่นของตัวอักษร (Topological feature extraction) การพิจารณากระทำกับโครงสร้างของลายเส้นตัวอักษรที่ถูกแบ่งออกเป็น 8 ส่วน ในแนวตั้ง แนวนอน และในแนวเส้นทะแยงมุมทั้งสอง ในขั้นตอนต่อไปทำการแยกพิจารณาในแต่ละส่วน (Octant) หาคุณสมบัติทางโทโพโลยี่ของลายเส้น (Stroke) ของตัวอักษร ซึ่งจะได้รหัสแทนตัวอักษรที่เรียกว่า A-code จากการรวมรหัสแสดงคุณลักษณะเด่นในแต่ละส่วนของตัวอักษรที่ถูกแบ่งทั้ง 8 ส่วน เนื่องจากการพิจารณากระทำกับลักษณะโครงสร้างของลายเส้นของตัวอักษรอย่างละเอียดจึงสามารถเก็บคุณลักษณะเด่นของตัวอักษรได้ครบถ้วน ส่งผลให้ความถูกต้องของการรู้จำมีความถูกต้องสูงเกินกว่า 95 % แต่การแบ่งพื้นที่การพิจารณาออกเป็นส่วนๆ ที่มีพื้นที่เล็กเกินไป จะทำให้การพิจารณาโครงสร้างของลายเส้นในแต่ละส่วนอาจไม่ได้ลักษณะลายเส้นที่แท้จริงในกรณีที่ลายเส้นตัวอักษรที่มีความหนา แต่แนวทางแก้ไขสามารถทำได้โดยเพิ่มเติมส่วนของการทำตัวอักษรให้บาง ซึ่งจะได้โครงร่างของลายเส้นตัวอักษรที่แท้จริง แล้วจึงนำไปใช้พิจารณา ก็จะสามารถแก้ปัญหาส่วนนี้ได้ และคาดว่าจะได้ความถูกต้องแน่นอนแม้ว่ารูปแบบ (Font) ของตัวอักษรอาจจะแตกต่างกัน

1.3 แนวทางของวิทยานิพนธ์

จากผลงานวิจัยที่ผ่านๆ มา ระบบการทำงานมุ่งเน้นไปในทางด้านการค้นคิดวิธีการรู้จำตัวอักษรให้ได้ประสิทธิภาพและความถูกต้องของการรู้จำสูงสุดเพียงด้านเดียว ทำให้ละเลยไม่ได้กล่าวถึงส่วนของการเตรียมข้อมูลสำหรับการรู้จำ[2],[5],[6],[7],[8] หรืออาจกล่าวไว้เพียงคร่าวๆ และสามารถทำงานได้เพียงในขอบเขตที่จำกัด[9] ส่วนของการเตรียมข้อมูลนี้เป็นขั้นตอนหนึ่งของส่วนจัดการล่วงหน้าของระบบ (Preprocessing process) ที่มีความสำคัญไม่น้อยไปกว่าส่วนของรู้จำ (Recognition) ในการประยุกต์ใช้งานจริงของระบบการรู้จำตัวอักษร อินพุทของระบบจะมีลักษณะเป็นเอกสารบนแผ่นกระดาษที่บรรจุข้อความเป็นบรรทัดหรือเป็นย่อหน้า และยังอาจจะมีรูปภาพกราฟิกเป็นส่วนอธิบายข้อความรวมอยู่ด้วย เพื่อให้ระบบการรู้จำตัวอักษรทำงานได้อย่างมีประสิทธิภาพและถูกต้องแม่นยำ จึงจำเป็นจะต้องมีขบวนการทำงานเข้ามาจัดการกับงานในส่วนนี้ นั่นคือส่วนจัดการล่วงหน้าโดยจะทำการวิเคราะห์และระบุส่วนประกอบหน้าเอกสาร แยกส่วนของรูปภาพที่ไม่ใช่ส่วนที่ต้องการรู้จำจัดเก็บไว้อย่างถูกต้องและสามารถนำกลับมาใช้งานได้อีกภายหลังเมื่อต้องการ ส่วนของข้อความที่ประกอบด้วยตัวอักษรเรียงต่อกันเป็นคำ เป็นประโยคหรือเป็นย่อหน้าก็ตาม ก็จะต้องมีขบวนการทำงานที่สามารถแยกตัวอักษรเป็นตัวอักษรเดี่ยวๆ (Isolated character) ก่อนส่งต่อไปให้กับส่วนของการรู้จำ

วิทยานิพนธ์ฉบับนี้จึงขอนำเสนอระบบการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารสำหรับการรู้จำตัวอักษร ซึ่งประกอบด้วยขั้นตอนการทำงาน 3 ส่วนใหญ่ๆ ที่พอจะอธิบายโดยสังเขปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 1 การอ่านข้อมูลจากแฟ้มข้อมูลกราฟิก

เป็นการอ่านแฟ้มข้อมูลกราฟิกของภาพหน้าเอกสารที่ได้จากเครื่องกวาดตรวจภาพ (Image scanner) ซึ่งใช้เป็นอุปกรณ์อินพุทให้กับระบบ แล้วทำการแปลงข้อมูลกราฟิกให้เป็นข้อมูลไบนารี่ (Binary data) สำหรับการประมวลของเครื่องคอมพิวเตอร์ในขั้นตอนต่อไป

ส่วนที่ 2 การวิเคราะห์และระบุส่วนประกอบบนหน้าเอกสาร

ในส่วนนี้ได้นำเอาเทคนิคใหม่ นั่นคือการค้นหาขอบเขตโดยอาศัยทฤษฎีเซลลูล่าออโตมาตา (Zone detection using Cellular Automata) มาประยุกต์ใช้ในการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร ส่วนที่เป็นรูปภาพกราฟิกจะทำการจัดเก็บในรูปแบบ (Format) ของแฟ้มข้อมูลกราฟิกรูปแบบ TIFF อย่างถูกต้อง และส่วนที่เป็นข้อความจะถูกส่งผ่านไปยังส่วนของการรู้จำต่อไป

ส่วนที่ 3 การรู้จำตัวอักษร

ประกอบด้วยส่วนของการแยกตัวอักษรออกจากคำหรือประโยค (Segmentation) ที่อาศัยเทคนิคของการติดตามรอยขอบของภาพ แยกภาพตัวอักษรออกได้อย่างถูกต้อง ในส่วนของการรู้จำ ภาพตัวอักษรแต่ละตัวจะถูกแบ่งออกเป็น 8 ส่วน ณ.จุดศูนย์ถ่วงของตัวอักษร และในแต่ละส่วนจะอาศัยคุณสมบัติทางโทโพโลยี่ของลายเส้นตัวอักษรกำหนดเป็นรหัส (Q-code) ขึ้น ณ.จุดศูนย์ถ่วงในส่วนนั้นๆ โดยที่ตัวอักษรแต่ละตัวจะมีรหัสแทนเป็นเลขฐาน 16 จำนวน 9 หลัก ซึ่งรหัสแทนตัวอักษรทั้งหมดจะถูกจัดเก็บไว้เป็นพจนานุกรมของการรู้จำต่อไป

เนื้อหาทั้งหมดของวิทยานิพนธ์ฉบับนี้ ซึ่งประกอบด้วยขั้นตอนการทำงาน การทดลอง และผลของการทดลอง ได้อธิบายไว้อย่างละเอียดโดยแยกออกเป็นบทตามลำดับขั้นของการทำงานดังต่อไปนี้

บทที่ 2 กล่าวถึงอุปกรณ์ที่ใช้ในการเก็บข้อมูลและลักษณะของข้อมูล

2.1 เครื่องตรวจกวาดภาพ

2.1.1 ลักษณะและการทำงานของเครื่องตรวจกวาดภาพ

2.1.2 เครื่องตรวจกวาดภาพที่ใช้ในงานวิจัย

2.2 ลักษณะของข้อมูล

2.2.1 ลักษณะของข้อมูลกราฟิก

2.2.2 การแปลงภาพกราฟิกเป็นข้อมูลไบนารี่

2.3 ตัวอย่างข้อมูลที่ใช้ในงานวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 กล่าวถึงการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร

3.1 บทนำ

3.2 ทฤษฎีที่นำมาใช้

3.2.1 ทฤษฎีเซลลูล่าออโตมาตา

3.2.2 เทคนิคการติดตามรอยขอบของภาพ

3.3 การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร

3.3.1 การกำหนดขอบเขต

3.3.2 การแยกและคัดลอกขอบเขต

3.4 การจัดเก็บส่วนประกอบของหน้าเอกสาร

บทที่ 4 กล่าวถึงการรู้จำตัวอักษร

4.1 บทนำ

4.2 ทฤษฎีที่นำมาใช้

4.2.1 ทฤษฎีฟีเจอร์คอนเซนเทรชั่น

4.2.2 การคำนวณจุดศูนย์ถ่วงของภาพ

4.2.3 การคำนวณความกว้างและความสูงของภาพตัวอักษร

4.3 การแยกภาพตัวอักษรออกจากภาพประโยค

4.4 การพิจารณาลักษณะเด่นของภาพตัวอักษร

4.4.1 การปรับแต่งจุดอ้างอิง

4.4.2 การพิจารณาคุณสมบัติทางโทโพโลยี่ของลายเส้นของภาพตัวอักษร

4.5 การจัดเรียงรูปประโยคหลังการวิเคราะห์

บทที่ 5 กล่าวถึงโปรแกรมระบบและผลการทดลอง

5.1 ขั้นตอนการทำงานของระบบ

5.1.1 โปรแกรมสำเร็จรูปสำหรับเก็บภาพจากเครื่องตรวจกวาดภาพ

5.1.2 โปรแกรมระบบที่พัฒนาขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2.1 อุปกรณ์ที่ต้องการสำหรับการทำงานของระบบ

5.1.2.2 ขั้นตอนการทำงานของโปรแกรมที่พัฒนาขึ้น

5.2 การทดลองการทำงานของโปรแกรม

บทที่ 6 กล่าวถึงบทสรุปของวิทยานิพนธ์

6.1 สรุปการทำงานของระบบ

6.1.1 ข้อมูลอินพุทกับการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร

6.1.2 ข้อมูลภาพตัวอักษรกับการรู้จำ

6.2 ข้อจำกัดของการทำงานของระบบ

6.2.1 ข้อจำกัดของการทำงานในส่วนของการวิเคราะห์

และระบุส่วนประกอบของหน้าเอกสาร

6.2.2 ข้อจำกัดของการทำงานในส่วนของการรู้จำตัวอักษร

6.3 ข้อเสนอแนะสำหรับแนวทางพัฒนาระบบในอนาคต

ภาคผนวก ก. แฟ้มข้อมูลกราฟิกรูปแบบ TIFF Version 5.0

ภาคผนวก ข. ตารางรหัสแทนอักขระ

ภาคผนวก ค. รูปแบบตัวอักษรและคา Q-code

ภาคผนวก ง. โปรแกรมระบบที่พัฒนาขึ้น

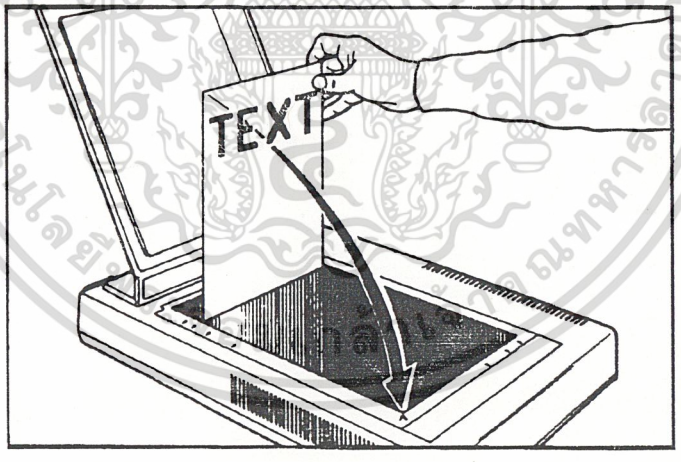
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

อุปกรณ์ที่ใช้ในการเก็บข้อมูลและลักษณะของข้อมูล

2.1 เครื่องตรวจกวาดภาพ

เครื่องตรวจกวาดภาพหรืออิมเมจสแกนเนอร์ (Image scanner) เป็นอุปกรณ์ช่วยสนันสนุนการทำงานของเครื่องคอมพิวเตอร์ชนิดหนึ่ง ทำหน้าที่เสมือนตาของเครื่องคอมพิวเตอร์ ใช้เก็บรูปภาพหรือภาพตัวอักษรที่พิมพ์อยู่บนแผ่นกระดาษ นำเข้าไปเก็บไว้ในหน่วยความจำหรืออุปกรณ์เก็บข้อมูลสำรองของเครื่องคอมพิวเตอร์ เช่น ฟลอปปี้ดิสก์ หรือฮาร์ดดิสก์ เป็นต้น ได้มีการนำเอาเครื่องสแกนเนอร์มาประยุกต์ใช้เป็นเครื่องอ่านตัวอักษรด้วยแสง (Optical Character Reader, OCR) เพื่อใช้ในงานวิจัยทางด้านการรู้จำตัวอักษร (Character recognition) โดยทำหน้าที่เป็นอุปกรณ์เก็บข้อมูล ตัวอย่างข้อมูลอาจมีลักษณะเป็นหน้ากระดาษ ที่ประกอบด้วยรูปภาพและตัวอักษรพิมพ์เรียงเป็นประโยคข้อความ เมื่อนำแผ่นกระดาษดังกล่าวมาตรวจกวาดหรือสแกนด้วยสแกนเนอร์ ภาพที่ปรากฏบนหน้าเอกสารทั้งหมดไม่ว่าจะเป็นรูปภาพหรือตัวอักษรจะถูกเปลี่ยนเป็นข้อมูลไบนารี่และเก็บบันทึกลงแฟ้มข้อมูลในรูปแบบหรือฟอร์แมตแบบกราฟิก ซึ่งจะถูกนำไปประมวลผลในขั้นตอนต่อๆ ไป



รูปที่ 2.1 แสดงตัวอย่างภาพเครื่องตรวจกวาดภาพ

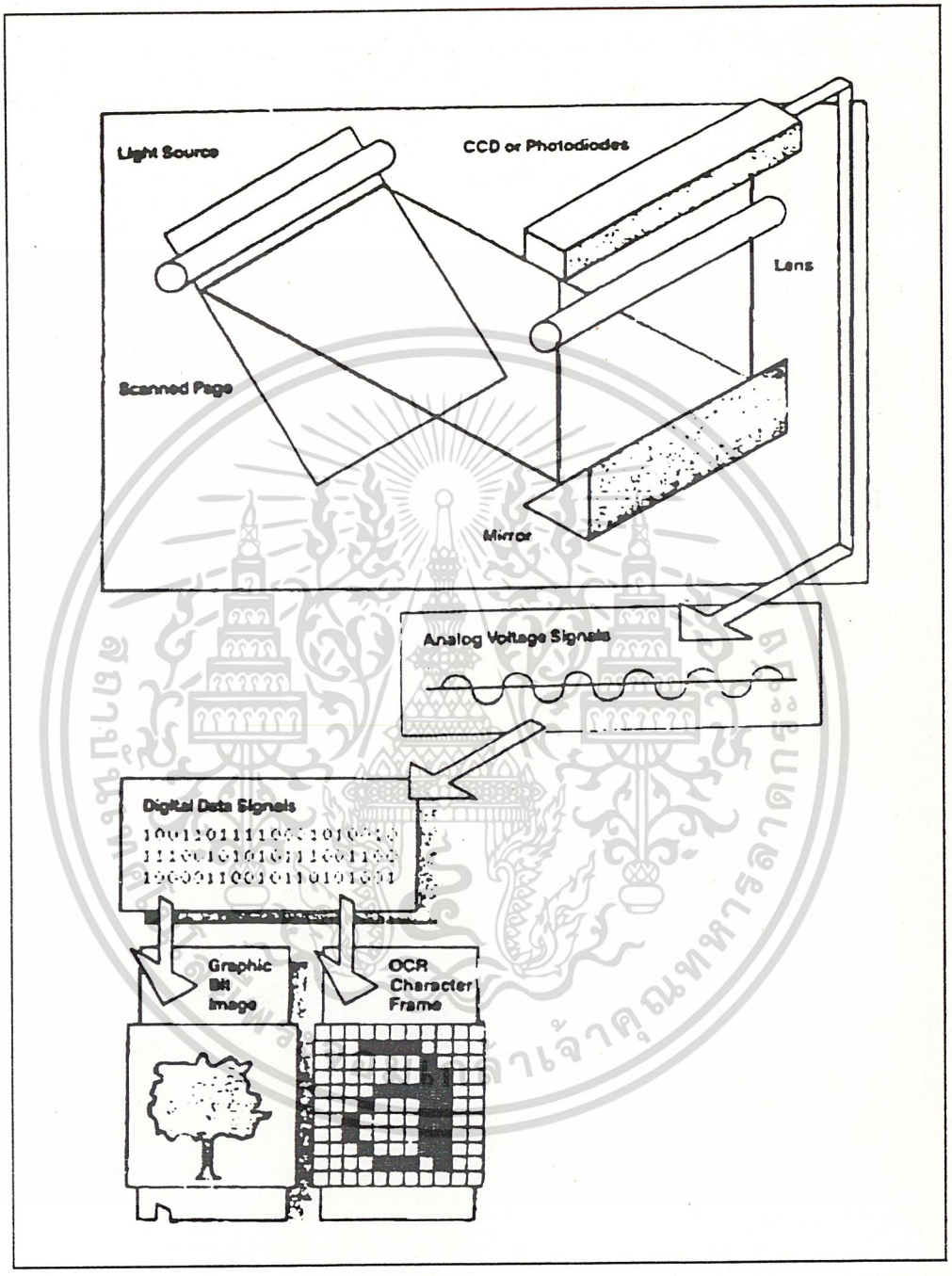
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 ลักษณะและการทำงานของเครื่องตรวจกวาดภาพ

เครื่องตรวจกวาดภาพทำงานโดยอาศัยหลักการสะท้อนของแสง เมื่อฉายแสงไปบนแผ่นกระดาษที่มีรูปภาพหรือข้อความพิมพ์อยู่ แล้วเก็บบันทึกแสงที่สะท้อนกลับออกมาในปริมาณที่ต่างกัน ส่วนประกอบของเครื่องตรวจกวาดภาพ ภายในตัวเครื่องจะมีหลอดฟลูออเรสเซนต์ ที่จะให้แสงความถี่ต่ำฉายออกมากระทบกับหน้ากระดาษที่จะตรวจกวาด ส่วนที่เป็นพื้นสีดำซึ่งอาจจะเป็นข้อความหรือรูปภาพ ก็จะดูดซับแสงเอาไว้หรือสะท้อนออกมาบางส่วน และส่วนที่เป็นพื้นขาวก็จะสะท้อนแสงออกมามากกว่า จากนั้นจะมีตัวรับสัญญาณภาพ (Photosensor) ที่เรียกว่า CCD (Charge Couple Device) ซึ่งจะประกอบด้วยเซลรับแสงเป็นจำนวนมากเรียงเป็นแถว โดยที่เซลแต่ละตัวจะสร้างความต่างศักย์ทางไฟฟ้าขึ้นเป็นสัดส่วนกับปริมาณแสงที่รับเข้ามา ค่าความต่างศักย์จะถูกแปลงให้เป็นค่าทางตัวเลข ไบนารี่ (binary data; 0 หรือ 1)

ความละเอียด (Resolution) ของเครื่องตรวจกวาดภาพจะวัดกันด้วยค่าจุดต่อนิ้ว (Dot per inch, dpi) ซึ่งเป็นค่าตัวเลขแน่นอนค่าหนึ่งที่ได้จากความสัมพันธ์ของจำนวนเซลในหนึ่งอาร์เรย์ และพื้นที่ทั้งหมดที่ตรวจกวาด ยกตัวอย่างเช่น อาร์เรย์ประกอบด้วยเซลรับแสงจำนวน 2,400 หน่วย สามารถครอบคลุมพื้นที่ในแนวตรงได้ 8 นิ้ว จะได้ค่าความละเอียดของเครื่องตรวจกวาดภาพ (Hardware resolution) เป็น 300 dpi ซึ่งเมื่อนำมาใช้ตรวจกวาดรูปภาพ ก็จะได้ภาพที่มีความละเอียดของภาพ (Image resolution) ขนาด 300 dpi เครื่องตรวจกวาดภาพทั่วๆ ไปยังสามารถตรวจกวาดภาพและให้ความละเอียดที่ต่ำกว่าหรือสูงกว่าได้ โดยอาศัยซอฟต์แวร์ช่วยในการจัดการกำหนดความละเอียดของภาพ การทำงานของเครื่องตรวจกวาดภาพสามารถแสดงได้ดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

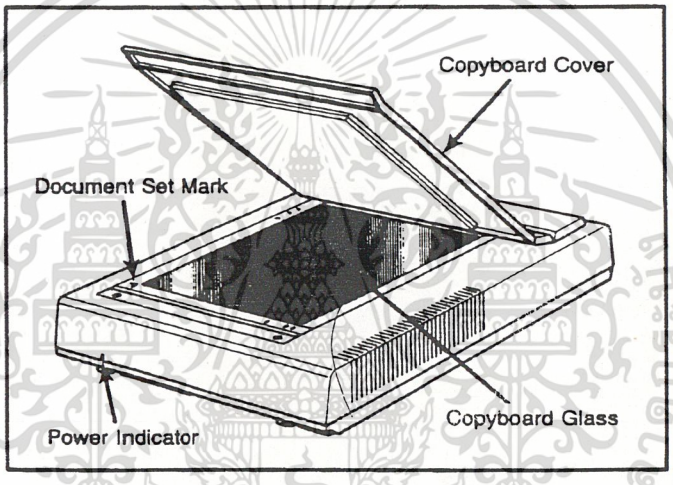


รูปที่ 2.2 แสดงหลักการทำงานของเครื่องตรวจกวาดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 เครื่องตรวจกวาดภาพที่ใช้ในงานวิจัย[21]

เครื่องกวาดตรวจกวาดภาพที่ใช้ในงานวิจัยนี้เป็นเครื่องตรวจกวาดภาพชนิดตั้งโต๊ะ ผลิตโดยบริษัท Hewlett Packard รุ่น ScanJet สามารถตรวจกวาดภาพเอกสารที่มีขนาดใหญ่ได้ถึงขนาดกระดาษ A4 และสามารถเลือกระดับความละเอียดของภาพ (Scanning resolution) ได้ตั้งแต่ 38 - 600 จุดต่อนิ้ว (Dots per inch) สามารถปรับค่าความละเอียดของภาพได้โดยอาศัยซอฟแวร์ รูปภาพหรือข้อความที่ตรวจกวาดได้จะถูกจัดเก็บลงแฟ้มข้อมูลกราฟิกในฟอร์แมต TIFF ข้อมูลที่จัดเก็บมีลักษณะเป็น Packed binary data (มีค่าเป็น ตัวเลขค่า 0 กับ 1) โดยที่ข้อมูลที่มีค่าเป็น 1 จะแทนส่วนที่เป็นพื้นดำที่เป็นข้อความหรือรูปภาพ และข้อมูลที่มีค่าเป็น 0 จะแทนส่วนที่เป็นพื้นสว่างที่เป็นช่องว่างบนแผ่นกระดาษ



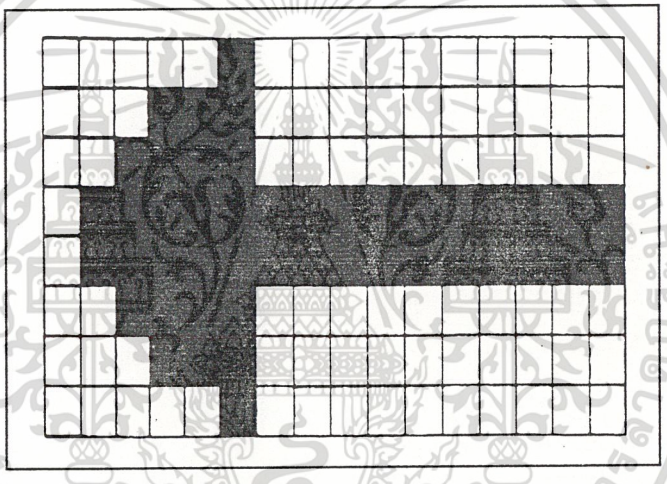
รูปที่ 2.3 แสดงภาพเครื่องตรวจกวาดภาพที่ใช้ในงานวิจัยนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ลักษณะของข้อมูล

2.2.1 ลักษณะของข้อมูลภาพกราฟิก[22]

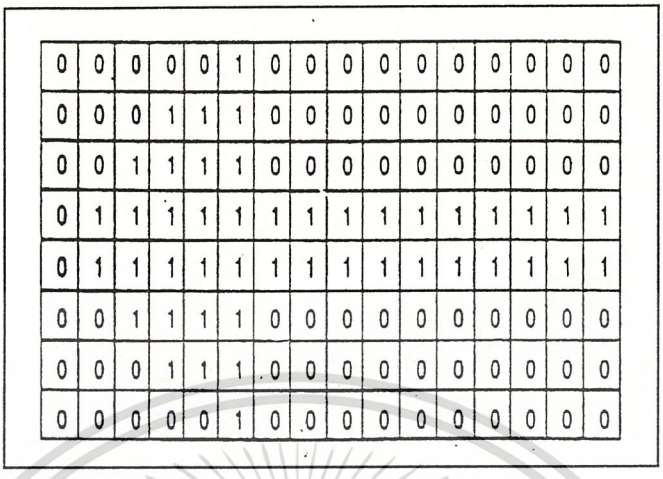
เนื่องจากข้อมูลที่จัดเก็บได้จากเครื่องตรวจกวาดภาพจะอยู่ในแฟ้มข้อมูลภาพกราฟิก นั่นคือเป็นตัวเลขค่าไบนารี่คือ เลข 0 กับ 1 ของจุดภาพหนึ่งๆ ถูกย่อรวมกันจุดเก็บอยู่ในรูปตัวเลขฐาน 16 ก่อนที่จะนำข้อมูลดังกล่าวมาป้อนให้กับเครื่องคอมพิวเตอร์เพื่อทำการประมวลผลต่อไป จะต้องแปลงข้อมูลกราฟิกในรูปเลขฐาน 16 นี้กลับให้เป็นข้อมูลไบนารี่ในรูปตัวเลข 0 กับ 1 เสียก่อน แฟ้มข้อมูลกราฟิกที่ได้จากเครื่องกวาดตรวจภาพที่ใช้ในงานวิจัยนี้ เป็นแฟ้มข้อมูลรูปแบบ TIFF มีลักษณะเป็นบิทแม็พ (Bitmap pattern) ซึ่งก็คือกลุ่มของจุดสี่เหลี่ยมอาจจะเป็นสีขาวหรือสีดำมาเรียงต่อกันเป็นตาราง ซึ่งจะต่อรวมกันเป็นแสดงออกมาเป็นรูปภาพ ตัวอย่างของรูปภาพแบบบิตแม็พแสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 แสดงตัวอย่างของ Bitmap pattern

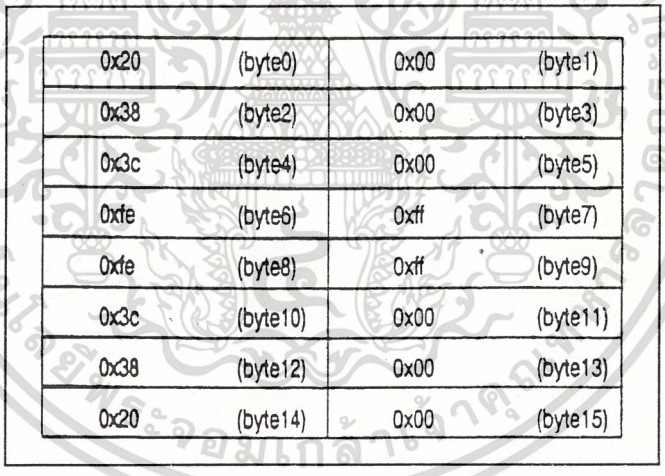
Bitmap patten ที่แสดงดังรูปที่ 2.4 สามารถที่จะแสดงให้อยู่ในรูปของตัวเลข 0 กับ 1 ได้ดังรูปที่ 2.5 และ Bitmap data ซึ่งเป็นข้อมูลที่เก็บบันทึกจริงในแฟ้มข้อมูลกราฟิกจะมีลักษณะเป็นตัวเลขฐาน 16 ดังแสดงในรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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

รูปที่ 2.5 แสดง Bitmap pattern ในรูปของตัวเลข 0 กับ 1



0x20	(byte0)	0x00	(byte1)
0x38	(byte2)	0x00	(byte3)
0x3c	(byte4)	0x00	(byte5)
0xfe	(byte6)	0xff	(byte7)
0xfe	(byte8)	0xff	(byte9)
0x3c	(byte10)	0x00	(byte11)
0x38	(byte12)	0x00	(byte13)
0x20	(byte14)	0x00	(byte15)

รูปที่ 2.6 แสดง Bitmap data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การเปลี่ยนข้อมูลบิตแม็พเป็นข้อมูลไบนารี่

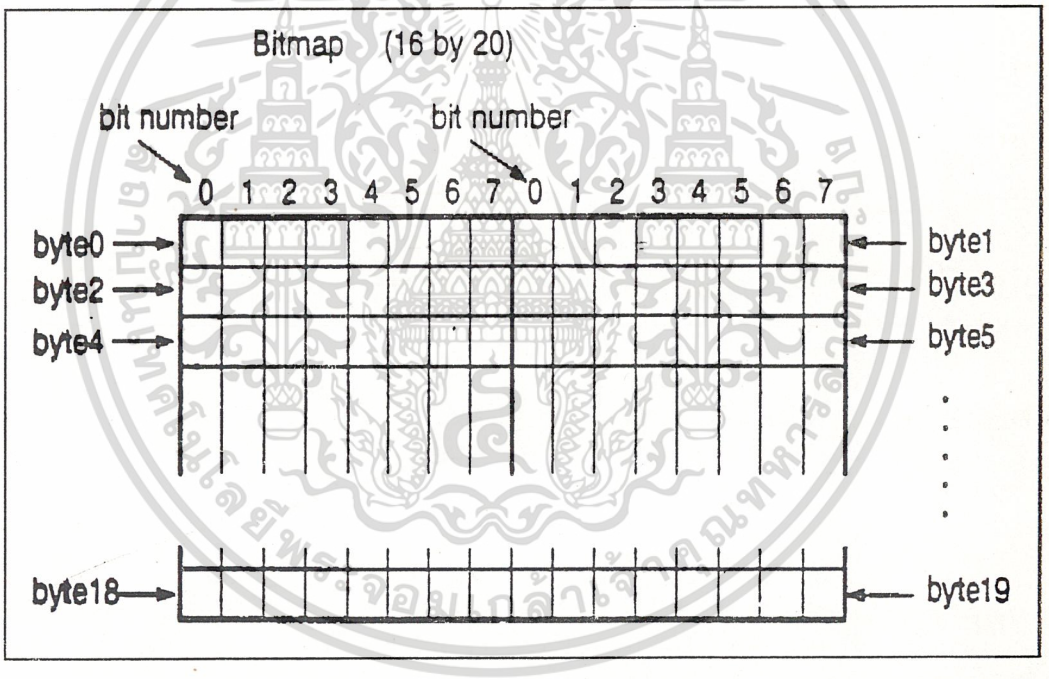
ข้อมูลภาพกราฟิกแบบ Bitmap มีอยู่ 2 ลักษณะ โดยพิจารณาจากความกว้างของภาพ และนำมาใช้เป็นหลักเกณฑ์ในการเปลี่ยนข้อมูลจากข้อมูลที่อยู่ในลักษณะเลขฐาน 16 เป็นเลขฐาน 2 คือ

(1) ถ้าความกว้างเป็นผลคูณของ 8 มีเงื่อนไขของโครงสร้างดังนี้

(ก) ข้อมูลในรูปของตัวเลข 0 หรือ 1 จะถูกจับเป็นกลุ่มโดยเรียงกันไปจำนวน 8 บิท นั่นก็คือ 1 ไบท์

(ข) ลำดับของไบท์ของข้อมูลจะนับจากบนลงล่าง และซ้ายไปขวา

(ค) ลำดับของบิทในแต่ละไบท์จะนับจากซ้ายไปขวา และบิททางซ้ายมือสุด (Leftmost bit) นับเป็นบิทที่ 0 และบิททางขวามือสุด (Rightmost bit) นับเป็นบิทที่ 7



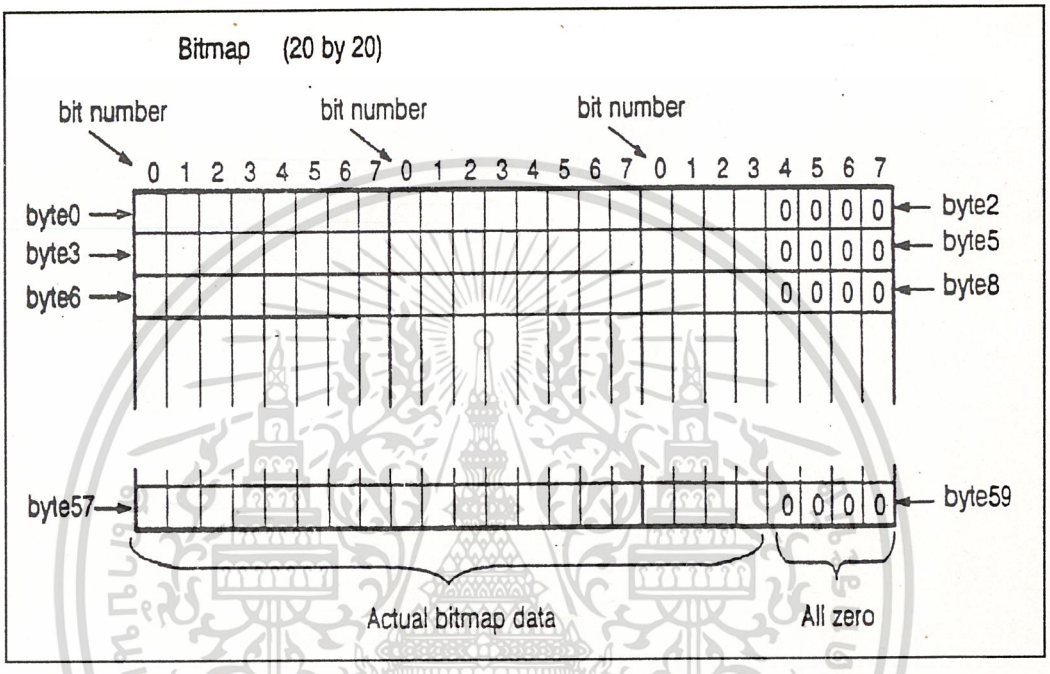
รูปที่ 2.7 แสดงโครงสร้าง Bitmap ความกว้างเป็นผลคูณของ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) ถ้าความกว้างไม่เป็นผลคูณของ 8 จะมีเงื่อนไขดังนี้

(ก) จะนับความกว้างเป็นผลคูณของ 8 ที่น้อยที่สุดที่มีค่ามากกว่าความกว้างที่แท้จริงของรูป

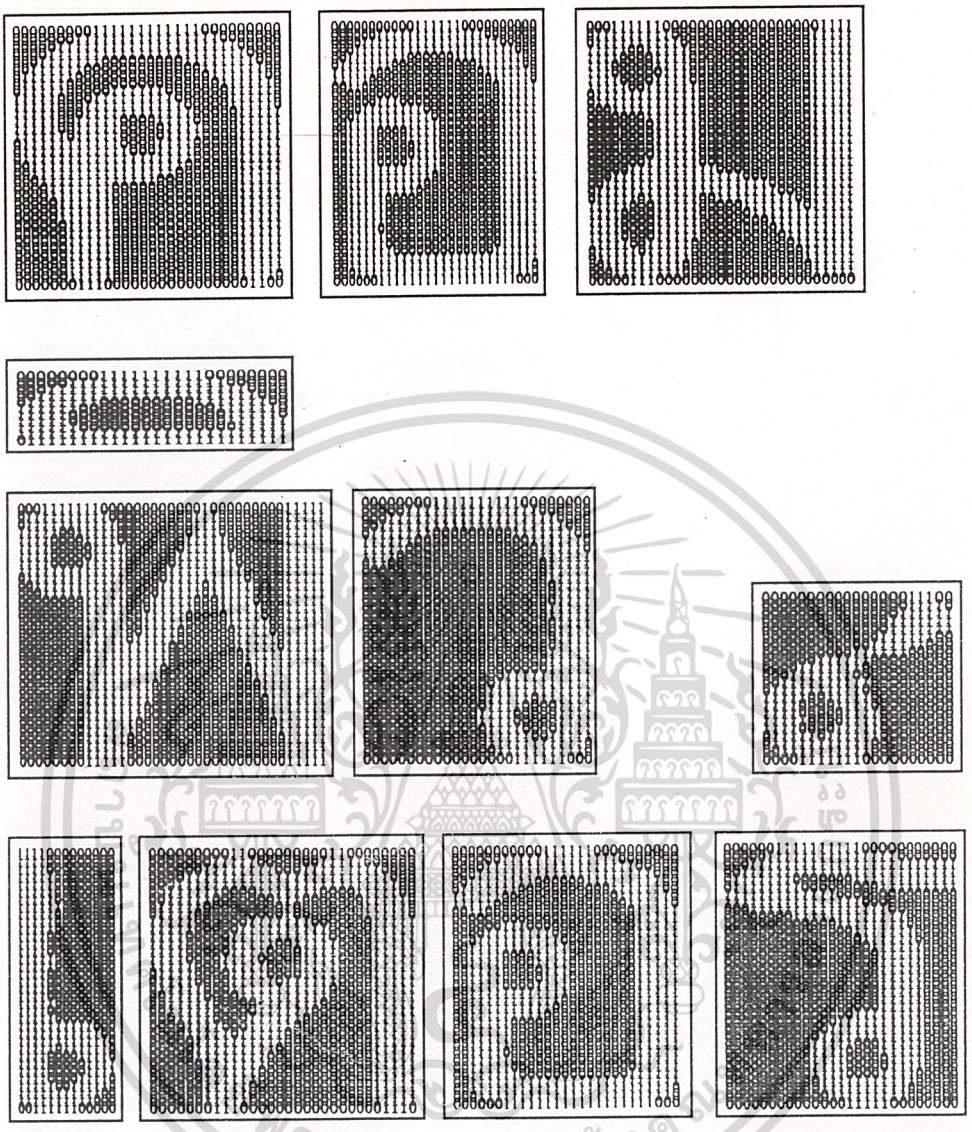
(ข) บิทที่เหลืออยู่นอกเหนือจากความกว้างที่แท้จริง จะถูกเติมด้วยตัวเลข 0



รูปที่ 2.8 แสดงโครงสร้าง Bitmap ความกว้างไม่เป็นผลคูณของ 8

เมื่อทราบรูปแบบของ Bitmap แล้ว ก็ทำการเปลี่ยนข้อมูลภาพที่มีลักษณะเป็นเลขฐาน 16 ให้เป็นเลขฐาน 2 โดยหนึ่งบิตของเลขฐาน 2 จะแทนจุดภาพหนึ่งจุด บิตที่มีค่าเป็น 0 ก็จะแทนจุดสว่างหรือจุดที่เป็นพื้นกระดาษสีขาว และบิตที่มีค่าเป็น 1 ก็จะแทนจุดดำหรือมืดทึบ ซึ่งก็คือจุดที่อาจเป็นตัวอักษรหรือจุดดำของรูปภาพ ก็จะได้ข้อมูลภาพตัวอักษรที่ถูกต้องและพร้อมที่จะส่งไปประมวลผลในขั้นตอนต่อไป ตัวอย่างของภาพตัวอักษรที่ถูกเปลี่ยนจุดภาพให้เป็นค่าไบนารี่แสดงได้ดังรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



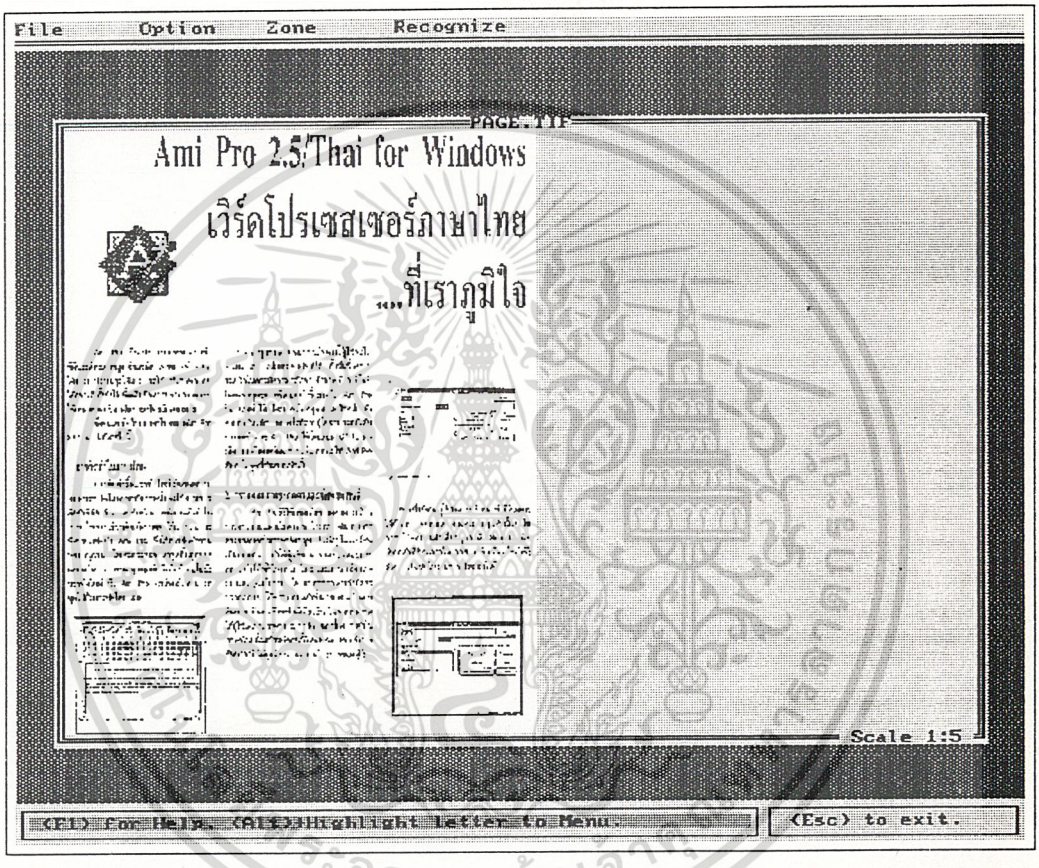
รูปที่ 2.9 แสดงตัวอย่างภาพตัวอักษรในลักษณะไบนารี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.3 ตัวอย่างข้อมูลที่ใช้ในงานวิจัย

ข้อมูลที่ใช้ในงานวิจัยนี้จะมีลักษณะเป็นภาพหน้าเอกสารหนึ่งหน้าขนาดไม่เกินขนาดกระดาษ A4 บนหน้ากระดาษจะประกอบด้วยรูปภาพและข้อความที่มีทั้งตัวอักษรภาษาไทยและตัวอักษรภาษาอังกฤษ ตัวอย่างของภาพหน้าเอกสารในรูปแบบของภาพกราฟิกซึ่งจัดเก็บจากเครื่องตรวจกวาดแสดงด้วยโปรแกรมระบบที่พัฒนาขึ้น ดังรูปที่ 2.10



รูปที่ 2.10 แสดงตัวอย่างของภาพหน้าเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

21210

ตัวอย่างภาพตัวอักษรที่ใช้ในการรู้จำ ประกอบด้วยตัวอักษรภาษาไทยจำนวน 44 ตัว สระและวรรณยุกต์ไทยจำนวน 19 ตัว เลขไทย 10 ตัว ตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก ชุดละ 26 ตัว และตัวเลขอารบิค 10 ตัว รูปแบบ (Font) ต่างๆ ของตัวอักษรทั้งหมดที่ใช้ ได้แสดงไว้ในภาคผนวก ค. ตอนท้ายของวิทยานิพนธ์ฉบับนี้

รูปที่ 2.11 แสดงตัวอย่างของชุดตัวอักษรที่ใช้ในงานวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร

3.1 บทนำ

การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร เป็นขบวนการทำงานหนึ่งในส่วนของการจัดการล่วงหน้า (Preprocessing process) ของระบบการรู้จำตัวอักษร (Character recognition) ส่วนประกอบต่างๆ บนหน้าเอกสารได้กลายเป็นอุปสรรคหนึ่งของระบบการรู้จำตัวอักษร ตัวอย่างเช่นในกรณีของเอกสารหนึ่งหน้ากระดาษที่เป็นข้อความ ถ้าพิจารณาลักษณะโครงร่าง (Layout) ของหน้าเอกสาร จะพบว่าหน้าเอกสารมีลักษณะเป็นเส้นทึบสีดำๆ เรียงกันอยู่ในแนวนอนเป็นเส้นๆ และอยู่รวมกันเป็นกลุ่มๆ ซึ่งที่จริงแล้วเส้นสีทึบดำๆ ก็คือภาพของตัวอักษรที่พิมพ์เรียงกันอยู่เป็นบรรทัด และกลุ่มของเส้นดำหลายๆ เส้นที่เห็นก็คือย่อหน้าของข้อความ เราพบว่าแต่ละบรรทัดของข้อความถูกแบ่งแยกออกจากกันด้วยช่องว่างในแนวนอน ซึ่งก็คือช่องว่างระหว่างบรรทัด และช่องว่างในแนวตั้ง ซึ่งก็คือช่องว่างระหว่างคอลัมน์ เมื่อนำเอาลักษณะโครงร่างของหน้าเอกสารดังกล่าวนี้มาพิจารณาก็สามารถใช้ช่องว่างในแนวนอนและแนวตั้งนี้เป็นตัวแบ่งแยกบริเวณของข้อความหรือย่อหน้าของข้อความที่พิมพ์อยู่บนหน้าเอกสารออกมาได้ และจะได้ส่งต่อภาพของกลุ่มตัวอักษรเหล่านั้นให้กับส่วนของการรู้จำต่อไปได้อย่างถูกต้อง สำหรับกรณีของปัญหาที่ซับซ้อนกว่านี้ บนหน้าเอกสารหนึ่งอาจจะไม่ได้มีเพียงตัวอักษรล้วนๆ พิมพ์อยู่อย่างเดียว แต่อาจจะประกอบด้วย รูปภาพประกอบข้อความ ภาพสัญญลักษณ์ หรือภาพโลโกกราฟิก พิมพ์รวมอยู่ด้วย ซึ่งกรณีนี้ถ้าไม่มีการแยกแยะส่วนประกอบต่างๆ บนหน้าเอกสารให้ถูกต้องแล้ว อาจทำให้การรู้จำตัวอักษรผิดพลาด หรือทำไม่ได้เลยในกรณีที่ข้อมูลที่ส่งผ่านให้กับส่วนของการรู้จำตัวอักษรกลายเป็นรูปภาพไม่ใช่ภาพตัวอักษร เพราะฉะนั้นขบวนการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารจึงเป็นขั้นตอนการทำงานที่สำคัญมากและจะขาดเสียมิได้ เพื่อการจัดเตรียมข้อมูลให้กับระบบการรู้จำตัวอักษรจะเป็นไปได้อย่างถูกต้อง

3.2 ทฤษฎีที่นำมาใช้

3.2.1 ทฤษฎีเซลลูล่าออโตมาตา (Cellular Automata)[17]

เทคนิคที่นำมาประยุกต์ใช้วิเคราะห์และระบุส่วนประกอบของหน้าเอกสารเป็นเทคนิคใหม่โดยอาศัยทฤษฎีเซลลูล่าออโตมาตา (Cellular Automata) ซึ่งเป็นทฤษฎีว่าด้วยการแก้ปัญหาทฤษฎีหนึ่ง อาศัยหลักการที่กล่าวว่า การแก้ปัญหาที่ไม่ว่าจะเป็นเรื่องใดก็ตาม จะยึดหลักการแบ่งปัญหาออกเป็นส่วนย่อยๆ แล้วค่อยๆ จัดการกับส่วนย่อยๆ เหล่านั้นก่อน เมื่อจัดการกับองค์ประกอบย่อยๆ เหล่านั้นได้ ก็จะทำให้สามารถจัดการกับปัญหาใหญ่ทั้งหมดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนย่อยๆ ที่ถูกแบ่งแยกออกจะเรียกอีกอย่างหนึ่งว่า เซล (Cell) การจัดการกับแต่ละเซลแล้วจะสามารถจัดการกับระบบรวมหรือปัญหาใหญ่ได้ทั้งหมดนั้น มาจากความจริงที่ว่า ปัญหาใหญ่ที่สามารถแบ่งแยกย่อยออกไปได้นั้น จะเกี่ยวข้องกับปัญหาย่อยหรือเซลที่ถูกแยกออกมา และเมื่อไล่แก้ปัญหาย่อย ก็เหมือนกับแก้ปัญหาใหญ่ทีละเล็กละน้อย จนในที่สุดเมื่อแก้ปัญหาย่อยได้หมด ก็เป็นการแก้ปัญหาใหญ่ได้ทั้งหมดด้วย

การแก้ปัญหาส่วนย่อยหรือเซล นอกจากจะใช้สถานะหรือรายละเอียดที่ได้จากภายในเซลเองแล้ว ก็ยังสามารถใช้สถานะของเซลอื่นๆ ที่อยู่รอบๆ เซลนั้นมาใช้ช่วยในการแก้ปัญหาได้ด้วย ซึ่งสุดท้ายจะนำไปสู่การแก้ปัญหาใหญ่ทั้งหมด

วิธีการแก้ปัญหาของทฤษฎีเซลลูล่าออโตมาตา ยึดหลักใหญ่ๆ อยู่ 2 ข้อคือ

1. แยกสิ่งที่สนใจและต้องการแก้ไขหรือหาคำตอบ ออกเป็นส่วนย่อยๆ หรือเซล

2. แก้ไขปัญหาส่วนย่อยทีละส่วนให้หมด โดยอาศัยข้อมูลจากภายในส่วนย่อย และข้อมูลจากส่วนย่อยที่อยู่รอบๆ นั้นด้วย

3.2.2 เทคนิคการติดตามรอยขอบของภาพ (Contour Following)[19]

เทคนิคการติดตามรอยขอบของภาพ ถูกนำมาใช้ในการแยกและคัดลอกส่วนของรูปภาพใดๆ ที่อยู่บนรูปใหญ่ ข้อมูลภาพที่จะนำมาประมวลผลด้วยเทคนิคนี้จะต้องอยู่ในรูปของข้อมูลไบนารี่ นั่นคือจุดภาพจะแสดงด้วยตัวเลข 0 กับ 1 เท่านั้น โดยที่จุดภาพที่มีค่าเป็น 1 แทนจุดดำหรือจุดที่เป็นส่วนของรูปภาพ และจุดภาพที่มีค่าเป็น 0 แทนจุดขาวหรือจุดที่เป็นช่องว่างบนกระดาษหรือพื้นเบื้องหลัง

การทำงานของเทคนิคการติดตามรอยขอบของภาพ เป็นการเดินไต่ไปตามขอบระหว่างส่วนที่เป็นรูปภาพ (Image) กับส่วนที่เป็นพื้นเบื้องหลัง (Background) โดยจะตรวจกวาดไปทุกๆ จุดภาพ (Pixel) โดยจะเริ่มจากจุดมุมซ้ายบนของข้อมูลภาพ ตรวจกวาดไปในทิศทางจากซ้ายไปขวา และเลื่อนลงจากบนลงล่าง เมื่อตรวจกวาดมาพบจุดภาพใดๆ ที่มีค่าของจุดภาพเป็น 1 ก็จะเปลี่ยนลักษณะการเคลื่อนที่ไปยังจุดภาพจุดถัดไปเสียใหม่ โดยมีเงื่อนไขของการเคลื่อนที่ดังต่อไปนี้

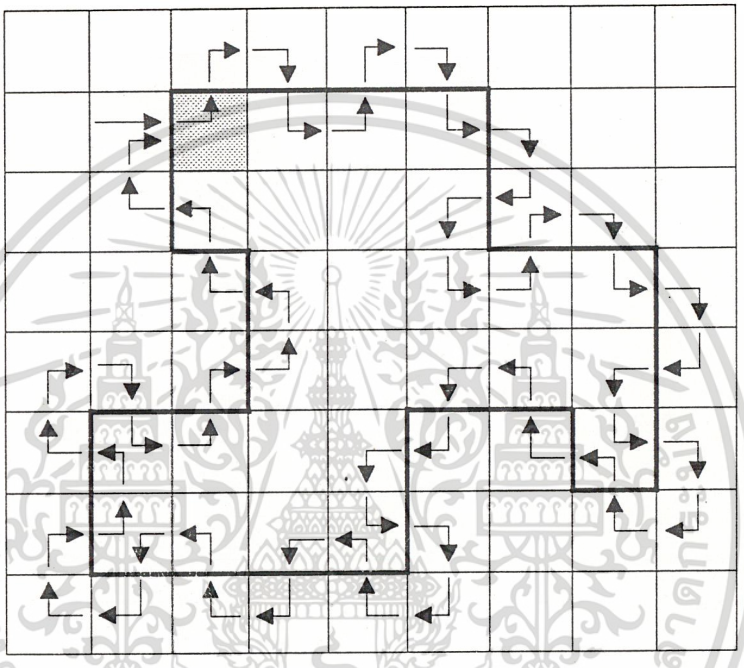
1. ถ้าจุดที่อยู่ปัจจุบันเป็นจุดของภาพหรือมีค่าของจุดเป็น 1 ให้เลี้ยวซ้าย แล้วก้าวเดินตรงไปข้างหน้าไปยังจุดถัดไป

2. ถ้าจุดที่อยู่ปัจจุบันเป็นพื้นเบื้องหลังหรือมีค่าของจุดเป็น 0 ก็ให้เลี้ยวขวา แล้วก้าวเดินตรงไปข้างหน้าไปยังจุดถัดไป

3. การเคลื่อนที่จะสิ้นสุดลง เมื่อจุดที่อยู่ปัจจุบันเป็นจุดเดียวกันกับจุดเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 3.1 ได้แสดงลักษณะการทำงานของเทคนิคการติดตามรอยขอบภาพ ซึ่งจะแสดงการเคลื่อนที่ไปตามจุดต่างๆ ที่เป็นขอบของภาพ เริ่มจากจุดที่ถูกแรเงาไว้ซึ่งเป็นจุดของภาพจุดแรกที่ตรวจกวาดมาพบ การเคลื่อนที่จะเป็นไปตามเงื่อนไขที่กำหนด เมื่อการเคลื่อนที่วนกลับมาถึงจุดที่เป็นจุดเริ่มต้น ก็จะทราบจุดที่เป็นขอบของภาพได้ทั้งหมด



รูปที่ 3.1 แสดงลักษณะการติดตามรอยขอบของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร[16]

ในการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารได้นำทฤษฎีเซลลูล่าออโตมาตาและเทคนิคการติดตามขอบของภาพมาประยุกต์ใช้ รายละเอียดของการทำงานในขั้นตอนนี้ สามารถแบ่งการทำงานออกได้ 2 ขั้นตอนใหญ่ๆ และอธิบายได้ดังต่อไปนี้

3.3.1 การกำหนดขอบเขต (Blocking zones)

การทำงานในขั้นตอนนี้เริ่มต้นด้วยการเตรียมข้อมูลภาพหน้าเอกสารหนึ่งหน้ากระดาษในลักษณะของข้อมูลไบนารี่ นั่นคือแต่ละจุดของภาพจะถูกแทนด้วยตัวเลขฐานสองเท่านั้น คือตัวเลข 0 กับ 1 โดยที่จุดภาพที่มีค่าเป็น 0 จะแทนพื้นสีขาวหรือช่องว่างบนหน้ากระดาษ และจุดภาพที่มีค่าเป็น 1 จะแทนส่วนที่เป็นลายเส้นของตัวอักษรหรือลายเส้นของรูปภาพ โดยจะเตรียมข้อมูลลักษณะนี้ไว้ 2 ชุดเพื่อใช้ประกอบการวิเคราะห์

สำหรับข้อมูลชุดที่หนึ่ง จุดภาพแต่ละจุดจะถูกกำหนดสถานะของจุด โดยพิจารณาจากจุดภาพที่ล้อมรอบจุดที่กำลังพิจารณา นับจำนวนจุดภาพรอบๆ ที่มีค่าเป็น 1 และพิจารณาลักษณะของจุดภาพรอบๆ ที่ปรากฏ ซึ่งจะถูกนำมาใช้เป็นเงื่อนไขในการกำหนดสถานะของจุด โดยจะมีเงื่อนไขอยู่ด้วยกัน 4 เงื่อนไขดังต่อไปนี้

1. ถ้าจุดภาพที่กำลังพิจารณาอยู่มีค่าเป็น 1 และจำนวนจุดภาพรอบข้างมีจำนวนน้อยกว่าจำนวนที่กำหนดไว้ แล้วจุดภาพที่กำลังพิจารณาจะถูกกำหนดให้เป็น **จุดตาย** เนื่องจากมีจำนวนจุดภาพรอบข้างน้อยเกินไป

2. ถ้าจุดภาพที่กำลังพิจารณาอยู่มีค่าเป็น 1 และจำนวนจุดภาพรอบข้างมีจำนวนมากกว่าหรือเท่ากับจำนวนที่กำหนดไว้ แล้วจุดภาพที่กำลังพิจารณาจะถูกกำหนดให้เป็น **จุดเป็น**

3. ถ้าจุดภาพที่กำลังพิจารณาอยู่มีค่าเป็น 0 และจำนวนจุดภาพรอบข้างมีจำนวนมากกว่าหรือเท่ากับจำนวนที่กำหนดไว้ แล้วจุดภาพที่กำลังพิจารณาจะถูกกำหนดให้เป็น **จุดเกิด**

4. ถ้าจุดภาพที่กำลังพิจารณาอยู่มีค่าเป็น 0 และจำนวนจุดภาพรอบข้างมีจำนวนน้อยกว่าจำนวนที่กำหนดไว้ แล้วจุดภาพที่กำลังพิจารณาอยู่จะยังคงสถานะเดิมต่อไปและค่าของจุดภาพยังคงมีค่าเหมือนเดิมไม่เปลี่ยนแปลง

เมื่อทราบสถานะของจุดภาพจุดหนึ่งแล้ว จะทำการเปลี่ยนค่าของจุดภาพให้มีค่าที่สอดคล้องกับสถานะของจุดภาพนั้นๆ แต่การเปลี่ยนค่าของจุดภาพจะกระทำกับจุดภาพในข้อมูลภาพชุดที่สอง ณ.ตำแหน่งพิกัดของจุดภาพเดียวกันกับพิกัดของจุดภาพในข้อมูลภาพชุดที่หนึ่ง ทั้งนี้ก็เพราะว่าจุดภาพที่ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนค่าตามสถานะใหม่แล้วจะได้ไม่ส่งผลกระทบต่อการพิจารณากำหนดสถานะของจุดภาพต่อไปในข้อมูลภาพต้นฉบับ สำหรับเงื่อนไขในการเปลี่ยนค่าของจุดภาพใหม่ถูกกำหนดโดยเงื่อนไข 3 ลักษณะดังนี้

1. **จุดตาย** ซึ่งเดิมมีค่าเป็น 1 จะถูกเปลี่ยนให้มีค่าเป็น 0
2. **จุดเป็น** ซึ่งเดิมมีค่าเป็น 1 ยังคงให้มีค่าเป็น 1 ต่อไป
3. **จุดเกิด** ซึ่งเดิมมีค่าเป็น 0 จะถูกเปลี่ยนให้มีค่าเป็น 1

การพิจารณากำหนดสถานะของจุดภาพและเปลี่ยนค่าใหม่ให้กับจุดภาพจะกระทำกับทุกๆ จุดภาพในข้อมูลภาพ เริ่มจากจุดภาพที่มุมบนซ้ายมือ และกวาดตรวจไปบนคอลัมน์ในทิศทางจากซ้ายไปขวา และเลื่อนแถวลงมาในทิศทางจากบนลงล่าง

การประยุกต์ใช้งานทฤษฎีเซลลูล่าออโตมาตากับการกำหนดขอบเขตของส่วนประกอบของหน้าเอกสารในงานวิจัยเรื่องการรู้จำตัวอักษรของวิทยานิพนธ์ฉบับนี้ มีการปรับแต่งเงื่อนไขของการพิจารณาสถานะของจุดภาพบางประการให้เหมาะสมกับวัตถุประสงค์ของงานมากขึ้น เนื่องจากจุดมุ่งหมายของการกำหนดขอบเขตของส่วนประกอบของหน้าเอกสารคือ ขอบเขตในลักษณะสี่เหลี่ยมที่เป็นย่อหน้าของข้อความหรือขอบเขตสี่เหลี่ยมที่เป็นรูปภาพ ดังนั้นการพิจารณาจุดภาพที่ล้อมรอบจุดภาพที่กำลังพิจารณา จะละเว้นไม่พิจารณาจุดภาพในแนวเส้นทะแยงมุมทั้งสอง คงเหลือพิจารณาเฉพาะจุดรอบข้างที่อยู่ด้านบน ด้านขวา ด้านล่าง และด้านซ้าย รวม 4 จุดเท่านั้น ซึ่งจะช่วยในการค้นหาขอบเขตในลักษณะสี่เหลี่ยมทำได้ง่ายขึ้น อีกทั้งยังช่วยลดการคำนวณลงได้อีกมาก เป็นผลให้การทำงานเป็นไปได้รวดเร็วยิ่งขึ้นด้วย

เงื่อนไขการพิจารณาสถานะของจุดของภาพที่ถูกประยุกต์ตามเหตุผลที่ได้กล่าวมาแล้วข้างต้น มีละเอียดของเงื่อนไขดังนี้

1. ถ้ากำหนดให้จุด X เป็นจุดใดๆ ที่มีค่าเป็น 1 แล้ว จุด X จะมีสถานะเป็น **จุดตาย** ก็ต่อเมื่อมีจุดรอบข้างที่มีค่าเป็น 1 น้อยกว่า 2 จุด

	1	
0	X	0
	0	

	0	
1	X	0
	0	

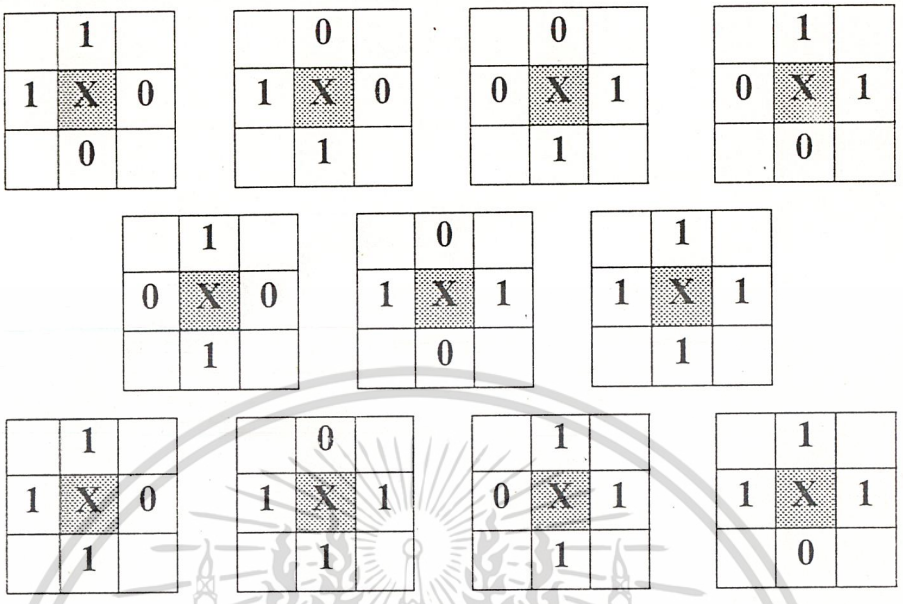
	0	
0	X	0
	1	

	0	
0	X	1
	0	

รูปที่ 3.2 แสดงลักษณะของจุดตาย

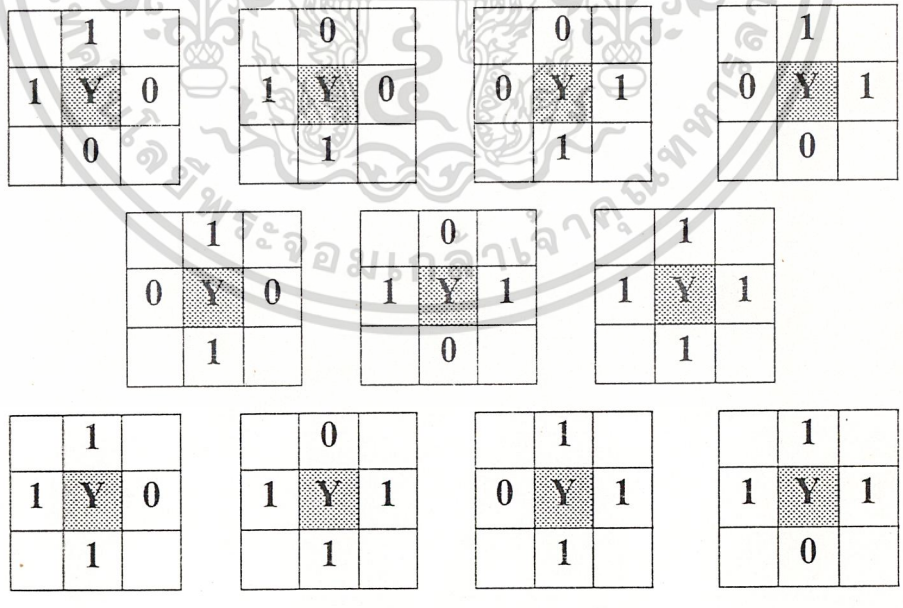
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ถ้ากำหนดให้จุด X เป็นจุดใดๆ มีค่าเป็น 1 แล้ว จุด X จะมีสถานะเป็น **จุดเป็น** ก็ต่อเมื่อมีจุดรอบข้างที่มีค่าเป็น 1 มากกว่าหรือเท่ากับ 2 จุด



รูปที่ 3.3 แสดงลักษณะของจุดเป็น

3. ถ้ากำหนดให้จุด Y เป็นจุดใดๆ ที่มีค่าเป็น 0 แล้ว จุด Y จะมีสถานะเป็น **จุดเกิด** ก็ต่อเมื่อมีจุดรอบข้างที่มีค่าเป็น 1 มากกว่าหรือเท่ากับ 2 จุด

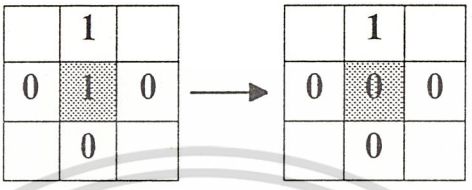


รูปที่ 3.4 แสดงลักษณะของจุดเกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

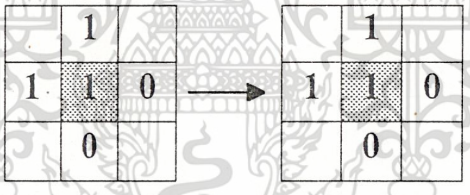
สำหรับเงื่อนไขของการเปลี่ยนค่าใหม่ของจุดภาพตามสถานะของจุดภาพที่หาได้นั้น ยังคงใช้เงื่อนไขเดิม ซึ่งมีรายละเอียดดังต่อไปนี้

1. **จุดตาย** จุดภาพที่มีสถานะเป็นจุดตาย ซึ่งค่าของจุดภาพนั้นมีค่าเป็น 1 จะถูกเปลี่ยนให้มีค่าเป็น 0 ดังตัวอย่างในรูปที่ 3.5



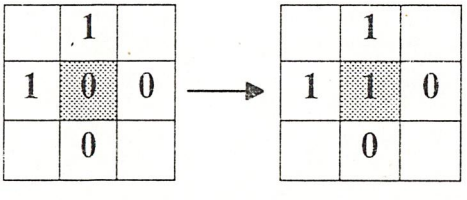
รูปที่ 3.5 แสดงตัวอย่างของจุดตายที่มีค่าเป็น 1 ถูกเปลี่ยนให้มีค่าเป็น 0

2. **จุดเป็น** จุดภาพที่มีสถานะเป็นจุดเป็น ซึ่งค่าของจุดภาพนั้นมีค่าเป็น 1 จะยังคงมีค่าเป็น 1 ต่อไปไม่เปลี่ยนแปลง ดังตัวอย่างในรูปที่ 3.6



รูปที่ 3.6 แสดงตัวอย่างของจุดเป็นที่มีค่าเป็น 1 ยังคงมีค่าเป็น 1 ต่อไป

3. **จุดเกิด** จุดภาพที่มีสถานะเป็นจุดเกิด ซึ่งค่าของจุดภาพนั้นมีค่าเป็น 0 จะถูกเปลี่ยนให้มีค่าเป็น 1 ดังตัวอย่างในรูปที่ 3.7



รูปที่ 3.7 แสดงตัวอย่างของจุดเกิดที่เดิมมีค่าเป็น 0 ถูกเปลี่ยนให้มีค่าเป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

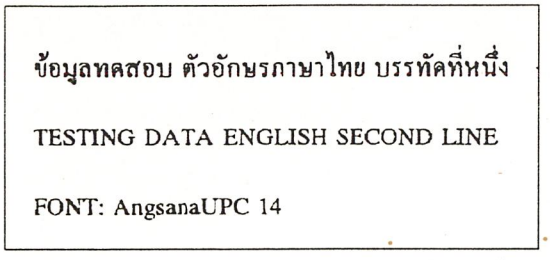
การประยุกต์ใช้ทฤษฎีเซลลูล่าออโตมาตา ในการกำหนดขอบเขตของส่วนประกอบของหน้าเอกสาร จะสามารถกำหนดได้เพียงโครงร่าง (Layout) ของขอบเขตของส่วนประกอบของหน้าเอกสารเท่านั้น ขอบเขตต่างๆ ของส่วนประกอบบนหน้าเอกสารยังคงอยู่กระจัดกระจายบนหน้ากระดาษซึ่งยังไม่เหมาะที่จะแยกคัดลอกออกมา จึงจำเป็นต้องมีขั้นตอนการทำงานอีกขั้นตอนหนึ่งที่จะจับกลุ่มของขอบเขตต่างๆ ที่กระจัดกระจายกันอยู่ รวมเข้าด้วยกันให้ถูกต้องและเหมาะสม ตัวแปรตัวหนึ่งที่ถูกนำมาใช้เป็นตัวกำหนดการจับรวมกลุ่มของขอบเขตของส่วนประกอบของหน้าเอกสารที่กระจัดกระจายกันอยู่ คือ ค่าความหยาบของจุดภาพ

ค่าความหยาบของจุดภาพ (Coarseness value) เป็นค่าคงที่ใดๆ ที่ใช้กำหนดระยะห่างของจุดภาพที่เป็นจุดภาพในขอบเขตต่างๆ ที่กระจัดกระจายอยู่ ตัวอย่างเช่น ถ้าจุดภาพของ 2 ขอบเขตใดๆ อยู่ในช่วงระยะห่างของค่าความหยาบของจุดภาพที่กำหนด ก็จะทำการรวม 2 ขอบเขตนั้นเข้าเป็นขอบเขตหรือพื้นที่เดียวกัน เป็นต้น ในการกำหนดค่าความหยาบของจุดภาพ ถ้ากำหนดให้ค่าความหยาบของจุดภาพมีค่าน้อยเกินไป จะทำให้ไม่สามารถจับรวมกลุ่มของขอบเขตที่กระจัดกระจายกันอยู่ได้ และถ้ากำหนดให้ค่าความหยาบของจุดภาพมีค่ามากเกินไป ก็จะทำให้ขอบเขตที่กระจัดกระจายอยู่รวมกลุ่มกันเป็นขอบเขตใหญ่เพียงขอบเขตเดียว ซึ่งจะไม่สามารถแยกส่วนประกอบของหน้าเอกสารได้ ดังนั้นการกำหนดขอบเขตของส่วนประกอบของหน้าเอกสารที่ถูกต้องและเหมาะสม จึงจำเป็นจะต้องกำหนดค่าความหยาบของจุดภาพที่ใช้ให้เหมาะสมด้วยเช่นกัน

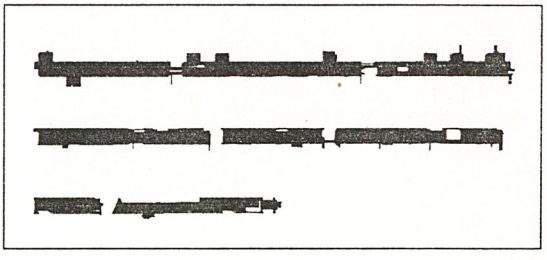
ค่าความหยาบของจุดภาพที่เหมาะสม ที่จะสามารถกำหนดขอบเขตของส่วนประกอบของหน้าเอกสารได้อย่างถูกต้อง จะไปสอดคล้องกับตัวแปรอีกตัวหนึ่งนั่นก็คือ ค่าความละเอียดของจุดภาพของข้อมูลภาพ (Image resolution) ซึ่งค่าความละเอียดของจุดภาพเป็นค่าคงที่ ที่ถูกกำหนดไว้ตั้งแต่ขั้นตอนกวาดตรวจภาพหน้าเอกสารด้วยเครื่องตรวจกวาดภาพ ค่าความหยาบของจุดภาพที่สอดคล้องกับค่าความละเอียดของจุดภาพ สามารถกำหนดค่าขึ้นได้จากการทดลอง ซึ่งในการทดลองได้กำหนดค่าความหยาบของจุดภาพที่ใช้ทดสอบไว้ทั้งหมด 5 ระดับคือ 2, 5, 10, 15 และ 20 จุด (Pixel) ทำการทดสอบกับข้อมูลภาพที่ตรวจกวาดมาด้วยค่าความละเอียดของจุดภาพในระดับต่างๆ กัน 4 ระดับคือ 150, 200, 300 และ 400 จุดต่อนิ้ว (Dot per inch, dpi) ผลของการทดลองได้ค่าของตัวแปรทั้งสองที่สามารถเลือกใช้ให้เหมาะสมกับงานได้

ตัวอย่างภาพของข้อมูลที่ใช้ทดลอง และลักษณะของขอบเขตของข้อมูลที่เกิดขึ้นเมื่อกำหนดค่าความหยาบของจุดภาพ และค่าความละเอียดของจุดภาพ ให้มีค่าต่างๆ กันแสดงได้ในรูปที่ 3.8, 3.9, 3.10 และ 3.11

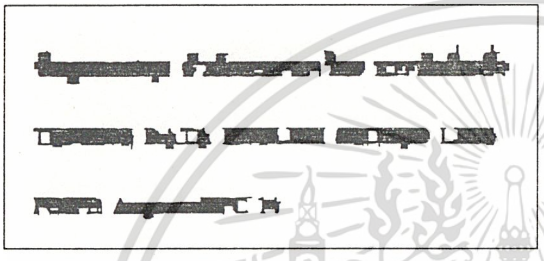
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



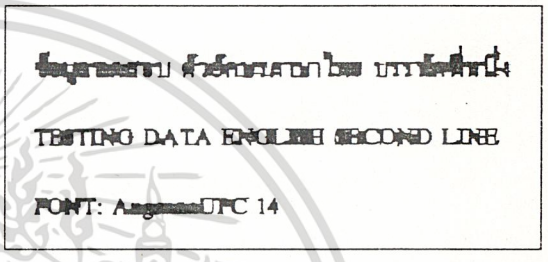
(ก)



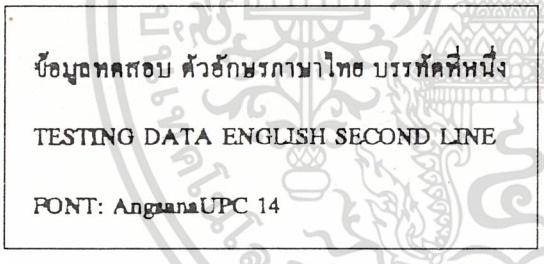
(ข)



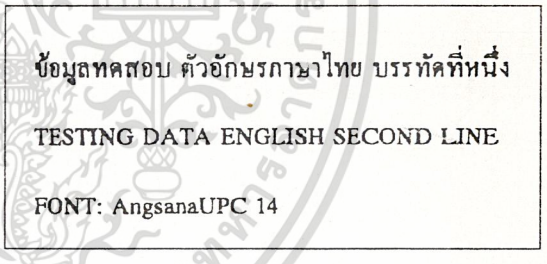
(ค)



(ง)



(จ)



(ฉ)

รูปที่ 3.8 แสดงตัวอย่างข้อมูลภาพในขั้นตอนการกำหนดขอบเขต

(ก) ภาพปกติ มีความละเอียดของจุดภาพเท่ากับ 400 จุดต่อนิ้ว

(ข) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 20

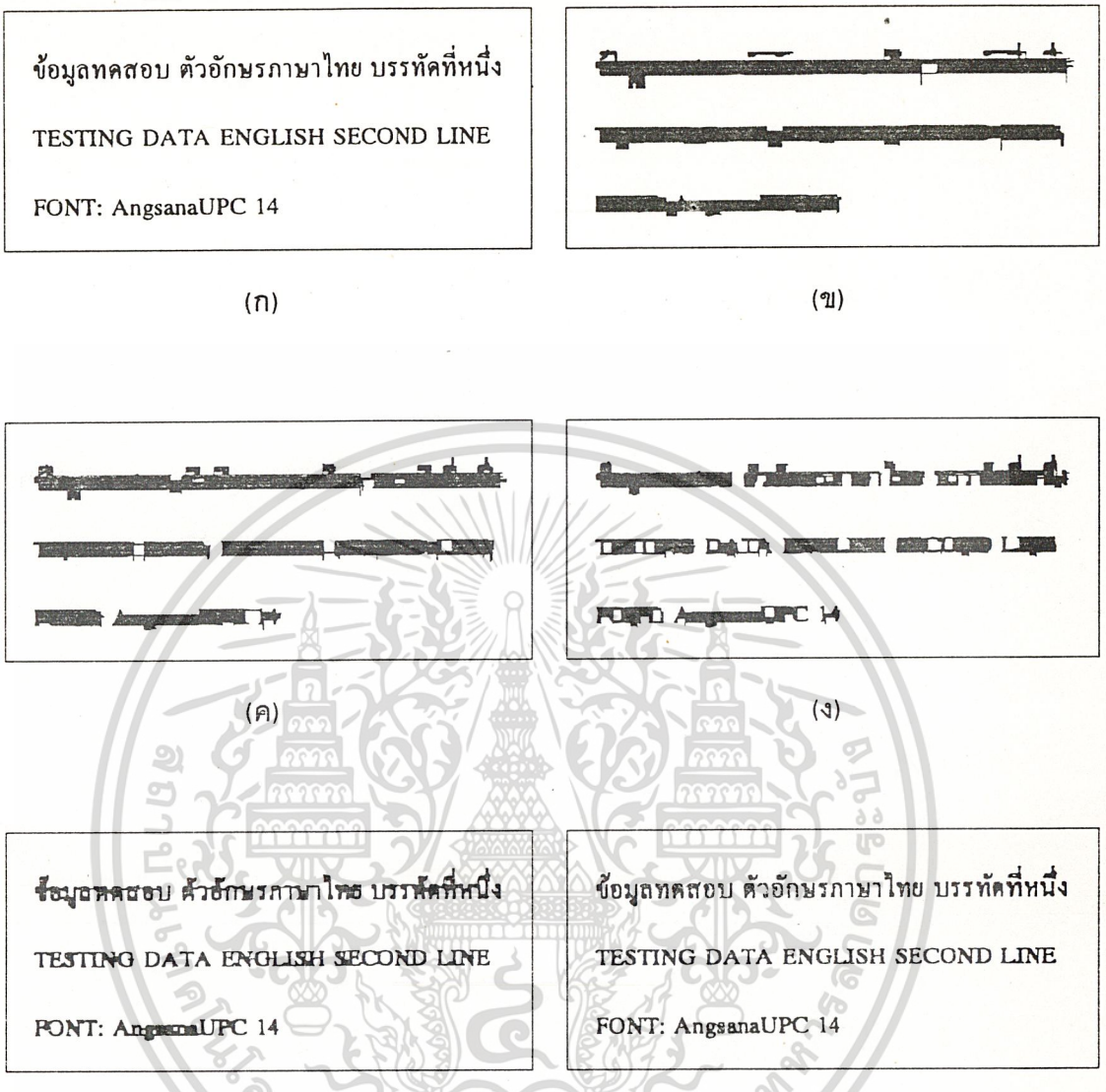
(ค) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 15

(ง) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 10

(จ) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 5

(ฉ) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงตัวอย่างข้อมูลภาพในขั้นตอนการกำหนดขอบเขต

(ก) ภาพปกติ มีความละเอียดของจุดภาพเท่ากับ 300 จุดต่อนิ้ว

(ข) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 20

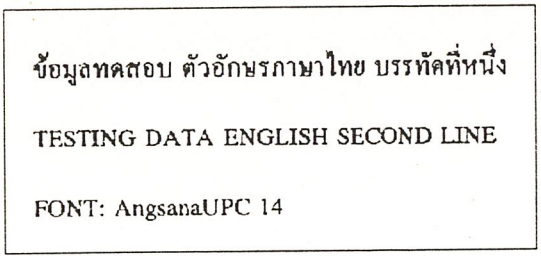
(ค) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 15

(ง) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 10

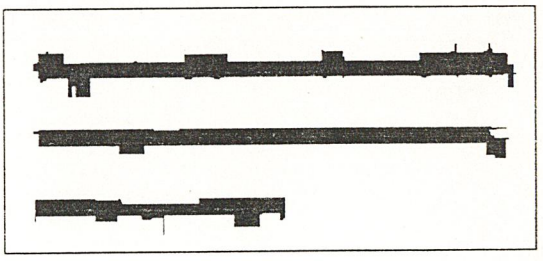
(จ) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 5

(ฉ) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 2

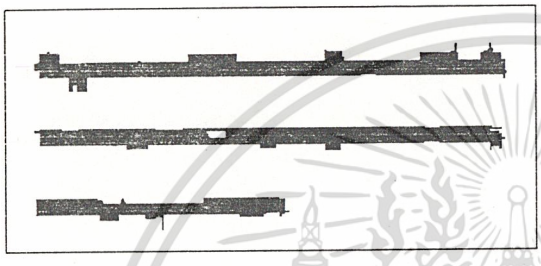
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



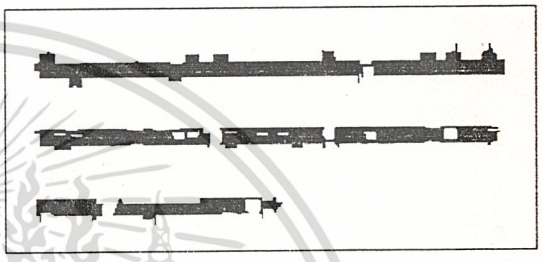
(ก)



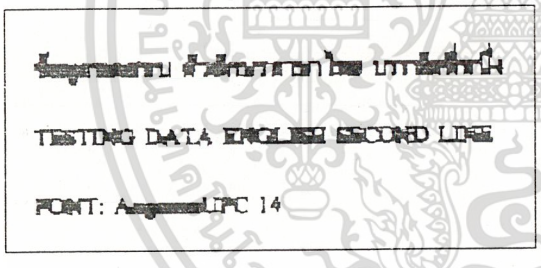
(ข)



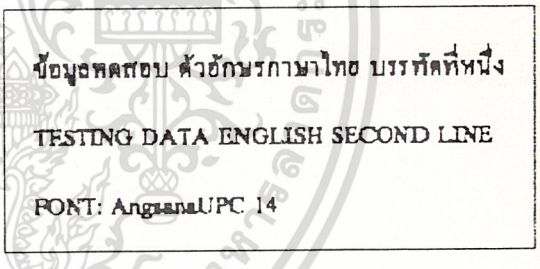
(ค)



(ง)



(จ)



(ฉ)

รูปที่ 3.10 แสดงตัวอย่างข้อมูลภาพในขั้นตอนการกำหนดขอบเขต

(ก) ภาพปกติ มีความละเอียดของจุดภาพเท่ากับ 200 จุดต่อนิ้ว

(ข) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 20

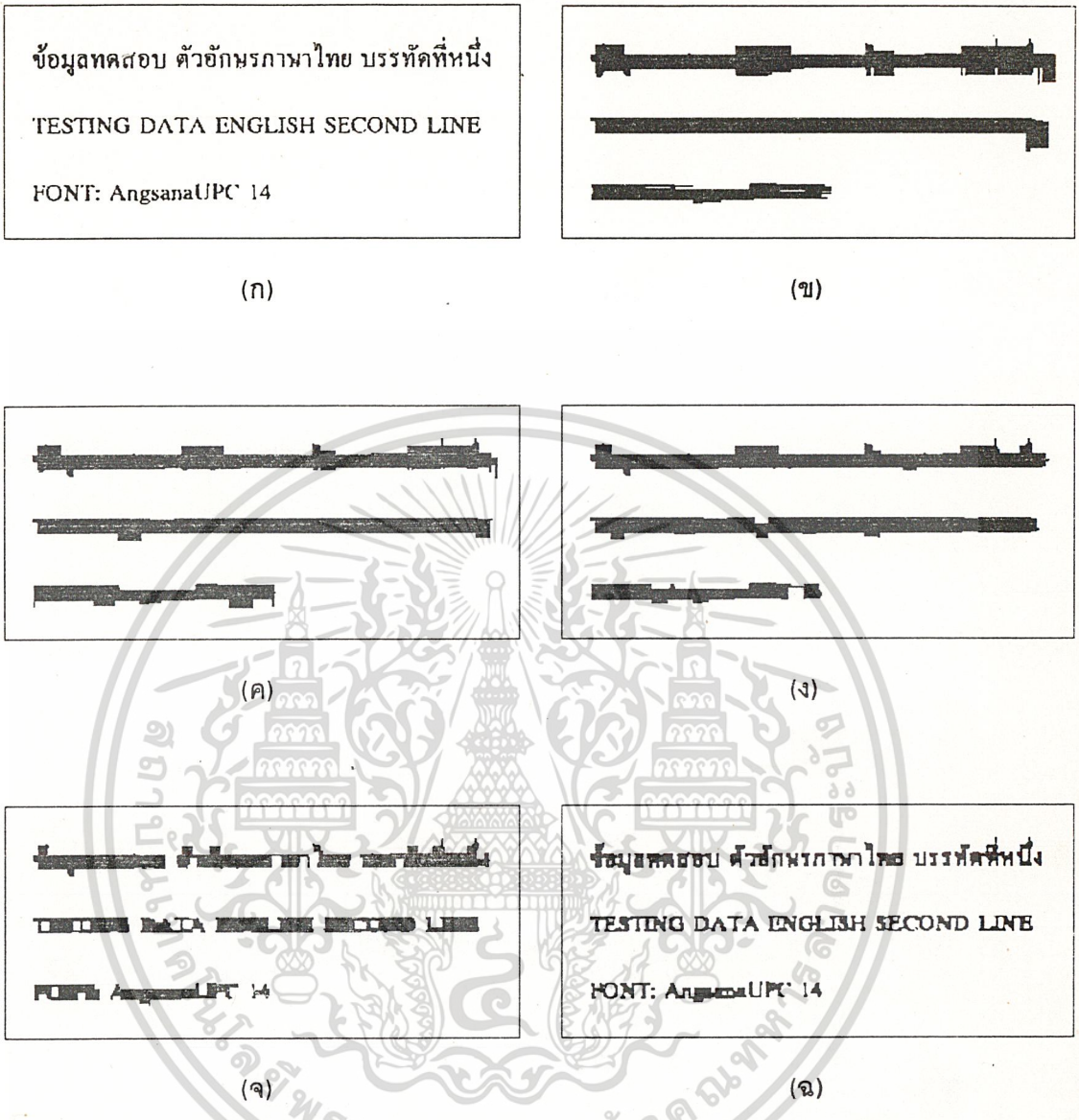
(ค) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 15

(ง) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 10

(จ) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 5

(ฉ) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงตัวอย่างข้อมูลภาพในขั้นตอนการกำหนดขอบเขต

(ก) ภาพปกติ มีความละเอียดของจุดภาพเท่ากับ 150 จุดต่อนิ้ว

(ข) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 20

(ค) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 15

(ง) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 10

(จ) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 5

(ฉ) ภาพแสดงขอบเขต เมื่อกำหนดค่า Coarseness เท่ากับ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตารางที่ 3.1 ได้สรุปค่าของตัวแปรทั้งสองคือ ค่าความละเอียดของข้อมูลภาพ และค่าความหยาบของจุดภาพ ที่เหมาะสมและสามารถนำไปใช้งานทั่วๆ ไปได้ แต่อย่างไรก็ตามค่าของตัวแปรทั้งสองที่เหมาะสมจริงๆ จะขึ้นอยู่ว่าต้องการนำผลลัพธ์ที่ได้จากการกำหนดขอบเขตของส่วนประกอบของหน้าเอกสารไปใช้งานต่อไปอย่างไร ยกตัวอย่างเช่น ถ้าต้องการค้นหาโครงร่างอย่างหยาบของหน้าเอกสารว่าประกอบด้วยอะไรบ้าง ก็จะต้องกำหนดค่าความหยาบของจุดภาพมีค่ามาก แต่ถ้าต้องการค้นหาโครงร่างให้ละเอียดมากขึ้น ก็จะต้องกำหนดค่าความหยาบของจุดภาพให้มีค่าลดน้อยลงเป็นต้น (พิจารณารูป 3.8 ถึง 3.11 ประกอบ)

ตารางที่ 3.1 แสดงค่าของตัวแปรที่ใช้ในการกำหนดขอบเขต

ความละเอียดของภาพ (หน่วย : จุดต่อนิ้ว) Image resolution (Unit : dot per inch, dpi)	ความหยาบของจุดภาพ (หน่วย : จุด) Coarseness value (Unit : pixel)
400	15-20
300	10
200	10
150	5-10

3.3.2 การแยกและคัดลอกขอบเขต (Extracting rectangular zones)

การคัดลอกขอบเขตที่เป็นส่วนประกอบของหน้าเอกสารออกจากภาพหน้าเอกสาร ได้นำเทคนิคการติดตามรอยขอบของภาพมาใช้ในการแยกและคัดลอกขอบเขตออก การทำงานเริ่มตรวจกวาดจุดภาพจากจุดมุมบนด้านซ้ายไปในทิศทางจากซ้ายไปขวา และเลื่อนลงจากบนลงล่าง เมื่อตรวจกวาดมาพบจุดภาพที่เป็นจุดใดจุดหนึ่งของขอบเขต ซึ่งมีค่าของจุดภาพเป็น 1 ก็จะเคลื่อนที่ติดตามรอยขอบของขอบเขตไป โดยอาศัยเทคนิคการติดตามรอยขอบของภาพ ในขณะที่เคลื่อนที่ติดตามรอยขอบของภาพไป ตำแหน่งหรือพิกัดของจุด (coordinate) ที่เป็นขอบของภาพจะถูกบันทึกเก็บไว้ด้วย และเมื่อเคลื่อนที่ไปจนรอบนั่นคือกลับมาอยู่ที่จุดเริ่มต้นแล้ว ก็จะนำพิกัดของจุดที่เป็นขอบภาพทั้งหมดมาคำนวณหาจุดพิกัดที่มีค่ามากที่สุดและน้อยที่สุดทั้งในแนวนอนและแนวตั้ง มากำหนดเป็นจุดมุมทั้งสี่ของรูปสี่เหลี่ยม แต่ละจุดมุมของรูปสี่เหลี่ยมอธิบายได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

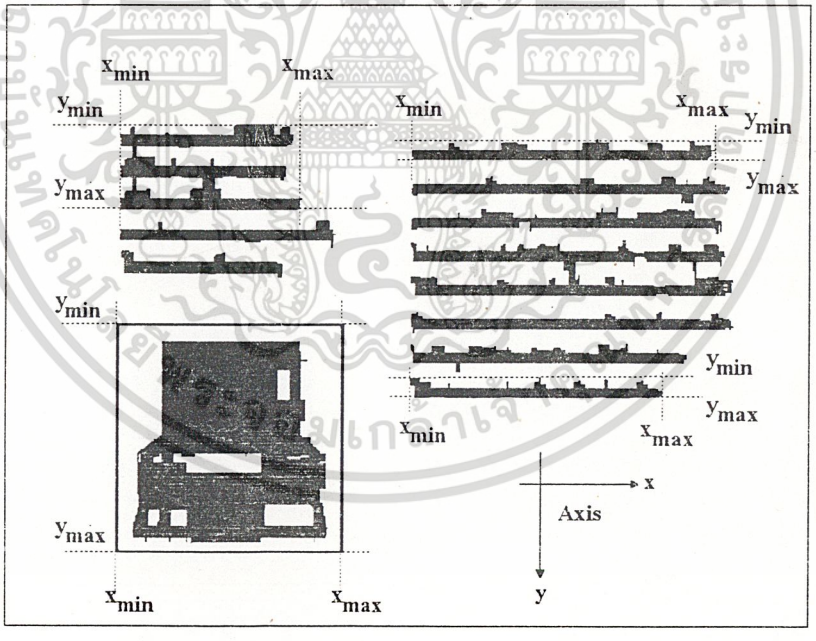
1. พิกัดของจุดมุมบนด้านซ้ายคือ พิกัดของจุดขอบที่มีค่าน้อยที่สุดในแนวตั้งหรือแนวแกน y กับพิกัดของจุดขอบที่มีค่าน้อยที่สุดในแนวนอนหรือแนวแกน x (y_{min},x_{min})

2. พิกัดของจุดมุมบนด้านขวาคือ พิกัดของจุดขอบที่มีค่าน้อยที่สุดในแนวตั้งหรือแนวแกน y กับพิกัดของจุดขอบที่มีค่ามากที่สุดในแนวนอนหรือแนวแกน x (y_{min},x_{max})

3. พิกัดของจุดมุมล่างด้านซ้ายคือ พิกัดของจุดขอบที่มีค่ามากที่สุดในแนวตั้งหรือแนวแกน y กับพิกัดของจุดขอบที่มีค่าน้อยที่สุดในแนวนอนหรือแนวแกน x (y_{max},x_{min})

4. พิกัดของจุดมุมล่างด้านขวาคือ พิกัดของจุดขอบทีมีค่ามากที่สุดในแนวตั้งหรือแนวแกน y กับพิกัดของจุดขอบที่มีค่ามากที่สุดในแนวนอนหรือแนวแกน x (y_{max},x_{max})

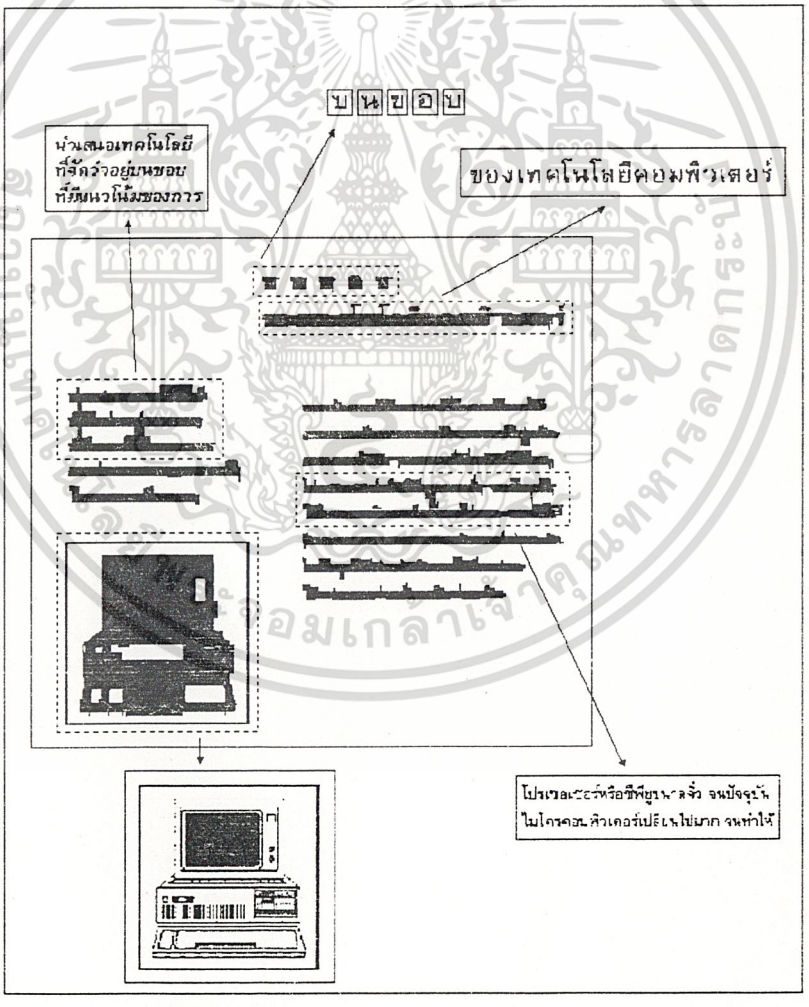
โดยกำหนดให้จุดมุมบนด้านซ้ายของข้อมูลภาพทั้งหมดเป็นจุดเริ่มต้น (Origin point) มีพิกัดของจุดเป็น (0,0) แนวแกน x คือแกนอ้างอิงในแนวนอน และแนวแกน y คือแกนอ้างอิงในแนวตั้ง ในรูปที่ 3.12 แสดงตัวอย่างลักษณะพิกัดของจุดของขอบเขตหนึ่ง



รูปที่ 3.12 แสดงตัวอย่างพิกัดของขอบเขต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคำนวณหาจุดมุมทั้ง 4 ของรูปสี่เหลี่ยมได้แล้ว ก็จะสามารถกำหนดขอบเขตของส่วนประกอบของหน้าเอกสารนั้นๆ ได้ในลักษณะของพื้นที่รูปสี่เหลี่ยม จุดภาพทั้งหมดที่อยู่ภายในบริเวณรูปสี่เหลี่ยมนั้นจะถูกคัดลอกออกจากภาพหน้าเอกสาร แล้วนำแยกเก็บไว้ต่างหากเพื่อการประมวลในขั้นตอนต่อไป จุดภาพในบริเวณขอบเขตรูปสี่เหลี่ยมที่ถูกคัดลอกจุดภาพออกไปแล้ว ก็จะถูกเปลี่ยนค่าของจุดภาพภายในบริเวณนั้นๆ ให้มีค่าเป็น 0 หรือกล่าวอีกนัยหนึ่งว่าเปลี่ยนจุดดำของจุดภาพให้กลายเป็นจุดขาวหรือจุดพื้นให้หมด จากนั้นก็จะย้อนกลับไปยังจุดเริ่มต้นที่พบขอบเขตก่อนหน้านี้ที่เพิ่งจะคัดลอกจุดภาพออกไป เริ่มตรวจกวาดหาจุดภาพที่เป็นขอบเขตอื่นต่อไป เพื่อที่จะกำหนดขอบเขตและคัดลอกขอบเขตนั้นๆ ออกมา การทำงานจะทำงานวนไปในลักษณะเช่นนี้ไปเรื่อยๆ จนสามารถกำหนดและแยกคัดลอกขอบเขตที่เป็นส่วนประกอบของหน้าเอกสารออกมาได้ทั้งหมด



รูปที่ 3.13 แสดงตัวอย่างขอบเขตต่าง ๆ ที่หาได้ของภาพข้อมูลตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การจัดเก็บส่วนประกอบของหน้าเอกสาร

ในกรณีที่หน้าเอกสารหนึ่งไม่ได้มีเพียงข้อความที่เป็นตัวอักษรพิมพ์อยู่เพียงอย่างเดียว แต่อาจจะมีรูปภาพกราฟิกพิมพ์รวมอยู่ด้วย เมื่อภาพหน้าเอกสารถูกนำมาผ่านขั้นตอนของการกำหนดและแยกคัดลอกขอบเขตที่เป็นส่วนประกอบของหน้าเอกสารออกแล้ว ส่วนประกอบของหน้าเอกสารที่เป็นส่วนของรูปภาพจะไม่ส่งผ่านไปขั้นตอนต่อไปนั่นคือขั้นตอนของการรู้จำตัวอักษร แต่จะทำการจัดเก็บลงแฟ้มข้อมูลภาพกราฟิกแทน ฟอร์แมตที่กำหนดไว้สำหรับบันทึกข้อมูลที่เป็นส่วนที่เป็นรูปภาพบนหน้าเอกสารเลือกใช้ฟอร์แมต TIFF ซึ่งเป็นฟอร์แมตเดียวกันกับฟอร์แมตของข้อมูลภาพต้นฉบับ หรือข้อมูลภาพทั้งหน้าเอกสารที่ได้จากเครื่องตรวจกวาดภาพ แฟ้มข้อมูลรูปภาพที่เป็นส่วนประกอบหนึ่งของหน้าเอกสารนี้สามารถเรียกใช้งานได้ด้วยโปรแกรมแสดงภาพกราฟิกทั่วๆ ไป เพื่อที่จะนำมาแก้ไขเพิ่มรวมเข้ากับเอกสารอื่น หรือพิมพ์ออกมาทางเครื่องพิมพ์ ซึ่งรายละเอียดของโครงสร้างของแฟ้มข้อมูลภาพกราฟิก TIFF Version 5 ได้กล่าวไว้ในภาคผนวก ก.

สำหรับส่วนประกอบของหน้าเอกสารที่อาจเป็นย่อหน้าของข้อความ อาจเป็นประโยค หรืออาจจะเป็นคำ จะถูกส่งต่อไปยังขั้นตอนต่อไปคือขั้นตอนของการรู้จำตัวอักษร ซึ่งจะต้องทำการแยกภาพตัวอักษรเดี่ยวๆ ออกจากภาพย่อหน้า ประโยค หรือคำเสียก่อน แล้วทำการคำนวณหารหัสแทนตัวอักษรนั้นๆ เพื่อใช้ในการรู้จำตัวอักษรนั้นต่อไป โดยรายละเอียดของการทำงานทั้งหมดจะได้กล่าวต่อไปในบทที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การรู้จำตัวอักษร

4.1 บทนำ

การรู้จำตัวอักษรเป็นขบวนการทำงานหนึ่งที่ทำให้เครื่องคอมพิวเตอร์สามารถรู้จักภาพตัวอักษรที่พิมพ์อยู่บนหน้ากระดาษ สามารถบอกได้ว่าตัวอักษรที่พบเป็นตัวอักษรอะไร และยังสามารถจดจำภาพตัวอักษรนั้นๆ ได้ เพื่อที่ว่าเมื่อใดก็ตามหากพบภาพตัวอักษรนี้อีกครั้ง ก็จะสามารถระบุภาพตัวอักษรนั้นได้อย่างถูกต้อง ได้มีการศึกษาค้นคว้าอย่างแพร่หลายเพื่อหาวิธีการในการรู้จำตัวอักษร วิธีการที่แตกต่างกันหลายๆ วิธีถูกนำเสนอพร้อมกับประสิทธิภาพของการรู้จำที่แตกต่างกันด้วย แต่ทั้งนี้วิธีการแต่ละวิธีจะขึ้นกับชนิดของข้อมูล และวัตถุประสงค์ที่ต้องการนำผลลัพธ์ของการรู้จำไปใช้งานในลักษณะใด ในงานวิจัยเรื่องการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารและการรู้จำตัวอักษรของวิทยานิพนธ์ฉบับนี้ ในส่วนของวิธีการของการรู้จำตัวอักษร ได้เลือกเอาวิธีการพิจารณาคุณสมบัติทางโทโพโลยีของลายเส้นตัวอักษรมาใช้ เนื่องจากการทำงานไม่ยุ่งยากซับซ้อน ซึ่งส่งผลโดยตรงให้การทำงานเป็นไปได้อย่างรวดเร็ว และประสิทธิภาพของการรู้จำในส่วนของความถูกต้องอยู่ในเกณฑ์ที่ดี

แต่ก่อนที่จะถึงขั้นตอนของการรู้จำตัวอักษร จะต้องมีขบวนการทำงานอีกชุดหนึ่งที่จะจัดเตรียมข้อมูลป้อนให้กับส่วนของการรู้จำตัวอักษรได้อย่างถูกต้องและเหมาะสม เอาท์พุทหรือผลลัพธ์ที่ได้จากขั้นตอนของการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร ซึ่งเป็นขั้นตอนการทำงานก่อนหน้านี้ของระบบทั้งหมด จะส่งผ่านข้อมูลมาให้มีลักษณะเป็นภาพของกลุ่มตัวอักษรพิมพ์อยู่รวมกันเป็นย่อหน้าของข้อความ หรืออาจเรียงกันเป็นประโยค หรือไม่ก็ประกอบกันเป็นคำๆ ดังนั้นในขั้นตอนของการเตรียมข้อมูลให้กับส่วนของการรู้จำตัวอักษรจึงเป็นการแยกภาพตัวอักษรออกเป็นตัวอักษรเดี่ยวๆ เพื่อนำไปประมวลผลต่อในขั้นตอนต่อไป

ในส่วนของการรู้จำตัวอักษร ได้อาศัยคุณสมบัติทางโทโพโลยีของลายเส้นของตัวอักษร มากำหนดเป็นรหัสแทนตัวอักษรแต่ละตัว ตัวอักษรแต่ละตัวจะมีรหัสที่มีลักษณะเป็นตัวเลขฐานสิบหกจำนวน 9 หลักที่แตกต่างกัน รวบรวมกันไว้เป็นพจนานุกรมสำหรับการรู้จำต่อไป เมื่อมีภาพตัวอักษรที่ต้องการรู้จำก็จะนำภาพตัวอักษรนั้นมาคำนวณหารหัสแทนตัวอักษรโดยพิจารณาคุณสมบัติทางโทโพโลยีของลายเส้นตัวอักษร แล้วนำรหัสที่คำนวณหาได้ไปค้นหาในพจนานุกรมว่าภาพตัวอักษรนั้นเป็นตัวอักษรอะไร

และในขั้นตอนสุดท้ายจะเป็นการแสดงและเก็บบันทึกผลลัพธ์ที่ได้ของการรู้จำตัวอักษร ภาพตัวอักษรที่ผ่านการรู้จำและสามารถระบุได้ว่าเป็นตัวอักษรอะไรแล้ว จะเก็บภาพตัวอักษรแต่ละตัวลงแฟ้มข้อมูลอักขระ (ASCII text file) โดยจัดเรียงตำแหน่งของตัวอักษรบนตำแหน่งเดียวกันกับตำแหน่งที่ภาพตัวอักษรนั้นปรากฏบนภาพคำ หรือภาพประโยค และบนย่อหน้านั้นอย่างถูกต้อง แฟ้มข้อมูลที่จัดเก็บสามารถเรียกใช้งาน เพื่อแก้ไขเพิ่มเติม ได้โดยง่ายด้วยโปรแกรมเวิร์ดโพเซสเซอร์ทั่วๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

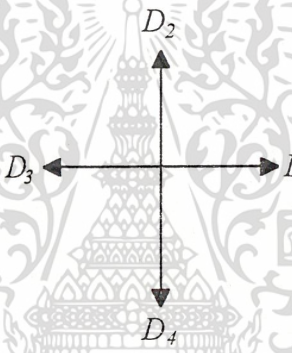
การทำงานในแต่ละขั้นตอนจะได้อธิบายเป็นหัวข้อๆ โดยละเอียดต่อไปดังนี้

4.2 ทฤษฎีที่นำมาใช้

4.2.1 ทฤษฎีฟีเจอร์คอนเซนเทรชั่น[15]

เทคนิคทำงานหนึ่งในทฤษฎีฟีเจอร์คอนเซนเทรชั่นที่ได้นำมาใช้ในงานวิจัยนี้ คือวิธีการค้นหาคุณลักษณะเด่น (Feature extraction) ซึ่งเป็นการเข้ารหัสให้กับจุดภาพใดๆ โดยพิจารณาคุณสมบัติทางโทโพโลยีจุดนั้นๆ ส่วนหนึ่งของทฤษฎีฟีเจอร์คอนเซนเทรชั่นที่นำมาใช้ มีรายละเอียดของการทำงานดังนี้

จุดของภาพที่พิจารณาจะเป็นจุดขาว หรือจุดที่เป็นพื้นเบื้องหลัง (Background) และมีค่าของจุดเป็น 0 จะถูกเข้ารหัสโดยพิจารณาจากลายเส้นของภาพตัวอักษรที่ปรากฏอยู่รอบๆ จุดขาวนั้น ซึ่งได้กำหนดทิศทางที่ใช้ในการพิจารณาไว้ดังรูป 4.1

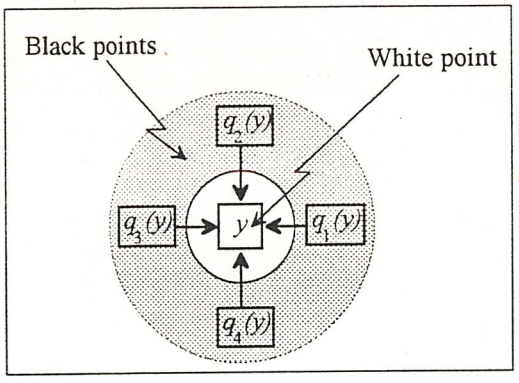


รูปที่ 4.1 แสดงการกำหนดทิศทางทั้ง 4 ทิศ

พิจารณารูปที่ 4.2 ถ้ากำหนดให้ $q_i(y)$ เป็นฟังก์ชั่นๆ หนึ่ง และจะมีค่าเป็น 1 ถ้ามีจุดดำหนึ่งจุดปรากฏอยู่ในทิศทาง D_i จากจุด y และจะมีค่าเป็น 0 ถ้าไม่ปรากฏจุดดำเลยในทิศทางเดียวกัน จุดขาว y จะถูกเข้ารหัสมีค่าเท่ากับ Q_k $(k=0,1,...,15)$ โดยที่ค่า k สามารถคำนวณได้จากสมการ 4.1

$$k(y) = \sum_{j=1}^{4} 2^{4-j} \bullet q_j(y) \qquad (4.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



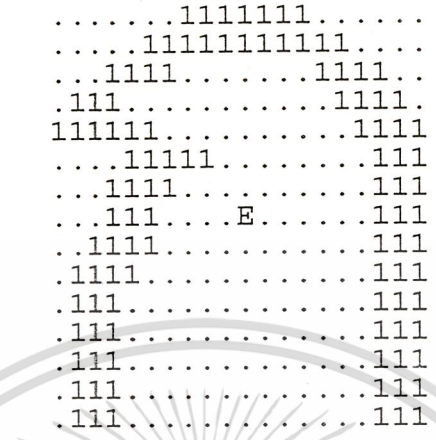
รูปที่ 4.2 แสดงการพิจารณาลักษณะของจุดขาว

ค่าใหม่ของจุดขาวที่เป็นผลมาจากการเข้ารหัส จะเรียกอีกอย่างว่า Q-code สามารถอธิบายลักษณะของจุดดำที่ปรากฏอยู่รอบๆ ได้ด้วยภาพสัญญลักษณ์ดังแสดงในตารางที่ 4.1 และเพื่อความสะดวกในการอ้างอิงถึงค่าของจุดภาพที่ถูกเข้ารหัส จึงใช้เลขฐาน 16 แทนค่า Q-codes ทั้งหมด ตัวอย่างเช่น **0** แทน Q_0, **1** แทน Q_1, ..., **A** แทน Q_{10}, **B** แทน Q_{11}, ..., **E** แทน Q_{14}, และ **F** แทน Q_{15} เป็นต้น

ตารางที่ 4.1 ภาพสัญญลักษณ์ของ Q-codes

Q-code	ภาพสัญญลักษณ์	*Q-code*	ภาพสัญญลักษณ์
Q_0	○	Q_8	○ |
Q_1	○ —	Q_9	⌟
Q_2	| ○	Q_{10}	| |
Q_3	⌞	Q_{11}	⌴
Q_4	¯ ○	Q_{12}	⌝
Q_5	=	Q_{13}	⌝
Q_6	⌜	Q_{14}	⌜⌝
Q_7	⊏	Q_{15}	□

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดง Q-code ณ.ตำแหน่งจุดศูนย์ถ่วงของภาพตัวอักษร ก

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การคำนวณหาจุดศูนย์ถ่วงของภาพ[5]

ภาพตัวอักษรในลักษณะของข้อมูลไบนารี่ (จุดที่มีค่าเป็น 1 แทนจุดดำ และจุดที่มีค่าเป็น 0 แทนจุดขาว) จะถูกนำมาคำนวณหาจุดศูนย์ถ่วงของภาพตัวอักษร ด้วยสมการคณิตศาสตร์ 2 สมการ โดยที่สมการ 4.2 จะคำนวณตำแหน่งพิกัดของจุดศูนย์ถ่วงในแนวแกน x และสมการ 4.3 จะคำนวณตำแหน่งพิกัดของจุดศูนย์ถ่วงในแนวแกน y จุดศูนย์ถ่วงของภาพตัวอักษรแสดงดังรูป 4.3

$$I_x = \frac{\sum_j \sum_i i \bullet F(i,j)}{\sum_j \sum_i F(i,j)} \qquad (4.2)$$

$$I_y = \frac{\sum_i \sum_j j \bullet F(i,j)}{\sum_i \sum_j F(i,j)} \qquad (4.3)$$

เมื่อ $F(i,j)$ คือจุดภาพใดๆ ของภาพตัวอักษรที่มีค่าเป็น 0 หรือ 1

i , j คือตำแหน่งพิกัดของจุดภาพในแนวแถวและคอลัมน์ตามลำดับ

I_x , I_y คือตำแหน่งพิกัด (Coordinate) ของจุดศูนย์ถ่วง

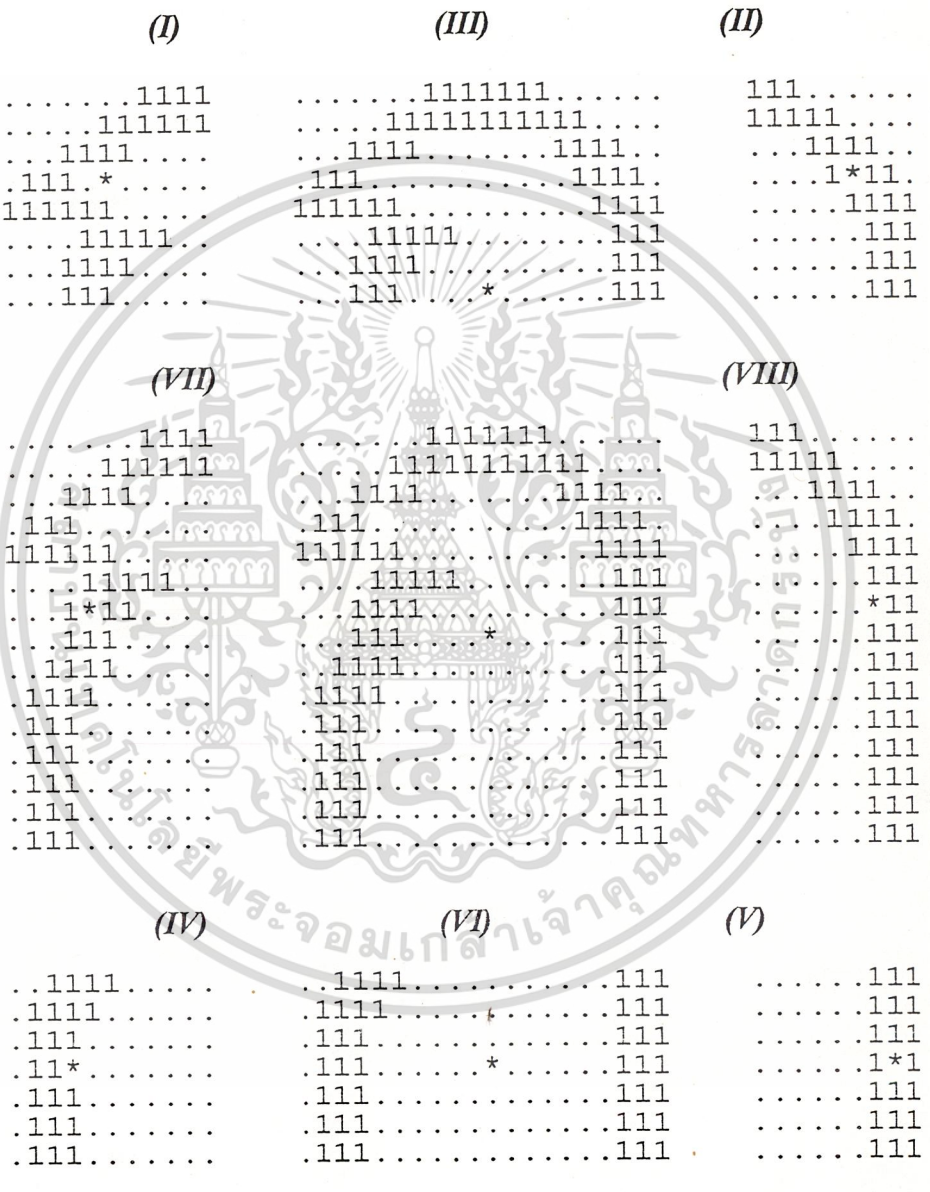
```
.......1111111.....
.....11111111111...
...1111.....1111..
.111.........1111.
111111.........1111
....11111.......111
...1111.........111
...111...*......111
..1111.........111
.1111..........111
.111...........111
.111...........111
.111...........111
.111...........111
.111...........111
```

* = *Center of gravity*

รูปที่ 4.4 แสดงจุดศูนย์ถ่วงของภาพตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ณ.ตำแหน่งจุดศูนย์ถ่วงของภาพนี้ จะนำมาใช้ประกอบการรู้จำตัวอักษรใน 2 ลักษณะคือ หนึ่งใช้เป็นจุดอ้างอิงในการแบ่งภาพตัวอักษรออกเป็น 4 ส่วนด้วยเส้นตรงที่ลากในแนวดิ่ง และเส้นตรงที่ลากในแนวนอนผ่านจุดศูนย์ถ่วงของภาพ ซึ่งสามารถกำหนดพื้นที่ของภาพตัวอักษรได้ทั้งหมด 8 ส่วน ดังแสดงในรูปที่ 4.5 และในลักษณะที่สองคือ ใช้จุดศูนย์ถ่วงของภาพนี้เป็นจุดอ้างอิงในการพิจารณาคุณสมบัติทางโทโพโลยี่ของลายเส้นตัวอักษร คำนวณหาค่า Q-code ณ.จุดนี้ ซึ่งจะเป็นรหัสแทนตัวอักษรค่าหนึ่งในเก้ารหัสของตัวอักษรหนึ่งๆ



รูปที่ 4.5 แสดงพื้นที่ของภาพตัวอักษรที่ถูกแบ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การคำนวณหาความกว้างและความสูงของตัวอักษร[9]

ภาพตัวอักษรแต่ละตัวในขณะอยู่ในขั้นตอนของการแยกภาพตัวอักษรออกจากภาพประโยคหรือคำ จะมีการบันทึกตำแหน่งพิกัดของจุดภาพของตัวอักษรไว้ด้วย เพื่อจะบันทึกขนาดความกว้างและความสูงของตัวอักษร และตำแหน่งพิกัดของภาพตัวอักษรว่าปรากฏอยู่ในแถวและคอลัมน์ใดของภาพย่อหน้าของประโยค หรือตำแหน่งคอลัมน์ใดของภาพคำ

จุดภาพของภาพตัวอักษรจุดแรกที่ตรวจกวาดมาพบจะถูกบันทึกตำแหน่งพิกัดไว้ แนวแกนอ้างอิงตำแหน่งพิกัดกำหนดให้แนวนอนเป็นแกน x และแนวตั้งเป็นแกน y และกำหนดตัวแปรที่ใช้ในการคำนวณไว้ดังนี้

ให้ $p_n(x,y)$: จุดภาพตัวอักษรใด ณ. ตำแหน่งพิกัด x และ y

$n = 1,2,3,...N$

$n = 1$: ให้เป็นจุดแรกของภาพตัวอักษร

$n = N$: ให้เป็นจุดสุดท้ายของภาพตัวอักษร

$p_{n+1}(x,y)$: จุดภาพของภาพตัวอักษรจุดถัดไปที่ตรวจกวาดพบ

x_{min} : ตำแหน่งพิกัดที่มีค่าน้อยที่สุดในแนวแกน x

x_{max} : ตำแหน่งพิกัดที่มีค่ามากที่สุดในแนวแกน x

y_{min} : ตำแหน่งพิกัดที่มีค่าน้อยที่สุดในแนวแกน y

y_{max} : ตำแหน่งพิกัดที่มีค่ามากที่สุดในแนวแกน y

ขั้นตอนการคำนวณหาความกว้างและความสูงของตัวอักษร มีรายละเอียดดังนี้

1. บันทึกตำแหน่งพิกัดจุดภาพของภาพตัวอักษรจุดแรกที่ตรวจกวาดพบ และกำหนดให้ตำแหน่งพิกัดที่มีค่าน้อยที่สุดและมากที่สุดทั้งแกน x และแกน y มีค่าเท่ากับตำแหน่งพิกัดของจุดภาพนี้ด้วย $(x = x_{min} = x_{max}, y = y_{min} = y_{max})$

2. ตรวจกวาดไปยังจุดภาพของตัวอักษรจุดถัดไป บันทึกตำแหน่งพิกัดแล้วทำการเปรียบเทียบตามเงื่อนไขต่อไปนี้

2.1 ถ้าตำแหน่งพิกัดของจุดภาพในแนวแกน x มีค่าน้อยกว่า x_{min} แล้ว ให้บันทึกตำแหน่งพิกัด x_{min} เสียใหม่ด้วยตำแหน่งพิกัด x ของจุดภาพปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ถ้าตำแหน่งพิกัดของจุดภาพในแนวแกน x มีค่ามากกว่า x_{max} แล้ว ให้บันทึกตำแหน่งพิกัด x_{max} เสียใหม่ด้วยตำแหน่งพิกัด x ของจุดภาพปัจจุบัน

2.3 ถ้าตำแหน่งพิกัดของจุดภาพในแนวแกน y มีค่าน้อยกว่า y_{min} แล้ว ให้บันทึกตำแหน่งพิกัด $y_{min.}$ เสียใหม่ด้วยตำแหน่งพิกัด y ของจุดภาพปัจจุบัน

2.4 ถ้าตำแหน่งพิกัดของจุดภาพในแนวแกน y มีค่ามากกว่า y_{max} แล้ว ให้บันทึกตำแหน่งพิกัด y_{max} เสียใหม่ด้วยตำแหน่งพิกัด y ของจุดภาพปัจจุบัน

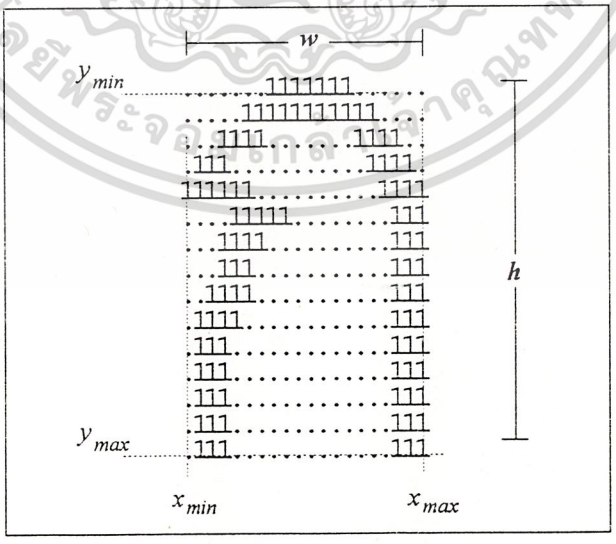
3. ตรวจกวาดไปยังจุดภาพของภาพตัวอักษรจุดถัดไป แล้วย้อนกลับไปพิจารณาเงื่อนไขในข้อที่ 2 โดยทำวนเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงจุดภาพจุดสุดท้ายของภาพตัวอักษร ก็จะได้ตำแหน่งพิกัดที่มีค่ามากที่สุดและน้อยที่สุดของภาพตัวอักษร ซึ่งจะนำมาคำนวณหาความกว้างและความสูงของภาพตัวอักษรได้ด้วยสมการ 4.4 และ 4.5

$$w = x_{max} - x_{min} + 1 \qquad (4.4)$$

$$h = y_{max} - y_{min} + 1 \qquad (4.5)$$

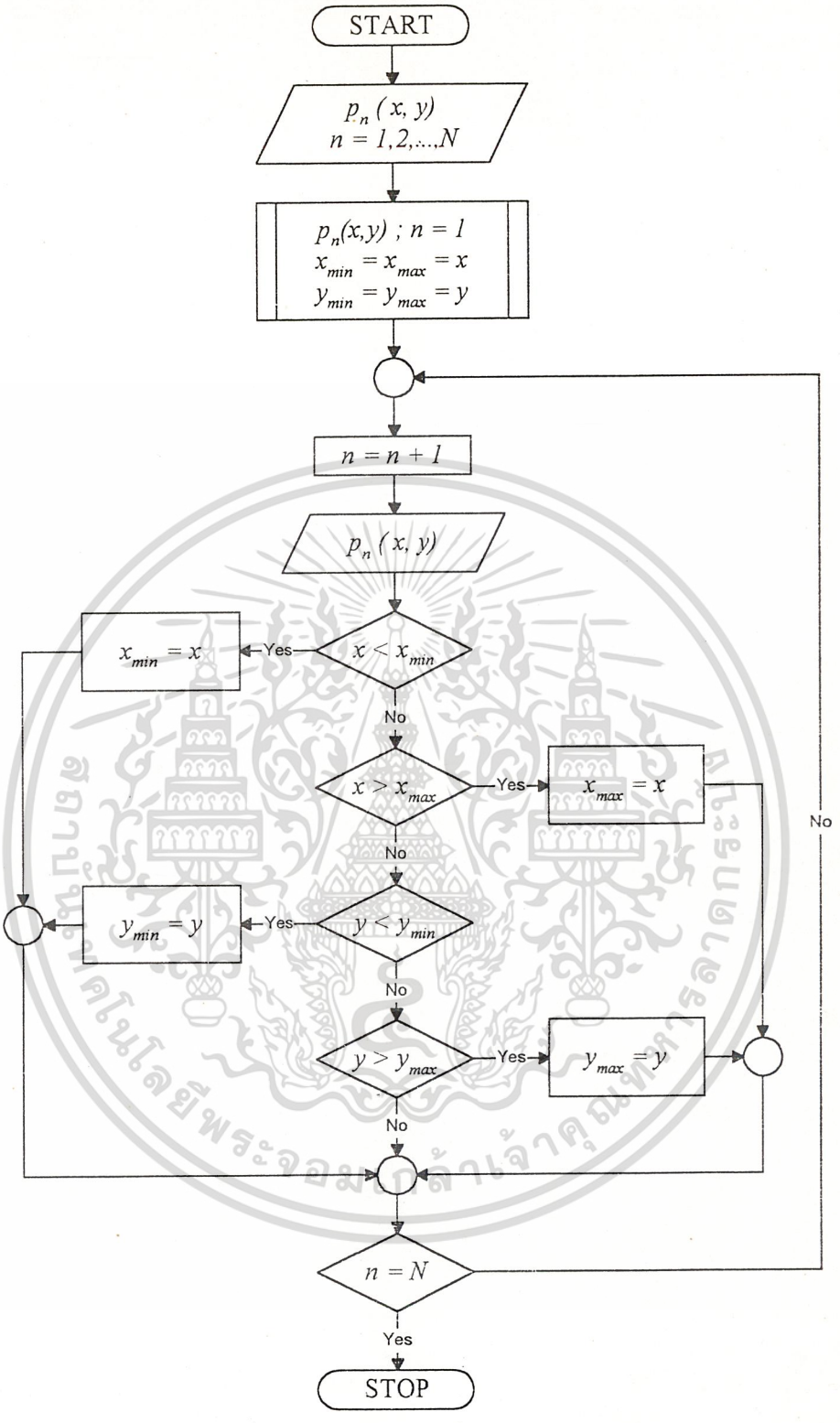
เมื่อกำหนดให้ w : คือความกว้างของภาพตัวอักษร

h : คือความสูงของภาพตัวอักษร



รูปที่ 4.6 แสดงภาพตัวอักษรและความกว้างและความสูงของตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



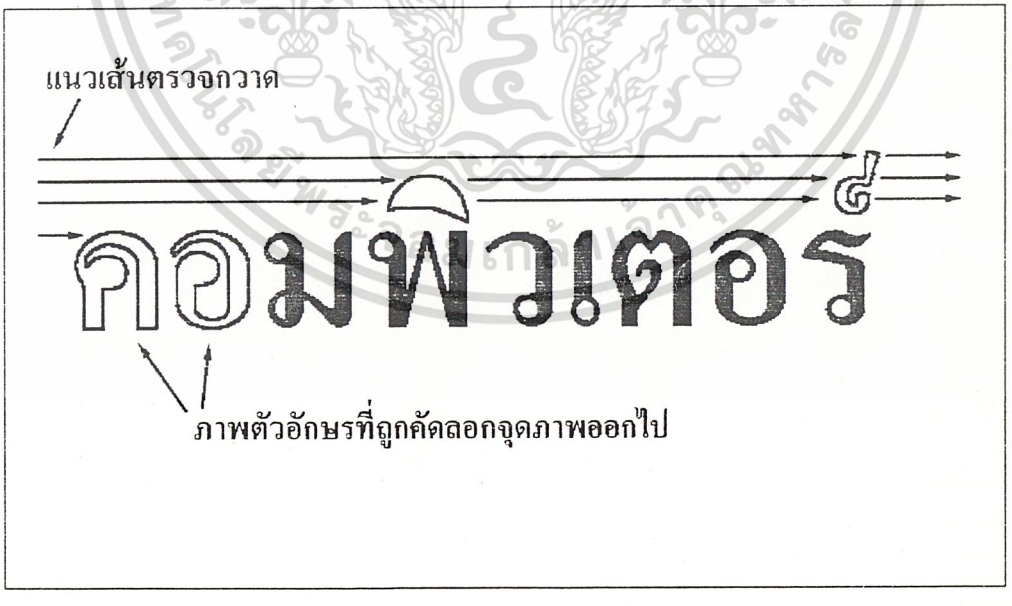
รูปที่ 4.7 แสดงผังการทำงานของการหาความกว้างและความสูงของตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การแยกภาพตัวอักษรออกจากภาพประโยค[11]

ข้อมูลที่ส่งผ่านมาจากขั้นตอนการแยกส่วนประกอบของหน้าเอกสาร จะมีลักษณะเป็นขอบเขตของภาพตัวอักษรหลายๆ ตัวพิมพ์เรียงต่อกันเป็นประโยค และประโยคหลายๆ ประโยครวมเป็นย่อหน้า การรู้จำตัวอักษรจะกระทำได้กับภาพตัวอักษรแต่ละตัวเท่านั้น ดังนั้นจึงจำเป็นจะต้องมีขบวนการทำงานอีกขบวนการหนึ่ง นั่นก็คือขบวนการแยกภาพตัวอักษรออกจากภาพประโยค เป็นการเตรียมข้อมูลให้กับส่วนของการรู้จำตัวอักษร โดยจะทำการแยกภาพตัวอักษรแต่ละตัวออกเป็นภาพตัวอักษรตัวเดี่ยวๆ จากภาพประโยค วิธีการและเทคนิคที่นำมาใช้ก็คือเทคนิคการติดตามรอยขอบของภาพ ซึ่งรายละเอียดของการทำงานของเทคนิคการติดตามรอยขอบของภาพนี้ได้อธิบายไว้ก่อนหน้านี้แล้วในบทที่ 3

การแยกภาพตัวอักษรออกจากภาพประโยค เริ่มจากการตรวจกวาดไปบนจุดภาพของข้อมูลภาพข้อความ จากจุดภาพที่มุมบนซ้ายมือสุดไปทางขวาและเลื่อนบนลงล่าง เมื่อตรวจกวาดมาพบจุดภาพที่เป็นจุดใดจุดหนึ่งของลายเส้นตัวอักษร หรือจุดภาพที่มีค่าเป็น 1 ก็จะนำเอาเทคนิคการติดตามรอยขอบของภาพมาใช้เพื่อค้นหาจุดที่เป็นขอบของภาพตัวอักษร ทำให้ทราบค่าพิกัดของจุดภาพทุกจุดของตัวอักษร แล้วทำการคัดลอกจุดภาพของตัวอักษรนั้นออกมาไว้ต่างหาก จากนั้นจะลบจุดภาพตัวอักษรจากข้อมูลภาพข้อความเดิมออกไป โดยเปลี่ยนภาพตัวอักษรให้เป็นช่องว่างหรือเปลี่ยนค่าของจุดภาพที่เดิมมีค่าเป็น 1 ให้มีค่าเป็น 0 แล้วทำการตรวจกวาดค้นหาภาพตัวอักษรตัวต่อไป โดยภาพตัวอักษรที่ถูกค้นพบและลบออกไปจะได้ไม่ส่งผลกระทบกับภาพตัวอักษรตัวอื่นที่อยู่ข้างเคียง

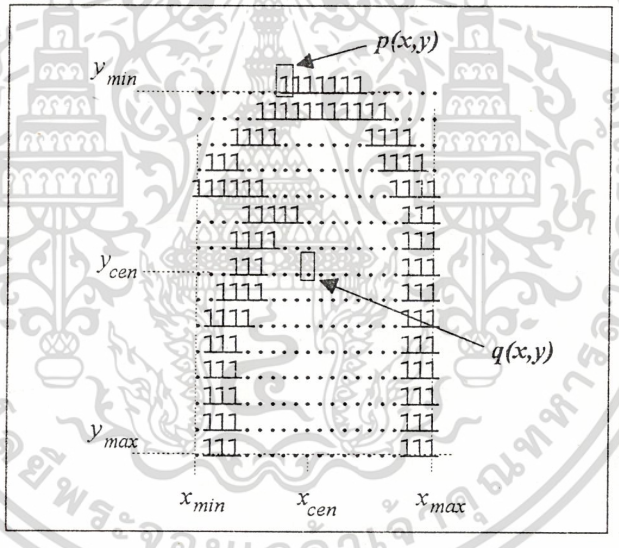


รูปที่ 4.8 แสดงการแยกภาพตัวอักษรออกจากภาพประโยค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพตัวอักษรตัวหนึ่งๆ จะถูกคัดลอกออกจากภาพประโยคในลักษณะที่ไม่ได้เรียงตัวอักษรจากซ้ายไปขวาเช่นเดียวกับการเขียน แต่จะถูกคัดลอกออกมาโดยขึ้นอยู่กับตำแหน่งของภาพตัวอักษรที่ตรวจกวาดไปพบเป็นตัวแรกเมื่อตรวจกวาดจากมุมบนซ้ายมือไปทางขวา และจากแถวบนเลื่อนลงล่าง ดังนั้นจะต้องเก็บบันทึกตำแหน่งพิกัดของภาพตัวอักษรที่ปรากฏบนภาพข้อความไว้ เพื่อที่ว่าเมื่อทำการรู้จำตัวอักษรนั้นๆ แล้ว จะได้เรียงลำดับตัวอักษรกลับให้อยู่ ณ.ตำแหน่งเดิมที่ถูกต้องในข้อความ

จุดพิกัดของจุดภาพที่เป็นลายเส้นตัวอักษรจุดแรกที่ตรวจกวาดพบ *(p(x,y))* จะถูกเก็บบันทึกไว้ และเมื่อทราบจุดขอบของภาพตัวอักษรทั้งหมด ก็จะสามารถคำนวณหาความกว้างและความสูงของตัวอักษรได้ จากนั้นคำนวณหาจุดกึ่งกลางของความกว้างและความสูง (x_{cen}, y_{cen}) ของตัวอักษร แล้วปรับตำแหน่งพิกัด (Coordinate) ของจุดภาพจุดแรกที่ตรวจกวาดพบให้มาอยู่ที่จุดกึ่งกลางของภาพตัวอักษร ซึ่ง ณ.ตำแหน่งพิกัดของจุดภาพนี้ *(q(x,y))* จะถูกเก็บบันทึกไว้อ้างอิงตำแหน่งของภาพตัวอักษร โดยที่ x คือคอลัมน์ และ y คือแถวที่ภาพตัวอักษรปรากฏ ในรูป 4.9 แสดงจุดพิกัดต่างๆ ที่ใช้ในกำหนดเป็นจุดอ้างอิงตำแหน่งของภาพตัวอักษร



รูปที่ 4.9 แสดงจุดอ้างอิงตำแหน่งของภาพตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การพิจารณาลักษณะเด่นของภาพตัวอักษร

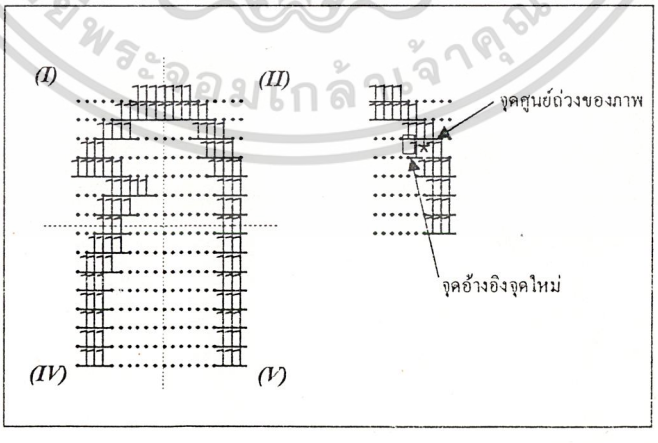
ภาพตัวอักษรตัวหนึ่งๆ จะถูกนำมาคำนวณหาจุดศูนย์ถ่วงของภาพตัวอักษร ซึ่ง ณ. ตำแหน่งจุดศูนย์ถ่วงของภาพตัวอักษรนี้จะถูกใช้เป็นจุดอ้างอิงในการแบ่งพื้นที่ของจุดภาพของภาพตัวอักษรสำหรับการพิจารณาหาลักษณะเด่นของภาพตัวอักษร ดังแสดงในรูปที่ 4.5

ในแต่ละพื้นที่ๆ ของภาพตัวอักษรที่แบ่งได้ จะถูกนำมาคำนวณหาจุดศูนย์ถ่วงของภาพลายเส้นของภาพตัวอักษรที่ปรากฏในพื้นที่นั้นๆ ทำให้ภาพตัวอักษรหนึ่งๆ จะมีจุดอ้างอิงเกิดขึ้นทั้งหมด 9 จุด ซึ่งต่อไปก็จะนำเอาจุดอ้างอิงเหล่านี้มาใช้เป็นตำแหน่งในการพิจารณาคุณสมบัติทางโทโพโลยี่ของลายเส้นของภาพตัวอักษร และเข้ารหัสเป็นค่า Q-codes

ในการคำนวณหาจุดศูนย์ถ่วงของภาพในแต่ละพื้นที่ของภาพตัวอักษรที่ถูกแบ่งออก เพื่อค้นหาจุดที่จะใช้เป็นจุดอ้างอิงในการพิจารณาคุณสมบัติโทโพโลยี่ ตำแหน่งของจุดอ้างอิงในบางพื้นที่อาจไม่สอดคล้องกับทฤษฎีการพิจารณาคุณสมบัติทางโทโพโลยี่ กล่าวคือกรณีที่จุดศูนย์ถ่วงที่คำนวณได้เป็นจุดภาพที่เป็นส่วนของลายเส้นของภาพตัวอักษร ซึ่งเป็นจุดภาพที่มีค่าเป็น 1 จะไม่สามารถนำมาพิจารณาคุณสมบัติทางโทโพโลยี่ได้ ในกรณีเช่นนี้จะต้องทำการปรับแต่งจุดอ้างอิงที่ใช้ ไปยังจุดอื่นที่ถูกต้องและเหมาะสม

4.4.1 การปรับแต่งตำแหน่งของจุดอ้างอิง

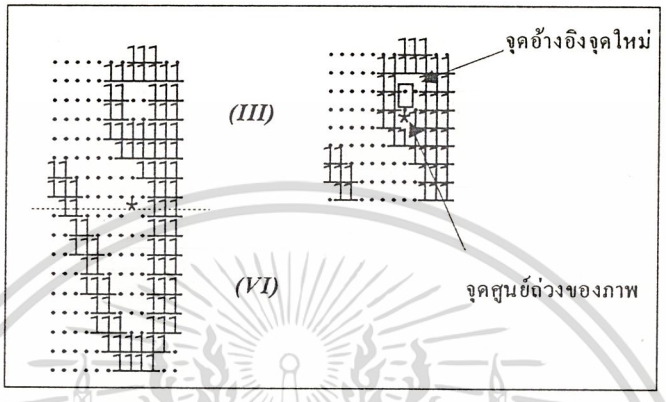
1. ในพื้นที่ส่วนที่ 1, 2, 4, 5, 7 และ 8 ลักษณะของภาพลายเส้นของตัวอักษรในพื้นที่เหล่านี้จะปรากฏอยู่ในแนวตั้งด้านซ้ายหรือด้านขวา หรืออยู่ในแนวตั้งแล้วต่อเนื่องไปด้านบนหรือด้านล่าง ดังนั้นการเลือกจุดอ้างอิงจุดใหม่จึงเลือกปรับจุดศูนย์ถ่วงของพื้นที่นั้นๆ ไปในแนวนอน ด้านใดด้านหนึ่งที่คำนวณได้ว่าจุดภาพที่มีค่าเป็น 0 ที่อยู่ใกล้ที่สุด แล้วเลือกใช้จุดภาพนั้นเป็นจุดอ้างอิง



รูปที่ 4.10 แสดงจุดอ้างอิงที่ปรับแต่งจากจุดศูนย์ถ่วงในพื้นที่ส่วนที่ 2 ของตัวอักษร ก

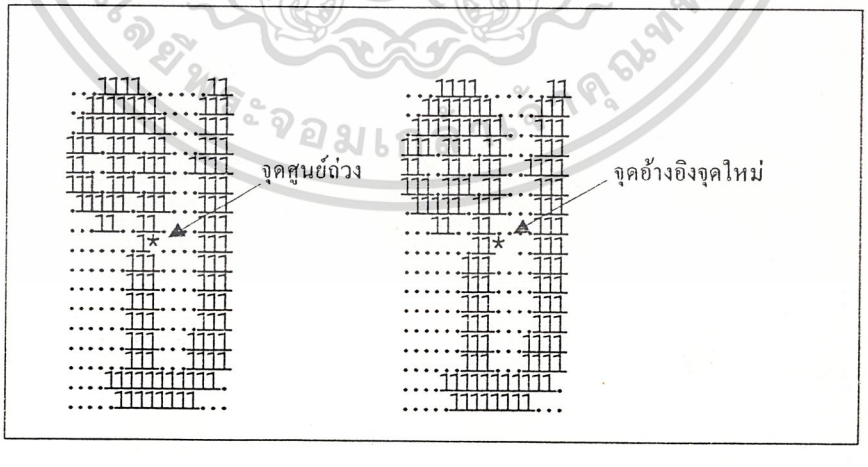
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ในพื้นที่ส่วนที่ 3 และ 6 จะพบว่าลักษณะของภาพลายเส้นของตัวอักษรปรากฏอยู่ทั้งด้านซ้ายและขวาของพื้นที่ หรืออาจต่อเนื่องจากซ้ายไปขวาโดยมีลายเส้นอยู่ด้านบนหรือด้านล่าง ดังนั้นจุดอ้างอิงจุดใหม่จึงเลือกปรับจุดศูนย์ถ่วงขึ้นลงในแนวตั้ง คำนวณหาจุดภาพที่มีค่าเป็น 0 ที่อยู่ใกล้ที่สุด แล้วเลือกจุดภาพนั้นเป็นจุดอ้างอิง



รูปที่ 4.11 แสดงจุดอ้างอิงที่ถูกปรับแต่งในพื้นที่ส่วนที่ 3 ของตัวอักษร ง

3. สำหรับภาพตัวอักษรทั้งตัว ถ้าจุดศูนย์ถ่วงของภาพที่คำนวณได้เป็นจุดภาพที่มีค่าเป็น 1 จะทำการปรับตำแหน่งของจุดที่จะใช้อ้างอิงเสียใหม่ โดยเลือกปรับแต่งไปเป็นจุดภาพที่มีค่าเป็น 0 ที่ใกล้ที่สุดในทิศทางใดทิศทางหนึ่งของทั้ง 4 ทิศทางคือ ด้านบน ด้านขวา ด้านล่าง และด้านซ้าย

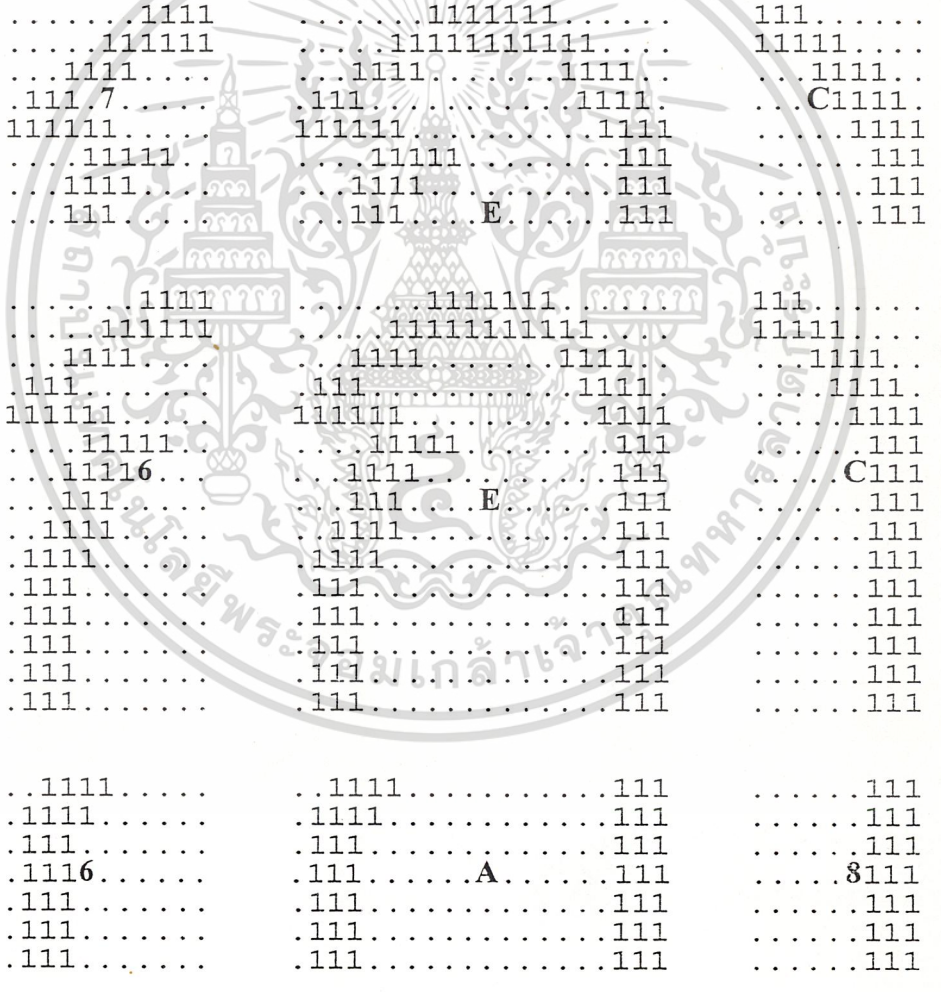


รูปที่ 4.12 แสดงจุดอ้างอิงที่ถูกปรับแต่งของภาพตัวอักษร ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 การพิจารณาคุณสมบัติทางโทโพโลยีของลายเส้นของภาพตัวอักษร

ในแต่ละส่วนของภาพตัวอักษรที่ถูกแบ่งออกแล้ว จะถูกนำมาพิจารณาหาคุณสมบัติทางโทโพโลยีของลายเส้นของภาพตัวอักษร โดยอาศัยทฤษฎีเฟียเจอร์คอนเซนเทรชั่น ซึ่งเป็นการพิจารณาลายเส้นของตัวอักษรที่ปรากฏล้อมรอบจุดอ้างอิงจุดหนึ่งๆ ในแต่ละพื้นที่ของภาพตัวอักษร และลักษณะของลายเส้นของภาพตัวอักษรนี้ จะถูกนำมาใช้เข้ารหัสให้กับจุดภาพ ณ.ตำแหน่งจุดอ้างอิงที่ใช้พิจารณาได้เป็นรหัส Q-code โดยจะมีรหัส Q-code ทั้งหมด 9 รหัสจากจุดอ้างอิง 9 จุดสำหรับตัวอักษรหนึ่งตัว ซึ่งรหัส Q-code นี้จะนำมาใช้ระบุตัวอักษรที่ต้องการรู้จำว่าเป็นตัวอักษรอะไร รูปที่ 4.13 ได้แสดงรหัส Q-codes ของภาพตัวอักษร ก ที่จุดอ้างอิงทั้ง 9 จุด และในตารางที่ 4.2 แสดงตัวอย่างรหัส Q-code ของตัวอักษรภาษาไทย ในรูปแบบหรือฟอนต์ AngsanaUPC ขนาด 14 points



รูปที่ 4.13 แสดงรหัส Q-code ของภาพตัวอักษร ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 แสดงตัวอย่างค่า Q-code

ฟอนต์ : AngsanaUPC ขนาด 14 points

ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*
ก	7CE68A6CE	ข	FAF8BBEBB	ฃ	FAFABBFBB	ค	FCF68ECEE
ฅ	FCF68ECEE	ฆ	FA3DBEFBB	ง	8FFCFB3DF	จ	CCFCBBCFF
ฉ	7CE83E7DF	ช	F6F9BBFFF	ซ	FBF9BFFFF	ฌ	7AEABF7BF
ญ	7AB8BE7BE	ฎ	6CE5BF7DF	ฏ	6CED9F7DF	ฐ	77FCBBFFF
ฑ	EEA8AEFEB	ฒ	FAE6BFFBF	ณ	FABA3FFBE	ด	6CFC8EFEF
ต	6CFC8EFEF	ถ	7CE3AB7CF	ท	FEB88ACEE	ธ	F5799BDDF
น	22A3FEC9B	บ	28A99BD9B	ป	E88D9BD9B	ผ	E8F6CEFBE
ฝ	E88FBEF9B	พ	2AB8CACBE	ฟ	E88CBECBE	ภ	8CE28ADCE
ม	28A8BBD9B	ย	F8E79BD9F	ร	75772F5DD	ล	DFF8AB7CF
ว	6CE0FF6DD	ศ	7DF68ECEE	ษ	38B7DFD9F	ส	76F82A76E
ห	29B68ACEE	ฬ	ABE62ABFE	อ	FCF79BCFF	ฮ	76F79F7FF
ฯ	7EE8887EE	ะ	E9B79FF9B	า	6CC08F6CC	ิ	93EAFB7DF
ี	9BBFFB73F	ึ	9F97FB7DF	ื	9BB7DB73F	ุ	6CECCF7DF
ู	EABCBBFBB	เ	8299FF933	แ	F6703FF7D	ใ	FCE03FF3D
ไ	BCE03FB3D	ๆ	FCE08FFCC	็	899BFB77F	่	7FFC6F7FE
้	E9AD6FDBF	๊	EAEB9A7BE	๋	93FC6FFF6	์	096F5FF96
๑	6CF3BFFFF	๒	2B2BFFBFF	๓	EEEBAB7EF	๔	69977BF7F
๕	DB677B77F	๖	D338DB7DD	๗	2CDBBE7FE	๘	999FDB757
๙	77772B757	๐	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงตัวอย่างค่า Q-code

ฟอนต์: TimesNewRomanPS ขนาด 10 points

ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*
A	E3E7DE73F	B	7DF7DFD7F	C	6CE39B757	D	6CE39B7DF
E	7DF7DFD7F	F	7DF7C7D5F	G	6DE3DB7DF	H	7DB7DED7E
I	C6F93FD77	J	86FF2916D	K	7DFFDED77	L	60F39B797
M	EEAEBBFFB	N	CCB3BEFFB	O	6CE39B7DF	P	6DE747DDF
Q	6CE37B7DF	R	6DF73ED7F	S	7CF3DFDFD	T	EEF93FD77
U	6CA39B7DB	V	7DACBB39B	W	FAFBBABBE	X	7DB63ED77
Y	7DBD7FC7B	Z	6DF79FD77				
a	6DF73B7DF	b	3333DB7DF	c	6CE36B757	d	9DD39B7DF
e	7DF36B757	f	9C792FD57	g	76F7DF757	h	33373ADDE
i	C2993FD3C	j	0C9F993DD	k	CD2F3F37B	l	C2993FD3C
m	EEBBAFFEE	n	CCE33A7DE	o	6CE39B7DF	p	6CE7677DF
q	7CECDD73F	r	C9B92FD67	s	7DF3DFD7F	t	F2983CC77
u	CCA33B3DB	v	76AC6B39B	w	FAFCBABBE	x	36F73ED77
y	76B969D9B	z	6DF79FD77				
1	C2993FD3C	2	9CE9DF5DD	3	6DF3DD577	4	6E9FFC7DF
5	757FD957D	6	7573DB7DF	7	9DE9694DD	8	7DF7DFD7F
9	7CE5DD7DF	0	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การจัดเรียงรูปประโยคหลังการวิเคราะห์

ตัวอักษรแต่ละตัวที่ถูกแยกออกจากภาพประโยค ก่อนที่จะส่งผ่านไปทำการวิเคราะห์และรู้จำ จะถูกบันทึกค่าตำแหน่งพิกัด และคำนวณหาขนาดความกว้างและความสูงของตัวอักษร ซึ่งก็จะนำไปคำนวณหาค่าตำแหน่งของตัวอักษร (x_{cen}, y_{cen}) ในภาพประโยคได้ เมื่อภาพตัวอักษรถูกส่งไปวิเคราะห์ และได้ผลลัพธ์ออกมาแล้วว่าเป็นตัวอักษรอะไร รหัส ASCII ของภาพตัวอักษรนั้นจะถูกนำมาจัดเรียงลำดับก่อนการจัดบันทึกลงไฟล์ โดยจะจัดเรียงตัวอักษรตามค่า y_{cen} ของตัวอักษรแต่ละตัวจากค่าน้อยไปมาก เพื่อจัดเรียงตัวอักษรให้อยู่ในบรรทัดที่ถูกต้อง และจัดเรียงตามค่า x_{cen} จากค่าน้อยไปหามาก เพื่อจัดเรียงตัวอักษรให้อยู่ในตำแหน่งที่ประกอบขึ้นเป็นคำที่ถูกต้อง ตัวอย่างของการจัดเรียงตัวอักษรมีขั้นตอนการจัดเรียงดังนี้

ขั้นตอนที่ 1 ตัวอย่างข้อความเดิมเมื่อยังไม่ถูกแยกตัวอักษรออกจากประโยค

คอมพิวเตอร์กราฟิคเป็นเทค (บรรทัดที่ 1)
โนโลยีทางคอมพิวเตอร์แขนงใหม่ (บรรทัดที่ 2)

ขั้นตอนที่ 2 เมื่อผ่านขั้นตอนการแยกตัวอักษรออก จะแยกตัวอักษรในแต่ละบรรทัดออกมาโดยมีลำดับดังนี้

ฟปคอมพวเตอรกราคเนเทค (บรรทัดที่ 1)
โโในลยทางคอมพวเตอรแขนงหม (บรรทัดที่ 2)

ขั้นตอนที่ 3 ในบรรทัดที่ 1 แม้ว่าตัวอักษร "ไม้ไต่คู้" จะถูกแยกออกมาก่อนก็ตาม เมื่อจัดเรียงตัวอักษรตามค่า x_{cen} จะได้ตัวอักษร "ค" เป็นตัวอักษรตัวแรกแล้วตามด้วยตัวอักษร "อ" เรียงต่อไปเรื่อยๆ จนหมดบรรทัด จากนั้นจะมาจัดเรียงตัวอักษรตามค่า y_{cen} ก็จะได้ตัวอักษร "โ" อยู่ในบรรทัดที่ 2 และจัดเรียงตามค่า x_{cen} ก็จะสามารถเรียงตัวอักษรในบรรทัดที่ 2 ได้ เมื่อจัดเรียงตัวอักษรทั้งหมดแล้ว จะได้ตัวอักษรในแฟ้มข้อมูลอักขระเรียงเป็นประโยคข้อความในลักษณะเดิมเหมือนกับข้อมูลอินพุทในตอนแรก

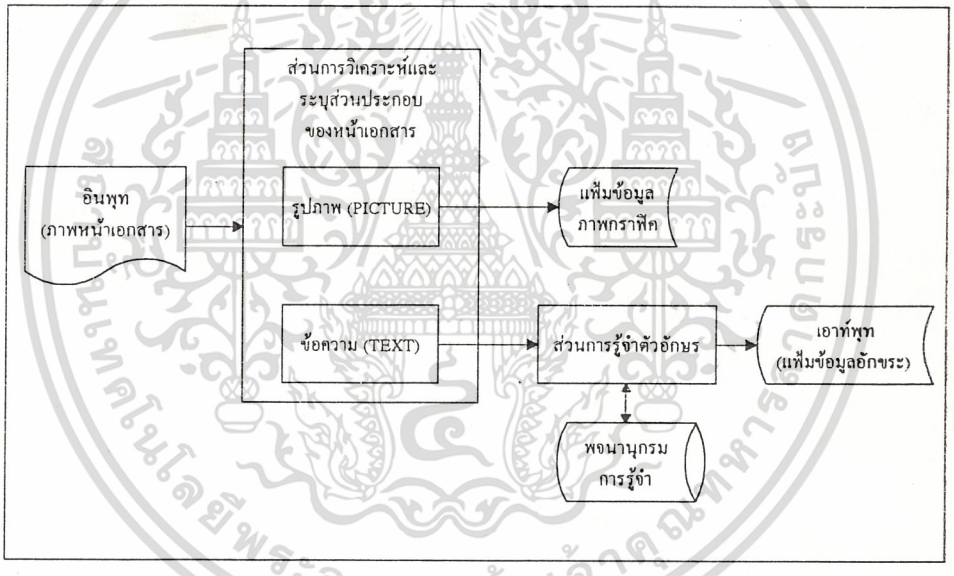
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

โปรแกรมระบบและผลการทดลอง

5.1 ขั้นตอนการทำงานของระบบ

การทำงานของระบบทั้งหมดประกอบด้วยขบวนการทำงานในแต่ละขั้นตอนดังแสดงในรูปที่ 5.1 โดยเริ่มต้นจากข้อมูลภาพหน้ากระดาษเอกสารถูกตรวจกวาดด้วยเครื่องตรวจกวาดภาพ ในขั้นตอนต่อไปเป็นการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร ส่วนประกอบของหน้าเอกสารที่เป็นรูปภาพจะถูกแยกออกจัดเก็บไว้ และในส่วนที่เป็นข้อความตัวอักษรจะถูกส่งต่อไปให้กับส่วนของการรู้จำตัวอักษร ทำการรู้จำตัวอักษรโดยมีพจนานุกรมสำหรับการรู้จำเป็นฐานข้อมูล ผลลัพธ์ที่ได้คือแฟ้มข้อมูลอักขระของข้อความบนหน้าเอกสาร



รูปที่ 5.1 แสดงการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

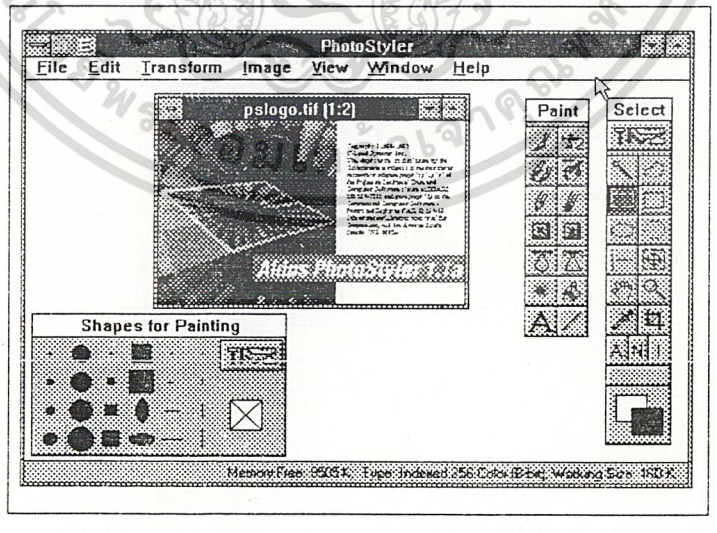
ในการทำงานของระบบจะมีทั้งโปรแกรมคอมพิวเตอร์ที่นำมาใช้และพัฒนาขึ้นเองสำหรับการทำงานในแต่ละขั้นตอน โดยแบ่งโปรแกรมที่ใช้งานเป็น 2 ส่วนคือ

(1) โปรแกรมสำเร็จรูปสำหรับเก็บภาพจากเครื่องตรวจกวาดภาพ

(2) โปรแกรมที่พัฒนาขึ้นสำหรับการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร และการรู้จำตัวอักษร

5.1.1 โปรแกรมสำเร็จรูปสำหรับเก็บภาพจากเครื่องตรวจกวาดภาพ

โปรแกรมสำเร็จรูปสำหรับเก็บภาพหน้าเอกสารที่เลือกใช้มีชื่อว่า PhotoStyler Version 1.1a ซึ่งประกอบด้วยดีไวซ์ไดรว์เวอร์สามารถควบคุมการทำงานของเครื่องตรวจกวาดยี่ห้อ Hewett Package รุ่น ScanJet ที่เป็นเครื่องตรวจกวาดที่ใช้ในงานวิจัยนี้ได้เป็นอย่างดี โดยผู้ใช้สามารถเลือกระดับความละเอียดของการตรวจกวาด (Scanning resolution) ได้ตั้งแต่ 38 - 600 dpi (Dot per inch) และขนาดของหน้าเอกสารที่สามารถตรวจกวาดได้มีขนาดใหญ่ได้ถึงขนาด 8.5 x 14 นิ้ว ข้อมูลอินพุทในลักษณะของภาพหน้าเอกสารสามารถเก็บบันทึกไว้ในแฟ้มข้อมูลกราฟิกได้ในหลายๆ รูปแบบ เช่น TIF, BMP, PCX หรือ GIF สำหรับงานวิจัยนี้ได้เลือกรูปแบบ TIF ซึ่งเป็นรูปแบบแฟ้มข้อมูลที่เหมาะสมกับการจัดเก็บข้อมูลภาพจากเครื่องตรวจกวาดภาพ ทั้งนี้จากข้อได้เปรียบของโครงสร้างของแฟ้มข้อมูลแบบนี้ซึ่งมีลักษณะการจัดเก็บข้อมูลที่ไม่แน่นอนตายตัว ดังนั้นจึงเป็นการง่ายสำหรับการเพิ่มเติมรายละเอียดของข้อมูลเมื่อมีการแก้ไขหรือปรับแต่งข้อมูลภาพภายหลัง (รายละเอียดของโครงสร้างของแฟ้มข้อมูลแบบ TIFF กล่าวไว้ในส่วนภาคผนวก ก. ตอนท้ายของวิทยานิพนธ์นี้)



รูปที่ 5.2 แสดงภาพโปรแกรม PhotoStyler

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 โปรแกรมระบบที่พัฒนาขึ้น

โปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นนี้พัฒนาขึ้นด้วยภาษา C โดยใช้ตัวคอมไพล์เลอร์ชื่อ Borland C++ Version 3.1 ของบริษัท Borland International ทำงานบนสภาวะของระบบปฏิบัติการดอส (DOS Enviroment) บนเครื่องไมโครคอมพิวเตอร์ตระกูล IBM ทั่วๆ ไป

5.1.2.1 อุปกรณ์ที่ต้องการสำหรับการทำงานของระบบ

1. เครื่องไมโครคอมพิวเตอร์ตระกูล IBM ที่ใช้ตัวประมวลผลกลาง (CPU) 80386 ขึ้นไป
2. หน่วยความจำหลักไม่น้อยกว่า 4 Mbytes
3. ระบบจัดการหน่วยความจำขยาย (Expand Memory Management, EMS)
4. ฮาร์ดิสก์จำนวน 1 ตัวและมีเนื้อที่ว่างประมาณ 5 MBytes
5. จอภาพแสดงผลแบบ VGA
6. ฟลอปปี้ดิสก์ไดร์ฟขนาด 5.25 นิ้ว หรือขนาด 3.5 นิ้ว จำนวน 1 ตัว
7. เมาส์ พร้อมไดร์เวอร์ Microsoft Mouse Version 8.02 ขึ้นไป
8. เครื่องตรวจกวาดภาพ Hewlett Package รุ่น ScanJet พร้อม Interface card
9. โปรแกรมควบคุมเครื่องตรวจกวาดภาพ PhotoStyler Version 1.1a
10. โปรแกรมการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารและการรู้จำตัวอักษร

5.1.2.2 ขั้นตอนการทำงานของโปรแกรมที่พัฒนาขึ้น

การทำงานทั้งหมดของโปรแกรมการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารสำหรับการรู้จำตัวอักษร มีการสั่งงานด้วยระบบพูลดาวน์เมนูโดยผ่านทางคีย์บอร์ดหรือเมาส์เพื่อความสะดวกและทำงานได้อย่างเป็นขั้นเป็นตอน โดยจะขออธิบายการใช้งานโปรแกรมเป็นขั้นตอนได้ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 1 Run โปรแกรมโดยการพิมพ์

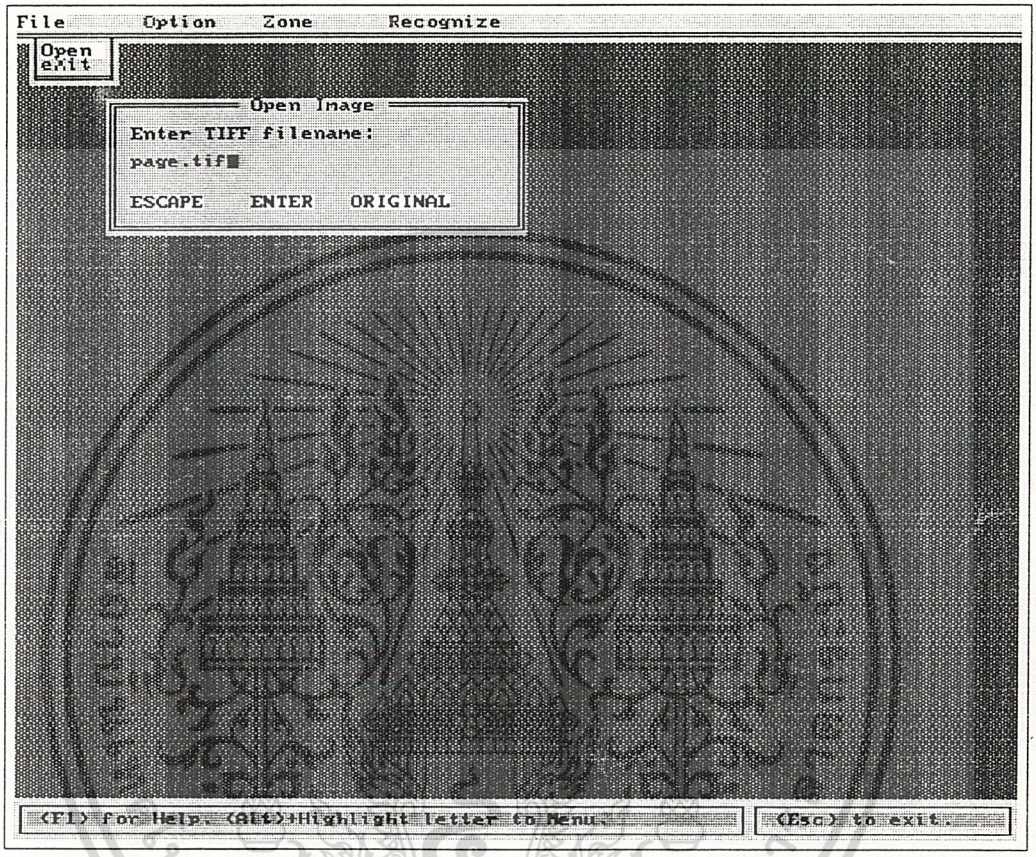
C:>OCR <ENTER>

จากนั้นก็จะปรากฏภาพหน้าจอการทำงานพร้อมระบบพูลดาวน์เมนูสำหรับสั่งงาน ดังแสดงในรูป 5.3 ซึ่งพร้อมที่จะทำงานในขั้นตอนต่อไปทันที

รูปที่ 5.3 แสดงภาพหน้าจอของโปรแกรมที่พัฒนาขึ้น

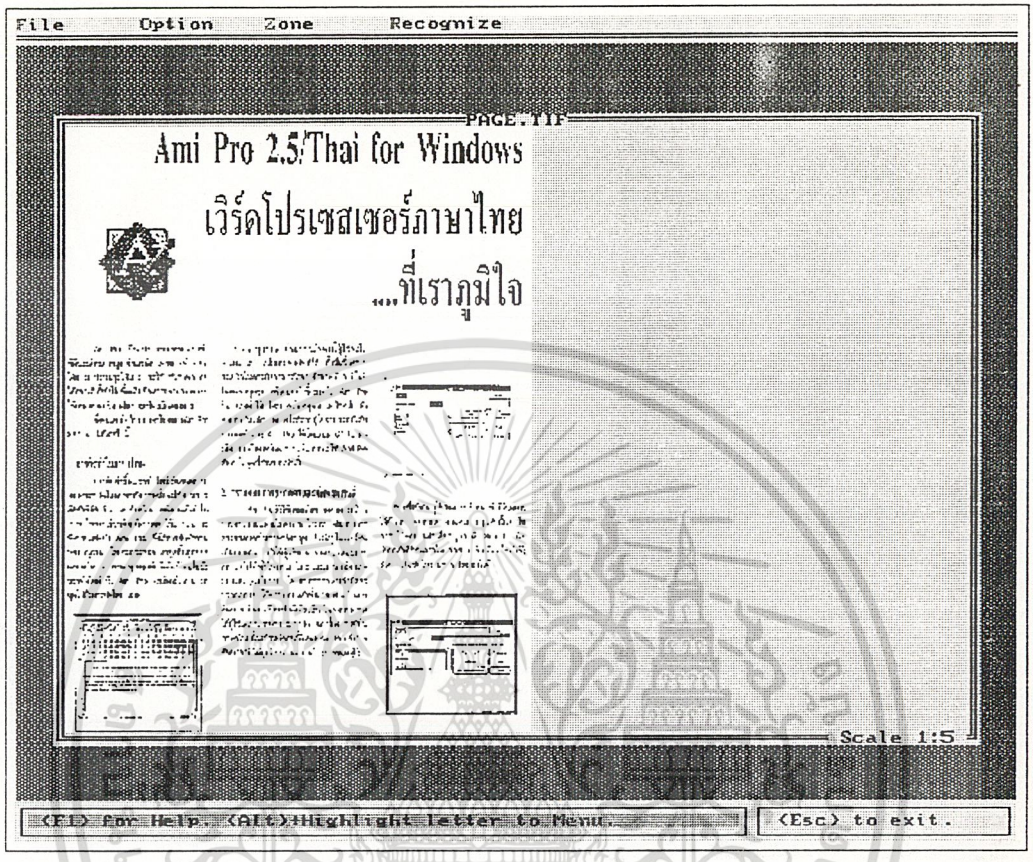
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 เลือกเมนู File บนพูลดาวน์เมนู แล้วเลือกฟังก์ชั่น Open เพื่อเปิดแฟ้มข้อมูลรูปภาพหน้าเอกสารซึ่งถือว่าเป็นอินพุทของโปรแกรม โดยพิมพ์ชื่อแฟ้มข้อมูลที่ต้องการ ดังแสดงในรูป 5.4 หลังจากนั้นจะปรากฏวินโดว์แสดงรูปกราฟิคของภาพหน้าเอกสาร ดังแสดงในรูปที่ 5.5



รูปที่ 5.4 แสดงวินโดว์ของการเปิดแฟ้มข้อมูล

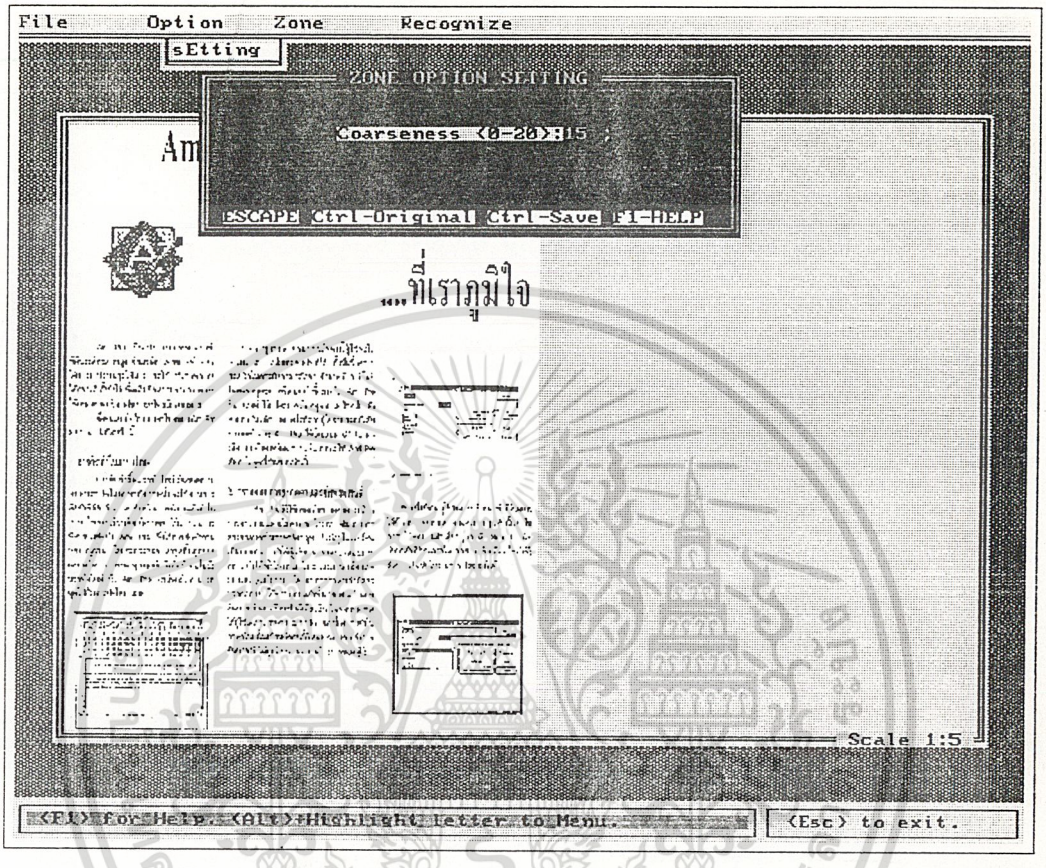
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงตัวอย่างภาพหน้าเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

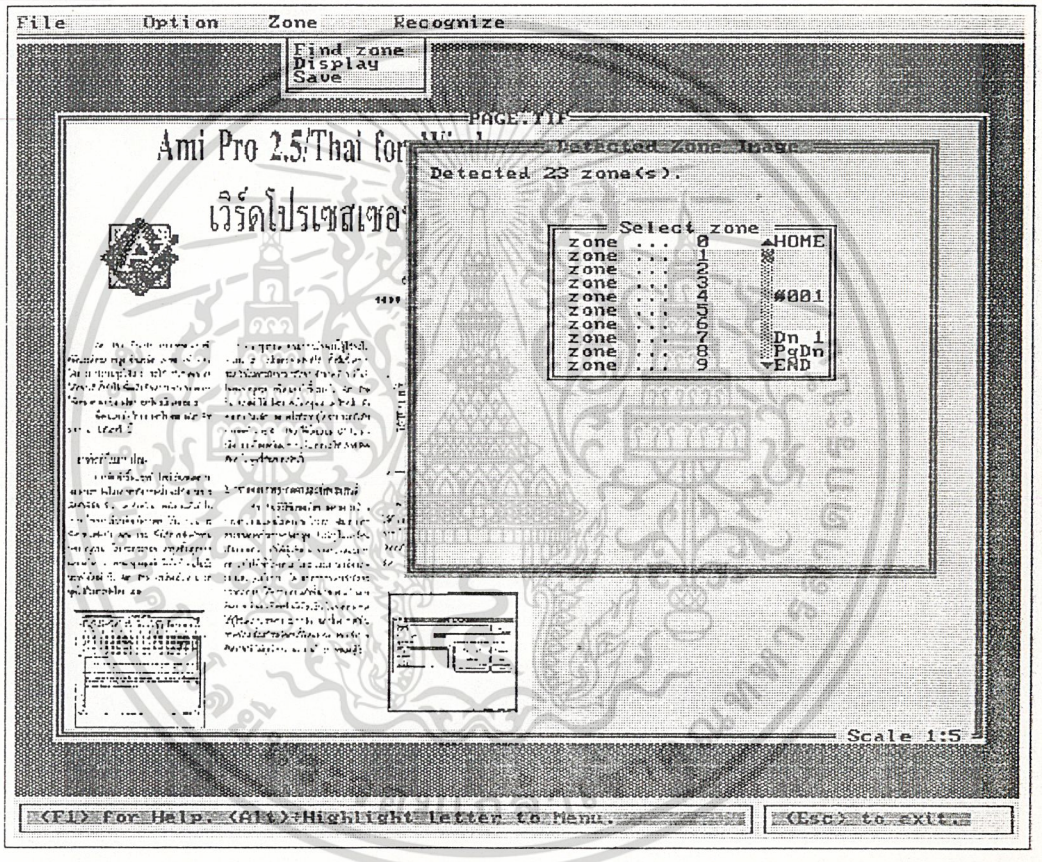
ขั้นตอนที่ 3 เลือกเมนู Setting บนพูลดาวน์เมนู เพื่อติดตั้งค่าความหยาบของจุดภาพ (Coarseness) ซึ่งจะถูกนำมาใช้คำนวณหาขอบเขตของส่วนประกอบของหน้าเอกสาร



รูปที่ 5.6 แสดงวินโดว์ของการติดตั้งค่า Coarseness

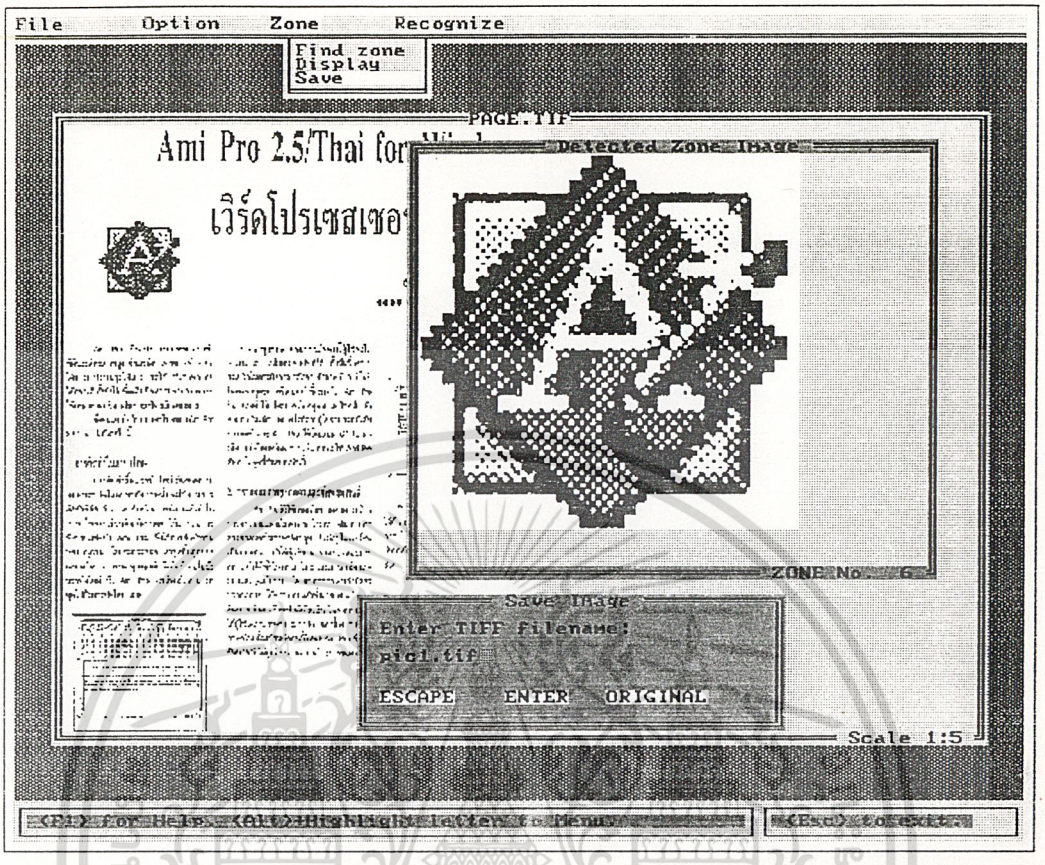
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 4 เลือกเมนู Zone บนพูลดาวน์เมนู แล้วเลือกฟังก์ชั่น Find zone เพื่อทำการคำนวณหาขอบเขตของส่วนประกอบของหน้าเอกสารโดยอาศัยทฤษฎีเซลลูล่าออโตมาตา และเมื่อการคำนวณประมวลเสร็จเรียบร้อย จะสามารถเลือกดูภาพของส่วนประกอบของหน้าเอกสารแต่ละส่วนได้ โดยเลือกฟังก์ชั่น View zone ในพลูดาวน์เมนู Zone ซึ่งจะปรากฏวินโดว์ชื่อ "Detected zone image" ดังแสดงในรูป 5.7 และสำหรับส่วนประกอบของหน้าเอกสารไม่ว่าจะเป็นส่วนที่เป็นข้อความของตัวอักษรหรือส่วนที่เป็นรูปภาพ สามารถจัดเก็บได้โดยเลือกฟังก์ชั่น Save zone ในเมนู Zone ซึ่งจะปรากฏวินโดว์ให้ตั้งชื่อแฟ้มข้อมูลที่ต้องการจัดเก็บ โดยรูปแบบ (Format) ของแฟ้มข้อมูลจะเป็นแฟ้มข้อมูลภาพกราฟิคแบบ TIFF ดังแสดงในรูป 5.8



รูปที่ 5.7 แสดงจำนวนของส่วนประกอบต่างๆ บนหน้าเอกสารที่คำนวณได้

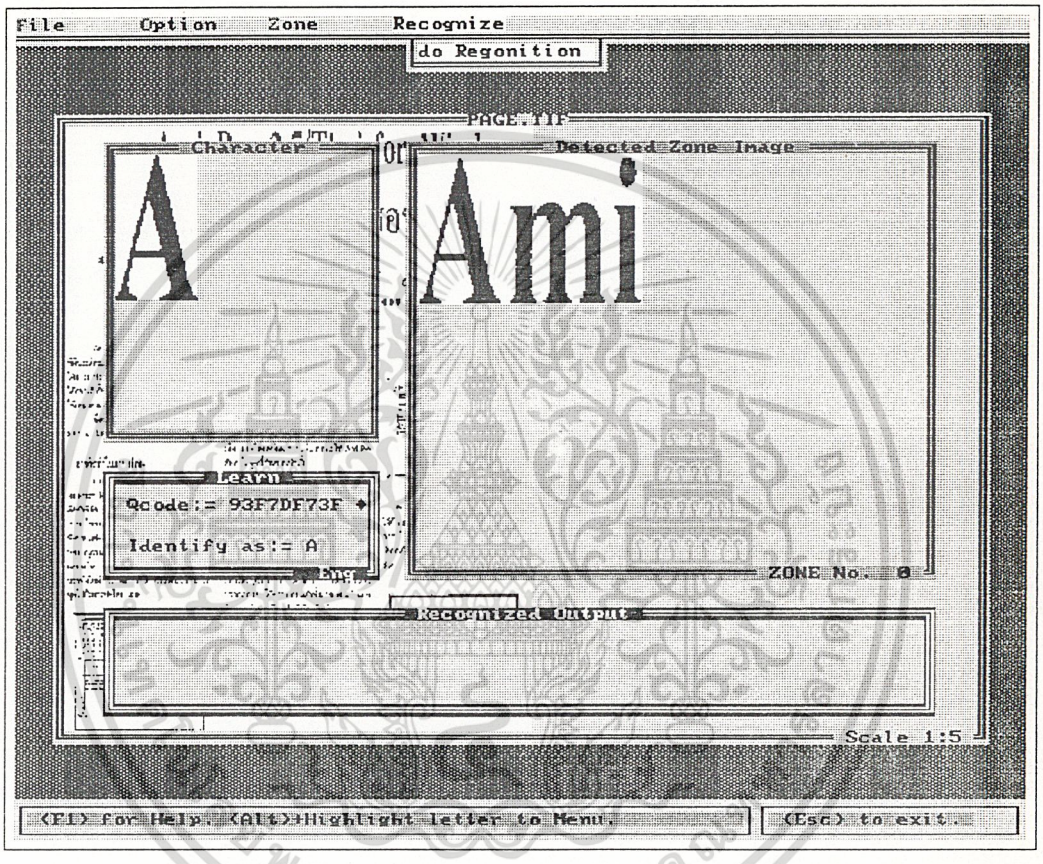
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 แสดงวินโดว์ของการจัดเก็บภาพส่วนประกอบของหน้าเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

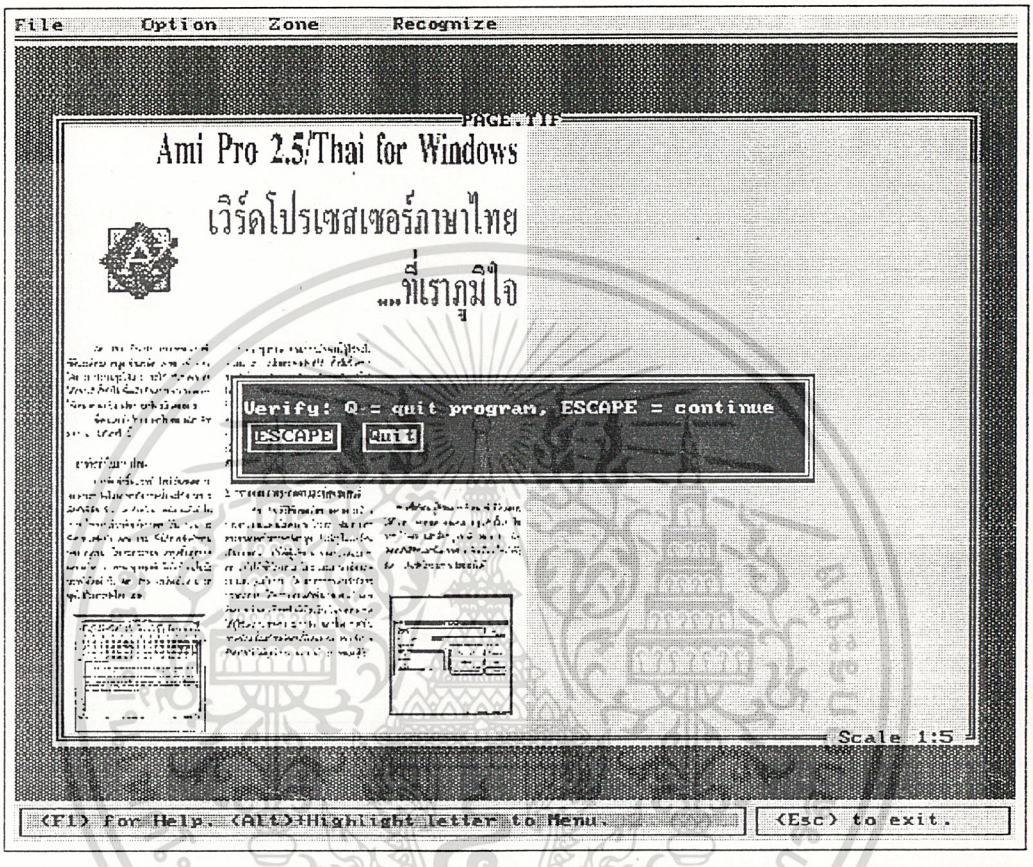
ขั้นตอนที่ 5 เลือกเมนู Recognition เพื่อทำการวิเคราะห์และรู้จำตัวอักษร โปรแกรมจะทำการประมวลผลโดยแยกภาพตัวอักษรแต่ละตัวออกจากภาพประโยคเสียก่อน แล้วคำนวณหารหัสแทนตัวอักษรและจะนำไปค้นหาในพจนานุกรมเพื่อระบุและรู้จำว่าภาพตัวอักษรนั้นเป็นตัวอักษรอะไร ในกรณีไม่ปรากฏว่ามีรหัสแทนตัวอักษรในพจนานุกรมสำหรับการรู้จำ ก็จะปรากฏวินโดว์แสดงรหัสตัวอักษรและรูปภาพตัวอักษรแล้วให้ผู้ใช้ระบุภาพตัวอักษรนั้นเป็นตัวอักษรอะไร ดังแสดงในรูป 5.9



รูปที่ 5.9 แสดงภาพการทำงานในส่วนการรู้จำตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 6 เมื่อสิ้นสุดการทำงานแล้ว ต้องการจะออกจากโปรแกรม สามารถทำได้โดยเลือกเมนู File บนพูลดาวน์เมนูแล้วเลือกฟังก์ชั่น eXit จากนั้นจะปรากฏวินโดว์แสดงข้อความขอคำยืนยันอีกครั้ง หากผู้ใช้ตอบตกลงก็จะออกจากโปรแกรมไปสู่ระบบปฏิบัติการดอสทันที



รูปที่ 5.10 แสดงวินโดว์การสิ้นสุดการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การทดสอบการทำงานของโปรแกรม

การทดสอบการทำงานของโปรแกรมกระทำบนเครื่องไมโครคอมพิวเตอร์ตระกูล IBM ที่ใช้หน่วยประมวลผลกลาง (CPU) 80386 ความเร็วสัญญาณนาฬิกา (CPU clock) 25 MHz และมีหน่วยความจำหลัก(Main memory) ขนาด 4 เมกะไบต์ ทำการทดสอบการทำงานของระบบในหัวข้อต่างๆ ดังต่อไปนี้

<u>การทดสอบที่ 1</u> การวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร

ภาพหน้าเอกสารดังแสดงในรูป 5.11 ถูกตรวจกวาดด้วยเครื่องตรวจกวาดภาพ โดยกำหนดความละเอียดของการตรวจกวาดที่ระดับต่างๆ กัน 3 ระดับคือ 150, 200 และ 300 จุดต่อนิ้ว ทำการทดสอบการทำงานในส่วนของการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร ได้ผลของการทดสอบดังแสดงในตารางที่ 5.1

รูปที่ 5.11 ตัวอย่างหน้าเอกสารที่ใช้ในการทดสอบ

(ขนาดจริงของภาพ 6.94" x 8.25" ความละเอียดของภาพ 200 จุดต่อนิ้ว

ถูกย่อลง 25 % เพื่อการแสดงผล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 แสดงเวลาที่ใช้ในการวิเคราะห์และระบุส่วนประกอบหน้าเอกสาร

Dimension Width x Height	Resolution (dpi)	File size (Bytes)	Time (Minutes)
7.38" x 9.72" Width 1,107 Pixels Height 1,459 Lines	150	199 K.	1.3
7.36" x 6.79" Width 1,473 Pixels Height 1,359 Lines	200	246 K.	1.42
3.47" x 6.77" Width 1,041 Pixels Height 2,031 Lines	300	261 K.	2.05

<u>การทดสอบที่ 2</u> การรู้จำตัวอักษร

ข้อมูลภาพตัวอักษรต้นฉบับในรูปแบบ (Font) ที่แตกต่างกันถูกนำมาทดสอบความสามารถในการรู้จำและวิเคราะห์ตัวอักษรแต่ละตัว โดยแบ่งข้อมูลที่ใช้ทดสอบเป็น 3 กลุ่มคือ กลุ่มที่หนึ่งเป็นข้อมูลตัวอักษรภาษาไทย ซึ่งประกอบด้วยตัวอักษรภาษาไทย 44 ตัว สระและวรรณยุกต์ไทย 20 ตัว จำนวน 9 รูปแบบ กลุ่มที่สองเป็นข้อมูลตัวอักษรภาษาอังกฤษ ซึ่งประกอบด้วยตัวอักษรพิมพ์ใหญ่และตัวอักษรพิมพ์เล็กชุดละ 26 ตัว จำนวน 5 รูปแบบ และกลุ่มที่สามเป็นข้อมูลตัวเลขไทย 10 ตัว จำนวน 9 รูปแบบ และตัวเลขอารบิค 10 ตัว จำนวน 5 รูปแบบ

ตารางที่ 5.2 แสดงผลของการวิเคราะห์ชุดตัวอักษรต้นฉบับภาษาไทยทั้งหมด 9 รูปแบบ และตารางที่ 5.3 แสดงผลการวิเคราะห์ชุดตัวอักษรต้นฉบับภาษาอังกฤษทั้งหมด 5 รูปแบบ เพื่อหารหัสแทนตัวอักษรแต่ละตัวในแต่ละรูปแบบ เก็บไว้เป็นพจนานุกรมสำหรับการรู้จำ ผลการวิเคราะห์ที่ผิดพลาดคือการคำนวณได้รหัสแทนตัวอักษรที่มีค่าเหมือนกัน ซึ่งเกิดจากลักษณะของตัวอักษรที่มีลักษณะคล้ายกันมาก เช่น ตัวอักษร "ค" กับตัวอักษร "ศ" และตัวอักษร "ด" กับตัวอักษร "ต" เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2 แสดงผลของการวิเคราะห์ชุดตัวอักษรต้นฉบับภาษาไทย

ชื่อรูปแบบตัวอักษร	จำนวนตัวอักษร	ผลการวิเคราะห์				
		ถูกต้อง		ผิด		ตัวอักษรที่มีรหัสซ้ำกัน
AngsanaUPC	44	42	95.45%	2	4.55 %	ค,ต ค,ต
BrowalliaUPC	44	44	100 %	0	0 %	
CordiaUPC	44	43	93.18 %	3	6.82 %	ค,ศ ต,ต ฎ,ฏ
DilleniaUPC	44	44	100 %	0	0 %	
EucrosiaUPC	44	43	97.73 %	1	2.27 %	ฎ,ฏ
FreesiaUPC	44	44	100 %	0	0 %	
IrisUPC	44	41	93.18 %	3	6.82 %	ช,ซ ฎ,ฏ ญ,ณ
JasmineUPC	44	37	84.09 %	7	15.91 %	ก,ถ ข,ฃ,ฆ ค,ต,ด,ต,ศ
KodchiangUPC	44	41	93.18 %	3	6.82 %	ก,ถ ฃ,ฆ ช,ซ

ตารางที่ 5.3 แสดงผลการวิเคราะห์ชุดตัวอักษรต้นฉบับภาษาอังกฤษ

ชื่อรูปแบบตัวอักษร	จำนวนตัวอักษร	ผลการวิเคราะห์				
		ถูกต้อง		ผิด		ตัวอักษรที่มีรหัสซ้ำกัน
CourierNew	52	51	96.15 %	1	3.85 %	D,O
Helvetica	52	51	96.15 %	1	3.85 %	D,O
LetterGothic	52	51	96.15 %	1	3.85 %	D,O
NewsGothic	52	51	96.15 %	1	3.85 %	D,O
TimesNewRomanPS	52	50	92.31 %	2	7.69 %	B,E D,O

หลังจากนั้นได้นำข้อมูลภาพตัวอักษรในแต่ละรูปแบบ ที่ได้จากการตรวจกวาดใหม่อีก 20 ครั้ง มาทำการทดสอบประสิทธิภาพของการรู้จำ และผลการทดสอบแสดงอยู่ในตารางที่ 5.4, 5.5, 5.6 และ 5.7 ผลการวิเคราะห์มีตัวแปรหลายตัวที่ส่งผลกระทบกับการรู้จำ เช่น ลายเส้นตัวอักษรขาด แหว่ง หรือหมึกพิมพ์ซึมมากเกินไป ซึ่งทำให้คำนวณได้รหัสแทนตัวอักษรที่มีค่าไม่ตรงกับรหัสของตัวอักษรต้นฉบับ และในที่นี้ได้กำหนดให้เป็นการวิเคราะห์ที่ผิดพลาด ดังแสดงในตารางผลการรู้จำในช่องผลการวิเคราะห์ผิด

ตารางที่ 5.4 แสดงผลการรู้จำตัวอักษรชุดทดสอบตัวอักษรไทย

ชื่อรูปแบบตัวอักษร	จำนวนตัวอักษร	ผลการวิเคราะห์			
		ถูกต้อง		ผิด	
AngsanaUPC	880	777	88.30 %	103	11.70 %
BrowalliaUPC	880	787	89.43 %	93	10.57 %
CordiaUPC	880	822	93.41 %	58	6.59 %
DilleniaUPC	880	787	89.43 %	93	10.57 %
EucrosiaUPC	880	798	90.68 %	82	9.32 %
FreesiaUPC	880	792	90 %	88	10 %
IrisUPC	880	729	82.84 %	151	17.16 %
JasmineUPC	880	829	94.20 %	51	5.8 %
KodchiangUPC	880	781	88.75 %	99	11.25 %

ตารางที่ 5.5 แสดงผลของการรู้จำตัวอักษรชุดทดสอบตัวอักษรอังกฤษ

ชื่อรูปแบบตัวอักษร	จำนวนตัวอักษร	ผลการวิเคราะห์			
		ถูกต้อง		ผิด	
CourierNew	780	666	85.38 %	114	14.62 %
Helvetica	780	735	93.88 %	45	6.12 %
LetterGothic	780	696	89.23 %	84	10.77 %
NewsGothic	780	700	89.74 %	80	10.26 %
TimesNewRomanPS	780	700	89.74 %	80	10.26 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.6 แสดงผลของการรู้จำตัวอักษรชุดทดสอบตัวเลขไทย

ชื่อรูปแบบตัวอักษร	จำนวนตัวอักษร	ผลการวิเคราะห์			
		ถูกต้อง		ผิด	
AngsanaUPC	100	84	84 %	16	16 %
BrowalliaUPC	100	72	72 %	28	28 %
CordiaUPC	100	71	71 %	29	29 %
DilleniaUPC	100	88	88 %	12	12 %
EucrosiaUPC	100	83	83 %	17	17 %
FreesiaUPC	100	84	84 %	16	16 %
IrisUPC	100	80	80 %	20	20 %
JasmineUPC	100	86	86 %	14	14 %
KodchiangUPC	100	84	84 %	16	16 %

ตารางที่ 5.7 แสดงผลของการรู้จำตัวอักษรชุดทดสอบตัวเลขอารบิค

ชื่อรูปแบบตัวอักษร	จำนวนตัวอักษร	ผลการวิเคราะห์			
		ถูกต้อง		ผิด	
CourierNew	100	84	84 %	16	16 %
Helvetica	100	84	84 %	16	16 %
LetterGothic	100	81	81 %	19	19 %
NewsGothic	100	92	92 %	19	8 %
TimesNewRomanPS	100	92	92 %	19	8 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.8 ได้สรุปผลการทำงานในส่วนของการรู้จำตัวอักษร โดยได้สรุปผลของการวิเคราะห์และทดสอบจากข้อมูลที่แสดงอยู่ในตารางที่ 5.2 ถึงตารางที่ 5.7 การวิเคราะห์และกำหนดรหัสแทนตัวอักษรสำหรับตัวอักษรต้นฉบับ ซึ่งประกอบด้วยตัวอักษรภาษาไทย 44 ตัว ตัวเลขไทย 10 ตัว จำนวน 9 รูปแบบ ตัวอักษรภาษาอังกฤษพิมพ์ใหญ่ 26 ตัว ตัวพิมพ์เล็ก 26 ตัว ตัวเลขอารบิค 10 ตัว จำนวน 5 รูปแบบ รวมตัวอักษรต้นแบบทั้งหมด 676 ตัวอักษร และสามารถกำหนดรหัสแทนตัวอักษรได้รหัสที่แตกต่างกันสำหรับตัวอักษรแต่ละตัวได้ถูกต้องรวม 653 ตัว และวิเคราะห์ผิดพลาดโดยคำนวณได้รหัสที่ซ้ำกันรวมทั้งสิ้น 23 ตัว และเมื่อทำการทดสอบการรู้จำโดยใช้รหัสแทนตัวอักษรของตัวอักษรต้นฉบับเป็นพจนานุกรมสำหรับการรู้จำ ทดสอบกับตัวอักษรในแต่ละรูปแบบโดยพิมพ์และตรวจกวาดใหม่ รูปแบบละ 20 ครั้ง ได้ตัวอักษรทดสอบรวมทั้งหมด 13,200 ตัวอักษร ผลการรู้จำปรากฏว่ารู้จำได้ถูกต้องรวม 11,765 ตัว และวิเคราะห์ผิดพลาดรวม 1,435 ตัว โดยพบว่าข้อผิดพลาดของการรู้จำเกิดจากคุณภาพของภาพตัวอักษร เช่น ลายเส้นของตัวอักษรขาดแหว่ง หรือหมึกพิมพ์ซึมมาก ทำให้ลายเส้นไม่สมบูรณ์และผิดเพี้ยนไป ทำให้คำนวณได้รหัสแทนตัวอักษรนั้นๆ เป็นรหัสใหม่ซึ่งไม่เคยปรากฏในพจนานุกรมสำหรับการรู้จำ

ตารางที่ 5.8 แสดงผลสรุปการรู้จำตัวอักษร

	ชุดตัวอักษรต้นฉบับ		ชุดตัวอักษรทดสอบ	
จำนวนตัวอักษรทั้งหมด	676		13,200	
	ผลการวิเคราะห์		ผลการรู้จำ	
	ถูกต้อง	ผิด	ถูกต้อง	ผิด
	653	23	11,765	1,435
เฉลี่ย	96.60 %	3.4 %	89.13 %	10.87 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุป

6.1 สรุปการทำงานของระบบ

งานวิจัยนี้ประกอบด้วยการทำงานต่างๆ หลายขั้นตอน ซึ่งสามารถสรุปการทำงานในแต่ละขั้นตอนได้เป็นหัวข้อดังต่อไปนี้

6.1.1 ข้อมูลอินพุทกับการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร

ข้อมูลที่เป็นอินพุทให้กับระบบของงานวิจัยนี้ นับเป็นปัญหาใหญ่อันดับแรกของงานวิจัยนี้ ทั้งนี้เนื่องจากอินพุทของระบบนั่นก็คือภาพหน้าเอกสาร 1 หน้ากระดาษ ซึ่งโดยทั่วๆ ไปนิยมใช้กระดาษขนาด A4 ซึ่งมีขนาดกว้างประมาณ 8 นิ้วและยาวประมาณ 11 นิ้ว การจัดเก็บภาพหน้าเอกสารให้อยู่ในรูปแบบที่เครื่องคอมพิวเตอร์สามารถนำไปประมวลผลได้ทำได้โดยการตรวจกวาดผ่านเครื่องตรวจกวาดภาพ ภาพหน้าเอกสารก็จะถูกแปลงให้อยู่ในรูปข้อมูลไบนารี่ ซึ่งหนึ่งไบท์ของข้อมูลก็คือหนึ่งจุดภาพหรือพิกเซล (Pixel) ของภาพหน้าเอกสารนั่นเอง ภาพหน้าเอกสารบนกระดาษขนาด A4 เมื่อตรวจกวาดมาด้วยความละเอียดของการตรวจกวาด 200 จุดต่อนิ้ว ก็จะประกอบด้วยจุดภาพซึ่งเป็นความกว้างของภาพเท่ากับ 1,600 จุด และความสูงของภาพเท่ากับ 2,200 จุด ทำให้ภาพหน้าเอกสารนี้ประกอบด้วยจุดภาพทั้งสิ้น 3,520,000 จุด ซึ่งนั่นก็คือข้อมูลไบนารี่ขนาด 3,520,000 ไบท์ด้วยเช่นกัน

การประมวลผลภาพเพื่อวิเคราะห์และระบุส่วนประกอบที่มีอยู่บนหน้าเอกสารโดยอาศัยทฤษฎีเซลลูล่าออโตมาตา เป็นวิธีการประมวลผลภาพโดยพิจารณาจุดภาพแต่ละจุดเปรียบเทียบกับจุดที่อยู่ล้อมรอบ จึงเกิดการคำนวณประมวลผลซ้ำๆ กันเป็นจำนวนมาก และเสียเวลาประมวลผลมากด้วยเช่นกัน ยิ่งข้อมูลขนาดใหญ่มากเท่าไรก็ยิ่งต้องใช้เวลาของการประมวลผลมากขึ้นเท่านั้น แต่ว่าผลลัพธ์ของวิธีการที่เลือกนำมาใช้นี้ ให้ความถูกต้องแม่นยำสูงมาก และการแก้ปัญหาในเรื่องของเวลาที่ใช้ในการประมวลสามารถทำได้ ด้วยการเพิ่มประสิทธิภาพทางฮาดแวร์ให้กับเครื่องคอมพิวเตอร์ ไม่ว่าจะเป็นการเพิ่มหน่วยความจำ (RAM) หรือเลือกใช้หน่วยประมวลผลกลาง (CPU) ที่มีความเร็วของการประมวลผลที่เร็วขึ้น

6.1.2 ข้อมูลภาพตัวอักษรกับการรู้จำ

คุณภาพของข้อมูลภาพตัวอักษรนับเป็นปัญหาอีกประการหนึ่งของงานวิจัยนี้ เทคนิคที่เลือกใช้สำหรับการวิเคราะห์และรู้จำตัวอักษร โดยการพิจารณาคุณสมบัติทางโทโพโลยี่ของลายเส้นของตัวอักษร เป็นวิธีการที่ไม่ยุ่งยากและซับซ้อนจนเกินไป มีการคำนวณประมวลผลไม่มากนักสามารถทำการวิเคราะห์และรู้จำได้อย่างรวดเร็ว และให้ความถูกต้องแม่นยำสูง ด้วยคุณภาพของข้อมูลภาพตัวอักษรต้นฉบับที่สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชัดเจน ประสิทธิภาพของการรู้จำสูงเฉลี่ยถึง 96.60 % ในส่วนของการผิดพลาดของการรู้จำเกิดจากตัวอักษรที่มีลักษณะคล้ายกันมาก เช่น "ค" กับ "ศ" หรือ "ด" กับ "ต" และ "D" กับ "O" เป็นต้น และถ้าคุณภาพของข้อมูลภาพมีลักษณะไม่สมบูรณ์ เช่น ลายเส้นขาด แหว่ง หรือติดกัน จะทำให้การวิเคราะห์และรู้จำไม่สามารถทำได้ แต่ในส่วนนี้ได้แก้ไขด้วยการเพิ่มเติมส่วนติดต่อกับผู้ใช้เข้าไปในโปรแกรมการทำงานของระบบเพื่อให้ผู้ใช้ระบุว่าตัวอักษรนั้นเป็นตัวอักษรอะไร

อีกกรณีหนึ่งที่จะทำให้การรู้จำตัวอักษรผิดพลาดไปได้ก็คือ ภาพหน้าเอกสารเอียงซึ่งเกิดจากการวางกระดาษบนเครื่องตรวจกวาดภาพไม่ตรงเมื่อทำการตรวจกวาดภาพหน้าเอกสาร กรณีดังกล่าวพบว่าถ้าหน้าเอกสารเอียงไปด้านใดด้านหนึ่งไม่ว่าจะเป็นด้านซ้ายหรือขวาประมาณ 1 องศาจากแนวนอน ผู้ใช้อาจจะยังไม่สังเกตุเห็นว่าภาพหน้าเอกสารนั้นเอียงไป และเมื่อทำการรู้จำตัวอักษรก็พบว่าผลของการรู้จำผิดพลาดไปไม่มากนัก (ทำการทดสอบกับประโยคยาวประมาณ 3 นิ้วมีตัวอักษรทั้งหมด 22 ตัว ผลการรู้จำผิดพลาดไป 3 ตัว หรือผิดพลาด 13.64 %) แต่ถ้าภาพหน้าเอกสารเอียงไปมากกว่า 2 องศา ผู้ใช้จะสามารถมองเห็นว่าภาพนั้นเอียงไปได้อย่างชัดเจน กรณีภาพหน้าเอกสารยิ่งมีขนาดใหญ่มากขึ้นเท่าไร ก็จะยิ่งเห็นว่าภาพนั้นเอียงไปชัดเจนมากขึ้นด้วยแม้ว่ามุมที่เอียงไปจะเอียงไปเพียงเล็กน้อยก็ตาม และเมื่อทดลองทำการรู้จำต่อไป ก็พบว่าผลของการรู้จำผิดพลาดสูงขึ้นมาก (ผิดพลาดประมาณ 75 %) การรู้จำที่ผิดพลาดก็คือการคำนวณได้รหัส Q-code แทนภาพตัวอักษรค่าที่แตกต่างไปไม่ถูกต้องตามรูปแบบของตัวอักษรนั้นๆ ซึ่งไม่ควรที่จะทำการรู้จำเพราะรหัสแทนตัวอักษรที่ผิดพลาดนั้นจะถูกเพิ่มเติมลงในพจนานุกรม ซึ่งอาจทำให้การรู้จำตัวอักษรอื่นต่อไปผิดพลาดไปด้วย ดังนั้นแนวทางแก้ไขสามารถทำได้โดยจัดวางหน้ากระดาษเสียใหม่ให้ตรง แล้วทำการตรวจกวาดใหม่อีกครั้ง

6.2 ข้อจำกัดของการทำงานของระบบ

6.2.1 ข้อจำกัดของการทำงานในส่วนการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร

เนื่องจากการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสารเป็นการประมวลผลภาพหน้าเอกสารทั้งหน้า ขนาดมิติของหน้าเอกสารและความละเอียดของจุดภาพเป็นตัวแปรที่สำคัญ 2 ตัวที่กำหนดขนาดของข้อมูลภาพ ขนาดมิติที่ใหญ่จะทำให้ข้อมูลภาพมีขนาดใหญ่ และความละเอียดของจุดภาพที่มีค่ามากก็ทำให้ข้อมูลภาพมีขนาดใหญ่มากด้วยเช่นกัน ดังนั้นการทำงานในขั้นตอนนี้จึงถูกจำกัดด้วยขนาดของหน่วยความจำของเครื่องคอมพิวเตอร์ ขนาดของข้อมูลภาพหน้าเอกสารซึ่งเป็นข้อมูลอินพุทของระบบจะต้องมีขนาดไม่ใหญ่เกินกว่าขนาดหน่วยความจำของเครื่องคอมพิวเตอร์ที่ใช้ ยกตัวอย่างเช่น ถ้าเครื่องไมโครคอมพิวเตอร์ที่ใช้มีหน่วยความจำ (RAM) ขนาด 4 เมกะไบต์ ในส่วนของระบบปฏิบัติการของเครื่องคอมพิวเตอร์และโปรแกรมการทำงานของระบบต้องการหน่วยความจำคิดรวมประมาณ 1 เมกะไบต์ ดังนั้นขนาดของข้อมูลภาพจะต้องมีขนาดไม่ใหญ่เกินกว่า 3 เมกะไบต์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2 ข้อจำกัดของการทำงานในส่วนของการรู้จำตัวอักษร

เนื่องจากเทคนิคและวิธีการที่เลือกใช้ในการรู้จำตัวอักษรเป็นการแนะนำวิธีการใหม่ที่ดัดแปลงจากวิธีการของงานวิจัยที่ผ่านมา วิธีการนี้ยังมีข้อจำกัดของการทำงาน เช่น คุณภาพของภาพตัวอักษรที่จะทำการรู้จำจะต้องมีคุณภาพที่ดี ลายเส้นของจะต้องสมบูรณ์ไม่ขาดแหว่ง หรือลายเส้นของภาพตัวอักษรต้องไม่ผิดเพี้ยนเนื่องจากหมึกพิมพ์ที่ซึมเปื้อนมากเกินไป รูปแบบตัวอักษรก็ต้องเป็นรูปแบบเดียวกันที่ได้คำนวณหารหัสแทนตัวอักษรเก็บไว้เป็นพจนานุกรมสำหรับการรู้จำแล้ว ไม่ว่าจะเป็นขนาดเล็กใหญ่ก็ดี ลักษณะตัวปกติหรือตัวเอียงก็ดี อย่างไรก็ตาม การเตรียมข้อมูลที่ดีเป็นไปตามเงื่อนไขที่กำหนดดังเช่นที่กล่าวมาข้างต้น ยังพอจะให้ประสิทธิภาพของการรู้จำที่สามารถยอมรับได้

6.3 ข้อเสนอแนะสำหรับแนวทางพัฒนาระบบในอนาคต

งานวิจัยนี้เน้นการวิเคราะห์และระบุส่วนประกอบของหน้าเอกสาร ซึ่งสามารถนำไปใช้งานได้จริง แต่ว่าการระบุว่าส่วนประกอบของหน้าเอกสารว่าส่วนใดเป็นส่วนของรูปภาพ หรือส่วนใดเป็นส่วนของข้อความยังไม่สามารถระบุได้เองโดยอัตโนมัติ ยังคงต้องอาศัยผู้ใช้เป็นผู้ตัดสินใจเลือก อีกทั้งข้อมูลภาพที่นำมาใช้ในการวิเคราะห์และประมวลผลมีลักษณะเป็นข้อมูลภาพขาวดำเท่านั้น หากได้มีการเพิ่มเติมอัลกอริทึมสำหรับการระบุส่วนประกอบของหน้าเอกสารโดยอัตโนมัติ และการจัดการกับข้อมูลภาพสีได้ ก็จะเป็นการเพิ่มประสิทธิภาพการทำงานในส่วนนี้มากยิ่งขึ้น และในส่วนของการรู้จำตัวอักษร การเพิ่มเติมความสามารถในการย่อและขยาย การปรับหมุนภาพ และการกำจัดส่วนรบกวน (Noise) ของภาพตัวอักษร จะทำให้การรู้จำมีความถูกต้องแม่นยำสูงมากขึ้น ผู้วิจัยคาดหวังไว้ว่า แม้งานวิจัยชิ้นนี้จะไม่สามารถทำงานครอบคลุมในทุกสภาวะแวดล้อมของการทำงานได้ แต่ก็น่าจะเป็นแนวทางหรือจุดเริ่มต้นให้กับการพัฒนาระบบสำหรับนักวิจัยรุ่นต่อๆ ไป ในการค้นคิดหาวิธีการและเทคนิคเพื่อพัฒนาระบบให้สามารถทำงานได้ดีมีประสิทธิภาพยิ่งขึ้น ซึ่งจะสามารถนำไปใช้งานได้อย่างแพร่หลายต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ผศ. ชม กิ้มปาน, "ทฤษฎีการจดจำรูปแบบเบื้องต้น", ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าวิทยาเขตเจ้าคุณทหารลาดกระบัง กรุงเทพฯ, พิมพ์ครั้งที่ 1, กรกฎาคม 2525

2.KIMPAN, C., A. ITOH and PROF. K. KAWANISHI, "Recognition of Printed Thai Characters Using a Matching Methods", IEE Proceeding, Vol. 130, Pt. E, No. 6, November 1983.

3. KIMPAN, C., "Printed Thai Characters Recognition", Dissertation of Doctor of Engineering, King Mogkut's Institute of Technology Chaokhuntaharn Ladkrabang, 1986.

4. KIMPAN, C., A. ITOH and PROF. K. KAWANISHI, "Fine Classification of Printed Thai Character Recogniton Using the Karhunen-love Expansion", IEE Proceeding, Vol.134, Pt. E, No. 5, September 1987.

5. KIMPAN, C., "Printed Thai Characters Recognition Using Topological Properties Method", INT.J.Electronics, 1986, Vol. 60, No. 3, pp. 303-329.

6. ประสาร ดังติสานนท์, "การจดจำรูปแบบตัวอักษรคัดลายมือภาษาไทย โดยวิธีแยกลักษณะเด่น", วิทยานิพนธ์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2529

7. สุรพันธ์ เอื้อไพบูลย์, "การจดจำอักษรลายมือเขียนภาษาไทยโดยการพิจารณาหัวของตัวอักษร", วิทยานิพนธ์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2531

8. ชาย เกษมอมรกุล, "การออกแบบพจนานุกรมสำหรับการเรียนรู้อักษรลายมือไทย-อังกฤษอัตโนมัติบนเครื่องไมโครคอมพิวเตอร์", วิทยานิพนธ์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2532

9. สุรสิทธิ์ ราตรี, "การรู้จำอักษรตัวพิมพ์ภาษาไทยโดยวิธีค้นหาลักษณะโครงสร้างลายเส้น", วิทยานิพนธ์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2533

10. SOMSAK WALAIRACHT, CHAI KASEMAMORNKUL and DR.CHOM KIMPAN, "Handprinted Thai Character Recognition by Feature Concentration Method", Computer Proceeding of Asian Language (CPAL), Asian Institute of Technology, September 1989.

11. อนุชิต จารุวนารัตน์, สุรสิทธิ์ ราตรี และรศ. ดร. ชม กิ้มปาน, "การแยกภาพตัวอักษรภาษาไทยออกจากภาพประโยค", การประชุมวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 10, คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, พฤษภาคม 2530

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. สมศักดิ์ วลัยรัชต์, สุรสิทธิ์ ราตรี และรศ. ดร. ชม กิ้มปาน, "การจดจำรูปแบบอักษรพิมพ์ภาษาไทย", บทความประชุมวิชาการและนิทรรศการ "การประยุกต์ใช้คอมพิวเตอร์ในทางวิศวกรรม", มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่, กุมภาพันธ์ 2531

13. สมศักดิ์ วลัยรัชต์ และรศ. ดร. ชม กิ้มปาน, "การรู้จำอักษรลายมือเขียนภาษาไทยของไมโครคอมพิวเตอร์โดยการพิจารณาลักษณะลายเส้นขณะที่ลากไป", การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 13, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่, พฤศจิกายน 2533

14. สมศักดิ์ วลัยรัชต์ และรศ. ดร. ชม กิ้มปาน, "การแยกและจัดกลุ่มตัวอักษรตัวพิมพ์หลายรูปแบบ", การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 14, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่, พฤศจิกายน 2534

15. YASUSHIRO YAMADA, TAKAHIKO KAWATANI, HIROSHI KANEKO and TOSHIO TSUTSUMIDA, "Handprinted Numerical Recognition by Feature Concentration Method", Review of the Electrical Communication Laboratories, Vol. 26, No. 11-12, November-December, 1978.

16. DAVID WEBER, "Detecting Text Regions Using Cellular Automata", The C Users Journal, August 1993

17. ALASTAIR D. McAULAY, "Optical Computer Architectures, The Application of Optical Concepts to Next Generation Computers", A Wiley-Interscience Publication, John Wiley & Sons, Inc.

18. ศิวัฒน์ ศิวะบวร, "Cellular Automata กับปัญหาเขาวงกต", คอมพิวเตอร์รีวิว ฉบับที่ 108 ประจำเดือนสิงหาคม 2536

19. RICHAD O. DUDA, PETER E. HART, "Pattern Classification and Scene Analysis", Stanford Research Institute, Menlo Park, California.

20. HEWLETT-PACKARD, "A Guide to the Tag Image File Format Version 5.0", Hewett-Packard Company, August 1991.

21. HEWLETT-PACKARD, "ScanJet User's Guide", Hewett-Packard Company, April 1987.

22. HEWLETT-PACKARD, "X Window System Programming: Xlib", Hewett-Packard Company, October 1989.

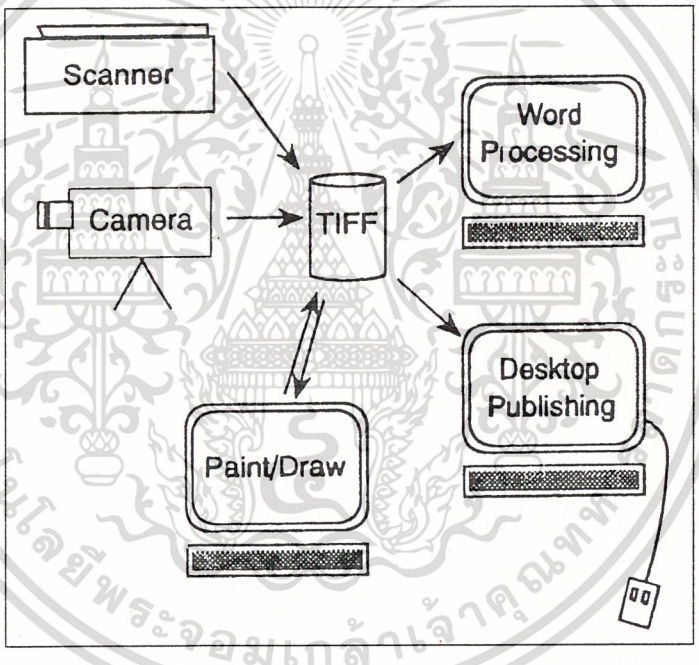
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

แฟ้มข้อมูลกราฟิกรูปแบบ TIFF Version 5.0

ภาพรวมของแฟ้มข้อมูลกราฟิกรูปแบบ TIFF

แฟ้มข้อมูล Tag Image File Format หรือเรียกสั้นๆ ว่า TIFF ถูกพัฒนาขึ้นโดยบริษัท Aldus Corporation เพื่อใช้จัดเก็บข้อมูลภาพแบบ Bit-map เราพบว่าโปรแกรมด้าน Desktop-publishing เกือบทั้งหมด รวมถึงโปรแกรมสำหรับสแกนเนอร์ต่างสนับสนุน TIFF ทั้งนั้น ความสามารถในการจัดเก็บรายละเอียดทั้งหมดของภาพไว้ได้ ทำให้ TIFF เป็นที่นิยมและถูกพัฒนาต่อมาขึ้นเป็นซูเปอร์เซ็ตของรูปแบบแฟ้มข้อมูลอื่นๆ ที่ใช้เก็บภาพ



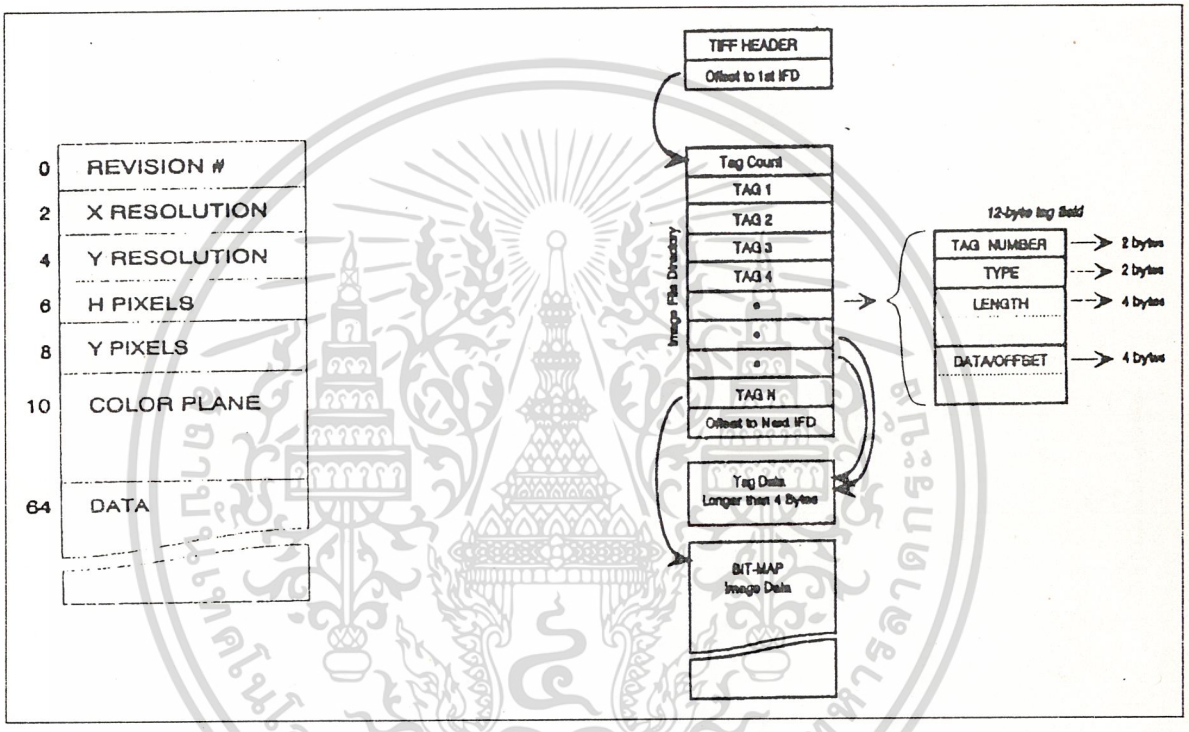
รูปที่ 1 ผังงานของ Scanned images ในงาน Desktop Publishing

ข้อแตกต่างหนึ่งที่ TIFF ต่างจากแฟ้มข้อมูลแบบอื่นๆ คือข้อมูลส่วนใหญ่ที่เก็บอยู่จะไม่ถูกกำหนดตำแหน่งแน่นอนตายตัว โดยมีเพียงข้อมูล 8 ไบต์แรกเท่านั้นที่เป็นตำแหน่งพิเศษและเก็บไว้ที่ตำแหน่งนี้เสมอ ส่วนอื่นๆ ที่เหลือจะถูกอ้างอิงกับส่วนเริ่มต้นของแฟ้มข้อมูล จึงเป็นการง่ายต่อการเพิ่มเติมข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดิมทีแล้วแฟ้มข้อมูลของโปรแกรมวาดภาพทั่วๆ ไป ใช้รูปแบบแฟ้มข้อมูลแบบแน่นอน (Fixed-format organization) ซึ่งเราสามารถทราบโครงสร้างและตำแหน่งของข้อมูลได้ก่อนการเปิดอ่านแฟ้มข้อมูล ยกตัวอย่าง เช่น X resolution จะเก็บอยู่ตำแหน่งไบท์ที่ 2 และ 3 และส่วนของข้อมูลจะเริ่มต้นที่ตำแหน่งไบท์ที่ 64 เป็นต้น

สำหรับแฟ้มข้อมูล TIFF ซึ่งใช้เทคนิคที่แตกต่างกัน โดยแบ่งเป็น Tags เก็บรายละเอียดของข้อมูล ในแต่ละ Tag จะมีความยาว 12 ไบท์อธิบายลักษณะแต่ละส่วนของภาพ ตัวอย่างเช่น ในแฟ้มข้อมูลจะประกอบด้วย Tag สำหรับ X and Y resolution และ Tag สำหรับตำแหน่งของข้อมูล (StripOffsets)



รูปที่ 2 ตัวอย่าง Fixed-file format รูปที่ 3 ตัวอย่างโครงสร้างทั่วไปของ TIFF

หากพิจารณาถึงโครงสร้างของแฟ้มข้อมูล TIFF พบมีการจัดเก็บรายละเอียดหลายอย่างที่ยังไม่มีความจำเป็น แต่ในทางกลับกันพบว่ากลายเป็นข้อได้เปรียบกว่าแฟ้มข้อมูลแบบตายตัว โดยเมื่อต้องการเพิ่มเติม Tag ใหม่ ไม่จำเป็นต้องเขียนโปรแกรมในส่วนจัดเก็บใหม่ ตัวอย่างเช่น ถ้าเพิ่มเติม Tag ที่เก็บ DateTime ในขณะที่อีกโปรแกรมหนึ่งไม่ Support ข้อมูลของ Tag DateTime โอกาสที่จะเกิดข้อผิดพลาดก็จะไม่เกิดขึ้นในกรณีที่มี Non-support tag อยู่ในแฟ้มข้อมูล ต่างกับแฟ้มข้อมูลแบบตายตัวเมื่อเพิ่มเติมข้อมูลเข้าไปจะต้องเขียนโปรแกรมส่วนจัดการใหม่ และเมื่อโปรแกรมที่ใช้งานที่เป็นโปรแกรมรุ่นเก่า ซึ่งต้องการข้อมูลบางอย่างตรงตำแหน่งที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

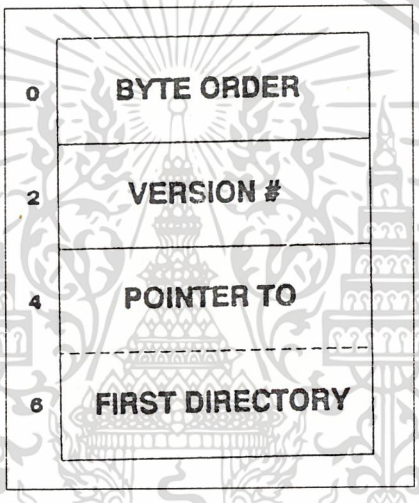
กำหนด กลับพบว่าข้อมูลในตำแหน่งนั้นเปลี่ยนไปหรือหาข้อมูลไม่พบเลยก็ได้ ผลลัพธ์ที่ได้ก็คือการทำงานที่ผิดพลาด

โครงสร้างทั่วไปของ TIFF

แฟ้มข้อมูลกราฟิก TIFF หนึ่งๆ จะประกอบด้วยส่วนประกอบหลัก 3 ส่วนดังต่อไปนี้

1. Header Field

ในส่วนหัวนี้จะมีความยาว 8 ไบท์ ประกอบด้วยข้อมูลดังแสดงในรูปที่ 4



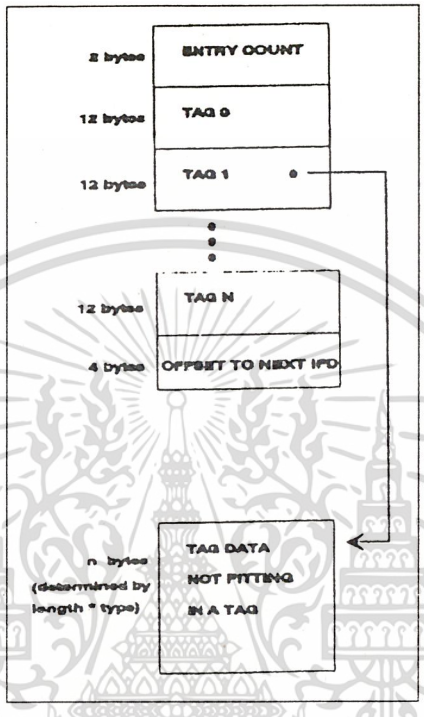
รูปที่ 4 The TIFF Header

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. The Image File Directory (IFD)

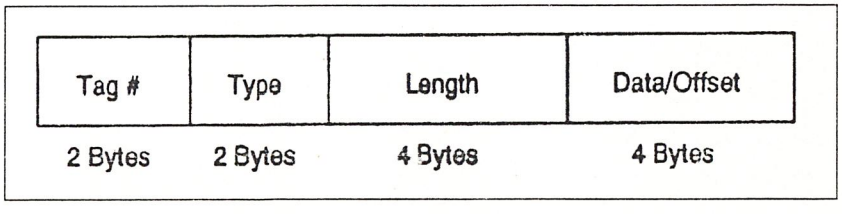
ในหนึ่ง IFD จะประกอบด้วย 3 ส่วนคือ Entry count, Tag ต่างๆ แต่ละอันยาว 12 ไบท์เรียงต่อกัน และส่วนที่จะชี้ไปยังตำแหน่งของ IFD ชุดต่อไป ส่วนนี้ยาว 4 ไบท์ แสดงในรูปที่ 5



รูปที่ 5 Image File Directoy

3.The Tags Field

สามารถที่ Tag ที่แตกต่างกันได้ถึง 50 Tags โดยแต่ละ Tag จะมีความยาว 12 ไบท์ ซึ่งจะประกอบด้วย 4 ส่วนคือ Tag number, Type of tag value, Number of values in tag และ Tag value of offset



รูปที่ 6 The 12 bytes of a TIFF tag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง Tag ที่สำคัญและคำอธิบาย

1. The ImageWidth Tag

ImageWidth Tag = 256(100H)

Type = SHORT or LONG

Number of distinct values = 1

Value range : All positive integers in Type

ImageWidth บอกความกว้างของรูปภาพมีหน่วยเป็น Pixels ความกว้างจะยึดถือตามแกน X หรือตามแนวนอน ความกว้างของภาพอาจเรียกอีกอย่างว่าเป็นจำนวนคอลัมน์ในรูปภาพ Tag นี้เป็น Tag ที่สำคัญเพราะจะไม่สามารถอ่านรูปภาพได้เลยหากไม่ทราบค่าความกว้างของรูปเสียก่อน

2. The ImageLength Tag

ImageLength Tag = 257 (101H)

Type = SHORT or LONG

Number of distinct values = 1

Value range : All positive interger in Type

ImageLength บอกความยาวของรูปภาพมีหน่วยเป็น Pixels ความยาวจะยึดถือตามแกน Y หรือตามแนวตั้ง ความยาวของรูปภาพอาจเรียกอีกอย่างว่าเป็นจำนวนแถวในรูปภาพ สำหรับ Tag นี้ก็เป็น Tag ที่สำคัญเช่นเดียวกัน เพราะจะไม่สามารถอ่านรูปภาพได้เลย หากไม่ทราบความยาวของรูป

3. The StripOffsets Tag

StripOffsets Tag = 273 (111H)

Type = SHORT or LONG

Number of distinct values

= one per strip for single plane bitmaps

= number of planes x strips/plane for multi-plane bitmaps

Vulue range : All positive integers in Type

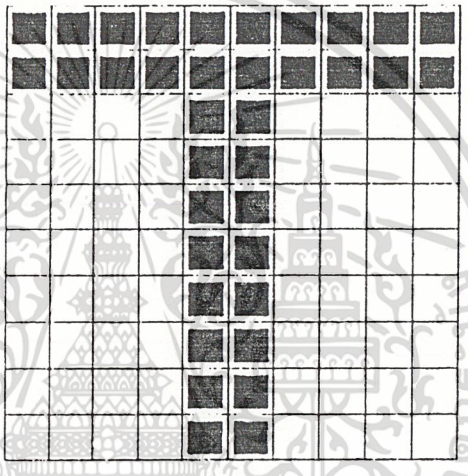
StripOffsets เป็น Tag ที่สำคัญและจำเป็นต้องทราบค่า เพราะ Tag นี้จะบอกตำแหน่งว่าส่วนของรูปภาพเริ่มต้นที่ตำแหน่งใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเปิดอ่านแฟ้มข้อมูลกราฟิก TIFF นี้ ผู้ใช้จะต้องทราบค่าที่สำคัญอย่างน้อย 3 ค่าคือ ความกว้าง ความยาว และตำแหน่งของข้อมูลภาพที่เก็บในแฟ้มข้อมูลชนิดนี้ ซึ่งก็คือค่าต่างๆ ที่เก็บอยู่ใน Tag ทั้ง 3 ที่กล่าวมาแล้วข้างต้นนั่นเอง

ตัวอย่างแฟ้มข้อมูลแบบ TIFF

An Intel format 10 x 10 pixel Black and White TIFF Image



300 DPI Image
BINARY (Line art) image
No data compression
INTEL byte order

Used	Tag #	Tag Name	Value
__x__	0FEH	NewSubfileType	__1___
__x__	100H	ImageWidth	__10__
__x__	101H	ImageLength	__10__
__x__	102H	BitsPerSample	__1___
__x__	103H	Compression	__1___
__x__	106H	PhotometricInterp.	__0___
__x__	107H	Thresholding	__1___
__x__	111H	*StripOffsets*	_@198
__x__	115H	SamplesPerPixel	__1___
__x__	116H	RowsPerStrip	__10__
__x__	117H	StripByteCounts	__20__
__x__	11AH	XResolution	_300__
__x__	11BH	YResolution	_300__
__x__	128H	ResolutionUnit	__2___

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I. Header

Offset	Value	
000	49 49	Note byte order is INTEL format (49 is ASCII 'I').
002	2A 00	Version #42; Note byte reversal.
004	08 00 00 00	Offset to first IFD.

II. Image File Directory

008 0E 00 14 tags in this directory

	Offset	Tag	Type	Length	Value/Offset	
Tag 1	00A	FE 00	04 00	01 00 00 00	00 00 00 00	(NewSubfileType = Full Res)
Tag 2	016	00 01	03 00	01 00 00 00	0A 00 00 00	(Image Width = 10 pixels)
Tag 3	022	01 01	03 00	01 00 00 00	0A 00 00 00	(Image Length = 10 pixels)
Tag 4	02E	02 01	03 00	01 00 00 00	01 00 00 00	(BitsPerSample = 1)
Tag 5	03A	03 01	03 00	01 00 00 00	01 00 00 00	(No compression, pack data)
Tag 6	046	06 01	03 00	01 00 00 00	00 00 00 00	(Black is 1)
Tag 7	052	07 01	03 00	01 00 00 00	01 00 00 00	(Thresholding type = Line Art)
Tag 8	05E	11 01	04 00	01 00 00 00	C6 00 00 00	(Offset to raster data)
Tag 9	06A	15 01	03 00	01 00 00 00	01 00 00 00	(1 sample/pixel)
Tag 10	076	16 01	04 00	01 00 00 00	0A 00 00 00	(10 Rows per Strip)
Tag 11	082	17 01	04 00	01 00 00 00	14 00 00 00	(20 Bytes per Strip)
Tag 12	08E	1A 01	05 00	01 00 00 00	B6 00 00 00	(Offset to X resolution)
Tag 13	09A	1B 01	05 00	01 00 00 00	BE 00 00 00	(Offset to Y resolution)
Tag 14	0A6	28 01	03 00	01 00 00 00	02 00 00 00	(Resolution units are inches)

0B2 00 00 00 00 (No more Image File Directories)

III. Extended tag information (as needed)

0B6 2C 01 00 00 01 00 00 00 (300 dpi X resolution)
0BE 2C 01 00 00 01 00 00 00 (300 dpi Y resolution)

IV. Raster data (as needed)

0C6	FF C0	(10 black pixels)
0C8	FF C0	(10 black pixels)
0CA	0C 00	(2 black pixels)
0CC	0C 00	(2 black pixels)
0CE	0C 00	(2 black pixels)
0D0	0C 00	(2 black pixels)
0D2	0C 00	(2 black pixels)
0D4	0C 00	(2 black pixels)
0D6	0C 00	(2 black pixels)
0D8	0C 00	(2 black pixels)

Strip offset pointers (as needed)
None needed.

Strip byte counts (as needed)
None needed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ตารางรหัสแทนอักขระ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	SP	0	@	P	`	p				ฐ	ภ	ะ	เ	๐
1	SOH	DC1	!	1	A	Q	a	q			ก	ฑ	ม	ั	แ	๑
2	STX	DC2	"	2	B	R	b	r			ข	ฒ	ย	า	โ	๒
3	ETX	DC3	#	3	C	S	c	s			ฃ	ณ	ร	ำ	ใ	๓
4	EOT	DC4	$	4	D	T	d	t			ค	ด	ฤ	ิ	ไ	๔
5	ENQ	NAK	%	5	E	U	e	u			ฅ	ต	ล	ี	ๅ	๕
6	ACK	SYN	&	6	F	V	f	v			ฆ	ถ	ฦ	ึ	ๆ	๖
7	BEL	ETB	'	7	G	W	g	w			ง	ท	ว	ื	็	๗
8	BS	CAN	(	8	H	X	h	x			จ	ธ	ศ	ุ	่	๘
9	HT	EM	)	9	I	Y	i	y			ฉ	น	ษ	ู	้	๙
A	LF	SUB	*	:	J	Z	j	z			ช	บ	ส		๊	
B	VT	ESC	+	;	K	[	k	{			ซ	ป	ห		๋	
C	FF	FS	,	<	L	\	l	|		ฌ	ผ	ฬ		์	
D	CR	GS	-	=	M	]	m	}			ญ	ฝ	อ			
E	SO	RS	.	>	N	^	n	~			ฎ	พ	ฮ			
F	SI	US	/	?	O	_	o	DEL			ฏ	ฟ	ฯ			

หมายเหตุ

ตารางรหัสภาษาอังกฤษใช้รหัสตามมาตราฐาน ISO 646-1983

ตารางรหัสภาษาไทยใช้รหัสมาตราฐานอุตสาหกรรม มอก. 620-2533

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

รูปแบบตัวอักษรและค่า Q-code

1. Font : AngsanaUPC

กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถ

ทธนบปผฝพฟภมยรลวศษสหฬอฮ

ฯ ั า ิ ี ึ ื ุ ู เ แ โ ใ ไ ๅ ็ ่ ้ ๊ ๋ ์ ๎ ๏ ๆ ๗ ๘ ๙ ๚

๑๒๓๔๕๖๗๘๙๐

ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*
ก	7CE68A6CE	ข	EAF8BBEBB	ฃ	FAFABBFBB	ค	FCF68ECEE
ฅ	FCF68ECEE	ฆ	FA3DBEFBB	ง	8FFCFB3DF	จ	CCFCBBCFF
ฉ	7CE83E7DF	ช	F6F9BBFFF	ซ	FBF9BFFFF	ฌ	7AEABF7BF
ญ	7AB8BE7BE	ฎ	6CE5BF7DF	ฏ	6CED9F7DF	ฐ	77FCBBFFF
ฑ	EEA8AEFEB	ฒ	FAE6BFFBF	ณ	FABA3FFBE	ด	6CFC8EFEF
ต	6CFC8EFEF	ถ	7CE3AB7CF	ท	FEB88ACEE	ธ	F5799BDDF
น	22A3FEC9B	บ	28A99BD9B	ป	E88D9BD9B	ผ	E8F6CEFBE
ฝ	E88FBEF9B	พ	2AB8CACBE	ฟ	E88CBECBE	ภ	8CE28ADCE
ม	28A8BBD9B	ย	F8E79BD9F	ร	75772F5DD	ล	DFF8AB7CF
ว	6CE0FF6DD	ศ	7DF68ECEE	ษ	38B7DFD9F	ส	76F82A76E
ห	29B68ACEE	ฬ	ABE62ABFE	อ	FCF79BCFF	ฮ	76E79F7FF
ฯ	7EE8887EE	ะ	E9B79FF9B	ั	6CC08F6CC	ิ	93EAFB7DF
ี	9BBFFB73F	ึ	9F97FB7DF	ื	9BB7DB73F	ุ	6CECCF7DF
ู	EABCBBFBB	เ	8299FF933	แ	F6703FF7D	ใ	FCE03FF3D
ไ	BCE03FB3D	ๆ	FCE08FFCC	็	899BFB77F	่	7FFC6F7FE
้	E9AD6FDBF	๊	EAEB9A7BE	๋	93FC6FFF6	์	096F5FF96
๑	6CF3BFFFF	๒	2B2BFFBFF	๓	EEEBAB7EF	๔	69977BF7F
๕	DB677B77F	๖	D338DB7DD	๗	2CDBBE7FE	๘	999FDB757
๙	77772B757	๐	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Font : BrowalliaUPC

กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถ
ทธนบปผฝพฟภมยรลวศษสหฬอฮ
ฯ ั า ิ ี ึ ื ุ ู เ แ โ ใ ไ ๅ ็ ่ ้ ๊ ๋ ์
๑๒๓๔๕๖๗๘๙๐

ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code
ก	7CE28A6CE	ข	FAF8BBFBF	ฃ	7AF4BBFBF	ค	7CF68EFEF
ฅ	6CF68EFEF	ฆ	7AFFBEFBF	ง	8FFFBB7DD	จ	7CF8BB5FF
ฉ	7CE3FBFDF	ช	F7F9BBFFFF	ซ	E7E9BBFFF	ฌ	7AEFBF7BF
ญ	7AE8BB7BF	ฎ	DEE5B97DF	ฏ	DCEDBFFDF	ฐ	8773BF5FF
ฑ	FEB8AEEEF	ฒ	7AEFBFFBF	ณ	FAE8EF7BE	ด	6CFBAEFEF
ต	6CFFAEFEF	ถ	7CE88B7CF	ท	3EB68ACEE	ธ	F7E39BDDF
น	F2A8FEC3B	บ	F8A39BD9B	ป	E88D9BD9B	ผ	F8ABBE7BB
ฝ	E88FBE79B	พ	2BF8CABBE	ฟ	E89CBEBBE	ภ	CCE28ADCE
ม	F8AFBBD9B	ย	F8E79B79F	ร	77782FCDD	ล	7CF8CE7FF
ว	9CE0899DD	ศ	7DF68E7EF	ษ	22B3DFB6F	ส	7DF82B7FF
ห	FFA8AECEB	ฬ	ABE82ABFE	อ	7CF79BFDF	ฮ	76F79FCDF
ฯ	F9F899FFB	ั	E9BC9BF9B	า	6CE08F6CC	ิ	93E9FEFFF
ๆ	9B99FBFDF	็	9F9FFB7DF	่	9BFF3BFDF	้	7DECCD7FF
๊	FA3CBBCBB	๋	20FFDF3D3	์	767,033,777	๐	62E03363C
๑	B2E033B3C	๒	FCE08FFCC	๓	9B6EFE757	๔	57F57F57F
๕	E9B76FC6B	๖	9BEFFA7BE	๗	93FE6FFFB	๘	999F2FF67
๙	6CF7BBFFF	๐	23BBBFBFF	฿	EEBFAA7EE	เ	999B3FF5F
แ	96FB3FF67	โ	3337BB7FD	ใ	B896FE7BF	ไ	999BFB77F
ๆ	999B3FFFF	ฯ	63E39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Font : CordiaUPC

กขฃคฅฆงจฉชฌญฎฏฐฑฒณดตถ

ทธนบปผฝพฟภมยรลวศษสหฬอฮ

ฯ ั า ิ ี ึ ื ุ ู เ แ ไ ๅ ็ ่ ้ ๊ ๋ ์ ๎

๑๒๓๔๕๖๗๘๙๐

ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code
ก	7CE28A6CE	ข	FAE8BBFBF	ฃ	6AE8BBFFF	ค	FEFB8EFEF
ฅ	6CFB8EFEF	ฆ	FAFFBEF9F	ง	2FFCBB3DF	จ	6CF8CBEFF
ฉ	7CF7FBFDF	ช	FDE8BBFFF	ซ	6DE0BBFFF	ฌ	7AEAEF7BF
ญ	7AEFBA7BE	ฎ	6EED9BFDF	ฏ	6EED9BFDF	ฐ	857CBC7FF
ฑ	EEF8AAEEF	ฒ	7AFEEFFBF	ณ	EAEFFF7BE	ด	7DFB8EFEF
ต	6CFB8EFEF	ถ	7CEF8B7CF	ท	FEB88ECEE	ธ	75739BDDF
น	F2ABFEC9B	บ	F8A39BD9B	ป	E88D9BD9B	ผ	F8ABBE79B
ฝ	E88FBE79B	พ	2BB6CACBE	ฟ	E89BBECBE	ภ	7CEF8ADCE
ม	F8AFBED9B	ย	FAF79BC9B	ร	7CF72F56D	ล	6CF8BA7FE
ว	6CE0FF6CD	ศ	FEFB8EFEF	ษ	ABF3DFBFF	ส	EEFCBA7FE
ห	FFABAEC6B	ฬ	FD9BBABFE	อ	6CF79FEFF	ฮ	76F79FFDF
ฯ	F9F899FFB	ะ	E9BC9BF9B	า	6CE08F6CC	ิ	93E9FEFFF
ี	9B99FBFDF	ึ	9F9FFB7DF	ื	9BFF3BFDF	ุ	7DECCD7FF
ู	FA3CBBCBB	เ	20FFDF3D3	โ	767,033,777	ใ	62E03363C
ไ	B2E033B3C	ๆ	FCE08FFCC	็	9B6EFE757	่	57F57F57F
้	E9B76FC6B	๊	9BEFFA7BE	๋	93FE6FFFB	์	999FEFF67
๑	7CFFBBFFF	๒	2B3F9FBFF	๓	6EBFAA7EE	๔	66FB3BF7F
๕	F6FB3FF77	๖	333F9B5DD	๗	E89BBA7BE	๘	66EEFB757
๙	9BBF3F7FF	๐	6CEC9B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Font : DilleniaUPC

กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถ

ทธนบปผฝพฟภมยรลวศษสหฬอฮ

ฯ ั า ิ ี ึ ื ุ ู เ โ ใ ไ ๅ ็ ่ ้ ๊ ๋ ์

๑๒๓๔๕๖๗๘๙๐

ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code
ก	6CE28A6CE	ข	FAF8BBFBF	ฃ	7AF4BBFBF	ค	6CF68EFEF
ฅ	6EF68EFEF	ฆ	7AFFBEF9F	ง	0FFCBBCDF	จ	7CF8BB5FF
ฉ	5CECFEF3F	ช	FDE8BBFFF	ซ	6DE0BBFFF	ฌ	7AEFEF7BF
ญ	7AEFBB7BF	ฎ	DCE5B97DF	ฏ	DCE5BB7DF	ฐ	8778BB5FF
ฑ	7EB8AAEEF	ฒ	7AEFBFFBF	ณ	FAEFEF7BE	ด	ECFB8EFEF
ต	6CFFAEFEF	ถ	6CE8AA7CF	ท	FEB68ACEE	ธ	F5E39BDDF
น	F2ABFEC3B	บ	F8A39BD9B	ป	F8AD9BD9B	ผ	F8ABBEFBB
ฝ	F8AFBEF9B	พ	FBBBBACBE	ฟ	FABBBACBE	ภ	8CEF8ADCE
ม	F8AFBED9B	ย	F8A79F79F	ร	7778FFFDD	ล	7DFFCA7FE
ว	6CE0FF6DD	ศ	7CF68EFEF	ษ	ABF3DFBFF	ส	EEFCBA7FE
ห	FFABAECEB	ฬ	F99BBACFE	อ	7CE79FFFF	ฮ	76F79FFDF
ฯ	FEB087FCE	ะ	E6AC9FF9B	า	6CE0876CC	ิ	93EFFC7DF
ี	9BBFFB7DF	ึ	9F9FFB77F	ุ	9BBFFB7BF	ู	6CECCC7FF
ฺ	EABDBBFBB	เ	207FDF3D3	แ	7CF033777	ไ	7CF03373C
ๅ	BCE033B3C	ๆ	FCE08FFCC	็	9BEBFBFDF	่	557,557,557
้	F6379BC9B	๊	EAEF9AFBE	๋	937C67FFB	์	8F9F0FFF7
๑	7CFFBBFFF	๒	2B3F9FBFF	๓	6EBFAA7EE	๔	66FB3BF7F
๕	F6FB3FF77	๖	333F9B5DD	๗	E89BBA7BE	๘	66EEFB757
๙	9BBF3F7FF	๐	6CEC9B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Font : EucrosiaUPC

กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถ

ทธนบปผฝพฟภมยรลวศษสหฬอฮ

ฯ ั า ิ ี ึ ื ุ ู เ โ ใ ไ ๋ ่ ้ ๊ ๋ ์

๑๒๓๔๕๖๗๘๙๐

ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code
ก	6CE28A6CE	ข	FAB79BDBF	ฃ	7AF89BDBF	ค	ECF68EFEF
ฅ	6CF68EFEF	ฆ	6AFFBBD9F	ง	1FFCBB7DF	จ	5CD8BBCFF
ฉ	5CCD3BFDF	ช	7AEAFF7BF	ซ	7AE8BF7BE	ฌ	8EE5B97DF
ญ	8EE5B97DF	ฎ	877BBB5FF	ฏ	FEE8AEDEB	ฐ	7AEFEFFBF
ฑ	7AE8FF7BE	ฒ	ECFBAEFEF	ณ	6CFBAEFEF	ด	7CE8AB7CF
ต	2EB88ACEE	ถ	75739B7DF	ท	F2A3FBC9B	ธ	F8A39BC9B
น	F88C9BC9B	บ	F8EBBE7BB	ป	F88FBE79B	ผ	2BB6CACBE
ฝ	F89BBACBE	พ	8CE28ADCE	ฟ	F8AFBBD9B	ภ	FAF79BD9B
ม	75782FEDD	ย	F8E79B79F	ร	7CE8CB7CF	ล	DFFFAB7CF
ว	ECE0FFEDD	ศ	6DF68EFEF	ษ	28B3DFC6F	ส	56F82B7FE
ห	F9B88ECEB	ฬ	F6BBBECFF	อ	5CD79FCFF	ฮ	77F79FFDF
ฯ	FEB087FCE	ะ	E6B79BF9B	า	6CE0876CC	ิ	6CEBBE7DF
ี	9B9FFB7DF	ึ	8FFFFB77F	ื	9BFFFB7DF	ุ	6CECCC7DF
ู	EABDBBFBB	เ	207FDF3D3	โ	76C03377C	ใ	7CE03373C
ไ	BCE033B3C	ๆ	FCE08FFCC	็	666BFB7DF	่	557,557,557
้	E8379FD9F	๊	EAEFFA7BE	๋	937C67EE6	์	66672FF4F
๑	6CFBBBFFF	๒	2E3F9FBFF	๓	EEBFAA7EE	๔	699B7FF7F
๕	1FFB7FF57	๖	3CB7BB5DD	๗	E893BA7BE	๘	6BBBFB7DF
๙	E6BBEFF7F	๐	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Font : FreesiaUPC

กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถ
ทธนบปผฝพฟภมยรลวศษสหฬอฮ
ฯ ั า ิ ี ึ ื ุ ู เ โ ใ ไ ็ ่ ้ ๊ ๋ ์
๑๒๓๔๕๖๗๘๙๐

ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code
ก	7CE28A6CE	ข	FAF79BDBF	ฃ	FEF79B5BF	ค	6CF68EFEF
ฅ	6CF68EFEF	ฆ	6AFFCED9F	ง	1FFCBB7DF	จ	7CF8BBCFF
ฉ	7CE3FBFDF	ช	FAB79BDFF	ซ	FAF79BDFF	ฌ	7AEFFF7BF
ญ	7AE8BA7BE	ฎ	DCE5B97DF	ฏ	DCEDB97DF	ฐ	8778BF5FF
ฑ	FEF88AFEF	ฒ	7AFFEFFBF	ณ	7AE8FF7BE	ด	6CFBAEFEF
ต	6CFBAFFEF	ถ	7CE88B7CF	ท	FEB68ACEE	ธ	75739B7DF
น	F2A8FE33B	บ	F8A39BC9B	ป	E8A39BC9B	ผ	F8EBBE7BB
ฝ	E8AFBF79B	พ	FBB8CABBE	ฟ	E89BBABBE	ภ	7CE28ADCE
ม	F8AFBEDBB	ย	FAF79FCBF	ร	757829FDD	ล	7DF8CB7FE
ว	FCE0F9FDD	ศ	7CF68EFEF	ษ	28B36FBFF	ส	7CF8CB7FE
ห	FFB68ACEB	ฬ	FFEBBEBDF	อ	5CD79FFFF	ฮ	76F79FFDF
ฯ	FEB08FFCC	ั	E9A36FF9F	า	9CC08F9CC	ิ	6CEE3C7DF
ี	9AB7DB7DF	ึ	9FB97B7DF	ื	9BFFFB7DF	ุ	6CE8CF7FF
ู	E8BCBBF9B	เ	20FFDF3D3	โ	76703777C	ใ	F6E037F7C
ไ	B6C037B7C	ๆ	FCE08FFCC	็	999FFB7DF	่	FFF82FC67
้	E6BD6FC9F	๊	BAEB6A7BE	๋	93FC6FFF6	์	869F6FF67
๑	6CF3BFFFF	๒	23B39FBFF	๓	EEBCBF7EF	๔	699B7BFFF
๕	9BF77B77F	๖	333F9B5DF	๗	EEBCBAFBF	๘	999FFE757
๙	EB9B3B7FF	๐	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. Font : IrisUPC

กขคฆงจฉชฌญฎฏฐฑฒณดตถ

ทธนบปผฝพฟภมยรลวศษสหฬอฮ

ฯ ั า ิ ี ึ ื ุ ู เ แ ไ ใ ๅ ่ ้ ๊ ๋ ์ ๎

๑๒๓๔๕๖๗๘๙๐

ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*
ก	7CE28ACCE	ข	E8A89BE9B	ฃ	EAB8BBEBB	ค	7CF68EFEF
ฅ	6CF68E7EF	ฆ	EAAFBEFBB	ง	0FFCFF3DF	จ	7CFCBBFFF
ฉ	7CE3FBF3F	ช	E738BBEFB	ซ	E738BBEFB	ฌ	7ABABF7BF
ญ	7AB8BE7BE	ฎ	6CED9B7DF	ฏ	6CED9B7DF	ฐ	977CBBFFF
ฑ	EEB88AEEE	ฒ	EAEBDFFBF	ณ	7AB8BF7BE	ด	EEFB8F7EF
ต	EEBB8F7EF	ถ	7CE8AA7CE	ท	3EB68ACEE	ธ	F7C39B7DF
น	22ABFEC3B	บ	28A39BC9B	ป	288C9BC9B	ผ	28A39B79B
ฝ	28879B79B	พ	2AB8CACBE	ฟ	E88BBECBB	ภ	7CE28ADCE
ม	28AFBBD9B	ย	38B79BF9F	ร	F5772FDDD	ล	EDF8CA7CE
ว	6CE0FF6DD	ศ	6DF68EFEF	ษ	28B3DFCFF	ส	76F8CA7FE
ห	29A88ECEB	ฬ	F6B82ABFE	อ	7CE79BCFF	ฮ	C6E79BFDF
ฯ	F9B096FDE	ะ	E6AC6BF9B	า	6CE08F6CC	ิ	93EBFB7DF
ี	9B97DB7DF	ึ	9E97DB76F	ื	9BF7DB7DF	ุ	7DE8CD7FF
ู	EABCBBFBB	เ	20FFDF3D3	แ	CF78D3C57	ไ	76E03377C
ๅ	B6E033B7C	ๆ	FCE08FFCC	็	676BFBF7F	่	57F57F57F
้	E8BD6FF9F	๊	EA9FBA7BE	๋	93FC6FEE6	์	969B6B77F
๑	6CFBBFFFF	๒	23BBBFBFF	๓	EEBFABFEF	๔	699B5FF5F
๕	9B6B5FF57	๖	3CBFBB5DF	๗	E897BA7BE	๘	999BDBF7F
๙	9B9F3FF7F	๐	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Font : JasmineUPC

กขฃคฅฆงจฉชซฌญฎฏฐฑฒณดตถ
ทธนบปผฝพฟภมยรลวศษสหฬอฮ
ฯ ั า ิ ี ึ ื ุ ู เ แ โ ใ ไ ๅ ็ ่ ้ ๊ ๋ ์ ๎
๑๒๓๔๕๖๗๘๙๐

ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code
ก	7CE33A7DE	ข	FEE39B7DF	ฃ	FEE39B7DF	ค	6CFB9EFFE
ฅ	6CFB9EFFE	ฆ	FEE39B7DF	ง	3CC39B3DF	จ	7CEC9B7DF
ฉ	7CE33B7DF	ช	F7B39B7DF	ซ	F7B79B7DF	ฌ	EABBBF7BF
ญ	EAB3BE7BE	ฎ	DCE7997DF	ฏ	DCE5997DF	ฐ	757C9B7DF
ฑ	FEE33ADDE	ฒ	7AEBBBFBF	ณ	EABBBEFBE	ด	6CFB9FFFF
ต	6CFB9EFFF	ถ	7CE33A7DE	ท	BCB73ADDE	ธ	75739B7DF
น	C6A33B3DB	บ	C6A39B3DB	ป	2C879B3DB	ผ	6CBBBEFFF
ฝ	9C8BBEFDB	พ	AEFBBEFFB	ฟ	2C8BBEFFF	ภ	7CE93A6DE
ม	C6A39BDDB	ย	7CF79BDDF	ร	75789B5DD	ล	7DF78B7CF
ว	9CE89B5DD	ศ	6CFB9EFFE	ษ	27B3DFBFB	ส	7DF78B77F
ห	B7F73ADDE	ฬ	27FBBEFFF	อ	7CE39B7DF	ฮ	77E39B7DF
ฯ	FEB08FFCE	ะ	39A76BF9B	า	6CE08F6CC	ิ	63EF3E7DF
◌ั	6ABFDB7DF	◌	9FBFFB77F	◌	9BEB3B7DF	◌	6FFBFF7FF
◌	E8AC9BF9B	เ	F0FFDFFD3	โ	76703F77C	ใ	FDF03FF7C
ไ	BCC03FB3D	ๆ	FCE08FFCC	◌	76BBFEF57	◌	57F57F57F
◌	E9A76F39B	◌	EABFBA7BE	+	93FC6FFFB	◌	86972F76F
๑	ECFB9EFFF	๒	2B2BBFBFA	๓	EEBF8A7CE	๔	699B2BF4F
๕	EBFB2BFF7	๖	3C3F9B7DD	๗	E893BA7BE	๘	699BFB757
๙	E9BB7AFEE	๐	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Font : KodchiangUPC

กขขคฅงจฉชฌญฎฏฐฑฒณดตถ
ทธนบปผฝพฟภมยรลวศษสหฬอฮ
ฯ ั า ็ ่ ้ ๊ ๋ ์ ๎ ๏ ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ ๚ ๛
๑๒๓๔๕๖๗๘๙๐

ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code
ก	6CE33A7DE	ข	F6A39B7DB	ฃ	D6A39B7DB	ค	6CFB9AFDE
ฅ	6CFB9EFDE	ฆ	D6A39B7DB	ง	2ED39B3DF	จ	7CE39B7DF
ฉ	7CE33BCDF	ช	F7A39B7DB	ซ	F7A39B7DB	ฌ	EEB3BFFBF
ญ	EEBBBEFBE	ฎ	DCE39B7DF	ฏ	DCE39BDDF	ฐ	77E39B7DF
ฑ	FCE33A7DE	ฒ	6AFBBBFBF	ณ	EEBBBFBFE	ด	6EFB9E7FF
ต	6CFBBE7FF	ถ	6CE33A7DE	ท	CCE33ADDE	ธ	7DF39B7DF
น	E6A32BCDB	บ	E6A39BCDB	ป	E6839BCDB	ผ	66ABBE7DB
ฝ	E68BBE7DB	พ	AEE93ABFE	ฟ	ECBBBABFE	ภ	8CE93ADDE
ม	E6A39BDDB	ย	7DB79BD6B	ร	757DBDDDD	ล	7DF78B7CF
ว	6CE39B5DD	ศ	7DF68ECEE	ษ	38B7DFD9F	ส	76F32A76E
ห	C7B73ADDE	ฬ	EEFBBABFE	อ	7CF79B7DF	ฮ	77F79B7DF
ฯ	79B0967DD	ะ	28AC6B36B	า	6CE03F6DC	ิ	82E6AEF3E
ี	08FFFEFF9	ึ	08FFFEFF9	ื	0AAFFEFB9	ุ	F398FFCF3
ู	28A82B3DB	เ	C6F83FC77	แ	C6783FCFC	ไ	76F03F77C
ๅ	E6D03FE7C	ๆ	FCE08FFCC	็	9FBFBE7FF	่	77F77F77F
้	28A89BD9B	๊	FAF36AFBE	๋	93FEEFFFB	์	89FF2FF67
๑	6CEBBF7DF	๒	2FB39FBFF	๓	EEBBBB7DF	๔	999F7BF5F
๕	9B6F7BF57	๖	3333DB7DD	๗	98BBBB7BF	๘	999FFB757
๙	9BBB3B77F	๐	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. Font : Courier New

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890
```

ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*
A	D3E7DE73F	B	7DF7DFD7F	C	62E36B757	D	6CE39B7DF
E	7CF79FD7F	F	7CF757D4F	G	62E3DF757	H	7DB7DED7E
I	C67937D77	J	86C39B3DD	K	69B7CEF77	L	C2799BD9F
M	EEA39EFFB	N	ECB3BEFFE	O	6CE39B7DF	P	6DE747DDF
Q	6CEC7B7DF	R	6DE7CED7F	S	7DF3DB57D	T	EEF937D77
U	6CA39B7DB	V	7DAC6B76B	W	EEBBBAFFE	X	39B7DED77
Y	76B937D7B	Z	6DF79FD77				
a	5DD73B7DF	b	33B99B7DF	c	62E36B757	d	9DB33B7DF
e	7DF7D7FF7	f	D57923D57	g	66E5DD7DF	h	33333A7DE
i	C2993FD3C	j	0C9F993DD	k	C926DEB5B	l	C2993FD3C
m	EEEBAFFEE	n	CCE33A7DE	o	6CE39B7DF	p	CCE7677DF
q	66ECDD7DF	r	CEB923D57	s	7DF3DB53D	t	F233F3757
u	CCA33B3DB	v	76A36B76B	w	7DBBBAFFE	x	39B7DED77
y	7DAD6DD6B	z	6DF79FD77				
1	C2C937D3D	2	6DE75757D	3	6DD3DD57D	4	6EE7DD7DF
5	75739B7DD	6	6577DBFDF	7	EDE099EDD	8	7DF7DFD7F
9	7DE5DD7DF	0	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. Font : Helvetica

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890

ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*
A	63E6CE7DF	B	7DF7DFF7F	C	63E36B757	D	6CE39B7DF
E	757757F77	F	757,626,757	G	63E3DF7DF	H	39B6CEFFE
I	77F77F77F	J	08F39B399	K	B6B6CEB7E	L	20F3F33F3
M	EEAAABEEB	N	38B2CEEBB	O	6CE39B7DF	P	7DF6267DF
Q	6CE37F7DF	R	7DF6CE77F	S	7DF3DFDFF	T	C6F82FC66
U	28A39B39B	V	36A36B36B	W	AAEBBABBE	X	C6B63ED77
Y	36BC6FC6B	Z	8DD7B7D77				
a	7DF73F93F	b	B3F39B7DF	c	63E36B757	d	9B939B7DF
e	7DF7DFFF7	f	D7982FC66	g	6CE5DD7DF	h	B3F28A6CE
i	C2993FD3C	j	0C9F993DD	k	26AE3EB7B	l	77F77F77F
m	EEBAAFEEE	n	6CE28A6CE	o	6CE39B7DF	p	6CEE6F7DF
q	6CECEE7DF	r	FFF20F6F6	s	7DF7DFD3F	t	F29837C77
u	28A39B39B	v	36AC6B36B	w	AAEBBABBE	x	C6B93ED77
y	36AD66D6B	z	82D7F7D77				
1	FF908FFCC	2	6DE757D7D	3	6DF3DFD7D	4	9EEFFC7DF
5	75739B7DF	6	7DF7DB7DF	7	EDD929DDD	8	7DF7DFD7F
9	7DE7DD7DF	0	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. Font : LetterGothic

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890

ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*
A	93E6CE73E	B	7DF3DBF7F	C	63E39B757	D	6CE39B7DF
E	7573E3F77	F	757,207,757	G	63E3DF7DF	H	39B28AFFE
I	C67937D77	J	0CC39B3DD	K	B6BA3EE57	L	2073E33E3
M	EEA28EEEB	N	E8B2BEEBE	O	6CE39B7DF	P	6DE6267DF
Q	6CE39B7DF	R	6DE63E77F	S	747CDDD7D	T	C67827C66
U	28A39B39B	V	39B36BC6B	W	39BBBEBBE	X	36B9CED77
Y	39BC67C6B	Z	4DD717D77				
a	5DD73B7DF	b	BCB39B7DF	c	63E36B757	d	6B939B7DF
e	7DF3EBFF7	f	957207C56	g	6CE5DD7DF	h	3CB68A6CE
i	C2993FD3C	j	0C9F993DD	k	262ECEB5B	l	ECC087ECC
m	6EF2AF6EE	n	6CE28A6CE	o	6CE39B7DF	p	6CEE6E7DF
q	6CE3EE7DF	r	BCE2076C6	s	7C7CDFD7D	t	F2F36BC57
u	22A3EB33B	v	36AC6B36B	w	BBBBBABBE	x	36B93ED77
y	36AB69D6B	z	8DD7B7D77				
l	62CE3D53D	2	6DF757D7D	3	6DD3DD57D	4	6EE76B7DF
5	7577D9FDF	6	75779B7DF	7	EDD669DDD	8	7DF7DBD7F
9	7DE5DD7FF	0	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. Font : NewsGothic

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890

ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*	ตัวอักษร	*Q-code*
A	9CE6CE7DF	B	7DF7DFF7F	C	63E36B757	D	6CE39B7DF
E	757757F77	F	75762FF57	G	6CE3DF7DF	H	39B6CEFFE
I	77F77F77F	J	08FC9BC99	K	A6B63AF7E	L	20F3F33F3
M	EEAAABEEB	N	E8B2BEEBE	O	6CE39B7DF	P	7DF6267DF
Q	6CE37B7DF	R	7DF63E77F	S	7CF3DFD7D	T	C6F82FC66
U	28A39B39B	V	36AC9B36B	W	AAEBBABBE	X	C6B63ED7D
Y	36BC6FC6B	Z	8DD7B7D77				
a	7DF73B9DF	b	BCB39B7DF	c	6CE36B757	d	6B939B7DF
e	7DF7DFFF7	f	F7F82FC66	g	7DF7DF777	h	33B28A6CE
i	C2993FD3C	j	0C9F993DD	k	26A63EB7B	l	77F77F77F
m	EEBAAFEEE	n	6CE28A6CE	o	6CE39B7DF	p	6CEE6F7DF
q	6CE3EE7DF	r	FF920F6F6	s	7DFCDFD3D	t	FFF83CC77
u	28A39B39B	v	C6AC6BC6B	w	AAEBBABBE	x	C6B6CED77
y	36AD6DD9B	z	8DD7F7D77				
1	C2F93FD3D	2	6DE757D7D	3	6DD3DF57D	4	9EEFFB73F
5	75739B7DF	6	7577DB7DF	7	FDD9695DD	8	7DF7DFD7F
9	7DE5DD7DF	0	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. Font : TimesNewRomanPS

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890

ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code	ตัวอักษร	Q-code
A	E3E7DE73F	B	7DF7DFD7F	C	6CE39B757	D	6CE39B7DF
E	7DF7DFD7F	F	7DF7C7D5F	G	6DE3DB7DF	H	7DB7DED7E
I	C6F93FD77	J	86FF2916D	K	7DFFDED77	L	60F39B797
M	EEAEBBFFB	N	CCB3BEFFB	O	6CE39B7DF	P	6DE747DDF
Q	6CE37B7DF	R	6DF73ED7F	S	7CF3DFDFD	T	EEF93FD77
U	6CA39B7DB	V	7DACBB39B	W	FAFBBABBE	X	7DB63ED77
Y	7DBD7FC7B	Z	6DF79FD77				
a	6DF73B7DF	b	3333DB7DF	c	6CE36B757	d	9DD39B7DF
e	7DFC6B757	f	9C792FD57	g	76F7DF757	h	33373ADDE
i	C2993FD3C	j	0C9F993DD	k	CD2F3F37B	l	C2993FD3C
m	EEBBAFFEE	n	CCE33A7DE	o	6CE39B7DF	p	6CE7677DF
q	7CECDD73F	r	C9B92FD67	s	7DF3DFD7F	t	F2983CC77
u	CCA33B3DB	v	76AC6B39B	w	FAFCBABBE	x	36F63ED77
y	76B969D9B	z	6DF79FD77				
1	C2993FD3C	2	9CE9DF5DD	3	6DF3DD577	4	6E9FFC7DF
5	757FD957D	6	7573DB7DF	7	9DE9694DD	8	7DF7DFD7F
9	7CE5DD7DF	0	6CE39B7DF				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

โปรแกรมการทำงานของระบบ


```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <alloc.h>
#include "wtwg.h"

/* extern prototypes */
extern void readtif(char *buffer,int prior,int *width,int *length);
extern void zone(int coarse,int order);
extern int showzone(void);
extern void savetif(char *name,int num);
extern void read_zone(int n);
extern void beep(void);

/* extern variable */
int img_width,img_length;

/* local global variable */
static int zone_no=0;

/* local prototypes */
static void openfile(void);
static void exitsys(void);
static void setopt(void);
static void calzone(void);
static void displayzone(void);
static void savezone(void);
static void recog(void);

/* global local variables */
char Mon[1]={1};
static int coarse=0;
static int order=0;
static int prior=0;

WMENU  Fmenu[] = {
/* mu_entry mu_func   mu_help mu_menu mu_enable mu_highlight mu_key */
"Open",     openfile, "Open", NULL,   Mon,      0,           'O',
"eXit",     exitsys,  "Exit", NULL,   Mon,      1,           'X',
 NULL,      NULL,      NULL,  NULL,   NULL,     0,            0
};

WMENU  Omenu[] = {
/* mu_entry   mu_func mu_help    mu_menu mu_enable mu_highlight mu_key */
"sEtting",    setopt, "Setting", NULL,   NULL,     0,           'E',
 NULL,        NULL,    NULL,     NULL,   NULL,     0,            0
};

WMENU  Zmenu[] = {
/* mu_entry  mu_func      mu_help      mu_menu mu_enable mu_highlight mu_key */
"Find zone", calzone,     "Find zone", NULL,   NULL,     0,           'F',
"Display" ,  displayzone, "Display",   NULL,   NULL,     0,           'D',
"Save",      savezone,    "Save",      NULL,   NULL,     0,           'S',
 NULL,       NULL,        NULL,        NULL,   NULL,     0,            0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
};

WMENU  Rmenu[] = {
/* mu_entry      mu_func  mu_help           mu_menu mu_enable mu_highlight mu_key */
"do Regonition", recog,   "Do recognition", NULL,   NULL,     3,           'R',
NULL,            NULL,     NULL,            NULL,   NULL,     0,           0
};

WMENU  topmenu[] = {
/* mu_entry   mu_func mu_help        mu_menu mu_enable mu_highlight mu_key */
"File",         NULL,  "FILE",       Fmenu,  Mon,      0,           ALT_F,
"Option",       NULL,  "OPTION",     Omenu,  Mon,      0,           ALT_O,
"Zone",         NULL,  "ZONE",       Zmenu,  Mon,      0,           ALT_Z,
"Recognize",    NULL,  "RECOGNIZE",  Rmenu,  Mon,      0,           ALT_R,
 NULL,          NULL,   NULL,        NULL,   NULL,     0,           0
};

WFORM option_form[] = {
/* label                        user     picklist format func  len HOLD x   y */
"Coarseness <0-20>:",           &coarse, NULL,    "%i",  NULL, 3,  0,   28, 3,
 NULL,                          NULL,    NULL,     NULL,  NULL, 0,  0,   35, 10
};

//"Order <1=Row, 2=Column>:", &order,  NULL,    "%i",  NULL, 3,  0,   28, 4,

char *title="\n\
             \n  The Document Analysis\n\
             \n        and\n\
             \n  Character Recognition\n\
             \n\n";


char *about="\n\
             \n   Author: Mr. Somsak  Walairacht\n\
             \n  Version: 1.00\n\
             \n Released: 1 Jan. 1994\n\
             \n\n\n";

int main(void)
{
        int key;
        char buff[80];

        printf("\nSelect memory management, where image will be stored\n");
        printf("\n\t  0. always stored on disk.");
        printf("\n\t  1. stored to EMS RAM. If no room, items are swapped to disk.");
        printf("\n\t255. never store on disk, very important data.");
        printf("\n\nYour selection: ");
        scanf("%d",&prior);

        winit('G');
        wpulldown(topmenu);
        whelp_install("zone");
        whelp_ptr="ABOUT_HELP";
        wsetattr(LIGHTGRAY);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        wsetc(140);
        wopen(1,wyabsmax-1,wxabsmax-22,1,48,1,112,0);
        wputs(" <F1> for Help. <Alt>+Highlight letter to Menu. ");
        wabandon();
        wopen(wxabsmax-19,wyabsmax-1,19,1,48,1,112,0);
        wputs(" <Esc> to exit. ");
        wabandon();
        wsetattr(31);
        wpromptc(NULL,title,NULL);
        beep();
        wpromptc(" About ",about,NULL);
        beep();
        wflush(); /* flush keyboard */
        do {
                key=wgetc();
                if(key==ESCAPE) {
                        beep();
                        wexit();
                }
        } while(key != 'Q');
        return 0;
}


static void openfile(void)
{
        char buffer[20];
        int key;

        buffer[0]=0;
        wsetlocation(WLOC_ATCUR,5,4);
        wsetattr(112);
        key=wprompts(" Open Image ",
        " Enter TIFF filename:\n\n ", buffer,sizeof(buffer));
        if(key==ESCAPE) return;
        else {
                readtif(buffer,prior,&img_width,&img_length);
                Omenu[0].mu_enable="1";         /* enable setting option */
                Fmenu[1].mu_enable="1";         /* enable exit */
        }
        return;
}

static void exitsys(void)
{
        beep();
        wexit();
}

static void setopt(void)
{
        wsetlocation(WLOC_ATCUR,2,2);
        wscanform(" ZONE OPTION SETTING ",option_form);
        Zmenu[0].mu_enable="1"; /* enable find zone */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return;
}

static void calzone(void)
{
        int order=2;                    /* set order to row major */
        Fmenu[0].mu_enable=NULL;        /* disable open file */
        Omenu[0].mu_enable=NULL;        /* disable setting */
        zone(coarse,order);
        Zmenu[0].mu_enable=NULL;        /* disable find zone */
        Zmenu[1].mu_enable="1";         /* enable display zone */
        return;
}

static void displayzone(void)
{

        zone_no=showzone();
        Zmenu[2].mu_enable="1"; /* enable save zone */
        Rmenu[0].mu_enable="1"; /* enable recognition */
        return;
}

static void savezone(void)
{
        char buffer[20];
        int key;

        buffer[0]=0;
        wsetlocation(WLOC_ATWIN,5,40);
        wsetattr(48);
        key=wprompts(" Save Image ",
                " Enter TIFF filename:\n\n ", buffer,sizeof(buffer));
        if(key==ESCAPE) return;
        else savetif(buffer,zone_no);
        return;
}


static void recog(void)
{
        read_zone(zone_no);
        beep();
        return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <math.h>
#include <alloc.h>
#include <string.h>
#include <conio.h>
#include <graphics.h>
#include "read.h"
#include "wtwg.h"

#define farmemcpy(dest,src,num) \
        movedata(FP_SEG(src),FP_OFF(src),FP_SEG(dest),FP_OFF(dest),num)

typedef WHEAP *WHEAP_PTR;
WHEAP_PTR *hlist;
WINDOW *indicate;

/* extern prototype */
extern void beep(void);

/* local variables */
static char masktable[8]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};
static unsigned int numbertype='II';
static FILEINFO fi;
static char path[40];
static char pathmsg[40];
static int priority;

/* local prototype */
void readtif(char *tiffile,int p,int *width,int *length);
void strmfe(char *new,char *old,char *ext);
int readheap(char *p,unsigned int n);
static int dosetup(FILEINFO *fi);
static int doclosedown(FILEINFO *fi);
static unsigned long fgetlong(FILE *fp);
static unsigned int fgetword(FILE *fp);
static int unpacktiff(FILE *fp,FILEINFO *fi);
static int readline(char *p,FILE *fp,int bytes,FILEINFO *fi);
static void convertline(char *dest,char *source,FILEINFO *fi);
static void setdefaults(FILEINFO *fi);
static void decodetag(FILEINFO *fi,FILE *fp);
static int allocheap(long n);
static int writeheap(char *p,unsigned int n);


/* Read TIFF routine start here */

void readtif(char *tiffile,int p,int *width,int *length)
{
        FILE *fp;

        strmfe(path,tiffile,"TIF");
        strupr(path);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if((fp=fopen(path,"rb")) != NULL) {
                priority=p;
                fi.setup=dosetup;
                fi.closedown=doclosedown;
                unpacktiff(fp,&fi);
                *width=fi.width;
                *length=fi.depth;
        }
        else {
                beep();
                sprintf(pathmsg," %s not found. ",path);
                wpromptc(" Error Open File ",pathmsg,NULL);
        }
        return;
}


/* unpack a TIFF file */
static int unpacktiff(FILE *fp,FILEINFO *fi)
{
        char *p,*pr;
        unsigned long l;
        int i,j,n;
        int run=0;

        wopen(wxabsmax-19,wyabsmax-1,19,1,112,1,112,1);
        indicate=w0;
        wputs(" Please wait...");

        /* get the number types */
        numbertype=fgetword(fp);

        /* see if it's real */
        if(numbertype=='II' || numbertype=='MM') {

                /* trash the version */
                fgetword(fp);

                /* seek through the image directories */
                while((l=fgetlong(fp)) != 0L) {

                        fseek(fp,l,SEEK_SET);

                        /* set default values */
                        setdefaults(fi);

                        /* decode all the tags */
                        n=fgetword(fp);
                        for(i=0;i<n;++i) decodetag(fi,fp);

                        /* save the file position */
                        l=ftell(fp);

                        /* figure the bits - limit only monochrome pictures */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                fi->bits=fi->samples*fi->bitspersample;

                /* figure the bytes */
                if(fi->bits <= 8)
                        fi->bytes=pixels2bytes(fi->width)*fi->bits;
                else fi->bytes=fi->width/8;

                if(fi->width != 0 &&
                   fi->depth != 0 &&
                   fi->offset != 0L) {
                        if((p=malloc(fi->bytes))==NULL)
                                werror(1,"Error allocate p in unpacktiff()");
                        if((pr=malloc(fi->width))==NULL)
                                werror(1,"Error allocate pr in unpacktiff()");
                        (fi->setup)(fi);
                        if(fi->count==1L) {
                                fseek(fp,fi->offset,SEEK_SET);
                                for(i=0;i<fi->depth;++i) {
                                        if(!(i%(int)(fi->depth/10))) {
                                                wgoto(0,0);
                                                wprintf("%3d%% Completed.",run);
                                                run+=10;
                                        }
                                        if(readline(p,fp,fi->bytes,fi) != fi->bytes) {
                                                free(pr);
                                                free(p);
                                                wheap_freeall();
                                                return(BAD_READ);
                                        }
                                        convertline(pr,p,fi);
                                        writeheap(pr,i);
                                }
                        }
                        else {
                                for(l=0;l<fi->count;++l) {
                                        fseek(fp,fi->offset+(l*sizeof(long)),SEEK_SET);
                                        fseek(fp,fgetlong(fp),SEEK_SET);
                                        for(j=0;j<(int)fi->rowsperstrip;++j) {
                                                if((int)((l*fi->rowsperstrip)+j) >= fi->dept
h) break;

                                                if(readline(p,fp,fi->bytes,fi) != fi->bytes)
 {
                                                        free(pr);
                                                        free(p);
                                                        wheap_freeall();
                                                        return(BAD_READ);
                                                }
                                                convertline(pr,p,fi);
                                                writeheap(pr,(int)(l*fi->rowsperstrip)+j);
                                        }
                                }
                        }
                        free(pr);
                        free(p);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                wclear();
                                wputs(" <Esc> to exit. ");
                                wbury();
                                (fi->closedown)(fi);
                                fseek(fp,1,SEEK_SET);
                        }
                }
                return(GOOD_READ);
        } else return(BAD_FILE);
}

static void convertline(char *dest,char *source,FILEINFO *fi)
{
        int i;

        switch(fi->bits) {
                case 8:
                        memcpy(dest,source,fi->width);
                        break;
                case 4:
                        for(i=0;i<fi->width;) {
                                dest[i++]=(*source>>4) & 0x0f;
                                dest[i++]=*source & 0x0f;
                                ++source;
                        }
                        break;
                case 1:
                        for(i=0;i<fi->width;++i) {
                                if(source[i>>3]&masktable[i&0x0007])
                                        dest[i]=0x00;
                                else
                                        dest[i]=0xff;
                        }
                        break;
        }
}

static void setdefaults(FILEINFO *fi)
{
        int i;

        fi->width=0;
        fi->depth=0;
        fi->bits=0;
        fi->threshhold=1;
        fi->samples=1;
        fi->compression=1;
        fi->offset=0L;
}

static void decodetag(FILEINFO *fi,FILE *fp)
{
        long length,offset,pos;
        int tag,type,i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
tag=fgetword(fp);
type=fgetword(fp);

if(type == TIFFlong) {
        length=fgetlong(fp);
        offset=fgetlong(fp);
}
else {
        length=(unsigned long)fgetword(fp);
        fgetword(fp);
        offset=(unsigned long)fgetword(fp);
        fgetword(fp);
}

switch(tag) {
        case SubfileType:
                break;
        case ImageWidth:
                fi->width=(unsigned int)offset;
                break;
        case ImageLength:
                fi->depth=(unsigned int)offset;
                break;
        case RowsPerStrip:
                if(type==TIFFlong) fi->rowsperstrip=offset;
                else fi->rowsperstrip=offset & 0xffffL;
                break;
        case StripOffsets:
                if(type==TIFFlong) fi->offset=offset;
                else fi->offset=offset & 0xffffL;
                fi->count=(int)length;
                break;
        case StripByteCounts:
                if(type==TIFFlong) fi->bytecount=offset;
                else fi->bytecount= offset & 0xffffL;
                break;
        case SamplesPerPixel:
                fi->samples=(int)offset;
                break;
        case BitsPerSample:
                if(length > 1L) {
                        pos=ftell(fp);
                        fseek(fp,offset,SEEK_SET);
                        fi->bitspersample=fgetword(fp);
                        fseek(fp,pos,SEEK_SET);
                } else fi->bitspersample=(int)offset;
                break;
        case Threshholding:
                fi->threshhold=(int)offset;
                break;
        case Compression:
                fi->compression=(int)offset;
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case Group3Options:
        break;
case Group4Options:
        break;
case FillOrder:
        break;
case PlanarConfiguration:
        break;
case CellWidth:
        break;
case CellLength:
        break;
case MinSampleValue:
        break;
case MaxSampleValue:
        break;
case PhotometricInterp:
        break;
case GrayResponseUnit:
        break;
case GrayResponseCurve:
        break;
case ColorResponseUnit:
        break;
case ColorResponseCurves:
        break;
case XResolution:
        break;
case YResolution:
        break;
case ResolutionUnit:
        break;
case Orientation:
        break;
case DocumentName:
        break;
case PageName:
        break;
case XPosition:
        break;
case YPosition:
        break;
case PageNumber:
        break;
case ImageDescription:
        break;
case Make:
        break;
case Model:
        break;
case FreeOffsets:
        break;
case FreeByteCounts:
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case ColorMap:
                        break;
        }
}

/* read a compressed PackBits line */
static int readline(char *p,FILE *fp,int bytes,FILEINFO *fi)
{
        int c,i,n=0;


        if(fi->compression==1) return(fread(p,1,bytes,fp));
        else if(fi->compression==0x8005) {
                do {
                        c=fgetc(fp) & 0xff;
                        if(c & 0x80) {
                                if(c != 0x80) {
                                        i = ((~c) & 0xff)+2;
                                        c=fgetc(fp);
                                        while(i--) p[n++] = c;
                                }
                        }
                        else {
                                i=(c & 0xff)+1;
                                while(i--) p[n++] = fgetc(fp);
                        }
                } while(n < bytes);
                return(n);
        }
        else return(0);
}

/* get a word from a TIFF file */
static unsigned int fgetword(FILE *fp)
{
        if(numbertype == 'II') return((fgetc(fp) & 0xff) + ((fgetc(fp) & 0xff) << 8));
        else return(((fgetc(fp) & 0xff) << 8) + (fgetc(fp) & 0xff));
}

/* get a long integer from a TIFF file */
static unsigned long fgetlong(FILE *fp)
{
        if(numbertype == 'II')
                return((unsigned long)(fgetc(fp) & 0xff) +
                  ((unsigned long)(fgetc(fp) & 0xff) << 8) +
                  ((unsigned long)(fgetc(fp) & 0xff) << 16) +
                  ((unsigned long)(fgetc(fp) & 0xff) << 24));
        else
                return(((unsigned long)(fgetc(fp) & 0xff) << 24) +
                  ((unsigned long)(fgetc(fp) & 0xff) << 16) +
                  ((unsigned long)(fgetc(fp) & 0xff) << 8) +
                  (unsigned long)(fgetc(fp) & 0xff));
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* make file name with specific extension */
void strmfe(char *new,char *old,char *ext)
{
        while(*old != 0 && *old != '.') *new++=*old++;
        *new++='.';
        while(*ext) *new++=*ext++;
        *new=0;
        return;
}

static int writeheap(char *p,unsigned int n)
{
        char *ptr=NULL;

        if((ptr=wheap_access(hlist[n],0))==NULL)
                return 0;
        else {
                farmemcpy(ptr,p,fi.width);
                wheap_deaccess(hlist[n],1);
                return 1;
        }
}


int readheap(char *p,unsigned int n)
{
        char *ptr=NULL;

        if((ptr=wheap_access(hlist[n],1))==NULL)
                return 0;
        else {
                farmemcpy(p,ptr,fi.width);
                wheap_deaccess(hlist[n],0);
                return 1;
        }
}


static int allocheap(long n)
{
        int i;

        if((hlist = malloc(fi.depth*sizeof(WHEAP_PTR)))==NULL)
                werror(1,"Error allocate hlist in allocheap()");
        for(i=0; i<fi.depth; i++) {
                if((hlist[i] = wheap_alloc(fi.width+1,priority,NULL))==NULL)
                        werror(1,"Error allocate hlist element in allocheap()");
        }
        return 1;
}

static int dosetup(FILEINFO *fi)
{
        if(!allocheap((long)fi->width*(long)fi->depth))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                werror(1,"Error allocate heap memory");
        return 0;
}

static int doclosedown(FILEINFO *fi)
{
        register int i,j;
        char *p,msg[15];
        float ScaleX,ScaleY,factor;
        int Xpos,Ypos,color;

        wopen(4,8,72,44,112,DOUBLE_BORDER,112,1);
        wtitle(path);

        if((p=malloc(fi->width))==NULL) {
                werror(1,"Error allocate p in doclosedown()");
        }

        if(fi->width>SCREENWIDE || fi->depth>SCREENDEEP) {
                ScaleX=(72*8) / (float)fi->width;
                ScaleY=(44*8) / (float)fi->depth;
                factor=(ScaleX < ScaleY)? ScaleX : ScaleY;

                sprintf(msg," Scale 1:%d ",(int)((1/factor)+0.9));
                wps(msg);

                for(j=0; j<fi->depth; j++) {
                        readheap(p,j);
                        Ypos=(float)j*factor+0.5;
                        for(i=0; i<fi->width; i++) {
                                color=(p[i]==0xff)? BLACK : WHITE;
                                Xpos=(float)i*factor+0.5;
                                putpixel(Xpos,Ypos,color);
                        }
                        if(j==fi->depth-2) break;
                }
        }
        else {
                wps(" Scale 1:1 ");
                for(j=0; j<fi->depth; j++) {
                        readheap(p,j);
                        for(i=0; i<fi->width; i++) {
                                if(p[i]==0xff) putpixel(i,j,BLACK);
                                else putpixel(i,j,WHITE);
                        }
                }
        }
        free(p);
        wabandon();
        beep();
        return(GOOD_READ);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <dos.h>
#include <conio.h>
#include <math.h>
#include <alloc.h>
#include <graphics.h>
#include "zone.h"
#include "read.h"
#include "wtwg.h"

/* extern prototypes & variables */
extern int readheap(char *p,unsigned int n);
extern int img_width,img_length;
extern WINDOW *indicate;

/* local variables */
char *image=NULL;
char *lines=NULL;
ZONE zone_list[MAX_ZONES];      /* this could be dynamically allocated */
int zone_count;                 /* index of zones */
int lines_per_buffer;           /* number of lines in a buffer */
int lines_per_page;             /* number of lines of image */
int bytes_per_line;             /* bytes in a scan line ie.,image width */
int num_sample;                 /* number of sample of image */
int remain_line;                /* remain line after sampling */
SAMPLEINFO sample[40];          /* zone extent, zone_count, line offset */
FILE *z_info;

/* extern prototypes */
extern void beep(void);

/* local prototypes */
void zone(int coarseness,int order);
void indicator(char title[15],int i);
static void block_zones(char *image,int dx,int dy,int coarseness);
static void sequence_zones(char *image,int dx,int dy);
static int extract_zone(char *image,int dx,int dy,int x,int y,ZONE *zone_ptr);
static void overlap_zones(ZONE *zone_array,int *array_size);
static void sort_zones(int dx,int dy,int order);
static char *get_page(int dx,int dy);
static void record_zone(int n);
static void combine_samples(void);
static void overlap_samples(ZONE *zone_array,int *array_size);

void zone(int coarseness,int order)
{
        char *ptr;
        int dx,dy,offset;
        long image_size;
        unsigned int i,j;
        FILE *out_fp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (coarseness < 0 || coarseness > 20) {
        /* test coarseness parameter */
        errno = EINVAL;
        return;
}
bytes_per_line = dx = img_width;
lines_per_page = dy = img_length;
image_size=(long)img_width * (long)img_length;
if(image_size > (long)2500000) {
        beep();
        wpromptc(NULL," Image too big to detect zone.\n",NULL);
        return;
}
else {
        wreopen(indicate);
        indicate=w0;
        wgoto(0,0);
        wputs(" Please wait...");
        if(image_size > BUFFER_SIZE) {
                /* sampling image along vertical */
                lines_per_buffer=MAX_MALLOC / bytes_per_line;
                num_sample=(float)lines_per_page / (float)lines_per_buffer + 0.5;
                remain_line=lines_per_page - ((num_sample-1) * lines_per_buffer);
                if((image=malloc(MAX_MALLOC))==NULL)
                        werror(1,"Error allocate image in zone()");
                memset(image,0x00,MAX_MALLOC);
                if((lines=malloc(bytes_per_line))==NULL)
                        werror(1,"Error allocate lines in zone()");
                ptr=image;
                for(i=0; i<num_sample; i++) {
                        offset=lines_per_buffer * i;
                        if(i==num_sample-1) {
                                for(j=0; j<remain_line; j++,image+=bytes_per_line) {
                                        readheap(lines,offset+j);
                                        memcpy(image,lines,bytes_per_line);
                                }
                        }
                        else {
                                for(j=0; j<lines_per_buffer; j++,image+=bytes_per_line) {
                                        readheap(lines,offset+j);
                                        memcpy(image,lines,bytes_per_line);
                                }
                        }
                        image=ptr;
                        block_zones(image,bytes_per_line,lines_per_buffer,coarseness);
                        sequence_zones(image,bytes_per_line,lines_per_buffer);
                        record_zone(i);
                }
                free(lines);
                free(image);
                /* combine zones in each sample */
                combine_samples();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        /* find overlaped zones of each sample & combine them */
                        overlap_samples(zone_list,&zone_count);
                        sort_zones(bytes_per_line,lines_per_page,order);

                }
                else {
                        /* image can be stored in buffer, no need sampling */
                        if((image=get_page(dx,dy))==NULL)
                                werror(1,"Error reading image");
                        block_zones(image,dx,dy,coarseness);
                        sequence_zones(image,dx,dy);
                        sort_zones(dx,dy,order);
                }
                wclear();
                wgoto(0,0);
                wsetattr(48);
                wputs(" <Esc> to exit. ");
                wabandon();
                beep();
                return;
        }
}

void indicator(char title[15],int i)
{
        wgoto(0,0);
        wprintf("%s %3d",title,i);
        return;
}


/**************** LOCAL FUNCTIONS ****************/

static char *get_page(int dx,int dy)
{
        int i;
        char *img_ptr,*p;

        if((image=malloc(dx * dy))==NULL) {
                werror(1,"Error allocate image in get_page()");
        }
        if((p=malloc(dx))==NULL) {
                werror(1,"Error allocate p in get_page()");
        }
        for(i=0,img_ptr=image; i<dy; i++,img_ptr+=dx) {
                readheap(p,i);
                memcpy(img_ptr,p,dx);
        }
        free(p);
        return image;
}


static void block_zones(char *image,int dx,int dy,int coarseness)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        int i,j,x,y,score;
        char *line,*pixel;

        for (y=1, line=image+dx ; y < dy-1 ; y++, line+=dx) {
                indicator(" Block zones  ",y);
                /* walk through buffer scanning for neighbors */
                for (x=1, pixel=line+1 ; x < dx-1 ; x++, pixel++) {
                        score = 0;
                        if (*(pixel-1) & ALIVE)      /* left */
                                score++;
                        if (*(pixel+1) & ALIVE)      /* right */
                                score++;
                        if (*(pixel-dx) & ALIVE)     /* up */
                                score++;
                        if (*(pixel+dx) & ALIVE)     /* down */
                                score++;
                        if (*pixel == WHITE && score >= LIFE_SCORE)
                                *pixel = PREGNANT;
                        else if (*pixel == BLACK && score <= DEATH_SCORE)
                                *pixel = SICK;
                }
        }
        for (pixel = image ; pixel < image + dy * dx ; pixel++) {
                /* birth and bury */
                if (*pixel == PREGNANT)
                        *pixel = BLACK;
                else if (*pixel == SICK)
                        *pixel = WHITE;
        }
        if (coarseness <= 0)            /* no need to coalesce */
                return;
        /* coalesce regions, based on coarseness, i.e. coarseness == 2
         * will close all horizontal/vertical gaps of 2 pixels */
        for (y=0, line=image+1 ; y < dy ; y++, line+=dx) {
                /* coalesce horizontally */
                for (x=1, pixel=line ; x < dx-coarseness ; x++, pixel++) {
                        if (*pixel == WHITE && *(pixel-1) == BLACK) {
                                for (i = 1 ; i <= coarseness ; i++) {
                                        if (*(pixel+i)) {
                                                for (i = 0 ; i < coarseness ; i++, pixel++)
                                                        *pixel = BLACK;
                                                pixel--;
                                                x += coarseness-1;
                                                break;
                                        }
                                }
                        }
                }
        }
        for (x=0, line=image+dx ; x < dx ; x++, line++) {
                /* coalesce vertically */
                for (y=1, pixel=line ; y < dy-coarseness ; y++, pixel+=dx) {
                        if (*pixel == WHITE && *(pixel-dx) == BLACK) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        for (i=1, j=dx ; i <= coarseness ; i++, j+=dx) {
                                if (*(pixel+j)) {
                                        for (i = 0 ; i < coarseness ; i++, pixel+=dx)
                                                *pixel = BLACK;
                                        pixel -= dx;
                                        y += (coarseness-1);
                                        break;
                                }
                        }
                }
        }
        return;
}


static void sequence_zones(char *image,int dx,int dy)
{
        int x,y;
        char *ptr;

        for (y=0,zone_count=0 ; y < dy && zone_count < MAX_ZONES ; y+=MIN_Y_ZONE) {
                /* extract zones from block images in y order */
                ptr = image + y * dx;
                for (x = 0 ; x < dx  ; x += MIN_X_ZONE)
                        if (*(ptr+x)) { /* found point */
                                if (zone_count >= MAX_ZONES)
                                        break;
                                while (x > 0 && *(ptr+x-1))     /* back up to left side */
                                        x--;
                                if(extract_zone(image,dx,dy,x,y,zone_list+zone_count)) {
                                        indicator(" Extract zones",zone_count);
                                        zone_count++;   /* get zone */
                                }
                        }
        }
        /* remove overlaping zones */
        overlap_zones(zone_list,&zone_count);
        return;
}


static void sort_zones(int dx,int dy,int order)
{
        int x,y,i,j,index,fudge_x,fudge_y,z=0;
        ZONE temp;

        if (order == COLUMN_MAJOR) {
                for (i = 0 ; i < zone_count-1 ; i++) {
                        indicator(" Sorting zones ",z+i);
                        /* sort on x1 */
                        for (j = i+1 ; j < zone_count ; j++) {
                                if (zone_list[j].x1 < zone_list[i].x1) {
                                        temp = zone_list[i];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                zone_list[i] = zone_list[j];
                                zone_list[j] = temp;
                        }
                }
        }
        for (i = 0 ; i < zone_count-1 ; i++) {
                /* order zones left to right, up to down */
                indicator(" Sorting zones ",z+i);
                x = zone_list[i].x2;
                y = zone_list[i].y1;
                fudge_x = zone_list[i].x1+dx/20;    /* 5% slippage on alignment */
                for (j = i+1, index = -1 ; j < zone_count ; j++) {
                        /* find any zones above and within x extent of zone_list[i] */
                        if (zone_list[j].x1 > x)
                                break;
                        if (zone_list[j].y1 < y && zone_list[j].x1 <= fudge_x) {
                                x = zone_list[j].x2;
                                y = zone_list[j].y1;
                                index = j;
                        }
                }
                if (index != -1) {
                        temp = zone_list[i];
                        zone_list[i] = zone_list[index];
                        zone_list[index] = temp;
                }
        }
}
else { /* ROW_MAJOR */
        /* already sorted in y1 order */
        for (i = 0 ; i < zone_count-1 ; i++) {
                indicator(" Sorting zones ",z+i);
                /* order zones up to down, left to right */
                y = zone_list[i].y2;
                x = zone_list[i].x1;
                fudge_y = zone_list[i].y1+dy/20;    /* 5% slippage on alignment */
                for (j = i+1, index = -1 ; j < zone_count ; j++) {
                        /* find any zones left of and within y extent of zone_list[i] */
                        if (zone_list[j].y1 > y)
                                break;
                        if (zone_list[j].x1 < x && zone_list[j].y1 <= fudge_y) {
                                y = zone_list[j].y2;
                                x = zone_list[j].x1;
                                index = j;
                        }
                }
                if (index != -1) {
                        temp = zone_list[i];
                        zone_list[i] = zone_list[index];
                        zone_list[index] = temp;
                }
        }
}
return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}


static int extract_zone(char *image,int dx,int dy,int x,int y,ZONE *zone_ptr)
{
        int ix,iy,minx,miny,maxx,maxy;  /* perimeter variables & min/max values */
        HEADING dir;                    /* current direction */
        char *previous,*next,*here;    /* buffer pointers */

        minx = maxx = ix = x;           /* preset min/max x/y and perimeter vars */
        miny = maxy = iy = y;
        dir  = GOING_UP;                /* starting direction */
        do      /* walk perimeter, recording min/max of rectangular region */
        {
                if (ix < minx)              /* update min/max */
                        minx = ix;
                if (ix > maxx)
                        maxx = ix;
                if (iy < miny)
                        miny = iy;
                if (iy > maxy)
                        maxy = iy;
                here = image + iy * dx + ix;    /* where are we? */
                next = here + dx;
                previous = here - dx;
                switch (dir)                /* based on current direction, */
                {                           /* look around for next direction */
                        case GOING_UP:
                        if (ix > 0 && *(here-1))
                        {
                                ix--;
                                dir = GOING_LEFT;
                                break;
                        }
                        if (iy > 0 && *previous)
                        {
                                iy--;
                                break;
                        }
                        if (ix < dx-1 && *(here+1))
                        {
                                ix++;
                                dir = GOING_RIGHT;
                                break;
                        }
                        if (iy < dy-1 && *next)
                        {
                                iy++;
                                dir = GOING_DOWN;
                                break;
                        }
                        break;
                        case GOING_RIGHT:
                        if (iy > 0 && *previous)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        iy--;
        dir = GOING_UP;
        break;
}
if (ix < dx-1 && *(here+1))
{
        ix++;
        break;
}
if (iy < dy-1 && *next)
{
        iy++;
        dir = GOING_DOWN;
        break;
}
if (ix > 0 && *(here-1))
{
        ix--;
        dir = GOING_LEFT;
        break;
}
break;
case GOING_DOWN:
if (ix < dx-1 && *(here+1))
{
        ix++;
        dir = GOING_RIGHT;
        break;
}
if (iy < dy-1 && *next)
{
        iy++;
        break;
}
if (ix > 0 && *(here-1))
{
        ix--;
        dir = GOING_LEFT;
        break;
}
if (iy > 0 && *previous)
{
        iy--;
        dir = GOING_UP;
        break;
}
break;
case GOING_LEFT:
if (iy < dy-1 && *next)
{
        iy++;
        dir = GOING_DOWN;
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        }
                        if (ix > 0 && *(here-1))
                        {
                                ix--;
                                break;
                        }
                        if (iy > 0 && *previous)
                        {
                                iy--;
                                dir = GOING_UP;
                                break;
                        }
                        if (ix < dx-1 && *(here+1))
                        {
                                ix++;
                                dir = GOING_RIGHT;
                                break;
                        }
                        break;
                }
        } while (ix != x || iy != y);   /* until we return to the start */
        for (iy=miny, here=image+miny*dx+minx; iy<=maxy; iy++,here+=dx)
                memset(here,WHITE,maxx-minx+1); /* white out the region */
        if ((maxx-minx+1 < MIN_X_ZONE) || (maxy-miny+1 < MIN_Y_ZONE))
                return 0;                       /* big enough? */
        if (minx > 0)                           /* expand dimensions by one pixel */
                minx--;
        if (maxx < dx-1)
                maxx++;
        if (miny > 0)
                miny--;
        if (maxy < dy-1)
                maxy++;
        zone_ptr->x1 = minx;            /* save zone */
        zone_ptr->y1 = miny;
        zone_ptr->x2 = maxx;
        zone_ptr->y2 = maxy;
        return 1;
}


static void overlap_zones(ZONE *zone_array,int *array_size)
{
        int i,j,ax1,ay1,ax2,ay2,bx1,by1,bx2,by2;

        for (i = 0 ; i < *array_size-1 ; i++)
        {   /* compare each zone against the rest */
                indicator(" Overlap zones",j);
                ax1 = (zone_array+i)->x1;
                ay1 = (zone_array+i)->y1;
                ax2 = (zone_array+i)->x2;
                ay2 = (zone_array+i)->y2;
                for (j = i+1 ; j < *array_size ; j++)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        bx1 = (zone_array+j)->x1;
                        by1 = (zone_array+j)->y1;
                        bx2 = (zone_array+j)->x2;
                        by2 = (zone_array+j)->y2;
                        if ((bx1>=ax2) || (ax1>=bx2) || (by1>=ay2) || (ay1>=by2))
                                continue;           /* no overlap */
                        bx1 = min(bx1,ax1);     /* combine zones */
                        by1 = min(by1,ay1);
                        bx2 = max(bx2,ax2);
                        by2 = max(by2,ay2);
                        (zone_array+i)->x1 = bx1;
                        (zone_array+i)->y1 = by1;
                        (zone_array+i)->x2 = bx2;
                        (zone_array+i)->y2 = by2;
                        (*array_size)--;        /* new zone count */
                        memmove((char *)(zone_array+j),(char *)(zone_array+j+1),
                                sizeof(ZONE)*(*array_size-j));  /* shift array to remove zone */
                        i = -1;                 /* start all over */
                        break;
                }
        }
        return;
}

static void overlap_samples(ZONE *zone_array,int *array_size)
{
        int i,j,ax1,ay1,ax2,ay2,bx1,by1,bx2,by2;

        for (i = 0 ; i < *array_size-1 ; i++)
        {   /* compare each zone against the rest */
                ax1 = (zone_array+i)->x1;
                ay1 = (zone_array+i)->y1;
                ax2 = (zone_array+i)->x2;
                ay2 = (zone_array+i)->y2;
                /* expand zone down (y2) by one pixel */
                if(ay2 < img_length-1)
                        ay2++;
                for (j = i+1 ; j < *array_size ; j++)
                {
                        bx1 = (zone_array+j)->x1;
                        by1 = (zone_array+j)->y1;
                        bx2 = (zone_array+j)->x2;
                        by2 = (zone_array+j)->y2;
                        /* expand zone up (y1) by one pixel */
                        if(by1 > 0)
                                by1--;
                        if ((bx1>=ax2) || (ax1>=bx2) || (by1>=ay2) || (ay1>=by2))
                                continue;           /* no overlap */
                        bx1 = min(bx1,ax1);     /* combine zones */
                        by1 = min(by1,ay1);
                        bx2 = max(bx2,ax2);
                        by2 = max(by2,ay2);
                        (zone_array+i)->x1 = bx1;
                        (zone_array+i)->y1 = by1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        (zone_array+i)->x2 = bx2;
                        (zone_array+i)->y2 = by2;
                        (*array_size)--;        /* new zone count */
                        memmove((char *)(zone_array+j),(char *)(zone_array+j+1),
                                sizeof(ZONE)*(*array_size-j));  /* shift array to remove zone */
                        i = -1;                 /* start all over */
                        break;
                }
        }
        return;
}

static void record_zone(int n)
{
        int i;

        sample[n].zones = zone_count;
        sample[n].line_no = n * lines_per_buffer;
        for(i=0; i<zone_count; i++) {
                sample[n].zone_info[i].x1=zone_list[i].x1;
                sample[n].zone_info[i].x2=zone_list[i].x2;
                sample[n].zone_info[i].y1=zone_list[i].y1;
                sample[n].zone_info[i].y2=zone_list[i].y2;
        }
        return;
}


static void combine_samples(void)
{
        int i,j,k;

        for(i=1; i<num_sample; i++) {
                for(j=0; j<sample[i].zones; j++) {
                        sample[i].zone_info[j].y1 += sample[i].line_no;
                        sample[i].zone_info[j].y2 += sample[i].line_no;
                }
        }
        for(k=0,zone_count=0; k<num_sample; k++) {
                zone_count+=sample[k].zones;
        }
        i=0;
        do {
                for(j=0; j<num_sample; j++) {
                        for(k=0; k<sample[j].zones; k++) {
                                zone_list[i].x1=sample[j].zone_info[k].x1;
                                zone_list[i].x2=sample[j].zone_info[k].x2;
                                zone_list[i].y1=sample[j].zone_info[k].y1;
                                zone_list[i].y2=sample[j].zone_info[k].y2;
                                i++;
                        }
                }
        } while(i!=zone_count);
        return;
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <conio.h>
#include <graphics.h>
#include <string.h>
#include <alloc.h>
#include "zone.h"
#include "wtwg.h"

/* extern variables & prototypes */
extern ZONE zone_list[MAX_ZONES];
extern int zone_count;
extern int img_width;
extern WINDOW *zone_win;

extern int readheap(char *p,unsigned int n);
extern void beep(void);

/* local prototype */
int showzone(void);
static void dispzone(int n);

int showzone(void)
{
        int i,n,key,ch,number;
        char buff[30],**choices;

        if((choices=(char **)malloc(MAX_ZONES))==NULL) {
                werror(1,"Error allocate number of choices");
        }
        for(i=0; i<zone_count; i++) {
                if((choices[i]=(char *)malloc(13))==NULL)
                        werror(1,"Error allocate size of choices");
                sprintf(choices[i]," zone ... %2d",i);
        }
        choices[zone_count]=NULL;
        wopen(32,10,40,30,112,DOUBLE_BORDER,48,1);
        wtitle(" Detected Zone Image ");
        sprintf(buff,"\n Detected %2d zone(s).",zone_count);
        wputs(buff);
        do {
                wflush();
                wsetlocation(WLOC_ATWIN,10,5);
                n=wpicklist(" Select zone ",choices);
                if(choices[n]==NULL) {
                        break;
                }
                else {
                        dispzone(n);
                        number=n;
                }
        } while(key!=ESCAPE);
        free(choices);
        wabandon();
        return(number);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
static void dispzone(int n)
{
        register int i,j;
        int x1,x2,y1,y2;
        char *p,buff[20];
        int zonewidth,zonelength;
        float sx,sy,fact;
        int c,xp,yp,np;

        sprintf(buff," ZONE No. %2d ",n);
        wps(buff);

        x1=zone_list[n].x1;     /* column offset */
        x2=zone_list[n].x2;
        y1=zone_list[n].y1;     /* row offset */
        y2=zone_list[n].y2;
        zonewidth=x2-x1+1;
        zonelength=y2-y1+1;
        if((p=malloc(img_width))==NULL) {
                werror(1,"Error allocate p in display_zone()");
        }
        if(zonewidth > (39*8) || zonelength > (29*8)) {
                sx=(39*8) / (float)zonewidth;
                sy=(29*8) / (float)zonelength;
                fact=(sx<sy)? sx : sy;
                np=(fact<1)? 1 : fact+0.9-(float)1;

                wclear();
                for(j=0; j<zonelength; j++) {
                        readheap(p,j+y1);
                        yp=(float)j*fact+0.5;
                        for(i=0; i<zonewidth; i++) {
                                c=(p[i+x1] == 0xff)? WHITE : BLACK;
                                xp=(float)i*fact+0.5;
                                putpixel(xp,yp,c);
                        }
                }
        }
        else {
                wclear();
                for(j=0; j<zonelength; j++) {
                        readheap(p,j+y1);
                        for(i=0; i<zonewidth; i++) {
                                if(p[i+x1] == 0xff) putpixel(i,j,WHITE);
                                else putpixel(i,j,BLACK);
                        }
                }
        }
        beep();
        free(p);
        return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>
#include <conio.h>
#include <string.h>
#include "read.h"
#include "zone.h"
#include "wtwg.h"

/* extern prototypes */
extern void strmfe(char *new,char *old,char *ext);
extern int readheap(char *p,unsigned int n);
extern ZONE zone_list[MAX_ZONES];
extern int img_width;

/* local prototypes */
void savetif(char *name,int num);
int fputword(FILE *fp,int n);
int fputlong(FILE *fp,long n);
static void tif2disk(FILE *fp,int left,int top,int right,int bottom);
static int writetiftag(FILE *fp,int tag,int type,long length,long offset);
static void packmonoline(char *dest,char *source,int offset,int width);

FILE *outtiff;

void savetif(char *savename,int num)
{
        int x1,x2,y1,y2,key=0;
        char name[20];

        x1=zone_list[num].x1;
        x2=zone_list[num].x2;
        y1=zone_list[num].y1;
        y2=zone_list[num].y2;
        strmfe(name,savename,"TIF");
        strupr(name);
        if((outtiff=fopen(name,"wb")) != NULL) {
                tif2disk(outtiff,x1,y1,x2,y2);
                fclose(outtiff);
        }
        else {
                werror(5,"Error creating TIFF");
        }
        return;
}

/* write a TIFF file */
static void tif2disk(FILE *fp,int left,int top,int right,int bottom)
{
        char *p,*pr;
        long pos;
        int i,j,bytes,width,depth;

        width=right-left+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        depth=bottom-top+1;
        bytes=(width+7)/8;

        /* allocate a line buffer */
        if((p=malloc(img_width)) == NULL)
                werror(1,"Error allocate p in tif2disk");
        if((pr=malloc(bytes)) == NULL )
                werror(1,"Error allocate pr in tif2disk");

        /* write the header */
        fputword(fp,'II');      /* INTEL format */
        fputword(fp,42);        /* Version #42; Note byte reversal */
        fputlong(fp,8L);        /* Offset to first IFD */
        fputword(fp,10);        /* 10 tags in this directory */

        writetiftag(fp,NewSubFile,TIFFshort,1L,1L);
        writetiftag(fp,ImageWidth,TIFFshort,1L,(long)width);
        writetiftag(fp,ImageLength,TIFFshort,1L,(long)depth);
        writetiftag(fp,BitsPerSample,TIFFshort,1L,1L);
        writetiftag(fp,Compression,TIFFshort,1L,1L);
        writetiftag(fp,PhotometricInterp,TIFFshort,1L,1L);
        writetiftag(fp,Threshholding,TIFFshort,1L,1L);
        writetiftag(fp,StripOffsets,TIFFlong,1L,150L);
        writetiftag(fp,SamplesPerPixel,TIFFshort,1L,1L);
        writetiftag(fp,RowsPerStrip,TIFFlong,1L,(long)depth);

        fputlong(fp,0L);        /* No more Image File Directories */

        /* Extended tag information, X-Y resolution  */
        for(i=0; i<4; i++)
                fputlong(fp,0L);

        /* write the bitmap */
        for(i=0;i<depth;++i) {
                readheap(p,i+top);
                packmonoline(pr,p,left,width);
                fwrite(pr,1,bytes,fp);
        }
        free(pr);
        free(p);
        putch(0x07);
        return;
}

/* write one TIFF tag to the IFD */
static int writetiftag(FILE *fp,int tag,int type,long length,long offset)
{
        fputword(fp,tag);
        fputword(fp,type);
        fputlong(fp,length);
        fputlong(fp,offset);
        return(ferror(fp));
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int fputword(FILE *fp,int n)
{
        fputc(n,fp);
        fputc((n >> 8),fp);
        return(ferror(fp));
}

int fputlong(FILE *fp,long n)
{
        fputc(n,fp);
        fputc((n >> 8),fp);
        fputc((n >> 16),fp);
        fputc((n >> 24),fp);
        return(ferror(fp));
}

void packmonoline(char *dest,char *source,int offset,int width)
{
        int i;
        char masktable[8]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};

        for(i=0;i<width;++i) {
                if(source[i+offset]==0x00)
                        dest[i>>3] |= masktable[i&0x0007];
                else if(source[i+offset]==0xff)
                        dest[i>>3] &= ~masktable[i&0x0007];
        }
        return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>
#include <conio.h>
#include <mem.h>
#include <dos.h>
#include <string.h>
#include <graphics.h>
#include "read.h"
#include "zone.h"
#include "chr.h"
#include "wtwg.h"

/* extern variables & prototypes */
extern ZONE zone_list[MAX_ZONES];
extern int zone_count;
extern int img_width;

extern int readheap(char *p,unsigned int n);
extern void display_char(char *ch,int cx,int cy);
extern char searchcode(char *ch,int cx,int cy,char *codecg);
extern int get_kb(void);
extern void beep(void);
extern void sort_pos(CHAR_INFO *st,int index);

void read_zone(int n);
static void raster_scan(char *image,int dx,int dy);
static void extract(char *image,int dx,int dy,int x,int y);
static void reccharinfo(int width,int height,int x,int y,char *codecg,char chrtype);
static char qcode(char *p,int cx,int cy,char *codecg);

FILE *txt; /* output text file pointer */
WINDOW *char_win,*recog_win;
unsigned int index; /* number of char segmented */
CHAR_INFO st[100]; /* array stores chars recognized */
char *msg="\n\
          \n  Zone image too big to fit memory.\n\
          \n  Save this zone and continue later.\n\
          \n\n";

void read_zone(int n)
{
        int i,j;
        int x1,x2,y1,y2;
        char *p,*ptr;
        int zonewidth,zoneheight;
        unsigned long mem_left;

        x1=zone_list[n].x1;     /* column offset */
        x2=zone_list[n].x2;
        y1=zone_list[n].y1;     /* row offset */
        y2=zone_list[n].y2;
        zonewidth=x2-x1+1;
        zoneheight=y2-y1+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mem_left=farcoreleft();
        if(mem_left < (long)zonewidth * (long)zoneheight) {
                beep();
                wpromptc(NULL,msg,NULL);
                return;
        }
        else {
                if((ptr=malloc(zonewidth * zoneheight))==NULL)
                        werror(1,"Error allocate image in read_zone()");
                if((p=malloc(img_width))==NULL)
                        werror(1,"Error allocate p in read_zone()");

                for(i=0; i<zoneheight; i++) {
                        readheap(p,i+y1);
                        for(j=0; j<zonewidth; j++) {
                                ptr[i*zonewidth+j]=p[j+x1];
                        }
                }
                free(p);
                raster_scan(ptr,zonewidth,zoneheight);
                free(ptr);
                wreopen(char_win); wclose();
                sort_pos(st,index);
                wreopen(recog_win); wabandon();
                return;
        }
}

static void raster_scan(char *image,int dx,int dy)
{
        int x,y;
        char *ptr;

        wopen(8,10,20,20,112,DOUBLE_BORDER,48,1);
        wtitle(" Character ");
        char_win=w0;

        wopen(8,44,64,6,112,DOUBLE_BORDER,95,1);
        wtitle(" Recognized Output ");
        recog_win=w0;

        index=0;
        for(y=0; y<dy; y++) {
                ptr = image + y * dx;
                for(x=0; x<dx; x++) {
                        if(*(ptr+x)) { /* found point */
                                extract(image,dx,dy,x,y);
                        }
                }
        }
        return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
static void extract(char *image,int dx,int dy,int x,int y)
{
        int ix,iy,minx,miny,maxx,maxy;  /* perimeter variables & min/max values */
        HEADING dir;                    /* current direction */
        char *previous,*next,*here;     /* buffer pointers */
        int i,j,cx,cy;
        int ii,jj;
        char *ch,code[9],type;

        minx = maxx = ix = x;           /* preset min/max x/y and perimeter vars */
        miny = maxy = iy = y;
        dir  = GOING_UP;                /* starting direction */

        /* walk perimeter, recording min/max of rectangular region */
        do {
                /* update min/max */
                if (ix < minx)
                        minx = ix;
                if (ix > maxx)
                        maxx = ix;
                if (iy < miny)
                        miny = iy;
                if (iy > maxy)
                        maxy = iy;
                here = image + iy * dx + ix;    /* where are we? */
                next = here + dx;
                previous = here - dx;
                switch (dir) {  /* based on current direction, */
                        /* look around for next direction */
                        case GOING_UP:
                                if (ix > 0 && *(here-1)) {
                                        ix--;
                                        dir = GOING_LEFT;
                                        break;
                                }
                                if (iy > 0 && *previous) {
                                        iy--;
                                        break;
                                }
                                if (ix < dx-1 && *(here+1)) {
                                        ix++;
                                        dir = GOING_RIGHT;
                                        break;
                                }
                                if (iy < dy-1 && *next) {
                                        iy++;
                                        dir = GOING_DOWN;
                                        break;
                                }
                                break;
                        case GOING_RIGHT:
                                if (iy > 0 && *previous) {
                                        iy--;
                                        dir = GOING_UP;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
        }
        if (ix < dx-1 && *(here+1)) {
                ix++;
                break;
        }
        if (iy < dy-1 && *next) {
                iy++;
                dir = GOING_DOWN;
                break;
        }
        if (ix > 0 && *(here-1)) {
                ix--;
                dir = GOING_LEFT;
                break;
        }
        break;
case GOING_DOWN:
        if (ix < dx-1 && *(here+1)) {
                ix++;
                dir = GOING_RIGHT;
                break;
        }
        if (iy < dy-1 && *next) {
                iy++;
                break;
        }
        if (ix > 0 && *(here-1)) {
                ix--;
                dir = GOING_LEFT;
                break;
        }
        if (iy > 0 && *previous) {
                iy--;
                dir = GOING_UP;
                break;
        }
        break;
case GOING_LEFT:
        if (iy < dy-1 && *next) {
                iy++;
                dir = GOING_DOWN;
                break;
        }
        if (ix > 0 && *(here-1)) {
                ix--;
                break;
        }
        if (iy > 0 && *previous) {
                iy--;
                dir = GOING_UP;
                break;
        }
        if (ix < dx-1 && *(here+1)) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        ix++;
                                        dir = GOING_RIGHT;
                                        break;
                                }
                                break;
                        }
                } while(ix != x || iy != y); /* until we return to the start */
                /* big enough? */
                if((maxx-minx+1 < 5) || (maxy-miny+1 < 5)) {
                        here=image+miny*dx+minx;
                        for(iy=miny; iy<=maxy; iy++,here+=dx) {
                                memset(here,WHITE,maxx-minx+1);
                        }
                }
                else {
                        cx=maxx-minx+1;
                        cy=maxy-miny+1;
                        if((ch=malloc(cx*cy))==NULL) {
                                werror(1,"Error allocate ch in extract_char()");
                        }
                        /* write buffer & white out the character */
                        for(j=0,jj=miny; j<cy; j++,jj++) {
                                for(i=0,ii=minx; i<cx; i++,ii++) {
                                        ch[j*cx+i] = image[jj*dx+ii];
                                }
                        }
                        here=image+miny*dx+minx;
                        for(i=miny; i<maxy; i++,here+=dx) {
                                memset(here,WHITE,cx);
                        }
                        display_char(ch,cx,cy);
                        type=qcode(ch,cx,cy,code);
                        reccharinfo(cx,cy,x,y,code,type);
                        free(ch);

                }
                return;
}


static void reccharinfo(int width,int height,int xpos,int ypos,char *codecg,char chrtype)
{
        int i;

        st[index].width = width;
        st[index].height = height;
        st[index].x = xpos;
        st[index].y = ypos;
        for(i=0; i<9; i++)
                st[index].c[i] = codecg[i];
        st[index].type = chrtype;
        index++;
        return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
static char qcode(char *p,int cx,int cy,char *codecg)
{
        int i,key_status,x,y;
        char c,chr,ch;
        FILE *code_file;

        chr=searchcode(p,cx,cy,codecg);
        if(chr==0x00) {
                beep();
                if((code_file=fopen("qcode.cod","a+t"))==NULL)
                        werror(0,"Error appending qcode.cod");
                wopen(8,34,20,6,112,DOUBLE_BORDER,31,1);
                wtitle(" Learn ");
                for(i=0; i<9; i++)
                        fputc(codecg[i],code_file);
                wprintf("\n Qcode:= %s\n",codecg);
                wprintf("\n Identify as:= ");
                x=wherex();
                y=wherey();
                do {
                        wps(" Thai ");
                        c=get_kb();
                        wgoto(x,y);
                        switch(c) {
                                case 0 :
                                        wps(" Eng ");
                                        wflush();
                                        c=wgetc();
                                        if(c!=0x29) wputc(c);
                                        break;
                                default :
                                        wputc(c);
                                        break;
                        }
                        wflush();
                        ch=wgetc(); /* wait for enter key */
                        if(ch==BACKSPACE) {
                                wgoto(x,y);
                                wputc(0xff);
                        }
                } while(!(ch==ENTER));
                for(i=0;i<7;i++)
                        fputc(0x20,code_file);
                fputc(c,code_file);
                fprintf(code_file,"\n");
                wclose();
                fclose(code_file);
                chr=c;
        }
        else {
                wreopen(recog_win);
                wputc(chr);
        }
        return chr;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <string.h>
#include "wtwg.h"

/* extern prototype */
extern void binary(char *p,int width,int height);
extern void div_tchar(void);
extern void get_code(char *code);

/* local prototype */
char searchcode(char *p,int width,int height,char *code);

char searchcode(char *p,int width,int height,char *code)
{
        char type,buf[9],dummy;
        FILE *cfp;
        char ch;
        int i,flag=0;

        if((cfp=fopen("qcode.cod","a+b"))==NULL)
                werror(1,"Error opening qcode.cod");
        rewind(cfp);
        binary(p,width,height);
        div_tchar();
        get_code(code);
        do {
                for(i=0; i<9; i++)
                        buf[i]=getc(cfp);       /* qcode from dictionary */
                for(i=0; i<7; i++)
                        dummy=getc(cfp);        /* skip tab 0x20 */
                type=getc(cfp);
                dummy=getc(cfp);        /* get "0D" */
                dummy=getc(cfp);        /* get "0A" */
                if(!(memcmp(buf,code,9))) {     /* matched */
                        flag=1;
                }
                else {
                        if(feof(cfp)) { /* end of file */
                                type=0x00;
                                flag=1;
                        }
                }
        } while(!flag);
        fclose(cfp);
        return type;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <math.h>
#include <alloc.h>
#include <conio.h>
#include <stdlib.h>
#include "wtwg.h"

/* local prototype */
void binary(char *p,int dx,int dy);
void div_tchar(void);
void get_code(char *buf);
static char cal_sub_cg(int line,int col,char *p,int q);

static char *table;
static wide,deep;
static char code[10];

void binary(char *p,int dx,int dy)
{
        int i,j;

        if((table=malloc(dx*dy))==NULL)
                werror(0,"Error allocate table in binary()");
        for(j=0; j<dy; j++) {
                for(i=0; i<dx; i++) {
                        if(p[j*dx+i] == 0xff) table[j*dx+i] = '1';
                        else table[j*dx+i] = '0';
                }
        }
        wide=dx;
        deep=dy;
        return;
}

void div_tchar(void)
{
        char *subtable,*ptr=table;
        int col[3],line[3];
        int r,c,k,v,y,z;
        int quadrant;

        col[0] = col[1]  = wide/2;
        line[0] = line[1] = deep/2;
        if((int)fmod(wide,2.0))
                col[1]++;
        if((int)fmod(deep,2.0))
                line[1]++;
        col[2] = col[0] + col[1];
        line[2] = line[0] + line[1];
        for(r=0; r<3; r++) {
                for(c=0; c<3; c++) {
                        switch(r) {
                                case 1:ptr += wide*line[0]; break;
                        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        switch(c) {
                                case 1:ptr += col[0]; break;
                        }
                        z = line[r]*col[c];
                        subtable = malloc(z);
                        y=0;
                        for(k=0; k<line[r]; k++) {
                                for(v=0; v<col[c]; v++,ptr++) {
                                        subtable[y++] = *ptr;
                                }
                                switch(c) {
                                        case 0: ptr += col[1]; break;
                                        case 1: ptr += col[0]; break;
                                }
                        }
                        quadrant = 3*r+c;
                        code[quadrant] = cal_sub_cg(line[r],col[c],subtable,quadrant);
                        free(subtable);
                        ptr=table;
                }
        }
        code[9]='\0';
        free(table);
        return;
}

static char cal_sub_cg(int line,int col,char *p,int q)
{
        int total=0,cgx,cgy,direct,n,flag;
        int r,c,x,y;
        char *ptr=p,*pr,*p1=p;
        long tempx=0,tempy=0;

        for(r=0; r<line; r++) {
                for(c=0; c<col; c++,ptr++) {
                        if((*ptr)=='1') {
                        total++;        /* sum number of '1' found on quadrant */
                        tempx+=c;       /* sum coordinate of '1' on column */
                        tempy+=r;   /* sum coordinate of '1' on row */
                        }
                }
        }
        if(total!=0) {
                cgx=tempx/total;
                if(fmod(tempx,total)>=total/2)
                        cgx++;
                cgy=tempy/total;
                if(fmod(tempy,total)>=total/2)
                        cgy++;
        }
        else {
                cgx=col/2;
                if(fmod(col,2.0)==1)
                        cgx++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                cgy=line/2;
                if(fmod(line,2.0)==1)
                        cgy++;
                cgx--; cgy--;
        }
        p += cgy*col+cgx;
        if(*p=='1') {   /* adjust CG to point '0' */
                x=cgx; y=cgy;
                switch(q) {
                        case 0:
                        case 1:
                        case 3:
                        case 4:
                        case 6:
                        case 7: n=1; flag=0;
                                do {
                                        for(direct=0,pr=p; direct<2; direct++) {
                                                switch(direct) {
                                                        case 0: if(flag) break;
                                                                if((x=cgx+n) > col-1) break;
                                                                else {
                                                                        pr=p1; /* reset */
                                                                        pr+=y*col+x;
                                                                        if(*pr=='0') {
                                                                                if(*(pr+col)=='1' ||
*(pr-col)=='1') break;
                                                                                else {
                                                                                        cgx=x;
                                                                                        flag=1;
                                                                                        break;
                                                                                }
                                                                        }
                                                                } /* end else */
                                                        case 1: if(flag) break;
                                                                if((x=cgx-n) < 0) break;
                                                                else {
                                                                        pr=p1; /* reset */
                                                                        pr+=y*col+x;
                                                                        if(*pr=='0') {
                                                                                if(*(pr+col)=='1' ||
*(pr-col)=='1') break;
                                                                                else {
                                                                                        cgx=x;
                                                                                        flag=1;
                                                                                        break;
                                                                                }
                                                                        }
                                                                } /* end else */
                                                } /* end switch direction */
                                        }
                                        if(cgx+n>col-1 && cgx-n<0) flag=1;
                                        else n++;
                                } while(!flag);
                                break; /* end case q=0,1,3,4,6,7 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 2:
                case 5: n=1; flag=0;
                        do {
                                for(direct=0,pr=p; direct<2; direct++) {
                                        switch(direct) {
                                                case 0: if(flag) break;
                                                        if((y=cgy-n) < 0) break;
                                                        else {
                                                                pr=p1; /* reset */
                                                                pr+=y*col+x;
                                                                if(*pr=='0') {
                                                                        if(*(pr+x)=='1' || *
(pr-x)=='1') break;
                                                                        else {
                                                                                cgy=y;
                                                                                flag=1;
                                                                                break;
                                                                        }
                                                                }
                                                        } /* end else */
                                                case 1: if(flag) break;
                                                        if((y=cgy+n) > line-1) break;
                                                        else {
                                                                pr=p1; /* reset */
                                                                pr+=y*col+x;
                                                                if(*pr=='0') {
                                                                        if(*(pr+x)=='1' || *
(pr-x)=='1') break;
                                                                        else {
                                                                                cgy=y;
                                                                                flag=1;
                                                                                break;
                                                                        }
                                                                }
                                                        } /* end else */
                                        } /* end switch direction */
                                }
                                if(cgy-n<0 && cgy+n>line-1) flag=1;
                                else n++;
                        } while(!flag);
                        break; /* end case q=2,5 */
                case 8: n=1; flag=0;
                        do {
                                for(direct=0,pr=p; direct<4; direct++) {
                                        switch(direct) {
                                                case 0: if(flag) break;
                                                        if((x=cgx+n) > col-1) break;
                                                        else {
                                                                pr=p1; /* reset */
                                                                pr+=y*col+x;
                                                                if(*pr=='0') {
                                                                        cgx=x;
                                                                        flag=1;
                                                                        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        }
                                } /* end else */
                        case 1: if(flag) break;
                                if((y=cgy-n) < 0) break;
                                else {
                                        pr=p1; /* reset */
                                        pr+=y*col+x;
                                        if(*pr=='0') {
                                                cgy=y;
                                                flag=1;
                                                break;
                                        }
                                } /* end else */
                        case 2: if(flag) break;
                                if((x=cgx-n) < 0) break;
                                else {
                                        pr=p1; /* reset */
                                        pr+=y*col+x;
                                        if(*pr=='0') {
                                                cgx=x;
                                                flag=1;
                                                break;
                                        }
                                } /* end else */
                        case 3: if(flag) break;
                                if((y=cgy+n) > line-1) break;
                                else {
                                        pr=p1; /* reset */
                                        pr+=y*col+x;
                                        if(*pr=='0') {
                                                cgy=y;
                                                flag=1;
                                                break;
                                        }
                                } /* end else */
                                } /* end switch direction */
                        }
                        if(cgx+n>col-1 && cgx-n<0 && cgy-n<0 && cgy+n>line-1)
                                flag=1;
                        else n++;
                } while(!flag);
                break; /* end case q=8 */
        } /* end switch q */
        p=pr; /* update CG point */
}
/* calculate Q-code at CG */
tempy=0; tempx=0; ptr = p;
ptr++;
for(c=cgx+1; c<col && tempy==0; c++,ptr++) {
        if(*ptr=='1') {
                tempx+=8;
                tempy++;
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        tempy=0; ptr = p;
        ptr-=col;
        for(c=cgy-1; c>=0 && tempy==0; c--,ptr-=col) {
                if(*ptr=='1') {
                        tempx+=4;
                        tempy++;
                }
        }
        tempy=0; ptr = p;
        ptr--;
        for(c=cgx-1; c>=0 && tempy==0; c--,ptr--) {
                if(*ptr=='1') {
                        tempx+=2;
                        tempy++;
                }
        }
        tempy=0; ptr = p;
        ptr+=col;
        for(c=cgy+1; c<line && tempy==0; c++,ptr+=col) {
                if(*ptr=='1') {
                        tempx+=1;
                        tempy++;
                }
        }
        return (char)tempx;
}

void get_code(char *buf)
{
        int c;

        for(c=0; c<9; c++) {
                switch(code[c]) {
                        case 10:                        /* code = 'A' */
                        case 11:                        /* code = 'B' */
                        case 12:                        /* code = 'C' */
                        case 13:                        /* code = 'D' */
                        case 14:                        /* code = 'E' */
                        case 15:
                                buf[c] = code[c]+55;    /* code = 'F' */
                                break;
                        default:
                                buf[c] = code[c]+48;    /* code = '1'-'9' */
                                break;
                }
        }
        return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <bios.h>
#include <dos.h>

#define RIGHT  0x01
#define LEFT   0x02

typedef unsigned char       BYTE;
typedef unsigned int        WORD;

#define LOWORD(l)           ((WORD)(l))
#define HIWORD(l)           ((WORD)(((DWORD)(l) >> 16) & 0xFFFF))
#define LOBYTE(w)           ((BYTE)(w))
#define HIBYTE(w)           ((BYTE)(((WORD)(w) >> 8) & 0xFF))

static char thai_kb[0x54] = {
        0x00,0x1B,0x00,0x2F,0x5F,0xC0,0xB6,0xD8,0xD6,0xA4,0xB5,0xA8,0xA2,0xAA,0x08,0x09,
        0xE6,0xE4,0xD3,0xBE,0xD0,0xD1,0xD5,0xC3,0xB9,0xC2,0xBA,0xC5,0x0D,0x00,0xBF,0xCB,
        0xA1,0xB4,0xE0,0xE9,0xE8,0xD2,0xCA,0xC7,0xA7,0x00,0x00,0x60,0xBC,0xBB,0xE1,0xCD,
        0xD4,0xD7,0xB7,0xC1,0xE3,0xBD,0x00,0x2A,0x00,0x20,0x00,0x00,0x00,0x00,0x00,0x00,
        0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x37,0x38,0x39,0x2D,0x34,0x35,0x36,0x2B,0x31,
        0x32,0x33,0x30,0x2E};

static char thai_kbs[0x54] = {
        0x00,0x1B,0x00,0xF1,0xF2,0xF3,0xF4,0xD9,0xDB,0xF5,0xF6,0xF7,0xF8,0xF9,0x08,0x09,
        0xF0,0x22,0xAE,0xB1,0xB8,0xEB,0x5A,0xB3,0xCF,0xAD,0xB0,0x2C,0x0D,0x00,0xC4,0xA6,
        0xAF,0xE2,0xAC,0xE7,0xEB,0xC9,0xC8,0xAB,0x2E,0x00,0x00,0x7E,0x28,0x29,0xA9,0xCE,
        0x25,0xEC,0x3F,0xB2,0xCC,0x3F,0x00,0x2A,0x00,0x20,0x00,0x00,0x00,0x00,0x00,0x00,
        0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xF7,0xF8,0xF9,0x2D,0xF4,0xF5,0xF6,0x2B,0xF1,
        0xF2,0xF3,0xF0,0x2E};

/* prototype */
int get_kb(void);
void beep(void);

int get_kb(void)
{
        int key,modifiers;
        int thai_key=1,eng_key=0;
        int last_key=0;

        key=bioskey(0);
        if(HIBYTE(key)==0x29 && !eng_key) {
                thai_key=!thai_key;
                return 0;
        }
        if(LOBYTE(key)!=0) {
                if(thai_key && (LOBYTE(key)>32)) {
                        modifiers=bioskey(2);
                        if((modifiers & RIGHT) || (modifiers & LEFT)) {
                                return(last_key=thai_kbs[HIBYTE(key)]);
                        }
                        return(last_key=thai_kb[HIBYTE(key)]);
                }
                else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        return(last_key=LOBYTE(key));
        }
        return(last_key=key);
}

void beep(void)
{
        long tm;

        sound(2000);
        tm=biostime(0,0L);
        tm+=1;
        while(biostime(0,0L)<=tm);
        nosound();
        return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include "chr.h"
#include "wtwg.h"

/* Extern variable */
extern WINDOW *recog_win;

/* Prototype */
void sort_pos(CHAR_INFO *st,int count);

void sort_pos(CHAR_INFO *st,int count)
{
        int x,y,i,j,index,fudge_x;
        CHAR_INFO temp;
        FILE *outtxt;

        if((outtxt=fopen("text.txt","at"))==NULL)
                werror(1,"Error creating text.txt");
        for(i=0; i<count-1; i++) {
                /* sort on x */
                for(j=i+1; j<count; j++) {
                        if(st[j].x < st[i].x) {
                                temp = st[i];
                                st[i] = st[j];
                                st[j] = temp;
                        }
                }
        }
        for(i=0; i<count-1; i++) {
                /* order of char: left to right, up to down */
                x = st[i].x;
                y = st[i].y;
                /* 5% slippage on alignment */
                fudge_x = st[i].x+st[i].width/20;
                for(j=i+1, index=-1; j<count; j++) {
                        /* find any chars above and within x extent of st[i] */
                        if(st[j].x > x)
                                break;
                        if(st[j].y < y && st[j].x <= fudge_x) {
                                x = st[j].x;
                                y = st[j].y;
                                index = j;
                        }
                }
                if(index != -1) {
                        temp = st[i];
                        st[i] = st[index];
                        st[index] = temp;
                }
        }
        wclear();
        for(i=0; i<count; i++) {
                wprintf("%c",st[i].type);
                fputc(st[i].type,outtxt);

        }
        fprintf(outtxt,"\n");
        fclose(outtxt);
        return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define GOOD_READ               0       /* return codes */
#define GOOD_WRITE              0
#define BAD_FILE                1
#define BAD_WRITE               1
#define BAD_READ                2
#define MEMORY_ERROR            3
#define WRONG_BITS              4

#define SCREENWIDE              640     /* mode 11,2-color VGA screen dimensions */
#define SCREENDEEP              480
#define STEP                    64      /* size of a step when panning */

#define HOME                    0x4700  /* cursor control codes */
#define CURSOR_UP               0x4800
#define CURSOR_LEFT             0x4b00
#define CURSOR_RIGHT            0x4d00
#define END                     0x4f00
#define CURSOR_DOWN             0x5000

/* TIFF object sizes */
#define TIFFbyte                1
#define TIFFascii               2
#define TIFFshort               3
#define TIFFlong                4
#define TIFFrational            5

/* TIFF tag names */
#define NewSubFile              254
#define SubfileType             255
#define ImageWidth              256
#define ImageLength             257
#define RowsPerStrip            278
#define StripOffsets            273
#define StripByteCounts         279
#define SamplesPerPixel         277
#define BitsPerSample           258
#define Compression             259
#define PlanarConfiguration     284
#define Group3Options           292
#define Group4Options           293
#define FillOrder               266
#define Threshholding           263
#define CellWidth               264
#define CellLength              265
#define MinSampleValue          280
#define MaxSampleValue          281
#define PhotometricInterp       262
#define GrayResponseUnit        290
#define GrayResponseCurve       291
#define ColorResponseUnit       300
#define ColorResponseCurves     301
#define XResolution             282
#define YResolution             283
#define ResolutionUnit          296
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define Orientation             274
#define DocumentName            269
#define PageName                285
#define XPosition               286
#define YPosition               287
#define PageNumber              297
#define ImageDescription        270
#define Make                    271
#define Model                   272
#define FreeOffsets             288
#define FreeByteCounts          289
#define ColorMap                320
#define Artist                  315
#define DateTime                306
#define HostComputer            316
#define ImageDescription        270
#define Software                305

#define pixels2bytes(n)         ((n+7)/8)

typedef struct {
        unsigned int width,depth,bytes,bits;
        unsigned int flags;
        unsigned int rowsperstrip;
        unsigned int count;
        unsigned int samples;
        unsigned int threshhold;
        unsigned int compression;
        unsigned int bitspersample;
        unsigned long bytecount;
        unsigned long offset;
        int (*setup)();
        int (*closedown)();
} FILEINFO;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* definitions */
#define MAX_MALLOC 0xffee       /* maximum malloc size */
#define BUFFER_SIZE 0xfe00

        /* blocking */
#define WHITE 0x00              /* white color in sample */
#define BLACK 0xff              /* black color in sample */
#define PREGNANT 0x01           /* ready to be black */
#define SICK 0xfe               /* ready to be white */
#define ALIVE 0x80              /* still alive */
#define LIFE_SCORE 2            /* neighbors for birth */
#define DEATH_SCORE 0           /* neighbors for death */

        /* sequencing */
#define MIN_X_ZONE 1            /* minimum allowable x extent for a zone */
#define MIN_Y_ZONE 1            /* minimum allowable y extent for a zone */
#define MAX_ZONES 200           /* maximum number of zones */
#define COLUMN_MAJOR 1          /* order zones with columns dominate */
#define ROW_MAJOR 2             /* order zones with rows dominate */

typedef struct { int x1,y1,x2,y2; } ZONE;
typedef enum { GOING_UP,GOING_RIGHT,GOING_DOWN,GOING_LEFT } HEADING;
typedef struct { short x,y,width,height; } DISPLAY_BOX;
typedef struct {
        ZONE zone_info[MAX_ZONES];
        int zones;
        int line_no;
} SAMPLEINFO;
```



```
typedef struct {
        int width,height;
        int x,y;
        char c[9];
        char type;
} CHAR_INFO;
```

หนังสืออ้างอิง
ห้ามนำออกนอกห้องสมุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้