การจดจำอักษรภาษาไทยลายมือเขียน

(THAI CHARACTER RECOGNITION)

โดย

น.ส.ศิริลักษณ์ เกตุคุ้ม 34107376

น.ส.สายใจ จันทร์มณี 34108417

วัน เดือน ปี 14 ม.ค. 2539
เลขทะเบียน 034419
เลขเรียกหนังสือ T387019 ศ6

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมอิเลคทรอนิคส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปริญญานิพนธ์สำหรับภาคการศึกษาที่ 2 ปีการศึกษา 2537

ภาควิชาวิศวกรรมศาสตร์อิเลคทรอนิคส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การจดจำตัวอักษรภาษาไทยลายมือเขียน (THAI CHARACTER RECOGNITION)

ผู้จัดทำ

น.ส.ศิริลักษณ์ เกตุคุ้ม เลขประจำตัว 34107376

น.ส.สายใจ จันทร์มณี เลขประจำตัว 34108417

อาจารย์ที่ปรึกษา

(.....)

อ.เทอดศักดิ์ ลิ่วหาทอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นเรื่องเกี่ยวกับการประมวลผลสัญญาณภาพ โดยใช้ Microcomputer เป็นตัววิเคราะห์และประมวลผล ซึ่งในที่นี้เป็นการประมวลสัญญาณภาพเกี่ยวกับการจดจำตัวอักษรภาษาไทยลายมือเขียน (Thai Character Recognition) เพื่อเป็นการพัฒนาระบบการป้อนข้อมูลเข้าสู่ computer โดยเฉพาะตัวอักษรที่เป็นลายมือเขียนโดยไม่ต้องป้อน ข้อมูลผ่านแป้นพิมพ์หรือ อ่านจากแฟ้มข้อมูลในหน่วยความจำ แต่จะป้อนข้อมูลโดยการ scan ตัวอักษรบนกระดาษเข้าสู่ computer โดยตรง และ ในการจดจำตัวอักษรภาษาไทยลายมือเขียน จะนำเอาลักษณะเด่นคือหัวของตัวอักษรมาใช้ในการจำแนกกลุ่มของตัวอักษร สำหรับโปรแกรมที่ใช้ในการประมวลผลจะเป็น program ที่เขียนด้วยภาษาซี

Abstract

This project concerns with the image processing which uses microcomputer to be analyzer and processor. The scope of this project is the hand writing thai character recognition for input data (hand writing thai character) to computer.

Hand writing thai character recognition will use imminent characteristic,that is the head of character, to classify group of character. Programe that is used in this project is written in C language.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

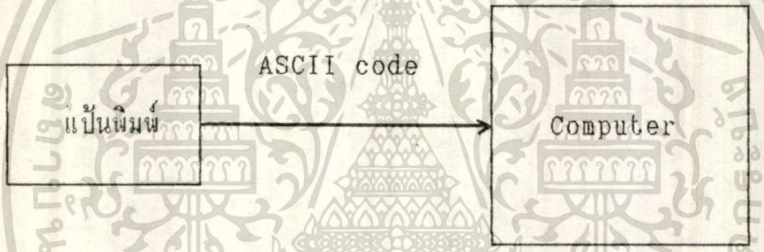
สารบัญ

บทที่	ชื่อเรื่อง	หน้า
	บทนำ	
1	ขั้นตอนการจดจำรูปแบบตัวอักษรภาษาไทย	1
2	ทฤษฎีในการจดจำตัวอักษรภาษาไทยลายมือเขียน	4
3	การเตรียมข้อมูลเพื่อการจำแนกกลุ่มของตัวอักษร	12
4	การตรวจสอบหัวและการแบ่งกลุ่มตัวอักษร	24
5	ผลการทดลองและสรุป	34
	บรรณานุกรม	40
	กิตติกรรมประกาศ	41
ภาคผนวก		
	ตัวอย่างโปรแกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ปัจจุบันนี้ Microcomputerได้เข้ามามีบทบาทในงานสาขาต่างๆ ของมนุษย์มากมาย เช่น การใช้ Computer เก็บและวิเคราะห์ข้อมูลทางด้านธุรกิจ คำนวณทางวิศวกรรมศาสตร์ เป็นต้น งานทุกอย่างที่มี Computer มาเกี่ยวข้องจะต้องมีการปฏิบัติการสิ่งหนึ่งเกิดขึ้นเสมอๆ สิ่งนั้นคือ การป้อนข้อมูลเข้า อุปกรณ์ในการป้อนข้อมูลที่มีการนำมาใช้กันมากกับเครื่อง Microcomputer ก็คือ แป้นพิมพ์ การป้อนข้อมูลโดยใช้แป้นพิมพ์นั้นเป็นการส่งรหัสให้แก่เครื่อง Computer เช่น ASCII code ของตัวอักษรต่างๆ รวมทั้งตัวอักษรภาษาไทยก็ได้มีการกำหนดรหัสแทนตัวอักษรขึ้นมาใช้กันหลายๆ แบบ

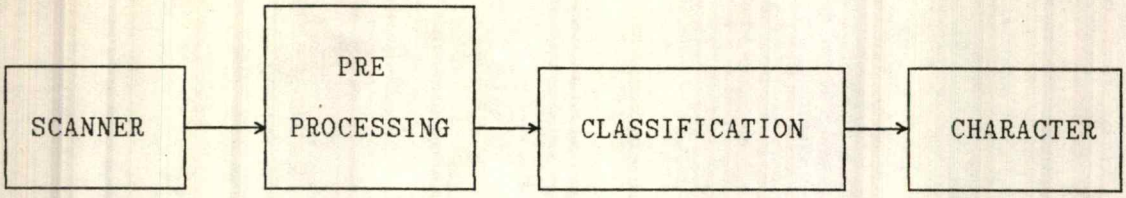


รูปที่ 1 แสดง Block Diagram ของการป้อนข้อมูลเข้าด้วยการใช้แป้นพิมพ์

จากรูปที่ 1 จะเห็นว่า Computer ไม่รับรู้ลักษณะของตัวอักษรที่ถูกพิมพ์เข้าไป แต่จะรับรู้เพียงรหัสแทนตัวอักษรเท่านั้น ดังนั้น จึงทำให้เกิดความยุ่งยากที่จะทำให้ Computer สามารถเข้าใจถึงลักษณะของตัวอักษรแบบต่างๆ ที่มนุษย์ใช้กันอยู่

ที่วิธีการอย่างหนึ่งในการป้อนข้อมูลเข้าสู่ Computer ได้ก็คือ การป้อนข้อมูลเข้าเครื่อง Computer โดยการใช้เครื่อง Scanner เมื่อรูปแบบของตัวอักษรได้ถูกอ่านเข้ามาด้วยเครื่อง Scanner แล้ว จะนำไปวิเคราะห์ออกมาว่า เป็นตัวอักษรตัวใดดังแสดงในรูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 แสดงการเก็บรูปแบบของตัวอักษร

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอวิธีในการจดจำตัวอักษรภาษาไทยลายมือเขียน โดยแบ่งเนื้อหาออกเป็นบทๆ ดังต่อไปนี้

บทที่ 1 เป็นการกล่าวถึงวัตถุประสงค์ และ ขั้นตอนในการจดจำตัวอักษรภาษาไทยลายมือเขียน

บทที่ 2 กล่าวถึงทฤษฎีที่นำมาในการจดจำตัวอักษรภาษาไทยลายมือเขียน เช่น การหาค่าตัวเลขต่อเนื่อง การหาจุดศูนย์ถ่วง การหาขอบของภาพ

บทที่ 3 เป็นการเตรียมข้อมูลเพื่อการจำแนกกลุ่มของตัวอักษร มีวิธีการหาตำแหน่งมาตรฐาน การทำให้บาง เพื่อลดจำนวนข้อมูลภาพ เป็นต้น

บทที่ 4 กล่าวถึงการตรวจสอบหัว และการแบ่งกลุ่มตัวอักษร

บทที่ 5 เป็นการแสดงผลการทดสอบ และสรุปผลการทดลองการจดจำตัวอักษรภาษาไทยลายมือเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ขั้นตอนการจดจำรูปแบบตัวอักษรภาษาไทย

ปริญญานิพนธ์เล่มนี้เป็นเรื่องเกี่ยวกับ "การจดจำลายมือเขียนภาษาไทย" รายละเอียดในการจดจำสามารถแยกออกเป็นขั้นตอนต่าง ๆ ได้ดังนี้

1.1 การเตรียมภาพตัวอักษร

เขียนตัวอักษรลงในกระดาษแล้วจัดเก็บโดยใช้เครื่องสแกนเนอร์ (scanner)ซึ่งภาพที่ได้จะเก็บอยู่ในรูปของ Bit-map File และจะประกอบด้วยส่วนที่เรียกว่า header ซึ่งเป็นส่วนที่แสดงรายละเอียดของภาพที่ scan เข้ามาว่ามีขนาดกว้างเท่าไร,ยาวเท่าไร,ขนาดของข้อมูลภาพจริง ๆ มีขนาดเท่าไร

1.2 การทำภาพสองระดับ (Threshold)

เพื่อที่จะใช้แยกสีของตัวอักษร ให้แตกต่างจากสีพื้น โดยใช้หลักการนับจำนวนจุดของค่าความเข้มแต่ละค่าว่ามีกี่จุดแล้วใช้ค่าความเข้มค่าหนึ่งเป็นตัวแบ่งว่าค่าความเข้มที่มีค่ามากกว่าค่าดังกล่าวให้มีค่าเป็น 1 และค่าความเข้มที่ต่ำกว่าให้มีค่าเป็น 0

1.3 ภาพสองระดับที่ได้จะแยกออกไปใช้ในสองขั้นตอน

1.3.1 ขั้นตอนการจัดเตรียมตัวอักษร (Preprocessing) ได้แก่ การทำตัวอักษรให้บาง

1.3.2 ขั้นตอนการหาส่วนหัวของตัวอักษร ได้แก่ การติดตามหาขอบของตัวอักษร,การติดตามหาขอบของตัวอักษร,การหาจุดศูนย์กลางของหัวตัวอักษร

1.4 รวมตำแหน่งของหัวตัวอักษรที่ได้เข้ากับตัวอักษรที่ทำให้บางแล้ว

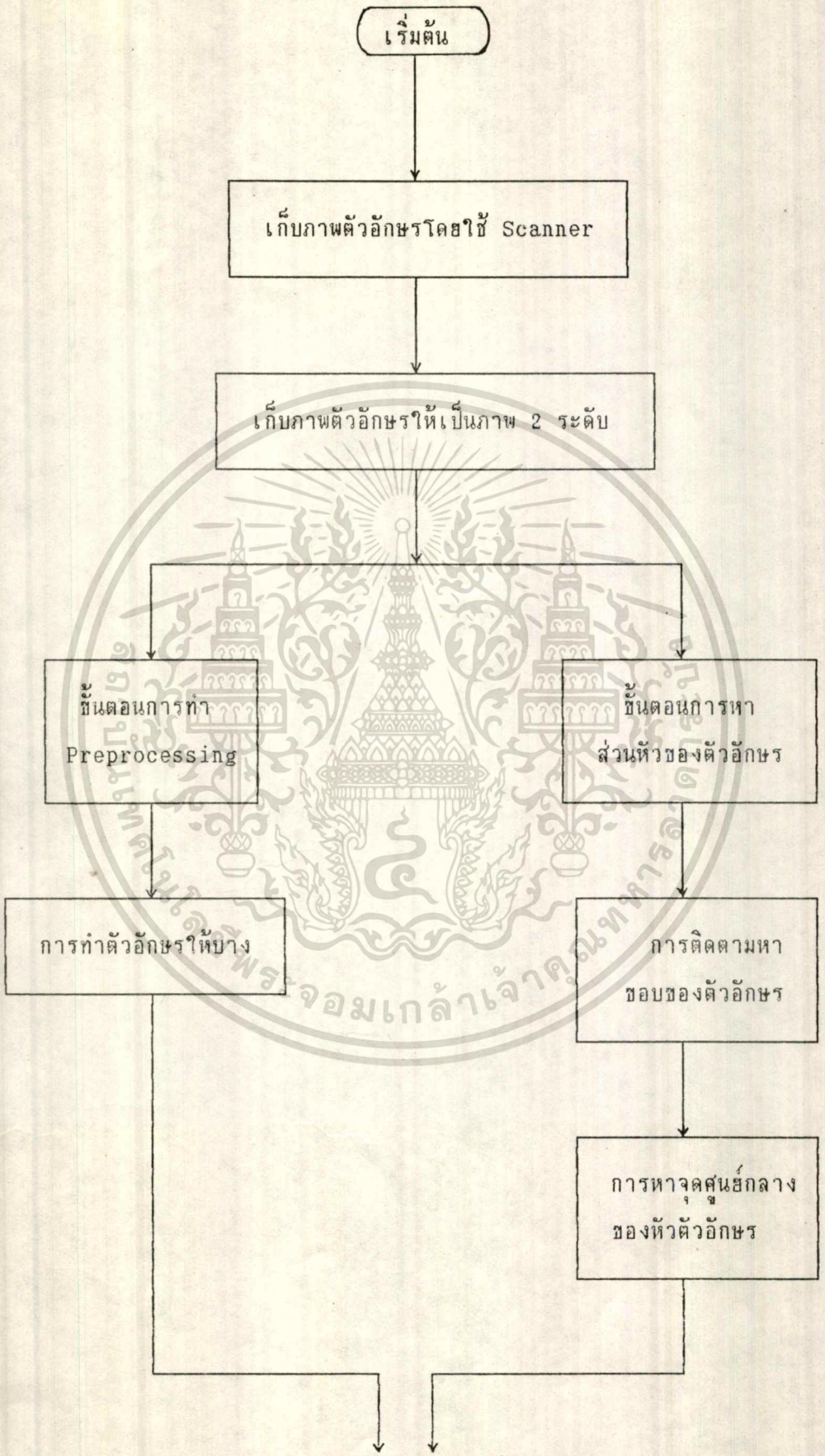
1.5 แบ่งขอบเขตของตัวอักษรออกเป็น 9 zone

1.6 หาตำแหน่งของหัวตัวอักษรในแต่ละ zone

1.7 ใช้ตำแหน่งของหัวตัวอักษรที่ได้จำแนกว่าเป็นตัวอะไร

ซึ่งขั้นตอนต่าง ๆ สามารถเขียนให้เป็นลำดับได้ ดังแผนภาพในหน้าต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีในการจดจำตัวอักษรภาษาไทย ลายมือเขียน

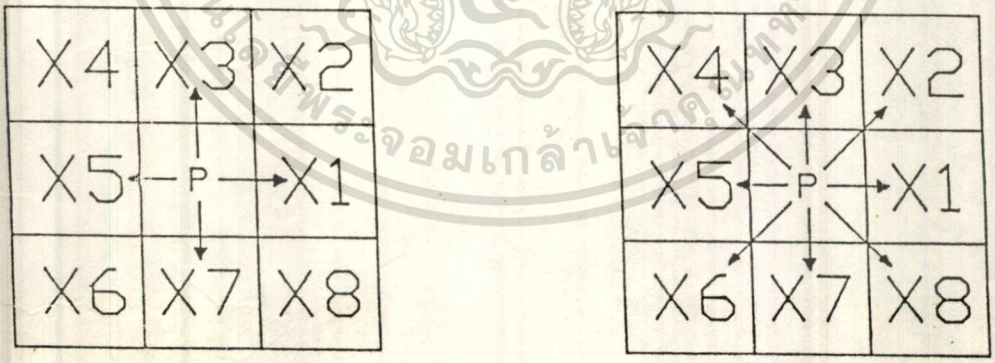
แนวทางในการพัฒนาระบบการจดจำรูปแบบตัวอักษรภาษาไทยให้สามารถนำไปใช้งานได้นั้น มีความจำเป็นอย่างยิ่งที่จะต้องศึกษาทฤษฎีที่เกี่ยวข้องกับภาพ หรือการประมวลผลภาพ (Imange Processing) พร้อมทั้งต้องมีความสังเกตลักษณะเด่นของตัวอักษรแต่ละตัว และลักษณะความคล้ายกันของตัวอักษรบางตัว ทั้งนี้เพราะว่า ลักษณะเหล่านี้นั้นเองที่จะเป็นตัวบ่งบอกถึงประสิทธิภาพของระบบการจดจำรูปแบบตัวอักษรภาษาไทย โดยเฉพาะตัวอักษรลายมือเขียนซึ่งมีรูปร่างที่ไม่แน่นอน

2.1 การคำนวณค่าตัวเลขต่อเนื่อง

ค่าตัวเลขต่อเนื่องเป็นคุณสมบัติทาง Topology ของจุดภาพที่อยู่ติดกันในรูปแบบ 2 ระดับ ค่าตัวเลขต่อเนื่องนี้จะเป็นตัวแสดงความสัมพันธ์ระหว่างจุดภาพข้างเคียง จุดภาพข้างเคียงบนรูปแบบ 2 ระดับใดๆ จะแบ่งได้เป็น 2 แบบคือ

1. จุดภาพข้างเคียงแบบ 4 จุดภาพ (4-Neighbourhood Pixel)
2. จุดภาพข้างเคียงแบบ 8 จุดภาพ (8-Neighbourhood Pixel)

กำหนดให้ p เป็นจุดภาพใดๆ บนรูปแบบ 2 ระดับ จุดภาพข้างเคียงของจุด p ทั้ง 2 แบบแสดงไว้ในรูปที่ 2.1ก และ 2.1ข ตามลำดับ



(ก)จุดข้างเคียง 4 จุด (ข)จุดข้างเคียง 8 จุด

รูปที่ 2.1 ลักษณะของจุดภาพข้างเคียง 2 แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าตัวเลขต่อเนื่องของจุดภาพข้างเคียงทั้ง 2 แบบสามารถได้จากสมการ (2.1) และสมการ (2.2)

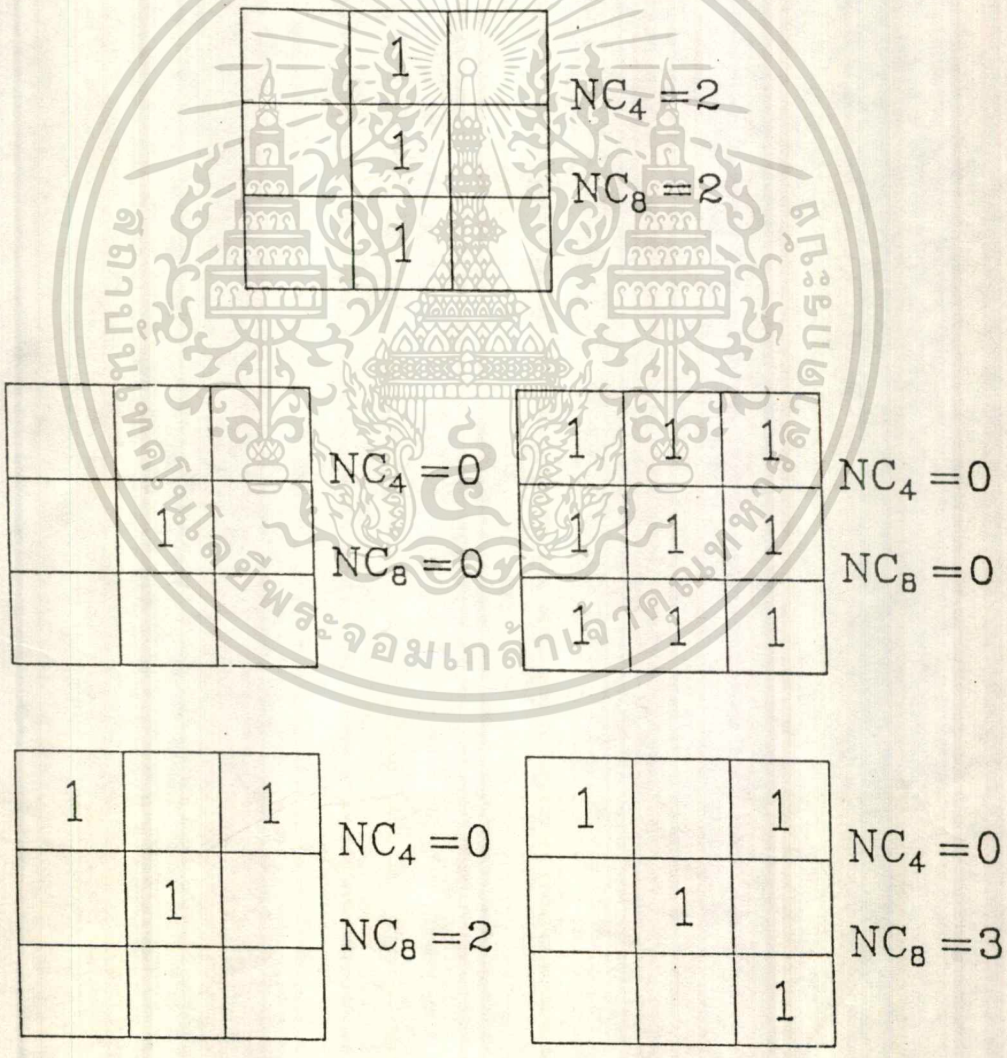
$$NC_4 = \sum_{Si} (X_k - X_k X_{k+1} X_{k+2}) \qquad (2.1)$$

$$NC_8 = \sum_{Si} (\bar{X}_k - \bar{X}_k \bar{X}_{k+1} \bar{X}_{k+2}) \qquad (2.2)$$

$$Si = \{1,3,5,7\},\ k \geqslant 9 \longrightarrow k = k-8$$

$$\bar{X} = (1-X)$$

ผลที่ได้จากสมการจะเป็นเลขจำนวนเต็มบวกที่มีค่าอยู่ระหว่าง 0 ถึง 4 ดังที่แสดงในรูป2.2



รูปที่ 2.2 ค่า NC_4 และ NC_8 ของจุดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าตัวเลขต่อเนื่องที่คำนวณได้ทั้ง 2 แบบ จะบ่งบอกถึงความสัมพันธ์ระหว่างจุดภาพออกมา ดังที่แสดงในตารางที่ 2.1

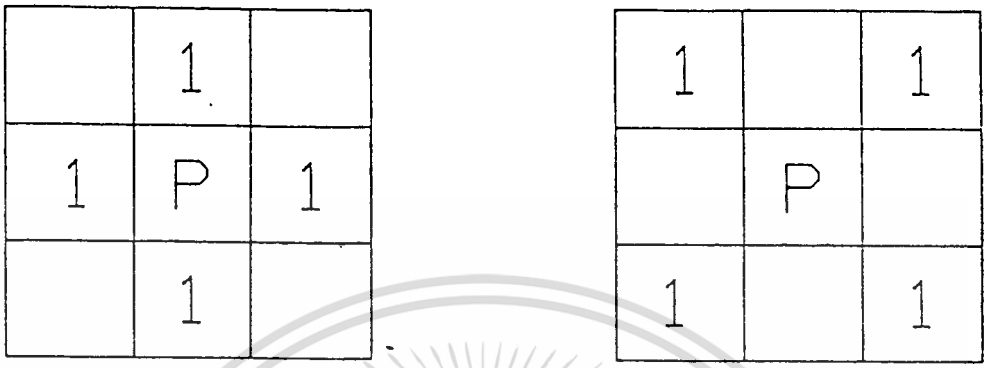
ตารางที่ 2.1 ตารางแสดงความสัมพันธ์ระหว่างจุดภาพ

ค่าตัวเลขต่อเนื่อง	จุดอิสระหรือจุดภายใน	จุดปลาย		จุดต่อเนื่อง					จุดแยก				จุดตัด	
NC_4	0	1	0	1	2	2	2	0	3	0	1	2	4	0
NC_8	0	1	1	2	2	2	1	2	1	3	3	2	0	4

ค่าตัวเลขต่อเนื่องของ จุดอิสระหรือจุดภายในคือ0, จุดปลายคือ1, จุดต่อเนื่องคือ2, จุดแยกคือ 3, จุดตัดคือ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการคำนวณ



รูปที่ 2.3 ตัวอย่างการคำนวณค่า NC_4 และ NC_8

จากรูปที่ 2.3ก) จะคำนวณค่า NC_4 และ NC_8 ได้ดังนี้

$$NC_4 = X_1+X_3+X_5+X_7 - (X_1*X_2*X_3)-(X_3*X_4*X_5)-(X_5*X_6*X_7)-(X_7*X_8*X_1)$$

$$= 1+1+1+1 - 1*0*1 - 1*0*1 -1*0*1 - 1*0*1$$

$$= 4$$

$$NC_8 = \overline{X}_1+\overline{X}_3+\overline{X}_5+\overline{X}_7 - (\overline{X}_1*\overline{X}_2*\overline{X}_3)-(\overline{X}_3*\overline{X}_4*\overline{X}_5)-(\overline{X}_5*\overline{X}_6*\overline{X}_7)-(\overline{X}_7*\overline{X}_8*\overline{X}_1)$$

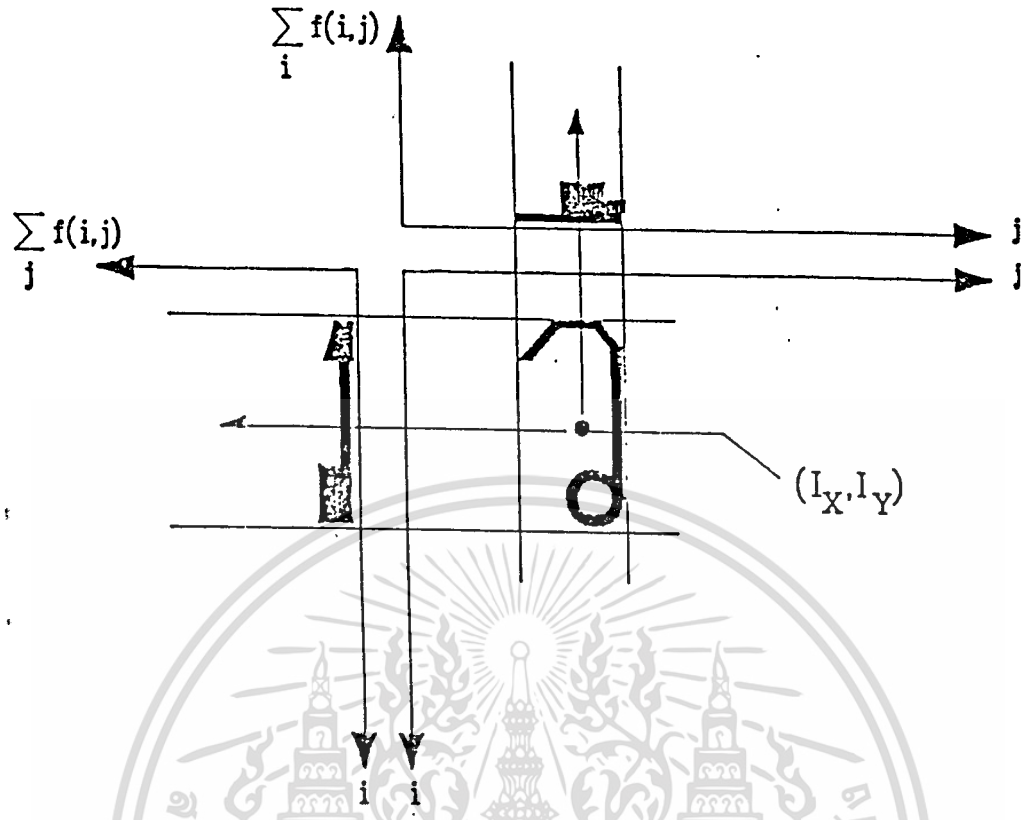
$$= 0+0+0+0 - 0*1*0 - 0*1*0 - 0*1*0 - 0*1*0$$

$$= 0$$

2.2 การหาจุดศูนย์ถ่วง

รูปแบบ 2 ระดับที่มีจุดภาพ ที่มีค่า 1 เป็นเนื้อของภาพ และมีค่า 0 เป็นฉากของภาพ จะสามารถหาตำแหน่งที่เป็นจุดศูนย์กลางของภาพได้ จุดนี้เรียกว่า "จุดศูนย์ถ่วง"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 จุดศูนย์ถ่วงของภาพ 2 ระดับ

จากรูปที่ 2.4 ส่วนที่เป็นสีทึบแสดงถึงส่วนที่เป็นเนื้อของภาพ ค่า i,j เป็นจำนวนภาพในแนว Row และแนว Column ตามลำดับ

ตำแหน่งของRow และ Column ที่เป็นตำแหน่งของจุดศูนย์ถ่วงจะหาได้จากสมการต่อไปนี้

$$I_x = \frac{\sum_i \sum_j j\ f(i,j)}{\sum_i \sum_j f(i,j)} \qquad (2.3)$$

$$I_y = \frac{\sum_i \sum_j i\ f(i,j)}{\sum_i \sum_j f(i,j)} \qquad (2.4)$$

เมื่อ f(i,j) เป็น function แสดงจุดภาพบนรูปแบบ 2 ระดับใดๆ

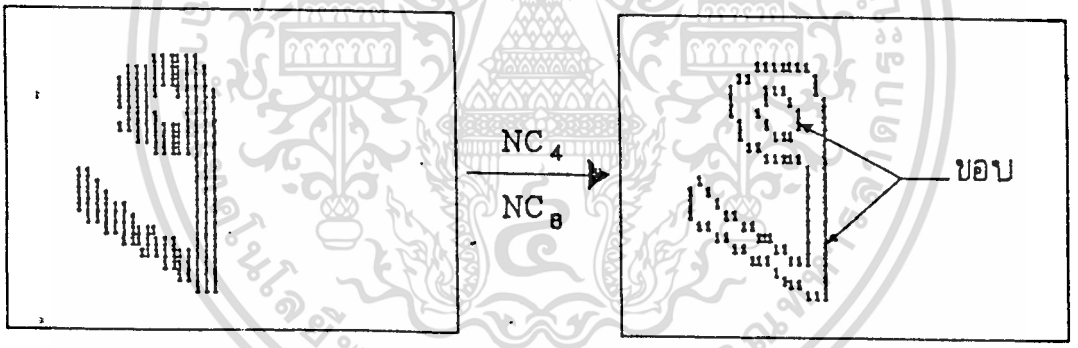
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การตรวจสอบขอบของภาพ

ขอบของภาพในรูปแบบ กับฉาก จะแบ่งจากกันโดยขอบของภาพ การหาขอบของภาพรูปแบบ 2 ระดับใดๆ มีวิธีการทำได้หลายวิธีเช่น การใช้คุณสมบัติทาง Topology ของจุดภาพการติดตามขอบของภาพ(Contour following) หรือวิธีการใช้ตารางหน้าต่าง สำหรับในปริญญานิพนธ์ฉบับนี้จะกล่าวถึงเพียง 2 วิธีเท่านั้นคือ

1. การหาขอบของภาพโดยวิธีการใช้ค่าตัวเลขต่อเนื่อง NC_4 และ NC_8 ที่คำนวณได้จากสมการที่ 2.1 และ 2.2

จุดภาพใดๆ ก็ตามที่มีค่า NC_4 และ NC_8 เป็นไปตามลักษณะของจุดปลายซึ่งก็หมายถึงว่าเป็นจุดภาพที่เป็นขอบดังแสดงในรูปที่ 2.5

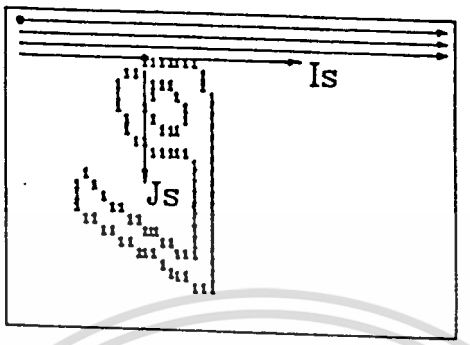


รูปที่ 2.5 ค่าตัวเลขต่อเนื่องที่เป็นขอบของภาพ

ดังนั้นจึงสามารถใช้ค่าตัวเลขต่อเนื่องที่หาได้บอกขอบของภาพได้ ทั้งขอบในและขอบนอกของภาพ แต่จะไม่สามารถบอกจำนวนจุดภาพที่เป็นขอบได้

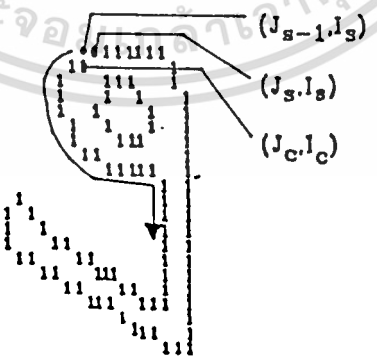
2.การหาขอบของภาพโดยการติดตามขอบเขตของภาพ การหาขอบภาพโดยวิธีนี้ จะต้องหาจุดเริ่มต้นของขอบก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
034719



รูปที่ 2.6 การ scan หาจุดภาพที่เป็นจุดเริ่มต้น

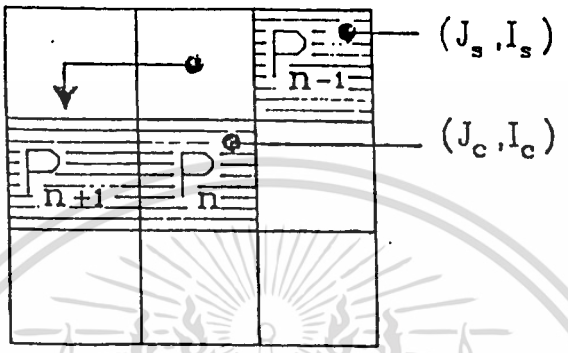
จากรูปแบบ 2 ระดับในรูปที่ 2.6 จะทำการ scan ไปบนรูปแบบจากบนลงล่างจนกระทั่งพบจุดที่เป็นภาพ ก็จะถือว่าจุดนั้นเป็นจุดเริ่มต้นของจุดภาพ(J_s, I_s) และจะกำหนดให้จุดก่อนหน้าที่จะถึงจุดเริ่มต้นของขอบนี้(J_s-1, I_s) เป็นตำแหน่งเริ่มต้นสำหรับการวนในทิศทวนเข็มนาฬิกา เพื่อติดตามหาขอบภาพจุดต่อไป(J_c, I_c) ดังรูปที่ 2.7



รูปที่ 2.7 จุดเริ่มต้นของการติดตามหาขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นวาง Window ขนาด 3*3 ที่ขอบภาพ(J_c,I_c) แล้วจะวนในทิศทางทวนเข็มนาฬิกา เพื่อติดตามหาขอบภาพจุดต่อไป ดังรูปที่ 2.8



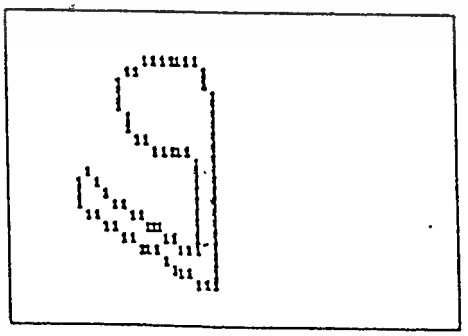
รูปที่ 2.8 การติดตามหาขอบภาพที่เป็นขอบ

เมื่อ P_{n-1} เป็นตำแหน่งของขอบภาพที่ผ่านมา

P_n เป็นตำแหน่งของขอบภาพที่กำลังพิจารณา

P_{n+1} เป็นตำแหน่งของขอบภาพต่อไป

จากรูปที่ 2.8 จะเห็นว่าจุดเริ่มต้นในการติดตามขอบจุดต่อไป(P_{n+1}) จะเป็นตำแหน่งที่ถัดจากจุดขอบภาพที่ผ่านมา(P_{n-1}) ในทิศทางทวนเข็มนาฬิกาเสมอ ด้วยวิธีการดังกล่าวนี้เมื่อติดตามขอบเขตของภาพไปจนกระทั่งพบตำแหน่งที่เริ่มต้นของขอบภาพ(J_s,I_s) ก็จะได้ขอบของภาพออกมา ดังรูปที่ 2.9



รูปที่ 2.9 ขอบของภาพที่ได้จากการติดตามขอบเขต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

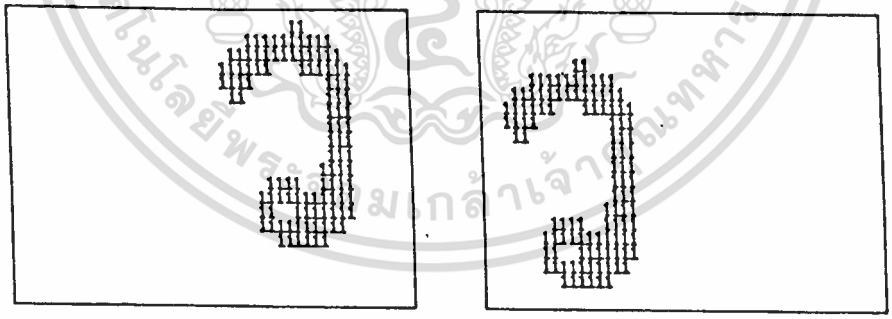
บทที่ 3
การเตรียมข้อมูลเพื่อการจำแนกกลุ่มของตัวอักษร

ปัญหาที่เกิดขึ้นเสมอในระบบการจำแนกรูปแบบของตัวอักษรภาษาไทย คือ รูปแบบตัวอักษรที่ใช้เป็น input pattern ของระบบมีสัญญาณรบกวนปะปนเข้ามาด้วย สัญญาณรบกวนนี้จะทำให้เกิดการผิดพลาดขึ้นในขบวนการจำแนกตัวอักษร ดังนั้นจึงมีความจำเป็นที่จะต้องมีการปรับปรุงคุณภาพของรูปแบบตัวอักษรก่อนที่จะนำไปจำแนกกลุ่มต่อไป จึงเรียกขบวนการในขั้นตอนนี้ว่า "ระบบการจัดการล่วงหน้า(Pre-processing)"

ระบบการจัดการล่วงหน้าที่กล่าวถึงในบทนี้ ประกอบไปด้วยวิธีการต่างๆ หลายขั้นตอนดังจะได้กล่าวถึงในหัวข้อต่อๆ ไป

3.1 การหาตำแหน่งมาตรฐาน

รูปแบบของตัวอักษรจะวางอยู่ที่ตำแหน่งต่างๆ กันภายในรูปแบบ 2 ระดับ 128*128 จุดภาพดังรูปที่แสดงไว้ในรูปที่ 3.1

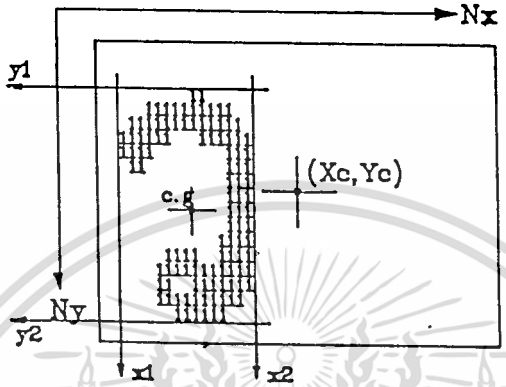


(ก) (ข)

รูปที่ 3.1 ตำแหน่งต่างๆ ของตัวอักษรภายในรูปแบบ

เพื่อความเป็นมาตรฐานของตำแหน่ง จึงต้องทำการย้ายตัวอักษรดังที่แสดงในรูปที่ 3.1ให้อยู่ตรงตำแหน่งที่มาตรฐาน ในที่นี้ได้ใช้จุดกึ่งกลางของรูปแบบ(X_c, Y_c) เป็นตำแหน่งมาตรฐาน

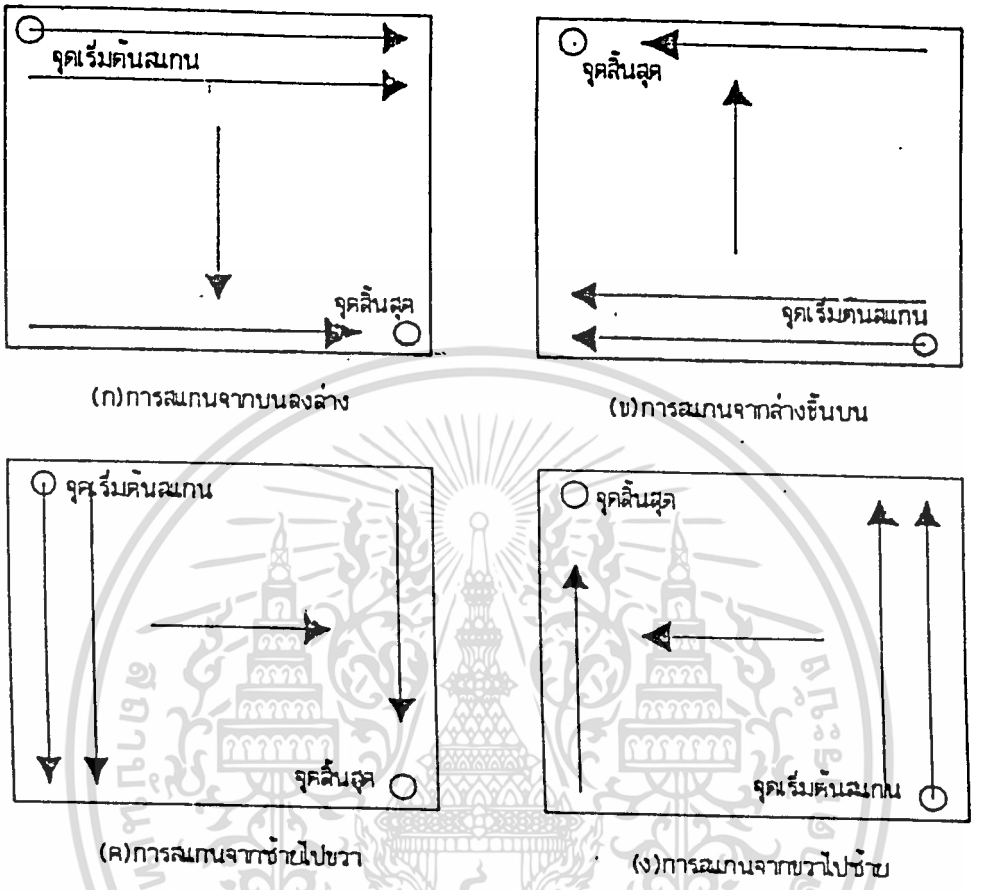
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การย้ายภาพตัวอักษรไปยังตำแหน่งมาตรฐาน
เมื่อ $X_c = N_x/2$ และ $Y_c = N_y/2$

การย้ายรูปแบบจะทำได้โดยการ scan หากรอบของตัวอักษรทั้ง 4 ด้าน คือ X_1, X_2, Y_1 และ Y_2 ซึ่งก็คือ กรอบ บน ล่าง ซ้าย ขวา ของตัวอักษร ดังแสดงในรูปที่3.2 ทิศของการ scan ทั้ง 4 ด้านแสดงในรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



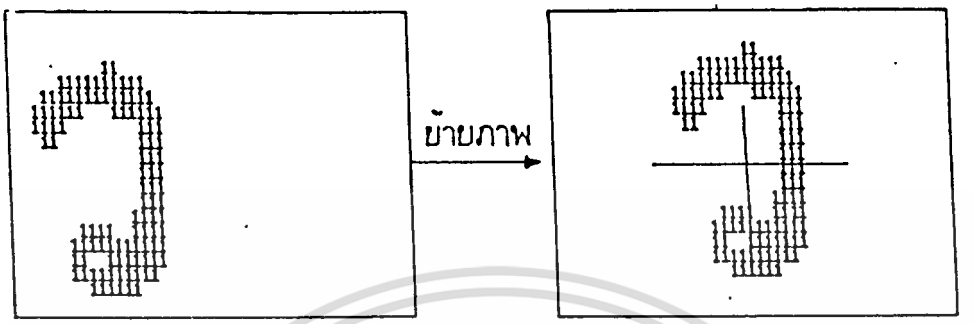
รูปที่ 3.3 ทิศทางของการ scan ทั้ง 4 แบบ

ก) การ scan จากบนลงล่าง ข) การ scan จากล่างขึ้นบน

ค) การ scan จากซ้ายไปขวา ง) การ scan จากขวาไปซ้าย

ตัวอักษรที่อยู่ในกรอบนี้จะนำมาหาตำแหน่งของจุดศูนย์ถ่วง(c.g) โดยใช้สมการที่ 2.3 และ 2.4 ในบทที่ 2 แล้วทำการย้ายภาพตัวอักษรจากตำแหน่งจุดศูนย์ถ่วงนี้ไปยังตำแหน่งมาตรฐาน ดังแสดงในรูปที่ 3.4

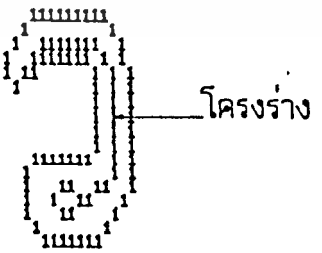
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ภาพตัวอักษรที่ย้ายมายังตำแหน่งมาตรฐาน

3.2 การทำตัวอักษรให้บาง (Thinning)

รูปแบบ 2 ระดับที่ได้มาจากการ scan จะมีภาพของตัวอักษรในรูปแบบที่มีขนาดความกว้างมากกว่า 1 จุดภาพเสมอ แต่ในระบบการจดจำที่ใช้คุณสมบัติทาง Topology ของรูปแบบเป็นตัวจำแนกกลุ่มนั้น จะต้องการเฉพาะส่วนที่เป็นโครงร่าง(Skeleton) ของตัวอักษรเท่านั้น โครงร่างของตัวอักษรก็คือ ภาพของตัวอักษรที่มีความกว้างเพียง 1 จุดภาพ และอยู่ในตำแหน่งที่สมมาตรบนเนื้อของตัวอักษร ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 โครงร่างของตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจึงต้องทำตัวอักษรในรูปแบบให้บางลง ด้วยวิธีการทำให้บาง(Thinning) วิธีการทำให้บางนี้ก็มีการวิจัยและพัฒนากันหลายๆ วิธี

n_3	n_2	n_1
n_4	P	n_0
n_5	n_6	n_7

รูปที่ 3.6 ตารางหน้าต่างสำหรับการทำให้บาง

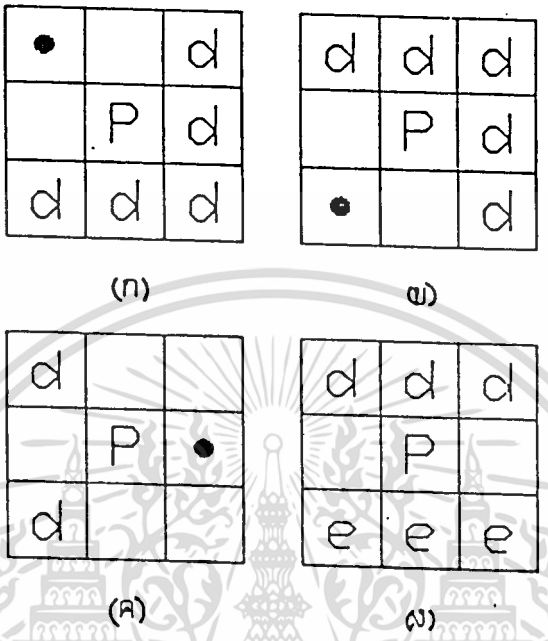
ตารางหน้าต่างในรูปที่ 3.6 ใช้นำมาพิจารณาส่วนของภาพตัวอักษรบนรูปแบบที่ต้องการทำให้บาง โดยที่จุดข้างเคียงของจุดภาพ P ซึ่งก็คือ จุดภาพ $n_0, n_1, n_2, \ldots, n_7$ จะเป็นตัวกำหนดลักษณะขอบ(Edge) ของจุดภาพ P ใน 4 ลักษณะคือ

1. จุดภาพ P เป็นขอบซ้ายถ้ามี n_4 เป็นจุดสว่าง($n_4 = 0$)
2. จุดภาพ P เป็นขอบขวาถ้ามี n_0 เป็นจุดสว่าง($n_0 = 0$)
3. จุดภาพ P เป็นขอบบนถ้ามี n_2 เป็นจุดสว่าง($n_2 = 0$)
4. จุดภาพ P เป็นขอบล่างถ้ามี n_6 เป็นจุดสว่าง($n_6 = 0$)

ในขบวนการทำให้บางถือว่า ขอบของภาพมิใช่โครงร่างของตัวอักษร ดังนั้น เมื่อพบขอบก็จะกำหนดค่าของจุดภาพนั้นให้มีค่าค่าหนึ่ง ที่ไม่เป็น 1(flagging) แต่ถ้าพบจุดภาพที่มีความกว้างเพียง 1 จุดภาพก็จะกำหนดค่าของจุดภาพนั้นให้เป็นค่า 1(unflagging)

การทดสอบจุดภาพที่เป็นขอบ กระทำโดยการเปรียบเทียบตารางหน้าต่างในรูปที่ 2.6 ที่สแกนจากบนลงล่างบนรูปแบบของตัวอักษร กับตารางหน้าต่างในรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ตารางหน้าต่างสำหรับการตรวจสอบหาขอบ

จากรูปที่ 3.7 P และ • เป็นจุดมืด(มีค่าเป็น 1) ส่วน d และ e เป็นค่าที่ไม่สนใจ (don't care)

เมื่อพิจารณารูปที่ 3.7(ก),(ข),(ค) จะแบ่งการพิจารณาเป็น 2 ลักษณะคือ

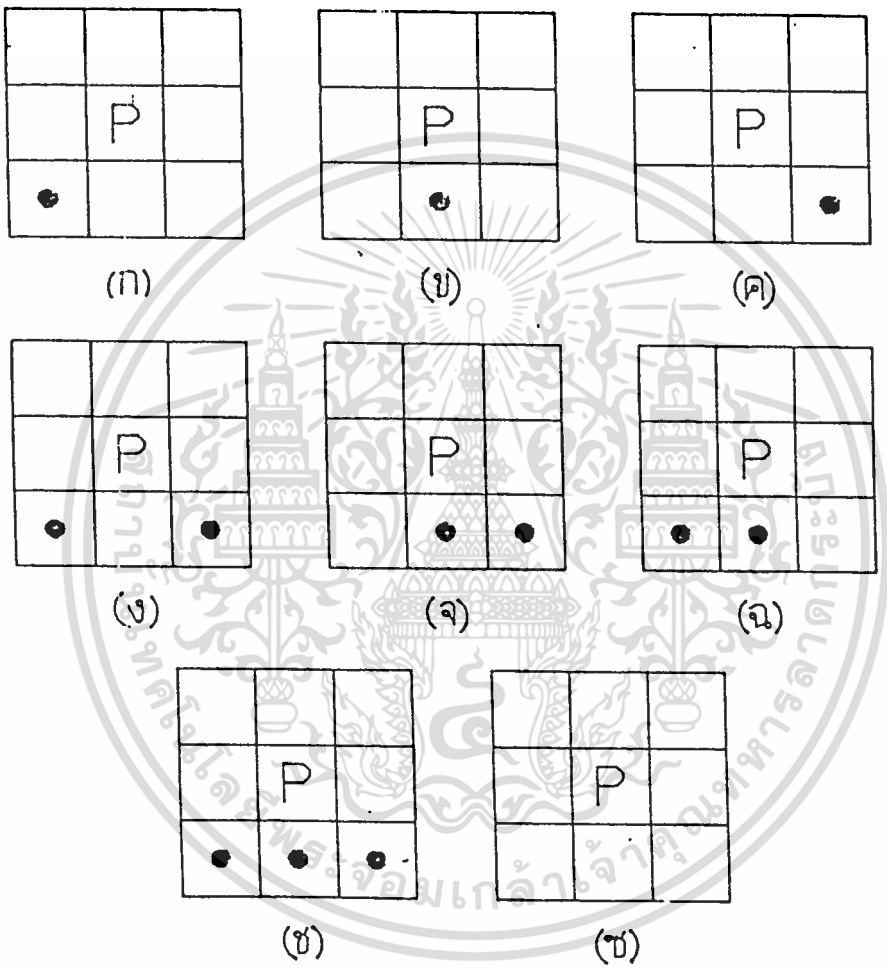
1. ถ้า d ทุกตัวเป็นจุดสว่าง (มีค่าเป็น 0) จะได้ว่า P คือ จุดภาพที่เป็นจุดปลายของโครงร่าง

2. ถ้า d อย่างน้อย 1 จุดเป็นจุดมืด (มีค่าเป็น 1) จะได้ว่า P คือ จุดภาพที่เป็นโครงร่างของตัวอักษร

จุด P ใดๆ ที่มีคุณสมบัติตาม 2 ลักษณะดังกล่าว จะถูกกำหนดค่าให้เป็น 1 (unfiaged)

ตารางหน้าต่างในรูปที่ 3.7(ง) จะมีการพิจารณาที่ซับซ้อนขึ้นโดยแยกการพิจารณาออกมาดังแสดงโดยตารางหน้าต่างในรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 ตารางหน้าต่างของจุดภาพโครงร่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.7(ง) ถ้า d และ e อย่างน้อย 1 จุด เป็นจุดมืดจะกำหนดให้จุด P เป็นจุดโครงร่าง ในกรณีนอกเหนือจากนี้จะพิจารณาตามตารางหน้าต่างในรูปที่ 3.8

จากรูป 3.8 ตารางหน้าต่าง (ก),(ข),(ค) จะกำหนดให้จุด P เป็นจุดปลายของโครงร่าง ตารางหน้าต่าง(ง) กำหนดให้ P เป็นโครงร่างตารางหน้าต่าง(จ) และ (ฉ) กำหนดให้ P เป็นจุดที่ทำให้เกิดความกว้างขนาด 2 จุดภาพในแนวเอียง ตารางหน้าต่าง(ช) กำหนดให้จุด P เป็นกิ่งสั้น(protrusion) ของโครงร่าง ส่วนตารางหน้าต่าง(ซ) กำหนดให้จุด P เป็นจุดอิสระที่เกิดขึ้น

เพื่อลดความยุ่งยากในการพิจารณาเงื่อนไขของขอบและโครงร่างลง จึงได้นำเอาเงื่อนไขทั้งหมดของขอบมาเขียนเป็นนิพจน์ทางตรรก ทำให้ง่ายต่อการเขียน program ยิ่งขึ้น ดังแสดงในสมการที่ (3.1),(3.2),(3.3) และ (3.4) ตามลำดับ

ขอบขวา: $B0 = N4*(N2+N3+N5+N6)*(N6+\overline{N7})*(\overline{N1}+N2)$ (3.1)

ขอบล่าง: $B6 = N2*(N0+N1+N3+N4)*(N4+\overline{N5})*(N0+\overline{N7})$ (3.2)

ขอบซ้าย: $B4 = N0*(N1+N2+N6+N7)*(N2+\overline{N3})*(\overline{N5}+N6)$ (3.3)

ขอบขวา: $B2 = N6*(N0+N4+N5+N7)*(N0+\overline{N1})*(\overline{N3}+N4)$ (3.4)

เมื่อ * : operator AND

\+ : operator OR

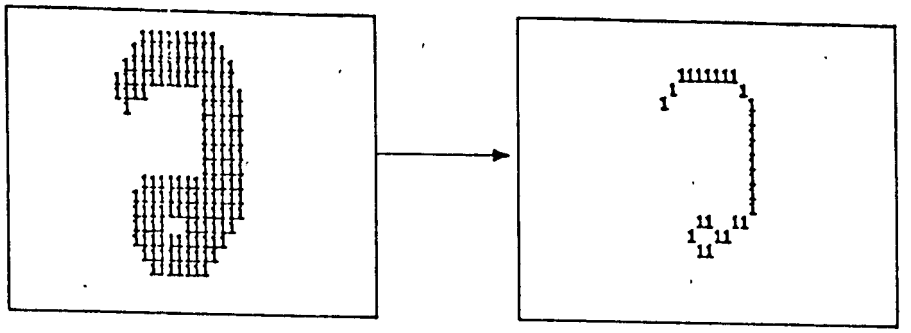
\- : operator NOT

และ B4,B0,B2,B6 เป็นค่าของจุดภาพ P ที่มีลักษณะเป็นขอบซ้าย ขอบขวา ขอบบน และขอบล่างตามลำดับ

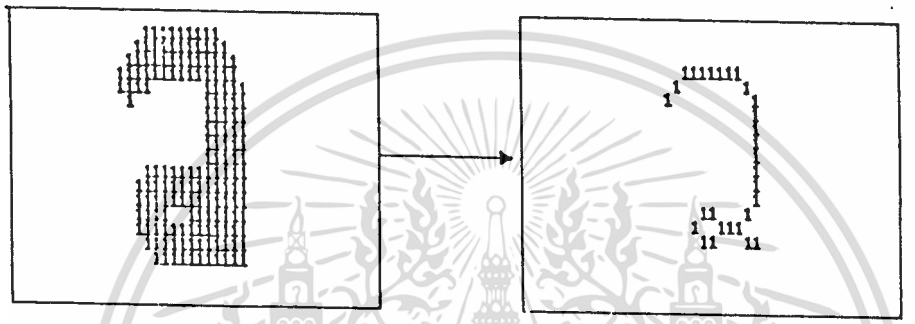
โดยการscanไปบนรูปแบบ 2 ระดับที่ต้องการทำให้บาง ทำการสแกนใน 2 ทิศทาง คือ scanจากบนลงล่างและscanจากซ้ายไปขวา(รูปที่ 2.3(ก) และ (ค)) ในทิศทางแรกจะพิจารณาขอบ B4 และ B0 และในทิศทางที่ 2 จะพิจารณาขอบ B2 และ B6 ตามลำดับ

ด้วยการกระทำแบบวนรอบ(Iteration) จนกระทั่งเหลือแต่จุดภาพที่เป็นโครงร่างก็จะได้ตัวอักษรที่บางออกมาดังแสดงในรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูปที่ 3.9 โครงร่างของตัวอักษร

(ก) โครงร่างที่สมบูรณ์

(ข) โครงร่างที่มีส่วนเกิน

ขั้นตอนการทำตัวอักษรให้บางทั้งหมดแสดงไว้ใน Algorithm a:

Algorithm a:

```
        While (new edge point unflaged)
    {
        : forward scanning
        if B4 or B0
        (flaged)
        end if
        delete edge point
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

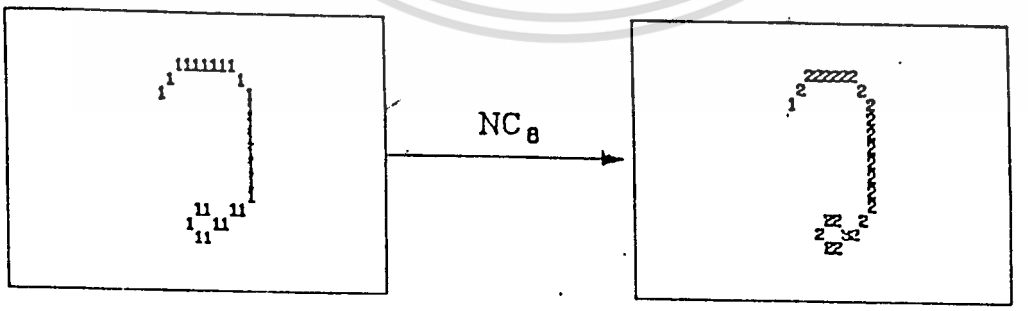
```
: leftward scanning
if B2 or B6
(flaged)
end if
delete edge point
}
```

โครงร่างของตัวอักษรที่ได้จากการทำให้บางจะเก็บเป็นรูปแบบที่จะนำไปหาค่าคุณสมบัติทาง Topology ต่อไป

ข้อดีของการทำให้บาง คือ ลดจำนวนข้อมูลที่เป็นภาพตัวอักษรลงไปเป็นจำนวนมาก ทำให้ลดเวลาของการประมวลผลในขั้นตอนต่อไปได้ แต่ทั้งนี้การทำให้บางก็จะใช้เวลาในการประมวลผลมาก ถ้ารูปแบบ 2 ระดับที่เข้ามามีความหนามากๆ

3.4 ค่าตัวเลขที่ต่อเนื่องของรูปแบบตัวอักษร

รูปแบบสองระดับที่เป็นโครงร่างซึ่งปราศจากส่วนเกินของตัวอักษร เมื่อนำมาทำการ scan จากบนลงล่าง และคำนวณหาค่าตัวเลขต่อเนื่องของจุดภาพที่เป็น 1 โดยใช้สมการที่ (2.1) และ (2.2) ก็จะได้รูปแบบของตัวอักษรออกมา ดังแสดงในรูปที่ 3.10 ค่าตัวเลขต่อเนื่องของรูปแบบตัวอักษรในรูปที่ 3.10 นี้จะแสดงถึงความสัมพันธ์ระหว่างจุดภาพที่อยู่ติดกัน โดยมีความหมายดังที่แสดงไว้ในตารางที่ 2.1



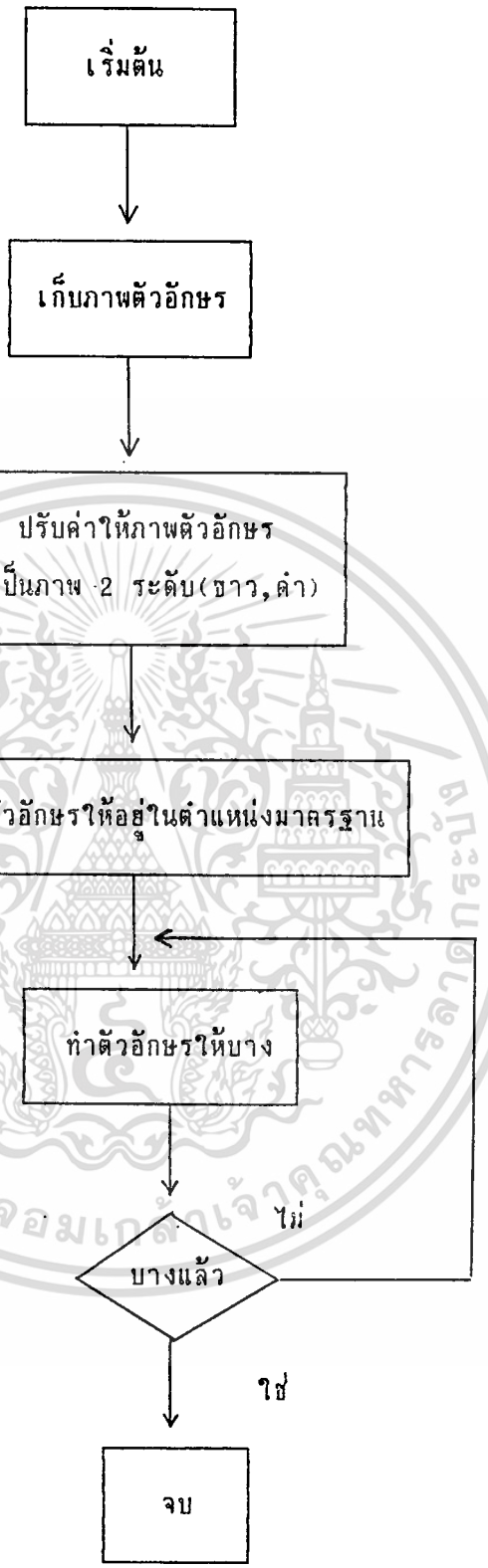
รูปที่ 3.10 ค่าตัวเลขต่อเนื่องบนโครงร่างของตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของตัวอักษรที่ผ่านการเตรียมข้อมูลแล้วจะเป็นรูปแบบที่มีความสมบูรณ์ พร้อมที่จะนำไปจำแนกออกเป็นกลุ่ม และกลุ่มย่อย ในขั้นตอนต่อไปของระบบการจดจำ

สรุปกรรมวิธีเกี่ยวกับการเตรียมข้อมูลของตัวอักษร เพื่อการจดจำตัวอักษรในระบบการจัดการล่วงหน้า แสดงโดยผังงานในรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ผังงานแสดงระบบการจัดการล่วงหน้า

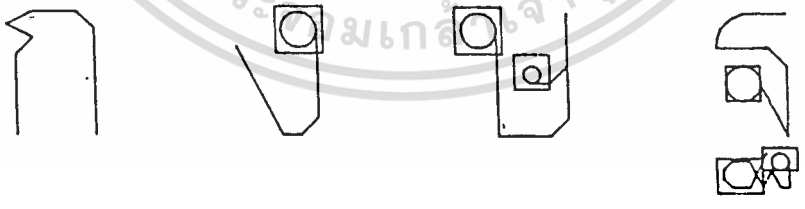
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การตรวจสอบหัวและการแบ่งกลุ่มตัวอักษร

ภาพของตัวอักษรแต่ละตัวที่ผ่านเข้ามาในระบบการจดจำ จะถูกนำไปเปลี่ยนให้เป็นรูปแบบสองระดับของตัวอักษรด้วยวิธีการดังกล่าวไว้แล้วในบทที่ 2 รูปแบบสองระดับที่ได้นี้จะนำไปลอกแบบออกเป็น 2 รูปแบบ ของตัวอักษรส่วนที่หนึ่งใช้เป็นรูปแบบสำหรับการตรวจสอบหาหัวของตัวอักษร และรูปแบบของตัวอักษรในส่วนที่สองใช้เป็นรูปแบบสำหรับระบบการจัดการล่วงหน้า ซึ่งระบบการจัดการล่วงหน้านี้ได้แสดงถึงรายละเอียดไว้แล้วในบทที่ 4 ดังนั้นเนื้อหาในบทนี้ก็จะกล่าวถึง ส่วนที่มีความสำคัญที่สุดของระบบการจำแนกตัวอักษร (CLASSIFICATION) ออกเป็นกลุ่ม ๆ

4.1 การตรวจสอบหัวตัวอักษร (Head detection)

หัวของตัวอักษรในภาษาไทย จะมีลักษณะเป็นวงปิดขนาดเล็ก (เมื่อเทียบกับส่วนอื่น ๆ ในตัวอักษร) วางอยู่ตามตำแหน่งต่าง ๆ และมีจำนวนต่าง ๆ กันออกไป สำหรับรูปแบบของตัวอักษรแต่ละตัวดังแสดงในรูปที่ 4.1

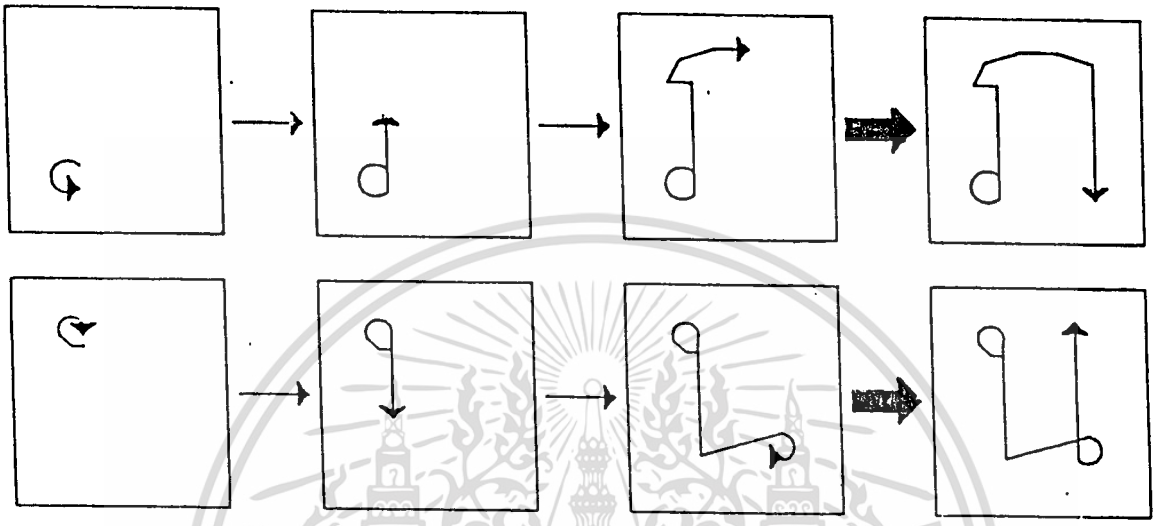


(ก) (ข) (ค) (ง)

รูปที่ 4.1 จำนวนและตำแหน่งของหัว
(ก) ตัวอักษรที่ไม่มีหัว (ข) ตัวอักษรที่มีหัว 1 หัว
(ค) ตัวอักษรที่มีหัว 2 หัว (ง) ตัวอักษรที่มีหัว 3 หัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะพบว่าตัวอักษรในภาษาไทยทั้งหมดจะสามารถที่จะแบ่งได้เป็น ตัวอักษรที่ไม่มีหัว มีหัว 1 หัวมีหัว 2 หัว และมีหัว 3 หัว ดังแสดงในรูปที่ 4.1 (ก),(ข),(ค) และ (ง) ตามลำดับ



รูปที่ 4.2 ลักษณะการเขียนตัวอักษร

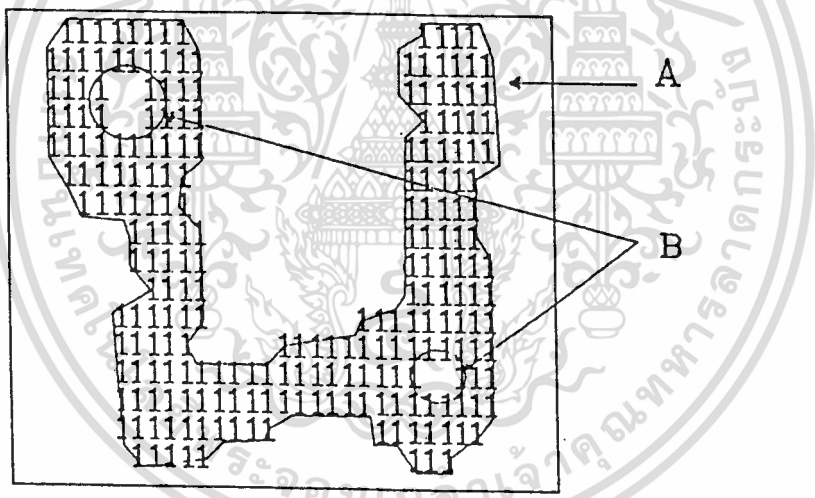
เมื่อพิจารณาถึงการเขียนตัวอักษรภาษาไทยด้วยแล้ว ก็จะเป็นที่สังเกตุได้อย่างชัดเจนในการเขียนตัวอักษรที่มีหัวนั้นจะเริ่มต้นขึ้นต้นที่การเขียนหัวของตัวอักษรก่อนเสมอ ดังจะเห็นได้จากขั้นตอนการเขียนตัวอักษร "ภ" และตัวอักษร "น" ในรูปที่ 4.2

ดังนั้นจึงจะถือได้ว่า หัวของตัวอักษรภาษาไทยเป็นลักษณะที่เด่นชัดอย่างหนึ่งของตัวอักษร ซึ่งในปริญญานิพนธ์ฉบับนี้จะได้นำเอาลักษณะเด่นอันนี้มาพิจารณาถึงความแตกต่างในเรื่องของจำนวนและตำแหน่งเพื่อใช้เป็นเครื่องมือในการจำแนกกลุ่มของตัวอักษร

จากรูปแบบสองระดับของตัวอักษรภาษาไทยในรูปที่ 4.3 จะเห็นได้ว่าขอบของภาพตัวอักษรจะแยกกันใน 2 ลักษณะคือ ส่วนที่เป็นขอบนอก (A)และขอบใน (B) ขอบในนี้เองที่ประกอบกันขึ้นเป็นหัวของตัวอักษร

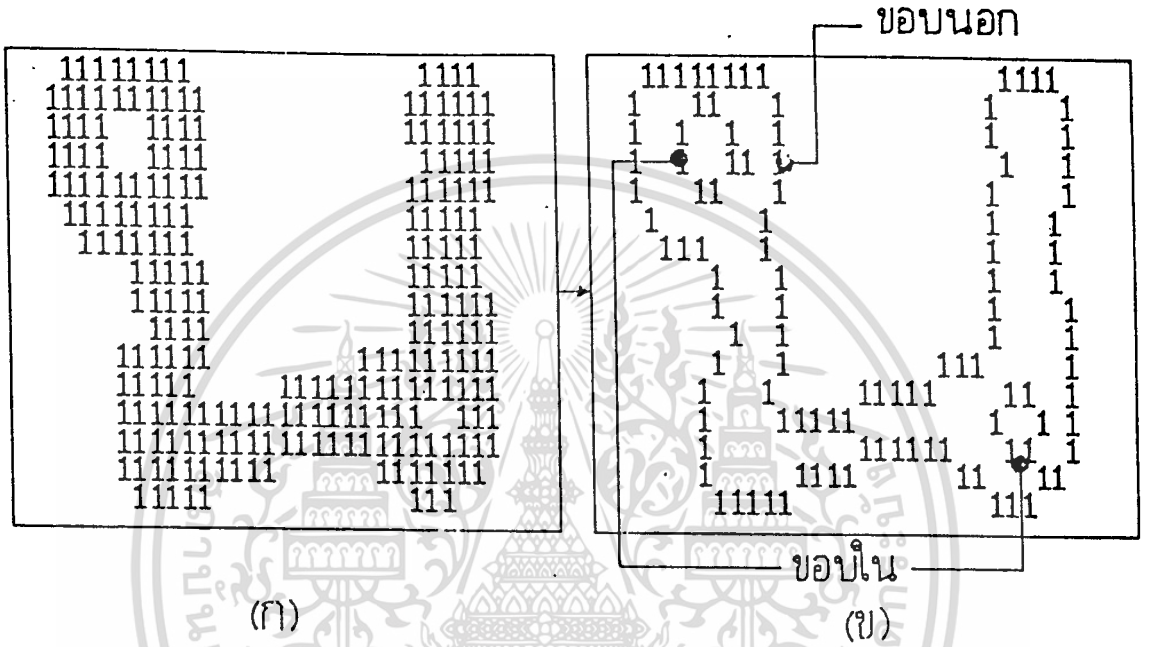
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจสอบหัวของตัวอักษรจะเริ่มต้นโดยทำการ scan จากบนลงล่าง (รูปที่ 4.3 (ก)) ไปบนรูปแบบสองระดับที่ยังไม่ได้มีการทำให้บาง แล้วคำนวณหาค่าตัวเลขต่อเนื่อง NC_4 และ NC_8 (โดยใช้สมการที่ (2.1) และ (2.2)) ของจุดภาพใด ๆ บนรูปแบบและทำการเก็บค่าไว้ในรูปแบบ ก็จะทำให้สามารถหาส่วนที่เป็นขอบในและขอบนอกของตัวอักษรออกมาได้ดังรูปที่ 4.4



รูปที่ 4.3 รูปแบบสองระดับสำหรับการตรวจสอบหัว

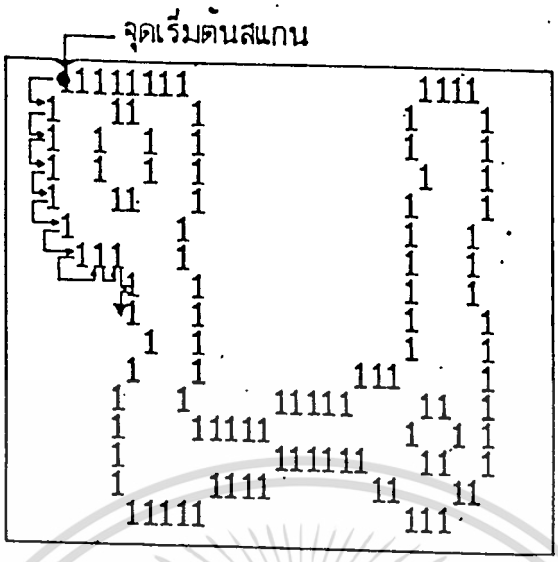
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ขอบของตัวอักษร

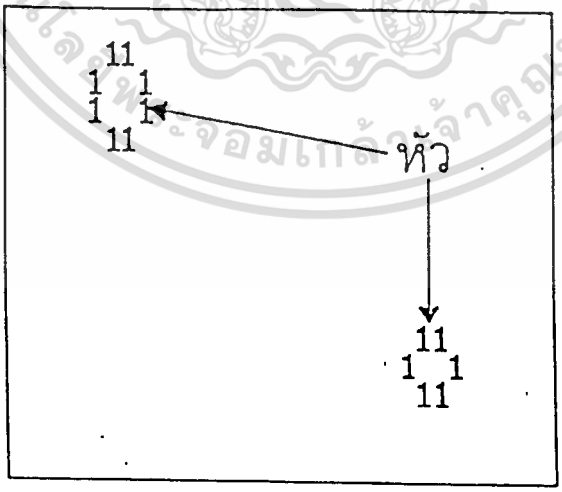
ขั้นตอนต่อไปนี้จะทำการ scan จากบนลงล่าง จนรูปแบบที่เป็นขอบของภาพ(รูปที่ 4.4(ข)) จนกระทั่งพบจุดภาพที่เป็นขอบจุดแรก จะใช้จุดภาพนี้เป็นจุดเริ่มต้นของการติดตามขอบเขตของตัวอักษร ดังแสดงในรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การติดตามขอบภาพเพื่อการตรวจสอบหัว

ในขณะที่ทำการติดตามขอบเขตของภาพ จะบันทึกตำแหน่งและจำนวนของจุดภาพไว้ และเมื่อการติดตามขอบเขตของภาพสิ้นสุดลงที่จุดเริ่มต้น ก็จะใส่ค่า 0ลงไปในตำแหน่งของจุดภาพที่ได้บันทึกไว้(ตัดขอบนอกออกไปจากรูปแบบ) จะได้รูปแบบที่เป็นขอบของภาพตัวอักษรออกมาดังรูปที่ 4.6



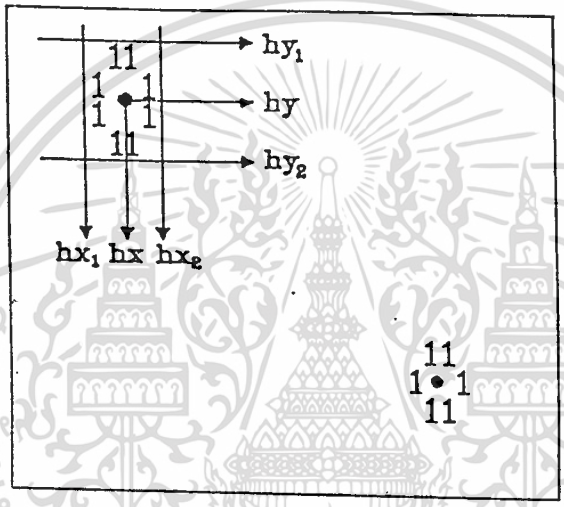
รูปที่ 4.6 ขอบของภาพตัวอักษรที่เป็นหัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อถึงขั้นตอนนี้จะพบว่าหัวของตัวอักษรก็จะปรากฏออกมาดังแสดงในรูปที่ 4.6

ถ้ากำหนดให้ H เป็นจำนวนสูงสุดของหัวในตัวอักษรภาษาไทยแล้ว(ในที่นี้ H = 3)ขั้นตอนต่อไปก็จะทำการติดตามขอบเขตที่เป็นหัวของตัวอักษร โดยกระทำเป็นจำนวน H ครั้ง ซึ่งในแต่ละครั้งก็จะทำการบันทึกเอาตำแหน่งที่ใช้แทนหัวและจำนวนจุดภาพของแต่ละหัวไว้

ตำแหน่งที่ใช้แทนหัวจะหาได้จากการทำ scan ภาพใน 4 ทิศทาง (รูปที่ 4.3) เพื่อหากรอบ บน ล่าง ซ้าย และขวาของหัวดังรูปที่ 4.7



รูปที่ 4.7 การหาตำแหน่งของหัว

ตำแหน่งที่ใช้แทนหัวซึ่งต่อไปจะเรียกว่า ตำแหน่งของหัว จะหาได้จากสมการที่ (4.1) และ (4.2)

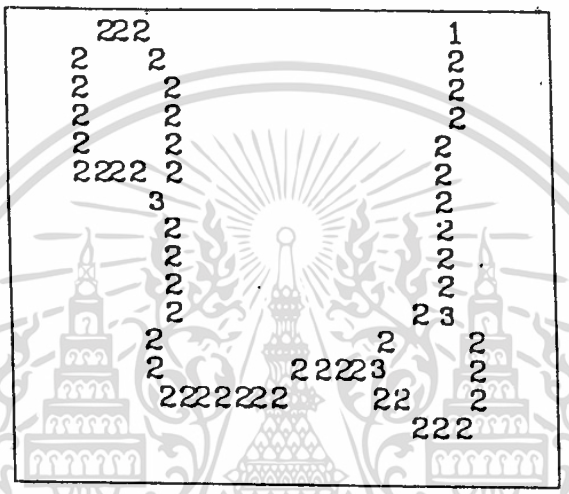
$$hx = (hx_2 - hx_1)/2 \qquad (4.1)$$

$$hy = (hy_2 - hy_1)/2 \qquad (4.2)$$

เมื่อนำเอารูปแบบสองระดับของตัวอักษรที่ผ่านการเตรียมข้อมูลแล้วมาหาค่า NC_8 ก็จะได้รูปแบบตัวอักษรที่ประกอบด้วยค่าตัวเลขต่อเนื่องดังรูปที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะเห็นได้ว่ารูปแบบสองระดับสำหรับใช้ตรวจสอบหัว และรูปแบบสองระดับสำหรับการเตรียมข้อมูลจะกระทำแยกจากกัน โดยจะกระทำกระบวนการใดก่อนก็ได้



รูปที่ 4.8 รูปแบบที่มีค่าตัวเลขต่อเนื่อง

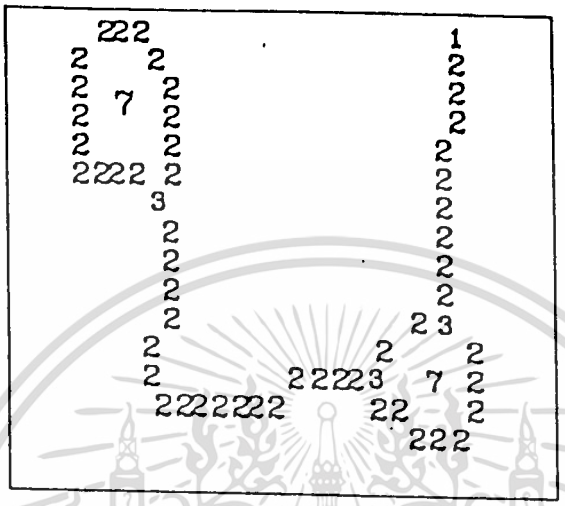
1 คือ จุดปลาย

2 คือ จุดต่อเนื่อง

3 คือ จุดแยก

รวมเอาตำแหน่งของหัวที่คำนวณได้จากสมการ(4.1) และ (4.2) เข้ากับรูปแบบที่มีค่าตัวเลขต่อเนื่องของตัวอักษร จะได้รูปแบบของตัวอักษรออกมาดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 รูปแบบตัวอักษรสำหรับการจำแนกกลุ่ม
1 คือ จุดปลาย 2 คือ จุดต่อเนื่อง
3 คือ จุดแยก 7 คือ จุดหัว

รูปแบบตัวอักษรในรูปที่ 4.9 นั้นจะเป็นรูปแบบที่มีคุณสมบัติต่างๆ ครบถ้วนพร้อมที่จะนำไปจำแนกกลุ่มต่อไป

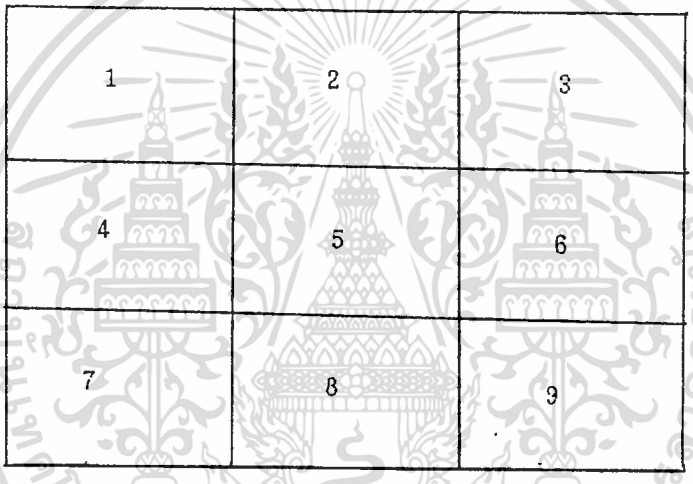
4.2 <u>การแบ่งกลุ่มของตัวอักษร</u> (Rought Classification)

จากการตรวจสอบหัวของตัวอักษรในหัวข้อที่ผ่านมา จะสามารถหาจำนวนและตำแหน่งของหัวในตัวอักษรแต่ละตัวที่ผ่านเข้ามาในระบบการจดจำได้ โดยจะแบ่งตัวอักษรออกเป็น 4 กลุ่ม(class) ตามจำนวนของหัวดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มที่ 1 (class1) คือ กลุ่มที่ไม่มีหัว(no head)
กลุ่มที่ 2 (class2) คือ กลุ่มที่มีหัว 1 หัว(one head)
กลุ่มที่ 3 (class3) คือ กลุ่มที่มีหัว 2 หัว(two head)
กลุ่มที่ 4 (class4) คือ กลุ่มที่มีหัว 3 หัว(three head)

รูปแบบของตัวอักษรในกลุ่มที่ 2 และกลุ่มที่ 3 จะถูกนำมา scan หาขอบเขตตัวของตัวอักษร เพื่อที่จะใช้แบ่งตัวอักษรออกเป็น 9 เขต(Zone) ดังรูปที่ 4.10



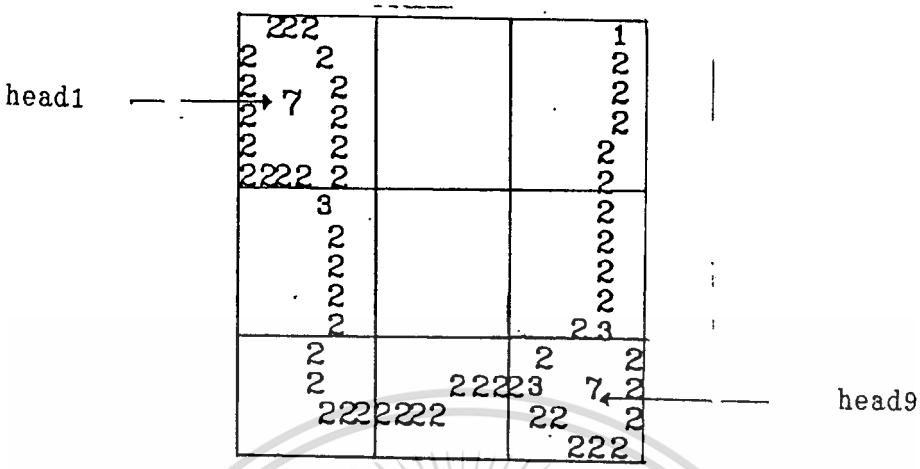
รูปที่ 4.10 การแบ่งเขตของตัวอักษรกลุ่มที่ 2 และกลุ่มที่ 3

เมื่อสังเกตถึงจำนวนหัวของตัวอักษรใน 9 เขตที่แบ่งได้ ก็จะสามารถแบ่งกลุ่มย่อย (Sub Class) ของตัวอักษรได้ดังนี้

กลุ่มย่อย C11 คือ กลุ่มย่อยที่มีหัวอยู่ 1 หัว
กลุ่มย่อย C12 คือ กลุ่มย่อยที่มีหัวอยู่ 2 หัว

ตัวอักษรในแต่ละกลุ่มย่อยจะถูกตรวจสอบเพื่อหาว่ามีหัวอยู่ในเขตไหน และสามารถจำแนกออกเป็นตัวอักษรต่าง ๆ แล้วแสดงการเปรียบเทียบให้เห็นได้ด้วยการ load ตัวอักษรจาก CU FONT ตามค่ารหัส ASCII ที่ สามารถหาได้จาก program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 ตำแหน่งของหัวตัวอักษรในกลุ่มย่อย

รูปที่ 4.11 แสดงถึงตำแหน่งของหัว ที่ปรากฏอยู่ตามเซตต่าง ๆ head1 = 1 และ head9 = 1 มีความหมายว่า ตัวอักษร "น" มีหัวอยู่ในเซตที่ 1 และเซตที่ 9 ตามลำดับ

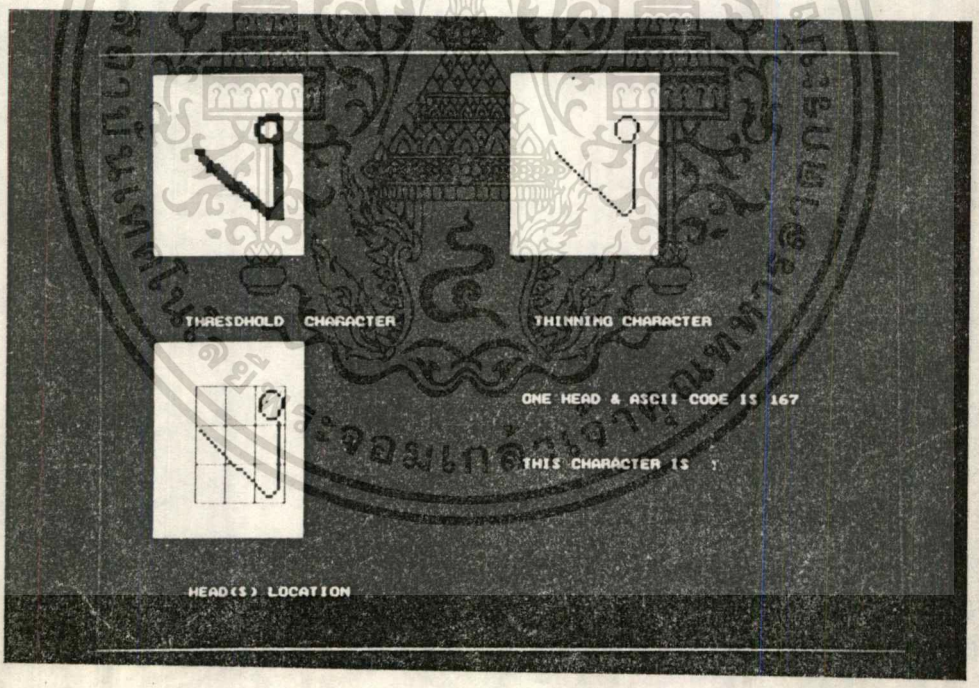
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ผลการทดลองและสรุป

ผลการทดลอง

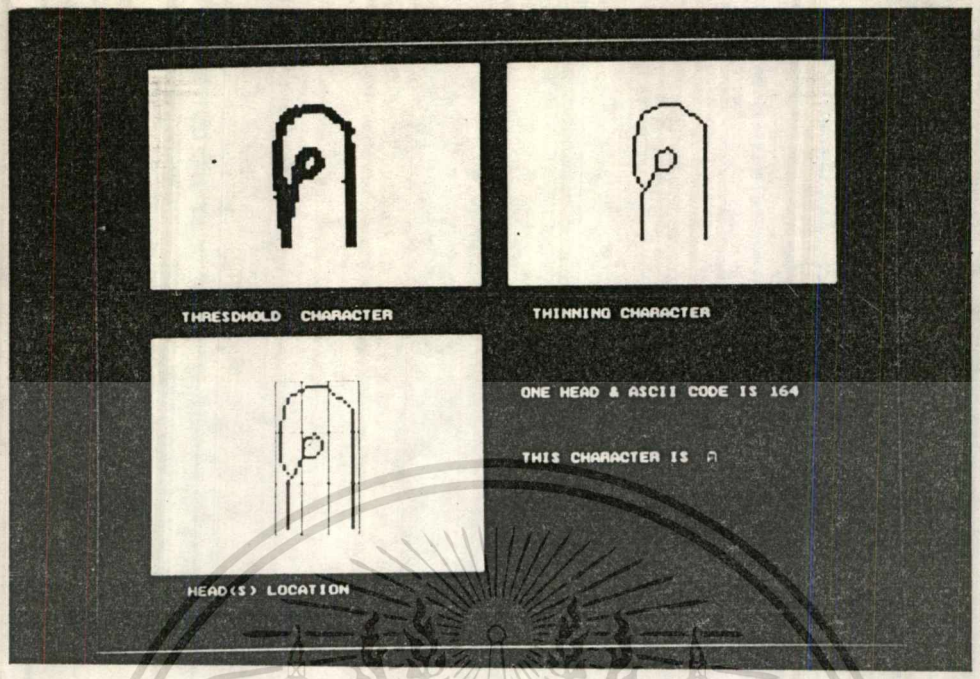
จากขั้นตอนการจดจำตัวอักษร โดยการหาจุดศูนย์กลางของหัวตัวอักษรในแต่ละ zone และแยกออกเป็นตัวอักษรที่มี 1 หัวและ 2 หัว จะได้ผลการทดลองดังรูป

ตัวอักษรที่มีหัว 1 หัว

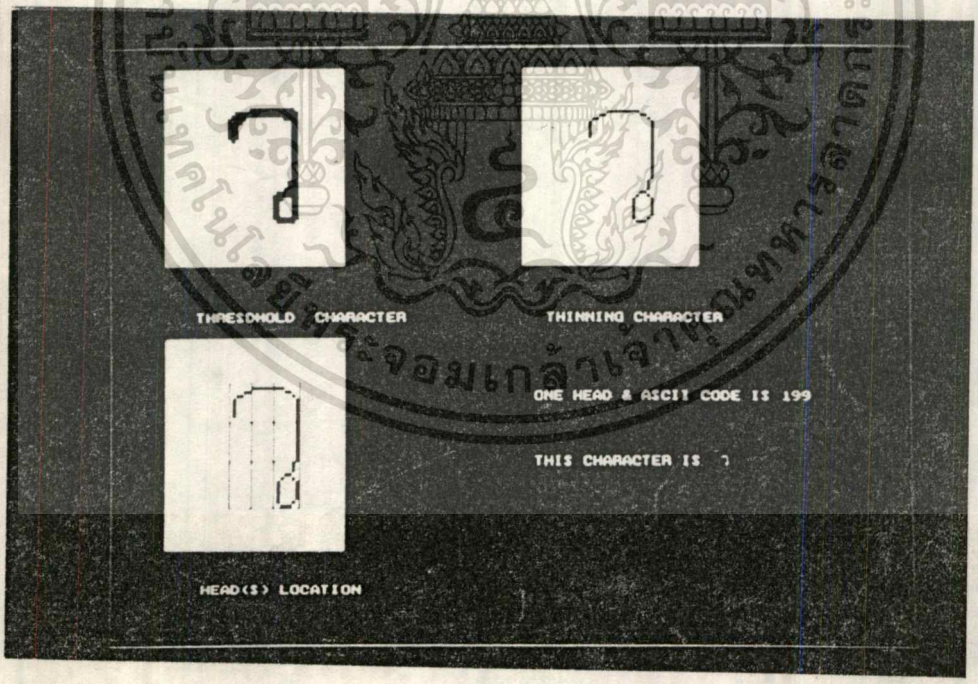


รูปที่ 5.1 แสดงตัวอักษรที่มีหัวอยู่ใน zone ที่ 3(ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

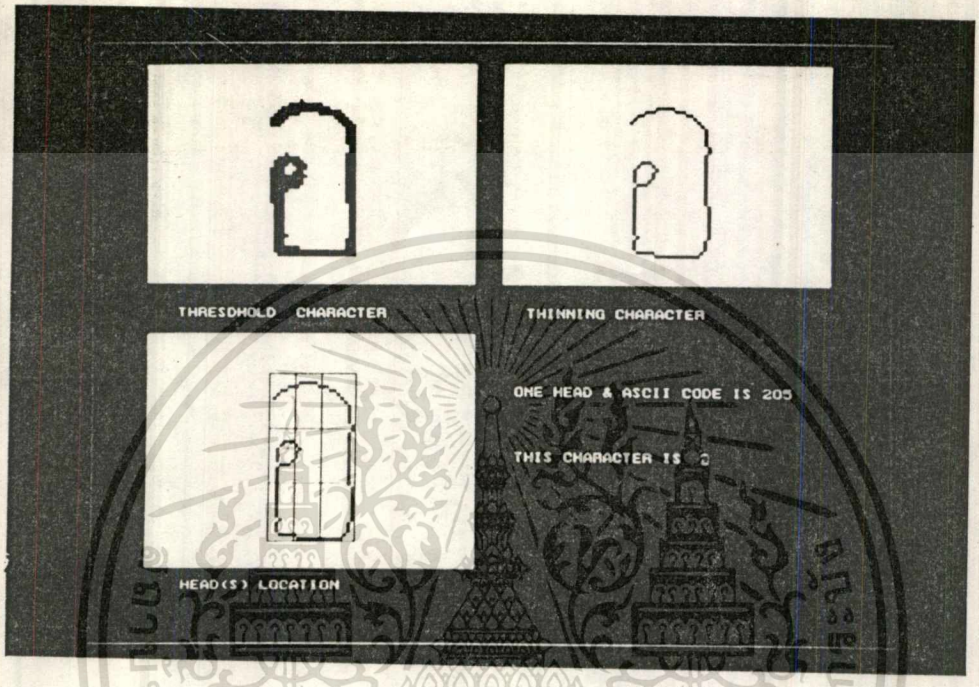


รูปที่ 5.2 แสดงตัวอักษรที่มีหัวอยู่ใน zone ที่ 5(ค)



รูปที่ 5.3 แสดงตัวอักษรที่มีหัวอยู่ใน zone ที่ 9(ว)

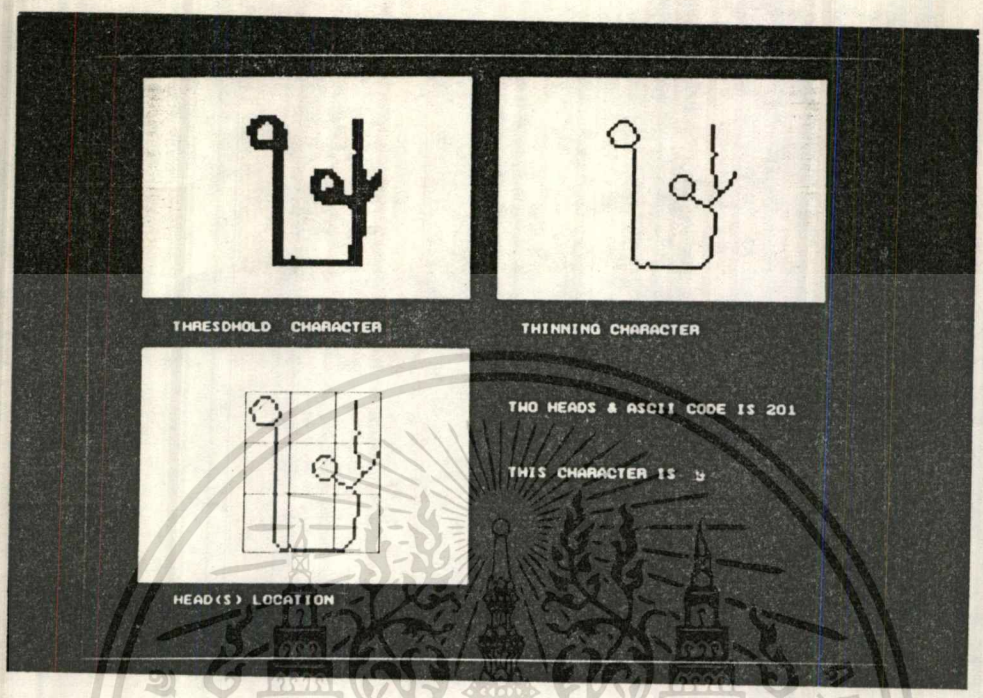
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



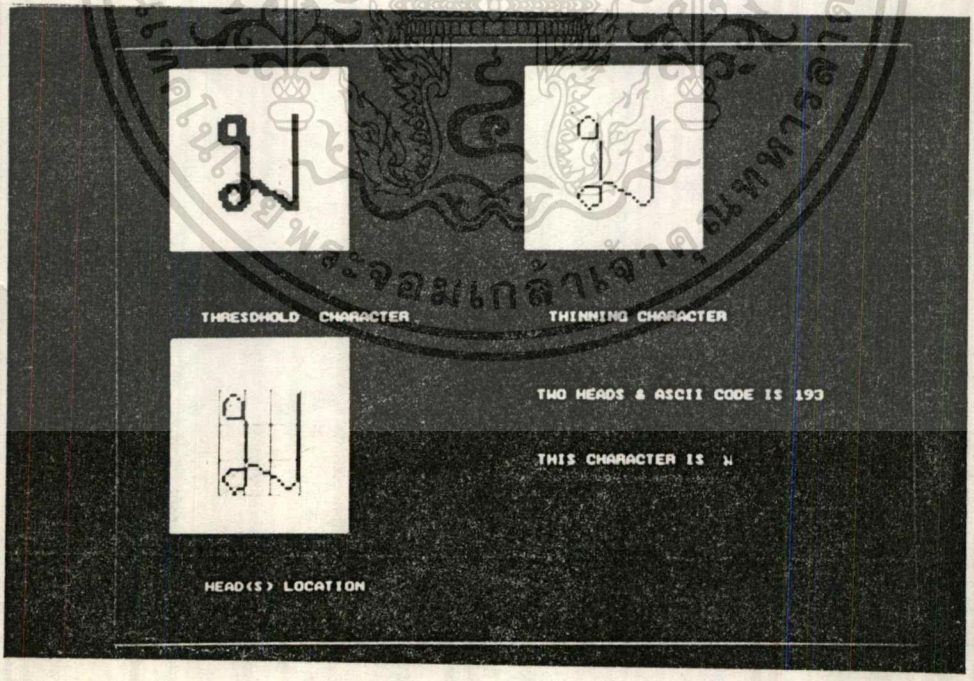
รูปที่ 5.4 แสดงตัวอักษรที่มีหัวอยู่ใน zone ที่ 4 (อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรที่มีหัว 2 หัว

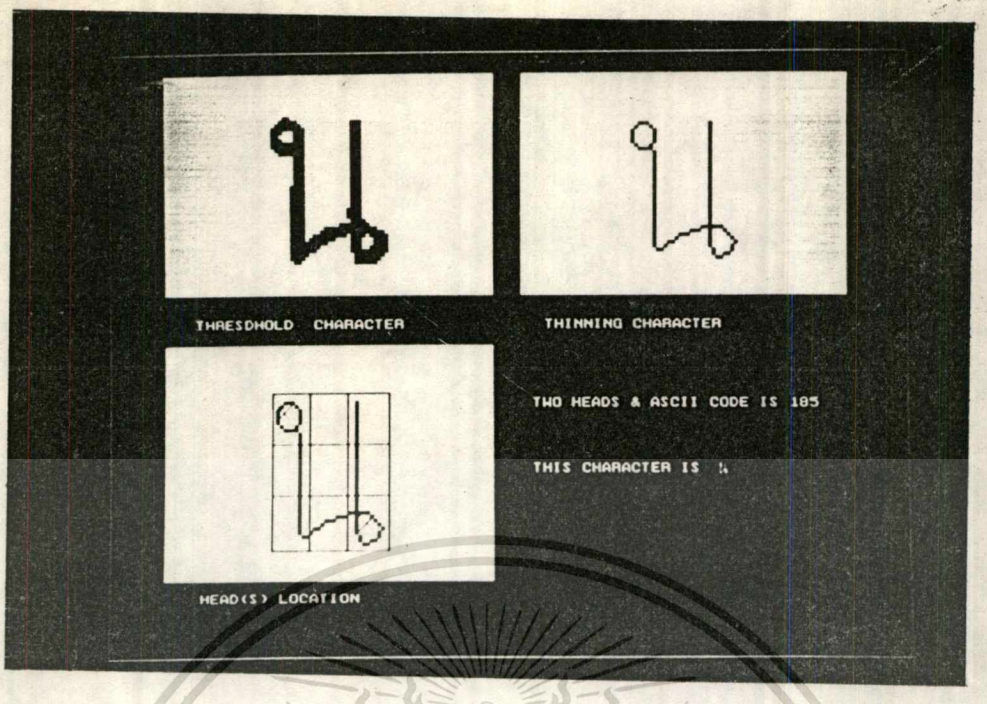


รูปที่ 5.5 แสดงตัวอักษรที่มีหัวอยู่ใน zone ที่ 1 และ 5(ษ)

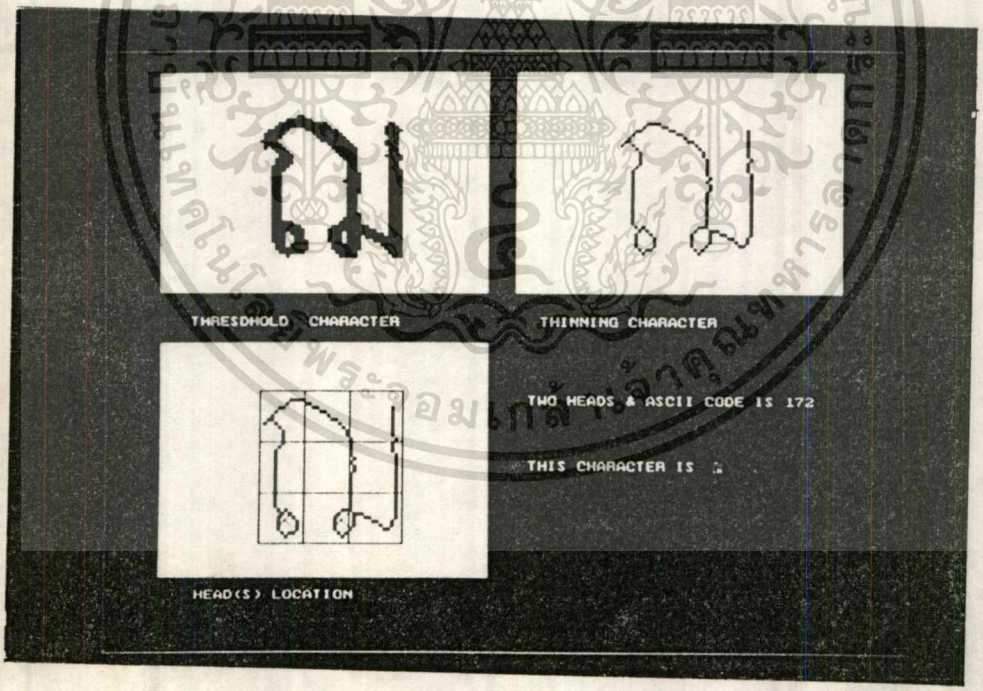


รูปที่ 5.6 แสดงตัวอักษรที่มีหัวอยู่ใน zone ที่ 1 และ 7(ม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 แสดงตัวอักษรที่มีหัวอยู่ใน zone ที่ 1 และ 9(น)



รูปที่ 5.8 แสดงตัวอักษรที่มีหัวอยู่ใน zone ที่ 7 และ 9(ฌ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

จากผลการทดลองใช้ลายมือ ตัวอย่าง 5 ลายมือ จะพบว่าการจดจำตัวอักษรลายมือเขียนภาษาไทยในปริญญานิพนธ์นี้จะมีความถูกต้องถึง 80 % ถ้าตัวอักษรที่ผ่านเข้ามาในขั้นตอนการจดจำนี้มีตำแหน่งหัวของตัวอักษรอยู่ในแต่ละส่วนอย่างแน่นอน หากตัวอักษรใดมีตำแหน่งหัวของตัวอักษรอยู่ก้ำกึ่งกันระหว่างแต่ละส่วนจะเกิดข้อผิดพลาดขึ้น นอกจากนี้ ในกรณีที่ตัวอักษรที่เข้ามาในระบบการจดจำมีตำแหน่งหัวของตัวอักษรอยู่ในส่วนเดียวกันหรือมีความคล้ายคลึงกัน ก็ยังไม่สามารถจดจำได้

ส่วนข้อจำกัดที่สำคัญอีกข้อหนึ่งก็คือ ความหนาของลายมือเขียนที่ผ่านเข้ามาในระบบการจดจำจะต้องมีความหนาพอที่จะทำให้สามารถแยกขอบของตัวอักษรออกมาได้ เพราะขอบของตัวอักษรเป็นส่วนสำคัญที่จะใช้หาตำแหน่งของหัวตัวอักษร ถ้าตัวอักษรบางจนทำให้การติดตามขอบของตัวอักษรผิดพลาดแล้วจะไม่สามารถหาตำแหน่งของหัวที่ถูกต้องได้

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. RAFAEL C. GONZALES , RICHARD E. WOOD "DIGITAL IMAGE PROCESSING" ADDISON-WILEY PLUBISHING COMPANY ,1992.
2. ยืน ภู่วรวรรณ "การประมวลผลสัญญาณดิจิตอล" วารสารเซมิคอนดัคเตอร์อิเลคทรอนิคส์ เล่ม 76-79, ซีเอ็ดยูเคชั่น, 2530.
3. ประสาร ตังติสานนท์ "การจดจำรูปแบบตัวอักษรลัดลายมือภาษาไทยโดยวิธีแยกลักษณะเด่น" วิทยานิพนธ์ ปีการศึกษา 2529 ภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สจล.
4. สุรพันธ์ เอื้อไพบูลย์ "การจดจำตัวอักษรลายมือภาษาไทยโดยการพิจารณาหัวของตัวอักษร" วิทยานิพนธ์ ปีการศึกษา 2531 สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สจล.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ สุรพันธ์ เอื้อไพบูลย์ ที่ได้ให้แนวทางในการทำปริญญานิพนธ์นี้ขึ้นมา
อาจารย์ เทอดศักดิ์ ลิ่วหาทอง ที่กรุณาให้คำปรึกษาในการเขียนโปรแกรม
ขอบคุณ ศิริเดช สมโภชน์ ณัฐพงศ์ ที่ให้คำปรึกษา และเป็นกำลังใจ
กราบขอบพระคุณ คุณพ่อ คุณแม่ ที่ให้การสนับสนุนในทุก ๆ ด้านเสมอมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <dos.h>
#include <alloc.h>
#include <conio.h>
#include <process.h>
#include <graphics.h>
#include <math.h>
#define BUFSIZE 4000 /* internal graphics buffer size */
#define scale 3


void show_pics(long int biwidth,long int biheigth,
     unsigned char new_mat[128][128]);
void thresdhold(long int biwidth,long int biheigth);
void convert(long int biwidth,long int biheigth,FILE *fp);
void position(long int biwidth,long int biheigth);
void thin(long int biwidth,long int biheigth);
void show_pic(long int biwidth,long int biheigth,
     unsigned char new_mat[128][128]);
void edge_detc(long int biwidth,long int biheigth);
void head_ctr(void);
void scan(int upper_x,int upper_y,int lower_x,int lower_y);
void show_pic_out(long int biwidth,long int biheigth,
     unsigned char new_mat[128][128]);
void cross(void);
void check(void);
void ld_font(int asc);
typedef struct tagBITMAPFILEHEADER
{
     int bfType;
     long int bfSize;
     int bfReserved1;
     int bfReserved2;
     long int bfoffBits;
} BITMAPFILEHEADER;

typedef struct tagBITMAPINFOHEADER
{
     long int biSize;
     long int biWidth;
     long int biHeight;
     int      biPlanes;
     int  biBitCount;

     long int biCompression;
     long int biSizeImage;
     long int biXPelsPerMeter;
     long int biYPelsPerMeter;
     long int ClrUsed;
} BITMAPINFOHEADER;

 typedef struct tagRGBQUAD
{
  unsigned char rgbBlue;
  unsigned char rgbGreen;
  unsigned char rgbRed;
  unsigned char rgbReserved;
} RGBQUAD;
  unsigned static char far *matrix;
  unsigned static char far temp[128][128];
  unsigned static char far plot[128][128];
  unsigned static char far pic[128][128];
  int dot[18];
  float huge *hist_array;
  int max_x,max_y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  int num_head;
  int xa,xb,xc,ya,yb,yc,x1,y1,x2,y2;
  int H1,H2,H3,H4,H5,H6,H7,H8,H9;

void main(void)
{
  char String[50];
  int max_dot;
  unsigned int  min_val,max_val, temp_val;
  int  i, j;
  long int bi_width,bi_heigth,offset,num;

BITMAPFILEHEADER  *BmFile;
BITMAPINFOHEADER  *BmInfoHeader;
RGBQUAD           *rgbquad;
FILE    *fp;


  clrscr();
 BmFile = (BITMAPFILEHEADER  *)malloc( sizeof(BITMAPFILEHEADER));
   if (!BmFile){
     printf("\nError: Unable to allocate histogram array\n");
     exit(1);
    }
 BmInfoHeader = (BITMAPINFOHEADER *)malloc( sizeof(BITMAPINFOHEADER))
   if (!BmInfoHeader){
      printf("\nError: Unable to allocate histogram array\n");
      exit(1);
    }
rgbquad= (RGBQUAD *)malloc( sizeof(RGBQUAD)) ;
  if (!rgbquad){
      printf("\nError: Unable to allocate histogram array\n");
      exit(1);
   }

  printf("Enter Bitmap File: ");
  gets(String);
  fp = fopen(String,"r+b");
  if(fp==NULL)
  {
      printf("I can't open your file.");
      exit(0);
  }
  fread(BmFile,sizeof(BITMAPFILEHEADER),1,fp);
  fread(BmInfoHeader,sizeof(BITMAPINFOHEADER),1,fp);
  fread(rgbquad,sizeof(RGBQUAD),1,fp);


  offset   =  BmFile->bfoffBits;
  fseek(fp, offset , SEEK_SET);


 min_val=0;
 max_val=255;
 bi_width = BmInfoHeader->biWidth;
 bi_heigth = BmInfoHeader->biHeight;
 free(BmFile);
 free(BmInfoHeader);
 free(rgbquad);

 if( bi_width>128 || bi_heigth>128)
 {
   printf("\n Your file is too big I can't do anything!!");
   getch();
   exit(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

```
   }

   hist_array = (float *)farcalloc(max_val-min_val+1, sizeof(float));
  if (!hist_array){
    printf("\nError: Unable to allocate histogram array\n");
    exit(1);
  }

  matrix = (unsigned char  *)farcalloc(bi_width*bi_heigth,
       sizeof(unsigned char ));
  if (!matrix){
    printf("\nError: Unable to allocate matrix array\n");
    exit(1);
  }
 convert(bi_width,bi_heigth,fp);

/* Step through the input matrix.  Make sure each value is within
   the proper range for an index into the histogram array.  Then
   increment the histogram value at this index point */

  for (i=0; i<bi_width; i++)
  {
     for (j=0; j<bi_heigth; j++)
     {
      temp_val =plot[i][j];
      temp_val = temp_val > max_val ? max_val : temp_val;
      temp_val = temp_val < min_val ? min_val : temp_val;
      hist_array[min_val+temp_val] += 1.0;
     }
  }

        thresdhold( bi_width,bi_heigth);
        position(bi_width,bi_heigth);
        show_pics(bi_width,bi_heigth,pic);

        edge_detc(bi_width,bi_heigth);
    //  show_pic(bi_width,bi_heigth,plot);
        head_ctr();

        thin(bi_width,bi_heigth);
        check();
        show_pic_out(bi_width,bi_heigth,pic);
        cross();

        getch();
        closegraph();

  }
   void convert(long int biwidth,long int biheigth,FILE *fp)
  {
   int i,j;
   for(i=biheigth-1;i>0;i--)
    {
       for(j=0;j<biwidth;j++)
      {
        fread(matrix,sizeof( unsigned char),1,fp);
        plot[i][j]=(*matrix);
      }
    }
     farfree(matrix);
  }
   void thresdhold(long int biwidth,long int biheigth)
  {
   int i,j,thold;
  /*  printf("\n Enter threshold value from graph:   ");
      scanf("%d",&thold);*/
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    thold=170;
    for(i=0;i<biheigth;i++)
     {
       for(j=0;j<biwidth;j++)
         {
            if(plot[i][j]<thold)
                 temp[i][j]=0;
             else
                 temp[i][j]=255;
         }
     }
}
 void position(long int biwidth,long int biheigth)
{
 int i,j,l,m,flag,xc,yc;
 int Lx,Ly,a,b;
 int top,bottom,left,right,shift_x,shift_y;
 long int Ix,Iy,iF,jF,sum,k;

 xc=biwidth/2;
 yc=biheigth/2;

 //scan from top to bottom
 flag=1;
 i=1;j=0;
 while( temp[i][j] !=0 && flag==1 )
  {
     j++;
     if(j==biwidth)
     {
       j=0;
       i++;
      }
     if(i==biheigth)
       flag=0;
  }
 top=i;
 //scan from bottom to top
   flag=1;
   i=biheigth-1;j=biwidth-1;
   while (temp[i][j] !=0 && flag==1)
  {
    j--;
    if(j==0)
    {
      j=biwidth-1;
      i--;
    }
    if(i==0)
    flag=0;
  }
 bottom=i;
 //scan from left to right
   flag=1;
   i=1;j=0;
   while (temp[i][j] !=0 && flag==1)
  {
    i++;
    if(i==biheigth)
    {
      i=1;
      j++;
    }
    if(j==biwidth)
    flag=0;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 left=j;
 //scan from right to left
   flag=1;
   i=biheigth-1;j=biwidth-1;
   while (temp[i][j] !=0 && flag==1)
  {
    i--;
    if(i==0)
    {
      i=biheigth-1;
      j--;
    }
   if(j==0)
   flag=0;
  }
 right=j;
sum=0;iF=0;jF=0;
 for(i=top;i<bottom;i++)
  {
   for(j=left;j<right;j++)
    {
      sum+=temp[i][j];
      jF+=(j*temp[i][j]);
    }
  }
 for(j=left;j<right;j++)
  {
   for(i=top;i<bottom;i++)
    {
      iF+=(i*temp[i][j]);
    }
  }
if(sum==0)
 {
   printf("\n No image ???  Error!! in your threshold  value. ");
   exit(0);
 }

 Iy=iF/sum;
 Ix=jF/sum;
 shift_x=xc-Ix;
 shift_y=yc-Iy;
 for(i=0;i<biheigth;i++)
     {
       for(j=0;j<biwidth;j++)
         {
            pic[i][j]=0;
         }
     }


   for(i=top;i<bottom;i++)
    {
     for(j=left;j<right;j++)
      {
         if(temp[i][j]==255)
         pic[i+shift_y][j+shift_x]=0;
         else
         pic[i+shift_y][j+shift_x]=(temp[i][j]+1);

      }
    }
x1=left+shift_x;
x2=right+shift_x;
y1=top+shift_y;
y2=bottom+shift_y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  for(i=0;i<biheigth;i++)
   {
     for(j=0;j<biwidth;j++)
       {
         if(pic[i][j]==0)
          temp[i][j]=0;
         else
          temp[i][j]=1;

       }
   }
}
 void edge_detc(long int biwidth,long int biheigth)
{
   int i,j,k,l,p=0,g;
   int pix1,pix2,pix3,pix4;
   int pix5,pix6,pix7,pix0,pix8;
   int A,B,C,D,E,F,G,H,NC4,NC8;
   int begin_i,begin_j,new_i,new_j,ipoint,jpoint;
   int di,dj,f,kk;
   int id[1000],jd[1000];
   int iF,jF;

   for(j=0;j<biwidth;j++)
   {
     for(i=0;i<biheigth;i++)
     {
       plot[i][j]=0; /*clear plot array*/
     }
   }
  /*window for mark edge*/
    for(j=0;j<biwidth;j++)
    {
      for(i=0;i<biheigth;i++)
      {
        if(pic[i][j]==1)
         {
           pix1=pic[i][j+1];
           pix2=pic[i-1][j+1];
           pix3=pic[i-1][j];
           pix4=pic[i-1][j-1];
           pix5=pic[i][j-1];
           pix6=pic[i+1][j-1];
           pix7=pic[i+1][j];
           pix8=pic[i+1][j+1];
           A=pix1-(pix1*pix2*pix3);
           B=pix3-(pix3*pix4*pix5);
           C=pix5-(pix5*pix6*pix7);
           D=pix7-(pix7*pix8*pix1);
           E=(1-pix1)-((1-pix1)*(1-pix2)*(1-pix3));
           F=(1-pix3)-((1-pix3)*(1-pix4)*(1-pix5));
           G=(1-pix5)-((1-pix5)*(1-pix6)*(1-pix7));
           H=(1-pix7)-((1-pix7)*(1-pix8)*(1-pix1));
           NC4=A+B+C+D;
           NC8=E+F+G+H;

           if((NC8==1 && NC4!=3)||(NC8==1 && NC4!=2))
               {
                 plot[i][j]=1;
               }
             else
               {
                 plot[i][j]=0;
               }
         }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
 /*find head of characters*/
  i=0;j=0;f=0;
  while(plot[i][j] !=1)
   {
    j++;
    if(j==biwidth)
     {
       i++;
       j=0;
     }
    if(i==biheigth)
    {
       f=1;
       break;
     }
   }
 begin_i=i;begin_j=j;
 /*mask the first point*/
 plot[begin_i][begin_j]=15;
 /*save array of first point*/
 id[p]=begin_i;jd[p]=begin_j;p++;
 /*assign first point begin to scan*/
 new_i=i-1;
 new_j=j;
 /*assign window to decide head of character*/

 ipoint=new_i-i;  /*difference between current & next co-ordinate*/
 jpoint=new_j-j;
 new_i=i; new_j=j;  kk=1;

/* start scanning*/
 while(kk<=8 && f==0)
  {
    di=ipoint+jpoint;   /*assign next co-ordinate*/
    dj=jpoint-ipoint;

    if(di !=0)
     { di=di/abs(di);}
    if(dj !=0)
     { dj=dj/abs(dj);}
    i=new_i+di; j=new_j+dj;

    if(plot[i][j]!=0)
     {
        if((i==begin_i)&&(j==begin_j))
         {
           di=0;dj=0;
           break;
         }
        else
         {
           plot[i][j]=15;
           id[p]=i; jd[p]=j; p++;
           ipoint=new_i-i;
           jpoint=new_j-j;
           new_i=i; new_j=j;  kk=1;
         }
     }
    else
    {
      ipoint=di;jpoint=dj;
      kk++;
    }
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
     return;
    }
   else
    for(i=0;i<p;i++)
    {
      plot[id[i]][jd[i]]=0;
    }
}

 void thin(long int biwidth,long int biheigth)
       /*thinning input character*/
{
  int i,j;
  int flag=1;
  int k0,k1,k2,k3,k4,k5,k6,k7;
  int sum1,sum2,sum3,sum4,sum5,sum6;
  int sum7,sum8,sum9,sum10,sum11,sum12;


  while(flag==1)
 {
     flag=0;
     for(j=0;j<biwidth;j++)
     {
       for(i=0;i<biheigth;i++)
        {
          if(pic[i][j]==1)
           {
             k0=pic[i][j+1];
             k1=pic[i-1][j+1];
             k2=pic[i-1][j];
             k3=pic[i-1][j-1];
             k4=pic[i][j-1];
             k5=pic[i+1][j-1];
             k6=pic[i+1][j];
             k7=pic[i+1][j+1];
             sum1=k0¦k4¦k5¦k7;
             sum2=k0¦k1¦k3¦k4;
             sum3=k0¦~k1;
             sum4=~k3¦k4;
             sum5=k4¦~k5;
             sum6=k0¦~k7;
             if(k2==0)
               {
                 if((k6 & sum1 & sum3 & sum4)==1)
                 pic[i][j]=8;
               }
             else
              if(k6==0)
                {
                  if((k2 & sum2 & sum5 & sum6)==1)
                  pic[i][j]=8;
                }
           }
        }
 }
   for(i=0;i<biheigth;i++)
   {
     for(j=0;j<biwidth;j++)
     {
       if(pic[i][j]==1)
       {
         k0=pic[i][j+1];
         k1=pic[i-1][j+1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        k3=pic[i-1][j-1];
        k4=pic[i][j-1];
        k5=pic[i+1][j-1];
        k6=pic[i+1][j];
        k7=pic[i+1][j+1];
        sum7=k1¦k2¦k6¦k7;
        sum8=k2¦k3¦k5¦k6;
        sum9=k2¦~k3;
        sum10=~k5¦k6;
        sum11=k6¦~k7;
        sum12=~k1¦k2;
        if(k4==0)
         {
           if((k0 &sum7 &sum9 &sum10)==1)
           pic[i][j]=8;
         }
        else
        if(k0==0)
          {
            if((k4 &sum8 &sum11 &sum12)==1)
            pic[i][j]=8;
          }
         }
    }
  }
   for(i=0;i<biheigth;i++)
 {
   for(j=0;j<biwidth;j++)
    {
      if(pic[i][j]==8)
     {
       flag=1;
       pic[i][j]=0;
     }
    }
  }
}
  show_pic(biwidth,biheigth,pic);
}

   void head_ctr()
{

     int i,j,xbound,ybound;
     xa=x1;xb=xa+((x2-x1)/3);xc=xb+((x2-x1)/3);
     ya=y1;yb=ya+((y2-y1)/3);yc=yb+((y2-y1)/3);
     num_head=0;head=0;
     H1=0;H2=0;H3=0;H4=0;H5=0;H6=0;
     H7=0;H8=0;H9=0;
     /*find head in each zone*/
             /*zone1*/
       scan(xa,ya,xb,yb);
             /* zone 2 */
       scan(xb+1,ya,xc,yb);
             /* zone 3*/
       scan(xc+1,ya,x2,yb);
             /*zone 4*/
       scan(xa,yb+1,xb,yc);
             /*zone 5*/
       scan(xb+1,yb+1,xc,yc);
             /*zone 6*/
       scan(xc+1,yb+1,x2,yc);
             /*zone 7*/
       scan(xa,yc+1,xb,y2);
             /*zone 8*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            /*zone 9*/
        scan(xc+1,yc+1,x2,y2);


  }

   void scan(int upper_x,int upper_y,int lower_x,int lower_y)
  {
    int  i,j;
    int xaz1,xbz1,yaz1,ybz1;
    int cx,cy;
    int ctx,cty;

    i=upper_y;j=upper_x;
    while(plot[i][j]!=1)
   {
     j++;
     if(j==lower_x+1)
     {
       j=upper_x;
       i++;
     }
     if(i==lower_y+1)
     {
       i=-1;
       break;
      }
   }
       xaz1=i+1; //upper

       i=lower_y;j=lower_x;
       while(plot[i][j]!=1)
   {
       j--;
       if(j==upper_x-1)
    {
       j=lower_x;
       i--;
    }
       if(i==upper_y-1)
     {
       i=1;
       break;
      }

}
   xbz1=i-1;//lower
   i=upper_y;j=upper_x;
   while(plot[i][j]!=1)
{
   i++;
   if(i==lower_y+1)
    {
      i=upper_y;
      j++;
    }
    if(j==lower_x+1)
     {
       j=-1;
       break;
     }
} ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
    yaz1=j+1; //left
    i=lower_y;j=lower_x;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
{
    i--;
    if(i==upper_y-1)
    {
      i=lower_y;
      j--;
    }
    if(j==upper_x-1)
     {
       j=1;
       break;
      }

}
    ybz1=j-1;//right
    ctx=xaz1+((xbz1-xaz1)/2);
    cty=yaz1+((ybz1-yaz1)/2);

    switch(head)
{
     case 0:
     dot[0]=ctx;dot[1]=cty;
     plot[dot[0]][dot[1]]=1;
     if(dot[0]!=0 && dot[1]!=0)
     {
      num_head++;
      H1=1;
     }
     break;
     case 1:
     dot[2]=ctx;dot[3]=cty;
     plot[dot[2]][dot[3]]=1;
     if(dot[2]!=0 && dot[3]!=0)
     {
      num_head++;
      H2=1;
     }
     break;
     case 2:
     dot[4]=ctx;dot[5]=cty;
     plot[dot[4]][dot[5]]=1;
     if(dot[4]!=0 && dot[5]!=0)
     {
      num_head++;
      H3=1;
     }
     break;
     case 3:
     dot[6]=ctx;dot[7]=cty;
     plot[dot[6]][dot[7]]=1;
     if(dot[6]!=0 && dot[7]!=0)
     {
      num_head++;
      H4=1;
     }
     break;
     case 4:
     dot[8]=ctx;dot[9]=cty;
     plot[dot[8]][dot[9]]=1;
     if(dot[8]!=0 && dot[9]!=0)
     {
      num_head++;
      H5=1;
     }
     break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      dot[10]=ctx;dot[11]=cty;
      plot[dot[10]][dot[11]]=1;
      if(dot[10]!=0 && dot[11]!=0)
      {
       num_head++;
       H6=1;
      }
      break;
      case 6:
      dot[12]=ctx;dot[13]=cty;
      plot[dot[12]][dot[13]]=1;
      if(dot[12]!=0 && dot[13]!=0)
      {
       num_head++;
       H7=1;
      }
      break;
      case 7:
      dot[14]=ctx;dot[15]=cty;
      plot[dot[14]][dot[15]]=1;
      if(dot[14]!=0 && dot[15]!=0)
      {
       num_head++;
       H8=1;
      }
      break;
      case 8:
      dot[16]=ctx;dot[17]=cty;
      plot[dot[16]][dot[17]]=1;
      if(dot[16]!=0 && dot[17]!=0)
      {
       num_head++;
       H9=1;
       }
      break;
    }
 head++;

}
  void cross(void)
{
  int i,k,l;
  k=max_y/2;
  l=max_x/10;
  setcolor(0);
  for(i=0;i<9;i++)
   {
     if(dot[2*i]!=0 && dot[2*i+1]!=0)
     line((dot[2*i+1]-2)*scale+l,(dot[2*i]-2)*scale+k,
         (dot[2*i+1]+2)*scale+l,(dot[2*i]+2)*scale+k);
     line((dot[2*i+1]-2)*scale+l,(dot[2*i]+2)*scale+k,
         (dot[2*i+1]+2)*scale+l,(dot[2*i]-2)*scale+k);
    }
}
  void check()
{
  int a,b;
  asc=0;
  if(num_head==1)
 {
   if(H1==1) asc=162;
   else if(H3==1) asc=167;
   else if(H4==1) asc=168;
   else if(H5==1) asc=164;
   else if(H7==1) asc=182;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    else if(H8==1) asc=195;
    else if(H9==1) asc=199;
    else asc=0;
  }
    else  if(num_head==2)
  {
    if(H1==1 && H3==1) asc=203;
    else if(H1==1 && H7==1) asc=193;
    else if(H1==1 && H5==1) asc=201;
    else if(H4==1 && H9==1) asc=169;
    else if(H7==1 && H8==1) asc=172;
    else if(H7==1 && H9==1) asc=179;
    else asc=0;
  }
 else asc=0;

}


  void show_pics(long int biwidth,long int biheigth,
       unsigned char new_mat[128][128])
{
  int i,j,k,m;
  char string[30];
  struct palettetype pal;
  int gdriver = DETECT, gmode, errorcode;
  setgraphbufsize(BUFSIZE);
  /* initialize graphics and local variables */
  initgraph(&gdriver, &gmode, "");

  /* read result of initialization */
  errorcode = graphresult();
  if (errorcode != grOk)  /* an error occurred */
  {
     printf("Graphics error: %s\n", grapherrormsg(errorcode));
     printf("Press any key to halt:");
     getch();
     exit(1); /* terminate with an error code */
  }
  getpalette(&pal);

  /* create gray scale */
   for (i=0; i<pal.size; i++)
     setrgbpalette(pal.colors[i], i*8, i*8, i*8);
   max_x=getmaxx();
   max_y=getmaxy();


     for(i=0;i<biheigth;i++)
      {
        for(j=0;j<biwidth;j++)
          {
              for(k=0;k<3;k++)
              {
                for(m=0;m<3;m++)
                {
                     if(new_mat[i][j]==0)
                      putpixel(j*scale+k+max_x/10,i*scale+m+20,15);
                    else
                      putpixel(j*scale+k+max_x/10,i*scale+m+20,0);
                }
              }
          }
      }
       setcolor(15);
       line(0,0,max_x,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      line(0,max_y,max_x,max_y);
      line(0,0,0,max_y);
      line(max_x,0,max_x,max_y);
      sprintf(string,"THRESDHOLD  CHARACTER");
      outtextxy(max_x/10+5,max_y/4+40+40,string);
}

  void show_pic(long int biwidth,long int biheigth,
       unsigned char new_mat[128][128])
{

  int i,j,k,m;
  char string[30];
     for(i=0;i<biheigth;i++)
      {
        for(j=0;j<biwidth;j++)
          {
             for(k=0;k<3;k++)
             {
               for(m=0;m<3;m++)
                {
                  if(new_mat[i][j]==0)
                    putpixel(j*scale+k+(max_x/2),i*scale+m+20,15);
                  else
                    putpixel(j*scale+k+(max_x/2),i*scale+m+20,0);
                }
             }
          }
      }
      free(new_mat);
      sprintf(string,"THINNING CHARACTER");
      outtextxy(max_x/2+10,max_y/4+40+40,string);
}
  void show_pic_out(long int biwidth,long int biheigth,
       unsigned char new_mat[128][128])
{

     int i,j,k,m,l;
     int hx,hy;
     int h1,h2,h3;
     char code[50];
     l=max_x/10;

     for(i=0;i<biheigth;i++)
      {
        for(j=0;j<biwidth;j++)
          {
              for(k=0;k<3;k++)
              {
                for(m=0;m<3;m++)
                 {
                   if(new_mat[i][j]==0)
                     putpixel(j*scale+k+l,i*scale+m+(max_y/2),15);
                   else
                     putpixel(j*scale+k+l,i*scale+m+(max_y/2),0);
                 }
              }
          }
      }
      free(new_mat);
       k=max_y/2;
      setcolor(0);
      line(xa*scale+l,ya*scale+k,x2*scale+l,ya*scale+k);
      line(xa*scale+l,yb*scale+k,x2*scale+l,yb*scale+k);
      line(xa*scale+l,yc*scale+k,x2*scale+l,yc*scale+k);
      line(xa*scale+l,y2*scale+k,x2*scale+l,y2*scale+k);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานที่มีการนำไปใช้

```
      line(xa*scale+1,ya*scale+k,xa*scale+1,y2*scale+k);
      line(xb*scale+1,ya*scale+k,xb*scale+1,y2*scale+k);
      line(xc*scale+1,ya*scale+k,xc*scale+1,y2*scale+k);
      line(x2*scale+1,ya*scale+k,x2*scale+1,y2*scale+k);

      setcolor(15);
      sprintf(code,"HEAD(S) LOCATION");
      outtextxy(max_x/10+10,max_y-50,code);
      if(asc!=0)
        ld_font(asc);


    setcolor(15);
      if(asc!=0)
      {
        if(num_head==1)
          sprintf(code,"ONE HEAD & ASCII CODE IS %d",asc);
       else if(num_head==2)
          sprintf(code,"TWO HEADS & ASCII CODE IS %d",asc);
       }
       else sprintf(code,"NO HEAD, I DON'T KNOW.");
           outtextxy(max_x-340,max_y-200,code);
}
 void ld_font(int asc)
{
    unsigned char normal_fon[5120];
    FILE   *fp;
    long int offset_y;
    int x,y;
    unsigned char bits,bytes,check;
    char fon_name[20];
    char code[30];

    sprintf(fon_name,"normal.fon");
    fp = fopen(fon_name,"r+b");
    if(fp==NULL)
  {
      printf("I can't open your file.");
      exit(0);
  }
   offset_y=asc*20;
   fseek(fp,offset_y,SEEK_SET);
   fread(normal_fon,sizeof(unsigned char),20,fp);
   setcolor(0);
    for(y=0;y<20;y++)
     {
       bytes=normal_fon[y];
       check=0x80;
       for(x=0;x<8;x++)
        {
           bits=bytes&check;
           if(bits!=0)
             putpixel((max_x-178)+x,max_y-158+y,15);
           else
             putpixel((max_x-178)+x,max_y-158+20+y,0);
           check=check>>1;
       }
        setcolor(15);
        sprintf(code,"THIS CHARACTER IS ");
        outtextxy(max_x-340,max_y-150,code);
     }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้