

การจดจำตัวอักษรภาษาไทยเขียนเป็นภาษาไทยโดยพิจารณาหัวของตัวอักษร

RECOGNITION OF HAND-WRITTEN THAI CHARACTER

CONSIDERING THE HEAD OF CHARACTER



วิทยาลัยแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล

สาขาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

เล่มนี้พิมพ์ในโอกาสที่พระอภิมหาเจ้าคุณเทพวราภรณ์ เจ้าคณะจังหวัดสุพรรณบุรี

ปีการศึกษา 2531



เลขหมู่ _____
เลขทะเบียน 2440
วัน, เดือน, ปี 4/12/2552

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ _____ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

แบบฟอร์มการให้คะแนนการสอบวิทยานิพนธ์

สำหรับนักศึกษาระดับมหาบัณฑิต

ชื่อนักศึกษา นายสุรพันธ์ เอื้อไพฑูริย์ เลขประจำตัว 29.0014

ชื่อเรื่องวิทยานิพนธ์ การจดจำตัวอักษรลายมือเขียนภาษาไทยโดยการพิจารณาหัวของ
ตัวอักษร (Recognition of Hand-written Thai
Character Considering the Head of Character)

ชื่ออาจารย์ผู้ควบคุมการสอบ	ลายมือชื่อ	ผลการสอบ
รศ. มนัส	สังวาลศิลป์	ดี
ดร. บุญวัฒน์	อัครชู	ดีเยี่ยม
รศ. ดร. ชม	กัมปาน	ดี
ดร. วรวัฒน์	ลัมภอกา	ดี
ผศ. ครรชิต	ไมตรี	ผ่าน

วันเดือนปี ที่สอบ 17 พฤศจิกายน 2531 เวลา 10.00 น. สถานที่ ห้อง A-305



การจดจำตัวอักษรลายมือเขียนภาษาไทย โดยการพิจารณาหัวของตัวอักษร

Recognition of Hand-written Thai character

considering the Head of character

สุรพันธุ์ เอื้อ ไพบูลย์

SURAPHUN AIRPHAIBOON

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ มนัส สังวรศิลป์

รองศาสตราจารย์ ดร.ชม กิมปาน

ADVISOR

ASSOCIATE PROFESSOR MANUS SANGWORASILP

ASSOCIATE PROFESSOR DR. CHOM KIMPAN

วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

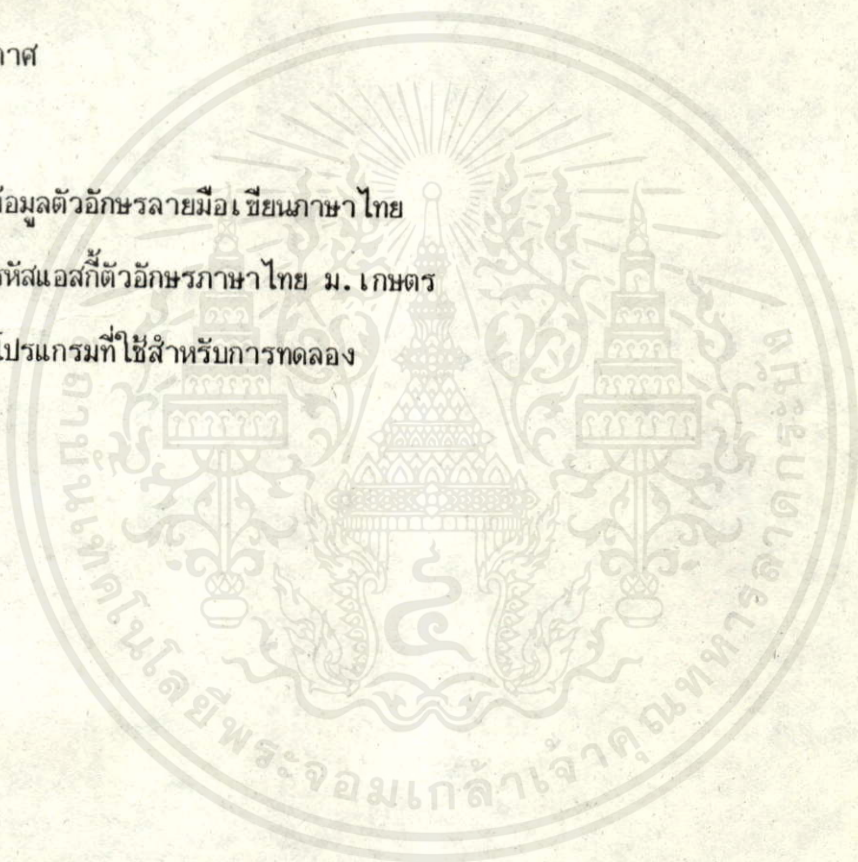
ปีการศึกษา 2531

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษานานับ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract	ข
บทที่ 1 บทนำ	1
บทที่ 2 วัตถุประสงค์และทฤษฎีที่นำมาใช้	4
2.1 วัตถุประสงค์	4
2.2 ความเป็นมาของการจดจำรูปแบบตัวอักษรภาษาไทย	4
2.3 ทฤษฎีที่นำมาใช้	5
2.3.1 การเก็บรูปแบบตัวอักษรโดยใช้กล้อง CCD	5
2.3.2 การคำนวณค่าตัวเลขต่อเนื่อง	8
2.3.3 การหาจุดศูนย์กลาง	13
2.3.4 การตรวจสอบขอบของภาพ	14
บทที่ 3 การเตรียมข้อมูลเพื่อการจำแนกกลุ่มของตัวอักษร	18
3.1 การกำจัดสัญญาณรบกวน	18
3.2 การหาตำแหน่งและขนาดมาตรฐาน	20
3.3 การทำตัวอักษรให้บาง	23
3.4 การกำจัดส่วนเกิน	29
3.5 ค่าตัวเลขที่ต่อเนื่องของรูปแบบตัวอักษร	32
บทที่ 4 การตรวจสอบหัวและการแบ่งกลุ่มของตัวอักษร	35
4.1 การตรวจสอบหัวของตัวอักษร	35
4.2 การแบ่งกลุ่มของตัวอักษรแบบหยายน	43
4.3 การแบ่งกลุ่มของตัวอักษรแบบละเอียด	46
4.4 เทคนิคและวิธีการแยกตัวอักษรที่คล้ายกัน	47

	หน้า
4.4.1 การแบ่งเขตย่อย	47
4.4.2 การทหารหัตถ์สรุปร่าง	49
4.4.3 การหาลักษณะหัวของตัวอักษร	50
4.4.4 การหาอัตราส่วนความกว้างต่อความสูงของตัวอักษร	51
บทที่ 5 การทดลองจำแนกกลุ่มและจดจำตัวอักษร	59
บทที่ 6 บทสรุป	74
กิตติกรรมประกาศ	75
บรรณานุกรม	76
ภาคผนวก ก ข้อมูลตัวอักษรลายมือเขียนภาษาไทย	79
ภาคผนวก ข รหัสแอสกีตัวอักษรภาษาไทย ม. เกษตร	95
ภาคผนวก ค โปรแกรมที่ใช้สำหรับการทดลอง	97



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพประกอบ

	หน้า
รูปที่ 1.1 บล็อกไดอะแกรมของการป้อนข้อมูลเข้าด้วยการใช้แป้นพิมพ์	1
รูปที่ 1.2 การเก็บรูปแบบของตัวอักษร	2
รูปที่ 2.1 การรับภาพตัวอักษรจากกล้อง CCD	6
รูปที่ 2.2 ลักษณะการสแกนของกล้อง CCD	6
รูปที่ 2.3 การเปลี่ยนสัญญาณ อนาล็อก เป็น ดิจิตอล	7
รูปที่ 2.4 ฮาร์ดแวร์สำหรับการทำรูปแบบสองระดับ	7
รูปที่ 2.5 ลักษณะของรูปแบบสองระดับ	8
รูปที่ 2.6 ลักษณะของจุดภาพข้างเคียง 2 แบบ	9
รูปที่ 2.7 ค่า NC_4 และ NC_8 ของจุดภาพ	10
รูปที่ 2.8 ตัวอย่างการคำนวณค่า NC_4 และ NC_8	12
รูปที่ 2.9 จุดศูนย์ถ่วงของภาพสองระดับ	13
รูปที่ 2.10 ค่าตัวเลขต่อเนื่องที่เป็นขอบของภาพ	14
รูปที่ 2.11 การสแกนหาจุดภาพที่เป็นจุดเริ่มต้น	15
รูปที่ 2.12 จุดเริ่มต้นของการติดตามหาขอบภาพ	15
รูปที่ 2.13 การติดตามหาจุดภาพที่เป็นขอบ	16
รูปที่ 2.14 ขอบของภาพที่ได้จากการติดตามขอบเขต	16
รูปที่ 3.1 ตารางหน้าตาสำหรับการกำจัดสัญญาณรบกวน	18
รูปที่ 3.2 การกำจัดสัญญาณรบกวนบนรูปแบบ	19
รูปที่ 3.3 ตำแหน่งต่างๆของตัวอักษรภายในรูปแบบ	20
รูปที่ 3.4 การย้ายภาพตัวอักษรไปยังตำแหน่งมาตรฐาน	20
รูปที่ 3.5 ทิศทางของการสแกนทั้ง 4 แบบ	21
รูปที่ 3.6 ภาพตัวอักษรที่ย้ายมายังตำแหน่งมาตรฐาน	22

	หน้า
รูปที่ 3.7 การหาขนาดมาตรฐาน	22
รูปที่ 3.8 ตัวอักษรที่มีขนาดมาตรฐาน	23
รูปที่ 3.9 โครงร่างของตัวอักษร	24
รูปที่ 3.10 ตารางหน้าตาสำหรับการทำให้บาง	24
รูปที่ 3.11 ตารางหน้าตาสำหรับการตรวจสอบหาขอบ	25
รูปที่ 3.12 ตารางหน้าตาต่างของจุดภายในโครงร่าง	26
รูปที่ 3.13 โครงร่างของตัวอักษร	28
รูปที่ 3.14 ตารางหน้าตาต่างสำหรับการกำจัดส่วนเกิน	30
รูปที่ 3.15 บล็อก ไดอะแกรมของการกำจัดส่วนเกิน	32
รูปที่ 3.16 ค่าตัวเลขต่อเนื่องบนโครงร่างของตัวอักษร	33
รูปที่ 3.17 ผังงาน แสดงระบบการจัดการล่วงหน้า	34
รูปที่ 4.1 จำนวนและตำแหน่งของหัว	35
รูปที่ 4.2 ลักษณะการเขียนตัวอักษร	36
รูปที่ 4.3 รูปแบบสองระดับสำหรับการตรวจสอบหัว	37
รูปที่ 4.4 ขอบของตัวอักษร	38
รูปที่ 4.5 การติดตามขอบภาพเพื่อการตรวจสอบหัว	39
รูปที่ 4.6 ขอบของภาพตัวอักษรที่เป็นหัว	40
รูปที่ 4.7 การหาตำแหน่งของหัว	41
รูปที่ 4.8 รูปแบบที่มีค่าตัวเลขต่อเนื่อง	42
รูปที่ 4.9 รูปแบบตัวอักษรสำหรับการจำแนกกลุ่ม	43
รูปที่ 4.10 การแบ่งเขตของตัวอักษรกลุ่มที่ 1 และ กลุ่มที่ 3	44
รูปที่ 4.11 การแบ่งเขตของตัวอักษรกลุ่มที่ 2	45
รูปที่ 4.12 ตำแหน่งหัวของตัวอักษรในกลุ่มย่อย	46
รูปที่ 4.13 ตำแหน่งของจุดปลาย	46

	หน้า
รูปที่ 4.14 ตำแหน่งของหัวที่อาจจะไม่แน่นอน	48
รูปที่ 4.15 การแบ่งเขตย่อย	48
รูปที่ 4.16 การแบ่งเขตย่อย	49
รูปที่ 4.17 รหัสรูปร่างของตัวอักษร	49
รูปที่ 4.18 ลักษณะของหัว	50
รูปที่ 4.19 รหัสรูปร่างของหัว	50
รูปที่ 4.20 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มที่ 1	51
รูปที่ 4.21 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มที่ 2	52
รูปที่ 4.22 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย H1	52
รูปที่ 4.23 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย H2	53
รูปที่ 4.24 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย H3	53
รูปที่ 4.25 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มที่ 3	54
รูปที่ 4.26 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย C12	54
รูปที่ 4.27 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย C13	55
รูปที่ 4.28 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย C14	55
รูปที่ 4.29 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย C33	56
รูปที่ 4.30 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย C34	56
รูปที่ 4.31 โครงร่างแบบต้นไม้สำหรับตัวอักษรระดับบน	57
รูปที่ 4.32 โครงร่างแบบต้นไม้สำหรับตัวอักษรระดับล่าง	57
รูปที่ 4.33 ผังงานของระบบการจดจำตัวอักษร	58
รูปที่ 5.1 ระบบที่ใช้ในการทดลอง	59
รูปที่ 5.2 รูปแบบของตัวอักษรที่ได้จากกล้อง CCD	60
รูปที่ 5.3 รูปแบบสองระดับของตัวอักษร ขนาด 64 X 64 จุดภาพ	61
รูปที่ 5.4 รูปแบบที่มาตรฐาน	61

	หน้า
รูปที่ 5.5 ผลที่ได้จากการตรวจสอบหัว	63
รูปที่ 5.6 ผลที่ได้จากการเตรียมข้อมูลตัวอักษร	64
รูปที่ 5.7 ผลจากการรวมรูปแบบ	64
รูปที่ 5.8 ผลจากการแบ่งเขตของตัวอักษร	65
รูปที่ 5.9 ผลการจดจำตัวอักษร "ม"	66
รูปที่ 5.10 ผลการจดจำตัวอักษร "ธ"	66
รูปที่ 5.11 ผลการจดจำตัวอักษร "ง"	67
รูปที่ 5.12 ผลการจดจำตัวอักษร "ด"	68
รูปที่ 5.13 ผลการจดจำตัวอักษร "ถ"	68
รูปที่ 5.14 ผลการจดจำตัวอักษร "ณ"	69
รูปที่ 5.15 ผลการจดจำตัวอักษร "ฐ"	69
รูปที่ 5.16 ผลการจดจำตัวอักษร "ช"	70
รูปที่ 5.17 ผลการจดจำตัวอักษร "ข"	70
รูปที่ 5.18 ผลการจดจำตัวอักษรที่มีลักษณะ เอน	71
รูปที่ 5.19 ผลการจดจำตัวอักษรที่มีลักษณะ เอน ไปทางด้านหน้ามากๆ	72
รูปที่ 5.20 ผลการจดจำตัวอักษรที่มีลักษณะ เอน ไปทางด้านหลังมากๆ	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตารางแสดงความสัมพันธ์ระหว่างจุดภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ลักษณะของตัวอักษรภาษาไทยนอกจากจะประกอบไปด้วยเส้นตรง เส้นโค้ง แล้วยังประกอบด้วยส่วนที่เป็นวงกลมเล็กๆ อยู่ด้วยซึ่งจะเรียกส่วนนี้ว่า "หัว" ของตัวอักษร หัวของตัวอักษรแต่ละตัวจะมีตำแหน่งที่ต่าง ๆ กันออกไปหรือบางตัวอักษรก็ไม่มีหัวเลย จึงถือได้ว่าหัวของตัวอักษรก็เป็นลักษณะที่เด่นชัดอย่างหนึ่งสำหรับตัวอักษรภาษาไทย จะสามารถสังเกตได้อย่างหนึ่งว่า ในการเขียนตัวอักษรภาษาไทยตามกรรมวิธีการเขียนที่ถูกต้องแล้วนั้น ส่วนมากแล้ว จะเริ่มโดยการเขียนหัวของตัวอักษรก่อนเสมอ ยกเว้นตัวอักษรบางตัวที่ไม่มีหัวเท่านั้น

ดังนั้น วิทยานิพนธ์ฉบับนี้จึงได้นำเสนอถึง การจดจำตัวอักษรภาษาไทย ที่เป็นลายมือเขียน (หรือตัวพิมพ์ก็ได้) โดยการนำเอาลักษณะเด่นคือ หัวของตัวอักษร มาใช้ในการจำแนกกลุ่มของตัวอักษร (Rough Classification) และใช้คุณสมบัติทางโทโพโลยี (Topology) ของตัวอักษร ในการแยกตัวอักษรออกจากกลุ่ม (Fine Classification) ตลอดจนนำเอาเทคนิคในการแยกตัวอักษรหลาย ๆ วิธีมาใช้แยกตัวอักษรที่มีความคล้ายคลึงกันออกจากกันด้วย

จากการทดลองจดจำตัวอักษรด้วยวิธีที่นำเสนอนี้ โดยใช้ตัวอักษรลายมือเขียนจากบุคคล 15 คน (แสดงไว้ในภาคผนวก ก) ปรากฏว่าสามารถจดจำตัวอักษรได้ผลดีถึง 98.82% ผิดพลาด 1.18% ใช้เวลาในการประมวลผล 1.35 วินาที/ ตัวอักษร (เขียนโปรแกรมด้วยภาษาซี บนเครื่อง IBM PC/AT)

Abstract

Most of Thai character characteristic composed of straight lines and curves. Moreover of these, there always presents a small circle portion, thus called **Head of character**. The head of character appears in various locations and some character have no head. In general, almost of Thai characters are written by the head first. By this dominant characteristic, the presentation of head, we can classify and recognize Thai characters.

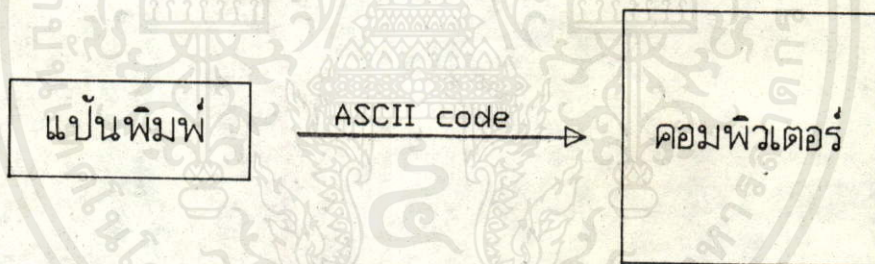
This thesis shall present the recognition of hand-written (or typing style) of Thai characters. The special characteristic of Thai character (Head presentation) is used in rough classification, and some other topological properties are used in fine classification. Additionally some more special techniques are involved in classification of almost similar characters.

Some experimental data is investigated by this proposed technique, the recognition result of 15 hand-written copies accurate 98.82% and the processing time is 1.35 sec / character.

บทที่ 1

บทนำ

ปัจจุบันนี้ไมโครคอมพิวเตอร์ได้เข้ามามีบทบาทในงานสาขาต่าง ๆ ของมนุษย์มากมาย เช่น การใช้คอมพิวเตอร์ เก็บและวิเคราะห์ข้อมูลทางด้านธุรกิจ คำนวณข้อมูลทางวิทยาศาสตร์และวิศวกรรม ฯลฯ งานทุกอย่างที่มีคอมพิวเตอร์เกี่ยวข้องอยู่ด้วยแล้ว จะต้องมีการปฏิบัติการสิ่งหนึ่งเกิดขึ้นเสมอๆ สิ่งนั้นคือ การป้อนข้อมูลเข้า (Data entry) อุปกรณ์ในการป้อนข้อมูล ที่มีการนำมาใช้กันมากกับเครื่องไมโครคอมพิวเตอร์ ก็คือแป้นพิมพ์ (Keyboard) การป้อนข้อมูลโดยใช้แป้นพิมพ์นั้น เป็นการส่งรหัสให้แก่เครื่องคอมพิวเตอร์เช่น รหัสแอสกี (ASCII) ของตัวอักษรต่าง ๆ รวมทั้งตัวอักษรภาษาไทย ก็ได้มีการกำหนดรหัสแทนตัวอักษรขึ้นมาใช้กันหลาย ๆ แบบ



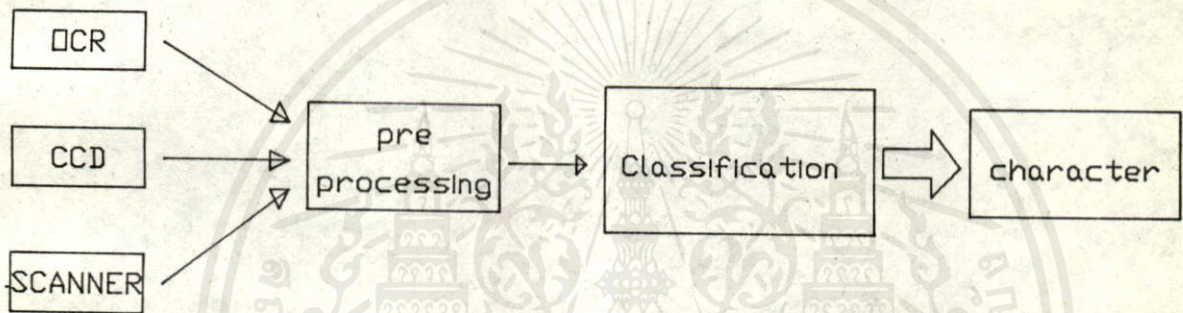
รูปที่ 1.1 บล็อกไดอะแกรมของการป้อนข้อมูลเข้าด้วยการใช้แป้นพิมพ์

จากบล็อกไดอะแกรมในรูปที่ 1.1 จะเห็นว่าคอมพิวเตอร์ไม่รับรู้ลักษณะของตัวอักษรที่ถูกพิมพ์เข้าไป แต่จะรับรู้เพียงรหัสแทนตัวอักษรเท่านั้น ดังนั้นจึงทำให้เกิดความยุ่งยากที่จะทำให้คอมพิวเตอร์สามารถเข้าใจถึงลักษณะของตัวอักษรแบบต่าง ๆ ที่มนุษย์ใช้กันอยู่

วิธีการอย่างหนึ่งที่จะสามารถพัฒนาความฉลาดของคอมพิวเตอร์ ในการเรียนรู้รูปแบบของตัวอักษรได้ก็คือ การป้อนข้อมูลเข้าเครื่องคอมพิวเตอร์โดยการใช้อุปกรณ์รับรู้ทางแสง ที่สามารถเปลี่ยน

สัญญาณแสง ให้เป็นสัญญาณทางไฟฟ้าที่คอมพิวเตอร์นำไปประมวลผลได้ อุปกรณ์เหล่านี้ มีหลายชนิด เช่น เครื่องอ่านตัวอักษรด้วยแสง หรือ OCR (Optical Character Reader) [1] กล้องรับภาพ และ เครื่องสแกนเนอร์ (Scanner) ชนิดต่าง ๆ

เมื่อรูปแบบของตัวอักษร ได้ถูกอ่านเข้ามาด้วยอุปกรณ์รับรู้ทางแสง ก็จะต้องนำเอาสัญญาณทางไฟฟ้าของตัวอักษรไปวิเคราะห์ออกมาว่า สัญญาณไฟฟ้านั้น เป็นสัญญาณไฟฟ้าของตัวอักษรตัวใด ดังแสดงในรูปที่ 1.2



รูปที่ 1.2 การเก็บรูปแบบของตัวอักษร

วิทยานิพนธ์ฉบับนี้ นำเสนอถึงวิธีการใหม่ในการจดจำตัวอักษรภาษาไทยที่เป็น ลายมือเขียน เนื้อหาในบทต่าง ๆ มีดังต่อไปนี้

บทที่ 1 เป็นการแนะนำระบบการป้อนข้อมูลเข้าเครื่องคอมพิวเตอร์ (Data Entry) ด้วย อุปกรณ์ต่าง ๆ

บทที่ 2 จะกล่าวถึงประวัติความเป็นมาถึงวิธีการที่จะให้คอมพิวเตอร์ สามารถเรียนรู้และ แยกแยะลักษณะของตัวอักษร รวมทั้งทฤษฎีทั้งหลายที่จะนำไปใช้ สำหรับการจดจำตัวอักษรลายมือเขียน ภาษาไทยในวิทยานิพนธ์นี้ เช่นการหาค่าตัวเลขต่อเนื่อง (Connectivity number) การหาจุดศูนย์กลาง (Center of Gravity) ของภาพสองระดับ ฯลฯ

บทที่ 3 เป็นการเตรียมรูปแบบข้อมูลของตัวอักษรภาษาไทย (Pre-processing) ก่อนที่จะนำไปจำแนก เช่น การรวมวิธีในการลดสัญญาณรบกวน การรวมวิธีในการทำให้บาง การกำจัดส่วนเกินของตัว

อักษรลายมือเขียน เป็นต้น

บทที่ 4 กล่าวถึงการตรวจสอบหาลักษณะเด่น ซึ่งก็คือหัวของตัวอักษรตลอดจนการใช้ลักษณะของหัว ในการแบ่งกลุ่มตัวอักษร การแยกตัวอักษรแต่ละตัวออกจากกลุ่มย่อย และ การพิจารณาตัวอักษรที่มีความคล้ายคลึงกันมากบางตัว

บทที่ 5 เป็นการทดลองจดจำตัวอักษรลายมือเขียนภาษาไทย โดยใช้ข้อมูลลายมือเขียนภาษาไทย จากผู้เขียนจำนวน 15 คน

บทที่ 6 เป็นการสรุป และวิเคราะห์ผลการทดลอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์การค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

วัตถุประสงค์และทฤษฎีที่นำมาใช้

แนวทางที่จะเป็นไปได้ ในการพัฒนาระบบการจดจำรูปแบบตัวอักษรภาษาไทยให้สามารถนำไปใช้งานได้ นั้น มีความจำเป็นอย่างยิ่งที่จะต้องศึกษาถึงทฤษฎี ที่เกี่ยวข้องกับภาพหรือ การประมวลผลภาพ (Image processing) พร้อมทั้งต้องมีการสังเกตลักษณะเด่นของตัวอักษรแต่ละตัว ลักษณะความคล้ายกันของตัวอักษรบางตัว ทั้งนี้ก็เพราะว่า ลักษณะเหล่านี้เอง ที่จะเป็นตัวบ่งบอกถึงประสิทธิภาพของระบบการจดจำรูปแบบตัวอักษรภาษาไทย โดยเฉพาะตัวอักษรลายมือเขียน ซึ่งมีรูปร่างที่ไม่แน่นอน

2.1 วัตถุประสงค์

จุดมุ่งหมายของวิทยานิพนธ์นี้ก็คือ ต้องการที่จะพัฒนาระบบของ การป้อนข้อมูล เข้าเครื่องคอมพิวเตอร์ ให้มีความชาญฉลาดใกล้เคียงกับการรับข้อมูลของมนุษย์ โดยเฉพาะการป้อนข้อมูล ที่เป็นตัวอักษรลายมือเขียนในภาษาไทยทุก ๆ ตัวอักษร เพื่อที่จะได้นำไปประยุกต์ใช้ให้เป็นประโยชน์ กับงานทางด้านต่าง ๆ เช่นการเขียนข้อความ การมองเห็นของหุ่นยนต์ การเขียนหนังสือของคนพิการ เหล่านี้เป็นต้น

2.2 ความเป็นมาของการจดจำรูปแบบตัวอักษรภาษาไทย

เมื่อเทคโนโลยีทางด้านอุปกรณ์รับรู้ทางแสง มีการพัฒนาขึ้นมา จนมีประสิทธิภาพสูงพอที่จะนำมาใช้งานได้ ในต่างประเทศก็ได้มีการทำวิจัยกันอย่างกว้างขวางเกี่ยวกับเรื่อง การจดจำรูปแบบ (Pattern Recognition) เช่น การจดจำรูปแบบของตัวอักษรในภาษาต่าง ๆ (Character Recognition) [2]- [6] ตัวอักษรที่ใช้ส่วนมากแล้วจะเป็นตัวพิมพ์ (Printed Character) ส่วนผลที่ได้จากการจดจำตัวอักษรจะ ได้ผลมากหรือน้อย ก็ขึ้นอยู่กับกรรมวิธีที่ใช้วิเคราะห์และแยกแยะตัวอักษร รวมทั้งขึ้นอยู่กับลักษณะ โครงสร้างของตัวอักษรในแต่ละภาษาด้วย

กรรมวิธีในการแยกแยะตัวอักษรที่ใช้กันอยู่ในปัจจุบันจะแบ่งได้ เป็น 2 ประเภทคือ

1. การวิเคราะห์ตัวอักษรโดยการใช้คณิตศาสตร์ (Mathematic analysis) [18], [17]
2. การวิเคราะห์โครงสร้างของตัวอักษร (Syntactic analysis) [18], [19]

การวิเคราะห์และแยกแยะตัวอักษร โดยการใช้สมการทางคณิตศาสตร์นั้น สามารถทำได้เร็วกว่า การวิเคราะห์โครงสร้าง แต่ถ้ากลุ่มของตัวอักษร มีความคล้ายคลึงกันมากแล้ว วิธีนี้จะแยกแยะตัวอักษรออกมาโดยมีความผิดพลาดสูง คณิตศาสตร์ที่นำมาใช้ เช่น สมการการตัดสินใจแบบเชิงเส้น (Linear decision function) หรือการวัดการกระจายโดยการใช้ K-L expansion (Karhunen-Loeve expansion) หรือการคำนวณค่าลักษณะต่อเนื่องของจุดภาพ (Connectivity) เป็นต้น

ส่วนการวิเคราะห์โครงสร้างของตัวอักษรนั้น ถ้าหากว่า จำนวนของตัวอักษรที่ต้องการจดจำมีมากแล้ว เวลาในการจดจำก็จะมากขึ้นไปด้วย ตัวอย่างของการวิเคราะห์โครงสร้าง เช่น การเข้ารหัสของลายเส้นแบบต่าง ๆ การพิจารณาของเส้น ส่วนโค้ง (Curve) เป็นต้น

ในประเทศไทย ก็มีการทำวิจัยในเรื่องการจดจำรูปแบบตัวอักษรภาษาไทยกันบ้างแต่ไม่ค่อยจะแพร่หลาย ทั้งนี้ก็อาจจะเป็นเพราะอุปกรณ์รับรู้ทางแสงนี้มิได้ใช้กันน้อยมาก เนื่องจากมีราคาสูง ส่วนกรรมวิธีในการวิเคราะห์นั้นก็มีการใช้วิธีการที่แตกต่างกันออกไป เช่น การใช้ Linear Decision Function [2] การใช้ K-L expansion [3] การวิเคราะห์ความโค้ง (Concave) และเว้า (Convex) [5], [6] การหาลักษณะเด่นของตัวอักษร (Feature concentrated) [20] เป็นต้น ซึ่งส่วนใหญ่แล้ว จะเป็นการจดจำตัวอักษรที่เป็นตัวพิมพ์ และผลการจดจำที่ได้ ก็ยังไม่เพียงพอที่จะนำไปประยุกต์ใช้ ในสายงานต่าง ๆ ได้

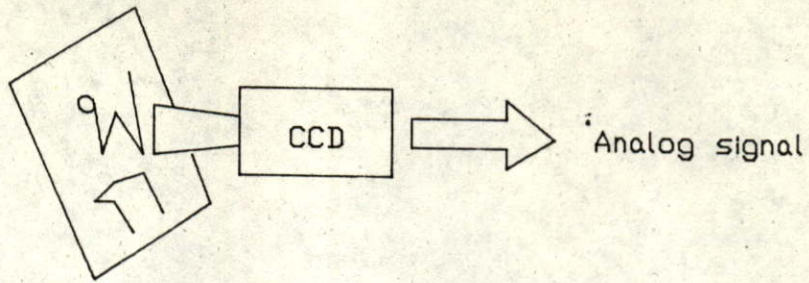
2.3 ทฤษฎีที่นำมาใช้

2.3.1 การเก็บรูปแบบตัวอักษรโดยการใช้กล้อง CCD

กล้อง CCD จะทำการเปลี่ยนสัญญาณแสงจากตัวอักษรที่ผ่านเข้ามา ให้กลายเป็นสัญญาณไฟฟ้าที่ต่อเนื่อง (Analog signal)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

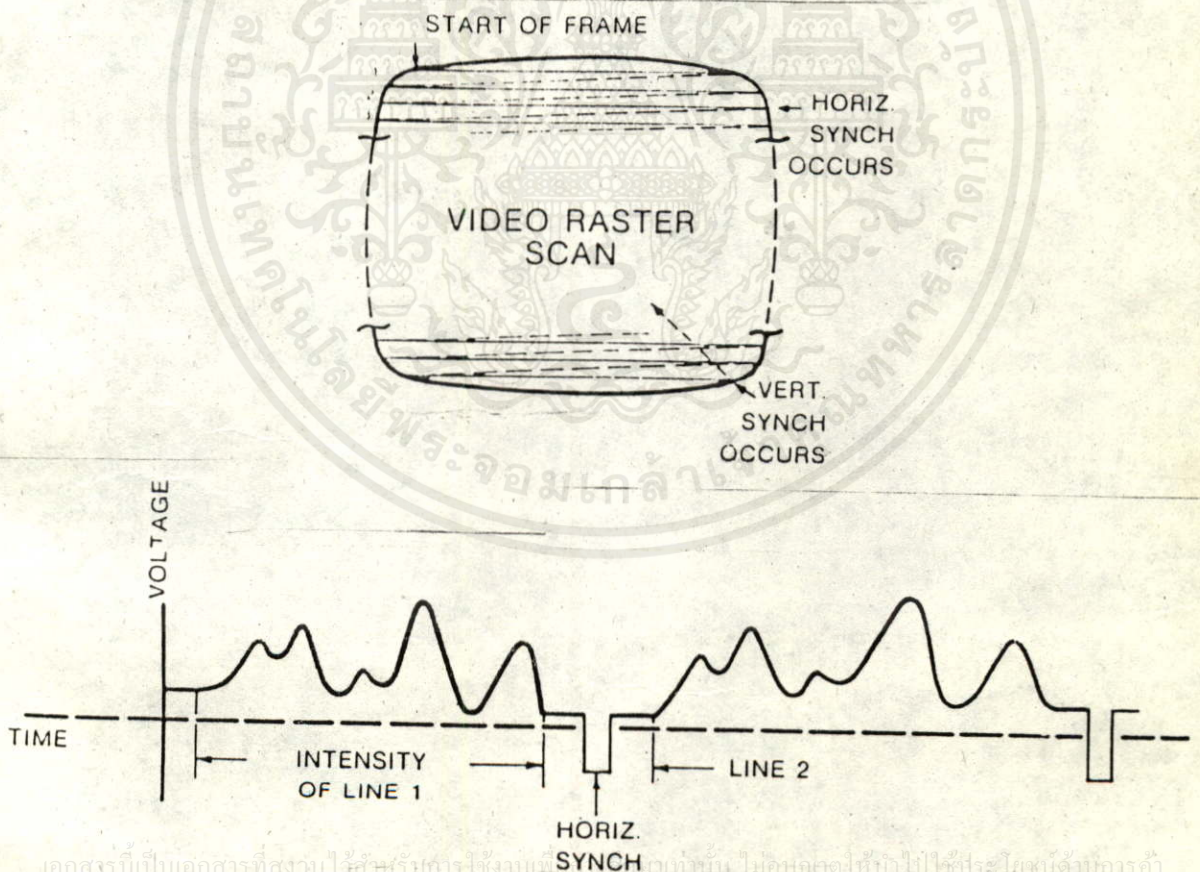
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การรับภาพตัวอักษรจากกล้อง CCD

สัญญาณไฟฟ้าที่ต่อเนื่องนี้คือ ค่าความเข้มของแสงแต่ละจุดบนตัวอักษรที่กล้องสแกน (Scan)

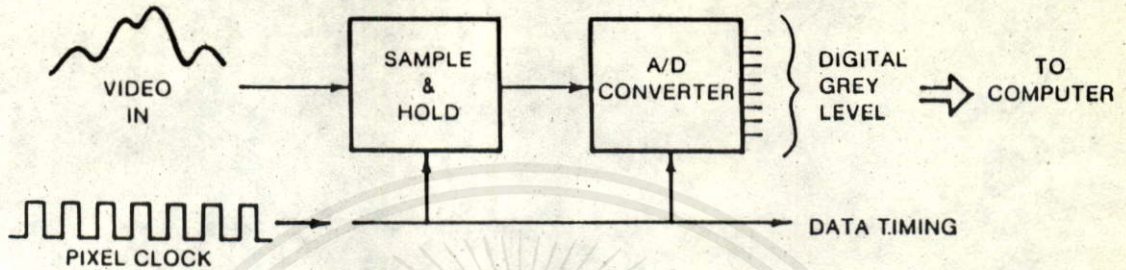
ไปนั่นเอง ลักษณะการสแกนของกล้องแสดงในรูปที่ 2.2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นการก้า
ไม่ว่ากรณีใดๆทั้งนี้ อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 ลักษณะการสแกนของกล้อง CCD

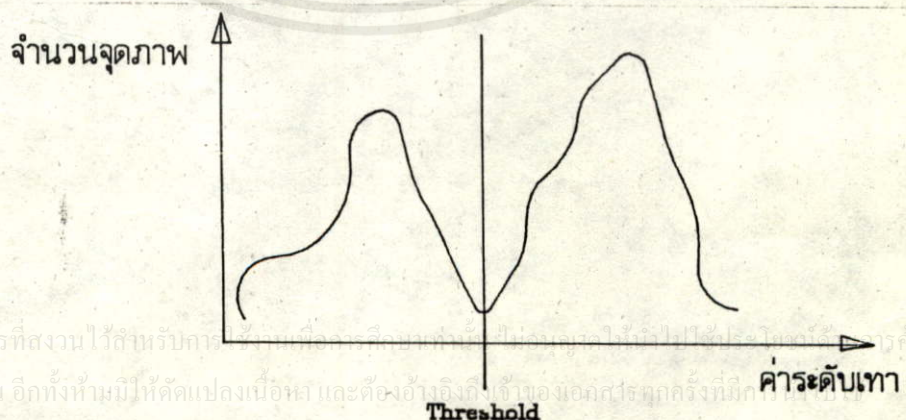
สัญญาณไฟฟ้าที่ต่อเนื่องในรูป 2.2 จะถูกนำมาสุ่ม (Sampling) แล้วทำการเปลี่ยนให้เป็นสัญญาณที่ไม่ต่อเนื่อง (Digital Signal) ด้วยอุปกรณ์เปลี่ยนสัญญาณ A/D Converter ดังรูปที่ 2.3



รูปที่ 2.3 การเปลี่ยนสัญญาณ อนาล็อก เป็น ดิจิตอล

สัญญาณในแต่ละเส้นสแกนก็จะถูกเปลี่ยน ให้เป็นสัญญาณแบบดิจิตอลหลาย ๆ จำนวน ขึ้นอยู่กับความถี่ของสัญญาณนาฬิกา แต่ละข้อมูลดิจิตอลที่ได้จะแทน 1 จุดภาพของการสแกนหรือ 1 Picture element (Pixel) ส่วนค่าของข้อมูลในแต่ละ Pixel นี้เรียกว่าค่าระดับเทา (Gray level)

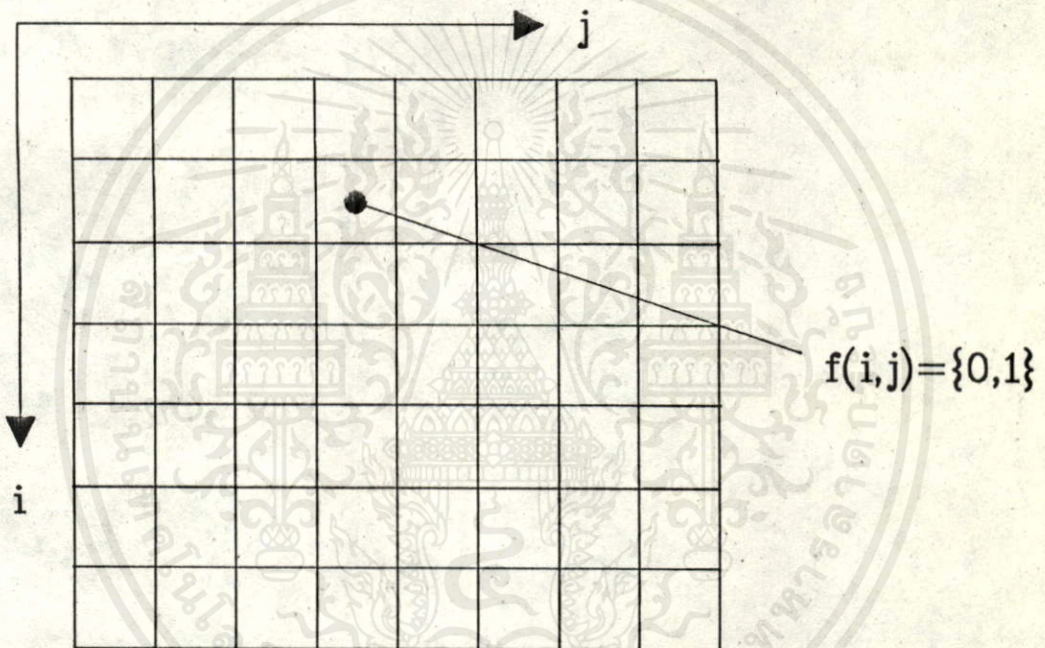
สำหรับรูปแบบที่ไม่ต้องการความละเอียดทางด้านระดับของแสงหรือระดับเทา ก็จะต้องทำการเปลี่ยนรูปแบบระดับเทา (Gray level pattern) ให้เป็นรูปแบบ 2 ระดับ (Binary Pattern) ด้วยการเลือกค่า ชัดเริ่ม (Threshold) ที่เหมาะสม



รูปที่ 2.4 ฮิสโตแกรมสำหรับการทำรูปแบบสองระดับ

ค่าขีดเริ่ม (Threshold) ที่เลือกอาจจะมีมากกว่า 1 ค่าก็ได้ โดยการพิจารณาค่าขีดเริ่มที่เหมาะสมจาก ฮิสโตแกรม (Histogram) ในรูปที่ 2.4 หรืออาจจะทำให้มีการคำนวณ ค่าขีดเริ่ม (Threshold) ขึ้นมาเองโดยอัตโนมัติ [21] ก็ได้

รูปแบบ 2 ระดับที่ได้จะมีค่าความสว่างในแต่ละจุดภาพได้สองค่า เช่น อาจจะใช้ค่า 1 แทนจุดภาพที่สว่าง หรือเป็นสีขาว และใช้ 0 แทนจุดภาพที่มืด หรือเป็นสีดำก็ได้ ดังแสดงในรูปที่ 2.5



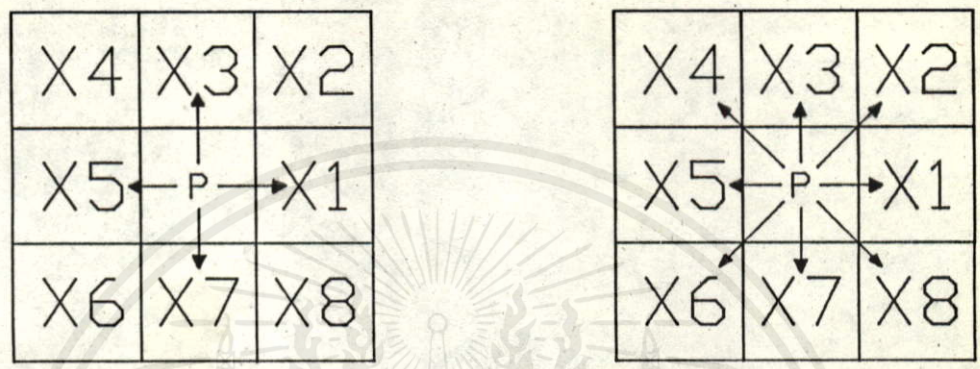
รูปที่ 2.5 ลักษณะของรูปแบบสองระดับ

2.3.2 การคำนวณค่าตัวเลขต่อเนื่อง (Connectivity number)

ค่าตัวเลขต่อเนื่อง [7] เป็นคุณสมบัติทาง โทโพโลยี (Topological property) ของจุดภาพ ที่อยู่ติดกันในรูปแบบสองระดับ ค่าตัวเลขต่อเนื่องนี้จะเป็นตัวแสดงความสัมพันธ์ระหว่างจุดภาพข้างเคียง (Neighbourhood pixel) จุดภาพข้างเคียงบนรูปแบบสองระดับใด ๆ จะแบ่งได้เป็น 2 แบบ คือ

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

กำหนดให้ P เป็นจุดภาพใด ๆ บนรูปแบบสองระดับ จุดภาพข้างเคียงของจุด P ทั้ง 2 แบบ แสดงไว้ในรูปที่ 2.6 (ก) และ (ข) ตามลำดับ



(ก)จุดข้างเคียง 4 จุด

(ข)จุดข้างเคียง 8 จุด

รูปที่ 2.6 ลักษณะของจุดภาพข้างเคียง 2 แบบ

(ก) 4-Neighbourhood pixel

(ข) 8-Neighbourhood pixel

ค่าตัวเลขต่อเนื่องของจุดภาพข้างเคียงทั้ง 2 แบบสามารถหาได้จากสมการ (2.1) และ

สมการ (2.2)

$$NC_4 = \sum_{k=1}^4 (X_k - X_k X_{k+1} X_{k+2}) \tag{2.1}$$

$$NC_8 = \sum_{k=1}^8 (\bar{X}_k - \bar{X}_k \bar{X}_{k+1} \bar{X}_{k+2}) \tag{2.2}$$

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

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

เอกสารนี้เป็นผลที่ได้จากการคำนวณโดยสมการ (2.1) หรือ (2.2) จะได้เป็นเลขจำนวนเต็มบวกที่มีค่าไม่มากกว่าค่าหนึ่ง อีกทั้งค่าที่ได้ขึ้นอยู่กับเงื่อนไขและต้องอ้างอิงถึงค่าของเอกสารทุกครั้งที่มีการนำไปใช้ออยู่ระหว่าง 0 ถึง 4 ดังที่แสดงในรูป 2.7

	1		$NC_4 = 2$
	1		
	1		

			$NC_4 = 0$	1	1	1	$NC_4 = 0$
	1			$NC_8 = 0$	1	1	
				1	1	1	

1		1	$NC_4 = 0$	1		1	$NC_4 = 0$
	1			$NC_8 = 2$		1	
						1	

รูปที่ 2.7 ค่า NC_4 และ NC_8 ของจุดภาพ

ค่าตัวเลขต่อเนื้อที่คำนวณได้ทั้ง 2 แบบ จะบ่งบอกถึงความสัมพันธ์ระหว่างจุดภาพ ออกมา
ดังที่แสดงในตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตารางแสดงความสัมพันธ์ระหว่างจุดภาพ

ค่าตัวเลข ต่อเนื่อง	จุดอิสระ หรือ จุดภายใน	จุดปลาย	จุดต่อเนื่อง				จุดแยก			จุดตัด				
NC ₄	0	1	0	1	2	2	0	3	0	1	2	4	0	
			0	1	2	2	0	3	0	1	2	4	0	
			0	1	2	2	0	3	0	1	2	4	0	
			0	1	2	2	0	3	0	1	2	4	0	
NC ₆	0	1	1	1	2	2	1	2	1	3	3	2	0	4
			1	1	2	2	1	2	1	3	3	2	0	4
			1	1	2	2	1	2	1	3	3	2	0	4
			1	1	2	2	1	2	1	3	3	2	0	4

ค่าตัวเลขต่อเนื่องของ จุดอิสระหรือจุดภายในคือ 0, จุดปลายคือ 1, จุดต่อเนื่องคือ 2, จุดแยกคือ 3, จุดตัดคือ 4

ตัวอย่างการคำนวณ

	1	
1	P	1
	1	

(ก)

1		1
	P	
1		1

(ข)

รูปที่ 2.8 ตัวอย่างการคำนวณค่า NC_4 และ NC_8 จากรูปที่ 2.8 (ก) จะคำนวณค่า NC_4 และ NC_8 ได้ดังนี้

$$\begin{aligned} NC_4 &= X_1 + X_3 + X_5 + X_7 - X_1 * X_2 * X_3 - X_3 * X_4 * X_5 - X_5 * X_6 * X_7 - X_7 * X_8 * X_1 \\ &= 1+1+1+1 - 1*0*1 - 1*0*1 - 1*0*1 - 1*0*1 \\ &= 4 \end{aligned}$$

$$\begin{aligned} NC_8 &= \bar{X}_1 + \bar{X}_3 + \bar{X}_5 + \bar{X}_7 - \bar{X}_1 * \bar{X}_2 * \bar{X}_3 - \bar{X}_3 * \bar{X}_4 * \bar{X}_5 - \bar{X}_5 * \bar{X}_6 * \bar{X}_7 - \bar{X}_7 * \bar{X}_8 * \bar{X}_1 \\ &= 0+0+0+0 - 0*1*0 - 0*1*0 - 0*1*0 - 0*1*0 \\ &= 0 \end{aligned}$$

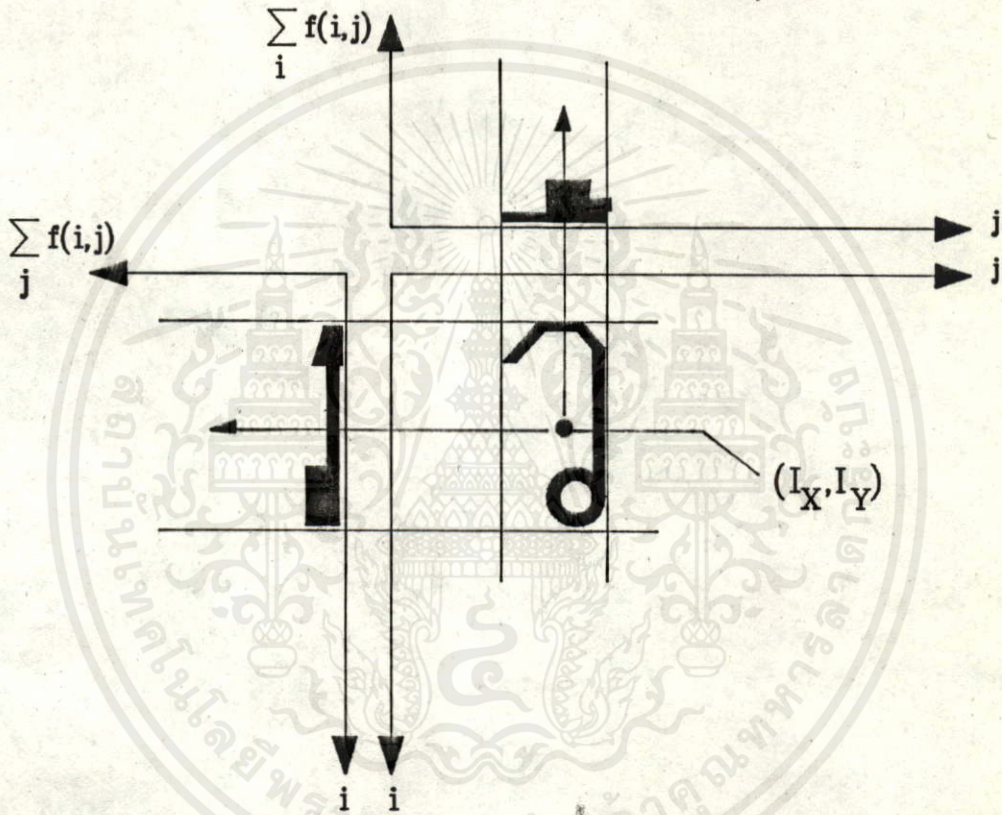
จากรูปที่ 2.8 (ข) จะคำนวณค่า NC_4 และ NC_8 ได้ดังนี้

$$\begin{aligned} NC_4 &= X_1 + X_3 + X_5 + X_7 - X_1 * X_2 * X_3 - X_3 * X_4 * X_5 - X_5 * X_6 * X_7 - X_7 * X_8 * X_1 \\ &= 0+0+0+0 - 0*1*0 - 0*1*0 - 0*1*0 - 0*1*0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} NC_8 &= \bar{X}_1 + \bar{X}_3 + \bar{X}_5 + \bar{X}_7 - \bar{X}_1 * \bar{X}_2 * \bar{X}_3 - \bar{X}_3 * \bar{X}_4 * \bar{X}_5 - \bar{X}_5 * \bar{X}_6 * \bar{X}_7 - \bar{X}_7 * \bar{X}_8 * \bar{X}_1 \\ &= 1+1+1+1 - 1*0*1 - 1*0*1 - 1*0*1 - 1*0*1 \\ &= 4 \end{aligned}$$

2.3.3 การหาจุดศูนย์กลางถ่วง (Center of Gravity)

รูปแบบสองระดับที่มีจุดภาพ ที่มีค่า 1 เป็นเนื้อของภาพ และมีค่า 0 เป็นฉาก (Background) ของภาพ จะสามารถหาตำแหน่งที่เป็นจุดศูนย์กลางของภาพได้ จุดนี้เรียกว่าจุดศูนย์กลางถ่วง (Center of Gravity) [18]



รูปที่ 2.9 จุดศูนย์กลางถ่วงของภาพสองระดับ

จากรูปที่ 2.9 ส่วนที่เป็นสีทึบ แสดงถึงส่วนที่เป็นเนื้อของภาพ ค่า i, j เป็นจำนวนจุดภาพในแนวแถว (Row) และแนวสดมภ์ (Column) ตามลำดับ

เอกสารนี้เป็นตำแหน่งของแถว และสดมภ์ ที่เป็นตำแหน่งของจุดศูนย์กลางถ่วงจะหาได้จากสมการต่อไปนี้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

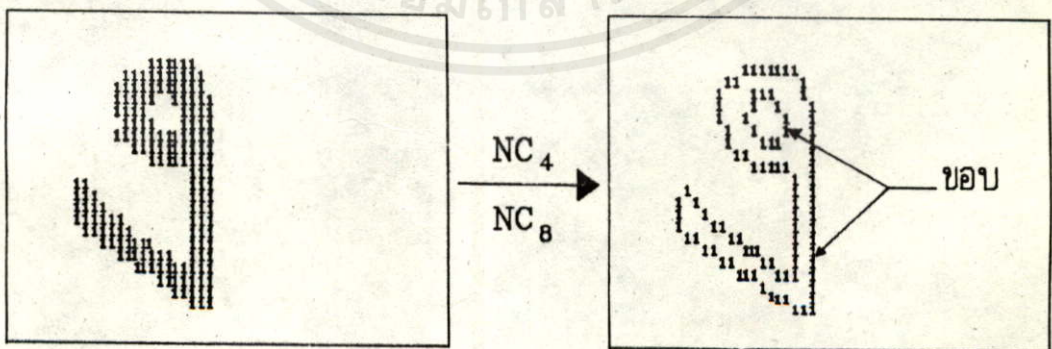
เมื่อ $f(i,j)$ เป็นฟังก์ชันแสดงจุดภาพบนรูปแบบ 2 ระดับใด ๆ

2.3.4 การตรวจสอบขอบของภาพ (Edge detection)

ขอบเขตของภาพในรูปแบบ กั้นฉาก จะแบ่งแยกจากกันโดยขอบของภาพ (Edge) การหาขอบเขตของภาพบนรูปแบบสองระดับใด ๆ มีวิธีการทำได้หลาย ๆ วิธี เช่นการใช้คุณสมบัติทางโทโพโลยีของจุดภาพ^[14] การติดตามขอบเขตของภาพ (Contour following)^{[11],[12]} หรือวิธีการใช้ตารางหน้าต่าง^[22] สำหรับในวิทยานิพนธ์ฉบับนี้นั้นจะกล่าวถึงเพียง 2 วิธีแรกเท่านั้น คือ

1. การหาขอบของภาพโดยการใช้ค่าตัวเลขต่อเนื่อง NC_4 และ NC_8 ที่คำนวณได้จากสมการ 2.1 และ 2.2 ในหัวข้อ 2.3.2

จุดภาพใด ๆ ก็ตาม ที่มีค่า NC_4 และ NC_8 เป็นไปตามลักษณะของจุดปลาย (End point) ซึ่งก็หมายถึงว่าเป็นจุดภาพที่เป็นขอบดังแสดงในรูปที่ 2.10



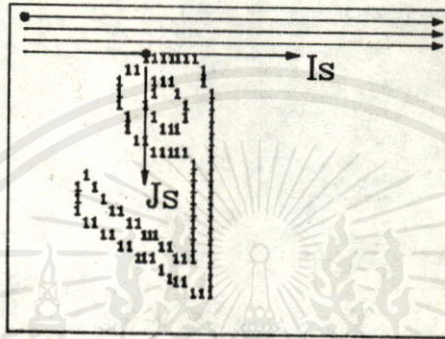
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.10 ค่าตัวเลขต่อเนื่องที่เป็นขอบของภาพ

ดังนั้น จึงสามารถใช้ค่าตัวเลขต่อเนื่องกันทำได้ นอกขอบเขตของภาพได้ทั้งขอบในและขอบนอกของภาพ แต่จะไม่สามารถบอกจำนวนจุดภาพที่เป็นขอบได้

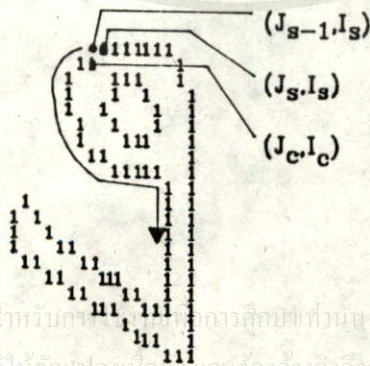
2. การหาขอบของภาพ โดยการติดตามขอบเขตของภาพ (Contour following)

การหาขอบภาพโดยวิธีนี้ จะต้องหา จุดเริ่มต้นของขอบก่อน



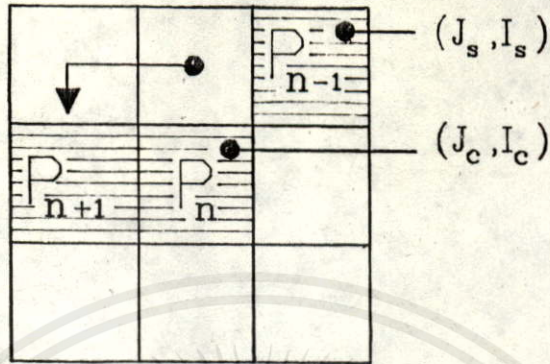
รูปที่ 2.11 การสแกนหาจุดภาพที่เป็นจุดเริ่มต้น

จากรูปแบบสองระดับในรูปที่ 2.11 จะทำการสแกนไปบนรูปแบบจากบนลงล่าง (Forward scanning) จนกระทั่งพบจุดที่เป็นภาพ ก็จะถือว่าจุดนั้นเป็นจุดเริ่มต้นของขอบภาพ (J_s, I_s) และจะกำหนดให้จุดก่อนหน้าที่จะถึงจุดเริ่มต้นของขอบนี้ (J_{s-1}, I_{s-1}) เป็นตำแหน่งเริ่มต้นสำหรับการวนในทิศทางวนเข็มนาฬิกา เพื่อติดตามหาขอบภาพจุดต่อไป (J_c, I_c) ดังรูป 2.12



รูปที่ 2.12 จุดเริ่มต้นของการติดตามหาขอบภาพ

จากนั้นวาง Window ขนาด 3 X 3 ที่ขอบภาพ (J_c, I_c) แล้วจะวนไปในทิศทางทวนเข็มนาฬิกา เพื่อติดตามหาขอบภาพจุดต่อไป ดังรูปที่ 2.13



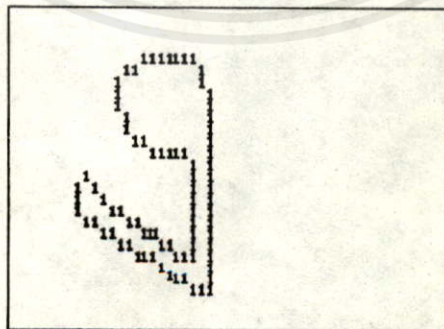
รูปที่ 2.13 การติดตามหาจุดภาพที่เป็นขอบ

เมื่อ P_{n-1} เป็นตำแหน่งของขอบภาพที่ผ่านมา

P_n เป็นตำแหน่งของขอบภาพที่กำลังพิจารณา

และ P_{n+1} เป็นตำแหน่งของขอบภาพต่อไป

จากรูปที่ 2.13 จะเห็นว่าจุดเริ่มต้นในการติดตามขอบจุดต่อไป (P_{n+1}) จะเป็นตำแหน่งที่ถัดจากจุดขอบภาพที่ผ่านมา (P_{n-1}) ในทิศทางทวนเข็มนาฬิกาเสมอ ด้วยวิธีการดังกล่าวนี้ เมื่อติดตามขอบเขตของภาพไปจนกระทั่งพบตำแหน่งเริ่มต้นของขอบ (J_c, I_c) ก็จะได้ขอบของภาพออกมา ดังรูปที่ 2.14



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครู ไร้งานเพื่อการศึกษาค้นคว้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่มีวรรณคดีฯ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.14 ขอบของภาพที่ได้จากการติดตามขอบเขต



ขอบที่ได้นี้ จะเป็นเพียงขอบนอกเท่านั้น แต่สามารถที่จะบอกจำนวนจุดภาพที่เป็นขอบได้ ด้วย
เหตุนี้ ในวิทยานิลเนศฉบับนี้ จึงได้นำเอาวิธีการอันนี้ไปประยุกต์ใช้ในการตรวจสอบหาส่วนที่เป็นหัวของตัว
อักษรในภาษาไทย ดังจะได้กล่าวถึงรายละเอียดในบทที่ 4 ต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12440

บทที่ 3

การเตรียมข้อมูลเพื่อการจำแนกกลุ่มของตัวอักษร

ปัญหาที่เกิดขึ้นเสมอในระบบการจำแนกรูปแบบของตัวอักษรภาษาไทย คือ รูปแบบตัวอักษรที่ใช้เป็น อินพุท (Input pattern) ของระบบมีสัญญาณรบกวน (Noise) ปะปนเข้ามาด้วย สัญญาณรบกวนนี้จะทำให้เกิดการผิดพลาดขึ้นในขั้นตอนการจำแนกตัวอักษร ดังนั้นจึงมีความจำเป็นที่จะต้องมีการปรับปรุงคุณภาพของรูปแบบตัวอักษร ก่อนที่จะนำไปจำแนกกลุ่มต่อไป จึงเรียกขบวนการในขั้นตอนนี้ว่า ระบบการจัดการล่วงหน้า (Pre-processing)

ระบบการจัดการล่วงหน้าดังกล่าวถึงในบทนี้ประกอบไปด้วยเทคนิคและวิธีการต่าง ๆ หลายขั้นตอนดังจะ ได้กล่าวถึงในหัวข้อต่อ ๆ ไป

3.1 การกำจัดสัญญาณรบกวน

สัญญาณรบกวนที่ปรากฏในรูปแบบของตัวอักษรที่พบคือสัญญาณรบกวนที่มีลักษณะเป็น จุดอิสระ (Isolate point) และรูภายใน (Isolate Hold) ^[15] การกำจัดสัญญาณรบกวนทั้ง 2 ลักษณะนี้จะใช้ตารางหน้าต่าง (Window) ขนาด 3 X 3 pixel ในรูปที่ 3.1 สแกนไปบนรูปแบบสองระดับที่มีสัญญาณรบกวน

X_3	X_2	X_1
X_4	X	X_0
X_5	X_6	X_7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 ตารางหน้าต่างสำหรับการกำจัดสัญญาณรบกวน

เมื่อทำสหสัมพันธ์ (Correlation) ของตารางหน้าต่าง ในรูปที่ 3.1 กับรูปแบบของตัวอักษร และกระทำตาม Algorithm A:

Algorithm A:

สแกนไปบนรูปแบบ ทุก ๆ จุดภาพ

{

ถ้า จุดภาพข้างเคียงของจุดภาพ X ทั้ง 8 จุดภาพมีค่าเป็น 0

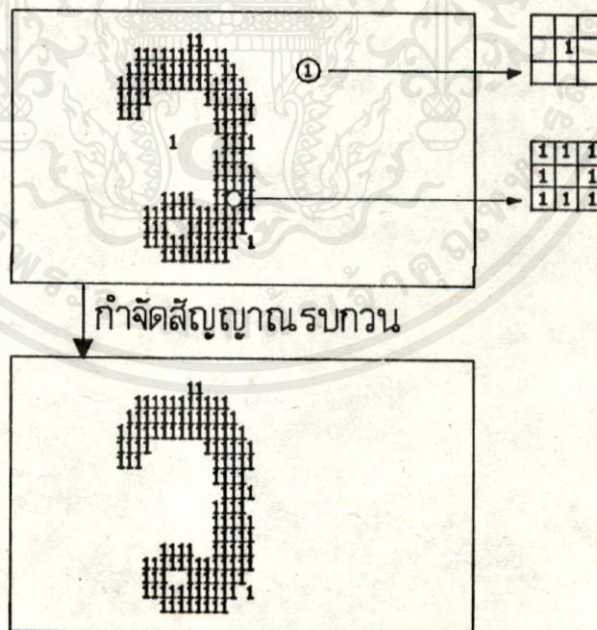
ทำให้จุดภาพ X มีค่าเป็น 0 ตาม

ถ้า จุดภาพข้างเคียงของจุดภาพ X ทั้ง 8 จุดภาพมีค่าเป็น 1

ทำให้จุดภาพ X มีค่าเป็น 1 ตาม

}

ผลที่ได้จากการทำ Algorithm A: จะทำให้สัญญาณรบกวนในรูปแบบถูกกำจัดออกไปดัง
แสดงในรูปที่ 3.2

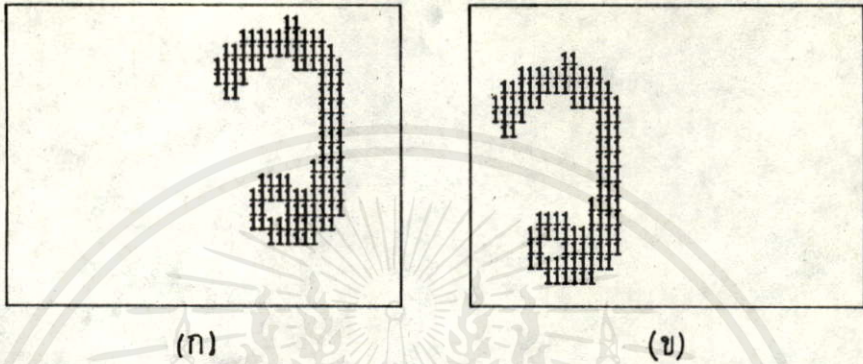


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 การกำจัดสัญญาณรบกวนบนรูปแบบ

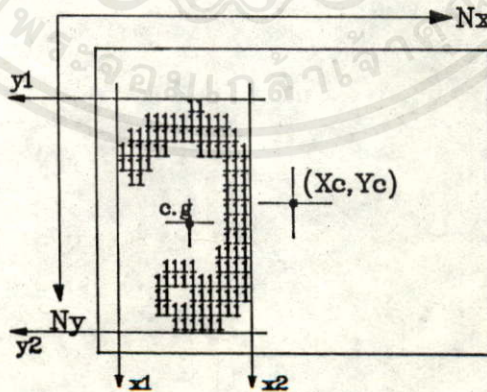
3.2 การหาตำแหน่งและขนาดมาตรฐาน

รูปแบบของตัวอักษรที่กำหนดลักษณะรอบกวนออกไปแล้ว จะวางอยู่ที่ตำแหน่งต่าง ๆ กันภายในรูปแบบสองระดับขนาด 64×64 จุดภาพ ดังที่แสดงไว้ในรูปที่ 3.3



รูปที่ 3.3 ตำแหน่งต่างๆของตัวอักษรภายในรูปแบบ

เพื่อความเป็นมาตรฐานของตำแหน่ง จึงต้องทำการย้ายตัวอักษรดังที่แสดงในรูปที่ 3.3 ให้อยู่ตรง ตำแหน่งที่มาตรฐาน [15] ในที่นี้ ได้ใช้จุดกึ่งกลางของรูปแบบ (X_c, Y_c) เป็นตำแหน่งมาตรฐาน

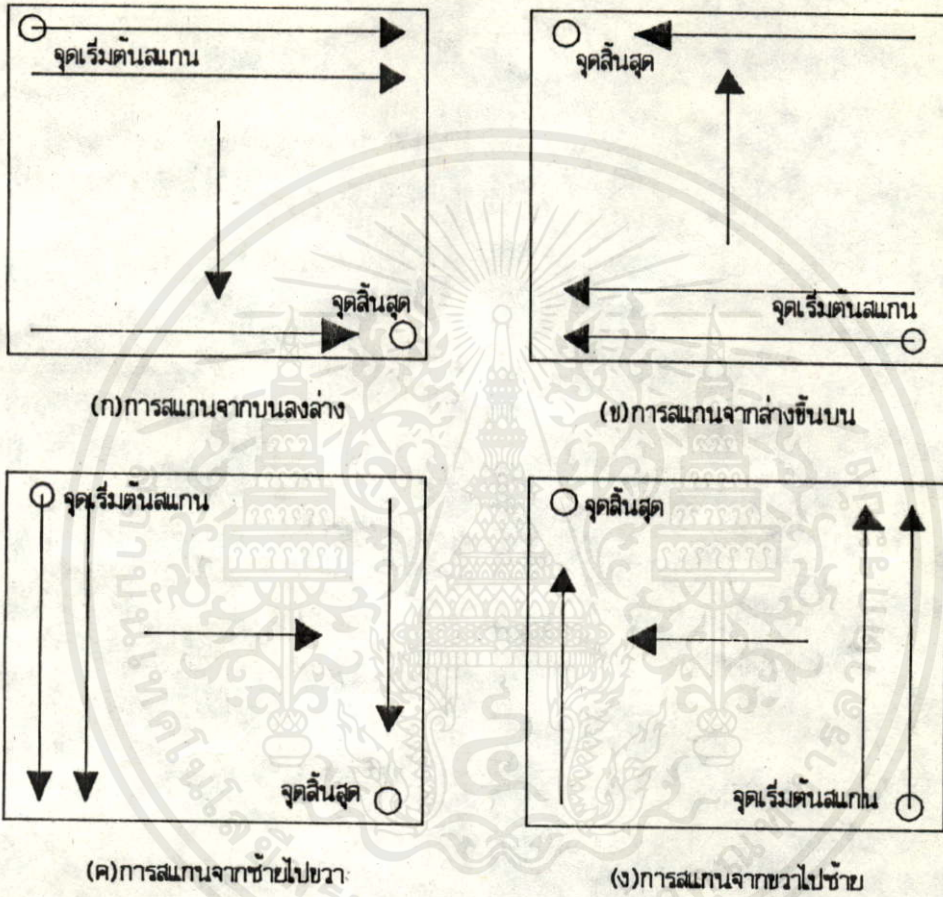


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามคัดลอกไปเผยแพร่โดยไม่ได้รับอนุญาตจากสำนักพิมพ์

รูปที่ 3.4 การย้ายภาพตัวอักษรไปยังตำแหน่งมาตรฐาน

$$\text{เมื่อ } X_c = N_x / 2 \text{ และ } Y_c = N_y / 2$$

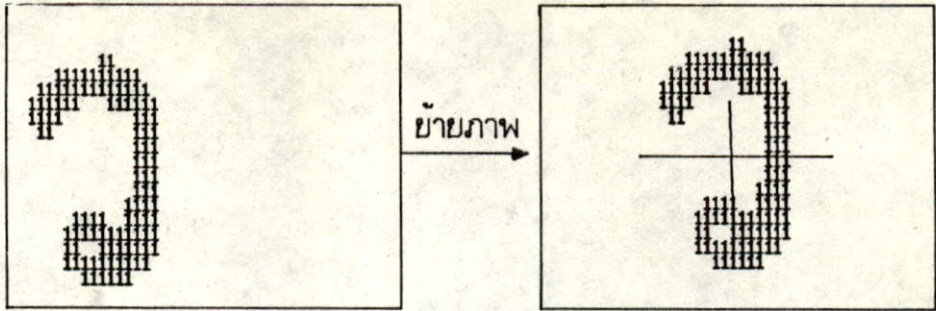
การย้ายรูปแบบจะทำได้โดยการสแกนหากรอบ (border) ของตัวอักษรทั้ง 4 ด้านคือ X1, X2, Y1 และ Y2 ซึ่งก็คือกรอบบน ล่าง ซ้าย ขวา ของตัวอักษร ดังแสดงในรูปที่ 3.4 ทิศของการสแกนทั้ง 4 ด้านแสดงในรูปที่ 3.5



รูปที่ 3.5 ทิศทางของการสแกนทั้ง 4 แบบ

- (ก) การสแกนจากบนลงล่าง (ข) การสแกนจากล่างขึ้นบน
(ค) การสแกนจากซ้ายไปขวา (ง) การสแกนจากขวาไปซ้าย

ตัวอักษรที่อยู่ในกรอบนี้จะนำมาหาตำแหน่งของจุดศูนย์กลาง (c.g) โดยใช้สมการที่ (2.3) และ (2.4) ในบทที่ 2 แล้วทำการย้ายภาพตัวอักษรจากตำแหน่งจุดศูนย์กลางนี้ไปยังตำแหน่งมาตรฐานดังแสดงในรูปที่ 3.6



รูปที่ 3.6 ภาพตัวอักษรที่ย้ายมายังตำแหน่งมาตรฐาน

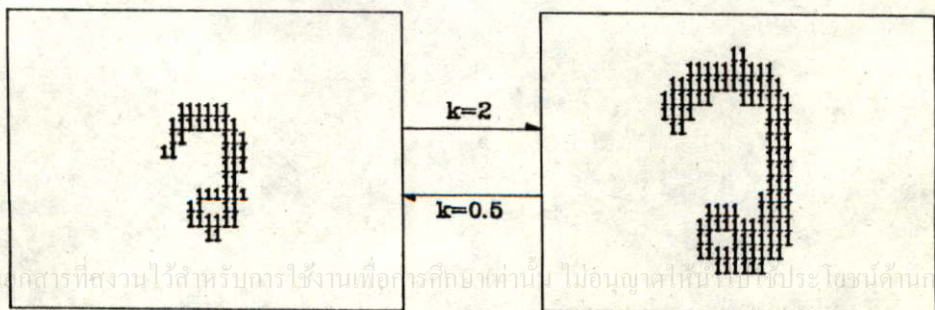
รูปแบบของตัวอักษรที่อยู่ในตำแหน่งมาตรฐานแล้วอาจจะมีความแตกต่างกันในการเก็บข้อมูลภาพแต่ละครั้ง ดังนั้นจะต้องทำรูปแบบของตัวอักษรให้อยู่ในรูปแบบที่มีขนาดมาตรฐานเดียวกันก่อน [15]

การทำรูปแบบตัวอักษรให้มีขนาดมาตรฐานสามารถทำได้โดยการย่อหรือขยายรูปแบบตัวอักษรในอัตราส่วนที่คงที่ค่าหนึ่ง อัตราส่วนนี้หาได้จากสมการที่ (3.1)

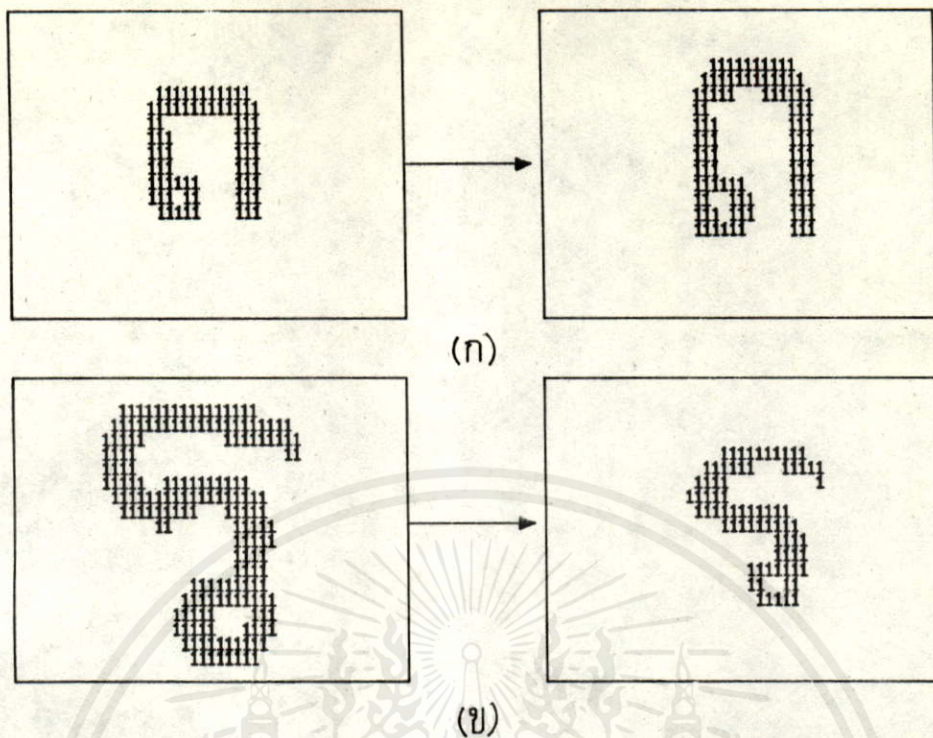
$$k = \min \{Lx-2 / x2-x1-1, Ly-2 / y2-y1-1\} \quad (3.1)$$

ในสมการ (3.1) ตัวแปร Lx และ Ly คือค่าเริ่มต้น (Initial Value) ที่กำหนดให้ในการทดลอง กำหนดให้ $Lx = 40$ และ $Ly = 32$ (สำหรับรูปแบบขนาด 64×64 จุดภาพ) ค่า $x1, x2, y1$ และ $y2$ คือตำแหน่งของแนวและสดมภ์ที่เป็นกรอบของตัวอักษร ซึ่งได้จากการสแกนหากรอบดังแสดงในรูปที่ 3.4

ค่า k คือค่าอัตราส่วนของการย่อหรือขยายรูปแบบดังแสดงในรูปที่ 3.7



รูปที่ 3.7 การหาขนาดมาตรฐาน



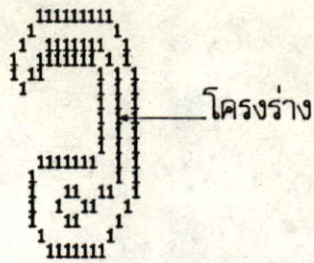
รูปที่ 3.8 ตัวอักษรที่มีขนาดมาตรฐาน

ตัวอย่างของตัวอักษรบางตัวที่อยู่ในรูปแบบที่มีขนาดมาตรฐานแสดงในรูปที่ 3.8

3.3 การทำตัวอักษรให้บาง (Thinning)

รูปแบบสองระดับที่ได้มาจากอุปกรณ์รับรู้อทางแสง เช่น OCR หรือ Scanner หรือแม้กระทั่งกล้องที่วีก็ตามจะมีภาพของตัวอักษรในรูปแบบที่มีขนาดความกว้างมากกว่า 1 จุดภาพเสมอ แต่ในระบบการจดจำที่ใช้คุณสมบัติทางโทโปโลยีของรูปแบบเป็นตัวจำแนกกลุ่มนั้น จะต้องการเฉพาะส่วนที่เป็นโครงร่าง (Skeleton) ของตัวอักษรเท่านั้น โครงร่างของตัวอักษรก็คือภาพของตัวอักษรที่มีความกว้างเพียง 1 จุดภาพและอยู่ในตำแหน่งที่สมมาตรบนเนื้อของตัวอักษร ดังแสดงในรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 โครงร่างของตัวอักษร

ดังนั้นจึงต้องทำตัวอักษรในรูปแบบให้บางลง ด้วยวิธีการทำให้บาง (Thinning) ^{[8], [10]} วิธีการทำให้บางนี้มีการวิจัยและพัฒนากันหลายวิธีแต่วิทยานิพนธ์นี้ได้รับปรุงวิธีการของ Naccache และ Shinghal ^[10] เพื่อให้เหมาะสมกับลักษณะของส่วนโค้งในตัวอักษรภาษาไทย

n_3	n_2	n_1
n_4	P	n_0
n_5	n_6	n_7

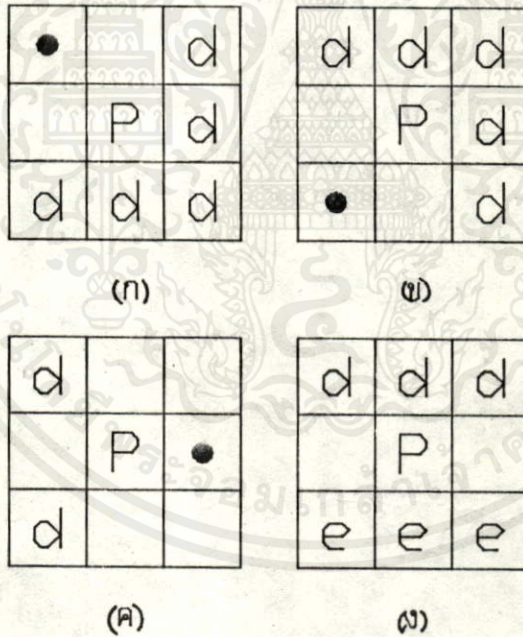
รูปที่ 3.10 ตารางหน้าตาสำหรับการทำให้บาง

ตารางหน้าตาต่างในรูปที่ 3.10 ใช้นำมาพิจารณาสวนของภาพตัวอักษรบนรูปแบบที่ต้องการทำให้บางโดยที่จุดข้างเคียงของจุดภาพ P ซึ่งก็คือจุดภาพ $n_0, n_1, n_2, \dots, n_7$ จะเป็นตัวกำหนดลักษณะขอบ (Edge) ของจุดภาพ P ใน 4 ลักษณะคือ จะต้องอ้างอิงถึงตัวเองเอกสารทุกครั้งที่มีการนำไปใช้

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_8 เป็นจุดสว่าง ($n_8 = 0$)

ในขั้นตอนการทำให้บางถือว่า ขอบของภาพไม่ใช่โครงร่างของตัวอักษร ดังนั้น เมื่อพบขอบก็จะกำหนดค่าของจุดภาพนั้นให้เป็นค่าค่าหนึ่ง ที่ไม่เป็น 1 (flagging) แต่ถ้าพบจุดภาพที่มีความกว้างเพียง 1 จุดภาพก็จะกำหนดค่าของจุดภาพนั้นให้เป็นค่า 1 (Unflagging)

การทดสอบจุดภาพที่เป็นขอบ กระทำโดยการเปรียบเทียบตารางหน้าต่างในรูป 3.10 ที่ สแกนจากบนลงล่างบนรูปแบบของตัวอักษร กับ ตารางหน้าต่างในรูปที่ 3.11



รูปที่ 3.11 ตารางหน้าต่างสำหรับการตรวจสอบหาขอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม กรุณาแจ้งที่มาของเอกสารฉบับนี้

จากรูปที่ 3.11 P และ • เป็นจุดมืด (มีค่าเป็น 1) ส่วน d และ e เป็นค่าที่ไม่สนใจ

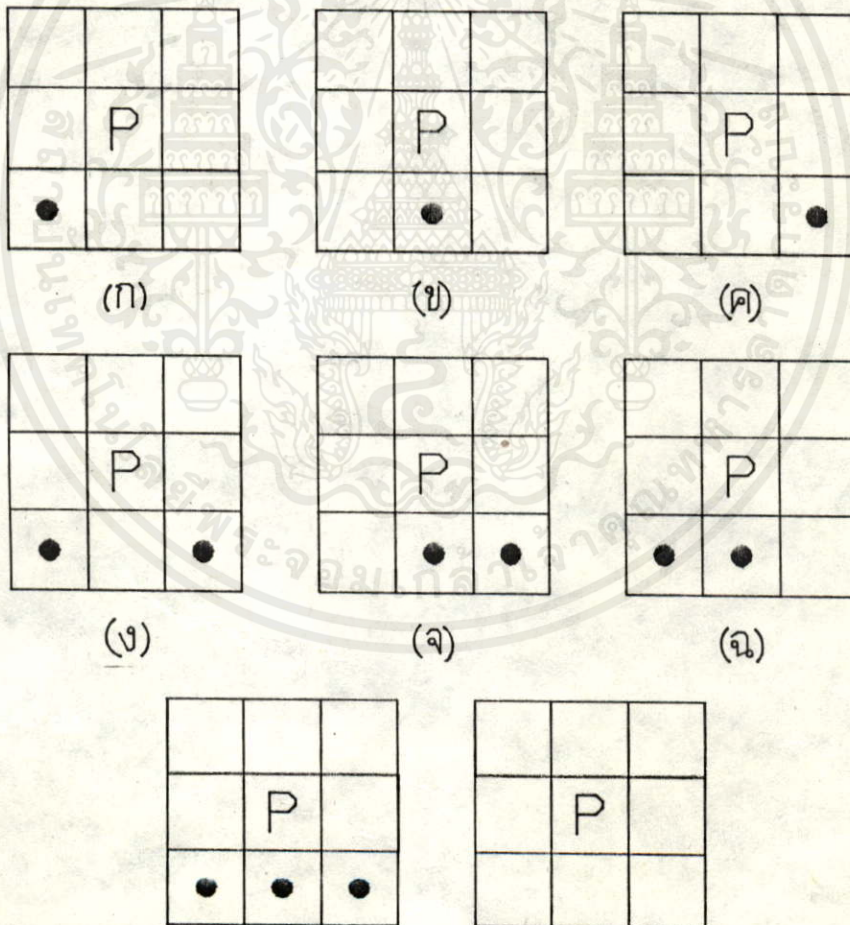
(don't care)

เมื่อนิยามารูปที่ 3.11 (ก), (ข), (ค) จะแบ่งการพิจารณาเป็น 2 ลักษณะคือ

1. ถ้า d ทุกตัวเป็นจุดสว่าง (มีค่าเป็น 0) จะได้ว่า P คือจุดภาพที่เป็นจุดปลาย (end point) ของโครงร่าง
2. ถ้า d อย่างน้อย 1 จุดเป็นจุดมืด (มีค่าเป็น 1) จะได้ว่า P คือจุดภาพที่เป็นโครงร่างของตัวอักษร

จุด P ใด ๆ ที่มีคุณสมบัติตาม 2 ลักษณะดังกล่าวจะถูกกำหนดค่าให้เป็น 1 (Unflagged)

ตารางหน้าตาต่างในรูปที่ 3.11 (ง) จะมีการพิจารณาที่ซับซ้อนขึ้นโดยแยกการพิจารณาออกมา
 ดังแสดงโดยตารางหน้าตาต่างในรูปที่ 3.12



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับค (ช) งานเพื่อการศึกษาเท่านั้น (ช) มาตราให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะตีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 ตารางหน้าตาต่างของจุดภาพโครงร่าง

จากรูป 3.11 (ง) ถ้า d และ e อย่างน้อย 1 จุดเป็นจุดมืดจะกำหนดให้จุด P เป็นจุด
โครงร่าง ในกรณีนอกเหนือจากนี้จะพิจารณาตามตารางหน้าต่างในรูปที่ 3.12

จากรูป 3.12 ตารางหน้าต่าง (ก), (ข), (ค) จะกำหนดให้จุด P เป็นจุดปลายของโครง
ร่าง ตารางหน้าต่าง (ง) กำหนดให้ P เป็นจุดโครงร่าง ตารางหน้าต่าง (จ) และ (ฉ) กำหนดให้
P เป็นจุดที่ทำให้เกิดความกว้างขนาด 2 จุดภายในแนวเอียง ตารางหน้าต่าง (ช) กำหนดให้จุด P
เป็นกิ่งสั้น (protrusion) ของโครงร่าง ส่วนตารางหน้าต่าง (ซ) กำหนดให้ จุด P เป็นจุดอิสระที่
เกิดขึ้น

เพื่อลดความยุ่งยากในการพิจารณาเงื่อนไขของขอบและโครงร่างลง จึงได้นำเอาเงื่อนไข
ทั้งหมดของขอบมา เขียนเป็นนิพจน์ทางตรรก ทำให้ง่ายต่อการเขียนโปรแกรมยิ่งขึ้น ดังแสดงในสมการ
ที่ (3.1), (3.2), (3.3) และ (3.4) ตามลำดับ

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

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

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

$$\text{ขอบบน: } B2 = N6 * (N0 + N4 + N5 + N7) * (N0 + \bar{N1}) * (\bar{N3} + N4) \quad (3.4)$$

เมื่อ * คือ operator AND

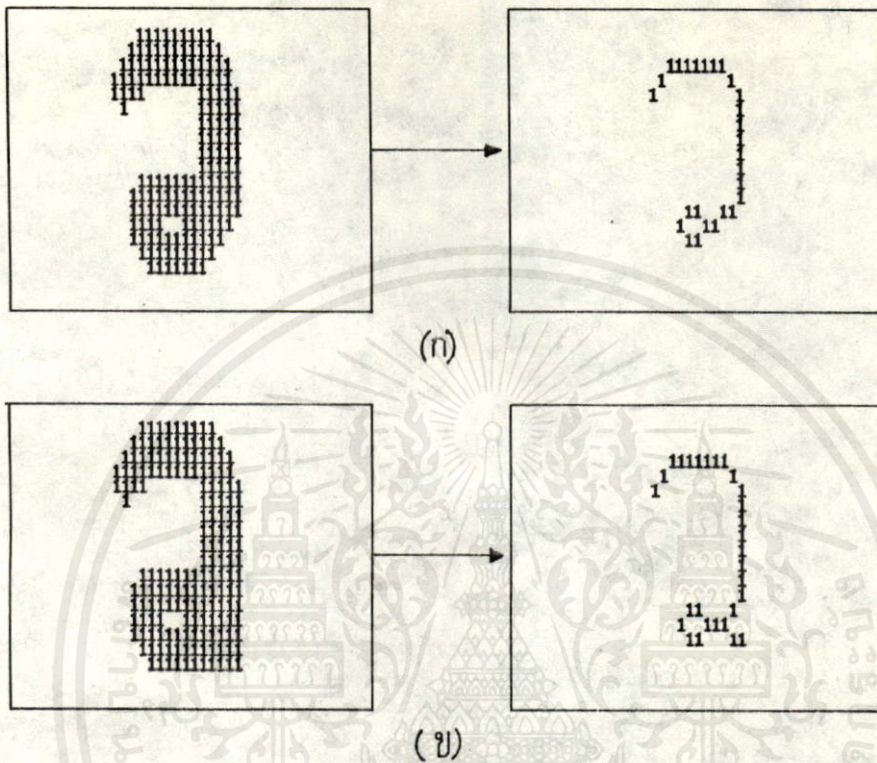
+ คือ operator OR

- คือ operator NOT

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

โดยการสแกนไปบนรูปแบบสองระดับที่ต้องการทำให้บาง ทำการสแกนใน 2 ทิศทาง คือ
สแกนจากบนลงล่าง และ สแกนจากซ้ายไปขวา (รูปที่ 3.5 (ก) และ (ค)) ในทิศทางแรกจะพิจารณา
ขอบ B4 และ B0 และในทิศทางที่สองจะพิจารณาขอบ B2 และ B6 ตามลำดับ

ไม่ว่ากรณีใดด้วยการกระทำ แบบวนรอบ (Iteration) จนกระทั่งเหลือแต่จุดภาพที่เป็นโครงร่างก็จะ
ได้ตัวอักษรที่บางออกมาดังแสดงในรูปที่ 3.13



รูปที่ 3.13 โครงร่างของตัวอักษร

(ก) โครงร่างที่สมบูรณ์ (ข) โครงร่างที่มีส่วนเกิน

ขั้นตอนการทำตัวอักษรให้บางทั้งหมดแสดงไว้ใน Algorithm B:

Algorithm B:

while (new edge points unflagged)

{

เอกสารนี้ : forward scanning ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ออกแบบโดยทีมวิจัยที่ล้มเหลวและต้องอึ้งอึ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

if B4 or B0

(flagged)

```

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

```

}

โครงร่างของตัวอักษรที่ได้จากการทำให้บางจะเก็บ เป็นรูปแบบที่จะนำไปหาค่าคุณสมบัติทางโทโพโลยีต่อไป

ข้อดีของการทำให้บางก็คือจะลดจำนวนข้อมูลที่เป็นภาพตัวอักษรลงไปเป็นจำนวนมาก ทำให้ลดเวลาของการประมวลผลในขั้นตอนต่อไปได้ แต่ทั้งนี้การทำให้บางก็จะใช้เวลาในการประมวลผลมากถ้ารูปแบบสองระดับที่เข้ามามีความหนาแน่นมาก ๆ

3.4 การกำจัดส่วนเกิน (Elimination of Redundancy)

โครงร่างของภาพตัวอักษรที่ได้จากการทำให้บางแล้ว ส่วนมากจะมีส่วนเกินที่เป็นกิ่ง (Redundant bits) ปรากฏออกมาด้วยเสมอ ดังจะเห็นได้จากรูปที่ 3.13 (ข) ส่วนเกินนี้จะทำให้เกิดความผิดพลาดขึ้นต่อระบบการจำแนกกลุ่มของตัวอักษรในภายหลัง ดังนั้นจึงต้องกำจัดส่วนเกินนี้ออกไปจากรูปแบบ

วิธีการกำจัดส่วนเกิน ^[23] จะต้องมีการคำนวณค่าตัวเลขต่อเนื่อง NC_g ในสมการ (2.2) และพิจารณาค่าตัวเลขต่อเนื่องจากตารางที่ 1

กำหนดให้

$$L = \{ l_{i,j} \mid i=1,2,\dots,m ; j=1,2,\dots,n \}$$

ไม่จำกัดว่า L เป็นรูปแบบสองระดับขนาด $m \times n$ ที่เก็บโครงร่างของตัวอักษรที่มีส่วนเกินปรากฏอยู่ (ในที่นี้ $m = 64$ และ $n = 64$)

กำหนดให้

$$M = \{ m_{i,j} \mid i=1,2,\dots,m ; j=1,2,\dots,n \}$$

$$N = \{ n_{i,j} \mid i=1,2,\dots,m ; j=1,2,\dots,n \}$$

M และ N เป็น accumulated pattern ที่มีค่าจุดภาพเริ่มต้นทุก ๆ จุดเป็น 0 และกำหนดให้ค่าเริ่มต้นของจำนวนจุดภาพที่เป็นส่วนเกินคือ จำนวนเต็มบวก K (ในที่นี้ให้ $K = 5$)

X_3	X_2	X_1
X_4	X	X_0
X_5	X_6	X_7

รูปที่ 3.14 ตารางหน้าตาต่างสำหรับการกำจัดส่วนเกิน

จากการใช้ตารางหน้าตาต่างในรูปที่ 3.14 สแกนไปบนรูปแบบสองระดับ L ในสอง ทิศทางคือ สแกนจากบนลงล่าง และ สแกนจากซ้ายไปขวา (ดังแสดงในรูป 3.5 (ก) และ (ค)) และพิจารณาการกระทำเป็น 4 ขั้นตอนคือ

ขั้นตอนที่ 1

กระทำแบบขนาน (Parallel operation) ต่อจุดภาพที่มีค่า 1 ของรูปแบบ L ในทิศทาง สแกนจากบนลงล่าง จำนวน K ครั้ง ดังนี้

if ($NC_{e_1} = 1$)

{ $l_{i,j} = 0 ; m_{i,j} = 1$ }

else

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หาก $if (NC_{e_1} = 2 ; ; NC_{e_1} = 3)$ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

{ $m_{i,j} = 2$ }

ขั้นตอนที่ 2

กระทำแบบอนุกรม (Series Operation) ต่อจุดภาพที่มีค่า 1 ของรูปแบบ M ในทิศทาง
สแกนจากซ้ายไปขวา (ดังแสดงในรูป 3.5 (ค)) จำนวน K ครั้งดังนี้

if (x1 or x2 or x3 or x4 or x5 or x6 or x7 = 2)

{ m_{1,j} = 2 }

ขั้นตอนที่ 3

กระทำแบบอนุกรมต่อจุดภาพที่มีค่า 2 ของรูปแบบ M ในทิศทาง สแกนจากบนลงล่าง ดังนี้

if (m_{1,j} = 2)

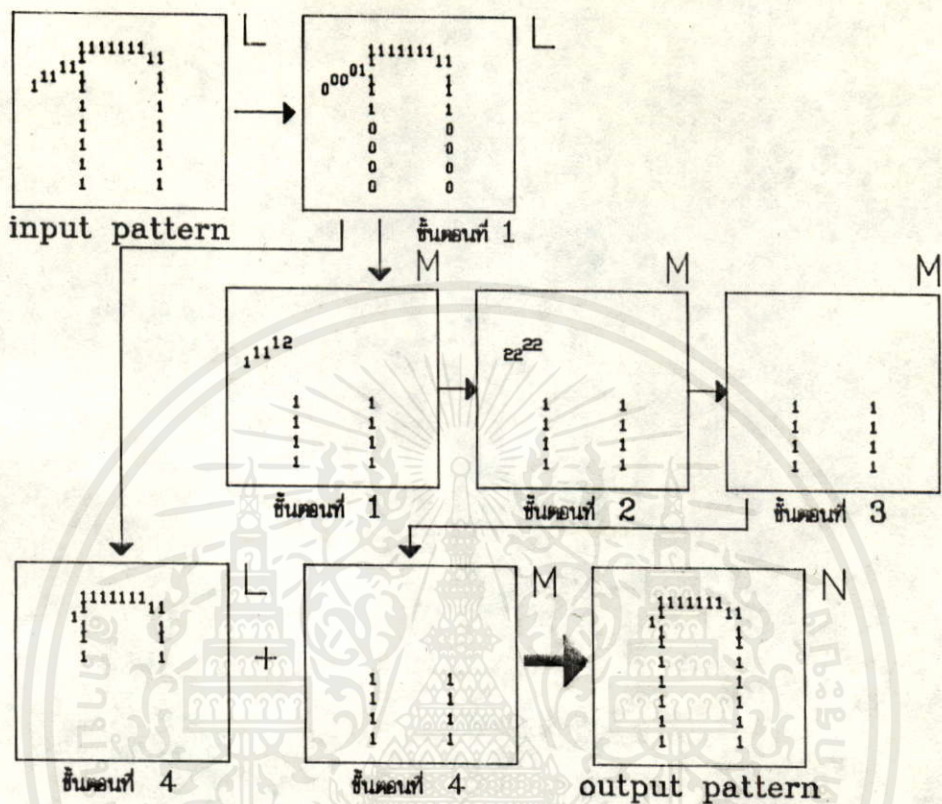
{ m_{1,j} = 0 }

ขั้นตอนที่ 4

รวมเอารูปแบบ M และรูปแบบ L เข้าด้วยกันและเก็บค่าไว้ในรูปแบบ N ดังนี้

n_{1,j} = m_{1,j} + l_{1,j}

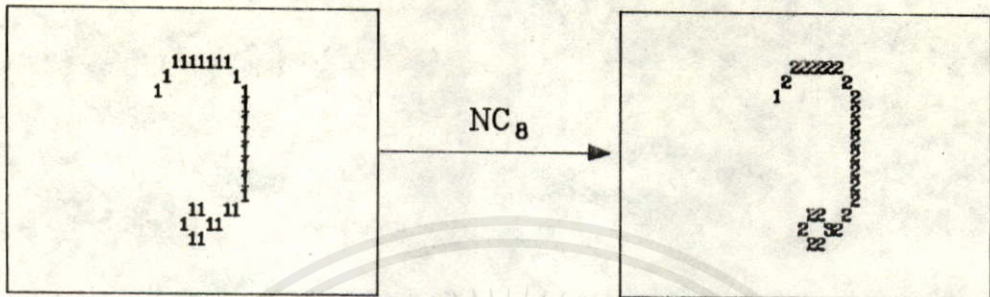
รูปแบบสองระดับ N ที่ได้จากขั้นตอนที่ 4 จะเป็นโครงร่างของตัวอักษรที่ถูกกำจัดเอาส่วน
เกินออกไปแล้ว บล็อกไดอะแกรมของการกำจัดส่วนเกินแสดงไว้ในรูปที่ 3.15



รูปที่ 3.15 บล็อกไดอะแกรมของการกำจัดส่วนเกิน

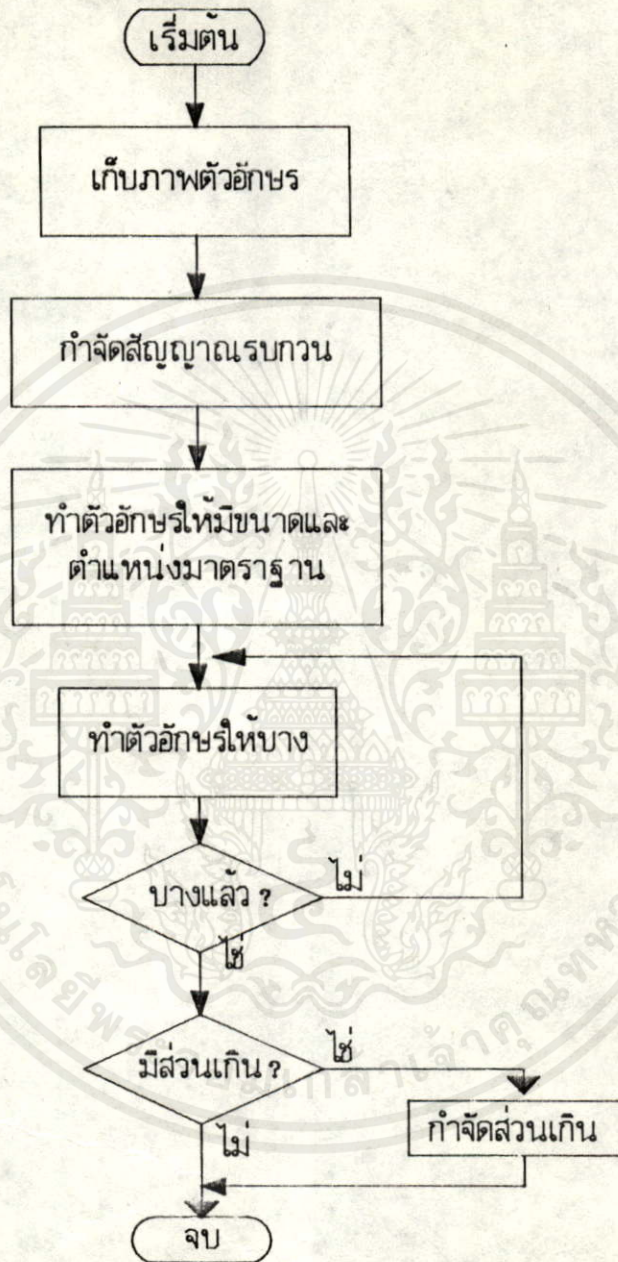
3.5 ค่าตัวเลขที่ต่อเนื่องของรูปแบบตัวอักษร

รูปแบบสองระดับที่เป็นโครงร่างซึ่งปราศจากส่วนเกินของตัวอักษร เมื่อนำมาทำการสแกนจากบนลงล่าง และคำนวณหาค่าตัวเลขต่อเนื่องของจุดภาพที่เป็น 1 (จุดภาพที่เป็นเนื้อของตัวอักษร) โดยใช้สมการที่ (2.1) และ (2.2) ก็จะได้รูปแบบของตัวอักษรออกมา ดังแสดงในรูปที่ 3.16 ค่าตัวเลขต่อเนื่องของรูปแบบตัวอักษรในรูปที่ 3.16 นี้จะแสดงถึงความสัมพันธ์ ระหว่างจุดภาพที่อยู่ติดกันโดยมีไม่ว่ากรณีใดๆ จุดภาพหนึ่งภาพหนึ่งให้ค่าต่อเนื่องหนึ่งค่า และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ความหมายดังที่แสดงไว้ในตารางที่ 2.1



รูปที่ 3.16 ค่าตัวเลขต่อเนื่องบนโครงสร้างของตัวอักษร

รูปแบบของตัวอักษรที่ผ่านการเตรียมข้อมูลแล้วจะเป็นรูปแบบที่มีความสมบูรณ์ พร้อมทั้งจะนำไปจำแนกออกเป็นกลุ่ม และกลุ่มย่อย ในขั้นตอนต่อไปของระบบการจดจำ สรุปรวมวิธีเกี่ยวกับการเตรียมข้อมูลของตัวอักษรเพื่อการจดจำตัวอักษรในระบบการจัดการล่องหน้า แสดงโดยผังงาน ในรูปที่ 3.17



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นการก้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งรูปที่ 3.17 ผังงาน แสดงระบบการจัดการลวงหน้า

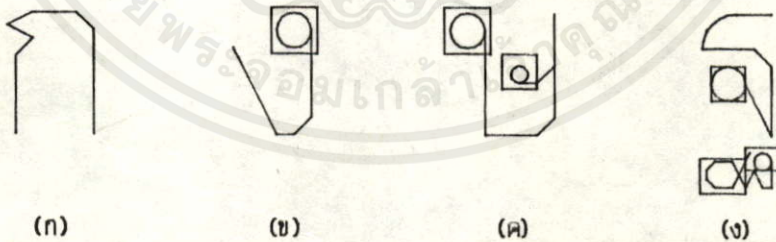
บทที่ 4

การตรวจสอบหัวและการแบ่งกลุ่มของตัวอักษร

ภาพของตัวอักษรแต่ละตัวที่ผ่านเข้ามาในระบบการจดจำ จะถูกนำไปเปลี่ยนให้เป็นรูปแบบสองระดับของตัวอักษรด้วยวิธีการดังที่ได้กล่าวไว้แล้วในบทที่ 2 รูปแบบสองระดับที่ได้นี้จะนำไปลอกแบบออกเป็น 2 รูปแบบ รูปแบบ ของตัวอักษรส่วนที่หนึ่งใช้เป็นรูปแบบสำหรับการตรวจสอบหาหัวของตัวอักษร ^[15] และรูปแบบของตัวอักษรในส่วนที่สองใช้เป็นรูปแบบสำหรับระบบการจัดการล่วงหน้า ซึ่งระบบการจัดการล่วงหน้านี้ได้แสดงถึงรายละเอียดไว้แล้วในบทที่ 3 ดังนั้นเนื้อหาในบทนี้ก็จะได้กล่าวถึง ส่วนที่มีความสำคัญที่สุดของระบบการจำแนกตัวอักษร (Classification) ออกเป็นกลุ่ม ๆ

4.1 การตรวจสอบหัวของตัวอักษร (Head detection)

หัวของตัวอักษรในภาษาไทย จะมีลักษณะเป็นวงปิดขนาดเล็ก (เมื่อเทียบกับส่วนอื่นๆ ในตัวอักษร) วางอยู่ตามตำแหน่งต่าง ๆ และมีจำนวนต่าง ๆ กันออกไป สำหรับรูปแบบของตัวอักษรแต่ละตัว ดังแสดงในรูปที่ 4.1



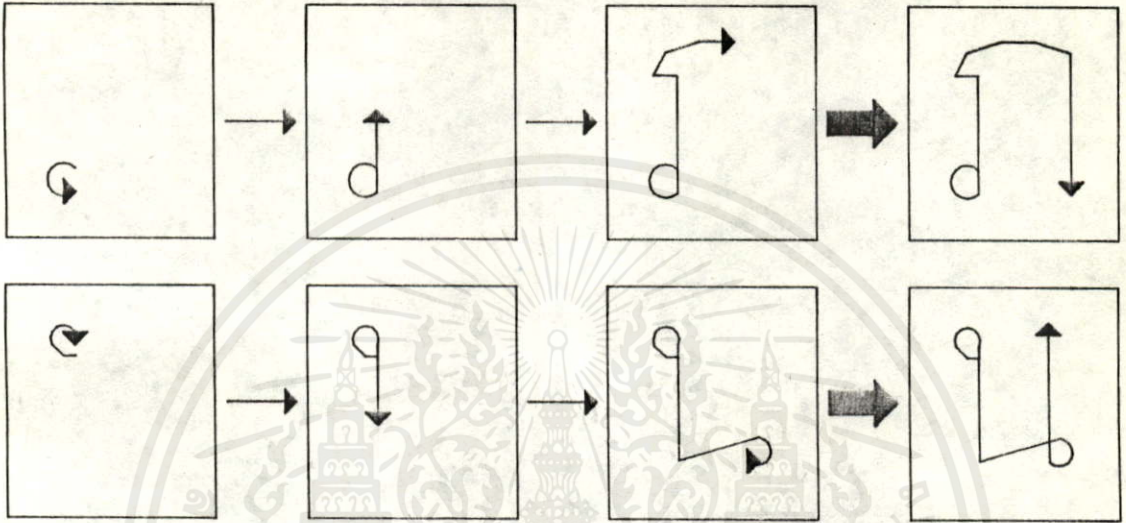
รูปที่ 4.1 จำนวนและตำแหน่งของหัว

(ก) ตัวอักษรที่ไม่มีหัว (ข) ตัวอักษรที่มีหัว 1 หัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้ง (ค) ตัวอักษรที่มีหัว 2 หัว (ง) ตัวอักษรที่มีหัว 3 หัว การนำไปใช้

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



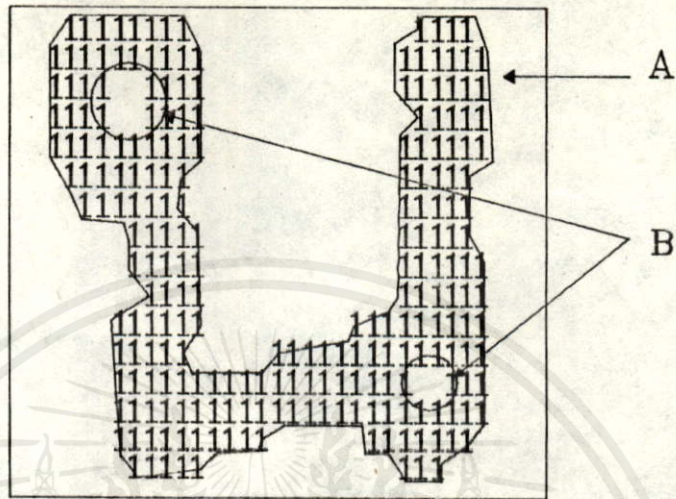
รูปที่ 4.2 ลักษณะการเขียนตัวอักษร

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

ดังนั้นจึงถือได้ว่า หัวของตัวอักษรภาษาไทยเป็นลักษณะที่เด่นชัดอย่างหนึ่งของตัวอักษร ซึ่งในวิทยานิพนธ์ฉบับนี้จะได้นำเอาลักษณะเด่นอันนี้มาพิจารณาถึงความแตกต่างในเรื่องของจำนวนและตำแหน่งเพื่อใช้เป็นเครื่องมือในการจำแนกกลุ่มของตัวอักษร

จากรูปแบบสองระดับของตัวอักษรภาษาไทยในรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

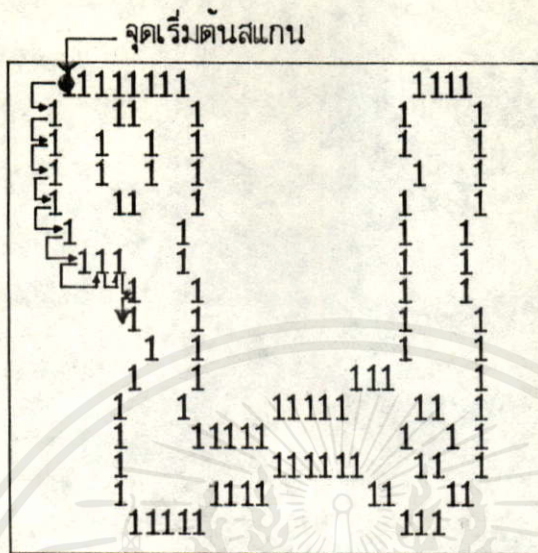


รูปที่ 4.3 รูปแบบสองระดับสำหรับการตรวจสอบหัว

จะเห็นได้ว่าขอบของภาพตัวอักษรจะแยกกันเป็น 2 ลักษณะคือ ส่วนที่เป็นขอบนอก (A) และขอบใน (B) ขอบในนี้เองที่ประกอบกันขึ้นเป็นหัวของตัวอักษร

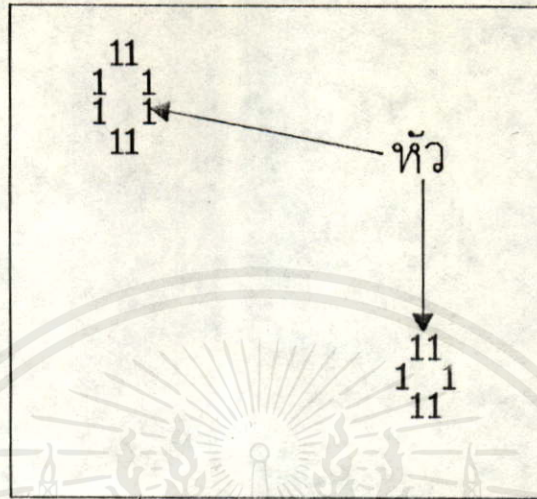
การตรวจสอบหัวของตัวอักษรจะเริ่มต้นโดยการทำ สแกนจากบนลงล่าง (รูปที่ 3.5 (ก)) ไปบนรูปแบบสองระดับที่ยังไม่ได้มีการทำให้บาง แล้วคำนวณหาค่าตัวเลขต่อเนื่อง NC_4 และ NC_8 (โดยใช้สมการที่ (2.1) และ (2.2)) ของจุดภาพใด ๆ บนรูปแบบและทำการเก็บค่าไว้ในรูปแบบ ก็จะทำให้สามารถหาส่วนที่เป็นขอบในและขอบนอกของตัวอักษรออกมาได้ดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปไว้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การติดตามขอบภาพเพื่อการตรวจสอบหัว

ในขณะที่ทำการติดตามขอบเขตของภาพอยู่นี้ ก็จะมีบันทึกเอาตำแหน่งและจำนวนของจุดภาพไว้ และเมื่อการติดตามขอบเขตของภาพสิ้นสุดลงที่จุดเริ่มต้น ก็จะได้ค่า 0 ลงไปในตำแหน่งของจุดภาพที่ได้บันทึกไว้ (ตัดขอบนอกออกไปจากรูปแบบ) จะได้รูปแบบที่เป็นขอบของภาพตัวอักษรออกมาดังรูปที่ 4.6



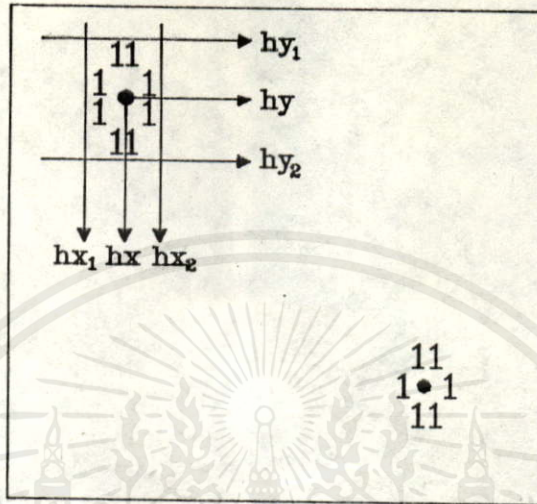
รูปที่ 4.6 ขอบของภาพตัวอักษรที่เป็นหัว

เมื่อถึงขั้นตอนนี้จะพบว่าหัวของตัวอักษรก็จะปรากฏออกมาดังแสดงในรูปที่ 4.6

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

ตำแหน่งที่ใช้แทนหัวจะหาได้จากการทำสแกนภาพใน 4 ทิศทาง (รูปที่ 3.5) เพื่อหากรอบบน ล่าง ซ้าย และขวาของหัวดังรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 การหาตำแหน่งของหัว

ตำแหน่งที่ใช้แทนหัวซึ่งต่อไปจะเรียกว่าตำแหน่งของหัว จะหาได้จากสมการที่ (4.1) และ

(4.2)

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

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

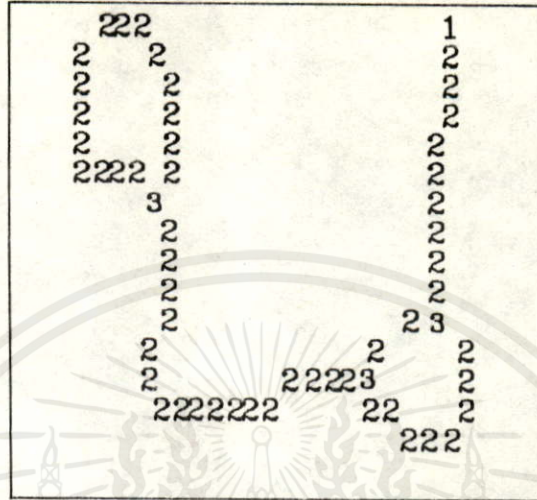
เมื่อนำเอารูปแบบสองระดับของตัวอักษรที่ผ่านการเตรียมข้อมูลแล้วมาหาค่า NC_g ก็จะได้

รูปแบบตัวอักษรที่ประกอบด้วยค่าตัวเลขต่อเนื่องดังรูปที่ 4.8

ดังนั้นจะเห็นได้ว่ารูปแบบสองระดับสำหรับใช้ตรวจสอบหัว และรูปแบบสองระดับสำหรับการเตรียมข้อมูล จะกระทำแยกจากกัน โดยจะกระทำกระบวนการใดก่อนก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาก่อนอื่น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

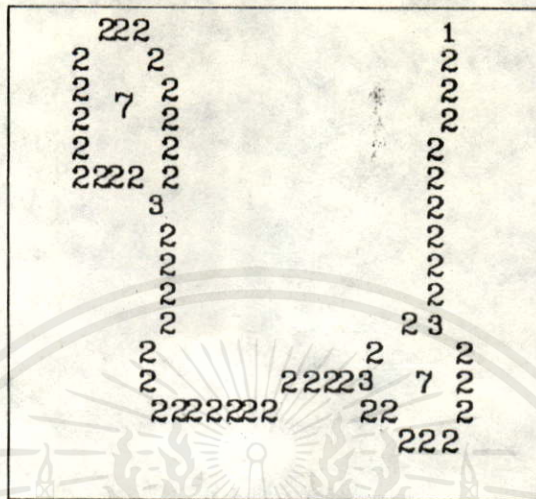


รูปที่ 4.8 รูปแบบที่มีค่าตัวเลขต่อเนื่อง

1 คือจุดปลาย : 2 คือจุดต่อเนื่อง : 3 คือจุดแยก

รวมเอาตำแหน่งของหัวที่คำนวณได้จาก สมการ (4.1) และ (4.2) เข้ากับรูปแบบที่มี

ค่าตัวเลขต่อเนื่องของตัวอักษร ก็จะได้รูปแบบของตัวอักษรออกมาดังรูปที่ 4.9



รูปที่ 4.9 รูปแบบตัวอักษรสำหรับการจำแนกกลุ่ม

1 คือจุดปลาย : 2 คือจุดต่อเนื่อง : 3 คือจุดแยก : 7 คือจุดหัว

รูปแบบของตัวอักษรในรูปที่ 4.9 นี้จะเป็นรูปแบบที่มีคุณสมบัติต่าง ๆ ควบถ้วนพร้อมที่จะนำไปจำแนกกลุ่มต่อไป

4.2 การแบ่งกลุ่มของตัวอักษรแบบทหยาบ (Rought Classification)

จากกรรมวิธีการตรวจสอบหัวของตัวอักษรในหัวข้อที่ผ่านมา ก็จะสามารถหาจำนวนและตำแหน่งของหัวในตัวอักษรแต่ละตัวที่ผ่านเข้ามาในระบบการจดจำได้ โดยจะแบ่งตัวอักษรออกเป็น 4 กลุ่ม (class) ตามจำนวนของหัวดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าการฉ้อโกง กลุ่มที่ 1 (Class1) คือกลุ่มที่ไม่มีหัว (No head) เช่น ตัวอักษร ก ช ไป๋

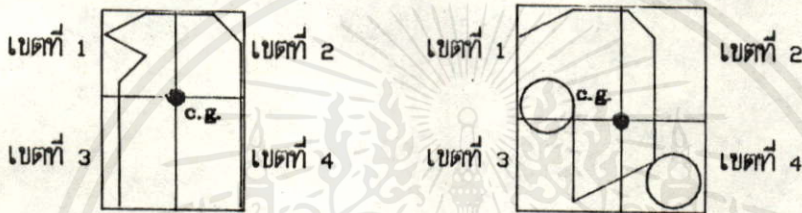
กลุ่มที่ 2 (Class2) คือกลุ่มที่มีหัว 1 หัว (One head) เช่น ตัวอักษร บ ย ภ

กลุ่มที่ 3 (Class3) คือกลุ่มที่มีหัว 2 หัว (Two head) เช่น ตัวอักษร ฉ ม ห

กลุ่มที่ 4 (Class4) คือกลุ่มที่มีหัว 3 หัว (Three head) ได้แก่ ตัวอักษร จู

รูปแบบของตัวอักษรในกลุ่มที่ 1 และกลุ่มที่ 3 จะถูกนำมาคำนวณหาจุดศูนย์กลางของตัวอักษร โดยใช้สมการที่ (2.3) และ (2.4) เพื่อที่จะใช้จุดนี้แบ่งตัวอักษรออกเป็น 4 เขต (Zone) ดังรูปที่

4.10



รูปที่ 4.10 การแบ่งเขตของตัวอักษรกลุ่มที่ 1 และ กลุ่มที่ 3

เมื่อสังเกตถึงตำแหน่งหัวของตัวอักษรใน 4 เขตที่แบ่งได้ ก็จะสามารถแบ่งกลุ่มย่อย (Sub Class) ของตัวอักษร ได้ดังนี้

กลุ่มย่อย C11 คือกลุ่มย่อยที่มีหัวอยู่ในเขต 1 ทั้ง 2 หัว

กลุ่มย่อย C12 คือกลุ่มย่อยที่มีหัวอยู่ในเขต 1 และเขต 2

กลุ่มย่อย C13 คือกลุ่มย่อยที่มีหัวอยู่ในเขต 1 และเขต 3

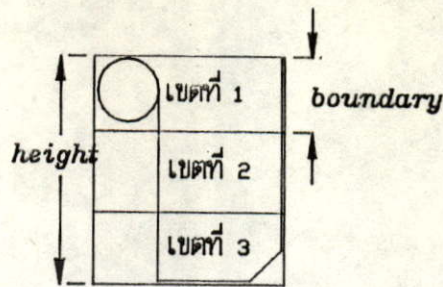
กลุ่มย่อย C14 คือกลุ่มย่อยที่มีหัวอยู่ในเขต 1 และเขต 4

กลุ่มย่อย C23 คือกลุ่มย่อยที่มีหัวอยู่ในเขต 2 และเขต 3

กลุ่มย่อย C33 คือกลุ่มย่อยที่มีหัวอยู่ในเขต 3 ทั้ง 2 หัว

กลุ่มย่อย C34 คือกลุ่มย่อยที่มีหัวอยู่ในเขต 3 และ เขต 4

สำหรับตัวอักษรในกลุ่มที่ 2 จะแบ่งออกเป็น 3 เขตดังรูปที่ 4.11



รูปที่ 4.11 การแบ่งเขตของตัวอักษรกลุ่มที่ 2

โดยที่ความกว้างของแต่ละเขตจะหาได้จาก

$$\text{boundary} = \text{height} / 3$$

จากการสังเกตหัวของตัวอักษรในกลุ่มที่ 2 นี้ก็จะสามารถแบ่งตัวอักษรออกเป็นกลุ่มย่อยตามตำแหน่งของหัวในเขตที่แบ่งได้เป็น 3 กลุ่มย่อยดังนี้

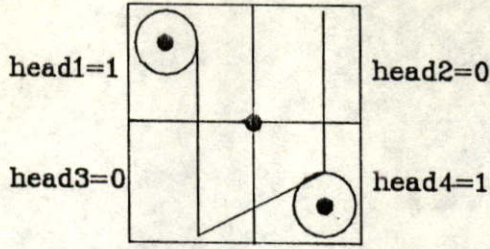
กลุ่มย่อย H1 (Upper Head) คือกลุ่มย่อยที่มีหัวอยู่ในเขต 1

กลุ่มย่อย H2 (Middle Head) คือกลุ่มย่อยที่มีหัวอยู่ในเขต 2

กลุ่มย่อย H3 (Lower Head) คือกลุ่มย่อยที่มีหัวอยู่ในเขต 3

ตัวอักษรในกลุ่มย่อย H1, H2 และ H3 นี้ยังจะนำไปแบ่งเขตออกเป็น 4 เขต ด้วยวิธีการเช่นเดียวกันกับการแบ่งเขตของตัวอักษร ในกลุ่มที่ 1 และกลุ่มที่ 3 เพื่อประโยชน์ในการจำแนกกลุ่มของตัวอักษรอย่างละเอียด ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

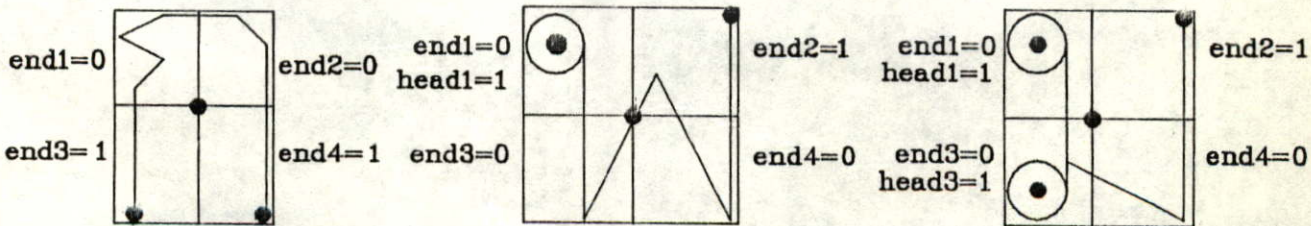


รูปที่ 4.12 ตำแหน่งหัวของตัวอักษรในกลุ่มย่อย

รูปที่ 4.12 แสดงถึงตำแหน่งของหัว ที่ปรากฏอยู่ตามเขตต่าง ๆ $head1 = 1$ และ $head4 = 1$ มีความหมายว่า ตัวอักษร "น" มีหัวอยู่ในเขตที่ 1 และ เขตที่ 4 ตามลำดับ ส่วน $head2 = 0$ และ $head3 = 0$ ก็หมายความว่าไม่มีหัวอยู่ในเขตที่ 2 และเขตที่ 3 ตามลำดับ ส่วนตัวอักษรในกลุ่มที่ 4 นี้มีเพียงตัวเดียว คือ "ง" ดังนั้นจึงไม่กระทำการใด ๆ กับตัวอักษรในกลุ่มนี้ เพราะถือว่าเป็นผลลัพธ์ขั้นสุดท้ายของระบบการจดจำแล้ว

4.3 การแบ่งกลุ่มของตัวอักษรแบบละเอียด (Fine classification)

ตัวอักษรในกลุ่มย่อยที่ได้จากหัวข้อ 4.2 จะถูกนำมาแยกแยะความแตกต่างของแต่ละตัวอักษร ในหัวข้อนี้ ด้วยวิธีการพิจารณาถึงตำแหน่งจุดปลาย (end point) ของตัวอักษรในแต่ละเขตที่แบ่ง ดังรูปที่ 4.13



(ก) (ข) (ค)

เอกสารนี้ (ก) เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ (ข) ยานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ (ค) ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.13 ตำแหน่งของจุดปลาย

ในรูปที่ 4.13 แสดงถึงตำแหน่งของจุดปลายที่อยู่ในเขตต่าง ๆ ส่วน $end3 = 1$ และ $end4 = 1$ ในรูปที่ 4.13 (ก) นั้นหมายความว่า ตัวอักษร "ก" มีจุดปลายอยู่ในเขตที่ 3 และเขตที่ 4 ตามลำดับ เช่นเดียวกัน $end1 = 0$ และ $end2 = 0$ ก็หมายความว่าไม่มีจุดปลายอยู่ในเขตที่ 1 และเขตที่ 2 ตามลำดับ

สำหรับตัวอักษรในรูปที่ 4.13 (ข) และ (ค) ก็จะมีการพิจารณาในลักษณะเดียวกัน

4.4 เทคนิคและวิธีการแยกตัวอักษรที่คล้ายกัน

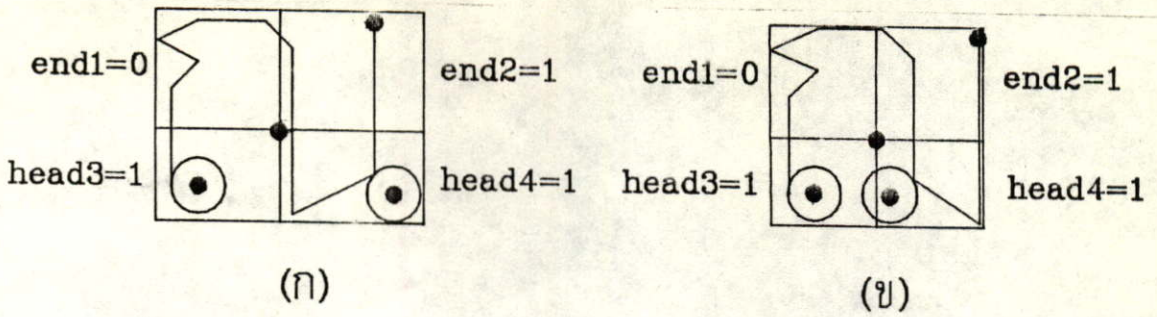
ในกลุ่มย่อยของตัวอักษรหลาย ๆ กลุ่ม จะมีตัวอักษรที่คล้ายกันอยู่หลายตัว เช่น ในกลุ่มย่อยของตัวอักษร "ช" กับ "บ" หรือ "ฉ" กับ "ณ" กับ "ณ" เป็นต้น ดังนั้นจึงได้นำเสนอถึงเทคนิคและวิธีการหลาย ๆ แบบ ในการแยกตัวอักษรแต่ละตัวออกจากกัน เพื่อเพิ่มประสิทธิภาพของระบบการจดจำให้สูงขึ้นเทคนิคต่าง ๆ ที่นำเสนอมีดังนี้

1. การแบ่งเขตย่อย (Sub Zone)
2. การหารหัสรูปร่าง (Feature Code) บางส่วนของตัวอักษร ^[13]
3. การหาลักษณะหัวของตัวอักษร (Head Style)
4. การหาอัตราส่วนความกว้างต่อความสูงของตัวอักษร (Width per Height ratio)

4.4.1 การแบ่งเขตย่อย (Sub Zone)

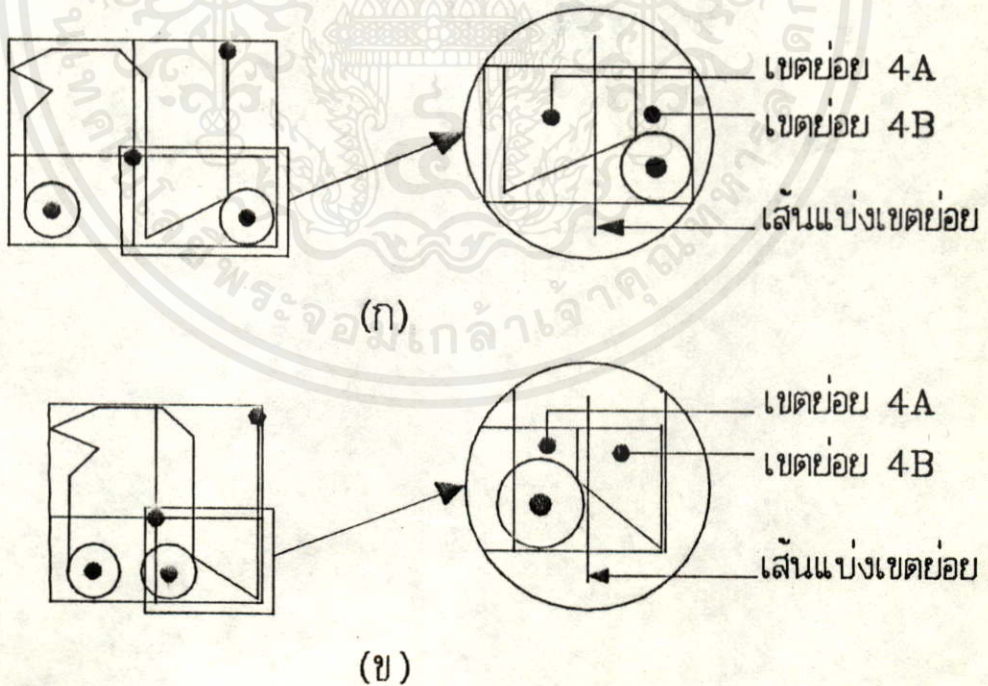
การแบ่งเขตของตัวอักษรออกเป็น 4 เขต ตามขั้นตอนของการแบ่งกลุ่มของตัวอักษรแบบขยายในหัวข้อ 4.2 นั้น บางกรณีจะเกิดความไม่แน่นอนในเรื่องตำแหน่งของหัว เช่น ตัวอย่างในรูปที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 ตำแหน่งของหัวที่อาจจะไม่แน่นอน

จากรูปที่ 4.14 จะเห็นว่าตำแหน่งของหัวตัวอักษรทั้งรูป (ก) และรูป (ข) เป็นตำแหน่งเดียวกันคืออยู่ในเขตที่ 3 และเขตที่ 4 แต่จากการพิจารณารูปร่างของตัวอักษรทั้ง 2 ตัวนี้พบว่า ตัวอักษร "ณ" จะมีหัวที่ ค่อนไปทางด้านขวาของเขตที่ 4 เสมอ ดังนั้นจึงพิจารณาแบ่งเขตที่ 4 นี้ออกเป็นเขตย่อย (Sub Zone) อีก 2 เขต เพื่อการตัดสินใจดังรูปที่ 4.15

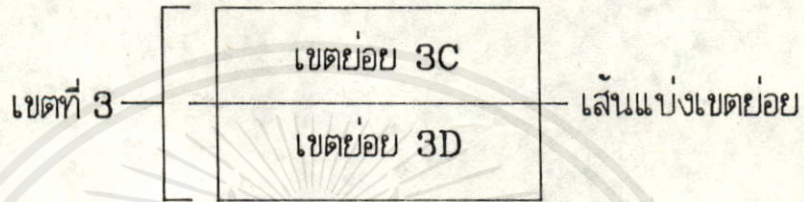


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.15 การแบ่งเขตย่อย

เขตย่อยทั้ง 2 เขตคือ เขตย่อย 4A และเขตย่อย 4B จะแบ่งใหม่พื้นที่เท่าๆ กัน โดยใช้เส้นแบ่งครึ่งเขต ดังรูปที่ 4.15

การแบ่งเขตย่อยนี้อาจจะแบ่ง ในอีกทิศทางหนึ่งคือ ในแนวระดับก็ได้ดังรูป 4.16

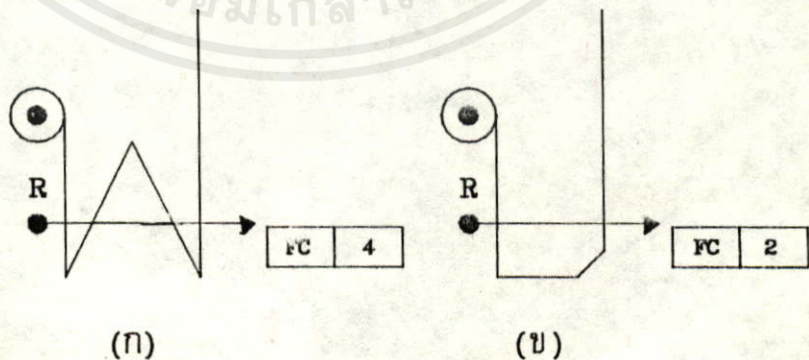


รูปที่ 4.16 การแบ่งเขตย่อย

การเลือกลักษณะของการแบ่งเขตย่อย ก็ขึ้นอยู่กับลักษณะของตัวอักษรในกลุ่มย่อย เป็นสำคัญ

4.4.2 การหารหัสรูปร่าง (Feature code) บางส่วนของตัวอักษร [13]

รหัสรูปร่างของตัวอักษรจะหาได้จากการกำหนดจุดภาพที่ใช้ เป็นจุดอ้างอิงของรหัส แล้วสแกนหาจุดภาพที่เป็นตัวอักษร ไปในทิศทางที่ต้องการ เมื่อได้จำนวนจุดภาพที่เป็นตัวอักษรแล้วก็บันทึกจำนวนจุดภาพนั้นไว้เป็นรหัสของรูปร่าง FC ดังรูปที่ 4.17

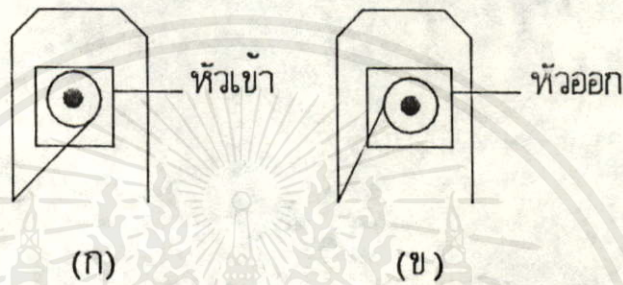


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอก รูปที่ 4.17 รหัสรูปร่างของตัวอักษร หล่อรังที่มีคนนำไปใช้

จุด R ในรูปที่ 4.17 (ก) และ (ข) เป็นจุดอ้างอิงของการทหารหัส ตำแหน่งของจุด R จะแตกต่างกันออกไปตามแต่ละกลุ่มย่อยของตัวอักษร

4.4.3 การหาลักษณะหัวของตัวอักษร (Head Style)

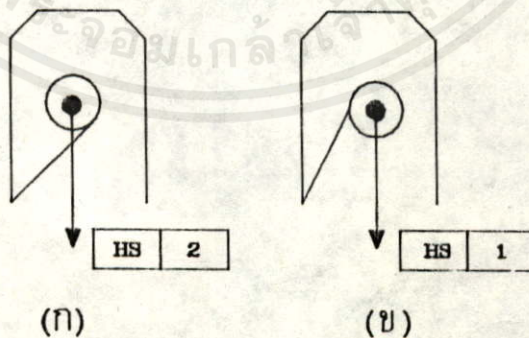
หัวของตัวอักษรภาษาไทยจะมีลักษณะที่แตกต่างกันอยู่ 2 แบบ คือ หัวที่มีลักษณะมีวนเข้า (รูปที่ 4.18 (ก)) กับหัวที่มีลักษณะมีวนออก (รูปที่ 4.18 (ข))



รูปที่ 4.18 ลักษณะของหัว

(ก) หัวมีวนเข้า (ข) หัวมีวนออก

วิธีง่าย ๆ ในการตรวจสอบลักษณะของหัว คือการพิจารณาหัวรูปร่างในทิศทางใดทิศทางหนึ่ง โดยใช้ตำแหน่งหัวเป็นจุดอ้างอิงของการทหารหัส ดังรูปที่ 4.19



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.19 รหัสรูปร่างของหัว หากนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกหนึ่งข้อที่ต้องคำนึงถึงคือว่าเอกสารฉบับนี้เป็นการนำไปใช้ (ก) รหัสรูปร่างของหัวมีวนเข้า (ข) รหัสรูปร่างของหัวมีวนออก

รหัสของรูปร่าง ของตัวอักษรในรูปที่ 4.19 (ก) และ (ข) จะเป็นตัวบอกถึงลักษณะของหัว HS

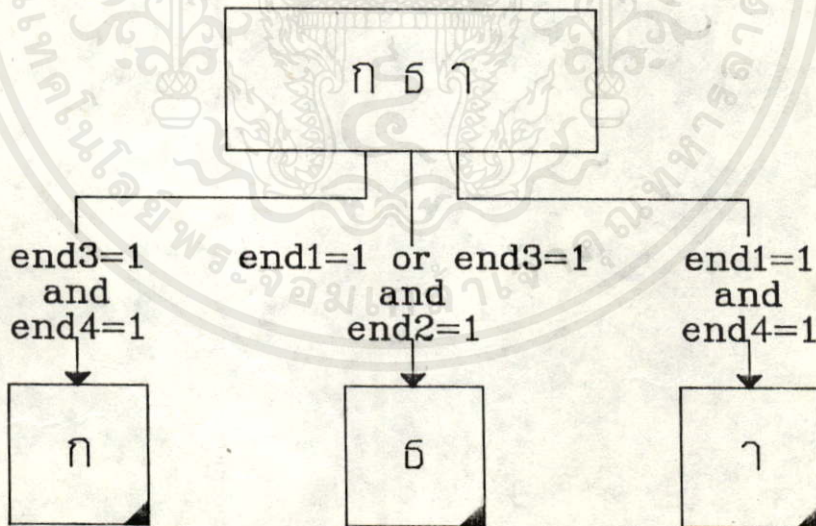
HS = 1 ตัวอักษรจะมีหัวม้วนออก และ

HS = 2 ตัวอักษรจะมีหัวม้วนเข้า

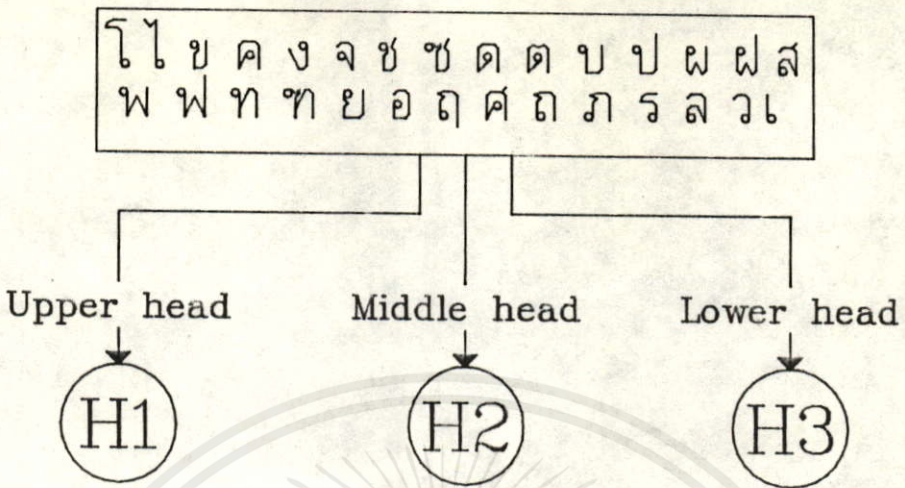
4.4.4 การหาอัตราส่วนความกว้างต่อความสูงของตัวอักษร (Width per Height ratio)

ตัวอักษร "ข" กับตัวอักษร "บ" จะมีลักษณะที่คล้ายกันมากที่สุด แต่ก็ยังมีลักษณะที่แตกต่างกันที่ความกว้างของตัวอักษร ดังนั้นจึงนำเอาอัตราส่วนของความกว้างต่อความสูงของตัวอักษร WH ratio มาใช้เป็นตัวยกตัวอักษร 2 ตัวนี้ออกจากกัน

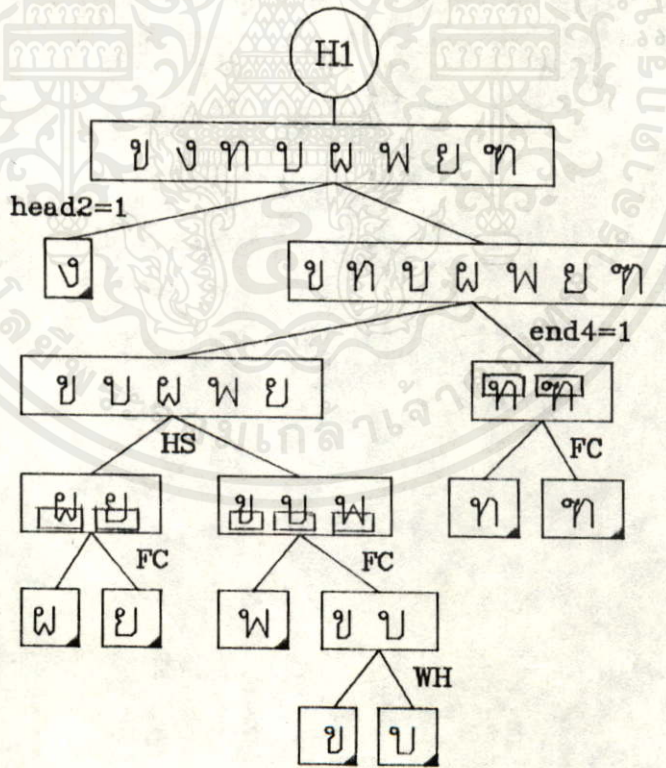
จากการแบ่งกลุ่มของตัวอักษร การแบ่งกลุ่มย่อยตลอดจนเทคนิคต่าง ๆ ในการแยกตัวอักษรออกจากกลุ่มย่อย ที่กล่าวมาในบทนี้ จะสามารถจำแนกกลุ่มและแยกแยะตัวอักษรลายมือเขียนภาษาไทยได้ โดยการทำให้โครงร่างแบบต้นไม้ (Tree structure) ดังแสดงในรูปที่ 4.20 ถึงรูปที่ 4.32



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครู ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และทำซ้ำ หรือเผยแพร่เอกสารจากผู้อื่นที่นำเข้าไปใช้
รูปที่ 4.20 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มที่ 1

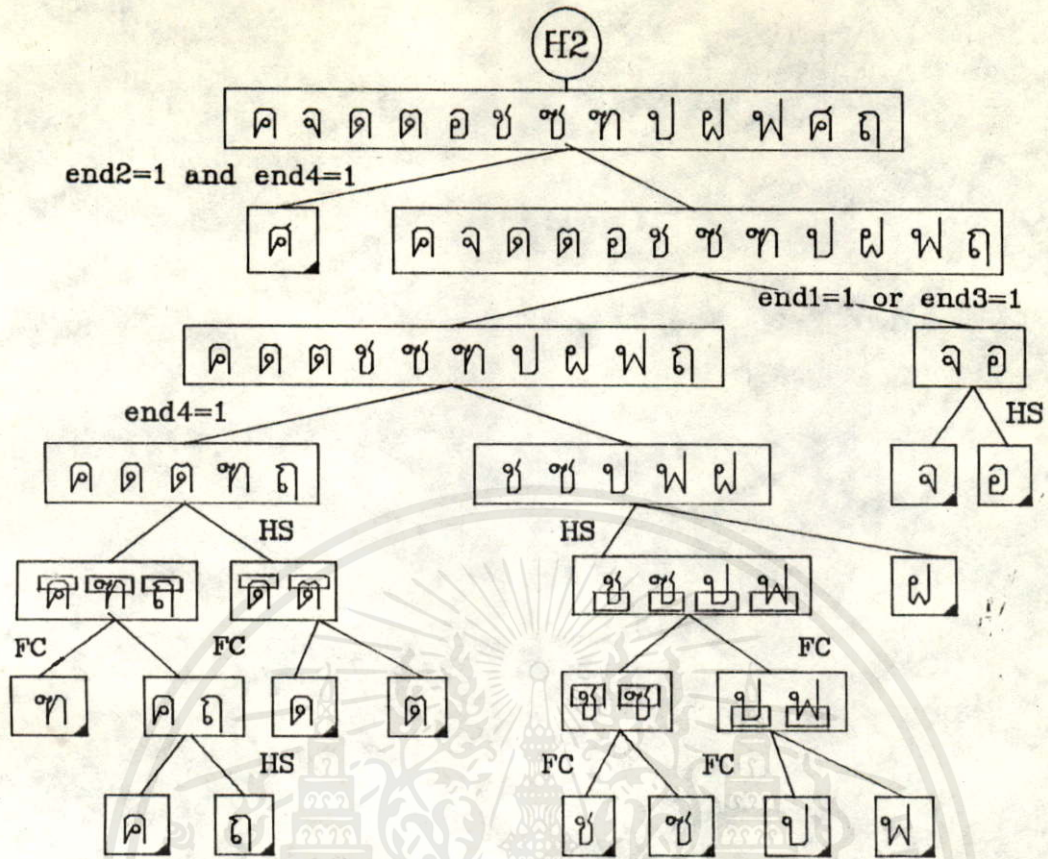


รูปที่ 4.21 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มที่ 2

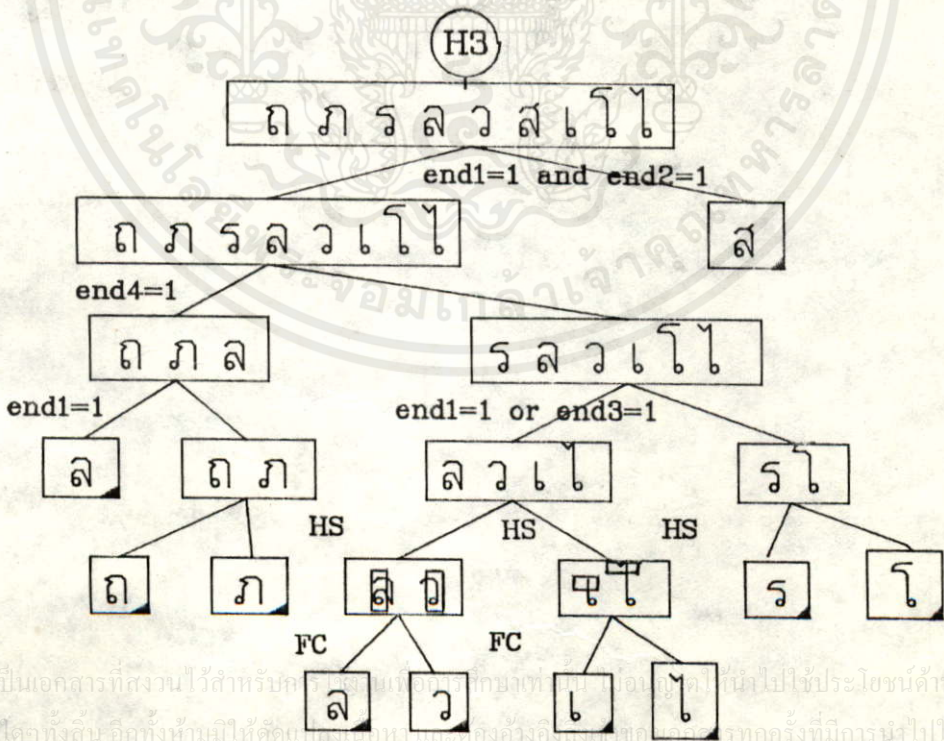


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.22 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย H1

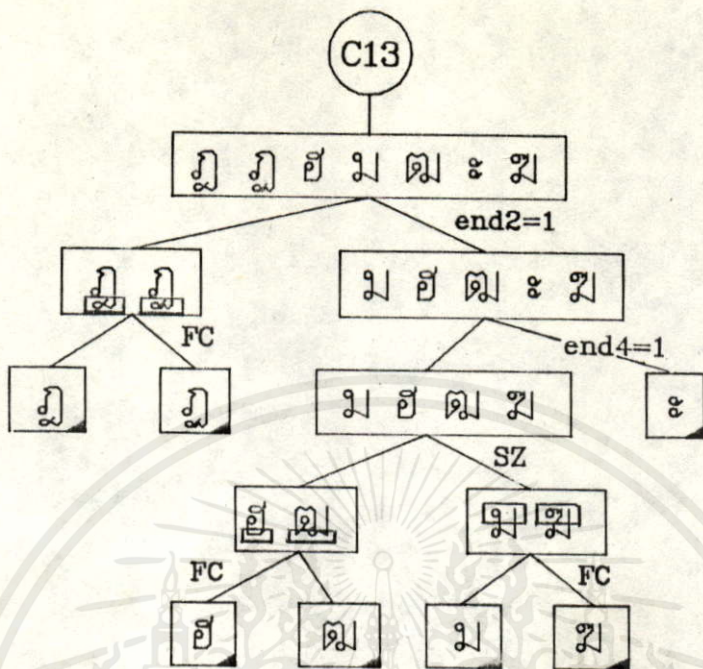


รูปที่ 4.23 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย H2

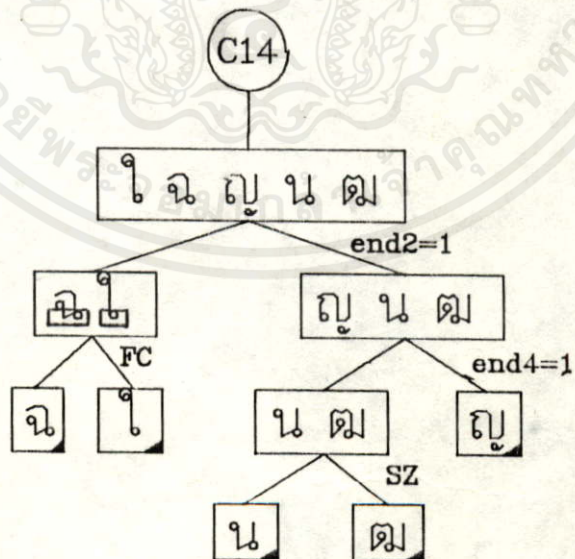


รูปที่ 4.24 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย H3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับคณาจารย์และบุคลากรในมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรตีพิมพ์หรือทำซ้ำโดยไม่ได้รับอนุญาตจากมหาวิทยาลัย

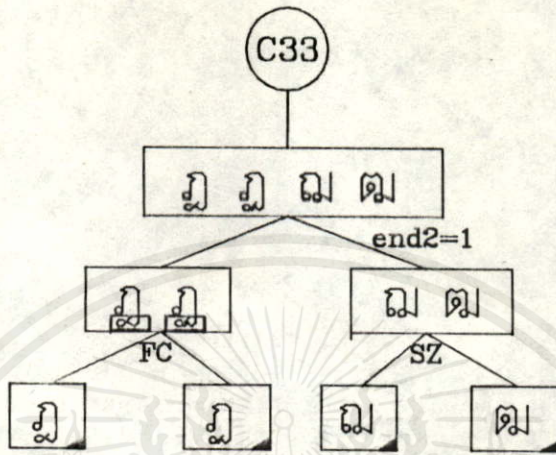


รูปที่ 4.27 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย C13

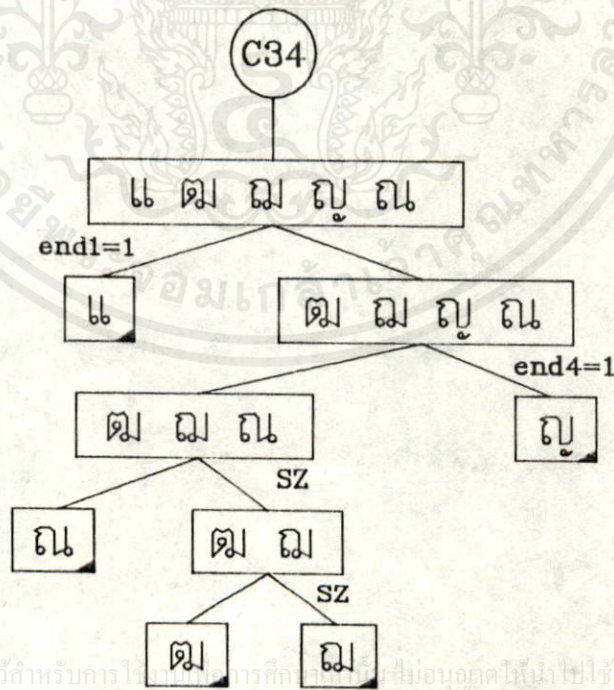


รูปที่ 4.28 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย C14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



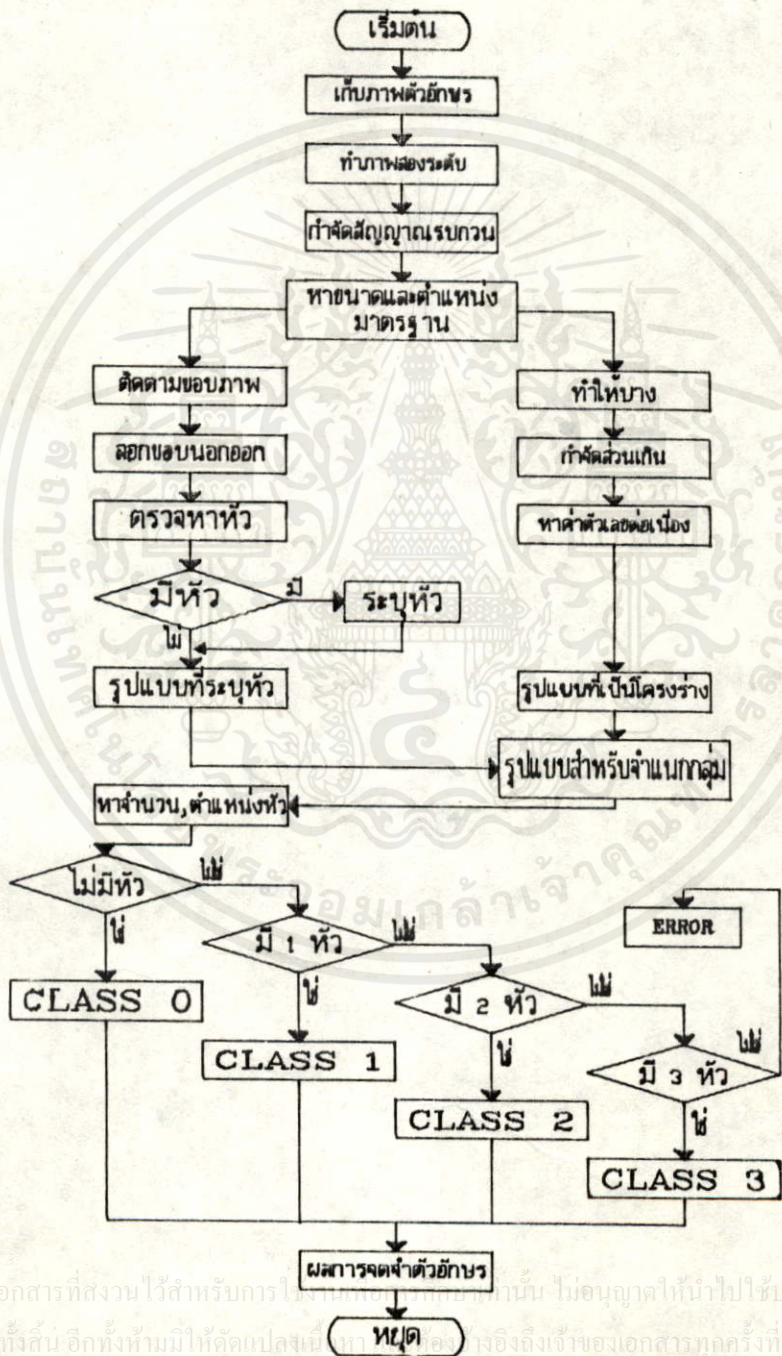
รูปที่ 4.29 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย C33



รูปที่ 4.30 โครงร่างแบบต้นไม้สำหรับตัวอักษรในกลุ่มย่อย C34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในการศึกษา ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีคนนำไปใช้

จากโครงร่างแบบต้นไม้อำนาจสำหรับการจัดจำตัวอักษรนี้ จะเห็นว่าตัวอักษรบางตัว ถูกจัดให้อยู่ในหลายๆ กลุ่มย่อย เช่น ตัวอักษร "ฉ" จัดให้อยู่ในกลุ่มย่อย C13, C14 และ C33 ที่ทำเช่นนี้เพราะเป็นการป้องกันความผิดพลาดของระบบการจัดจำ ในกรณีที่ตัวอักษรลายมือ เขียนที่เป็นอินพุท มีลักษณะที่ผิดเพี้ยน เช่นตัวอักษรมีการเอียงหน้า เอียงหลังมากเกินไป หรือหัวของตัวอักษรอยู่ในตำแหน่งที่ไม่ถูกต้อง เป็นต้น ผังงานในรูปที่ 4.33 แสดงถึงขั้นตอนทั้งหมดสำหรับระบบการจัดจำตัวอักษรลายมือเขียน

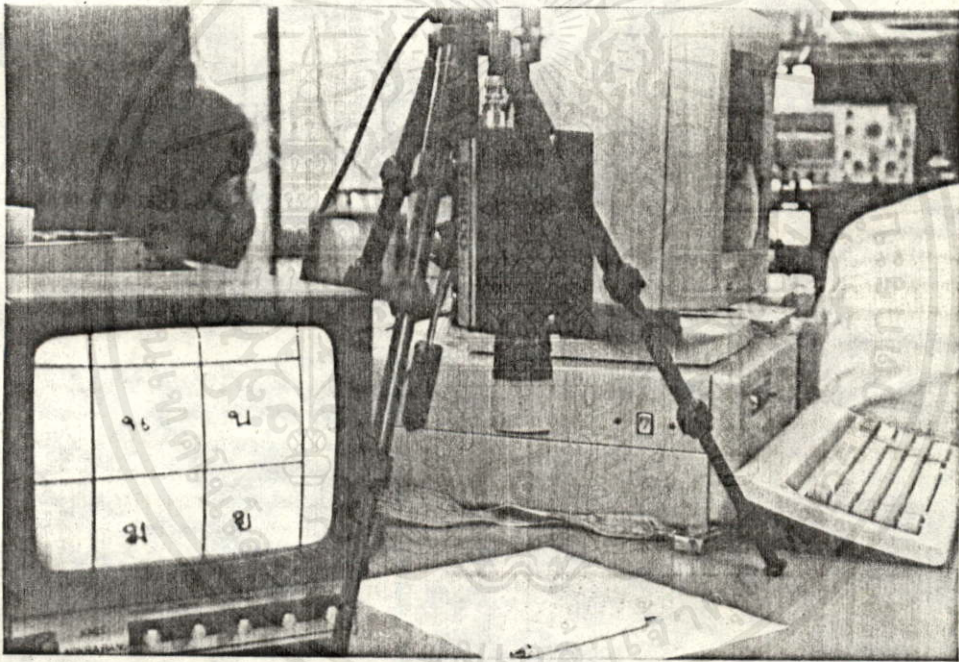


รูปที่ 4.33 ผังงานของระบบการจัดจำตัวอักษร

บทที่ 5

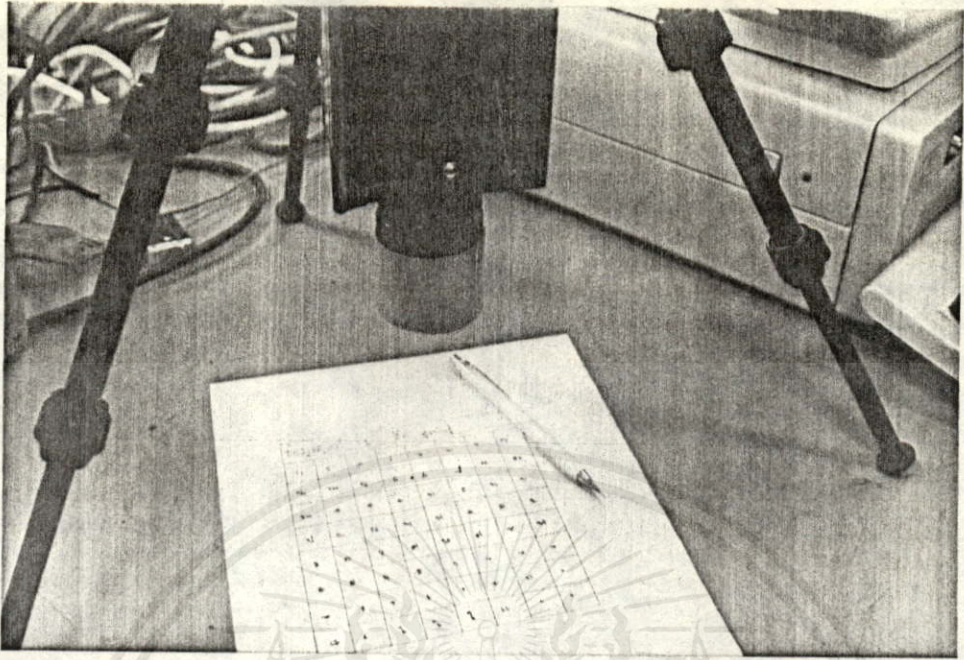
การทดลองจำแนกกลุ่มและจดจำตัวอักษร

นำเอาตัวอักษรลายมือเขียนที่เขียนโดยผู้เขียน 15 คน คนละ 1 ชุด ชุดละ 64 ตัวอักษรมาผ่านขั้นตอนการจดจำตัวอักษรที่นำเสนอ โดยเริ่มต้นที่การเก็บภาพเข้ามาด้วยกล้อง CCD ที่ติดตั้งขยายไว้ ซึ่งเป็นส่วนหนึ่งในระบบดิจิทัลไจเซอร์ (Digitizer) ที่สร้างขึ้นมาใช้งานเอง ภายในห้องทดลอง ดังแสดงในรูปที่ 5.1

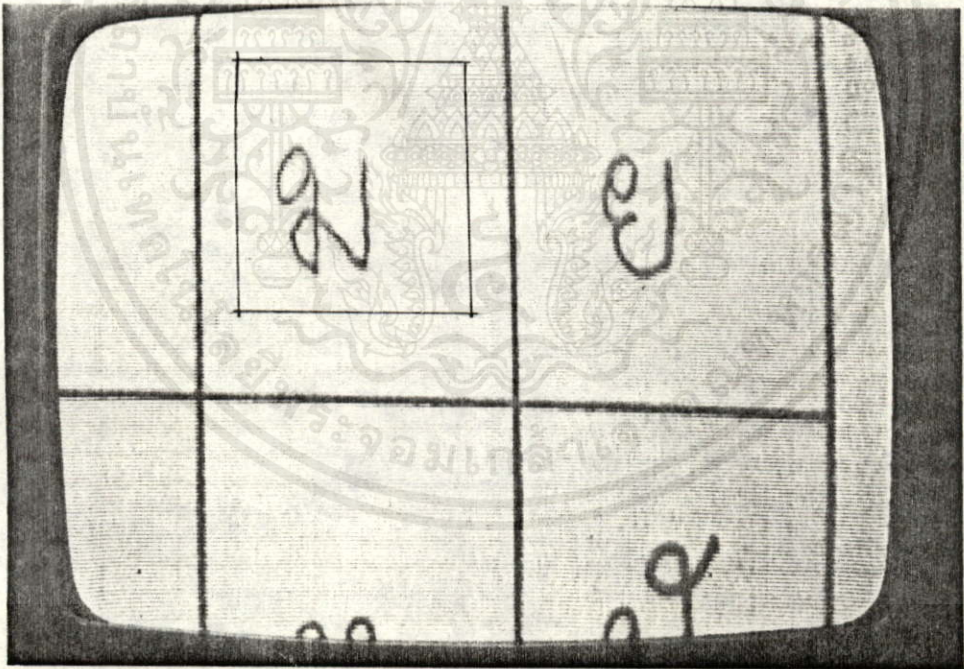


รูปที่ 5.1 ระบบที่ใช้ในการทดลอง

ข้อมูลภาพที่เป็นระดับเทา 256 ระดับของตัวอักษรขนาด 64 X 64 จุดภาพ ดังแสดงไว้ในรูปที่ 5.2 (ก) และ (ข) จะถูกส่งมายังเครื่อง IBM PC/AT (ทำงาน ที่ความถี่ของสัญญาณนาฬิกา 8 MHz) แล้วจึงทำการเปลี่ยนข้อมูลภาพ ให้เป็นรูปแบบสองระดับของตัวอักษรขนาด 64 X 64 จุดภาพ ผลที่ได้แสดงในรูปที่ 5.3 มิให้คิดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



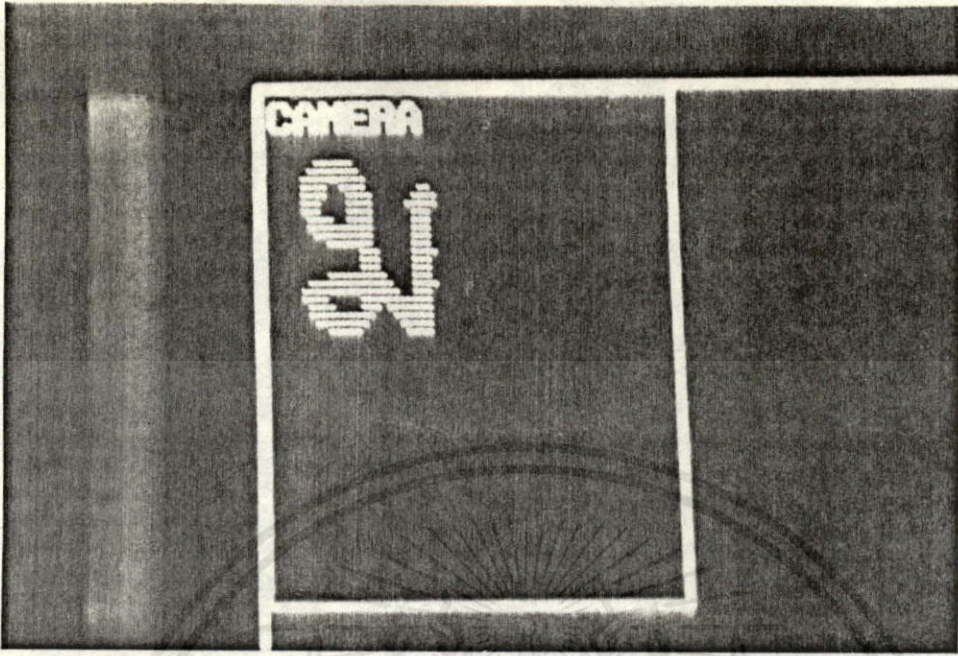
(ข)

รูปที่ 5.2 รูปแบบของตัวอักษรที่ได้จากกล้อง CCD

เอกสารนี้เป็นเอกสารของงานวิจัยที่ได้รับทุนวิจัยจากสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามทำซ้ำ (ก) การถ่ายภาพของตัวอักษร จากเอกสารทุกครั้งที่มีการนำไปใช้

(ข) รูปแบบของตัวอักษรขนาด 64 X 64 จุดภาพ



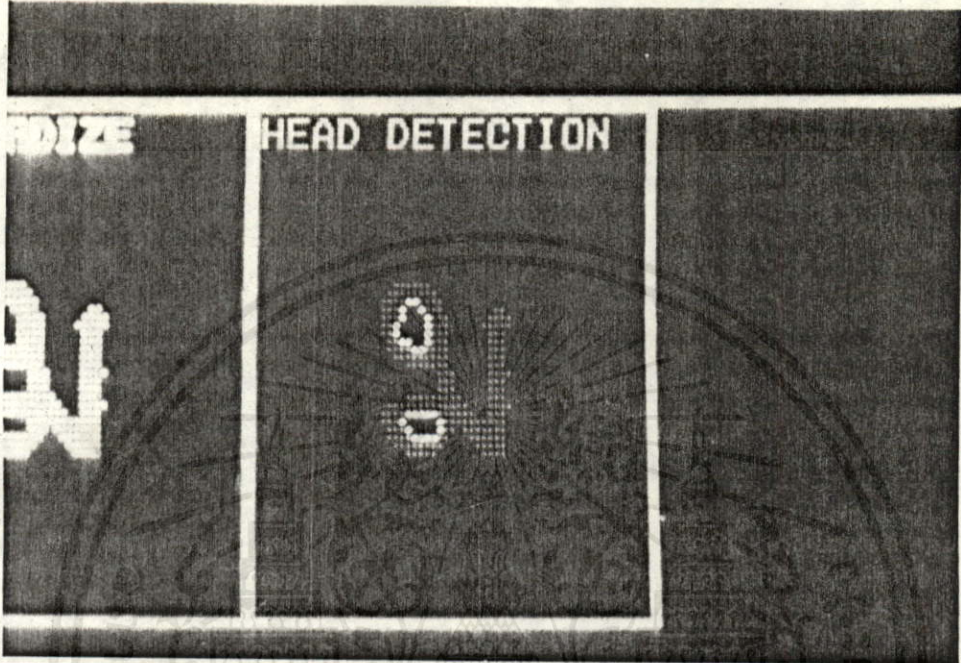
รูปที่ 5.3 รูปแบบสองระดับของตัวอักษร ขนาด 64 X 64 จุดภาพ

รูปแบบของตัวอักษรในรูปที่ 5.3 เป็นรูปแบบที่ยังไม่มี ขนาด และตำแหน่งที่มาตรฐาน ดังนั้นจึงทำให้เป็นรูปแบบที่มาตรฐานก่อน ดังแสดงในรูปที่ 5.4

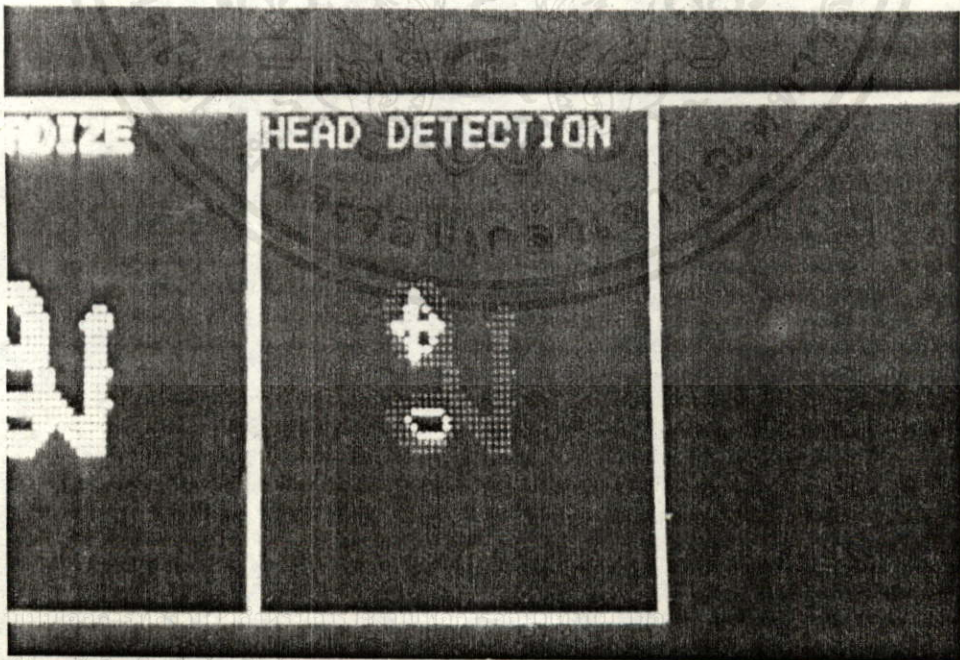


รูปที่ 5.4 รูปแบบที่มาตรฐาน

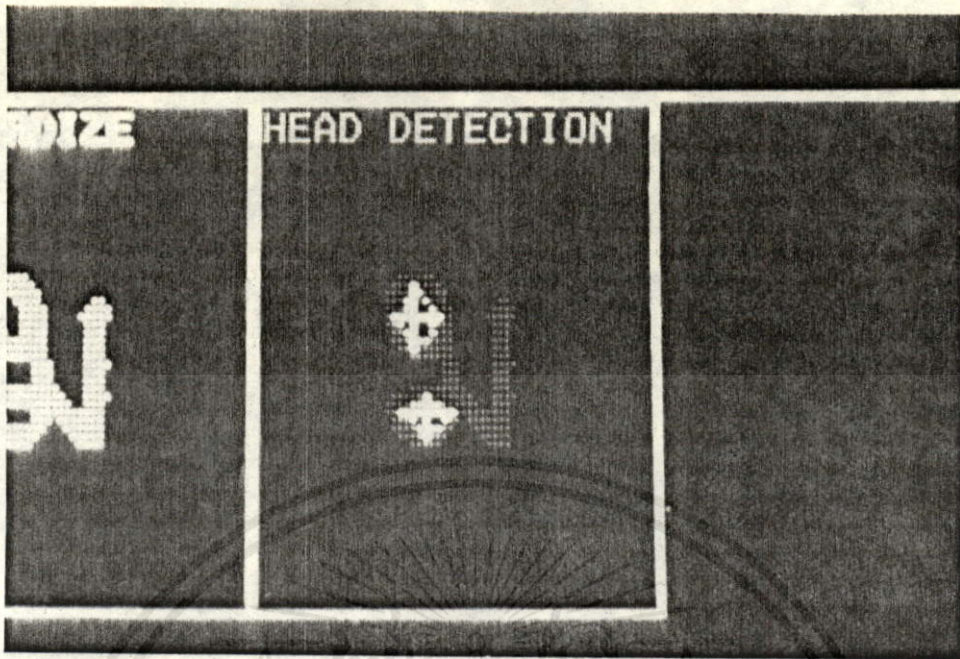
รูปแบบของตัวอักษรจะถูกลอกแบบออกมาเป็น 2 ส่วนเหมือน ๆ กัน โดยรูปแบบส่วนแรก จะนำไปทำการตรวจสอบหาจำนวนและตำแหน่งหัวของตัวอักษรในรูปแบบ ด้วยวิธีการที่ได้เสนอมานี้ ผลที่ได้จากการตรวจสอบหัว แสดงในรูปที่ 5.5 (ก), (ข) และ (ค)



(ก)



(ข)



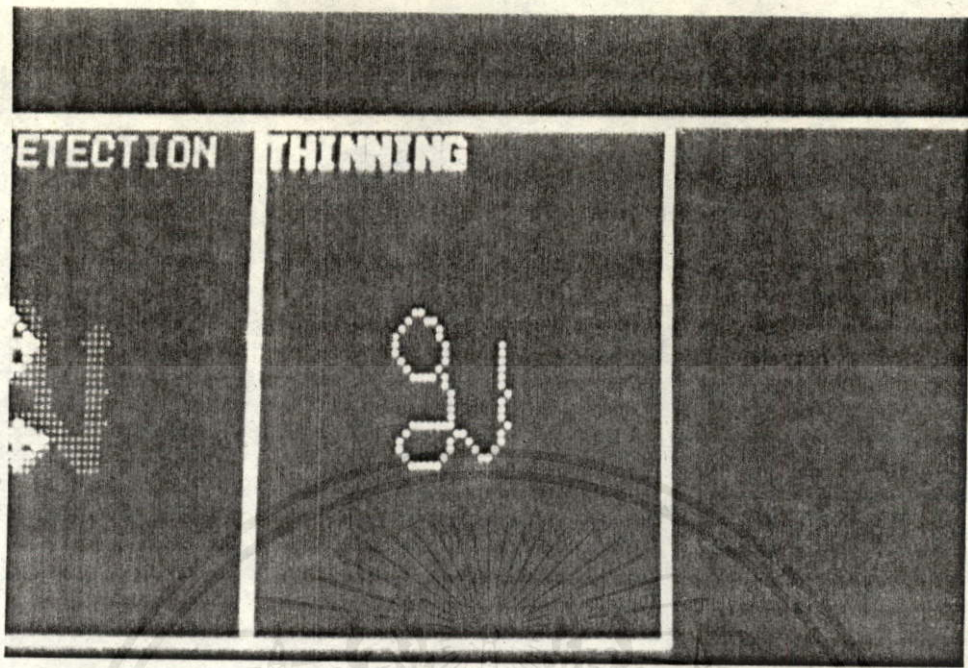
(ค)

รูปที่ 5.5 ผลที่ได้จากการตรวจสอบหัว

(ก) การตรวจสอบหาจำนวนหัว

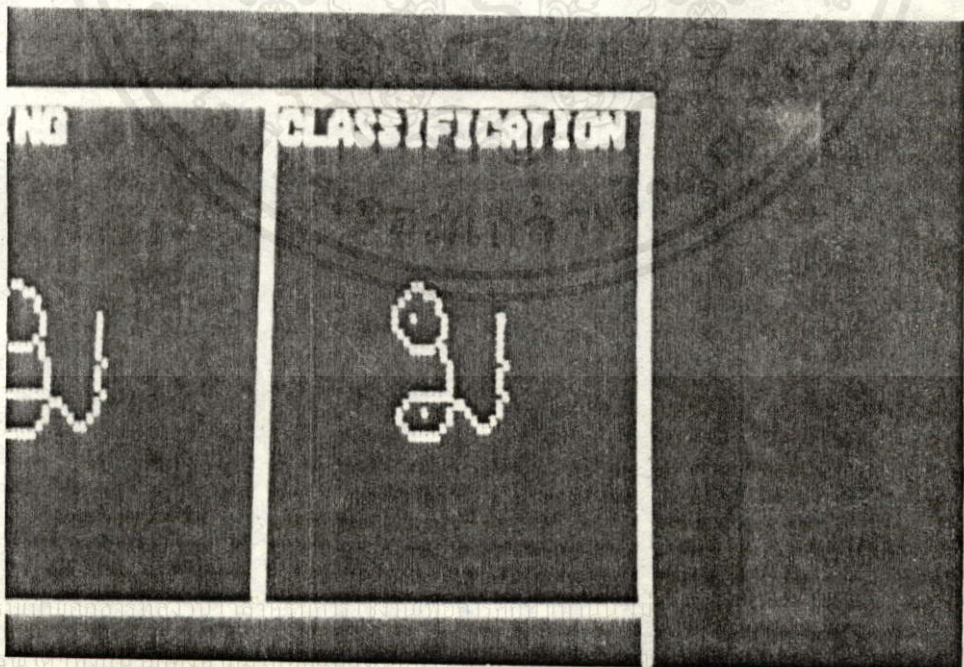
(ข), (ค) การหาตำแหน่งของหัวแต่ละหัว

ส่วนอีกรูปแบบหนึ่งก็จะนำไปผ่านระบบการจัดการล่วงหน้า (Pre-processing) ซึ่งจะประกอบด้วยวิธีการต่างๆคือ การทำตัวอักษรให้บาง การกำจัดส่วนเกิน และการหาคุณสมบัติทางโทโมโลยีของโครงร่าง ทำให้ได้ผลออกมาเป็นโครงร่างที่สมบูรณ์แบบของตัวอักษรดังแสดงในรูปที่ 5.6

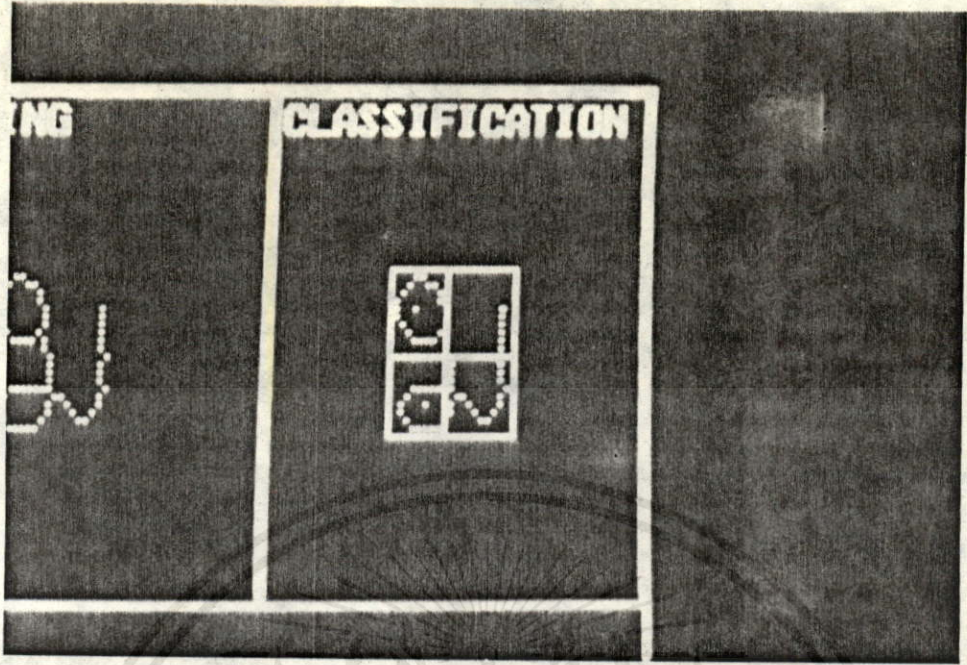


รูปที่ 5.6 ผลที่ได้จากการเตรียมข้อมูลตัวอักษร

เมื่อรวมเอาารูปแบบในรูปที่ 5.6 กับตำแหน่งของหัวที่ได้จากรูปที่ 5.5 เข้าด้วยกันจะได้รูปแบบของตัวอักษรที่พร้อมจะนำไปจำแนกกลุ่ม ดังรูปที่ 5.7 รูปแบบของตัวอักษรที่พร้อมจะนำไปจำแนกกลุ่มนี้ ประกอบด้วยจุดภาพที่เป็น จุดต่อเนื่อง จุดปลาย และจุดหัว



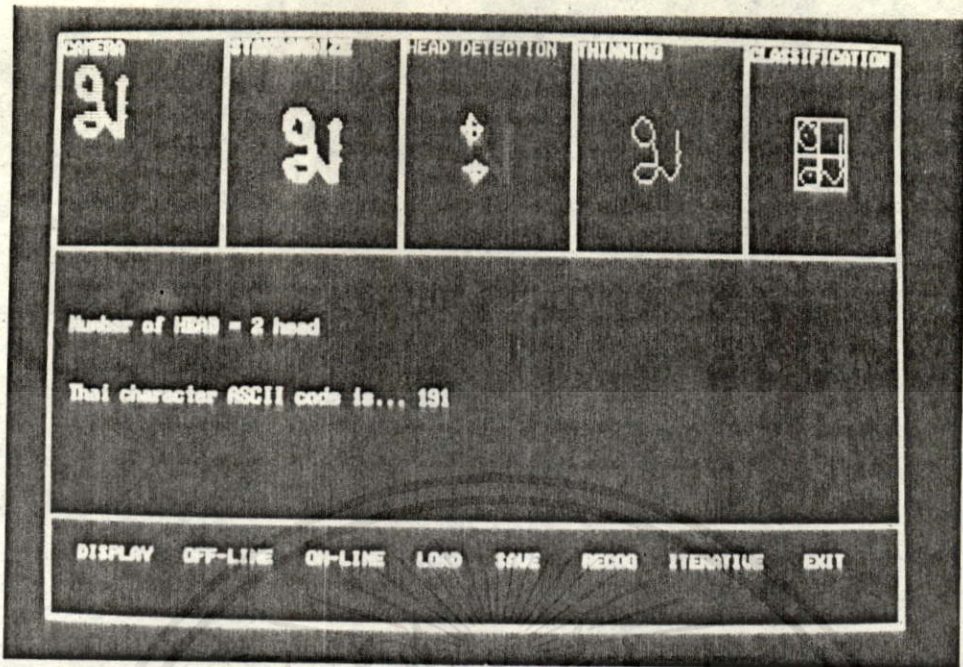
รูปที่ 5.7 ผลจากการรวมรูปแบบ



รูปที่ 5.8 ผลจากการแบ่งเขตของตัวอักษร

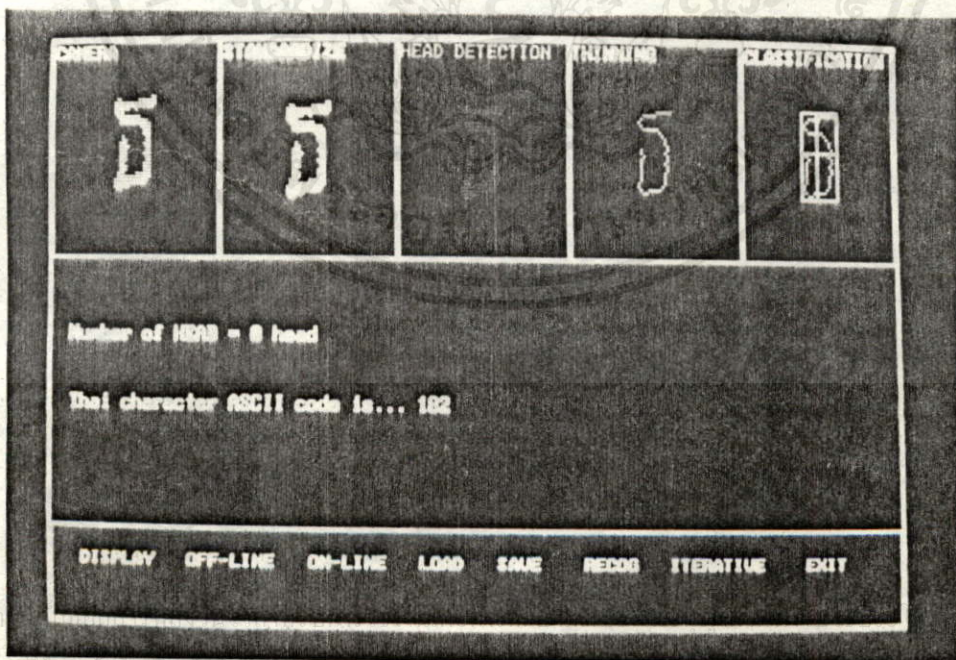
รูปที่ 5.8 เป็นการจำแนกกลุ่มของตัวอักษร โดยพิจารณาจำนวนและตำแหน่งหัวของตัวอักษรในแต่ละเขตที่แบ่ง และการจำแนกกลุ่มย่อย โดยพิจารณาตำแหน่งของจุดปลายในแต่ละเขต

ในขั้นตอนสุดท้าย ตัวอักษรก็จะถูกแยกแยะออกมาโดยการพิจารณารายละเอียดต่างๆ ตามโครงร่างแบบต้นไม้ที่เสนอไว้ในตอนท้ายของบทที่ 4 ผลที่ได้จากการจดจำตัวอักษร จะแสดงออกมาเป็นรหัสแอสกี (ASCII Code) ของตัวอักษรตัวนั้น ดังรูปที่ 5.9



รูปที่ 5.9 ผลการจดจำตัวอักษร "ม"

ตัวอักษร "ธ" เป็นตัวอักษรในกลุ่มที่ 1 (Class1) คือตัวอักษรที่ไม่มีหัว (No head) ดัง
ผลการทดลองในรูปที่ 5.10

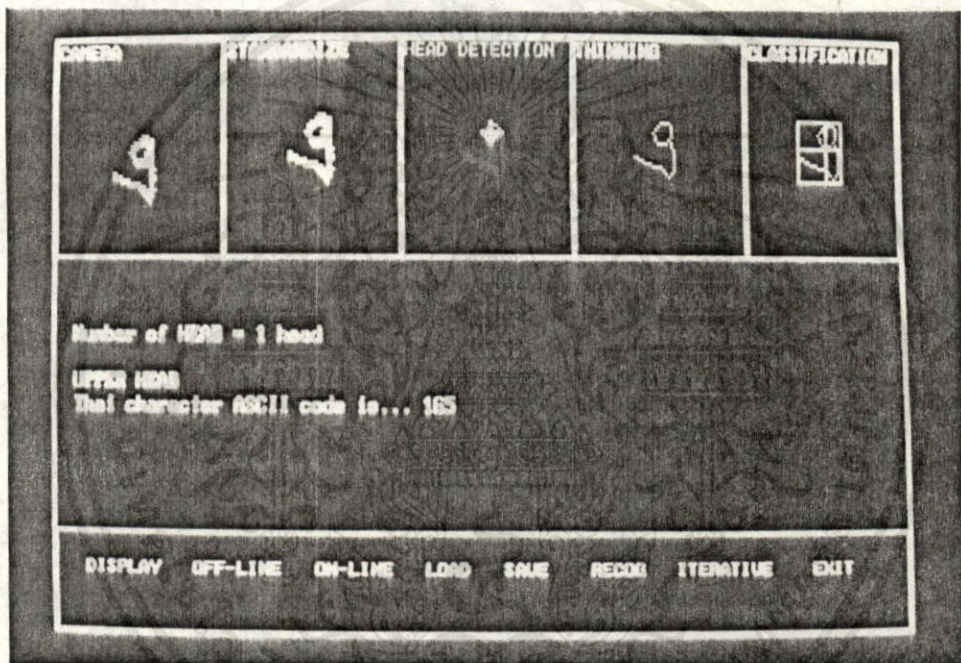


รูปที่ 5.10 ผลการจดจำตัวอักษร "ธ"

ตัวอักษร "ง" เป็นตัวอักษรในกลุ่มที่ 2 (Class2) คือตัวอักษรที่มีหัว จำนวน 1 หัว (One head) และหัวอยู่ในระดับบน (Upper head) ดังผลการทดลองในรูปที่ 5.11

ตัวอักษร "ด" เป็นตัวอักษรในกลุ่มที่ 2 (Class2) คือตัวอักษรที่มีหัว จำนวน 1 หัว (One head) และหัวอยู่ในระดับกลาง (Middle head) ดังผลการทดลองในรูปที่ 5.12

ตัวอักษร "ถ" เป็นตัวอักษรในกลุ่มที่ 2 (Class2) คือตัวอักษรที่มีหัว จำนวน 1 หัว (One head) และหัวอยู่ในระดับล่าง (Lower head) ดังผลการทดลองในรูปที่ 5.13



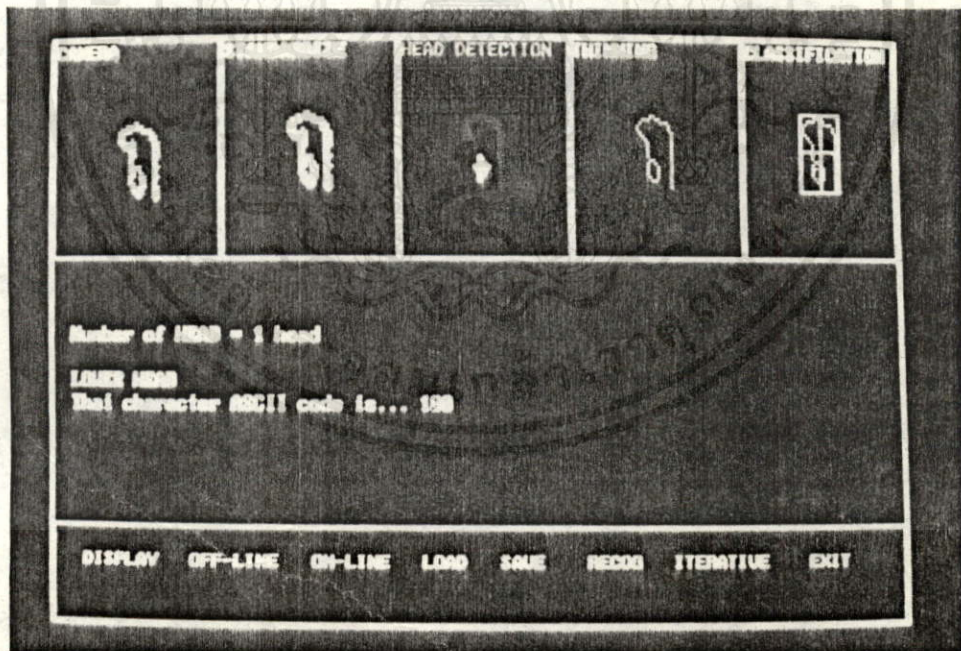
รูปที่ 5.11 ผลการจดจำตัวอักษร "ง"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

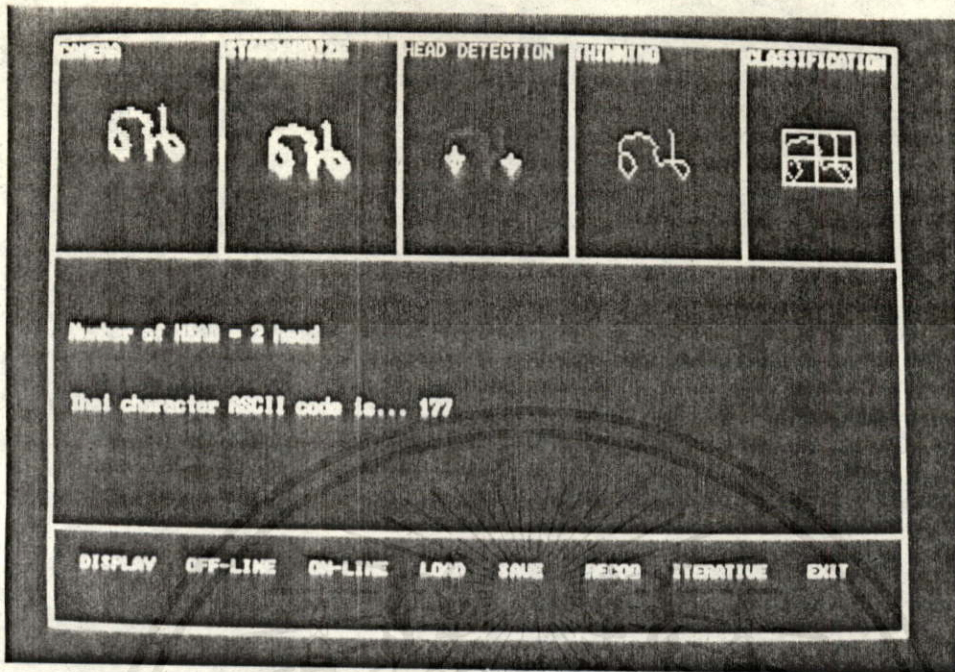


รูปที่ 5.12 ผลการจดจำตัวอักษร "ด"



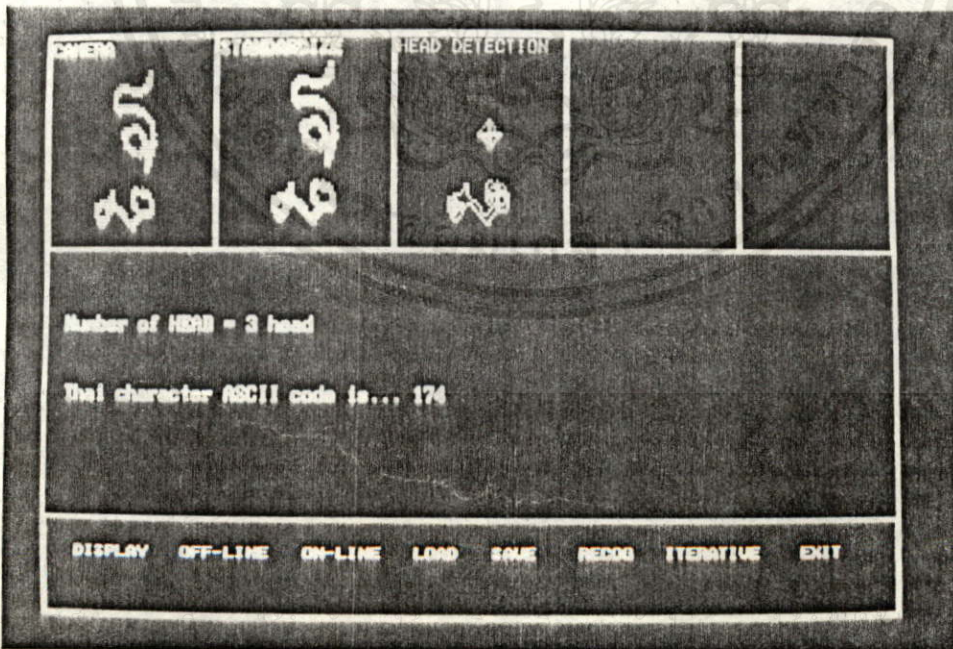
รูปที่ 5.13 ผลการจดจำตัวอักษร "ถ"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในการศึกษาเท่านั้น ไม่ควรเผยแพร่ไปให้ประชาชน
ไม่ว่ากรณี ตัวอักษร อีกรูปหนึ่ง "ถ" เป็นตัวอักษรในกลุ่มที่ 3 (Class3) คือตัวอักษรที่มีหัว จำนวน 2 หัว (Two head) ดังผลการทดลองในรูปที่ 5.14



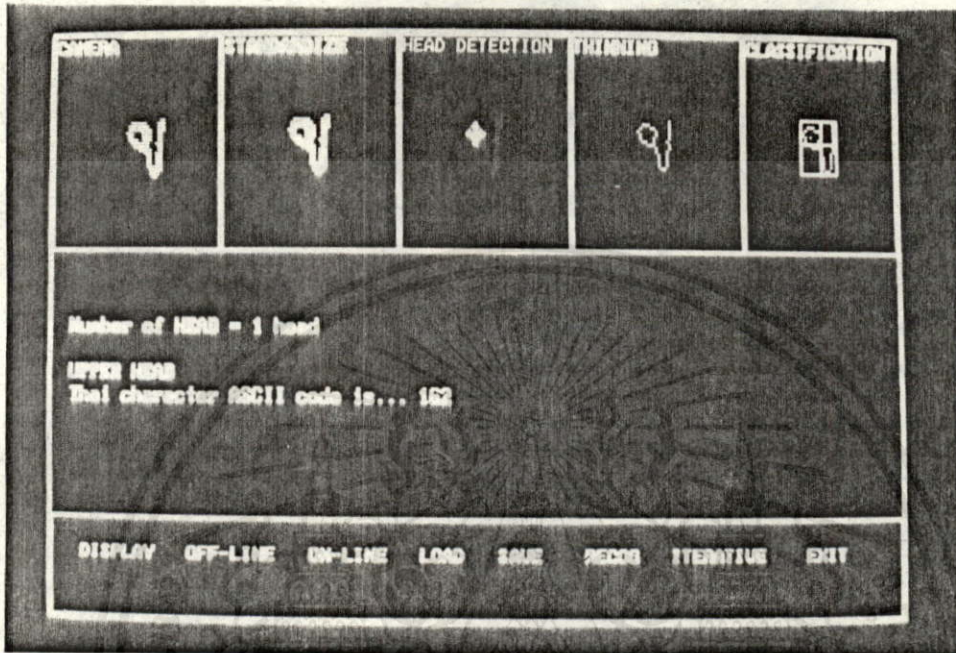
รูปที่ 5.14 ผลการจดจำตัวอักษร "ณ"

ตัวอักษร "ฐ" เป็นตัวอักษรในกลุ่มที่ 4 (Class4) คือตัวอักษรที่มีหัว จำนวน 3 หัว (Three head) ดังผลการทดลองในรูปที่ 5.15

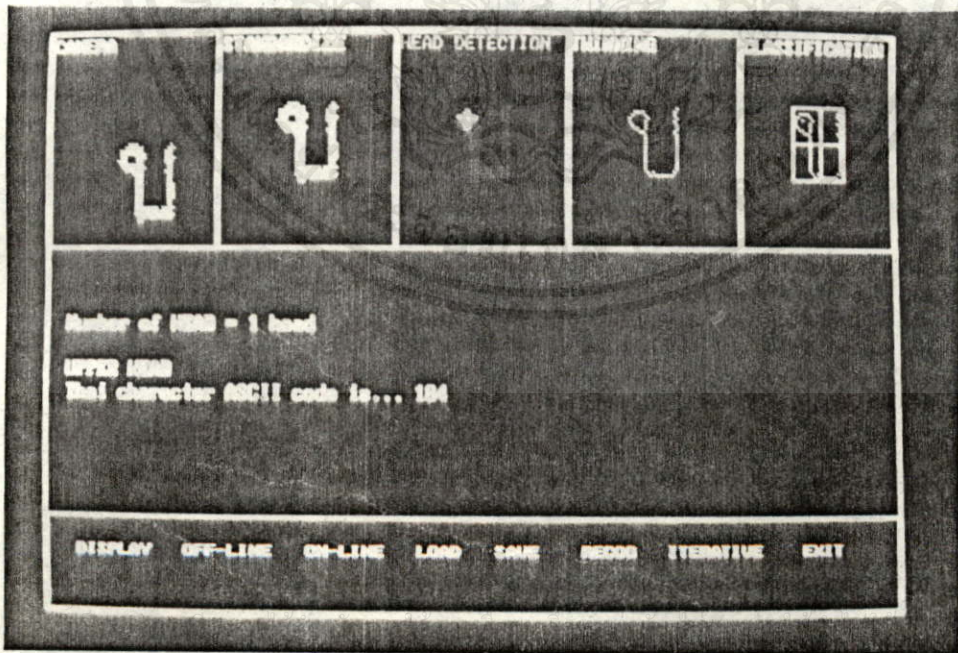


รูปที่ 5.15 ผลการจดจำตัวอักษร "ฐ"

ตัวอักษร "ข" และตัวอักษร "บ" จะใช้อัตราส่วนของ ความกว้างต่อความสูง ของตัวอักษร (WH Ratio) เป็นตัวจำแนก ถ้าความกว้างไม่เกิน $1/3$ ของความสูง ก็จะได้ผลของการจดจำออกมา เป็น ตัวอักษร "ข" ดังผลการทดลองในรูปที่ 5.16 และ 5.17



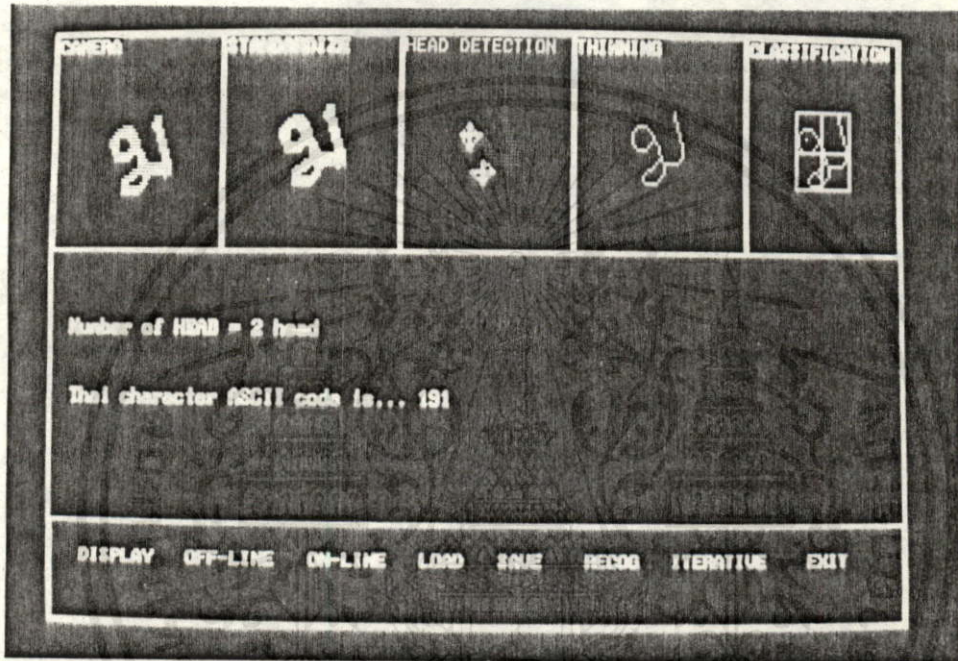
รูปที่ 5.16 ผลการจดจำตัวอักษร "ข"



รูปที่ 5.17 ผลการจดจำตัวอักษร "บ"

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งในการนำไปใช้

ในบางครั้ง ตัวอักษรที่ใช้ในการทดลอง จะมีลักษณะ เอนหน้าหรือ เอนหลัง ตามลักษณะการเขียนของผู้เขียนแต่ละคน ถ้าตัวอักษรมีการเอนมากๆ แล้ว จะทำให้ผลการจำแนกกลุ่มผิดพลาด แต่ถ้าตัวอักษรนั้นมีการเอน โดยที่ไม่ทำให้คุณสมบัติของหัว และคุณสมบัติทางโทโพโลยีของตัวอักษรเปลี่ยนแปลงไป แล้ว ผลการจดจำตัวอักษรก็จะถูกต้องเหมือนเดิม ดังแสดงในรูปที่ 5.18

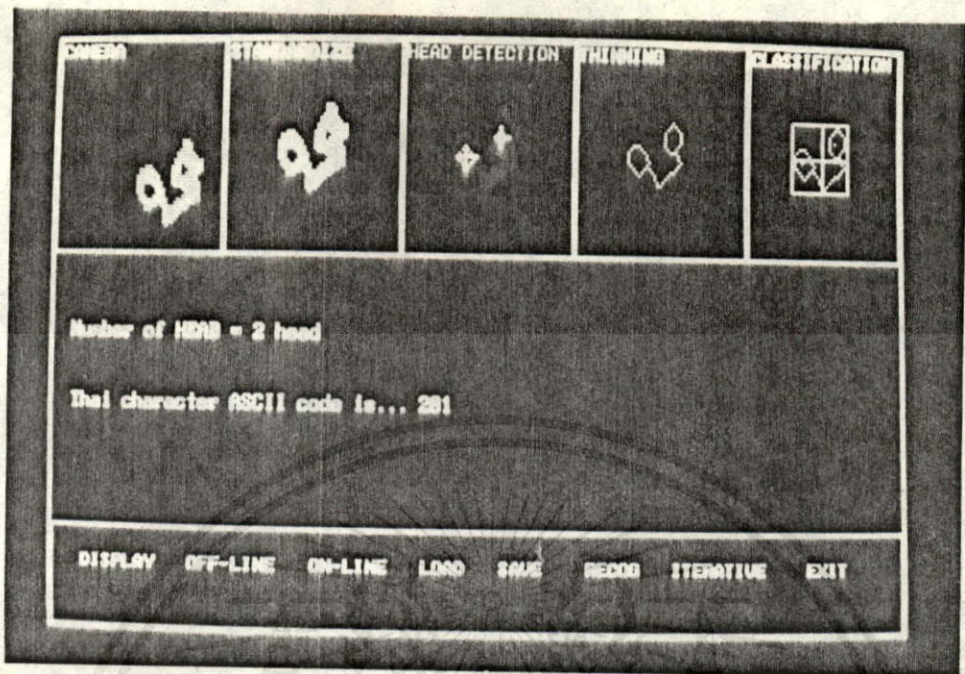


รูปที่ 5.18 ผลการจดจำตัวอักษรที่มีลักษณะเอน

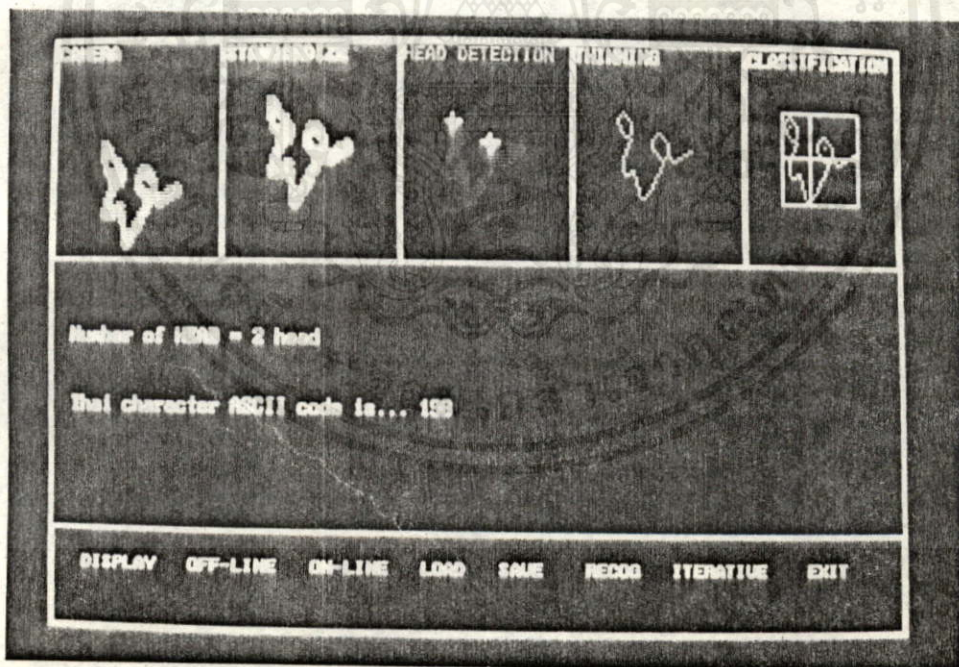
จากการทดลองจดจำตัวอักษร "ฮ" ที่มีลักษณะเอนไปทางด้านหน้ามากๆ จนทำให้คุณสมบัติต่างๆ เปลี่ยนไปจากเดิม พบว่าผลการทดลองจะเกิดการผิดพลาด โดยจะได้ผลการจดจำออกมาเป็นรหัสแอสกีของตัวอักษร "ฬ" ดังแสดงในรูปที่ 5.19

ในลักษณะเดียวกัน ตัวอักษร "ฬ" ที่มีลักษณะเอนไปทางด้านหลังมากๆ ก็พบว่าผลการทดลองจะเกิดการผิดพลาดขึ้นเช่นเดียวกัน โดยจะได้ผลการจดจำออกมาเป็นรหัสแอสกีของตัวอักษร "ฮ" ดังแสดงในรูปที่ 5.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.19 ผลการจดจำตัวอักษรที่มีลักษณะ เอน ไปทางด้านหน้ามากๆ

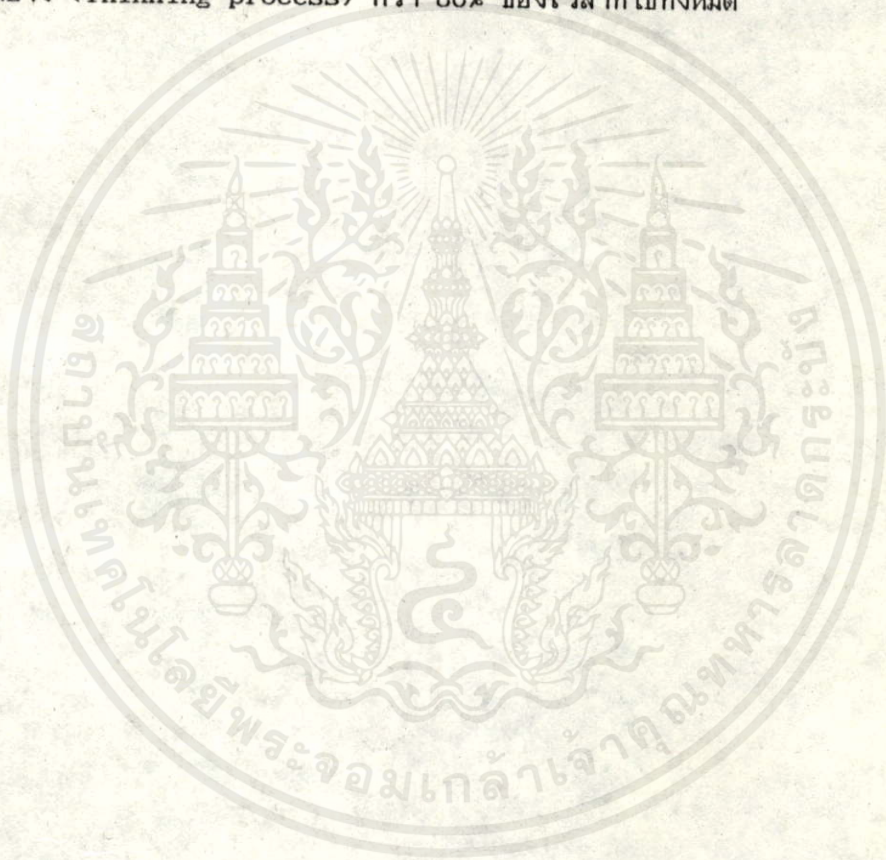


รูปที่ 5.20 ผลการจดจำตัวอักษรที่มีลักษณะ เอน ไปทางด้านหลังมากๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาควิชางานที่ออกวาทศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการศึกษา
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองจับเวลาในการประมวลผลทุกขั้นตอน ของการจดจำตัวอักษรแต่ละตัว จะพบว่า ตัวอักษร "ฐ" ใช้เวลาในการจดจำน้อยที่สุด คือ 2.3 วินาที สำหรับเครื่อง IBM PC/XT (ทำงานที่ 4.77 MHz) และ 1.1 วินาที สำหรับเครื่อง IBM PC/AT (ทำงานที่ 8 MHz)

ตัวอักษร "ฎ" ใช้เวลาในการจดจำมากที่สุด คือ 2.8 วินาที สำหรับเครื่อง IBM PC/XT และ 1.6 วินาที สำหรับเครื่อง IBM PC/AT สำหรับเวลาที่ใช้ในการประมวลผลนี้ จะใช้ไปในขั้นตอนของการทำให้บาง (Thinning process) กว่า 80% ของเวลาที่ใช้ทั้งหมด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุป

จากผลการทดลองจดจำตัวอักษรลายมือเขียนภาษาไทย จากผู้เขียน 15 คน จำนวน 960 ตัวอักษร พบว่าวิธีการจดจำตัวอักษรโดยการพิจารณาหัวของตัวอักษร ที่นำเสนอในวิทยานิพนธ์ฉบับนี้ ให้ผลในการจดจำตัวอักษรที่มีประสิทธิภาพสูง จากตัวอักษรทั้งหมด 960 ตัวอักษร สามารถจดจำตัวอักษรได้ 949 ตัวอักษร ผิดพลาด 11 ตัวอักษร ในจำนวน 11 ตัวอักษรนี้แบ่งเป็น ตัวอักษรที่จำแนกออกมาผิดกลุ่มจำนวน 7 ตัวอักษร และตัวอักษรที่จำแนกกลุ่มไม่ได้จำนวน 4 ตัวอักษร

สำหรับ เวลาที่ใช้ในการประมวลผลการจดจำตัวอักษรประมาณ 2.55 วินาที / ตัวอักษร เมื่อทำงานบนเครื่อง IBM PC/XT และ ประมาณ 1.35 วินาที / ตัวอักษร เมื่อทำงานบนเครื่อง IBM PC/AT

ความผิดพลาดที่เกิดขึ้นในการทดลองจดจำตัวอักษรนั้น เกิดขึ้นเพราะตัวอักษรมีลักษณะที่ผิดไปจากตัวอักษรโดยทั่วไป เช่น ตัวอักษรไม่มีหัว หรือ ตัวอักษรมีส่วนที่ติดกันจนทำให้เกิดเป็นวงปิดที่คล้ายกับหัวของตัวอักษร หรือ ตัวอักษรอยู่ในลักษณะที่เอียงไปทางด้านใดด้านหนึ่งมากเกินไป เป็นต้น

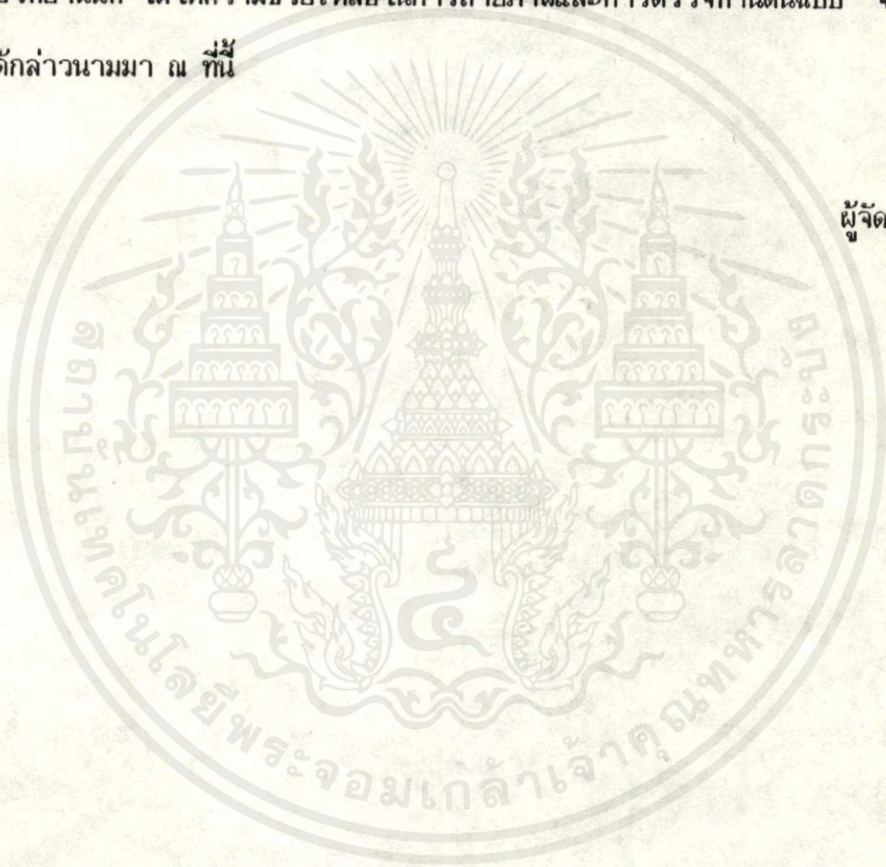
ดังนั้น วิธีการที่จะทำให้การจดจำตัวอักษรเกิดความผิดพลาดน้อยลง ก็จะต้องเขียนตัวอักษรให้บรรจง ไม่เขียนหวัดจนเกินไป และเนื่องจากเวลาในการจดจำตัวอักษรส่วนมากจะใช้เวลาในการทำให้บาง เพราะฉะนั้นตัวอักษรที่เขียนด้วยปากกาที่มีปลายขนาดใหญ่ จึงใช้เวลาในการจดจำนานกว่าตัวอักษรที่เขียนด้วยปากกาที่มีปลายขนาดเล็ก ด้วยสาเหตุดังกล่าว ถ้าได้มีการปรับปรุงวิธีการทำให้บาง ให้ทำงานได้เร็วขึ้นแล้ว การจดจำตัวอักษรก็จะเร็วขึ้น จนสามารถนำเอาวิธีการจดจำตัวอักษรที่ได้นำเสนอในวิทยานิพนธ์ฉบับนี้ ไปประยุกต์ใช้งานได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี ก็เนื่องด้วยการอนุเคราะห์ให้คำปรึกษาที่ดีจาก รองศาสตราจารย์ มนัส สังวรศิลป์ และ รองศาสตราจารย์ ดร.ชม กัมปาน ตลอดจนอาจารย์ทุกท่านในแผนกวิชาอิเล็กทรอนิกส์ ที่ให้ความช่วยเหลือ และเป็นกำลังใจ กระทั่งคุณ เรืองชัย วีระพงษ์ และ คุณ ชานูชัย นิลิทธิวิทยานนท์ ได้ให้ความช่วยเหลือในการถ่ายภาพและการตรวจทานต้นฉบับ จึงใคร่ขอขอบพระคุณท่านผู้ได้กล่าวนามมา ณ ที่นี้

ผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. FRED W.M. STENTIFORD.: "Automatic Feature for Optical Character Recognition Using an Evolutionary Search Procedure,"IEEE Trans.Pattern Anal. Mach. Intell, vol.PAMI-7,pp.349-355,May 1985.
2. KIMPAN,C. : "Printed Thai character recognition using topological properties method " .INT.J.ELECTRONICS, vol.60, no.3, pp.303-329, 1986.
(Invited paper)
3. KIMPAN,C., ITOH,A.,and KAWANISHI,K.: "Fine classification of printed Thai characters recognition using Karhunen-Loeve expansion" .IEE PROCEEDINGS ,vol.134,Pt.E,no.5,pp.257-264,September 1987.
4. KIMPAN,C.: "Printed Thai Characters Recognition", Dissertation of Doctor of Engineering, King Mongkut's Institute of Technology Chaokhun Taharn Ladkrabang, 1986.
5. HIRANVANICHAKORN,P., AGUI,T., and NAKAJIMA,M.: "A Recognition Method of Thai Characters". The Transactions of the IECE of Japan, vol.E65, no.12, December 1982.
6. HIRANVANICHAKORN,P. , AGUI,T. , and NAKAJIMA,M.: " Recognition of Thai Characters by using local features". The Transactions of the IECE of Japan, vol.E67, no.8, August 1984.
7. YOKOI,S.,TORIWAKI,J.,and FUKUMURA,T."Topological properties in Digitized Binary Pictures", IECE , 56-D , No.11 (in Japanese).
8. STEFANELLI,R. and ROSENFELD,A.: "Some Parallel Thinning Algorithms for Digital Pictures," Journal of the Association for Computing Machinery,

- vol.18,no.2,pp.255-264,April 1971.
9. CARLO ARCELLI and GABRIELLA SANNITI DI BAJA.: "A Width-Independent Fast Thinning Algorithm," IEEE Trans. Pattern Anal. Mach. Intell, vol.PAMI-7, no.4,pp.463-474,July 1985.
 - 10.K.S. Fu, R.C. Gonzalez and C.S.G. Lee.: " Skeleton of a Region,"ROBOTICS : CONTROL, SENSING, VISION, AND INTELLIGENCE, McGRAW-HILL International Editions, pp.411-416,1987.
 - 11.PAVLIDIS, T.: "Shape Description by Region Analysis," Structural Pattern Recognition, Springer-Verlag Berlin Heidelberg,New York,pp.216-257,1977.
 - 12.RICHARD O. DUDA and PETER E. HART.: "Contour Following,"PATTERN CLASSIFICATION AND SCENE ANALYSIS, A Wiley-Interscience Publication, pp. 290-293,1973.
 - 13.ARNOLD L.KNOLL.: "Experiments with "Characteristic Loci" for Recognition of Handprinted Characters," IEEE Transactions on Computers, pp.366-372, April 1969.
 - 14.สุรพันธุ์ เอื้อไพฑูริย์, " การเตรียมข้อมูลสำหรับการจำแนกรูปแบบตัวอักษรภาษาไทย",บทความทางวิชาการ Seminar MI, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง กรุงเทพฯ ,พฤศจิกายน 2529
 - 15.สุรพันธุ์ เอื้อไพฑูริย์ , ดร.บุญวัฒน์ อัดชู และ รศ.มณัส สังวรศิลป์, "การจำแนกกลุ่มของตัวอักษรภาษาไทยโดยการพิจารณาหัวของตัวอักษร" ,บทความสำหรับการประชุมทางวิชาการ "การประยุกต์ใช้คอมพิวเตอร์ในทางวิศวกรรม", มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่, หน้า 293-304 ,กุมภาพันธ์ 2531
 - 16.H.F. Li and P.W.F. Poon.: "Real-time on-line pattern recognition of eye position and movement," .IEE PROCEEDINGS, vol.132,Pt.E, no.6,pp.135-154, November 1985.

17. SIMON KAHAN , THEO PAVLIDIS and HENRY S. BAIRD.:"On the Recognition of Printed Characters of Any Font and Size,"IEEE Trans.Pattern Anal. Mach. Intell, vol.PAMI-9,no.2,pp.274-288,March 1987.
18. HUSSEIN ALMUALLIM and SHOICHIRO YAMAGUCHI.:" A Method of Recognition of Arabic Cursive Handwriting,"IEEE Trans.Pattern Anal. Mach. Intell, vol. PAMI-9,no.5,pp.715-722,September 1987.
19. ROGER W. EHRICH and KENNETH J. KOEHLER.:"Experiments in the Contextual Recognition of Cursive Script,"IEEE Transactions on Computers,pp.182-194 ,Febuary 1975.
20. ประสาร ตั้งติสานนท์, " การจดจำรูปแบบตัวอักษรคัดลายมือภาษาไทย โดยวิธีแยกลักษณะเด่น ", วิทยานิพนธ์สำหรับ ปริญญาวิศวกรรมศาสตรมหาบัณฑิต, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า ลาดกระบัง , 2529
21. F. Cheevasuvit.:"Edge Detection whit Auto-adapti6ve Threshold"ARTIFICIAL INTELLIGENCE , King Mongkut's Institute of Technology Chaokhun Taharn Ladkrabang,1987.
22. ช่างชัย พิสิทธิวิทยานนท์, "การติดตามขอบวัตถุด้วยการใช้ตารางหน้าต่าง", บทความทางวิชาการ Seminar MI, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง, พฤศจิกายน 2529
23. ครรชิต ไมตรี, "การกำจัดส่วนเกินของตัวอักษรลายมือเขียน", วิศวสาร ลาดกระบัง, ปีที่ 2 ฉบับที่ 1 , มีนาคม 2522

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครู ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลชุดที่ <u>1</u>				ชื่อ <u>สุรพันธ์</u>			
ก	ข	ค	ด	จ	ฉ	ช	ซ
ฅ	ฉ	ฐ	ฎ	ฏ	ฐ	ฑ	ฒ
ณ	ด	ต	ถ	ท	ธ	ฒ	ณ
ป	ย	ฝ	พ	ฟ	ภ	ม	ย
ธ	ด	ว	ล	ฬ	ล	ฬ	ฬ
อ	ฮ	อ	ฮ	า	า	า	อ
ใ	อ	อ	อ	อ	อ	อ	+
อ	อ	อ	อ	อ	อ	อ	อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

ข้อมูลชุดที่ <u>2</u>				ชื่อ <u>ยพชษา</u>			
ก	ข	ค	ง	จ	ฉ	ช	ซ
ฅ	ฉ	ช	ซ	ฅ	ฉ	ช	ซ
ด	ต	ถ	ด	ต	ถ	ด	ต
ป	ผ	ฝ	พ	ฟ	ภ	ม	ย
ร	ล	ว	ศ	ษ	ส	ห	ฬ
อ	ฮ	อ	๕	๖	๗	๘	๙
๑	๒	๓	๔	๕	๖	๗	๘
๙	๐	๑	๒	๓	๔	๕	๖

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอมูลชุดที่ <u>3</u>				ชื่อ <u>เริงชัย</u>			
ก	ข	ค	ง	จ	ฉ	ช	ซ
ฅ	ฉ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ
ณ	ด	ต	ถ	ท	ธ	ฒ	บ
ป	ผ	ฝ	พ	ฟ	ภ	ม	ย
ร	ล	ว	ศ	ษ	ส	ห	ฬ
อ	ฮ	ฤ	ฌ	า	ก	ข	ฃ
ฅ	๑	๒	๓	๔	๕	๖	๗
๘	๙	๐	๑	๒	๓	๔	๕

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลชุดที่ <u>4</u>				ชื่อ <u>นักบรรณารักษ์ วิทยาลัยศรีนครินทร์</u>			
ก	ข	ค	ง	จ	ฉ	ช	ฌ
ฆ	ฌ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ
ณ	ด	ต	ถ	ท	ธ	ฒ	บ
ป	ผ	ฝ	พ	ฟ	ภ	ม	ย
ร	ล	ว	ล	ช	ส	ห	ฬ
อ	ฮ	อ	ฮ	า	า	า	า
๗	๑	๒	๓	๔	๕	๖	๗
๘	๘	๙	๐	๑	๒	๓	๔

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลชุดที่ <u>5</u>				ชื่อ <u>มหาวิทยาลัย ทรัพย์สินทางปัญญา</u>			
๑	๒	๓	๔	๕	๖	๗	๘
๙	๑๐	๑๑	๑๒	๑๓	๑๔	๑๕	๑๖
๑๗	๑๘	๑๙	๒๐	๒๑	๒๒	๒๓	๒๔
๒๕	๒๖	๒๗	๒๘	๒๙	๓๐	๓๑	๓๒
๓๓	๓๔	๓๕	๓๖	๓๗	๓๘	๓๙	๔๐
๔๑	๔๒	๔๓	๔๔	๔๕	๔๖	๔๗	๔๘
๔๙	๕๐	๕๑	๕๒	๕๓	๕๔	๕๕	๕๖
๕๗	๕๘	๕๙	๖๐	๖๑	๖๒	๖๓	๖๔
๖๕	๖๖	๖๗	๖๘	๖๙	๗๐	๗๑	๗๒

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอมูลชุดที่ <u> ๘ </u>				ชื่อ <u> ไฉนุญะ </u>			
ก	ข	ฅ	ค	๖๖	๗	๘	๑
ฉ	ช	ซ	ฅ	ญ	ฎ	ฏ	ฐ
ท	ฒ	ณ	ด	ต	ถ	ฑ	ฒ
ฬ	บ	ป	พ	ฝ	ฟ	ภ	ภา
ม	ย	ร	ล	ว	ศ	ษ	ส
ห	ฬ	อ	ฮ	ฤ	ฌ	โ	ใ
เ	๒	๓	๔	๕	๖	๗	๘
๙	๐	๑	๒	๓	๔	๕	๖

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลชุดที่ <u>8</u>				ชื่อ <u>ดุลขวัจน์</u> <u>จอมแสง</u>			
ก	ข	ค	ง	จ	ฉ	ช	ซ
ฅ	ฉ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ
ณ	ด	ต	ถ	ท	ธ	ฒ	บ
ป	พ	ฟ	ภ	ผ	ภ	ม	ย
ร	ล	ว	ศ	ษ	ส	ห	ฬ
อ	ฮ	ฤ	ุ	า	ำ	๑	๒
๓	๔	๕	๖	๗	๘	๙	๐
๑	๒	๓	๔	๕	๖	๗	๘
๙	๐	๑	๒	๓	๔	๕	๖

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำมาไปใช้

ข้อมูลชุดที่ <u>9</u>				ชื่อ <u>นางสาว</u>			
ก	ข	ค	ด	ฉ	ช	ฉ	ช
ซ	ฅ	ฌ	ญ	ฎ	ฏ	ฐ	ฑ
ฑ	ฒ	ณ	ด	ด	ด	ด	ด
น	บ	ป	พ	ผ	ฝ	ฝ	ภ
ม	ย	ร	ล	ว	ด	ช	ค
ร	ช	อ	ฮ	อ	อ	อ	อ
ค	ข	ค	ก	ก	ก	ก	ก
ก	ข	ค	ด	ด	ด	ด	ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอมูลชุดที่ 10				ชื่อ กาญจนะ			
ก	ข	ค	ง	จ	ฉ	ช	ซ
ฌ	ฉ	ฉ	ฌ	ฌ	ฌ	ฌ	ฌ
ฉ	ค	ค	ค	ค	ค	ค	ค
ข	ข	ข	ข	ข	ข	ข	ข
จ	จ	จ	จ	จ	จ	จ	จ
ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ	ฉ
ช	ช	ช	ช	ช	ช	ช	ช
ซ	ซ	ซ	ซ	ซ	ซ	ซ	ซ
ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ
ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ	ฌ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอมูลชุดที่ _____ //				ชื่อ _____			
ก	ข	ค	ง	ฉ	ช	ซ	ฅ
ฉ	ฉ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ
ณ	ด	ต	ถ	ท	ธ	ฒ	ป
ป	ฝ	ฝ	พ	พ	ภ	ม	ย
ร	ล	ล	ด	ช	ส	ห	ฬ
อ	ฮ	ภ	อ	า	ค	น	น
จ	จ	ข	เ	เ	ช	ฉ	ท
ฉ	เ	ย	อ	ก	๖๖	๐	๑

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลชุดที่ <u>12</u>				ชื่อ <u>กฤษฎิ์</u>			
ก	ข	ค	ง	จ	ฉ	ช	ซ
ฃ	ฅ	ง	ฉ	ซ	ฌ	ฎ	ฏ
ฑ	ณ	ต	ท	น	ป	ผ	ฝ
ฟ	ภ	ม	ร	ฤ	ล	ฦ	ว
ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ
ะ	ั	า	ำ	ิ	ี	ึ	ื
ุ	ู	ฺ	฻	฼	฾	฿	เ
แ	โ	ใ	ไ	ๅ	ๆ	็	่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลชุดที่ <u>13</u>				ชื่อ <u>กาญจนา</u>			
ก	ข	ค	ง	จ	ฉ	ช	ซ
ฅ	ฌ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ
ณ	ด	ต	ถ	ท	ธ	น	บ
ป	ผ	ฝ	พ	ฟ	ภ	ม	ย
ร	ล	ว	ศ	ษ	ส	ห	ฬ
อ	ฮ	อ	ฮ	า	ะ	ั	ุ
เ	แ	อ	บ	เ	อ	ผ	ท
ส	ศ	ย	อ	โ	บ	อ	อ

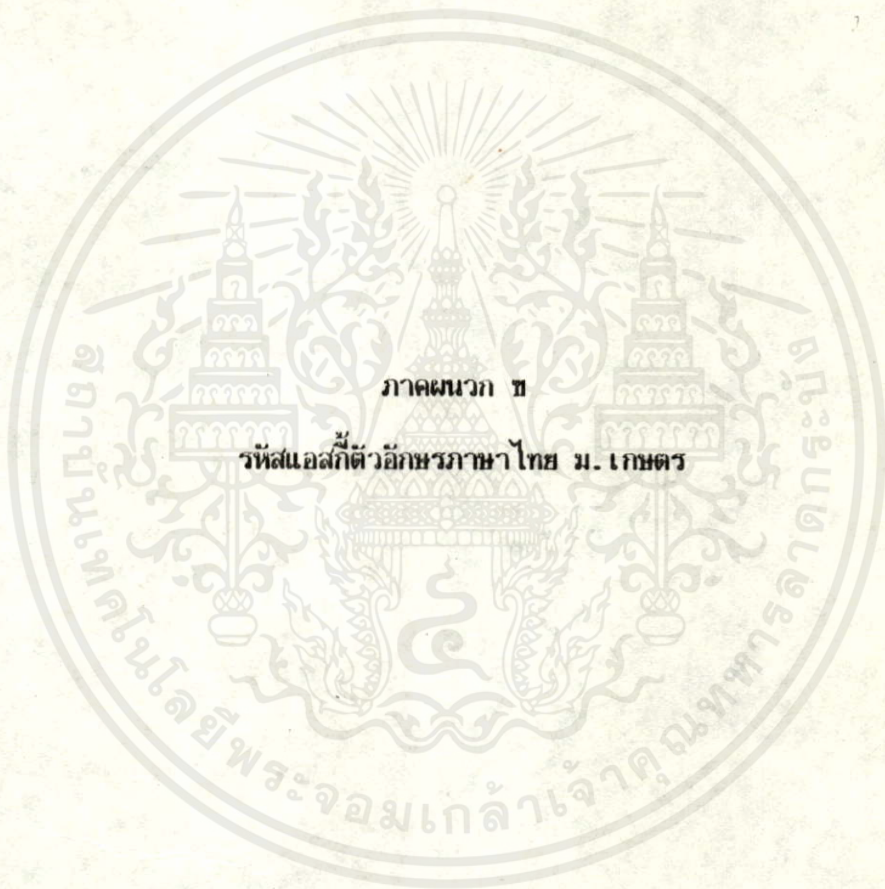
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลชุดที่ <u>14</u>				ชื่อ <u>วราพร</u>			
ก	ข	ช	ด	ต	น	ง	จ
ค	ช	ซ	ฎ	ฏ	ฐ	ฑ	ฒ
ณ	ด	ต	ฎ	ฏ	ฐ	ฑ	ฒ
ฬ	ณ	ด	ฎ	ฏ	ฐ	ฑ	ฒ
ร	ล	ว	ศ	ษ	ห	ฬ	ฬ
อ	ฮ	ก	ข	ค	ด	ด	ด
เ	อ	ข	ค	ด	ด	ด	ด
ส	ซ	ซ	ด	ด	ด	ด	ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลชุดที่ <u>15</u>				ชื่อ <u>ค.ศ. ค.บ.ค.</u>			
ก	ข	ค	ง	จ	ฉ	ช	ซ
ฅ	ฌ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ
ณ	ด	ต	ถ	ท	ธ	น	บ
ป	ผ	ฝ	พ	ฟ	ภ	ม	ย
ร	ล	ว	ศ	ษ	ค	ห	ฬ
อ	ฮ	ก	ข	ก	ค	น	ว
ฌ	ง	บ	ง	า	ง	ฬ	+
จ	จ	ง	จ	ก	ค	ค	ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of \language\turboc\include\smr.h

Page 1

8/14/1988 17:09:51

```

1:
2: /* Header for thai character recognition program */
3: /* by Suraphun Airphaiboon KMITL      3/4/1988 */
4: #define  TIMERMODE  182
5: #define  FREQSCALE  (long) 1190000
6: #define  TIMESCALE  (long) 1230
7: #define  T_MODEPORT 67
8: #define  FREQPORT   66
9: #define  BEEPPOINT  97
10: #define  ON          79
11: #define  STOP        0x0d
12: #define  TV_SEG      0x7000
13:
14: unsigned char  PTO[64][64] ; /* input,thin,redun */
15: unsigned char  PT1[64][64] ; /* head location */
16: unsigned char  PT2[64][64] ; /* end,branch,head */
17: unsigned char  PT3[64][64] ; /* redun buffer */
18: unsigned char  PT4[64][64] ; /* "-----" */
19: unsigned char  ID[10],JD[10] ; /* delete for redun */
20: unsigned char  IH[ 5],JH[ 5] ; /* keep head point */
21: unsigned char  IB[ 5],JB[ 5] ; /* keep branch ...' */
22: unsigned char  IE[ 5],JE[ 5] ; /* keep end .....' */
23: char          *G ; /* file name for load */
24: char          *H ; /* file name for save */
25: char          *R ; /* file name for recog */
26: char          buff[2048] ;
27: char          FF ;
28: int           HEAD ;
29: int           BRAN ;
30: int           VECTOR ;
31: int           END ;
32: int           REST ;
33: int           xm,ym ;
34: int           x1,x2,y1,y2,A,B ;
35: int           xa,xb,xc,ya,yb,yc ;
36: int           xa1,ya1,xb1,yb1,hx,hy ;
37: int           end1,con1,hed1 ;
38: int           end2,con2,hed2 ;
39: int           end3,con3,hed3 ;
40: int           end4,con4,hed4 ;
41: int           end5,con5,hed5 ;
42: int           end6,con6,hed6 ;
43: int           end7,con7,hed7 ;
44: int           end8,con8,hed8 ;
45: int           end9,con9,hed9 ;
46: int           count1,count2,count3,count7 ;
47:
48:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of listhe.c

Page 1

8/14/1988 17:18:28

```

1: /* *****
2: *          RECOGNITION THAI CHARACTER          *
3: *          file name PATT.C run on EGA monitor  *
4: *          by Mr.JO                             *
5: *          ***** */
6: #include <stdio.h>
7: #include <math.h>
8: #include <dos.h>
9: #include <conio.h>
10: #include <graphics.h>
11: #include <smr.h> /* Manus,Suraphun,Rerngchai Algorithm HEADER */
12: int GraphDriver,GraphMode;
13:
14: main ()
15: {
16:     char c ;
17:     int i,flag=0 ;
18:     GraphDriver = DETECT; /* Request auto-detection
19:     initgraph( &GraphDriver, &GraphMode, "" );
20:     setcolor(10);outtextxy(210,132,"THAI CHARACTER RECOGNITION");
21:     for(i=0 ; i<9 ;++i) {
22:         delay(180);
23:     }
24:     clearviewport();
25:     seffect ();
26:     setbkcolor (0);
27:     setcolor (10);
28:     outtextxy (25,307,"DISPLAY OFF-LINE ON-LINE LOAD") ;
29:     setcolor (10);
30:     outtextxy (330,307,"SAVE RECOG ITERATIVE EXIT") ;
31:     setcolor (15) ;
32:     line(0,0,639,0);line(639,0,639,349);
33:     line(639,349,0,349);line(0,349,0,0);
34:     setcolor(15);
35:     line(638,348,1,348);line(1,348,1,290);
36:     line(1,290,638,290);line(638,290,638,348);
37:     do
38:     {
39:         c = getch () ;
40:         switch (c)
41:         {
42:             case 'd':
43:                 flag = 0 ;display () ;getch () ;tone (1800,5) ;
44:                 break ;
45:             case 'r':
46:                 flag = 0 ;clscrn(); recog () ; tone (1500,5) ;
47:                 break ;
48:             case 'o':
49:                 flag = 0 ; autom () ; tone (1500,5) ;
50:                 break ;

```

Listing of listhe.c

Page 2

8/14/1988 17:18:28

```

51:         case 'l':
52:             flag = 0 ; load ( ) ;      tone (1300,5) ;
53:             break ;
54:         case 's':
55:             flag = 0 ; save ( ) ;      tone (1400,5) ;
56:             break ;
57:         case 'i':
58:             if (flag == 0)
59:             {
60:                 gotoXY (3,24) ;
61:                 printf (" Please wait. I'm recieveing image ") ;
62:                 rs_232 ( ) ;tone (675,3) ;
63:                 savpat ( ) ;display ( ) ;tone (1000,5) ;
64:                 gotoXY (3,24) ;
65:                 printf ("hit any key to return...           ") ;
66:                 getch ( ) ;
67:             }
68:             break ;
69:         case 'e':
70:             break ;
71:         default :
72:             tone (440,-30) ;
73:     }
74: } while ( c != 'e') ;
75: closegraph();
76: }
77: clrscrn()
78: {
79:     setviewport(2,2,127,127,1);
80:     clearviewport();
81:     setviewport(130,2,256,127,1);
82:     clearviewport();
83:     setviewport(259,2,384,127,1);
84:     clearviewport();
85:     setviewport(387,2,515,127,1);
86:     clearviewport();
87:     setviewport(518,2,637,127,1);
88:     clearviewport();
89:     setviewport(1,130,350,256,1);
90:     clearviewport();
91:     setviewport(0,0,639,349,1);
92: }
93: recog ( )
94: {
95:     int i,bk=0 ;
96:     clear_patt ( ) ;
97:     setcolor(11);rectangle(1,1,128,128);
98:     load ( ) ;dpy ( ) ;
99:     setcolor(13);outtextxy (4,4,"CAMERA") ;
100: getch ( ) ;

```

ไม่ว่ากรณีใดๆก็ตาม อีลทั้งหมดมิให้คัดแบบสงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of listhe.c

Page 3

8/14/1988 17:18:28

```

101:          setcolor(11);rectangle(129,1,257,128);
102:   move () ;size () ;display1 () ;
103:   setcolor(11);outtextxy(132,4,"STANDARDIZE") ;
104:   getch () ;
105:          setcolor(11);rectangle(258,1,385,128);
106:   setcolor(12);outtextxy(261,4,"HEAD DETECTION");
107:   headc () ;
108:   if (HEAD>=3)
109:     { class_3 () ;bk=1 ; }
110:   if (bk==0)
111:     {
112:   getch () ;
113:          setcolor(11);rectangle(386,1,516,128);
114:   setcolor(10);outtextxy(388,4,"THINNING") ;
115:   thinn () ;redun () ;display3 () ;
116:   getch () ;
117:          setcolor(11);rectangle(517,1,638,128);
118:   setcolor(14);outtextxy(520,4,"CLASSIFICATION") ;
119:
120:   if (HEAD==0)
121:     { class_0 () ; }
122:   else
123:   if (HEAD==1)
124:     { class_1 () ; }
125:   else
126:   if (HEAD==2)
127:     { class_2 () ; }
128:   }
129:   delh () ;
130:   HEAD=0 ;BRAN=0 ;END=0 ;
131: }
132:
133: autom () /* automatic recognition */
134: {
135:   int k ;
136:   while ( !kbhit() )
137:     {
138:       clear_patt () ;
139:       gotoXY (3,24) ;
140:       printf ("### Please wait. I'm recieveing image ###") ;
141:       rs_232 () ;tone (678,3) ;
142:       gotoXY (3,24) ;
143:       printf ("hit any key for stop...") ;
144:       savpat () ;display () ;headc () ;
145:       if (HEAD>=3)
146:         { class_3 () ;k=1 ; }
147:       if (k==0)
148:         {
149:           thinn () ;redun () ;
150:           if (HEAD==0)

```

Listing of listhe.c

Page 4

8/14/1988 17:18:28

```

151:         { class_0 () ; }
152:     else
153:     if (HEAD==1)
154:         { class_1 () ; }
155:     else
156:     if (HEAD==2)
157:         { class_2 () ; }
158:     }
159:     delh () ;
160:     HEAD=0 ;BRAN=0 ;END=0 ;
161: }
162: }
163:
164: display ()
165: {
166:     int i,j ;
167:     for (i=0 ; i<64 ; i++)
168:     {
169:         for (j=0 ; j<64 ; j++)
170:         {
171:             if (PT0[i][j] == 1)
172:                 { putpixel ((j*2),(i*2),14) ; }
173:             else
174:                 { }
175:         }
176:     }
177: }
178: dpy ()
179: {
180:     int i,j ;
181:     for (i=0 ; i<64 ; i++)
182:     {
183:         for (j=0 ; j<64 ; j++)
184:         {
185:             if (PT0[i][j] == 1)
186:                 { putpixel ((j*2),(i*2),13) ; }
187:             else
188:                 { }
189:         }
190:     }
191: }
192: display1 ()
193: {
194:     int i,j ;
195:     for (i=0 ; i<64 ; i++)
196:     {
197:         for (j=0 ; j<64 ; j++)
198:         {
199:             if (PT0[i][j] == 1)
200:                 { putpixel ((j*2+128),(i*2),11) ; }

```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of listhe.c

Page 5

8/14/1988 17:18:28

```

201:         else
202:             if (PT0[i][j] == 2)
203:                 { putpixel ((j*2+128),(i*2),12) ; }
204:             else
205:                 if (PT0[i][j] == 3 || PT1[i][j] == 4)
206:                     { putpixel ((j*2+128),(i*2),14) ; }
207:             else
208:                 if (PT0[i][j] == 20)
209:                     { putpixel ((j*2+128),(i*2),15) ; }
210:             else
211:                 { }
212:         }
213:     }
214: }
215: dsp ()
216: {
217:     int i,j ;
218:     for (i=0 ; i<64 ; i++)
219:     {
220:         for (j=0 ; j<64 ; j++)
221:         {
222:             if (PT1[i][j]==1)
223:                 { putpixel ((j*2+256),(i*2),14) ; }
224:             else
225:                 if (PT1[i][j]==20)
226:                     { putpixel ((j*2+256),(i*2),15) ; }
227:             else
228:                 if (PT1[i][j]==15)
229:                     { putpixel ((j*2+256),(i*2),12) ; }
230:             else
231:                 if (PT1[i][j]==2)
232:                     { putpixel ((j*2+256),(i*2),13) ; }
233:             else
234:                 if (PT1[i][j]==3)
235:                     { putpixel ((j*2+256),(i*2),11) ; }
236:             else
237:                 if (PT1[i][j]==0)
238:                     { }
239:         }
240:     }
241: }
242: dspA ()
243: {
244:     int i,j ;
245:     for (i=0 ; i<64 ; i++)
246:     {
247:         for (j=0 ; j<64 ; j++)
248:         {
249:             if (PT0[i][j]==1)
250:                 { putpixel ((j*2+256),(i*2),12) ; }

```

Listing of listhe.c

Page 6

8/14/1988 17:18:28

```

251:         else
252:             { }
253:     }
254: }
255: }
256: display3 ()
257: {
258:     int i,j ;
259:     for (i=0 ; i<64 ; i++)
260:     {
261:         for (j=0 ; j<64 ; j++)
262:         {
263:             if (PT0[i][j] == 1)
264:                 { putpixel ((j*2+384),(i*2),10) ; }
265:             else
266:                 if (PT0[i][j] == 2)
267:                     { putpixel ((j*2+384),(i*2),12) ; }
268:                 else
269:                     if (PT0[i][j] == 3 || PT0[i][j] == 4)
270:                         { putpixel ((j*2+384),(i*2),14) ; }
271:                     else
272:                         { }
273:                 }
274:         }
275:     }
276: dipl ()
277: {
278:     int i,j ;
279:     for (i=0 ; i<64 ; i++)
280:     {
281:         for (j=0 ; j<64 ; j++)
282:         {
283:             if (PT2[i][j] == 1)
284:                 { putpixel ((j*2+512),(i*2),15) ; }
285:             else
286:                 if (PT2[i][j] == 2)
287:                     { putpixel ((j*2+512),(i*2),14) ; }
288:                 else
289:                     if (PT2[i][j] == 3 || PT2[i][j] == 4)
290:                         { putpixel ((j*2+512),(i*2),13) ; }
291:                     else
292:                         if (PT2[i][j] == 20)
293:                             { putpixel ((j*2+512),(i*2),10) ; }
294:                 }
295:         }
296:     }
297:     getch ();
298:     setcolor (13);
299:     line ((y1*2+512),(x1*2),(y2*2+512),(x1*2)) ;
300:     line ((y2*2+512),(x1*2),(y2*2+512),(x2*2)) ;
300:     line ((y2*2+512),(x2*2),(y1*2+512),(x2*2)) ;

```

ไม่ว่ากรณีใดๆทั้งสิ้น ออกกฎหมายใหม่ลดแบบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of listhe.c

Page 7

8/14/1988 17:18:28

```

301:   line ((y1*2+512),(x2*2),(y1*2+512),(x1*2)) ;
302:   line ((y1*2+512),(xb*2),(y2*2+512),(xb*2)) ;
303:   line ((y1*2+512),(xc*2),(y2*2+512),(xc*2)) ;
304:   line ((yb*2+512),(x1*2),(yb*2+512),(x2*2)) ;
305:   line ((yc*2+512),(x1*2),(yc*2+512),(x2*2)) ;
306: }
307: dip02 ()
308: {
309:   int i,j ;
310:   for (i=0 ; i<64 ; i++)
311:   {
312:     for (j=0 ; j<64 ; j++)
313:     {
314:       if (PT2[i][j] == 1)
315:         { putpixel ((j*2+512),(i*2),15) ; }
316:       else
317:         if (PT2[i][j] == 2)
318:           { putpixel ((j*2+512),(i*2),14) ; }
319:         else
320:           if (PT2[i][j] == 3 || PT2[i][j] == 4)
321:             { putpixel ((j*2+512),(i*2),13) ; }
322:           else
323:             if (PT2[i][j] == 20)
324:               { putpixel ((j*2+512),(i*2),10) ; }
325:         }
326:     }
327:   getch () ;
328:     setcolor (13) ;
329:   line ((y1*2+512),(x1*2),(y2*2+512),(x1*2)) ;
330:   line ((y2*2+512),(x1*2),(y2*2+512),(x2*2)) ;
331:   line ((y2*2+512),(x2*2),(y1*2+512),(x2*2)) ;
332:   line ((y1*2+512),(x2*2),(y1*2+512),(x1*2)) ;
333:   line ((y1*2+512),(xm*2),(y2*2+512),(xm*2)) ;
334:   line ((ym*2+512),(x1*2),(ym*2+512),(x2*2)) ;
335: }
336:
337: crosshead ()
338: {
339:   getch () ;
340:   setcolor (11) ;
341:   line ((yal*2+256),(hx*2),(yb1*2+256),(hx*2)) ;
342:   line ((hy*2+256),(xa1*2),(hy*2+256),(xb1*2)) ;
343: }
344:
345: tone (freq,time)      /* sound */
346: int freq,time ;
347: {
348:   sound(freq) ;
349:   delay((time+64)) ;
349:   nosound() ;

```

เป็นการตีโดฯทางสนออกหนังสือพิมพ์ให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of listhe.c

Page 8

8/14/1988 17:18:28

```
351: }
352: seffect ()
353: {
354:     int i;
355:     for(i=1500 ; i<2550 ; i=i+20)
356:     {
357:         sound(i);delay(2);nosound();
358:     }
359: }
360:
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of tools.c

Page 1

8/14/1988 17:29:38

```

1: #include <stdio.h>
2: #include <dos.h>
3: #define TV_SEG 0x7000
4: extern char buff[2048] ;
5: extern unsigned char PT0[64][64] ; /* input,thin,redun */
6: extern unsigned char PT1[64][64] ; /* head location */
7: extern unsigned char PT2[64][64] ; /* end,branch,head */
8: extern unsigned char PT3[64][64] ; /* redun buffer */
9: extern unsigned char PT4[64][64] ; /* "-----" */
10: extern char *G ; /* file name for load */
11: extern char *H ; /* file name for save */
12: extern char *R ; /* file name for recog*/
13: extern int xm,ym ;
14: extern int x1,x2,y1,y2,A,B ;
15: extern unsigned char IH[ 5],JH[ 5] ; /* keep head point */
16: extern unsigned char IB[ 5],JB[ 5] ; /* keep branch ...' */
17: extern unsigned char IE[ 5],JE[ 5] ; /* keep end .....' */
18:
19: load () /* load pattern 64 X 64 pixel */
20: {
21:     int i,j ;
22:     long int k = 0,step = 0 ;
23:     FILE *fp ;
24:     gotoXY (3,24) ;
25:     printf ("Input filename to load : ") ;
26:     gets (G) ;
27:     if ( (fp = fopen(G,"r")) != NULL)
28:     {
29:         gotoXY (3,24) ;
30:         printf ("Waiting for loading.... ") ;
31:         for (i=0 ; i<2 ; i++)
32:         {
33:             fseek (fp,(long) i*2048,0) ;
34:             fread (buff,2048,1,fp) ;
35:             for (j=0 ; j<2048 ; j++)
36:             {
37:                 pokeb (TV_SEG,j+k,buff[j]) ;
38:             }
39:             k = k + 2048 ;
40:         }
41:         fclose (fp) ;
42:         for (i=0 ; i<64 ; i++)
43:         {
44:             for (j=0 ; j<64 ; j++)
45:             {
46:                 PT0[i][j] = peekb(TV_SEG,j+step) ;
47:             }
48:             step = step + 64 ;
49:         }
50:         gotoXY (3,24) ;

```

สารบัญเนื้อหา การที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ออกทั้งหมดมีเหตุผลแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of tools.c

Page 2

8/14/1988 17:29:38
"); ;

```

51:     printf ("
52:     }
53:     else
54:     {
55:         gotoXY (3,24) ;
56:         printf ("file not found           ") ;
57:         getch () ;
58:         gotoXY (3,24) ;
59:         printf ("                               ") ;
60:     }
61: }
62:
63: save ()    /* save pattern 64 X 64 pixel */
64: {
65:     int i,j ;
66:     long int k = 0,step = 0 ;
67:     FILE *fp ;
68:     gotoXY (3,24) ;
69:     printf ("Input filename to save : ") ;
70:     gets (H) ;
71:     if ( (fp = fopen(H,"w")) != NULL)
72:     {
73:         gotoXY (3,24) ;
74:         printf ("Waiting for saving....") ;
75:         for (i=0 ; i<64 ; i++)
76:         {
77:             for (j=0 ; j<64 ; j++)
78:             {
79:                 pokeb (TV_SEG,j+step,PT0[i][j]) ;
80:             }
81:             step = step + 64 ;
82:         }
83:         for (i=0 ; i<2 ; i++)
84:         {
85:             for (j=0 ; j<2048 ; j++)
86:             {
87:                 buff[j] = peekb (TV_SEG,j+k) ;
88:             }
89:             fseek (fp,(long) i*2048,0) ;
90:             fwrite (buff,2048,1,fp) ;
91:             k = k + 2048 ;
92:         }
93:         fclose (fp) ;
94:     }
95:     gotoXY (3,24) ;
96:     printf ("                               ") ;
97: }
98: }
99:
100: savpat () /* save pattern 64 X 64 pixel to array */

```

Listing of tools.c

Page 3

8/14/1988 17:29:38

```

101: {
102:     int i,j,k ;
103:     long int step = 4096 ;
104:     for (i=0 ; i<64 ; i++)
105:     {
106:         for (j=0 ; j<64 ; j++)
107:         {
108:             k = (peekb(TV_SEG,j+step)) - 0x20 ;
109:             if (k == 0)
110:                 { k = 1 ; }
111:             else
112:                 { k = 0 ; }
113:             PT1[i][j] = k ;
114:         }
115:         step = step - 64 ;
116:     }
117:     /* swap */
118:     for (i=0 ; i<64 ; i++)
119:     {
120:         for (j=0 ; j<64 ; j++)
121:         {
122:             PT0[i][j] = PT1[i][64-j] ;
123:         }
124:     }
125:     for (i=59 ; i<64 ; i++)
126:     {
127:         for (j=0 ; j<64 ; j++)
128:         {
129:             PT0[i][j] = 0 ;
130:         }
131:     }
132:     for (j=0 ; j<64 ; j++)
133:     { PT0[0][j] = 0 ; PT0[1][j] = 0 ; }
134: }
135: delh ()
136: {
137:     int t ;
138:     for (t=0 ; t<5 ; t++)
139:     {
140:         IH[t] = 0 ; JH[t] = 0 ;
141:         IB[t] = 0 ; JB[t] = 0 ;
142:         IE[t] = 0 ; JE[t] = 0 ;
143:     }
144: }
145: clear_patt ()
146: {
147:     int i,j ;
148:     for (i=0 ; i<64 ; i++)
149:     {
150:         for (j=0 ; j<64 ; j++)

```

การนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ
 วิศวกรรมศาสตร์ วิทยาลัยพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Listing of tools.c

Page 4

8/14/1988 17:29:38

```

151:     {
152:         PT1[i][j] = 0 ;PT2[i][j] = 0 ;
153:         PT3[i][j] = 0 ;PT4[i][j] = 0 ;
154:     }
155: }
156: for (i=0 ; i<4096 ; i++)
157:     { pokeb (TV_SEG,i,0) ; }
158: }
159: border () /* find pattern border */
160: {
161:     int x=0,y=0 ;
162:     while ( (PT0[x][y]) == 0)
163:     {
164:         y++ ;
165:         if (y==63)
166:             { y=0 ;x++ ;}
167:     }
168:     x1 = x-1 ; /* upper bound */
169:     x=63 ;y=63 ;
170:     while ( (PT0[x][y]) == 0)
171:     {
172:         y-- ;
173:         if (y==0)
174:             { y=63 ;x-- ;}
175:     }
176:     x2 = x+1 ; /* lower bound */
177:     x=0 ;y=0 ;
178:     while ( (PT0[x][y]) == 0)
179:     {
180:         x++ ;
181:         if (x==63)
182:             { x=0 ;y++ ;}
183:     }
184:     y1 = y-1 ; /* left bound */
185:     x=63 ;y=63 ;
186:     while ( (PT0[x][y]) == 0)
187:     {
188:         x-- ;
189:         if (x==0)
190:             { x=63 ;y-- ;}
191:     }
192:     y2 = y+1 ; /*right bound */
193: }
194:
195: center_g () /* fine center of gravity */
196: {
197:     int i,j ;
198:     A = 0 ;B = 0 ;
199:     for (i=x1 ; i<x2 ; i++)
200:     {

```

ส่วนนี้เป็นเอกสารสงวนลิขสิทธิ์โรงเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of tools.c

8/14/1988 17:29:38

Page 5

```

201:     for (j=y1 ; j<y2 ; j++)
202:     {
203:         A = A + ((PTO[i][j]) * j) ;
204:         B = B + (PTO[i][j]) ;
205:     }
206: }
207: ym = A / B ;
208: A = 0 ;B = 0 ;
209: for (j=y1 ; j<y2 ; j++)
210: {
211:     for (i=x1 ; i<x2 ; i++)
212:     {
213:         A = A + ((PTO[i][j]) * i) ;
214:         B = B + (PTO[i][j]) ;
215:     }
216: }
217: xm = A / B ;
218: }
219: fill ()
220: {
221:     int ix,iy,jx,jy,np ;
222:     for (iy=1 ; iy<63 ; iy++)
223:     {
224:         for (ix=1 ; ix<63 ; ix++)
225:         {
226:             np=0 ;
227:             for (jy=(iy-1) ; jy<=(iy+1) ; jy++)
228:             {
229:                 for (jx=(ix-1) ; jx<=(ix+1) ; jx++)
230:                 {
231:                     np = np + PTO[jx][jy] ;
232:                 }
233:             }
234:             if ( PTO[ix][iy]==1 && np==1 )
235:             { PTO[ix][iy] = 0 ; }
236:             if ( PTO[ix][iy]==1 && np==2 )
237:             { PTO[ix][iy] = 1 ; }
238:         }
239:     }
240: }
241: move ()
242: {
243:     int xm,ym,ix,iy,i,j ;
244:     border () ;
245:     xm = (x1+x2)/2 ;ym = (y1+y2)/2 ;
246:     for (iy=y1 ; iy<=y2 ; iy++)
247:     {
248:         for (ix=x1 ; ix<=x2 ; ix++)
249:         {
250:             PT4[ix-xm+32][iy-ym+32] = PTO[ix][iy] ;

```

การนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม้วางกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of tools.c

Page 6

8/14/1988 17:29:38

```

251:     }
252:   }
253:   for (i=0 ; i<64 ; i++)
254:   {
255:     for (j=0 ; j<64 ; j++)
256:     {
257:       PT0[i][j] = PT4[i][j] ;PT4[i][j] = 0 ;
258:     }
259:   }
260:   x1 = x1-xm+32 ;x2 = x2-xm+32 ;
261:   y1 = y1-ym+32 ;y2 = y2-ym+32 ;
262: }
263:
264:

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of thinn.c

Page 1

8/14/1988 17:47:21

```

1: extern unsigned char    PT0[64][64] ;    /* input,thin,redun */
2: extern unsigned char    PT1[64][64] ;    /* head location    */
3: extern unsigned char    PT2[64][64] ;    /* end,branch,head  */
4: extern unsigned char    PT3[64][64] ;    /* redun buffer     */
5: extern unsigned char    PT4[64][64] ;    /* "-----"       */
6: extern unsigned char    ID[10],JD[10] ;   /* delete for redun */
7: extern unsigned char    IH[ 5],JH[ 5] ;   /* keep head point  */
8: extern unsigned char    IB[ 5],JB[ 5] ;   /* keep branch ...' */
9: extern unsigned char    IE[ 5],JE[ 5] ;   /* keep end .....' */
10: extern int              HEAD ;
11: extern int              BRAN ;
12: extern int              END ;
13: extern int              VECTOR ;
14: extern int              xm,ym ;
15: extern int              x1,x2,y1,y2,A,B ;
16: thinn ()
17: {
18:     int i ;
19:     int j,flag = 1 ;
20:     int k0,k1,k2,k3,k4,k5,k6,k7 ;
21:     int sumA,sumB,sumC,sumD,sumE,sumF ;
22:     int sumG,sumH,sumI,sumJ,sumK,sumL ;
23:     while (flag == 1)
24:     {
25:         flag = 0 ;
26:         for (j=0 ; j<64 ; j++)
27:         {
28:             for (i=0 ; i<64 ; i++)
29:             {
30:                 if (PT0[i][j] == 1)
31:                 {
32:                     k0 = PT0[i][j+1] ;
33:                     k1 = PT0[i-1][j+1] ;
34:                     k2 = PT0[i-1][j] ;
35:                     k3 = PT0[i-1][j-1] ;
36:                     k4 = PT0[i][j-1] ;
37:                     k5 = PT0[i+1][j-1] ;
38:                     k6 = PT0[i+1][j] ;
39:                     k7 = PT0[i+1][j+1] ;
40:                     sumK = k0 | k4 | k5 | k7 ;
41:                     sumL = k0 | k1 | k3 | k4 ;
42:                     sumE = k0 | ~k1 ;sumF = ~k3 | k4 ;
43:                     sumG = k4 | ~k5 ;sumH = k0 | ~k7 ;
44:                     if (k2 == 0)
45:                     {
46:                         if ( (k6 & sumK & sumE & sumF) == 1 )
47:                             PT0[i][j] = 8 ;
48:                     }
49:                 }
50:                 else if (k6 == 0)

```

สารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 หรือการตีพิมพ์ใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่เอกสารนี้โดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of thinn.c

Page 2

8/14/1988 17:47:21

```

51:         {
52:             if ( (k2 & sumL & sumG & sumH) == 1 )
53:                 PT0[i][j] = 8 ;
54:         }
55:     }
56: }
57: }
58:
59: for (i=0 ; i<64 ; i++)
60: {
61:     for (j=0 ; j<64 ; j++)
62:     {
63:         if (PT0[i][j] == 1)
64:         {
65:             k0 = PT0[i][j+1] ;
66:             k1 = PT0[i-1][j+1] ;
67:             k2 = PT0[i-1][j] ;
68:             k3 = PT0[i-1][j-1] ;
69:             k4 = PT0[i][j-1] ;
70:             k5 = PT0[i+1][j-1] ;
71:             k6 = PT0[i+1][j] ;
72:             k7 = PT0[i+1][j+1] ;
73:             sumI = k1 | k2 | k6 | k7 ;
74:             sumJ = k2 | k3 | k5 | k6 ;
75:             sumA = k2 | ~k3 ; sumB = ~k5 | k6 ;
76:             sumC = k6 | ~k7 ; sumD = ~k1 | k2 ;
77:             if (k4 == 0)
78:             {
79:                 if ( (k0 & sumI & sumA & sumB) == 1 )
80:                     PT0[i][j] = 8 ;
81:             }
82:             else
83:             if (k0 == 0)
84:             {
85:                 if ( (k4 & sumJ & sumC & sumD) == 1 )
86:                     PT0[i][j] = 8 ;
87:             }
88:         }
89:     }
90: }
91: for (i=0 ; i<64 ; i++)
92: {
93:     for (j=0 ; j<64 ; j++)
94:     {
95:         if (PT0[i][j] == 8)
96:         {
97:             flag = 1 ;
98:             PT0[i][j] = 0 ;
99:         }
100:     }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกสิ่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of thinn.c

Page 3

8/14/1988 17:47:21

```

101:     }
102: }
103: }
104:
105: redun ( )
106: {
107:     int i,j,k,t,y,f,x=0 ;
108:     int pix0,pix1,pix2,pix3 ;
109:     int pix4,pix5,pix6,pix7,pix8 ;
110:     int A,B,C,D,E,F,G,H,NC8,NC4 ;
111:
112:     for (t=0 ; t<5 ; t++)
113:     {
114:         for (i=0 ; i<64 ; i++)
115:         {
116:             for (j=0 ; j<64 ; j++)
117:             {
118:                 if (PT0[i][j] == 1)
119:                 {
120:                     pix1 = PT0[i][j+1] ;pix2 = PT0[i-1][j+1] ;
121:                     pix3 = PT0[i-1][j] ;pix4 = PT0[i-1][j-1] ;
122:                     pix5 = PT0[i][j-1] ;pix6 = PT0[i+1][j-1] ;
123:                     pix7 = PT0[i+1][j] ;pix8 = PT0[i+1][j+1] ;
124:                     A = pix1-(pix1*pix2*pix3) ;
125:                     B = pix3-(pix3*pix4*pix5) ;
126:                     C = pix5-(pix5*pix6*pix7) ;
127:                     D = pix7-(pix7*pix8*pix1) ;
128:                     E = (1-pix1)-((1-pix1)*(1-pix2)*(1-pix3)) ;
129:                     F = (1-pix3)-((1-pix3)*(1-pix4)*(1-pix5)) ;
130:                     G = (1-pix5)-((1-pix5)*(1-pix6)*(1-pix7)) ;
131:                     H = (1-pix7)-((1-pix7)*(1-pix8)*(1-pix1)) ;
132:                     NC4 = A + B + C + D ;
133:                     NC8 = E + F + G + H ;
134:                     if ( (NC8==1 && NC4==1) || (NC8==1 && NC4==0) )
135:                         { ID[x] = i ;JD[x] = j ;x++ ;PT3[i][j] = 1 ; }
136:                     else
137:                         if ( NC8==3 || NC4==3 )
138:                             { PT3[i][j] = 2 ; }
139:                     else
140:                         if ( NC4==2 && NC8==2 )
141:                             {
142:                                 if ( (pix1+pix5)==2 || (pix3+pix7)==2 )
143:                                     { }
144:                                 else
145:                                     { PT3[i][j] = 2 ; }
146:                             }
147:                 }
148:             }
149:         }
150:     }
for (y=0 ; y<x ; y++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

แม้ว่ากรณี for (y=0 ; y<x ; y++) เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of thinn.c

Page 4

8/14/1988 17:47:21

```

151:      { PT0[ ID[y] ][ JD[y] ] = 0 ; }
152:      x = 0 ;
153:  }
154:  for (t=0 ; t<5 ; t++)
155:  {
156:      for (j=0 ; j<64 ; j++)
157:      {
158:          for (i=0 ; i<64 ; i++)
159:          {
160:              if (PT3[i][j]==1)
161:              {
162:                  if (PT3[i-1][j-1]==2 || PT3[i-1][j]==2 || PT3[i-1][j+1]==2
163:                      || PT3[i][j+1]==2 || PT3[i+1][j-1]==2 || PT3[i+1][j]==2
164:                      || PT3[i+1][j+1]==2)
165:                      { PT3[i][j] = 2 ; }
166:              }
167:          }
168:      }
169:      for (i=0 ; i<64 ; i++)
170:      {
171:          for (j=0 ; j<64 ; j++)
172:          {
173:              if (PT3[i][j]==2)
174:              { PT3[i][j] = 0 ; }
175:          }
176:      }
177:      for (i=0 ; i<64 ; i++)
178:      {
179:          for (j=0 ; j<64 ; j++)
180:          {
181:              PT0[i][j] = PT0[i][j] + PT3[i][j] ;
182:          }
183:      }
184: /* for (i=0 ; i<64 ; i++)
185:  {
186:      for (j=0 ; j<64 ; j++)
187:      {
188:          if (PT4[i][j] == 0)
189:              { PT0[i][j] = 0 ; }
190:          else
191:              { PT0[i][j] = 1 ; }
192:      }
193:  } */
194:
195: /*--- fine NC4,8 of [PT0] --> [PT2] ----*/
196: for (i=0 ; i<64 ; i++)
197: {
198:     for (j=0 ; j<64 ; j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆก็ตาม หากมีข้อผิดพลาดประการใด ขออภัยและต้องอภัยถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of thinn.c

Page 5

8/14/1988 17:47:21

```

199:      {
200:          if (PT0[i][j] == 1)
201:          {
202:              pix1 = PT0[i][j+1] ;pix2 = PT0[i-1][j+1] ;
203:              pix3 = PT0[i-1][j] ;pix4 = PT0[i-1][j-1] ;
204:              pix5 = PT0[i][j-1] ;pix6 = PT0[i+1][j-1] ;
205:              pix7 = PT0[i+1][j] ;pix8 = PT0[i+1][j+1] ;
206:              A = pix1-(pix1*pix2*pix3) ;
207:              B = pix3-(pix3*pix4*pix5) ;
208:              C = pix5-(pix5*pix6*pix7) ;
209:              D = pix7-(pix7*pix8*pix1) ;
210:              E = (1-pix1)-((1-pix1)*(1-pix2)*(1-pix3)) ;
211:              F = (1-pix3)-((1-pix3)*(1-pix4)*(1-pix5)) ;
212:              G = (1-pix5)-((1-pix5)*(1-pix6)*(1-pix7)) ;
213:              H = (1-pix7)-((1-pix7)*(1-pix8)*(1-pix1)) ;
214:              NC4 = A + B + C + D ;
215:              NC8 = E + F + G + H ;
216:              if ( (NC4==1 && NC8==1) || (NC4==0 && NC8==1) )
217:                  { PT2[i][j] = 1 ;IE[END] = i ;JE[END] = j ;END++ ;
218:                  }
219:              else
220:                  if ( (NC4==1 && NC8==2) || (NC4==2 && NC8==0) || (NC4=
221:                  =2 && NC8==2) || (NC4==2 && NC8==1) || (NC4==0 && NC8==2) )
222:                      { PT2[i][j] = 2 ; }
223:                  else
224:                      if ( (NC4==3 && NC8==1) || (NC4==0 && NC8==3) || (NC4=
225:                      =1 && NC8==3) || (NC4==2 && NC8==2) )
226:                          { PT2[i][j] = 3 ;IB[BRAN] = i ;JB[BRAN] = j ;BRAN++
227:                          ; }
228:                      else
229:                          if ( (NC4==4 && NC8==0) || (NC4==0 && NC8==4) )
230:                              { PT2[i][j] = 3 ;IB[BRAN] = i ;JB[BRAN] = j ;BRAN++
231:                              ; }
232:                      }
233:                  }
234:          }
235:      }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of head.c

Page 1

8/14/1988 17:52:24

```

1: extern unsigned char PT0[64][64] ; /* input,thin,redun */
2: extern unsigned char PT1[64][64] ; /* head location */
3: extern unsigned char PT2[64][64] ; /* end,branch,head */
4: extern unsigned char PT3[64][64] ; /* redun buffer */
5: extern unsigned char PT4[64][64] ; /* "-----" */
6: extern unsigned char ID[10],JD[10] ; /* delete for redun */
7: extern unsigned char IH[ 5],JH[ 5] ; /* keep head point */
8: extern unsigned char IB[ 5],JB[ 5] ; /* keep branch ...' */
9: extern int HEAD,BRAN ; /* number of head & branch */
10: extern int VECTOR ; /* code-vector for CLASS_0,2 */
11: extern int xm,ym ;
12: extern int x1,x2,y1,y2,A,B ;
13: extern int xa1,xb1,ya1,yb1,hx,hy ;
14:
15: headc ()
16: {
17:     int i,j,k,h,f=0 ;
18:     int ix,iy,jx,jy ;
19:     int bx,by,dx,dy ;
20:     int xs,ys,kk,w ;
21:     int pix0,pix1,pix2,pix3 ;
22:     int pix4,pix5,pix6,pix7,pix8 ;
23:     int E,F,G,H,NC8,p=0 ;
24:     char id[200],jd[200] ; /*delete*/
25:     for (i=0 ; i<64 ; i++)
26:     {
27:         for (j=0 ; j<64 ; j++)
28:         {
29:             PT1[i][j] = 0 ;
30:         }
31:     }
32:         /* find topological properties */
33:     for (i=0 ; i<64 ; i++)
34:     {
35:         for (j=0 ; j<64 ; j++)
36:         {
37:             pix0 = PT0[i][j] ;
38:             if (pix0 == 1)
39:             {
40:                 pix1 = PT0[i][j+1] ;pix2 = PT0[i-1][j+1] ;
41:                 pix3 = PT0[i-1][j] ;pix4 = PT0[i-1][j-1] ;
42:                 pix5 = PT0[i][j-1] ;pix6 = PT0[i+1][j-1] ;
43:                 pix7 = PT0[i+1][j] ;pix8 = PT0[i+1][j+1] ;
44:                 E = (1-pix1)-((1-pix1)*(1-pix2)*(1-pix3)) ;
45:                 F = (1-pix3)-((1-pix3)*(1-pix4)*(1-pix5)) ;
46:                 G = (1-pix5)-((1-pix5)*(1-pix6)*(1-pix7)) ;
47:                 H = (1-pix7)-((1-pix7)*(1-pix8)*(1-pix1)) ;
48:                 NC8 = E + F + G + H ;
49:                 if ( NC8==1)
50:                 {

```

Listing of head.c

Page 2

8/14/1988 17:52:24

```

51:             PT1[i][j] = 1 ;
52:         }
53:     else
54:     {
55:         PT1[i][j] = 0 ;
56:     }
57:     }
58: }
59: }
60: dspA ( ) ;
61: for (h=0 ; h<5 ; h++)
62: {
63:     ix=0 ;iy=0 ;
64:     while ( PT1[ix][iy] != 1 )
65:     {
66:         iy++ ;
67:         if (iy==63)
68:             { iy=0 ;ix++ ; }
69:         if (ix==63)
70:             { f=1 ;break ; }
71:     }
72:     xs=ix ;ys=iy ;
73:     PT1[xs][ys] = 15 ;id[p] = xs ;jd[p] = ys ;p++ ;
74:     jx=ix-1 ;jy=iy ;
75:     /*-start contour-*/
76:     bx=jx-ix ;by=jy-iy ;
77:     jx=ix ;jy=iy ;
78:     kk=1 ;
79:     while ( kk<=8 && f==0 )
80:     {
81:         dx=bx+by ;dy=by-bx ;
82:         if (dx != 0)
83:             { dx = dx / abs(dx) ; }
84:         if (dy != 0)
85:             { dy = dy / abs(dy) ; }
86:         ix=jx+dx ;iy=jy+dy ;
87:         if (PT1[ix][iy] != 0)
88:         {
89:             if ( (ix==xs) && (iy==ys) )
90:             {
91:                 break ;
92:             }
93:             else
94:             {
95:                 PT1[ix][iy] = 15 ;id[p] = ix ;jd[p] = iy ;p++ ;
96:                 bx=jx-ix ;by=jy-iy ;
97:                 jx=ix ;jy=iy ;
98:                 kk=1 ;
99:                 }
100:         }

```

สารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 หากกรณีใด ๆ ที่เห็น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of head.c

Page 3

8/14/1988 17:52:24

```

101:     else
102:     {
103:         bx=dx ;by=dy ;
104:         kk++ ;
105:     }
106: }
107: if (p==1)
108:     { break ; }
109: else
110: if ( p<25 && p>3 )
111: {
112:     h_ct () ; /* find head location */
113:     HEAD++ ;
114:     for (i=0 ; i<p ; i++)
115:         { PT1[ id[i] ][ jd[i] ] = 0 ; }
116: }
117: else
118: {
119:     for (i=0 ; i<p ; i++)
120:         { PT1[ id[i] ][ jd[i] ] = 0 ; }
121: }
122: p = 0 ;
123: seffect () ;
124: dsp () ;
125: }
126: gotoXY (3,13) ;printf ("Number of HEAD = %d head",HEAD) ;
127: }
128:
129: h_ct () /* find head center */
130: {
131:     int x=0,y=0 ;
132:
133:     while ( (PT1[x][y]) != 15)
134:     {
135:         y++ ;
136:         if (y==63)
137:             { y=0 ;x++ ;}
138:     }
139:     xal = x-1 ; /* upper bound */
140:     x=63 ;y=63 ;
141:     while ( (PT1[x][y]) != 15)
142:     {
143:         y-- ;
144:         if (y==0)
145:             { y=63 ;x-- ;}
146:     }
147:     xbl = x+1 ; /* lower bound */
148:     x=0 ;y=0 ;
149:     while ( (PT1[x][y]) != 15)
150:     {

```

เอกสารนี้จัดทำขึ้นเพื่อแจกจ่ายแก่บุคลากรในหน่วยงานที่สนใจศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการศึกษา
 ฝากอธิบดีฯ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of head.c

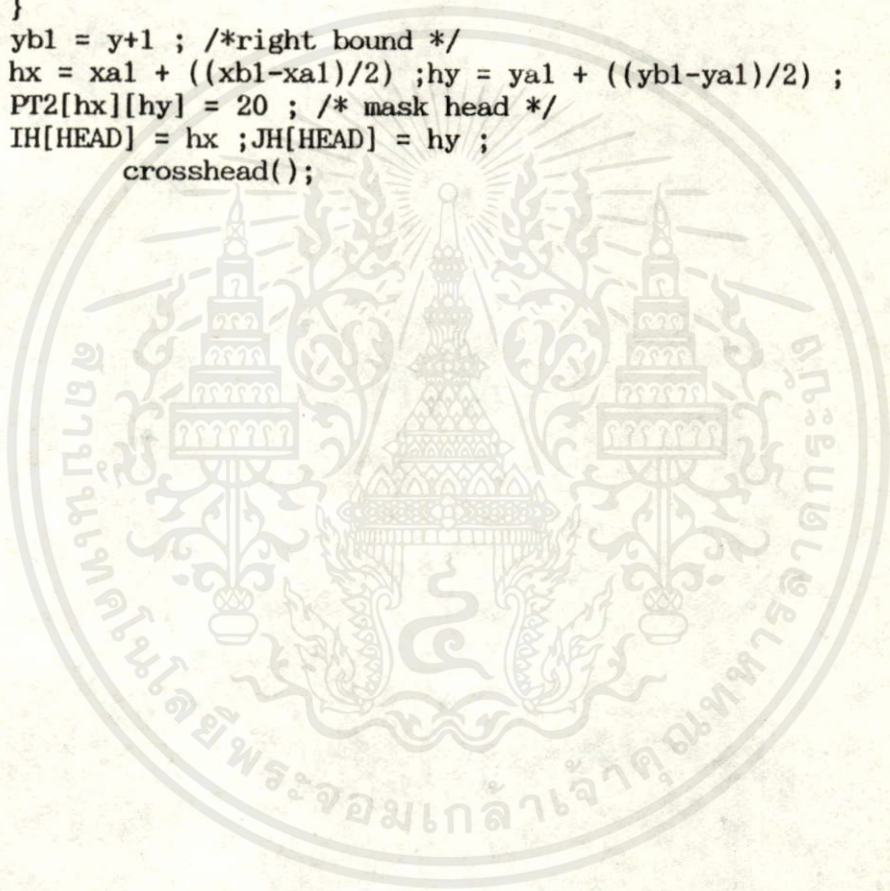
Page 4

8/14/1988 17:52:24

```

151:         x++ ;
152:         if (x==63)
153:             { x=0 ;y++ ;}
154:     }
155:     ya1 = y-1 ; /* left bound */
156:     x=63 ;y=63 ;
157:     while ( (PT1[x][y]) != 15)
158:     {
159:         x-- ;
160:         if (x==0)
161:             { x=63 ;y-- ;}
162:     }
163:     yb1 = y+1 ; /*right bound */
164:     hx = xa1 + ((xb1-xa1)/2) ;hy = ya1 + ((yb1-ya1)/2) ;
165:     PT2[hx][hy] = 20 ; /* mask head */
166:     IH[HEAD] = hx ; JH[HEAD] = hy ;
167:     crosshead();
168: }
169:
170:

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class0.c

Page 1

8/14/1988 17:56:17

```

1: /* classifier NO head pattern */
2: extern unsigned char PT2[64][64] ; /* end,branch,head */
3: extern int VECTOR,REST ;          /* code-vector for CLASS_0,2 */
4: extern int xm,ym ;
5: extern int x1,x2,y1,y2,A,B ;
6: extern int endl,end2,end3,end4 ;
7:
8: class_0 () /* [no head] */
9: {
10:     int i,j,k ;
11:     border () ;
12:     center_g () ;
13:     dip02 () ;
14:     /* find vector code */
15:     for (i=x1 ; i<=x2 ; i++)
16:     {
17:         if (PT2[i][ym] != 0)
18:             { VECTOR++ ; }
19:         else
20:             { }
21:     }
22:     /* fine NC8 in each 4-zone */
23:     /* Z1 */
24:     for (i=x1 ; i<=xm ; i++)
25:     {
26:         for (j=y1 ; j<=ym ; j++)
27:         {
28:             if (PT2[i][j]==1)
29:                 { endl1++ ; }
30:         }
31:     }
32:     /* Z2 */
33:     for (i=x1 ; i<=xm ; i++)
34:     {
35:         for (j=ym+1 ; j<y2 ; j++)
36:         {
37:             if (PT2[i][j]==1)
38:                 { end2++ ; }
39:         }
40:     }
41:     /* Z3 */
42:     for (i=xm+1 ; i<x2 ; i++)
43:     {
44:         for (j=y1 ; j<=ym ; j++)
45:         {
46:             if (PT2[i][j]==1)
47:                 { end3++ ; }
48:         }
49:     }
50:     /* Z4 */

```

49: } การที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 50: การแก้ไข ทั้งต้น ออกใหม่ให้มีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class0.c

Page 2

8/14/1988 17:56:17

```

51:   for (i=xm+1 ; i<x2 ; i++)
52:   {
53:       for (j=ym+1 ; j<y2 ; j++)
54:       {
55:           if (PT2[i][j]==1)
56:               { end4++ ; }
57:       }
58:   }
59:
60:   if ( (end1==1 || end3==1) && end2==1 && VECTOR>=3)
61:       { REST = 182 ; }
62:   else
63:       if ( end3==1 && end4==1 && VECTOR==1)
64:           { REST = 161 ; }
65:       else
66:           if ( (end1==1 || end3==1) && end4==1 && VECTOR==1)
67:               { REST = 206 ; }
68:           else
69:               { REST = 224 ; }
70:   good ( ) ;
71:   VECTOR = 0 ; REST = 0 ;
72:   end1=0 ;end2=0 ;end3=0 ;end4=0 ;
73: }
74:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class1.c

Page 1

8/14/1988 17:59:02

```

1: /* classifier ONE head pattern */
2: extern unsigned char PT0[64][64] ;
3: extern unsigned char PT2[64][64] ; /* end,branch,head */
4: extern unsigned char IH[ 5],JH[ 5] ; /* keep head point */
5: extern unsigned char IB[ 5],JB[ 5] ; /* keep branch ...' */
6: extern unsigned char IE[ 5],JE[ 5] ; /* keep end .....' */
7: extern int VECTOR ;
8: extern int xm,ym ;
9: extern int x1,x2,y1,y2 ;
10: extern int hed1,hed2,hed3,hed4,hed5,hed6,hed7,hed8,hed9 ;
11: extern int end1,end2,end3,end4,end5,end6,end7,end8,end9 ;
12: extern int count1,count2,count3,count7 ;
13: extern int REST ;
14: extern int xa,xb,xc,ya,yb,yc ;
15: int hsty,vtor,xh,yh ;
16: int vtor_R ;
17: float wid=0.0,hig=0.0 ;
18: class_1 () /* [one head] */
19: {
20:     int i,j,k ;
21:     int xbound,ybound ;
22:     border () ;
23:                                     /* Boundary of each zone */
24:     xbound=(x2-x1)/3 ;
25:     ybound=(y2-y1)/3 ;
26:                                     /* devide to 9-zone */
27:     xa=x1 ;xb=x1+xbound ;xc=xb+xbound ;
28:     ya=y1 ;yb=y1+ybound ;yc=yb+ybound ;
29: /*     dipl () ; */
30:                                     /* fine NC8 in each 9-zone */
31:     /* Z1 */
32:     for (i=xa ; i<=xb ; i++)
33:     {
34:         for (j=ya ; j<=yb ; j++)
35:         {
36:             if (PT2[i][j]==20)
37:             {
38:                 hed1 = 1 ;
39:             }
40:         }
41:     }
42:     /* Z2 */
43:     for (i=xa ; i<=xb ; i++)
44:     {
45:         for (j=yb+1 ; j<=yc ; j++)
46:         {
47:             if (PT2[i][j]==20)
48:             {
49:                 hed2 = 1 ;
50:             }

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ทำการตีค่าทั้งสามนี้ } จึงห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีตรรกะไปใช้

Listing of class1.c

Page 3

8/14/1988 17:59:02

```

101:      {
102:          if (PT2[i][j]==20)
103:          {
104:              hed7 = 1 ;
105:          }
106:      }
107:  }
108:  /* Z8 */
109:  for (i=xc+1 ; i<=x2 ; i++)
110:  {
111:      for (j=yb+1 ; j<=yc ; j++)
112:      {
113:          if (PT2[i][j]==20)
114:          {
115:              hed8 = 1 ;
116:          }
117:      }
118:  }
119:  /* Z9 */
120:  for (i=xc+1 ; i<=x2 ; i++)
121:  {
122:      for (j=yc+1 ; j<=y2 ; j++)
123:      {
124:          if (PT2[i][j]==20)
125:          {
126:              hed9 = 1 ;
127:          }
128:      }
129:  }
130:  /* ----- */
131:  if ( hed1==1 || hed2==1 || hed3==1 ) /*up head */
132:  { m1A ( ) ; }
133:  else
134:  if ( hed4==1 || hed5==1 || hed6==1 ) /*md head */
135:  { m1B ( ) ; }
136:  else
137:  if ( hed7==1 || hed8==1 || hed9==1 ) /*lo head */
138:  { m1C ( ) ; }
139:  good ( ) ;
140:  hed1=0 ;hed2=0 ;hed3=0 ;hed4=0 ;hed5=0 ;
141:  hed6=0 ;hed7=0 ;hed8=0 ;hed9=0 ;VECTOR=0 ;
142:  end1=0 ;end2=0 ;end3=0 ;end4=0 ;end5=0 ;
143:  end6=0 ;end7=0 ;end8=0 ;end9=0 ;xh=0 ;yh=0 ;
144:  hsty=0 ;vtor=0 ;wid=0.0 ;hig=0.0 ;vtor_R=0 ;
145:  xa=0 ;xb=0 ;xc=0 ;ya=0 ;yb=0 ;yc=0 ;
146:  }
147:
148: m1A ( )
149: {
150:     int i,j;

```

Listing of class1.c

Page 4

8/14/1988 17:59:02

```

151:   int x,a ;
152:   gotoXY (3,15) ;printf ("UPPER HEAD") ;
153:   border () ;center_g () ;zone4 () ; /* devide 4 zone */
154:   dip02 () ;
155:   if ((end1==1 || end3==1) && end4==0)
156:   {
157:       REST = 165 ;
158:   }
159:   else
160:   {
161:       if (end4==1)
162:       {
163:           /* crossh */
164:           vtor = 0 ;
165:           for (j=y1 ; j<=y2 ; j++)
166:           {
167:               if ( PTO[IH[0]][j]==1 )
168:                   { vtor++ ; }
169:               else
170:                   { }
171:           }
172:           if ( vtor<5 )
173:               { REST = 181 ; }
174:           else
175:               { REST = 175 ; }
176:       }
177:       else
178:       {
179:           /* head style */
180:           hsty = 0 ;
181:           for (i=IH[0] ; i<=x2 ; i++)
182:           {
183:               if (PTO[i][ JH[0] ]==1)
184:                   { hsty++ ; }
185:           }
186:           if (hsty==1)
187:           {
188:               /* out head */
189:               /* crossbase */
190:               VECTOR = 0 ;
191:               for (j=y1 ; j<=y2 ; j++)
192:               {
193:                   if ( PTO[x2-4][j]==1 )
194:                       { VECTOR++ ; }
195:                   else
196:                       { }
197:               }
198:               if ( VECTOR<=2 )
199:               {
200:                   /* crossh */

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในกิจการการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ถ้ากรณีใดๆทั้งสิ้น อีกทั้งห้า และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class1.c

Page 5

8/14/1988 17:59:02

```

201:         vtor = 0 ;
202:         for (j=y1 ; j<=y2 ; j++)
203:         {
204:             if ( PTO[IH[0]][j]==1 )
205:                 { vtor++ ; }
206:             else
207:                 { }
208:         }
209:         if ( vtor<=3 )
210:         {
211:             /* ratrio */
212:             for (x=y1 ; x<=y2; x++)
213:             {
214:                 if (PTO[x2-3][x]==1)
215:                     { a = x ; }
216:             }
217:             wid = y2 - a ;hig = x2 - x1 ;
218:             if ( wid <= ((1.0/3.0)*hig) )
219:                 { REST = 162 ; }
220:             else
221:                 { REST = 184 ; }
222:         }
223:         else
224:             { REST = -11 ; }
225:     }
226:     else
227:     {
228:         REST = 188 ;
229:     }
230: }
231: else
232: if (hsty>1)
233: {
234:     /* in head */
235:     /* crossbase */
236:     VECTOR = 0 ;
237:     for (j=y1 ; j<=y2 ; j++)
238:     {
239:         if ( PTO[x2-4][j]==1 )
240:             { VECTOR++ ; }
241:         else
242:             { }
243:     }
244:     if ( VECTOR<=2 )
245:         { REST = 192 ; }
246:     else
247:         { REST = 186 ; }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 248: }
 249: else
 250: การพิมพ์ใดๆทั้งสิ้น { ถ้าท่านมีให้ข้อเสนอแนะ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class1.c

Page 6

8/14/1988 17:59:02

```

251:         REST = -11 ;
252:     }
253: }
254: }
255: }
256:
257: mlB ()
258: {
259:     int i,j ;
260:     int istart=0,jstart=0,istop=0,jstop=0 ;
261:     int angle=0,crsl=0 ;
262:     gotoXY (3,15) ;printf ("MIDLE HEAD") ;
263:     border () ;center_g () ;zone4 () ; /* devide 4 zone */
264:     dip02 () ;
265:     if (end4==1 && end2==1)
266:     {
267:         /* crossleft */
268:         crsl = 0 ;
269:         for (j=y1 ; j<=JH[0] ; j++)
270:         {
271:             if (PT0[ IH[0] ][j]==1)
272:                 { crsl++ ; }
273:             else
274:                 { }
275:         }
276:         if (crsl==1)
277:             { REST = 175 ; }
278:         else
279:             { REST = 197 ; }
280:     }
281:     else
282:     {
283:         if (end1==1 || end3==1)
284:         {
285:             /* head style */
286:             hsty = 0 ;
287:             for (i=IH[0] ; i<=x2 ; i++)
288:             {
289:                 if (PT0[i][ JH[0] ]==1)
290:                     { hsty++ ; }
291:             }
292:             if (hsty==1)
293:                 { REST = 166 ; }
294:             else
295:                 if (hsty>1)
296:                     { REST = 202 ; }
297:             else
298:                 { REST = -11 ; }
299:         }
300:     }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้สำหรับกรเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆก็ตาม หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูงและต้องอภัยถึงเจ้าของเอกสารทุกครั้งที่มีคนนำไปใช้

Listing of class1.c

Page 7

8/14/1988 17:59:02

```

301:      {
302:          if (end4==1)
303:          {
304:              /* head style */
305:              hsty = 0 ;
306:              for (i=IH[0] ; i<=x2 ; i++)
307:              {
308:                  if (PTO[i][ JH[0] ]==1)
309:                      { hsty++ ; }
310:              }
311:              if (hsty==1)
312:              {
313:                  /* crossh */
314:                  vtor = 0 ;
315:                  for (j=JH[0] ; j<=y2 ; j++)
316:                  {
317:                      if ( PTO[IH[0]][j]==1 )
318:                          { vtor++ ; }
319:                      else
320:                          { }
321:                  }
322:                  if (vtor==2)
323:                      { REST = 163 ; }
324:                  else
325:                      { REST = 175 ; }
326:              }
327:              else
328:              if (hsty>1)
329:              {
330:                  /* crossup */
331:                  VECTOR = 0 ;
332:                  for (j=y1 ; j<=y2 ; j++)
333:                  {
334:                      if ( PTO[x1+3][j]==1 )
335:                          { VECTOR++ ; }
336:                      else
337:                          { }
338:                  }
339:                  if ( VECTOR<=2 )
340:                      { REST = 178 ; }
341:                  else
342:                      { REST = 179 ; }
343:              }
344:              else
345:              {
346:                  REST = -11 ;
347:              }
348:          }
349:      } else
350:      {

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class1.c

8/14/1988 17:59:02

Page 8

```

351:      /* head style */
352:      hsty = 0 ;
353:      for (i=IH[0] ; i<=x2 ; i++)
354:      {
355:          if (PTO[i][ JH[0] ]==1)
356:              { hsty++ ; }
357:      }
358:      if (hsty==1)
359:      {
360:          /* tail angle */
361:          irstart = (IE[0] + 10) ; jstart = JE[0] ;
362:          while (PTO[irstart][jstart] != 1)
363:          {
364:              jstart-- ;
365:          }
366:          jstop = jstart ;
367:          istop = irstart ;
368:          angle = abs (JE[0] - jstop) ;
369:          if (angle <= 3)
370:          {
371:              /* crossbase */
372:              VECTOR = 0 ;
373:              for (j=y1 ; j<=y2 ; j++)
374:              {
375:                  if ( PTO[x2-4][j]==1 )
376:                      { VECTOR++ ; }
377:                  else
378:                      { }
379:              }
380:              if ( VECTOR<=2 )
381:                  { REST = 185 ; }
382:              else
383:                  { REST = 189 ; }
384:          }
385:          else
386:          {
387:              /* crossh */
388:              vtor = 0 ;
389:              for (j=JH[0] ; j<=y2 ; j++)
390:              {
391:                  if ( PTO[IH[0]][j]==1 )
392:                      { vtor++ ; }
393:                  else
394:                      { }
395:              }
396:              if (vtor<=2)
397:                  { REST = 168 ; }
398:              else
399:                  { REST = 169 ; }
400:          }

```

กรุณาเป็นเอกสารที่สงวนไว้สำหรับกรณีสืบค้นเอกสารเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

กรุณาใส่ทุกสิ่งทุกอย่างที่คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class1.c

Page 9

8/14/1988 17:59:02

```

401:         }
402:         else
403:         if (hsty>1)
404:         {
405:             REST = 187 ;
406:         }
407:     }
408: }
409: }
410: istart=0 ;jstart=0 ;istop=0 ;jstop=0 ;
411: angle=0 ;crsl=0 ;
412: }
413:
414: mlC ()
415: {
416:     int i,j ;
417:     gotoXY (3,15) ;printf ("LOWER HEAD") ;
418:     border () ;center_g () ;zone4 () ; /* devide 4 zone */
419:     dip02 () ;
420:     if ((end1==1 || end3==1) && end2==1)
421:     {
422:         REST = 199 ;
423:     }
424:     else
425:     {
426:         if (end4==1)
427:         {
428:             if (end1==1 || end3==1)
429:             {
430:                 REST = 195 ;
431:             }
432:             else
433:             {
434:                 /* crossh */
435:                 vtor = 0 ;
436:                 for (i=x1 ; i<=IH[0] ; i++)
437:                 {
438:                     if (PT0[i][ JH[0] ]==1)
439:                     { vtor++ ; }
440:                     else
441:                     { }
442:                 }
443:                 if (vtor==1 || vtor==3)
444:                 { REST = 190 ; }
445:                 else
446:                 { REST = 180 ; }
447:             }
448:         }
449:     }
450: }

```

นี่เป็นเอกสารที่วางไว้สำหรับครู ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ถ้ากรณใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

Listing of class1.c

Page 10

8/14/1988 17:59:02

```

451:         if (end1==1 || end3==1)
452:         {
453:             /* crossh */
454:             vtor = 0 ;
455:             for (i=x1 ; i<=IH[0] ; i++)
456:             {
457:                 if (PTO[i][ JH[0] ]==1)
458:                     { vtor++ ; }
459:                 else
460:                     { }
461:             }
462:             if (vtor==1)
463:             {
464:                 /* in head */
465:                 if ((x2-x1) > 35)
466:                 {
467:                     REST = 212 ;
468:                 }
469:                 else
470:                 {
471:                     REST = 208 ;
472:                 }
473:             }
474:             else
475:             {
476:                 /* out head */
477:                 /* cross_R */
478:                 vtor_R = 0 ;
479:                 for (j=JH[0] ; j<=y2 ; j++)
480:                 {
481:                     if (PTO[ IH[0] ][j]==1)
482:                         { vtor_R++ ; }
483:                     else
484:                         { }
485:                 }
486:                 if (vtor_R==1)
487:                     { REST = 196 ; }
488:                 else
489:                     { REST = 195 ; }
490:             }
491:         }
492:         else
493:         {
494:             if ((x2-x1) > 35)
495:             {
496:                 REST = 210 ;
497:             }
498:             else
499:             {
500:                 REST = 193 ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับสารนิพนธ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 หากท่านใดต้องการข้อมูลเพิ่มเติม กรุณาติดต่อขอข้อมูลจากศูนย์บริการข้อมูลและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class1.c

Page 11

8/14/1988 17:59:02

```

501:         }
502:     }
503: }
504: }
505: }
506:
507: zone4 ()
508: {
509:     int i,j ;
510:     /* Z1 */
511:     for (i=x1 ; i<=xm ; i++)
512:     {
513:         for (j=y1 ; j<=ym ; j++)
514:         {
515:             if (PT2[i][j]==1)
516:             {
517:                 endl++ ;
518:             }
519:             else
520:             if (PT2[i][j]==20)
521:             {
522:                 hed1++ ;
523:             }
524:         }
525:     }
526:     /* Z2 */
527:     for (i=x1 ; i<=xm ; i++)
528:     {
529:         for (j=ym+1 ; j<=y2 ; j++)
530:         {
531:             if (PT2[i][j]==1)
532:             {
533:                 end2++ ;
534:             }
535:             else
536:             if (PT2[i][j]==20)
537:             {
538:                 hed2++ ;
539:             }
540:         }
541:     }
542:     /* Z3 */
543:     for (i=xm+1 ; i<=x2 ; i++)
544:     {
545:         for (j=y1 ; j<=ym ; j++)
546:         {
547:             if (PT2[i][j]==1)
548:             {
549:                 end3++ ;
550:             }

```

สารนี้เป็นเอกสารที่สงวนไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 หารณีใดๆทั้งสิ้น อีก ห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class1.c

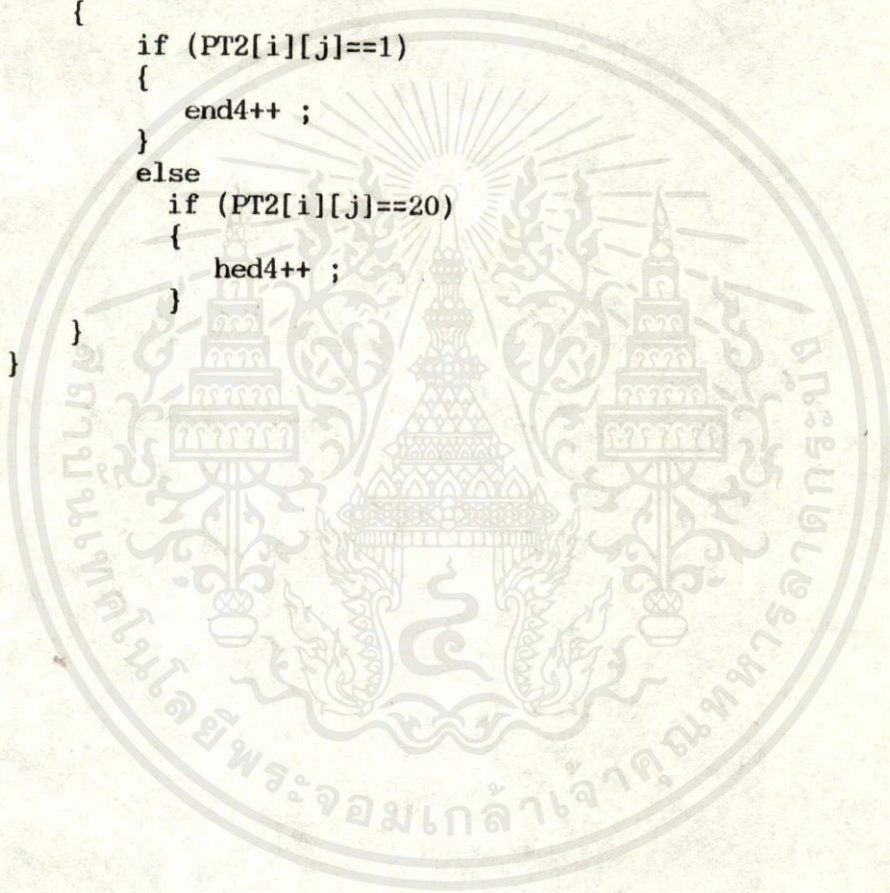
Page 12

8/14/1988 17:59:02

```

551:         else
552:             if (PT2[i][j]==20)
553:             {
554:                 hed3++ ;
555:             }
556:         }
557:     }
558:     /* Z4 */
559:     for (i=xm+1 ; i<=x2 ; i++)
560:     {
561:         for (j=ym+1 ; j<=y2 ; j++)
562:         {
563:             if (PT2[i][j]==1)
564:             {
565:                 end4++ ;
566:             }
567:             else
568:             if (PT2[i][j]==20)
569:             {
570:                 hed4++ ;
571:             }
572:         }
573:     }
574: }
575:
576:
577:

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class2.c

Page 1

8/14/1988 18:08:42

```

1: /* classifier for TWO head pattern */
2: extern unsigned char PT0[64][64] ; /* end,branch,head */
3: extern unsigned char PT2[64][64] ;
4: extern unsigned char IH[ 5],JH[ 5] ;
5: extern int VECTOR ; /* code-vector for CLASS_0,2 */
6: extern int xm,ym ;
7: extern int x1,x2,y1,y2,A,B ;
8: extern int end1,con1,hed1 ;
9: extern int end2,con2,hed2 ;
10: extern int end3,con3,hed3 ;
11: extern int end4,con4,hed4 ;
12: extern int REST ;
13:
14: class_2 () /* [two head] */
15: {
16:     int i,j,k,sw ;
17:     border () ;center_g () ;dip02 () ;
18:     /* fine NC8 in each 4-zone */
19:     /* Z1 */
20:     for (i=x1 ; i<=xm ; i++)
21:     {
22:         for (j=y1 ; j<=ym ; j++)
23:         {
24:             if (PT2[i][j]==1)
25:             {
26:                 end1++ ;
27:             }
28:             else
29:                 if (PT2[i][j]==20)
30:                 {
31:                     hed1++ ;
32:                 }
33:         }
34:     }
35:     /* Z2 */
36:     for (i=x1 ; i<=xm ; i++)
37:     {
38:         for (j=ym+1 ; j<=y2 ; j++)
39:         {
40:             if (PT2[i][j]==1)
41:             {
42:                 end2++ ;
43:             }
44:             else
45:                 if (PT2[i][j]==20)
46:                 {
47:                     hed2++ ;
48:                 }
49:         }
50:     }

```

Listing of class2.c

Page 2

8/14/1988 18:08:42

```

51:  /* Z3 */
52:  for (i=xm+1 ; i<=x2 ; i++)
53:  {
54:      for (j=y1 ; j<=ym ; j++)
55:      {
56:          if (PT2[i][j]==1)
57:          {
58:              end3++ ;
59:          }
60:          else
61:              if (PT2[i][j]==20)
62:              {
63:                  hed3++ ;
64:              }
65:      }
66:  }
67:  /* Z4 */
68:  for (i=xm+1 ; i<=x2 ; i++)
69:  {
70:      for (j=ym+1 ; j<=y2 ; j++)
71:      {
72:          if (PT2[i][j]==1)
73:          {
74:              end4++ ;
75:          }
76:          else
77:              if (PT2[i][j]==20)
78:              {
79:                  hed4++ ;
80:              }
81:      }
82:  }
83:  cat2 ( ) ;
84:  hed1 = 0 ;hed2 = 0 ;hed3 = 0 ;hed4 = 0 ;
85:  end1 = 0 ;end2 = 0 ;end3 = 0 ;end4 = 0 ;
86: }
87:
88: cat2 ( )
89: {
90:     int i,j ;
91:     int h3_a=0,h3_b=0,h4_a=0,h4_b=0 ;
92:     int x3,x4,dist ;
93:     if (hed1==2) /*-----*/
94:     {
95:         REST = 198 ;
96:     }
97:     else
98:         if (hed1==1 && hed2==1) /*-----*/
99:         {
100:             if (end4==1)

```

Listing of class2.c

Page 3

8/14/1988 18:08:42

```

101:         { REST = 200 ; }
102:     else
103:         { REST = 198 ; }
104:     }
105: else
106: if (hed1==1 && hed3==1) /*-----*/
107: {
108:     if (end2==0)
109:     {
110:         VECTOR = 0 ;
111:         for (j=y1 ; j<=y2 ; j++)
112:         {
113:             if ( PTO[IH[1]][j]==1 )
114:                 { VECTOR++ ; }
115:             else
116:                 { }
117:         }
118:         if ( VECTOR<5 )
119:             { REST = 172 ; }
120:         else
121:             { REST = 173 ; }
122:     }
123: else
124: {
125:     if (end4==1)
126:     {
127:         REST = 204 ;
128:     }
129: else
130: {
131:     /* subhead3 */
132:     h3_a = 0 ; h3_b = 0 ;
133:     x3 = xm + ((x2 - xm) / 2) ;
134:     for (i=xm ; i<=x3 ; i++)
135:     {
136:         for (j=y1 ; j<=ym ; j++)
137:         {
138:             if (PT2[i][j]==20)
139:                 { h3_a++ ; }
140:         }
141:     }
142:     for (i=x3 ; i<x2 ; i++)
143:     {
144:         for (j=y1 ; j<=ym ; j++)
145:         {
146:             if (PT2[i][j]==20)
147:                 { h3_b++ ; }
148:         }
149:     }
150:     if (h3_a==1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ได้อิงค่าทั้งสั้น อีกทั้งห้ามเผยแพร่โดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรณีนำไปใช้

Listing of class2.c

Page 4

8/14/1988 18:08:42

```

151:         {
152:             /* hed distance */
153:             dist = abs(JH[0]-JH[1]) ;
154:             if (dist<=3)
155:                 { REST = 203 ; }
156:             else
157:                 { REST = 176 ; }
158:         }
159:     else
160:     if (h3_b==1)
161:     {
162:         /* cross */
163:         VECTOR = 0 ;
164:         for (j=y1 ; j<=y2 ; j++)
165:         {
166:             if ( PTO[IH[0]][j]==1 )
167:                 { VECTOR++ ; }
168:             else
169:                 { }
170:         }
171:         if ( VECTOR>3 )
172:             { REST = 164 ; }
173:         else
174:             {
175:                 if (end2 != 0)
176:                     { REST = 191 ; }
177:                 else
178:                     { REST = 173 ; }
179:             }
180:     }
181:     else
182:     {
183:         gotoXY (37,16) ;printf ("%d..%d",h3_a,h3_b) ;
184:     }
185: }
186: }
187: }
188: else
189: if (hed1==1 && hed4==1) /*-----*/
190: {
191:     if (end2==1)
192:     {
193:         if (end4==1)
194:         {
195:             REST = 171 ;
196:         }
197:         else
198:         {
199:             /* subhead4 */
200:             h4_a = 0 ;h4_b = 0 ;

```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแบบลงมือหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class2.c

Page 5

8/14/1988 18:08:42

```

201:         x4 = ym + ((y2 - ym) / 2) ;
202:         for (i=xm ; i<=x2 ; i++)
203:         {
204:             for (j=ym ; j<=x4 ; j++)
205:             {
206:                 if (PT2[i][j]==20)
207:                     { h4_a++ ; }
208:             }
209:         }
210:         for (i=xm ; i<=x2 ; i++)
211:         {
212:             for (j=x4 ; j<=y2 ; j++)
213:             {
214:                 if (PT2[i][j]==20)
215:                     { h4_b++ ; }
216:             }
217:         }
218:         if (h4_a==1)
219:             { REST = 176 ; }
220:         else
221:         if (h4_b==1)
222:             { REST = 183 ; }
223:         else
224:             { gotoXY (37,16) ;printf ("%d..%d",h4_a,h4_b) ; }
225:         }
226:     }
227: else
228: {
229:     if (end1==1 || end3==1)
230:     {
231:         if (end2==1)
232:             { REST = 209 ; }
233:         else
234:             { REST = 167 ; }
235:     }
236:     else
237:     {
238:         REST = 211 ;
239:     }
240: }
241: }
242: else
243: if (hed2==1 && hed3==1) /*-----*/
244: {
245:     REST = 201 ;
246: }
247: else
248: if (hed3==2) /*-----*/
249: {
250:     if (end2==1)

```

สารบัญ (เอกสารการสอน) ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 การทำซ้ำโดยไม่ได้รับอนุญาตถือว่าผิดและต้องแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class2.c

Page 6

8/14/1988 18:08:42

```

251:  {
252:      /* subhead3 */
253:      h3_a = 0 ;h3_b = 0 ;
254:      x3 = xm + ((x2 - xm) / 2) ;
255:      for (i=xm ; i<=x3 ; i++)
256:      {
257:          for (j=y1 ; j<=ym ; j++)
258:          {
259:              if (PT2[i][j]==20)
260:                  { h3_a++ ; }
261:          }
262:      }
263:      for (i=x3 ; i<x2 ; i++)
264:      {
265:          for (j=y1 ; j<=ym ; j++)
266:          {
267:              if (PT2[i][j]==20)
268:                  { h3_b++ ; }
269:          }
270:      }
271:      if (h3_a==1)
272:          { REST = 176 ; }
273:      else
274:          if (h3_b==1)
275:              { REST = 170 ; }
276:          else
277:              { gotoXY (37,16) ;printf ("%d..%d",h3_a,h3_b) ; }
278:      }
279:      else
280:      {
281:          /* cross */
282:          VECTOR = 0 ;
283:          for (j=y1 ; j<=y2 ; j++)
284:          {
285:              if ( PTO[IH[1]][j]==1 )
286:                  { VECTOR++ ; }
287:              else
288:                  { }
289:          }
290:          if ( VECTOR<5 )
291:              { REST = 172 ; }
292:          else
293:              { REST = 173 ; }
294:      }
295:      }
296:      else
297:          if (hed3==1 && hed4==1) /*-----*/
298:          {
299:              if (end1==1)
300:              {

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class2.c

Page 7

8/14/1988 18:08:42

```

301:      /* subhead3 */
302:      h3_a = 0 ;h3_b = 0 ;
303:      x3 = xm + ((x2 - xm) / 2) ;
304:      for (i=xm ; i<=x3 ; i++)
305:      {
306:          for (j=y1 ; j<=ym ; j++)
307:          {
308:              if (PT2[i][j]==20)
309:                  { h3_a++ ; }
310:          }
311:      }
312:      for (i=x3 ; i<x2 ; i++)
313:      {
314:          for (j=y1 ; j<=ym ; j++)
315:          {
316:              if (PT2[i][j]==20)
317:                  { h3_b++ ; }
318:          }
319:      }
320:      if (h3_a==1)
321:      {
322:          if (end1==1 || end3==1)
323:          {
324:              if (end2==1)
325:                  { REST = 209 ; }
326:              else
327:                  { REST = 167 ; }
328:          }
329:          else
330:          {
331:              REST = 211 ;
332:          }
333:      }
334:      else
335:      if (h3_b==1)
336:          { REST = 209 ; }
337:      else
338:          { gotoXY (37,16) ;printf ("%d.%.d",h3_a,h3_b) ; }
339:      }
340:      else
341:      {
342:          if (end4==1)
343:          {
344:              REST = 171 ;
345:          }
346:          else
347:          {
348:              /*subhead4 */
349:              h4_a = 0 ;h4_b = 0 ;
350:              x4 = ym + ((y2 - ym) / 2) ;

```

Listing of class2.c

Page 8

8/14/1988 18:08:42

```

351:         for (i=xm ; i<=x2 ; i++)
352:         {
353:             for (j=ym ; j<=x4 ; j++)
354:             {
355:                 if (PT2[i][j]==20)
356:                     { h4_a++ ; }
357:             }
358:         }
359:         for (i=xm ; i<=x2 ; i++)
360:         {
361:             for (j=x4 ; j<=y2 ; j++)
362:             {
363:                 if (PT2[i][j]==20)
364:                     { h4_b++ ; }
365:             }
366:         }
367:         if (h4_b==1)
368:             { REST = 177 ; }
369:         else
370:             if (h4_a==1)
371:             {
372:                 /*subhead3 */
373:                 h3_a = 0 ; h3_b = 0 ;
374:                 x3 = xm + ((x2 - xm) / 2) - 2 ;
375:                 for (i=xm ; i<=x3 ; i++)
376:                 {
377:                     for (j=y1 ; j<=ym ; j++)
378:                     {
379:                         if (PT2[i][j]==20)
380:                             { h3_a++ ; }
381:                     }
382:                 }
383:                 for (i=x3 ; i<x2 ; i++)
384:                 {
385:                     for (j=y1 ; j<=ym ; j++)
386:                     {
387:                         if (PT2[i][j]==20)
388:                             { h3_b++ ; }
389:                     }
390:                 }
391:                 if (h3_a==1)
392:                     { REST = 176 ; }
393:                 else
394:                     if (h3_b==1)
395:                         { REST = 170 ; }
396:                 else
397:                     { gotoXY (37,16) ;printf ("%d..%d",h3_a,h3_b) ; }
398:             }
399:         else
400:             { gotoXY (37,16) ;printf ("%d..%d",h4_a,h4_b) ; }

```

Listing of class2.c

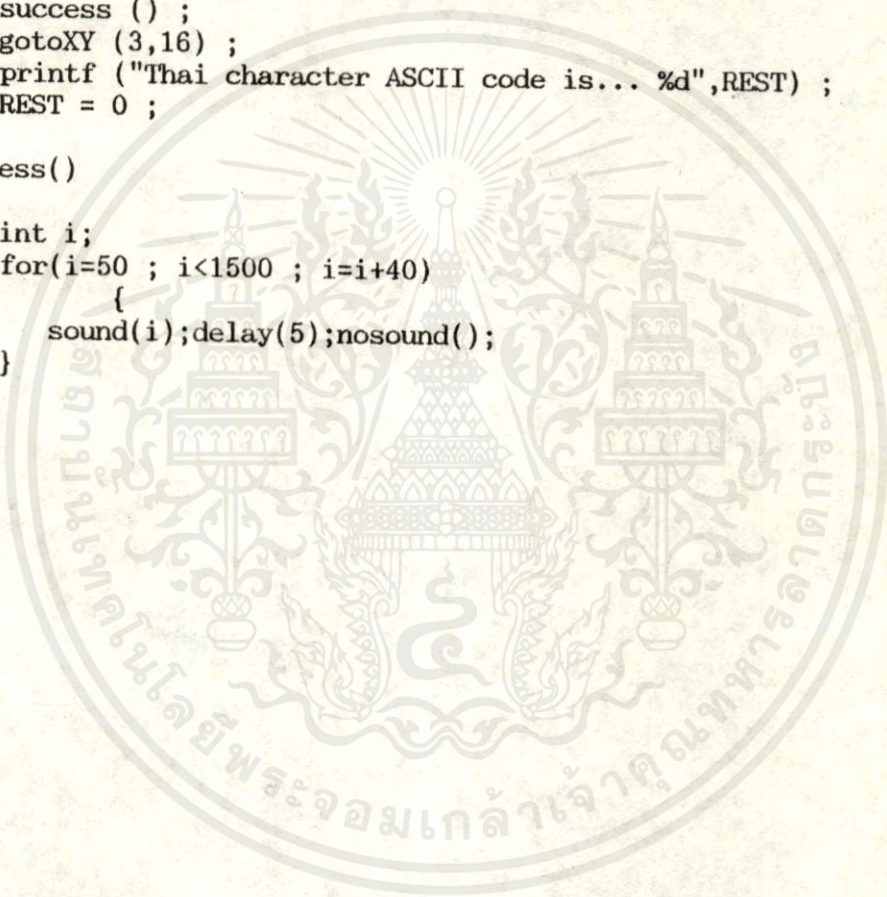
Page 9

8/14/1988 18:08:42

```

401:     }
402:   }
403: }
404: /*-----*/
405: good () ;
406: h3_a = 0 ;h3_b = 0 ;h4_a = 0 ;h4_b = 0 ;
407: x3 = 0 ;x4 = 0 ;dist = 0 ;VECTOR = 0 ;
408: }
409:
410: good ()
411: {
412:   success () ;
413:   gotoXY (3,16) ;
414:   printf ("Thai character ASCII code is... %d",REST) ;
415:   REST = 0 ;
416: }
417: success()
418: {
419:   int i;
420:   for(i=50 ; i<1500 ; i=i+40)
421:   {
422:     sound(i);delay(5);nosound();
423:   }
424: }
425:
426:

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 *ไม่ว่ากรณีใดๆก็ตาม อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of class3.c

Page 1

8/14/1988 18:16:44

```
1: /* classifier for THREE head pattern */
2: extern int REST ;
3:
4: class_3 ()
5: {
6:     REST = 174 ;
7:     good () ;
8: }
9:
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of ibm.asm

Page 1

8/14/1988 18:18:50

```

1: PGROUP   GROUP   PROG
2: PROG     SEGMENT  BYTE PUBLIC 'PROG'
3:          PUBLIC   _EGA,_NORMAL,_CLS,_RS_232,_INITPT,_SEND,
4:          ASSUME   CS:PGROUP
5:          mode    equ    [bp+4]
6:          esc     equ    27
7:          LF      equ    0ah
8:          CR      equ    0dh
9: ;-----
10: _EGA     PROC     NEAR
11:          push    ds
12:          PUSH    AX
13:          MOV     AX,0010H
14:          INT     10H
15:          POP     AX
16:          pop     ds
17:          RET
18: _EGA     ENDP
19:
20:          PUBLIC  _NORMAL
21: _NORMAL  PROC     NEAR
22:          PUSH    DS
23:          PUSH    AX
24:          MOV     AX,03H
25:          INT     10H
26:          POP     AX
27:          POP     DS
28:          RET
29: _NORMAL  ENDP
30:
31: ;-----
32: ;          CLS ( ) ;
33: ;-----
34:          PUBLIC  _CLS
35: _CLS     PROC     NEAR
36:          PUSH    BX
37:          PUSH    CX
38:          PUSH    DX
39:          PUSH    AX
40:          MOV     BH,00H
41:          MOV     AX,0600H
42:          MOV     CX,00H
43:          MOV     DX,184FH
44:          INT     10H
45:          POP     AX
46:          POP     DX
47:          POP     CX
48:          POP     BX
49:          RET
50: _CLS     ENDP

```

Listing of ibm.asm

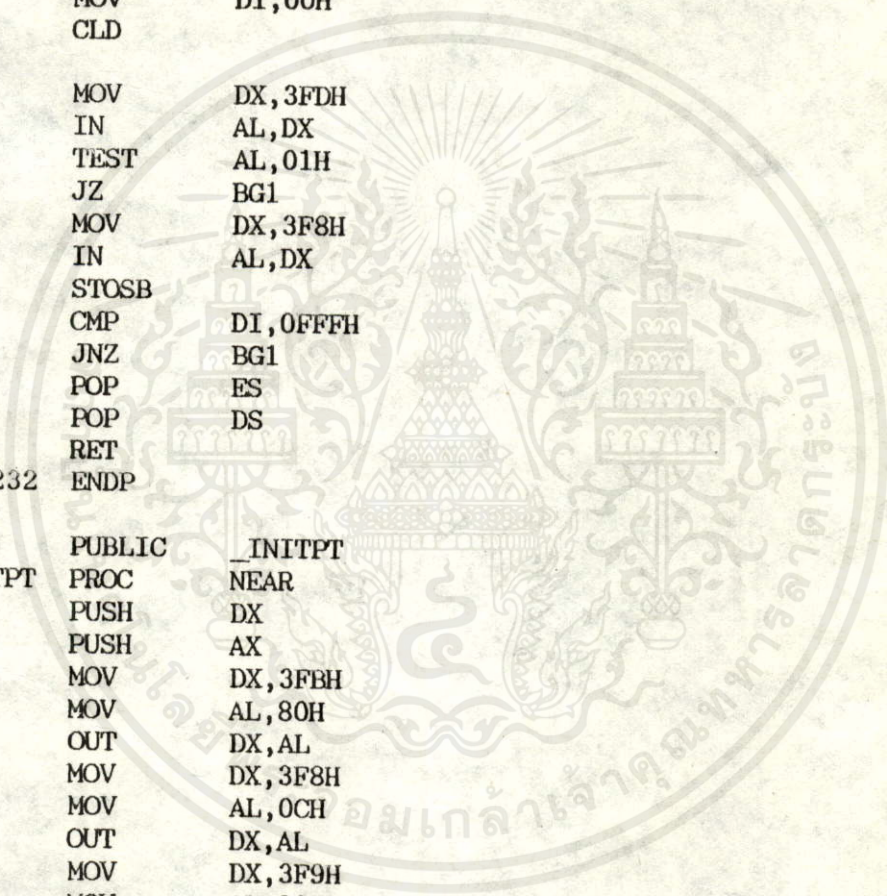
Page 2

8/14/1988 18:18:50

```

51:
52: ;-----
53: ;      RS_232 ( ) ;
54: ;-----
55:      PUBLIC      _RS_232
56: _RS_232 PROC      NEAR
57:      PUSH        DS
58:      PUSH        ES
59:      call        CHKCODE
60:      MOV         AX,7000H
61:      MOV         ES,AX
62:      MOV         DI,00H
63:      CLD
64: BG1:
65:      MOV         DX,3FDH
66:      IN          AL,DX
67:      TEST        AL,01H
68:      JZ          BG1
69:      MOV         DX,3F8H
70:      IN          AL,DX
71:      STOSB
72:      CMP         DI,0FFFH
73:      JNZ        BG1
74:      POP         ES
75:      POP         DS
76:      RET
77: _RS_232 ENDP
78:
79:      PUBLIC      _INITPT
80: _INITPT PROC      NEAR
81:      PUSH        DX
82:      PUSH        AX
83:      MOV         DX,3FBH
84:      MOV         AL,80H
85:      OUT         DX,AL
86:      MOV         DX,3F8H
87:      MOV         AL,0CH
88:      OUT         DX,AL
89:      MOV         DX,3F9H
90:      MOV         AL,00H
91:      OUT         DX,AL
92:      MOV         DX,3FBH
93:      MOV         AL,07H
94:      OUT         DX,AL
95:      MOV         DX,3FCH
96:      MOV         AL,03H
97:      OUT         DX,AL
98:      MOV         DX,3F9H
99:      MOV         AL,00H
100:     OUT         DX,AL

```



Listing of ibm.asm

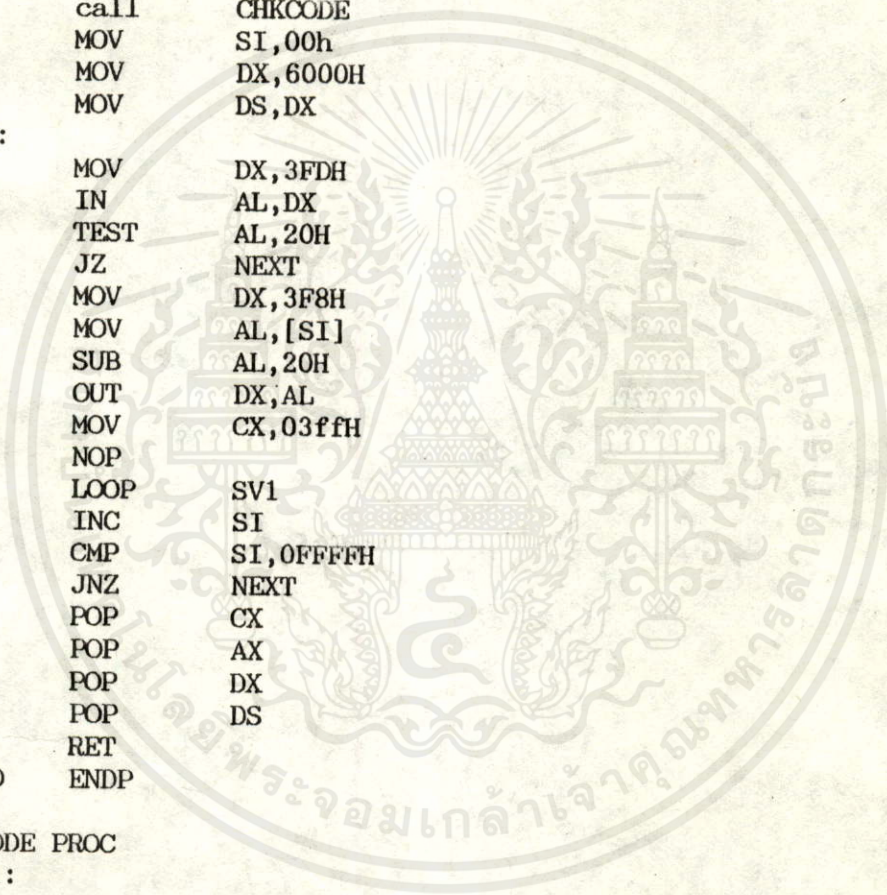
Page 3

8/14/1988 18:18:50

```

101:      POP      AX
102:      POP      DX
103:      ret
104:  _INITPT  ENDP
105:
106:      PUBLIC   _SEND
107:  _SEND  PROC   NEAR
108:      PUSH    DS
109:      PUSH    DX
110:      PUSH    AX
111:      PUSH    CX
112:      call    CHKCODE
113:      MOV     SI,00h
114:      MOV     DX,6000H
115:      MOV     DS,DX
116: NEXT:
117:      MOV     DX,3FDH
118:      IN      AL,DX
119:      TEST    AL,20H
120:      JZ      NEXT
121:      MOV     DX,3F8H
122:      MOV     AL,[SI]
123:      SUB     AL,20H
124:      OUT     DX,AL
125:      MOV     CX,03fh
126: SV1:   NOP
127:      LOOP   SV1
128:      INC    SI
129:      CMP    SI,0FFFFH
130:      JNZ    NEXT
131:      POP    CX
132:      POP    AX
133:      POP    DX
134:      POP    DS
135:      RET
136:  _SEND  ENDP
137:
138: CHKCODE PROC
139: WAIT1:
140:      MOV     DX,3FDH
141:      IN      AL,DX
142:      TEST    AL,20H
143:      JZ      WAIT1
144:      MOV     DX,3F8H
145:      MOV     AL,02H
146:      OUT     DX,AL
147: WAIT2:
148:      MOV     DX,3FDH
149:      IN      AL,DX
150:      TEST    AL,01

```



เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้ภายในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆก็ตาม หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูงและต้องอภัยถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Listing of ibm.asm

Page 4

8/14/1988 18:18:50

```

151:          JZ          WAIT2
152:          MOV         DX,3F8H
153:          IN           AL,DX
154:          CMP         AL,03
155:          JNZ         WAIT1
156:          RET
157: CHKCODE ENDP
158:
159: PROG      ENDS
160:          END
161:
162:

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้