การออกแบบระบบการรู้จำตัวอักษรเขียนภาษาไทย

โดย

สุวิทย์ นาคพีระยุทธ

ลัญฉกร วุฒิสิทธิกุลกิจ

โครงการวิจัยเลขที่ 151-MRD-2540

ทุนส่งเสริมการวิจัยคณะวิศวกรรมศาสตร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จฬ
วศ 15
007326

สถาบันวิจัยและพัฒนาคณะวิศวกรรมศาสตร์

คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

กรุงเทพฯ

สิงหาคม 2541

สถาบันวิจัยและพัฒนาของคณะวิศวกรรมศาสตร์ไม่รับผิดชอบ
ต่อผลเสียใด ๆ อันอาจเกิดจากการนำความคิดเห็นในเอกสาร
ฉบับนี้ไปใช้ ความคิดเห็นที่ปรากฏในเอกสารเป็นความคิดเห็น
ของผู้เขียนซึ่งไม่จำเป็นต้องเป็นความคิดเห็นของสถาบัน ฯ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การออกแบบระบบการรู้จำตัวอักษรเขียนภาษาไทย

โดย

สุวิทย์ นาคพีระยุทธ

วุฒิ วิศวกรรมศาสตร์มหาบัณฑิต

และ

ลัญฉกร วุฒิสิทธิกุลกิจ

วุฒิ วิศวกรรมศาสตร์ดุษฎีบัณฑิต

โครงการวิจัยเลขที่ 151-MRD-2540

ทุนส่งเสริมการวิจัยคณะวิศวกรรมศาสตร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สถาบันวิจัยและพัฒนาคณะวิศวกรรมศาสตร์

คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

กรุงเทพฯ

สิงหาคม 2541

2 ธ.ค. 2541

I18356485

บทคัดย่อ

รายงานฉบับนี้เกี่ยวข้องกับการศึกษา ค้นคว้า ออกแบบและพัฒนาระบบการรู้จำสำหรับตัวอักษรเขียนภาษาไทย โดยประกอบด้วยเนื้อหาของ การสร้างฐานข้อมูลลายมือเขียนภาษาไทยและภาษาอังกฤษ จำนวน 5 หมื่นกว่าตัวอักษร การพัฒนาโปรแกรมสำหรับใช้ในการปรับปรุงคุณภาพและการประมวลผลเบื้องต้นของภาพตัวอักษรเพื่อให้อยู่ในรูปแบบที่เหมาะสมต่อการนำไปใช้ในการรู้จำ และการออกแบบระบบรู้จำตัวอักษรพยัญชนะภาษาไทยโดยอาศัยวิธีการบ่งถึงความแตกต่างของตัวอักษรภาษาไทยจากคุณลักษณะของ จำนวนหัว ตำแหน่งของหัว และจำนวนจุดของการเปลี่ยนจากจุดขาวไปเป็นจุดดำ ผลที่ได้คือแนวทางการออกแบบระบบรู้จำเบื้องต้นที่จะเป็นพื้นฐานต่อการพัฒนาระบบรู้จำที่มีประสิทธิภาพต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Abstract

This report presents a study of approaches to the design and development of handwritten Thai character recognition systems. The contents of this report include the creation of a handwritten Thai and English character database (over 50 thousand characters), the development of a computer software for image pre-processing to improve the image quality and convert them into an appropriate format and size suitable for character recognition, and the design of a Thai character recognition system using distinctive features of Thai characters based on the existence and location of the heads of characters and the stroke changing sequence technique. The outcome from this study is a basic design approach useful to the future development of efficient recognition systems.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

กิตติกรรมประกาศ

คณะผู้วิจัยขอขอบคุณ คุณยุทธนา บัวเสน คุณอำนาจ เจนพานิชทรัพย์ และคุณประเสริฐ ฉอเรืองวิวัฒน์ ในฐานะที่ทั้งสามท่านมีส่วนสำคัญในการทำงานและความสำเร็จของโครงการนี้ ขอขอบพระคุณหัวหน้าห้องปฏิบัติการวิจัย DSPRL รศ.ดร.สมชาย จิตะพันธ์กุล ที่ให้ความสนับสนุนด้านอุปกรณ์วิจัยและสถานที่รวมไปถึงด้านบุคคลากรและคำปรึกษาที่เป็นประโยชน์ต่อการวิจัยเป็นอย่างยิ่ง และคณะผู้วิจัยขอขอบคุณคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้ความสนับสนุนโครงการวิจัยนี้เป็นอย่างดี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบาญ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เลขหมู่ ๓๗
๐๙๑๕
เลขทะเบียน ๐๐๗๓๒๖
วัน,เดือน,ปี ๒๙ ก.ค. ๔๒

1. บทนำ

1.1 ความเป็นมาของปัญหา

การศึกษาวิจัยและพัฒนาของระบบการรู้จำตัวอักษรเขียนได้เริ่มขึ้นอย่างจริงจังเมื่อ ประมาณ 40-50 ปีก่อน ซึ่งเป็นช่วงเวลาเดียวกันกับการเกิดของเครื่องคอมพิวเตอร์ แต่กระนั้นการนำระบบดังกล่าวมาใช้ประโยชน์ในทางปฏิบัตินั้นเพิ่งจะเริ่มเด่นชัดเมื่อต้นทศวรรตที่ผ่านมา โดยในปัจจุบันเราสามารถโปรแกรมให้เครื่องคอมพิวเตอร์มีความสามารถรู้จำและแยกแยะตัวอักษรเขียนได้ในระดับที่น่าพอใจ หากแต่ว่างานวิจัยเหล่านี้ยังจำกัดอยู่เฉพาะบางภาษาเท่านั้น เช่น ภาษาอังกฤษ และภาษาจีน ส่วนงานวิจัยสำหรับภาษาไทยนั้นในขณะนี้ยังมีอยู่น้อยมาก และเนื่องจากการที่ภาษาแต่ละภาษามีคุณลักษณะที่แตกต่างกันอย่างชัดเจน เช่น ภาษาอังกฤษประกอบด้วยตัวอักษรพื้นฐานเพียง 26 ตัวซึ่งนำมาใช้ประกอบเป็นคำต่างๆได้มากมาย ขณะที่คำในภาษาจีนแต่ละคำมีการเขียนที่แตกต่างกันโดยสิ้นเชิง ส่วนภาษาไทยนั้นประกอบด้วยพยัญชนะ 44 ตัว สระและวรรณยุกต์อีก 26 ตัว โดยวิธีเขียนนั้นมีถึง 3 ระดับ ด้วยเหตุนี้การพัฒนาวิธีการรู้จำตัวอักษรเขียนของแต่ละภาษาจึงต่างกันมาก ดังนั้นการนำความรู้และงานวิจัยที่มีอยู่มาประยุกต์ใช้กับภาษาไทยนั้นจะเป็นไปอย่างจำกัด ด้วยเหตุนี้จึงมีความจำเป็นในการศึกษาและพัฒนาระบบการรู้จำตัวอักษรเขียนที่เหมาะสมกับภาษาไทย

1.2 การสำรวจการวิจัยอื่นๆที่เกี่ยวข้อง

ดังที่กล่าวไว้ก่อนแล้วว่างานวิจัยเรื่องการรู้ตัวอักษรภาษาไทยยังมีอยู่ไม่มากนัก ในรายงานนี้จึงมิได้เน้นในส่วนการสำรวจการวิจัยอื่นๆเท่าใดนัก หากแต่ความรู้และแนวความคิดต่างๆที่ได้มีผู้นำเสนอทางด้านระบบรู้จำกับภาษาอื่นๆนั้นคณะผู้วิจัยได้ทำการศึกษาอย่างละเอียดและมีบางส่วนที่ได้นำมาประยุกต์ใช้กับภาษาไทย ทั้งนี้คณะผู้วิจัยจะขอกล่าวถึงการวิจัยที่เกี่ยวข้องแทรกไปกับแต่ละส่วนของรายงานฉบับนี้ตามความเหมาะสม

1.3 วัตถุประสงค์และขอบเขต

ศึกษาค้นคว้าและวิจัยเพื่อหาแนวทางการออกแบบและพัฒนาระบบการรู้จำสำหรับตัวอักษรเขียนภาษาไทย โดยมีเป้าหมายหลักคือ การสร้างฐานข้อมูลตัวอย่างของตัวอักษรเขียนภาษาไทยขนาดใหญ่สำหรับการทดสอบสมรรถนะอัลกอริธึมที่ใช้ในการรู้จำ พัฒนาและเขียนโปรแกรมคอมพิวเตอร์ประมวลภาพอักษรประเภทต่าง ๆ ที่จำเป็นต่อการพัฒนาระบบรู้จำ และนำเสนอแนวทางใหม่ ๆ ที่ใช้ในการรู้จำตัวอักษรภาษาไทย

1.4 ประโยชน์ที่ได้จากการวิจัยนี้

- ระบบที่มีความสามารถรู้จำตัวอักษรเขียนภาษาไทยเบื้องต้น
- ฐานข้อมูลตัวอย่างของตัวอักษรเขียนภาษาไทยขนาดใหญ่ (กว่า 5 หมื่นตัวอักษร) ซึ่งได้ทำการอัดเก็บในแผ่นเก็บข้อมูลความจุสูง โดยฐานข้อมูลนี้จะมีประโยชน์ต่องานวิจัยในระยะยาวทั้งสำหรับคณะวิจัยเองและจะเผยแพร่สำหรับนักวิจัยท่านอื่นๆ
- เสริมสร้างความรู้และความเข้าใจในการออกแบบระบบการรู้จำสำหรับตัวอักษรเขียนภาษาไทยแก่คณะผู้วิจัย และเป็นประโยชน์ต่อการเรียนการสอนทั้งในระดับปริญญาบัณฑิตและมหาบัณฑิต
- เผยแพร่ความรู้และประสบการณ์จากงานวิจัยในงานประชุมทางวิชาการ IEEE Asia Pacific Conference on Circuits and Systems: Microelectronics and Integration Systems, November 24-27, 1998, Chiangmai Plaza Hotel, Chiangmai, Thailand.

2. การสร้างฐานข้อมูลตัวอักษรเขียนภาษาไทย

ในการออกแบบและพัฒนาระบบรู้จำตัวอักษรเขียน องค์ประกอบหนึ่งที่มีความสำคัญอย่างมากคือ ฐานข้อมูลลายมือเขียน เพราะเป็นส่วนที่นำมาใช้ในการทดสอบสมรรถนะของวิธีหรืออัลกอริธึมที่พัฒนาขึ้น สำหรับงานวิจัยที่กระทำกันอยู่ในปัจจุบันกับภาษาต่างประเทศนั้น สถาบันวิจัยแต่ละแห่งก็มักจะมีฐานข้อมูลลายมือเขียนเป็นของตนเอง ยกตัวอย่างเช่นในกรณีของภาษาอังกฤษ มีฐานข้อมูลที่แตกต่างกันอยู่หลายแหล่ง เช่น CEDAR และ University of Essex แต่ละแห่งเก็บรวบรวมข้อมูลจากแหล่งที่แตกต่างกัน เช่น เก็บจากลายมือที่จ่าหน้าซองจดหมาย ซึ่งลายมือของผู้เขียนจากแต่ละประเทศก็มักจะมีลักษณะที่แตกต่างกันถึงแม้จะเขียนตัวอักษรตัวเดียวกัน นอกจากตัวอย่างลายมือเขียนจะเก็บจากแหล่งที่แตกต่างกันแล้ว รูปแบบในการเก็บก็ยังแตกต่างกันอีกด้วย ฐานข้อมูลบางแหล่งจะเก็บในรูปแบบของ bit map บางแหล่งจะเก็บเป็น grey scale จากการที่ฐานข้อมูลที่มีอยู่ในปัจจุบันมีความหลากหลายมาก ไม่มีฐานข้อมูลใดที่ได้รับการยอมรับให้เป็นมาตรฐานกลาง ส่งผลทำให้การเปรียบเทียบสมรรถนะของวิธีการรู้จำแบบต่างๆกระทำกันอย่างไม่มีระบบหรือไม่มีกฎเกณฑ์ตายตัวที่เหมาะสม ด้วยเหตุนี้การหาข้อสรุปที่ถูกต้องจึงเป็นไปอย่างยากลำบาก

สำหรับงานวิจัยกับตัวอักษรเขียนภาษาไทยในปัจจุบันยังมีอยู่อย่างจำกัด และยังไม่มีรายงานการสร้างฐานข้อมูลขนาดใหญ่ที่เป็นมาตรฐานสำหรับใช้ในการทดสอบ ดังนั้นโครงการวิจัยนี้จึงตระหนักถึงปัญหาและความจำเป็นที่จะต้องมีฐานข้อมูลลายมือเขียนภาษาไทยและได้ให้ความสำคัญมากเป็นพิเศษ งานส่วนนี้จึงเป็นส่วนที่จะต้องกระทำก่อนหรือกระทำควบคู่ไปกับการออกแบบและพัฒนาส่วนอื่นๆ เพื่อให้ฐานข้อมูลที่สร้างขึ้นมีคุณภาพที่ได้มาตรฐานและมีขนาดใหญ่เพียงพอ จึงได้มีการเตรียมการและวางแผนอย่างชัดเจนในรายละเอียดก่อนที่จะออกทำการเก็บข้อมูลตัวอย่างจริงจากกลุ่มคน ซึ่งที่ผ่านมาจะอาศัยนิสิตภายในมหาวิทยาลัย ในช่วงของการวางแผนได้กำหนดรายละเอียดต่างๆเป็น 5 หัวข้อหลักดังนี้

1. ชนิดของตัวอักษรที่เก็บ
2. แบบฟอร์มที่ใช้ในการเก็บตัวอย่างข้อมูล
3. รูปแบบการจัดเก็บ
4. โปรแกรมจัดการฐานข้อมูล
5. โปรแกรมสำหรับเปิดดูฐานข้อมูล

ในการทำแต่ละหัวข้อนั้นมีปัจจัยที่จะต้องพิจารณาต่าง ๆ กัน เช่นในส่วนของชนิดของตัวอักษรที่เก็บจะต้องเก็บให้ครบทุกประเภทของตัวอักษรในครั้งเดียว เพราะเป็นเรื่องที่ไม่ง่ายนักที่จะเก็บเพิ่มเติมเพียงบางส่วนภายหลัง ในส่วนของแบบฟอร์มที่ใช้ก็ต้องออกแบบให้ใกล้เคียงกับสภาพการใช้งานจริงเช่นขนาดของตัวอักษรต้องไม่ใหญ่หรือเล็กเกินไป เมื่อทั้ง 2 ขั้นตอนได้เสร็จสิ้นแล้วก็ทำการเก็บตัวอย่างข้อมูลจริง ขั้นตอนการเก็บนั้นเป็นช่วงที่ต้องใช้เวลามากเพื่อให้ได้จำนวนภาพอักษรจำนวนมากพอ จากนั้นก็นำตัวอย่างภาพทั้งหมดไปสแกนลงบนคอมพิวเตอร์ ในรูปแบบที่เหมาะสมต่อไป

หลังจากที่ได้สแกนภาพทั้งหมดลงในฮาร์ดดิสก์แล้ว จึงนำภาพอักษรเหล่านั้นไปผ่านกระบวนการประมวลภาพให้อยู่ในรูปแบบที่เหมาะสมกับการใช้ในการทดสอบระบบรู้จำต่อไป ในขั้นตอนนี้ต้องอาศัยโปรแกรมคอมพิวเตอร์จัดการเพื่อให้การประมวลต่างๆเป็นไปอย่างรวดเร็วและอัตโนมัติ ส่วนสุดท้ายของการสร้างฐานข้อมูลก็คือการตรวจสอบความถูกต้องของฐานข้อมูล ในส่วนนี้อาศัยโปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้น

โดยเฉพาะสำหรับเปิดดูภาพอักษรต่างๆในฐานข้อมูลในโหมดกราฟิก ทั้งนี้รายละเอียดของแต่ละส่วนจะได้กล่าวถึงต่อไป

2.1 ชนิดของตัวอักษรที่เก็บ

สำหรับฐานข้อมูลภาพตัวอักษรที่สร้างขึ้นประกอบด้วย พยัญชนะไทย 46 ตัว (รวม ฤ ฦ) สระไทย 16 ตัว วรรณยุกต์ไทย 4 ตัว สัญลักษณ์ภาษาไทยอื่น ๆ 4 ตัว และตัวเลขไทย 10 ตัว นอกจากนี้เพื่อให้ฐานข้อมูลมีความสมบูรณ์มากขึ้นจึงได้ทำการเก็บตัวอักษรภาษาอังกฤษรวมเข้ามาด้วย ในส่วนของภาษาอังกฤษ ประกอบด้วยตัวอักษรตัวใหญ่และตัวเล็กอย่างละ 26 ตัว และตัวเลขอารบิค 10 ตัว เหตุที่ได้เก็บอักษรภาพภาษาอังกฤษเข้ามาด้วยก็เพื่อให้เป็นประโยชน์ต่อการทำงานวิจัยในอนาคต รวมแล้วจะมีชนิดตัวอักษรทั้งหมด 142 แบบ ดังรายละเอียดที่แสดงในรูปที่ 2.1

พยัญชนะไทย 44 ตัว
ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ

สระไทย 18 ตัว
ะ า ิ ี ึ ื ุ ู เ แ ไ ใ โ ็ ั ฤ ๅ

วรรณยุกต์ไทย 4 ตัว
่ ้ ๊ ๋

สัญลักษณ์ภาษาไทยอื่นๆ 4 ตัว
์ ๎ ฯ ๆ

ตัวเลขไทย 10 ตัว
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙

อักษรภาษาอังกฤษ ตัวใหญ่ 26 ตัว
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

อักษรภาษาอังกฤษ ตัวเล็ก 26 ตัว
a b c d e f g h i j k l m n o p q r s t u v w x y z

ตัวเลขอารบิค
0 1 2 3 4 5 6 7 8 9

รูปที่ 2.1 *รายละเอียดของตัวเลขและตัวอักษรที่ได้ทำการเก็บ*

2.2 แบบฟอร์มที่ใช้ในการเก็บตัวอย่างข้อมูล

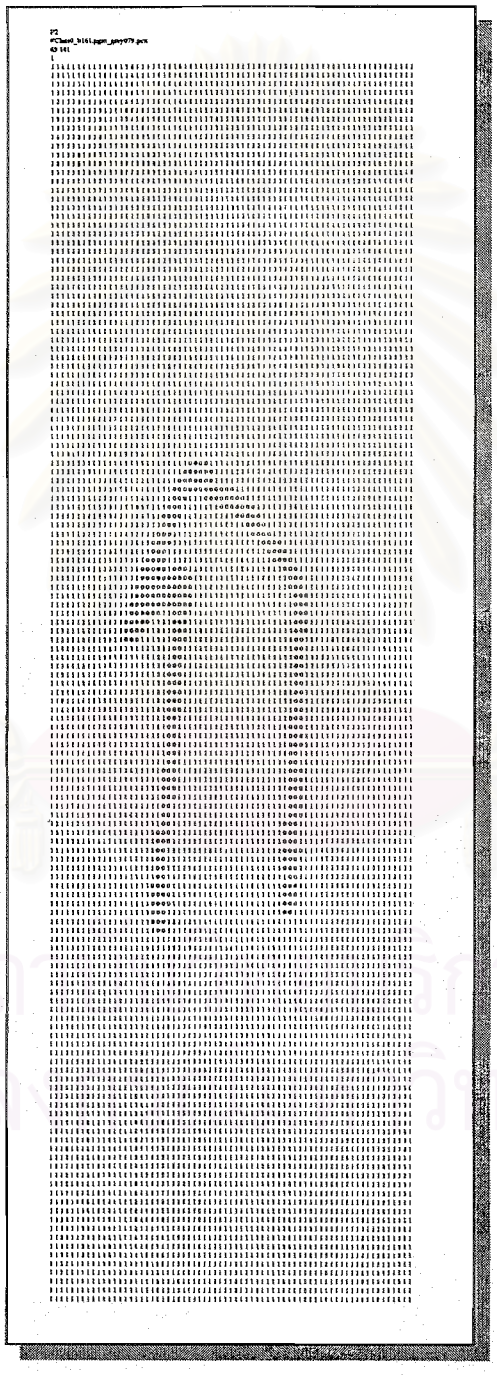
ในการเก็บข้อมูลนั้นเราได้ออกแบบฟอร์มเฉพาะสำหรับการเก็บข้อมูลซึ่งมีรูปแบบดังที่แสดงในรูปที่ 2.2 สังเกตว่าแต่ละตัวอักษรหรือตัวเลขจะมีช่องให้เขียนถึง 2 ช่อง (ฟอร์ม A และฟอร์ม B) ทั้งนี้มีจุดมุ่งหมายเพื่อเก็บข้อมูลจากการเขียนด้วยปากกา 2 ชนิดคือ ปากกาลูกลื่นสีน้ำเงิน และปากกาเมจิกสีดำ โดยทางกลุ่มวิจัยได้จัดเตรียมอุปกรณ์ต่างๆเหล่านี้ไว้ให้ เพื่อที่จะได้ตัวอย่างข้อมูลที่ใกล้เคียงกับสภาพการใช้งานจริง และอยู่ภายใต้เงื่อนไขที่กำหนด

รูปที่ 2.2 แบบฟอร์มที่ใช้ในการเก็บข้อมูลลายมือเขียน ขนาดเล็กกว่าฟอร์มจริงเล็กน้อย

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.3 รูปแบบการจัดเก็บ

แบบฟอร์มที่ออกแบบขึ้นในหัวข้อที่ 2.2 ได้ถูกนำไปใช้ในการเก็บตัวอย่างภาพตัวอักษร โดยได้ทำการสุ่มเก็บตัวอย่างลายมือเขียนจากนิสิตภายในมหาวิทยาลัยจำนวนทั้งสิ้น 200 คน จากนั้นนำข้อมูลที่เก็บได้ในแบบฟอร์มทั้ง 200 ชุดไปสแกนลงบนแผ่นฮาร์ดดิสก์ ข้อมูลเหล่านี้ได้เก็บอยู่ในรูปแบบข้อมูล grey scale 256 ระดับ ในไฟล์นามสกุล .PCX ซึ่งเป็นมาตรฐานสากลที่มีการใช้งานกันอยู่ทั่วไป ผลที่ได้ก็คือจำนวนไฟล์ข้อมูลทั้งสิ้น 200 ไฟล์



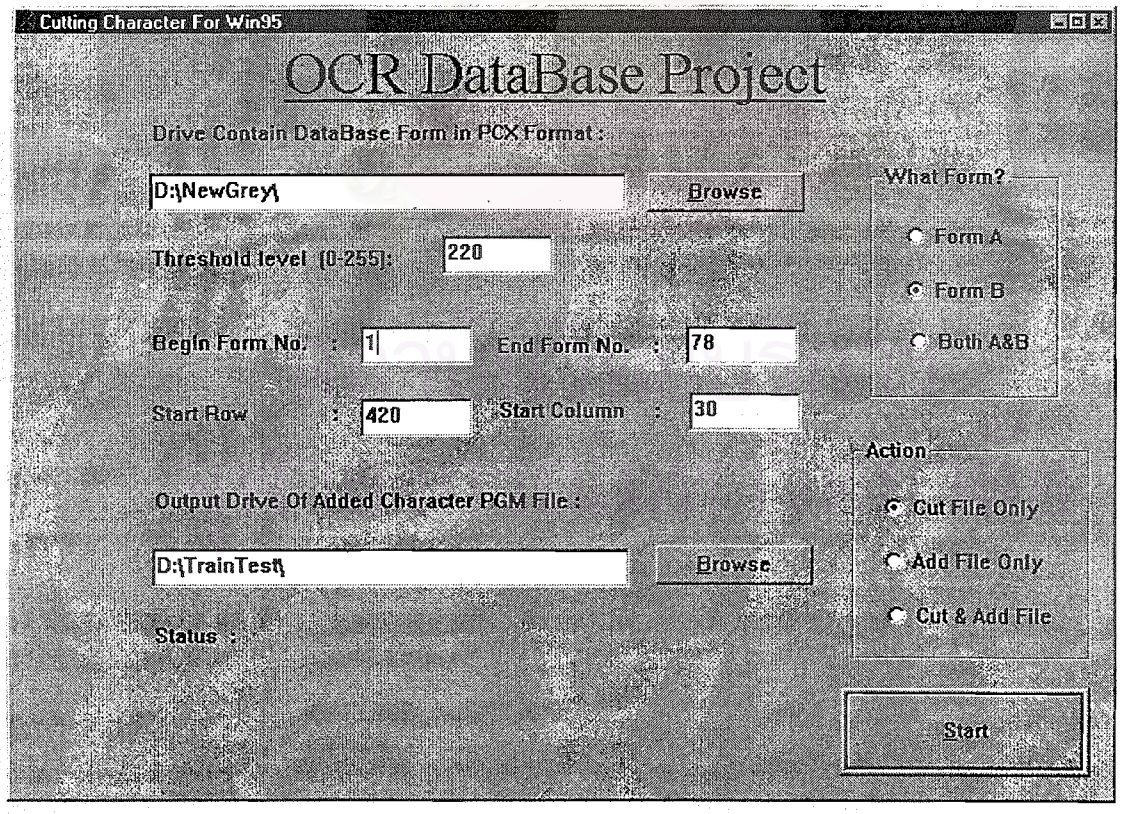
รูปที่ 2.3 ตัวอย่างของไฟล์ข้อมูลอักษร ถ

เนื่องจากการเก็บข้อมูลในลักษณะนี้ไม่สะดวกต่อการนำไปใช้ในการทดสอบสมรรถนะของระบบรู้จำได้โดยตรง จึงมีความจำเป็นที่จะต้องแปลงลักษณะการจัดเก็บข้อมูลให้อยู่ในรูปแบบที่เหมาะสมต่อการนำไปใช้งาน โดยปกติแล้วรูปแบบการจัดเก็บข้อมูลที่ต้องการก็คือไฟล์ข้อมูลที่ประกอบด้วยตัวอย่างภาพตัวอักษรชนิดเดียวกันจากตัวอย่างทั้งหมด เช่น อักษร ก จากลายมือเขียนของทุกคนในไฟล์เดียว ซึ่งหมายความว่าจะมีไฟล์หนึ่งไฟล์สำหรับตัวอักษรแต่ละชนิด และในแต่ละไฟล์จะมีภาพตัวอักษรทั้งสิ้น 200 ตัว

เพื่อให้ได้ไฟล์ข้อมูลในรูปแบบดังกล่าวจึงได้พัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถจัดการแยกตัวอักษรแต่ละตัวออกจากภาพที่ได้จากการสแกน แล้วนำตัวอักษรแบบเดียวกันจากตัวอย่างทุกชุดไปเก็บรวมกันในไฟล์เดียว นอกจากนี้โปรแกรมดังกล่าวยังทำหน้าที่อีกอย่างหนึ่งพร้อมๆกันไปคือ ทำการแปลงภาพ grey scale 256 ระดับให้กลายเป็นภาพขาวดำ 2 ระดับ ทั้งนี้เพราะระบบรู้จำที่พัฒนาขึ้นจะรู้จำกับภาพอักษรขาวดำ 2 ระดับเท่านั้น สำหรับไฟล์รูปแบบใหม่ได้จัดเก็บให้อยู่ในไฟล์ที่มีนามสกุล .pgm ซึ่งมีลักษณะดังแสดงในรูปที่ 2.3

2.4 โปรแกรมสำหรับจัดการฐานข้อมูล

ดังที่ได้กล่าวไว้แล้วว่าข้อมูลที่สแกนได้จะต้องนำมาแปลงให้อยู่ในรูปแบบที่ต้องการคือ .pgm ดังนั้นจึงได้ทำการพัฒนาโปรแกรมคอมพิวเตอร์ขึ้นเพื่อให้งานส่วนนี้สามารถกระทำได้อย่างรวดเร็วและอัตโนมัติ ทั้งนี้ได้พัฒนาโปรแกรมดังกล่าวขึ้นเป็นกราฟฟิกโหมดทั้งหมด เพื่อให้ผู้ใช้สามารถติดต่อและควบคุมโปรแกรมได้โดยง่าย ดูรูปที่ 2.4 และ 2.5 ประกอบ ในแต่ละส่วนของจอภาพจะมีหน้าที่และการใช้งานดังต่อไปนี้ จากรูปที่ 2.4 ก่อนจะทำการกด ปุ่ม Start เพื่อให้โปรแกรมทำการประมวลภาพและแปลงให้อยู่ในรูปแบบ .pgm นั้นต้องทำการตั้งค่าต่าง ๆ ให้ เรียบร้อยเสียก่อน ซึ่งค่าดังกล่าวมีรายละเอียดในรูปที่ 2.5



รูปที่ 2.4 *หน้าจอของโปรแกรมที่ใช้ทำฐานข้อมูล*

รายละเอียดของค่าพารามิเตอร์ที่ต้องตั้งค่า
Drive Contain Database Form in PCX Format จะเป็น drive ที่มีไฟล์ .PCX ซึ่งสแกนมาจากแบบฟอร์มที่ออกแบบไว้โดยเฉพาะ และไฟล์ที่จะเริ่มต้นตัด จะต้องมีชื่อขึ้นต้นด้วย Grey และ ตามด้วยตัวเลข 3 หลัก เช่น Grey001
Threshold Level (0- 255) เป็นค่าที่จะเปลี่ยนจาก Grayscale ไปสู่ไฟล์ภาพขาวดำ ซึ่งถ้าตั้งค่านี้มากไปจะมีสัญญาณรบกวนเยอะมากตามไปด้วย แต่ถ้าตั้งน้อยเกินไป ข้อมูลของภาพบางส่วนจะหายไป ในการเลือกค่าที่เหมาะสมเพื่อความรวดเร็วจะต้องทำการลองผิดลองถูกโดยอาศัยโปรแกรมอื่นมาก่อน
Begin Form No. ไฟล์ที่จะเริ่มทำการตัดหรือเพิ่ม ใส่ค่าเป็นเลขที่ของแบบฟอร์ม เช่น 1
End Form No. ไฟล์ที่จะสิ้นสุดการตัดหรือเพิ่ม ใส่ค่าเป็นเลขที่ของแบบฟอร์ม เช่น 78
Start Row. เป็นค่าของแถวที่จะเริ่มต้นสำหรับใช้ในการค้นหาตำแหน่งของกรอบภาพ
Start Column เป็นค่าคอลัมน์ที่จะเริ่มต้นสำหรับใช้ในการค้นหาตำแหน่งของกรอบภาพ
Output Drive of Added Character PGM File จะต้องต่อเป็นชุดใหญ่เท่านั้นคือชุด A หรือ B (ปากกาเมจิก และ ปากกาลูกลื่น) ซึ่งแบ่งเป็น 2 แบบดังนี้ • Drive ที่มีไฟล์ .PGM อยู่แล้ว โปรแกรมจะอ่านจาก Zip File ที่ มีตัวอักษรที่ผ่านขบวนการตัดแล้วนำไปต่อท้ายไฟล์ที่อยู่ใน drive นั้น • Drive นั้นไม่มีไฟล์ .PGM อยู่โปรแกรมจะสร้างไฟล์ขึ้นมาใหม่
What Form? มีให้เลือก 3 แบบ คือ จะตัดหรือเพิ่มข้อมูล • เฉพาะแบบฟอร์ม B • เฉพาะแบบฟอร์ม B • ทั้งแบบฟอร์ม A และ B
Action มีให้เลือก 3 แบบ คือ • ตัดไฟล์อย่างเดียว ผลที่ได้จะเป็น Zip File อยู่ใน directory ที่โปรแกรมอยู่ เรียงชื่อตามแบบฟอร์มที่ตัด เช่น Grey001.zip Grey002.zip เป็นต้น โดยชื่อไฟล์ของตัวอักษรแต่ละชนิดได้กำหนดไว้ตายตัว • เพิ่มไฟล์อย่างเดียว จะต้องมี Zip File ที่ได้จากการตัดมาแล้วอยู่ใน directory ที่โปรแกรมนี้อยู่ • ทั้งตัดและเพิ่มไฟล์

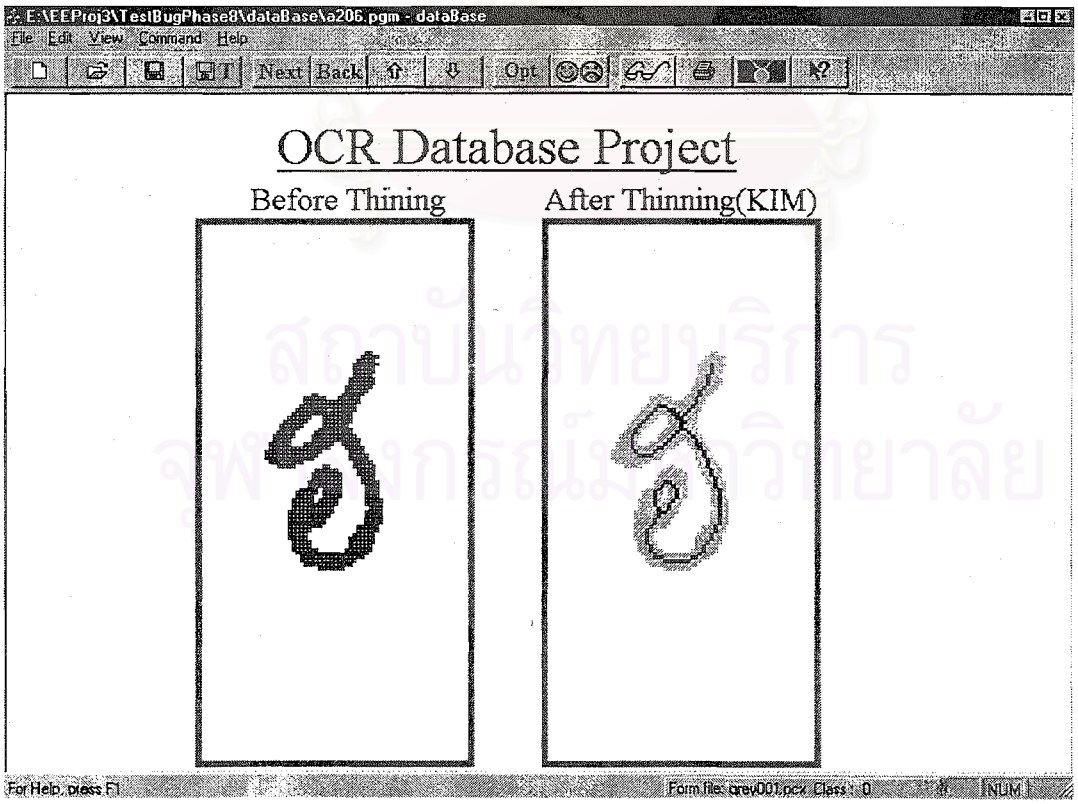
รูปที่ 2.5 รายละเอียดของพารามิเตอร์ต่างๆที่ต้องมีการตั้งค่า

2.5 โปรแกรมสำหรับเปิดดูฐานข้อมูลในโหมดกราฟฟิก

ในระหว่างที่กำลังสร้างฐานข้อมูลภาพอักษรอยู่นั้น ปัญหาหนึ่งที่ต้องคำนึงถึงก็คือความถูกต้องของการเก็บข้อมูลภาพ ดังนั้นเพื่อตรวจสอบว่าภาพที่ได้เป็นไปตามที่ต้องการหรือไม่ จึงได้ทำการพัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถแสดงภาพอักษรบนจอคอมพิวเตอร์ในโหมดกราฟฟิก โปรแกรมดังกล่าวมีขีดความสามารถในการเปิดไฟล์ .pgm และแสดงภาพที่อยู่ในไฟล์ได้ทุกตัว โดยสามารถค้นหาภาพที่ต้องการได้ทั้งแบบเป็นลำดับและข้ามไปยังตำแหน่งของภาพที่ต้องการได้ทันที นอกจากนี้โปรแกรมนี้ยังฟังก์ชันต่างๆ เช่น สามารถ

เลือกบันทึกเฉพาะภาพอักษรที่สนใจลงในไฟล์ใหม่ สามารถนำกรรมวิธีการประมวลภาพ เช่น การทำภาพตัวอักษรให้บาง มาทดสอบกับภาพในฐานข้อมูลได้เพื่อใช้ในการเปรียบเทียบการภาพเดิม อีกทั้งโปรแกรมนี้ยังอนุญาตให้แสดงผลออกทางเครื่องพิมพ์ได้ รูปที่ 2.6 แสดงหน้าจอของโปรแกรมที่ใช้ดึงภาพจากฐานข้อมูลออกมาดูและทดสอบกับกรรมวิธีการประมวลภาพบางอย่าง ซึ่งในรูปแสดงการทดสอบ การทำให้ตัวอักษรบาง สำหรับหน้าที่ของฟังก์ชันต่างๆทั้งหมดที่มีรวมถึงรายละเอียดคำอธิบายสามารถดูได้จากในรูปที่ 2.7 สังเกตว่าที่บรรทัดล่างสุดของจอจะมีแถบแสดงสถานะของไฟล์ข้อมูลที่กำลังดูอยู่ เช่น จะบอกว่าไฟล์ที่แสดงผลนั้นมาจากข้อมูลดิบหมายเลขอะไร และ คลาสอะไร จากในรูปตัวอย่าง จะทราบได้ทันทีว่าตัวอักษรนี้มาจากไฟล์ grey001.pcx หรือข้อมูลดิบหมายเลข 001 และมีค่าของคลาสเป็น 0 หมายเหตุ คำว่าคลาสในที่นี้ได้กำหนดให้เป็นค่าที่บ่งถึงความยากง่ายในการอ่านภาพตัวอักษรนั้นๆ ซึ่งในขณะนี้ยังมิได้กำหนดชัดเจนตายตัวลงไปว่าจะมีกี่คลาสและแต่ละคลาสควรจะมีคุณสมบัติอย่างไร แต่กระนั้นในเบื้องต้นคณะผู้วิจัยได้มีการกำหนดคร่าวๆไว้ว่าคลาส 0 หมายถึงตัวอักษรที่อ่านได้ง่ายและมีองค์ประกอบต่างๆที่สำคัญของตัวอักษรครบถ้วน เช่นมีหัวตัวอักษรที่ชัดเจนไม่กำกวม คลาส 1 หมายถึงตัวอักษรที่พอจะอ่านได้ไม่ยากนักโดยที่องค์ประกอบบางอย่างของภาพอาจจะไม่สมบูรณ์ เช่น หัวตัวอักษรไม่เป็นวงกลมที่ชัดเจนแต่พอจะเดาได้ว่าเป็นอักษรตัวใด คลาสที่ 2 หมายถึงตัวอักษรที่อ่านได้ยากเนื่องจากลายมือที่เขียนมีความหวัดมาก

โปรแกรมที่พัฒนาขึ้นนี้นอกจากจะอนุญาตให้เลือกใช้ฟังก์ชันทั้งหมดนี้ได้จากใน menu bar แล้ว ยังได้สร้างปุ่มอำนวยความสะดวกต่างๆมากมายลงใน tool bar อีกด้วย ดูรายละเอียดการใช้งานของ tool bar ได้ในรูปที่ 2.8 ส่วนการแสดงผลออกทางเครื่องพิมพ์ก็ได้กำหนดรูปแบบของพิมพ์ที่เป็นระเบียบ โดยจะมีข้อมูลอื่นๆที่เกี่ยวกับภาพอักษร ได้แก่ ชื่อของไฟล์ที่กำลังเปิดใช้งานและ Directory ค่าของคลาส บอกที่มาของตัวอักษรเพื่อสะดวกในการอ้างอิงกับไปสู่ข้อมูลดิบ วันที่ที่ใช้งาน และอื่นๆ ดูตัวอย่างได้ในรูปที่ 2.9

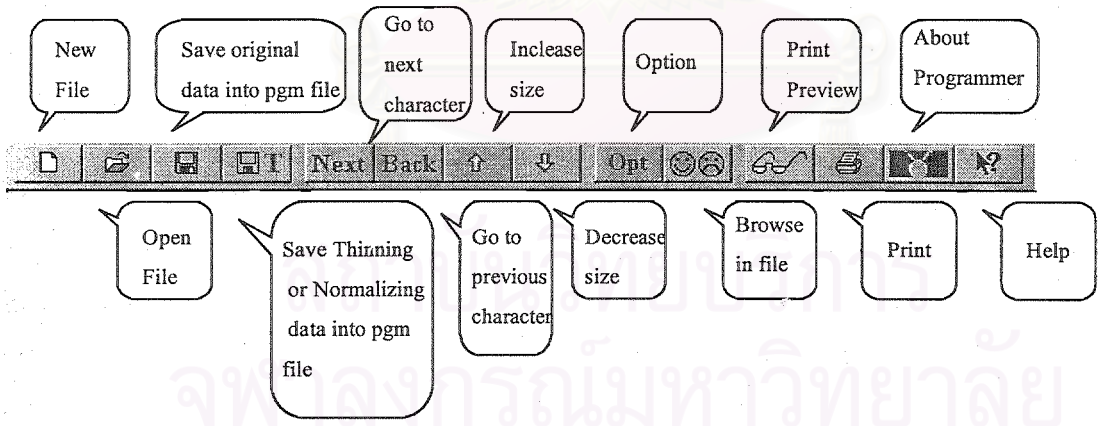


รูปที่ 2.6 หน้าจอของโปรแกรมที่ใช้เปิดดูฐานข้อมูล

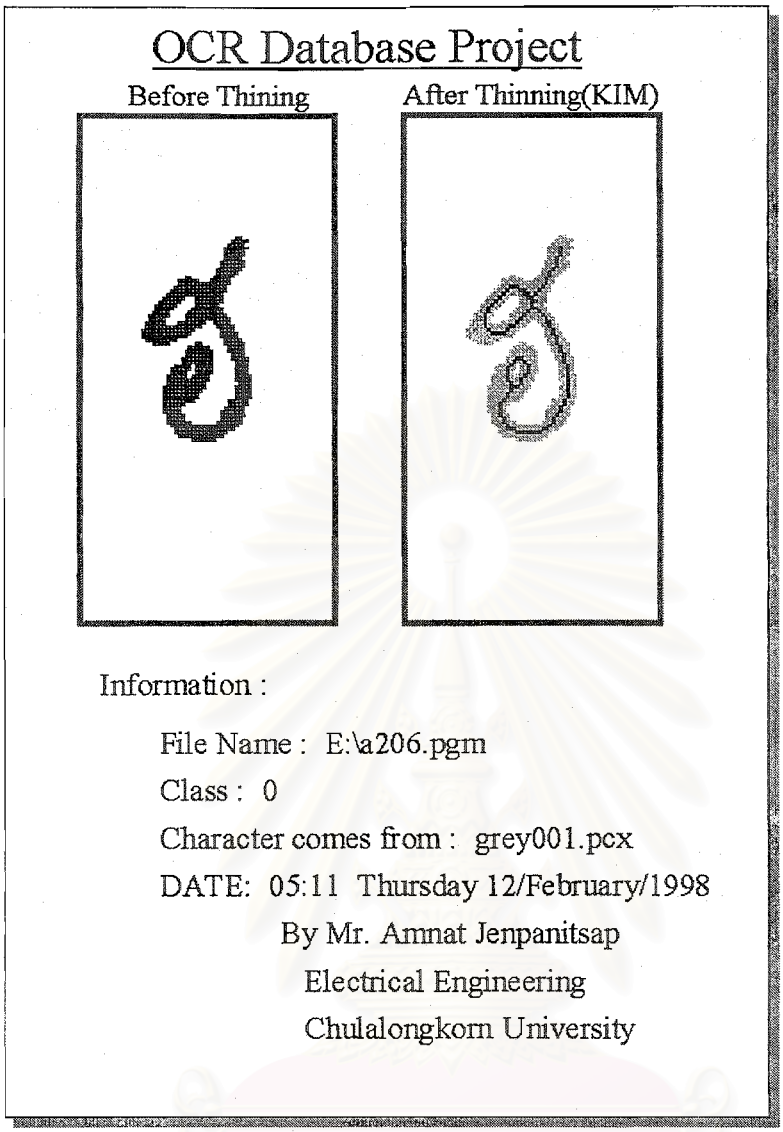
ฟังก์ชันต่างๆ

- **New** เป็นการเริ่มใหม่ จะต้องใช้ฟังก์ชันนี้ก่อนทุกครั้งถ้าจะเปิดไฟล์ใหม่
- **Open** สำหรับการเปิดไฟล์ ที่เป็น นามสกุล .pgm
- **Save** บันทึกไฟล์ดั้งเดิม(Original data) ที่แสดงผลบนจอภาพโดยเป็นไฟล์ .pgm
- **Save Thin** บันทึกไฟล์หลังการผ่านอัลกอริธึมโดยเป็นไฟล์ .pgm
- **Next** ไปที่ตัวอักษรที่อยู่ถัดไปในไฟล์เดียวกัน
- **Back** กลับไปยังตัวอักษรที่อยู่ก่อนหน้าในไฟล์เดียวกัน
- **Option** ระบุโหมดในการแสดงผล
- **Scale Up** ขยายขนาดของการแสดงผล
- **Scale Down** ลดขนาดของการแสดงผล
- **Browse** แสดงตัวอักษรที่เรียงกันในไฟล์นั้นและสามารถกระโดดไปยังตัวอักษรใดๆได้
- **Print Preview** แสดงภาพก่อนการพิมพ์
- **Print** แสดงผลออกทางเครื่องพิมพ์ ดูรูปที่ 2.9 ประกอบ
- **Help** บอกวิธีใช้งาน
- **About** บอกชื่อผู้พัฒนาโปรแกรม

รูปที่ 2.7 รายละเอียดของฟังก์ชันต่างๆ



รูปที่ 2.8 แสดงฟังก์ชันที่ใช้บน Toolbar



รูปที่ 2.9 แสดงภาพที่พิมพ์ออกมาจากเครื่องพิมพ์

2.6 ผลที่ได้จากการสร้างฐานข้อมูล

จากกระบวนการต่างๆที่ใช้ในการสร้างฐานข้อมูลที่กล่าวมาทั้งหมดนี้ ผลที่ได้ก็คือฐานข้อมูลที่มีขนาดใหญ่เพียงพอในการนำมาทดสอบสมรรถนะของระบบรู้จำตัวอักษรเขียนภาษาไทยและภาษาอังกฤษได้อย่างมีประสิทธิภาพ โดยมีจำนวนทั้งสิ้น 5 หมื่นกว่าตัวอักษร ทั้งนี้ได้เก็บรวบรวมไว้เป็นไฟล์ 2 รูปแบบ คือ grey scale format และ binary format และกระทำการอัดเก็บในแผ่น CD-ROM พร้อมที่จะนำมาใช้งานได้ทันที

นอกจากนี้โปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นทั้งสองส่วนสามารถนำมาใช้ช่วยในการเก็บข้อมูลเพิ่มเติมได้อย่างสะดวกรวดเร็วและมีประสิทธิภาพ ส่วนที่ผู้วิจัยจะต้องทำเพิ่มก็เพียงแต่ทำการเก็บรวบรวมลายมือเขียนจากแหล่งต่างๆเพิ่ม โดยใช้แบบฟอร์มที่ได้ออกแบบไว้ จากนั้นก็ทำการสแกนภาพตัวอย่างทั้งหมดลงในคอมพิวเตอร์ ส่วนที่เหลือจะสามารถกระทำได้โดยอาศัยโปรแกรมคอมพิวเตอร์ ซึ่งข้อมูลใหม่เหล่านี้จะถูกผนวกเข้าไปกับฐานข้อมูลเดิมที่มีอยู่แล้วได้โดยอัตโนมัติ

3. การปรับปรุงคุณภาพและประมวลผลเบื้องต้น (Pre-Processing) ของภาพตัวอักษร

หลังจากที่ได้เก็บตัวอย่างข้อมูลภาพจำนวนมากเรียบร้อยแล้ว ก่อนที่จะนำภาพเหล่านี้มาใช้กับระบบรู้จำ โดยปกติแล้วจะต้องนำภาพอักษรเหล่านี้มาผ่านกระบวนการประมวลผลเบื้องต้นเพื่อปรับปรุงคุณภาพของภาพให้ดีขึ้นหรือเพื่อให้อยู่ในรูปแบบที่เหมาะสมและใช้งานได้กับระบบรู้จำ ในส่วนนี้มีความสำคัญมากเพราะอาจส่งผลกระทบโดยตรงต่อประสิทธิภาพการรู้จำของแต่ละระบบ ดังนั้นจึงมีความจำเป็นที่จะต้องพัฒนาโปรแกรมคอมพิวเตอร์สำหรับงานในส่วนนี้ ในโครงการนี้ได้แบ่งการประมวลผลเบื้องต้นออกเป็น 5 ประเภท คือ

- การกำจัดสัญญาณรบกวน
- การทำภาพตัวอักษรให้บาง
- การปรับขนาดของภาพ
- การหมุนภาพ
- การทำให้ภาพโย้

โดยรายละเอียดของแต่ละหัวข้อจะกล่าวต่อไปข้างล่าง

3.1 การกำจัดสัญญาณรบกวน (Noise Reduction)

โดยปกติแล้วภาพตัวอักษรที่ได้จากการสแกนจากแบบฟอร์มนั้นจะไม่สมบูรณ์ 100% มักจะมีจุดดำบางจุดที่ขาดหายไปหรือมีจุดดำที่ไม่ต้องการเกินขึ้นมา สิ่งเหล่านี้อาจเกิดขึ้นจากกระดาษเก็บข้อมูล เครื่องสแกนเนอร์ หรือผู้กรอกลายมือเอง ดังนั้นจึงมีความจำเป็นที่จะต้องทำการลบส่วนของสัญญาณรบกวนเหล่านั้นออกไป โดยที่จะต้องไม่ทำให้ข้อมูลหลักของภาพตัวอักษรมีการเปลี่ยนแปลงรูปร่างหรือผิดเพี้ยนไปจากเดิม สำหรับในงานวิจัยนี้เราได้ทำการเขียนโปรแกรมคอมพิวเตอร์ที่สามารถกำจัดสัญญาณรบกวนที่มีลักษณะเป็นจุดเดี่ยวหรือรูเดี่ยวได้ ในการกำจัดสัญญาณรบกวนเหล่านี้สามารถกระทำได้โดยใช้หน้าต่างขนาด 3x3 ดังที่แสดงในรูปที่ 3.1 และ 3.2 มาทาบลงบนภาพอักษรโดยเริ่มที่ตำแหน่งหัวมุมบนด้านซ้าย และทำการเลื่อนหน้าต่างดังกล่าวไปทีละจุดตามแนวนอนจนสุด แล้วเลื่อนหน้าต่างดังกล่าวลงมาทีละหนึ่งแถวแล้วทำเหมือนเดิมจนถึงท้ายสุดของบรรทัด ณที่ตำแหน่งต่างๆหากมีจุดของข้อมูลภาพมีลักษณะเดียวกับค่าในหน้าต่างของรูปที่ 3.1 และ 3.2 ก็ทำการเปลี่ยนจุดกลางให้เป็นค่าเดียวกันกับจุดรอบข้าง

0	0	0
0	1	0
0	0	0

รูปที่ 3.1 *สัญญาณรบกวนแบบรูเดี่ยว*

1	1	1
1	0	1
1	1	1

รูปที่ 3.2 *สัญญาณรบกวนแบบจุดเดี่ยว*

รูปที่ 3.3 แสดงตัวอย่างภาพตัวอักษรจากฐานข้อมูลจริงที่สร้างขึ้นก่อนและหลังจากผ่านกระบวนการกำจัดสัญญาณรบกวน จะเห็นว่าวิธีดังกล่าวสามารถกำจัดสัญญาณรบกวนบางส่วนได้คือประเภทจุดเดี่ยวหรือรูเดี่ยว หากแต่ว่ายังมีสัญญาณรบกวนอื่นๆที่ยากต่อการกำจัด เช่น สัญญาณรบกวนที่มีขนาดใหญ่กว่าหนึ่งจุด ทั้งนี้เพราะการกำจัดสัญญาณรบกวนเหล่านี้อาจจะทำให้ข้อมูลของภาพตัวอักษรบางส่วนสูญหายไปด้วย เช่นไม้เอกซึ่งมีลักษณะคล้ายคลึงกับสัญญาณรบกวน

ก่อน	หลัง
	
	
	
	
	

ก่อน	หลัง
	
	
	
	
	

รูปที่ 3.3 ตัวอย่างภาพตัวอักษรก่อนและหลังการกำจัดสัญญาณรบกวน

3.2 การทำภาพตัวอักษรให้บาง (Thinning)

การทำให้ภาพตัวอักษรบางเป็นกระบวนการที่เปลี่ยนแปลงภาพข้อมูลให้เหลือเฉพาะเส้นโครงร่างของภาพตัวอักษร (Skeleton) เท่านั้น โดยข้อมูลของภาพที่ผ่านกระบวนการทำตัวอักษรให้บางนั้นจะเหลือความกว้างของเส้นตัวอักษร 1 จุดภาพ โครงร่างมาจากแกนกลางของตัวอักษร และต้องต่อเนื่องไม่ขาดตอน ซึ่งโดยทั่วไปแล้วพอจะสรุปคุณสมบัติที่ดีของการทำให้ตัวอักษรบางได้ดังต่อไปนี้

- โครงร่างที่ได้ควรมีความต่อเนื่องไม่ขาดตอน (preserving connectivity of skeletons)
- โครงร่างที่ได้ต้องมีความกว้างเท่ากับ 1 จุดภาพ (converging to skeletons of unit width)
- โครงร่างที่ได้ต้องอยู่ประมาณกึ่งกลางของภาพเดิม (closely approximating the medial axis)
- โครงร่างที่ได้ไม่ขึ้นอยู่กับลักษณะของขอบ (possessing insensitivity to boundary) เช่นภาพตัวอักษรแบบเดียวกันสองตัว ตัวหนึ่งมีขอบที่ขรุขระ ส่วนอีกตัวมีขอบราบเรียบ การทำให้บางที่ดีควรจะให้โครงร่างที่คล้ายกัน

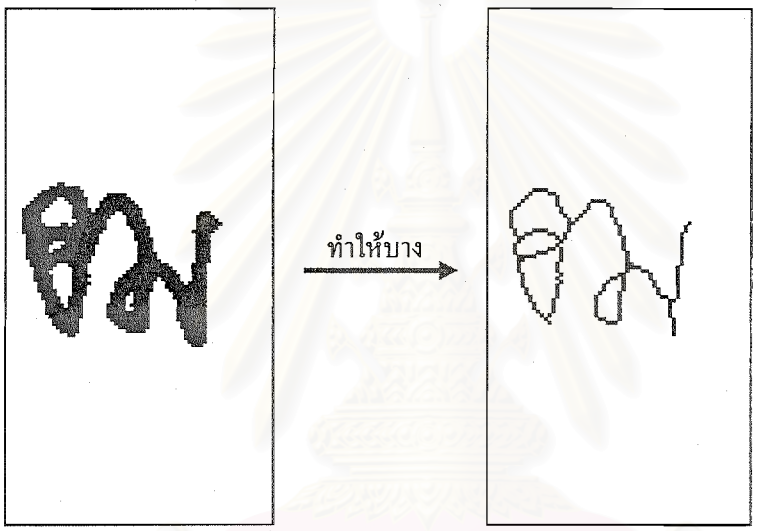
การทำให้ตัวอักษรบางส่วนใหญ่จะใช้วิธีลบจุดภาพส่วนเกินหรือจุดภาพที่ไม่ต้องการซ้ำหลายรอบๆ โดยเฉพาะจุดภาพที่อยู่ขอบของภาพเพื่อที่จะได้โครงร่างที่อยู่ประมาณแกนกลางของภาพเดิม จนคงเหลือความ

กว้างของเส้นอักษร 1 จุดภาพ ซึ่งเป็นจุดภาพที่ไม่สามารถลบได้ เมื่อลบแล้วจะทำให้ภาพไม่ต่อเนื่อง หรือขาดตอน แนวทางหรือวิธีการทำตัวอักษรให้บางสามารถแบ่งออกได้เป็น 2 วิธีใหญ่ ดังนี้

1. การทำให้บางแบบลำดับ (Sequential Processing Thinning) การทำให้บางในลักษณะนี้ในรอบๆ หนึ่งของการลบจุดภาพที่ไม่ต้องการจะลบทีละจุด และการลบของจุดๆหนึ่งในแต่ละรอบจะมีผลกระทบกับการลบของจุดถัดไป

2. การทำให้บางแบบขนาน (Parallel Processing Thinning) เป็นการทำให้บางโดยในแต่ละรอบจะลบจุดภาพพร้อมกันทุกจุด ดังนั้นในแต่ละรอบของการลบ การตัดสินใจลบจุดภาพจุดหนึ่งจะไม่มีผลกับอีกจุดภาพหนึ่ง

รูปที่ 3.4 แสดงถึงภาพก่อนและหลังการทำให้ตัวอักษร ฒ บาง จะเห็นว่าภาพที่ได้ยังคงเก็บรักษาข้อมูลสำคัญๆไว้ได้นั่นคือเรายังสามารถระบุได้ว่าตัวอักษรนี้เป็นตัวอักษร ฒ



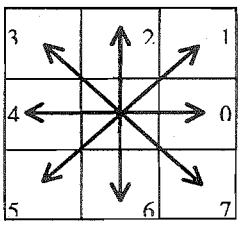
รูปที่ 3.4 ตัวอย่างภาพตัวอักษรก่อนทำให้บาง และหลังการทำให้บาง

การทำภาพตัวอักษรให้บางยังสามารถนำมาใช้ในการหาลักษณะเฉพาะของภาพตัวอักษรที่เหมือนกันของตัวอักษรเดียวกัน และลักษณะที่แตกต่างจากตัวอักษรตัวอื่น โดยที่ลักษณะเฉพาะนั้นถ้าหาจากภาพของตัวอักษรที่ยังหนาอยู่จะลำบาก เช่น จุดปลาย (End Points) จุดแยก (Junctions) เป็นต้น ดังนั้นการทำให้ตัวอักษรบางจึงเป็นส่วนสำคัญส่วนหนึ่งในการพัฒนาระบบรู้จำตัวอักษรเขียนภาษาไทย ในบทนี้จะกล่าวถึงวิธีการทำให้ภาพตัวอักษรบางที่ได้พัฒนาโปรแกรมคอมพิวเตอร์ขึ้นภายใต้โครงการวิจัยนี้ รวมไปถึงการวิเคราะห์และเปรียบเทียบสมรรถนะของแต่ละวิธีเพื่อหาวิธีที่เหมาะสมกับภาษาไทย

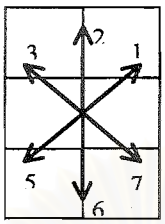
วิธีที่ 1: A New Safe-Point Thinning Algorithm Based on the Freeman Chain Code Tracing

เป็นวิธีการทำให้บางแบบลำดับแบบหนึ่ง ซึ่งทางคณะผู้วิจัยได้ทำการดัดแปลงมาจากวิธี *"A New Safe-Point Thinning Algorithm Based on the Mid-Crack Code Tracing"* ของ Frank Y. Shih and Wai-Tak Wong [1] ส่วนที่แตกต่างไปก็คือการใช้ Freeman Chain Code แทนการใช้ Mid-Crack Code (ดูความแตกต่างระหว่าง code ทั้งสองประเภทได้ในรูปที่ 3.5) วิธีการนี้มีขั้นตอนดังนี้คือ ขั้นแรกจะต้องทำการค้นหาส่วนที่เป็นขอบของภาพตัวอักษร จากนั้นทำการพิจารณาจุดที่อยู่บนขอบของภาพทีละจุดโดยใช้วิธีวิ่งรอบวงขอบ (Contour) ไปเรื่อยๆ ณที่

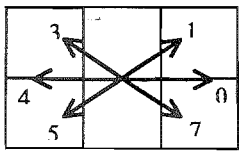
แต่ละจุดก็จะตรวจสอบดูว่าจุดนั้นเป็นจุดที่สามารถลบทิ้งได้หรือไม่ คือเมื่อลบแล้วจะไม่ทำให้โครงร่างของภาพเปลี่ยนไปหรือขาด จุดดังกล่าวมีชื่อเรียกว่า nonsafe border pixel ถ้าหากจุดนั้นมีคุณสมบัติดังกล่าวก็จะลบจุดนั้นทันที ซึ่งแสดงให้เห็นว่าก็ลบจุดภาพนั้นส่งผลต่อการพิจารณาของจุดถัดไป ซึ่งเป็นลักษณะของการทำให้บางแบบลำดับนั่นเอง



The Freeman chain codes



Mid-crack codes on the vertical crack

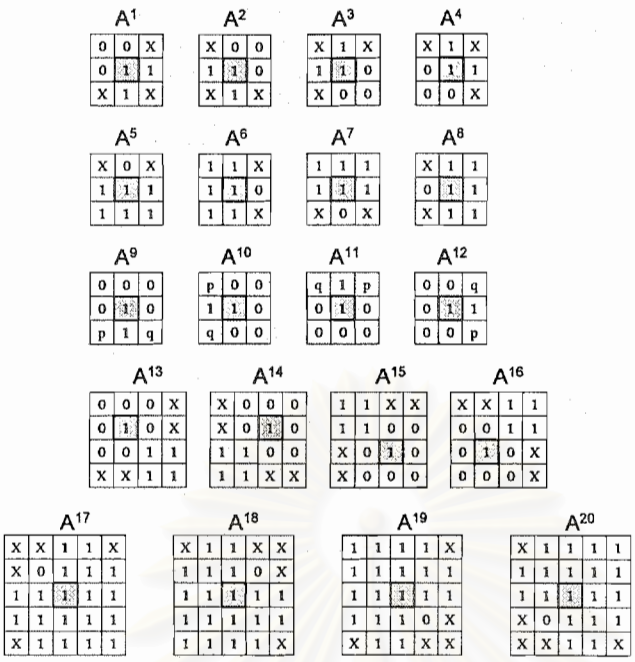


Mid-crack codes on the horizontal crack

รูปที่ 3.5 *The Freeman chain codes และ Mid-crack codes*

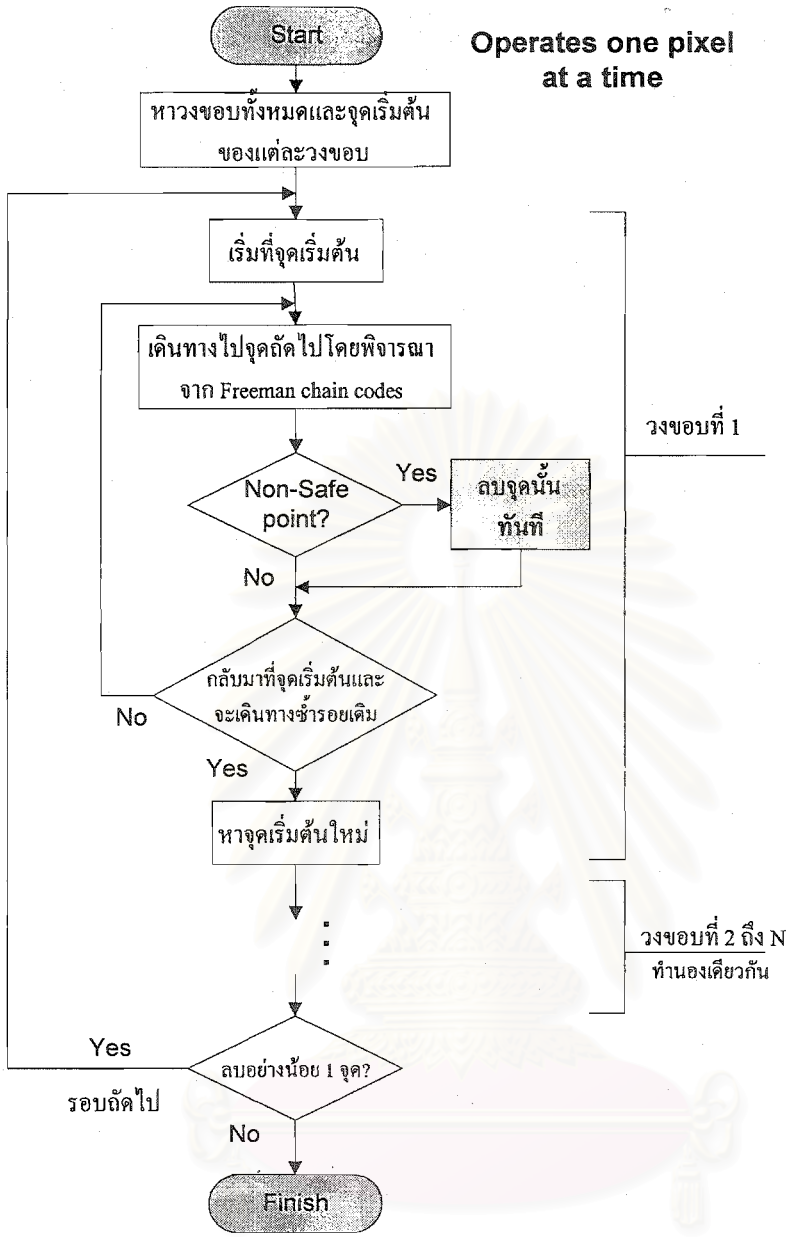
ก่อนที่จะทำการวิ่งวนรอบวงขอบจะต้องทำการหาจำนวนวงขอบทั้งหมดที่มีอยู่ในภาพก่อน ซึ่งสามารถกระทำได้โดยการอ่านภาพทุกจุดหนึ่งรอบก่อน แล้วในแต่ละวงขอบจะใช้ Freeman Chain Codes ดังในรูปที่ 3.5 เพื่อพิจารณาหาตำแหน่งของจุดถัดไป จากจุดกลางจุดหนึ่งที่จะพิจารณาเพื่อหาจุดที่จะเดินทางต่อไปให้เลือกจุดภาพดำที่มีเวคเตอร์ที่ค่าน้อยสุดก่อน ดูรูปที่ 3.5 ประกอบ จะเห็นว่าการพิจารณาจะเริ่มที่จุดที่เวคเตอร์ชี้ไปทางด้านขวาของจุดกลางซึ่งคือเวคเตอร์หมายเลข 0 หากจุดดังกล่าวเป็นจุดดำก็จะเดินไปที่จุดดังกล่าวเลย แต่หากจุดนี้เป็นจุดขาวก็จะพิจารณาจุดต่อไปคือจุดที่เวคเตอร์ชี้ไปทางขวาด้านบนซึ่งคือเวคเตอร์หมายเลขเลข 1 แล้วดูว่าจุดดังกล่าวเป็นจุดดำหรือไม่ เราจะทำตามกระบวนการดังกล่าวนี้ไปเรื่อยๆจนครบทุกจุดบนวงขอบ ส่วนวิธีการที่จะตรวจสอบว่าเป็นจุด nonsafe border pixel หรือไม่นั้นจะใช้หน้าต่าง (Templates) ขนาด 3x3 ตามที่อธิบายไว้ในบทความ [2] ดูรายละเอียดของหน้าต่างแต่ละแบบในรูปที่ 3.6 หากผลการทดสอบพบว่าตรงกับหน้าต่างใดหน้าต่างหนึ่งจุดภาพนั้นก็จะถูกลบทิ้งทันทีหรือก็คือการปรับให้จุดดังกล่าวมีค่าเป็น 0 นั่นเอง สำหรับขั้นตอนการทำงานของวิธีให้ดูจากรูปที่ 3.7

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.6 *หน้าต่าง (Templates) ที่แสดงถึง nonsafe border pixel (also thinning templates)*

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

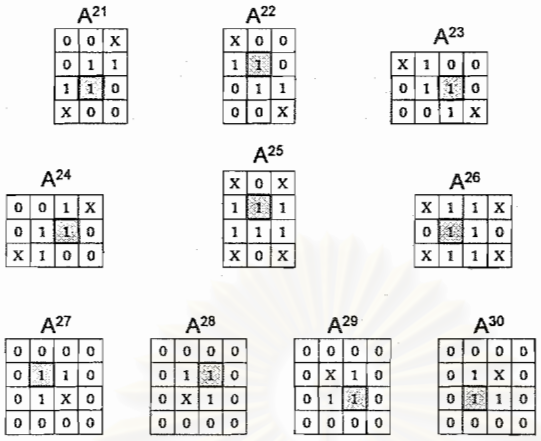


รูปที่ 3.7 *A New Safe-Point Thinning Algorithm based on the Freeman Chain Code Tracing*

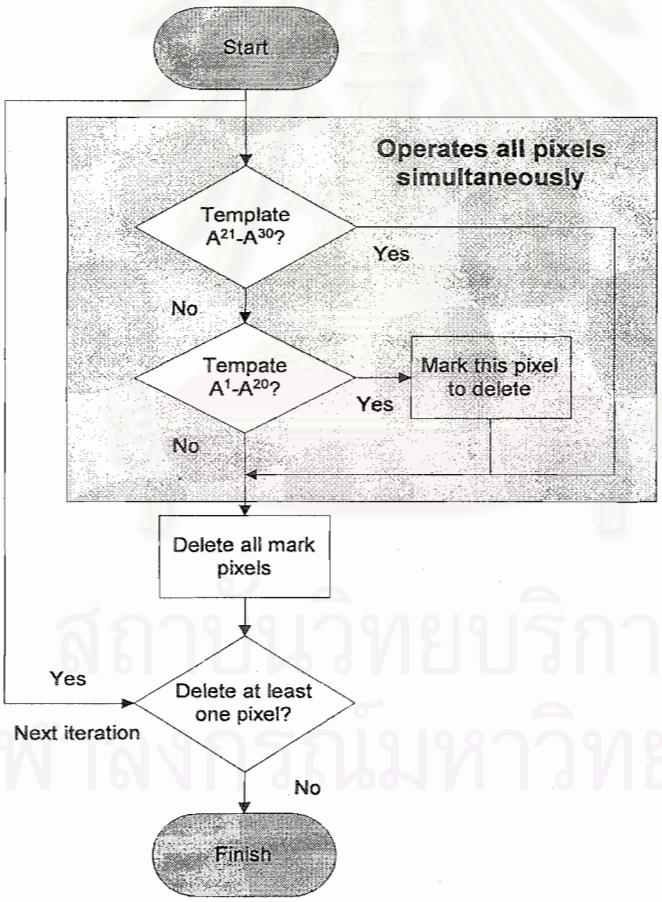
วิธีที่ 2: The One-Pass Algorithm in the Square Grid

วิธีนี้ถูกพัฒนาขึ้นโดย Ben K. Jang and Roland T. Chin [2] ในปีค.ศ. 1992 หลักการของวิธีนี้คือในหนึ่งรอบจะทำการพิจารณาจุดทุกจุดในภาพพร้อมๆกันทีเดียว และจะทำการตัดสินว่ามีจุดใดบ้างที่สามารถลบทิ้งได้ ทั้งนี้การตัดสินของแต่ละจุดจะอาศัยหน้าต่าง (templates) 2 ประเภท คือ thinning templates (รูปที่ 3.6) และ restoring templates (รูปที่ 3.8) โดยที่การพิจารณาแต่ละจุดจะไม่เกี่ยวข้องกับจุดอื่นๆเลย หากพบว่าจุดใดที่มีรูปแบบเหมือนกับ thinning templates หนึ่ง ก็จะจดไว้ว่าจุดนี้มีคุณสมบัติเข้าข่ายที่จะลบทิ้งได้โดยอาจจะไม่ทำให้ภาพเสียหาย แต่กระนั้นถ้าหากจุดดังกล่าวตรงกับ restoring templates หนึ่งแล้วจะต้องไม่ทำการลบจุดดังกล่าว เพราะจะทำให้ข้อมูลในภาพเสียหายได้ เมื่อพิจารณาครบทุกจุดในหนึ่งรอบแล้ว จะทำการลบจุดที่ได้จดไว้ทั้ง

หมดพร้อมๆกันทีเดียว จากนั้นก็ทำการพิจารณารอบใหม่ซ้ำๆกันจนกว่าจะพบว่าไม่มีจุดที่สามารถลบได้อีกแล้ว ดูขั้นตอนต่างๆของวิธีนี้ในรูปที่ 3.9 สังเกตว่าวิธีนี้เป็นวิธีทำให้อักษรบางแบบขนาน



รูปที่ 3.8 Restoring Template

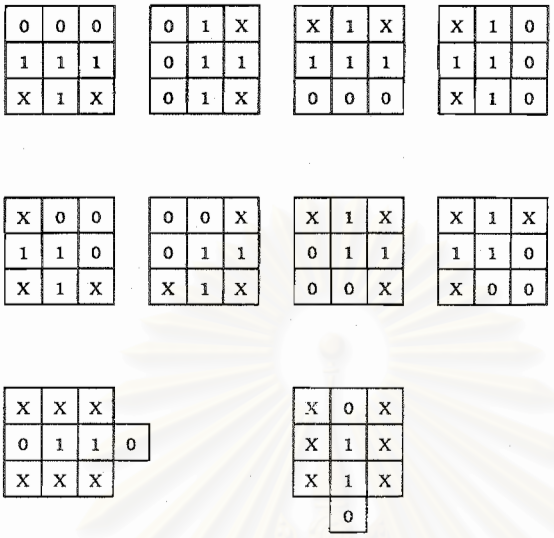


รูปที่ 3.9 *The One-Pass Algorithm in the Square Grid*

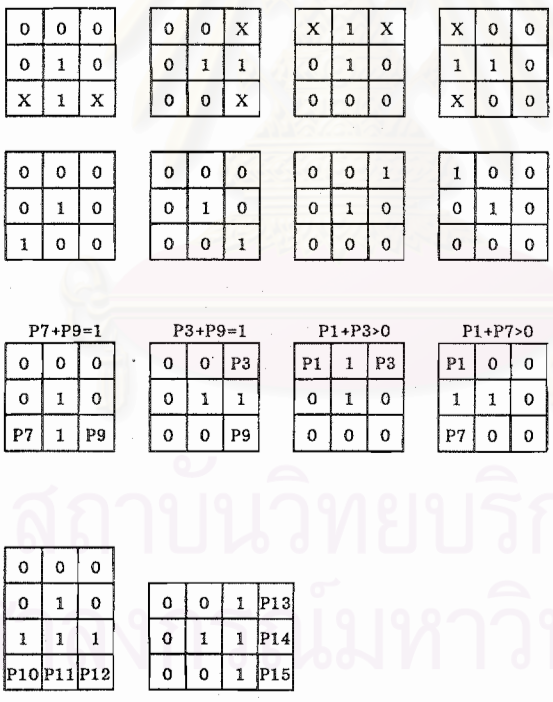
วิธีที่ 3: High-Speed Thinning Processor

วิธีนี้ได้รับการพัฒนาโดย Y.S. Kim, W.S. Chol และ S.W. Kim [3] วิธีนี้จะมีหน้าต่าง (Templates) 2 ชนิด คือ Thinning Templates และ Trimming Templates (ดูรูปที่ 3.10 และ 3.11 ประกอบ) สำหรับใช้ลบจุดภาพที่

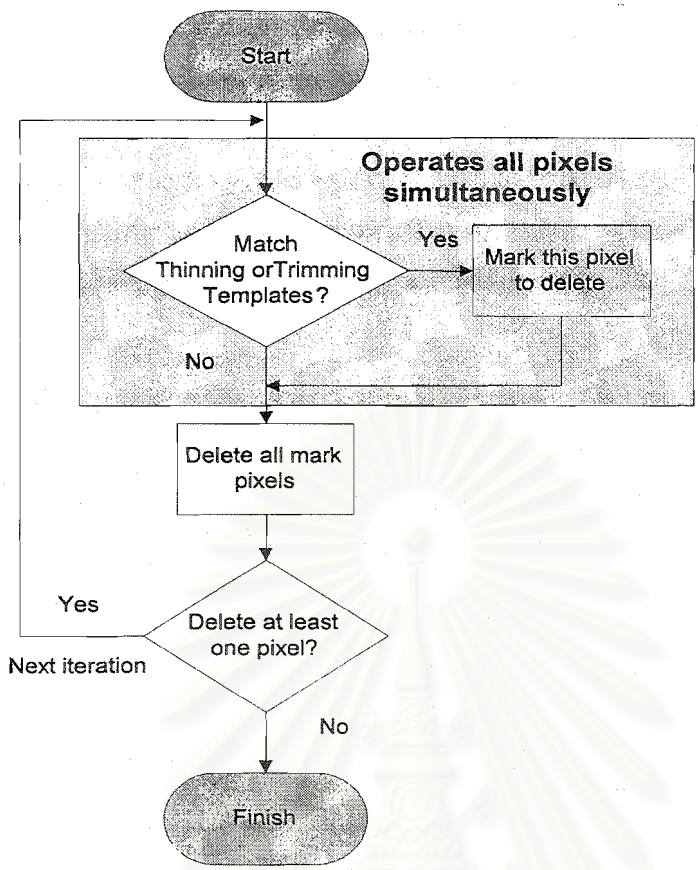
ตรงกับหน้าต่างใดหน้าต่างหนึ่ง ทั้งนี้ Thinning Templates มีไว้สำหรับลบจุดภาพที่อยู่บริเวณขอบ (boundary pixels) ของภาพ ส่วน Trimming Templates ใช้สำหรับลบจุดภาพที่อยู่บริเวณมุม (corner pixels) ของภาพ โดยมีแผนภาพการทำงานโดยรวมจะมีลักษณะดังในรูปที่ 3.12



รูปที่ 3.10 *Thinning Templates*



รูปที่ 3.11 *Trimming Templates*



รูปที่ 3.12 *The High-Speed Thinning Processor*

ผลการทดสอบและวิเคราะห์ผล

ในการวิเคราะห์และตัดสินว่าวิธีใดเป็นวิธีที่เหมาะสมนั้นกระทำได้ค่อนข้างยากมาก เพราะยังไม่มีวิธีการเปรียบเทียบที่มีประสิทธิภาพและเป็นที่ยอมรับกันทั่วไป ด้วยเหตุนี้การเปรียบต่างๆที่ผ่านมาบางบทความจึงอาศัยสายตามนุษย์ในการตัดสิน ซึ่งหลังจากที่คณะผู้วิจัยได้พิจารณาอย่างถี่ถ้วนแล้วจึงได้เลือกใช้วิธีการทดสอบด้วยสายตา และผลจากสังเกตและวิเคราะห์ด้วยสายตาได้สรุปคุณลักษณะของแต่ละวิธีไว้ดังนี้

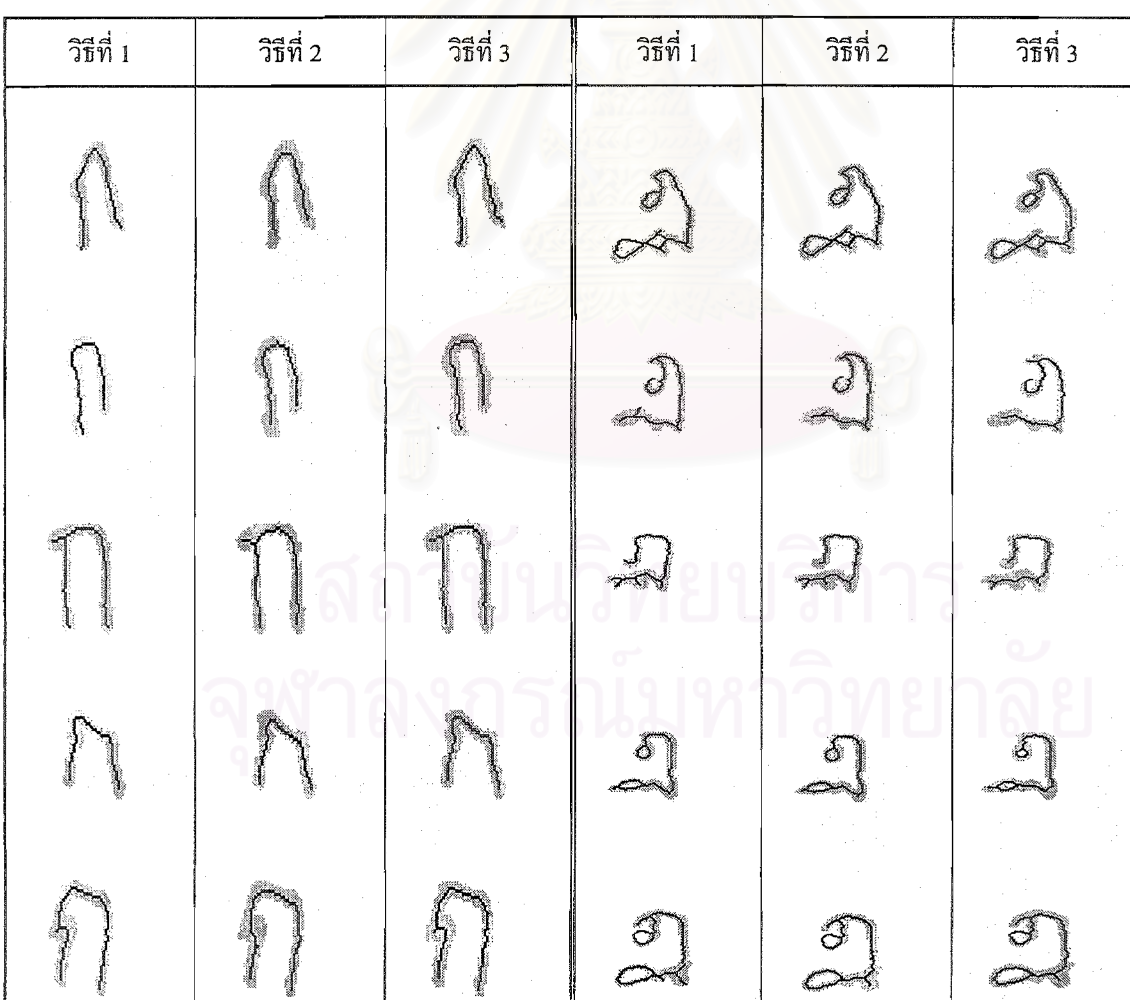
ผลลัพธ์หลังการทำให้บาง	วิธีที่ 1	วิธีที่ 2	วิธีที่ 3
1. ความต่อเนื่องของโครงร่าง	ต่อเนื่อง	ต่อเนื่อง	ต่อเนื่อง
2. ความกว้างโครงร่างเท่ากับ 1 จุด	ดี	ดีที่สุด	ดี
3. โครงร่างประมาณกึ่งกลางของภาพเดิม	ดีที่สุด	ดี	พอใช้
4. โครงร่างขึ้นอยู่กับลักษณะขอบ	น้อย	น้อยที่สุด	มาก
5. ประสิทธิภาพในการลดข้อมูล	ดี	ดีที่สุด	ต่ำที่สุด
6. จำนวนรอบเฉลี่ยต่อตัวอักษร	5.821	4.304	6.512
7. เวลาเฉลี่ยต่อตัวอักษร (มิลลิวินาที)	12.58	10.96	7.44
9. การตัดหยัก เช่น ใน ต, ค	น้อย	มากที่สุด	น้อย
8. ระยะร่นของตัดจุดปลายจากปลายภาพจริงๆ	น้อยที่สุด	มากที่สุด	น้อย
9. ความซับซ้อนของในการเขียนโปรแกรม	มากที่สุด	น้อยที่สุด	น้อย

โดยสรุปแล้ววิธีการทำให้บางแต่ละวิธีมีข้อดีและข้อด้อยที่แตกต่างกันไปดังนี้

- วิธีที่ 1 A New Safe-Point Thinning Algorithm Based on the Freeman Chain Code Tracing เป็นวิธีที่มีขั้นตอนยุ่งยากทำให้การนำมาเขียนเป็นโปรแกรมมีความซับซ้อนที่สุด และวิธีนี้ยังใช้เวลาในการประมวลภาพอักษรนานที่สุดและมีจำนวนการวนรอบมากที่สุด แต่จุดเด่นของวิธีนี้ก็คือการที่สามารถหาโครงร่างที่ประมาณกึ่งกลางภาพตัวอักษรเดิมได้ดีที่สุด เพราะใช้วิธีวนรอบขอบแล้วตัดจุดที่ไม่ต้องการทิ้งไปจนถึงแกนกลาง และข้อดีที่สำคัญอีกประการเมื่อกระบวนการทำให้บางเสร็จสิ้น ผลที่ได้ก็คือรหัสเวคเตอร์ของโครงร่างทันที

- วิธีที่ 2 The One-Pass Algorithm in the Square Grid เป็นวิธีที่มีประสิทธิภาพในการลดข้อมูลสูงที่สุดคือใช้จำนวนรอบน้อยที่สุด แต่มีข้อเสียตรงที่จุดปลายของอักษรจะถูกร่น (Endpoint Reduction) มากที่สุด ส่วนคุณสมบัติอื่นๆ อยู่ในระดับที่น่าพอใจทีเดียว นอกจากนี้ยังสามารถออกแบบฮาร์ดแวร์ในการแยกประมวลพร้อมๆกันได้ เพราะวิธีนี้เป็นวิธีการทำให้บางแบบขนาน

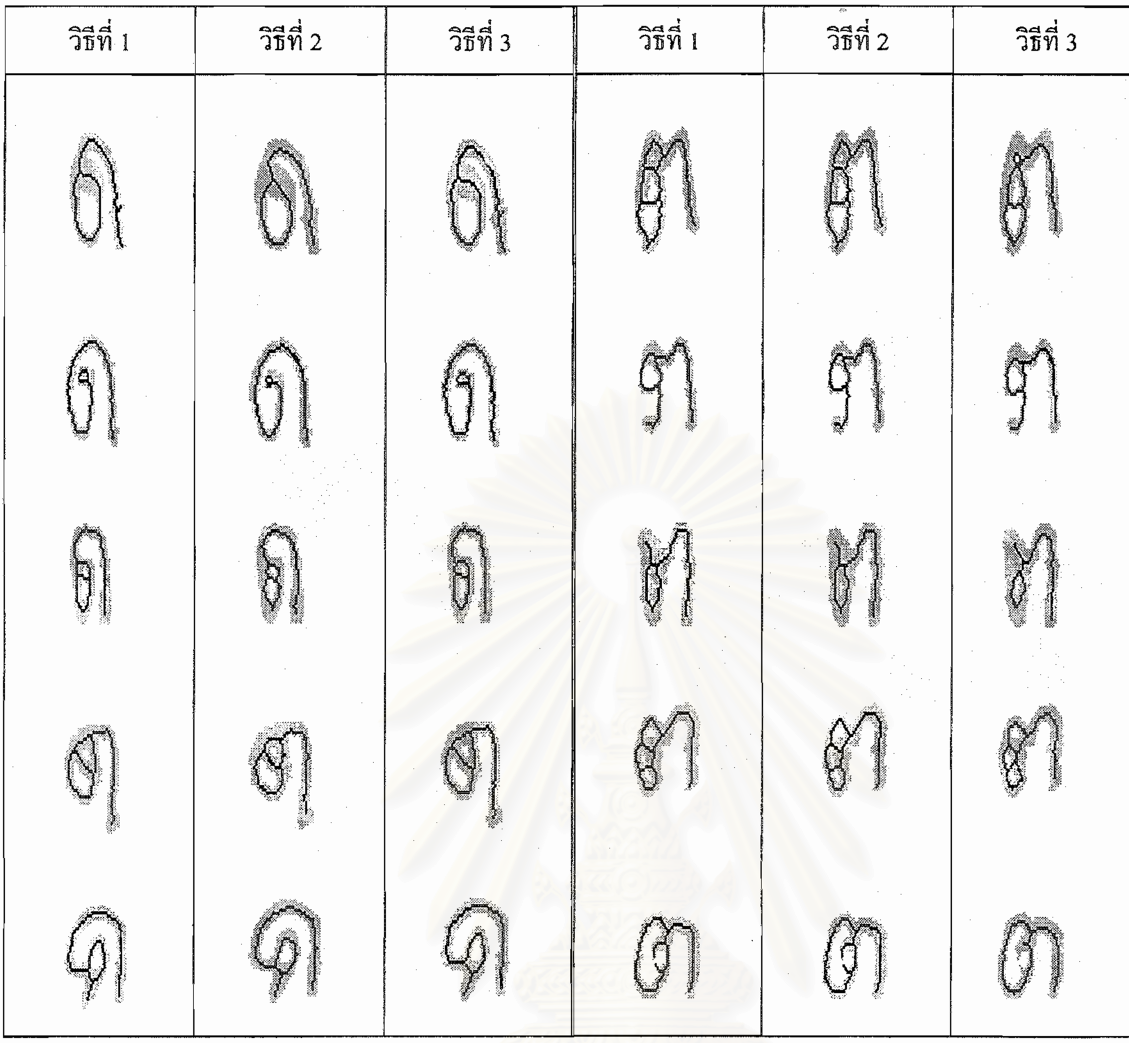
- วิธีที่ 3 High-Speed Thinning Processor ข้อดีของวิธีคือเรื่องของเวลา ถึงแม้จะใช้จำนวนรอบสูงที่สุด แต่หากเปรียบเทียบคุณสมบัติในด้านอื่นวิธีนี้อาจไม่เหมาะสมที่จะนำมาใช้ในทางปฏิบัติ



รูปที่ 3.13 *ภาพตัวอักษรตัวอย่างก่อนและหลังการทำให้บาง เรียงตามวิธี*

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.13 *ภาพตัวอักษรตัวอย่างก่อนและหลังการทำให้บาง เรียงตามวิธี (ต่อ)*

3.3 การปรับขนาดของภาพ (Shape Normalization)

การปรับขนาดของภาพ คือ กรรมวิธีการในการทำให้ภาพของตัวอักษรให้มีขนาดเท่ากัน ทั้งนี้เพื่อขจัดปัญหาเรื่องภาพของอักษรที่เวลาเขียนนั้นอาจจะมีขนาดเล็กและใหญ่ไม่เท่ากัน ขบวนการนี้มีความสำคัญกับระบบรู้จำบางประเภทที่กำหนดว่าข้อมูลที่จะรู้จำต้องมีขนาดที่แน่นอนตายตัว การปรับขนาดของภาพก็นั้นคล้ายกับการย่อขนาดหรือขยายขนาด สำหรับภาพที่ขนาดใหญ่เกินไปหรือขนาดเล็กเกินไปตามลำดับ ทั้งนี้กรรมวิธีการปรับขนาดของภาพแยกเป็น 2 วิธี [4] คือ

- การปรับขนาดแบบเชิงเส้น (Linear Shape Normalization)
- การปรับขนาดแบบไม่เชิงเส้น (Non-Linear Shape Normalization)

การปรับขนาดแบบเชิงเส้น (Linear Shape Normalization)

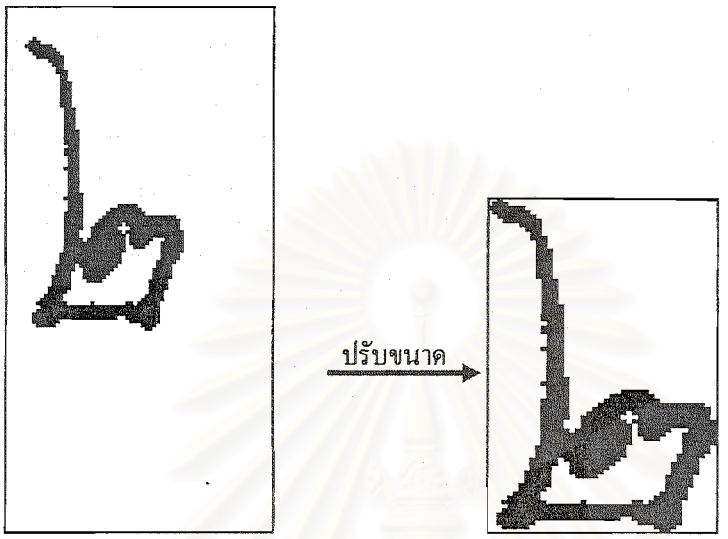
เป็นวิธีการที่มีเป้าหมายในการแปลงภาพจากภาพหนึ่งให้เป็นอีกภาพหนึ่งซึ่งมี ตำแหน่ง ขนาด มุมที่หมุนไป และมุมเอียงที่คงที่ โดยใช้กรอบหรือโมเมนต์ที่จำกัด [4] ทั้งนี้จะได้ความสัมพันธ์ระหว่างจุดภาพเก่า (x, y) กับจุดใหม่ (x', y') ดังสมการ (3.1) และ (3.2)

ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

$$x' = a_1 x + a_2 y + a_3 \quad (3.1)$$

$$y' = a_4 x + a_5 y + a_6 \quad (3.2)$$

โดย $a_1, a_2, .., a_6$ เป็นค่าคงที่ใดๆ ทำให้ภาพหลังการปรับขนาดจะมีรูปร่างเป็นสัดส่วนโดยตรงกับภาพเดิม ดังเช่น ในรูป 3.14

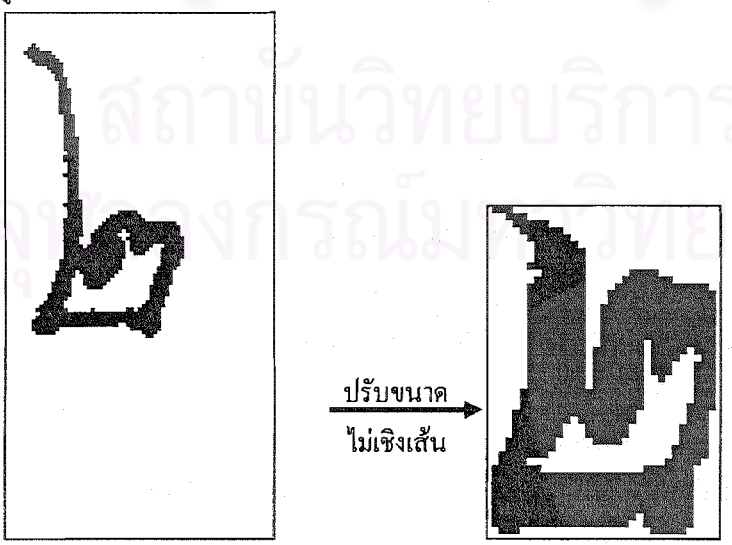


รูปที่ 3.14 *ภาพตัวอักษรก่อนและหลังการปรับขนาด*

รูปที่ 3.14 แสดงผลของการปรับขนาดของภาพ สังเกตว่ากรอบของภาพเริ่มต้นนั้นมีขนาดใหญ่และตัวอักษรก็เขียนชิดไปทางมุมซ้ายด้านบน หลังจากที่ได้ทำการปรับขนาดของภาพแล้วก็จะได้ภาพที่มีขนาดใกล้เคียงกับกรอบโดยที่กรอบของภาพก็มีขนาดลดลงตามที่ต้องการด้วย

การปรับขนาดแบบไม่เชิงเส้น (Non-Linear Shape Normalization)

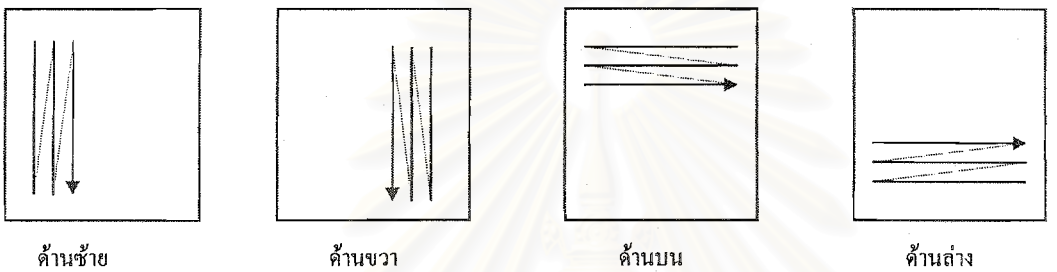
ซึ่งจะพบว่าภาพหลังการปรับขนาดจะมีรูปร่างไม่เป็นสัดส่วนโดยตรงกับภาพเดิม อาจจะมีรูปร่างบิดเบี้ยวไป ดังเช่น ในรูป 3.15



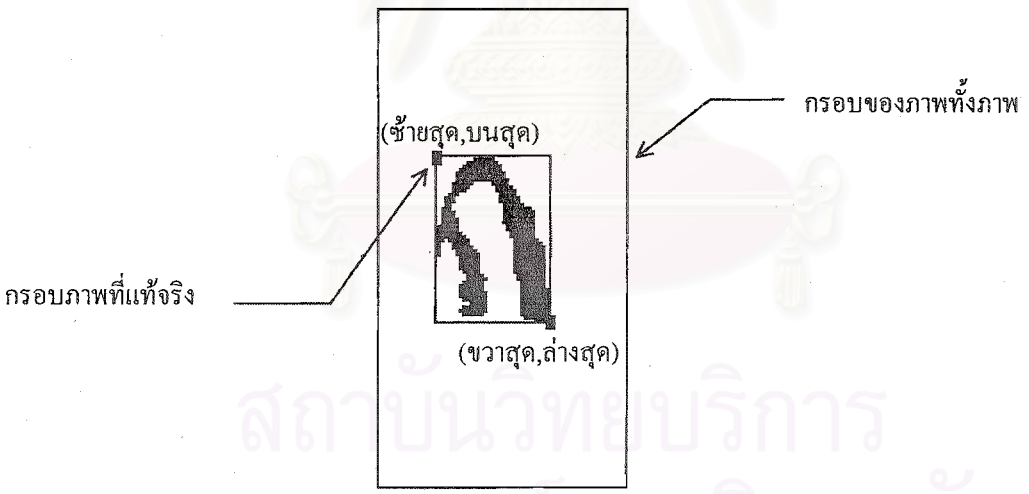
รูปที่ 3.15 *ภาพตัวอักษรก่อนและหลังการปรับขนาดแบบไม่เชิงเส้น*

วิธีการทดสอบ

วิธีการปรับขนาดของภาพตัวอักษรทั้งแบบเชิงเส้นและไม่เชิงเส้นได้นำมาจากบทความของ Seong-Whan Lee, Jeong-Seon Park and Yuan Y.Tang [4] ซึ่งได้เสนอวิธีการปรับขนาดแบบไม่เชิงเส้นไว้หลายวิธี แต่สำหรับงานวิจัยนี้ได้เลือกเพียง 2 วิธีโดยพิจารณาจากความเหมาะสมทางด้านเวลา ความซับซ้อนในการคำนวณ และความถูกต้องเมื่อนำไปใช้ในการรู้จำ ที่ได้ทดสอบและเขียนอยู่ในบทความนั้น โดยวิธีการปรับขนาดทั้งหมดจะใช้ Feature Projection และ Feature density equalization ในการปรับขนาดของภาพจาก IxJ เป็น MxN โดยก่อนที่จะทำการปรับขนาดของภาพจะต้องมีการหาขนาดของกรอบภาพที่แท้จริง (IxJ) ก่อน วิธีการก็คือ หาจุดซ้ายสุด บนสุด ขวาสุด และล่างสุดของภาพให้ได้ [5] โดยมีทิศทางการหาดังรูป 3.16 ซึ่งจะได้กรอบภาพที่แท้จริงที่ตำแหน่ง (ซ้ายสุด, บนสุด) ถึงตำแหน่ง (ขวาสุด, ล่างสุด) ดังรูป 3.17



รูปที่ 3.16 *แสดงทิศทางการหากรอบภาพที่แท้จริงและตำแหน่งกรอบที่ได้*



รูปที่ 3.17 *แสดงทิศทางการหากรอบภาพที่แท้จริงและตำแหน่งกรอบที่ได้*

หลังจากนั้นก็จะทำการปรับขนาดจากภาพที่อยู่ในกรอบเป็นภาพใหม่ (MxN) โดยแต่ละวิธีมีรายละเอียดดังต่อไปนี้

การปรับขนาดแบบเชิงเส้น (Linear Shape Normalization) วิธีการนี้ภาพหลังการปรับขนาดจะขึ้นอยู่กับขนาดของ input image เท่านั้น ไม่ขึ้นอยู่กับลักษณะของ input image ซึ่งจะพบว่าภาพหลังการปรับขนาดจะมีรูปร่างเป็นสัดส่วนโดยตรงกับภาพเดิม

Feature projection functions ในแนวราบและแนวดิ่งเป็นดังสมการ (3.3) และ (3.4) ตามลำดับ

$$H(i) = 1 \tag{3.3}$$

$$V(i) = 1 \tag{3.4}$$

โดย i = 1,2,...,I และ j = 1,2,...,J

Feature density equalization เมื่อใช้ H(i) และ V(j) ของ input image จะพบว่าตำแหน่งจุดภาพใหม่หลังการปรับขนาด (m, n) หาได้จากสมการ

$$m = \sum_{k=1}^{i} H(k) \times \frac{M}{\sum_{k=1}^{I} H(k)} \tag{3.5}$$

$$m = \sum_{k=1}^{i} H(k) \times \frac{M}{\sum_{k=1}^{I} H(k)} \tag{3.6}$$

โดย i = 1,2,...,I ; j = 1,2,...,J; และ m = 1,2,...,M; n = 1,2,...,N

ซึ่งเมื่อแทนสมการที่ (3.5) ด้วยสมการที่ (3.3) และสมการที่ (3.6) ด้วย (3.4) จะได้

$$m = i \times \frac{M}{I} \tag{3.7}$$

$$n = j \times \frac{N}{J} \tag{3.8}$$

การปรับขนาดแบบไม่เชิงเส้น (Non-Linear Shape Normalization) วิธีการนี้ภาพหลักการปรับขนาดนอกจากจะขึ้นอยู่ขนาดยังขึ้นอยู่กับจำนวนจุดดำใน input image ซึ่งจะพบว่าภาพหลังการปรับขนาดจะมีรูปร่างไม่เป็นสัดส่วนโดยตรงกับภาพเดิม อาจจะมีรูปร่างบิดเบี้ยวไป

Feature projection functions ในแนวราบและแนวดิ่งจะหาได้จากจำนวนจุดภาพดำในแนวนั้น ดังสมการ (3.9) และ (3.10) ตามลำดับ

$$H(i) = \sum_{j=1}^{J} f(i,j) + \alpha_H \tag{3.9}$$

$$V(i) = \sum_{i=1}^{I} f(i,j) + \alpha_V \tag{3.10}$$

โดย i = 1,2,...,I และ j = 1,2,...,J

และ *f(i,j)* มีค่าเป็น 1 เมื่อจุด (i, j) เป็นจุดดำ และ 0 เมื่อเป็นจุดขาว

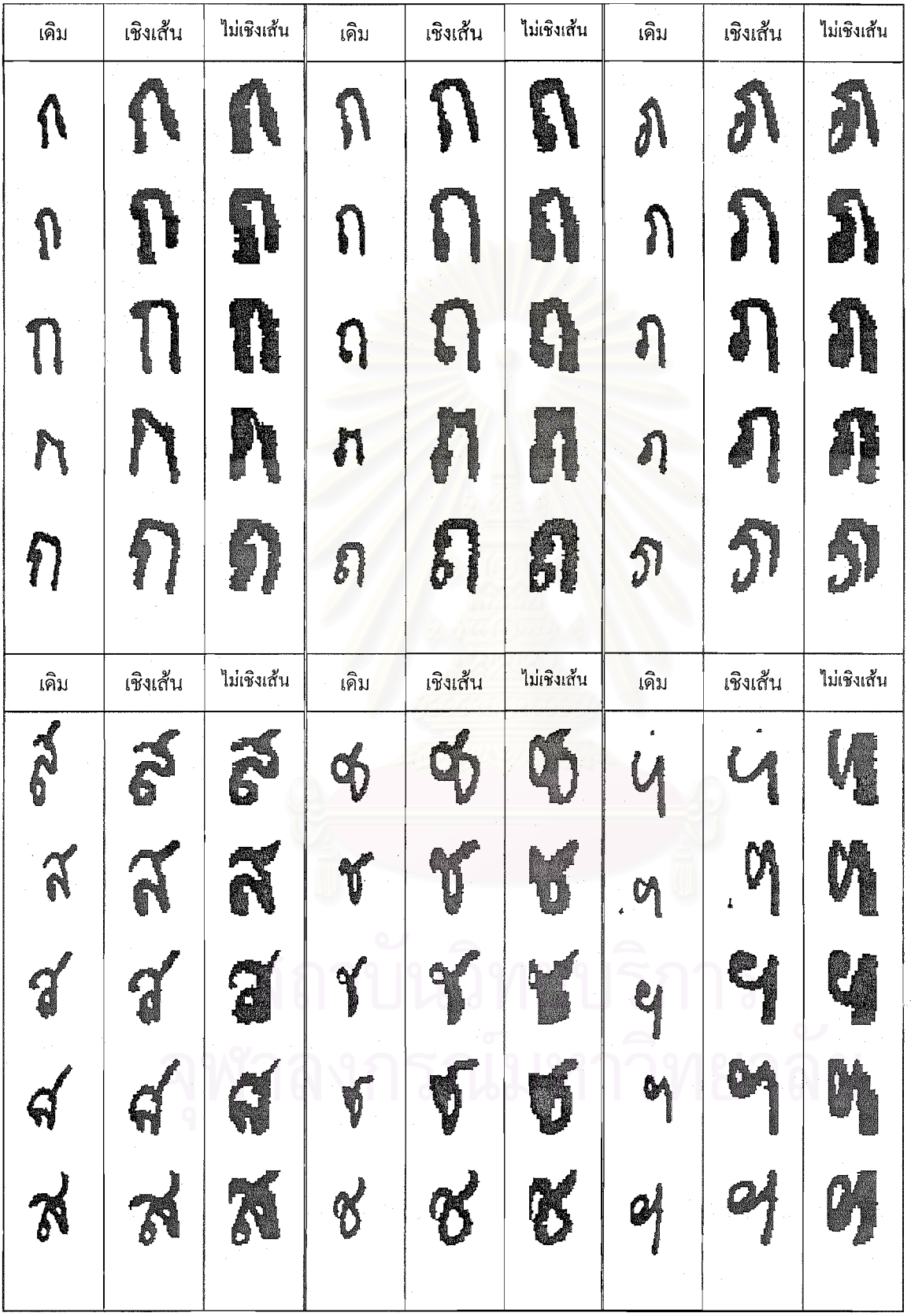
Feature density equalization เมื่อใช้ H(i) และ V(j) ของ input image จะพบว่าตำแหน่งจุดภาพใหม่หลังการปรับขนาด (m,n) หาได้จากสมการ

$$m = \sum_{k=1}^{i} H(k) \times \frac{M}{\sum_{k=1}^{I} H(k)} \tag{3.11}$$

$$m = \sum_{k=1}^{i} H(k) \times \frac{M}{\sum_{k=1}^{I} H(k)} \tag{3.12}$$

โดย i = 1,2,...,I ; j = 1,2,...,J; และ m = 1,2,...,M; n = 1,2,...,N

ผลการทดสอบและวิเคราะห์ผล



รูปที่ 3.18 *ภาพตัวอักษรตัวอย่างก่อนและหลังการปรับขนาดทั้ง 2 วิธี*

I18356485

พิจารณาจากผลการทดสอบในรูป 3.18 ที่พบว่าแต่ละวิธีมีจุดเด่นจุดด้อยต่างกันดังนี้

ผลหลังการปรับขนาด	เชิงเส้น	ไม่เชิงเส้น
1. ขนาดและรูปร่าง	สัดส่วนกับรูปเดิม	ไม่เป็นสัดส่วนกับรูปเดิม
2. ความเหมือนของตัวอักษรเดียวกันที่เขียนจากแต่ละคน	เหมือนน้อยกว่า	เหมือนมากกว่า
2. เวลาเฉลี่ยต่อตัวอักษร (มิลลิวินาที)	24.21	29.85
3. ทนต่อสัญญาณรบกวนที่ไม่อยู่ติดตัวภาพตัวอักษร	ไม่ดีเลย	ดีมาก
4. ความถูกต้องเมื่อใช้โครงข่ายประสาทเทียมในการรู้จำเทียบกับตอนที่ไม่ได้ปรับขนาด	เพิ่มขึ้น	เพิ่มขึ้นมากกว่า

เมื่อพิจารณาการปรับขนาดทั้งสองวิธีจะพบว่าวิธีแบบเชิงเส้นจะสามารถให้รูปภาพตัวอักษรที่คล้ายคลึงการภาพเดิมมากกว่า และใช้เวลาในการประมวลสั้นกว่าเล็กน้อย หากแต่ข้อด้อยของวิธีนี้ก็คือ ในกรณีที่ภาพอักษรมีสัญญาณรบกวน เช่นในกรณีตัวอย่างอักษร ฯ ในรูปที่ 3.18 ภาพตัวอักษรที่ออกมาจะไม่ชิดขอบทางด้านซ้ายเลย และสัดส่วนของภาพตามแนวแกนนอนและแกนตั้งเปลี่ยนไปอย่างมาก ดังนั้นหากนำวิธีนี้มาใช้กับระบบรู้จำแบบเปรียบเทียบจุดข้อมูลแต่ละจุดโดยตรงแล้ว ความถูกต้องของระบบรู้จำจะไม่ถูกต้องนัก จึงได้ข้อสรุปว่าการปรับขนาดของภาพแบบไม่เชิงเส้นจะมีข้อได้เปรียบกว่าวิธีเชิงเส้น ในสภาพที่ภาพตัวอักษรมีสัญญาณรบกวนนอกจากนั้นวิธีไม่เชิงเส้นยังสามารถที่จะจัดการกับหางของตัวอักษรที่มีความยาวไม่เท่ากันให้มีความยาวพอๆ กันได้ เช่น กรณี ส และ ช

เพื่อให้เกิดความมั่นใจยิ่งขึ้นว่าการวิเคราะห์ผลดังที่กล่าวมานี้มีความถูกต้อง จึงได้ทำการทดสอบกับระบบรู้จำซึ่งสร้างขึ้นโดยใช้โครงข่ายประสาทเทียม (ในลักษณะการทำงานคล้าย Pattern Matching) และผลที่ได้พบว่าการใช้วิธีปรับขนาดไม่เชิงเส้นให้อัตราการรู้จำที่ดีกว่าการใช้วิธีปรับขนาดเชิงเส้นตามที่ได้คาดหมายไว้ หมายเหตุ: การปรับขนาดทั้งสองวิธีก็มีส่วนทำให้อัตราการรู้จำที่ดีขึ้นกว่าการไม่มีการปรับขนาดเลย

3.4 การหมุนภาพ (Rotation)

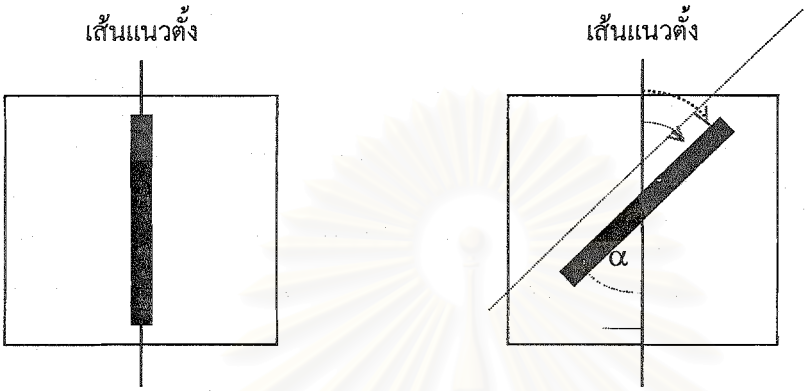
การหมุนภาพ คือ วิธีการหมุนภาพของตัวอักษรเป็นมุมค่าต่างๆ โดยการหมุนภาพนั้นจะไม่ทำให้ขนาดของภาพเปลี่ยนไปแต่อย่างใด การหมุนภาพมีประโยชน์สำหรับใช้ในการแก้ปัญหาของภาพตัวอักษรที่วางตัวหมุนไปจากแนวปกติ ซึ่งอาจจะเกิดจากพฤติกรรมการเขียนของแต่ละบุคคล หรือเนื่องมาจากการวางกระดาษที่เอียงในระหว่างการสแกนข้อมูลภาพ การหมุนเอียงของภาพอาจจะส่งผลกระทบต่อประสิทธิภาพการรู้จำของระบบ ดังแสดงในรูป 3.19

การที่จะนำการหมุนภาพไปประยุกต์กับระบบรู้จำเราจะต้องทราบค่ามุมที่ภาพตัวอักษรหมุนไปเท่าไร หลังจากนั้นก็หมุนภาพกลับด้วยมุมเท่านั้น ดังรูป 3.20 ที่สมมติว่าภาพตัวอักษรหมุนไปเป็นมุม α กับแนวตั้ง เมื่อทราบถึงมุมที่จะหมุนไปก็อาศัยวิธีการทางคณิตศาสตร์พื้นฐานในการหาพิกัดของภาพหลังทำการหมุน (x',y') โดยจะหามาได้จากพิกัดเดิม (x,y) และมุมที่เอียงไปจากเส้นแนวตั้ง ดังรูป 3.21 และอาศัยสมการ (3-13) และ (3-14)

$$x' = r \cdot \sin\alpha \quad (3\text{-}13)$$

$$y' = r \cdot \cos\alpha \quad (3\text{-}14)$$

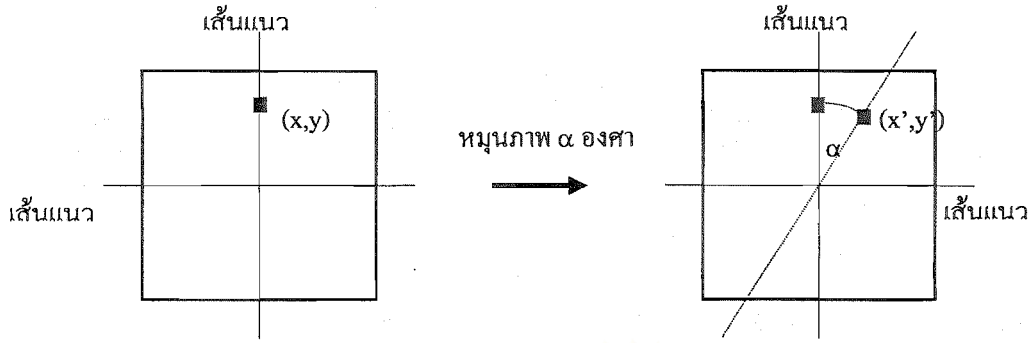
โดยที่ $r = \sqrt{x^2 + y^2}$



รูปที่ 3.19 *ตัวอย่างการหมุนภาพไปเป็นมุม α กับแนวตั้ง*

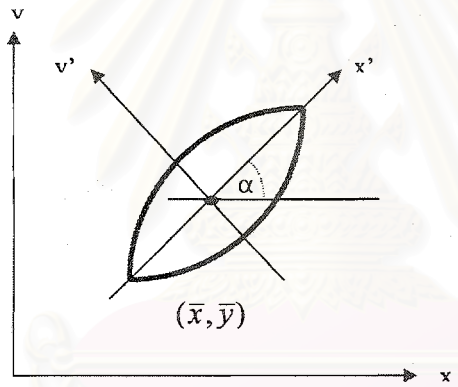
รูปที่ 3.20 *ภาพของตัวอักษรก่อนและหลังการหมุน*

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.21 *พิกัดที่เปลี่ยนไปเมื่อทำการหมุนภาพ*

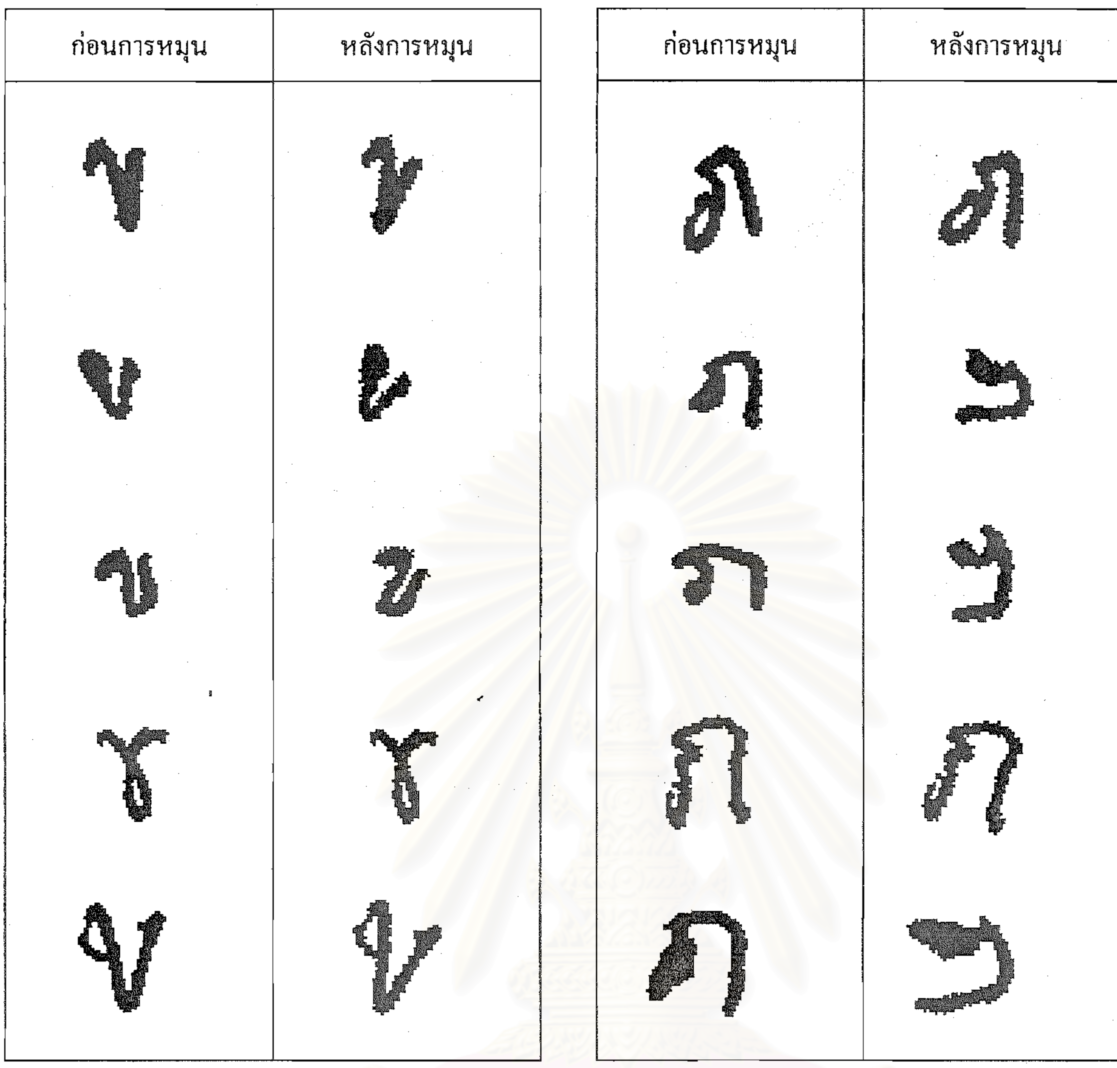
ในการประยุกต์กับระบบรู้จำซึ่งต้องการขจัดปัญหาของภาพตัวอักษรที่หมุน สิ่งที่สำคัญคือจะต้องหามุมที่ตัวอักษรหมุนไปจากแนวปกติ แล้วจึงทำการหมุนกลับด้วยมุมเท่านั้นโดยเทียบกับจุดศูนย์กลางมวล $(\overline{x}, \overline{y})$ ดังรูป 3.22 ซึ่งการหามุมที่ภาพตัวอักษรวางตัวนี้จะใช้หลักการของโมเมนต์ (Moment Descriptors) ซึ่งจะสามารถหาจุดศูนย์กลางมวล (Center of mass) และแนวการวางตัว (Orientation, α)



รูปที่ 3.22 *แนวการวางตัวของภาพ*

ผลการทดสอบและการวิเคราะห์ผล

หลังการหาจุดศูนย์กลางมวลและแนวการวางตัว แล้วนำค่ามุมที่ได้ไปหมุนภาพตัวอักษรตัวอย่าง ผลที่ได้แสดงดังรูป 3.23 จากรูป 2.23 จะเห็นได้ว่าตัวอักษรเดียวกันที่เขียนจากคนต่างกันเมื่อนำมาหมุนภาพกลับด้วยมุมที่มีค่าเท่ากับแนวการวางตัวที่หาจากหลักการโมเมนต์ แล้วจะให้ผลที่ไม่ดี (ตัวอักษรของแต่ละคนวางตัวต่างแนวกัน) ทั้งนี้เนื่องมาจากภาพของตัวอักษรนั้นมีความซับซ้อนไม่มีเหมือนวัตถุรูปทรงเรขาคณิตธรรมดา ทำให้การคำนวณหาจุดศูนย์กลางมวล (Center of Mass) และแนวการวางตัว (Orientation) จากหลักการโมเมนต์ (Moment Descriptors) ได้แนวการวางตัวการวางตัวที่ไม่ถูกต้อง หรือไม่ใช่มุมที่ต้องการ ทำให้เมื่อนำมาใช้ในการหมุนภาพกลับจึงได้ผลไม่ดี ดังนั้นสิ่งที่สำคัญของการแก้ไขภาพตัวอักษรที่วางตัวหมุนไปจากแนวปกติก็คือ ต้องหามุมที่ภาพนั้นหมุนไปจากแนวปกติเท่าไรก่อน



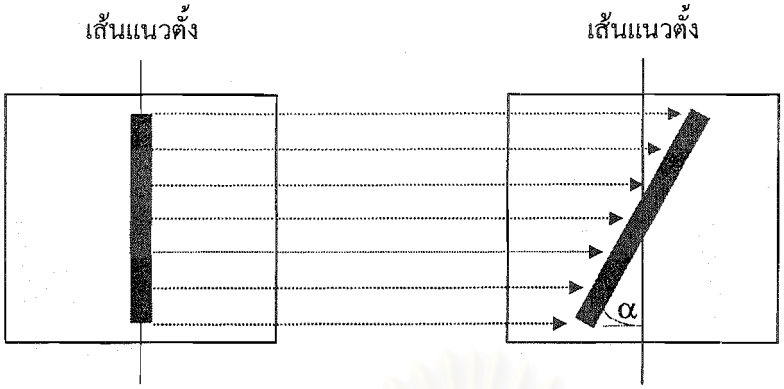
รูปที่ 3.23 *แสดงภาพตัวอักษรตัวอย่างก่อนและหลังการหมุนภาพ*

3.5 การทำให้ภาพโย้ (Slant)

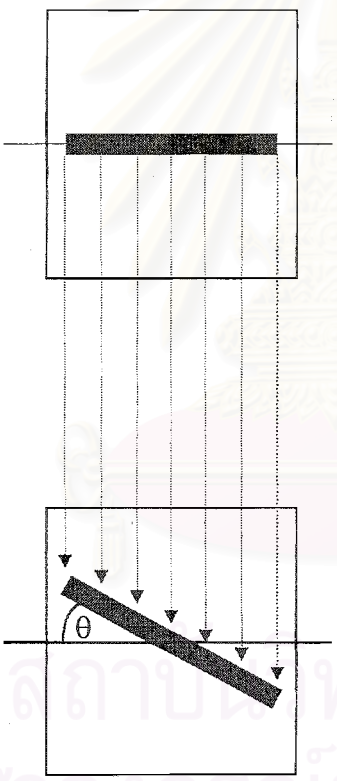
การทำให้ภาพโย้ คือ การเอียงภาพของตัวอักษรในแนวตั้งหรือแนวนอนแนวใดแนวหนึ่ง หรือทั้งสองแนว สิ่งที่แตกต่างระหว่างการหมุนภาพกับการทำให้ภาพโย้ คือภาพที่ได้หลังจากการทำให้โย้จะมีขนาดเปลี่ยนไป ส่วนการหมุนภาพนั้นภาพที่ได้ยังมีขนาดเท่ากับภาพเดิมเพียงแต่เปลี่ยนมุมในการวางเท่านั้น โดยจะแบ่งการทำให้ภาพโย้เป็น 2 แนว คือ

- **แนวราบ (Horizontal)** พิกัดในแนวราบ (x) จะเปลี่ยนไป โดยภาพจะทำมุมกับเส้นแนวตั้งเปลี่ยนไปจากเดิม ดังรูปที่ 3.24
- **แนวดิ่ง (Vertical)** พิกัดในแนวดิ่ง (y) จะเปลี่ยนไป โดยภาพจะทำมุมกับเส้นแนวนอนเปลี่ยนไปจากเดิม ดังรูปที่ 3.25

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

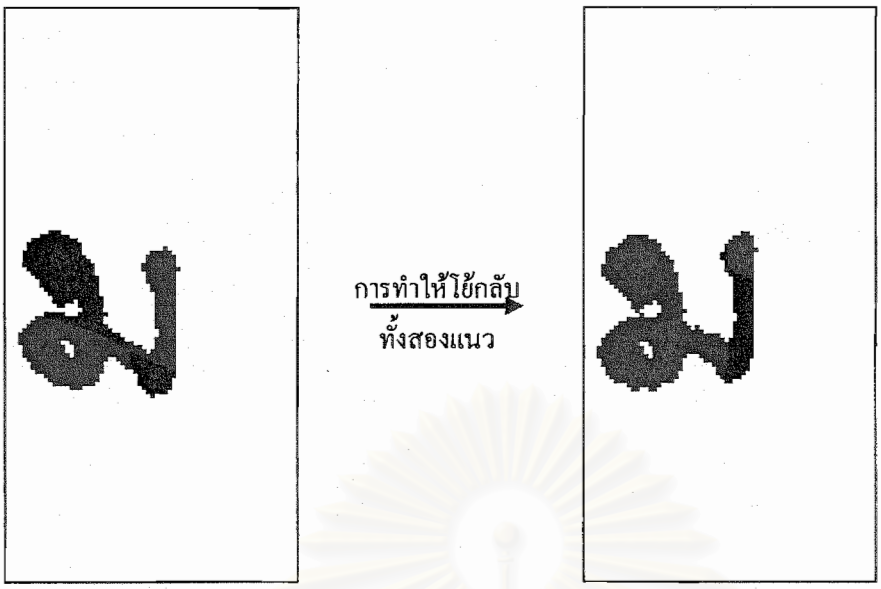


รูปที่ 3.24 *การทำให้ภาพโย้ในแนวราบเป็นมุม α กับเส้นแนวตั้ง*



รูปที่ 3.25 *การทำให้ภาพโย้ในแนวดิ่งเป็นมุม θ กับเส้นแนวนอน*

เมื่อเราจะนำการทำให้ภาพโย้ไปประยุกต์กับระบบรู้จำนั้น เราจะต้องหามุมที่ภาพตัวอักษรโย้ไปในแนวราบและแนวดิ่งเท่าไร หลังจากนั้นก็โย้ภาพกลับด้วยมุมเท่านั้น ดังรูป 3.26



รูปที่ 3.26 ภาพของตัวอักษรที่โย้ และหลังการจากโย้กลับ

สาเหตุสำคัญที่ต้องมีการทำให้ภาพโย้คือ โดยปกติแล้วตัวเอียงธรรมดา (Italic) นั้นจะโย้ทางแนวราบเท่านั้น หากเราต้องการปรับให้ตรง (ตัวเอียงเป็นปัญหาสำคัญปัญหาหนึ่งในการรู้จำ) โดยทำการหมุนทั้งภาพจะเป็นวิธีการได้ผลไม่ตรงกับความต้องการนัก ดังรูป 3.27

รูปที่ 3.27 ภาพของตัวอักษรที่โย้ และหลังการจากหมุน

ผลการทดสอบและวิเคราะห์ผล

รูปที่ 3.28 แสดงตัวอย่างภาพของตัวอักษรก่อนและหลังการโย้ จะเห็นได้ว่าจากการโย้กลับทั้งสองแนว ถึงแม้ว่าจะไม่สามารถทำให้ตัวอักษรกลับมาวางตัวในแนวปกติ แต่ว่าก็สามารถทำให้ตัวอักษรเดียวกันวางอยู่ใน

แนวเดียวกันได้พอสมควร (มองด้วยตาเปล่า) ซึ่งน่าจะมีประโยชน์กว่าวิธีการหมุนกลับ ในการใช้กับระบบรู้จำที่ได้ออกแบบไว้ ที่เป็นลักษณะคล้าย Matching เพราะหากทำให้ตัวอักษรเดียวกันคล้ายกันได้มากเท่าไรก็จะมีความถูกต้องมากเท่านั้น (และน่าจะแก้ปัญหาตัวเอียงแบบ *Italic* ในระบบรู้จำตัวพิมพ์ได้) แต่หากจะไปใช้ในระบบรู้จำลายมือเขียนที่ใช้ลักษณะแตกต่างระหว่างตัวอักษรในการรู้จำและพิจารณาแยกแยะอาจจะไม่มีความจำเป็นก็ได้ อาจจะทำให้เสียเวลาในการประมวลผลเปล่า เพราะในลายมือเขียนมีความซับซ้อนมากการทำให้ตัวอักษรเดียวกันของทุกคนเหมือนกันคงทำได้ลำบาก

ก่อนการโย้	หลังการโย้		ก่อนการโย้	หลังการโย้		ก่อนการโย้	หลังการโย้
๘	๘		๓	๓		ข	ข
๘	๘		๓	๓		ข	ข
๘	๘		๓	๓		ข	ข
๘	๘		๓	๓		ข	ข
๘	๘		๓	๓		ข	ข
๑	๑		๙	๙		๕	๕
๑	๑		๙	๙		๕	๕
๑	๑		๙	๙		๕	๕
๑	๑		๙	๙		๕	๕
๑	๑		๙	๙		๕	๕

รูปที่ 3.28 *แสดงภาพตัวอักษรตัวอย่างก่อนและหลังการโย้กลับ*

4. การหาลักษณะเฉพาะที่บ่งถึงความแตกต่างของภาพตัวอักษร (Distinctive Feature Extraction)

ในบทนี้จะกล่าวถึงองค์ประกอบอีกส่วนหนึ่งที่มีความสำคัญโดยตรงต่อสมรรถนะของระบบรู้จำที่พัฒนาขึ้น และเป็นส่วนที่เป็นพื้นฐานที่จำเป็นต่อการค้นหาแนวทางการรู้จำอักษรภาษาไทย ส่วนนี้มีชื่อเรียกว่า การหาลักษณะเฉพาะที่บ่งถึงความแตกต่างของภาพตัวอักษร (Distinctive feature extraction) โดยปกติแล้วการที่มนุษย์สามารถแยกแยะอักษรแต่ละตัวออกจากกันได้นั้นก็เพราะอักษรแต่ละตัวมีลักษณะเฉพาะของตัวเองที่มีความแตกต่างจากอักษรตัวอื่นๆ และเมื่อเราเห็นอักษรดังกล่าวแล้วก็จะบ่งบอกได้ว่าอักษรดังกล่าวคืออักษรอะไร ในการพัฒนาระบบรู้จำก็เช่นกันจำเป็นที่จะต้องค้นหาลักษณะที่แตกต่างของตัวอักษรแต่ละตัว เพื่อนำไปใช้ในการตัดสินใจและแยกแยะตัวอักษรเหล่านั้น ลักษณะเฉพาะที่ทำให้อักษรแต่ละตัวแตกต่างนั้นสามารถที่จะพิจารณาได้ในหลายลักษณะ ยกตัวอย่างเช่น หัวของตัวอักษรสามารถนำมาใช้ในการแยกแยะระหว่าง ก และ ถ เพราะ ก ต่างจาก ถ ตรงที่ไม่มีหัว หมายเหตุ: หัวของตัวอักษรเป็นคุณลักษณะพิเศษอย่างหนึ่งของตัวอักษรภาษาไทย ที่สามารถนำมาใช้บ่งถึงความแตกต่างของตัวอักษรได้เป็นอย่างดี สำหรับโครงการนี้ได้เลือกทำการวิจัยกับคุณลักษณะที่คาดว่าจะเป็นประโยชน์กับการรู้จำอักษรภาษาไทยทั้งสิ้น 4 ประเภท ดังต่อไปนี้

- ขอบของภาพ (Edge)
- จุดปลายของภาพ (Endpoint)
- หัวของตัวอักษร (Head of Character)
- จำนวนการเปลี่ยนจากจุดขาวเป็นจุดดำ (Stoke Changing)

สำหรับแนวทางในการหาคุณลักษณะทั้ง 4 ประเภท จะได้กล่าวถึงต่อไปข้างล่าง

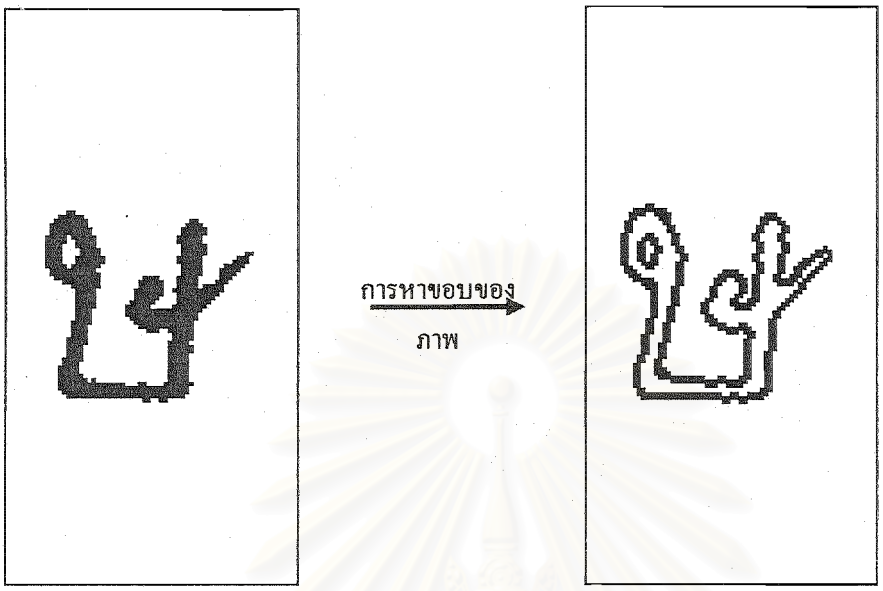
4.1 การหาขอบของภาพ (Edge Detection)

การหาขอบของภาพ คือ การหาขอบของภาพตัวอักษร และผลที่ได้จะเป็นขอบของภาพ โดยเนื้อในของภาพจะถูกตัดออกไป ดังแสดงในรูปที่ 4.1 ทั้งนี้ Ioannis Pitas [6] กล่าวว่า "Edges are basic image feature. They carry useful information about object boundaries which can be used for image analysis, object identification and for image filtering applications as well." และจะพบว่างานวิจัยเกี่ยวกับระบบรู้จำเป็นจำนวนมาก [7,8] ใช้ขอบของภาพเป็นลักษณะที่บ่งความแตกต่างในการรู้จำ

ถึงแม้ว่าวิธีการหาขอบของภาพมีหลายวิธี เช่น Sobel edge detector algorithm, Prewitt edge detector algorithm เป็นต้น แต่เมื่อพิจารณาจากบทความหลายบทความ [7,8] ซึ่งได้นำขอบของภาพมาใช้ในการรู้จำตัวอักษรทั้งตัวพิมพ์และลายมือเขียนภาษาอังกฤษ พบว่านิยมใช้ Kirsch Masks (templates 3x3 ดังรูปที่ 4.2) กันอย่างมาก ดังนั้นเราจึงเลือกที่จะใช้ Kirsch Mask ในการหาขอบของภาพ โดยจะหาขอบในแนวต่างๆ ทั้งหมด 4 แนว คือ

1. แนวตั้ง (Vertical Direction)
2. แนวนอน (Horizontal Direction)
3. แนวเส้นทแยงมุมหลักหรือแนวเส้นทแยงมุมขวา (Right-Diagonal Direction)
4. แนวเส้นทแยงมุมรองหรือแนวเส้นทแยงมุมซ้าย (Left-Diagonal Direction)

สำหรับขั้นตอนการหาขอบภาพโดยละเอียดให้ดูได้จากแผนภาพในรูปที่ 4.3



รูปที่ 4.1 ภาพของตัวอักษรและขอบของภาพที่หาได้

-1	0	1
-1	0	1
-1	0	1

Vertical

1	1	1
0	0	0
-1	-1	-1

Horizontal

0	1	1
-1	0	1
-1	-1	0

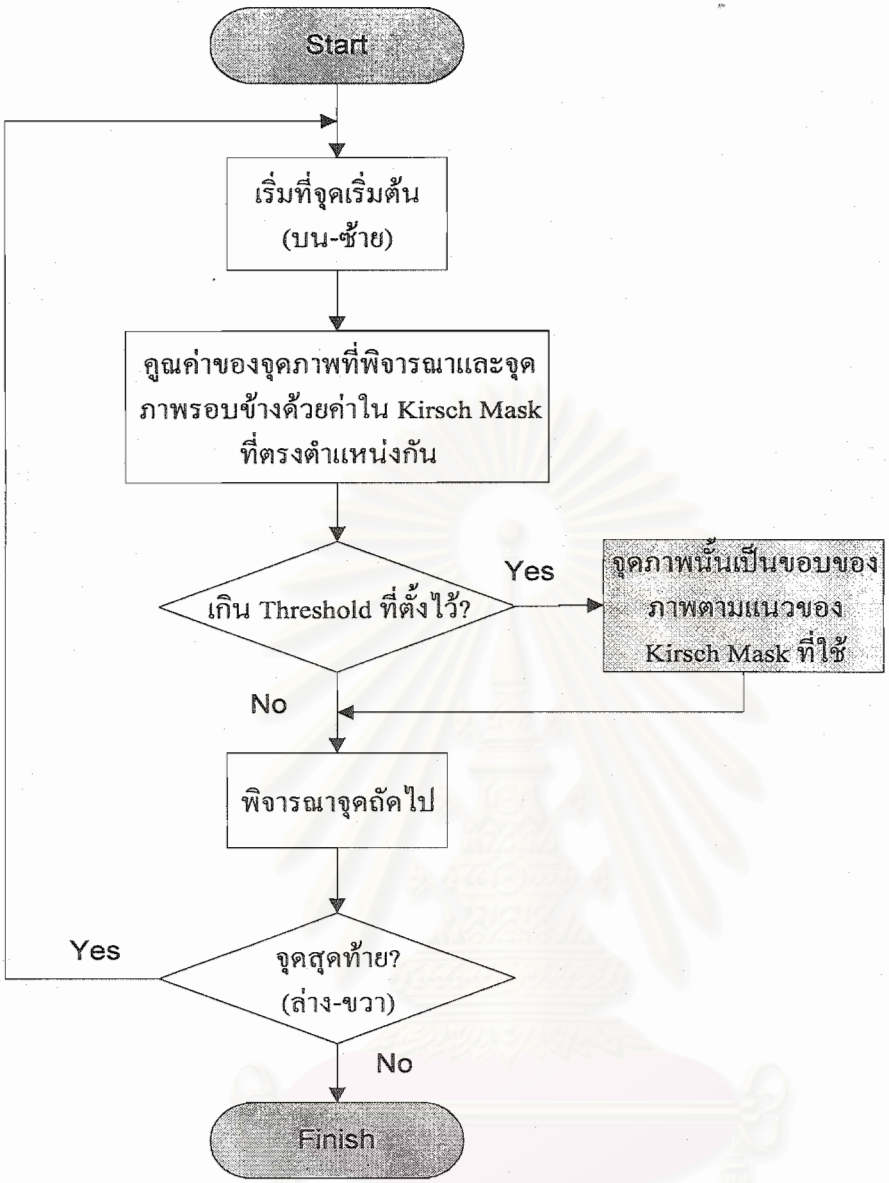
Right-Diagonal

1	1	0
1	0	-1
0	-1	-1

Left-Diagonal

รูปที่ 4.2 แสดง Kirsch Masks ที่ใช้หาขอบในแนวต่างๆ

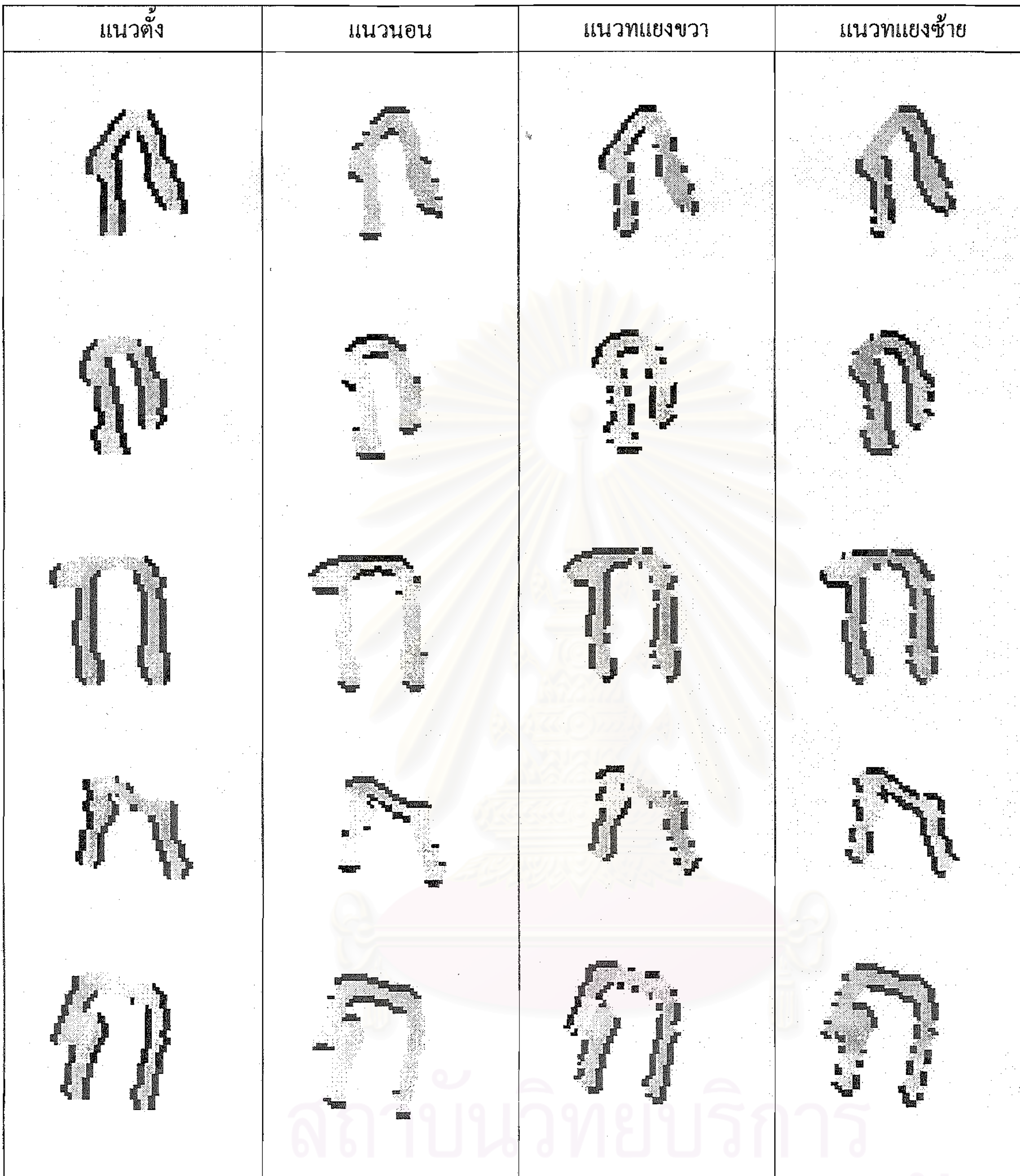
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.3 *ขั้นตอนการหาขอบของภาพโดยวิธี Kirsch Masks*

ผลการทดสอบและการวิเคราะห์

รูปที่ 4.4 แสดงตัวอย่างภาพของตัวอักษรที่ผ่านกระบวนการหาขอบของภาพ ตามแนวตั้ง แนวนอน แนวทแยงขวา และแนวทแยงซ้าย ตามลำดับ โดยจุดสีเข้มแทนขอบ จุดสีอ่อนแทนภาพเดิม จากรูปเราสามารถนำมาใช้ประโยชน์ในการแยกแยะตัวอักษรได้พอสมควร เช่น อักษร ก จะไม่ค่อยมีขอบในแนวตั้งทางด้านบนของตัวอักษร แต่จะมีขอบในแนวตั้งอย่างชัดเจนทั้ง 2 ข้าง อักษรตัวอื่นๆก็จะมีคุณลักษณะของขอบที่แตกต่างกันไป ซึ่งทำให้เราสามารถนำลักษณะที่แตกต่างดังกล่าวมาช่วยในการแยกกลุ่มตัวอักษรภาษาไทยได้โดยเพื่อประยุกต์ใช้กับระบบรู้จำต่อไป



รูปที่ 4.4 *แสดงตัวอย่างภาพหลังจากหาขอบของภาพ*

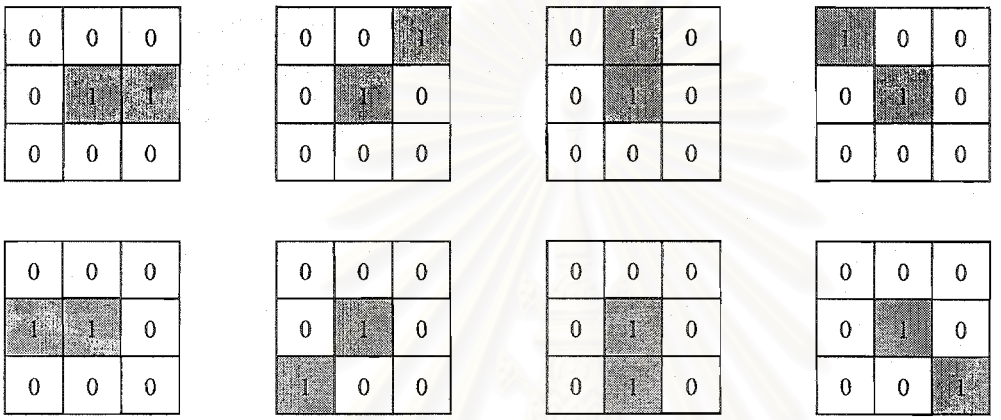
รูปที่ 4.4 *แสดงตัวอย่างภาพหลังจากหาขอบของภาพ (ต่อ)*

4.2 การหาจุดปลาย (End Points)

จุดปลายคือจุด ที่มีจุดที่อยู่รอบข้างเพียงจุดเดียว ตำแหน่งและจำนวนของจุดปลายเป็นลักษณะบ่งถึงความแตกต่างของตัวอักษรอย่างหนึ่งที่มีประโยชน์และสามารถนำมาประยุกต์ใช้ในกระบวนการรู้จำได้ ยกตัวอย่างเช่น 0 หากเขียนอย่างถูกต้องแล้วจะไม่มีจุดปลายอยู่เลย ในขณะที่ ก จะมีจุดปลาย อยู่ 2 จุดทางด้านล่างซ้ายและขวาของภาพตัวอักษร เป็นต้น จะเห็นว่าเมื่อเราทราบตำแหน่งและจำนวนของจุดปลายของภาพอักษร ข้อมูลดังกล่าวนี้สามารถนำมาใช้ในการจัดกลุ่มตัวอักษรได้เป็นอย่างดีโดยเฉพาะกับตัวอักษรภาษาไทย

การหาจุดปลายของภาพตัวอักษรสามารถกระทำได้ไม่ยาก หากแต่จะต้องนำภาพอักษรไปผ่านกระบวนการทำให้ภาพบางดังที่ได้อธิบายไว้ในบทที่ 3 เมื่อได้ภาพที่บางแล้วก็นำไปทาบกับหน้าต่าง (Template) ขนาด 3x3 ดังในรูปที่ 4.5 ทีละจุด หากมีจุดใดในภาพที่ตรงกันกับหน้าต่างใดหน้าต่างหนึ่ง จุดนั้นก็จะถือว่าเป็นจุดปลาย

ในการเขียนโปรแกรมเพื่อทำงานดังกล่าวอาจจะอาศัยวิธีหาน้ำหนัก (weight) ของจุดภาพดำที่กำลังพิจารณา (P) ก็ได้ โดยน้ำหนักที่ว่านี้จะได้จากผลบวกของจุดรอบข้างทั้ง 8 ตำแหน่ง (n0-n7) ของจุดที่กำลังพิจารณา แล้วนำน้ำหนักของจุดที่พิจารณานั้นมาเปรียบเทียบกับจุดปลายทั้ง 8 แบบ ซึ่งหาน้ำหนักไว้ก่อนแล้ว ดูรูปที่ 4.6 ประกอบ สังเกตว่านำหนักที่ให้กับจุดรอบข้างจะมีค่าเป็น 2^i สำหรับตำแหน่งของ n_i หากพบว่าผลบวกของจุดรอบข้างมีค่าเท่ากับน้ำหนักของจุดปลายแบบใดแบบหนึ่งทั้ง 8 แบบ ก็จะกล่าวว่าจุดที่กำลังพิจารณานั้นเป็นจุดปลาย โดยรวมแล้วการหาจุดปลายมีขั้นตอนต่างๆดังแสดงในรูปที่ 4.7



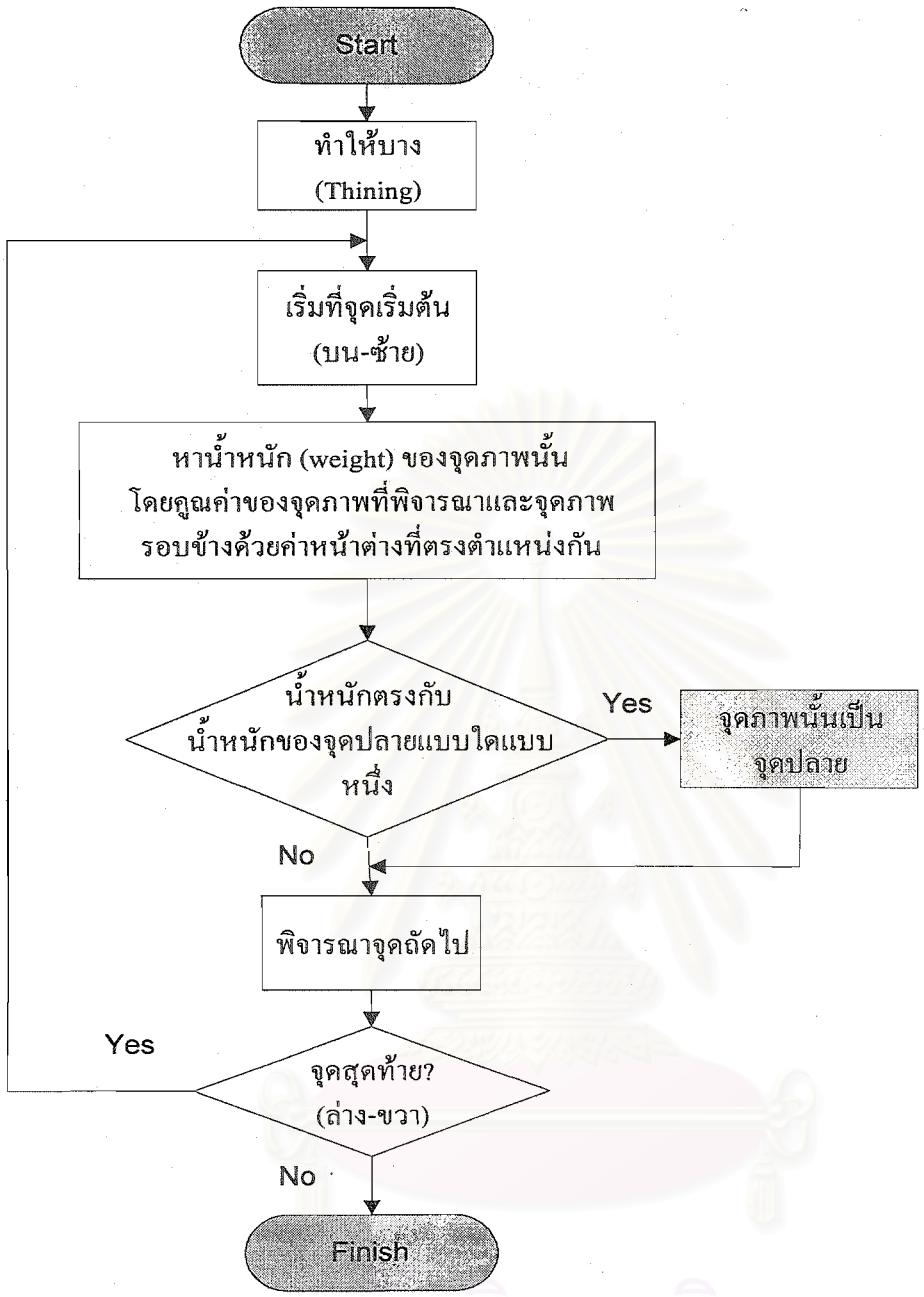
รูปที่ 4.5 หน้าต่างที่ใช้ในการเปรียบเทียบเพื่อหาจุดปลายเมื่อพิจารณาจากจุดรอบข้าง

n3	n2	n1
n4	P	n0
n5	n6	n7

8	4	2
16	P	1
32	64	128

รูปที่ 4.6 แสดงหน้าต่างที่ใช้หาจุดปลาย และน้ำหนักของจุดรอบข้าง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



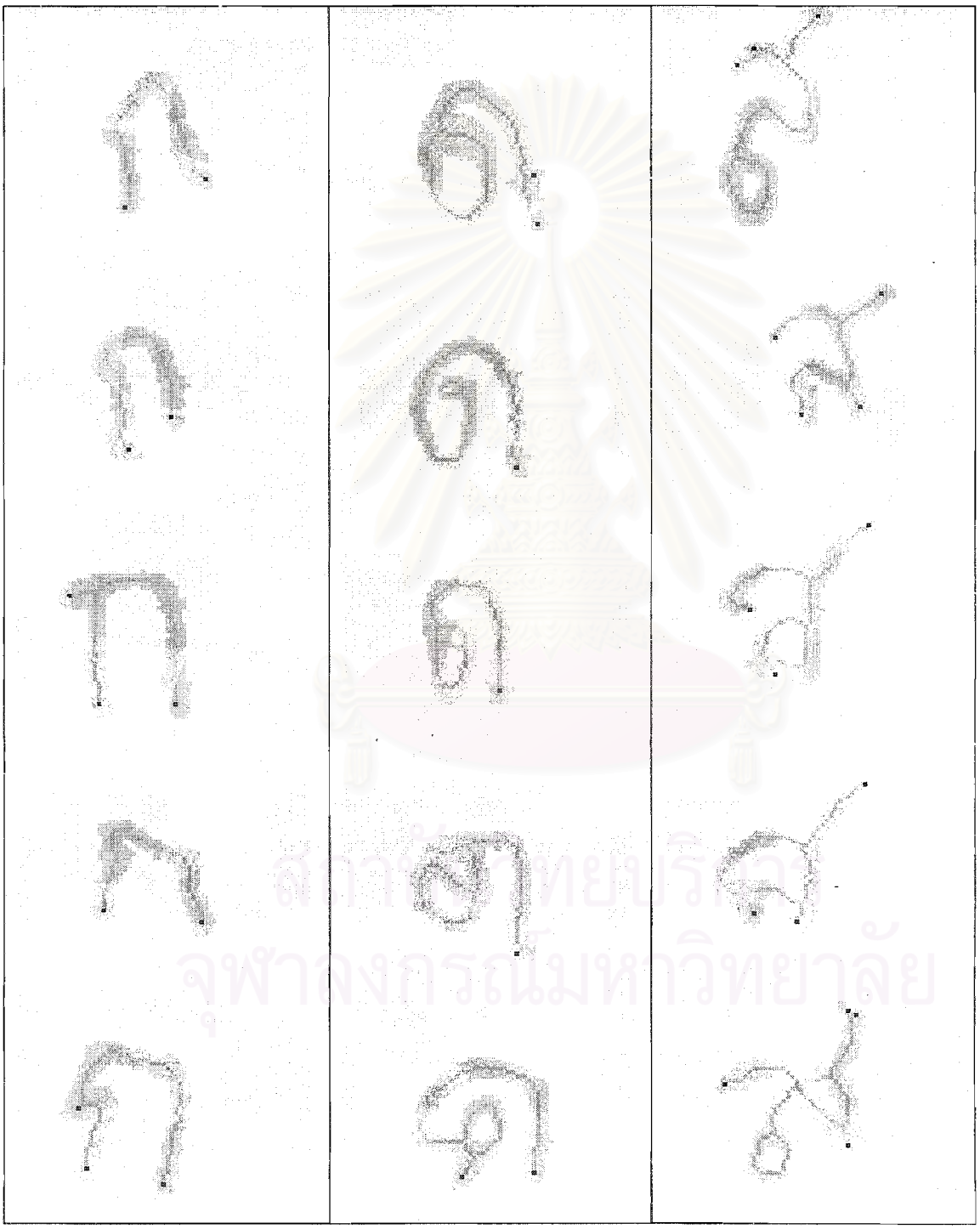
รูปที่ 4.7 *ขั้นตอนการหาจุดปลายของภาพ*

ผลการทดสอบและวิเคราะห์ผล

รูปที่ 4.8 แสดงตัวอย่างภาพของตัวอักษรและจุดปลายที่หาได้ โดยจุดปลายคือจุดที่มีสีเข้มที่สุด จุดสีเข้มรองลงมาคือโครงร่างที่ได้จากการทำให้บาง และจุดที่จางสุดคือจุดภาพเดิม คุณภาพหรือความถูกต้องของจุดปลายที่หาได้จะขึ้นอยู่กับคุณภาพของการทำให้บาง ซึ่งหากวิธีการทำให้บางก่อให้เกิดแง่งที่ไม่ต้องการแล้ว จะทำให้ได้จุดปลายที่หาได้มิได้เป็นจุดปลายของตัวอักษรที่แท้จริง เช่น อักษร ค ตัวแรกของรูปที่ 4.8 จะมีจุดปลายที่ตำแหน่งตรงกลางด้านขวาของภาพด้วย ซึ่งโดยปกติแล้วอักษร ด จะไม่มีจุดปลายอยู่ที่ตำแหน่งดังกล่าว จะเห็นว่าการที่เราพบจุดปลายในตำแหน่งที่ผิดจะส่งผลกระทบโดยตรงต่อการตัดสินใจในกระบวนการแยกแยะตัวอักษรในขั้นต่อ

ไป ดังนั้นจึงมีความจำเป็นที่จะต้องขจัดหรือลดผลกระทบของปัญหาดังกล่าวให้น้อยที่สุด ซึ่งอาจจะกระทำได้ในขั้นตอนของการทำให้ตัวอักษรบาง หรือพัฒนากระบวนการแยกแยะตัวอักษรที่มีขีดความสามารถที่สูงขึ้น

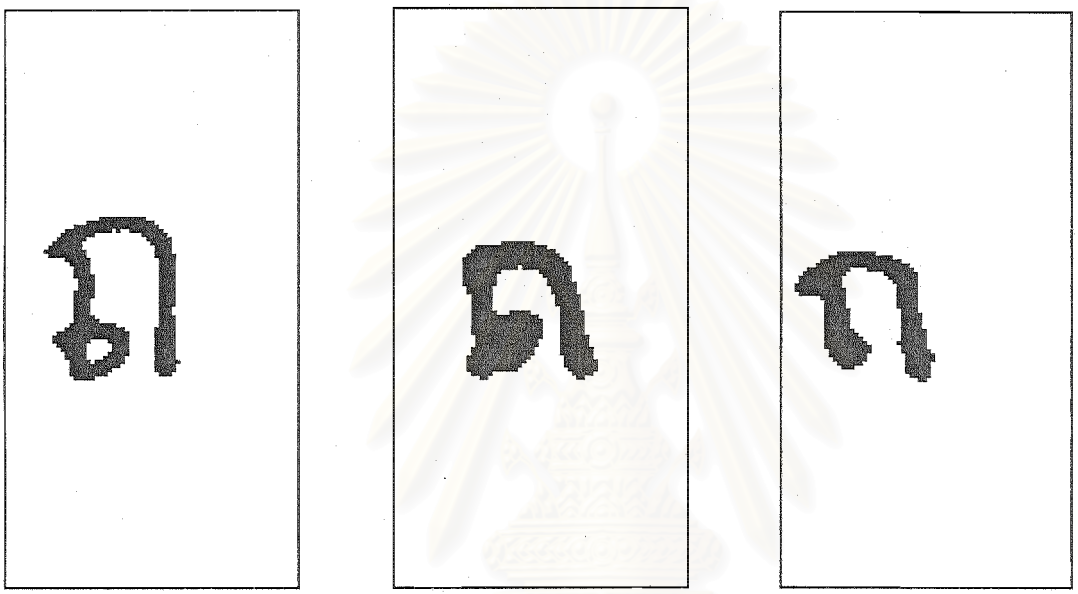
โดยสรุปแล้วตำแหน่งของจุดปลายทำให้เราสามารถแยกแยะตัวอักษรได้เป็นกลุ่มอย่างคร่าว เช่น ก, ด, ค จะไม่มีจุดปลายในส่วนขวาบน ส่วนกลุ่มอักษร ส, ฮ, ช จะมีจุดปลายในส่วนขวาบน ดังนั้นจึงสามารถในไปประยุกต์ใช้ในระบบรู้จำได้เป็นอย่างดี



รูปที่ 4.8 *แสดงภาพตัวอักษรตัวอย่างและจุดปลายที่หาได้ (ใช้การทำให้บางวิธีที่ 2 ที่อธิบายในบทที่ 3)*

4.3 การหาหัว (Head)

ลักษณะสำคัญอีกอย่างหนึ่งของภาษาไทยที่แตกต่างจากภาษาอื่น คือ หัวของตัวอักษร โดยตัวอักษรบางส่วนจะมีหัวแต่บางส่วนจะไม่มีหัว นอกจากนั้นจำนวนและตำแหน่งของหัวยังแตกต่างกันไปในแต่ละตัวอักษรอีกด้วย ลักษณะที่แตกต่างนี้มีประโยชน์อย่างมากในการนำมาใช้ในการแยกแยะกลุ่มตัวอักษรภาษา จากการศึกษาภาพตัวอักษรลายมือเขียนในฐานข้อมูลที่เก็บขึ้น พบว่าโดยทั่วไปแล้วภาพของตัวอักษรที่มีหัวมีอยู่ 3 รูปแบบ คือ แบบเห็นหัวชัดเจนมีรูกลวงภายในหัว แบบเห็นหัวชัดเจนแต่ไม่มีรูกลวงภายใน และแบบหัวไม่ชัดเจน ซึ่งตัวอย่างของทั้ง 3 แบบได้แสดงไว้ดังรูปที่ 4.9



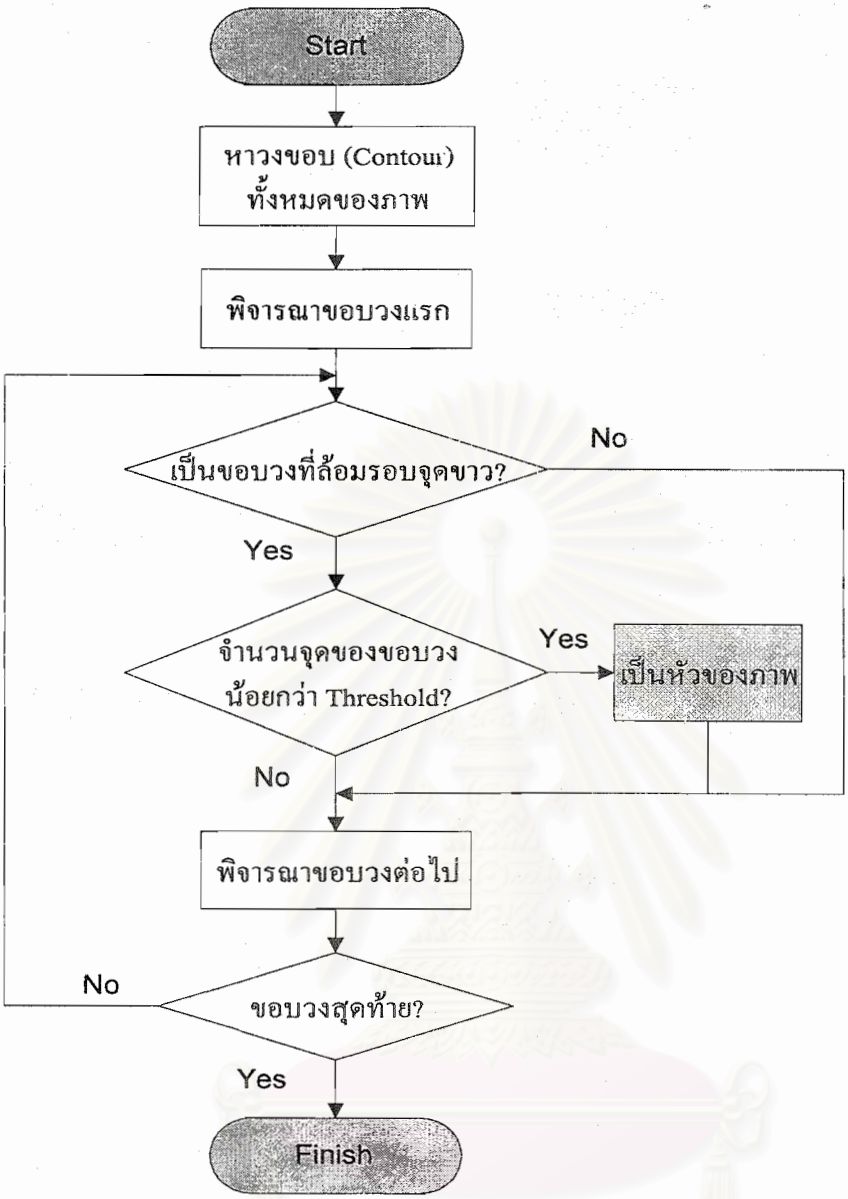
หัวชัดเจนมีรูตรงกลาง หัวชัดเจนไม่มีรูตรง หัวไม่ชัดเจน

รูปที่ 4.9 ภาพของอักษรที่มีหัวทั้ง 3 แบบ

สำหรับกรรมวิธีในการหาหัวที่ได้ทำการศึกษาจะใช้ได้กับภาพตัวอักษรที่มีหัวในลักษณะเป็นแบบที่ 1 เท่านั้น (แบบเห็นหัวชัดเจนมีรูตรงกลาง) โดยเทคนิคในการหาหัวมีขั้นตอนดังแสดงในรูปที่ 4.10

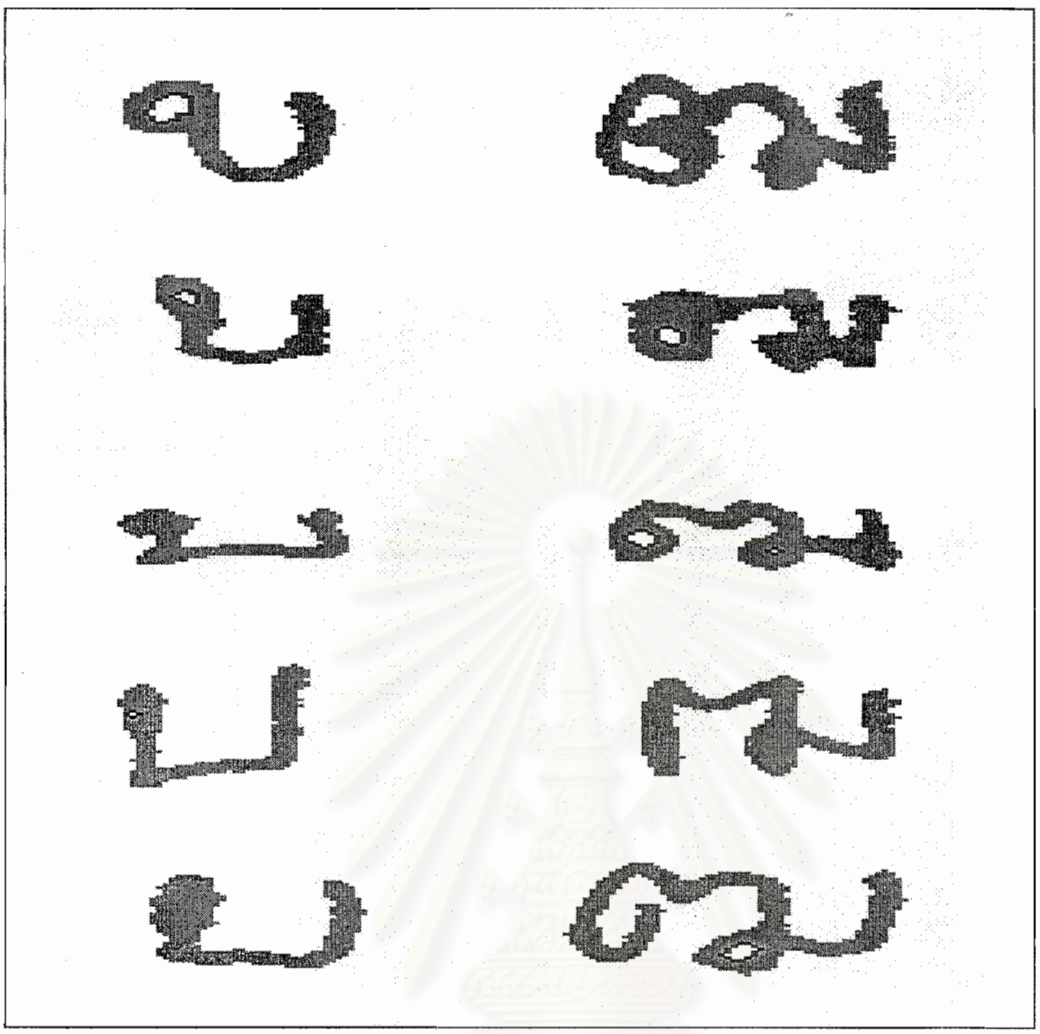
ผลการทดสอบและวิเคราะห์ผล

พิจารณาตัวอย่างของผลที่ได้จากการหาหัวในรูปที่ 4.11 สังเกตว่าวิธีการหาหัวที่เสนอไว้จะหาได้เฉพาะภาพตัวอักษรที่มีหัวแบบที่ 1 (แบบที่มีหัวชัดเจนและมีรูตรงกลาง) เท่านั้น ซึ่งปัญหาก็คือในภาพตัวอักษรในฐานข้อมูลจะมีลักษณะที่มีหัวแบบที่ 1 ไม่มากนัก ดังนั้นแสดงว่าหากไม่พบหัวแบบที่ 1 ก็ยังไม่สามารถระบุได้ว่าตัวอักษรนั้นไม่มีหัว จำเป็นที่จะต้องหาวิธีการหาหัวแบบที่ 2 และ 3 ต่อไป ทั้งนี้จากประสบการณ์งานวิจัยพบว่า แนวทางในการหาหัวประเภทที่เหลือมิใช่เรื่องที่ง่ายเท่าใดนัก



รูปที่ 4.10 *ขั้นตอนการหาหัวของภาพ*

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.11 *แสดงภาพตัวอักษรตัวอย่างและหัวที่หาได้*

4.4 การหาจำนวนการเปลี่ยนจากจุดขาวเป็นจุดดำ (Stoke Changing)

คุณลักษณะที่บ่งถึงความแตกต่างของภาพตัวอักษรอีกแบบหนึ่งคือ วิธีการหาจำนวนการเปลี่ยนจากจุดขาวเป็นจุดดำทางแนวนอน หรือแนวตั้ง แนวใดแนวหนึ่ง โดยจุดที่เปลี่ยนจากขาวเป็นดำในแนวนอนและแนวตั้งได้แสดงดังรูปที่ 4.12 ในการหาจำนวนการเปลี่ยนจากจุดขาวเป็นจุดดำจะต้องกระทำเหมือนๆกันในทุกๆแถวหรือในทุกๆคอลัมน์ ดูตัวอย่างผลการคำนวณหาจำนวนการเปลี่ยนจากจุดขาวเป็นดำได้ในรูปที่ 4.13 จำนวนการเปลี่ยนจากจุดขาวเป็นจุดดำสามารถนำมาใช้เป็นลักษณะบ่งถึงความแตกต่างได้เหมือนกัน

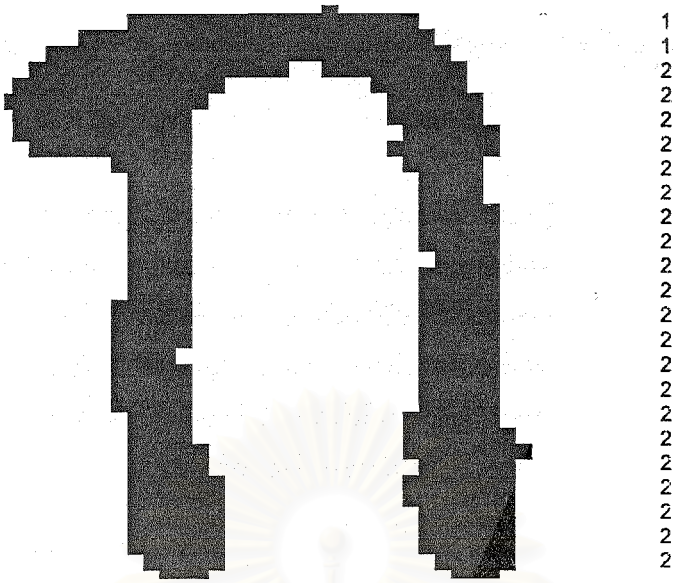
x	x	x
x	0	1
x	x	x

จุดที่เปลี่ยนจากขาวเป็นดำในแนวนอน

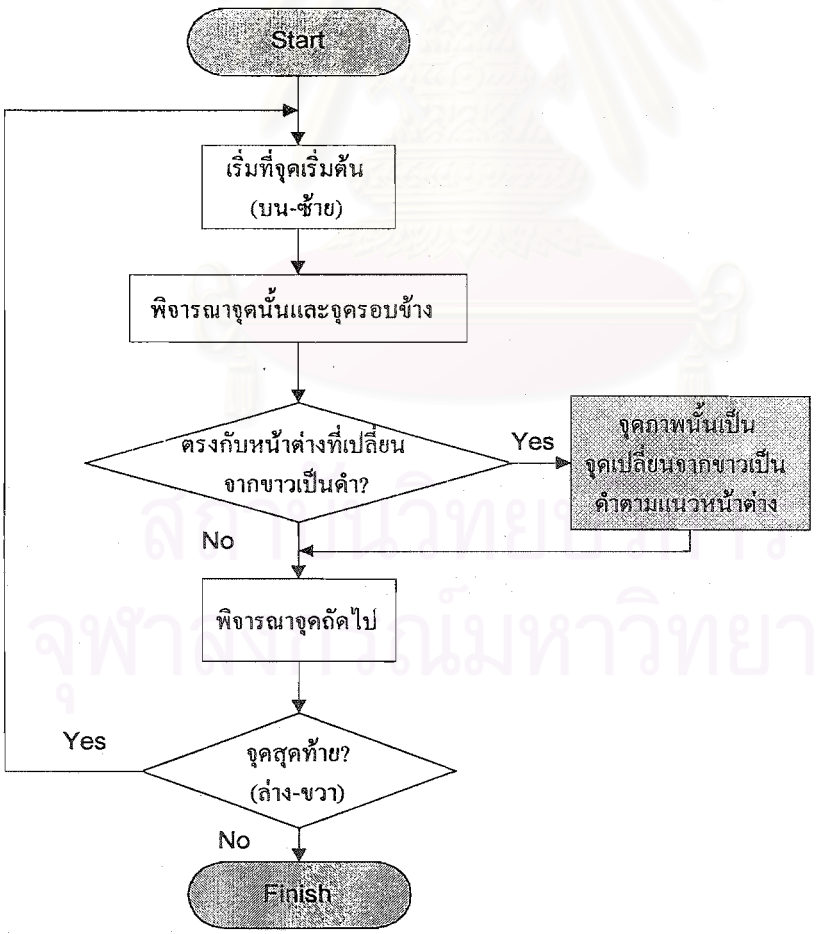
x	x	x
x	0	x
x	1	x

จุดที่เปลี่ยนจากขาวเป็นดำในแนวตั้ง

รูปที่ 4.12 *จุดที่เปลี่ยนจากขาวเป็นดำ (x เป็นดำหรือขาวก็ได้)*



รูปที่ 4.13 *ภาพของตัวอักษรตัวอย่างพร้อมจำนวนการเปลี่ยนจากจุดขาวเป็นดำ*



รูปที่ 4.14 *ขั้นตอนการหาจำนวนการเปลี่ยนจากจุดขาวเป็นดำ*

5. แนวทางการออกแบบระบบรู้จำตัวอักษรพยัญชนะลายมือเขียนภาษาไทยเบื้องต้น

5.1 บทนำ

ในบทนี้กล่าวถึงแนวทางการออกแบบและการทดสอบเบื้องต้นของระบบรู้จำตัวอักษรพยัญชนะลายมือเขียนภาษาไทยจำนวน 44 ตัว คือตัวอักษร ก ถึง ฮ โดยอาศัยวิธีการดึงคุณสมบัติเฉพาะที่บ่งถึงความแตกต่างของตัวอักษร เช่น จำนวนของหัว ตำแหน่งของหัว ลักษณะของหัวคือหัวเข้าหรือหัวออก และจำนวนการเปลี่ยนจากจุดขาวไปเป็นจุดดำ เป็นต้น โดยในส่วนแรกของบทนี้จะกล่าวถึงรายละเอียดของวิธีการแบ่งกลุ่มตัวอักษรพยัญชนะไทยทั้ง 44 ตัวโดยอาศัยลักษณะบ่งต่างเหล่านี้ จากนั้นก็จะเป็นส่วนของการทดสอบประสิทธิภาพและการวิเคราะห์ผลของแต่ละวิธีโดยได้ทำการทดสอบกับภาพตัวอักษรในฐานข้อมูลที่สร้างขึ้น

5.2 แนวทางการแบ่งกลุ่มอักษรพยัญชนะไทย

5.2.1 การแบ่งกลุ่มตัวอักษรตามจำนวนเกาะของตัวอักษร

การแบ่งกลุ่มตัวอักษรประเภทแรกที่จะกล่าวถึงคือการแบ่งตามจำนวนของเกาะที่มีอยู่ในภาพตัวอักษร เกาะในที่นี้นิยามให้เป็นส่วนของภาพตัวอักษรที่อยู่อย่างโดดเดี่ยว ไม่เชื่อมต่อกับส่วนอื่น ๆ ของภาพ สำหรับในกรณีของตัวอักษรพยัญชนะภาษาไทยสามารถแบ่งออกได้เป็น 2 กลุ่มใหญ่คือ กลุ่มที่มีเพียง 1 เกาะ และกลุ่มที่มี 2 เกาะ โดยสมาชิกของแต่ละกลุ่มมีรายละเอียดดังที่แสดงในตารางที่ 5.1

ตารางที่ 5.1 *การจำแนกกลุ่มตัวอักษรตามจำนวนเกาะของตัวอักษร*

จำนวนเกาะของตัวอักษร	อักษรในกลุ่ม	จำนวน(ตัวอักษร)
1 เกาะ	ตัวอักษรพยัญชนะทุกตัว ยกเว้น ญ และ ฐ	42
2 เกาะ	ญ ฐ	2
	รวมทั้งสิ้น	44

5.2.2 การแบ่งกลุ่มตัวอักษรโดยอาศัยจำนวนหัวของตัวอักษร

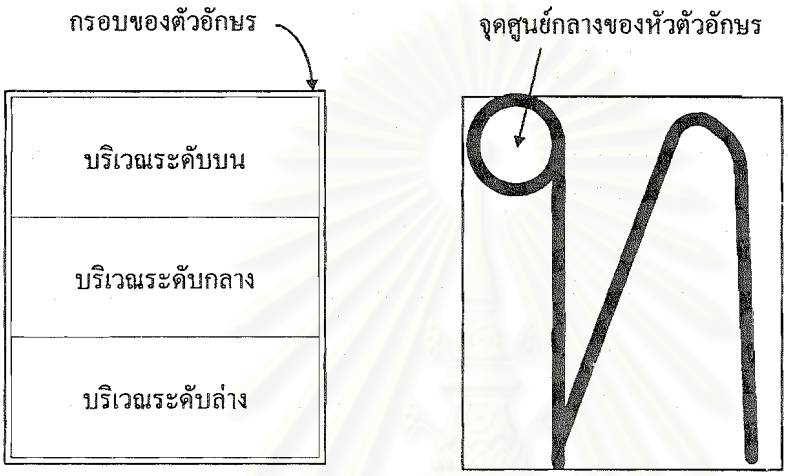
การแบ่งกลุ่มตัวอักษรอีกวิธีหนึ่งที่มีความเหมาะสมโดยเฉพาะกับตัวอักษรภาษาไทยก็คือการนับจำนวนหัวของตัวอักษร ซึ่งเราสามารถแบ่งออกได้เป็น 4 กลุ่มใหญ่ คือ กลุ่มที่ไม่มีหัวเลย เช่น ก กลุ่มที่มีหัว 1 หัว เช่น ข กลุ่มที่มี 2 หัว เช่น น และกลุ่มที่มี 3 หัว คือ ฐ สำหรับกรรมวิธีในการหาหัวนั้นสามารถดูได้จากบทที่ 4 ส่วนรายละเอียดของการแบ่งทั้งหมดได้แสดงไว้ในตารางที่ 5.2

ตารางที่ 5.2 *การจำแนกกลุ่มตัวอักษรโดยอาศัยจำนวนหัวของตัวอักษร*

จำนวนหัวของตัวอักษร	ตัวอักษร	จำนวน
ไม่มีหัว	ก ธ	2
1 หัว	ข ฃ ค ฅ ง จ ช ซ ฌ ด ต ถ ท บ ป ผ ฝ พ ฟ ภ ย ร ล ว ศ ส อ	27
2 หัว	ม ฉ ฌ ญ ฎ ฏ ฒ ณ น ม ษ ห ฬ ฮ	14
3 หัว	ฐ	1
	รวมทั้งสิ้น	44

5.2.3 การแบ่งกลุ่มตัวอักษรโดยพิจารณาจากระดับของหัวตัวอักษร

สำหรับงานวิจัยนี้ได้แบ่งระดับหัวของตัวอักษรออกเป็นทั้งหมด 3 ระดับ คือ บน กลาง และล่าง ดังที่แสดงในรูปที่ 5.1 ในการพิจารณาว่าหัวของตัวอักษรอยู่ในระดับใดนั้น ขั้นแรกจะต้องทำการหาขอบเขตหรือกรอบของภาพตัวอักษรก่อน และทำการแบ่งความสูงของภาพออกเป็น 3 ส่วนเท่า ๆ กัน จากนั้นพิจารณาดูว่าหัวตัวอักษรตกอยู่ในบริเวณใด ทั้งนี้ให้ใช้ตำแหน่งจุดศูนย์กลางของหัวตัวอักษรเป็นเครื่องชี้วัดระดับ โดยหากจุดศูนย์กลางหัวของตัวอักษรตกอยู่ในบริเวณพื้นที่ของระดับใดก็จะระบุว่าตัวอักษรนั้นมีหัวอยู่ในระดับดังกล่าว



รูปที่ 5.1 *การกำหนดบริเวณระดับของหัวตัวอักษร*

ในกรณีที่ตัวอักษรมีจำนวนหัวมากกว่าหนึ่งหัว จำเป็นจะต้องมีการให้ลำดับความสำคัญของหัวตัวอักษรเพื่อความสะดวกในการจำแนกกลุ่มตัวอักษร ในงานวิจัยนี้ได้กำหนดลำดับความสำคัญไว้ดังนี้

- กำหนดให้ระดับบนมีความสำคัญมากที่สุด ส่วนระดับกลางมีความสำคัญรองลงมา และระดับล่างมีความสำคัญน้อยที่สุด
- ภายในระดับเดียวเดียวกันก็ยังมีการแบ่งลำดับความสำคัญ โดยกำหนดให้หัวของตัวอักษรที่อยู่ใกล้ขอบด้านซ้ายของภาพมากกว่ามีลำดับความสำคัญที่สูงกว่าหัวที่อยู่ห่างออกไป

จากกฎเกณฑ์การจัดลำดับความสำคัญนี้ จะสามารถระบุว่าตัวอักษร ห มี 2 หัว โดยหัวที่หนึ่งอยู่ระดับบน และหัวที่สองก็อยู่ระดับบนด้วย ในขณะที่ตัวอักษร น มีหัวที่หนึ่งอยู่ระดับบน และหัวที่สองอยู่ระดับล่าง เป็นต้น สำหรับตัวอักษรที่ไม่มีหัวจะไม่ถูกพิจารณาเรื่องระดับของหัวตัวอักษร เมื่อนำกฎเกณฑ์นี้มาใช้กับตัวอักษรพยัญชนะไทยจะสามารถจำแนกกลุ่มของตัวอักษรออกได้ดังนี้

- ตัวอักษรที่มีเพียง 1 หัว สามารถแบ่งออกได้เป็น 3 กลุ่ม คือ กลุ่มที่มีหัวอยู่ในระดับบน กลาง และล่าง
- ตัวอักษรที่มี 2 หัว สามารถแบ่งออกได้เป็น 5 กลุ่ม โดยพิจารณาลำดับความสำคัญจากหัวที่หนึ่งและสองตามลำดับดังนี้ (หัวที่หนึ่ง, หัวที่สอง) ได้แก่ ระดับบน,บน ระดับบน,กลาง ระดับบน,ล่าง ระดับกลาง,ล่าง และระดับล่าง,ล่าง

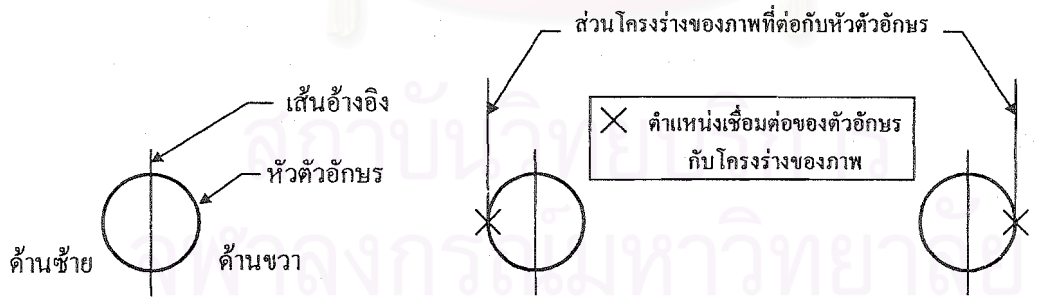
ตารางที่ 5.3 การจำแนกกลุ่มตัวอักษรโดยอาศัยระดับของหัวตัวอักษร

จำนวนหัว	ตำแหน่งหัวที่ 1	ตำแหน่งหัวที่ 2	ตัวอักษร	จำนวน
1	บน	-	ข ฃ ง ฑ ท บ ผ พ ย	9
	กลาง	-	ค ฅ จ ด ต ศ อ ช ซ ป ฝ ฟ	12
	ล่าง	-	ฉ ภ ร ล ว ส	6
2	บน	บน	ห	1
	บน	กลาง	ษ ฬ ฮ	3
	บน	ล่าง	ฆ น ม	3
	กลาง	ล่าง	ฉ ญ ฎ ฏ ฒ	5
	ล่าง	ล่าง	ฌ ณ	2
			รวมทั้งสิ้น	41

จากตารางที่ 5.3 มีข้อสังเกตที่น่าสนใจอย่างหนึ่งคือตัวอักษร ห ซึ่งอยู่ในกลุ่มตัวอักษรที่มีจำนวนหัวเท่ากับ 2 เมื่อพิจารณาระดับของหัวตัวอักษรเพียงอย่างเดียวก็สามารถที่จะแยกตัวอักษร ห ออกมาได้ ดังนั้นโดยหลักการแล้วตัวอักษร ห จึงไม่จำเป็นต้องนำไปพิจารณาลักษณะบ่งความต่างชนิดอื่นต่อไป

5.2.4 การแบ่งกลุ่มตัวอักษรโดยพิจารณาจากลักษณะของหัวอักษร

ลักษณะของหัวตัวอักษรภาษาไทยสามารถแบ่งอย่างคร่าว ๆ ได้เป็น 2 รูปแบบตามตำแหน่งของจุดเชื่อมต่อระหว่างหัวกับโครงร่างของตัวอักษร ดูรูปที่ 5.2 ประกอบ กำหนดให้เส้นอ้างอิงคือเส้นที่ตัดผ่านจุดศูนย์กลางของหัวตัวอักษร จากนั้นให้พิจารณาว่าโครงร่างของตัวอักษรเชื่อมต่อกับหัวของตัวอักษรทางด้านใด (ซ้ายหรือขวา) เมื่อเทียบกับเส้นอ้างอิง



รูปที่ 5.2 การกำหนดการต่อเชื่อมทางด้านซ้ายหรือขวาของหัวตัวอักษร

ในตารางที่ 5.4 ได้จำแนกตัวอักษรโดยอาศัยการต่อเชื่อมของหัวตัวอักษร ตามจำนวนหัวและระดับของหัวตัวอักษรที่ได้จำแนกไว้ในหัวข้อย่อยก่อนหน้านี้แล้ว

ตารางที่ 5.4 *การจำแนกกลุ่มตัวอักษรโดยดูจากการต่อเชื่อมของหัวตัวอักษร*

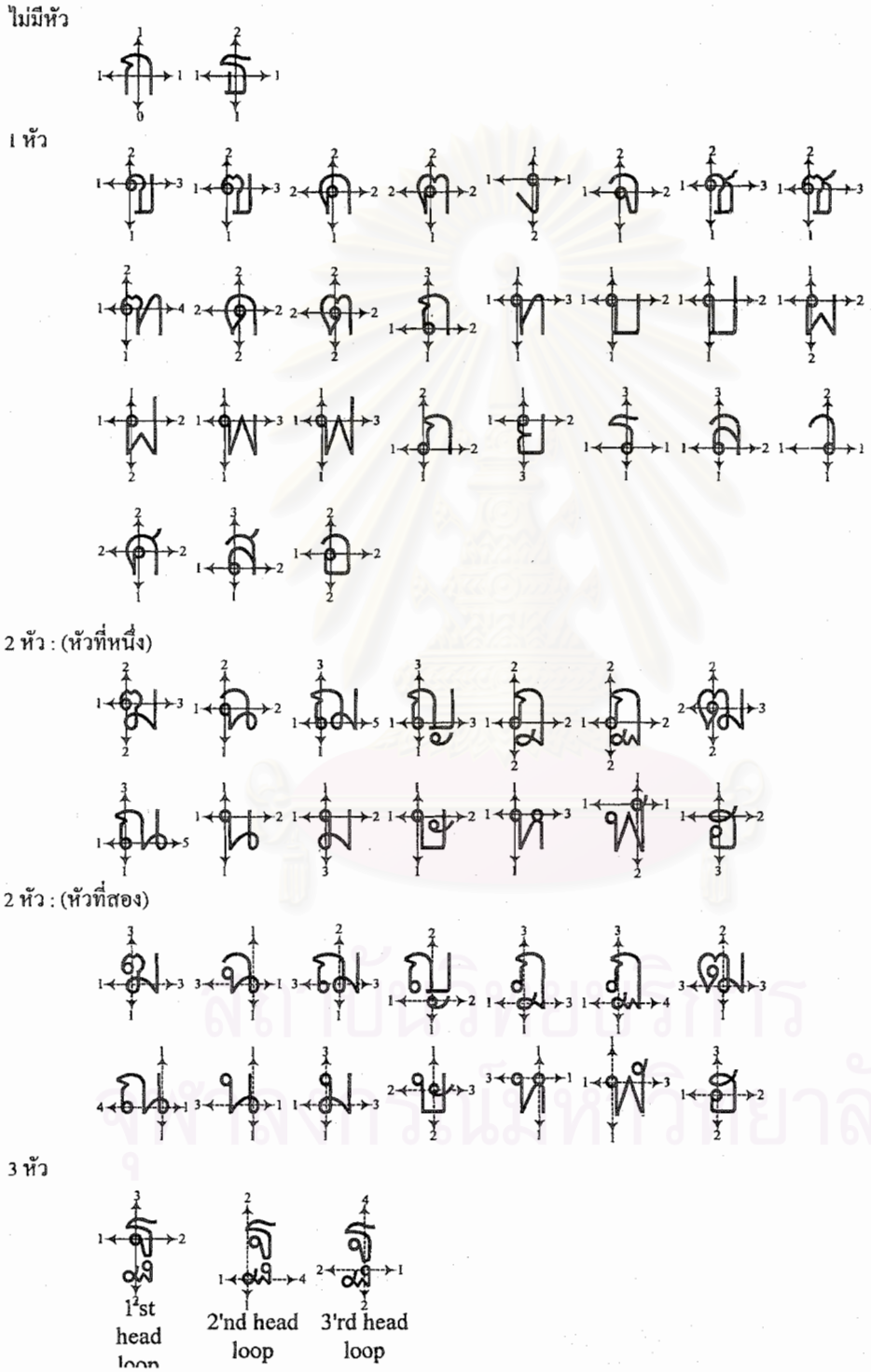
จำนวนหัว	ระดับ	ตำแหน่งหัว		ตัวอักษร	จำนวน
		หัวที่ 1	หัวที่ 2		
1	บน	ซ้าย	-	ฃ ฅ ผ ย	4
		ขวา	-	ข ง ท บ พ	5
	กลาง	ซ้าย	-	ค ฅ อ ซ ฝ ศ	6
		ขวา	-	จ ด ต ช ป ฟ	6
	ล่าง	ซ้าย	-	ถ ล ส	3
		ขวา	-	ภ ร ว	3
2	บน,กลาง	ขวา	ซ้าย	ฮ	1
		ขวา	ขวา	ษ พ	2
	บน,ล่าง	ซ้าย	ขวา	ฆ	1
		ขวา	ซ้าย	ม	1
		ขวา	ขวา	น	1
	กลาง,ล่าง	ซ้าย	ขวา	ญ	1
		ขวา	ซ้าย	ฉ	1
		ขวา	ขวา	ฌ ฏ ฒ	3
	ล่าง,ล่าง	ซ้าย	ซ้าย	ณ	1
		ซ้าย	ขวา	ฌ	1
				รวมทั้งสิ้น	40

แนวทางการแบ่งกลุ่มตัวอักษรตามคุณลักษณะบ่งความต่างที่กล่าวมาทั้งหมดนี้น่าจะเป็นประโยชน์ในการพิจารณาแบ่งกลุ่มตัวอักษรพยัญชนะไทยทั้ง 44 ตัวออกเป็นกลุ่มย่อยได้ดีระดับหนึ่ง หากแต่ว่าการพิจารณาเพียงคุณลักษณะเท่าที่กล่าวมาข้างต้นยังไม่เพียงพอที่จะแยกกลุ่มตัวอักษรทั้งหมดออกจากกันได้ ดังนั้นจึงจำเป็นต้องมีกรรมวิธีอื่นที่จะมาช่วยในการพิจารณาเพิ่มเติม ซึ่งในที่นี้ได้เลือกใช้คุณลักษณะของการเปลี่ยนจากจุดดำเป็นจุดขาว

5.2.5 การจำแนกตัวอักษรไทยโดยอาศัยจำนวนการเปลี่ยนจากจุดขาวไปเป็นจุดดำ

จากการแบ่งกลุ่มย่อยโดยอาศัยกรรมวิธีต่าง ๆ ที่ได้กล่าวมาข้างต้น ขั้นต่อไปก็คือการแยกตัวอักษรภายในกลุ่มย่อยแต่ละกลุ่มออกจากกันโดยอาศัยการหาจำนวนการเปลี่ยนของจุดขาวไปเป็นจุดดำ (ดูรายละเอียดวิธีการหาจำนวนการเปลี่ยนของจุดขาวไปเป็นจุดดำได้ในบทที่ 4) สำหรับแนวความคิดที่ใช้ในงานวิจัยนี้มีความแตกต่างจากการใช้งานที่ผ่านมา ตรงที่เราจะหาจำนวนการเปลี่ยนของจุดขาวไปเป็นจุดดำเฉพาะตำแหน่งที่มีหัวเท่านั้น โดยจะพิจารณาทั้งในแนวตั้งและแนวนอนและมีจุดเริ่มต้นอยู่ที่จุดศูนย์กลางของหัวตัวอักษร นั่นคือหากตัวอักษรมีเพียงหัวเดียวก็จะมีการหาจำนวนการเปลี่ยนจากจุดขาวไปเป็นจุดดำเพียง 4 ครั้งคือ ขึ้น ลง ซ้าย และ ขวา สำหรับกรณีที่ตัวอักษรไม่มีหัวก็จะอาศัยจุดกึ่งกลางของภาพเป็นจุดเริ่มต้นแทนการใช้จุดกึ่งกลางของหัว สังเกตว่าในการ

หาจุดกึ่งกลางของภาพจะต้องนำภาพตัวอักษรไปผ่านกระบวนการหากรอบของภาพก่อนซึ่งได้อธิบายไว้ในบทที่ 3 แล้ว สำหรับรายละเอียดการหาจำนวนการเปลี่ยนจากจุดขาวไปเป็นจุดดำของตัวอักษรแต่ละตัวได้แสดงไว้ใน รูปที่ 5.3 จุดเด่นของวิธีนี้อยู่ตรงที่สามารถช่วยลดปริมาณการคำนวณหาจำนวนจุดการเปลี่ยนจากขาวเป็นดำลงอย่างมาก เพราะคำนวณเพียง 4 ทิศทางต่อหนึ่งหัวเท่านั้น



รูปที่ 5.3 *รายละเอียดการหาจำนวนการเปลี่ยนจากจุดขาวไปเป็นจุดดำ*

5.3 การทดสอบและการวิเคราะห์ผล

สำหรับการทดสอบเพื่อหาประสิทธิภาพของวิธีการที่กล่าวมานี้ ได้ทำการทดสอบกับภาพอักษรในฐานข้อมูลที่สร้างขึ้นจำนวนทั้งสิ้น 100 ชุด รวมตัวอักษรทั้งสิ้น 4,400 ตัว ตัวอักษรทั้งหมดที่เลือกใช้ในการทดสอบเป็นตัวอักษรที่มีหัวชัดเจนอ่านได้ง่าย การทดสอบภายใต้เงื่อนไขดังกล่าว หมายความว่าผู้เขียนจะต้องควบคุมการเขียนให้ได้ตัวอักษรที่มีหัวชันเจน การทดสอบภาพตัวอักษรได้กระทำทีละตัว โดยนำภาพแต่ละภาพมาหาลักษณะบ่งความแตกต่างแต่ละชนิดที่กล่าวมาข้างต้นทั้งหมด และทำการการทดสอบวิธีการจำแนกแต่ละวิธีดังจะได้กล่าวต่อในส่วนต่อไป

5.3.1 ผลการทดสอบจากจำนวนเกาะของตัวอักษร

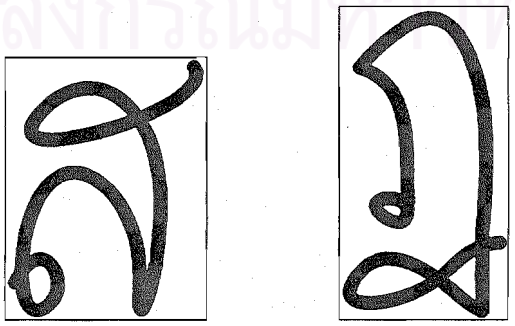
จากผลการทดสอบพบว่าวิธีการนี้สามารถแยกแยะตัวอักษรทั้ง 2 กลุ่มได้ถูกต้อง 100% นั่นคือสามารถแยกได้ว่าตัวอักษร ญ และ ฐ อยู่ในกลุ่มที่มี 2 เกาะ และตัวอักษรที่เหลืออยู่ในกลุ่ม 1 เกาะ ดังนั้นหากพบว่าภาพอักษรที่ทดสอบมี 2 เกาะ จะทราบได้ทันทีเลยว่าภาพตัวอักษรนี้จะเป็นได้ 2 แบบคือ ญ และ ฐ จากนั้นก็อาศัยวิธีการอื่น ๆ เพื่อทำการแยกต่อไปว่าเป็นตัวอักษรใด

5.3.2 ผลการทดสอบจากจำนวนหัวของตัวอักษร

จากผลการทดสอบในตารางที่ 5.5 พบว่าวิธีการนี้ไม่สามารถแยกแยะตัวอักษรทั้ง 4 กลุ่มได้ถูกต้อง 100% ยกตัวอย่างเช่น ตัวอักษร ส ซึ่งอยู่ในกลุ่มของตัวอักษรที่มีหัวเดียว แต่เนื่องจากลายมือเขียนของคนบางคนเป็นดังในรูปที่ 5.4 ซึ่งทำให้ภาพตัวอักษรกลายเป็นภาพที่มี 2 หัว ส่งผลให้การตัดสินใจผิดพลาดได้ หรือตัวอักษร ฎ ที่มีเพียง 2 หัว แต่ภาพตัวอักษร ฎ บางภาพอาจจะมี 3 หัวได้ ดังที่แสดงในรูปที่ 5.4 สำหรับกรณีตัวอักษรที่มี 3 หัวซึ่งมีเพียงตัวอักษรเดียวคือ ฐ ซึ่งจากการทดสอบพบว่าวิธีนี้สามารถแยกตัวอักษร ฐ ได้ถูกต้องพียง 91% เท่านั้น

ตารางที่ 5.5 ผลการทดสอบโดยอาศัยจำนวนหัวของตัวอักษร

จำนวนหัวของตัวอักษร	ตัวอักษร	ความถูกต้อง (%)
ไม่มีหัว	ก ธ	98%
1 หัว	ข ฃ ค ฅ ง จ ช ซ ฌ ด ต ถ ท บ ป ผ ฝ พ ฟ ภ ย ร ล ว ศ ส อ	96%
2 หัว	ฆ ฉ ฌ ญ ฎ ฏ ฒ ณ น ม ษ ห ฬ ฮ	97%
3 หัว	ฐ	91%



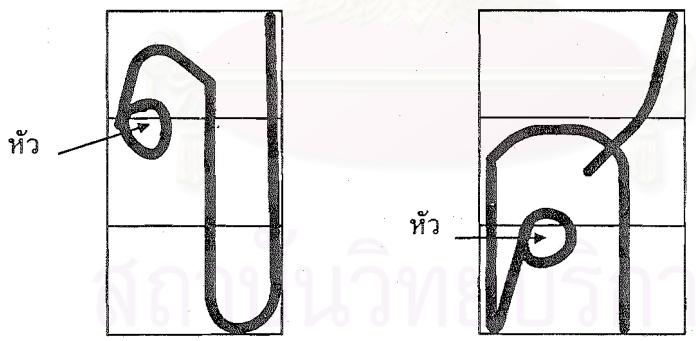
รูปที่ 5.4 ตัวอย่างภาพตัวอักษรที่ทำให้การตัดสินใจผิดพลาด

5.3.3 ผลการทดสอบจากระดับของหัวตัวอักษร

พิจารณาผลการทดสอบในตารางที่ 5.6 กรณีที่มีจำนวนหัวเพียง 1 หัว จะสังเกตเห็นว่าเปอร์เซนต์ความถูกต้องของกลุ่มที่มีหัวอยู่ในระดับกลางมีค่อนข้างต่ำกว่าระดับอื่น เมื่อทำการตรวจโดยละเอียดพบว่าเกิดจากสาเหตุ 2 ประการคือ หนึ่งมาจากการที่นับจำนวนหัวผิด และสองเนื่องมาจากการเขียนที่ทำให้ระดับของหัวมิได้อยู่ในระดับที่ควรจะเป็น ดังเช่นตัวอักษร ข ในรูปที่ 5.5 ที่ควรจะมีหัวอยู่ที่ระดับบน แต่จากรูปหัวอยู่ที่ระดับกลาง ส่งผลให้การตัดสินใจผิดพลาด อีกตัวอย่างหนึ่งก็คือตัวอักษร ศ ที่ควรจะมีหัวอยู่ที่ระดับกลาง แต่จากในรูปที่ 5.5 สังเกตว่าหัวของภาพตัวอักษรอยู่ที่ระดับล่าง ซึ่งทำให้การตัดสินใจผิดพลาด

ตารางที่ 5.6 ผลการทดสอบการจำแนกกลุ่มตัวอักษรโดยอาศัยระดับของหัวตัวอักษร

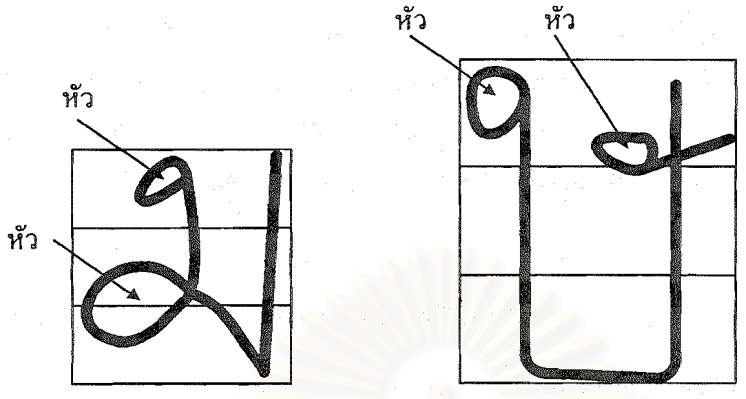
จำนวนหัว	ตำแหน่งหัวที่ 1	ตำแหน่งหัวที่ 2	ตัวอักษร	ความถูกต้อง (%)
1	บน	-	ข ฃ ง ฑ ท บ ผ พ ย	95%
	กลาง	-	ค ฅ จ ด ต ศ อ ช ซ ป ฝ ฟ	91%
	ล่าง	-	ถ ภ ร ล ว ส	98%
2	บน	บน	ห	99%
	บน	กลาง	ย พ ฮ	83%
	บน	ล่าง	ฆ น ม	90%
	กลาง	ล่าง	ฉ ญ ฎ ฏ ฒ	93%
	ล่าง	ล่าง	ฌ ณ	98%



รูปที่ 5.5 ตัวอย่างภาพตัวอักษรที่ทำให้การตัดสินใจผิดพลาดเนื่องจากหัวอยู่ในระดับที่ผิด

สำหรับกรณีที่ตัวอักษรมี 2 หัว เปอร์เซนต์ความถูกต้องของการรู้จำมีค่าค่อนข้างต่ำโดยเฉพาะเมื่อตัวอักษรมีหัวอยู่ในระดับกลาง สาเหตุก็เนื่องมาจากการที่นับจำนวนหัวผิดทำให้การตัดสินใจผิดพลาดไปตั้งแต่แรก และอีกสาเหตุหนึ่งก็มาจากการที่หัวตัวอักษรมิได้อยู่ในระดับที่ถูกต้อง ทำให้การเรียงตำแหน่งของหัวผิดพลาด ส่งผลให้การตัดสินใจผิดพลาด ดูตัวอย่างของภาพตัวอักษร ม และ ษ ในรูปที่ 5.6 ที่หัวอยู่ผิดระดับและทำให้การเรียงตำแหน่งหัวผิดไป เช่น ม ที่จัดอยู่ในกลุ่มที่มี 2 หัว และอยู่ตำแหน่ง บน ล่าง แต่จากในรูปที่ 5.6 ภาพนี้มีหัวอยู่

ที่ บน กลาง ส่วนตัวอักษร ษ ซึ่งจัดอยู่ในกลุ่ม บน กลาง แต่จากในรูปมีหัวอยู่ที่ บน บน ทั้งสองกรณีทำให้การตัดสินใจผิดพลาด



รูปที่ 5.6 ตัวอย่างภาพตัวอักษรที่ทำให้การตัดสินใจผิดพลาดเนื่องจากหัวอยู่ในระดับที่ผิด

5.3.4 ผลการทดสอบโดยพิจารณาจากลักษณะของหัวอักษร

ตารางที่ 5.7 ผลการทดสอบการจำแนกกลุ่มตัวอักษร โดยดูจากการต่อเชื่อมของหัวตัวอักษร

จำนวนหัว	ระดับ	ตำแหน่งหัว		ตัวอักษร	ความถูกต้อง (%)
		หัวที่ 1	หัวที่ 2		
1	บน	ซ้าย	-	ฃ ฑ ผ ย	80%
		ขวา	-	ข ง ท บ พ	95%
	กลาง	ซ้าย	-	ค ฅ อ ซ ฝ ศ	87%
		ขวา	-	จ ด ต ช ป ฟ	88%
	ล่าง	ซ้าย	-	ฉ ล ส	96%
		ขวา	-	ภ ร ว	99%
2	บน,กลาง	ขวา	ซ้าย	ฮ	86%
		ขวา	ขวา	ษ พ	80%
	บน,ล่าง	ซ้าย	ขวา	ฆ	50%
		ขวา	ซ้าย	ม	96%
		ขวา	ขวา	น	99%
	กลาง,ล่าง	ซ้าย	ขวา	ญ	75%
		ขวา	ซ้าย	ฉ	97%
		ขวา	ขวา	ฏ ฎ ฒ	90%
	ล่าง,ล่าง	ซ้าย	ซ้าย	ณ	98%
		ซ้าย	ขวา	ฌ	95%

จากผลการทดสอบของวิธีการพิจารณาจากลักษณะหัวของตัวอักษรในตารางที่ 5.7 จะเห็นได้ว่าความถูกต้องในการรู้จำมีเปอร์เซนต์ที่ต่ำลงกว่าเดิมเมื่อเทียบกับผลในตารางที่ 5.6 สำหรับตัวอักษรบางตัว เช่น ฆ จากเดิมที่มีค่าความถูกต้องในการรู้จำของกลุ่ม (ฆ น ม) ที่สูงถึง 90% พบว่ามีค่าลดลงเหลือเพียง 50% เมื่อทำการแยกเฉพาะตัว ฆ ออกมา และในขณะที่ตัวอักษรบางตัวเช่น น มีค่าความถูกต้องในการรู้จำที่สูงขึ้นเป็น 99% สาเหตุของการที่ความถูกต้องในการแยกตัวอักษร ฆ ต่ำมากก็เนื่องมาจากการที่หัวของตัวอักษรที่หนึ่งซึ่งควรจะอยู่ที่ระดับบน กลับพบว่ามีภาพอักษร ฆ จำนวนมากที่มีหัวที่หนึ่งอยู่ในระดับกลาง และสาเหตุที่สองเกิดจากการเขียนที่ชิดกันของโครงร่างตัวอักษรซึ่งไปสัมผัสกับหัวที่หนึ่ง ทำให้มีจุดต่อเชื่อมของหัวที่หนึ่งทั้ง 2 ด้าน ซึ่งส่งผลให้การตัดสินใจผิดพลาด

5.3.5 ผลการทดสอบโดยอาศัยจำนวนการเปลี่ยนจากจุดขาวไปเป็นจุดดำ

สำหรับในรายงานผลงานวิจัยฉบับนี้จะไม่มีรายละเอียดของผลการทดสอบของวิธีการแยกแยะตัวอักษรโดยอาศัยจำนวนการเปลี่ยนจากจุดดำไปเป็นจุดขาว เพราะในส่วนนี้มีจุดประสงค์เพื่อเป็นเพียงการนำเสนอแนวทางการออกแบบระบบรู้จำในรูปแบบใหม่ ๆ จึงยังมีความจำเป็นที่จะต้องทำการศึกษาต่อไปอีกมาก แต่กระนั้นผลการทดสอบในชั้นต้นพบว่าเป็นวิธีการอีกรูปแบบหนึ่งที่น่าสนใจ แต่ก็ยังคงประสบกับปัญหาที่คล้ายคลึงกับวิธีอื่น ๆ คือยังไม่สามารถแยกตัวอักษรออกได้ถูกต้อง 100% ซึ่งสำหรับวิธีการนี้คณะผู้วิจัยก็จะได้ทำการศึกษาและวิจัยต่อไป

5.3.6 ข้อจำกัดของวิธีการที่ได้นำเสนอ

ข้อจำกัดบางประการที่ทำให้วิธีนี้ยังไม่ดีพอกับการนำไปใช้งานทางปฏิบัติ

1. ตัวอักษรบางตัวให้ค่าจำนวนหัวเท่ากันและมีการเปลี่ยนของจุดขาวเป็นดำที่เหมือนกันทุกประการ จึงไม่สามารถแยกแยะตัวอักษรเหล่านี้ออกจากกันได้ เช่น ด และ ต เป็นต้น ดูรายละเอียดของตัวอักษรอื่น ๆ ที่มีปัญหาในลักษณะเดียวกันในตารางที่ 5.8
2. ในการแยกกลุ่มตัวอักษรนั้นได้กระทำทีละขั้นตามลำดับ ในแต่ละขั้นหากพบว่าตัวอักษรที่ทดสอบตกอยู่ในกลุ่มใดก็จะตัดสินเลยว่าตัวอักษรนั้นอยู่ในกลุ่มดังกล่าว เพราะฉะนั้นหากการแยกแยะในขั้นต้น ๆ เกิดความผิดพลาด การตัดสินใจภายหลังก็จะผิดไปอย่างที่แก้ไขไม่ได้เลย ยกตัวอย่างเช่น หากการเขียนตัวอักษรไม่มีการควบคุมที่ดีพอ ทำให้ภาพตัวอักษรไม่มีหัวที่ชัดเจนหรืออยู่ในระดับที่ผิด การแบ่งกลุ่มก็จะผิดพลาดไปโดยสิ้นเชิงตั้งแต่ขั้นแรก

ตารางที่ 5.8 กลุ่มตัวอักษรที่ไม่สามารถแยกแยะได้

รูปแบบการเปลี่ยนจากขาวเป็นดำ	กลุ่มตัวอักษร
(2,2,2,2)	ด, ต
(2,2,2,1)	ค, ด, ศ
(1,1,3,1)	พ, ฟ
(1,1,2,2)	ผ,ฝ
(1,1,3,1)	ช, ซ
(1,1,2,1)	ข, บ, ช

จากปัญหาต่างๆเหล่านี้คณะผู้วิจัยยังจำเป็นที่จะต้องทำการศึกษาค้นคว้าเพิ่มเติมอีกมากเพื่อหาวิธีที่มีประสิทธิภาพมากขึ้นกว่านี้ และสามารถที่จะนำมาใช้รู้จำตัวอักษรที่มีการเขียนที่มีข้อจำกัดที่น้อยลง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6. ข้อสรุปและข้อเสนอแนะ

โครงการวิจัยนี้ได้ทำการศึกษา ค้นคว้า พัฒนา แนวทางในการออกแบบระบบรู้จำตัวอักษรเขียนภาษาไทย ซึ่งผลของงานวิจัยที่ผ่านมาได้ผลงานต่างๆออกมาดังพอจะสรุปได้ดังนี้

1. ได้สร้างฐานข้อมูลตัวอักษรเขียนภาษาไทยและอังกฤษที่เป็นระบบและมีขนาดใหญ่เพียงพอ (5 หมื่นกว่าตัวอักษร) ต่อการนำมาทดสอบระบบรู้จำได้อย่างมีประสิทธิภาพ โดยฐานข้อมูลภาพดังกล่าวได้บรรจุอยู่ในแผ่น CD-ROM จำนวนหนึ่งแผ่น สามารถที่จะนำไปใช้ทดสอบกับระบบรู้จำได้ทันที ทั้งนี้ได้ทำการเก็บข้อมูลทั้งหมดเป็น 2 รูปแบบ คือ bitmap และ grey scale 256 ระดับตามมาตรฐานสากล เพื่อให้เหมาะกับการนำไปใช้งานตามต้องการ อีกทั้งยังได้มีการพัฒนาโปรแกรมคอมพิวเตอร์ที่ช่วยในการเพิ่มขนาดของฐานข้อมูลให้มีขนาดใหญ่ขึ้นได้โดยสะดวก และยังสามารถตรวจสอบหรือดูอักษรภาพทั้งหมดได้ในโหมดกราฟฟิก

2. ได้ทำการศึกษาวิธีการปรับปรุงคุณภาพและการประมวลผลภาพเบื้องต้นที่จำเป็นต่อการพัฒนาระบบรู้จำทั้งหมด 5 หัวข้อหลัก คือ การกำจัดสัญญาณรบกวน (noise reduction) การทำตัวอักษรให้บาง (thinning) การปรับรูปปกติของตัวอักษร (normalization) การหมุนภาพ (rotation) และการทำให้ภาพโย้ (slant) ทั้งนี้ในแต่ละหัวข้อจะมีแนวทางการทำได้ต่างๆกันไปอีก โดยในโครงการนี้ได้ทำการพัฒนาโปรแกรมคอมพิวเตอร์ให้สามารถประมวลผลภาพได้ทั้ง 5 หัวข้อด้วยวิธีการต่างๆหลายวิธี และได้ทำการทดสอบสมรรถนะเพื่อวิเคราะห์ถึงข้อดีและข้อด้วยของแต่ละวิธีโดยเน้นไปที่ตัวอักษรภาษาไทยเป็นหลัก

3. ได้ทำการศึกษาถึงลักษณะเฉพาะสำหรับบ่งถึงความแตกต่างของตัวอักษร ที่มีความเหมาะสมและสามารถนำมาใช้ในการแยกแยะตัวอักษรภาษาไทยทั้งสิ้น 4 ประเภท คือ ขอบของภาพ (Edge) จุดปลายของภาพ (Endpoint) หัวของตัวอักษร (Head of Character) และจำนวนการเปลี่ยนจากจุดขาวเป็นจุดดำ (Stoke Changing) และได้แสดงให้เห็นว่าลักษณะที่บ่งถึงความแตกต่างทั้ง 4 ประเภทนี้มีประโยชน์อย่างมากต่อการนำไปใช้ในการรู้จำตัวอักษรโดยเฉพาะกับตัวอักษรไทย ทั้งนี้ได้ทำการพัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถดึงลักษณะที่บ่งถึงความแตกต่างทั้ง 4 ประเภทไว้เรียบร้อยแล้ว สามารถนำมาประยุกต์ใช้ได้กับระบบรู้จำอื่นๆที่อาจจะพัฒนาต่อไปในอนาคต

4. ได้นำเสนอแนวทางการพัฒนาระบบรู้จำแบบใหม่โดยอาศัยคุณลักษณะเฉพาะที่บ่งถึงความแตกต่าง 2 อย่าง คือจำนวนของหัวและจำนวนการเปลี่ยนของจุดขาวไปเป็นจุดดำ มาประยุกต์ใช้ในการรู้จำ ซึ่งผลการทดสอบในเบื้องต้นพบว่าวิธีการนี้สามารถนำมาใช้ได้ดีภายใต้สภาพแวดล้อมที่กำหนดเท่านั้น และยังมีอักษรบางกลุ่มที่ยังไม่สามารถแยกออกจากกันได้

โดยรวมแล้วผลงานวิจัยที่ได้กระทำมาทั้งหมดถึงแม้ว่าจะยังไม่เพียงพอต่อการนำไปสร้างระบบรู้จำที่มีประสิทธิภาพเพื่อการนำไปใช้งานในทางปฏิบัติ แต่ขอบเขตของการศึกษาค้นคว้าและวิจัยก็ได้กระทำอย่างละเอียดครอบคลุมในทุกแง่มุมของหลักการพื้นฐานต่างๆที่จำเป็นต่อการพัฒนาระบบรู้จำในระดับที่น่าพอใจ และเป็นประโยชน์อย่างยิ่งต่อคณะผู้วิจัยสำหรับการนำไปใช้ทดสอบแนวความคิดใหม่ๆที่มีประสิทธิภาพในงานวิจัยในอนาคตต่อไป

ข้อเสนอแนะ

จากประสบการณ์งานวิจัยที่ผ่านมาได้ข้อสังเกตที่น่าสนใจอย่างหนึ่งว่า ประสิทธิภาพของระบบรู้จำนอกจากจะขึ้นอยู่กับกรรมวิธีที่ใช้ในการแยกแยะตัวอักษรแล้ว คุณภาพของภาพตัวอักษรที่ใช้ทดสอบเป็นอีกปัจจัยหนึ่งที่มีผลกระทบต่อผลลัพธ์ที่ได้อย่างเห็นได้ชัด บางระบบอาจจะมีเปอร์เซนต์ความถูกต้องของการรู้จำที่สูงมากเมื่อทดสอบกับภาพอักษรชุดหนึ่ง แต่อาจจะให้ผลที่แตกต่างไปโดยสิ้นเชิงเมื่อนำมาทดสอบกับภาพอักษรอีกชุดหนึ่ง ดังนั้นในการเปรียบเทียบระบบรู้จำที่ต่างกันจำเป็นจะต้องมีการทดสอบกับฐานข้อมูลชุดเดียวกัน นอกจากนี้คณะผู้วิจัยยังเห็นควรที่จะมีการแบ่งประเภทของฐานข้อมูลออกเป็น 2-3 ประเภทตามคุณภาพของภาพอักษร ทั้งนี้เพราะคณะวิจัยเชื่อว่าเป็นเรื่องยากมากที่สามารถสร้างระบบรู้จำที่ให้ผล 100% ได้ หากไม่มีการควบคุมคุณภาพของลายมือที่เขียนเลย ยกตัวอย่างเช่นในบางกรณีแม้แต่สายตามนุษย์ยังไม่สามารถแยกแยะลายมือของคนบางคนได้ เมื่อพิจารณาโดยรวมแล้วพบว่าปริมาณงานวิจัยกับภาษาไทยยังมีอยู่น้อยมาก หัวข้อวิจัยนี้น่าจะต้องมีการศึกษาค้นคว้าต่อไปอีกมาก เพื่อให้ได้ระบบรู้จำที่มีประสิทธิภาพและขีดความสามารถสูงขึ้นกว่าที่เป็นอยู่

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เอกสารอ้างอิง

[1] Frank Y. Shih and Wai-Tak Wong, "A New Safe-Point Thinning Algorithm Based on the Mid-Crack Code Tracing," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 2, February 1995, pp. 370-377.

[2] Ben K. Jang and Roland T. Chin, "One-Pass Parallel Thinning: Analysis, Properties, and Quantitative Evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 11, November 1992, pp. 1129-1140.

[3] Y. S. Kim, W. S. Choi and S. W. Kim, "High-Speed Thinning Processor for Character Recognition System," *IEEE Transactions on Consumer Electronics*, Vol. 38, No. 4, November 1992, pp. 762-766.

[4] Seong-Whan Lee, Jeong-Seon Park and Yuan Y.Tang, "Performance Evaluation of Nonlinear Shape Normalization Methods for the Recognition of Large-Set Handwritten Characters", *Proceedings of the Second International Conference on Document Analysis and Recognition*, 1993, pp. 402-407.

[5] กิตติพงษ์ เจนวิถีสุข, การรู้จำอักษรพิมพ์ภาษาไทยโดยใช้นิวรอลเน็ตเวิร์กและวิธีซินแทกติก, วิทยานิพนธ์ปริญญามหาบัณฑิต, บัณฑิตวิทยาลัย, จุฬาลงกรณ์มหาวิทยาลัย, 2538.

[6] Ioannis Pitas, **Digital Image Processing Algorithms**, Prentice Hall, 1993.

[7] Stefan Knerr, Leon Personnaz and Gerard Dreyfus, "Handwritten Digit Recognition by Neural Networks with Single-Layer Training", *IEEE Transactions on Neural Networks*, Vol. 3, No. 6, pp. 962-968, November 1992.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

บทความ (ฉบับร่าง) ที่ได้รับการตอบรับให้ตีพิมพ์ในการประชุมนานาชาติ IEEE Asia Pacific Conference on Circuits and Systems: Microelectronics and Integration Systems, November 24-27, 1998, Chiangmai, Thailand.

DISTINCTIVE FEATURE ANALYSIS FOR THAI HANDWRITTEN CHARACTER RECOGNITION BASED ON MODIFIED STROKE CHANGING SEQUENCE

Choruengwiwat,P. , Jitapunkul,S. , Wuttisittikulkuj,L. , Seehapan,P.

Digital Signal Processing Research Laboratory, Department of Electrical Engineering,

Faculty of Engineering, Chulalongkorn University, Bangkok 10330, THAILAND

E-Mail : jsomchai@chula.ac.th

ABSTRACT

This paper was proposed to used modify Stroke Changing Sequence (SCS) and distinctive feature to recognize Thai handwritten character. From character head center, Stroke Changing Sequence will be drawn both horizontal and vertical to detect stroke changing. The experimental results are 92% of recognition rate, 4% of both error and undecision. The processing speed is about 5.8 character per second.

I. Introduction

There were several researches of Thai hand written recognition [[1],[2],[3]]. Airphaboon,S., Sangworasil,M.[1] proposed technique by considering the head of character. Tanprasert,C. ,and Tanprasert,T. [2] used simulated light sensitive model technique for Thai digit recognition and by same technique by Oliver,V., Wangsuya,S. and Coomans,D. [3] but for Thai hand written characters. The Thai hand written recognition system that proposed by Tanprasert,C. and Tanprasert,T. [2] in Thai digit and Oliver, Wangsuya,S. [3] in Thai consonant aim to used light sensitive for filter a feature in each direction that design for Thai characters. The features in character are very important on decisions in recognition system.

Phanit,W. [4] had shown that distinctive feature analysis technique was very efficient method to classify and separated Thai characters into groups. However, Gwo-En,and Jhing-Fa Wang [[5],[6]] had propose to used stroke changing sequence (SCS) technique to encode handwritten numeral character for recognition which is a very simple and fast technique that gave an excellent result.

However, unfortunately, the SCS was executed in horizontal line proposed by Gwo-En, and Jhing-Fa Wang [[5],[6]] not suitable to Thai handwritten character recognition at all. This was mainly due to the code obtained from horizontal SCS for a Thai character might be same as one obtained for another Thai characters. So, we had modified this technique by adding vertical SCS in accordance with horizontal SCS.

II. Distinctive Feature in Thai Characters

Thai characters compose of 44 characters which can classify into several feature such as number of heads, roof of character etc. By using distinctive feature analysis to distinguish a character from another, Phanit,W. [4] had shown that the good result could be obtained for Thai printed characters. But, for Thai hand written character, this technique had the limitation because each character may very from one writer to another in size, shape and writing style.

Airphaboon,S. and Sangworasil,M. [1] had proposed to use number of head to classify Thai hand written character in groups as shown in Table 1.

Number of heads	Members	Total
No Head	ก, ธ	2
1 Head	ข,ฃ,ค,ฅ,ง,จ,ช,ฌ,ด,ต,ถ, ท,บ,ป,ผ,ฝ,พ,ฟ,ภ,ย,ร,ล,ว,	27

	ศ,ส,อ	
2 Head	ฆ,ฉ,ฌ,ญ,ฎ,ฏ,ฒ,ณ,น,ม,ษ, ห,ฬ,ฮ	14
3 Head	ฐ	1
	Grand total	44

Table 1 Classification Thai consonant by used number of head.

The SCS is one of the fastest technique and gives suitable features, which was proposed and experimental by Gwo-En and Jhing-Fa Wang[[5],[6]], that:

"The crossing method algorithm was selected because it required little computational overhead in extracting the required features from the data. The crossing number is calculated from the number of dark blocks of pixels that may be found in a particular line of data. For instance, if the data line being checked contains a single line of dark pixels, then the crossing number will be set to one. The crossing method, starting from the top left corner, searches for the crossing number for each line of the character horizontally. A crossing number sequence is defined as sequence."

For example, a SCS of this handwritten Thai consonant character "ก" is 121,shown in Fig. 1.

```
1111111000000001111111  1
1111000000000000011111  1
1100000000000000011111  1
1000000111111110000111  2
1000000111111111000011  2
1100000011111111100011  2
1111000001111111100011  2
1111000111111111100011  2
1110000011111111100011  2
1110001111111111110011  2
1110001111111111110011  2
1100011111111111110011  2
1100011111111111110011  2
1100011111111111110011  2
1100011111111111100011  2
1100011111111111100011  2
1100011111111111100011  2
1100011111111111111111  1
1100011111111111111111  1
```

Horizontal SCS 121

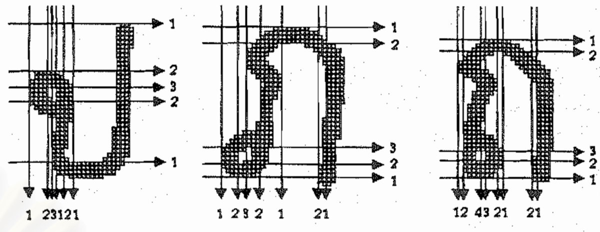
Fig 1 Thai consonant character "ก" has horizontal SCS

III. Problem in Stroke Changing Sequence (SCS)

Stoke changing sequence codes are obtained from examination of the changing of pixels value in each stroke line. The result from each stroke is then arranged to obtain the sequence order.

A) Problems associated in the implementation of horizontal SCS (HSCS) following by Gwo-En and Jhing-Fa Wang[[5],[6]].

There are 44 consonants in Thai language and horizontal stroke changing technique will give the same coded sequence for some of Thai Characters, for example, the character "ถ" and "ภ.", see in Fig 2. Due to this reason, the vertical changing sequence was also applied to assist the analysis but there may be disturbance at along with the boundary of character.



(a) (b) (c)

Fig. 2 HSCS and VSCS in Thai character

(a) "ป" have HSCS = 12321 VSCS = 123121

(b) "ภ" have HSCS = 12321 VSCS = 1232121

(b) "ภ" have HSCS = 12321 VSCS = 12432121

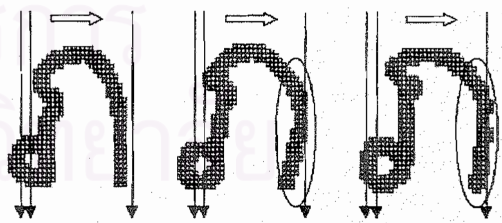
B) First modification of SCS by vertical SCS (VSCS)

Due to the problem associated with horizontal SCS, thus, vertical stroke was applied to enhance the capacity of group classification but problem persisted for the varying written characters, see VSCS in Fig. 2. Using both HSCS and VSCS, it gave several codes for some characters and some codes had long code sequence, which will produce many template sequences. For example, the character "ฐ" stroke sequence is very sensitive to writing characteristic. Small changes in writing may result in different codes for the SCS.

As discussion above, the problem in the VSCS is represented in the Fig. 3.

The problem

- From the changing of shape.
- From the disturbance of the character boundary.



(a) (b) (c)

VSCS : 12321 123212 123212

Fig. 3. Vertical SCS problems

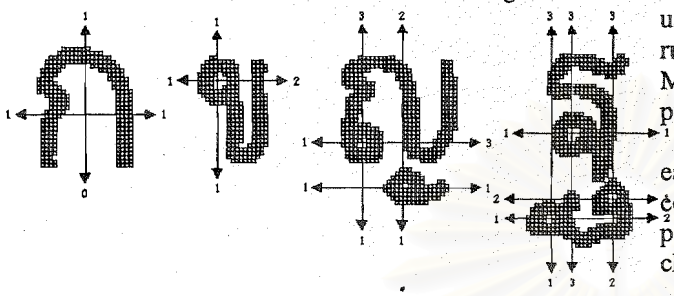
(a) Normal character (b) Problem in style written

(c) Problem in noise at edge of image

IV. Modified Stroke Changing Sequence

In order to improve the disadvantages obtained by both distinctive feature and HSCS and VSCS, both HSCS and VSCS will be considered respectively to character head center. We will take SCS both horizontal and vertical to the character heads center.

To obtain stroke changing code, both horizontal and vertical lines will be drawn from each head center in left and right direction for horizontal line and up and down direction for vertical line as shown on Fig. 4.



(a) (b) (c) (d)

Fig. 4 SCS from center of head in Thai character
(a) Case have not head (b) Case have one head
(c) Case have two heads (d) Case have three heads

Table 2 shows modified SCS code, (left, up, right, down) respectively, for Thai character in each case by considering number of head.

		SCS at center of character		
Case	Character	1'st head	2'nd head	3'rd head
0 head	ก	(1,1,2,1)		
1 head	ข	(2,2,2,1)		
2 head	ญ	(1,3,3,1)	(1,2,1,1)	
3 head	ฐ	(1,3,1,3)	(1,3,2,1)	(2,3,1,2)

Table 2 The code of modified SCS in each case of character head.

However, if this technique was only used to classify Thai character, it will give a same SCS code for some different characters, as shown in Table 3.

SCS	Group of characters
(2,2,2,2)	ด, ต
(2,2,2,1)	ค, ฅ, ฆ
(1,1,3,1)	พ, ฟ
(1,1,2,2)	ผ,ฝ
(1,1,3,1)	ข, ช
(1,1,2,1)	ข, บ, ช

Table 3 The members character that give same SCS

Fig. 5 shows modified SCS from head center of all 44 Thai consonant characters.

The advantage of using modified SCS is that, it is not necessary to scan several stroke lines as proposed by Gwo-En and Jhing-Fa Wang[[5],[6]] to obtain stroke changing code. But, only 4 stroke lines from each character head will be obtained.

V. Experimental results

The Thai handwritten recognition system based on modified SCS and number of head was developed using Visual C ++ on Windows 95. The system used was run on Pentium 133 MHz Personal Computer with 56 Mbyte Memory. Input data was acquired from tablet pen base and stored in standard binary image.

Thai consonant characters with 5 characters for each pattern were analyzed to set the modified SCS codes. Then, testing characters was taken from 28 persons and wrote 5 times for each character, 6160 characters in total

SCS	Group of characters	% right group
(2,2,2,2)	ด, ต	98
(2,2,2,1)	ค, ฅ, ฆ	97
(1,1,3,1)	พ, ฟ	92
(1,1,2,2)	ผ,ฝ	96
(1,1,3,1)	ข, ช	86
(1,1,2,1)	ข, บ, ช, ป	88

Table 4 Results in character that give same SCS.

From table 4, in the case of modified SCS that produced same code, the percentage group of characters "ข,ช" and "ข,ช,บ,ป" is lower that others. In this case, the characters may have 2 possible actions; one head and two heads. By considering number of head, character "ข, ช,ช,ช" should have one head but some people write style may caused, two heads, as shown in Fig. 6. Hence, the character "ข,ข,ช,ช" will be considered in both actions.

VI. Conclusion

By using modified SCS with another distinctive feature, each Thai handwritten character can be classified into each group with the result are 92% recognition rate, 4% error and 4% undecision. The processing speed is about 5.8 characters per second.

Acknowledgement

The author would like to acknowledge the effort of hand writing contributors. This research is supported by the Digital Signal Processing Research Laboratory and the Telecommunication Consortium Scholarship, National Science and Technology Development Agency (NSTDA).

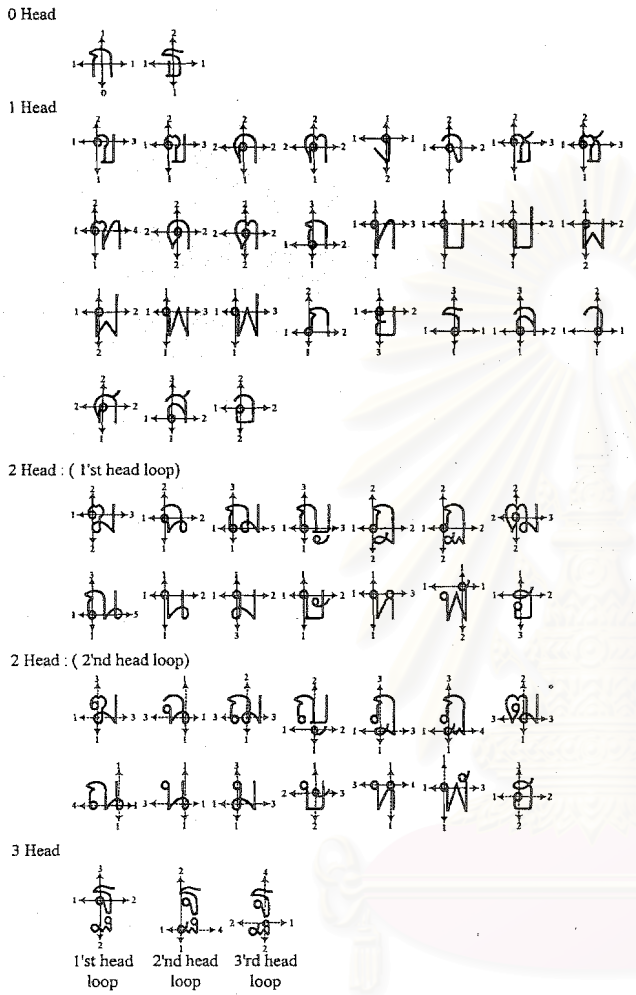
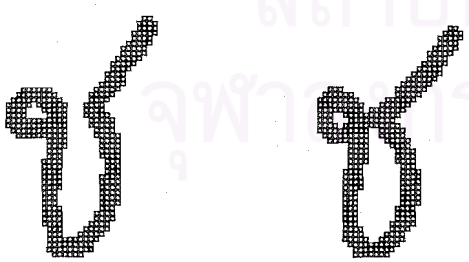


Fig. 5 Modified SCS from head center of Thai consonant characters.



(a) (b)

Fig. 6 The problem in number of heads (a) Normal (b) Have 2 loops of head

References

[1] Suraphun Airphaboon, Manas Sangworasil, "Recognition of Handwritten Thai Character Considering The Head of Character", IEEE Inter Conference on Image Proceeding, Singapore 1989.

[2] Chularat Tanprasert ,Thitipong Tanprasert, "Variable Simulated Light Sensitive Model for Handwritten Thai Digit Recognition", Proceedings of the Symposium on Natural Language Processing in Thailand, 17-21 March 1995.

[3] Oliver de vil, Sujint Wangsuya and Danny Coomans, "On Thai Character Recognition", IEEE International Conference on Published, Vol. 4, 1995.

[4] Phanit W. "Distinctive Feature Thai Character Recognition", Master's thesis, Chulalongkorn University, Thailand 1997.

[5] Gwo-En,Jhing-Fa Wang, "A New Approach for Recognition of Unconstrained Handwritten Numverals",IEEE TENCON, 1993.

[6] Gwo-En,Jhing-Fa Wang, "A New Hierarchical Approach for Recognition of Unconstrained Handwritten Numerals", IEEE Transactions on Consumer Electornics, Vol.40 No.3 August 1994.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย