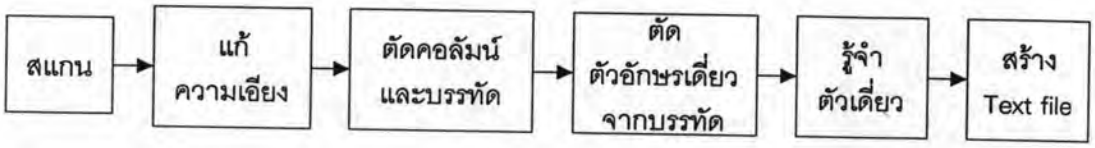
บทที่ 3
กระบวนการที่ใช้ในระบบรู้จำอักษรภาษาไทย

กล่าวนำ

ระบบรู้จำอักษรภาษาไทย คือ ระบบที่เปลี่ยนภาพเอกสารภาษาไทยที่ได้จากการสแกนมาเป็น Text File โดยขั้นตอนจะเริ่มจากนำเอกสารที่อยู่ในรูปกระดาษมาสแกนแล้วเก็บในรูปแบบแฟ้มรูปภาพ จากนั้นทำการแก้ความเอียงซึ่งเกิดจากการใส่เอกสารเอียง ตัดคอลัมน์และบรรทัด ทำการดึงจุดภาพที่ติดกันเป็นเกาะ (หมายถึงจุดภาพที่มีตำแหน่งต่อเนื่องกัน ซึ่งอาจเป็นอักษรเดี่ยว หลายตัวอักษร หรือเป็นสัญญาณรบกวนก็ได้) ออกจากบรรทัด แล้วแยกแต่ละเกาะออกจากกันเป็นอักษรเดี่ยว จากนั้นส่งตัวอักษรเดี่ยวไปให้กระบวนการแยกแยะได้เป็นรหัส ASCII สุดท้ายนำมาเรียงกลับให้เหมือนกับเอกสารต้นแบบในรูป Text File เราสามารถเขียนแผนภาพกระบวนการหลักๆ ของระบบได้ดังรูปที่ 3.1

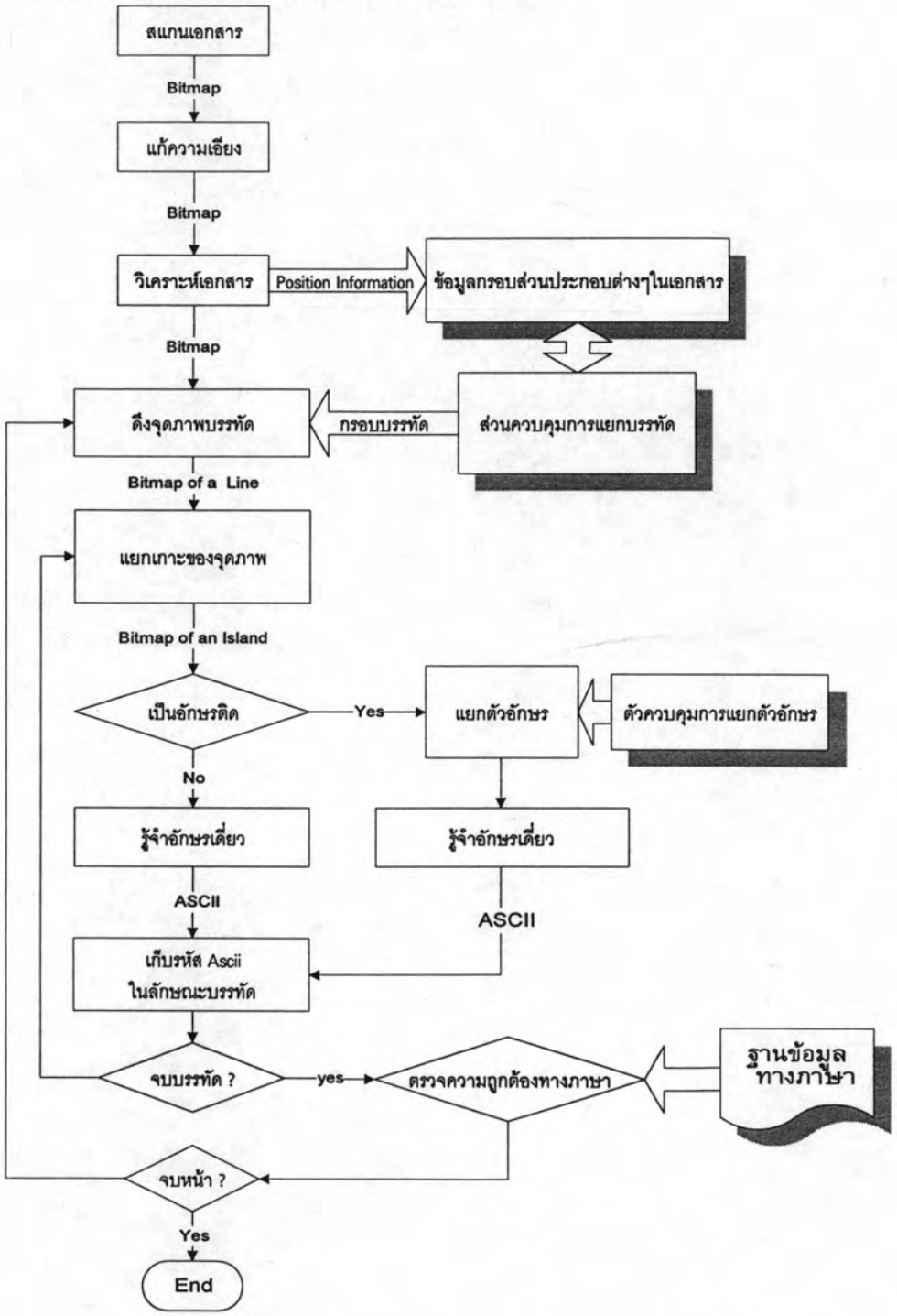


รูปที่ 3.1 กระบวนการหลักของระบบรู้จำเอกสารภาษาไทย

การเปลี่ยนแปลงข้อมูลเริ่มต้นที่ระบบรับมาคือภาพเอกสารที่สแกนมาได้ (ในลักษณะบิทแมป) จนเป็น Text file มีขั้นตอนดังนี้

1. เมื่อผ่านการแก้ความเอียงเราจะได้ภาพเอกสารที่มีเส้นแนวบรรทัดอยู่ในแนวระดับ
2. จากนั้นผ่านส่วนแยกคอลัมน์และแยกบรรทัดซึ่งผลที่จะได้คือข้อมูลของกรอบของแต่ละบรรทัดและลำดับของแต่ละกรอบ
3. ระบบจะตัดจุดภาพของแต่ละบรรทัดออกมาตามกรอบที่ได้ ข้อมูลตอนนี้คือ บิทแมปของแต่ละบรรทัด
4. ระบบจะดึงเกาะของจุดที่ต่อเนื่องกันที่อยู่ในบรรทัดซึ่งอาจจะเป็นอักษรเดี่ยว อักษรติด เศษอักษรที่ขาด สัญญาณรบกวนก็ได้ ผลขณะนี้คือ บิทแมปของแต่ละเกาะ พร้อมทั้งหาว่าเกาะที่ได้เป็นลักษณะใดเมื่อเทียบกับอักษรในบรรทัดเดียวกัน
5. ภาพเกาะจะผ่านส่วนตรวจสอบว่าเป็นอักษรติดหรือไม่โดยใช้ ***ลักษณะบ่งความต่างของอักษรไทย*** โดยถ้าติดจะตัดแยกเป็นอักษรเดี่ยว ดังนั้นผลที่ได้คือ บิทแมปของอักษรเดี่ยว
6. เข้าขั้นตอนการรู้จำอักษรเดี่ยวมีขั้นตอนย่อยดังนี้
   - นำภาพอักษรมาหากลุ่มซึ่งกำหนดไว้ดังในตารางที่ 3.6
   - หา ***ลักษณะบ่งความต่างของอักษรไทย*** ตามกลุ่มที่ได้แบ่งไว้
   - นำลักษณะบ่งความต่างที่ได้มาเทียบกับฐานข้อมูลให้คำตอบเป็น ASCII

7. นำ ASCII มาเรียงเป็นบรรทัดแล้วตรวจสอบความถูกต้องทางด้านภาษา แล้วเขียนเป็น Text file เราสามารถเขียน flow chart การทำงานของระบบดังนี้



รูปที่ 3.2 แสดง Flow Chart ของระบบรู้จำเอกสารภาษาไทย

เราจะแบ่งเนื้อหาในบทนี้เป็น 3 ส่วนหลักคือ

1. ส่วนวิเคราะห์เอกสารประกอบด้วย การแก้ความเอียง แยกคอลัมน์และแยกบรรทัด การดึงเกาะของจุดภาพออกจากบรรทัด(ซึ่งรวมการกำจัดสัญญาณรบกวนด้วย) การหาระดับของอักษร การเพิ่มความถูกต้องด้วยข้อมูลทางภาษา และการสร้าง Text file
2. กรรมวิธีในการแยกตัวอักษรที่ติดกัน (Segmentation of Connected Characters)
3. การรู้จำอักษรเดี่ยว (Recognition of Single Character)

3.1 ส่วนวิเคราะห์เอกสาร

3.1.1 การแก้ความเอียง

เนื่องจากขั้นตอนการหากรอบของบรรทัดนั้นใช้กรอบแบบ 4 เหลี่ยม การที่เอกสารเอียงจะทำให้การตัดบรรทัดและตัดคอลัมน์ผิดพลาดได้ อีกทั้งขั้นตอนการหาระดับของอักษรจะเทียบกับอักษรข้างเคียง ถ้าเอกสารเอียงจะก่อให้เกิดความผิดพลาดเช่นกัน

เราทำการทดลองหาความเอียงที่เกิดจากการวางกระดาษไม่ตรง โดยให้ผู้ที่เคยใช้สแกนเนอร์ทำการวางเอกสารแล้วสแกนจำนวน 10 คน คนละ 5 หน้ากระดาษได้ผลดังตาราง

ตารางที่ 3.1 แสดงจำนวนหน้าที่ทำเอกสารเอียงในมุมขนาดต่างๆ

องศาที่เอียง(ค่าสัมบูรณ์)	0 ถึง 2.5	2.5 ถึง 5	5 ถึง 7.5	มากกว่า 7.5	รวม
จำนวน(หน้า)เอกสารที่เอียง	29	21	0	0	50

จากการทดลองพบว่ามีการเอียงทั้งแบบตามและทวนเข็มนาฬิกาซึ่งมีค่าไม่เกิน 5 องศา เรากำหนดช่วงปลอดภัยเป็น 50 % ดังนั้นอัลกอริทึมที่เราจะต้องสร้างขึ้นต้องสามารถแก้ความเอียงได้อย่างต่ำ 7.5 องศา แบ่งขั้นตอนเป็น 2 ขั้นคือหาองศาความเอียง และปรับให้ตรง

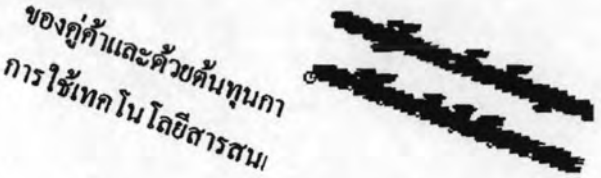
3.1.1.1 การหาองศาที่เอกสารเอียง

มีบทความหลายบทความเสนอวิธีเช่น Stuart C. Hinds et. al (1990) ใช้ run-length encoding ร่วมกับ Hough Transform ซึ่งบอกว่าได้แก้จุดเสียของ Hough Transform ซึ่งใช้การคำนวณมาก โดยการลดจำนวนข้อมูลเข้า Hough Transform ซึ่งเดิมเป็นจำนวนจุดภาพทั้งหมดเป็นการใช้ horizontal และ vertical black run-lengths ซึ่งเมื่อเราพิจารณาแล้วก็ยังต้องใช้ Hough Transform ซึ่งค่อนข้างยุ่งยากและเสียเวลามากอยู่ดี

เวลาที่เสียไปนั้นวิเคราะห์แล้วมาจากต้องการหาองศาที่แน่นอนของความเอียง ซึ่งจริงๆแล้วขั้นตอนต่อไปคือการตัดบรรทัดและหาระดับของบรรทัดนั้นสามารถทำให้ทนต่อความเอียงเล็กน้อยได้ (ประมาณ 1 องศา โปรดดูในบทที่ 4 จะแสดงตารางที่เปรียบเทียบผลการรู้จำเอกสารที่เอียง 1 องศา และที่ไม่เอียงซึ่งให้ผลไม่ต่างกัน แต่ถ้าเอกสารเอียงเกิน 1 องศา จะเกิดความผิดพลาดขึ้น) ดังนั้นเราจึงเลือกวิธีที่ให้ความถูกต้องพอสมควรแต่ยุ่งยากน้อยดังวิธีที่จะเสนอดังนี้

ขั้นตอนการหาความเอียงมีดังนี้

1. Paste คือการลากเส้นต่อจุดภาพในแนวนอนเคียงกันที่ห่างกันไม่เกิน N จุดภาพ ซึ่งภาพอักษรจะรวมกันเป็นกลุ่มก้อนซึ่งค่า N จากการทดลองกับขนาดอักษร 10-24 point แล้วได้ N = 20
2. จากสมมุติฐานที่ว่า เส้นบรรทัดด้านล่างจะเรียบ ดังนั้นผลจากการรวมจุดภาพเอกสารให้เป็นกลุ่มก้อนโดยขั้นตอนที่ 1 จะได้ผลดังรูปที่ 2.3



รูปที่ 3.3 แสดงภาพอักษรก่อนและหลังการ Paste

เราทำการหาขอบของภาพหลังการ Paste แล้วเลือกบางจุดบนขอบที่เป็นเส้นล่างของบรรทัด มีหลักคือถ้าเราเริ่มหาขอบโดยจุดเริ่มต้นเป็นซ้ายสุดแล้ววนทวนเข็ม ความยาวของขอบ 1 ใน 3 ของขอบที่ได้จะเป็นเส้นล่างของบรรทัด

3. เราเลือกจุดมากลุ่มหนึ่งแล้วเข้าสูตร regression ได้ slope = $\frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2}$

การ Paste มีขั้นตอนดังนี้

```
for y = 1 to line_height
    for x = 1 to line_width
        if ( pixel ( x , y ) = 1 )      set = N
        else        set=set-1
        if ( set > 0 ) put a dot in temporary image
```

การหาขอบของภาพหลังการ Paste โดยวนทวนเข็ม มีอัลกอริทึมดังนี้

```
1  หาจุดเริ่มต้นทางซ้ายสุด
2  จุดถัดไป = ฟังก์ชันหาจุดถัดไป(จุดเริ่มต้น)
3  while( จุดถัดไป ≠ จุดเริ่มต้น) do
       จุดถัดไป = ฟังก์ชันหาจุดถัดไป(จุดถัดไปเดิม)
       เก็บจุดถัดไปไว้ใน stack
```

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ส่วนฟังก์ชันหาจุดถัดไป เนื่องจากเป็นการหาจุดรอบทวนเข็ม ดังนั้นจุดถัดไปจึงเป็นจุดที่พบจุดแรกเมื่อเวียนทวนเข็มด้วย

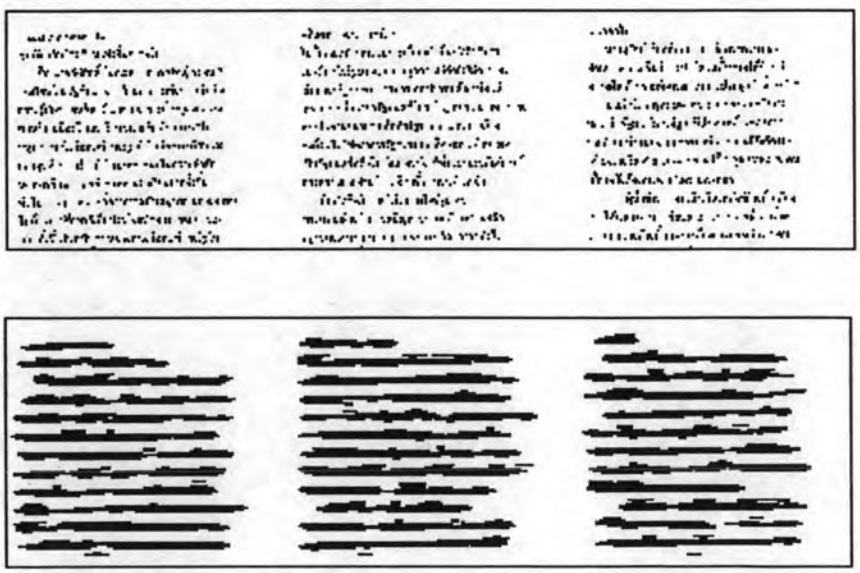
3.1.1.2 การหมุนเอกสาร

เมื่อได้มุมที่ต้องการหมุนเอกสารแล้วจึงนำมาสู่กระบวนการหมุนเอกสาร(รอบมุมซ้ายบน) มีกระบวนการดอนดังนี้

```
for all ( pixel(x,y) = 1 )
        radius = sqrt(x^2 + y^2)
        new_theta = tan^-1( y / x ) + มุมที่ต้องการหมุน
        new_x = radius cos( new_theta )
        new_y = redius sin( new_theta )
        put_dot( new_x , new_y )
```

3.1.2 แยกคอลัมน์และแยกบรรทัด

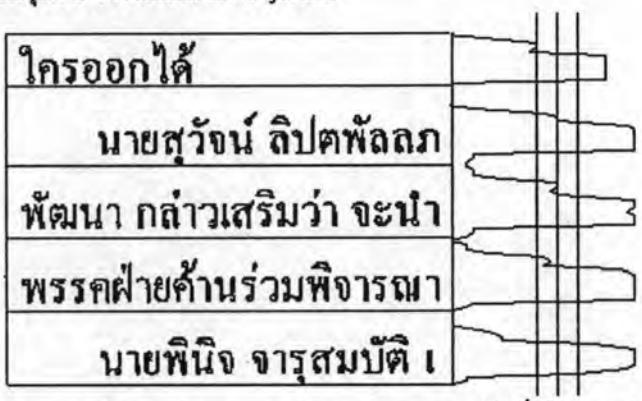
เริ่มต้นด้วยใช้วิธีการ Paste ในลักษณะเช่นเดิม แต่ทำในแนวดิ่งด้วยซึ่งจะทำให้สระระดับบน สระระดับล่างและอักษรระดับกลางจะรวมกันเป็นบรรทัดเดียว ทั้งนี้เพื่อให้การตัดคอลัมน์ไม่ตัดผิดไปตัดระหว่างอักษรในบรรทัด และไม่ตัดบรรทัดผิดไปตัดช่องว่างระหว่างระดับอักษร (ข้อนี้สำคัญเพราะต่างจากอักษรภาษาอังกฤษซึ่งไม่มีช่องระหว่างระดับอักษร)



รูปที่ 3.4 แสดงภาพก่อนและหลังการ Paste ทั้งแนวนอนและแนวดิ่ง

3.1.2.1 *การแยกคอลัมน์* ใช้ Vertical Projection คือการรวมจุดในแนวดิ่ง ใช้ช่วงที่ไม่มีจุดภาพเป็นจุดที่จะตัดคอลัมน์

3.1.2.2 *การแยกบรรทัด* ใช้ Horizontal Projection คือการรวมจุดในแนวนอน เราจะใช้ช่วงที่ไม่มีจุดภาพเป็นจุดที่จะตัดคอลัมน์ไม่ได้เพราะเมื่อบรรทัดเอียงเล็กน้อยทำให้ไม่มีช่องทะลุที่จะตัดบรรทัดได้เราจึงต้องใช้ช่องระหว่างภูเขาที่เกิดจากการ Project แต่ปัญหาเกิดขึ้นอีกว่าใช้เกณฑ์อะไรบอกว่าจุดนั้นต่ำพอที่จะเป็นหุบเขาได้ ซึ่งทำได้ง่ายโดยเทียบกับยอดเขา แต่ปัญหาต่อมาเกิดอีกว่าถ้าบางบรรทัดมีอักษรน้อย ยอดเขาจะต่ำ อาจถูกมองว่าเป็นหุบเขาได้ เราจึงต้องปรับปรุงการทำ Horizontal Projection



รูปที่ 3.5 แสดง Modified Horizontal Projection และระดับที่ใช้ในการตัดบรรทัด

ในการทำ Horizontal Projection ตามปรกตินั้นเราจะได้ลักษณะภูเขา 1 ลูกต่อ 1 บรรทัด ถ้าบรรทัดยาวใกล้เคียงกันความสูงของยอดภูเขาแต่ละลูกจะใกล้เคียงกัน แต่ในกรณีที่บางบรรทัดสั้น เช่น จบย่อหน้า หัวเรื่อง จะทำให้ภูเขาบางลูกเล็ก เราปรับ Horizontal Projection โดยเรียกว่า ***Modified Horizontal Project*** โดยแบ่งแนวนอนเป็น 6 ส่วน (เลือก 6 เพราะส่วนที่เหลือของย่อหน้านั้นปกติจะมีความยาวมากกว่า 1 ใน 6 ของบรรทัด ถ้าบรรทัดใดสั้นกว่านั้นจะยอมให้เป็นความผิดพลาดไป) แล้วเลือกค่า Horizontal Projection ช่วงที่มีค่ามากที่สุดของแต่ละบรรทัด จะได้ผลดังรูปที่ 3.5 เห็นว่าบรรทัดแรก ส่วนที่แบ่งเป็น 6 ส่วนในส่วนหลังๆ จะไม่มีค่าที่ได้จากการ Project เลย แต่ยังได้จากส่วนแรก ดังนั้นลักษณะภูเขาที่ได้ทางด้านขวาจึงใกล้เคียงกับบรรทัดอื่นที่มีอักษรเต็ม

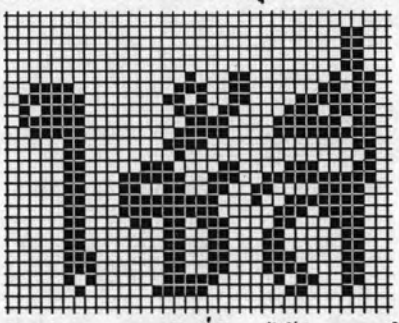
จากนั้นทำการหาจุดตัดโดยตรวจสอบหาหุบเขาซึ่งใช้หลักการว่า

" จุดตัด = จุดที่ต่ำที่สุดระหว่างความชันที่เป็นลบซึ่งต่อเนื่องจากยอดถึงพื้นราบ
กับความชันที่เป็นบวกซึ่งต่อเนื่องจากพื้นราบถึงยอดเขา "

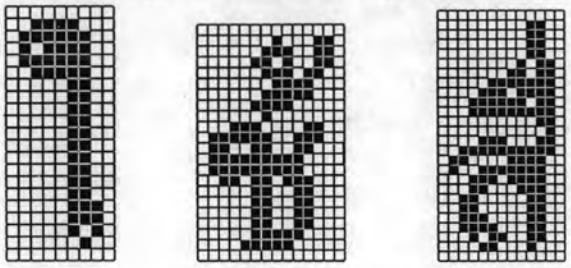
การที่จะตรวจว่าต่อเนื่องจากยอดถึงพื้นหรือไม่นั้นเราจะกำหนดความสูงที่อนุโลมว่าเป็นยอดแต่ไม่ใช่จุดที่สูงสุดจริงๆเพราะทุกยอดสูงไม่เท่ากันและจะกำหนดส่วนที่อนุโลมความสูงเป็นที่ราบซึ่งไม่ใช่ศูนย์เพราะจุดตัดของบางบรรทัดอาจไม่เป็นศูนย์ โดยถ้าเส้นกราฟตัดผ่านจุดทั้ง 2 นั้นแล้วจะถือว่าเป็นส่วนที่มีความชันเป็นลบต่อเนื่อง สำหรับส่วนที่ความชันเป็นบวกก็ทำเช่นเดียวกัน

3.1.3 การดึงเกาะของอักษร

เกาะของอักษรหมายถึง กลุ่มก้อนของจุดภาพที่ต่อเนื่องกันดังภาพตัวอย่างบรรทัดนี้ประกอบด้วย " ใช้ สี่ " ซึ่งสามารถแยกได้เป็น ใ 1 เกาะ ช้ 1 เกาะ เนื่องจากจุดภาพของ ช ติดกับ ไม้โท และ สี่ 1 เกาะ



รูปที่ 3.6 แสดงตัวอย่างภาพบรรทัดที่ตัดมาได้ก่อนการดึงเกาะของอักษร



(ก) (ข) (ค)

รูปที่ 3.7 แสดงเกาะของอักษรที่ดึงได้ทั้ง 3 เกาะ ก) ใ ข) ช้ ค) สี่

อัลกอริทึมดังนี้

1 หาจุดเริ่มต้นทางซ้ายสุด โดยสแกนจากบนลงล่างแล้วไล่ไปทางขวา ให้เป็นจุดปัจจุบัน

2 push จุดที่อยู่ข้างเคียงจุดปัจจุบันทั้งหมดลงใน stack พร้อมลบจุดที่ push จากภาพ

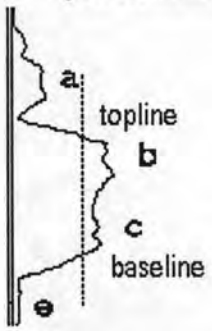
3 if ยังมีจุดอยู่ใน stack do pop จุดขึ้นมากำหนดให้เป็นจุดปัจจุบัน ย้อนทำขั้น 2
else stop

โดยตำแหน่งกรอบของเกาะอักษรเทียบกับภาพบรรทัดที่ตัดมาจะถูกเก็บไว้โดยแทนชื่อตัวแปรคือ border_top แทนขอบบน border_bot แทนขอบล่าง border_right แทนขอบขวา border_left แทนขอบซ้าย

3.1.4 การหาระดับของอักษร

บรรทัดนี้เมื่อทำการ Horizontal projection จะได้ผลดังรูปล่าง

รูปที่ 3.8 แสดงภาพบรรทัดที่ตัดได้ และ Horizontal projection ที่ได้



จากรูปแบบการ horizontal project เป็นรูปภูเขา จะเห็นยอด a เป็นผลมาจากสระบน ส่วน b และ c เป็นผลจากส่วนกลาง และยอด e เกิดจากสระล่าง มี อัลกอริทึมการหาระดับดังนี้

1 เราจะทำการหายอดของภูเขา = max

2 ลากเส้นตัดภูเขา ทำการเก็บช่วงที่สูงกว่า max/N (ค่า N ได้จากการทดลองซึ่งจะตัดภูเขาตรงกลางซึ่งเป็นระดับกลางได้เต็มลูก)

3. หาช่วงที่มีความกว้างที่สุดนั่นคือจะได้ส่วนกลาง

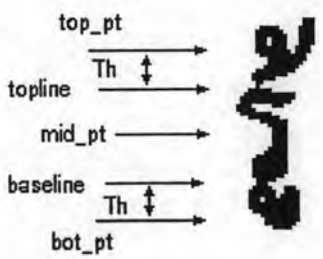
4. จุดตัดทั้ง 2 คือเส้นแบ่งระดับบนและล่าง

ให้ topline = จุดตัดแรกของช่วงที่เลือกและ baseline = จุดตัดหลังของช่วงที่เลือก

แต่จากการแก้ความเอียง เราไม่ได้ทำให้ตรงพอดี ดังนั้นความเอียงเพียงเล็กน้อยจะทำให้อักษรระดับมีพิกัด y ตรงกับอักษรระดับกลาง หรือสระระดับล่างตรงกับอักษรระดับกลาง ดังนั้นการทำ horizontal projection จึงทำในช่วงจำกัดเมื่อเทียบกับเกาะของอักษรที่ดึงออกมาได้ ช่วงจำกัดนี้ต้องสอดคล้องกับเงื่อนไขความเอียง คือ เนื่องจากเราให้เอกสารเอียงได้ไม่เกิน 1 องศาหลังจากแก้ความเอียงแล้ว ดังนั้นช่วงจำกัดต้องไม่ยาวเกินไปเพื่อไม่ให้ระดับบรรทัดผิดมาก และต้องไม่น้อยเกินไปจนหาระดับบรรทัดไม่ได้

3.1.5 การตรวจสอบลักษณะของอักษร

ลักษณะของอักษรจะกำหนดตามตารางที่ 2.2 การตรวจสอบทำได้โดยนำตำแหน่งกรอบภาพเกาะของอักษรที่เทียบกับบรรทัด มาเปรียบเทียบกับเส้นตัดสินใจซึ่งเส้นบนจะสูงกว่าเส้น topline (ที่ได้จากการหาระดับของอักษร) เป็นค่า Th และเส้นตัดสินใจล่างจะต่ำกว่าเส้น baseline เป็นค่า Th เช่นกัน เรียกทั้ง 2 เส้นว่า top_pt และ bot_pt ตามลำดับ ที่ไม่ใช้เส้น topline และ baseline ตรงๆ เพราะถ้าเป็นอักษรบนหรือล่างจริงๆ จะเกินค่าตัดสินใจแน่นอน แต่ถ้าใช้ topline ตรงๆ อักษรอย่างเช่น ส หรือ ศ จะถูกตัดสินว่าเป็นอักษรลักษณะที่ 2 และเส้นตัดสินใจอีกเส้นคือ mid_pt ซึ่งอยู่กลางระหว่าง topline และ baseline



รูปที่ 3.9 แสดงเส้นตัดสินระดับ

เราจะตัดสินว่าเป็นอักษรลักษณะใดตามตารางที่ 2.2 ในบทที่ 2 ด้วยเกณฑ์ดังนี้

เป็นลักษณะที่ 1 เมื่อ border_top สูงกว่า top_pt และ border_bot สูงกว่า mid_pt

เป็นลักษณะที่ 2 เมื่อ border_top สูงกว่า top_pt และ border_bot อยู่ระหว่าง mid_pt และ bot_pt

เป็นลักษณะที่ 3 เมื่อ border_top และ border_bot อยู่ระหว่าง top_pt และ bot_pt

เป็นลักษณะที่ 4 เมื่อ border_bot ต่ำกว่า bot_pt และ border_top อยู่ระหว่าง mid_pt และ top_pt

เป็นลักษณะที่ 5 เมื่อ border_top ต่ำกว่า mid_pt และ border_bot ต่ำกว่า bot_pt

สำหรับอักษรลักษณะที่ 6 นั้นจะให้เป็นลักษณะที่ 5 ไปก่อนแต่จะใช้ประโยชน์ตอนตัดแยกอักษรที่ติด ถ้าเป็นอักษรดังรูปที่ 3.9 คือเป็นทั้งลักษณะที่ 2 3 และ 4 จะส่งไปส่วนตัดแยกพิเศษ

3.1.6 การตรวจสอบความถูกต้องทางภาษา

การตรวจสอบขั้นนี้มีเป้าหมายในการเพิ่มความถูกต้องของการรู้จำ เราใช้แนวทาง 2 อย่าง อย่างแรกคือใช้กฎการผสมคำของอักษรไทย คือ เรียง พยัญชนะ + สระ + วรรณยุกต์ อย่างหลังคือแก้ไขตามความผิดพลาดที่มักเกิดขึ้นเนื่องจากการรู้จำผิด***ซึ่งไม่ใช่กฎแต่เป็นลักษณะเฉพาะของโปรแกรม***

ข้อมูลที่จะนำมาใช้ช่วยในการตรวจสอบคือ

- ตำแหน่งกรอบของเกาะจุดภาพเทียบกับจุดภาพ
- กฎการผสมคำ
- ความน่าจะเป็นของอักษรหรือกลุ่มอักษรที่จะเกิดขึ้น

ตัวอย่างการใช้กฎการผสมคำในการตรวจสอบ เช่น เราสามารถแก้คำผิดอย่างเช่น คำว่า " ขี " จุดภาพของสระอีจะล้ำหน้า " ข " (สามารถตรวจสอบได้จากตำแหน่งกรอบของเกาะจุดภาพเทียบกับจุดภาพของบรรทัดของอักษรทั้ง 2) ทำให้การเรียงเป็น อี + ข ซึ่งผิด

ตัวอย่างการใช้ความผิดพลาดที่มักเกิดขึ้นในการตรวจสอบ (ซึ่งใช้กฎการผสมคำร่วมด้วย) เช่น คำว่า " ขั้ " ไม้โทอาจถูกรู้จำเป็น การันต์ และการที่ไม้หันอากาศตามด้วยการันต์นั้นเป็นไปไม่ได้ตามกฎการผสมคำ ดังนั้นเมื่อพบลักษณะดังกล่าวจะแก้การันต์เป็นไม้โท หรือ อักษร "ฐ" เป็นอักษรที่รู้จำผิดได้ง่าย ศัพท์ที่ใช้บ่อยเช่น "รัฐบาล" ถูกรู้จำเป็น "รัชุบาล" เมื่อตรวจพบคำว่า "รัชุบาล" จะแก้ "ช" เป็น "ฐ"

3.1.7 การสร้าง Text file

เมื่อได้ ASCII code ของบรรทัดที่ตัดมาได้แล้วจึงนำมาเขียนลงแฟ้มข้อมูลตามบรรทัด เมื่อขึ้นคอลัมน์ใหม่ จะเขียนบอกว่าเป็นคอลัมน์ใหม่ ซึ่งสามารถเปิดอ่านได้ด้วยโปรแกรมแก้ไขแฟ้มภาษาไทยทั่วไป เช่น CU Writer หรือ MS. Word 6

3.2 กรรมวิธีในการแยกตัวอักษรที่ติดกัน (Segmentation of Connected Characters)

จากการทดสอบสแกนแบบตัวอักษรต่างๆ กัน(ดังผลที่จะแสดงในบทที่ 4) พบว่าเมื่อใช้สแกนเนอร์ ความละเอียด 300 จุดต่อนิ้ว สแกนตัวอักษรขนาดต่ำกว่า 16 point เกิดตัวอักษรติดกันมากจนมีผลกับอัตราความถูกต้อง ซึ่งเป็นผลมาจากการใช้ความละเอียดของสแกนเนอร์ไม่เพียงพอ (เพราะต้องการความเร็วในการสแกน ลดขนาดแฟ้มข้อมูลที่ต้องการเก็บ ราคาของสแกนเนอร์) ตัวอักษรที่ติดกันนั้นมีรูปแบบหลากหลาย ปัญหาเกิดขึ้นคือ เราจะสามารถรู้ได้อย่างไรว่ากลุ่มของจุดภาพที่เราได้เป็นอักษรที่ติดกันหรือไม่ และถ้าทราบแล้วว่าเป็นอักษรที่ติดกันแล้วจะแยกให้เป็นอักษรเดี่ยวอย่างไร

แนวคิดที่นำเสนอในบทนี้ได้เราได้ประสบการณ์จากการทดลองและพยายามสรุปรูปแบบการติดกันของตัวอักษร พบว่าเราสามารถจัดกลุ่มตัวอักษรที่ติดกันได้ ทำให้สามารถตรวจสอบและแยกตัวอักษรที่ติดกันได้อย่างมีประสิทธิภาพ (สำหรับกรณีที่หน้ากระดาษมีหมึกเลอะไม่มาก การขีดเส้นใต้ ฯลฯ เช่นหน้ากระดาษที่ได้จากการพิมพ์ด้วยเครื่องพิมพ์เลเซอร์หรือหนังสือพิมพ์ที่สะอาด)

ดังนั้นปัญหาใหญ่ของบทนี้ คือ ***การหาวิธีตรวจสอบกลุ่มของจุดภาพว่าเป็นอักษรติดหรือไม่และเป็นกลุ่มใดแล้วตัดแยกตัวอักษรที่ติดกันตามวิธีของแต่ละกลุ่ม***

ข้อมูลที่เราได้รับจากส่วนตัดบรรทัดคือความสูงของบรรทัด ซึ่งทำให้เราประมาณความกว้างตัวอักษรได้เส้นแบ่งระดับของตัวอักษรระดับบน กลางและล่าง ประกอบกับโครงสร้างของตัวอักษรไทย ด้วยข้อมูลเพียงเท่านี้เราต้องทำการตัดแยก ยกตัวอย่างเช่น คำว่า ปี ถ้านับจุดแนวนอนแล้วเราจะพบว่าส่วนบนของจุดภาพมีจำนวนมากเกินไปเพราะไม่มีพยัญชนะไทยตัวใดมีจุดในส่วนบนมาก

ในบทนี้จะกล่าวถึงหลักการในการแบ่งกลุ่มตัวอักษรที่ติดกัน วิธีการตรวจสอบว่าเป็นอักษรติดหรือไม่และเป็นกลุ่มใด วิธีการแยกตัวอักษรออกจากกันของแต่ละกลุ่ม และการส่งข้อมูลให้ส่วนแยกแยะ

สำหรับเครื่องมือที่จะช่วยในการตรวจสอบหาจุดตัดอันหนึ่งคือการนับช่องว่างจากกรอบแต่ละด้านถึงตัวอักษรนั้นจะอยู่ในส่วนรู้จำตัวเดี่ยว

3.2.1 ประเภทของตัวอักษรที่ติดกัน

จากตัวอย่างของตัวอักษรที่ติดกันที่ได้การทดลอง เราพบว่าการแบ่งกลุ่มของอักษรที่ติดกันนั้น การแบ่งด้วยลักษณะของตัวอักษรเป็นวิธีแบ่งที่ดีด้วยเหตุผลดังนี้

- เนื่องจากมีอักษร 6 ลักษณะ ดังนั้นเมื่อจับคู่ระหว่างลักษณะจะมีจำนวนกลุ่มจำนวนจำกัด
- เนื่องจากแต่ละกลุ่มมีลักษณะเฉพาะทำให้ตรวจสอบว่าเป็นกลุ่มใดและพิจารณาจุดที่จะตัดได้ง่าย

การจับคู่ทั้งหมดคือ เลือก 2 ลักษณะจาก 6 ลักษณะและสามารถสลับได้ แต่ เราพบว่าหลายคู่นั้นไม่สามารถติดกันได้เช่น ระดับบน(เช่น ไม้เอก) กับระดับล่าง(เช่น สระอู) ทำให้กลุ่มเหลือแค่ 9 กลุ่ม และอีกกลุ่มสำหรับแบบที่ซับซ้อนมากกว่า 2 ระดับที่ติดกัน

เราสามารถแบ่งประเภทตามลักษณะ (ที่แสดงในตารางที่ 2.2) ของตัวอักษรที่ติดกัน ดังตารางที่ 3.2 และแสดงร้อยละที่พบสำหรับแบบอักษรต่างๆ ที่ขนาดต่างๆ กัน ในตารางที่ 3.3

ตารางที่ 3.2 แสดงแบบของตัวอักษรที่ติดกัน และตัวอย่าง

แบบที่	อักษรลักษณะที่ติดกัน	ตัวอย่าง
1	1 ติด 1	บัติ ค์กิ ชีวิ
2	1 ติด 2 (อยู่คอลัมน์เดียวกัน)	ปี ปี
3	1 ติด 2 (อยู่คนละคอลัมน์ 1 อยู่หน้า)	ร์ไ ร์ไ
4	1 ติด 2 (อยู่คนละคอลัมน์ 2 อยู่หน้า)	ปรี
5	1 ติด 6	อี่
6	1 ติด 3	ด้ ศี สั ช่ ย์
7	2 ติด 3	ไม
8	3 ติด 5	นุ ญ ดู ผู รู ฐ
9	3 ติด 3	รท เภ นภ รท รา รบ รข รจ ยย
10	ที่ติดกันมากกว่า 2 เช่น แบบที่ 2 รวมกับ 4 แบบที่ 6 รวมกับ 7 แบบที่ 1 รวมกับ 2 รวมกับ 4	 สี่ รูป มั้สี่

ตารางที่ 3.3 ร้อยละที่พบของแบบของตัวอักษรที่ติดกัน

อักษรติด	AngsanaUPC				CordiaUPC			
แบบที่	12	14	16-18	20-24	12	14	16-18	20-24
1	2.5	0.2	2.1	-	4.2	6.5	2.1	0
2	10.7	14.5	39.0	-	9.9	20.1	17.5	100
3	3.7	9.2	1.5	-	3.1	1.8	3.8	0
4	0.2	0.2	0.2	-	0.2	0.1	0.3	0
5	15.5	38.1	0.2	-	16.2	11.6	6.2	0
6	13.7	5.9	28.1	-	14.1	17.4	16.5	0
7	0.1	0.2	0.2	-	3.2	0.1	0.2	0
8	52.9	31.0	28.2	-	24.9	34.8	51.1	0
9	0.1	0.2	0.2	-	14.1	5.4	0.2	0
10	0.6	0.5	0.5	-	10.1	1.2	2.1	0

**หมายเหตุ ข้อมูลได้จากอักษรทดสอบในตารางที่ 4.1 (อักษรที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์)*

ตารางที่ 3.4 แสดงจำนวนอักษรที่ทำการทดสอบและที่ติดสำหรับอักษรแบบและขนาดต่าง ๆ

	AngsanaUPC				CordiaUPC			
	12	14	16-18	20-24	12	14	16-18	20-24
จำนวนอักษรทดสอบ(ตัว)	6714	1356	3380	9242	3156	4145	6263	10995
จำนวนอักษรที่ติด(ตัว)	510	65	44	0	258	174	88	22
% ที่ติด	7.6	4.8	1.3	0.0	8.2	4.2	1.4	0.2

**หมายเหตุ ข้อมูลได้จากอักษรทดสอบในตารางที่ 4.1 (อักษรที่พิมพ์ด้วยเครื่องพิมพ์เลเซอร์)*

3.2.2 ขั้นตอนการตรวจสอบว่าเป็นอักษรที่ติดกันหรือไม่

เมื่อเราได้เกาะของอักษรและเส้นแบ่งระดับอักษรแล้วเราสามารถแบ่งลักษณะของอักษรได้เป็น 6 ลักษณะตามตารางที่ 2.2

รูปที่ 3.10 แสดงตัวอย่างเกาะของอักษรที่ติดกันแบบต่างๆ

จะเห็นได้ว่าการติดกันของอักษรทำให้ลักษณะที่ได้ผิดไป เช่น เมื่อลักษณะที่ 3 ติดกับลักษณะที่ 1 ทำให้ดูเป็นลักษณะที่ 2 ดังนั้นเมื่อเราได้ลักษณะจากส่วนโปรแกรมตรวจสอบลักษณะแล้วเราจะทราบว่าถ้าเป็นอักษรปกติจะเป็นอักษรใดบ้างและถ้าเป็นอักษรที่ติดกันจะมีต้นเหตุความผิดพลาดแบบใด (ตามที่แบ่งตามตารางที่ 3.2) ซึ่งเป็นข้อมูลสำคัญที่จะให้กับโปรแกรมส่วนพิจารณาว่าเกาะของอักษรที่ได้เป็นอักษรติดหรือไม่ ดังแสดงในตารางที่ 3.5

ตารางที่ 3.5 แสดงว่าลักษณะที่ได้สามารถเป็นอักษรปกติแบบใดและเป็นอักษรติดแบบใดได้

ลักษณะที่ได้	ความผิดพลาดต้นเหตุ (แบบตามในตารางที่ 3.2)	อักษรปกติ
1	1,5,10	๑๘๔๕ ๑๒๗๑๘๑๙๐
2	2,3,4,6,7,10	ปฝฟไไไ ๒๖๗๘
3	9,10	อักษรระดับ 3
4	8,10	ฤฦฦฦฐๆ
5	-	ๆ ๆ

สรุปผลได้ว่าถ้าตรวจสอบได้ว่าเป็น

ลักษณะที่ 5 ส่งไปส่วนรู้จำตัวเดี่ยวได้ทันที

ลักษณะที่ 4 ส่งไปส่วนตัดแยกของลักษณะที่ 4 นำส่วน*ล่าง*ที่เหลือไปส่วนรู้จำตัวเดี่ยว ส่วนกลางที่เหลือไปส่วนตัดแยกของลักษณะที่ 3

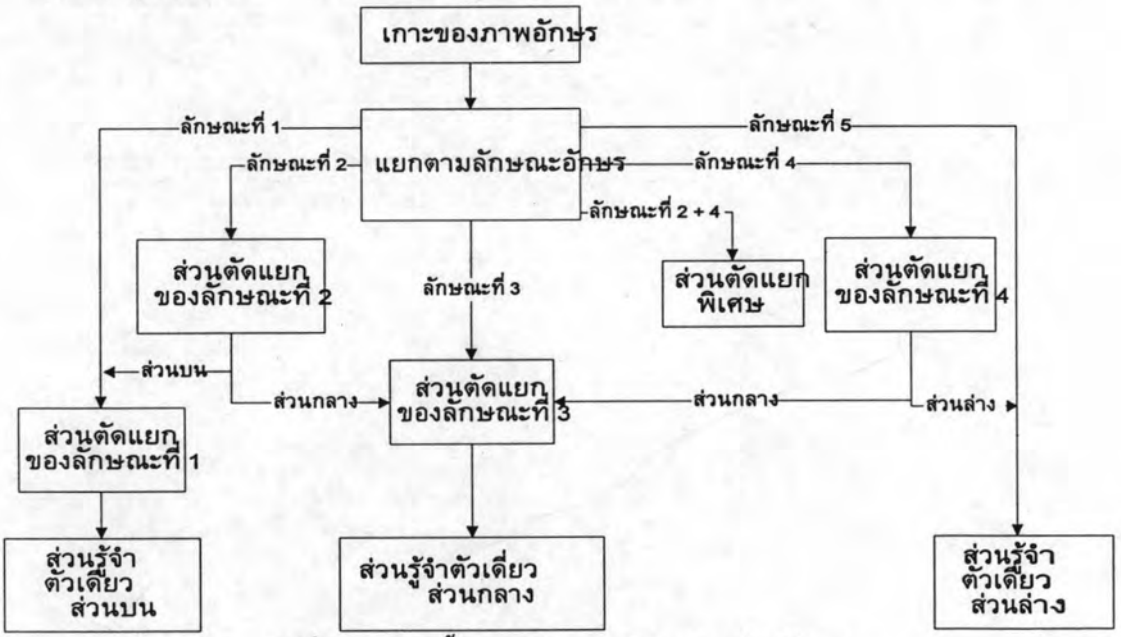
ลักษณะที่ 2 ส่งไปส่วนตัดแยกของลักษณะที่ 2 นำส่วน*บน*ที่เหลือไปส่วนตัดแยกของลักษณะที่ 1 ส่วนกลางที่เหลือไปส่วนตัดแยกของลักษณะที่ 3

ลักษณะที่ 1 ตัดแยกด้วยส่วนตัดแยกของลักษณะที่ 1 แล้วส่งไปส่วนรู้จำตัวเดี่ยว

ลักษณะที่ 3 ตัดแยกด้วยส่วนตัดแยกของลักษณะที่ 3 แล้วส่งไปส่วนรู้จำตัวเดี่ยว

ลักษณะที่ 6 ตัดแยกด้วยส่วนตัดแยกของลักษณะที่ 4 และ 2 ตามลำดับ

แสดงดังรูปที่ 3.11



รูปที่ 3.11 แสดงขั้นตอนหลักๆ การทำงานของส่วนตัดแยก

จากขั้นตอนจะเห็นได้ว่ามีส่วนทำงานหลักๆ คือส่วนตัดแยกของลักษณะที่ 1 2 3 4 และ พิเศษ ซึ่งมีหลักการทำงานเหมือนๆ กัน

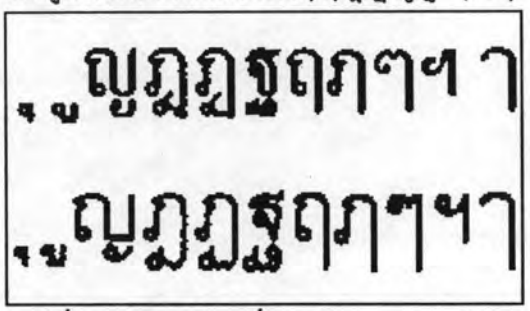
ขั้นตอนการทำงานหลักของส่วนตัดแยกคือ

1 หาว่าติดหรือไม่และแบบใด

2 หาจุดที่จะตัด

3.2.3 ส่วนตัดแยกของลักษณะที่ 4

สระล่างมี 2 ตัวคือ อุ และ อู ส่วนอักษรที่ปนมาจะมี ฤ ฦ ฏ ฎ ญ ฐ ๆ ฯ ๅ



รูปที่ 3.12 แสดงอักษรที่จะถูกตัดแยกของระดับ 4

ดังนั้นในการพิจารณาเราจะต้องแยกเป็นประเภทดังนี้

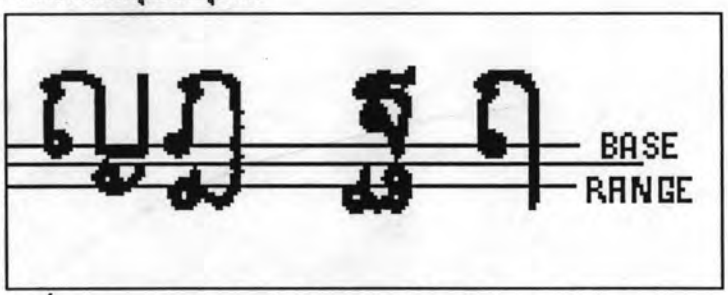
1 หางตรง ได้แก่ ฤ ฦ ๆ ๅ

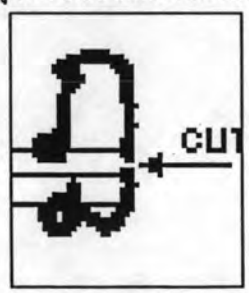
2 สระ อุ อู

3 ส่วนประกอบของอักษรอื่น (ที่ไม่ใช่หางตรง)

ขั้นตอนการแยกแยะดังนี้

1 ตรวจสอบความกว้างของภาพอักษรถ้ามีส่วนแคบ(ซึ่งจะเป็นส่วนเส้นตรงในแนวดิ่งของ "ๆ หรือ ๅ") มากกว่าส่วนกว้างตัดสินว่าเป็น ส่งไปให้ส่วนรู้จำตัวเดี่ยว

2 นับช่องว่างทางซ้ายของภาพในช่วงพิจารณาจากซ้ายไปจนพบจุดภาพ ช่วงพิจารณาคือบริเวณ baseline ให้ช่องว่างทางซ้ายที่ลึกที่สุดเป็นจุดตัด

รูปที่ 3.13 แสดงการหาจุดตัดโดยอาศัยเส้นระดับล่างและช่วงการพิจารณา



รูปที่ 3.14 แสดงการตัดแยกระดับกลางกับล่างออกจากกัน

การตัดในแนวนอนเป็นวิธีที่ง่ายแต่เกิดปัญหาบางอย่าง โปรดดูรูปจะเห็นว่าหางของ ฎ ฏ ของแบบอักษร AngsanaUPC ล้ำเข้าไปในตัวอักษรซึ่งเราต้องทำการลบทิ้งก่อนที่จะส่งส่วนที่เหลือข้างบนไปให้ส่วนรู้จำตัวเดี่ยว

รูปที่ 3.15 แสดงปัญหาที่เกิดจากการตัดแยก และวิธีแก้ไข

เมื่อตัดเรียบร้อยแล้วจึงส่งส่วนบนและส่วนล่างไปยังส่วนแยกแยะตัวเดี่ยวโดยยังเก็บข้อมูลที่ว่า 2 ส่วนนี้เคยติดกันไว้ด้วย เช่น ถ้าตรวจสอบได้ว่าส่วนล่างเป็นส่วนล่างของ ฎ และส่วนบนเป็น ภ จะให้คำตอบเป็น ฎ แต่ถ้าตรวจสอบส่วนล่างได้ว่าเป็นสระ อู และส่วนบนเป็น ภ จะให้คำตอบเป็น ภ

3.2.4 ส่วนตัดแยกของลักษณะที่ 3

การตรวจสอบว่าติดหรือไม่

การติดกันของลักษณะที่ 3 สามารถดูอย่างคร่าวๆ ได้ด้วย aspect ratio = ความกว้างของภาพ / ความสูงของภาพ ซึ่งอักษรที่ aspect ratio มากที่สุดคือ ฒ ณ ญ ฌ เกณฑ์คือถ้ากว้างมากกว่าอักษรที่กว้างที่สุดให้ตอบว่าเป็นอักษรติด

ปัญหาเกิดขึ้นสำหรับกรณีเช่น แ รว แคบกว่าเกณฑ์ เราใช้วิธีส่งไปให้ส่วนแยกแยะตัวเดี่ยวเลยโดยถือให้อักษรติดดังกล่าวเหมือนอักษรเดี่ยวตัวหนึ่ง แต่ผลการแยกแยะตัวเดี่ยวจะชี้ว่าเป็นแบบอักษร 2 ตัวติดกัน ข้อเสียคือทำให้จำนวน output ของตัวแยกแยะตัวเดี่ยวเพิ่มขึ้น

การหาจุดตัด

เราเลือกวิธีที่ง่ายและได้ผลดีของ Kahan and Pavlidis (1987) ซึ่งเขียนไว้ว่า "จุดเชื่อมของ 2 อักษรจะมีค่าของ Vertical Projection (V (x))เปลี่ยนแบบ Sharp minimum และทั้ง 2 คนเสนอว่าให้ใช้อัตราส่วนระหว่าง อนุพันธ์อันดับ 2 คือ V (x - 1) - 2 V (x) + V (x + 1) กับค่าของ Projection เป็นสมการเงื่อนไขในกาหาจุดตัด"

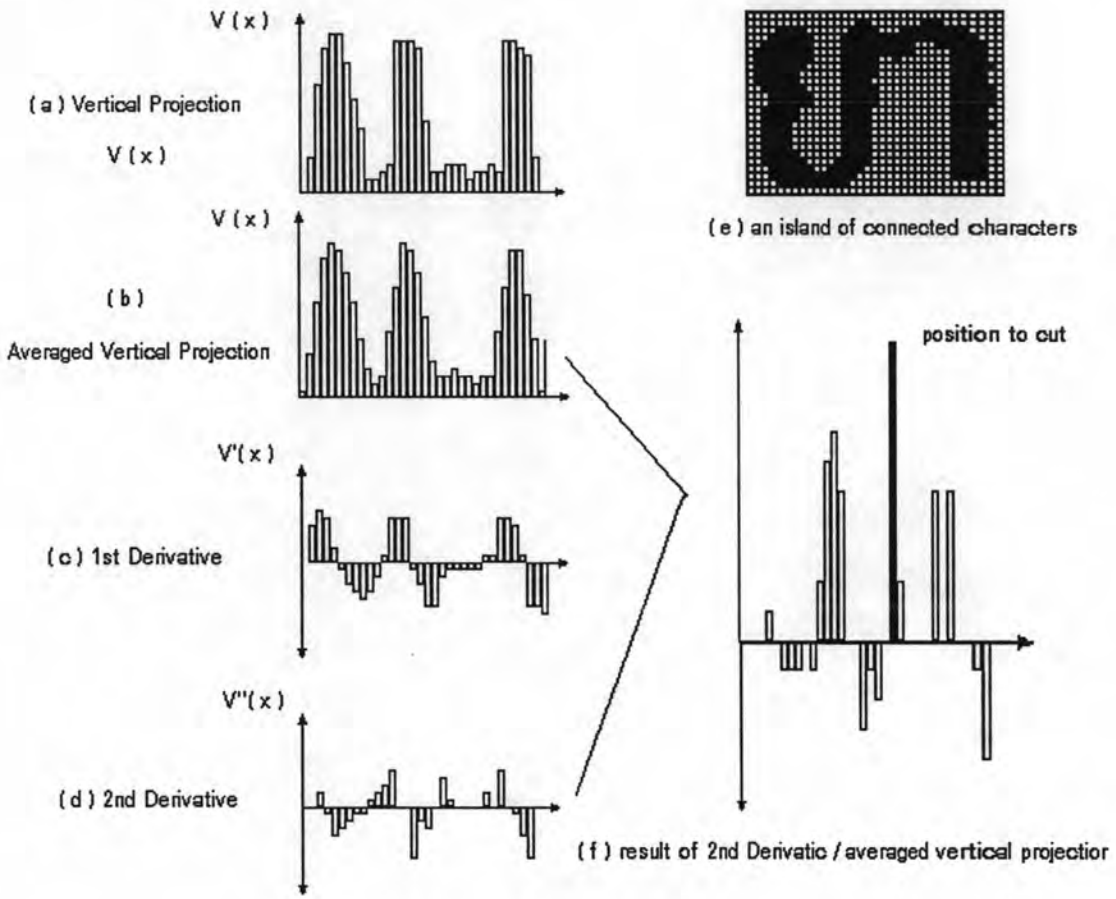
แต่เนื่องจากถ้าใช้ Vertical Projection โดยตรงจะให้ผลไม่ดีเนื่องจากสัญญาณรบกวน ดังนั้นเราจึงทำการเฉลี่ยค่า V (x) ที่ได้กับค่าข้างเคียงดังสมการ

$$\text{new } V(x) = \frac{V(x-1) + V(x) + V(x+1)}{3}$$

เราจึงหาจุดตัดได้ด้วยสมการ

$$Cut_position = \max\left(\frac{V(x-1) - 2*V(x) + V(x+1)}{V(x)}\right)$$

จากการทดลอง เราได้ผลดังรูปที่ 3.16



รูปที่ 3.16 แสดง (a) Vertical Projection ของอักษรที่ติด (b) Averaged Vertical Projection
(c) อนุพันธ์อันดับ 1 (d) อนุพันธ์อันดับ 2
(f) ผลการหารที่ได้จากสูตร เทียบกับ (e) อักษรที่ติด

สมการนี้สามารถแปลความหมายได้ว่า ค่าในวงเล็บจะมากก็ต่อเมื่อมีค่า Vertical Projection น้อยซึ่งถูกต้องเพราะจุดที่เกิดอักษรติดจะมีจุดภาพไม่มาก ส่วนอนุพันธ์อันดับ 1 นั้นแสดงการเปลี่ยนแปลงของจำนวนจุดในแนวดิ่งซึ่งมีค่าบวกมากเมื่อมีจุดเพิ่มมาก โปรดดูรูปที่ 3.16 (c) เห็นว่าอนุพันธ์อันดับ 1 มีค่ามากตรงจุดก่อนขาที่ 1 และขาที่ 2 ของ ย.ยักษ์ และก่อนขาของสระอา เมื่อหาอนุพันธ์อันดับ 2 แล้วอนุพันธ์อันดับ 2 จะมีค่ามากเมื่ออนุพันธ์อันดับ 1 เปลี่ยนจากน้อยไปมาก นั่นคือจากจำนวนจุดในแนวดิ่งเพิ่มน้อย เป็น จำนวนจุดในแนวดิ่งเพิ่มมาก จำนวนจุดในแนวดิ่งเพิ่มน้อยถ้าเราลองดูจากรูปที่ 3.16 (e) จะเห็นว่าเป็นส่วนเชื่อมโครงร่างคือส่วนเชื่อม ขาที่ 1 กับขาที่ 2 ของ ย.ยักษ์ และจากขาที่ 2 ของ ย.ยักษ์กับสระอา ดังนั้นจุดตัดจะต้องพร้อมด้วย 2 ประการคือ มีจุดภาพไม่มากและใกล้กับขาของอักษร

ถ้ามีอักษรติดกันมากกว่า 2 ตัว (ซึ่งสามารถตรวจได้จากดู Aspect ratio)จะทำการแยกจุดที่น่าจะตัดที่สุดก่อนแล้วนำส่วนที่ตัดได้มาตรวจสอบ ratio แล้วตัดจนไม่เหลืออักษรที่ติด

3.2.3 ส่วนตัดแยกของลักษณะที่ 2

อักษรที่เข้ามาในส่วนนี้คือ

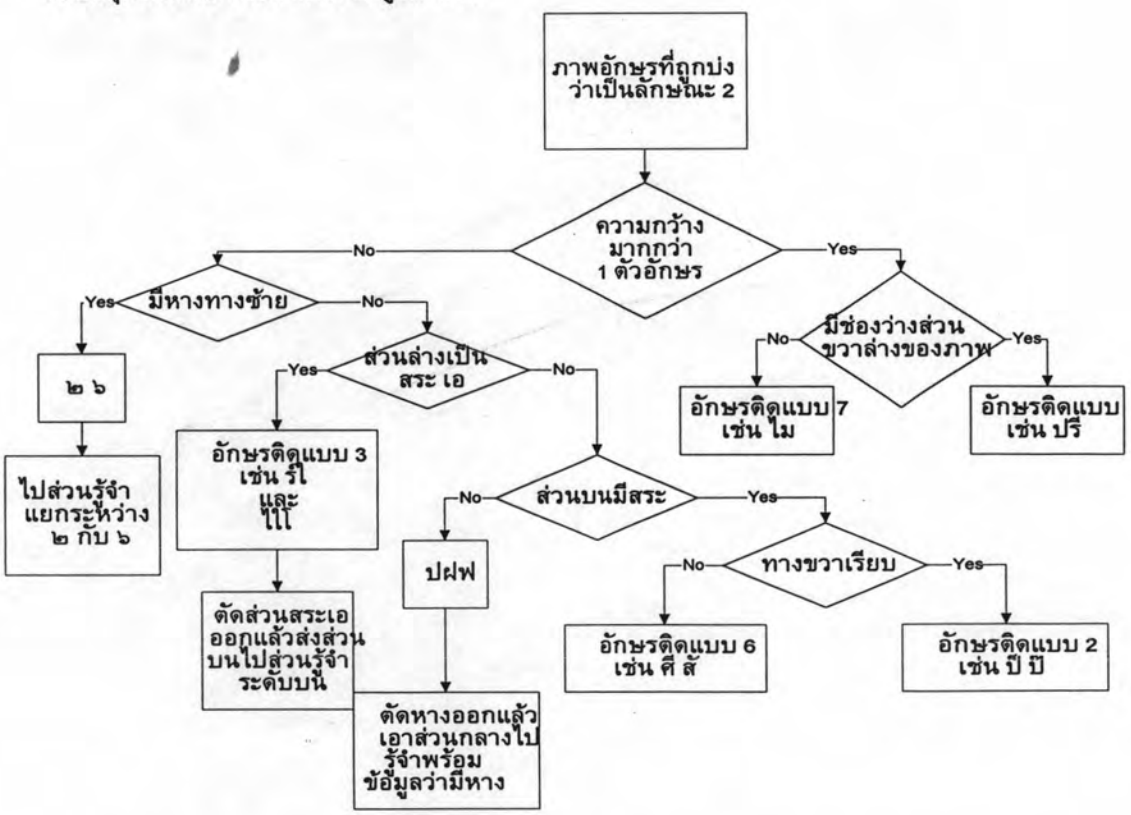
อักษรที่ติด

1. แบบที่ 2 เช่น ป๊ ป๊
2. แบบที่ 3 เช่น ร์ไ
3. แบบที่ 4 เช่น ปรี
4. แบบที่ 6 เช่น ศี สั
5. แบบที่ 7 เช่น ไม

อักษรที่เดี่ยว แบ่งเป็น 3 กลุ่มย่อย คือ

1. แบบมีหางทางซ้าย คือ ๒๖
2. มีหางทางขวา คือ ปฝฟ
3. ไไโ

ดังนั้นเราต้องหาลักษณะบ่งความต่างระหว่างอักษรเหล่านี้ และหาจุดตัดตามกลุ่ม ขั้นตอนการตรวจว่าเป็นกลุ่มใดทำตาม flow chart ในรูปที่ 3.17



รูปที่ 3.17 แสดง flow chart การแบ่งกลุ่มของอักษรที่เข้ามาในส่วนแยกแยะของลักษณะที่ 2

3.2.3 ส่วนตัดแยกของลักษณะที่ 2

อักษรที่เข้ามาในส่วนนี้คือ

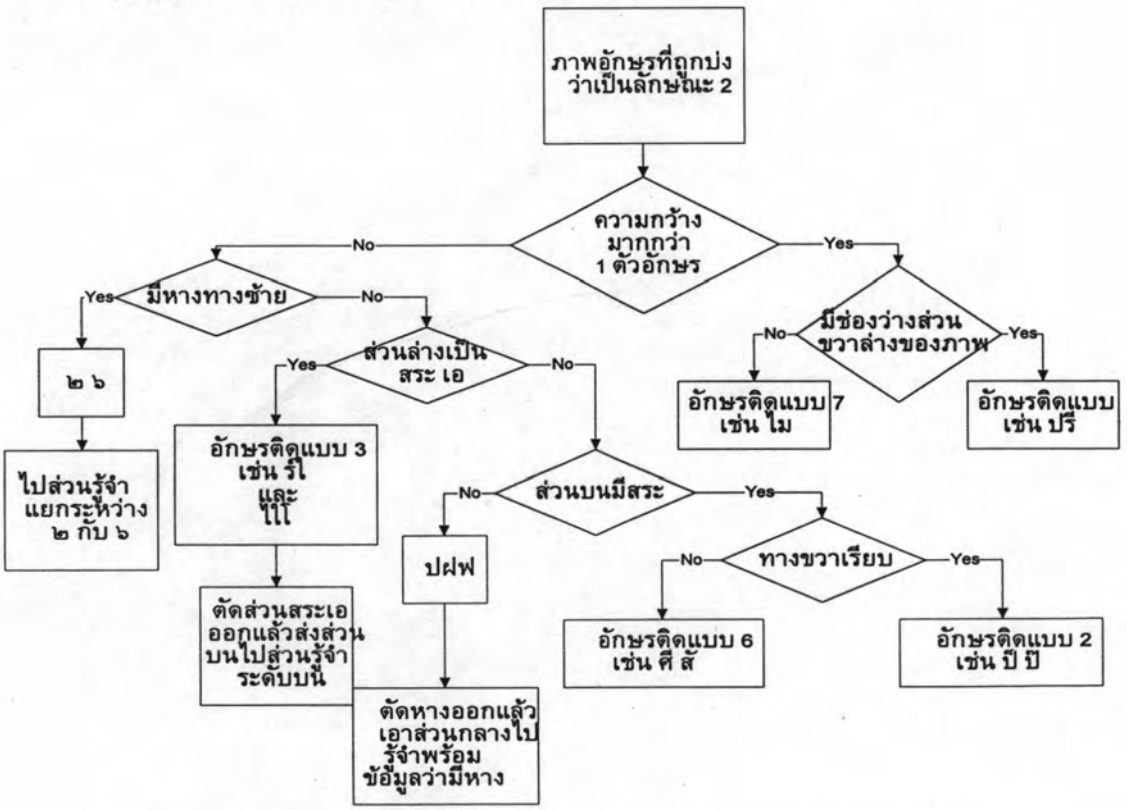
อักษรที่ติด

1. แบบที่ 2 เช่น ปี ปี
2. แบบที่ 3 เช่น ร์ไ
3. แบบที่ 4 เช่น ปรี
4. แบบที่ 6 เช่น ศี สั
5. แบบที่ 7 เช่น ไม

อักษรที่เดี่ยว แบ่งเป็น 3 กลุ่มย่อย คือ

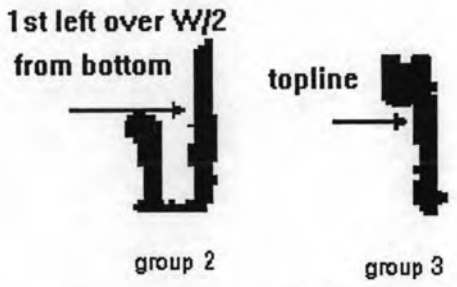
1. แบบมีหางทางซ้าย คือ ๒๖
2. มีหางทางขวา คือ ปฝฟ
3. ไไโ

ดังนั้นเราต้องหาลักษณะบ่งความต่างระหว่างอักษรเหล่านี้ และหาจุดตัดตามกลุ่ม ขั้นตอนการตรวจว่าเป็นกลุ่มใดทำตาม flow chart ในรูปที่ 3.17



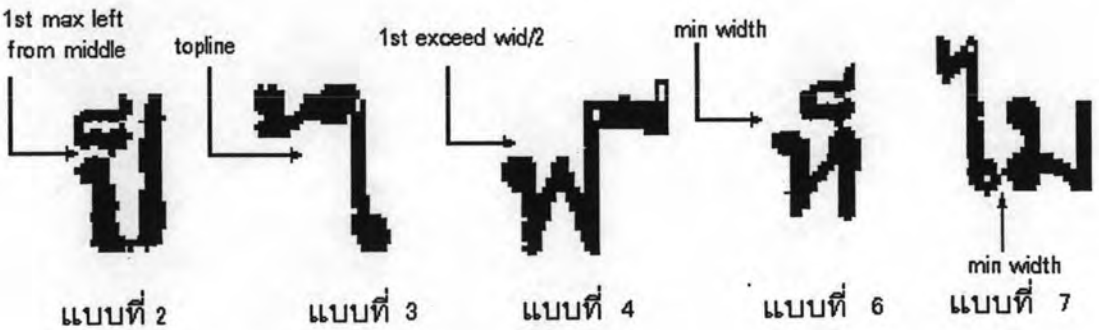
รูปที่ 3.17 แสดง flow chart การแบ่งกลุ่มของอักษรที่เข้ามาในส่วนแยกแยะของลักษณะที่ 2

- ตรวจสอบว่ามีมากกว่า 1 ตัวอักษรโดยเทียบความกว้างภาพอักษรกับความสูงของบรรทัดโดยใช้ค่าของอักษรเดี่ยวที่กว้างที่สุด(คือ ฒ ฌ ณ ญ) เป็นเกณฑ์
- ตรวจว่ามีหางทางซ้าย ทางขวา หรือส่วนบนมีสระหรือไม่โดยดูการกระจายของจุดที่เหนือ topline
- ตรวจว่าส่วนล่างเป็นสระเอหรือไม่โดยดูความกว้างของจุดที่อยู่ระหว่าง topline กับ baseline เทียบกับความสูงของบรรทัด
- เราตัดแยกอักษรเดี่ยวกลุ่มย่อยที่ 2 คือ ป ฝ ฟ โดยตัดเฉพาะหางออกโดยจะตัดที่ช่องทางซ้ายแรกจากล่างที่ยาวถึงหาง ดังรูปที่ 3.18
- เราตัดแยกอักษรเดี่ยวกลุ่มย่อยที่ 3 คือ ไ ใ โ โดยตัดที่ topline



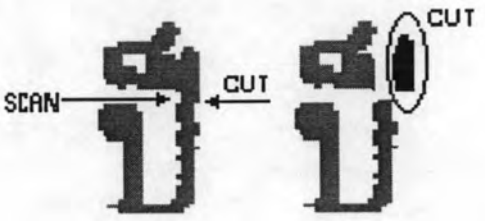
รูปที่ 3.18 แสดงจุดตัดแยกของอักษรเดี่ยวกลุ่มย่อยที่ 2 และ 3

- ตัดแยกตัวติดแบบที่ 2 โดยใช้ช่องว่างทางซ้ายที่มากที่สุดบริเวณเส้น topline
- ตัดแยกตัวติดแบบที่ 3 โดยใช้ topline แล้วส่งส่วนที่เหลือให้ส่วนตัดแยกลักษณะที่ 1 โดยให้ข้อมูลส่วนล่างเป็นสระเอ
- ตัดแยกตัวติดแบบที่ 4 โดยใช้ช่องว่างทางซ้ายแรกจากล่างที่ยาวถึง W / 2
- ตัดแยกตัวติดแบบที่ 6 โดยใช้จุดที่มีความกว้างน้อยที่สุดบริเวณ topline
- การตัดแยกตัวติดแบบที่ 7 จะไม่ตัดแนวนอนแต่จะตัดตามแนวดิ่งโดยหาจุดที่มีความหนาในแนวตั้งน้อยที่สุดบริเวณส่วนหน้าของภาพ ส่งส่วนแรกไปส่วนตัดแยกของลักษณะที่ 2 ส่งส่วนหลังไปส่วนตัดแยกของลักษณะที่ 3



รูปที่ 3.19 แสดงการหาจุดตัดของอักษรติดแบบต่างๆ

สำหรับแบบที่ 2 เราจะทำการตัดหางของ ป ฝ หรือ ฟ ออกก่อนโดยประมาณความกว้างของหางจากรอยตัดแล้วตัดตามความกว้างนั้นดังรูปที่ 3.20

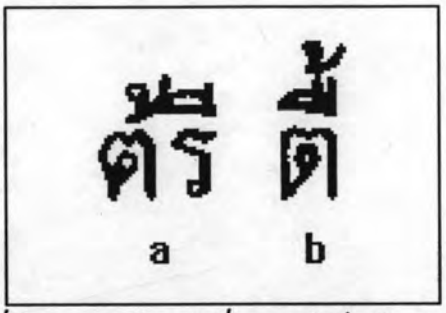


รูปที่ 3.20 แสดงการตัดแยกอักษรที่มีหางกับสระหรือวรรณยุกต์

3.2.4 ส่วนตัดแยกของลักษณะที่ 1

แบ่งจากที่มา 2 อย่างคือ อักษรระดับบน และที่เหลือจากการตัดแยกตัวอักษรติดแบบที่ 3

3.2.4.1 การตัดแยกอักษรติดระดับบน ซึ่งมี 2 แบบคือ 1 และ 5



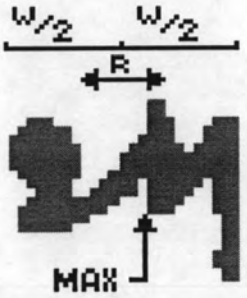
รูปที่ 3.21 แสดงอักษรติดที่ถูกระบุว่าเป็นลักษณะที่ 1
(a) แบบ 1 (ความกว้างเกินปกติ) (b) แบบที่ 5 (ความสูงเกินปกติ)

ตรวจสอบว่าเป็นแบบ 1 แบบ 5 หรือเป็นแบบปกติโดยใช้ความสูงและความกว้างของภาพอักษรเทียบกับความสูงของบรรทัด โดยแบบที่ 1 จะมีความกว้างมากกว่าปกติ แบบที่ 5 จะมีความสูงมากกว่าปกติ ถ้าไม่ต่างจากปกติให้ส่งไปส่วนรู้จำอักษรระดับ 1 ปกติ

- การตัดแยกอักษรติดแบบที่ 1 ทำได้ยากเพราะมีการซ้อนกันทำให้ข้อมูลหายไปและถ้าจะตัดให้ได้ผลดีจริงจะต้องตัดแนวทแยง แต่จากการทดสอบเมื่อเราใช้การตัดครึ่ง (เนื่องจากอักษรที่ติดกันในแนวนอนจะมีความกว้างใกล้เคียงกัน) ให้ผลการรู้จำถูกต้องพอสมควร (โปรดดูผลจากบทที่ 4)
- การตัดแยกอักษรติดแบบที่ 5 ใช้ตรวจหาส่วนที่แคบที่สุดบริเวณที่เป็นความสูงปกติของสระบน

3.2.4.2 การแยกภาพที่เหลือจากการตัดแยกอักษรติดแบบที่ 3

ส่วนที่เหลือมากจากการตัดตามแนวระดับนั้นมีลักษณะดังนี้



รูปที่ 3.22 แสดงการตัดแยกสระหรือวรรณยุกต์ติดกับ สระ ไ ไ โ

ในการหาจุดตัดตามแนวแกนตั้งเราจะเลือกช่วงการพิจารณาโดยให้อยู่ช่วงกึ่งกลางภาพเพราะความกว้างของ สระ โ ไ ไ ใกล้เคียงกับสระที่ติด เราจะนับช่องว่างจากข้างล่างของภาพในช่วงพิจารณาและ

"จุดตัด = ช่องว่างจากข้างล่างที่มีค่ามากที่สุดในช่วงพิจารณา"

เหตุผลที่ใช้อย่างนี้เนื่องจาก สระ ไ ไ โ นั้นมีปลายส่วนหน้าเป็นลักษณะเส้นในแนวตั้งและสระหรือวรรณยุกต์ด้านหน้าที่จะมาติดนั้นจะมีหาง ซึ่งหางจะมีความหนาน้อยกว่าปลายของสระ ไ ไ โ ตัวอย่างดังรูปที่ 3.22 หางของไม้โทแคบกว่าหางของสระ ไ

3.2.5 ส่วนตัดแยกพิเศษ

สำหรับกรณีที่เป็นทั้งระดับ 2 และ 4 จะส่งให้ส่วนตัดระดับ 4 ก่อนให้แยกส่วนล่างออก เพราะส่วนล่างมีความซับซ้อนน้อยกว่า แล้วเอาผลที่รู้จำได้เก็บไว้รวมกับผลที่ได้จากการตัดของระดับ 2

3.3 การแยกแยะอักษรเดี่ยว (Classification of Single Character)

จากบทที่ 2 ซึ่งกล่าวถึงงานที่ผ่านมาทำให้เราเห็นว่าการรู้จำที่ใช้วิธี SyntPR มีข้อได้เปรียบที่ได้ใช้ลักษณะการเขียนอักษรนั่นคือความสัมพันธ์ระหว่างเส้นที่ประกอบกันเป็นอักษร แต่เนื่องจากงานของ สนธยา เมรินทร์ (2537) มีข้อจำกัดคือ ต้องมีต้นแบบจำนวนเท่ากับจำนวนอักษรทั้งหมดทำให้เสียเวลาและหาส่วนประกอบของโครงสร้าง (เรียก primitive) ได้ยากเมื่ออักษรมีสัญญาณรบกวนมาก เช่น หัวของอักษรจะไม่มีรู ดังนั้นจะหา primitive วงกลมไม่เจอ เมื่อเราสังเกตจากอักษรไทยทั้งหมดพบว่าสามารถแบ่งเป็นกลุ่มที่มีโครงสร้างหยาบๆ เหมือนกันได้ จากรูปที่ 3.23 ถ้าใช้จำนวนเส้นแนวดิ่ง หลังคา และลักษณะพื้น เป็นลักษณะหลักในการแบ่งกลุ่ม จะเห็นได้ว่าอักษรในแนวนอนมีลักษณะหลักๆ หลายอย่างเหมือนกันแม้จะต่างอักษรและต่างแบบอักษรกัน เช่น แถวแรกมีเส้นแนวดิ่ง 1 เส้นเหมือนกัน แถวที่ 2 มีเส้นแนวดิ่ง 2 เส้นและมีหลังคาไม่มีพื้น แถวที่ 3 มีเส้นแนวดิ่ง 2 เส้น ไม่มีหลังคาและพื้น แถวที่ 4 มีเส้นแนวดิ่ง 2 เส้น มีพื้นเรียบและไม่มีหลังคา

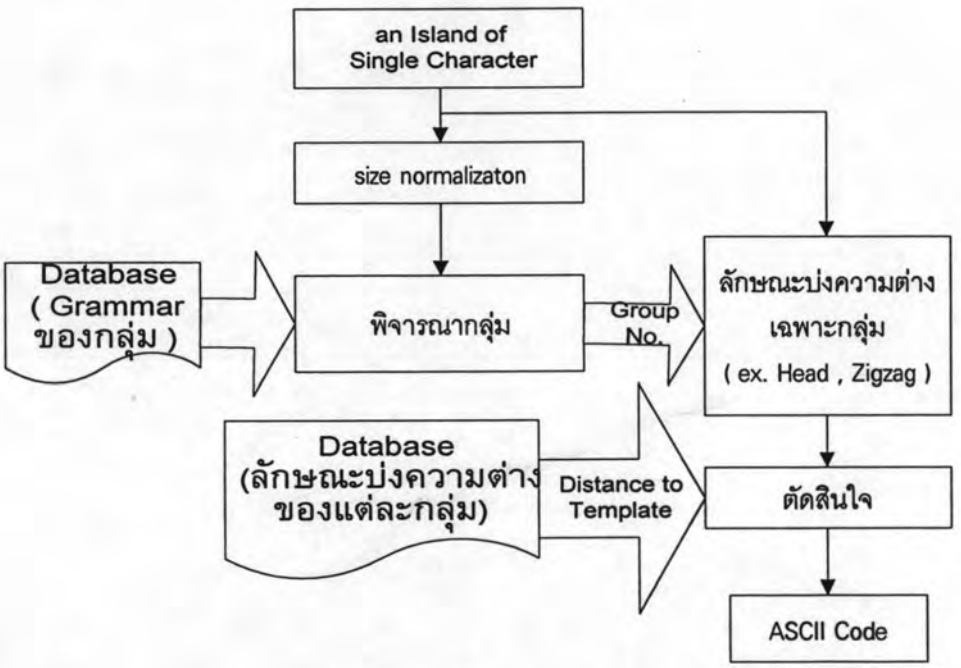
รูปที่ 3.23 แสดงลักษณะโครงสร้างหลักที่เหมือนกันระหว่างหลายๆ แบบอักษร

แนวคิดที่เสนอคือ ลดจำนวนอักษรต้นแบบโดยมีการแบ่งกลุ่มก่อนแล้วจึงแบ่งกลุ่มย่อยโดยใช้ลักษณะบ่งความต่างของภาษาไทย เมื่อเราสามารถทราบว่าอักษรที่จะพิจารณาอยู่ในกลุ่มใดแล้ว การแยกอักษรในกลุ่มกันเองทำได้ง่ายขึ้น เช่นการพิจารณาแยกแยะ ล กับ ส เพราะเราทราบบริเวณที่ควรจะพิจารณา และควรจะพิจารณาอะไร ในกรณีนี้ หัวไม่ใช่สิ่งสำคัญ ที่สำคัญคือมีหางหรือไม่เท่านั้น ซึ่งจะเรียกกระบวนการในส่วนนี้ว่าส่วนหาลักษณะบ่งความต่าง

การแบ่งกลุ่มเราอาศัยโครงสร้างหลัก โดยเน้นว่าส่วนประกอบของโครงสร้างหลักต้องหาง่าย ทนต่อสัญญาณรบกวน (เพื่อแก้ข้อจำกัดของเดิม) ทำให้จำนวน Grammar ที่ต้องใช้น้อยซึ่งมีผลให้เร็วขึ้นด้วย

ข้อมูลจุดภาพขาเข้าจะเข้ามาในลักษณะเกาะของจุดต่อเนื่อง ดังนั้นกรณีอักษรที่มีชิ้นส่วนมากกว่า 1 จึงต้องมีกรณีพิเศษในการพิจารณาเช่น ญ ฐ หรือ ะ (ญ กับ ฐ มักจะมีจุดภาพต่อเนื่องเป็นชิ้นเดียวกันทำให้ต้องพิจารณาทั้งเป็นแบบ ชิ้นส่วนมากกว่าและเท่ากับ 1) ซึ่งสามารถทำได้โดยกำหนดรหัสของแต่ละส่วนเป็นรหัสพิเศษ เช่น ญ ให้ระดับกลางเป็น Suspect_up_ญ ให้ระดับล่างเป็น Suspect_down_ญ เมื่อรู้จำหมดบรรทัดแล้วจึงมาตรวจสอบว่าถ้ามี Suspect_up_ญ + Suspect_down_ญ ให้รวม ASCII เป็น ญ ตัวเดียว ถ้าไม่ใช่เช่น Suspect_up_ญ + ก.ไก่ แสดงว่าไม่มีอะไรอยู่ได้ ญ แสดงว่ารู้จำ ญ ผิด อักษรที่ใกล้เคียงคือ ฌ จึงให้แก้ ASCII เป็น ฌ แทน

ขั้นตอนการทำงานของส่วนรู้จำหลักๆ แสดงดังรูปที่ 3.24



รูปที่ 3.24 แสดงขั้นตอนหลักในการรู้จำอักษรเดี่ยว

จากขั้นตอนการตัดแยก หลังจากการตัดแยกแล้วจะเหลืออักษรแค่ 3 ระดับเท่านั้นคือ บน กลาง และล่าง เราให้ระดับบน เป็นกลุ่มหนึ่ง ระดับล่างเป็นอีกกลุ่มต่างหาก สำหรับระดับกลางจะแยกกลุ่มย่อยอีก 7 กลุ่ม ซึ่งหลักการในการรู้จำของแต่ละระดับเหมือนกัน มีการแบ่งกลุ่ม มีการหาลักษณะบ่งความต่างเฉพาะกลุ่มย่อยเหมือนกัน

เนื้อหาต่อจากนี้จะแบ่งดังนี้

1. อธิบายเครื่องมือที่ใช้ในการหาโครงสร้างหลักและลักษณะบ่งความต่างในการแยกแยะภายในกลุ่ม
2. การรู้จำอักษรระดับกลาง
3. การรู้จำอักษรระดับบน
4. การรู้จำอักษรระดับล่าง

3.3.1 เครื่องมือที่ใช้ในการหาลักษณะบ่งความต่าง

เราเสนอเครื่องมือที่ใช้ในการรู้จำ ทั้งการแบ่งกลุ่มใหญ่และการแยกแยะภายในกลุ่มย่อยมีดังนี้

- การ Normalize ขนาดของตัวอักษร
- การหาจำนวนเส้นแนวดิ่ง
- การนับช่องว่างจากกรอบถึงจุดภาพของอักษร ทั้ง 4 ด้าน

3.3.1.1 การ Normalize ขนาดของตัวอักษร

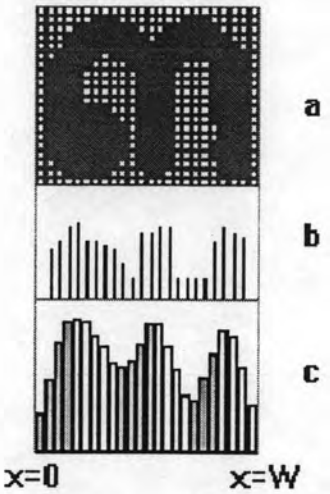
เนื่องจากกระบวนการในการหาส่วนประกอบของโครงสร้างหลักและลักษณะบ่งความต่างนั้น เราจะกำหนดกฎเกณฑ์บางอย่างเกี่ยวกับขนาดตายตัวเช่น การกำจัดสัญญาณรบกวนในการตรวจสอบลักษณะของหลังคา อีกเหตุผลหนึ่งคือเพื่อลดขนาดของข้อมูลในกรณีที่อักษรมีขนาดใหญ่มาก แต่กรณีที่ขนาดอักษรเข้าเล็กอยู่แล้วไม่สามารถเพิ่มข้อมูลขึ้นได้ ดังนั้นหลังการทำ Normalize แล้วอักษรทุกขนาดจะเหมือนกันหมด

12 point 14 point 16 point

รูปที่ 3.25 แสดงอักษร ช ที่ 12 14 และ 16 point

จากเหตุทั้ง 2 เราจึงต้องเลือกขนาดที่จะ normalize อักษรนั่นคือต้องเล็กที่สุดเพื่อให้ขนาดข้อมูลน้อยที่สุดและยังมีข้อมูลครบ จากรูปที่ 3.25 จะเห็นว่าหยักของอักษร ช จะยังเหลือพอให้ตรวจสอบได้สำหรับ 14 point สำหรับอักษรอื่นๆ ก็เช่นกัน ดังนั้นเราจะให้อักษรทุกขนาดเปลี่ยนขนาดมาเป็น 14 point สำหรับอักษรที่ขนาดต่ำกว่า 14 point ถ้าข้อมูล(เช่น หยัก)หายไปแล้วเมื่อขยายขนาดขึ้นก็ไม่มีประโยชน์ แต่ถ้าข้อมูลยังอยู่เมื่อขยายขนาดขึ้นมาก็จะสามารถตรวจพบได้โดยกระบวนการซึ่งทำมาเฉพาะสำหรับ 14 point

3.3.1.2 การหาจำนวนเส้นแนวดิ่ง



รูปที่ 3.26 a) แสดงจุดภาพอักษร b) แสดง Vertical Projection c) แสดงค่าเฉลี่ยของ Vertical Projection

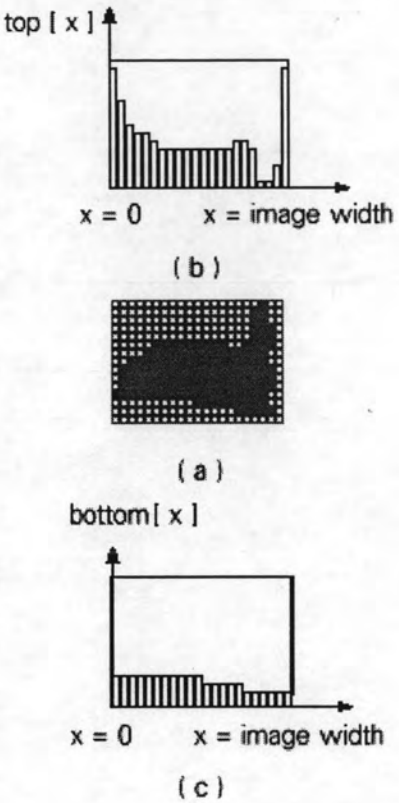
การหาลักษณะของเส้นแนวดิ่งใช้การรวมจุดตามแนวตั้ง (vertical projection) ซึ่งจะเกิดเป็นรูปภูเขาดังรูป b ค่าที่ได้เป็นฟังก์ชันของ x ให้ vert [x] แทนจำนวนจุดทั้งหมดในแนวตั้งที่ x และเราจะใช้ vert_av[x] เพื่อแทน ค่าเฉลี่ยของ vert [x] แล้วโดยใช้สูตร

$$vert_av\ [\ x\ ] = vert\ [\ x - 2\ ] / 2 + vert\ [\ x - 1\ ] + vert\ [\ x\ ] + vert\ [\ x + 1\ ] + vert\ [\ x + 2\ ] / 2$$

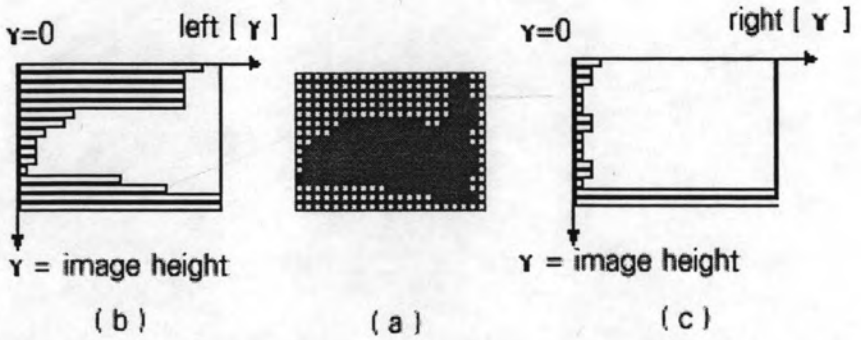
เพราะการใช้ vert [x] โดยตรงในการนับจำนวนจุดยอดเกิดความผิดพลาดมากเนื่องจากอาจได้มากกว่า 1 ยอดสำหรับภูเขาลูกหนึ่ง ซึ่งจะใช้ในการรู้จำผิดพลาด (ดังในรูป b) แต่เมื่อใช้ vert_av[x] ดังรูป c จะให้ผลดีกว่า เนื่องจากอักษรถูก Normalize แล้ววิธีนี้จึงใช้ได้กับอักษรทุกขนาด

นับยอดของ vert_av โดยให้ peak [n] เท่ากับตำแหน่งในแนวแกน x ที่เกิดยอดเขาขึ้น

3.3.1.3 การนับช่องว่างจากกรอบถึงจุดภาพของอักษร ทั้ง 4 ด้าน



รูปที่ 3.27 (a) แสดงจุดภาพอักษร (b) top [x] ของภาพตัวอย่าง (c) bottom [x] ของภาพตัวอย่าง



รูปที่ 3.28 (a) แสดงจุดภาพอักษร (b) left [x] ของภาพตัวอย่าง (c) right [x] ของภาพตัวอย่าง

top [x] แทน ขนาดช่องว่างด้านบน ณ จุด x (x = 0 ที่ซ้ายสุด)

bot [x] แทน ขนาดช่องว่างด้านล่าง ณ จุด x (x = 0 ที่ซ้ายสุด)

right [y] แทน ขนาดช่องว่างด้านข้างนับจากกรอบขวา ณ จุด y (y = 0 ที่บนสุด)

left [y] แทน ขนาดช่องว่างด้านข้างนับจากกรอบซ้าย ณ จุด y (y = 0 ที่บนสุด)

และใช้ H = image height , W = image width , LH = Line Height

3.3.2 การรู้จำอักษรระดับกลาง

สำหรับอักษรระดับกลางนั้นหลังจากตัดหางและส่วนล่างทิ้งไปแล้วจะเหลือสมาชิกดังนี้

ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฐ ฑ ฒ ณ ด ต ถ ท ธ
น บ ผ พ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ ะ า เ ๆ ฯ
๑ ๒ ๔ ๕ ๖ ๘ ๙ ๐
และสระ แอ มาจากความผิดพลาดจากส่วนตัดแยกระดับ 3

เนื่องจากตัวเลขบางตัวต้องแยกออกไปก่อนด้วยการตรวจสอบลักษณะบ่งความต่างบางอย่างเพราะว่าตัวเลขเหล่านั้นไม่มีโครงสร้างหลักเหมือนกับตัวอักษร ดังนั้นจะมีการแยกตัวเลขบางตัวออกไปก่อน ได้แก่เลข ๑ ๒ ๔ ๕ ๖ ๘ ๐ (แยกสระ ะ โดยดูความสูงอักษรเทียบกับความสูงของบรรทัด)

เราจะทำการแบ่งกลุ่มอักษรที่เหลือด้วยวิธีทาง SyntPR โดยเราเลือก primitive ซึ่งในที่นี้คือส่วนประกอบของโครงสร้างหลัก จากนั้นเมื่อดูอักษรที่เหลือจะพบส่วนประกอบของโครงสร้างหลัก 3 อย่าง คือ

1 การหาค่า vert_av คือจำนวนยอดของภูเขาที่เกิดขึ้น

2 การพิจารณาว่ามี "หลังคา" หรือไม่

3 การพิจารณาว่ามี "พื้นเรียบ" หรือไม่

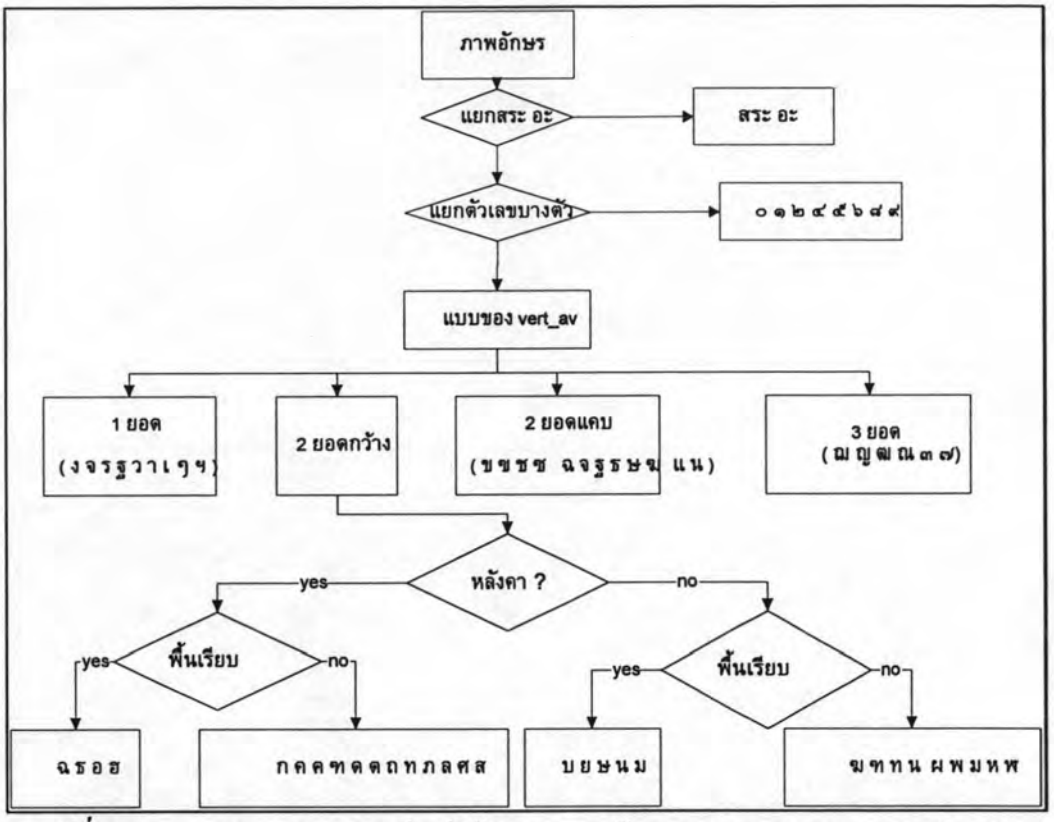
ทำให้แบ่งกลุ่มตามโครงสร้างหลักได้ 7 กลุ่ม ตามตารางที่ 3.6 ซึ่งตารางนี้ในแต่ละกลุ่มถือเป็น 1 Grammar ซึ่งจากแต่ละ Grammar นั้น เราสามารถสร้างสมาชิกในกลุ่มออกมาได้ครบตามที่ตารางแสดงไว้ และเมื่อมีอักษรที่ต้องการหากลุ่ม เราต้องหาส่วนประกอบของโครงสร้างหลักก่อน เมื่อจะทำการ parsing ว่าตรงกับ Grammar ใด เราจะใช้การตรวจสอบจากตารางได้

ตารางที่ 3.6 แสดงการแบ่งกลุ่มของอักษรระดับกลาง (ซึ่งแยกตัวเลขบางตัวออกไปแล้ว)

กลุ่มที่	ข้อมูลจาก vert_av	หลังคา	พื้นเรียบ	อักษรในกลุ่ม
1	มี 1 ยอด	-	-	ง จ ร ฐ ว า เ ๆ ฯ
2	มี 2 ยอดกว้าง	✓	×	ก ค ฅ ฑ ด ต ถ ท ภ ล ศ ส
3	มี 2 ยอดกว้าง	×	×	ฆ ฑ ท น ผ พ ม ห ฬ
4	มี 2 ยอดกว้าง	×	✓	บ ป ษ น ม
5	มี 2 ยอดกว้าง	✓	✓	ฉ ฐ อ ฮ
6	มี 2 ยอดแคบ	-	-	ข ฃ ช ซ ฉ จ ฐ ธ ษ ฆ แ น
7	มี 3 ยอดขึ้นไป	-	-	ฌ ญ ฒ ณ ๓ ๗

จะเห็นได้ว่าอักษรบางตัวอยู่ 2 กลุ่ม เนื่องจาก font ของอักษรที่ใช้แตกต่างกันมีผลกับการทำงานในส่วน vert_av การจะทำให้อักษรอยู่ในกลุ่มเดียวทุกตัวนั้นจะทำให้ขั้นตอน vert_av สิ้นเปลืองการคำนวณมากเกินไป จากนั้นจึงส่งไปยังส่วนแยกแยะเฉพาะกลุ่มทั้ง 7 ต่อไป

สามารถแสดงดัง flow chart ในรูปที่ 3.29



รูปที่ 3.29 แสดง Flow Chart การแบ่งกลุ่มอักษร การแยกตัวเลขและสระอะ ออกจากกระบวนการ

หมายเหตุ บางอักษรอยู่ในหลายกลุ่มเนื่องจากอักษรเดียวกันแต่ต่างแบบอักษรกันบางตัวมีโครงสร้างต่างกัน

3.3.2.1 ขั้นตอนการแยก สระอะ

เนื่องจากการหาระดับในขั้นตอนแรกจะให้ "สระอะ" อยู่ในลักษณะที่ 3 และขั้นตอนดึงเกาะของจุดภาพจะดึงสระอะได้ทีละครึ่งเดียว (ไม้หันอากาศ 2 อัน) ดังนั้นสระอะจะเป็นอักษรลักษณะที่ 3 ที่มีความสูงน้อยกว่าความสูงของบรรทัดมากจึงใช้เกณฑ์

" ถ้า image Height x 2 < Line Height แล้ว ตอบเป็น Suspect_สระอะ "

ในขั้นตอนการตรวจสอบขั้นสุดท้ายถ้ามี Suspect_สระอะ 2 ตัวติดกันจะรวมเป็น สระอะ ตัวเดียว แต่ถ้าไม่จะแก้เป็นเครื่องหมายลบ

3.3.2.2 ขั้นตอนการแยกตัวเลขบางตัว

สมาชิกในกลุ่มนี้มี ๑ ๒ ๔ ๕ ๖ ๘ ๐ (สำหรับในตัวโปรแกรมสามารถเลือกได้ว่าในเอกสารต้องการให้รู้จำตัวเลขหรือไม่)

การทำงานมี 2 ขั้นตอนคือ

1 ใช้ลักษณะบ่งความต่าง 5 อย่างในการแยกตัวเลขเหล่านี้ออกจากอักษร

2 แยกแยะตัวเลขที่ตรวจสอบมาได้ว่าเป็นเลขอะไร

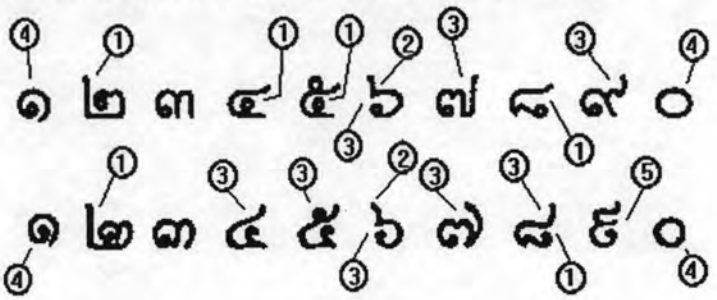
3.3.2.2.1 การแยกตัวเลขออกจากอักษร

จากการศึกษาเราเห็นได้ชัดเจนว่าสามารถแยกตัวเลขออกจากตัวอักษรได้อย่างง่ายโดยใช้ลักษณะบ่งความต่าง 5 อย่างแสดงดังตารางที่ 3.7 (จะอธิบายวิธีหาในหัวข้อต่อไป)

ตารางที่ 3.7 แสดงเงื่อนไขการตรวจสอบว่าเป็นตัวเลขไทยหรือไม่

กลุ่มที่	ตรวจตัวเลข	ลักษณะบ่งความต่าง
1.	๒ ๔ ๕ ๖ ๘	มีช่องทางขวาและความกว้างใกล้เคียงความสูง
2	๖	มีช่องว่างต่อเนื่องทางขวาและสูงมากกว่ากว้าง
3	๔ ๕ ๗ ๙	มีช่องว่างต่อเนื่องทางซ้าย
4	๐ ๑	กลม
5	๙ (AngsanaUPC)	มีช่องว่างทางขวา 2 ช่อง

เราแสดงตำแหน่งของลักษณะบ่งความต่างที่ปรากฏในตัวเลขไทยได้ดังรูปที่ 3.30



รูปที่ 3.30 แสดงตำแหน่งลักษณะบ่งความต่างของตัวเลขไทยแบบ (บน)CordiaUPC (ล่าง) AngsanaUPC

3.3.2.2.2 การแยกแยะตัวเลข

หลังจากแยกตัวเลขออกมาและแบ่งเป็นกลุ่มตามลักษณะบ่งความต่างที่แสดงดัง ตาราง 3.7 แล้ว งานต่อไปคือแยกแยะในกลุ่ม ลักษณะบ่งความต่างที่แยกในกลุ่มที่ใช้มีดังนี้

กลุ่มที่ 1 มีสมาชิกคือ ๒ ๔ ๕ ๖ ๘

- เป็น ๒ เมื่อ มีหางทางซ้ายและช่องทางซ้าย
- เป็น ๖ เมื่อ มีหางทางซ้ายและช่องทางขวา
- เป็น ๘ เมื่อ มีหางทางขวาและหัวอยู่ต่ำ
- เป็น ๔ เมื่อ มีหางทางขวา หัวไม่อยู่ต่ำ (เทียบกับ ๘)และ หลังคาเรียบ
- เป็น ๕ เมื่อ มีหางทางขวา หัวไม่อยู่ต่ำ และ หลังคาไม่เรียบ

กลุ่มที่ 3 มีสมาชิกคือ ๔ ๕ ๗ ๙

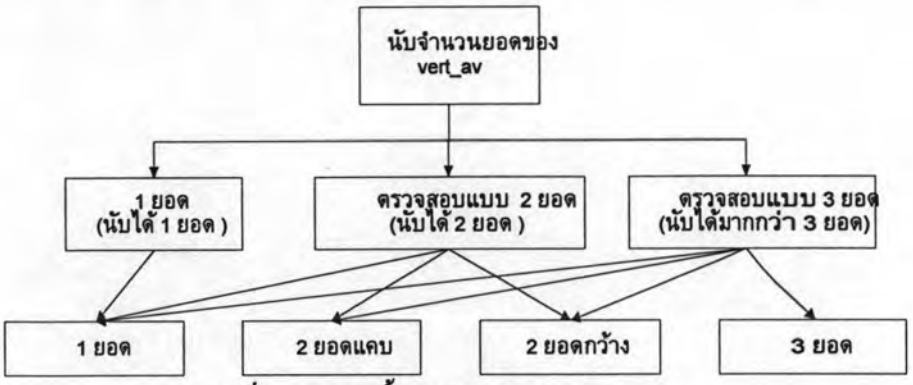
- เป็น ๔ เมื่อ พื้นเรียบและหลังคาเรียบ
- เป็น ๕ เมื่อ พื้นเรียบและหลังคาไม่เรียบ
- เป็น ๗ เมื่อ พื้นไม่เรียบและด้านขวาเรียบ
- เป็น ๙ เมื่อ พื้นไม่เรียบและด้านขวาไม่เรียบ

กลุ่มที่ 4 มีสมาชิกคือ ๐ ๑ - เป็น 0 เมื่อภายในกลวง

3.3.2.3 การหาแบบของ vert_av

แบบของ vert_av มี 4 ค่า คือ 1 ยอด 2 ยอดกว้าง 2 ยอดแคบ และ 3 ยอด

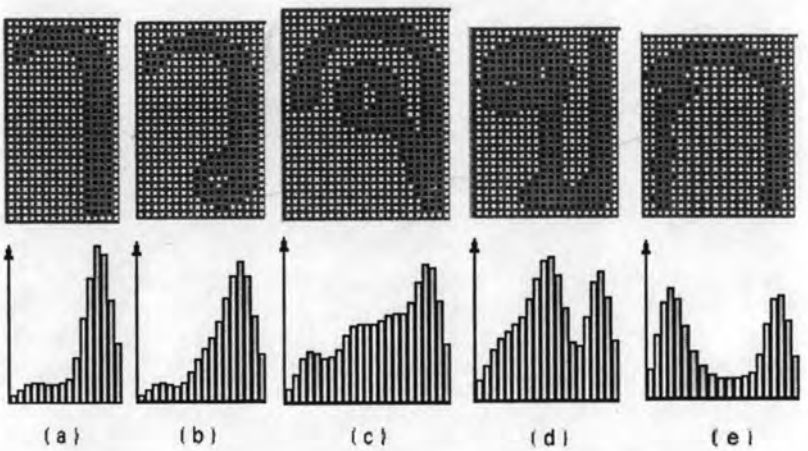
การนับยอดทำได้โดยดูการเปลี่ยนความชันจากบวกเป็นลบของ vert_av [x] ในหัวข้อ 3.1.2



รูปที่ 3.30 แสดงขั้นตอนการหาแบบของ vert_av

ส่วนตรวจสอบทำหน้าที่ 2 อย่างคือแก้ค่า n ให้ถูกต้อง และ peak [n] ตรงกับยอดภูเขาที่ต้องการ

- ***การตรวจสอบแบบ 2 ยอด***คือแยกอักษรที่จริงๆ ควรเป็น 1 ยอดแต่นับได้ 2 ยอด ตัวอย่างดังรูป

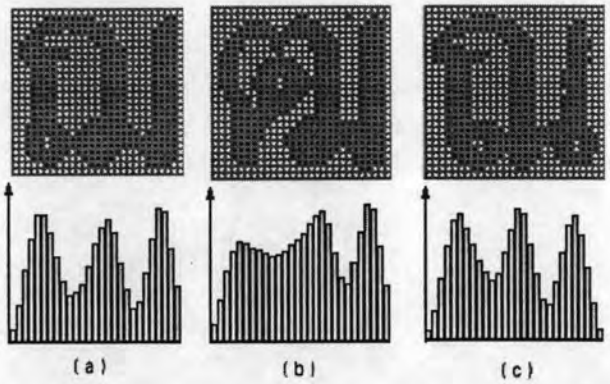


รูปที่ 3.31 แสดง Vertical Projection ของอักษรที่นับยอดเขาได้ 2 ยอด

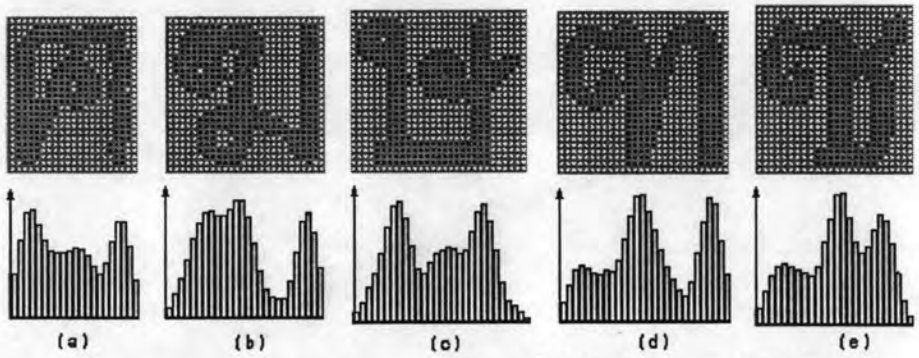
จากรูป a b และ c จะเห็นได้ว่านับได้ 2 ยอดแต่จริงๆ เป็นแบบ 1 ยอด เราสามารถตรวจยอดปลอมที่เกิดขึ้นได้คือยอดแรกจะมีค่าต่ำกว่ายอด 2 มาก สำหรับกรณีที่เป็นแบบ 2 ยอดจริงๆ เราจะทำการตรวจว่าเป็นแบบแคบหรือกว้างโดยตรวจดูระยะห่างระหว่าง 2 ยอดเทียบกับความกว้างของภาพอักษร โปรดดูในรูป d ซึ่งเป็นแบบ 2 ยอดแคบเห็นว่ามีระยะห่างแคบกว่ารูป e ซึ่งเป็นแบบ 2 ยอดแคบมาก

- *การตรวจสอบแบบ 3 ยอด* คือแยกอักษรที่จริงๆ ควรเป็น 1 หรือ 2 ยอดแต่นับได้ 3 ยอด แสดงตัวอย่างดังรูป 3.32 คือแบบ 3 ยอดจริง และรูปที่ 3.33 เป็นแบบที่นับผิด

อักษรที่มียอดภูเขา 3 ยอด ปกติจะมีลักษณะภูเขาดังนี้ คือ ยอดค่อนข้างกระจาย ความสูงใกล้เคียงกัน



รูปที่ 3.32 แสดง Vertical Projection ของอักษรแบบ 3 ยอดและนับได้ 3 ยอด



รูปที่ 3.33 แสดง Vertical Projection ของอักษรที่ไม่ใช่แบบ 3 ยอดแต่นับได้ 3 ยอด

ถ้าดูจากรูปที่ 3.33 จะเห็นว่ายอดของ vert_av มีความสูงไม่สม่ำเสมอและไม่กระจาย รวมทั้งค่า bot [peak [n]] ที่ ผิดจะมีค่าสูง เช่น ยอดที่ 2 ของ รูปที่ 3.33 (a) ยอดที่ 1 ของรูปที่ 3.33 (d) , (e)

3.3.2.4 การหาว่ามี "หลังคา" หรือไม่

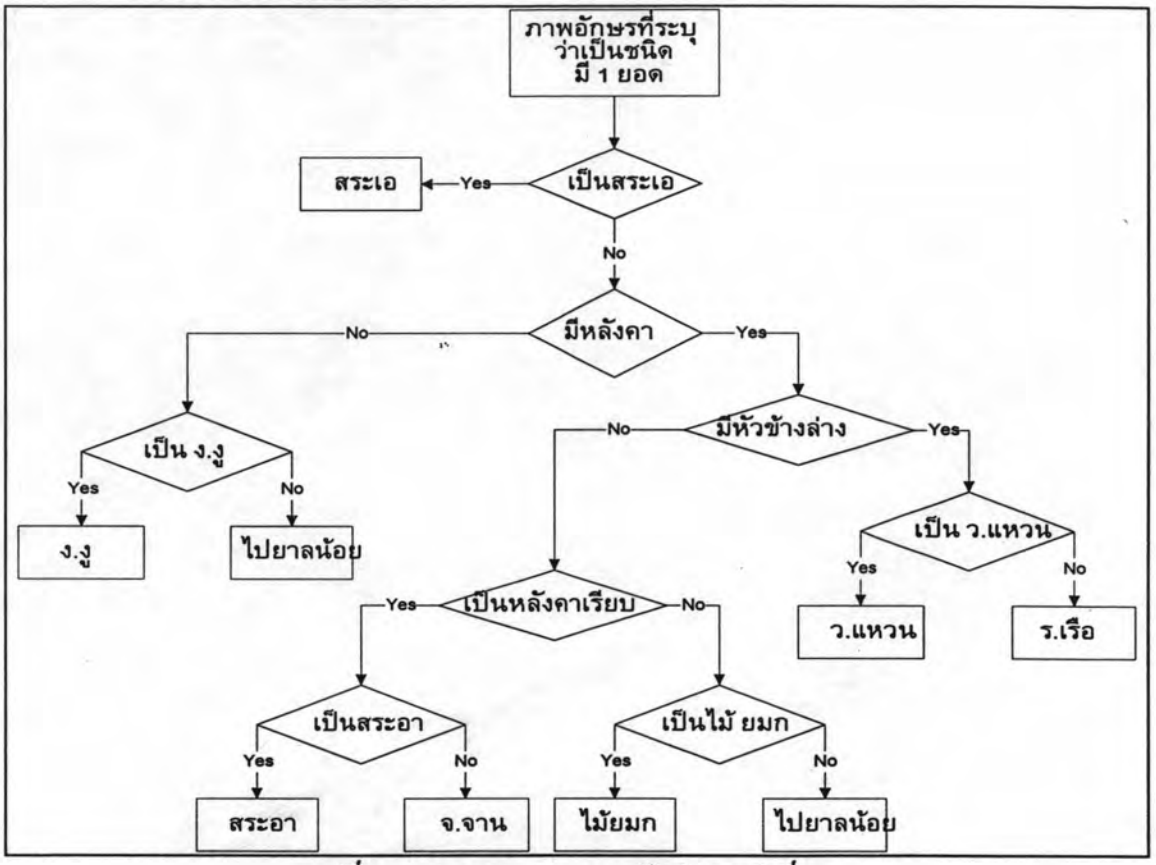
การมีหลังคาคือ มีส่วนที่ top มีค่าน้อยต่อเนื่องกันโดยไม่มีส่วนที่ top มีค่ามากมาคั่น

3.3.2.5 การหาว่ามี "พื้นเรียบ" หรือไม่

การมีพื้นเรียบคือ มีส่วนที่ bot มีค่าน้อยต่อเนื่องกันโดยไม่มีส่วนที่ bot มีค่ามากมาคั่น

3.3.2.6 ขั้นตอนการรู้จำกลุ่มที่ 1 (ง จ ร ฐ ว า เ ๆ ฯ)

มี flow chart การทำงานดังนี้



รูปที่ 3.34 แสดง flow chart การรู้จำอักษรกลุ่มที่ 1

หลักการตรวจสอบในขั้นตอนต่างๆ อธิบายได้ดังนี้

- การตรวจว่าเป็นสระเอหรือไม่ คือ ตรวจความกว้างของภาพว่ามีความแคบเทียบกับความสูงสม่ำเสมอในช่วงบนหรือไม่
- การตรวจว่ามีหัวในส่วนครึ่งล่างของภาพหรือไม่ โดย ใช้วิธีในข้อ 3.2.14 โดยใช้ ฟังก์ชัน Head (left , H / 2 , H)
- แยก “ง” ออกจาก “ฯ” โดยดูความกว้างภาพอักษรในช่วงล่างถ้าไม่แคบสม่ำเสมอตอบ “ง”
- แยก “ว” ออกจาก “ร” โดยดูว่าถ้าไม่มีช่องทางขวาหรือความหนาแนวตั้งของส่วนบนของภาพอักษรน้อยตอบ “ว”
- แยก “า” ออกจาก “จ” โดยดูว่าถ้าความหนาบริเวณช่วงกลางถึงล่างของภาพอักษรว่าแคบสม่ำเสมอตอบ “า”
- แยก “ฯ” ออกจาก “ๆ” ดูลักษณะที่ตัดได้ถ้าเป็น 3 ตอบ “ฯ”

3.3.2.7 ขั้นตอนการรู้จำกลุ่มที่ 2 (ก ด ต ท ด ต ถ ท ภ ล ศ ส)

เราแบ่งกลุ่มย่อยอักษรลงอีกเป็น 4 กลุ่มตามลักษณะของ bot [x] ที่ต่างกัน คือ

1 ก ถ ภ

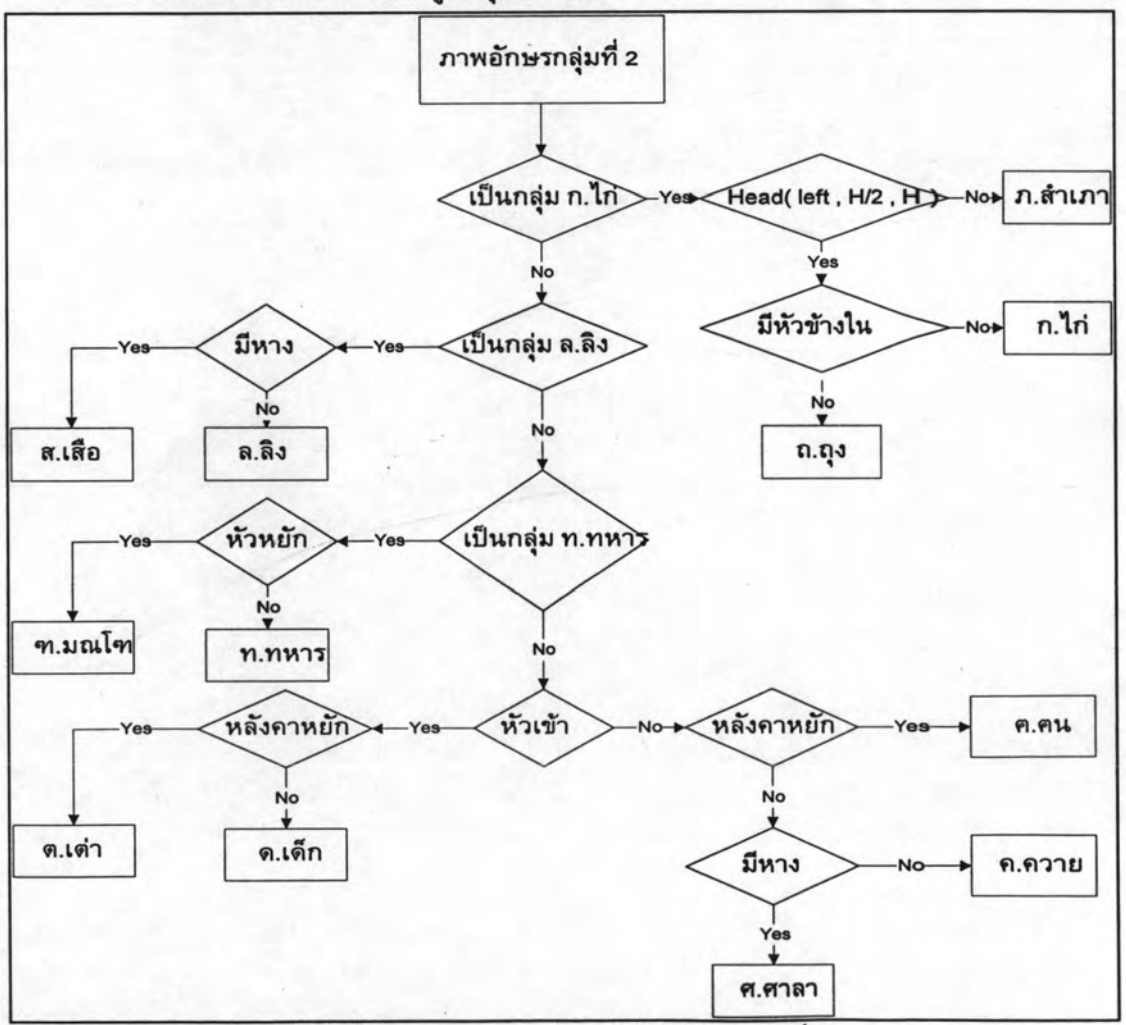
2 ค ต ด ต ศ

3 ท ท

4 ล ส

จากนั้นจึงแยกแยะในกลุ่มย่อยอีกที ซึ่งวิธีในการแยกกลุ่มย่อยนั้นเราใช้ฟังก์ชัน Head (left , H / 2 , H) และการดูแบบของหลังคาว่าเป็นแบบธรรมดา หยักหรือมีหาง นอกจากนี้ยังมีฟังก์ชันที่ต้องสร้างขึ้นมาใหม่ คือ หาว่าหัวของกลุ่ม ท.ทหาร เป็น ท.ทหาร หรือ ท.มณโฑ และการหาหัวของกลุ่ม ก.ไก่ ว่ามีหัวข้างในหรือไม่ รวม 2 ฟังก์ชัน

มี flow chart การทำงานการรู้จำกลุ่มที่ 2 ดังนี้



รูปที่ 3.35 แสดง flow chart การรู้จำอักษรกลุ่มที่ 2

หลักการตรวจสอบในขั้นตอนต่างๆ อธิบายได้ดังนี้

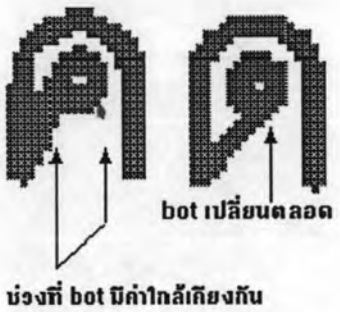
- ตรวจว่าเป็นกลุ่ม "ก" , "ท" หรือ "ล" โดยพิจารณา bot [x] ระหว่าง peak [1] และ peak [2]

  เป็นกลุ่ม "ก" เมื่อ bot [x] มีค่ามาก เกือบเท่ากับความสูงอักษร

  เป็นกลุ่ม "ท" เมื่อ bot [x] มีค่าเปลี่ยนแปลงเพิ่มขึ้นค่อนข้างคงที่

  เป็นกลุ่ม "ล" เมื่อมีหัวข้างในเหมือนกับการหาหัวของ ถ.ถุงที่จะกล่าวต่อไป

  ตรวจที่เหลือว่าเป็นแบบหัวเข้าหรือออกอธิบายโดยรูปที่ 3.36



รูปที่ 3.36 แสดงลักษณะบ่งความต่างระหว่างหัวเข้ากับหัวออก

- ตรวจว่าเป็นหัว "ถ" หรือไม่ อธิบายดังรูปที่ 3.37

รูปที่ 3.37 แสดงลักษณะบ่งความต่างของหัวข้างใน ถ.ถุง

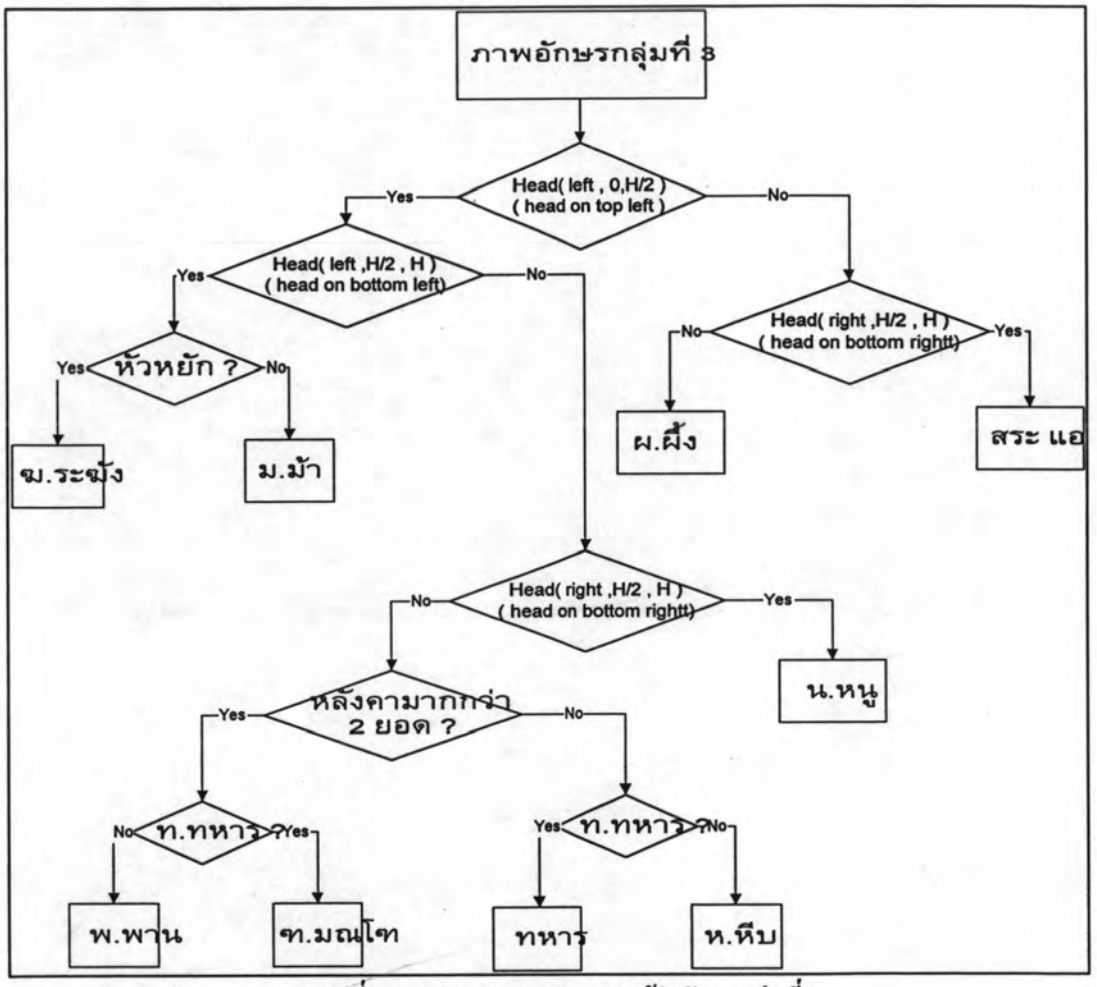
ใช้ลักษณะเช่นเดียวกับฟังก์ชัน Head โดยเปลี่ยนตัวแปรที่ใช้ในฟังก์ชัน Head คือเราแทนที่จะใช้ side[y] แต่เราจะนับช่องว่างจาก "แนว x ที่เลือก" ไปทางซ้ายจนกว่าจะถึงจุดดำ (ซึ่งคือขาหน้าของกลุ่ม ก.ไก่) แทน สำหรับแนว x ที่เลือกนี้คือช่องระหว่าง peak [1] กับ peak [2]

-ตรวจว่าเป็นหัวหยักหรือไม่

การตรวจว่าเป็นหัวหยักหรือไม่ เราใช้การพิจารณาแบบของหลังคา เนื่องจากถ้าเราพิจารณาจำนวนหยักของหลังคาจะพบว่า ท.ทหารจะมี 2 ยอด ส่วน ท.มณโฑ จะมี 3 ยอด ดังนั้นถ้าค่าของแบบหลังคาตอบมาเป็น 2 หรือ 3 (คือ มีหางหรือหยัก) แสดงว่าเป็น ท.ทหาร ถ้าตอบมาเป็น 4 คือมีมากกว่า 2 ยอดแสดงว่าเป็น ท.มณโฑ

3.3.2.8 ขั้นตอนการรู้จำกลุ่มที่ 3 (ฆ ฑ ท น ผ พ ม ห ฬ ฮ)

Flow Chart การรู้จำของอักษรกลุ่มที่ 3 มีดังนี้



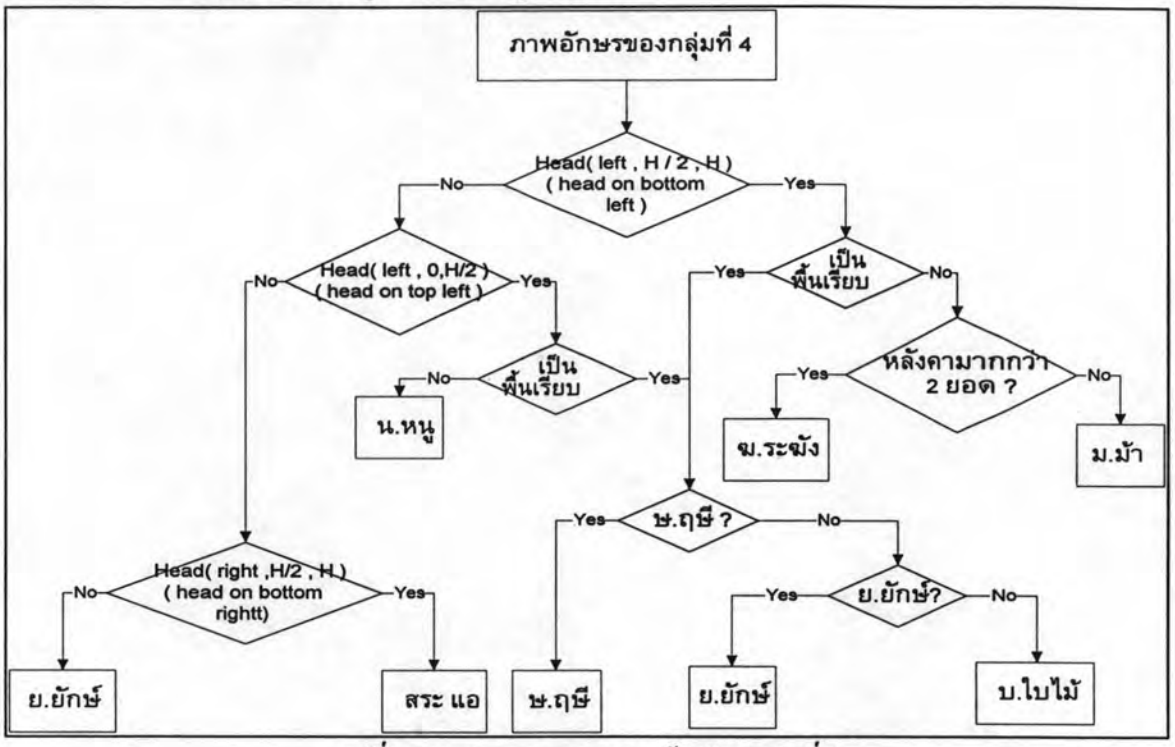
รูปที่ 3.38 แสดง flow chart การรู้จำอักษรกลุ่มที่ 3

ฟังก์ชัน Head ใช้ตามข้อ 3.2.14 การตรวจหัวหยักเหมือน 3.2.7

- การตรวจว่าเป็น ท.ทหารหรือไม่ ใช้การดู bot [x] ในช่วงระหว่าง peak [1] กับ peak [2] ว่าถ้ามีค่ามากกว่า "ห" หรือ "พ" ปกติ ให้ตอบว่าเป็น "ท"

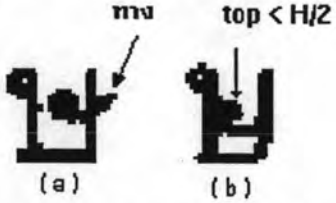
3.3.2.9 ขั้นตอนการรู้จำกลุ่มที่ 4 (บ ย ษ น ม ฆ ฒ)

Flow Chart การรู้จำของอักษรกลุ่มที่ 4 มีดังนี้



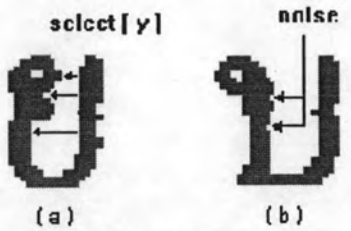
รูปที่ 3.39 แสดง flow chart การรู้จำอักษรกลุ่มที่ 4

- การตรวจว่าเป็นพื้นเรียบ (อีกครั้ง เนื่องจากการตรวจครั้งแรกเราใช้เกณฑ์ที่เน้นให้กลุ่มที่ 4 ไม่ไปปนกับกลุ่มอื่น จึงมีกลุ่มอื่นปนเข้ามาในกลุ่มนี้ ดังนั้นการตรวจคราวนี้คล้ายเดิมคือ หัวข้อ 3.2.5 แต่ใช้เกณฑ์ที่เข้มงวดมากขึ้น
- ตรวจว่าเป็น "ษ" หรือไม่ ใช้วิธีตรวจดังรูปที่ 3.40 คือตรวจหางที่ยื่นออกมาทางขวา หรือตรวจช่องระหว่าง peak [1] และ peak [2] ซึ่งควรจะเกือบถึงพื้น



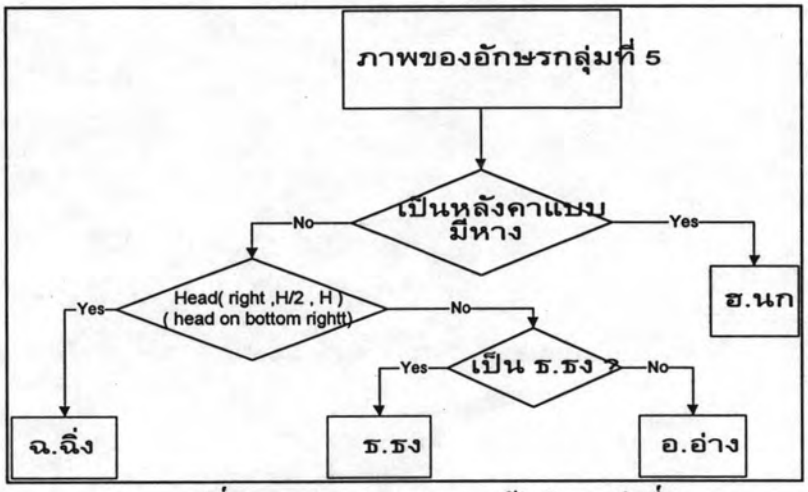
รูปที่ 3.40 แสดงลักษณะบ่งความต่างของ ษ แบบอักษร a) CordiaUPC b) AngsanaUPC

- ตรวจว่าเป็น ย.ยักษ์ หรือไม่ทำดังรูปที่ 3.41 คือเริ่มจากตัดอักษรเป็น 2 ส่วนแล้วพิจารณาความหยักของส่วนหน้าคล้ายกับการตรวจความหยักของหัวอักษร ซึ่ง "บ" จะเรียบกว่า "ย"



รูปที่ 3.41 แสดงการตรวจสอบหยักภายในซึ่งเป็นลักษณะบ่งความต่างระหว่าง ย.ยักษ์ กับ บ.ใบไม้

3.3.2.10 ขั้นตอนการรู้จำกลุ่มที่ 5 (ฉ ธ อ ฮ)

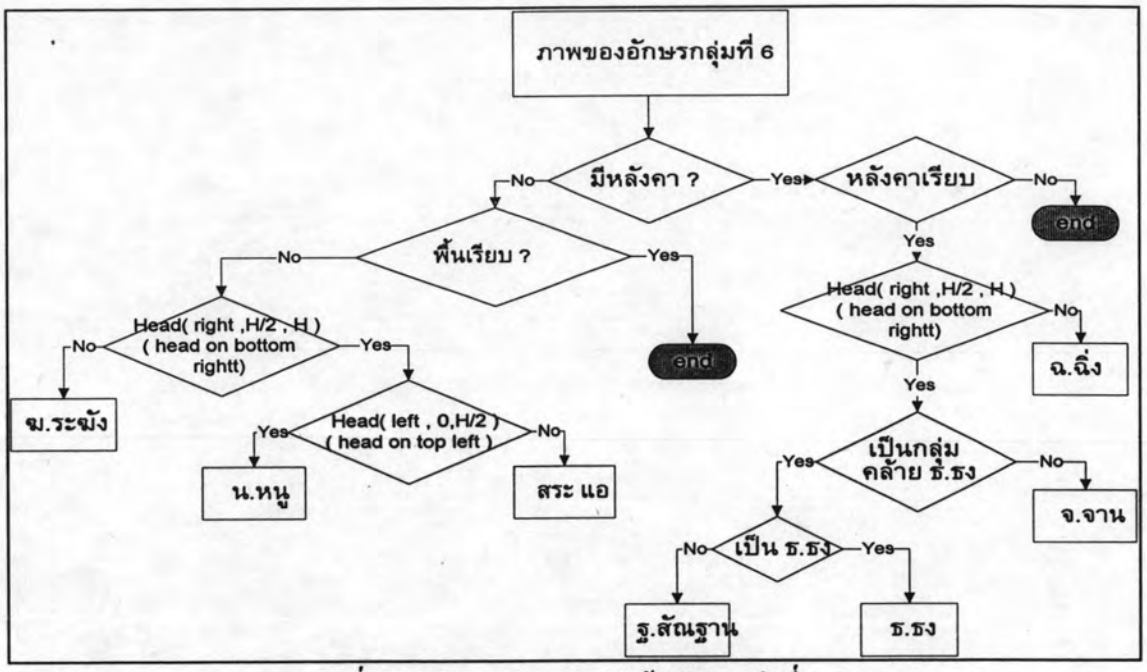


รูปที่ 3.42 แสดง flow chart การรู้จำอักษรกลุ่มที่ 2

- ตรวจว่าเป็น "ธ" โดยดูช่องทางขวา (เนื่องจาก "อ" ไม่มี)

3.3.2.11 ขั้นตอนการรู้จำกลุ่มที่ 6 (ข ฃ ช ซ ฉ จ ฐ ธ ษ ฆ แ น)

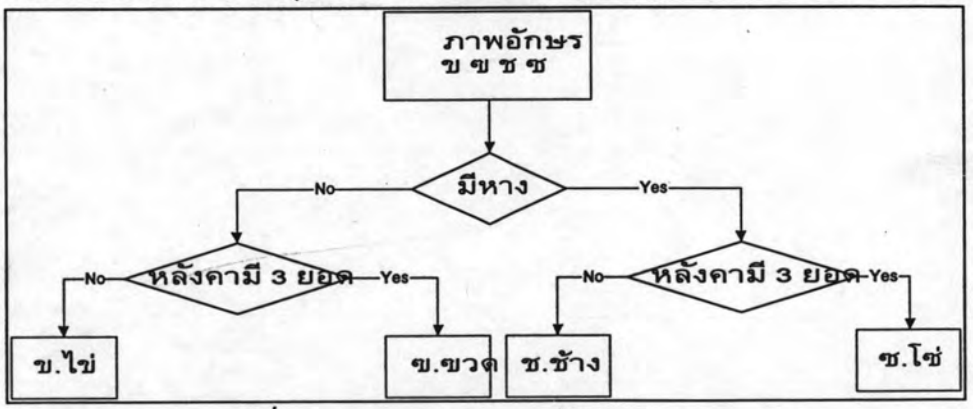
เราจะทำการแยกอักษรที่ไม่ใช่ ข ฃ ช ซ ออกมาก่อนด้วย flow chart ดังภาพ 4.28 ถ้าผ่านขั้นตอนนี้ไป (ส่วนที่แรเงาเขียนว่า end) จะเข้าสู่การแยกแยะ ข ฃ ช ซ ต่อไป



รูปที่ 3.43 แสดง flow chart การรู้จำอักษรกลุ่มที่ 6

- แยก "ธ" ออกจาก "ฐ" ด้วยความกว้างของฐานอักษรเนื่องจาก ธ จะกว้างและเรียบกว่า

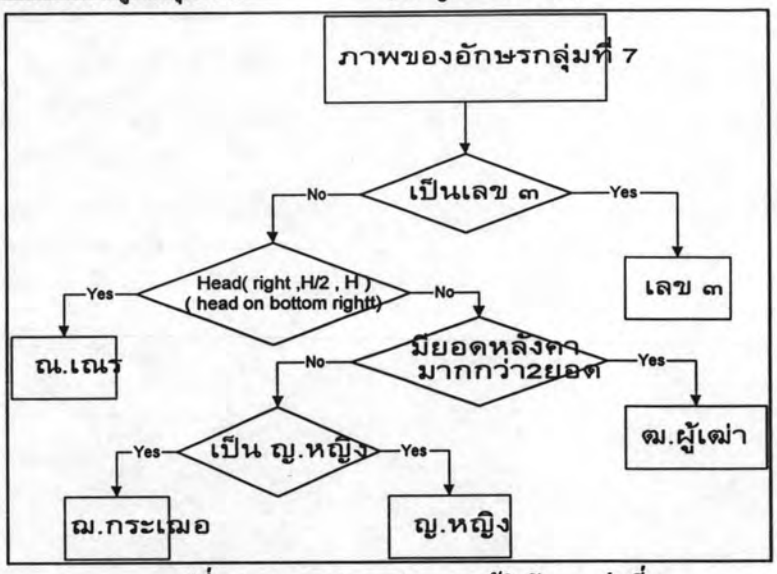
สำหรับอักษร ข ฃ ช ซ ซึ่งหลุดจากการแยกในชั้นแรก จะทำการแยกแยะด้วย flow chart ดังต่อไปนี้



รูปที่ 3.44 แสดง flow chart การรู้จำอักษร ข ฃ ช ซ

การตรวจว่าหลังคามี 3 ยอดหรือไม่ใช้วิธีในข้อ 3.2.13 ส่วนการตรวจหางใช้ตรวจด้านขวาว่ามียุบลงไปหรือไม่

3.3.2.12 ขั้นตอนการรู้จำกลุ่มที่ 7 (ณ ญ ฒ ณ ๓ ๗)



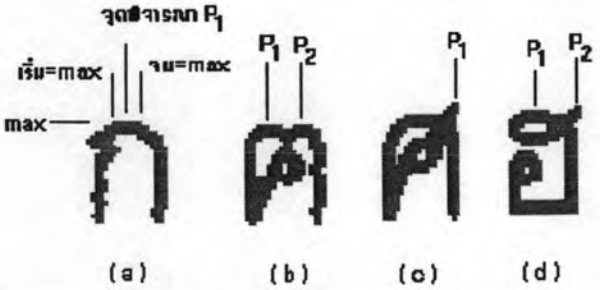
รูปที่ 3.45 แสดง flow chart การรู้จำอักษรกลุ่มที่ 7

- เป็นเลข ๓ เมื่อไม่มีช่องด้านบนระหว่าง Peak [2] กับ Peak [3]
- แยก "ญ" จาก "ฌ" ด้วยพื้นช่วงระหว่าง Peak [2] กับ Peak [3] ถ้าเรียบเป็น "ญ"

3.3.2.13 การแยกแยะแบบของหลังคา

จะให้ค่า 3 ค่าคือ

1 เป็นหลังคาเรียบธรรมดา
2 เป็นหลังคาหยัก (เช่น ต ค)
3 เป็นหลังคามีหาง (เช่น ส ศ)
4 ไม่ระบุ



รูปที่ 3.46 แสดงแบบของหลังคา a) เรียบธรรมดา b) มีหยัก c) , d) มีหาง

- เราตรวจสอบว่าเป็นหลังคาแบบใดจากตำแหน่งจุดเปลี่ยนความชันของ top [x] ซึ่งถ้ามีจุดเปลี่ยนความชันอยู่ริมซ้ายเช่นรูป 3.46 (c) และ (d) ถือเป็นแบบมีหาง และเป็นแบบเรียบเมื่อมีจุดเปลี่ยนความชัน 2 จุดเป็นแบบหยัก ถ้ามีจุดเดียวเป็นหลังคาธรรมดา

3.3.2.14 การพิจารณาหัวของอักษร

เราจะใช้สัญลักษณ์ Head(side , begin , end) แทนฟังก์ชันการหาหัว

- side หมายถึง ด้านที่จะหาหัว
- begin คือ y จุดเริ่มพิจารณา
- end คือ y จุดสุดท้ายที่ต้องการพิจารณา

โดยจะคืนค่าเป็น 1 เมื่อในช่วง begin ถึง end ในด้าน side มีหัว ถ้าไม่มีคืนค่า 0

รูปที่ 3.47 แสดงตัวอย่างการหาหัวของ ว.แหวนใช้ฟังก์ชัน Head (left , H / 2 , H)
(แสดงค่า index ไล่จาก H / 2 เริ่มด้วย 0 เป็น + เป็น 0 และจบด้วย -)

เราแสดงวิธีหาหัวดังตัวอย่าง ว.แหวน เราจะหา Head (left , H / 2 , H) โดยใช้หลักการว่าหัวของอักษรนั้น ถ้าไล่ตาม y จะเริ่มจากเรียบก่อน ดูความเปลี่ยนแปลงของค่า side แล้วมีค่า + ช่วงหนึ่ง แล้วมีค่า 0 ช่วงหนึ่งแล้วจบด้วยช่วงลบ ถึงเป็น 1 หัว โดยไม่จำเป็นต้องดู "ช่องในหัวอักษร" เพราะภาพอักษรส่วนใหญ่ถ้าตัวเล็กจะหาไม่เจอ

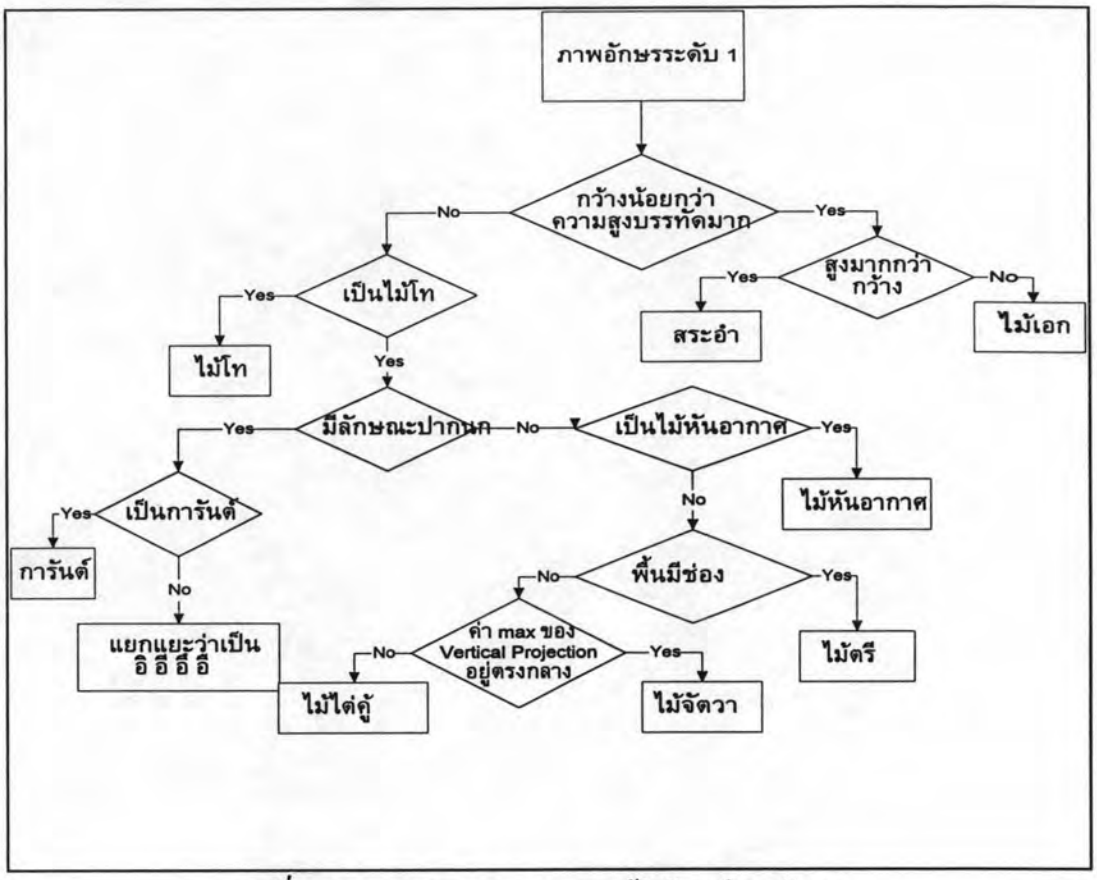
เนื่องจากเรา normalize ขนาดอักษรดังนั้นเราจึงสามารถกำหนดช่วงที่เหมาะสมเป็น "***ค่าคงที่***" ได้ โดยค่า ช่วง + = 2 ช่วง - = 2 ช่วง - = 2 ถ้าเป็นการหา Head (right , begin ,end) ก็จะเปลี่ยนตัวแปร left เป็น right

3.3.3 การรู้จำอักษรระดับบน

สำหรับอักษรระดับบนนั้น ได้รับจากทั้งตัวตัดแยกอักษรลักษณะที่ 1 และลักษณะที่ 2 มีสมาชิกดังนี้ [illegible] และเศษที่เหลือจาก ป ฝ ฟ ไ ๅ ไ

3.3.3.1 การรู้จำอักษรระดับบน ได้แก่ [illegible]

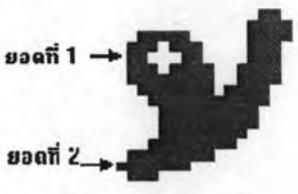
มี Flowchart การทำงานดังนี้



รูปที่ 3.48 แสดง flow chart แสดงการรู้จำอักษรในระดับบน

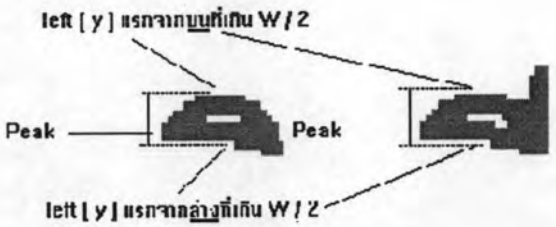
เราทำการแยกอักษรที่มีความกว้างน้อยเมื่อเทียบความสูงของบรรทัดไปก่อน(ได้แก่อักษร ' °) จากนั้นจึงแยกไม้โทออกจากที่เหลือ สำหรับอักษรที่เหลือสามารถแบ่งแยกได้ 2 ประเภทคือ อักษรที่มีลักษณะปากนก คือ 4 ตัว และประเภทที่ไม่มีลักษณะปากนก(วิธีตรวจสอบจะอธิบายต่อไป) จากนั้นจึงแยกย่อยสำหรับทั้ง 2 กลุ่มต่อไป ซึ่งการแยกประเภทที่ไม่มีลักษณะปากนกจะแสดงในหัวข้อย่อยนี้แต่การแยกอักษรที่มีลักษณะปากนกจะใช้ร่วมกับข้อ 3.3.2.1 ซึ่งจะขอยกไปอธิบายในข้อ 3.3.2

- ตรวจสอบว่าเป็นไม้โทได้โดยดูทางซ้ายว่ามี 2 ยอดหรือไม่ตามรูปที่ 3.49 (ที่ต้องแยกไม้โทก่อนการหาลักษณะปากนกเพราะไม้โทอาจให้ลักษณะปากนกหรือไม่ก็ได้ซึ่งทำให้ถึงหาลักษณะปากนกไปก็ไม่มีประโยชน์ จึงยอมเสียเวลาคำนวณก่อน)



รูปที่ 3.49 แสดงลักษณะบ่งความต่างของไม้โท

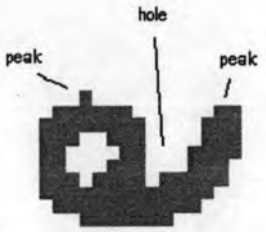
- การตรวจสอบลักษณะปากนกอธิบายได้ดังรูปที่ 3.50



รูปที่ 3.50 แสดงการหาลักษณะบ่งความต่างชนิดปากนก

โดยถ้าส่วนที่ยื่นออกมาทางซ้ายนั้นต่ำพอเทียบกับหลังคาของสระจึงตัดสินว่าเป็นปากนก

- ตรวจว่าเป็นไม้หันอากาศหรือไม่ด้วยการดู top [x] ว่ามีลักษณะดังรูปที่ 3.51 หรือไม่



รูปที่ 3.51 แสดงการตรวจไม้หันอากาศ

3.3.3.2 การรู้จำอักษรที่ได้รับจากตัวตัดแยกลักษณะที่ 1 โดยผ่านการตัดแบบ *"ความสูงเกิน"*

*ส่วนบน*ได้แก่ ่ ้ ็ ์ จะส่งให้ข้อ 3.3.1 ตามปกติแต่มีคำสั่งจำกัดคำตอบให้อยู่ในกลุ่ม

*ส่วนล่าง*ได้แก่ ุ ู ฺ

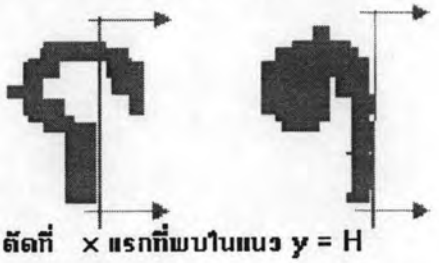
เราใช้การแยกไม้หันอากาศตามรูปที่ 3.51 จากนั้นใช้การตรวจ top [x] เพื่อแยกที่เหลือออกจากกันโดยทำการหาจุดยอด โดยดูตำแหน่งและจำนวน ดังรูปที่ 3.52

รูปที่ 3.52 แสดงลักษณะบ่งความต่างที่แยก อิ อี อึ อื ออกจากกัน

3.3.3.3 การรู้จำอักษรที่ได้รับจากตัวตัดแยกลักษณะที่ 2 ได้แก่ ไ ใ ๅ

รูปที่ 3.53 แสดงสระ ไ ใ ๅ ก่อนและหลังการตัดแยกในบทที่ 3

เราดูการเปลี่ยนแปลงของ top [x] เหมือนกับการดูแบบของหลังคาโดยถ้าเป็นหลังคาแบบหยักให้ตอบเป็นไม้มลาย ถ้าเป็นหลังคาธรรมดาให้ทำดังรูปที่ 3.54 ถ้ามีหางตอบ "โ"



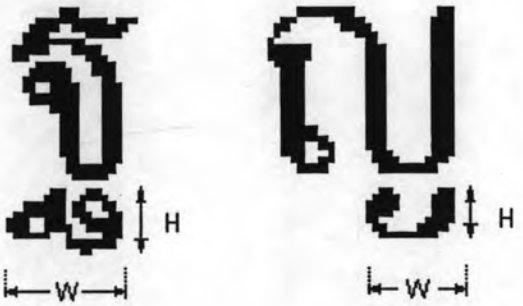
รูปที่ 3.54 แสดงลักษณะบ่งความต่างระหว่างสระ โ กับ ใ

3.3.4 การรู้จำอักษรระดับล่าง

สำหรับอักษรระดับล่างนั้นแบ่งที่มาได้ 2 แบบคือ
- จากลักษณะที่ 5 โดยตรง ได้แก่ สระ อุ อู ส่วนล่างของ ญ ฐ
- ได้รับจากทั้งตัวตัดแยกลักษณะที่ 4 ได้แก่ สระ อุ อู ส่วนล่างของ ญ ฎ ฏ ฐ ฤ ฦ

3.3.4.1 การรู้จำอักษรที่ได้รับจากลักษณะที่ 5 โดยตรง ได้แก่ สระ อุ อู ส่วนล่างของ ญ ฐ

ลักษณะส่วนล่างของ ฐ และ ญ จะเห็นว่ามีความกว้างมากกว่าความสูงมากดังนั้นถ้าพบว่ากว้างมากกว่าสูงมาก ให้ตอบเป็น Suspect_ฐ_ญ



รูปที่ 3.55 แสดงความกว้างและสูงของส่วนล่างของอักษร ฐ และ ญ

จากนั้นเมื่อส่วนตรวจสอบหลังการรู้จำทั้งบรรทัดมาตรวจสอบ ถ้าพบว่าอักษรระดับกลางที่อยู่บนอักษรลักษณะที่ 5 นั้นเป็น ฐ (อาจเป็น จ หรือ ช ได้เพราะมีลักษณะใกล้เคียงกัน) รวมอักษรทั้ง 2 เป็น ฐ หรือถ้าพบว่าอักษรที่อยู่บนเป็น ญ (อาจเป็น ณ ได้เพราะมีลักษณะใกล้เคียงกัน) รวมอักษรทั้ง 2 เป็น ญ

จากนี้ตัวที่เหลือคือ สระอุ กับ สระอู เราทำการหา vertical projection ถ้ามี 1 ยอดตอบเป็นสระอุ แต่ถ้าพบว่ามี 2 ยอดอาจมีส่วนล่างของ ญ ปนมา จะมีการตรวจสอบในลักษณะเหมือนกรณีหาปากนก แต่กลับกันเป็นปลายแหลมอยู่ช่วงบน ถ้าปลายอยู่ช่วงบนตอบเป็นสระอู ถ้าไม่ตอบเป็น ญ

เห็นได้ว่าสามารถทำได้ง่ายเนื่องจากมีจำนวนตัวอักษรที่ต้องแยกจำนวนน้อย ซึ่งเป็นเหตุผลให้เราต้องแยกการรู้จำอักษรระดับล่างตามที่มา

3.3.4.2 การรู้จำอักษรที่ได้รับจากตัดแยกลักษณะที่ 4 **ได้แก่ สระ อุ อู ส่วนล่างของ ญ ฎ ฏ ฐ ฤ ฦ**

เราใช้ประโยชน์จากการรู้จำระดับกลางด้วย โดยเราแยกหางแบบธรรมดาก่อนด้วยการตรวจสอบความกว้างโดยทำการหาความกว้างเฉลี่ยและถ้าไม่ต่างกันมากให้เป็นหางแบบธรรมดาแล้วตรวจสอบสวนบนต่อว่าเป็น ถ หรือ ภ หรือไม่ ถ้าเป็นเปลี่ยน ASCII ของ ถ เป็น ฤ และ ภ เป็น ฦ ถ้าไม่เป็นตอบว่าเป็นสระอุ เพราะใกล้เคียงที่สุด

จากนั้นทำการตรวจสอบเหมือน 3.4.1 เนื่องจากส่วนล่างของ ฎ และ ฏ คล้ายส่วนล่างของ ฐ เราจะให้เป็นคำตอบเดียวกัน จากนั้นถ้าส่วนบนเป็น ภ ให้แก้เป็น ฎ แล้วดูว่าเป็น ฎ หรือ ฏ โดยใช้ฐานข้อมูลทางภาษา