การรู้จำตัวอักษรคัดลายมือภาษาไทย

(HAND WRITTEN THAI CHARACTER RECOGNITION)

โดย

นาย นพพร คงดำ

วัน เดือน ปี 1 ตค 2541
เลขทะเบียน 038307
เลขเรียกหนังสือ T3932ๆ 06174ก

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038307

ปีการศึกษา 2539

การรู้จำตัวอักษรคัดลายมือภาษาไทย

(HAND WRITTEN THAI CHARACTER RECOGNITION)

โดย

นาย นพพร ดงดำ

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร. ชม กิ้มปาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2539

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การรู้จำตัวอักษรคัดลายมือภาษาไทย

(HAND WRITTEN THAI CHARACTER RECOGNITION)

ผู้จัดทำ นาย นพพร คงดำ

---------- อาจารย์ ที่ปรึกษา

(รองศาสตราจารย์ ดร. ชม กิ้มปาน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรู้จำตัวอักษรคัดลายมือภาษาไทย

นาย นพพร คงดำ 37013299
อาจารย์ที่ปรึกษา
รองศาสตราจารย์ ดร. ชม กิ้มปาน
ปีการศึกษา 2539

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการนำเสนอวิธีการในการรู้จำตัวอักษรคัดลายมือภาษาไทย โดยการพิจารณาจากลักษณะโครงสร้างของเส้นที่ลากขึ้นเป็นตัวอักษร และทำการออกแบบซอฟแวร์คอมพิวเตอร์ ควบคุมไมโครคอมพิวเตอร์ ให้สามารถวิเคราะห์ คุณสมบัติความสัมพันธ์ลักษณะลายเส้น ระหว่างจุดที่ประกอบขึ้นเป็นตัวอักษร โดยอาศัยหลักการความต่อเนื่องของลายเส้น ลักษณะของการลากเส้น และหลักไวยากรณ์ในการพิจารณาลักษณะโครงสร้างแล้วนำผลการวิเคราะห์ที่ได้ไปเปรียบเทียบกับฐานข้อมูลที่มีอยู่ ถ้าไม่ตรงกับฐานข้อมูลที่มีอยู่ก็จะให้ผู้ใช้ระบุว่าเป็นตัวอักษรอะไร จากนั้นก็นำข้อมูลที่ได้ไปเก็บเพิ่มเติมเพื่อใช้ในการวิเคราะห์ครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HAND WRITTEN THAI CHARACTER RECOGNITION

Nopporn Kongdum S7013299

Associate Professor Dr. Chom Kimpan Advisor

1996

Abstract

This thesis present the Syntactic Description of Handwritten Thai Characters Recognition by designing computer Software for automatically controlled micro-computer to be able analytical properties, the description for relationship drawing Line between point of characters with the principle of trace, connection trail and Syntactic Description of the Grammar . The result of analysis compares with a database and shows on the screen of computer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 1 บทนำ 1

บทที่ 2 ทฤษฎีและหลักการเบื้องต้น 3

2.1 วิธีในการรู้จำตัวอักษรต่างๆ 3

2.1.1 การรู้จำแบบซ้อนทับ 3

2.1.2 การรู้จำโดยการวิเคราะห์โครงสร้าง 3

2.2 ค่าความสัมพันธ์ระหว่างจุด 4

2.3 การตรวจเช็คความหนาของขอบภาพ 8

2.4 การตรวจเช็คความหนาของภาพทั้งหมด 10

บทที่ 3 การคำนวณและการสร้าง 12

3.1 การจัดการล่วงหน้า 12

3.1.1 การกำจัดสัญญาณรบกวน 13

3.1.2 การทำตัวอักษรให้บาง 13

3.1.3 การขจัดส่วนเกิน 15

3.2 การแยกตัวอักษรออกจากภาพประโยค 18

3.3 การหาหัวของตัวอักษร 19

3.4 การหาทิศทางโครงสร้างของตัวอักษร 20

3.5 การจำแนกตัวอักษร 25

บทที่ 4 การทดลองและผลการทดลอง 33

4.1 การทำงานของโปรแกรม 33

4.2 อุปกรณ์ที่ต้องนำมาใช้ในการทำงานของโปรแกรม 33

4.3 การติดตั้งโปรแกรม 33

4.4 ขั้นตอนการใช้งานโปรแกรม 34

บทที่ 5 บทวิจารณ์และสรุป 50

5.1 ปัญหาและข้อจำกัดในการทำงานของระบบ 50

5.2 ข้อเสนอแนะ 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

ภาคผนวก 51
ก. โครงสร้างของ Bitmap File..... 52
ข. รหัสแอสกี้ของอักขระในรหัส สมอ..... 55
ค. โปรแกรมการทำงาน..... 59
กิตติกรรมประกาศ..... 120
หนังสืออ้างอิง..... 121

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตารางหน้าต่างขนาด 3x3 4

2.2 จุดอิสระ 4

2.3 จุดปลาย 5

2.4 จุดต่อเนื่อง 6

2.5 จุดแยก 7

2.6 จุดตัด 7

2.7 จุดภายใน 8

2.8 ความสัมพันธ์ของจุด 8

2.9 ลักษณะของจุดตามสมการ N1 9

2.10 ลักษณะของจุดตามสมการ N3 9

2.11 ลักษณะของจุดตามสมการ N5 9

2.12 ลักษณะของจุดตามสมการ N7 9

2.13 ลักษณะของจุดตามสมการ N0 9

2.14 ลักษณะของจุดตามสมการ Ne 10

2.15 ลักษณะของจุดตามสมการ S 11

2.16 ลักษณะของจุดตามสมการ T 11

3.1 จุดรบกวน 13

3.2 จุดใกล้เคียง 8 ทิศทาง 15

3.3 Chain code 8 ทิศทาง 21

3.4 ทิศทางขึ้นบน 22

3.5 ทิศทางลงล่าง 22

3.6 ทิศทางไปทางขวา 22

3.7 ทิศทางไปซ้าย 23

ก.1 BitMap File Header 52

ก.2 BitMap Information 53

ก.3 BitMap Data 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่ **หน้า**

3.1 flow chart การจัดการล่วงหน้า..... 12
3.2 การตรวจเช็คความหนาจากบนลงล่าง..... 13
3.3 การตรวจเช็คความหนาจากล่างขึ้นบน..... 14
3.4 การตรวจเช็คความหนาจากซ้ายไปขวา..... 14
3.5 การตรวจเช็คความหนาจากขวาไปซ้าย..... 14
3.6 ภาพก่อนขจัดส่วนเกิน..... 16
3.7 ภาพหลังขจัดส่วนเกิน..... 16
3.8 flow chart การรู้จำตัวอักษร..... 17
3.9 แสดงจุดเริ่มต้นในการหาหัว..... 19
3.10 แสดงการเกิดจุดหัว..... 19
3.11 แสดงลักษณะของหัว..... 20
3.12 แสดงจุดเริ่มหาทิศทางสำหรับอักษรไม่มีหัว..... 20
3.13 แสดงจุดเริ่มหาทิศทางสำหรับอักษรมีหัว..... 21
3.14 แสดงการหาทิศทางของตัวอักษร ก และ ข..... 23
3.15 ตัวอย่างทิศทางของตัวอักษร..... 24
3.16 การแบ่งตัวอักษรตามหัว..... 27
3.17 ทิศทางของตัวอักษรไม่มีหัว..... 28
3.18 ทิศทางของอักษร 1 หัวและเป็นหัวชนิดซ้าย..... 29
3.19 ทิศทางของอักษร 1 หัวและเป็นหัวชนิดขวา..... 30
3.20 ทิศทางของอักษร 2 หัวและหัวเริ่มต้นเป็นชนิดซ้าย..... 31
3.21 ทิศทางของอักษร 2 หัวและหัวเริ่มต้นเป็นชนิดขวา..... 32
4.1 แสดงการทำงานของโปรแกรมเมื่อเริ่มต้นทำงาน..... 34
4.2 แสดงการพิมพ์ชื่อแฟ้มข้อมูลเพื่อการเปิด..... 35
4.3 แสดงข้อมูลภาพที่ทำการเปิด..... 36
4.4 แสดงข้อมูลภาพภายหลังการทำให้บาง..... 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

รูปที่ **หน้า**

4.5 แสดงข้อมูลภาพภายหลังการขจัดส่วนเกิน..... 38
4.6 แสดงผลการวิเคราะห์ตัวอักษร ก..... 39
4.7 แสดงผลการวิเคราะห์ตัวอักษร จ..... 40
4.8 แสดงผลการวิเคราะห์ตัวอักษร ง..... 41
4.9 แสดงผลการวิเคราะห์ตัวอักษร ช..... 42
4.10 แสดงผลการวิเคราะห์ตัวอักษร ฐ..... 43
4.11 แสดงผลการวิเคราะห์ตัวอักษร ม..... 44
4.12 แสดงผลการวิเคราะห์ตัวอักษร ฟ..... 45
4.13 แสดงผลการวิเคราะห์ตัวอักษร ฆ..... 46
4.14 แสดงผลการวิเคราะห์ตัวอักษร ฌ..... 47
4.15 แสดงผลการวิเคราะห์ตัวอักษร ต..... 48
ก.1 โครงสร้างของ BitMap File..... 52
ก.2 การเก็บภาพแบบ BitMap..... 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

โดยปกติแล้วในการจัดเก็บข้อมูลเอกสารต่างๆด้วยเครื่องคอมพิวเตอร์ จะต้องทำการพิมพ์เข้าไป โดยการใช้คีย์บอร์ดซึ่งก่อให้เกิดความไม่สะดวกและผิดพลาดได้ง่ายสำหรับผู้ที่พิมพ์ดีดไม่ชำนาญหรือ แม้จะพิมพ์ดีดชำนาญแล้วก็ตามเราก็จะต้องมาทำการพิมพ์ข้อมูลเข้าไปใหม่อีกทั้งที่มีข้อมูลอยู่แล้วบนเอกสารก่อให้เกิดการใช้เวลาและบุคลากรไปอย่างไม่คุ้มค่า และจะสิ้นเปลืองเนื้อที่ในการจัดเก็บข้อมูลมาก ถ้ามีการจัดเก็บเป็นแฟ้มข้อมูลรูปภาพ

ในปัจจุบันจึงได้มีการคิดวิธีที่จะนำเอกสาร ซึ่งอยู่ในรูปแบบของตัวพิมพ์หรือ ตัวเขียนที่อยู่บนกระดาษมาทำการจัดเก็บด้วย เครื่องคอมพิวเตอร์ โดยการสแกนผ่านเครื่องตรวจกวาดภาพ (scanner) แล้วนำภาพข้อความที่ได้นั้นไปผ่านกระบวนการรู้จำตัวอักษรเพื่อทำการแปลงภาพตัวอักษรไปเป็นรหัส ASCII ของตัวอักษรนั้น ซึ่งใช้เนื้อที่ในการจัดเก็บน้อยกว่าการจัดเก็บเป็นภาพหลายเท่า

การสอนให้คอมพิวเตอร์สามารถรู้จำตัวอักษรต่างๆได้นั้น ได้มีการศึกษาและทำการวิจัยในเรื่องต่างๆที่เกี่ยวข้องกันมาอย่างแพร่หลาย พร้อมทั้งมีการเรียนรู้และสร้างทฤษฎีของการรู้จำกันอย่างมากมาย ซึ่งในส่วนของการศึกษาที่ผ่านๆมาก็สามารถแบ่งพิจารณาวิธีในการรู้จำจากลักษณะของรูปแบบตัวอักษรที่แตกต่างกัน เช่นการรู้จำอักษรตัวพิมพ์ การรู้จำอักษรลายมือเขียน หรืออาจแบ่งโดยวิธี การพิจารณาจากวิธีในการรู้จำ เช่น การรู้จำโดยวิธีการซ้อนทับ (Matching method) การรู้จำโดยวิธีการพิจารณาโครงสร้างของตัวอักษร (Structure-Analysis) ซึ่งในแต่ละวิธีการก็มีลักษณะเด่นที่สามารถอธิบายลักษณะอักษร หรือรู้จำอักษรแต่ละประเภท และสามารถใช้แก้ปัญญาที่แตกต่างกันออกไป ในส่วนของปริญญานิพนธ์ฉบับนี้เป็นการศึกษาและออกแบบวิธีการรู้จำตัวอักษรคัดลายมือภาษาไทย โดยการพิจารณาจากลักษณะโครงสร้างของตัวอักษร และแบ่งประเภทของตัวอักษรตามจำนวนและตำแหน่งของหัวตัวอักษร จากนั้นก็หาทิศทางในการเขียนตัวอักษรซึ่งเป็นรูปแบบเฉพาะตัวอักษร

ลำดับในการรู้จำ (recognition) ภาพตัวอักษรว่าเป็นตัวอักษรอะไร จะเริ่มจากการตรวจกวาดภาพตัวอักษรนั้น ด้วยเครื่องตรวจกวาดภาพเพื่อจัดเก็บข้อมูลที่ได้ในลักษณะของค่าเลขดิจิตอลแล้วนำค่าดิจิตอลที่ได้มาทำการจัดการล่วงหน้า (pre processing) ก่อนนำข้อมูลที่ได้ไปสู่ขบวนการรู้จำ เมื่อทำการวิเคราะห์ได้แล้วว่าเป็นตัวอักษรอะไรก็จะส่งรหัส ASCII ภาษาไทยของอักษรตัวนั้นแสดงทางจอภาพแล้วจัดเก็บลงแฟ้มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุประสงค์

ปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อเป็นการศึกษาและออกแบบวิธีในการรู้จำตัวอักษรคัดลายมือภาษาไทยโดยการวิเคราะห์จากลักษณะโครงสร้างของตัวอักษร ซึ่งสามารถแบ่งออกได้เป็นแต่ละหัวข้อดังนี้

1. เป็นการศึกษาและออกแบบวิธีในการทำตัวอักษรให้บาง (thinning) เพื่อทำให้ตัวอักษรที่ได้จากเครื่องตรวจกวาดภาพ มีความหนาของลายเส้นเพียงจุดเดียว

2. ศึกษาและออกแบบวิธีการกำจัดส่วนเกินของตัวอักษร ที่ได้จากการทำให้บางเรียบร้อยแล้วเพื่อให้เหลือเฉพาะลักษณะโครงสร้างจริง ๆ ของตัวอักษร

3. ศึกษาและออกแบบวิธีการในการแยกภาพตัวอักษร ออกจากภาพประโยคข้อความพร้อมทั้งการหาตำแหน่งหัว และสถานะหัว (ซ้าย,ขวา) ของตัวอักษรนั้น

4. ศึกษาและออกแบบวิธีการหาทิศทางจากโครงสร้างของตัวอักษรเพื่อเก็บเป็นรูปแบบเฉพาะของแต่ละตัวอักษร

5. นำตำแหน่งหัวและรูปแบบทิศทางที่ได้มาทำการวิเคราะห์ว่าเป็นตัวอักษรอะไร

6. จัดเก็บตัวอักษรที่ผ่านการวิเคราะห์แล้วลงแฟ้มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการเบื้องต้น

2.1 วิธีการในการรู้จำตัวอักษรต่างๆ

2.1.1. การรู้จำแบบซ้อนทับ (Matching Method) วิธีการนี้จะเป็นการพิจารณาถึงความคล้ายคลึงของภาพตัวอักษรที่นำเข้ามากับภาพตัวอักษรที่กำหนดไว้ โดยจะมีการแบ่งออกเป็น 2 ขั้นตอน

1. การแบ่งกลุ่มตัวอักษรแบบหยาบๆ (Rough Classification Stage) ในขั้นตอนนี้จะมีการกำจัดสัญญาณรบกวนต่างๆ และมีการทำให้เบลอ เพื่อให้ภาพที่ได้มีความครบถ้วนมากขึ้นจากนั้นนำตัวอักษรที่ทำให้เบลอแล้วไปจัดแบ่งออกเป็นกลุ่ม โดยวิธีการวัดค่าสัมประสิทธิ์ของความเหมือนของแต่ละรูปแบบแล้วใช้การกระจายแบบคาร์ฮูเนนโลบ (Karhunen Lo'eve Expansion) หารูปแบบมาตราฐานสำหรับเก็บไว้เป็นตัวแทนของอักษรแต่ละตัวและกลุ่มตัวอักษร

2. การแยกตัวอักษรออกจากกลุ่ม(Fine Classification Stage) มีวิธีการอยู่ 2 วิธี คือ การแยกตัวอักษรออกจากกันโดยการซ้อนทับเป็นส่วน (Subpattern matching) หรือการใช้วิธีการสร้างฟังก์ชันการตัดสินใจแบบเชิงเส้น (Linear Decision Function) บนระนาบของไอเกนเวคเตอร์ที่ได้จากการกระจายแบบคาร์ฮูเนนโลบ สำหรับแยกตัวอักษรในแต่ละกลุ่มออกจากกัน

2.1.2. การรู้จำโดยการวิเคราะห์โครงสร้าง (Structural Analysis Method) มีการแบ่งออกเป็น 2 ขั้นตอน เหมือนกับวิธีแรก คือ

1. การทำตัวอักษรให้บางลงก่อนที่จะทำการวิเคราะห์ เพื่อทำการกำจัดสัญญาณรบกวนและสัญญาณปลีกย่อยต่างๆที่ไม่ต้องการออกไปก่อน ตัวอักษรที่ถูกทำให้บางแล้วจะถูกแบ่งออกให้เป็นกลุ่มๆ โดยอาศัยฟังก์ชันการตัดสินใจแบบเชิงเส้นที่ยึดหลักความกว้างและความสูงของตัวอักษร

2. ขั้นตอนนี้จะเป็นการ แบ่งตัวอักษรออกจากกลุ่มโดยการนำคุณสมบัติทางโทโพโลยีมาใช้แบ่งแยกความแตกต่างของตัวอักษรโดยการคำนวณหาความสัมพันธ์ระหว่างจุดต่างๆ บนโครงร่างของตัวอักษร การพิจารณาในส่วนที่เป็นวงจรปิดหรือหัวของตัวอักษร การพิจารณาในส่วนของจุดแยก ,จุดต่อ,จุดตัด และจุดปลาย การพิจารณาลักษณะของลายเส้น เป็นต้น ซึ่งจะใช้ได้ผลดีในอักษรที่มีลักษณะต่างๆ แต่จะไม่ได้ผลดีกับอักษรที่มีลักษณะไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมบูรณ์ เช่น อักษรที่เว้าแหว่ง ขาด หรือมีสัญญาณรบกวน จนทำให้รูปร่างของตัวอักษรเปลี่ยนไปอันจะทำให้การพิจารณาผิดพลาดไปด้วย

2.2 ค่าความสัมพันธ์ระหว่างจุด

การหาความสัมพันธ์ระหว่างจุด ด้วยการหาจำนวนจุดใกล้เคียงของจุดแต่ละจุดบนภาพ โดยการใช้ ตารางหน้าต่างขนาด 3 x 3 กับทุกจุดภาพ

X3	X2	X1
X4	X0	X8
X5	X6	X7

ตาราง ที่ 2.1 ตารางหน้าต่างขนาด 3 x 3

ให้ X0 เป็นจุดใดๆ บนรูปภาพของตัวอักษรที่จะทำการหาค่าความสัมพันธ์ระหว่างจุด และให้จุด X1 , X2 , X3 , X4 , X5 , X6 , X7 และ X8 เป็นจุดข้างเคียงของ X0 ดังแสดงใน ตาราง 2.1

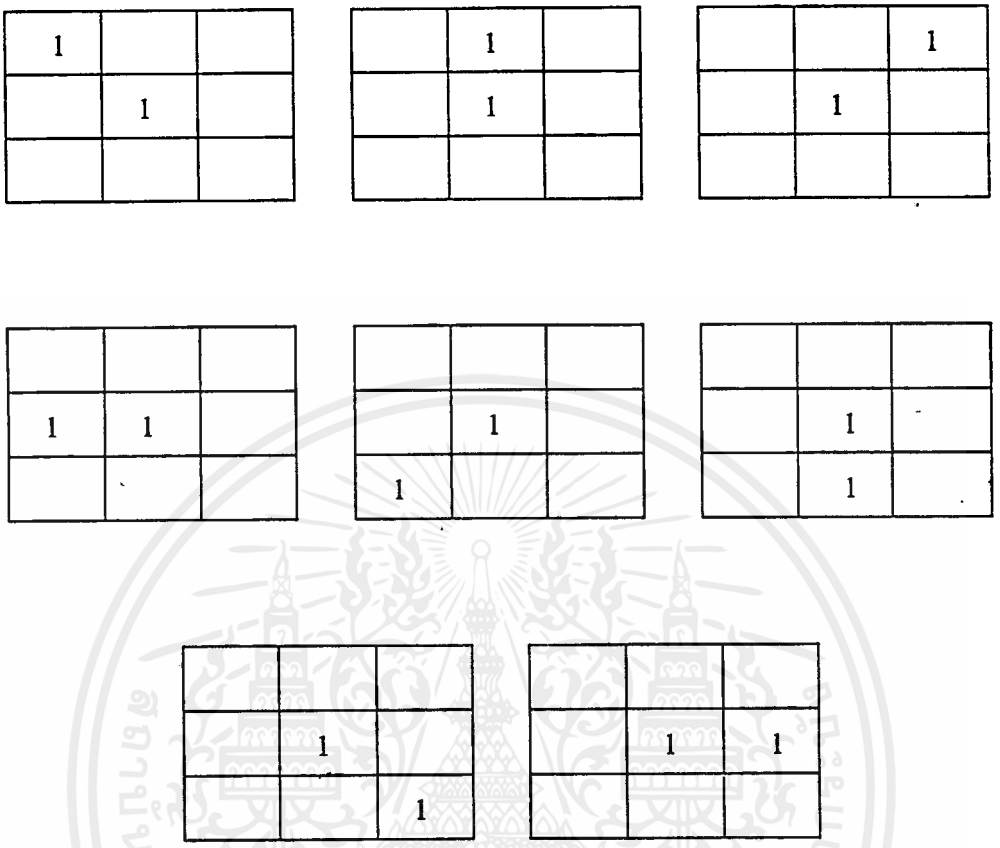
จำนวนจุดใกล้เคียง $N = X1+X2+X3+X4+X5+X6+X7+X8$

	1	

$N = 0$

ตาราง ที่ 2.2 จุดอิสระ

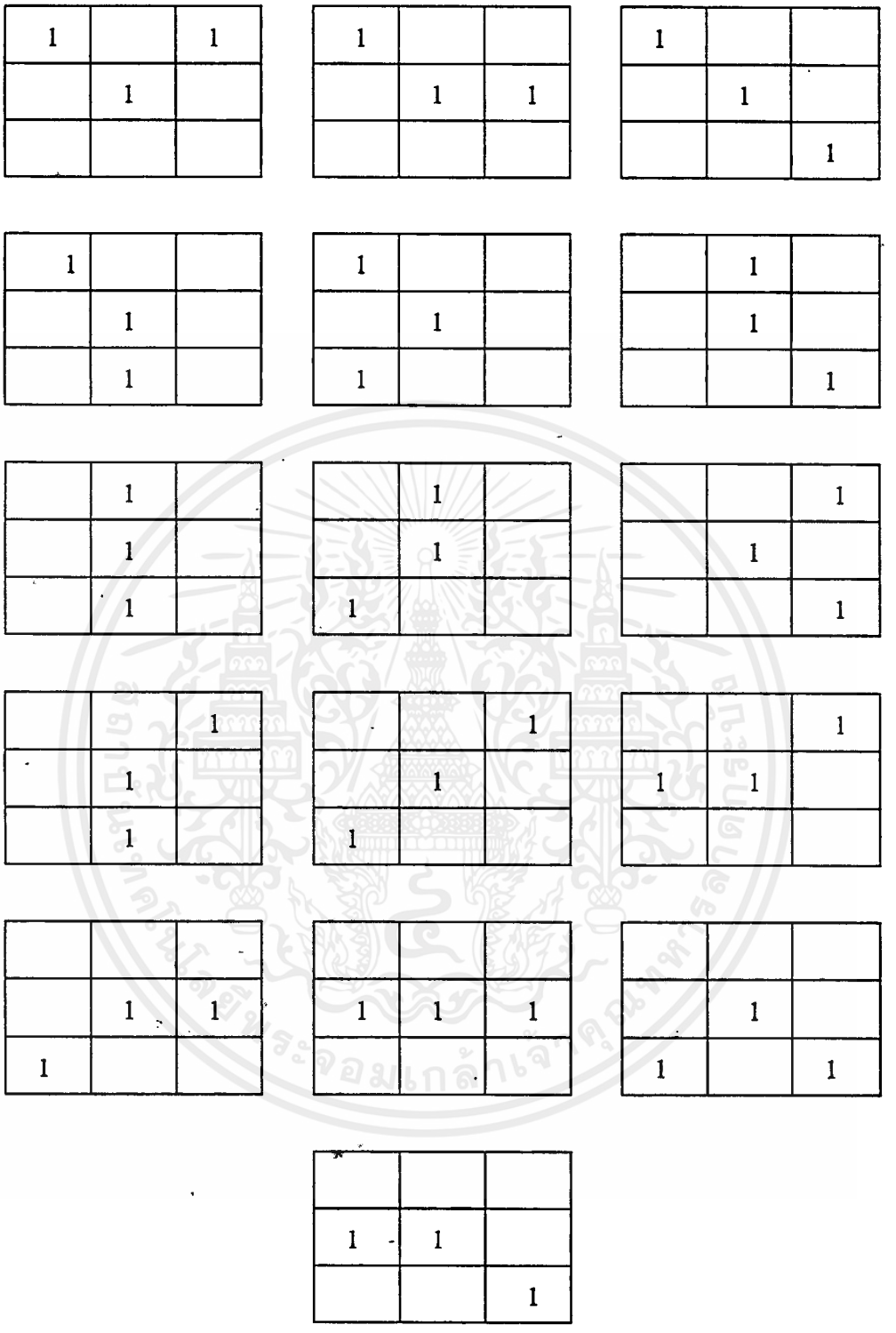
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



N = 1

ตาราง ที่ 2.3 จุดปลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



N=2

ตาราง ที่ 2.4 จุดต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1		1
	1	
		1

1		1
	1	
	1	

1		1
	1	
1		

	1	
	1	1
1		

		1
	1	
1		1

1		
	1	1
	1	

1		
	1	
1		1

		1
1	1	
	1	

	1	
1	1	
		1

N = 3

ตาราง ที่ 2.5 จุดแยก

	1	
1	1	1
	1	

1		1
	1	
1		1

N=4

ตาราง ที่ 2.6 จุดตัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1	1	1
1	1	1
1	1	1

N=8

ตาราง ที่ 2.7 จุดภายใน

ลักษณะจุด	จุดอิสระ	จุดปลาย	จุดต่อเนื่อง	จุดแยก	จุดตัด	จุดภายใน
ค่า N	0	1	2	3	4	8

ตาราง ที่ 2.8 ความสัมพันธ์ของจุด

2.3 การตรวจเช็คความหนาของขอบภาพ

ในการทำตัวอักษรให้บางโดยการลบส่วนที่เป็นขอบภาพของตัวอักษร เราจะต้องทราบก่อนว่าขอบภาพที่จะทำการลบนั้นมีความหนา มากกว่า 1 จุดภาพ หรือไม่ ถ้าขอบภาพนั้นมีความหนาเพียง 1 จุดภาพ แสดงว่าไม่สามารถลบจุดภาพนั้นได้

สมการในการหาความหนาของขอบภาพ

จุดที่มีความหนา 1 จุดภาพ (N1 | N3 | N5 | N7 | (!(N0|Ne))) = 1

N1 = X1 & (!(X8|X2))

N3 = X3 & (!(X2|X4))

N5 = X5 & (!(X4|X7))

N7 = X7 & (!(X6|X8))

N0 = (X2|x6) & (X4|X8)

Ne = (X1&X2&X3&(!X6)) | (X1&X2&X3&(!X6))

| (X1&X2&X3&(!X6)) | (X1&X2&X3&(!X6)))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N1,N3,N5,N7 เป็นจุดขอบภาพที่มุมใดมุมหนึ่งใน 4 มุม ไม่มีส่วนประกอบทั้ง 2 ด้านซึ่งจะทำให้มีความหนา 1 จุดภาพ

d	0	1
d	1	0
d	d	d

ตารางที่ 2.9 ลักษณะจุดตามสมการ N1

1	0	d
0	1	d
d	d	d

ตารางที่ 2.10 ลักษณะจุดตามสมการ N3

d	d	d
0	1	d
1	0	d

ตารางที่ 2.11 ลักษณะจุดตามสมการ N5

d	d	d
d	1	0
d	0	1

ตารางที่ 2.12 ลักษณะจุดตามสมการ N7

d	0	d
d	1	0
d	d	d

d	0	d
0	1	d
d	d	d

d	d	d
0	1	d
d	0	d

d	d	d
d	1	0
d	0	d

ตารางที่ 2.13 ลักษณะจุดตามสมการ N0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ne เป็นลักษณะของจุดขอบภาพที่เป็นส่วนเกิน จะต้องทำการลบออกไปด้วย

1	1	1
d	1	d
d	d	d

1	d	d
1	1	d
1	d	d

d	d	d
d	1	d
1	1	1

d	d	1
d	1	1
d	d	1

ตารางที่ 2.14 ลักษณะจุดตามสมการ Ne

2.4 การตรวจเช็คความหนาของภาพทั้งหมด

ในการทำตัวอักษรให้บางจะทำการลบส่วนที่เป็นขอบภาพของตัวอักษร จนกว่าภาพทั้งภาพมีความหนาเพียง 1 จุดภาพ โดยในแต่ละรอบจะทำการตรวจเช็คความหนาของขอบภาพ แล้วทำการลบจุดที่มีความหนามากกว่า 1 จุดภาพ จากนั้นจึงทำการตรวจเช็คความหนาของจุดภาพทั้งหมดอีกครั้ง ว่ายังมีจุดที่มีความหนามากกว่า 1 จุดภาพ หรือไม่ ถ้ามีจะต้องกลับไปทำรอบต่อไปอีก

สมการในการหาความหนาของจุดภาพ

N1 = X1 & (!(X8|X2))

N3 = X3 & (!(X2|X4))

N5 = X5 & (!(X4|X7))

N7 = X7 & (!(X6|X8))

S = X2&X4&X6&X8&(!(X1|X3|X5|X7))

T = (X2&X8&(!X5)) | (X2&X4&(!X7))

| (X4&X6&(!X1)) | (X6&X83&(!X3))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N1,N3,N5,N7 เป็นจุดภาพที่มุมใดมุมหนึ่งใน 4 มุม ไม่มีส่วนประกอบทั้ง 2 ด้านซึ่งจะทำให้มีความหนา 1 จุดภาพ เช่นเดียวกับ การตรวจเช็คความหนาของจุดที่เป็นขอบภาพ

สมการ S คือ จุดที่เป็น 4 แยก

0	1	0
1	1	1
0	1	0

ตารางที่ 2.15 ลักษณะจุด ตามสมการ S

สมการ T คือ จุด ที่ เป็น มุมฉากต่างๆ

d	1	d
d	1	1
d	d	d

d	1	d
1	1	d
d	d	d

d	d	d
d	1	1
d	1	d

d	d	d
1	1	d
d	1	d

ตารางที่ 2.16 ลักษณะจุด ตามสมการ T

ลำดับขั้นตอนในการตรวจเช็คความหนาของจุดภาพทั้งหมด

1. ถ้า (N1|N3|N5|N7) = 1 แสดงว่ามีความหนา มากกว่า 1 จุดภาพ
2. ถ้า S = 1 แสดงว่ามีความหนา 1 จุดภาพ
3. ถ้า T = 1 แสดงว่ามีความหนา มากกว่า 1 จุดภาพ
4. นอกจากนี้แสดงว่า มี ความหนา 1 จุดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

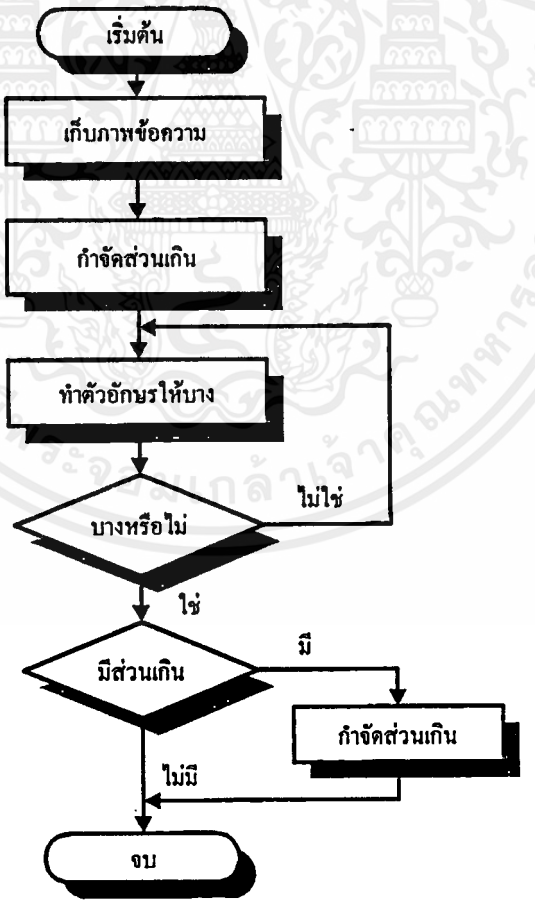
บทที่ 3
การคำนวณและการสร้าง

3.1 การจัดการล่วงหน้า

ในการวิเคราะห์โครงสร้างของตัวอักษรเพื่อหารูปแบบเฉพาะของตัวอักษร ตัวอักษรที่จะนำมาวิเคราะห์จะต้องมีความหนาของเส้น 1 จุดเท่านั้นคือ ข้อมูลที่ได้จากเครื่องตรวจกวาดภาพ จะต้องนำมาผ่านการจัดการล่วงหน้า (preprocessing) ก่อนที่จะนำไปสู่กระบวนการรู้จำ

การจัดการล่วงหน้า สามารถแบ่งออกเป็น 3 ขั้นตอนคือ

1. การกำจัดสัญญาณรบกวน
2. การทำตัวอักษรให้บาง
3. การกำจัดส่วนเกิน



รูปที่ 3.1 flow chart การจัดการล่วงหน้า

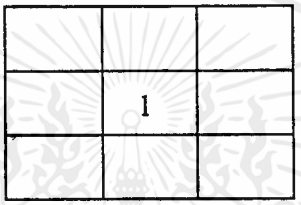
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1การกำจัดสัญญาณรบกวน

สัญญาณรบกวน คือจุดต่าง ๆ ที่เกิดขึ้นมาโดยที่จุดนั้นไม่ได้เป็นส่วนหนึ่งส่วนใดของตัวอักษร การตรวจหาจุดรบกวนจะตรวจหาจุดภาพที่ไม่มีจุดใกล้เคียงเลยและทำการลบจุดนั้นออกไป โดยจะทำการตรวจหาจุดรบกวนที่มีอยู่ทั้งภาพ

ลักษณะของสัญญาณรบกวน ก็คือ จุดอิสระต่างๆ

$$N = X1+X2+X3+X4+X5+X6+X7+X8 = 0$$

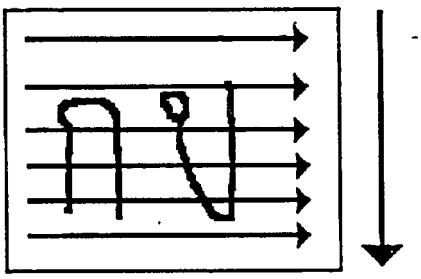


ตาราง ที่ 3.1 จุดรบกวน

3.1.2 การทำตัวอักษรให้บาง

ในการทำตัวอักษรให้บาง จะต้องมีการตรวจหาจุดภาพที่มีความหนามากกว่า 1 จุด จากขอบภาพเข้ามาทั้ง 4 ทิศ ถ้าจุดใดมีความหนามากกว่า 1 จุดภาพก็จะลบจุดนั้นออกไปโดยการตรวจเช็คจะทำจนกว่าไม่มีจุดที่มีความหนามากกว่า 1 จุดภาพ ทิศทางที่ใช้ในการตรวจความหนาของจุดภาพ 4 ทิศ คือ

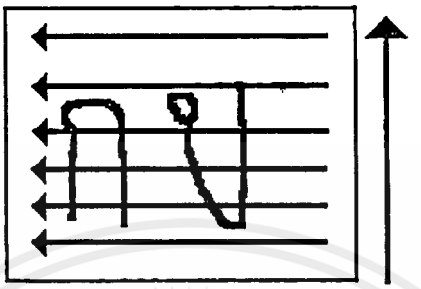
1. จากด้านบนลงด้านล่างของภาพโดยตรวจเช็คในแนวแกน X จากด้านซ้ายไปด้านขวา ทีละ 1 เส้น



รูปที่ 3.2 การตรวจเช็คจากด้านบนลงด้านล่าง

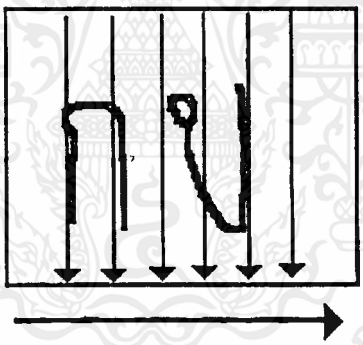
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากด้านล่างขึ้นด้านบนของภาพโดยตรวจเช็คในแนวแกน X จากด้านขวาไปด้านซ้าย ทีละ 1 เส้น



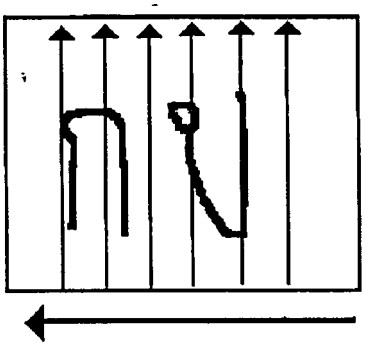
รูปที่ 3.3 การตรวจเช็คจากด้านล่างขึ้นด้านบน

3. จากด้านซ้ายไปด้านขวาของภาพโดยกระทำในแนวแกน Y จากด้านบนลงด้านล่าง ทีละ 1 เส้น



รูปที่ 3.4 การตรวจเช็คจากด้านซ้ายไปด้านขวา

4. จากด้านขวาไปด้านซ้ายโดยกระทำในแนวแกน Y จากด้านล่างขึ้นด้านบน ทีละ 1 เส้น



รูปที่ 3.5 การตรวจเช็คจากด้านขวาไปด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งนี้จากการตรวจเช็คทั้ง 4 ทิศทางจะทำให้ใช้เวลาในการทำให้ตัวอักษรบางน้อยกว่าการตรวจเช็คโดยใช้ทิศทางเดียว และภาพที่ผ่านการทำให้บางแล้วจะมีความสมดุลย์ทางโครงสร้างมากกว่าการทำให้บางในทิศทางเดียว

3.1.3 การขจัดส่วนเกิน

ภายหลังจากการทำตัวอักษรให้บางแล้ว ภาพโครงสร้างตัวอักษรที่ได้ยังมีส่วนเกินซึ่งเป็นลายเส้นเล็กๆ ที่แตกออกไปจากโครงสร้างของตัวอักษร จึงต้องมีการขจัดส่วนเกินนี้ออกไปด้วยเพื่อให้เหลือเฉพาะโครงร่างซึ่งเป็นโครงสร้างของตัวอักษรจริงๆ ก่อนที่จะนำไปสู่การรู้จำ

วิธีที่ใช้ในการขจัดส่วนเกิน คือการนับจำนวนจุดที่แยกออกไปจากจุดแยกและเกิดจุดปลายภายในจำนวนจุดที่กำหนดไว้ ว่าเป็นส่วนเกิน ก็จะทำการลบจุดแยกนั้นออกไปทั้งเส้น

ลักษณะของจุดแยกคือจุดที่มีจุดใกล้เคียงมากกว่า 2 จุด

จุดแยกคือจุดที่ (X1+X2+X3+X4+X5+X6+X7+X8) > 2

X3	X2	X1
X4	X0	X8
X5	X6	X7

ตารางที่ 3.2 จุดใกล้เคียง 8 ทิศทาง

จุด X0 คือจุดที่กำลังพิจารณาอยู่โดยที่จุด X1 - X8 เป็นจุดใกล้เคียงจุดที่กำลังพิจารณา

ลักษณะของจุดปลาย คือจุดที่มีจุดใกล้เคียงเพียง 1 จุด

(X1+X2+X3+X4+X5+X6+X7+X8) = 1

วิธีหาจำนวนจุดที่แยกออกไปจากจุดแยก

1. ทำการหาจุดใกล้เคียงของจุดปัจจุบัน แล้วทำการย้ายไปยังจุดใกล้เคียงนั้น และทำการเพิ่มจำนวนนับของจุดที่แยกออกไป พร้อมทั้งเก็บตำแหน่งเก่าไว้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำข้อ 1. จนกว่าจะพบจุดปลายหรือพบจุดแยกใหม่หรือจำนวนนับของจุดที่แยกออกไปมากกว่าจำนวนจุดที่กำหนดว่าเป็นส่วนเกิน

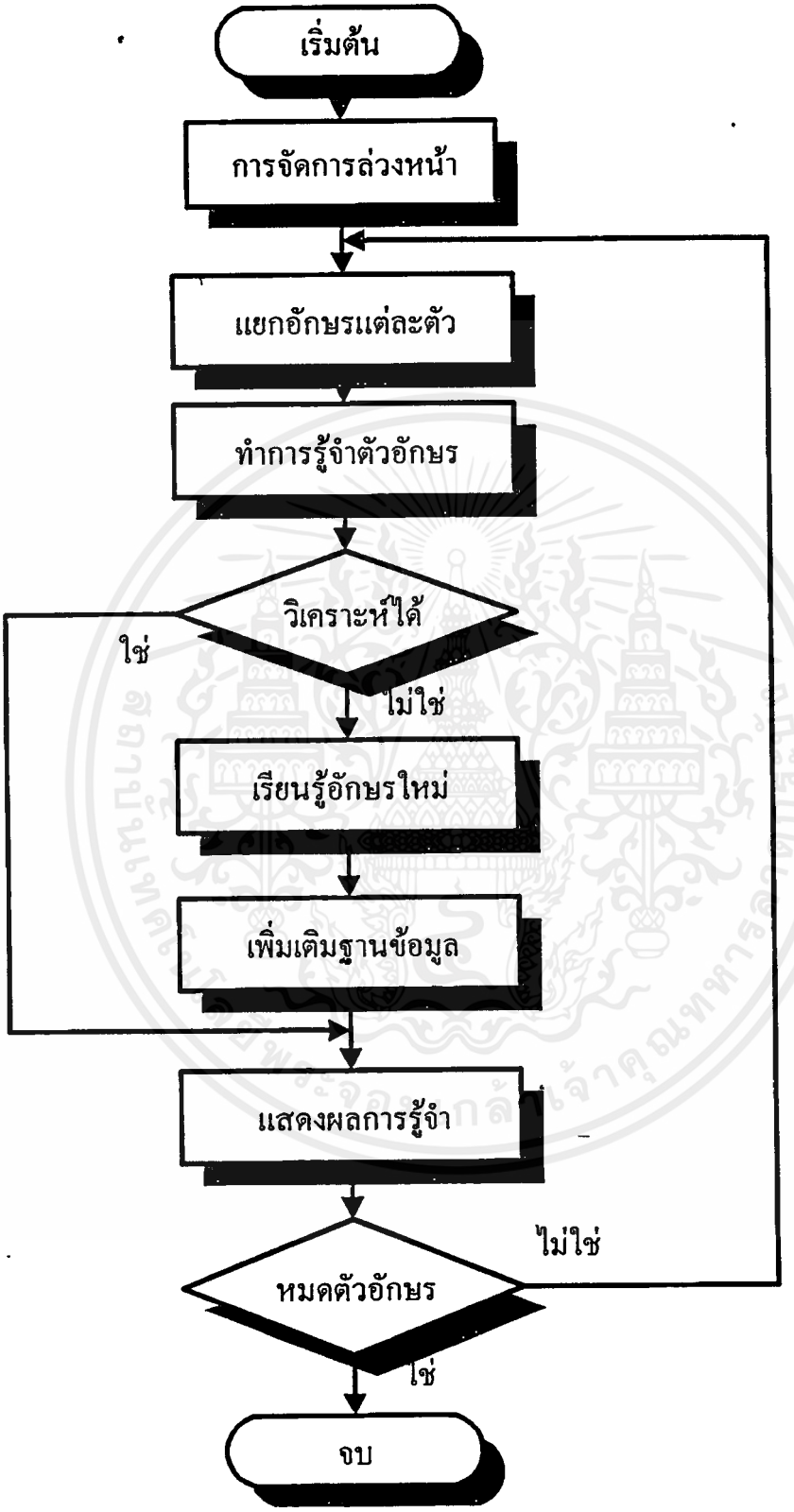
3. ถ้าจุดปัจจุบันเป็นจุดปลาย แสดงว่าทางแยกที่ผ่านมาเป็นส่วนเกินให้ทำการลบทางแยกนั้นออกไปทั้งแยก โดยนำตำแหน่งที่เก็บไว้จากข้อ 1. มาทำการลบ

รูปที่ 3.6 ภาพก่อนขจัดส่วนเกิน

รูปที่ 3.7 ภาพหลังขจัดส่วนเกิน

โดยข้อมูลที่ผ่านจากการจัดการล่วงหน้าแล้วจะนำไปสู่การรู้จำตัวอักษรทีละตัว ถ้าตัวไหนสามารถทำการรู้จำได้ก็จะแสดงออกมาให้ดูแต่ถ้าไม่สามารถวิเคราะห์ได้ก็จะแสดงภาพตัวอักษรนั้นพร้อมทั้งให้ผู้ใช้สามารถระบุว่าเป็นตัวอักษรอะไรแล้วจึงนำข้อมูลที่ได้ไปเก็บไว้ในฐานข้อมูลเพื่อนำไปใช้ในการวิเคราะห์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 flow chart การรู้จำตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การแยกตัวอักษรออกจากภาพประโยคบนหน้ากระดาษ

การแยกตัวอักษรออกจากภาพประโยคโดยการติดตามจุดภาพ เป็นการไต่ไปตามจุดภาพ (จุดที่เป็น 1) ลักษณะการไต่ จะเคลื่อนที่ ไปยังจุดภาพจุดถัดไปที่อยู่ใกล้เคียงจุดปัจจุบัน โดยมีลำดับขั้นตอนดังนี้

1. หาจุดใกล้เคียงของจุดปัจจุบัน โดยไม่รวมจุดเก่าที่ผ่านมาแล้ว

2. ถ้าจุดที่อยู่ปัจจุบันมีจุดใกล้เคียงเพียง 1 จุด ให้ จุดเก่าเท่ากับจุดปัจจุบัน และย้ายจุดปัจจุบัน ไปยังจุดใกล้เคียงนั้น

3. ถ้าจุดที่อยู่ปัจจุบันมีจุดใกล้เคียงมากกว่า 1 จุด ให้ย้ายไปยังจุดใกล้เคียงจุดใดจุดหนึ่ง และจัดเก็บจุดใกล้เคียงซึ่งเป็นจุดแยกที่เหลือไว้

4. ถ้าจุดที่อยู่ปัจจุบันไม่มีจุดใกล้เคียง ให้เลือกจุดแยกที่เก็บไว้เป็นจุดเริ่มต้นใหม่

5. การเคลื่อนที่จะสิ้นสุดลง เมื่อจุดที่อยู่ปัจจุบัน ไม่มีจุดใกล้เคียง และไม่มีจุดแยกที่จะเป็นจุดเริ่มต้นใหม่

เมื่อได้ขอบเขตของภาพ 1 ตัว แล้วก็จะทำการ คัดลอกภาพนั้นไปทำการวิเคราะห์ขั้นตอนในการคัดลอกภาพไปนั้น เราจะต้องมีการเก็บตำแหน่งที่ภาพนั้นอยู่ไว้ด้วยเพื่อ การนำมาจัดเรียงเป็นประโยคที่ถูกต้องหลังจากได้มีการวิเคราะห์ตัวอักษรนั้นแล้ว และเมื่อมีการคัดลอกภาพอักษรนั้นไป ก็จะทำการลบภาพอักษรนั้นออกจากภาพใหญ่ เพื่อจะได้ไม่รบกวนในการหาตัวอักษรตัวต่อไป

เทคโน

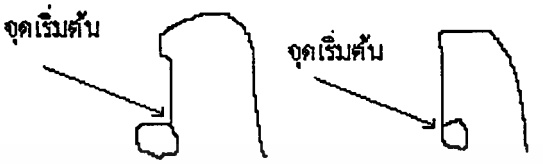
1. จากการ ตรวจกวาด จากมุมบนซ้ายลงด้านล่างขวาจะพบ โ ก่อน
2. ทำการติดตามจุดภาพของ โ จะได้ โ ออกมา
3. ทำการคัดลอก โ ไปทำการวิเคราะห์ โดยเก็บตำแหน่ง X , Y Center ของ โ ไว้ด้วยเพื่อการจัดเรียงประโยคภายหลัง
4. ลบภาพ โ ออกจากประโยค
5. ทำการ ตรวจกวาด หาตัวอักษรต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

3.3 การหาหัวของตัวอักษร

1. ตรวจหาจุดแยกที่มีจุดใกล้เคียงตั้งแต่ 3 จุดขึ้นไปเพื่อเป็นจุดเริ่มต้นในการหาหัว



รูปที่ 3.9 แสดงจุดเริ่มต้นในการหาหัว

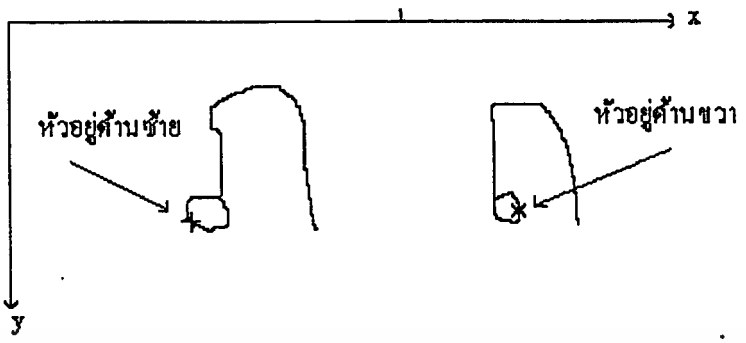
2. ทำการเคลื่อนย้ายไปยังจุดใกล้เคียงของจุดแยกนั้นทุกๆจุดแยกพร้อมๆกันที่ละจุด

3. ถ้ามีจุดแยกใดทำการเคลื่อนย้ายไปยังจุดใหม่ที่จุดแยกอื่นได้เคลื่อนย้ายมาแล้ว (หรือเกิดการชนกัน) แสดงว่าเกิดจุดหัวขึ้นมา

รูปที่ 3.10 แสดงการเกิดจุดหัว

4. ถ้าเกิดจุดหัวขึ้นมาจะกำหนดให้จุดเริ่มต้นของจุดแยกคือจุดหัวของตัวอักษร โดยที่สถานะของหัวตัวอักษร(ซ้าย,ขวา) จะได้จากการตรวจเช็คว่า ตำแหน่งที่ได้จากข้อ 3. มีค่าในแกน X น้อยกว่าหรือมากกว่าจุดที่เป็นจุดหัว ถ้ามีค่าน้อยกว่าแสดงว่าหัวนั้นเป็นหัวที่อยู่ทางด้านซ้าย แต่ถ้ามีค่ามากกว่าแสดงว่าเป็นหัวที่อยู่ทางด้านขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
038307



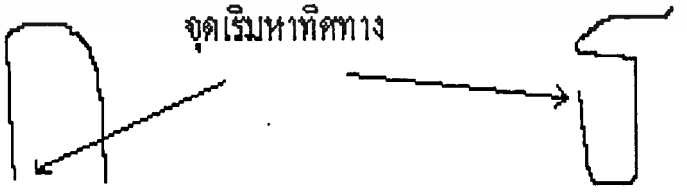
รูปที่ 3.11 แสดงลักษณะของหัว

5. เมื่อหาตำแหน่งได้แล้ว จะทำการลบวงกลมที่เป็นส่วนหัวออกไปให้หมด เพื่อให้เหลือเฉพาะส่วนที่เป็นโครงร่างของตัวอักษร เพื่อนำไปใช้ในการหาทิศทางต่อไป

3.4 การหาทิศทางโครงสร้างของตัวอักษร

ในการหาทิศทางเพื่อกำหนดเป็นรูปแบบเฉพาะของตัวอักษรจะต้องมีการกำหนดจุดเริ่มต้น ในการหาทิศทางของตัวอักษรโดยจุดเริ่มต้นจะแบ่งออกได้เป็น 2 กรณีคือ

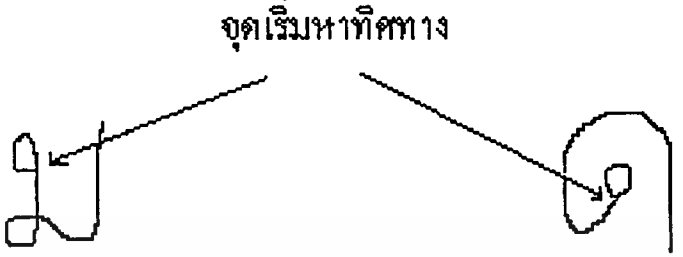
1. ตัวอักษรที่ไม่มีหัวจะกำหนดให้จุดเริ่มต้นในการหาทิศทางคือจุดปลายที่อยู่ทางด้านซ้ายของตัวอักษร



รูปที่ 3.12 แสดงจุดเริ่มหาทิศทางสำหรับตัวอักษรไม่มีหัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ตัวอักษรที่มีหัวจะกำหนดให้จุดเริ่มต้นในการหาทิศทางคือจุดหัวที่มีจุดแยกออกไปเพียงจุดเดียว



รูปที่ 3.13 แสดงจุดเริ่มหาทิศทางสำหรับตัวอักษรมีหัว

ลักษณะของทิศทางที่กำหนดเป็นรูปแบบของตัวอักษรจะใช้ทิศทาง 4 ทิศคือ

1. ทิศทางจากด้านล่างขึ้นด้านบนใช้สัญลักษณ์ตัว U
2. ทิศทางจากด้านบนลงด้านล่างใช้สัญลักษณ์ตัว D
3. ทิศทางจากด้านซ้ายไปด้านขวาใช้สัญลักษณ์ตัว R
4. ทิศทางจากด้านขวาไปด้านซ้ายใช้สัญลักษณ์ตัว L

การไล่หาทิศทางของลายเส้นจะใช้ตารางหน้าต่างขนาด 3 x 3 เพื่อหาลักษณะของ Chain Code 8 ทิศทาง

X3	X2	X1
X4	X0	X8
X5	X6	X7

ตารางที่ 3.3 Chain code 8 ทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาทิศทางทั้ง 4 ทิศจะมีวิธีการหาดังนี้

1.1 ถ้าจุดถัดไปเป็น X1 หรือ X2 หรือ X3 แสดงว่ามีทิศทางขึ้นบน

X3	X2	X1
	X0	

ตารางที่ 3.4 ทิศทางขึ้นบน

1.2 ถ้าจุดถัดไปเป็น X5 หรือ X6 หรือ X7 แสดงว่ามีทิศทางลงล่าง

	X0	
X5	X6	X7

ตารางที่ 3.5 ทิศทางลงล่าง

1.3 ถ้าจุดถัดไปเป็น X1 หรือ X8 หรือ X7 แสดงว่ามีทิศทางไปทางด้านขวา

		X1
	X0	X8
		X7

ตารางที่ 3.6 ทิศทางไปทางด้านขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ถ้าจุดถัดไปเป็น X3 หรือ X4 หรือ X5 แสดงว่ามีทิศทางไปทางด้านซ้าย

X3		
X4	X0	
X5		

ตารางที่ 3.7 ทิศทางไปทางด้านซ้าย

การหาทิศทางจะมีการกำหนดจำนวนจุดที่ก่อให้เกิดการเปลี่ยนทิศทางจากทิศทางหนึ่งไปยังอีกทิศทางหนึ่ง โดยที่จำนวนจุดที่กำหนดไว้ในที่นี้ใช้ 3 จุด

2. ทำการหาทิศทางตามข้อ 1. แล้วเพิ่มจำนวนนับของแต่ละทิศทางที่ได้

3. ถ้าจำนวนนับของทิศทางใด ถึงจำนวนจุดที่ก่อให้เกิดการเปลี่ยนทิศทางให้กำหนดทิศทางที่ได้เป็นไปตามทิศทางนั้น

4. ทำการหาทิศทางต่อไปจนกว่าจะมีการกหนดทิ ศทางใหม่ที่แตกต่างไปจากทิศทางเดิมที่กำหนดไว้จากข้อ 3.

5. ค่าทิศทางที่ได้ก็คือ ทิศทางที่ได้จากข้อ 3.

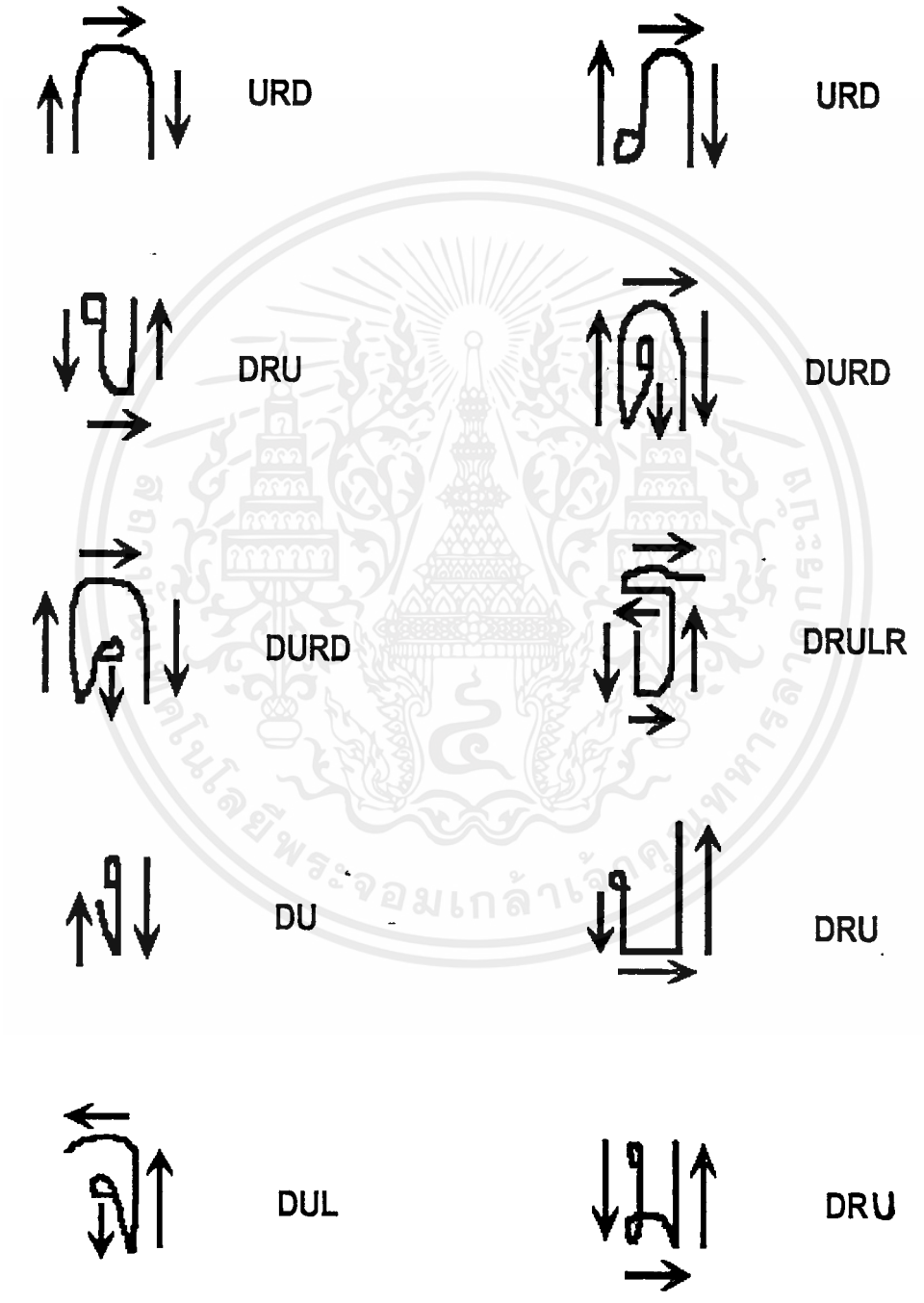
6. จากข้อ 5. ค่าทิศทางที่ได้จะเป็นทิศทางเดียวเท่านั้น ยังไม่ได้เป็นทิศทางทั้งหมดของตัวอักษรจะต้องมีการหาทิศทางต่อไป ต่อจากจุดเดิมอีกจนกว่าจะหมดโครงสร้างของตัวอักษรและนำทิศทางที่ได้เก็บเป็นรูปแบบของตัวอักษร

ตัวอย่างเช่น ตัวอักษร ก ทิศทางที่ได้จะเท่ากับ URD และตัวอักษร ข ทิศทางที่ได้จะเท่ากับ DRU

รูปที่ 3.14 แสดงการหาทิศทางของตัวอักษร ก และ ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการระบุว่าภาพนั้นเป็นตัวอักษรอะไร เราจะนำวิธีในการจำแนกต่างๆมาใช้ร่วมกัน เพื่อระบุว่าเป็นตัวอักษรอะไร เช่น



รูปที่ 3.15 ตัวอย่างทิศทางของตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างรูปแบบทิศทางของตัวอักษร ในรูป 3.15 พอจะอธิบายได้ดังนี้

ตัวอักษร ก จะเป็นตัวอักษรที่ไม่มีหัวและมีทิศทางเท่ากับ URD

ตัวอักษร ข มี 1 หัวและหัวอยู่ทางด้านซ้ายมีทิศทางเป็น DRU

ตัวอักษร ค มี 1 หัวและหัวอยู่ทางด้านขวามีทิศทางเป็น DURD

ตัวอักษร ง มี 1 หัวและหัวอยู่ทางด้านซ้ายมีทิศทางเป็น DU

ตัวอักษร จ มี 1 หัวและหัวอยู่ทางด้านซ้ายมีทิศทางเป็น DUL

ตัวอักษร ภ มี 1 หัวและหัวอยู่ทางด้านซ้ายมีทิศทางเป็น DUL

ตัวอักษร ด มี 1 หัวและหัวอยู่ทางด้านซ้ายมีทิศทางเป็น DURD

ตัวอักษร ธ เป็นตัวอักษรที่ไม่มีหัวและมีทิศทางเท่ากับ DRULR

ตัวอักษร ป มี 1 หัวและหัวอยู่ทางด้านซ้ายมีทิศทางเป็น DRU

ตัวอักษร ม มี 2 หัวและหัวอยู่ทางด้านซ้ายมีทิศทางเป็น DRU

จากตัวอย่างรูปแบบทิศทางของตัวอักษร ที่นำแสดงนี้ เป็นเพียงรูปแบบหนึ่งของตัวอักษรที่ได้จากการเขียนเท่านั้น

3.5 การจำแนกตัวอักษร (CLASSIFICATION)

1. การจำแนกตามจำนวนหัวของตัวอักษร

แบ่งออกได้เป็น 3 ประเภท คือ

1. อักษรไม่มีหัว
2. อักษรมี 1 หัว
3. อักษรมี 2 หัว

2. การจำแนกตามลักษณะหัวของตัวอักษร

แบ่งออกได้เป็น 2 ประเภท คือ

1. หัวอยู่ด้านซ้าย
2. หัวอยู่ด้านขวา

3. การจำแนกตามรูปแบบทิศทางของตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากการจำแนก ตาม 3 ขั้นตอน ข้างต้น แล้วจะทำให้ทราบได้ว่า ภาพที่ทำการวิเคราะห์นั้นคือตัวอักษรอะไร

แต่ยังมีอักษรบางตัว ที่มีจำนวนหัวลักษณะของหัว และรูปแบบทิศทางที่เหมือนกัน ทำให้ไม่ทราบว่าเป็นตัวอะไรแน่ จึงต้องมีวิธีการในการจำแนกตัวอักษรเพิ่มอีก คือ

4. การเทียบตำแหน่งของจุดปลายกับตำแหน่งของจุดหัว เช่น ตัวอักษร พ กับ ฟ , ผ กับ ฝ และ บ กับ ป ถ้าจุดปลายอยู่สูงกว่าจุดหัว ตามอัตราที่กำหนดไว้ จะถือว่าเป็นตัวอักษร ฟ , ฝ และ ป ตามลำดับ ส่วนตัวอักษร ถ กับ ฤ ถ้าจุดปลายอยู่ต่ำกว่าจุดหัว ตามอัตราที่กำหนดไว้ จะถือว่าเป็นตัวอักษร ฤ เช่นกัน

5. การหาอัตราส่วนความสูงเทียบกับความกว้างของตัวอักษร ใช้เพื่อแยกตัวอักษรบางคู่ เช่น ข กับ บ โดยถ้า อัตราส่วนที่ได้มีค่ามากกว่า 1.3 จะถือว่าเป็น ข

6. การเทียบตำแหน่งของจุดหัวกับตำแหน่งสูงสุดของตัวอักษร เช่น ตัวอักษร ด กับ ท หรือ ตัวอักษร ง กับ ถ้าจุดหัว เป็นจุดสูงสุดของตัวอักษร จะถือว่าเป็นตัวอักษร ท หรือ ง ตามลำดับ

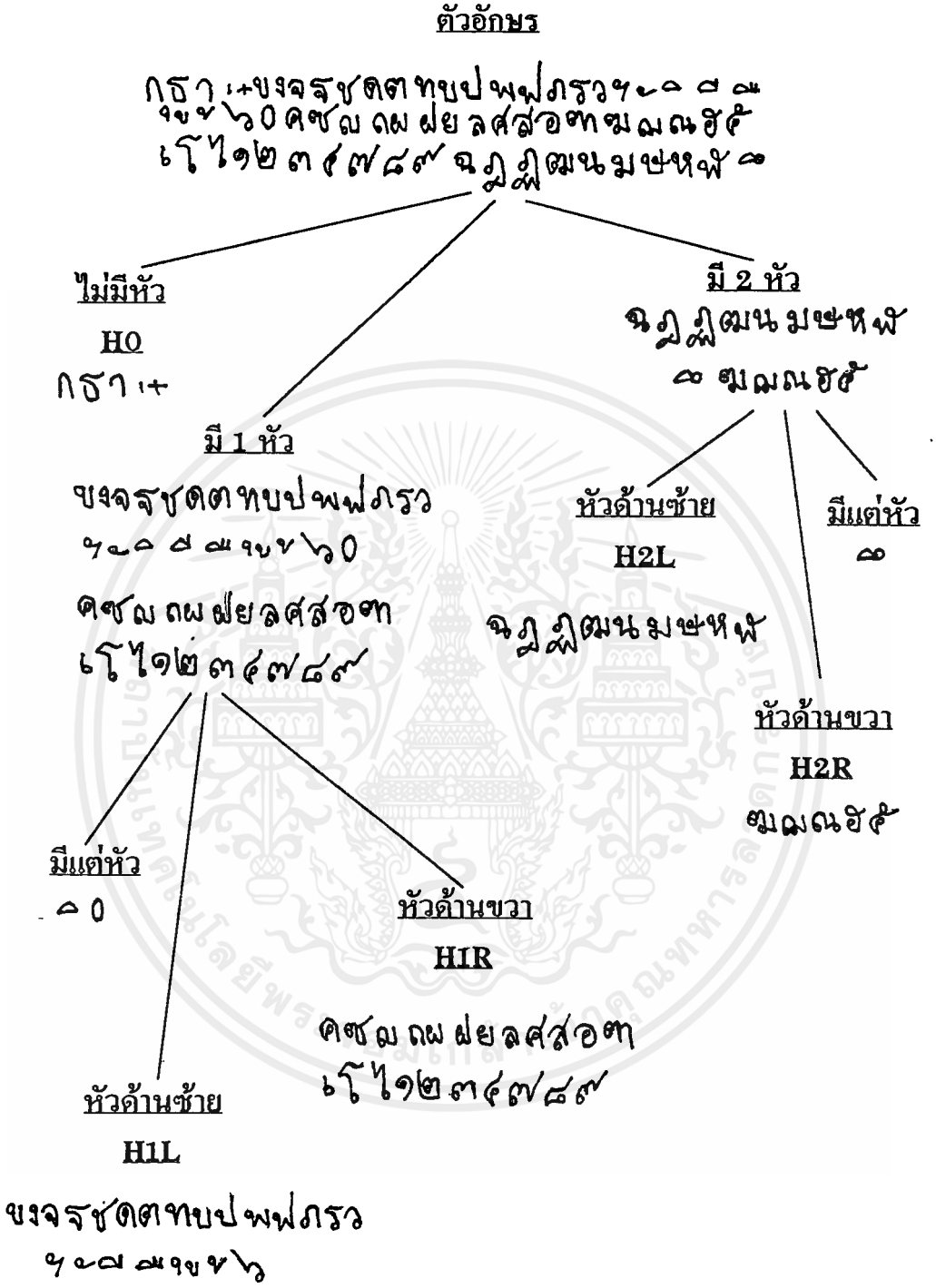
7. การเทียบตำแหน่งของจุดปลายกับตำแหน่งของจุดปลาย เช่น ตัวอักษร ก กับ า ถ้าจุดปลายกับจุดปลาย มี ระยะห่างในแกน Y สูงกว่าอัตราที่กำหนดไว้ จะถือว่าเป็นตัว า

8. การเปรียบเทียบขนาดของตัวอักษรนั้น กับ ขนาดโดยเฉลี่ยของตัวอักษร เช่น ตัวอักษร ข กับ ู ถ้าตัวอักษรมีขนาดเล็กกว่าขนาดโดยเฉลี่ยมาก แสดงว่าเป็นสระ ู

ต่อไปนี้จะเป็นตัวอย่าง Tree Diagram ในการจำแนกตัวอักษรต่างๆ

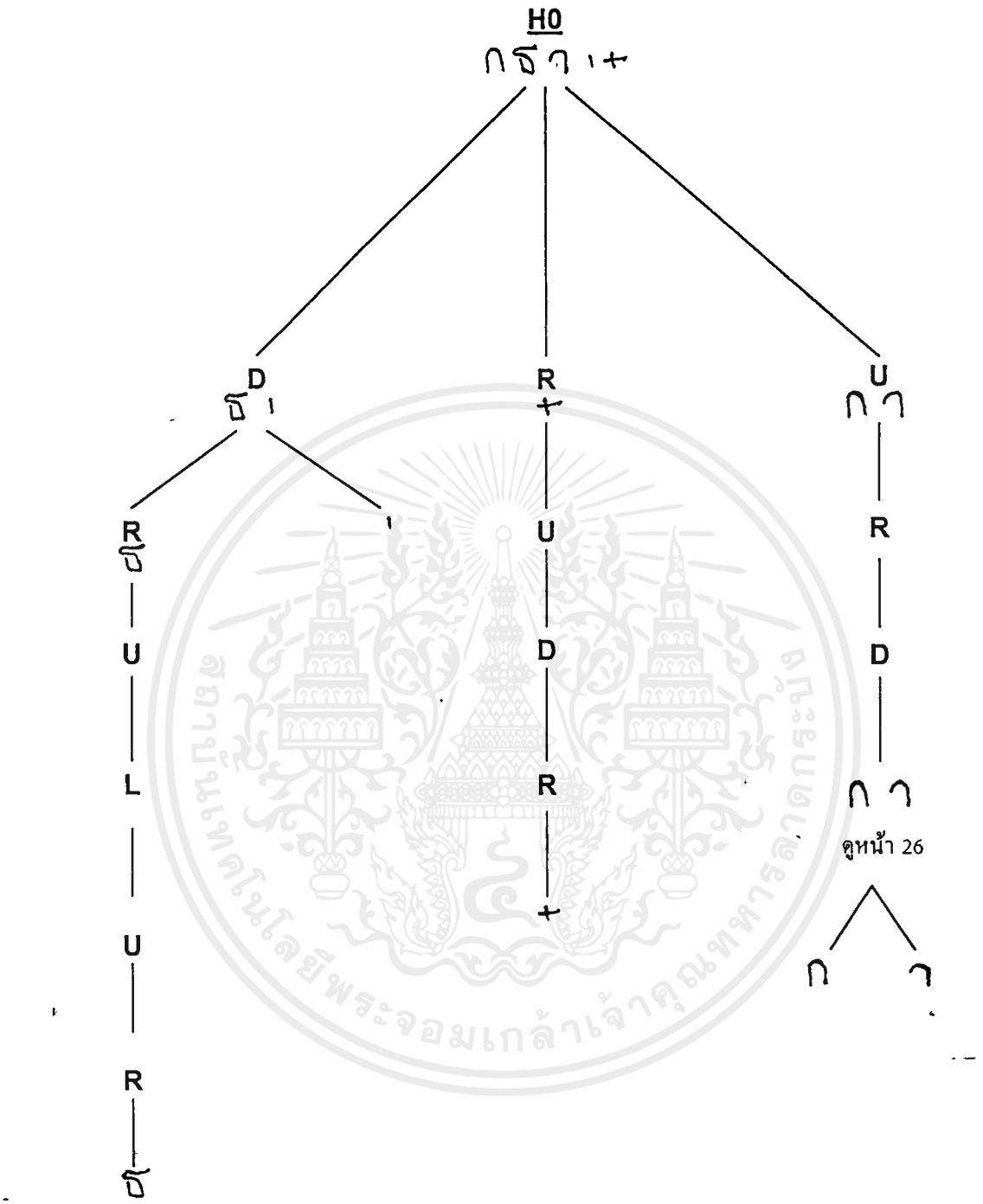
จากตัวอย่างที่จะนำมาแสดงนี้ เป็นเพียงรูปแบบหนึ่งของตัวอักษรที่ได้จากการเขียนโดยปกติทั่วๆไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



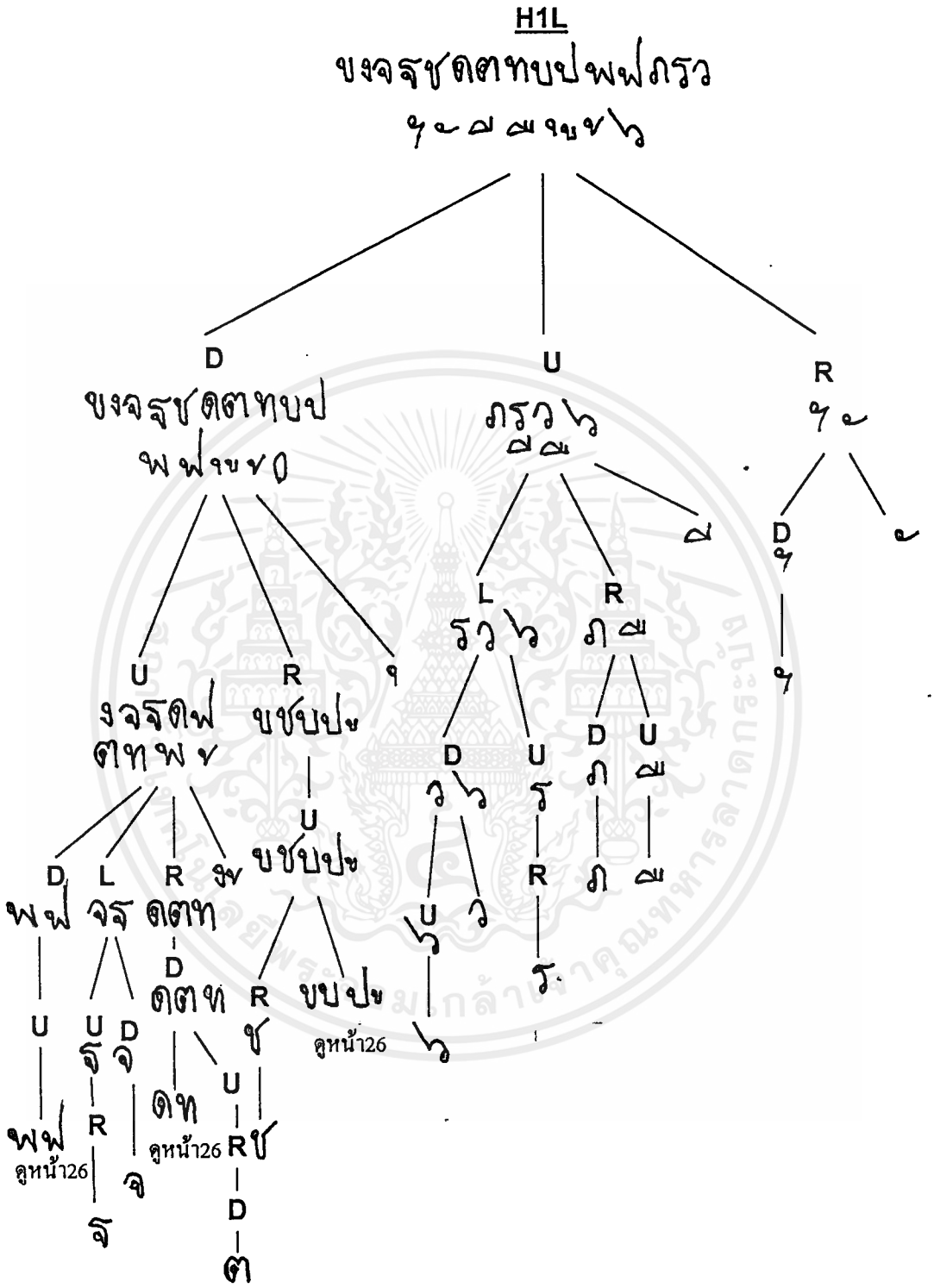
รูปที่ 3.16 การแบ่งตัวอักษรตามหัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



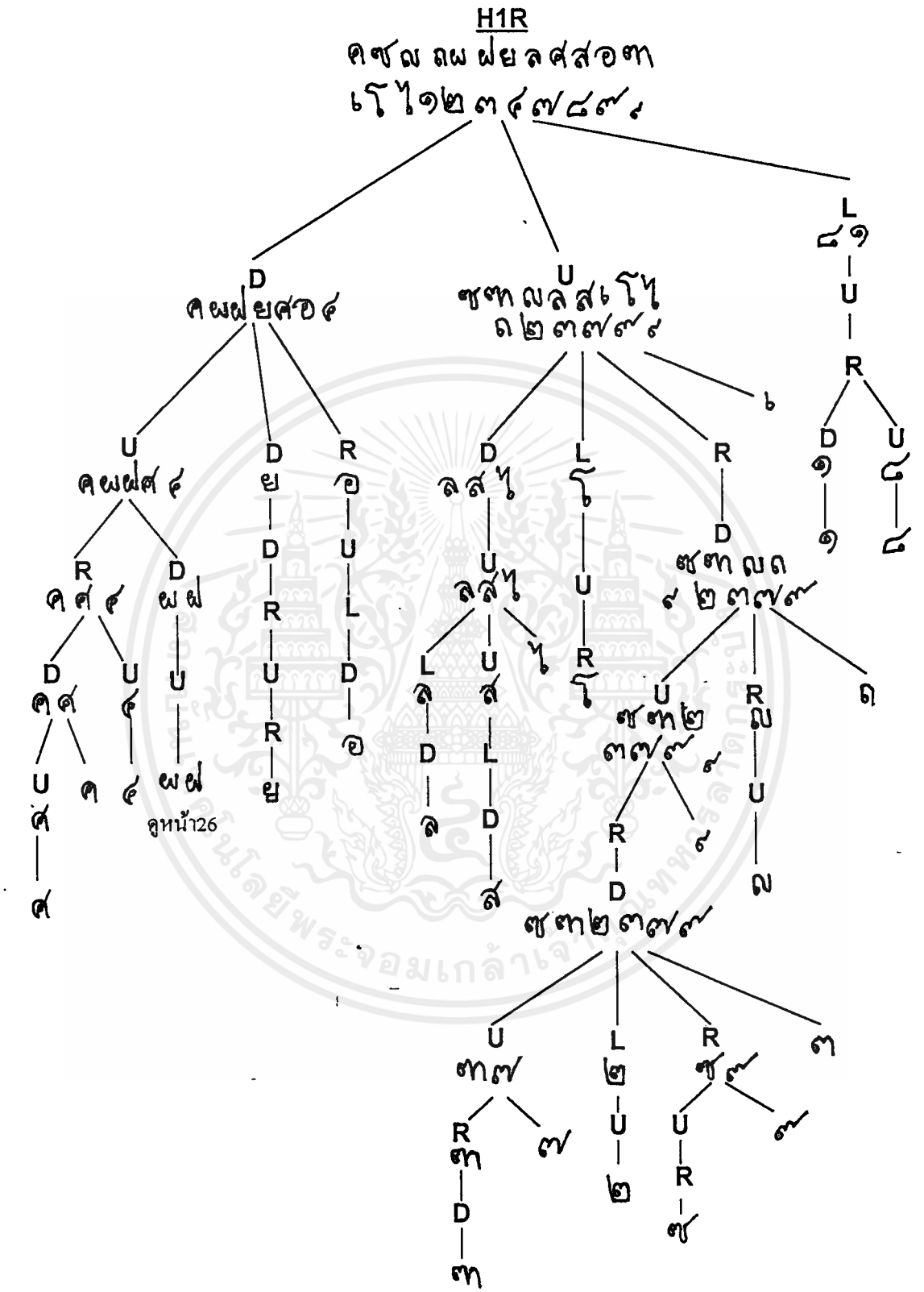
รูปที่ 3.17 ทิศทางของอักษรไม่มีหัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



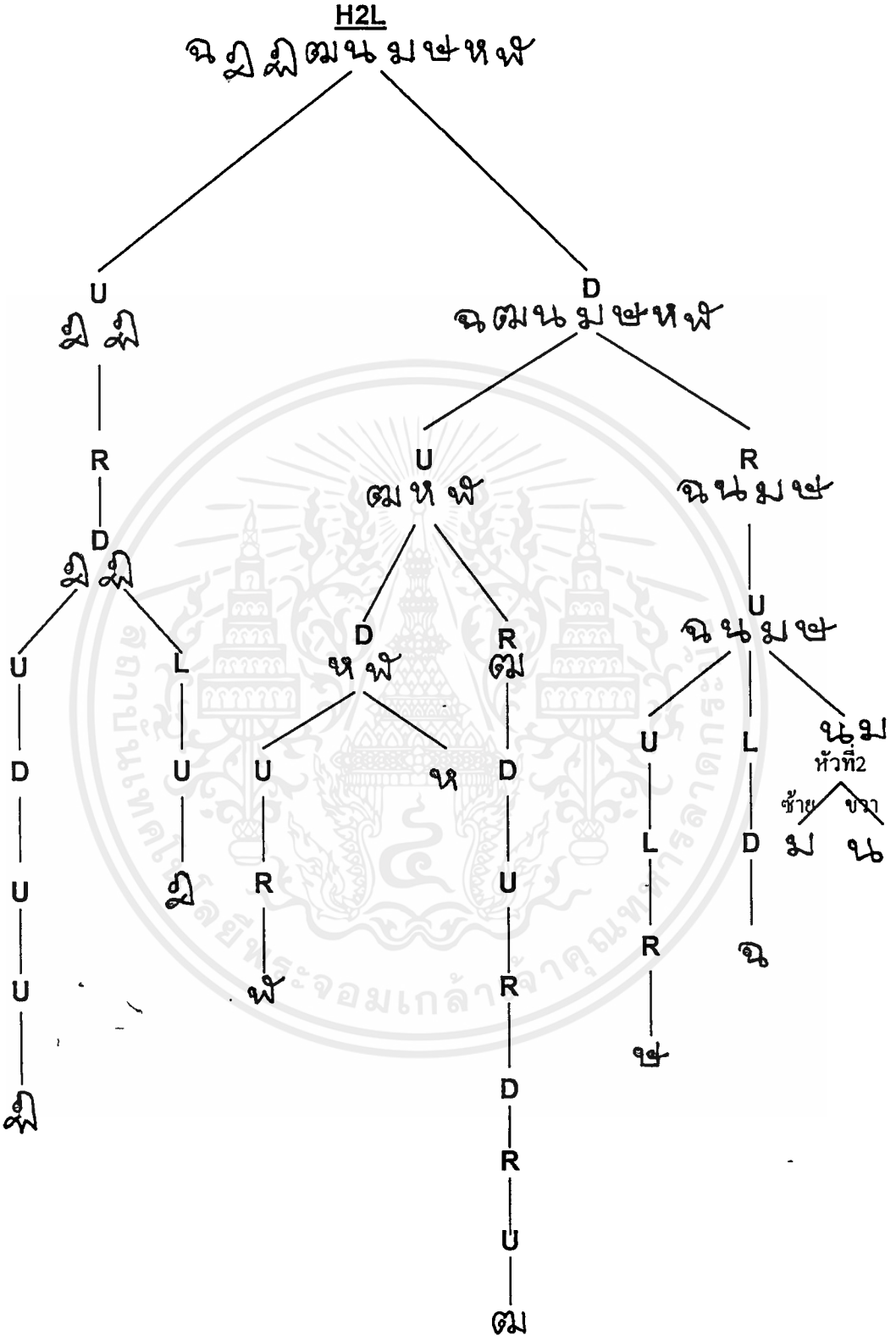
รูปที่ 3.18 ทิศทางของอักษร 1 หัว และเป็นหัวอยู่ด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



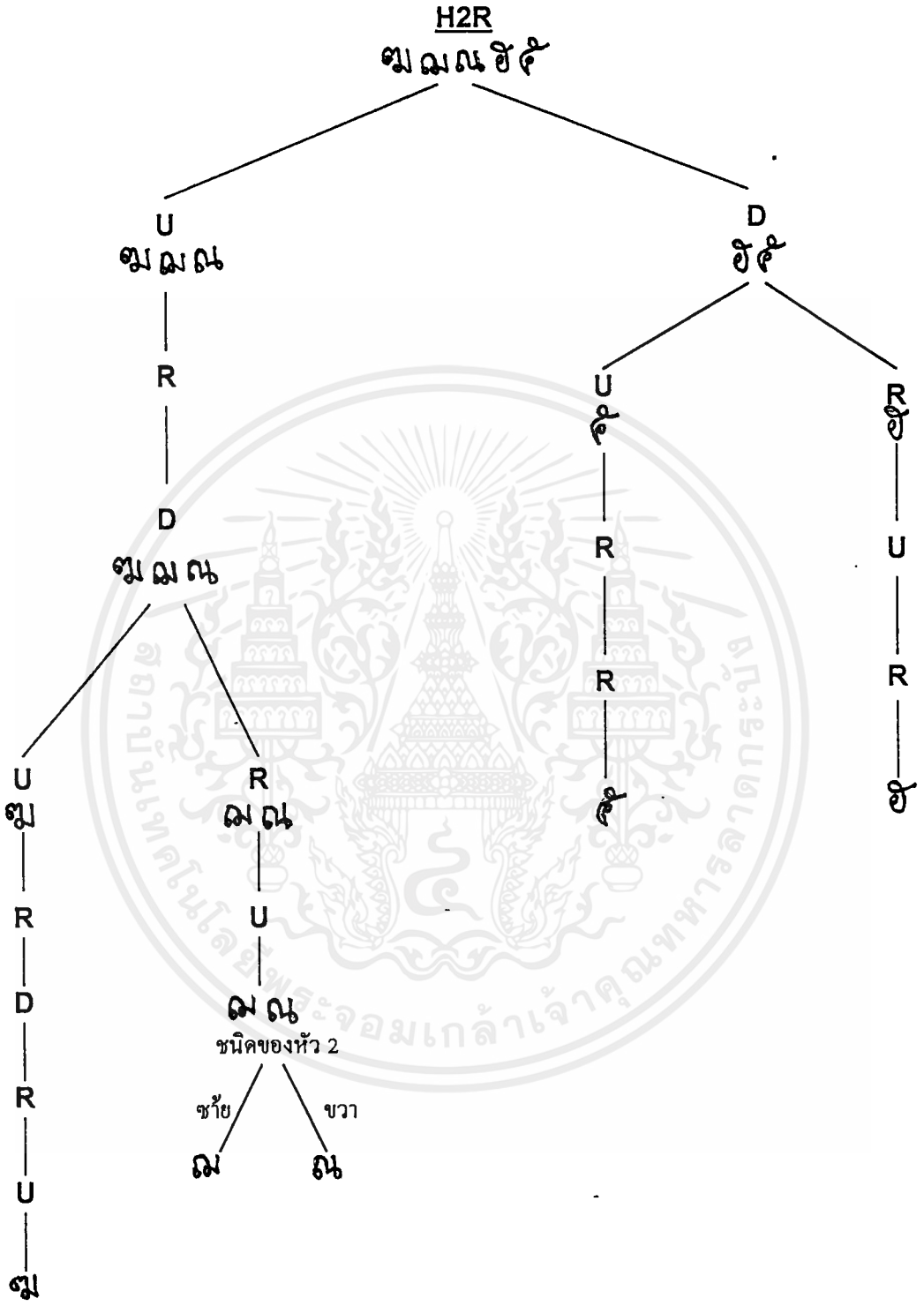
รูปที่ 3.19 ทิศทางของอักษร 1 หัว และเป็นหัวอยู่ด้านขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 ทิศทางของอักษร 2 หัว และหัวเริ่มต้นเป็นชนิดด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 ทิศทางของอักษร 2 หัว และหัวเริ่มต้นเป็นชนิดด้านขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทำงานของโปรแกรม

การทำงานของโปรแกรมเริ่มต้นจากการแปลงแฟ้มข้อมูลภาพลายมือเขียนภาษาไทย ซึ่งได้มาจากเครื่องตรวจกวาดภาพให้เป็นข้อมูลดิจิตอลที่พร้อมจะใช้ในการคำนวณ จากนั้นในขั้นตอนต่อไปจะเป็นการจัดการล่วงหน้า และทำให้ข้อมูลภาพที่ได้มีความถูกต้อง และเหมาะสมยิ่งขึ้นก่อนจะทำการแยกตัวอักษรแต่ละตัวมาทำการหาจำนวนหัว ลักษณะของหัว และทิศทางในการเขียนตัวอักษรแต่ละตัว เพื่อระบุว่าเป็นตัวอักษรอะไร โดยการเปรียบเทียบกับข้อมูลที่มีอยู่ในฐานข้อมูล ถ้าข้อมูลที่ได้จากการวิเคราะห์ไม่มีอยู่ในฐานข้อมูล ก็จะรอให้ผู้ใช้ระบุลงไปว่าภาพนั้นเป็นตัวอักษรอะไรเพื่อโปรแกรมจะได้นำไปเก็บไว้ในฐานข้อมูลเพิ่มเติมเพื่อใช้ในการวิเคราะห์ครั้งต่อไป

4.2 อุปกรณ์ที่ต้องนำมาใช้ในการทำงานของโปรแกรม

1. เครื่องไมโครคอมพิวเตอร์ไอบีเอ็ม หรือเครื่องที่คอมแพทิเบิล
2. หน่วยความจำแรม อย่างน้อย 4 เมกะไบต์
3. จอแสดงผลแบบ VGA เป็นอย่างต่ำ
4. ระบบปฏิบัติการ MS-DOS เป็นอย่างต่ำ
5. เครื่องตรวจกวาดภาพ
6. ไมโครซอร์ฟเมาส์หรือ เมาส์ที่คอมแพทิเบิล

4.3 การติดตั้งโปรแกรม

1. ใส่แผ่นติดตั้งไว้ที่ ดิสท์ไดรว์ A:
2. สร้างไดเรกทอรีย่อยไว้ที่ไดรว์ C:

   md c:\recog <enter>
3. คัดลอกแฟ้มข้อมูลในแผ่นติดตั้งไปที่ไดรว์ C: ในไดเรกทอรีย่อยที่สร้างไว้

   copy a:*.* c:\recog <enter>

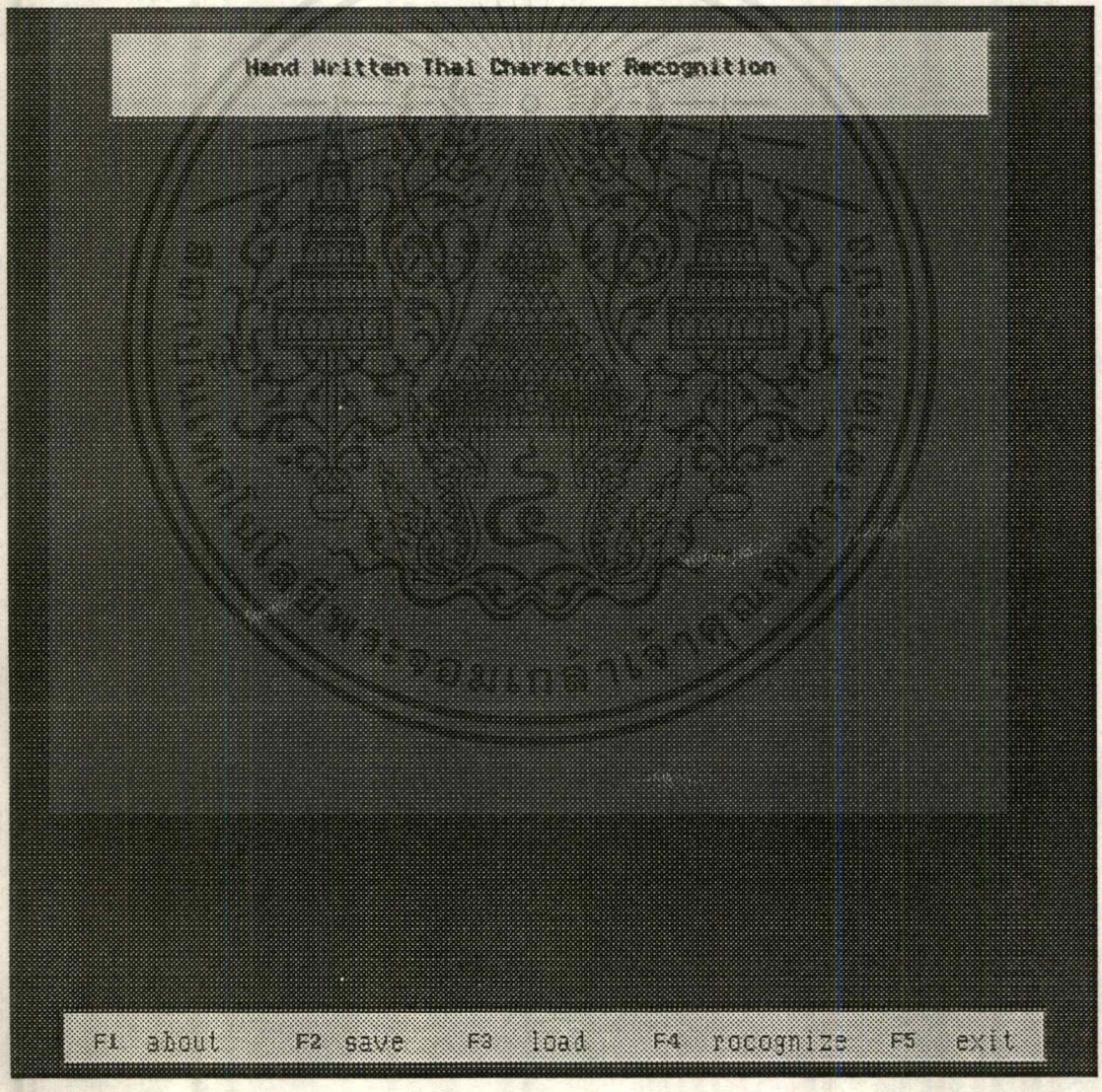
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ขั้นตอนการใช้งานโปรแกรม

การทำงานของโปรแกรมจะเป็นการทำงานผ่าน เมนู โดยการใช้คีย์บอร์ด เลือกขั้นตอนต่าง ๆ โดยจะมีขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 การเรียกการใช้งานโปรแกรม จะเริ่มต้นด้วยการเรียกชื่อโปรแกรมที่ดอสพรอมท์ คือ c:\recog\recog <enter>

โปรแกรมจะเริ่มการทำงาน โดยมีเมนูในการสั่งงานดังนี้

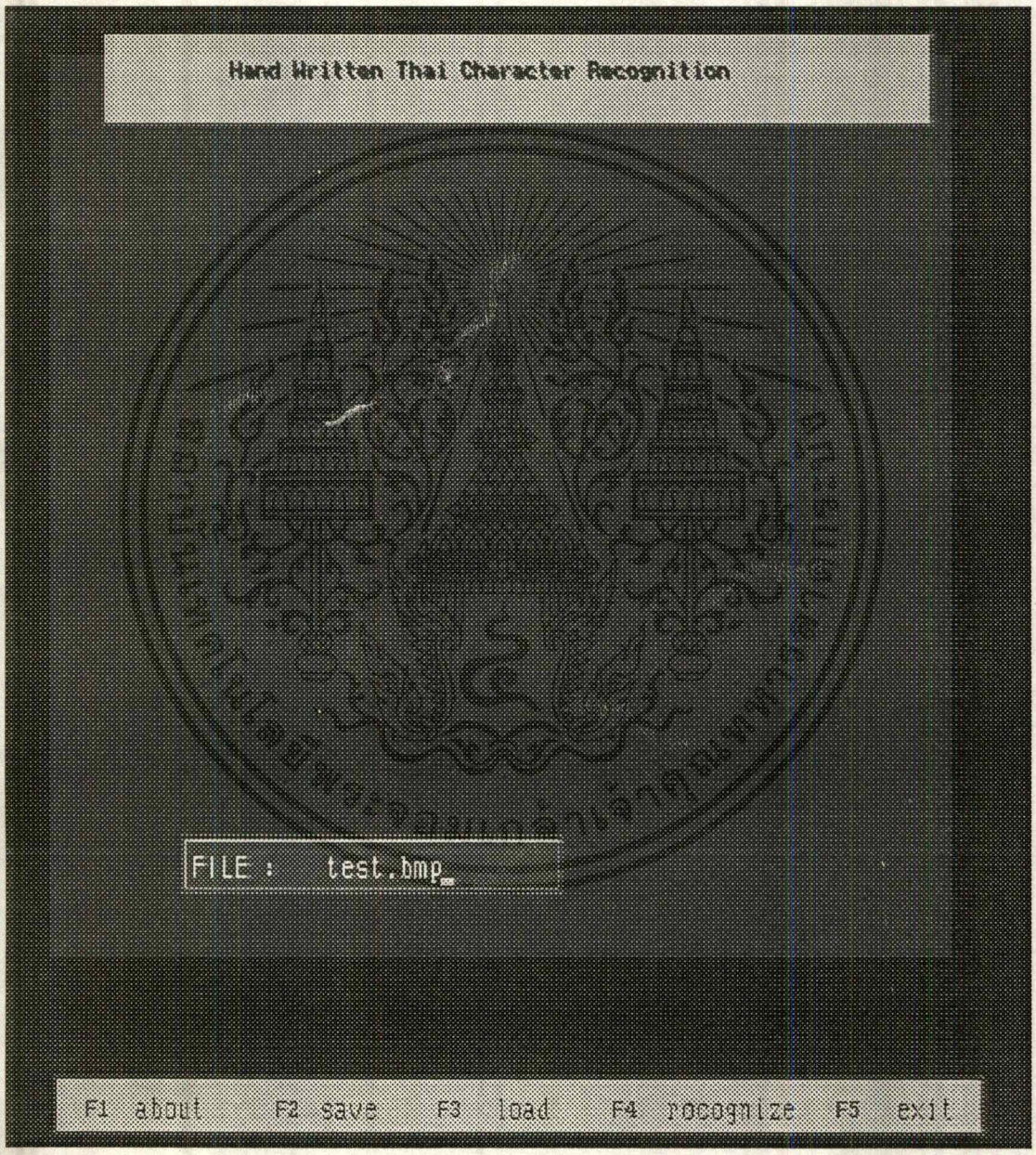


รูปที่ 4.1 แสดงการทำงานของโปรแกรมเมื่อเริ่มต้นทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 การเปิดแฟ้มข้อมูลภาพ <Load>

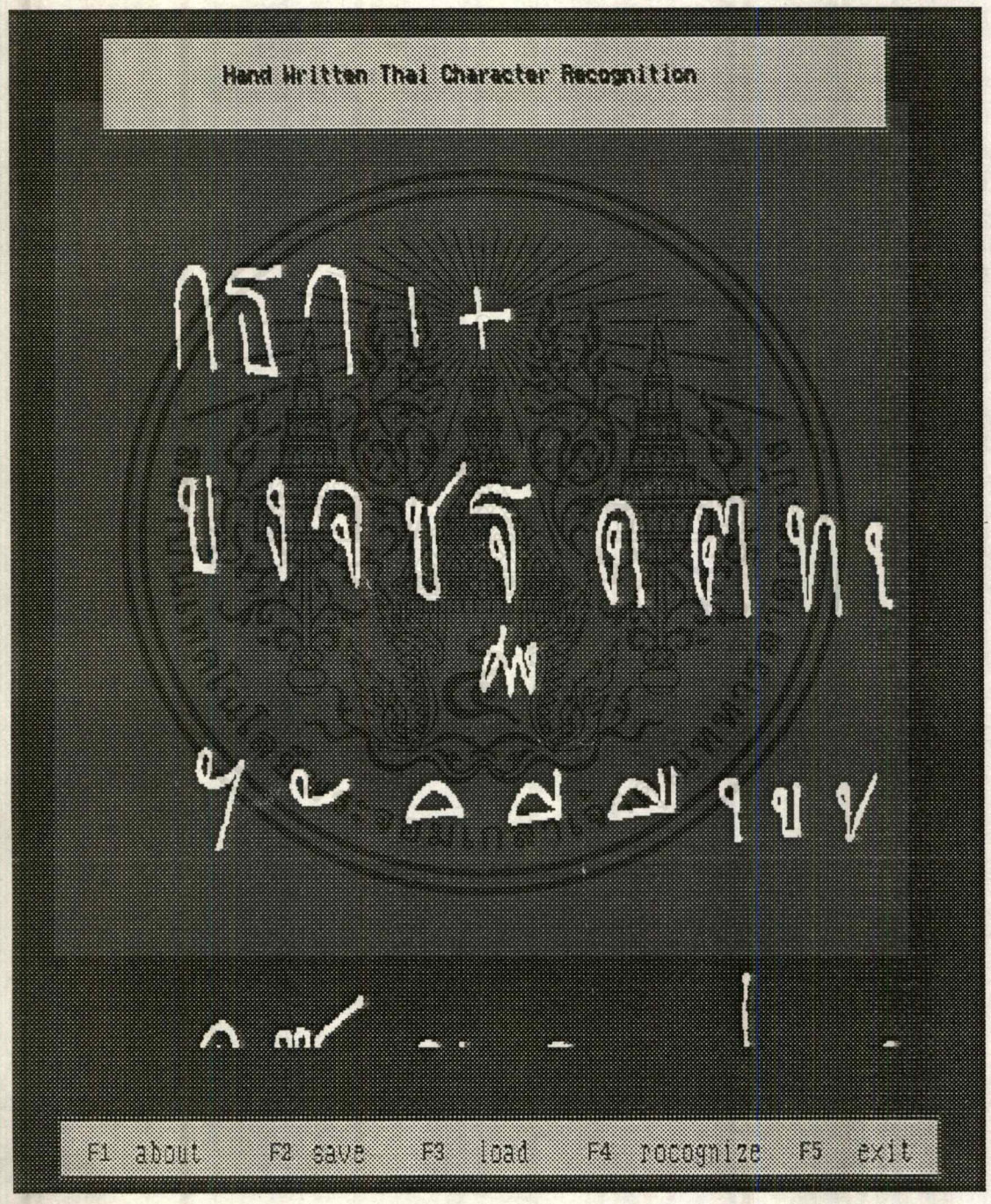
การเปิดแฟ้มข้อมูลภาพ โดยการกดปุ่ม F3 จากนั้นจะมีหน้าต่างขนาดเล็กขึ้นมาให้ทำการใส่ชื่อของภาพที่ต้องการจะเปิดดังตัวอย่าง



รูปที่ 4.2 แสดงการพิมพ์ชื่อแฟ้มข้อมูลภาพเพื่อการเปิด

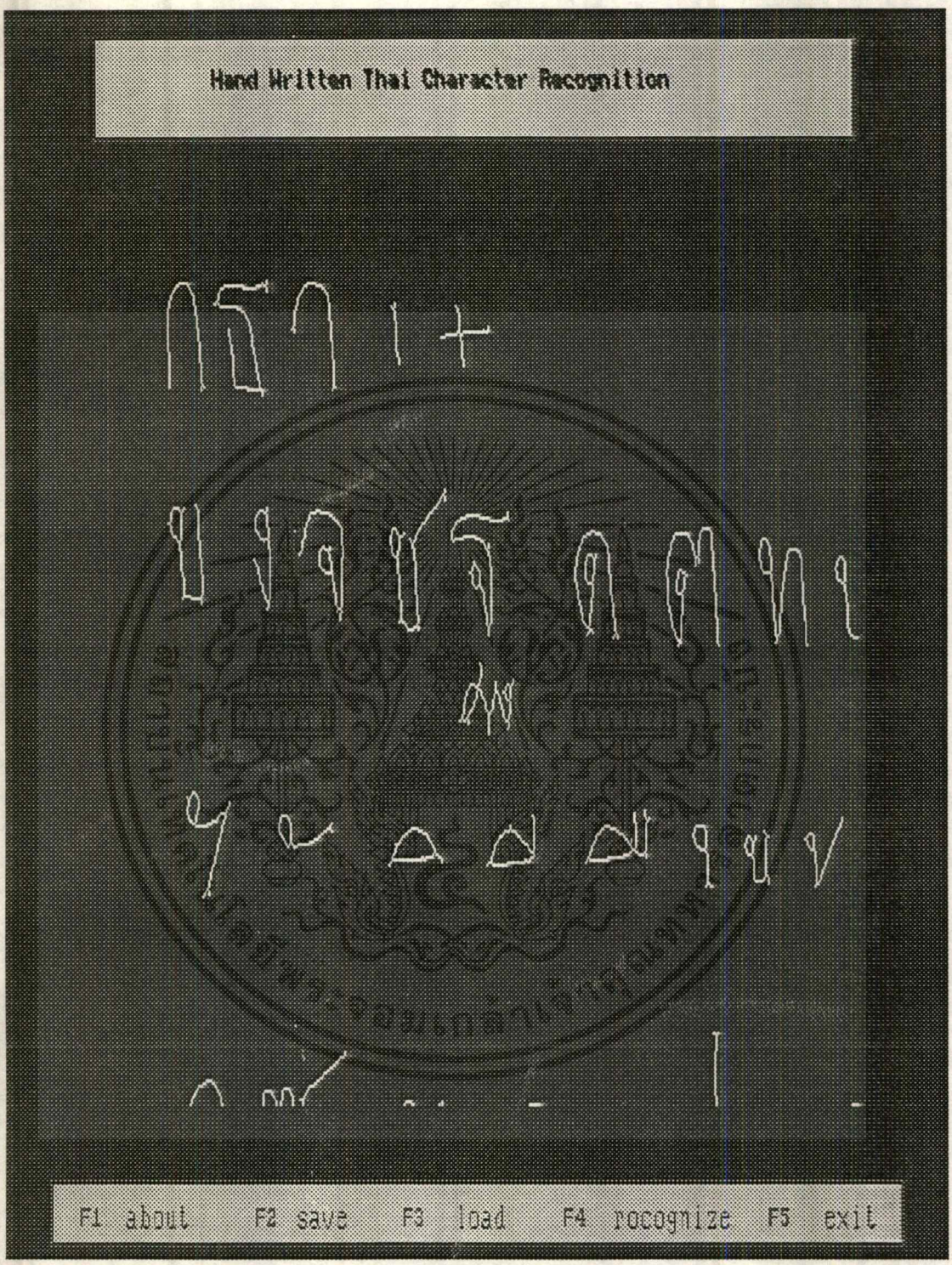
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการใส่ชื่อแฟ้มข้อมูลภาพเรียบร้อยแล้วให้กดปุ่ม enter จากนั้นโปรแกรมจะแสดงภาพแฟ้มข้อมูลที่เปิดขึ้นมา พร้อมทั้งทำการจัดการล่วงหน้าเพื่อให้ภาพที่ได้มีความหนาของลายเส้นเพียงจุดเดียว และมีการกำจัดส่วนเกินต่าง ๆให้เรียบร้อยก่อนการวิเคราะห์ต่อไป



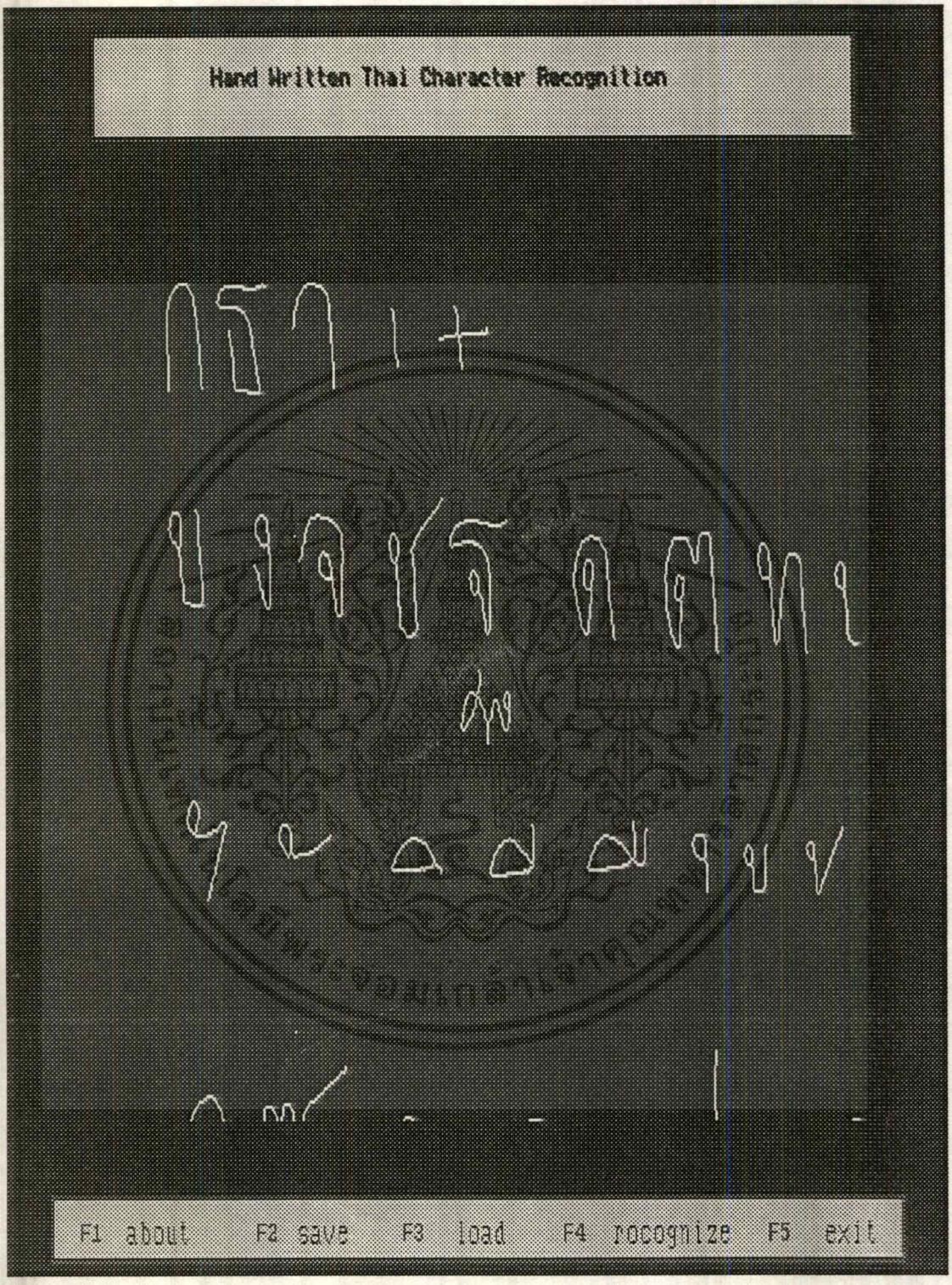
รูปที่ 4.3 แสดงข้อมูลภาพที่ทำการเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงข้อมูลภาพภาพภายหลังการทำให้บาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

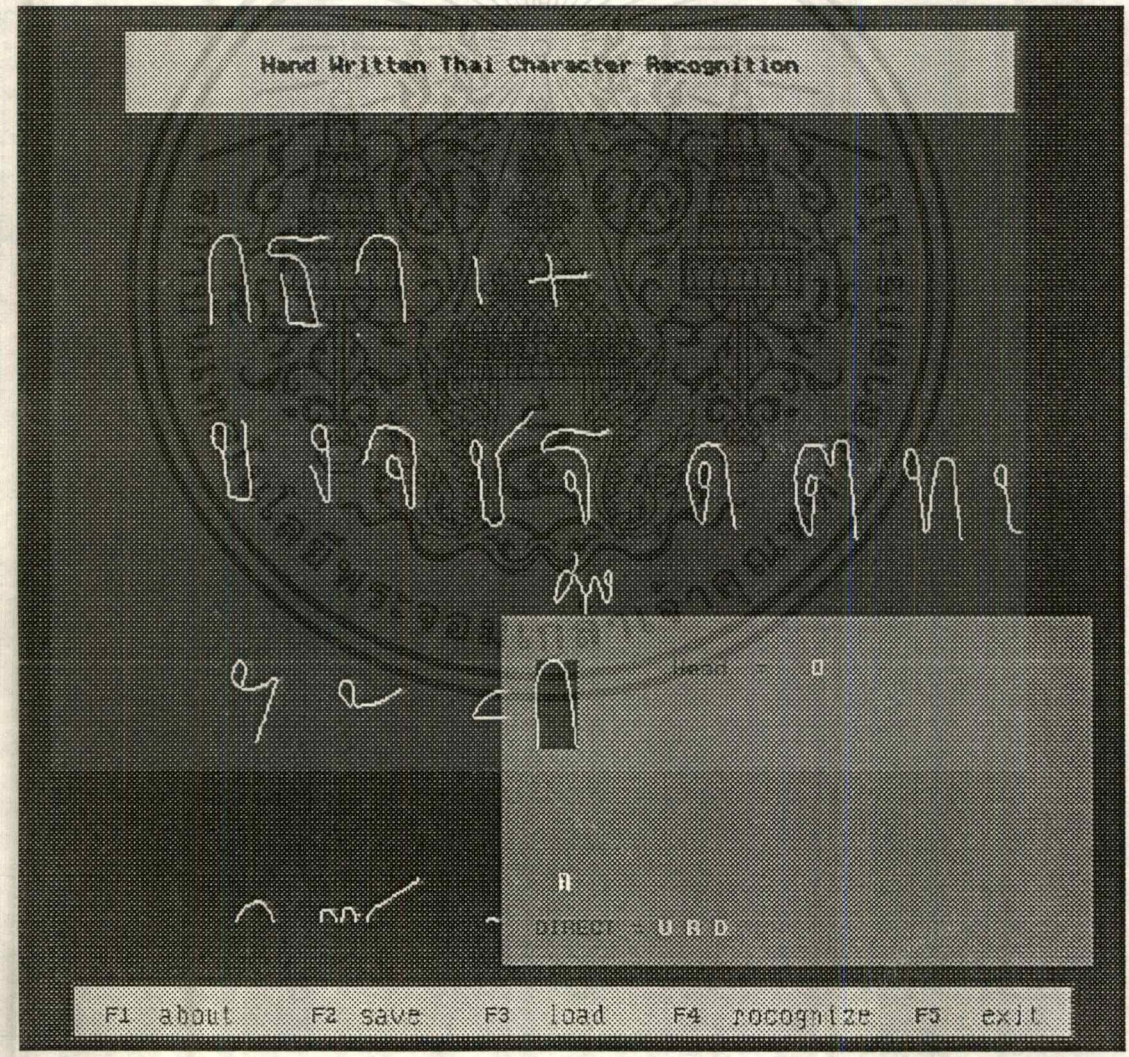


รูปที่ 4.5 แสดงข้อมูลภาพภาพภายหลังการขจัดส่วนเกิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

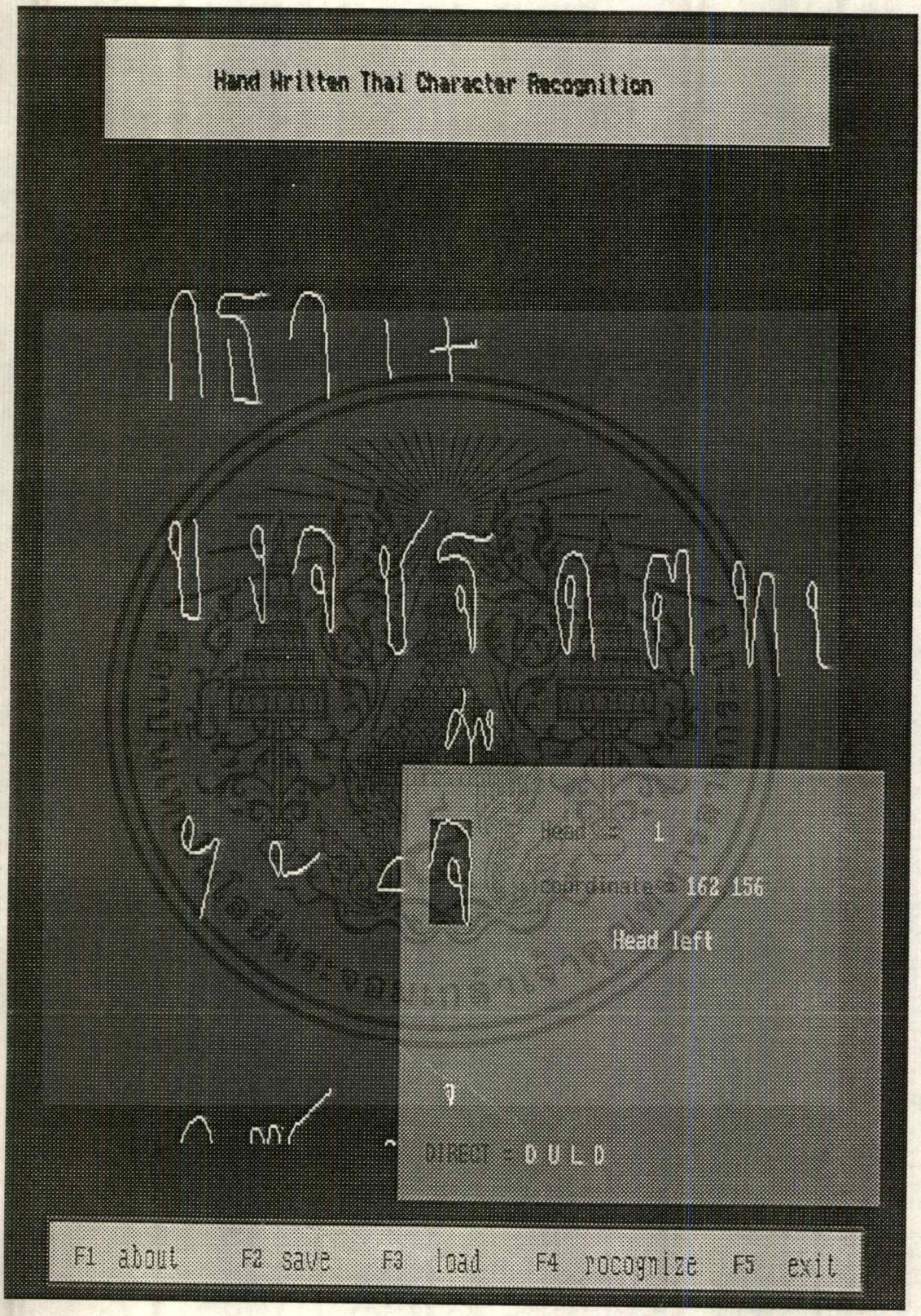
ขั้นตอนที่ 3 การวิเคราะห์ <Recog>

ทำการวิเคราะห์ภาพ โดยการกดปุ่ม F4 ซึ่งจะเป็นการวิเคราะห์ข้อมูลที่เปิดมาแล้วตามขั้นตอนที่ 2 เพื่อทำการแยกตัวอักษรแต่ละตัวมาทำการหาจำนวนหัว ลักษณะของหัว และทิศทางในการเขียนตัวอักษรแต่ละตัวเพื่อระบุว่าเป็นตัวอักษรอะไร โดยการเปรียบเทียบกับฐานข้อมูลที่มีอยู่ ถ้าข้อมูลที่ได้จากการวิเคราะห์ไม่มีอยู่ในฐานข้อมูล ก็จะรอให้ผู้ใช้ระบุลงไปว่าภาพนั้นเป็นตัวอักษรอะไรเพื่อโปรแกรมจะได้นำไปเก็บไว้ในฐานข้อมูลเพิ่มเติม เพื่อใช้ในการวิเคราะห์ครั้งต่อไป



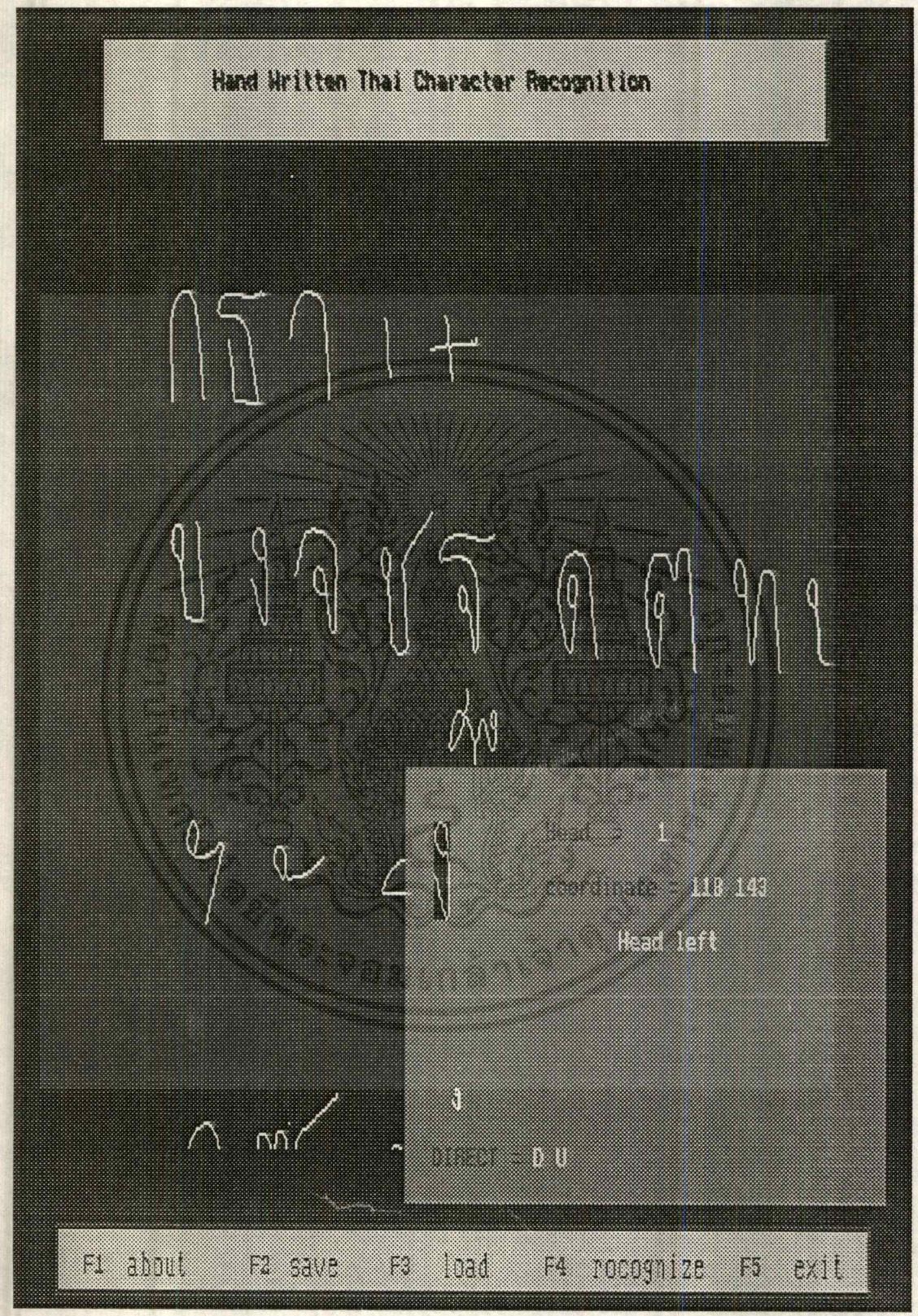
รูปที่ 4.6 แสดงผลการวิเคราะห์ภาพตัวอักษร ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



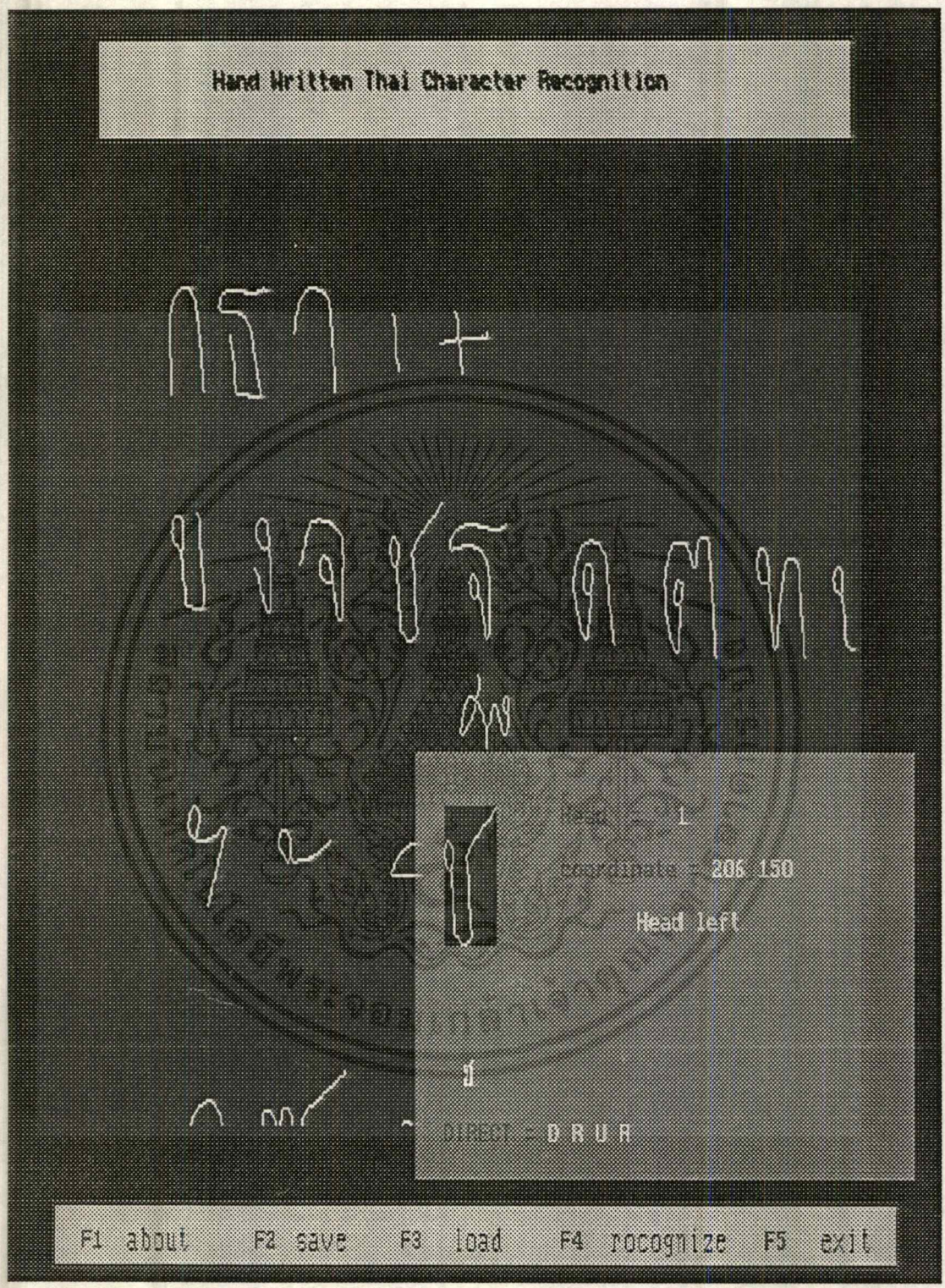
รูปที่ 4.7 แสดงผลการวิเคราะห์ภาพตัวอักษร จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



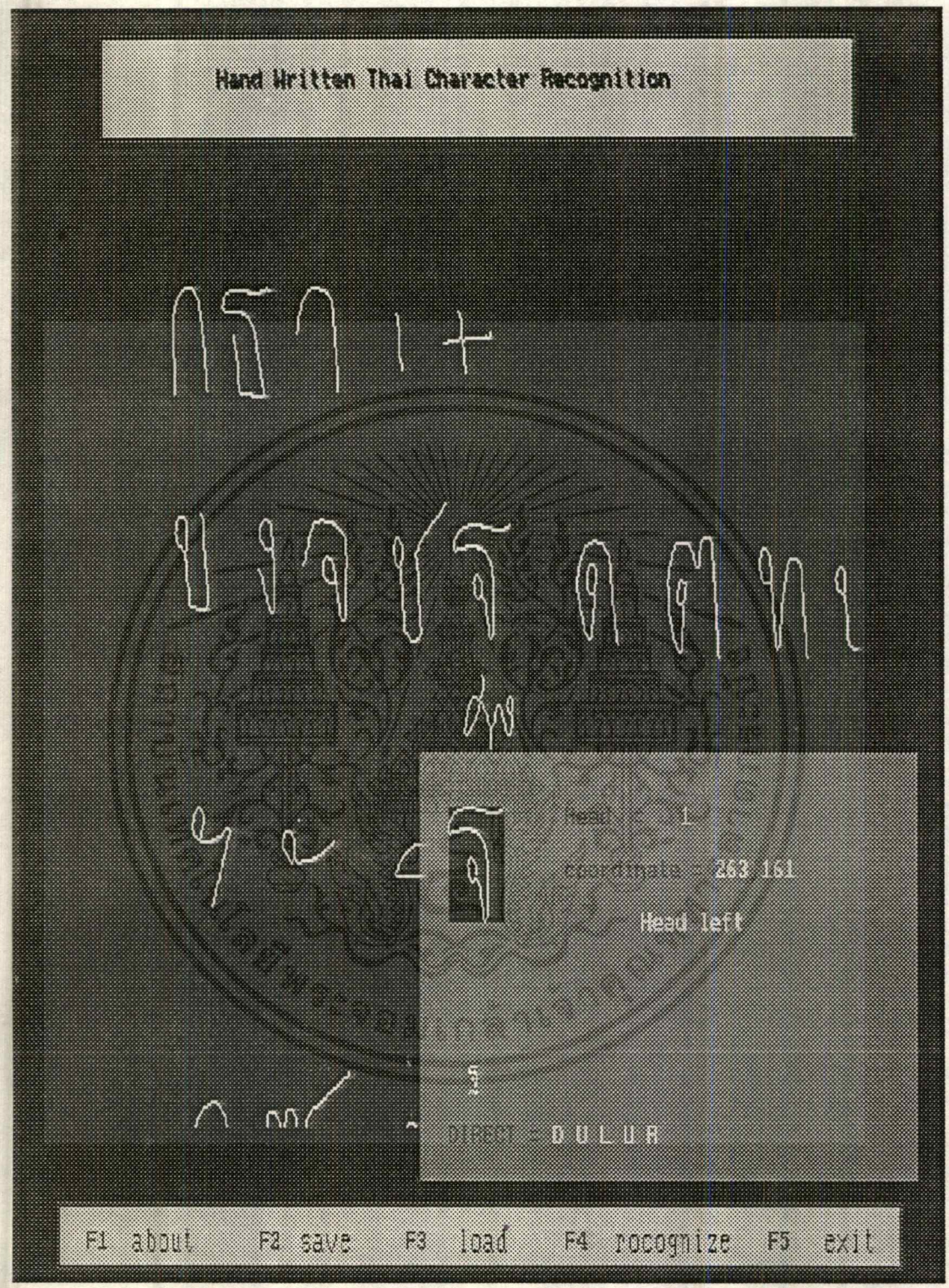
รูปที่ 4.8 แสดงผลการวิเคราะห์ภาพตัวอักษร ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



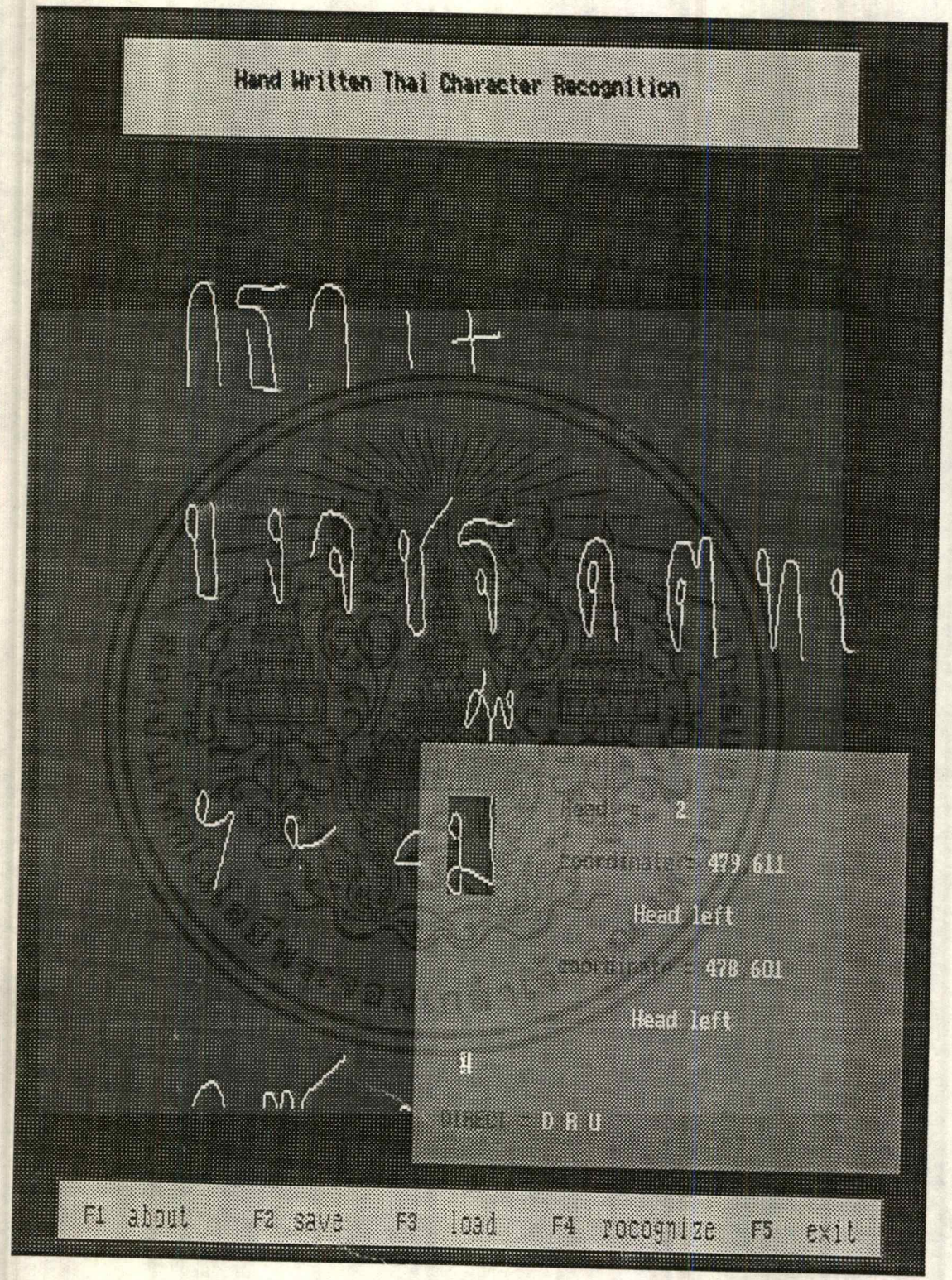
รูปที่ 4.9 แสดงผลการวิเคราะห์ภาพตัวอักษร ช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



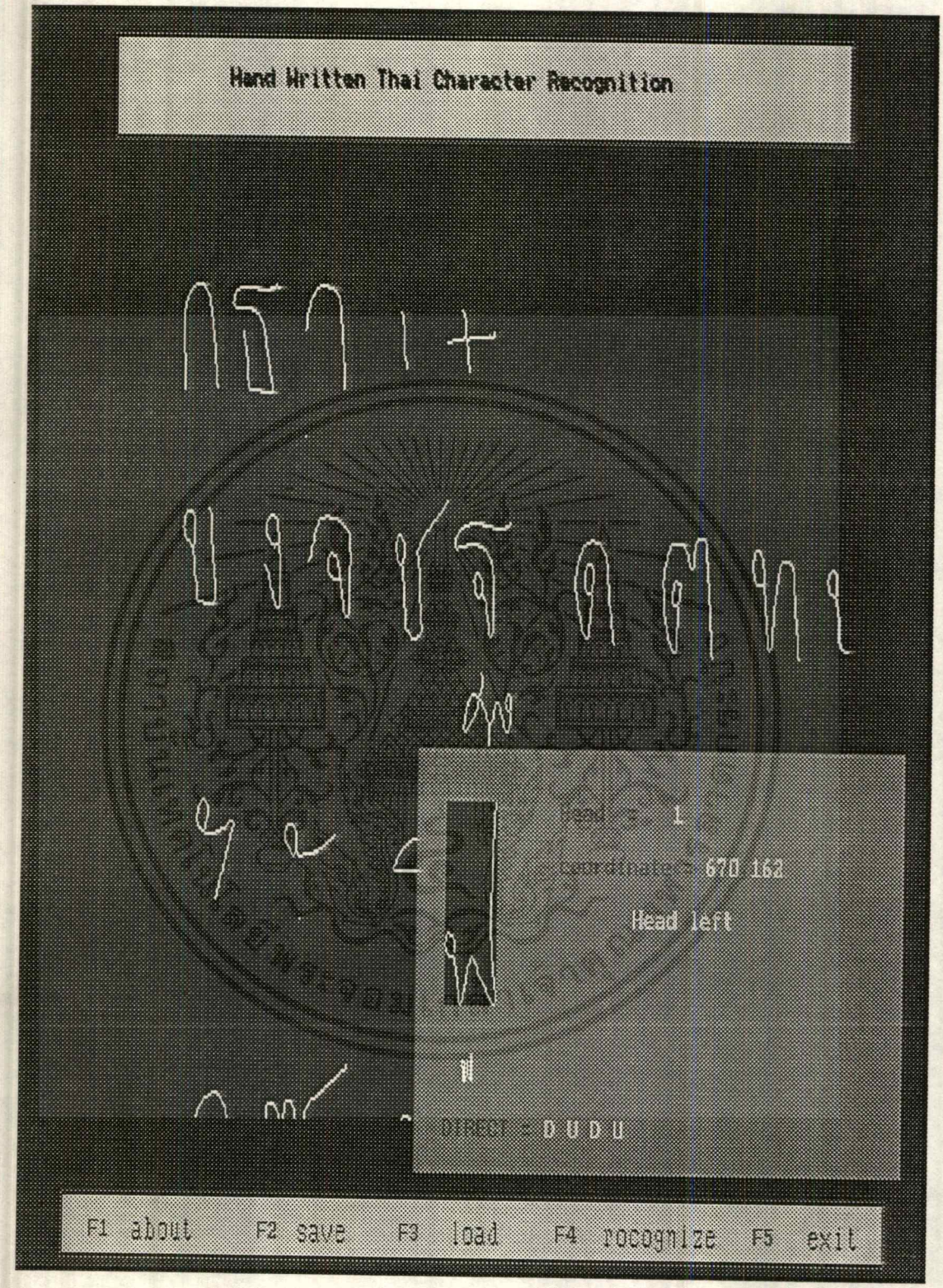
รูปที่ 4.10 แสดงผลการวิเคราะห์ภาพตัวอักษร ฐ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



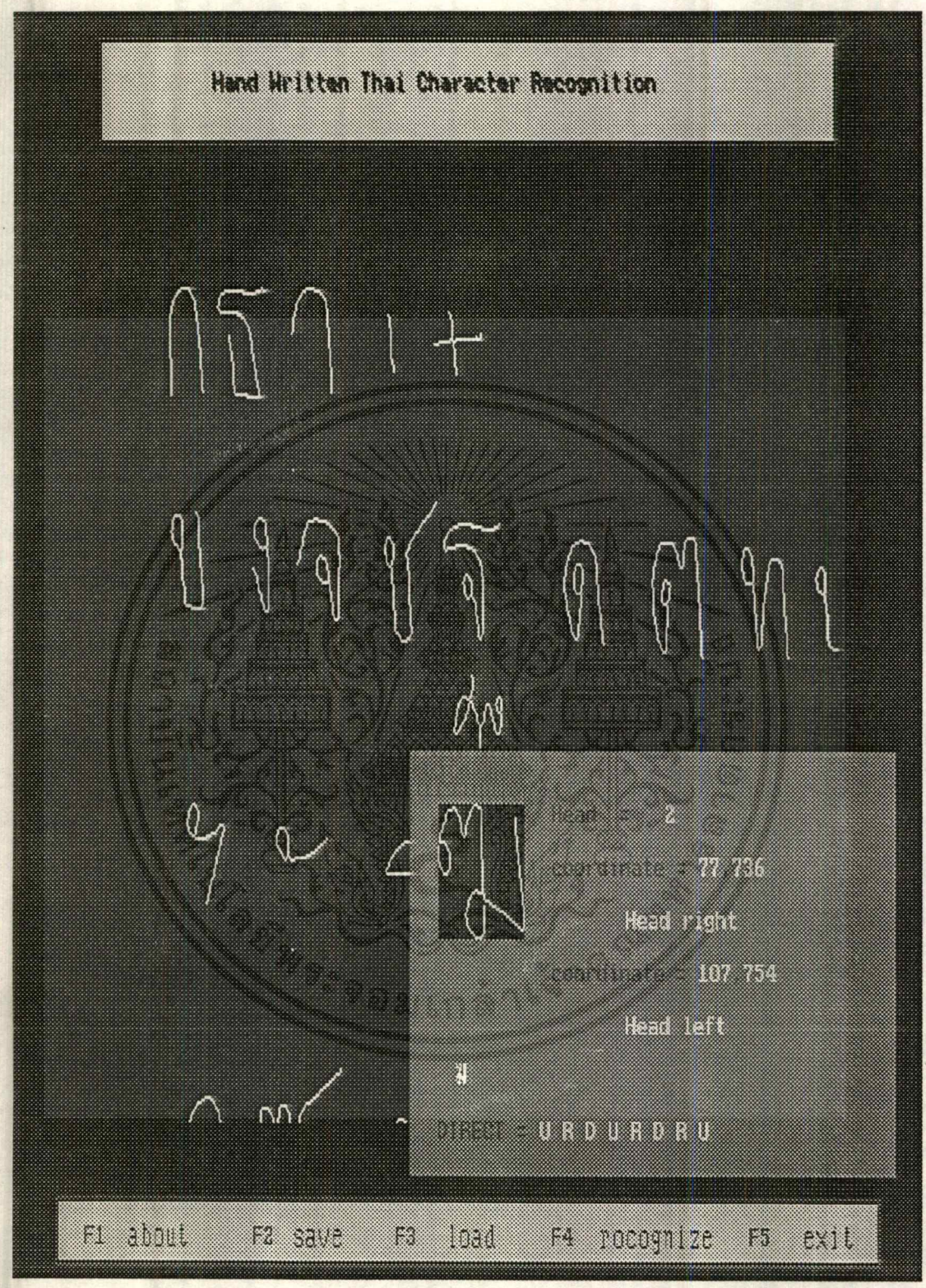
รูปที่ 4.11 แสดงผลการวิเคราะห์ภาพตัวอักษร ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



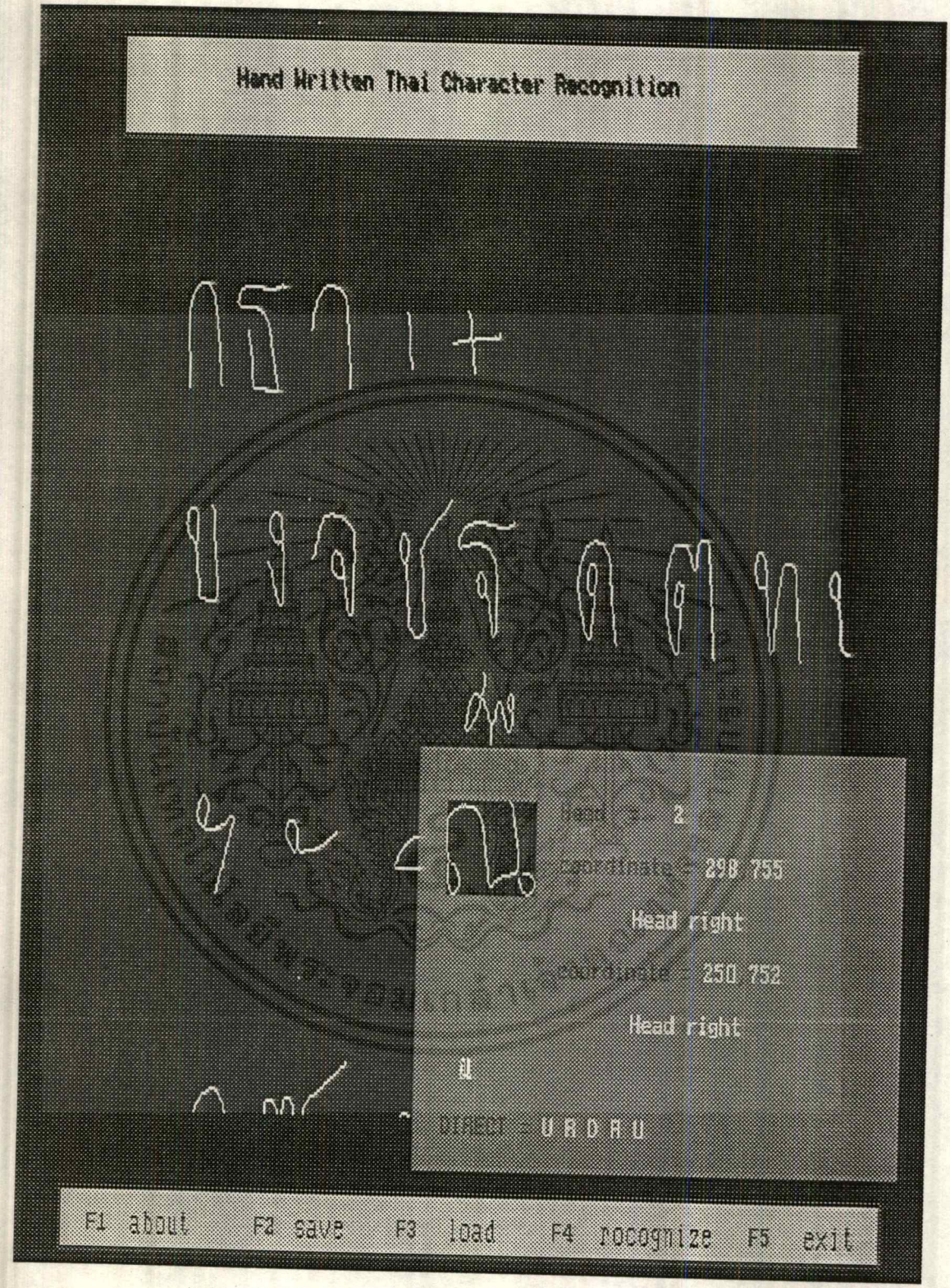
รูปที่ 4.12 แสดงผลการวิเคราะห์ภาพตัวอักษร ฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



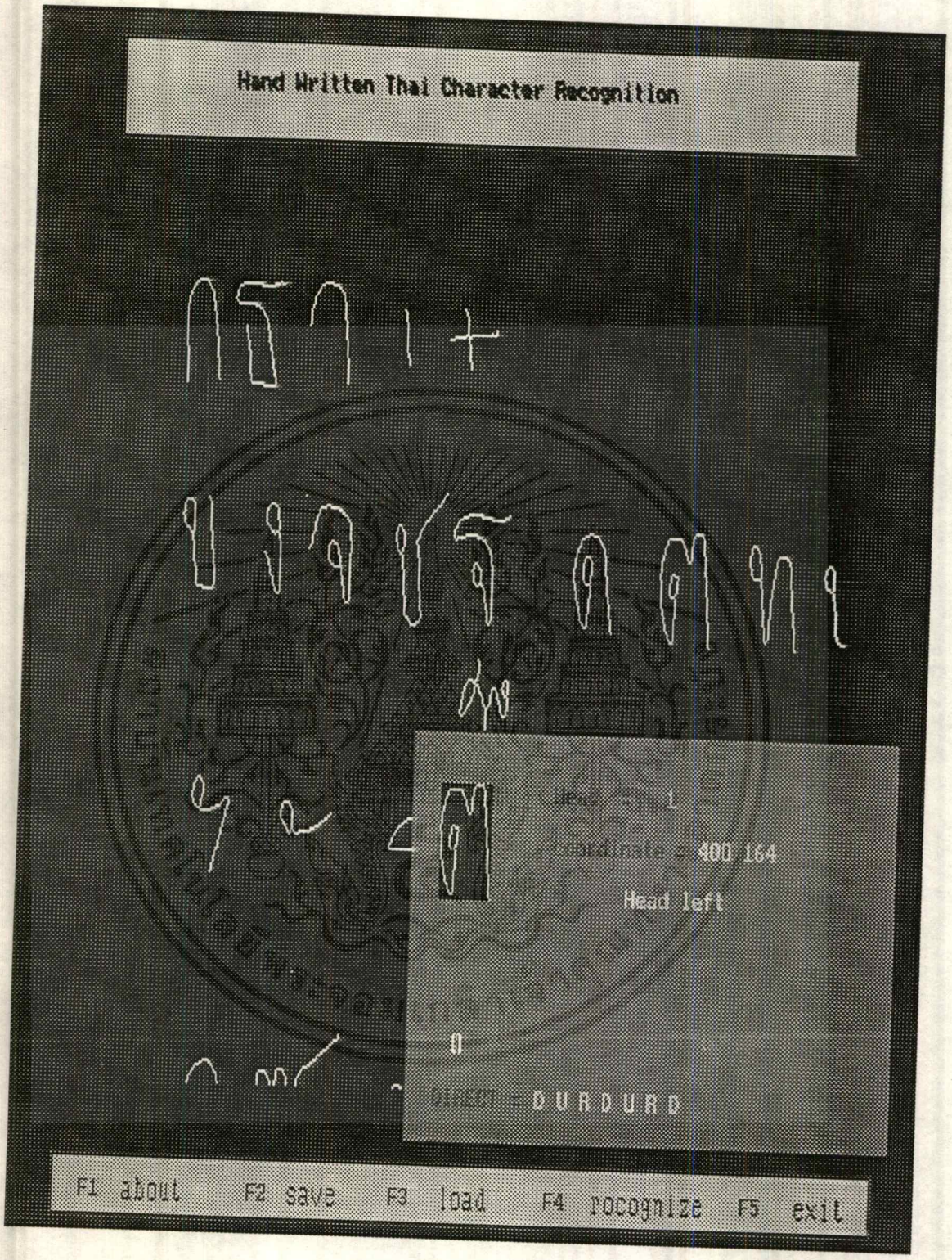
รูปที่ 4.13 แสดงผลการวิเคราะห์ภาพตัวอักษร ฆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงผลการวิเคราะห์ภาพตัวอักษร ณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงผลการวิเคราะห์ภาพตัวอักษร ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 4 การจัดเก็บผลจากการวิเคราะห์ <Save>

หลังจากการวิเคราะห์ ข้อมูลภาพเสร็จสิ้นแล้ว สามารถเก็บผลการวิเคราะห์ ลงแฟ้มข้อมูล โดยการกดปุ่ม F2 ซึ่งจะทำการเก็บ รหัส แอสกี้ ของตัวอักษร และตำแหน่งของตัวอักษร ลงแฟ้มข้อมูลที่ต้องการ เพื่อนำมาจัดเรียงเป็นประโยคในภายหลัง

ขั้นตอนที่ 5 การออกจากการทำงานของโปรแกรม <Exit>

การออกจากการทำงานของโปรแกรม ให้กดปุ่ม F5 ซึ่งโปรแกรมจะสิ้นสุดการทำงานทั้งหมด และออกจากโปรแกรมไปสู่ดอสพรอมท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

ปัญหาและข้อจำกัดในการทำงานของระบบ

1. ปัญหาและข้อจำกัดของข้อมูลอินพุท ลักษณะของข้อมูลอินพุทจะต้องมีการเขียนด้วยตัวบรรจงมีหัวชัดเจนและไม่เขียนตัวอักษรติดกัน

2. ปัญหาและข้อจำกัดของการทำงานส่วนของการจัดการล่วงหน้า คือการทำตัวอักษรให้บาง และการขจัดสัญญาณรบกวนต่างๆ ยังใช้เวลามากอยู่

ข้อเสนอแนะ

1. ฐานความรู้ในการรู้จำตัวอักษรจะต้องมีการเรียนรู้จากอักษรคัดลายมือจำนวนมากๆ เพื่อให้การใช้งานมีประสิทธิภาพมากขึ้น

2. ถ้าสามารถเพิ่มประสิทธิภาพในการจัดการล่วงหน้าให้มีความรวดเร็วยิ่งขึ้นก็จะทำให้เวลาทั้งหมดที่ใช้ในการรู้จำลดลงได้มาก

3. เครื่องไมโครคอมพิวเตอร์ที่ใช้ในการประมวลผล ควรจะมีความเร็วในการประมวลผลสูงเท่าที่จะเป็นไปได้

4. ควรมีพัฒนาเพิ่มเติมให้สามารถทำการวิเคราะห์ตัวอักษรที่เขียนหัวไม่ชัดเจนหรือเขียนแบบไม่มีหัวได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

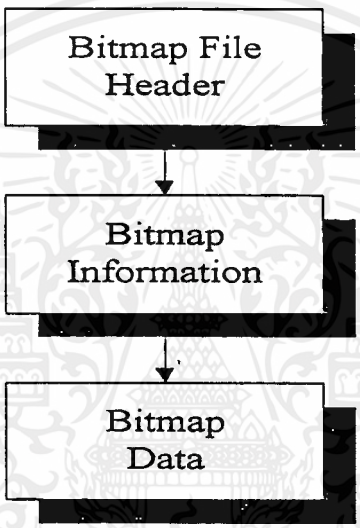
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โครงสร้างของ Bitmap File

ภาพ bitmap จะมีการจัดเก็บตามขนาดของสี ถ้าสีมากจะทำให้ภาพที่จัดเก็บมีขนาดใหญ่ตามไปด้วย ภาพ bitmap จะประกอบไปด้วยส่วนระบุชนิดของภาพ , ส่วนข้อมูลของภาพ , ขนาด , สี , อื่นๆ และส่วนสุดท้ายจะเป็นจุดภาพที่จะนำมาแสดงผล



รูปที่ ก.1 โครงสร้างของภาพ bitmap

Bitmap File Header

Byte #	Data	Details
1 - 2	File type	Must be ASCII text " BM "
3 - 6	Size of the file	In double word (32 bit integer)
7 - 10	Reserved for future use	Must be zero
11 - 14	Byte offset to bitmap data	Offset from the Bitmap File Header (i.e., the start of the file)

ตารางที่ ก.1 Bitmap File Header

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bitmap Information

Byte #	Data	Details
1 - 4	Number of byte in header	ขนาดของส่วนหัว 40 byte
5 - 8	Width of Bitmap	ความกว้างของภาพ (pixel)
9 - 12	Height of bitmap	ความสูงของภาพ (pixel)
13 - 14	Number of color planes	ต้องเท่ากับ 1
15 - 16	Number of bit per pixel	ส่วนที่บอกสีของภาพ 1 = 2 สี , 4 = 16 สี , 8 = 256 สี , 24 = 16M สี
17 - 20	Type of compression	0 = ไม่มีการอัดภาพ 1 = 8 bit per pixel 2 = 4 bit per pixel
21 - 24	Size of image	ไม่ได้ใช้งาน
25 - 28	Horizontal resolution	ไม่ได้ใช้งาน
29 - 32	Vertical resolution	ไม่ได้ใช้งาน
33 - 36	Number of color indexes used by the bitmap	ไม่ได้ใช้งาน
37 - 40	Number of color indexes important for displaying bitmap	ไม่ได้ใช้งาน

ตารางที่ ก.2 Bitmap Information

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bitmap Data

Byte #	Data	Details
41	Blue color value	ค่าสีน้ำเงินของภาพ
42	Green color value	ค่าสีเขียวของภาพ
43	Red color value	ค่าสีแดงของภาพ
44	Reserved for future use	ต้องเป็น 0
...	..... Remaining color palette entries	บอกค่าสีต่างๆ 4 bit เหมือนกันไปจนหมดภาพ

ตารางที่ ก.3 Bitmap Data

การเก็บภาพ bmp จะเก็บมุมล่างซ้ายไปขวา และไล่จนถึงบนสุด

รูปที่ ก.2 แสดงการเก็บภาพของ bmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รหัสแอสกี้ของอักขระในรหัส สมอ

ตัวเลขไทย (0-๙)

รหัสแอสกี้	อักขระ
240	0
241	๑
242	๒
243	๓
244	๔
245	๕
246	๖
247	๗
248	๘
249	๙

อักขระภาษาไทย (ก-ฮ)

รหัสแอสกี้	อักขระ
161	ก
162	ข
163	
164	ค
165	
166	ฆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสแอสกี้	อักขระ
167	ง
168	จ
169	ฉ
170	ช
171	ซ
172	ฌ
173	ญ
174	ฎ
175	ฏ
176	ฐ
177	ฑ
178	ฒ
179	ณ
180	ด
181	ต
182	ถ
183	ท
184	ธ
185	น
186	บ
187	ป
188	ผ
189	ฝ
190	พ
191	ฟ
192	ภ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสแอสกี้	อักขระ
193	ม
194	ย
195	ร
196	ฤ
197	ล
198	
199	ว
200	ศ
201	ษ
202	ส
203	ห
204	ฬ
205	อ
206	ฮ

อักขระพิเศษ

รหัสแอสกี้	อักขระ
207	ๆ
208	ะ
209	ั
210	า
211	ำ
212	ิ
213	ี
214	ึ
215	ื

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสแอสกี้	อักขระ
216	ุ
217	ู
224	เ
225	แ
226	โ
227	ใ
228	ไ
229	ๅ
230	ๆ
231	็
232	่
233	้
234	๊
235	๋
236	์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

โปรแกรมการทำงาน

RECOG.CPP

```
// Written By Nopporn Kongdum 3P 37013299
// Start 01/10/96
// Last Modified 30/03/97
// Written in Borland C++ version 3.1
// Compile in Huge Model

#include <ctype.h>
#include <string.h>
#include <bios.h>
#include <graphics.h>
#include <stdio.h>
#include <mem.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>

#define  HOME   18176
#define  END    20224
#define  CR     7181
#define  ES     283
#define  TAB    3849
#define  DEL    21248
#define  UP     18432
#define  DOWN   20480
#define  LEFT   19200
#define  RIGHT  19712
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define  SP     14624
#define  BS     3592
#define  PU     18688
#define  PD     20736
#define  CRT_PU 33792
#define  CRT_PD 30208

#define  F1     15104
#define  F2     15360
#define  F3     15616
#define  F4     15872
#define  F5     16128
#define  F6     16384
#define  F7     16640
#define  F8     16896
#define  F9     17152
#define  F10    17408

#define  INS    20992
#define  DEL    21248
#define  BS     3592

#define X_RANGE    1200 // 1500
#define Y_RANGE    1800 // 2000
#define Max_Dist   4  // old 8 distorttion  line   length < 10
#define P_Direct   4  // old 3
#define State_Ch   3  // number of point for change direct
#define Start_P    2  // (min=2) start point for new direct
#define X    300      // 380  coordinate for display char
#define Y    300
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define Xb   60
#define Yb   60
#define Xn   580
#define Yn   420

#define U    (unsigned char)0x00
#define D    (unsigned char)0x01
#define R    (unsigned char)0x02
#define L    (unsigned char)0x03

#define ND   (unsigned char)0xFF

#define LEARN   (int)1
#define RECOG   (int)2

extern int  read_image(char *st);
extern void show_image(int left,int up,int rght,int dw,int color,int back);
extern void write_pattern(void);
extern void read_pattern(void);
extern void logo(void);

void about(void);
void learning(void);
void save_ch_list(void);
void load_image(void);
void recognize(void);

unsigned char   ascii[256][20];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char thai_kb[0x54] = {

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

unsigned char thai_kbs[0x54] = {

0x00,0x1B,0x00,0xF1,0xF2,0xF3,0xF4,0xD9,0xDB,0xF5,0xF6,0xF7,0xF8,0xF9,0x08,0x09,
0xF0,0x22,0xAE,0xB1,0xB8,0xEB,0xEA,0xB3,0xCF,0xAD,0xB0,0x2C,0x0D,0x00,0xC4,0xA6,
0xAF,0xE2,0xAC,0xE7,0xEB,0xC9,0xC8,0xAB,0x2E,0x00,0x00,0x7E,0x28,0x29,0xA9,0xCE,
0x25,0xEC,0x3F,0xB2,0xCC,0x3F,0x00,0x2A,0x00,0x20,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xF7,0xF8,0xF9,0x2D,0xF4,0xF5,0xF6,0x2B,0xF1,
0xF2,0xF3,0x30,0x2E };

struct BMPHEAD
{
 unsigned char  id[2];              // file type : must be ASCII text "BM"
 long  filesize;            // Size of teh file : in double words (32 bit integers)
 int   reserved[2];         // Reserved for future use : must be zero
 long  headersize;          // byte offser to bitmap data : offset from the Bitmap file header (i.
e,the startof the file)
 long  infosize;            // number of bytes in header : currenty 40 bytes
 long  width;               // width of bitmap : in pixels
 long  depth;               // height of bitmap : in pixels
 int   blplanes;            // number of color planese : must be set to 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 int   bits;                  // number of bits per pixel : valid choices are 1,4,8,24;if not 24,
determines the size of the palette
 long  blcompression;   // type of compression : {0:no compression}{1:run legth 8 bits per
pixel}{2:run legth 4 bits per pixel}
 long  bisizeimage;       // size of image : in byte
 long  bixpelspermeter; // horizontal resolution : in pixels/meter
 long  biypelspermeter; // vertical   resolution : in pixels/meter
 long  biclrused;// number of color indexes used by the bitmap : zero indicates all color
important
 long  biclrimportant;  // number of color indexes important for displaying bitmap : a value of
size indicates all colors are important
}bmphead;
int    width;
int    Xmax,Ymax;

unsigned char x1,x2,x3,x4,x5,x6,x7,x8;

struct  point{
          int x;
          int y;
};
point   sect_[8],temp[50];

point   hp[250];   //  temp  for display head after delete
int     _hp;       //

struct node {
    int x_origin;   // origin of section
    int y_origin;
    int x_start;    // section which select
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int y_start;
    int x_old;      // old coordinate
    int y_old;
    int x_new;      // new coordinate
    int y_new;
};

node  br[16];
int   cb;

struct head1{
    int  x;         // coordinate of head
    int  y;
    int  y_min;     // y minimum of head
    int  y_max;     // y maximum o head
    int  status;    // status of head
};

head1 head_[8];
int   head;
point  edge_[8];  // point of edge
int    ne;        // number of edge

int   x_min,x_max,y_min,y_max;

struct stack{
        int  x;  // coordinate of section
        int  y;
        int  c_b;  // count of branch
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stack stack_s[8];   // save old section

point  old1[8],old2[8];
point  sect1[8],sect2[8];

unsigned long DIRECT;
int   nd;   //nuber of direct
int   begin_branch;

struct plist{          // Link list of pattern;
          unsigned long  pattern;
          unsigned char character;
          plist  *next;
          plist  *befor;
         };
typedef plist  *PLIST;

PLIST  H0,L0;
PLIST  H1L,L1L;
PLIST  H2L,L2L;
PLIST  H1R,L1R;
PLIST  H2R,L2R;

struct clist{
          unsigned char character;
          int     x;
          int     y;
          clist   *next;
          clist   *befor;
         };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
typedef clist  *CLIST;

CLIST  CH,CL;

int    Lrn_Rcg; // flag for learning or recognize

unsigned char   character;
unsigned char   Ch_learn;

unsigned char huge binary_image[X_RANGE/8][Y_RANGE];

struct time t1,t2;


char *headmsg = "Hand Written Thai Character Recognition";
char *help    = " about       save       load       rocognize       exit";
int  maxx,maxy;
char st[30];

#include "thin.cpp"
#include "lib.cpp"
#include "rec.cpp"


void main()
{
  unsigned ch;
  int driver = DETECT,mode;

   initgraph(&driver,&mode,"");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    readfont();
    logo();

    do
    {
      ch = bioskey(0);
      switch(ch)   {
            case F1: about();
                     break;
            case F2: save_ch_list();
                     break;
            case F3: load_image();
                     break;
            case F4: recognize();
                     break;
            }
    } while(ch!=F5);
    DEL_NODE();
    closegraph();
}

void about(void)
{
    board(50,150,460,300,103,9,3,1,1);
    displaythai(70,170,YELLOW,"Hand Written Thai Chracters Recognition");
    displaythai(70,190,YELLOW,"      Written By Nopporn Kongdun      ");
    displaythai(70,210,YELLOW,"            3P   37013299            ");
    displaythai(70,230,YELLOW," Associate Professor Dr. Chom Kimpan ");
    displaythai(70,250,YELLOW," King's Mongkut Institute of Ladkrabang");
    getch();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   logo();
   show_image(0,0,Xmax,Ymax,WHITE,BLUE);
 }
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>
// function learning()
//          learning input image and save new pattern in database
// image input must write fallow asscii code from 161(Hen)
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>
void learning(void)
{
int  xd,yd;

  load_image();   // load image (BITMAP)

  DEL_NODE();     //  clear memory for database

  read_pattern();    // read database(pattern)

  Lrn_Rcg  = LEARN;   // learning not get from keyboard

  Ch_learn = (unsigned char)161;     // start image for learning

  for(int y=1;y<=Ymax;y++)    // check all point in image
    for(int x=1;x<=Xmax;x++)
     {
       if(X0(x,y))              // if found part of character
          {
            recognize1(x,y);  //  recognize
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        switch (character) {

        case (unsigned char)173:
                xd = x_max - (x_max-x_min)/4;
                yd = y_max+1;
                while(X0(xd,yd)==0)
                    {
                      yd++;
                      if(yd==Ymax) break;
                    }
                del_stroke(xd,yd);
                break;

         case (unsigned char)176:
                xd = x_min + (x_max-x_min)/2;
                yd = y_max+1;
                while(X0(xd,yd)==0)
                    {
                      yd++;
                      if(yd==Ymax) break;
                    }
                del_stroke(xd,yd);
                break;
          }
       }  // end  found character
    }  // end loop
  write_pattern();  // save pattern
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// function save_ch_list()
//          save list of character after recognize to file
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
void save_ch_list(void)
{
FILE *fbmp;
CLIST T;

  getstr(200,350,15,"FILE : ",st);  // get file name

  if((fbmp=fopen(st,"w+b"))!=NULL)       // if cant open file
   {

     for(T=CH;T!=NULL;T=T->next) {
          fwrite(&T->character,1,1,fbmp);   // save  character
          fwrite(&T->x,2,1,fbmp);           // save position of character
          fwrite(&T->y,2,1,fbmp);
     }

     fclose(fbmp);            // close file
   }
}


//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// function load_image()
//          load image file to memmory(binary_image[..][..])
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
void load_image(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
logo();
getstr(200,350,15,"FILE : ",st);   // get file name
setfillstyle(1,YELLOW);
bar(110,348,498,372);
setcolor(RED);

switch(read_image(st))
  {
   case 1: outtextxy(210,360,"File not Found !"); // file not found
           getch();
           setfillstyle(1,BLUE);
           bar(110,348,498,372);
           show_image(0,0,Xmax,Ymax,WHITE,BLUE);
           break;
   case 2: outtextxy(210,360,"Not BMP File !");  // not bmp file
           getch();
           setfillstyle(1,BLUE);
           bar(110,348,498,372);
           show_image(0,0,Xmax,Ymax,WHITE,BLUE);
           break;
   case 3: outtextxy(210,360,"Not 2 gray level !");  // not 2 gray level
           getch();
           setfillstyle(1,BLUE);
           bar(110,348,498,372);
           show_image(0,0,Xmax,Ymax,WHITE,BLUE);
           break;
   case 0: show_image(0,0,Xmax,Ymax,WHITE,BLUE); // can load image
           fill_hole();                  // fill hole
           thinning();                   // thinning
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            show_image(0,0,Xmax,Ymax,WHITE,BLUE); // after thinning
//          write_image(st,"thin.bmp");
            clip_distort();                 // clip distort 2 time
            clip_distort();
            show_image(0,0,Xmax,Ymax,WHITE,BLUE); // after clip distort
//          write_image(st,"clip.bmp");
            break;
    }
}

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// function recognize()
//          recognize all character in image
//          and save  in CH(list of output)
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
void recognize()
{
int xd,yd;

  Lm_Rcg = RECOG;    // recognize

  DEL_NODE();        // clear memory for load pattern and save output

  read_pattern();    // load database(pattern)

  for(int y=1;y<=Ymax;y++)
    for(int x=1;x<=Xmax;x++)
     {
       if(X0(x,y))          // if found parts of character
          {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
recognize1(x,y); // recognize
// add output to CH(list of output after recognize)
ADDC(CH,CL,character,x_min+(x_max-x_min)/2,y_min+(y_max-y_min)/2);
switch (character) {

case (unsigned char)173:
        xd = x_max - (x_max-x_min)/4;
        yd = y_max+1;
        while(X0(xd,yd)==0)
          {
            yd++;
            if(yd==Ymax) break;
          }
        del_stroke(xd,yd);
        break;

case (unsigned char)176:
        xd = x_min + (x_max-x_min)/2;
        yd = y_max+1;
        while(X0(xd,yd)==0)
          {
            yd++;
            if(yd==Ymax) break;
          }
        del_stroke(xd,yd);
        break;

case (unsigned char)209:
        xd = x_min + (x_max-x_min)/2;
        yd = y_max+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                while(X0(xd,yd)==0)
                    {
                      yd++;
                      if(yd==Ymax) break;
                    }
                recognize1(xd,yd);
                if(character==(unsigned char)209)
                      CL->character  = (unsigned char)208;
                else
                 {
                      ADDC(CH,CL,character,x_min+(x_max-x_min)/2,y_min+(y_max-
y_min)/2);
                 }
                break;

          case (unsigned char)224:
                xd = x_min+1;
                yd = y_min+(y_max-y_min)/2;
                while(X0(xd,yd)==0)
                 {
                    xd++;
                    if(xd == Xmax) break;
                  }
                recognize1(xd,yd);
                if(character==(unsigned char)224)
                      CL->character = (unsigned char)225;
                else
                  {
                      ADDC(CH,CL,character,x_min+(x_max-x_min)/2,y_min+(y_max-
y_min)/2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    xd = x_min-1;
                    yd = y_min+(y_max-y_min)/2;
                    while(X0(xd,yd)==0)
                      {
                       xd--;
                       if(xd == 1) break;
                       }
                    recognize1(xd,yd);
                    if(character==(unsigned char)224)
                       (CL->befor)->character = (unsigned char)225;
                    else
                     {
                       ADDC(CH,CL,character,x_min+(x_max-x_min)/2,y_min+
(y_max-y_min)/2);
                       }
                     }
                   break;
              } //end switch
         } // end if
     } // end for
   write_pattern();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REC.CPP

```
// Written By Nopporn Kongdum 3P 37013299
// Start 01/10/96
// Last Modified 30/3/97

#define   true  (unsigned char)0x01
#define   false (unsigned char)0x00
#define   h_right  0x01  // status of head
#define   h_left   0x00
#define   HIGH_CH  8     // for detect character which same pattern

extern unsigned char get_kb(int x,int y,int color,int back);
extern unsigned char x1,x2,x3,x4,x5,x6,x7,x8;
extern unsigned char X0(int &x,int &y);
extern unsigned char X1_X8(int &x,int &y);
extern void clear_b(int &x,int &y);
extern void set_b(int &x,int &y);
extern int  Xmax,Ymax;
extern void show_int(int value);
extern void displaychar(int x, int y ,int color, int ch);
extern void show_win(int lft,int up,int rght,int dw,int bx,int by,
                     int color,int back,int type);

extern node  br[16];
extern int   cb;
extern head1 head_[8];
extern int   head;
extern PLIST H0,L0;
extern PLIST H1L,L1L;
extern PLIST H2L,L2L;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
extern PLIST H1R,L1R;
extern PLIST H2R,L2R;
extern CLIST CHAR;
extern int   n_ch;
extern unsigned char  character;
extern unsigned char  Ch_learn;
extern int   Lrn_Rcg;
extern point edge_[8];
extern int   ne;
extern unsigned long DIRECT;
extern int   nd;
extern int   x_min,x_max,y_min,y_max;
extern stack stack_s[8];   // save old section
extern point  old1[8],old2[8];   // save old point which delete
extern point  sect1[8],sect2[8];  // save branch of section

extern point  hp[250];
extern int    _hp;

void  del_head(int x,int y);
void  del_stroke(int x,int y);
unsigned char line_direct(int &x,int &y);
void  recognize3();
void  recognize2();
void  recognize1(int &x,int &y);
int   find_start(int &x,int &y); // if return 1 input is loop
void  check_head(int &n);
void  addnode(int x_or,int y_or,int x_s,int y_s,int x_o,int y_o,int x_n,int y_n);
void  delnode(int n);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function  recognize1(x,y)
//           Analysis chracter by find head status of head and direct
//           of line stroke
// Input
//             x,y -> point of found image charater
// Output
//           charracter  -> alphabet of image charater
//           x_min,y_min -> x,y minimum of character
//           x_max,y_max -> x,y maximum of character
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
void recognize1(int &x,int &y)
{
   int  x_or,y_or,x_s,y_s;
   int  x_or2,y_or2,x_s2,y_s2,x_n2,y_n2;
   int  n1,n2;  //  count old point which delete(for restor)
   int  b1,b2;  //  count branch of section;
   int  r1,r2;  //  used for for loop
   int  cp ;  // count point;

  // oldvect = getvect(0x1c);
  // setvect(0x1c,newint);

 START:

   _hp=0; // >>>>>>>>>>>>>>>for display head after delete only<<<<<<<<<<<<<

   ne =  0; // number of edge

   if(find_start(x,y)) ; //if return 1 then character input is loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
x_min = x_max = x;
y_min = y_max = y;

head = 0;  // count head

cb = 0;
cp = 0;

addnode(x,y,x,y,x,y,x,y); // first point

while(cb>0)
{
 for(int n=1;n<=cb;)
 {
  cp++;     // increment  point
  clear_b(br[n].x_old,br[n].y_old);  // don't use old point for search

  if((br[n].x_new)>x_max) x_max=br[n].x_new;
  else if((br[n].x_new)<x_min) x_min=br[n].x_new;
  if((br[n].y_new)>y_max) y_max=br[n].y_new;
  else if((br[n].y_new)<y_min) y_min=br[n].y_new;

  X1_8(br[n].x_new,br[n].y_new);   // find neighbour (8 direct)

  b1 = 0;       // find new branch
  if(x1) { b1++;sect1[b1].x=(br[n].x_new)+1;sect1[b1].y=(br[n].y_new)-1;}
  if(x2) { b1++;sect1[b1].x=(br[n].x_new)  ;sect1[b1].y=(br[n].y_new)-1;}
  if(x3) { b1++;sect1[b1].x=(br[n].x_new)-1;sect1[b1].y=(br[n].y_new)-1;}
  if(x4) { b1++;sect1[b1].x=(br[n].x_new)-1;sect1[b1].y=(br[n].y_new);}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(x5) { b1++;sect1[b1].x=(br[n].x_new)-1;sect1[b1].y=(br[n].y_new)+1;}
if(x6) { b1++;sect1[b1].x=(br[n].x_new)  ;sect1[b1].y=(br[n].y_new)+1;}
if(x7) { b1++;sect1[b1].x=(br[n].x_new)+1;sect1[b1].y=(br[n].y_new)+1;}
if(x8) { b1++;sect1[b1].x=(br[n].x_new)+1;sect1[b1].y=(br[n].y_new);}

if(b1==0) {   // don't have path for new search  [edge]
     set_b(br[n].x_old,br[n].y_old);  // restor old point
     ne++;
     edge_[ne].x = br[n].x_new;
     edge_[ne].y = br[n].y_new;
     delnode(n);
 }

 else if(b1==1)  {  // have 1 path
   set_b(br[n].x_old,br[n].y_old);  // restore old point
   br[n].x_old = br[n].x_new;  br[n].y_old = br[n].y_new;
   br[n].x_new = sect1[1].x; br[n].y_new = sect1[1].y;
   check_head(n);
 }

 else // is case of found  section
 {
   n1 = 1; old1[n1].x = br[n].x_old; old1[n1].y = br[n].y_old;
   clear_b(old1[n1].x,old1[n1].y);  // clear_b point which found section
   ++n1; old1[n1].x = br[n].x_new; old1[n1].y = br[n].y_new; // save point
   clear_b(old1[n1].x,old1[n1].y);  // clear_b point which found section
   // test for find orgin of section
   if(x1&x2&x3) { //if current is not origin change to origin
        --(br[n].y_new); clear_b(br[n].x_new,br[n].y_new);
        if((br[n].y_new)>y_max) y_max=br[n].y_new;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else if((br[n].y_new)<y_min) y_min=br[n].y_new;
        ++n1; old1[n1].x = br[n].x_new; old1[n1].y = br[n].y_new;
    }
    else if(x3&x4&x5) { //if current is not origin change
        --(br[n].x_new); clear_b(br[n].x_new,br[n].y_new);
        if((br[n].x_new)>x_max) x_max=br[n].x_new;
        else if((br[n].x_new)<x_min) x_min=br[n].x_new;
        ++n1; old1[n1].x = br[n].x_new; old1[n1].y = br[n].y_new;
    }
    else if(x5&x6&x7) { //if current is not origin change
        ++(br[n].y_new); clear_b(br[n].x_new,br[n].y_new);
        if((br[n].y_new)>y_max) y_max=br[n].y_new;
        else if((br[n].y_new)<y_min) y_min=br[n].y_new;
        ++n1; old1[n1].x = br[n].x_new; old1[n1].y = br[n].y_new;
    }
    else if(x7&x8&x1) { //if current is not origin change
        ++(br[n].x_new); clear_b(br[n].x_new,br[n].y_new);
        if((br[n].x_new)>x_max) x_max=br[n].x_new;
        else if((br[n].x_new)<x_min) x_min=br[n].x_new;
        ++n1; old1[n1].x = br[n].x_new; old1[n1].y = br[n].y_new;
    }
    if((x1&x2&x3)||(x3&x4&x5)||(x5&x6&x7)||(x7&x8&x1)) {   // if change origin find new
branch again
        X1_8(br[n].x_new,br[n].y_new);
        b1 = 0;
        if(x1) { b1++;sect1[b1].x=(br[n].x_new)+1;sect1[b1].y=(br[n].y_new)-1;}
        if(x2) { b1++;sect1[b1].x=(br[n].x_new)  ;sect1[b1].y=(br[n].y_new)-1;}
        if(x3) { b1++;sect1[b1].x=(br[n].x_new)-1;sect1[b1].y=(br[n].y_new)-1;}
        if(x4) { b1++;sect1[b1].x=(br[n].x_new)-1;sect1[b1].y=(br[n].y_new);}
        if(x5) { b1++;sect1[b1].x=(br[n].x_new)-1;sect1[b1].y=(br[n].y_new)+1;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(x6) { b1++;sect1[b1].x=(br[n].x_new)  ;sect1[b1].y=(br[n].y_new)+1;}
        if(x7) { b1++;sect1[b1].x=(br[n].x_new)+1;sect1[b1].y=(br[n].y_new)+1;}
        if(x8) { b1++;sect1[b1].x=(br[n].x_new)+1;sect1[b1].y=(br[n].y_new);}
} // end find  origin of section and new branch

x_or = br[n].x_new;   //  set_b origin of section
y_or = br[n].y_new;

delnode(n);

for(r1=1;r1<=b1;r1++)   //  clear_b all branch befor find new branch
    clear_b(sect1[r1].x,sect1[r1].y);

for(r1=1;r1<=b1;r1++) {   // find new branch  of all branch
    x_s = sect1[r1].x; y_s = sect1[r1].y;
    if(x_s>x_max) x_max=x_s;
    else if(x_s<x_min) x_min=x_s;
    if(y_s>y_max) y_max=y_s;
    else if(y_s<y_min) y_min=y_s;

    X1_8(x_s,y_s);

    b2 = 0;
    if(x1) { b2++;sect2[b2].x=(x_s)+1;sect2[b2].y=(y_s)-1;}
    if(x2) { b2++;sect2[b2].x=(x_s)  ;sect2[b2].y=(y_s)-1;}
    if(x3) { b2++;sect2[b2].x=(x_s)-1;sect2[b2].y=(y_s)-1;}
    if(x4) { b2++;sect2[b2].x=(x_s)-1;sect2[b2].y=(y_s);}
    if(x5) { b2++;sect2[b2].x=(x_s)-1;sect2[b2].y=(y_s)+1;}
    if(x6) { b2++;sect2[b2].x=(x_s)  ;sect2[b2].y=(y_s)+1;}
    if(x7) { b2++;sect2[b2].x=(x_s)+1;sect2[b2].y=(y_s)+1;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(x8) { b2++;sect2[b2].x=(x_s)+1;sect2[b2].y=(y_s);}
if(b2==1) {    // if new branch is not section add start node
   addnode(x_or,y_or,x_s,y_s,x_s,y_s,sect2[1].x,sect2[1].y);
   n++;
 }
else // in case of found section after branch from old section
 {
   x_or2 = x_s;  y_or2 = y_s;
   n2 = 0;
   if(x1&x2&x3) {
        --(y_or2); clear_b(x_or2,y_or2);
        if((y_or2)>y_max) y_max=y_or2;
        else if((y_or2)<y_min) y_min=y_or2;
        ++n2; old2[n2].x = x_or2; old2[n2].y = y_or2;
        }
   else if(x3&x4&x5) {
        --(x_or2); clear_b(x_or2,y_or2);
        if((x_or2)>x_max) x_max=x_or2;
        else if((x_or2)<x_min) x_min=x_or2;
        ++n2; old2[n2].x = x_or2; old2[n2].y = y_or2;
        }
   else if(x5&x6&x7) {
        ++(y_or2); clear_b(x_or2,y_or2);
        if((y_or2)>y_max) y_max=y_or2;
        else if((y_or2)<y_min) y_min=y_or2;
        ++n2; old2[n2].x = x_or2; old2[n2].y = y_or2;
     }
   else if(x7&x8&x1) {
        ++(x_or2); clear_b(x_or2,y_or2);
        if((x_or2)>x_max) x_max=x_or2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else if((x_or2)<x_min) x_min=x_or2;
        ++n2; old2[n2].x = x_or2; old2[n2].y = y_or2;
    }
if((x1&x2&x3)||(x3&x4&x5)||(x5&x6&x7)||(x7&x8&x1)) {
        X1_8(x_or2,y_or2);
        b2 = 0;
        if(x1) { b2++;sect2[b2].x=(x_or2)+1;sect2[b2].y=(y_or2)-1;}
        if(x2) { b2++;sect2[b2].x=(x_or2)  ;sect2[b2].y=(y_or2)-1;}
        if(x3) { b2++;sect2[b2].x=(x_or2)-1;sect2[b2].y=(y_or2)-1;}
        if(x4) { b2++;sect2[b2].x=(x_or2)-1;sect2[b2].y=(y_or2);}
        if(x5) { b2++;sect2[b2].x=(x_or2)-1;sect2[b2].y=(y_or2)+1;}
        if(x6) { b2++;sect2[b2].x=(x_or2)  ;sect2[b2].y=(y_or2)+1;}
        if(x7) { b2++;sect2[b2].x=(x_or2)+1;sect2[b2].y=(y_or2)+1;}
        if(x8) { b2++;sect2[b2].x=(x_or2)+1;sect2[b2].y=(y_or2);}
    }

for(r2=1;r2<=b2;r2++)
    clear_b(sect2[r2].x,sect2[r2].y);

for(r2=1;r2<=b2;r2++) {
    x_s2 = sect2[r2].x; y_s2 = sect2[r2].y;
    if(x_s2>x_max) x_max=x_s2;
    else if(x_s2<x_min) x_min=x_s2;
    if(y_s2>y_max) y_max=y_s2;
    else if(y_s2<y_min) y_min=y_s2;
    X1_8(x_s2,y_s2);
    if(x1) { x_n2=x_s2+1;y_n2=y_s2-1;}
    if(x2) { x_n2=x_s2  ;y_n2=y_s2-1;}
    if(x3) { x_n2=x_s2-1;y_n2=y_s2-1;}
    if(x4) { x_n2=x_s2-1;y_n2=y_s2;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(x5) { x_n2=x_s2-1;y_n2=y_s2+1;}
            if(x6) { x_n2=x_s2  ;y_n2=y_s2+1;}
            if(x7) { x_n2=x_s2+1;y_n2=y_s2+1;}
            if(x8) { x_n2=x_s2+1;y_n2=y_s2;}
            addnode(x_or2,y_or2,x_s2,y_s2,x_s2,y_s2,x_n2,y_n2);
            n++;
          }  // end for loop r2

        for(r2=1;r2<=b2;r2++)
            set_b(sect2[r2].x,sect2[r2].y);

        for(r2=1;r2<=n2;r2++)
            set_b(old2[r2].x,old2[r2].y);

       } // end in case of found new section after branch from section

    } // end loop r1 (loop all branch which have

  for(r1=1;r1<=b1;r1++)  //  restor all neighbour of section
      set_b(sect1[r1].x,sect1[r1].y);

  for(r1=1;r1<=n1;r1++)  //  restor old point  and orgin of section
     set_b(old1[r1].x,old1[r1].y);

  } // end  case of section
 }; // end for search new point for all node
};// end while

x = x_max; y = y_min;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if(cp<Max_Dist) { //    distortion   not character
        del_stroke(br[1].x_origin,br[1].y_origin);  // delete distort
        for(;y<=Ymax;y++)              // and find new character
          for(x=1;x<=Xmax;x++)
             if(X0(x,y))  goto START;
  }
  else {   // chracter
     recognize2();  // find direct pattern of character
     recognize3();  // find pattern of character in database
  }
//  setvect(0x1c,oldvect);
}

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function  recognize2
//           find direct pattern of character
// Input
//           edge_[..]  -> point of all edge of character
//           head_[..]  -> point of head and status of head
// Output
//
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
void  recognize2()
{
  int x,y; // start find
  int c,j;
  int n = 0; //top stack;
  int status_;
//  int begin_branch;
  unsigned long direct;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  unsigned long PATTERN;

// >>>>>>>>>>>>>>>>>>>>DISPLAY<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
   setfillstyle(1,CYAN);
   bar(X-20,Y-20,X+320,Y+140);
   show_win(x_min,y_min,x_max,y_max,WHITE,BLUE,1);
// >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
   for(j=1;j<=_hp;j++)     // DIPLAY HEAD
      putpixel(hp[j].x-x_min+X,hp[j].y-y_min+Y,WHITE);
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
   setcolor(RED);
   moveto(X+80,Y);
   outtext("Head  =  ");
   setcolor(YELLOW);
   show_int(head);
   for(int r1=1,r2=Y+20;r1<=head;r1++,r2+=20) {
      moveto(X+80,r2);
      setcolor(RED);
      outtext("coordinate = ");
      setcolor(YELLOW);
      show_int(head_[r1].x);
      setcolor(RED);
      outtext(",");
      setcolor(YELLOW);
      show_int(head_[r1].y);
      r2+=20;
      moveto(X+100,r2);
      setcolor(YELLOW);
      if(head_[r1].status==h_left)
            outtext("    Head left");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    else   outtext("    Head right");
  }
  moveto(X,Y+120);
  setcolor(RED);
  outtext("DIRECT = ");
  setcolor(YELLOW);
// >>>>>>>>>>>>>>>>>>>>DISPLAY<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
  if(head==0) {
     x = edge_[1].x;
     y = edge_[1].y;
     for(j=2;j<=ne;j++)
          if((edge_[j].x) < x) {
             x = edge_[j].x;
             y = edge_[j].y;
          }
  }
  else if(head==1) {
     x = head_[1].x;
     y = head_[1].y;
  }
  else  {
     X1_8(head_[1].x,head_[1].y);
     if((x1+x2+x3+x4+x5+x6+x7+x8)==1) {
          X1_8(head_[2].x,head_[2].y);
          if((x1+x2+x3+x4+x5+x6+x7+x8)==1) {
               if(head_[1].x<head_[2].x)  {
                    x = head_[1].x;
                    y = head_[1].y;
               }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                else {
                        x = head_[2].x;
                        y = head_[2].y;
                        status_ = head_[1].status;
                        head_[1].status = head_[2].status;
                        head_[2].status = status_;
                    }

            }
        else {
                x = head_[1].x;
                y = head_[1].y;
            }
      }
     else {
        x = head_[2].x;
        y = head_[2].y;
        status_ = head_[1].status;
        head_[1].status = head_[2].status;
        head_[2].status = status_;
     }
  }

  nd = 0;  //  number of direct;

  do{
        clear_b(x,y);

// >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>.
//putpixel(x-x_min+X,y-y_min+Y,BLUE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//delay(10);
// >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

        X1_8(x,y);
        c = (int)(x1+x2+x3+x4+x5+x6+x7+x8);
        begin_branch = 0;

        if(c>1) {    //  section

            if(x1&x2&x3) --y;
            else if(x3&x4&x5) --x;
            else if(x5&x6&x7) ++y;
            else if(x7&x8&x1) ++x;

            clear_b(x,y);
            X1_8(x,y);

            if(x8) { n++;temp[n].x=x+1;temp[n].y=y;clear_b(temp[n].x,temp[n].y);}
            if(x7) { n++;temp[n].x=x+1;temp[n].y=y+1;clear_b(temp[n].x,temp[n].y);}
            if(x6) { n++;temp[n].x=x  ;temp[n].y=y+1;clear_b(temp[n].x,temp[n].y);}
            if(x5) { n++;temp[n].x=x-1;temp[n].y=y+1;clear_b(temp[n].x,temp[n].y);}
            if(x4) { n++;temp[n].x=x-1;temp[n].y=y;clear_b(temp[n].x,temp[n].y);}
            if(x3) { n++;temp[n].x=x-1;temp[n].y=y-1;clear_b(temp[n].x,temp[n].y);}
            if(x2) { n++;temp[n].x=x  ;temp[n].y=y-1;clear_b(temp[n].x,temp[n].y);}
            if(x1) { n++;temp[n].x=x+1;temp[n].y=y-1;clear_b(temp[n].x,temp[n].y);}

            x = temp[n].x; y = temp[n].y;  --n;
            c = 1;
            begin_branch = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    if(c==0) {
        if(n>0) {
          x = temp[n].x; y = temp[n].y;  --n;
          c =1;
          begin_branch = 1;
         }
    }

    if(c==1) {
        direct = (unsigned long)line_direct(x,y);
// >>>>>>>>>>>>>>>>>>          >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>.
        if((direct!=(unsigned long)ND)&&(nd<32)) {
            if(nd==0) {
                PATTERN = direct;
                nd+=2;
            }
            else if((direct!=(unsigned long)(PATTERN>>(nd-2)))
                ||(begin_branch)) {
                    PATTERN|=(unsigned long)(direct<<nd);
                    nd+=2;
            }
        }
    }while(c!=0);

/*  if((PATTERN&(unsigned long)0x03)==(unsigned long)0x01)  {   // | start D
    if((PATTERN&(unsigned long)0xfff)==(unsigned long)0x989)  {   // |_/-\_|
        PATTERN>>=12;
        PATTERN<<=6;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PATTERN|=(unsigned long)0x11;
        nd-=6;
    }
else if((PATTERN&(unsigned long)0x3ff)==(unsigned long)0x189)  {   // |_/-\
        PATTERN>>=10;
        PATTERN<<=6;
        PATTERN|=(unsigned long)0x11;
        nd-=4;
    }
else if((PATTERN&(unsigned long)0x3ff)==(unsigned long)0x261)  {   // |/-\_|
        PATTERN>>=10;
        PATTERN<<=6;
        PATTERN|=(unsigned long)0x11;
        nd-=4;
    }
else if((PATTERN&(unsigned long)0xff)==(unsigned long)0x91)  {   // |/\_|
        PATTERN>>=8;
        PATTERN<<=6;
        PATTERN|=(unsigned long)0x11;
        nd-=2;
    }
else if((PATTERN&(unsigned long)0xff)==(unsigned long)0x61)  {   // |/-\
        PATTERN>>=8;
        PATTERN<<=6;
        PATTERN|=(unsigned long)0x11;
        nd-=2;
    }
else if((PATTERN&(unsigned long)0xff)==(unsigned long)0x49)  {   // |_/\
        PATTERN>>=8;
        PATTERN<<=6;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PATTERN|=(unsigned long)0x11;
            nd-=2;
        }

    else if((PATTERN&(unsigned long)0xff)==(unsigned long)0x89) {   //\_/-
            PATTERN>>=8;
            PATTERN<<=4;
            PATTERN|=(unsigned long)0x01;
        }
    else if((PATTERN&(unsigned long)0x3f)==(unsigned long)0x21) {   //\/-
PATTERN>>=8;
            PATTERN>>=6;
            PATTERN<<=4;
            PATTERN|=(unsigned long)0x01;
            nd-=2;
        }
    else if((PATTERN&(unsigned long)0x3f)==(unsigned long)0x09) {   //\_/
            PATTERN>>=6;
            PATTERN<<=4;
            PATTERN|=(unsigned long)0x01;
            nd-=2;
        }
    else if((PATTERN&(unsigned long)0xff)==(unsigned long)0xd9) {   //\_/
            PATTERN>>=8;
            PATTERN<<=6;
            PATTERN|=(unsigned long)0x39;
            nd-=2;
        }

    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if((PATTERN&(unsigned long)0x03)==(unsigned long)0x02)  {   // - start R

  if((PATTERN&(unsigned long)0x3ff)==(unsigned long)0x226)  { // -\_/-
        PATTERN>>=10;
        PATTERN<<=4;
        PATTERN|=(unsigned long)0x01;
        nd-=6;
  }
  else if((PATTERN&(unsigned long)0xff)==(unsigned long)0x26)  {  // -\_/
        PATTERN>>=8;
        PATTERN<<=4;
        PATTERN|=(unsigned long)0x01;
        nd-=4;
  }
  else if((PATTERN&(unsigned long)0x3f)==(unsigned long)0x06)  {  // -\/
        PATTERN>>=6;
        PATTERN<<=4;
        PATTERN|=(unsigned long)0x01;
        nd-=2;
  }
}

else if((PATTERN&(unsigned long)0x03)==(unsigned long)0x00)  {   // | start U

  if((PATTERN&(unsigned long)0x3ff)==(unsigned long)0x0c8)  {  //
        PATTERN>>=10;
        PATTERN<<=6;
        PATTERN|=(unsigned long)0x38;
        nd-=4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    else if((PATTERN&(unsigned long)0xff)==(unsigned long)0xc8)  {  //
            PATTERN>>=8;
            PATTERN<<=6;
            PATTERN|=(unsigned long)0x38;
            nd-=2;
        }
    else if((PATTERN&(unsigned long)0xff)==(unsigned long)0x38)  {  // -\/
            PATTERN>>=8;
            PATTERN<<=6;
            PATTERN|=(unsigned long)0x38;
            nd-=2;
        }
    }
*/
//   PATTERN |= (unsigned long)(0xffffffff<<nd);
    PATTERN |= (direct<<30);

    DIRECT = PATTERN;

    for(c=0;c<nd;c+=2) {
            unsigned  char  a;
            a = (unsigned char)((PATTERN&((unsigned long)(0x03)<<c))>>(c));
            if(a==U)  outtext("U ");
            else if(a==D)  outtext("D ");
            else if(a==L)  outtext("L ");
            else if(a==R)  outtext("R ");
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function   recognize3
//            analysis character by used head status of head and direct
//
// Input
//            head   -> head of character
//            head_[..]  -> coordinate and status of character
//            DIRECT     -> direct pattern of character
//      Database
//            H0  -> list patern of no head character
//            H1L -> list patern of 1 head character and head status = left
//            H1R -> list patern of 1 head character and head status = right
//            H2L -> list patern of 2 head character and head status = left
//            H2R -> list patern of 2 head character and head status = right
// Output
//            character -> ascii code of image character
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
void recognize3()
{
 int   i,found=0;
 PLIST T;

   if(head == 0) {
          for(T=H0;T!=NULL;T=T->next)
               if(T->pattern==DIRECT) {
                         found = 1;
                         character = T->character;
                         break;
                    }
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if(head == 1) {
        if(head_[1].status==h_left) {
            for(T=H1L;T!=NULL;T=T->next)
                if(T->pattern==DIRECT) {
                    found = 1;
                    character = T->character;
                    break;
                }
        }
        else {
            for(T=H1R;T!=NULL;T=T->next)
                if(T->pattern==DIRECT) {
                    found = 1;
                    character = T->character;
                    break;
                }
        }
 }
 else if(head==2) {
        if(head_[1].status==h_left) {
            for(T=H2L;T!=NULL;T=T->next)
                if(T->pattern==DIRECT) {
                    found = 1;
                    character = T->character;
                    break;
                }
        }
        else {
            for(T=H2R;T!=NULL;T=T->next)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if(T->pattern==DIRECT) {
                    found = 1;
                    character = T->character;
                    break;
                }
            }
}

if(found == 0)  {
 if(Lrn_Rcg == RECOG)
   {
       character = get_kb(X+20,Y+90,WHITE,CYAN);
   }
 else
   {
       character = Ch_learn;

       switch (Ch_learn)  {
          case (unsigned char)162 :  Ch_learn=(unsigned char)164;
                                     break;
          case (unsigned char)164 :  Ch_learn=(unsigned char)166;
                                     break;
          case (unsigned char)207 :  Ch_learn=(unsigned char)209;
                                     break;
          case (unsigned char)217 :  Ch_learn=(unsigned char)224;
                                     break;
          case (unsigned char)224 :  Ch_learn=(unsigned char)226;
                                     break;
          case (unsigned char)228 :  Ch_learn=(unsigned char)230;
                                     break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            case (unsigned char)236 :   Ch_learn=(unsigned char)240;
                                        break;
            default :     Ch_learn++;
        }
    }

if(head==0) {
        ADDP(H0,L0,DIRECT,character);
}
else if(head==1) {
        if(head_[1].status==h_left) {
                ADDP(H1L,L1L,DIRECT,character);
        }
        else {
                ADDP(H1R,L1R,DIRECT,character);
        }
       }
else if(head==2) {
        if(head_[1].status==h_left) {
                ADDP(H2L,L2L,DIRECT,character);
        }
        else {
                ADDP(H2R,L2R,DIRECT,character);
        }
       }
 }

 switch (character) {
       case (unsigned char)162:    // k
                    if(y_min<(head_[1].y_min-HIGH_CH))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                character = (unsigned char)187;   // p
            else {
                float ratio = (float)(((float)y_max-(float)y_min)/
                                        ((float)x_max-(float)x_min));
                if(ratio<1.5)
                    character = (unsigned char)186;  //  b
            }
            break;

case (unsigned char)187:    // p
            if(y_min>(head_[1].y_min-HIGH_CH)) {
                float ratio = (float)(((float)y_max-(float)y_min)/
                                        ((float)x_max-(float)x_min));
                if(ratio<1.5)
                    character = (unsigned char)186;  //  b
                else
                    character = (unsigned char)162;  // k
            }
            break;

case (unsigned char)186:    // k
            if(y_min<(head_[1].y_min-HIGH_CH))
                character = (unsigned char)187;   // p
            else {
                float ratio = (float)(((float)y_max-(float)y_min)/
                                        ((float)x_max-(float)x_min));
                if(ratio>1.5)
                    character = (unsigned char)162;  //  b
            }
            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case (unsigned char)167:
        if(y_min<head_[1].y_min)
                character = (unsigned char)168;
        break;

case (unsigned char)168:
        if(y_min==head_[1].y_min)
                character = (unsigned char)167;
        break;

case (unsigned char)172:
        if(head_[2].status==h_right)
                character = (unsigned char)179;
        break;

case (unsigned char)179:
        if(head_[2].status==h_left)
                character = (unsigned char)172;
        break;

case (unsigned char)182:
        if(y_max>(head_[1].y_max+HIGH_CH))
                character = (unsigned char)196;
        break;

case (unsigned char)196:
        if(y_max<(head_[1].y_max+HIGH_CH))
                character = (unsigned char)182;
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case (unsigned char)185:
        if(head_[2].status==h_left)
                character = (unsigned char)193;
        break;

case (unsigned char)193:
        if(head_[2].status==h_right)
                character = (unsigned char)185;
        break;

case (unsigned char)188:
        if(y_min<(head_[1].y_min-HIGH_CH))
                character = (unsigned char)189;
        break;

case (unsigned char)189:
        if(y_min>(head_[1].y_min-HIGH_CH))
                character = (unsigned char)188;
        break;

case (unsigned char)190:
        if(y_min<(head_[1].y_min-HIGH_CH))
                character = (unsigned char)191;
        break;

case (unsigned char)191:
        if(y_min>(head_[1].y_min-HIGH_CH))
                character = (unsigned char)190;
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

    displaychar(X+20,Y+90,WHITE,character);
    delay(350);
    displaychar(X+20,Y+90,BLUE,character);
}

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function Find_Start(x,y)
//          Find start point for search Head and scope of character
//          start point must Edge or origin of Section
// Input
//      x,y -> point of found Character(Black point)
// Output
//      x,y -> coordinate of Edge or origin of section of character
// Return code
//      1 -> input character is circle(loop)
//      0 -> not loop
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
int find_start(int &x,int &y)
{
  int x_start=x, y_start=y;          // old coordinate
  int x_old = x, y_old = y;
  int branch=0;        // count of neighbour

    X1_8(x,y);          // don't use old point

    branch = (int)(x1+x2+x3+x4+x5+x6+x7+x8); // branch = 1 is edge
    if(branch!=1)    // if in line or section find start of line or
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{               // origin of section
 if(x1) {  x++; y--; }    // set_b new point = new neighbour
 else if(x2) { y--; }     // protect = xy start
 else if(x3) { x--;y--; } // in loop do if xy_new = xy start is loop;
 else if(x4) { x--; }
 else if(x5) { x--;y++; }
 else if(x6) { y++; }
 else if(x7) { x++;y++; }
 else if(x8) { x++; }

do {
    clear_b(x_old,y_old); // clear_b old point
                          // for find status of neighbour
    X1_8(x,y);          // don't use old point

    set_b(x_old,y_old);  // restor old point in image

    x_old = x;                    // old point = current point
    y_old = y;

    branch = (int)(x1+x2+x3+x4+x5+x6+x7+x8);

    if(branch==1)           // if new neighbour = 1
     {
      if(x1) {  x++; y--; }   // set_b new point = new neighbour
      else if(x2) { y--; }
      else if(x3) { x--;y--; }
      else if(x4) { x--; }
      else if(x5) { x--;y++; }
      else if(x6) { y++; }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            else if(x7) { x++;y++; }
            else if(x8) { x++; }
            if((x==x_start)&&(y==y_start))    return(1); // is head only
          }
        else if(branch>1){
            if(x1&&x2) { y--; }           // find origin of section
            else if(x2&&x3) {  y--; }
            else if(x3&&x4) {  x--; }
            else if(x4&&x5) {  x--; }
            else if(x5&&x6) {  y++; }
            else if(x6&&x7) {  y++; }
            else if(x7&&x8) {  x++; }
            else if(x8&&x1) {  x++; }
        }
      } while(branch==1); //  repeat until found edge(cn=0) or section(cn>1)
    }
    if((branch==1)||(branch==0)) {  //  found edge
        ne++;
        edge_[ne].x = x ; edge_[ne].y = y;
    }
    return(0);
}
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function  addnode(x_or,y_or,x_s,y_s,x_o,y_o,x_n,y_n)
//          add new branch for search
// Input
//           x_or,y_or -> origin of this branched(or section)
//           x_s,y_s   ->  start point of this branched
//           x_o,y_o   ->  old point which searched
//           x_n,y_n   ->  new point for search
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Output
//          cb      ->  count of all branched(must increment)
//          br[..]  ->  save all branched
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
void addnode(int x_or,int y_or,int x_s,int y_s,int x_o,int y_o,
                int x_n,int y_n)
{
    ++cb;
    br[cb].x_origin = x_or;   // origin of section
    br[cb].y_origin = y_or;
    br[cb].x_start = x_s;    // section which select
    br[cb].y_start = y_s;
    br[cb].x_old = x_o;      //  old coordinate
    br[cb].y_old = y_o;
    br[cb].x_new = x_n;      //  new coordinate
    br[cb].y_new = y_n;
 }

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function    delnode(n)
//           delete branched which search ready
// Input
//              n -> number of branched which search ready
// Output
//              cb -> count of branch must decrement
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
void delnode(int n)
{
   for(int r=n;r<cb;r++) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        br[r].x_origin =  br[r+1].x_origin;  // origin of section
        br[r].y_origin =  br[r+1].y_origin;
        br[r].x_start  =  br[r+1].x_start;   // section which select
        br[r].y_start  =  br[r+1].y_start;
        br[r].x_old    =  br[r+1].x_old;     //  old coordinate
        br[r].y_old    =  br[r+1].y_old;
        br[r].x_new    =  br[r+1].x_new;     //  new coordinate
        br[r].y_new    =  br[r+1].y_new;
     }
     --cb;
}

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function  check_head(n)
//'          check occured head(loop)
//           if new point of this branch(n) = old point of other branch
//           and delete branched which occure head(2 branch)
// Input
//           n -> number of branch which change to new point
// output
//      if occured head
//           head -> increment number of head
//           head_[..] -> save coordinate of head and status of head
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
void check_head(int &n)
{
   int  xn,yn,l,xo,yo;
   int  branch,b1=0,b2=0,r1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(int t=1;t<=cb;t++) {

 if(((br[t].x_old)==(br[n].x_new))&&((br[t].y_old)==(br[n].y_new))) {


  X1_8(br[n].x_origin,br[n].y_origin);

  sect1[b1].x=(br[n].x_origin);  sect1[b1].y=(br[n].y_origin);
  sect2[b2].x=(br[t].x_origin); sect2[b2].y=(br[t].y_origin);

  if(x1) { b1++;sect1[b1].x=(br[n].x_origin)+1;sect1[b1].y=(br[n].y_origin)-1;}
  if(x2) { b1++;sect1[b1].x=(br[n].x_origin)  ;sect1[b1].y=(br[n].y_origin)-1;}
  if(x3) { b1++;sect1[b1].x=(br[n].x_origin)-1;sect1[b1].y=(br[n].y_origin)-1;}
  if(x4) { b1++;sect1[b1].x=(br[n].x_origin)-1;sect1[b1].y=(br[n].y_origin);}
  if(x5) { b1++;sect1[b1].x=(br[n].x_origin)-1;sect1[b1].y=(br[n].y_origin)+1;}
  if(x6) { b1++;sect1[b1].x=(br[n].x_origin)  ;sect1[b1].y=(br[n].y_origin)+1;}
  if(x7) { b1++;sect1[b1].x=(br[n].x_origin)+1;sect1[b1].y=(br[n].y_origin)+1;}
  if(x8) { b1++;sect1[b1].x=(br[n].x_origin)+1;sect1[b1].y=(br[n].y_origin);}

  for(r1=0;r1<=b1;r1++)
          clear_b(sect1[r1].x,sect1[r1].y);

  if(((br[t].x_origin)!=(br[n].x_origin))||((br[t].y_origin)!=(br[n].y_origin))) {

     X1_8(br[t].x_origin,br[t].y_origin);

     if(x1) { b2++;sect2[b2].x=(br[t].x_origin)+1;sect2[b2].y=(br[t].y_origin)-1;}
     if(x2) { b2++;sect2[b2].x=(br[t].x_origin)  ;sect2[b2].y=(br[t].y_origin)-1;}
     if(x3) { b2++;sect2[b2].x=(br[t].x_origin)-1;sect2[b2].y=(br[t].y_origin)-1;}
     if(x4) { b2++;sect2[b2].x=(br[t].x_origin)-1;sect2[b2].y=(br[t].y_origin);}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(x5) { b2++;sect2[b2].x=(br[t].x_origin)-1;sect2[b2].y=(br[t].y_origin)+1;}
        if(x6) { b2++;sect2[b2].x=(br[t].x_origin)  ;sect2[b2].y=(br[t].y_origin)+1;}
        if(x7) { b2++;sect2[b2].x=(br[t].x_origin)+1;sect2[b2].y=(br[t].y_origin)+1;}
        if(x8) { b2++;sect2[b2].x=(br[t].x_origin)+1;sect2[b2].y=(br[t].y_origin);}

        for(r1=0;r1<=b2;r1++)
                clear_b(sect2[r1].x,sect2[r1].y);
 }

head++;   // increment count of head

del_head(br[n].x_start,br[n].y_start);   //   del head

for(r1=0;r1<=b1;r1++)
        set_b(sect1[r1].x,sect1[r1].y);

for(r1=0;r1<=b2;r1++)
        set_b(sect2[r1].x,sect2[r1].y);

clear_b(br[n].x_start,br[n].y_start);
clear_b(br[t].x_start,br[t].y_start);

if(((br[t].x_origin)!=(br[n].x_origin))||((br[t].y_origin)!=(br[n].y_origin)))  {

        if(thick1(br[t].x_origin,br[t].y_origin))
                clear_b(br[t].x_origin,br[t].y_origin);

        if(thick1(br[n].x_origin,br[n].y_origin))
                clear_b(br[n].x_origin,br[n].y_origin);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if((br[n].y_origin)<(br[t].y_origin))  {  // select maxy
                head_[head].x = br[n].x_origin;
                head_[head].y = br[n].y_origin;
            }
        else {    head_[head].x = br[t].x_origin;
                head_[head].y = br[t].y_origin;
            }
        }
    else {
        xo=xn=br[n].x_origin ; yo=yn=br[n].y_origin;  // mothode 2
        l = 0;
        do {
            clear_b(xo,yo);
            X1_8(xn,yn);
            set_b(xo,yo);
            xo = xn; yo = yn;
            branch = 0;
            l++;
            if(x1) { ++xn;--yn;++branch; }
            if(x2) { --yn;++branch; }
            if(x3) { --xn;--yn;++branch; }
            if(x4) { --xn;++branch; }
            if(x5) { --xn;++yn;++branch; }
            if(x6) { ++yn;++branch; }
            if(x7) { ++xn;++yn;++branch; }
            if(x8) { ++xn;++branch; }
            if(branch==1) { temp[l].x = xo;  temp[l].y = yo;}
          } while((branch==1)&&(l<4));    // 3 point for check

        if(branch>1) {     // clip distort length 1-3 (<4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          for(branch = (l-1) ; branch>0; branch--)
                clear_b(temp[branch].x,temp[branch].y);
          }

       else { xo=br[n].x_origin ; yo=br[n].y_origin; }

       if((x1&x2)||(x2&x3)) {clear_b(xo,yo);--yo;}
       else if((x3&x4)||(x4&x5)){clear_b(xo,yo);--xo;}
       else if((x5&x6)||(x6&x7)){clear_b(xo,yo);++yo;}
       else if((x7&x8)||(x8&x1)){clear_b(xo,yo);++xo;}

       head_[head].x = xo; head_[head].y = yo;

       if(thick1(head_[head].x,head_[head].y))
           clear_b(head_[head].x,head_[head].y);
     } // else
   if((br[n].x_new)>(head_[head].x))  head_[head].status = h_right;
   else  head_[head].status = h_left;

   if(n>t) { delnode(n); delnode(t); n--; goto end_chk; }
   else { delnode(t); delnode(n); goto end_chk; }
   } // if found
  } // loop while
  n++;
  end_chk:
}

//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function   line_direct(x,y)
//            find direct of line
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Input
//              x,y -> start point for find direct
// Output
//            x,y -> end point of direct
// Return code of direct
//              U  ->   up direct
//              D  ->   down direct
//            L  ->   left direct
//            R  ->   right direct
//            ND ->   not enough point for direct
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<


unsigned char line_direct(int &x,int &y)
{
  unsigned char u=false,l=false,r=false,d=false;
  int  up=0,down=0,left=0,right=0;
  int  c;
  int  c_p=0;

  do{
        clear_b(x,y);
// >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>.
//putpixel(x-x_min+X,y-y_min+Y,BLUE);
//delay(10);
// >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        X1_8(x,y);
        c = (int)(x1+x2+x3+x4+x5+x6+x7+x8);

        if(c==1) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(x1) { ++x;--y;}
else if(x2) { --y;}
else if(x3) { --x;--y;}
else if(x4) { --x;}
else if(x5) { --x;++y;}
else if(x6) { ++y;}
else if(x7) { ++x;++y;}
else if(x8) { ++x;}

c_p++;

if(begin_branch==1) {
        if(c_p==Start_P) {
                c_p = 0;
                begin_branch = 2;
         }
}
else {
  if((uldllr)==false) {
        if(x1|x2|x3) {
                up++; down = 0;
                if(up==State_Ch)  {
                        u=true;
                        up = left = right = 0;
                }
           }
        else if(x5|x6|x7) {
                down++; up = 0;
                if(down==State_Ch) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    d=true;
                    down = left = right = 0;
                }
            }
        if(x3|x4|x5) {
            left++; right = 0;
            if(left==State_Ch) {
                l=true;
                up = down = left = 0;
            }
        }

    else if(x1|x8|x7) {
            right++; left = 0;
            if(right==State_Ch) {
                r=true;
                up = down = right = 0;
            }
    }
}

else {
    if(x2) up++;
    else if(x6) down++;
    else if(x4) left++;
    else if(x8) right++;

    if(u==true) {
        if((x5|x6|x7)|(left==Start_P)|(right==Start_P))
            return(U);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                    else if(d==true) {
                        if((x1|x2|x3)|(left==Start_P)|(right==Start_P))
                                return(D);
                    }
                    else if(l==true) {
                        if((x1|x8|x7)|(up==Start_P)|(down==Start_P))
                                return(L);
                    }
                    else if(r==true) {
                        if((x3|x4|x5)|(up==Start_P)|(down==Start_P))
                                return(R);
                    }
                }
            }
        }
    }while(c==1);

    if(c_p>P_Direct) {
        if(u==true) return(U);
        else if(d==true) return(D);
        else if(l==true) return(L);
        else if(r==true) return(R);
        else return(ND);
    }
    else return(ND);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function   del_stroke(x,y)
//            delelte stroke of character(all point of character)
// Input
//            x,y -> any point of character
// Output
//            None
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
void  del_stroke(int x,int y)
{
 int c,j;
 int n = 0; //top stack;

 do{
        clear_b(x,y);
        X1_8(x,y);

        c = (int)(x1+x2+x3+x4+x5+x6+x7+x8);

        if(c==1) {

                if(x1) { ++x;--y;}
                else if(x2) { --y;}
                else if(x3) { --x;--y;}
                else if(x4) { --x;}
                else if(x5) { --x;++y;}
                else if(x6) { ++y;}
                else if(x7) { ++x;++y;}
                else if(x8) { ++x;}
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else if(c>1) {    //  section

            if(x1&x2&x3) --y;
            else if(x3&x4&x5) --x;
            else if(x5&x6&x7) ++y;
            else if(x7&x8&x1) ++x;

            clear_b(x,y);
            X1_8(x,y);

            if(x1) { n++;temp[n].x=x+1;temp[n].y=y-1;clear_b(temp[n].x,temp[n].y);}
            if(x2) { n++;temp[n].x=x  ;temp[n].y=y-1;clear_b(temp[n].x,temp[n].y);}
            if(x3) { n++;temp[n].x=x-1;temp[n].y=y-1;clear_b(temp[n].x,temp[n].y);}
            if(x4) { n++;temp[n].x=x-1;temp[n].y=y;clear_b(temp[n].x,temp[n].y);}
            if(x5) { n++;temp[n].x=x-1;temp[n].y=y+1;clear_b(temp[n].x,temp[n].y);}
            if(x6) { n++;temp[n].x=x  ;temp[n].y=y+1;clear_b(temp[n].x,temp[n].y);}
            if(x7) { n++;temp[n].x=x+1;temp[n].y=y+1;clear_b(temp[n].x,temp[n].y);}
            if(x8) { n++;temp[n].x=x+1;temp[n].y=y;clear_b(temp[n].x,temp[n].y);}
            x = temp[n].x; y = temp[n].y;  --n;

        }
      else if(c==0) {
                if(n>0) {
                    x = temp[n].x; y = temp[n].y;  --n;
                    c = 1;
                }
      }
    }while(c!=0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
// Function   del_head(x,y)
//            delete line at position x,y until end of line(head)
// Input
//            x,y -> start of line
// Output
//            None
//<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
void  del_head(int x,int y)
{
  head_[head].y_min = head_[head].y_max = y;

  do{
       X1_8(x,y);
       clear_b(x,y);
// >>>>>>>>>>>>>>>>>>>>>FOR DISPLAY HEAD ONLY<<<<<<<<<<<<<
       _hp++;
       hp[_hp].x = x;
       hp[_hp].y = y;
// >>>>>>>>>>>>>>>>>>>>>CAN DELETE>>>>>>>>>>>>>>>>>>>>>>>>
       if(x1) { ++x;--y;}
       else if(x2) { --y;}
       else if(x3) { --x;--y;}
       else if(x4) { --x;}
       else if(x5) { --x;++y;}
       else if(x6) { ++y;}
       else if(x7) { ++x;++y;}
       else if(x8) { ++x;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(y<head_[head].y_min)   head_[head].y_min = y;
        else if(y>head_[head].y_max)  head_[head].y_max = y;

    }while(X0(x,y));
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ รองศาสตราจารย์ ดร. ชม กิ้มปาน ที่ให้คำแนะนำ ให้ข้อมูล และเป็นที่ปรึกษาตลอดมาในการจัดทำปริญญานิพนธ์ฉบับนี้จนสำเร็จลุล่วงด้วยดี

ขอขอบพระคุณ คณาจารย์ เจ้าหน้าที่ภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความช่วยเหลือในด้านต่างๆ

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ที่ให้คำแนะนำและความช่วยเหลือในด้านต่างๆด้วยดี

สุดท้ายต้องขอขอบพระคุณ บิดา-มารดา ของข้าพเจ้าที่อบรมสั่งสอนข้าพเจ้าด้วยดีตลอดมา

นายนพพร คงดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. รศ. ดร. ชม กิ้มปาน "ทฤษฎีการรู้จำรูปแบบเบื้องต้น" , ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพ , พิมพ์ครั้งที่ 1 , กรกฎาคม 2525.

2. Kimpan,C.,A. Itoh and Prof. K. Kawanishi, "Recognition of Printed Thai Characters using a Matching Method ", IEE Proceeding , Vol 130 , Pt.E,No.6, Novenber 1983.

3. Kimpan,C., "Printed Thai Characters Recognition Using Topological Properties Method" ,Int.J.Electtronics 1986,Vol.60,No.3,pp.303-329

4. ประสาน ตังติสานนท์, การจดจำรูปแบบตัวอักษรคัดลายมือภาษาไทยโดยวิธีแยกลักษณะเด่น",วิทยานิพนธ์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2529

5. สุรพันธ์ เอื้อไพบูลย์ "การจดจำรูปแบบตัวอักษรคัดลายมือภาษาไทย โดยการพิจารณาหัวของตัวอักษร" ,วิทยานิพนธ์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2531

6.. ชาย เกษมอมรกุล "การออกแบบพจนานุกรมสำหรับการเรียนรู้อักษรลายมือไทย-อังกฤษ อัตโนมัติบนเครื่องไมโครคอมพิวเตอร์"วิทยานิพนธ์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2532

7. สุรสิทธิ์ ราตรี, "การรู้จำตัวอักษรตัวพิมพ์ภาษาไทย โดยวิธีค้นหาลักษณะโครงสร้างลายเส้น"วิทยานิพนธ์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2532

8. สุริยัน อนุวัตร, "การรู้จำอักษรลายมือเขียนตัวพิมพ์ใหญ่ภาษาอังกฤษ ด้วยการพิจารณาลักษณะโครงสร้าง", การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่16 , หน้า 275-279, 2536

9. ทรงชัย วีระทวีมาศ , "การปรับปรุงอัลกอริทึมในการทำให้วัตถุในภาพบางอย่างรวดเร็ว", การประชุมวิชาการทางวิศวกรรมไฟฟ้าครั้งที่ 18 , หน้า 956-960 , 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้