การแยกตัวอักษรภาษาไทยที่ติดกัน

SEGMENTATION OF CONNECTED THAI CHARACTERS

จรรยา เกียรติศิริอนันต์

CHANYA KERATSIRIANAN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2543

ISBN 974-622-826-9

การแยกตัวอักษรภาษาไทยที่ติดกัน

SEGMENTATION OF CONNECTED THAI CHARACTERS

จรรยา เกียรติศิริอนันต์

CHANYA KERATSIRIANAN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2543

ISBN 974-622-826-9

SEGMENTATION OF CONNECTED THAI CHARACTERS

CHANYA KERATSIRIANAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN COMPUTER SCIENCE

AND INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUE OF TECHNOLOGY LADKRABANG

2000

ISBN 974-622-826-9

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การแยกตัวอักษรภาษาไทยที่ติดกัน

SEGMENTATION OF CONNECTED THAI CHARACTERS

ชื่อนักศึกษา นางสาวจรรยา เกียรติศิริอนันต์

รหัสประจำตัว 36064005

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.ดร.บุญธีร์ เครือตราชู

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.บุญธีร์	เครือตราชู	
รศ.ดร.วิเชียร	เปรมชัยสวัสดิ์	
รศ.ดร.ชม	กิ้มปาน	
ดร.วรพจน์	กรีสุระเดช	
รศ.ดร.บุญวัฒน์	อัตชู	

วัน/เดือน/ปี ที่สอบ 27 เมษายน 2543 เวลา 10.00 น. เป็นต้นไป

สถานที่สอบ ณ ห้อง 234 – 235 ชั้น 2 อาคารสำนักวิจัยและบริการคอมพิวเตอร์

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.มนัส สังวรศิลป์)

คณบดีบัณฑิตวิทยาลัย

วันที่ 31 **เดือน** พฤษภาคม **พ.ศ.** ๒๕๔๓

หัวข้อวิทยานิพนธ์	การแยกตัวอักษรภาษาไทยที่ติดกัน
นักศึกษา	นางสาวจรรยา เกียรติศิริอนันต์
รหัสประจำตัว	36064005
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
พ.ศ.	2543
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.บุญธีร์ เครือตราชู

บทคัดย่อ

การวิจัยนี้มุ่งสร้างระบบการแยกตัวอักษรภาษาไทย ที่มีลักษณะการติดกันในแนวนอนซึ่งมีจำนวนการติดของตัวอักษรตั้งแต่ 2 ตัวอักษรขึ้นไป เช่น รท ณะน มมาทนก เป็นต้น โดยใช้วิธี การหาเส้นทางเดินที่สั้นที่สุด (Shortest Path)[4] ซึ่งได้มีการทดลองมาแล้วในภาษาอังกฤษ มาลองใช้ในภาษาไทยพบว่ามีความผิดพลาดเกิดขึ้น และได้เสนอวิธีการแก้ไขเป็นแบบเฉียงแต่ก็ยังคงพบปัญหาอีกจึงทำการแก้ไขอีกโดยใช้วิธี Histogram การแสดงผลจะเป็นการสรุปความเป็นไปได้ในการแยกตัวอักษรในแต่ละวิธี ตัวอักษรที่ใช้เป็นตัวอย่างติดกันแน่นอน 2 ตัวอักษรทั้งหมด 381 ตัวอย่าง ในวิธี Shortest Path ตัดถูกต้อง 57.70% ในวิธี Histogram ตัดถูกต้อง 91.08 % และเฉพาะในวิธี Histogram ตัวอย่างที่เป็นอักษร 1 ตัวมีจำนวน 200 ตัวอย่างตัดถูกต้อง 92.00 % และตัวอย่างที่มีติดกันมากกว่า 2 ตัวอักษรขึ้นไปมีอยู่ 109 ตัวอย่าง ตัดถูกต้อง 82.57%

Thesis Title	Segmenttation of connected Thai Characters
Student	Miss.Chanya Keiatsirianan
Student ID.	36064005
Degree	Master of Science
Programme	Computer Science and Information Thecnology
Year	2000
Thesis Advisor	Assoc.Prof.Dr.Boontee Kruatrachue

ABSTRACT

This thesis attempts to isolate touching Thai characters in vertical detection from the scan image. The Touching character can be more than two characters for example รท ณะน มมาทนท. There are 490 examples from many articles and newspaper. The method used to isolate character is Shortest Path[4] which tested successfully on English character. The modified Shortest Path is present which some improvement result. For testing, uses 381 of two touching characters in Shortest Path segment correct 57.70 % and in Histogram segment correct 91.08 %. For testing in Histogram, 200 of one character segment correct 92.00 % and 109 of more than two touching characters segment correct 82.57 %.

กิตติกรรมประกาศ

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ครู อาจารย์ ที่เคยอบรมสั่งสอนเป็นอย่างสูงและให้ความช่วยเหลือพร้อมทั้งสนับสนุนด้านการศึกษามาโดยตลอด

ขอกราบขอบพระคุณเป็นอย่างสูงในความกรุณาของ ผศ.ดร.บุญธีร์ เครือตราชู ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้ซึ่งทุ่มเทแรงกายแรงใจในการควบคุมงานวิจัย ให้คำแนะนำในการแก้ปัญหาและข้อบกพร่องต่าง ๆ เป็นอย่างดียิ่ง

ขอขอบพระคุณ อ.ณัฐกร ทับทอง ภาควิชาฟิสิกส์ จุฬาลงกรณ์มหาวิทยาลัย ที่ช่วยเหลือแก้ไขและให้คำแนะนำแก่ผู้วิจัยเกี่ยวกับปัญหาของโปรแกรมภาษา C

ขอขอบคุณ คุณบุญช่วย ชาติทอง และเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศทุกท่าน ที่ช่วยประสานงานและให้ความสะดวกในทุกด้าน ตลอดระยะเวลาที่ผู้เขียนศึกษา ณ. สถาบันแห่งนี้

ขอขอบคุณ คุณเชิดชู ชัยขจรภัทร์ ที่ช่วยเหลือในด้านการจัดทำรูปเล่ม และให้คำปรึกษาในการแก้ปัญหา พร้อมทั้งให้กำลังใจในการทำงานต่าง ๆ ให้ลุล่วงไปด้วยดี

ขอขอบคุณบุคคลในครอบครัว และเพื่อน ๆ ทุกท่าน ที่ช่วยเหลือให้กำลังกาย และกำลังใจในการทำงานต่าง ๆ ให้ลุล่วงไปด้วยดี

สุดท้ายขอขอบคุณบัณฑิตวิทยาลัย ที่ได้ให้ทุนสนับสนุนการทำวิทยานิพนธ์ครั้งนี้

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

จรรยา เกียรติศิริอนันต์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

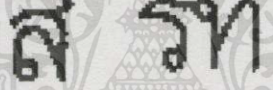
สารบัญรูป (ต่อ)

บทที่ 1

บทนำ

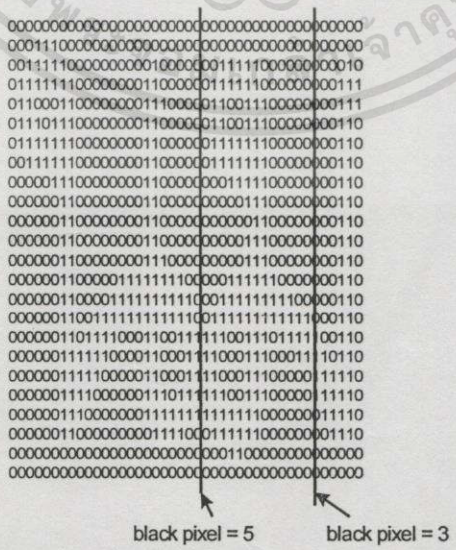
1.1 ความเป็นมาของงานวิจัย

จากการศึกษาในระบบของการจดจำตัวอักษรภาษาไทย (Thai OCR) พบว่าจะมีปัญหาอยู่ปัญหาหนึ่งที่พบอยู่เสมอคือ การมีตัวอักษรติดกัน (Touching character) เนื่องจากว่าภาษาไทยเป็นภาษาที่มีระดับมีทั้งสระ พยัญชนะ และวรรณยุกต์เป็นส่วนประกอบ ดังนั้นเมื่อนำไปใช้จะทำให้เกิดการติดกันได้มาก และหลายรูปแบบ ลักษณะการติดอาจจะติดกันในแนวดิ่ง (horizontal) เช่น สิ ยุ หรืออาจจะติดกันในแนวนอน (vertical) เช่น รน เม ดังแสดงไว้ในรูปที่ 1.1 จากการศึกษามาพบว่าการติดกันในแนวดิ่งนั้น ถ้าทราบเส้นบรรทัดก็จะสามารถแยกตัวอักษรที่ติดกันนั้นได้ง่าย แต่การติดกันในแนวนอนไม่สามารถใช้เส้นบรรทัดแยกออกได้ในทีเดียวทำให้มีความยากกว่าการแยกในแนวดิ่ง จึงทำให้งานวิจัยนี้จะเน้นการแยกตัวอักษรที่ติดกันในแนวนอนเป็นหลัก และได้ศึกษาการแยกตัวอักษรที่ติดกันมากกว่า 2 ตัวอักษรขึ้นไปอีกด้วย



รูปที่ 1.1 แสดงตัวอย่างการติดกันของตัวอักษรในลักษณะต่าง ๆ

และจากการศึกษามาพบว่าจุดตัดที่ใช้ตัดตัวอักษรออกจากกันมักจะใช้บริเวณที่เป็นค่าต่ำสุดซึ่งในความจริงจุดที่เป็นค่าต่ำสุดนี้ อาจจะไม่ใช่จุดตัดที่ถูกต้องเสมอไปดังแสดงในรูปที่ 1.2



รูปที่ 1.2 แสดงจุดตัดตัวอักษรที่ไม่ใช่บริเวณที่มีค่าต่ำสุด

จากปัญหาที่กล่าวมาข้างต้นทำให้งานวิจัยนี้ศึกษาการแยกตัวอักษรที่ติดกันในแนวนอน และมีจำนวนตัวอักษรที่ติดกัน 2 ตัวอักษร ตัวอักษรเดี่ยวๆ และตัวอักษรที่ติดกันมากกว่า 2 ตัวอักษรขึ้นไป พร้อมทั้งทำการหาจุดตัดที่ถูกต้องที่สุดเป็นจุดแยกของตัวอักษร

1.2 วัตถุประสงค์ของงานวิจัย

1) เพื่อศึกษาลักษณะการติดกันของตัวอักษรพร้อมทั้งวิธีต่าง ๆ ที่ใช้แยกตัวอักษรนั้น เพื่อนำมาประยุกต์ใช้ในภาษาไทย
2) เพื่อศึกษาปัญหาที่เกิดขึ้นพร้อมทั้งหาแนวทางในการแก้ไข เพื่อนำไปใช้งานได้อย่างมีประสิทธิภาพ
3) เพื่อเป็นแนวทางในการพัฒนาระบบการแยกตัวอักษรที่ติดกันต่อไปในอนาคต
4) เพื่อเปรียบเทียบวิธีการแยกตัวอักษรโดยวิธี Shortest path กับวิธี Histogram

1.3 ขอบเขตของงานวิจัย

1) การวิจัยนี้มุ่งศึกษาพัฒนาบนเครื่องไมโครคอมพิวเตอร์
2) กลุ่มตัวอักษรที่ใช้เป็นตัวอย่างประกอบด้วย ตัวอักษรภาษาไทยทั้งหมด, สระ ะ า เ ฤ ฦ, เลขอารบิค (0-9)
3) การวิจัยนี้เลือกศึกษาเฉพาะบทความภาษาไทยที่พบในหนังสือทั่ว ๆ ไปเช่น หนังสือพิมพ์ วารสาร ฯลฯ และมีลักษณะตัวอักษรที่ใช้เป็นตัวอย่างตัวตรงไม่เอียง ไม่หนา และเป็น font เดียวกันทั้งบทความ
4) ใช้ scanner ที่มี resolution 300 dpi เป็นตัว scan ข้อมูลที่ใช้เป็นตัวอย่าง
5) ข้อมูลตัวอย่างที่ใช้ทำการทดลองในงานวิจัยนี้ทั้งหมดจะเลือกตัวอักษรที่เห็นว่ามีการติดกันจากบทความในข้อ 3 ด้วยตาโดยใช้ Photoshop เป็นตัวช่วย
6) โปรแกรมที่ใช้เขียนเป็นโปรแกรมภาษา C

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ทราบถึงบทความต่าง ๆ ที่เกี่ยวข้อง รวมทั้งลักษณะการติดกันของตัวอักษรภาษาไทย
2) ทราบถึงทฤษฎีเกี่ยวกับ Shortest Path ที่นำไปใช้แก้ปัญหาจากภาษาอังกฤษสู่ภาษาไทย
3) ทราบลักษณะทาง Histogram ของตัวอักษรภาษาไทยและวิธีการนำไปใช้ในการแก้ปัญหาสามารถนำไปใช้แก้ปัญหาการติดกันของตัวอักษรในระบบ Thai OCR ได้

1.5 ขั้นตอนของงานวิจัย

1) ศึกษาบทความต่าง ๆ ที่มีความเกี่ยวข้องกับงานวิจัยนี้
2) เก็บข้อมูลตัวอย่างจากหนังสือทั่ว ๆ ไป เช่น หนังสือพิมพ์ พร้อมทั้งนำไปผ่าน scanner
3) นำข้อมูลที่ได้ในข้อ 2. ผ่านโปรแกรม Photoshop เพื่อหาตัวอย่างที่มีลักษณะติดกันเก็บเป็นตัวอย่างที่ใช้ทำการทดลอง
4) ศึกษาลักษณะการติดกันของตัวอักษรที่ใช้เป็นตัวอย่าง
5) เลือกตัวอย่างที่จะใช้ทดลองจากข้อมูลในข้อ 4. เพื่อใช้เป็นตัวอย่างจริงในงานวิจัย
6) เขียนโปรแกรม Shortest path พร้อมทั้งทดลองกับตัวอย่างที่มีอยู่
7) แก้ไขข้อผิดพลาดจากวิธี Shortest path พร้อมทั้งทดลองกับตัวอย่าง
8) ศึกษาวิธี Histogram เพื่อใช้แก้ปัญหาที่ยังแก้ไม่ได้จากวิธี Shortest path
9) เขียนโปรแกรม Histogram เพื่อแก้ไขปัญหา พร้อมทั้งทดสอบกับตัวอย่างที่มีอยู่
10) แก้ไขโปรแกรม Histogram เพื่อให้แก้ไขปัญหาต่าง ๆ ได้มากที่สุด
11) สรุปผลที่เป็นไปได้ทั้งหมด พร้อมทั้งทำการเสนอเป็นงานวิจัย

บทที่ 2
งานวิจัยที่เกี่ยวข้อง

2.1 การแยกตัวอักษรภาษาไทยที่ติดกันด้วยลักษณะเฉพาะของตัวอักษร[3]

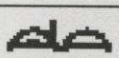
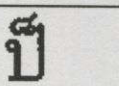
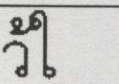
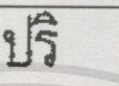
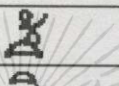
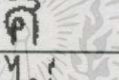
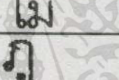
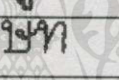
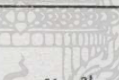
ในบทความนี้เสนอวิธีการในการตรวจสอบ และการแยกตัวอักษรภาษาไทยที่ติดกันโดยใช้ลักษณะเฉพาะของตัวอักษรภาษาไทยเป็นตัวช่วยในการแบ่งกลุ่ม และเป็นเงื่อนไขในการแยกแต่ละกลุ่มออกจากกัน

ในบทความได้ทำการแบ่งกลุ่มตามลักษณะของตัวอักษรภาษาไทยแต่ละตัวเป็นดังตารางที่ 2.1 และจากตารางที่ 2.1 สามารถแบ่งเป็นกลุ่มย่อยได้เป็น 10 กลุ่มตามลักษณะการติดกันของตัวอักษร เป็นดังตารางที่ 2.2 ซึ่งข้อมูลที่ใช้ในการทดลองได้มาจาก Microsoft Word 6 font AngsanaUPC และ CordiaUPC ที่มีขนาด 12 14 และ 16 โดยพิมพ์จาก Laser Jet 5L จากตารางพบว่าเปอร์เซ็นต์ที่ติดกันในแนวดิ่ง (แบบที่ 5 อี๋) มีมากกว่าที่ติดกันในลักษณะอื่น ๆ

ตารางที่ 2.1 แสดงการแบ่งกลุ่มของตัวอักษร

กลุ่ม	ระดับ	ตัวอย่าง
1	ด้านบน	๒๘๙๘๑๙๗+๑๗
2	ด้านบน+ตรงกลาง	ป ฝ ฟ ฟ ไ ไ โ ๒ ๖ ๗ ๘
3	ตรงกลาง	ฯ ะ เ า ก ข ฆ ค ต ฆ ง จ ฉ ช ซ ฌ ด ต ถ ท ธ น บ ผ พ ภ ม ย ร ล ว ศ ษ ส ห อ ฮ ๐ ๑ ๓ ๔ ๕ ๘
4	ด้านล่าง+ตรงกลาง	ฦ ฤ ๅ ญ ฎ ฏ ฐ
5	ด้านล่าง	ุ ู
6	เหนือด้านบน	่ ้ ๊ ๋ ๗ + ์

ตารางที่ 2.2 แสดงการแบ่งกลุ่มตามลักษณะการติด

กลุ่ม	กลุ่มอักษรที่ติดกัน	ตัวอย่าง	เปอร์เซ็นต์ที่พบ
1	1&1		0.2
2	1&2 (คอลัมน์เดียวกัน)		14.5
3	1&2 (ต่างคอลัมน์)		9.2
4	2&1 (ต่างคอลัมน์)		0.2
5	1&6		38.1
6	1&3		5.9
7	2&3		0.2
8	3&5		31.0
9	3&3		0.2
10	เกิน 2 ตัวอักษร		0.5

ในการศึกษาด้วยวิธีนี้จะมีการหาขอบเส้นบรรทัดด้านบนและด้านล่างของตัวอักษรด้วยวิธี horizontal projection ซึ่งจะมีลักษณะคล้ายภูเขา เส้นบรรทัดด้านบนและด้านล่างจะอยู่บริเวณจุดต่ำสุดหรือบริเวณตีนเขาและบริเวณตรงกลางจะเป็นเนื้อของตัวอักษร หลังจากนั้นแล้วสามารถแบ่งกลุ่มตามขอบเขตเส้นบรรทัดได้เป็นดังตารางที่ 2.3

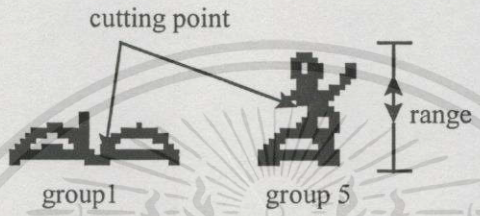
ตารางที่ 2.3 แสดงกลุ่มตามขอบเขตเส้นบรรทัด

กลุ่ม	กลุ่มที่มีโอกาสติดกันจากตาราง 2.2	ลักษณะตัวอักษรในกลุ่ม
1	1,5,10	◌ั ◌ิ ◌ี ◌ึ ◌ื ◌ุ ◌ู ◌็ ◌่ ◌๊ ◌๋
2	2,3,4,6,7,10	ป ฝ ฟ พ ไ ใ โ ๒ ๖ ๗ ๘
3	9,10	ตัวอักษรในกลุ่มที่ 3 ในตารางที่ 1
4	8,10	ฦ ฤ ๆ ฯ ญ ฎ ฏ ฐ
5	-	ฃ ฆ

ในบทความนี้ใช้วิธี Distinctive features เป็นวิธีในการแยกตัวอักษรที่ติดกัน โดยวิธีนี้จะแบ่งเป็นกลุ่มตามตารางที่ 2.3 ซึ่งในบทความนี้ยกตัวอย่างมาให้พิจารณา 2 กลุ่มคือ

2.1.1 กลุ่มที่ 1

ในกลุ่มนี้มีกลุ่มที่ติดกันเป็นไปได้ 2 กลุ่มย่อยคือกลุ่มที่ 1 หรือ 5 ในตารางที่ 2.2 เนื่องจากว่าความกว้างของตัวอักษรที่ติดกันในกลุ่มที่ 1 มากกว่าในกลุ่มที่ 5 ซึ่งมีความสูงมากกว่า ดังแสดงได้รูปที่ 2.1 ทำให้สามารถใช้ความแตกต่างที่เป็นอัตราส่วนของความกว้างและความสูงต่อความสูงของตัวอักษรเป็นตัวแยกทั้ง 2 กรณีออกจากกัน

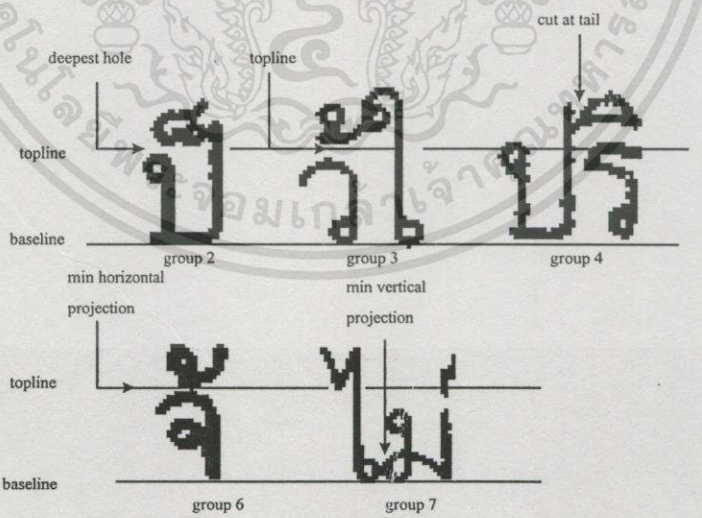


รูปที่ 2.1 แสดงการติดกันของกลุ่มที่ 1 หรือ 5

จุดตัดของกลุ่มที่ 1 จะอยู่บริเวณตรงกลางระหว่างตัวอักษร และจุดตัดของกลุ่มที่ 5 จะเป็นบริเวณที่น้อยที่สุดของ horizontal projection

2.1.2 กลุ่มที่ 2

กลุ่มที่ติดกันเป็นไปได้ในกลุ่มนี้คือ กลุ่มที่ 2,3,4,6 หรือ 7 ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แสดงตัวอย่างการติดกันในกลุ่ม 2,3,4,6 และ 7

การหาจุดตัดเราจะพิจารณาตามตารางที่ 2.4 โดยพิจารณาตาม tree ในรูปที่ 2.3

ตารางที่ 2.4 แสดงลักษณะเด่นที่ใช้แยกในกลุ่มที่ 2

Distinctive feature	group of connected characters							
	2	3	4	6	7	*	@	$
wider than 1 character	x	x	/	x	/	x	x	x
middle is "เ"	x	/	0	x	0	x	/	0
middle right has hole	0	0	/	0	x	0	0	0
upper has a vowel or a tone	/	/	0	/	0	x	x	0
tail on left	x	x	0	x	0	x	x	/
flat right	/	0	0	x	0	0	0	0

* ตัวอักษร ป ฝ ฟ พ ๗ ๘

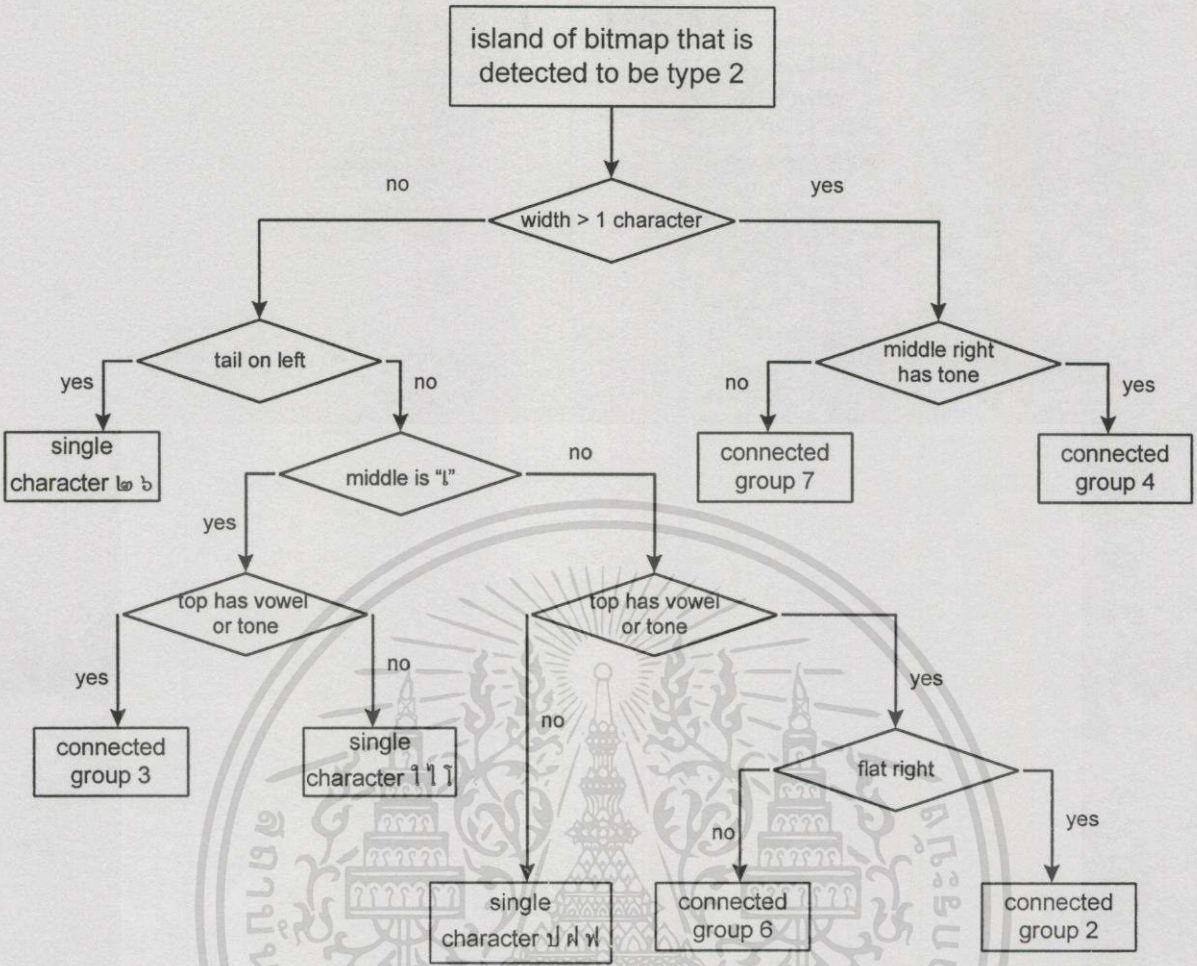
@ ตัวอักษร ใ ไ โ

$ ตัวอักษร ๒ ๖

x ไม่มี

/ มี

0 ไม่สนใจ



รูปที่ 2.3 แสดง tree ตามตารางที่ 2.4

ผลการทดลองของบทความนี้เป็นดังตารางที่ 2.5 และจากตารางพบว่าเปอร์เซ็นต์ถูกต้องประมาณ 95.6 % ซึ่งมีเปอร์เซ็นต์ความผิดพลาดอยู่ 5 % เนื่องมาจากความผิดพลาดดังนี้

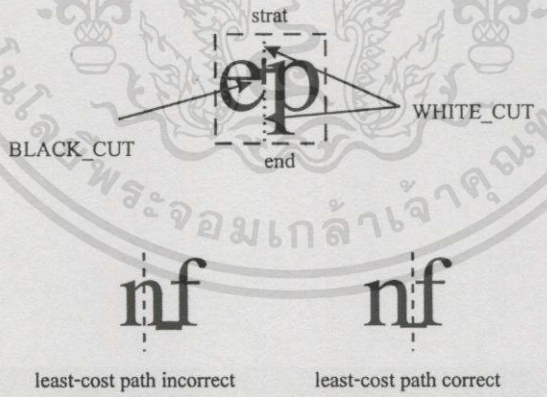
1) ความผิดพลาดที่เกิดจากความคล้ายกันจนแยกไม่ได้เช่น ฎ ที่คล้ายกับ ฎ
2) ความผิดพลาดที่เกิดจากหาจุดตัดผิดเมื่อมี noises หรือ overlapped characters การเกิด overlapped กันจะทำให้เวลาตัดบริเวณที่เป็นจุดตัดจะทำให้ตัวอักษรของแต่ละตัวไม่ถูกต้อง คือจะมีบางส่วนขาด บางส่วนเกิน

ตารางที่ 2.5 แสดงผลการทดลอง

ขนาดตัวอักษร	12	14	16
ตัวอักษรทั้งหมด	6000	6000	6000
ตัวอักษรที่ติดกัน	456	288	78
% ของตัวอักษรที่ติดกัน	7.6	4.8	1.3
จุดตัดที่ผิดพลาด	20	10	4
% ความถูกต้อง	95.6	96.5	94.8

2.2 การแยกตัวอักษรที่ติดกันโดยวิธี Neural Networks และ Shortest Path[4]

ในบทความนี้จะเป็นบทความที่เกี่ยวกับการจดจำตัวอักษรภาษาอังกฤษ โดยใช้ Neural Networks พร้อมทั้งมีการเสนอการแก้ปัญหาตัวอักษรที่ติดกันด้วยวิธี Shortest Path วิธี Shortest Path ที่กล่าวถึงนี้ เป็นการพิจารณาส่วนของตัวอักษรที่ติดกันดังแสดงในรูปที่ 2.4 โดยกำหนดขนาดของ bitmap เป็น n rows by m columns หรือ n x m กำหนดจุด strat และ end ภายนอก bitmap แสดงดังรูปที่ 2.4 และกำหนดว่า cost จากจุด start ไปจุดต่าง ๆ ใน row แรก หรือจากจุด end ไปจุดต่าง ๆ ใน row สุดท้ายมีค่าเป็น 0 ปัญหาการหาจุดตัดแก้ได้จากการค้นหา path ที่สั้นที่สุดจากจุด start ไปจุด end



รูปที่ 2.4 แสดงตัวอย่าง Shortest Path

การหา path จากจุด start ไปจุด end จะไปทุก ๆ จุดใน bitmap โดยมีการพิจารณาตามเงื่อนไขในการเดินจากจุดหนึ่งไปอีกจุดหนึ่ง ส่วนต่าง ๆ ของตัวอักษรที่ติดกัน 1 ตัว จะสามารถติดกับตัวอื่น ๆ ได้ 8 ด้านแต่การเคลื่อนที่หา path จะต้องเดินไปข้างหน้าไม่ถอยหลังจึงทำให้เหลือทางเดินที่เป็นไปได้ 3 ด้านคือตรงกลาง ซ้าย และขวา โดยการที่จะเดินไปทางใดจะพิจารณาตามเงื่อนไขดังนี้

1) ถ้าจุดตรงกลางเป็นจุดขาวค่า cost จะไม่คิด แต่ถ้าเป็นจุดดำค่า cost จะมีค่าเป็น 10

2) ถ้าจุดซ้ายหรือจุดขวาเป็นจุดขาวค่า cost จะมีค่าเป็น 1 แต่ถ้าเป็นจุดดำจะมีค่าเป็น $10\sqrt{2}$

3) การเดินจากจุด start จะต้องเดินตรงอย่างเดียว

ในการหา path ที่สั้นที่สุดจะเริ่มจากจุด start ลงมายังจุดใน row แรกเดินไปยัง row สุดท้าย โดยการเดินไปในแต่ละ row ถัดไปจะพิจารณาตามเงื่อนไขด้านบน ทางที่จะเดินต่อจะต้องมีค่า cost น้อยที่สุดในขณะนั้น หลังจากเลือกทางเดินจุดต่อไปได้จะทำการเก็บค่า cost ที่ได้ในจุดนั้นเป็นค่า cost รวม เมื่อเดินจนถึง row สุดท้ายก็จะทำการเลื่อน column ต่อไปจนหมดทุก column ซึ่งในแต่ละ column ก็จะมี path ที่มีค่า cost รวมต่าง ๆ กัน path ใดที่มีค่าcost รวมน้อยที่สุดจะถือว่าเป็น path ที่สั้นที่สุด (shortest path) ซึ่งจะเป็น path ที่ใช้เป็นเส้นตัดตัวอักษร

ผลในการตรวจสอบความถูกต้องจะนำส่วนที่ตัดได้ผ่านระบบ OCR ถ้าระบบรับรู้ว่าถูกต้องก็จะสามารถตัดตัวต่อไป แต่ถ้าไม่ถูกต้องก็จะต้องนำมาพิจารณาหา path ใหม่ ดังแสดงในรูปที่ 2.4 path ที่เลือกอาจจะไม่ใช่ path ที่ถูกต้องเสมอไปเนื่องจากว่าถ้าบริเวณอื่นมีค่า cost รวมน้อยกว่า หรือเรียกว่ามีลักษณะบางกว่าบริเวณที่ติดจริงก็จะทำให้เลือก path ที่ผิดได้

2.3 การแยกตัวอักษรที่ติดกันของ Printed Document Recognition[7]

การแยกตัวอักษรที่ติดกันด้วยวิธีนี้จะมีการกำหนดฟังก์ชั่นสำหรับตัวอักษรที่ติดกันจากอัตราส่วนของความแตกต่างลำดับ 2 ของ vertical pixel projection กับค่า vertical projection จากรูปที่ 2.5 แสดงตัวอย่างตัวอักษรที่ติดกัน "OO" การหาฟังก์ชั่นการแบ่งจะขึ้นอยู่กับ profile projection และ pixel projection ซึ่งการหา profile projection และ pixel projection สามารถกระทำได้ดังนี้

pixel projection จะเป็นการรวมจุดดำในแต่ละ column ให้เป็น PXP(k), เมื่อ k=1,2,...,LT และ LT เป็นความกว้างของตัวอักษรที่ติดกัน

profile projection จะเป็นการรวมจุดทั้งหมดที่พบเริ่มจากจุดดำจุดแรกจนถึงจุดดำที่พบจุดสุดท้ายในแต่ละ column ให้เป็น PFP(k)=TP(k)-BP(k), เมื่อ k=1,2,...,LT ซึ่ง TP เป็นจุดบนสุดของตัวอักษรที่ติดกัน และ BP เป็นจุดล่างสุดของตัวอักษรที่ติดกัน

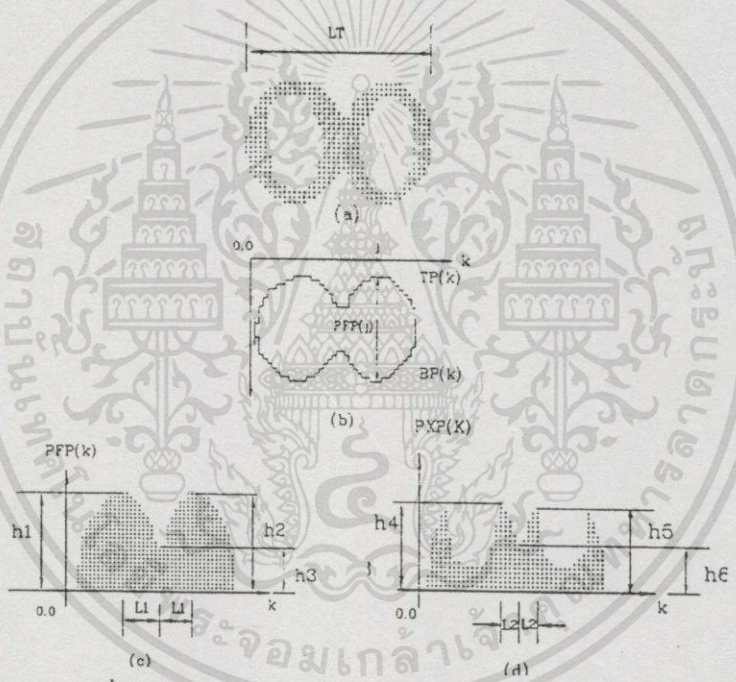
จากรูปที่ 2.5 แสดงฟังก์ชั่นการแบ่งที่ได้จาก pixel projection และ profile projection ตามสมการดังนี้

$$F^{\alpha}(k) = \left[\frac{PFP(k+L1)-2PFP(k)+PFP(k-L1)}{PFP(k)} \right]^{\alpha} \quad (2.1)$$

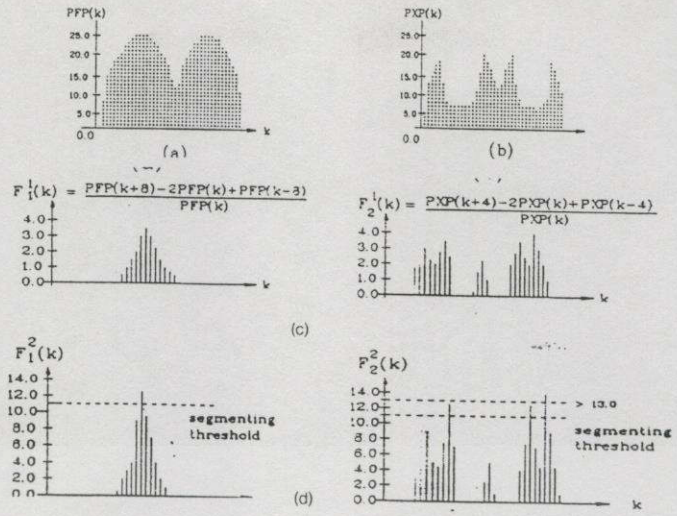
$$F^{\alpha}(k) = \left[\frac{PXP(k+L2)-2PXP(k)+PXP(k-L2)}{PXP(k)} \right]^{\alpha} \quad (2.2)$$

เมื่อค่า L1 และ L2 เป็นระยะระหว่าง column ปัจจุบัน กับ column ที่อยู่ใกล้ L1 กับ L2 ตัวอย่างการกำหนด L1และL2 เช่นถ้าตัวอักษร “time” มี font ขนาด 11 จะใช้ L1=8 และ L2=4 ส่วนค่า α จะเลือกค่าที่มากกว่า 1 จากรูปที่ 2.6 จะใช้ α =1 และ α=2 จุดตัดตัวอักษรที่ติดกันจะอยู่ในค่าฟังก์ชั่นการแบ่งที่เพิ่มขึ้นที่ α = 2 เมื่อ F1 และ F2 มากกว่า 2 จากรูปที่ 2.6 (c) และ(d) แสดงจุดตัดที่เป็นไปได้ 4 จุดสำหรับตัวอักษร “OO” ติดกัน ที่ threshold เท่ากับ 11 ในกรณีนี้จุดตัดที่ถูกต้องจะอยู่ในฟังก์ชั่น $F^2(k)$ การหาฟังก์ชั่นการแบ่งหาได้จาก

$$F(k) = \begin{cases} F^2(k), & \text{if } F^2(k) > T \;\&\; F^2(k) > T \\ F^2(k), & \text{if } F^2(k) > T \;\&\; F^2(k) < T \\ F^2(k), & \text{if } F^2(k) < T \;\&\; F^2(k) > T \\ 0, & \text{if } F^2(k) < T \;\&\; F^2(k) < T \end{cases} \qquad (2.3)$$

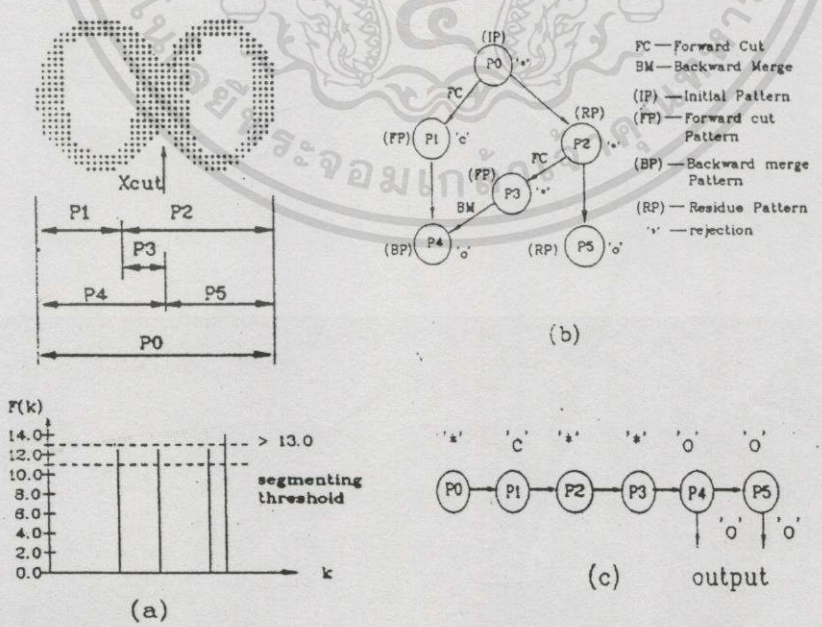


รูปที่ 2.5 (a) แสดงตัวอักษรที่ติดกัน “OO” (b) top และ bottom profiles
(c) projection profile PFP(k) (d) pixel projection PXP(k)



รูปที่ 2.6 (a) ตัวอักษรที่ติดกัน "OO" และ profile projection (b) ตัวอักษรที่ติดกัน "OO" และ pixel projection (c) ฟังก์ชั่นการแบ่งบน profile projection และ pixel projection ที่ $\alpha = 1$ (d) ฟังก์ชั่นการแบ่งบน profile projection และ pixel projection ที่ $\alpha = 2$

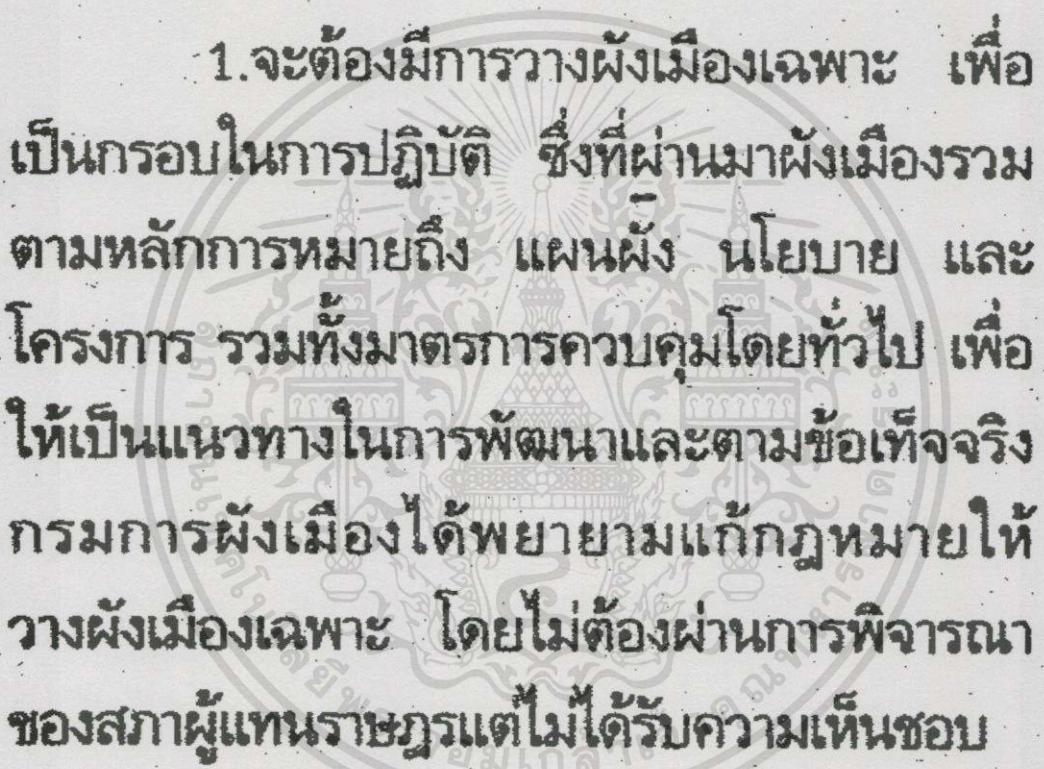
ในการหาจุดตัดจะได้จากการทำ dynamic recursive segmentation algorithm ซึ่งเป็นการใช้การแยกไปข้างหน้า หรือการรวมกลับหลัง ขึ้นอยู่กับข้อมูลออกของตัวอักษรที่ติดกัน เช่นในรูปที่ 2.7 แสดงการตัดตัวอักษร 2 ตัวคือ "OO" ในรูปที่ 2.7 (a) แสดงจุดตัดที่เป็นไปได้จากฟังก์ชั่นการแบ่ง F(k) จุดตัดจุดแรกจะแบ่งเป็น P1 กับ P2 ผ่านระบบจดจำจะได้ P1 เป็นตัวอักษร "C" ส่วน P2 ไม่เป็นตัวใดเลย P2 ก็จะถูกแบ่งออกเป็น P3 กับ P5 P3 ไม่มีตัวใดเลย ส่วน P5 ได้เป็นตัว "O" ดังนั้น P3 จะถูกรวมกับ P1 เป็น P4 ซึ่งตรวจได้ว่าเป็นตัว "O" เมื่อทั้ง 2 ข้างถูกจดจำได้จุดที่ได้ก็เป็นจุดตัด



รูปที่ 2.7 (a) segmented patterns (b) Graphic representation (c) Time sequence of segmentation

บทที่ 3
การเก็บข้อมูลตัวอักษรที่ติดกัน

ตัวอย่างที่ใช้ในการทดสอบได้มาจากการตัดบทความโดยทั่วไปจากหนังสือต่าง ๆ เช่นหนังสือพิมพ์, วารสาร ฯลฯ โดยลักษณะของบทความที่เลือกนั้นจะต้องมี font ของตัวอักษรแบบเดียวกัน ขนาดเท่ากัน และภายในบทความนั้นจะต้องเป็นตัวอักษรตัวตรงไม่เอียง ไม่หนา ดังรูปที่ 3.1 เป็นบทความบางส่วนที่ได้จากหนังสือวัฏจักรอาคารที่ดิน

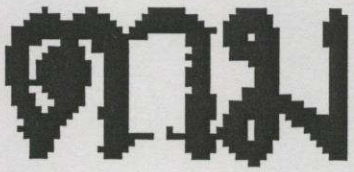

1.จะต้องมีการวางผังเมืองเฉพาะ เพื่อเป็นกรอบในการปฏิบัติ ซึ่งที่ผ่านมาผังเมืองรวมตามหลักการหมายถึง แผนผัง นโยบาย และโครงการ รวมทั้งมาตรการควบคุมโดยทั่วไป เพื่อให้เป็นแนวทางในการพัฒนาและตามข้อเท็จจริง กรมการผังเมืองได้พยายามแก้กฎหมายให้วางผังเมืองเฉพาะ โดยไม่ต้องผ่านการพิจารณาของสภาผู้แทนราษฎรแต่ไม่ได้รับความเห็นชอบ

รูปที่ 3.1 แสดงบทความจากหนังสือวัฏจักรอาคารที่ดิน

3.1 ขั้นตอนการเก็บตัวอักษรที่ติดกัน

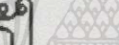
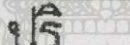
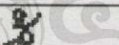
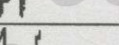
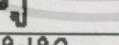
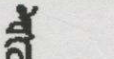
จากบทความข้างต้นนำมาผ่าน scanner ที่ resolution 300 dpi เก็บเป็นไฟล์ . tif แล้วนำไฟล์ที่ได้เข้าโปรแกรม Photoshop เพื่อหาตัวอักษรที่ติดกัน เก็บเป็นไฟล์ .bmp ดังแสดงในรูปที่ 3.2 จากการศึกษาลักษณะการติดกันของตัวอักษรทั้งหมดที่มีในบทความต่าง ๆ พบว่าลักษณะการติดกันจะเป็นไปตามตารางที่ 2.2 ในบทที่ 2 แต่เปอร์เซ็นต์ที่พบนั้นต่างออกไปดังแสดงไว้ในตารางที่ 3.1 จากตารางที่ 3.1 พบว่าการติดกันในแบบที่ 9 มีเปอร์เซ็นต์ที่พบสูงที่สุดและไม่สามารถแยกออกได้ด้วย

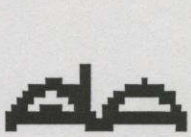
เส้นบรรทัดในทีเดียว ดังนั้นจึงทำการวิจัยในแบบนี้ ตัวอย่างตัวอักษรที่พบในตารางที่ 3.1 แสดงไว้ดังรูปที่ 3.3



รูปที่ 3.2 แสดงตัวอย่างไฟล์ .bmp

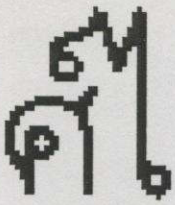
ตารางที่ 3.1 แสดงเปอร์เซ็นต์การติดกันของตัวอักษรจากบทความที่นำมาทดลอง

กลุ่ม	กลุ่มอักษรที่ติดกัน	ตัวอย่าง	% พบในบทความที่กล่าวถึง [3]	% พบจากหนังสือพิมพ์ทั่วไป
1	1&1		0.2	0.9
2	1&2 (คอลัมน์เดียวกัน)		14.5	11.2
3	1&2 (ต่างคอลัมน์)		9.2	8.9
4	2&1 (ต่างคอลัมน์)		0.2	0.7
5	1&6		38.1	9.4
6	1&3		5.9	13.8
7	2&3		0.2	2.7
8	3&5		31.0	3.4
9	3&3		0.2	39.0
10	เกิน 2 ตัวอักษร		0.5	10.1



group 1

group 2

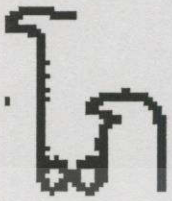


group 3

group 4

group 5

group 6



group 7

group 8

group 9

group 10

รูปที่ 3.3 แสดงตัวอย่างทั้ง10 กลุ่มในตาราง 3.1

จากไฟล์ .bmp จะนำไปผ่านโปรแกรม bitmap เพื่อทำให้เป็นภาพขาวดำ (.txt) ดังรูปที่ 3.4 ซึ่งเป็นรูปภาพขาวดำของตัวอักษร ร ที่ติดกับ ร

```
00000000000000000000000000000000
00000110000000000000000000000000
00011111111000000011111111100000
01111111111111101111111111111110
11100000011111111110000011111110
11100000000000111100000000000000
11111111110000111111111110000000
11111111111000011111111111000000
00000000111100000000000111100000
00000000001100000000000001100000
00000000001110000000000001110000
00000000000110000000000000110000
00000000000110000000000000110000
00000000000110000000000000110000
00000000000110000000000000110000
00000000111111000000000001111000
00000001111111000000000111111000
00000011111111000000001111111000
00000011001110000000011000110000
00000011001110000000011000110000
00000011111110000000001111110000
00000001111100000000001111100000
00000000000000000000000000000000
```

รูปที่ 3.4 แสดงตัวอย่างภาพขาวดำ

ตัวอย่างที่ใช้ในการทดสอบทั้งหมดได้มาจากบทความ 33 บทความ โดยเลือกตัวอักษรเฉพาะที่มีการติดกันในลักษณะแนวนอน มีตัวอักษรที่ติดกัน 2 ตัวอักษร 381 ตัวอย่าง ตัวอักษร 1 ตัวอักษร 200 ตัวอย่าง และตัวอักษรที่ติดกันมากกว่า 2 ตัวอักษร 109 ตัวอย่าง ดังตัวอย่างด้านล่าง

เภ ะห รบ รท นม ณะ รร ธท าร รา รพ ษท รณ รจ เม **รม นก** ผา ธน รน ะพ **หม ะ ทช าช ทพ ควc** ะห ษว รบ นะ ษน รป กา 00 **ปร ฒน ะจ รช อร ยง งท รช 77 มก ะม คร 84 ชอ ทน เล กง หร** กช ษฐ **ระ นเ** าว ฐบ เผ ษพ ษจ **จร** 54 ยน รถ 46 ขา 37 ฉล 99 63 **ะก ะร** นผ รถ **รอ** าบ **ธบ นก ะป ชน** าง ทว นล ฉะ ธญ **ลม** อง **มน** สพ าห ชจ **ชน** ชา ลง 56 ถง อล นด 58 ลล **ะ** ชว งร 22 **ะอ** งห ลค สห ชต กช เก 28 รง ชน ยง ยท ะช ะบ ะน ะม ทน สภ งห นง กว ษณ จร ยว **ฟร** 99 รช มพ มง ชน ธิป นล งห พร ชา ทร ะป รว ชน มห มว ปร งช ษส รษ ษร ลา รล ตา ฐา ขา

ษท ธา รบ รห ณะ นม เม รบ ษต ฐา รบ รห ธา **ตร นต มข**
าว มร ยา น รา ษฐ รร รว ธร ชา นม รก ะท รร เม รา
เค ณภ รบ รอ รท ะท รห รจ เม เด ษท ะห เม ณะ รา นม
รห รพ เภ ษฐ รบ รณ เภ ธน ะห **เม รท นม ฒน** นม รบ
รา ะห เภ รน ณะ รท ษท รา ษพ นม รา เภ **ธน ณะ ระ** นภ
รน นะ รท ก เม ฒน อง 00 นะ รน รบ ก ก รบ รบ ษท
รท **อฟ** นม ษท 99 รท **เภ รพ** 46 **รบ ยน น** ง เม เภ **ห าน**
ธิป อง รว รน รว รพ รบ ษน ษท รท รา รน ก ษท นภ
รณ ษณ รณ ษฐ ณภ ษฐ **รณ รน รว ฐา นม เภ กฎ รท รป**
ษา อา ยา ณย รม กฎ ณะน ธรก รษท รษฐ **สรจ แล** นมา
งรบ ทรพ รวม ศษ พฒน เมน ขรา พรม ชน รรจ ระบ รัชป
มพน สนค ารช กรง ารว มรบ ประ ราจ ราง ารพ ปรับ นม ะรบ
างพ ธาร านง กรา ณะท พนล เมอ มกบ ารท รณต งรบ ารา ารพ
นอท องช พาร าทค รมเ อรว เมน ตาม ระห ปีน ขรา รทม วงว
ณะม นมรา ประม ตารา บรษท ณะทค ดการ ะนลา การพ
รษทจ นมะน ษณะภ พระอ องรว ละสา ชนสง องสภ กระบ และ
ลงชน ชองก จะมม ชชาว ทาวเ นคนเ พมชน รบบร ะบรษท

ขารวม ราชบร สพรรณ ะระบบ างบวท สมทรส างชน รราชา
ารประ ณะกรร งบรษทท มมากนก งราคาช องนอนน กระทรว
ชวงระย ชนสญญาจ รบนเกาะ อตางทนม นคนยชมชา ารประสานง

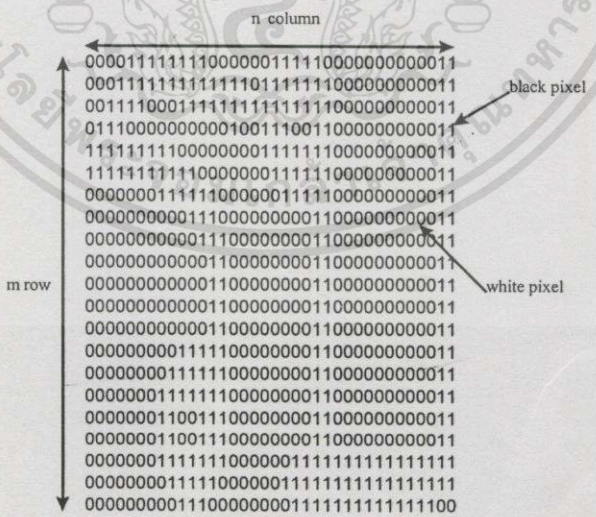
บทที่ 4
การแบ่งตัวอักษรที่ติดกันโดยใช้เส้นแบ่งที่สั้นที่สุด
(Shortest Path)

4.1 เส้นแบ่งที่สั้นที่สุดในแนวดิ่ง (Shortest Path ทางตรง)

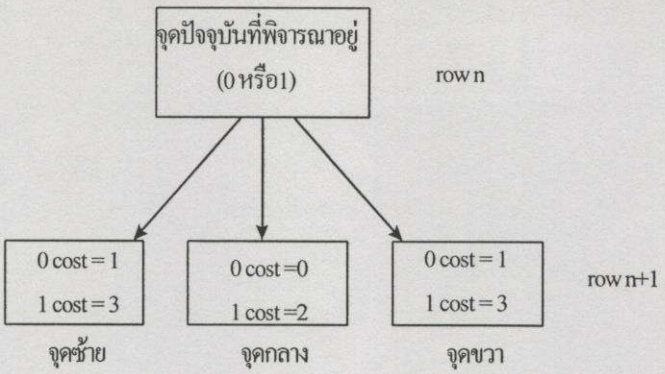
Shortest Path[4] ในกรณีจะเป็นการพิจารณาเลือกทางตรงก่อนเสมอถ้าทั้ง 3 ด้านที่เดินมีค่าในการเดินเท่ากัน ถึงแม้ว่าทางเฉียง 2 ด้านจะเป็นทางที่สั้นที่สุดก็ตาม ดังนั้นจึงเรียก Shortest Path ในกรณีนี้ว่า Shortest Path ทางตรง

Shortest Path ทางตรงได้มีการศึกษามาแล้วในภาษาอังกฤษ ดังที่กล่าวมาแล้วในบทที่ 2 และได้นำมาปรับปรุงใช้ในภาษาไทย โดยเริ่มแรกมีการกำหนดให้ตัวอย่างที่นำมาพิจารณาจะต้องเป็นตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษรทางแนวนอนเท่านั้น คือจะต้องมีการแบ่งออกเป็น 2 ตัวอักษร โดยใช้เส้นแบ่งที่สั้นที่สุด โดยคาดว่าตรงที่ตัวอักษรติดกันจะเป็นจุดที่มีจุดดำน้อยที่สุด ดังนั้นถ้ามีตัวอักษร 1 ตัวไม่ติดกันเข้ามา วิธีนี้จะแบ่งออกเป็น 2 ตัวทันที

การแยกตัวอักษรโดยใช้ Shortest Path ทางตรงจะพิจารณาตัวอักษรที่ติดกันแต่ละตัวอย่างเป็นขนาด n แถว (row) m หลัก (column) ดังแสดงในรูปที่ 4.1 ในการแยกจะทำการกำหนดค่าราคา (cost) ของแต่ละทางเดิน (path) โดยการกำหนดค่า cost แต่ละจุดที่ผ่านเป็นดังรูปที่ 4.2



รูปที่ 4.1 แสดงการกำหนดค่าในการแยกตัวอักษรโดยใช้ Shortest Path ทางตรง



รูปที่ 4.2 แสดงค่า cost ในแต่ละจุดของวิธี Shortest Path ทางตรง

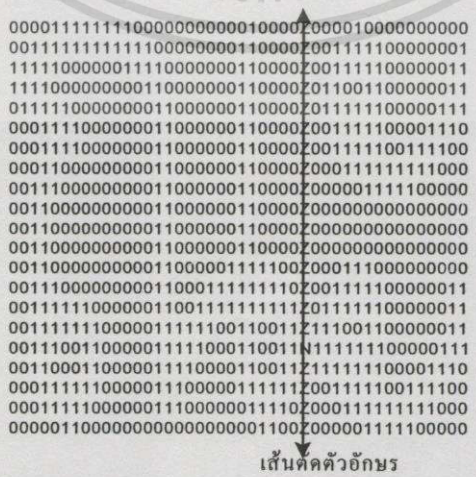
ในการหาเส้นทางเดินที่จะใช้ตัดตัวอักษร 2 ตัวที่ติดกัน จะเริ่มจากจุดที่อยู่ใน row บนสุดไปถึงจุดที่อยู่ใน row ท้ายสุด โดยทางเดินจาก row n ไป row n+1 จะต้องเป็นจุดที่ติดกันซึ่งเป็นไปได้ 3 ทาง ดังแสดงในรูปที่ 4.2 ถ้าเป็นทางดิ่งตรงจะมีค่า cost ที่เลื่อนจาก row n ไป row n+1 ต่ำสุด แต่ถ้าเป็นทางเฉียงจะมีค่าเท่ากันไม่ว่าจะเป็นเฉียงซ้ายหรือขวา โดยค่า cost ของจุดที่เป็นจุดดำจะมีค่า cost มากกว่าจุดขาว เพื่อให้เส้นทางเดินเดินไปตามจุดขาวและตัดเนื้อตัวอักษร(จุดดำ)น้อยที่สุด หลังจากได้เส้นทางเดินแล้วก็จะทำการเปลี่ยนจุดเริ่มต้นไปที่ column ถัดไป และ column ที่ให้ค่า cost น้อยที่สุดจะถูกเลือกเป็นเส้นตัดตัวอักษร (least-cost Path)

4.1.1 ผลการทดลองของวิธี Shortest Path ทางตรง

จากตัวอย่างของกลุ่มตัวอักษรที่ติดกัน 2 ตัวอักษร 381 ตัวอย่างพบว่า

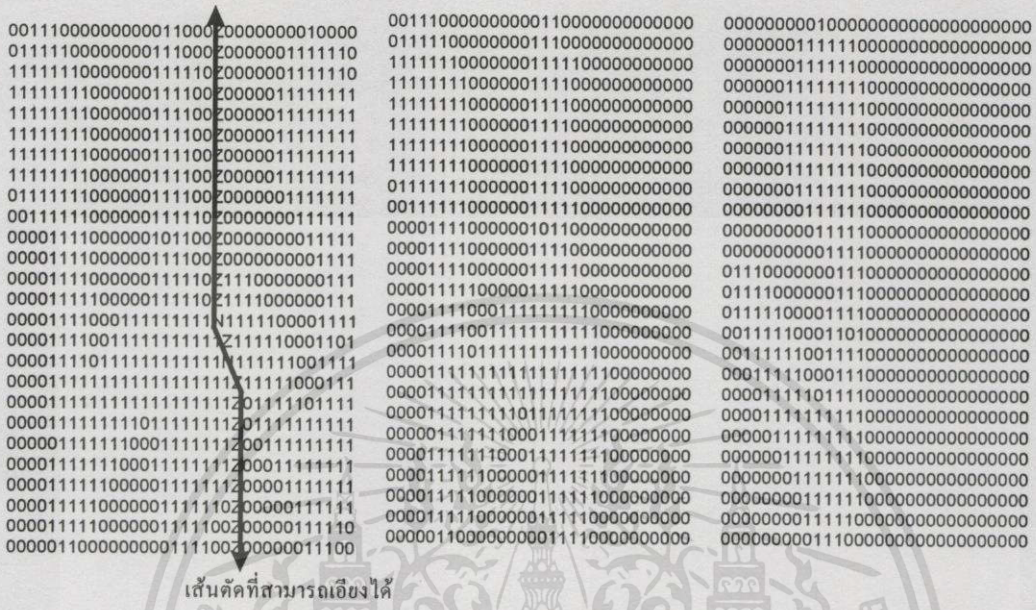
4.1.1.1 ตัวอย่างตัวอักษรที่ตัดถูก

1) ถ้าบริเวณที่ติดกันของตัวอักษรบางกว่าบริเวณอื่น ๆ จะทำให้จุดตัดมีความถูกต้อง ดังรูปที่ 4.3 เส้นตัดจะเป็นเส้นตัดที่ผ่านเนื้อตัวอักษรน้อยที่สุด



รูปที่ 4.3 แสดงตัวอย่างที่วิธี Shortest Path ทางตรง ตัดถูกต้อง

2) เส้นตัดตัวอักษรของวิธี Shortest Path ทางตรงที่ตัดถูกต้องสามารถโค้งตามลักษณะของตัวอักษรได้ทำให้เวลาแบ่งตัวอักษรออกเป็น 2 ส่วนจะได้เนื้อของตัวอักษรที่สมบรูณ์ ดังรูปที่ 4.4

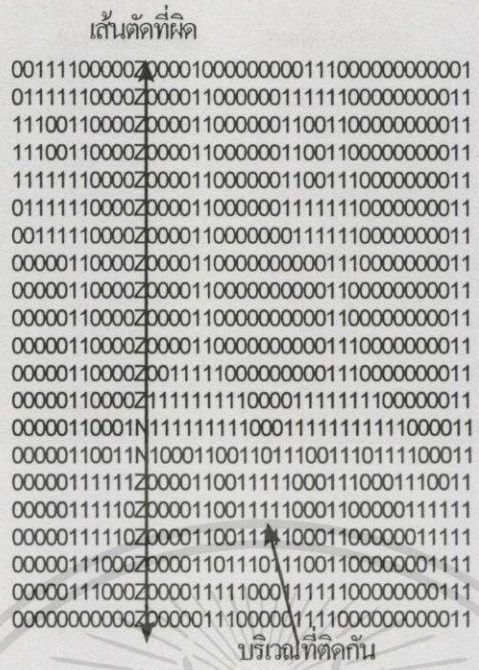


รูปที่ 4.4 แสดงตัวอย่างเส้นตัดที่สามารถโค้งได้ตามลักษณะของตัวอักษร

4.1.1.2 ตัวอย่างตัวอักษรที่ตัดผิด

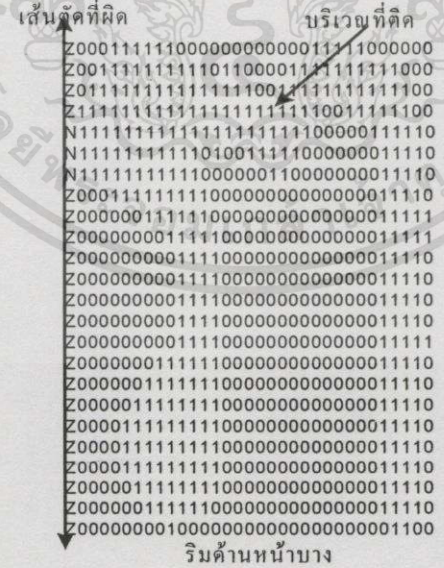
1) ถ้าบริเวณที่ติดกันของตัวอักษรหนากว่าบริเวณอื่น ๆ จะทำให้ค่า cost รวมของบริเวณที่ติดกันของตัวอักษรมากกว่าค่า cost รวมของบริเวณอื่น จึงทำให้ได้เส้นตัดที่ผิด ซึ่งสามารถแบ่งออกได้เป็น 2 กรณีคือ

1.1) ถ้าบริเวณที่ติดกันของตัวอักษรหนากว่าบริเวณกลางตัวอักษรจะทำให้ค่า cost รวมของบริเวณกลางตัวอักษรน้อยกว่าบริเวณที่ติดกันทำให้เลือกเส้นตัดที่ผิด ดังรูปที่ 4.5



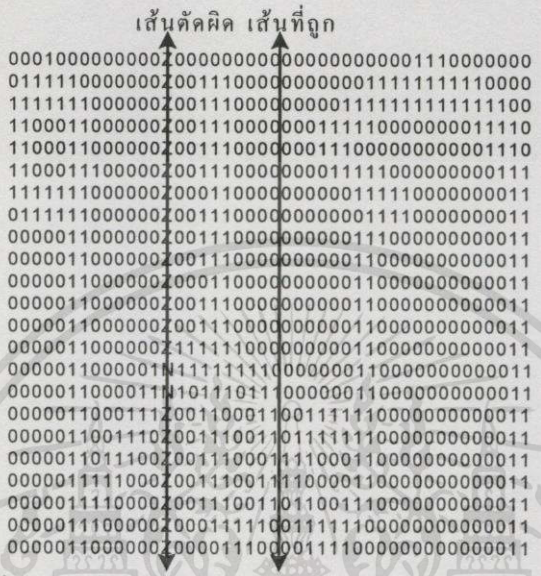
รูปที่ 4.5 แสดงตัวอย่างบริเวณกลางตัวอักษรมีความหนาน้อยกว่าบริเวณที่ติดกันของตัวอักษร

1.2) บริเวณที่ติดกันของตัวอักษรหนากว่าบริเวณริมด้านหน้า หรือริมด้านหลังของตัวอักษร ทำให้ค่า cost รวมของบริเวณริมด้านหน้าหรือริมด้านหลังด้านใดด้านหนึ่งมีค่า cost รวมน้อยกว่าบริเวณที่ติดกันทำให้ได้เส้นตัดที่ผิดดังรูปที่ 4.6



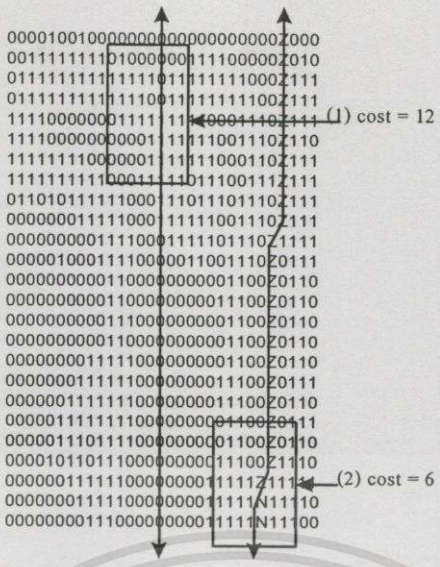
รูปที่ 4.6 แสดงตัวอย่างริมด้านหน้าของตัวอักษรมีความหนาน้อยกว่าบริเวณที่ติดกันของตัวอักษร

2) ค่า cost รวมที่ได้มีค่าเท่ากัน 2 ค่าซึ่งมี column ต่างกันทำให้เวลาพิจารณาเส้นตัดจะเลือกเส้นที่มี column น้อยกว่าเป็นเส้นตัด ซึ่งอาจจะเป็นเส้นตัดที่ถูก หรือผิดก็ได้ ดังรูปที่ 4.7 จากรูปพบว่าเส้นตัดเส้นแรกมี cost รวมเท่ากับ 4 มี column เป็น 13 ส่วนเส้นที่สองมี cost รวมเท่ากับ 4 เช่นกันแต่มี column มากกว่าจึงทำให้เลือกเส้นแรกเป็นเส้นตัดซึ่งเป็นเส้นตัดที่ผิด

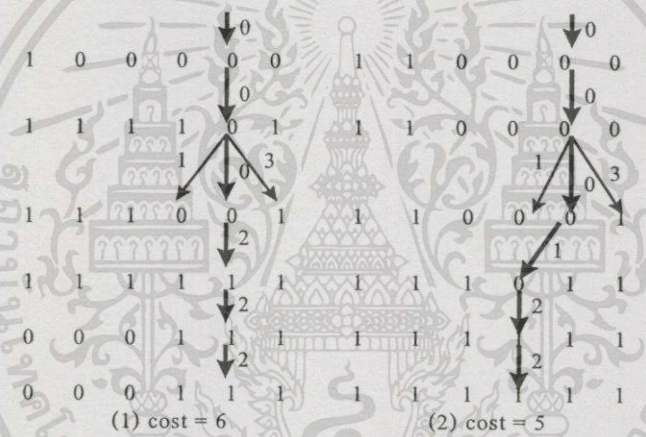


รูปที่ 4.7 แสดงตัวอักษรที่ติดกันมี cost รวมเท่ากัน 2 ค่าแต่มี column ต่างกันทำให้เลือกเส้นตัดผิด

3) จากรูปที่ 4.8 จะสังเกตพบว่าภายในกรอบสี่เหลี่ยม(1)น่าจะเป็นบริเวณที่มีเส้นตัดที่ถูกต้อง เพราะว่าเป็นส่วนที่บาง แต่ส่วนที่บางนี้มีลักษณะเฉียงลงจึงทำให้เวลามองทางตรง หรือคิดตาม Shortest Path ทางตรง ได้ค่า cost มากกว่าบริเวณอื่น ๆ จึงทำให้เลือกเส้นตัดที่ผิด เหตุที่ Shortest Path ทางตรงเลือกเส้นตัดผิดนั้นเนื่องมาจากการคำนวณค่า cost รวมของทั้งสองบริเวณลองมาพิจารณาค่า cost ภายในกรอบสี่เหลี่ยมทั้งสองได้มีการแสดงการคำนวณค่า cost รวมไว้ในรูปที่ 4.9 จากรูปจะพบว่าค่า cost รวมในกรอบสี่เหลี่ยมบนจะมีค่ามากกว่าค่า cost รวมในกรอบสี่เหลี่ยมล่างจึงทำให้เลือกเส้นตัดที่ผิดไป



รูปที่ 4.8 แสดงปัญหาเนื่องจาก Shortest Path ทางตรง

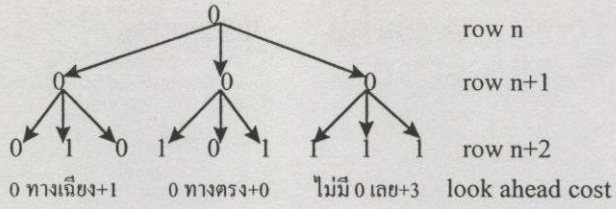


รูปที่ 4.9 แสดงการคำนวณค่า cost ใน Shortest Path ทางตรง

4.2 เส้นแบ่งที่สั้นที่สุดในแนวเฉียง (Shortest Path ทางเฉียง)

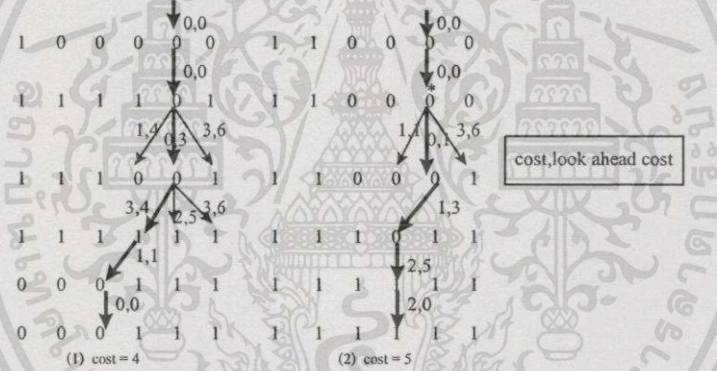
Shortest Path ในกรณีนี้จะพยายามเดินทางเฉียงในกรณีที่ทางเฉียงอาจจะเป็นเส้นทางเดินที่สั้นที่สุด โดยมองไปข้างหน้าอีก 1 ระดับ จึงเรียก Shortest Path ในกรณีนี้ว่า Shortest Path ทางเฉียงและจากข้อผิดพลาดของ Shortest Path ทางตรงในกรณีที่ตัดตรงเสมอ ถึงแม้ว่าทางเฉียงจะเป็นเส้นที่สั้นที่สุดสามารถที่จะแก้ไขได้ด้วย Shortest Path ทางเฉียง

Shortest Path ทางเฉียงมีวิธีคิดค่า cost รวมเหมือนกับในวิธี Shortest Path ทางตรงจะต่างกันตรงที่ใน Shortest Path ทางเฉียงจะมีการพิจารณาค่า look ahead cost เพิ่มขึ้นอีกหนึ่งค่าเพื่อใช้เป็นตัวบอกทิศทางทางเดินของ Path การพิจารณาค่า look ahead cost จะพิจารณาจาก row ที่ถัดไปอีกหนึ่ง row ดังรูปที่ 4.10

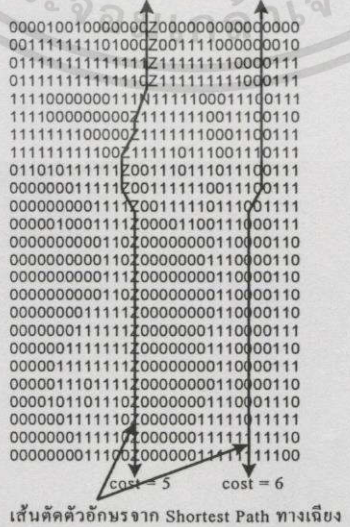


รูปที่ 4.10 แสดงการกำหนดค่า look ahead cost ของ Shortest Path ทางเฉียง

ในการทำ Shortest Path ทางเฉียง การคำนวณค่า cost รวมยังเหมือนเดิมคือดูว่าจุดใน row n+1 เป็น 0 หรือ 1 และเป็นแนวเฉียงหรือแนวตรง(ดังรูปที่4.2) แต่จะมีการคำนวณค่า look ahead cost ซึ่งใช้จุดในระดับ row n+2 มาช่วยในการเลือกว่าจะเลือก path ไหนแต่จะไม่นำมารวมเป็นค่า cost รวมในการเลื่อนจาก row n ไป row n+1 โดยในการเลือกจะเลือกไปทางที่มี cost + look ahead cost น้อยที่สุด ตัวอย่างการคำนวณ look ahead cost ดังรูปที่ 4.11 ผลการตัดด้วยวิธี Shortest path ทางเฉียงแสดงได้ดังรูปที่ 4.12



รูปที่ 4.11 แสดงการกำหนดค่า look ahead cost จากวิธี Shortest Path ทางเฉียง

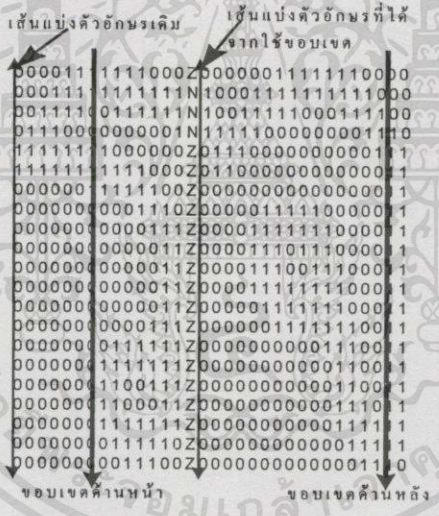


รูปที่ 4.12 แสดงผลการตัดจาก Shortest Path ทางเฉียง

ผลการตัดโดยวิธี Shortest Path ทางเฉียงจะเหมือนกับในวิธี Shortest Path ทางตรงจะต่างกันก็ตรงที่สามารถแก้ปัญหาในกรณีที่ Shortest Path ทางตรงตัดตรงเสมอถึงแม้ว่าทางเฉียงจะเป็นทางที่สั้นที่สุดได้ โดยสามารถเลือกทางเดินที่ถูกต้อง

4.3 การแก้ปัญหาริมด้านข้าง

จากปัญหาที่ริมด้านหน้า หรือริมด้านหลังบางกว่าบริเวณที่ติดจริง และเป็นปัญหาที่พบมากที่สุด ซึ่งสามารถทำการแก้ได้ด้วยการกำหนดขอบเขตด้านหน้าและด้านหลังให้ก่อนที่จะทำการหาเส้นทางเดิน ขอบเขตที่ใช้จะใช้ค่าความกว้างของตัวอักษรซึ่งได้จากการพิจารณาหาตัวอักษรที่มีความกว้างน้อยที่สุดเป็นขอบเขตด้านหน้าซึ่งก็คือสระ เ มีค่าเท่ากับ 7 และตัวอักษรที่มีความกว้างมากที่สุดเป็นขอบเขตด้านหลังคือตัวพยัญชนะ ฌ มีค่าเท่ากับ 30 เมื่อได้ขอบเขตแล้วจึงจะทำการหาเส้นทางเดินภายในขอบเขตที่กำหนด คือมากกว่า 7 น้อยกว่า 30 ผลการหาเส้นตัดตัวอักษรโดยการใช้ขอบเขตแสดงได้ดังรูปที่ 4.13



รูปที่ 4.13 แสดงการใช้ขอบเขตช่วยในการหาเส้นตัดตัวอักษร

4.4 ผลการทดลองจากวิธี Shortest Path

ผลของการแยกตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษรด้วยวิธี Shortest Path สามารถแสดงได้ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงผลการทดลองจากวิธี Shortest Path

วิธี	เปอร์เซ็นต์ความถูกต้อง (%)
Shortest Path ทางตรง	57.70
Shortest Path ทางเฉียง	58.22
Shortest Path ใช้ขอบเขต	76.24

จากตารางที่ 4.1 จะพบว่าวิธี Shortest Path สามารถแยกตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษรได้ถูกต้องเพียง 57.70 % เนื่องจากวิธีนี้จะเลือกเส้นตัดจากเส้นที่สั้นที่สุด หรือมีค่า cost รวมน้อยที่สุดเป็นเส้นตัดตัวอักษร แต่ในความจริงเส้นที่สั้นที่สุดอาจจะไม่ใช่เส้นตัดที่ถูกต้องเสมอไป

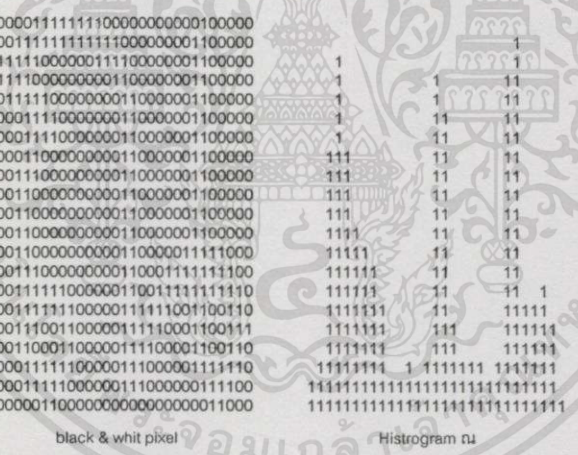
บทที่ 5

Histogram

ในวิธี Shortest Path ที่ผ่านมาในบทที่ 3 มีปัญหาของการตัดตัวอักษรที่สำคัญเกิดขึ้นอย่างหนึ่ง คือการที่เนื้อตัวอักษรที่ติดกันน้อยที่สุดไม่ใช่จุดตัดตัวอักษร ซึ่งการแก้ปัญหานี้จะต้องทำการหาจุดตัดที่ถูกต้องถึงแม้ว่าจุดตัดนั้นอาจจะไม่ใช่บริเวณที่มีเนื้อของตัวอักษรน้อยที่สุด โดยวิธีที่จะใช้คือวิธี Histogram ซึ่งวิธีนี้จะไม่คำนึงถึงว่าบริเวณที่มีเนื้อตัวอักษรติดกันน้อยที่สุดจะต้องเป็นจุดตัดที่ถูกต้องเสมอไป

5.1 การสร้าง Histogram

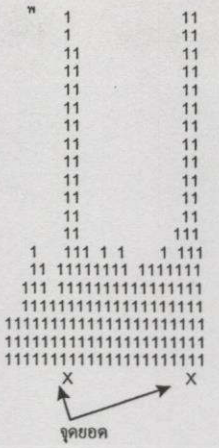
การสร้าง Histogram ของตัวอักษรแต่ละตัวได้จากการนับจำนวนจุดดำในแต่ละคอลัมน์ เช่นในคอลัมน์ที่ 1 มีจำนวนจุดดำเท่ากับ 2 ดังแสดงไว้ในรูปที่ 5.1



รูปที่ 5.1 แสดงตัวอย่าง Histogram ของตัวอักษร ณ

5.2 การกำหนดจุดยอด

ในการพิจารณา Histogram ของตัวอักษรจะมีการกำหนดจุดยอดให้กับตัวอักษรนั้น โดยกำหนดจากเปอร์เซ็นต์ความสูงของตัวอักษรซึ่งในการทดลองนี้ใช้มากกว่าหรือเท่ากับ 60 เปอร์เซ็นต์ของความสูงของตัวอักษรที่อยู่ภายในเส้นบรรทัด และแต่ละจุดที่ถือว่าเป็นจุดยอดจะต้องมีระยะห่างระหว่างจุดยอดที่ผ่านมากับจุดยอดใหม่มากพอควรขึ้นอยู่กับขนาดความสูงของตัวอักษรไม่เช่นนั้นจะถือว่าเป็นจุดยอดเดียวกันดังแสดงไว้ในรูปที่ 5.2 จากรูปแสดงตัวอักษร พ ที่มี 2 จุดยอด



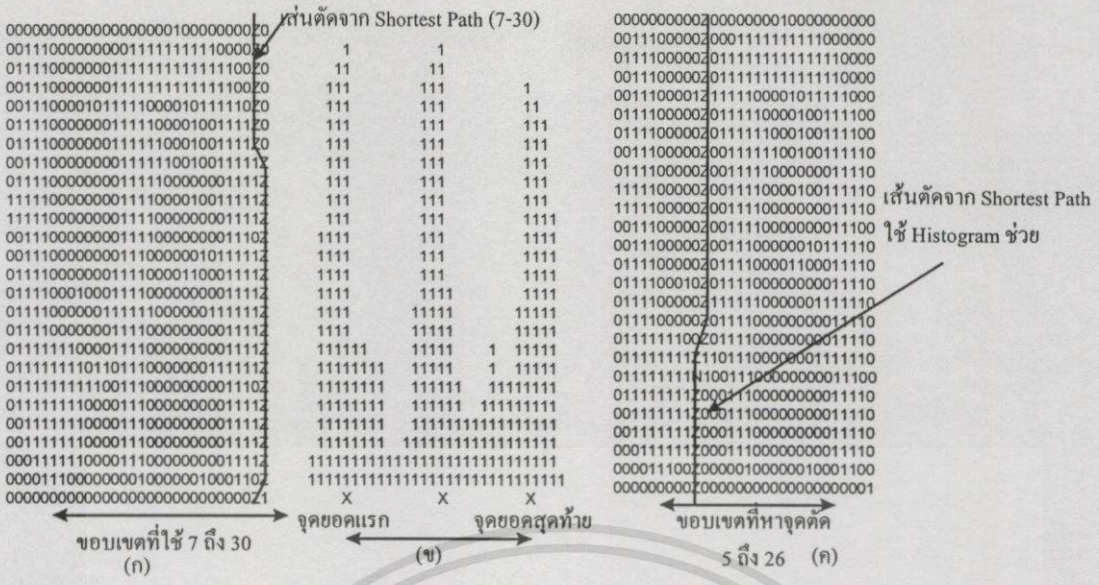
รูปที่ 5.2 แสดงตัวอย่างการกำหนดจุดยอด

5.3 การใช้ Histogram กับ Shortest Path

เนื่องจากการแก้ปัญหาริมด้านข้างของวิธี Shortest Path มีการกำหนดขอบเขตเป็นค่าคงที่ของความกว้างของตัวอักษรที่สั้นที่สุดคือ เ เป็นขอบเขตด้านหน้ามีค่าเท่ากับ 7 และความกว้างของตัวอักษรที่ยาวที่สุดคือ ณ เป็นขอบเขตด้านหลัง มีค่าเท่ากับ 30 ซึ่งการกำหนดนี้จะทำให้มีปัญหาเกิดขึ้น คือ

1) ถ้าความกว้างของตัวอักษรไม่ถึง 30 จะทำให้ขอบเขตที่กำหนดใช้ไม่ได้ผล
2) ถ้าจุดแยกตัวอักษรอยู่ก่อน 7 หรือหลัง 30 ขอบเขตนี้จะทำให้จุดแยกตัวอักษรที่แท้จริงหายไป
3) การกำหนดค่าขอบเขตเป็นค่าคงที่จะใช้ได้กับตัวอักษรเป็นกลุ่ม

จากปัญหาที่เกิดขึ้นนี้จึงทำการกำหนดขอบเขตใหม่โดยใช้ Histogram ของตัวอักษรมาช่วย เนื่องจากปัญหาริมด้านข้างที่เกิดขึ้นนั้นอาจจะอยู่หลังจุดยอดหลัง หรืออยู่ก่อนจุดยอดแรก เพราะฉะนั้นถ้ากำหนดให้ขอบเขตด้านหน้าเป็นตำแหน่งที่จุดยอดแรก และขอบเขตด้านหลังเป็นตำแหน่งของจุดยอดสุดท้ายจะทำให้ปัญหาทั้ง 3 ข้อหมดไป ดังแสดงไว้ในรูป 5.3 จากรูปจะสังเกตเห็น ตัวอักษรที่ติดกัน เก นั้นมีความกว้างเป็น 30 ซึ่งมีค่าพอดีกับขอบเขตที่กำหนดไว้ในตอนแรก (7 ถึง 30) จึงทำให้ขอบเขตที่ใช้ไม่มีผลเส้นตัดจึงยังคงตัดที่ริมด้านข้างเช่นเดิม ดังรูปที่ 5.3(ก) แต่จาก Histogram จุดยอดแรกมีค่าเท่ากับ 5 จุดยอดสุดท้ายมีค่าเท่ากับ 26 ดังรูปที่ 5.3(ข) เมื่อนำมาใช้เป็นขอบเขตจึงสามารถเปลี่ยนเส้นตัดตัวอักษรได้เป็นเส้นตัดถูกต้อง ดังรูปที่ 5.3(ค)



รูปที่ 5.3 แสดงตัวอย่างการใช้ Histogram ช่วยกำหนดขอบเขต

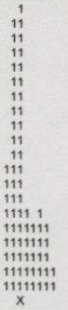
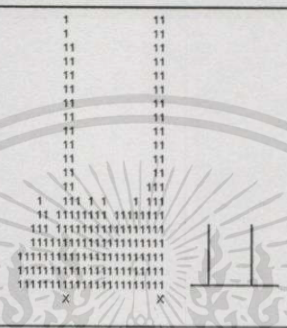
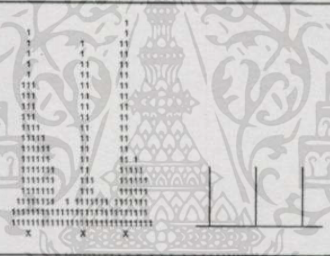
ผลของการใช้ Histogram ช่วยในการกำหนดขอบเขตจะทำให้ประสิทธิภาพในการแก้ปัญหาริมด้านข้างดีขึ้นกว่าการกำหนดไว้ด้วยค่าคงที่ แต่ปัญหาในการหาเส้นตัดที่ผิดในกรณีอื่น ๆ เช่นค่า cost ที่น้อยที่สุดไม่ใช่เส้นตัดที่ถูกต้องนั้น ยังคงแก้ไขด้วยวิธีนี้ไม่ได้

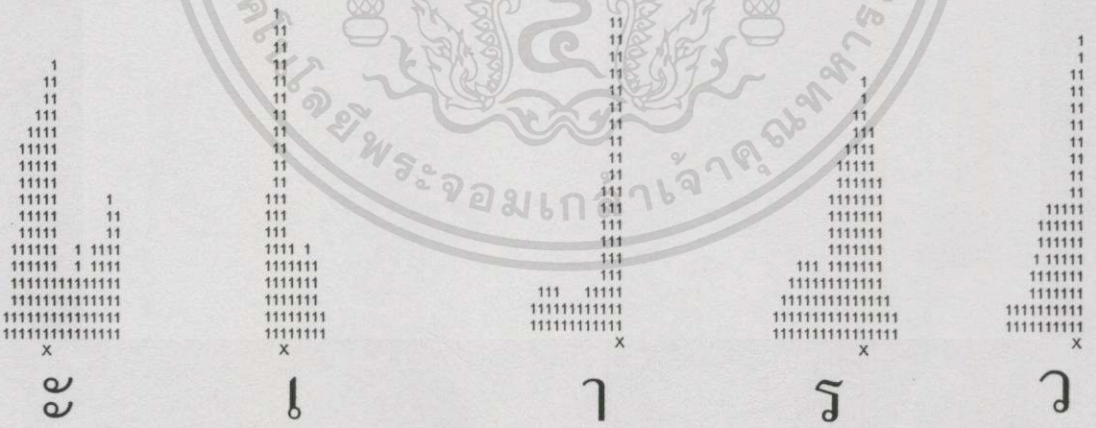
5.4 การแบ่งกลุ่มจาก Histogram

จากลักษณะ Histogram ของตัวอักษรแต่ละตัวพบว่ามีลักษณะเด่นที่สามารถนำมาพิจารณาช่วยแก้ปัญหาที่พบในวิธี Shortest Path ได้จึงทำการศึกษาลักษณะเด่น ๆ ของ Histogram ที่มีอยู่เพื่อนำมาช่วยแก้ปัญหาที่เกิดขึ้น

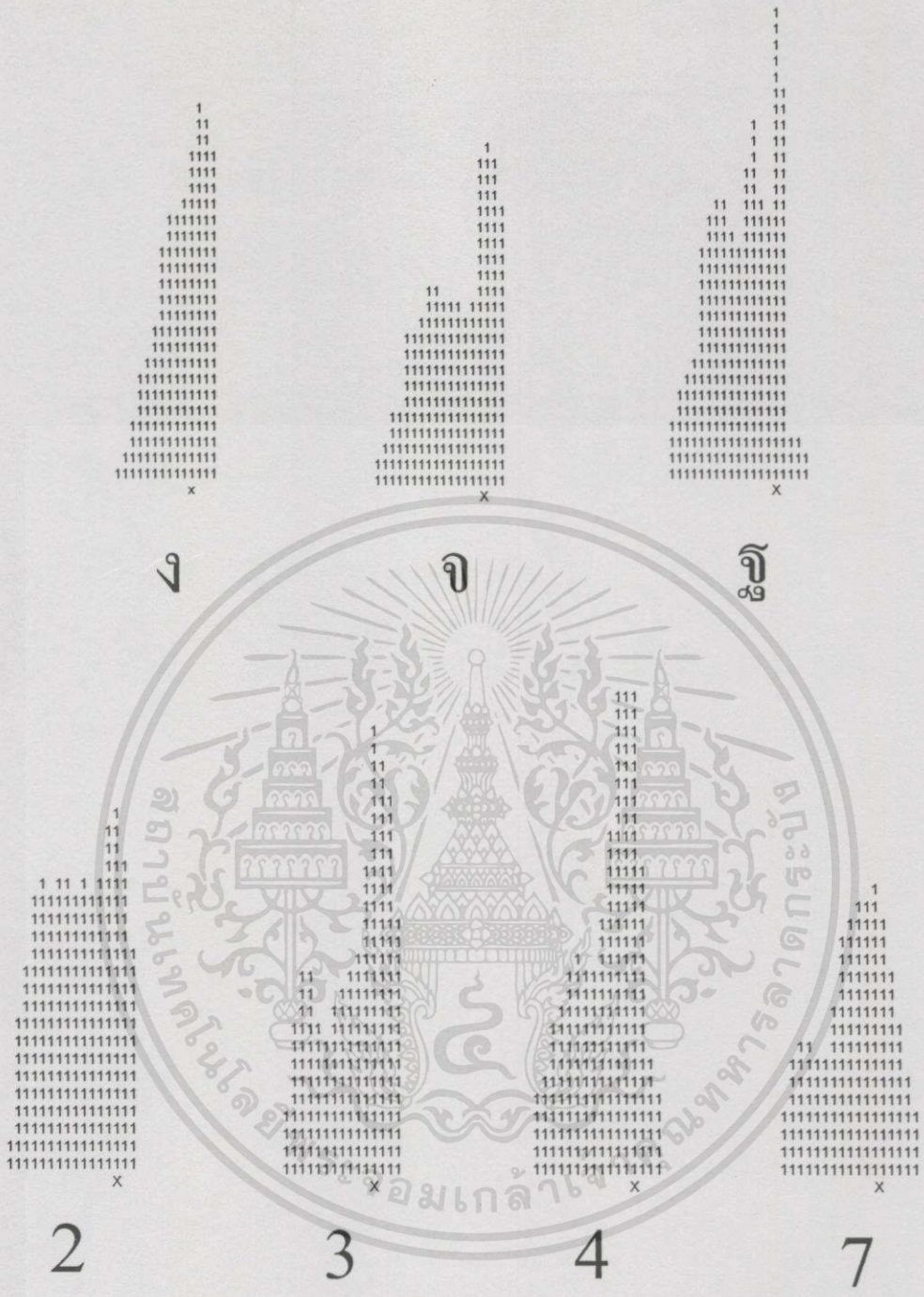
จากการกำหนดจุดยอดที่ได้จาก Histogram ของตัวอักษรแต่ละตัวจะสังเกตเห็นว่าลักษณะของจุดยอดจะเป็นจุดที่มีความสูงเด่นกว่าบริเวณอื่น ๆ ทำให้สามารถใช้ลักษณะเด่นนี้เป็นตัวช่วยในการแบ่งกลุ่มของตัวอักษรทั้งหมดออกเป็นกลุ่ม ๆ ได้ จากตัวอักษรที่ใช้เป็นตัวอย่างทั้งหมดสามารถแบ่งกลุ่มตามจำนวนจุดยอดของตัวอักษรแต่ละตัวได้เป็นกลุ่มใหญ่ ๆ 3 กลุ่มดังตารางที่ 5.1 และลักษณะ Histogram ของตัวอักษรแต่ละกลุ่มทั้งหมดในตารางที่ 5.1 ได้แสดงไว้ในรูปที่ 5.4

ตารางที่ 5.1 แสดงกลุ่มของตัวอักษรตามจำนวนจุดยอดที่ได้จาก Histogram

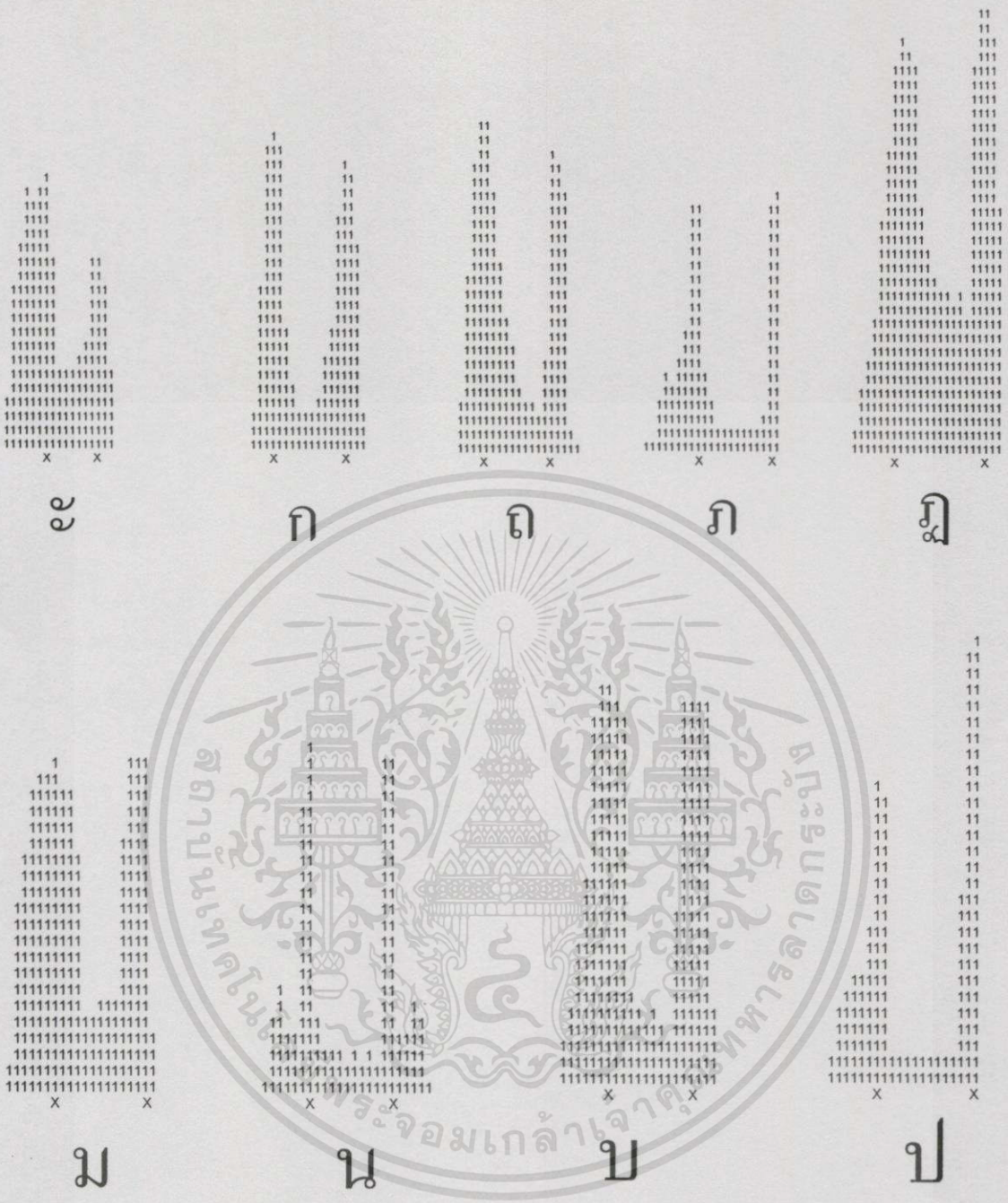
กลุ่ม	จำนวนจุดยอด (จุด)	ลักษณะตัวอย่างในกลุ่ม	ตัวอักษรที่พบ
1	1		ะเาๆจฐรว12347
2	2		ะกขฃคฅฆงจฉชซฌ ฎฏฐฑฒดตถทธนบปผ ฝพฟภมยลศษสหฬอ ฮฤฦ56890
3	3		ฌญฒณพห



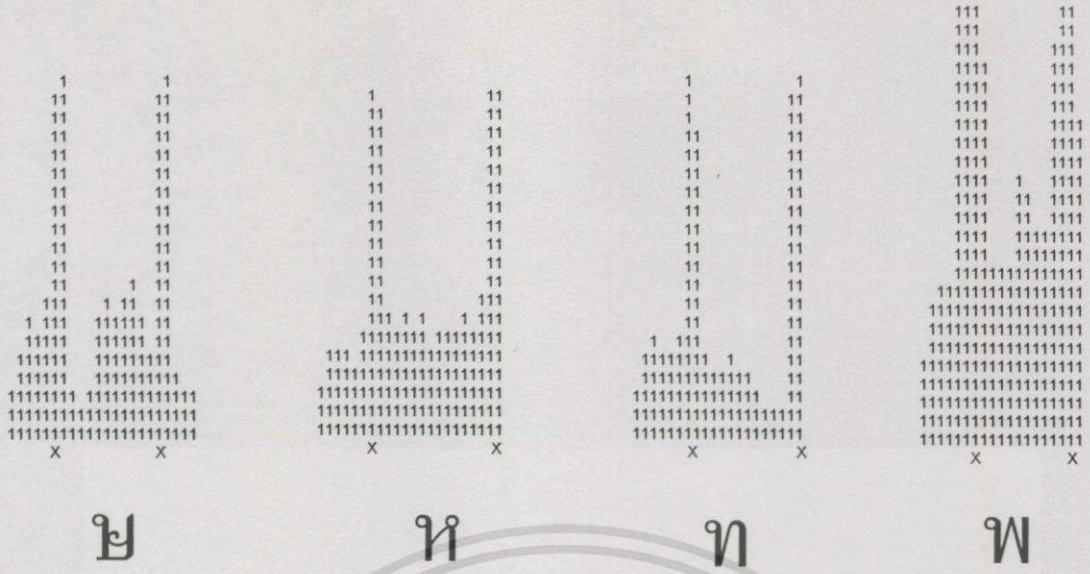
รูปที่ 5.4 แสดง Histogram ตัวอักษร



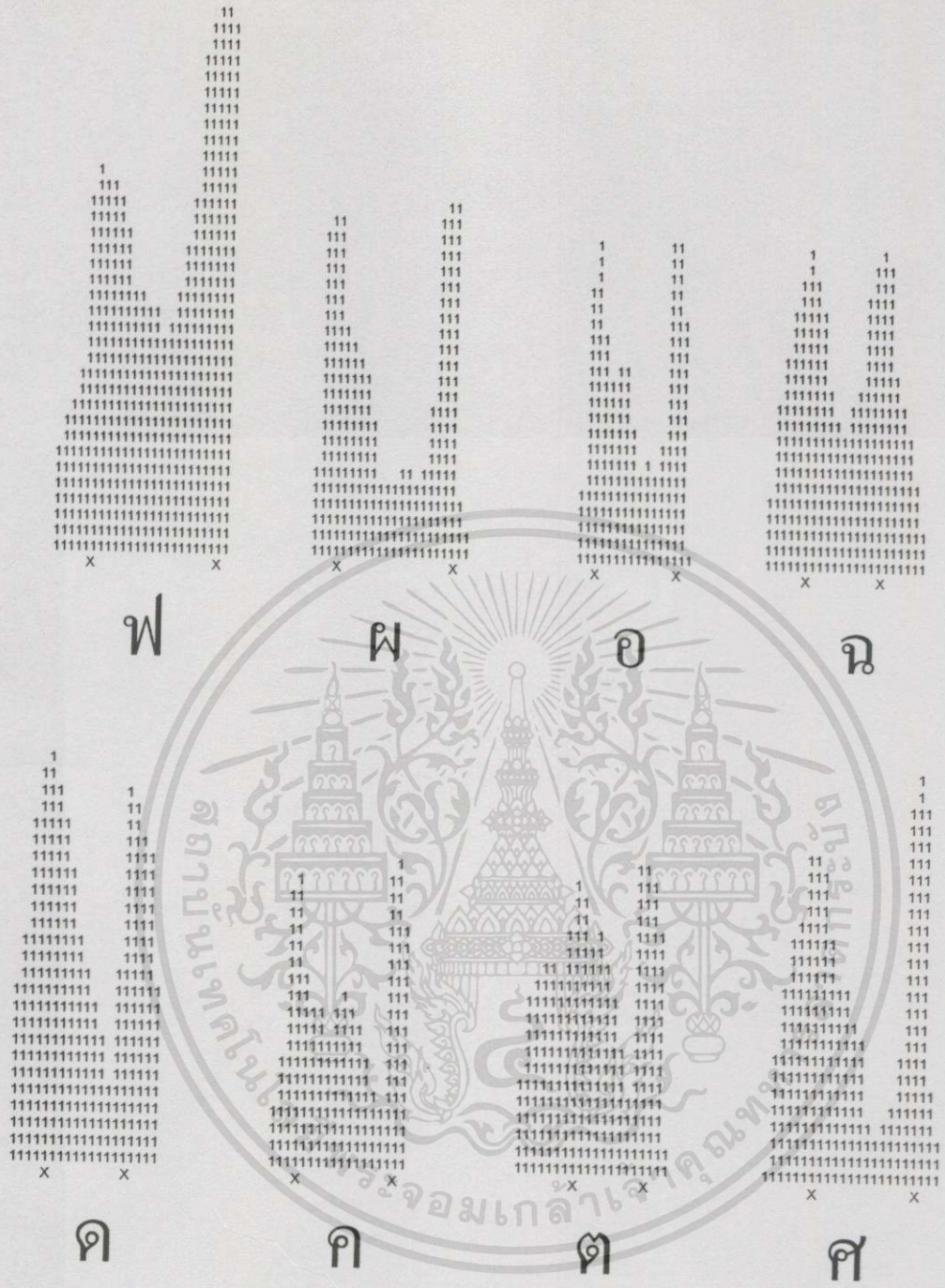
รูปที่ 5.4 (ต่อ)



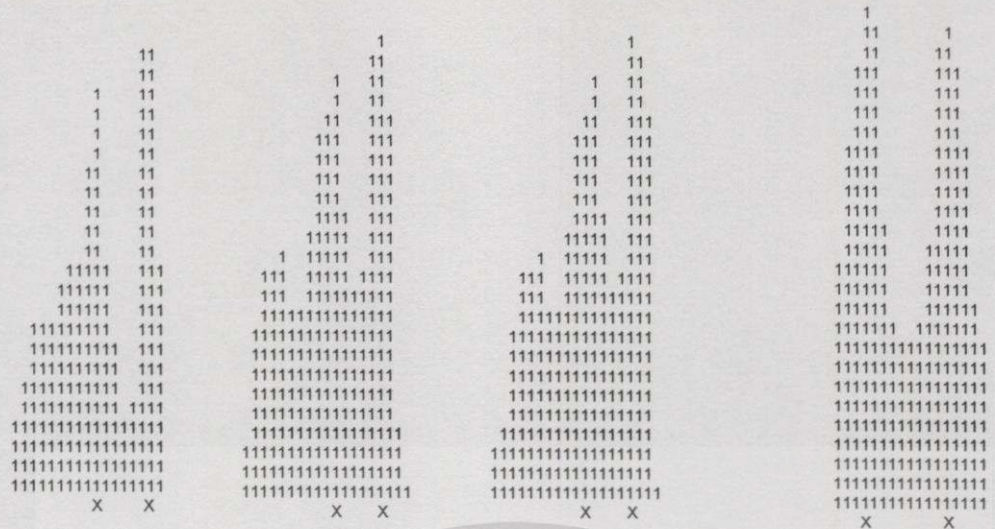
รูปที่ 5.4 (ต่อ)

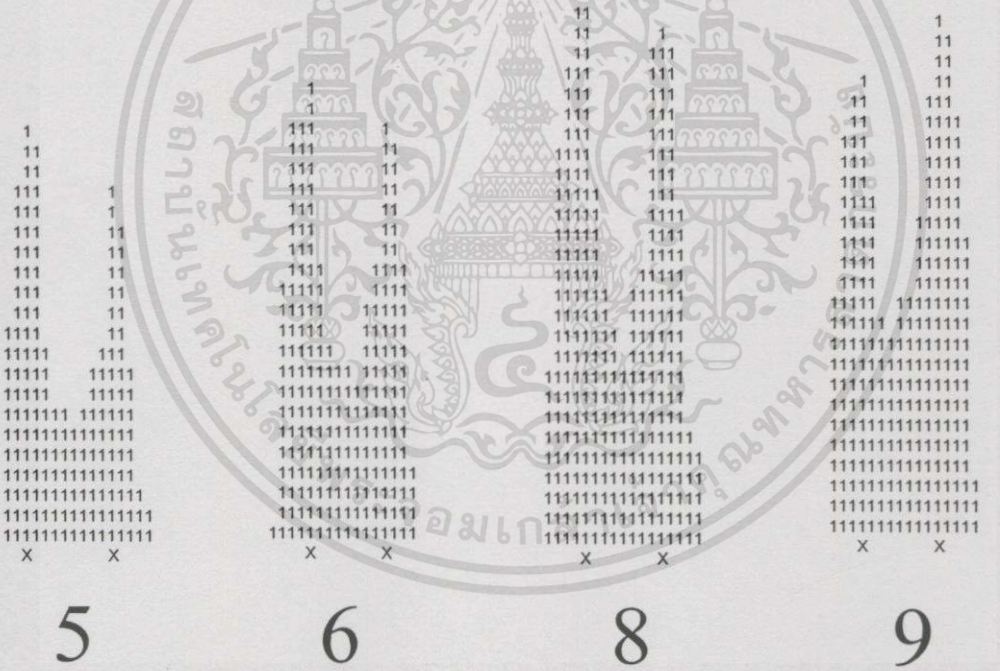


รูปที่ 5.4 (ต่อ)

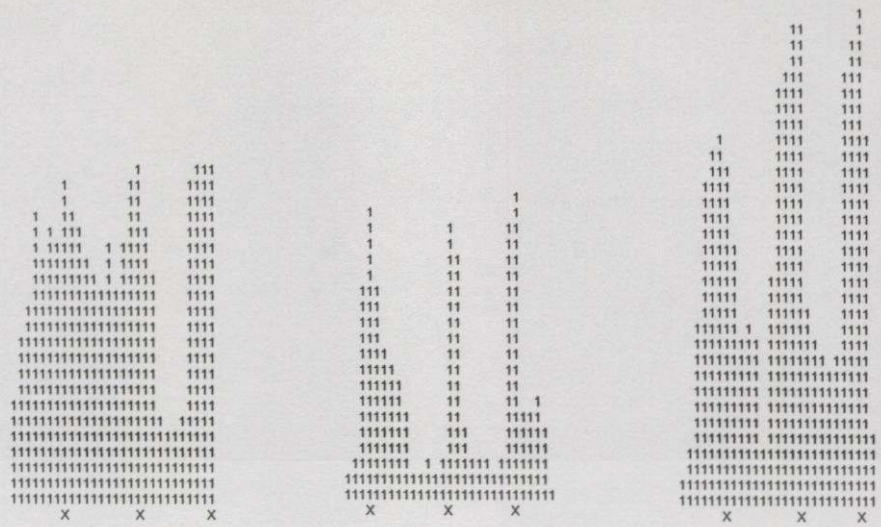


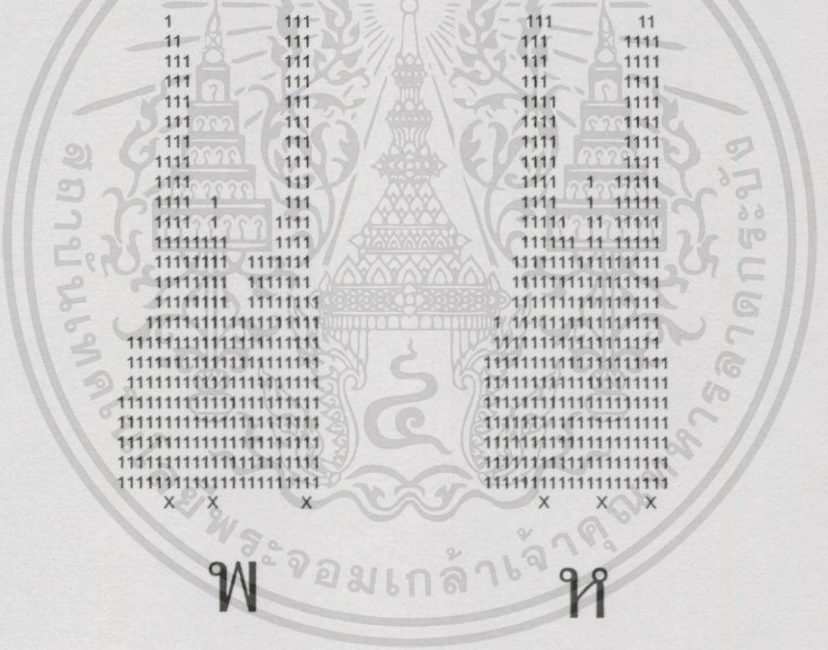
รูปที่ 5.4 (ต่อ)





รูปที่ 5.4 (ต่อ)





รูปที่ 5.4 (ต่อ)

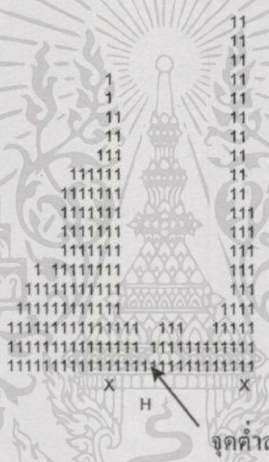
จากตารางที่ 5.1 พบว่าไม่มีเลขไทยในกลุ่มใดเลยเนื่องจากว่าเลขไทยเมื่อนำมาทำ Histogram จะไม่มีจุดยอดที่สูงเด่นเหมือนกับตัวอักษรตัวอื่น ๆ ดังแสดงในรูปที่ 5.5 ทำให้ไม่สามารถจัดเข้ากลุ่มใดได้ ดังนั้นจึงพบว่า Histogram เป็นปัญหาที่ไม่สามารถใช้กับเลขไทย หรือตัวอักษรที่ไม่มีจุดยอดเด่นได้

```
0000001          11
0000010
0011100       1  111
0100110
1101110        111111
0100100
0011011       1111111
```

รูปที่ 5.5 แสดงตัวอย่างเลข ๔ ไทย

5.5 การหาจุดต่ำสุด

จุดต่ำสุดเป็นจุดที่อยู่ระหว่างจุดยอด 2 จุด และเป็นจุดที่มีค่าจุดดำน้อยที่สุด ดังรูปที่ 5.6 จุดต่ำสุดมีความสำคัญในการใช้เป็นจุดสำหรับแบ่งตัวอักษรออกจากกัน เมื่อทราบว่าบริเวณที่จุดแยกตัวอักษรนั้นอยู่ระหว่างจุดยอดใด จุดต่ำสุดระหว่างจุดยอดนั้นจะเป็นจุดแยกตัวอักษร

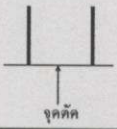
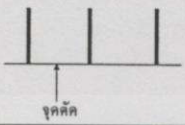
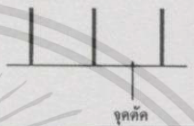
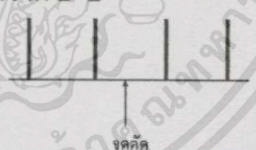
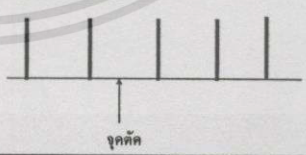
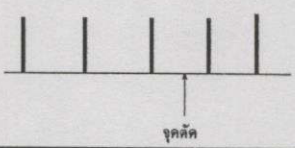
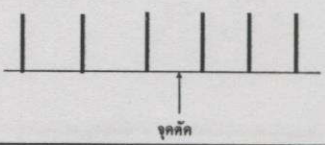


รูปที่ 5.6 แสดงจุดต่ำสุดระหว่าง 2 จุดยอด

5.6 การแยกตัวอักษรจากข้อมูลเข้าที่คาดว่าจะมีการติดกันแน่นอน และมีแค่ 2 ตัวอักษร

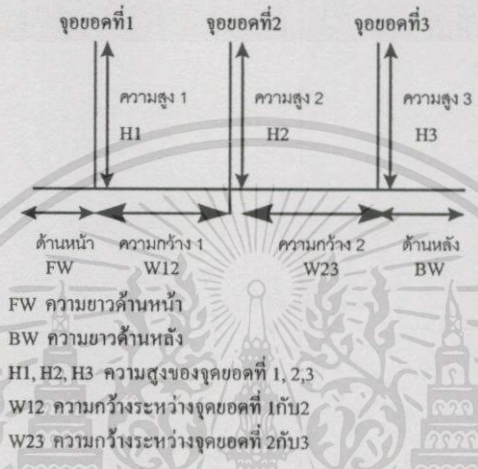
จากการกำหนดเริ่มต้นว่าตัวอักษรที่ใช้เป็นตัวอย่างในการทดสอบนั้นจะต้องติดกันแน่นอนและติดกัน 2 ตัวอักษรเท่านั้นเช่นเดียวกับในวิธี Shortest Path จากกลุ่มของตัวอักษรในตารางที่ 5.1 ถ้านำตัวอักษรมาผสมกันตามหลักภาษาไทยจะทำให้สามารถแบ่งลักษณะการติดกันของตัวอักษรในแนวนอนได้ดังตารางที่ 5.2 และจากลักษณะการติดของตัวอักษรในแนวนอนที่เป็นไปได้ในตารางที่ 5.2 เมื่อทราบจุดยอดรวมจะทำให้สามารถที่จะหาจุดตัดที่เป็นไปได้ได้ เช่น ถ้าจุดยอดรวมเป็น 2 จะมีจุดตัดที่เป็นไปได้อยู่ 1 จุด แสดงว่าจุดแยกตัวอักษรจะต้องเป็นจุดนี้ซึ่งก็คือจุดต่ำสุดที่อยู่ระหว่าง 2 จุดยอด แต่ถ้าจุดยอดรวมเป็นกรณีอื่น ๆ เช่น 3, 4 ซึ่งมีจุดตัดที่เป็นไปได้หลายจุดทำให้ต้องพิจารณาว่าจุดแยกตัวอักษรควรจะเป็นจุดใดที่สามารถแยกตัวอักษรได้ถูกต้อง

ตารางที่ 5.2 แสดงลักษณะการติดที่เป็นไปได้ของ 2 ตัวอักษรในแนวนอน

จำนวนจุดยอดรวม	จำนวนจุดตัดที่เป็นไปได้	บริเวณจุดตัดที่เป็นไปได้	ตัวอย่าง
2	1	2 เกิดจาก 1+1 	รา
3	2	3 เกิดจาก 1+2 	รบ าบ
		3 เกิดจาก 2+1 	ยง กว
4	3	4 เกิดจาก 1+3 	รณ่
		4 เกิดจาก 3+1 	ณะ
		4 เกิดจาก 2+2 	นม
5	2	5 เกิดจาก 2+3 	ธญ
		5 เกิดจาก 3+2 	ณภ
6	1	6 เกิดจาก 3+3 	ไม่พบตัวอย่าง

5.6.1 การแยกตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษร

การพิจารณาว่าจุดตัดนั้นควรจะเป็นจุดใดจะต้องมีการกำหนดเงื่อนไขเพื่อใช้เป็นตัวแยกตัวอักษรแต่ละกลุ่มในตารางที่ 5.1 ออกจากกัน เช่นแยกกลุ่ม 3 จุดยอดออกจากกลุ่ม 2 จุดยอด เป็นต้น โดยอาศัยลักษณะเด่นต่าง ๆ หลาย ๆ ลักษณะเช่น ตำแหน่ง ความสูง ความกว้าง ปริมาณข้อมูล และอื่น ๆ เป็นข้อมูลในการพิจารณาเป็นเงื่อนไขในการแบ่งกลุ่มออกเป็นกลุ่มย่อย ๆ ได้มีการกำหนดตัวแปรที่ใช้เป็นเงื่อนไขในการแบ่งกลุ่มย่อยดังรูปที่ 5.7



รูปที่ 5.7 แสดงตัวแปรในการใช้กำหนดเงื่อนไขในการแบ่งกลุ่มย่อย

5.6.1.1 การแบ่งกลุ่มย่อย

1) กลุ่มที่มี 3 จุดยอด

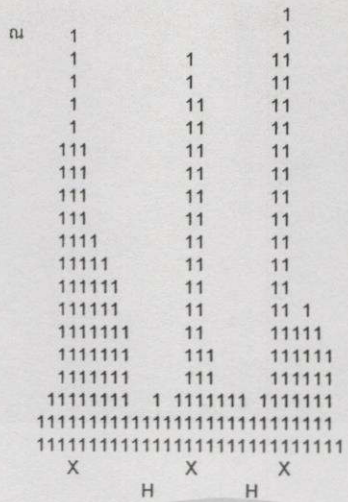
แบ่งได้เป็น 3 กลุ่มดังนี้

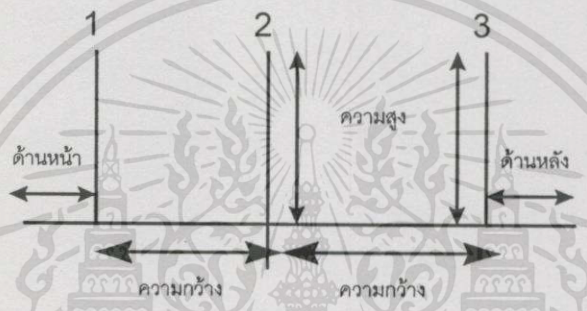
1.1) กลุ่มที่มี 3 จุดยอดแบบที่ 1

ตัวอักษรในกลุ่มนี้ประกอบด้วย ฌ ญ ณ การพิจารณาแยกกลุ่ม 3 จุดยอดในกลุ่มนี้ออกจากตัวอักษรตัวอื่น ๆ จะพิจารณาดังนี้คือ

1) $(0.1*H1 <= FW <= 0.2*H1) \ \& \ (BW < 0.4*H3)$
2) $|H2 - H3| <= 0.25*(H2+H3)$
3) $(0.3*(H1+H2) < W12) \ \& \ (W23 < 0.8*(H1+H2)$
4) $|W12 - W23| <= 0.2*H2$
5) sum1(ปริมาณจุดดำระหว่างจุดยอดที่ 1 กับ 2) < 0.24 * sumall(ปริมาณจุดดำทั้งหมดของภาพตัวอักษร)

ถ้า 3 จุดยอดใดเป็นตามเงื่อนไขนี้ก็จะเป็นกลุ่ม 3 จุดยอดแบบที่ 1 ตัวอย่าง 3 จุดยอดในกลุ่มนี้แสดงได้ดังรูปที่ 5.8





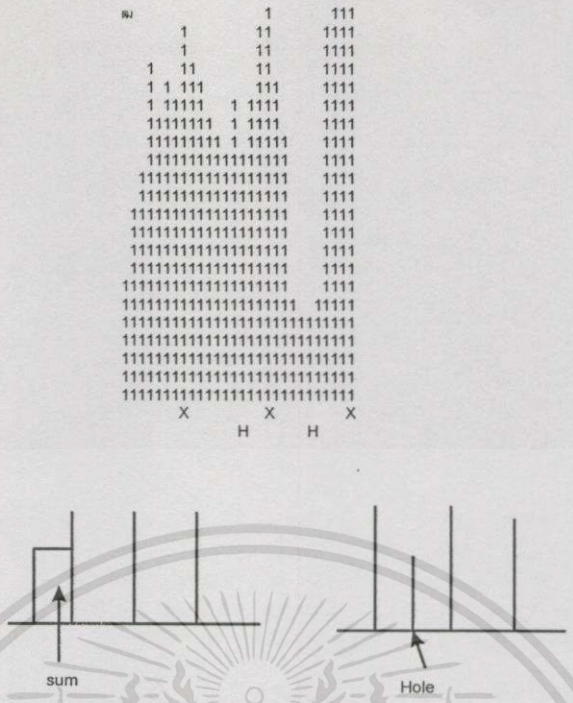
รูปที่ 5.8 แสดงการพิจารณาตัวอักษร 3 จุดยอดในแบบที่ 1

1.2) กลุ่มที่มี 3 จุดยอดแบบที่ 2

ตัวอักษรในกลุ่มนี้คือ ฒ การพิจารณาจะพิจารณาดังนี้

1) sum(ปริมาณจุดดำด้านหน้า) > 0.229*sumall(ปริมาณจุดดำทั้งหมดของภาพตัวอักษร)
2) BW < 0.4*H1
3) Hh1(ความสูงของจุดต่ำสุดจุดแรก) > Hh2
4) 0.6*(H2+H3) > W23 > 0.3*(H2+H3) & 0.2*(H1+H2) < W12 < 0.8*(H1+H2)

ตัวอย่างในกลุ่มนี้แสดงดังรูปที่ 5.9



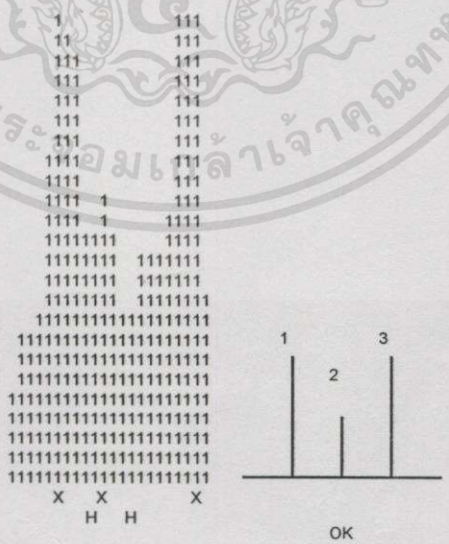
รูปที่ 5.9 แสดงการพิจารณาแยกตัวอักษร ฒ

1.3) กลุ่มที่มี 3 จุดยอดแบบที่ 3

ตัวอักษรในกลุ่มนี้ประกอบด้วยตัว พ ห การพิจารณาจะพิจารณาดังนี้

1) H2 < (H1 & H3) เมื่อ H1 ~ H3

2) FW < 0.4*H1 & BW < 0.4*H3 then group 1.3



รูปที่ 5.10 แสดงการพิจารณาแยกตัวอักษร พ ห

2) กลุ่มที่มี 2 จุดยอด

แบ่งออกเป็น 4 กลุ่มดังนี้

2.1) กลุ่มที่มี 2 จุดยอดแบบที่ 1

ตัวอักษรที่อยู่ในกลุ่มนี้ประกอบด้วย ก ค ฉ ฎ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ล ศ ห อ 5 6 8 9 0 ตัวอย่าง Histogram ของตัวอักษรในกลุ่มนี้แสดงดังรูปที่ 5.11 ตัวอักษรที่รวมอยู่ในกลุ่มนี้แต่ไม่พบตัวอย่าง 10 ตัว คือ ฃ ฅ ฆ ฏ ฑ ฝ ฬ ฮ ฤ ฦ

รูปที่ 5.11 แสดง Histogram ของตัวอักษรในกลุ่มจุดยอด 2 แบบที่ 1

การแยกตัวอักษรในกลุ่มนี้จะมีการพิจารณาดังนี้

1) (0.1*H1 < = FW < = 0.3*H1) & (BW < = 0.4*H2)

2) (0.3*(H1+H2) < W12 < = 0.9*(H1+H2)

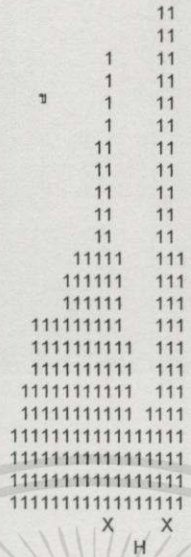
2.2) กลุ่มที่มี 2 จุดยอดแบบที่ 2

กลุ่มนี้จะมีลักษณะเด่นที่ความกว้างระหว่างจุดยอดจะแคบ ตัวอักษรที่อยู่ในกลุ่มนี้ประกอบด้วย ข ฃ ช ซ ธ อ การแยกกลุ่มนี้จะพิจารณาจาก

1) BW < = 0.4*H2

2) W12 < = 0.3*(H1+H2)

ตัวอย่างในกลุ่มนี้แสดงได้ดังรูปที่ 5.12



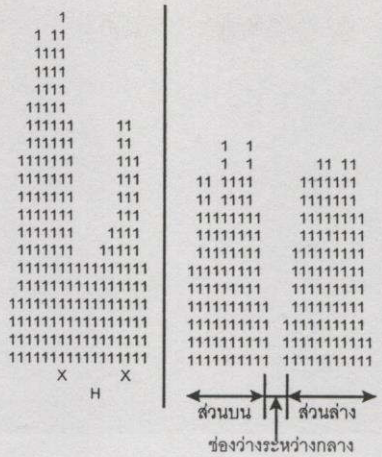
รูปที่ 5.12 แสดง Histogram ของตัวอักษรในกลุ่ม 2 จุดยอดแบบที่ 2

2.3) กลุ่มที่มี 2 จุดยอดแบบที่ 3

ในกลุ่มนี้ประกอบด้วยตัวอักษร สระ ะ ซึ่งการแยกสระ ะ ออกจากตัวอักษรตัวอื่น ๆ ไม่ว่าจะเป็นกรณีของสระ ะ ที่มี 1 หรือ 2 จุดยอดจะใช้วิธีในการแยกเช่นเดียวกันโดยจะพิจารณาจาก Histogram ทางด้านขวาง (Horizontal) ดังนี้

1) พิจารณาช่องว่างตรงกลางที่อยู่ระหว่างส่วนบนและส่วนล่าง
2) พิจารณาปริมาณความกว้างของส่วนบน และความกว้างของส่วนล่าง ซึ่งจะต้องมี ความกว้างใกล้เคียงกัน
3) พิจารณาบริเวณช่องว่างตรงกลางจะต้องไม่กว้างไปกว่าส่วนบนหรือส่วนล่าง และต้องมีปริมาณความกว้างของช่องว่างประมาณ 1 ใน 3 ของส่วนบนหรือส่วนล่าง

จากทั้ง 3 ข้อข้างต้นจะทำให้สามารถแยก ะ ออกจากตัวอักษรตัวอื่น ๆ ได้ดังแสดงในรูปที่ 5.13



รูปที่ 5.13 แสดง Histogram ของสระ ะ 2 จุดยอด

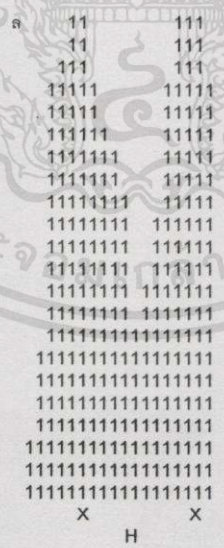
2.4) กลุ่มที่มี 2 จุดยอดแบบที่ 4

ตัวอักษรที่อยู่ในกลุ่มนี้ประกอบด้วย ย อ ล ส 8 5 6 การแยกกลุ่มนี้พิจารณาฮีสโตแกรมของตัวอักษรดังนี้

1) (FW < = 0.2*H1) & (BW < 0.4*H2)

2) sum(ปริมาณจุดดำทางซ้าย) > 0.25 *sumall(จุดดำทั้งหมดระหว่างจุดยอด)

ดังแสดงในรูปที่ 5.14



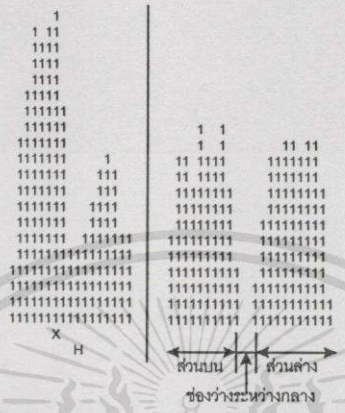
รูปที่ 5.14 แสดง Histogram ของตัวอักษรในกลุ่ม 2 จุดยอดแบบที่ 4

3) กลุ่มที่มี 1 จุดยอด

แบ่งเป็น 3 กลุ่มดังนี้

3.1) กลุ่มที่มี 1 จุดยอดแบบที่ 1

ตัวอักษรในกลุ่มนี้ประกอบด้วยสระ ะ การแยกจะพิจารณาเหมือนกับการพิจารณาสระ ะ ในแบบ 2 จุดยอด สระ ะ 1 จุดยอดแสดงได้ดังรูปที่ 5.15



รูปที่ 5.15 แสดงการแยกสระ ะ ออกจากตัวอักษรตัวอื่น ๆ

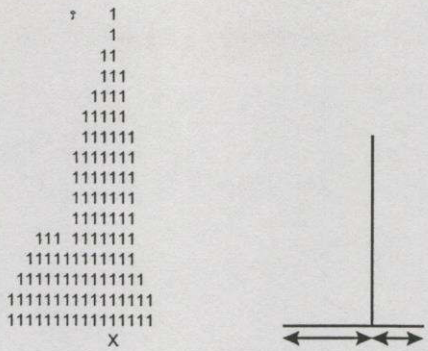
3. 2) กลุ่มที่มี 1 จุดยอดแบบที่ 2

ในกลุ่มนี้จะมีตัวอักษรสระ เ การพิจารณาสระ เ จะพิจารณาจากความยาวของด้านหน้าและด้านหลังของจุดยอดดังนี้ (FW < 0.4*H1) & (BW < 0.4*H1) สระ เ จะมีด้านหน้าที่สั้นมากและมีด้านหลังที่ยาวไม่มากนัก ดังรูปที่ 5.16

รูปที่ 5.16 แสดงการพิจารณาแยกสระ เ ออกจากตัวอักษรอื่น ๆ

3.3) กลุ่มที่มี 1 จุดยอดแบบที่ 3

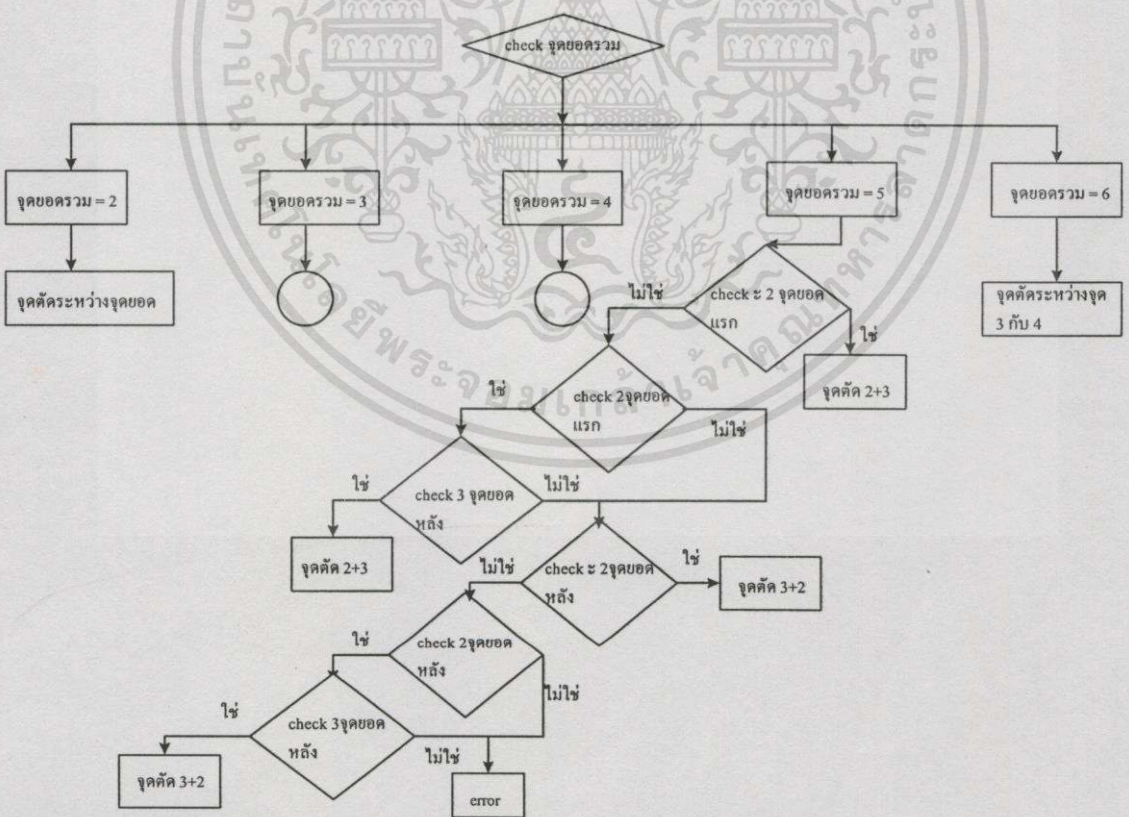
ในกลุ่มนี้จะประกอบไปด้วย า ง จ ฐ ร ว 1 2 3 4 7 การพิจารณาแยกกลุ่มนี้จะพิจารณาจากความยาวของด้านหน้าและด้านหลังของจุดยอดดังนี้ BW <= 0.4*H1 สรุปแล้วในกลุ่มนี้จะมีด้านหน้าของจุดยอดยาว และมีด้านหลังของจุดยอดสั้นดังรูปที่ 5.17



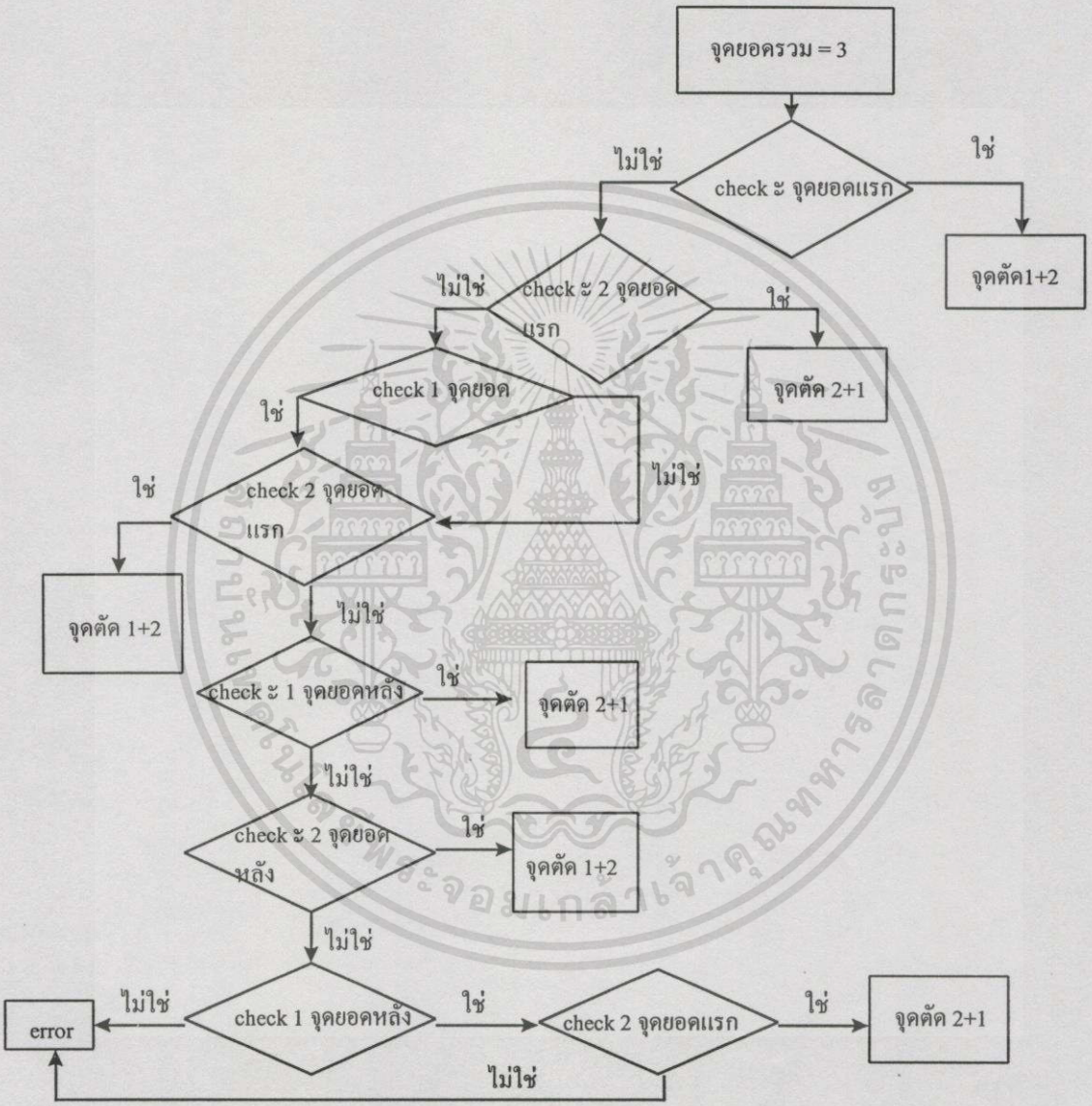
รูปที่ 5.17 แสดงการแยกตัวพยัญชนะ ร ออกจากตัวอักษรอื่น ๆ

5.6.1.2 วิธีการในการใช้แยกตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษร

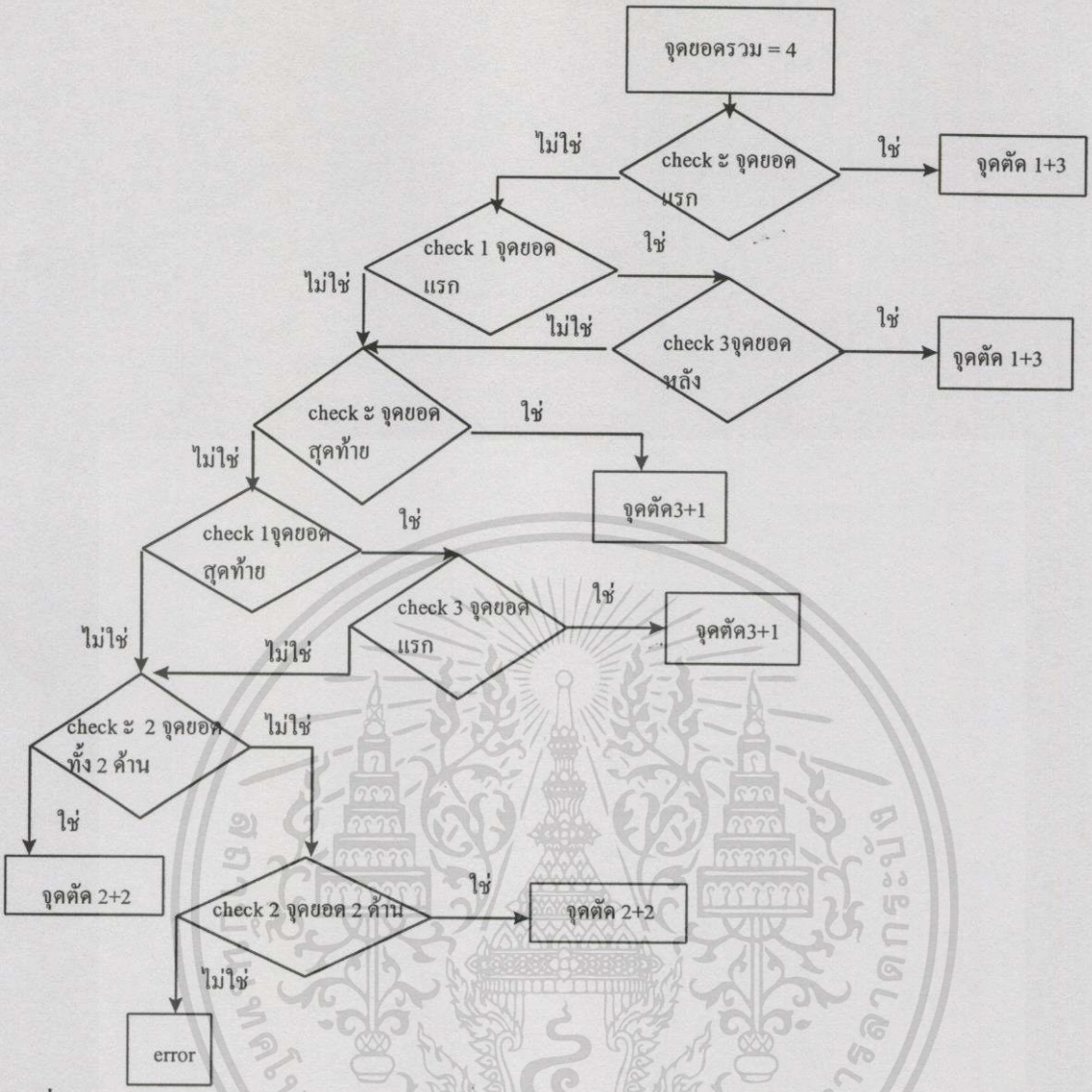
จากกลุ่มย่อยต่าง ๆ ด้านบน เมื่อทราบจำนวนจุดยอดรวมจะทำให้สามารถหาจุดแยกตัวอักษรได้โดยการทดสอบตัวอักษรทั้ง 2 ข้างของจุดตัดที่เป็นไปได้ว่าเป็นไปตามเงื่อนไขที่กำหนดไว้หรือไม่ ถ้าเป็นไปตามเงื่อนไขก็จะถือว่าจุดตัดนั้นเป็นจุดแยกตัวอักษรจริง ถ้าไม่ใช่ก็จะตรวจสอบจุดตัดที่เป็นไปได้ในจุดอื่น ๆ ที่เหลืออยู่ต่อไป การตรวจสอบจุดตัดของตัวอักษรสามารถเขียนเป็น flowchart ในการหาจุดตัดที่เป็นจุดแยกตัวอักษรได้ดังรูปที่ 5.18



รูปที่ 5.18 แสดง Flowchart การตัดตัวอักษร 2 ตัวด้วย Histogram



รูปที่ 5.18 (ต่อ)



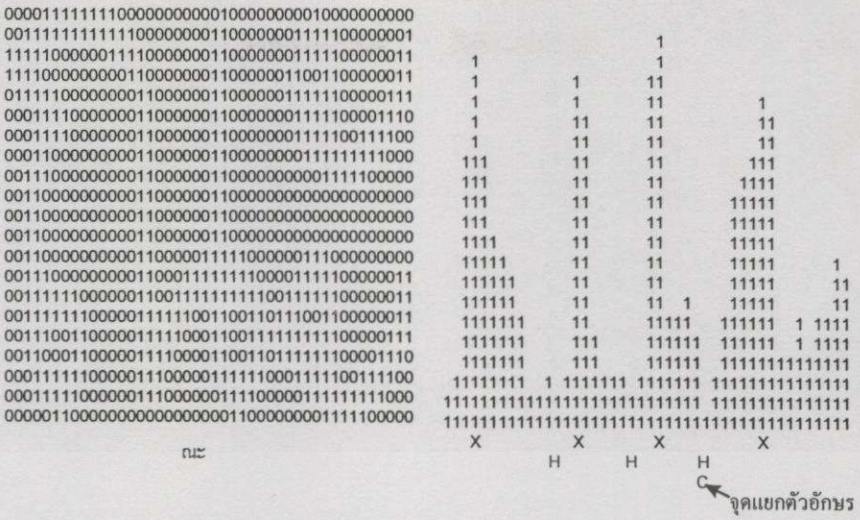
รูปที่ 5.18 (ต่อ)

5.6.2 ผลการแยกตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษร

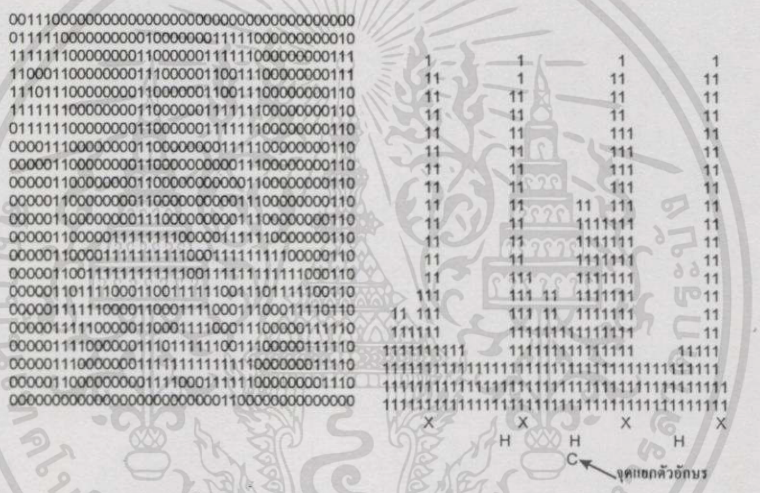
จากตัวอย่างของตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษรที่นำมาพิจารณาจำนวน 381 ตัวอย่าง สามารถแสดงผลการตัดตัวอักษรได้ดังนี้

5.6.2.1 ตัวอย่างตัวอักษรที่ตัดถูก

การแยกตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษรจำนวน 381 ตัวอย่างสามารถตัดได้ถูกต้อง 330 ตัวอย่างคิดเป็น 91.08 % ดังแสดงตัวอย่างไว้ในรูปที่ 5.19 และรูปที่ 5.20



รูปที่ 5.19 แสดงผลการแยกตัวอักษรที่ติดกัน 2 ตัวอักษร

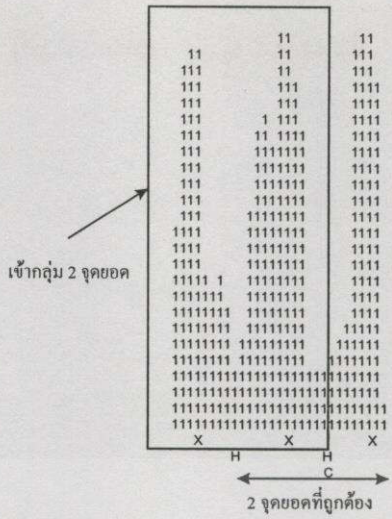


รูปที่ 5.20 แสดงผลการแยกตัวอักษรที่ติดกัน 2 ตัวอักษร

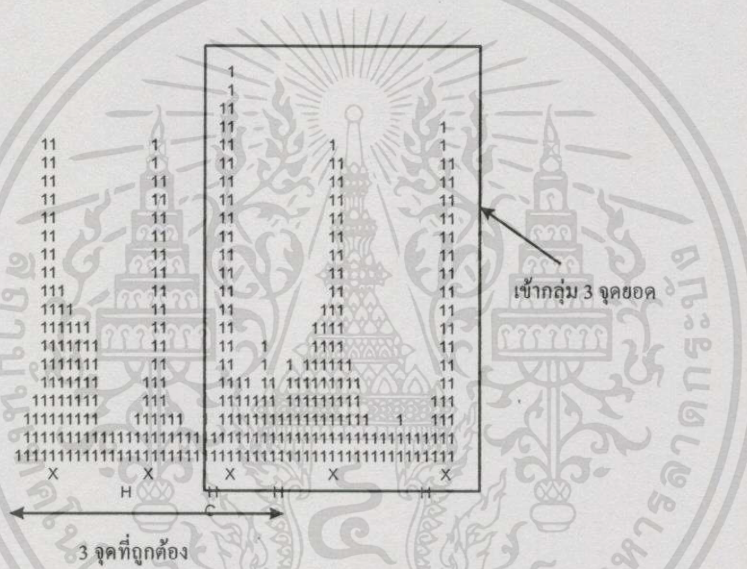
5.6.2.2 ตัวอย่างตัวอักษรที่ตัดผิด

การแยกตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษรจำนวน 381 ตัวอย่างตัดผิด 34 ตัวอย่างคิดเป็น 8.92 % สาเหตุเนื่องมาจาก

1) ไม่สามารถแยกออกจากกลุ่มได้ เช่น เม ตามปกติควรจะเป็นกลุ่ม 1 จุดยอดรวมกับกลุ่ม 2 จุดยอด (1+2) แต่ผลกับเป็นสระ เ รวมกับ ม ครึ่งหนึ่ง เข้ากลุ่ม 2 จุดยอดแทนแล้วอีกครึ่งหนึ่งของ ม กลายเป็น 1 จุดยอดไป ดังแสดงในรูปที่ 5.21 อีกตัวอย่างหนึ่งก็เช่นกันคือ ณภ จุดยอดสุดท้ายของ ณ รวมกับ ภ กลายเป็นกลุ่ม 3 จุดยอด จึงทำให้ตัดผิดดังแสดงในรูปที่ 5.22 จากผลทั้งหมดสามารถรวบรวมตัวอักษรที่รวมกันแล้วเข้ากลุ่มที่มีอยู่ได้ดังตารางที่ 5.3



รูปที่ 5.21 แสดงตัวอย่างตัวอักษรที่ตัดผิด เนื่องจากแยกออกจากกลุ่มไม่ได้

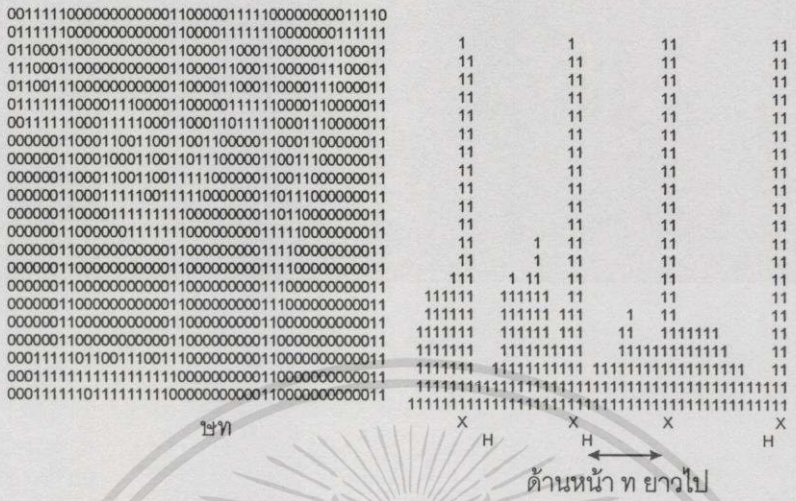


รูปที่ 5.22 แสดงตัวอย่างตัวอักษรที่ตัดผิด เนื่องจากแยกออกจากกลุ่มไม่ได้

ตารางที่ 5.3 แสดงตัวอักษรทั้งหมดที่แยกกลุ่มไม่ได้

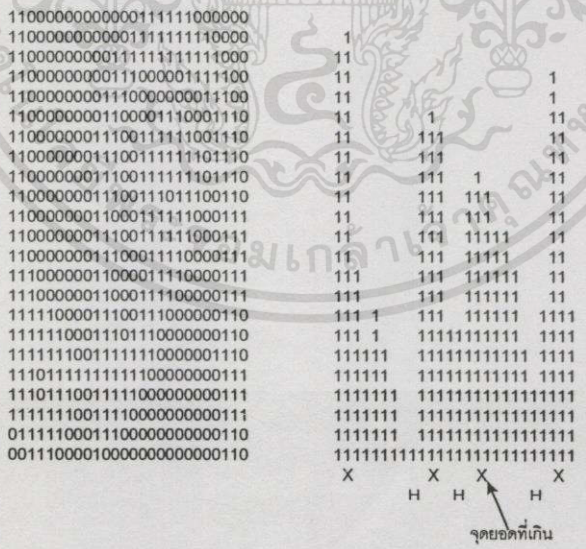
กลุ่มที่	ลักษณะที่ตัดผิด	ตัวอักษรที่พบ
1	1 จุดยอดรวมกับครึ่งหนึ่งของ 2 จุดยอด	ร**อ**, ร**ท**, ร**ห**, เ**ส**, เ**ม**, เ**ภ** ร**พ**, ก**า**, **ท**น, ะ**ป**
2	2 จุดยอด รวมกับ 1จุดยอดของ 3 จุดยอด	**ษณ**, ณภ

2) มีด้านหน้า หรือ ด้านหลัง ยาวกว่าค่าที่กำหนดเช่น ษท ตัว ท มีด้านหน้าที่ยาวกว่าค่าที่กำหนดไว้ ดังแสดงในรูปที่ 5.23

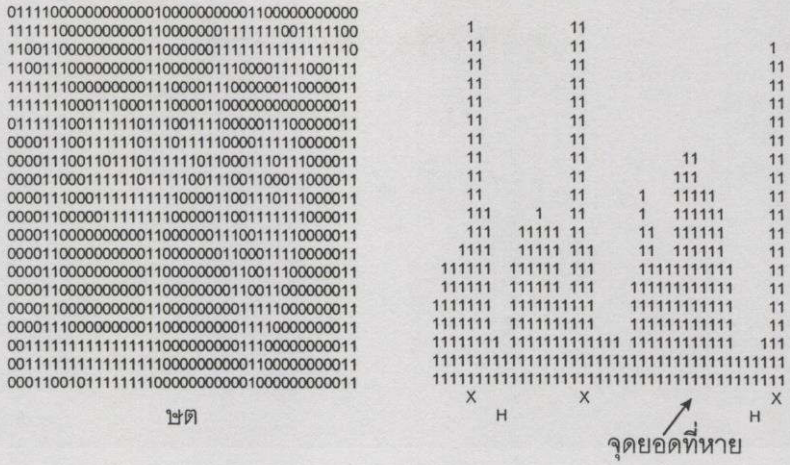


รูปที่ 5.23 แสดงตัวอย่างการตัดผิดเนื่องจากด้านหน้ายาวเกิน

3) มีจุดยอดเกิน หรือขาด จากที่กำหนด เช่น ด ตัว ด มีจุดยอดเกินเป็น 3 แทนที่จะเป็น 2 ดังแสดงในรูปที่ 5.24 และ ษต ตัว ต จุดยอดขาดไป 1 จุดจาก 2 จุดยอดเหลือ 1 จุดยอด ดังแสดงในรูปที่ 5.25

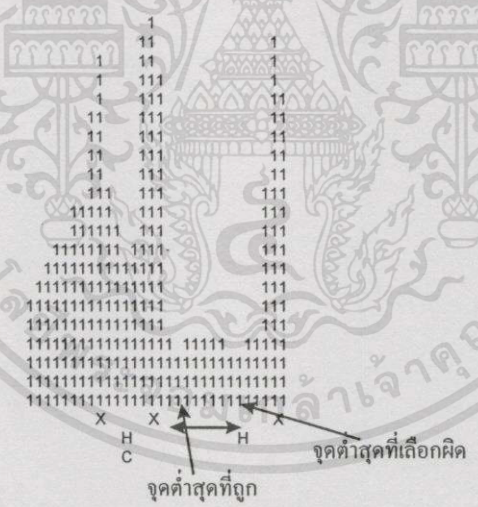


รูปที่ 5.24 แสดงตัวอย่างการตัดผิดเนื่องจากจุดยอดเกิน



รูปที่ 5.25 แสดงตัวอย่างการตัดผิดเนื่องจากจุดยอดขาด

4) การพิจารณาจุดต่ำสุดระหว่างจุดยอด หรือ Hole มีค่าเท่ากันหลายค่าในระหว่างจุดยอดนั้นการเลือกค่าบางค่าอาจจะทำให้มีผลต่อความยาวด้านหน้า หรือด้านหลังได้ ดังรูปที่ 5.26 ตัวอักษรที่ติดกัน **ซา** มีจุดต่ำสุดระหว่างจุดยอดที่ 2 กับ 3 หลายค่า ค่าที่เลือกทำให้ความยาวด้านหลัง ช ยาวเกินจากค่าที่กำหนดไว้



รูปที่ 5.26 แสดงผลการตัดผิดเนื่องจากการเลือกจุดต่ำสุด

5.7 การแยกตัวอักษรจากข้อมูลเข้าที่ไม่ทราบจำนวนตัวอักษรที่ติด

จากการแยกตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษร และลักษณะเด่นของ Histogram ของตัวอักษรที่ผ่านมาข้างต้น ทำให้นำมาลองใช้แยกตัวอักษรที่ไม่ทราบจำนวนตัวติดของตัวอักษร คือถ้ามีตัวอักษรเข้ามา 1 ตัวอักษรจะไม่แบ่งครึ่งตัวอักษรนั้นเหมือนกับวิธี Shortest Path หรือวิธี Histogram ในแบบแรก ในวิธีนี้จะสามารถแยกตัวอักษรที่เป็นตัวอักษร 1 ตัว หรือตัวอักษรที่ติดกัน 2 ตัว หรือตัวอักษรที่ติดกันมากกว่า 2 ตัวได้ ดังนั้นจึงมีการพิจารณาเงื่อนไขใหม่เพื่อใช้แยกตัวอักษรที่ไม่ทราบจำนวนตัวติดของตัวอักษร โดยมีการเปลี่ยนแปลง และเพิ่มเติมเงื่อนไขในการแยกดังนี้

5.7.1 การหา Distance

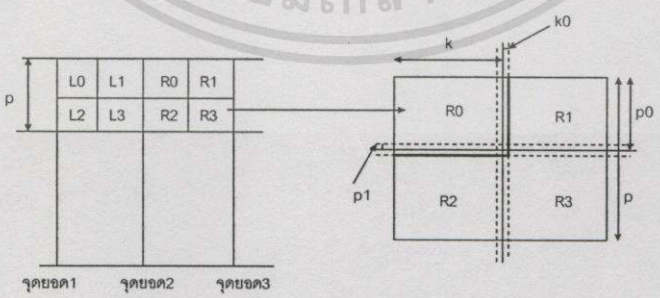
distance เป็นค่าที่เพิ่มขึ้นเพื่อช่วยในแยกกลุ่มที่ตัดผิดเนื่องจากไม่สามารถแยกกลุ่มได้ เช่น เม าน ณภ เป็นต้น การหาค่า distance ทำได้ดังนี้

การหาค่า distance Lและ R จะใช้ตัวอักษรที่อยู่ในรูปของ bitmap image นำเฉพาะส่วนบนมาแบ่งออกเป็น2 ข้าง(L,R) ในแต่ละข้างแบ่งออกเป็น 4 quandent โดยแต่ละ quandent มี overlap เท่ากับ 0.125*w12 จากนั้นก็หาค่า distance L และ R จากค่าเฉลี่ยของแต่ละกลุ่มซึ่งมีขั้นตอนการหาค่าดังนี้

1) นำตัวอักษรทั้งหมดมาจัดกลุ่มโดยตัวอักษรมีลักษณะรูปแบบที่คล้ายกัน เช่น ก กับ ถ อยู่กลุ่มเดียวกัน น กับ ม อยู่กลุ่มเดียวกัน หรือ ณ กับ ญ อยู่กลุ่มเดียวกัน
2) จากกลุ่มตัวอักษรและตัวอักษรที่เลือกมา หาค่าเฉลี่ยจำนวนจุดดำเทียบจำนวนจุดดำในแต่ละช่อง(L0,L1,L2,L3 R0,R1,R2,R3) โดยมีการแบ่ง quandent ตามรูปที่ 6
3) หาค่า distance L และ distance R จากสมการ (1) และ (2)

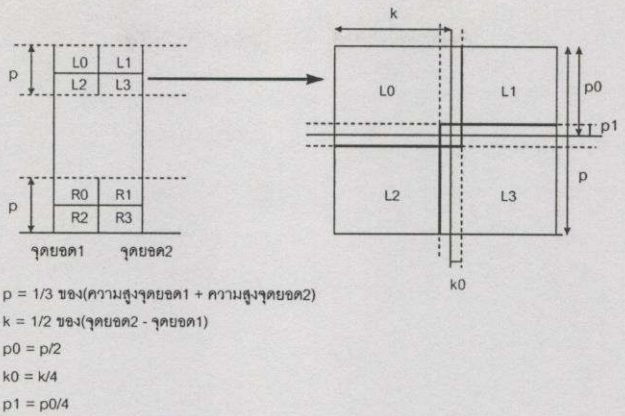
$$\text{distance L} = (L0-La0)^2 + (L1-La1)^2 + (L2-La2)^2 + (L3-La3)^2 \quad (1)$$

$$\text{distance R} = (R0-Ra0)^2 + (R1-Ra1)^2 + (R2-Ra2)^2 + (R3-Ra3)^2 (2)$$



p = 1/3 ของ(ความสูงจุดยอด1+ความสูงจุดยอด2)
p0 = p/2
p1 = p0/4
k = 1/2 ของ(จุดยอด2-จุดยอด1)
k0 = k/4

รูปที่ 5.27 แสดงการกำหนดพื้นที่ในการแบ่งกลุ่มย่อยของ 3 จุดยอด



รูปที่ 5.28 แสดงการกำหนดพื้นที่ในการแบ่งกลุ่มย่อยของ 2 จุดยอด

การหาค่า distance ในกลุ่ม 3 จุดยอดจะแบ่งได้เป็นกลุ่มค่า distance 7 กลุ่ม ซึ่งแต่ละกลุ่มจะมีค่า distance ที่ต่าง ๆ กันแต่มีตัวอักษรในกลุ่มเหมือนกันคือ ฉ ญ ฒ ส่วนในกลุ่ม 2 จุดยอดจะแบ่งออกเป็น 29 กลุ่มที่มีค่า distance ต่างกัน ในแต่ละกลุ่มมีตัวอักษรดังตารางที่ 5.4

ตารางที่ 5.4 แสดงกลุ่มต่าง ๆ ทั้ง 29 กลุ่มที่มีค่า distance ต่างกัน

กลุ่มที่	ตัวอักษรที่อยู่ในกลุ่ม	จำนวนกลุ่มย่อย
1	ก ฎ ถ ภ ค ต ด	3
2	น ม	4
3	บ ษ ป	3
4	ท ห	5
5	ล ส	3
6	ศ	1
7	อ 0 9 6 5 2	1
8	ฉ	1
9	พ ฟ	4
10	ย 9 6 5	1
11	ผ	1
12	ต	1
13	ด	1

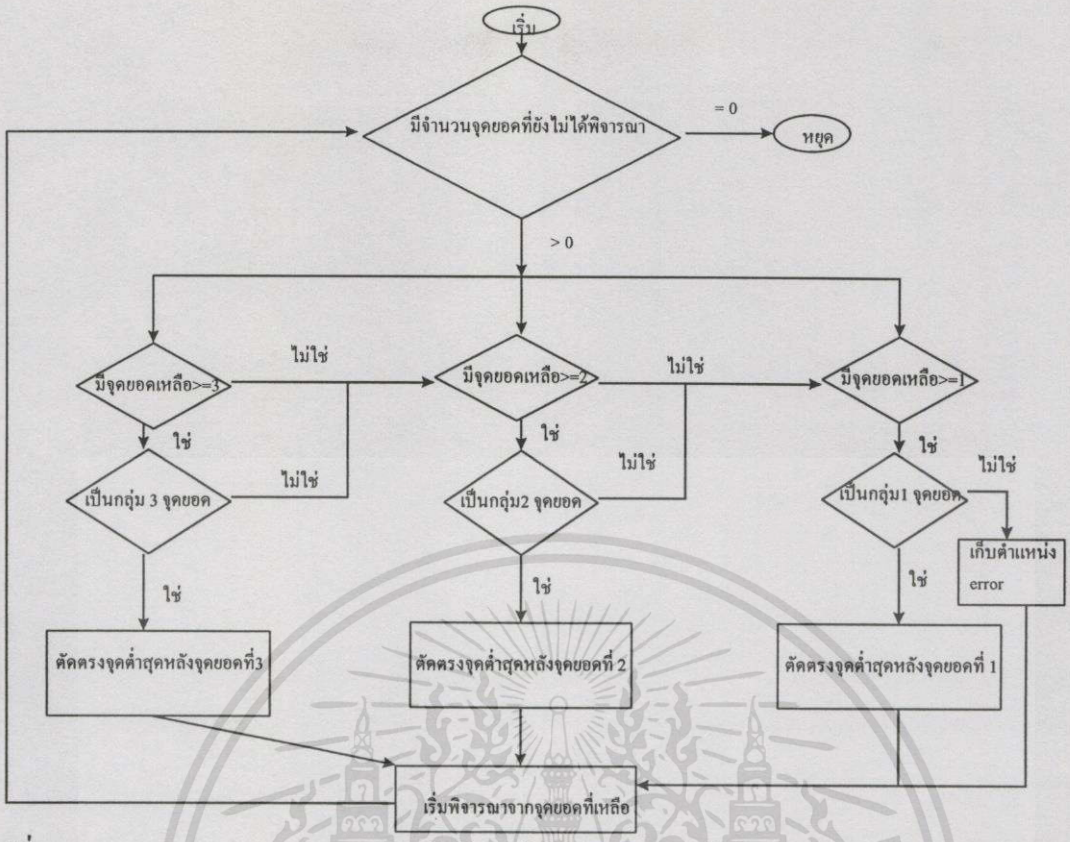
ค่า distance ที่แบ่งกลุ่มย่อยนี้จะเป็นเงื่อนไขที่เพิ่มเติมที่ใช้แยกตัวอักษรในกลุ่ม 2 จุดยอดแบบที่ 1 เช่น ก ภ ถ และในกลุ่ม 3 จุดยอดแบบที่ 1 เช่น ณ ญ ฌ และแบบที่ 2 เช่น ฒ ออกจากตัวอักษรในกลุ่มอื่น ๆ

5.7.2 วิธีในการแยกตัวอักษรที่ไม่ทราบจำนวนตัวติด

การพิจารณาหาจุดตัดของตัวอักษรที่ไม่ทราบจำนวนตัวติด จากความจริงที่ว่าตัวอักษรที่ติดกันจะต้องมีจุดตัดแน่นอนแต่ไม่ทราบว่าจุดตัดจะอยู่บริเวณใด ดังนั้นการพิจารณาหาจุดตัดจึงเริ่มจากการกำหนดจุดต่ำสุดระหว่างจุดยอดทั้งหมดที่มีอยู่ดังที่กล่าวมาแล้วข้างต้น แล้วนำแต่ละส่วนระหว่างจุดต่ำสุดไปพิจารณาตามเงื่อนไขของแต่ละกลุ่มที่กำหนดโดยการพิจารณาจะพิจารณาแบบ 3 2 1 คือการนำมาทีละ 3 จุดยอดพิจารณาก่อนโดยทำการพิจารณาในกลุ่ม 3 จุดยอด ถ้าไม่ใช่ก็จะลดจุดยอดลง 1 จุดยอดเหลือ 2 จุดยอดแล้วทำการพิจารณาในกลุ่ม 2 จุดยอด ถ้าไม่ใช่กลุ่ม 2 จุดยอดอีกก็จะทำการลดจุดยอดลงอีกเหลือ 1 จุดยอดแล้วพิจารณาในกลุ่ม 1 จุดยอดต่อไป ถ้าใช่ลงจุดยอดใดก็ตามก็จะถือว่าจุดต่ำสุดที่นำไปพิจารณานั้นเป็นจุดตัดของตัวอักษร ดังแสดงรูปแบบการตัดไว้ในรูปที่ 5.29 และ Flowchart ในการแยกแสดงไว้ในรูปที่ 5.30

เหตุผลที่ใช้การพิจารณาแบบ 3 2 1 เนื่องจากได้มีการทดสอบแบบอื่น ๆ เช่น 1 2 3 , 2 1 3 หรือ 3 1 2 ผลการตัดในแบบ 3 2 1มีเปอร์เซ็นต์ความถูกต้องมากที่สุด

รูปที่ 5.29 แสดงการแยกตัวอักษรแบบวิธี 3 2 1



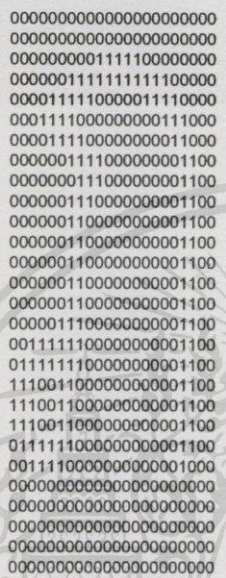
รูปที่ 5.30 แสดง flowchart ของการตัดตัวอักษรไม่ทราบจำนวนตัวติด

5.7.3 ผลการแยกตัวอักษรที่ไม่ทราบจำนวนตัวติด

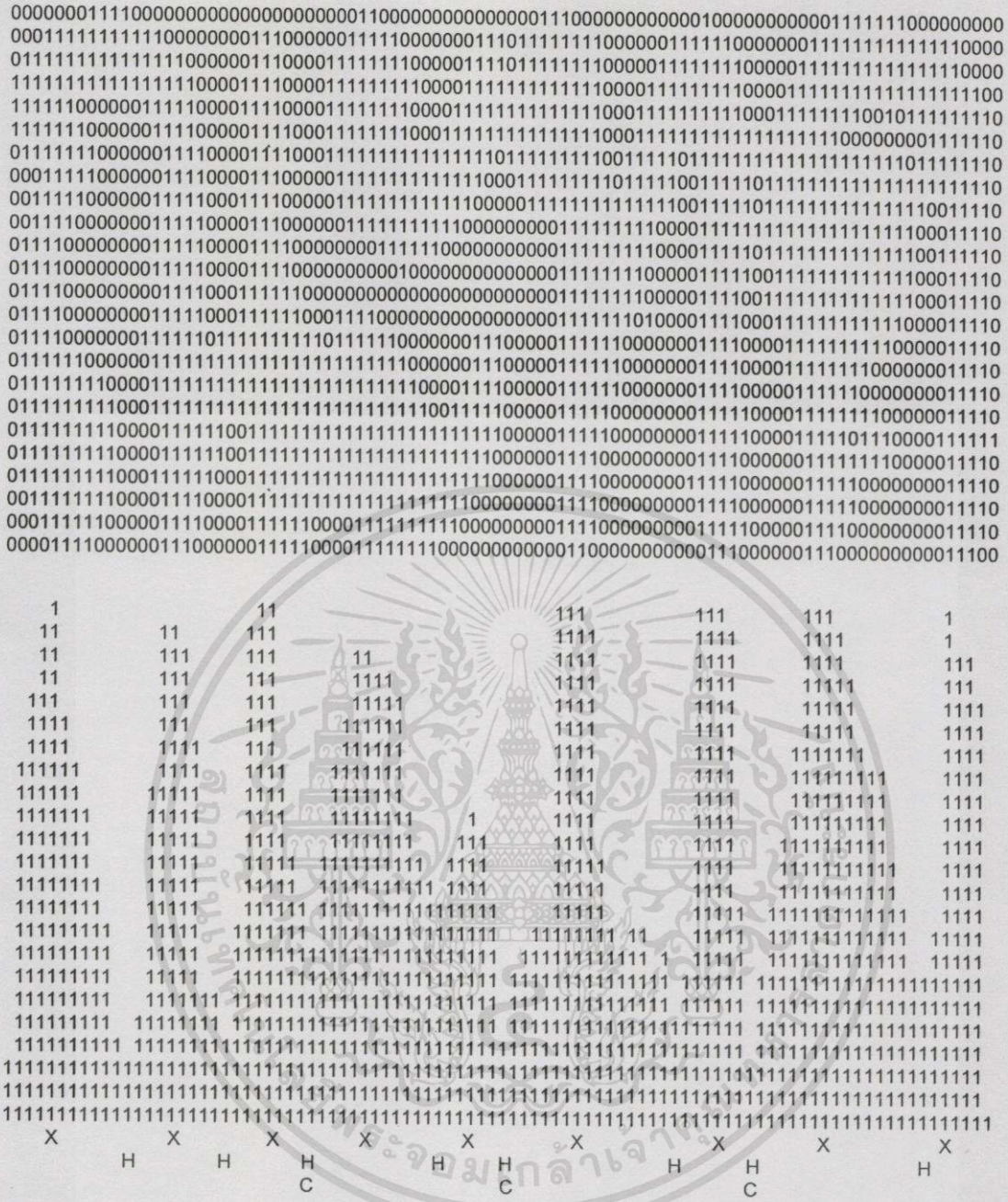
ในการแยกตัวอักษรที่ไม่ทราบจำนวนตัวติดจะสามารถตัดปัญหาการเข้ากลุ่มผิดได้ ซึ่งเมื่อรองนำตัวอย่างตัวอักษรที่ติดกัน 2 ตัวอักษรมาทำการแยกด้วยวิธีที่ใช้กับตัวอักษรที่ไม่ทราบจำนวนตัวติดจะให้ผลการตัดดีขึ้น จากตัวอย่างของตัวอักษรที่ติดกันแน่นอน 2 ตัวอักษรจำนวน 381 ตัวอย่าง สามารถตัดถูกต้อง 358 ตัวอย่างคิดเป็น 93.96 % ซึ่งเพิ่มขึ้นจากเดิม 2.78 % ตัดผิด 23 ตัวอย่างคิดเป็น 6.04 % และจากกลุ่มของตัวอักษรที่ติดกันมากกว่า 2 ตัวอักษรจำนวน 109 ตัวอย่าง และตัวอักษรที่ไม่ติดเลย(ตัวอักษรเดี่ยว ๆ) จำนวน 200 ตัวอย่าง สามารถตัดได้ผลดังนี้

5.7.3.1 ตัวอย่างตัวอักษรที่ตัดถูก

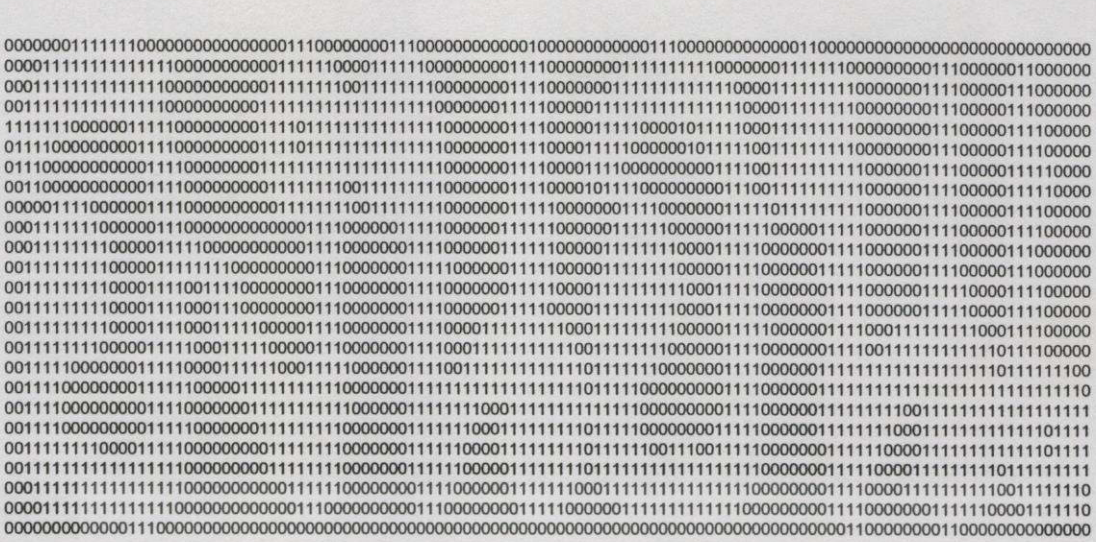
จากตัวอย่างที่เป็นตัวอักษร 1 ตัวอักษรมีจำนวน 200 ตัวอย่างตัดถูกต้อง 184 ตัวอย่างคิดเป็น 92.00 % ดังแสดงในรูปที่ 5.31 และจากตัวอย่างที่มีการติดกันของตัวอักษรมากกว่า 2 ตัวอักษร สามารถตัดได้ถูกต้อง 90 ตัวอย่างคิดเป็น 82.57 % ดังแสดงตัวอย่างที่ตัดถูกต้องไว้ในรูปที่ 5.32, รูปที่ 5.33 และรูปที่ 5.34

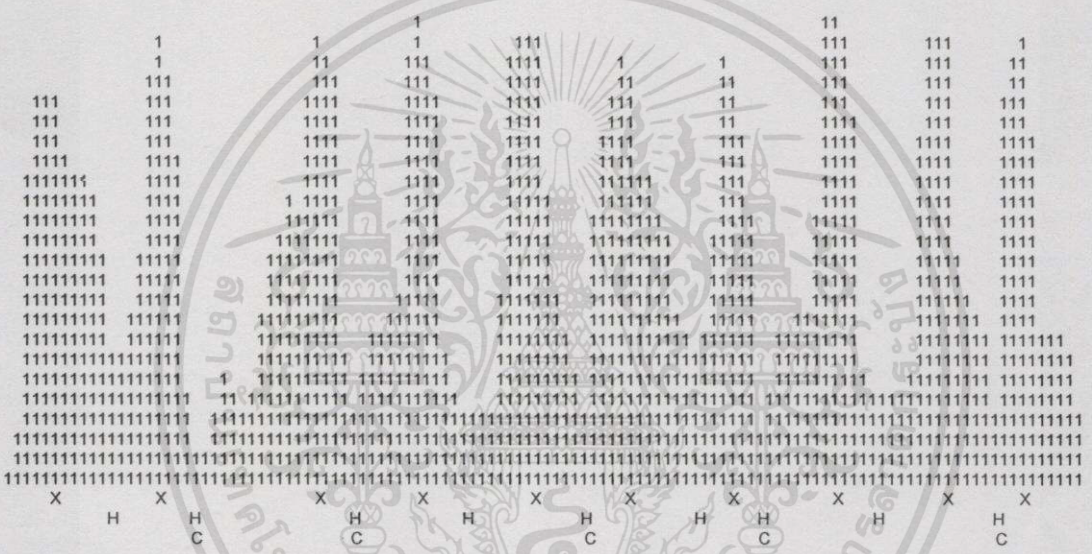


รูปที่ 5.31 แสดงผลการตัดตัวอักษร 1 ตัว

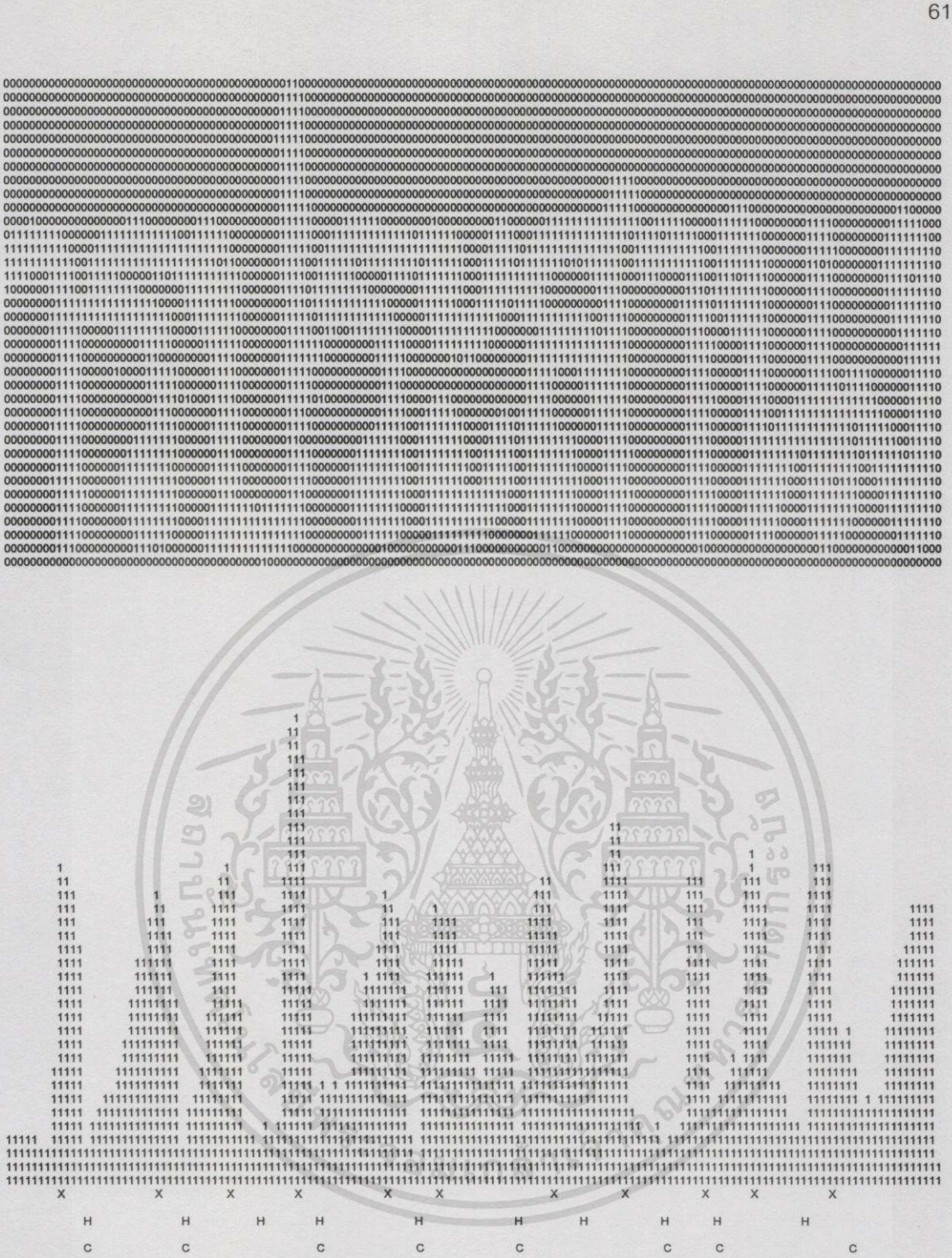


รูปที่ 5.32 แสดงการแยกตัวอักษรที่ติดกัน 4 ตัวอักษร





รูปที่ 5.33 แสดงผลการแยกตัวอักษรที่ติดกัน 6 ตัวอักษร

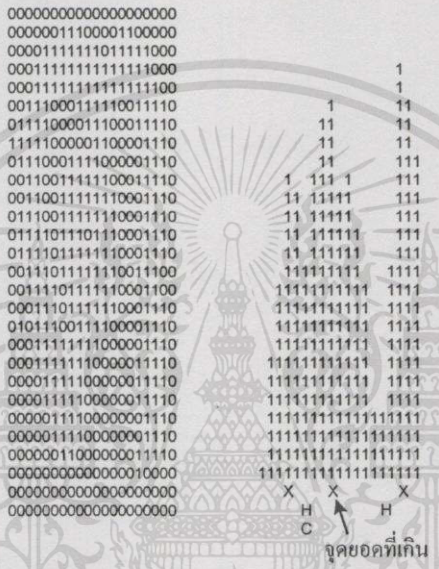


รูปที่ 5.34 แสดงผลการแยกตัวอักษรที่ติดกัน 9 ตัวอักษร

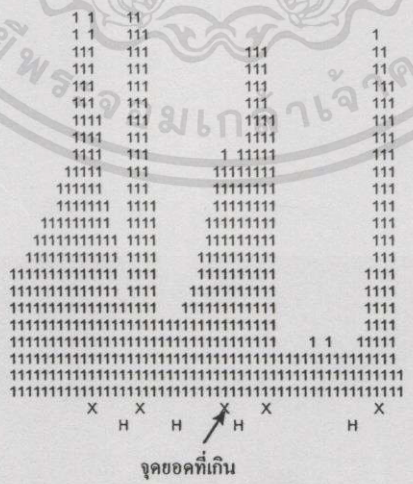
5.7.3.2 ตัวอย่างตัวอักษรที่ตัดผิด

ผลการแยกตัวอักษรเดี่ยว ๆ จำนวน 200 ตัวอย่างตัดผิด 16 ตัวอย่างคิดเป็น 8 % และผลการแยกตัวอักษรที่ติดกันมากกว่า 2 ตัวอักษรจำนวน 109 ตัวตัดผิด 19 ตัวคิดเป็น 17.43 % มีสาเหตุเนื่องจาก

1) จำนวนจุดยอดขาด หรือเกินจากที่กำหนด เช่น ตัวอักษร ต มีจุดยอดเป็น 3 ซึ่งเกินมา 1 จุดยอดทำให้ตัดผิดดังแสดงในรูปที่ 5.35 และ **ขรา** ตัว ร จะต้องมีจุดยอดเป็น 1 แต่กับมีจุดยอดเป็น 2 ทำให้ตัดผิดดังแสดงในรูปที่ 5.36

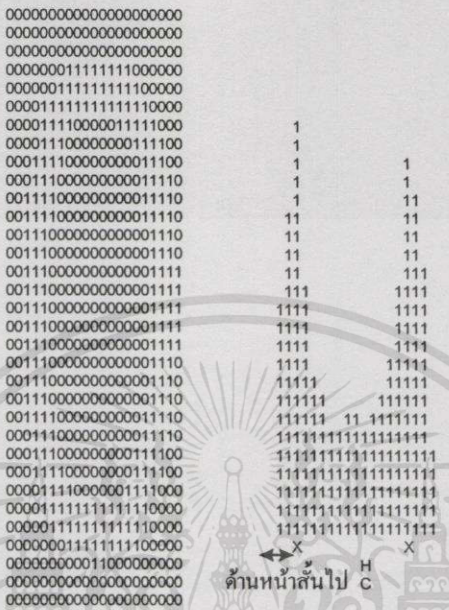


รูปที่ 5.35 แสดงผลตัดผิดของตัวอักษร 1 ตัวเนื่องจากจุดยอดเกิน

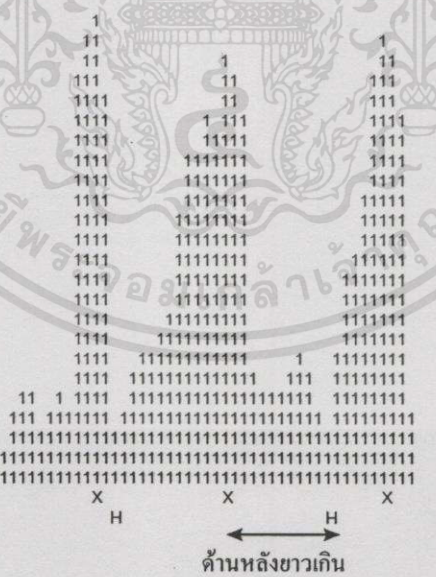


รูปที่ 5.36 แสดงผลตัดผิดของตัวติดมากกว่า 2 ตัวเนื่องจากจุดยอดเกิน

2) ความยาวด้านหน้าหรือด้านหลังมากกว่า หรือน้อยกว่าค่าที่กำหนด เช่น ตัวเลข 0 มีความยาวด้านหน้าน้อยกว่าค่าที่กำหนด ดังแสดงในรูปที่ 5.37 และ **ววว** ตัว ร มีด้านหลังที่มากกว่าค่าที่กำหนดไว้ ดังแสดงในรูปที่ 5.38

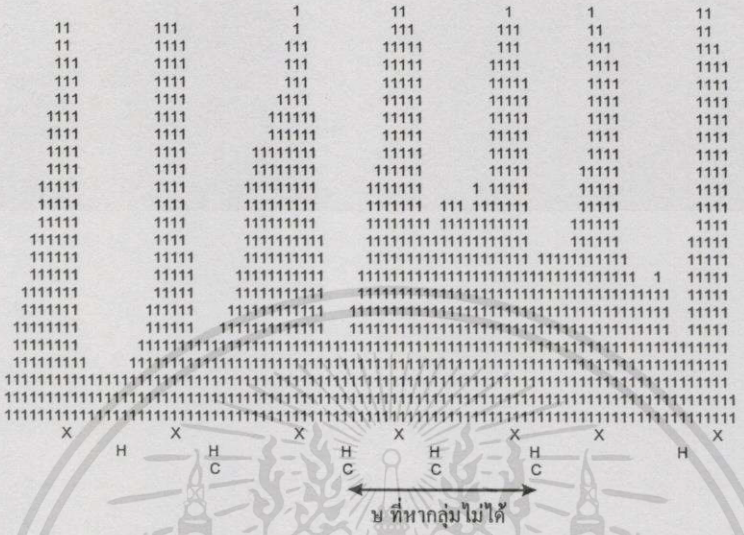


รูปที่ 5.37 แสดงผลตัดผิดของตัวอักษร 1 ตัวเนื่องจากความยาวด้านหน้าน้อยกว่าค่าที่กำหนด

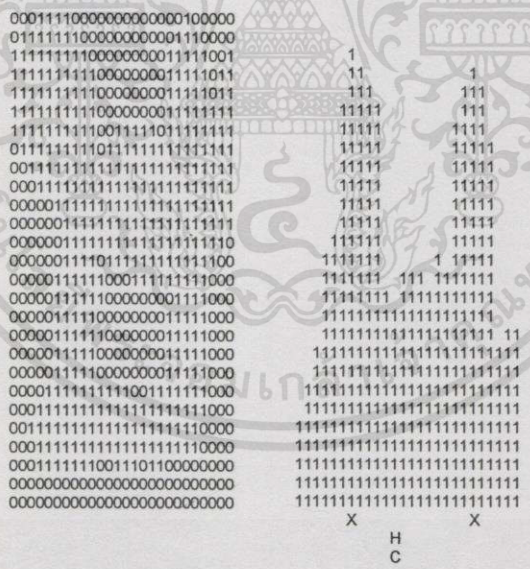


รูปที่ 5.38 แสดงผลตัดผิดของตัวติดมากกว่า 2 ตัวเนื่องจากความยาวด้านหลังมากกว่าค่าที่กำหนด

3) ค่า distance ที่ได้ของตัวอักษรไม่สามารถหากลุ่มเข้าได้ เนื่องจากมีค่ามากกว่าค่าที่กำหนดไว้ เช่น **บรษท** ตัว ษ มีค่า distance ที่มากกว่าค่าในกลุ่ม 2 จุดยอดทั้ง 29 กลุ่ม ดังนั้นจึงไม่สามารถจัดเข้ากลุ่มได้ ดังแสดงในรูปที่ 5.39 หรือตัวอักษร ษ มีค่า distance ไม่เข้ากลุ่มดังรูปที่ 5.40

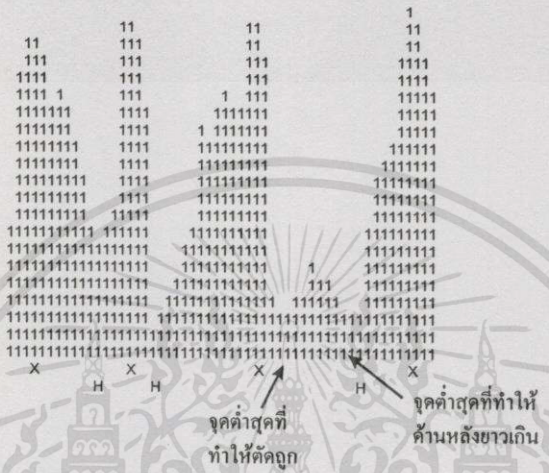


รูปที่ 5.39 แสดงผลตัดผิดของตัวติดมากกว่า 2 ตัวเนื่องจากมีค่า distance ที่เข้ากลุ่มไม่ได้



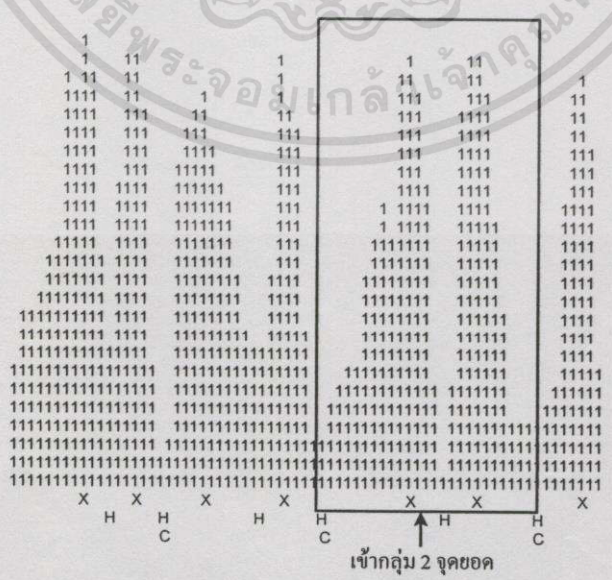
รูปที่ 5.40 แสดงผลตัดผิดของตัวอักษร 1 ตัวเนื่องจากมีค่า distance เข้ากลุ่มไม่ได้

4) การพิจารณาจุดต่ำสุดระหว่างจุดยอด หรือ Hole มีค่าเท่ากันหลายค่าในระหว่างจุดยอดนั้นทำให้การเลือกค่าบางค่าอาจจะทำให้ด้านหน้า หรือด้านหลังยาวเกินจากค่าที่กำหนด เช่นตัวอักษรที่ติดกัน **อรว** ระหว่างตัวอักษร ร กับ ว มีจุดต่ำสุดระหว่างจุดยอดหลายค่า จุดที่เลือกทำให้ความยาวด้านหลังของ ร ยาวเกินจากที่กำหนด ดังแสดงในรูปที่ 5.41 ตัวอักษร 1 ตัวไม่มีปัญหาเกี่ยวกับการเลือกจุดต่ำสุด



รูปที่ 5.41 แสดงผลตัดผิดของตัวติดมากกว่า 2 ตัวเนื่องจากการเลือกค่าจุดต่ำสุดที่ผิด

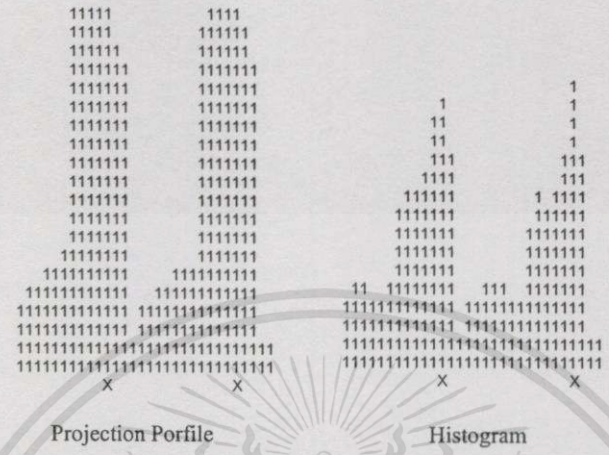
5) ไม่สามารถแยกกลุ่มได้ เช่น **ของก** ตัว งก ง รวมกับครึ่งหนึ่งของ ก ทำให้เข้ากลุ่ม 2 จุดยอด (2+1) แทนที่จะเป็น ง 1 จุดยอดแล้ว ก อีก 2 จุดยอด (1+2) ดังแสดงในรูปที่ 5.42 กรณีของตัวอักษร 1 ตัวไม่พบปัญหานี้เพราะไม่ได้รวมกับตัวอื่น



รูปที่ 5.42 แสดงผลตัดผิดของตัวติดมากกว่า 2 ตัวเนื่องจากเข้ากลุ่มอื่น

5.8 Profile Projection

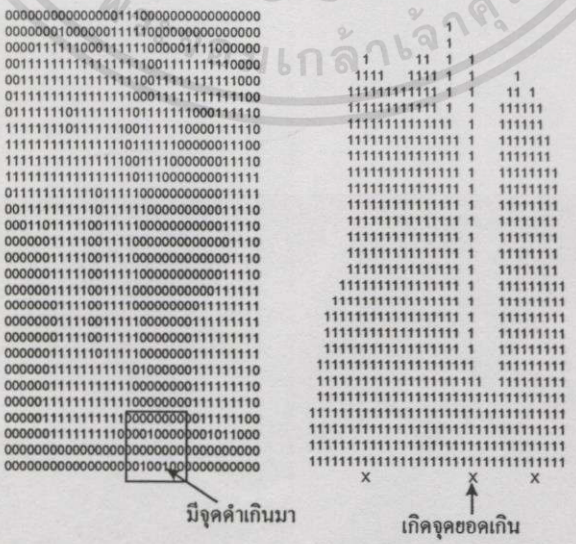
Profile Projection เป็นการนับจำนวนจุดขาว และจุดดำที่อยู่ระหว่างจุดดำที่พบจุดแรกจนถึงจุดดำที่พบจุดสุดท้ายภายในแต่ละคอลัมภ์ ดังรูปที่ 5.43



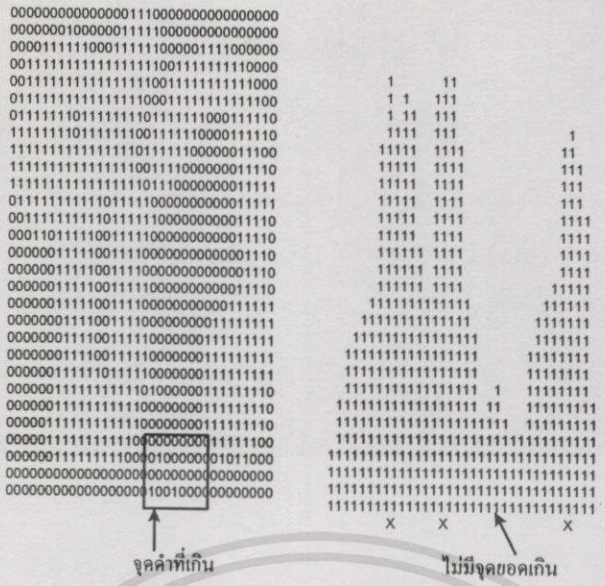
รูปที่ 5.43 แสดงตัวอย่าง Profile Projection เทียบกับ Histogram

จากรูป จะพบว่า Profile Projection จะมีลักษณะคล้ายกับ Histogram แต่จะมีลักษณะของจุดยอดหนา และยาวกว่า

Profile Projection จะมีข้อผิดพลาดที่มากกว่าของ Histogram เนื่องจากถ้าพบจุดดำที่เป็นจุดเกิน หรือจุดที่เกิดจาก error ไม่ว่ากรณีใดก็ตาม ไปอยู่บริเวณที่ใกล้กับจุดดำจริง หรือจุดสำคัญจะทำให้รูปแบบของจุดยอดเปลี่ยนไป ทำให้การกำหนดจุดยอดผิดไปจากความจริง ดังแสดงไว้ในรูปที่ 5.44 แต่ถ้าเป็นแบบ Histogram จุดดำเพียงเล็กน้อยนี้จะไม่เป็นปัญหา ดังแสดงไว้ในรูปที่ 5.45



รูปที่ 5.44 แสดงจุดผิดพลาดของ Profile Projection



รูปที่ 5.45 แสดงตัวอย่างของ Histogram ที่ต่างจาก Profile Projection

จากรูปที่ 5.44 ในวิธี Profile Projection จะพบว่าจุดดำที่เกินมาจะทำให้เกิดจุดยอดเกินมาอีก 1 จุดทำให้จุดยอดจริงถูกเปลี่ยนตำแหน่ง และมีลักษณะของจุดยอดเปลี่ยนไป ส่วนจากรูปที่ 5.45 ในวิธี Histogram จุดดำที่เกินมาไม่ทำให้มีจุดยอดเพิ่มขึ้น จากข้อผิดพลาดของวิธี Profile Projection นี้นับว่าเป็นจุดสำคัญมากในการกำหนดค่าต่าง ๆ ที่อาจจะผิดพลาดไปจึงทำให้ไม่นำวิธี Profile Projection นี้มาใช้ในการกำหนดจุดยอดของตัวอักษร

บทที่ 6
สรุปผลงานวิจัยและข้อเสนอแนะ

ในระบบการจดจำตัวอักษรภาษาไทย (Thai OCR) พบปัญหาที่สำคัญคือการมีตัวอักษรที่ติดกันรวมอยู่ในบทความที่นำมาใช้ทดสอบ และลักษณะการติดกันของตัวอักษรที่พบก็มีอยู่ด้วยกันหลายลักษณะ ซึ่งในงานวิจัยนี้ได้ทำการหาวิธีในการแยกตัวอักษรที่ติดกันในลักษณะแนวนอน เช่น วิธีที่นำมาทดสอบแยกตัวอักษรมีอยู่ 2 แบบคือ วิธี Shortest Path และวิธี Histogram ในวิธี Shortest Path ได้ผลการตัดถูกต้อง 57.70 % ส่วนในวิธี Histogram ได้ผลการตัดถูกต้อง 91.08 % ซึ่งไม่ว่าจะใช้วิธีใดก็ตามในการแยกตัวอักษร ถ้าจะให้ผลเป็นที่แน่นอนก็ต้องนำผลการแยกตัวอักษรที่ได้ไปผ่านระบบการจดจำเพื่อตรวจสอบว่าสามารถจดจำได้หรือไม่ จึงจะถือว่าเป็นผลการแยกที่ถูกต้องแน่นอนที่สุด ดังนั้นไม่ว่าจะใช้วิธีใดในการแยกตัวอักษรที่ติดกันออกจากกันก็ตามยังคงต้องอาศัยระบบการจดจำตัวอักษรช่วยในการตรวจสอบความถูกต้องร่วมอยู่ด้วยเสมอ และระบบการจดจำตัวอักษรที่จะสามารถเพิ่มประสิทธิภาพการจดจำที่ดีขึ้นก็จะต้องมีระบบการแยกตัวอักษรที่ได้ผลการแยกตัวอักษรที่ถูกต้องมากที่สุดเป็นตัวช่วยด้วยเช่นกัน ดังนั้นทั้ง 2 ระบบนี้จะต้องอาศัยซึ่งกันและกันจึงจะทำให้ระบบทั้งหมดมีความสมบรูณ์ที่สุด

ในงานวิจัยนี้การแยกตัวอักษรด้วยวิธี Shortest Path จะมีทั้งข้อดี ดังนี้

1) สามารถตัดตัวอักษรได้ดีถ้าบริเวณที่ติดกันของตัวอักษรเป็นบริเวณที่มีเนื้อของตัวอักษรน้อยที่สุด
2) เส้นตัดของตัวอักษรที่ได้สามารถเอียงตามลักษณะของเนื้อตัวอักษรทำให้เวลาแยกออกจากกันจะได้ความสมบรูณ์ของตัวอักษรทั้งคู่

ข้อเสียดังนี้

1) ถ้าบริเวณที่ติดกันไม่ใช่บริเวณที่มีเนื้อตัวอักษรน้อยที่สุดจะทำให้เลือกเส้นทางเดินผิด เนื่องจากวิธีนี้จะเลือกเส้นตัดจากเส้นทางเดินที่สั้นที่สุดเป็นเส้นตัดตัวอักษรเท่านั้น
2) ถ้ามีจุดดำ error เกิดอยู่ภายนอกตัวอักษรที่ติดกันอาจจะมีผลต่อการคิดค่าทางเดินได้ ทำให้อาจจะเลือกทางเดินผิด

ส่วนในวิธี Histogram ก็มีทั้งข้อดีดังนี้

1) สามารถตัดตัวอักษรได้ถูกต้องถึงแม้ว่าบริเวณที่ติดอาจจะไม่ใช่บริเวณที่มีเนื้อตัวอักษรน้อยที่สุดก็ตาม
2) สามารถแยกตัวอักษร 1 ตัวอักษร และตัวอักษรที่ติดกันมากกว่า 2 ตัวอักษรได้
3) ถ้ามีจุดดำ error ที่บริเวณภายนอกตัวอักษรจะไม่มีผลกระทบกับการกำหนดเงื่อนไข ซึ่งต่างจาก Profile Projection และ Shortest Path

4) ไม่กำหนดความกว้างของตัวอักษร

ข้อเสียดังนี้

1) ไม่สามารถใช้ได้กับตัวอักษรที่ไม่มีจุดยอดเด่น เช่นตัวเลขไทย เนื่องจากไม่สามารถกำหนดจำนวนจุดยอดได้
2) การกำหนดจุดยอดของตัวอักษรที่นำมาทดสอบมีความสำคัญมากถ้าจำนวนจุดยอดที่ได้ผิดจากจำนวนที่กำหนดไว้จะทำให้การตัดตัวอักษรผิดพลาด
3) การกำหนดค่าตัวแปรต่าง ๆ เช่นความยาวด้านหน้า หรือด้านหลัง อาจจะทำให้ผลของการแยกตัวอักษรผิดได้

ผลของการแยกตัวอักษรทั้ง 2 วิธีสามารถเขียนเป็นตารางแสดงเปอร์เซ็นต์ความถูกต้องเทียบกับจำนวนตัวอักษรที่ติดกัน 2 ตัวอักษร 381 ตัวอย่าง ไม่ทราบจำนวนตัวติด 309 ตัวอย่างได้ดังตารางที่ 6.1 จากตารางพบว่าในวิธี Histogram จะได้ผลการแยกตัวอักษรออกจากกันได้ถูกต้องมากที่สุด แต่ที่ยังมีข้อผิดพลาดเนื่องจาก

1) จำนวนจุดยอดเกินหรือขาด
2) ความยาวด้านหน้าหรือด้านหลัง มากหรือน้อยกว่าที่กำหนด
3) การเลือกจุดต่ำสุดที่มีหลายค่า

ซึ่งรายละเอียดของข้อผิดพลาดได้กล่าวไว้แล้วในบทที่ 5

ตารางที่ 6.1 แสดงผลการการแยกตัวอักษรในวิธีต่าง ๆ

วิธีที่ใช้	จำนวนตัวอักษรที่ติด	เปอร์เซ็นต์ความถูกต้อง
Shortest Path ทางตรง	2	57.70
Shortest Path ทางเฉียง	2	58.22
Shortest Path ใช้ขอบเขต	2	76.24
Histogram ติดกันแน่นอน	2	91.08
Histogram ไม่ทราบจำนวนตัวติด	1	92.00
Histogram ไม่ทราบจำนวนตัวติด	2	93.96
Histogram ไม่ทราบจำนวนตัวติด	มากกว่า 2	82.57

แนวทางการพัฒนาต่อไปในอนาคต

1) ปรับปรุงการกำหนดจุดยอดของ Histogram ให้ดีขึ้นเพราะจะทำให้วิธีนี้มีประสิทธิภาพเพิ่มเนื่องจากการกำหนดจุดยอดเป็นจุดสำคัญของวิธีนี้ ถ้าตัวอักษรที่เข้ามาทดสอบมีจุดยอดตรงตามที่กำหนด ไม่มีจุดยอดที่ขาด หรือเกิน จะทำให้เงื่อนไขที่ใช้สามารถใช้ตัดตัวอักษรได้ถูกต้อง การปรับปรุงการกำหนดจุดยอดอาจจะกำหนดด้วยวิธีอื่น หรือใช้ Histogram แบบอื่นมาช่วยให้สามารถกำหนดจุดยอดให้ได้ตำแหน่งที่แน่นอน
2) ทำการเพิ่มเติมให้มีการหาแยกส่วนต่าง ๆ ของตัวอักษรในบทความออกมาโดยมีการตรวจสอบว่ามีตัวอักษรใดบ้างติดกัน ตัวอักษรใดไม่ติด และแบ่งเป็นกลุ่มของตัวอักษรที่มีลักษณะเหมือนกันในแต่ละบทความ เพื่อสะดวกในการนำมาทดสอบต่อไป

บรรณานุกรม

1. Liang Su. Shridhar M. and Ahmadi M.,"Efficient Algorithms for Segmentation and Recognition of Printed Characters in Document Processing", Electrical of IEEE (1993) pp. 240-244
2. YILU,"Machine Printed Character Segmentation An Overview", Pattern RecognitionVol. 28 no. 4 (1994) pp. 67-79
3. Panich W. Jitapunkul S. and Choruengwiwat P.,"Segmentation of Connected Characters using Distinctive Features of Thai Characters in Thai Character Recognition System", 20th Electrical Engineering Conference (1997) pp. 338-342
4. Kahan S. Pavilids T. and Baird H.,"On the recognition of printed characters of any font any size",IEEE Trans Pattern Analysis Mach Intell Vol. 9 no. 2 (1987) pp. 274-287
5. Wang J. and Jean J.,"Segmentation of Merged Characters by Neural Networks and Shortest Path",Pattern Recognition Vol. 27 no.6 (1994) pp. 649-658
6. Jin Ho Kim. et. al.,"Segmentation of Touching Characters in Printed Korean/English Document Recognition", Electrical of IEEE (1996) pp.438-443
7. Liang Su. Shridhar M. and Ahmadi M.,"Segmentation of Touching Characters in Printed Document Recognition",Pattern Recognition Vol. 27 no. 6 (1994) pp. 825-840
8. Westall J.M. and Narasimha M.S.,"Vertex Directed Segmentation of Handwritten Numerals", Pattern Recognition Vol. 26 no.10 (1993) pp. 1473-1486

ภาคผนวก

การแยกตัวอักษรภาษาไทยที่ติดกันโดยวิธี Shortest Path

Segmentation of Thai connected characters with Shortest Path method

จรรยา เกียรติศิริอนันต์* และบุญธีร์ เครือตราชู**

*สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

3 หมู่ 2 ถนนฉลองกรุง เขตลาดกระบัง กรุงเทพฯ 10520

โทร (02)3269969 E-Mail:boontee@diamond.cc.kmitl.ac.th

บทคัดย่อ

การวิจัยนี้มุ่งสร้างระบบการแยกตัวอักษรภาษาไทย ที่มีลักษณะการติดในแนวนอน และมีจำนวนการติดของตัวอักษร 2 ตัว เช่น ษท นม โดยใช้วิธี Shortest Path ทางตรง [2] ซึ่งได้มีการทดลองมาแล้วในภาษาอังกฤษ มาลองใช้ในภาษาไทยพบว่ามีความผิดพลาดเกิดขึ้น และได้เสนอวิธีการแก้ไขเป็นแบบเฉียง เพื่อทำการแก้ไขปัญหาที่เกิดขึ้นให้ได้มากที่สุด การแสดงผลจะเป็นการสรุปความเป็นไปได้ในการแยกตัวอักษร ตัวอักษรที่ใช้เป็นตัวอย่างทั้งหมดมี 383 ตัวอย่าง

Abstract

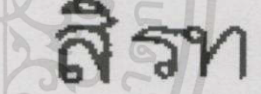
This research attempts to isolate touching Thai characters in vertical detection from the scan image. The touching chracter has two characters for example ษท นม. There are 383 examples from many articles and newspaper. The method used to isolate character is Shortest Path [2] which tested successfully on English character. The modified Shortest Path is present with some improvement result.

1. คำนำ

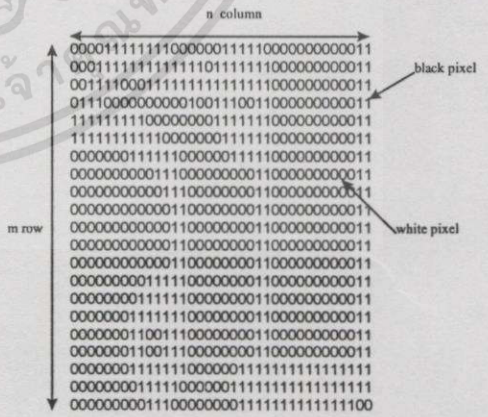
จากการศึกษาในระบบ OCR พบว่าจะมีปัญหาอยู่ปัญหาหนึ่งที่สำคัญคือ การที่มีตัวอักษรติดกัน (Touching character) เนื่องจากว่าภาษาไทยเป็นภาษาที่มีระดับมีทั้งสระ พยัญชนะ และวรรณยุกต์เป็นส่วนประกอบ ซึ่งเมื่อนำไปใช้รวมกันจะทำให้เกิดการติดกันได้มาก และหลายรูปแบบ ลักษณะการติดอาจจะติดกันในแนวดิ่ง (horizontal) เช่น สิ นุ หรืออาจจะติดกันในแนวนอน (vertical) เช่น ร ท ไ ม ดังแสดงไว้ในรูปที่ 1 จากบทความบทความหนึ่ง[1] ที่ศึกษามาพบว่าการติดกันในแนวดิ่งนั้น ถ้าทราบเส้นบรรทัดก็จะสามารถแยกตัวอักษรที่ติดกันนั้นได้ง่าย แต่การติดกันในแนวนอนนั้นไม่สามารถใช้เส้นบรรทัดแยกออกได้ทำให้มีความยากกว่าการแยกในแนวดิ่ง ดังนั้นจึงทำให้งานวิจัยนี้จะเน้นการแยกตัวอักษรในแนวนอนเป็นหลัก

2. การศึกษาการติดกันของตัวอักษรภาษาไทย

จากบทความ[1] ที่กล่าวมาข้างต้นจะมีลักษณะของตัวอักษรที่ติดกันเป็นดังตารางที่ 2 ซึ่งในการทดลองของบทความที่กล่าวมานั้นข้อมูลที่ใช้ได้มาจาก Microsoft Word 6 font AngsanaUPC และ CordiaUPC ที่มีขนาด 12 14 และ 16 จาก Laser Jet 5L แต่จากการที่ได้ศึกษามาพบว่า ถ้า scan ตัวอักษรจากสิ่งพิมพ์ทั่ว ๆ ไป เช่นหนังสือพิมพ์ จะทำให้เปอร์เซ็นต์ของการติดกันของตัวอักษรในตารางที่ 2 เปลี่ยนไปดังแสดงไว้ในช่องสุดท้ายในตารางที่ 2 จากตารางจะพบว่าเปอร์เซ็นต์ที่ติดกันในแนวนอนมีมากกว่า ตัวอย่างตัวอักษรที่ใช้ทดลองทั้ง 383 ตัวอย่าง เป็นจุดขาวดำ ได้มาจากการ scan บทความในหนังสือพิมพ์ทั่ว ๆ ไปที่ resolution 300 dpi แล้วตัดตัวอักษรเฉพาะที่เห็นว่ามีการติดกันโดยใช้โปรแกรม Photoshop



รูปที่ 1 การติดกันของตัวอักษรในแนวดิ่ง (สิ) และในแนวนอน (รท)



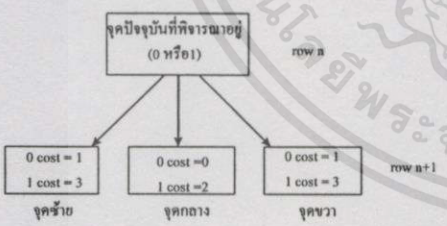
รูปที่ 2 การกำหนดค่าในการแยกตัวอักษรโดยใช้ Shortest Path ทางตรง

กลุ่ม	กลุ่มอักษรที่ติดกัน	ตัวอย่าง	% จาก [1]	% จากหนังสือพิมพ์ทั่วไป
1	1&1	บัติ ค์กิ	0.2	0.9
2	1&2 (คอลัมน์เดียวกัน)	ปี ปี	14.5	11.2
3	1&2 (ต่างคอลัมน์)	ร์ไ ร์ไ	9.2	8.9
4	2&1 (ต่างคอลัมน์)	ปรี	0.2	0.7
5	1&6	อี๋	38.1	9.4
6	1&3	ด้ ศึ สั	5.9	13.8
7	2&3	ไม	0.2	2.7
8	3&5	นุ ดู ผู	31.0	3.4
9	3&3	รท เภ	0.2	39.0
10	เกิน 2 ตัว	สี่ รูป	0.5	10.1

ตารางที่ 2 เปอร์เซ็นต์กลุ่มตัวอักษรที่ติดกันจากบทความที่กล่าวถึง [1] เทียบกับที่ทำการทดลอง

2.1 Shortest Path ทางตรง [2]

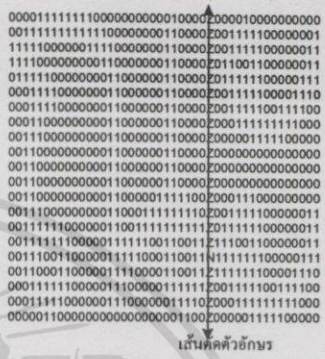
การแยกตัวอักษรโดยใช้ Shortest Path ทางตรงนี้ จะพิจารณาตัวอักษรที่ติดกันแต่ละตัวอย่างเป็นขนาด n แถว (row) m หลัก (column) ดังแสดงในรูปที่ 2 ในการแยกจะทำการกำหนดค่าราคา (cost) ของแต่ละทางเดิน (path) โดยการกำหนดค่า cost แต่ละจุดที่ผ่านเป็นดังรูปที่ 3



รูปที่ 3 ค่า cost ในแต่ละจุด

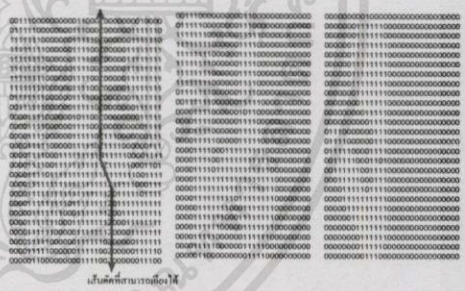
ในการหาเส้นทางเดินที่จะใช้ตัดตัวอักษร 2 ตัวที่ติดกัน จะเริ่มจากจุดที่อยู่ใน row บนสุดไปถึงจุดที่อยู่ใน row ท้ายสุด โดยทางเดินจาก row n ไป row n+1 จะต้องเป็นจุดที่ติดกันซึ่งเป็นไปได้ 3 ทาง ดังแสดงในรูปที่ 3 ถ้าเป็นทางดิ่งตรงจะมีค่า cost ที่เลื่อนจาก row n ไป row n+1 ต่ำสุด แต่ถ้าเป็นทางเฉียงจะมีค่าเท่ากันไม่ว่าจะเป็นเฉียงซ้ายหรือขวา โดยค่า cost ของจุดที่เป็นจุดดำจะมีค่า cost มากกว่าจุดขาว เพื่อให้เส้นทางเดินเดินไปตามจุดขาวและตัดเนื้อตัวอักษร(จุดดำ)น้อยที่สุด หลังจากได้เส้นทางเดินแล้วก็จะทำการเปลี่ยนจุดเริ่มต้นไปที่ column ถัดไป และ column ที่ให้ค่า cost น้อยที่สุดจะถูกเลือกเป็นเส้นตัดตัวอักษร (least-cost Path) จากตัวอย่างทั้ง 383 ตัวอย่างพบว่า

1. ถ้าบริเวณที่ติดกันของตัวอักษรบางกว่าบริเวณอื่น ๆ จะทำให้จุดตัดมีความถูกต้อง ดังรูปที่ 4



รูปที่ 4 ตัวอย่างที่วิธี Shortest Path ทางตรง ตัดถูกต้อง

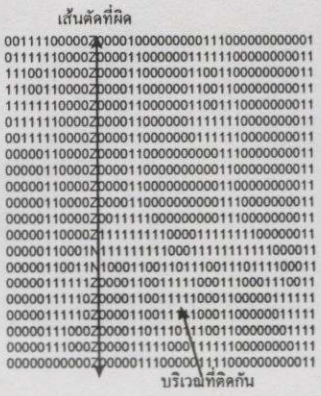
2. เส้นตัดตัวอักษรของวิธี Shortest Path ทางตรงสามารถโค้งตามลักษณะของตัวอักษรได้ทำให้เวลาแบ่งตัวอักษรออกเป็น 2 ส่วนจะได้เนื้อตัวอักษรที่สมบูรณ์ ดังรูปที่ 5



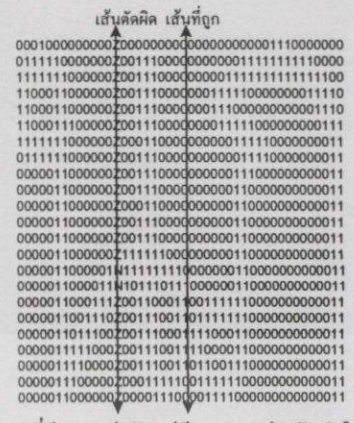
รูปที่ 5 ตัวอย่างเส้นตัดที่สามารถโค้งได้ตามลักษณะของตัวอักษร

3. ถ้าบริเวณที่ติดกันของตัวอักษรหนากว่าบริเวณอื่น ๆ จะทำให้เลือกเส้นตัดที่ผิด

3.1 ถ้าบริเวณที่ติดกันของตัวอักษรหนากว่าบริเวณกลางตัวอักษร จะได้เส้นตัดที่ผิด เช่นจากรูปที่ 6

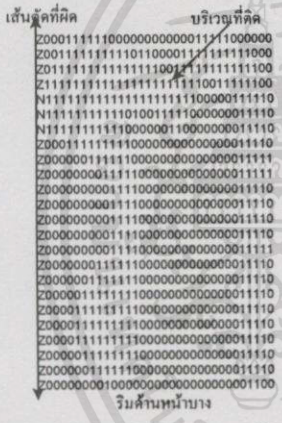


รูปที่ 6 การตัดผิดเนื่องจากความหนาของตัวอักษรน้อยกว่าบริเวณที่ติด

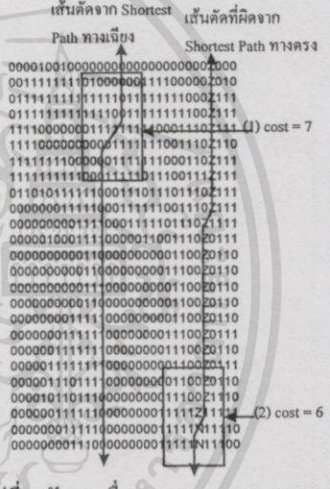


รูปที่ 8 ตัวอักษรที่มี cost เท่ากันแต่มี column ต่างกันทำให้เลือกเส้นตัดผิด

3.2 บริเวณที่ติดกันของตัวอักษรหนากว่าบริเวณริมด้านหน้าหรือริมด้านหลังของตัวอักษรทำให้ตัดผิด ดังรูปที่ 7



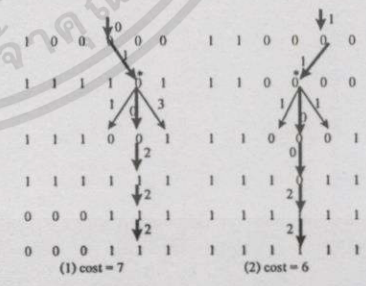
รูปที่ 7 ริมด้านหน้าของตัวอักษรที่มีความหนาน้อยกว่าบริเวณที่ติด



รูปที่ 9 ปัญหาเนื่องจาก Shortest Path ทางตรง

4. ค่า cost ที่ได้มีหลายค่าเท่ากันแต่มี column ต่างกันทำให้เวลาพิจารณาเส้นตัดจะเลือกเส้นที่มี column น้อยกว่าเป็นเส้นตัด ซึ่งอาจจะเป็นเส้นที่ถูก หรือผิดก็ได้ ดังรูปที่ 8 จากรูปพบว่าเส้นตัดแรกมี cost เท่ากับ 4 มี column เป็น 13 ส่วนเส้นที่สองมี cost เท่ากันแต่มี column มากกว่าจึงทำให้เลือกเส้นแรกเป็นเส้นตัดซึ่งเป็นเส้นที่ผิด

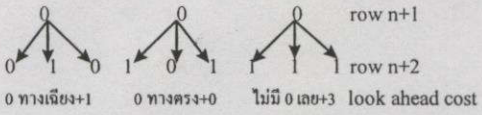
5. จากรูปที่ 9 พบว่าภายในกรอบสี่เหลี่ยม(1)น่าจะเป็นบริเวณที่มีเส้นตัดที่ถูกต้อง เพราะว่าเป็นส่วนที่บาง แต่ส่วนที่บางนี้มีลักษณะเฉียงลงจึงทำให้เวลามองทางตรง หรือคิดตาม Shortest Path ทางตรง ได้ค่า cost มากกว่าบริเวณที่ตัดผิด เหตุที่ Shortest Path ทางตรงตัดผิดนั้นลองมาพิจารณาค่า cost ภายในกรอบสี่เหลี่ยมซึ่งแสดงการคำนวณค่า cost ไว้ในรูปที่ 10



รูปที่ 10 ข้อผิดพลาดในการคำนวณค่า cost ใน Shortest Path ทางตรง

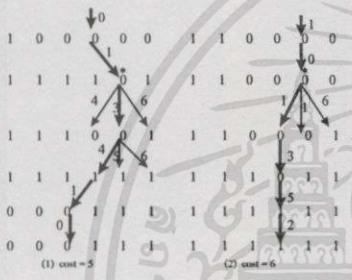
2.2 Shortest Path ทางเฉียง

วิธี Shortest Path ทางเฉียงจะมีลักษณะคล้ายกับ Shortest Path ทางตรงตรงที่การกำหนดค่า cost ในแต่ละจุดและการคิดค่า cost รวมเหมือนกัน แต่จะต่างกันตรงที่เวลาคิดทางเดินว่าจะเดินไปทางไหนต่อ จะมีการพิจารณาล่วงหน้าอีก 1 แถว เป็น row n+2 ดังรูปที่ 11



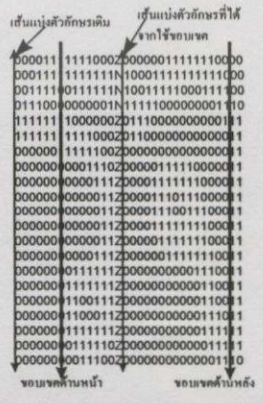
รูปที่ 11 การกำหนดค่า look ahead cost ของ Shortest Path ทางเฉียง

ในการทำ Shortest Path ทางเฉียงนั้นในการคิดค่า cost จะยังเหมือนเดิมคือดูว่าจุดใน row n+1 เป็น 0 หรือ 1 และเป็นแนวเฉียงหรือแนวตรง(ดังรูปที่3) แต่จะมีการคำนวณค่า look ahead cost ซึ่งใช้จุดในระดับ row n+2 มาช่วยในการเลือกว่าจะเลือก path ไหนแต่จะไม่นำมารวมเป็นค่า cost ในการเลื่อนจาก row n ไป row n+1 โดยในการเลือกจะเลือกไปทางที่มี cost + look ahead cost น้อยที่สุด ตัวอย่างการคำนวณ look ahead cost ดังรูปที่ 12



รูปที่ 12 การกำหนดค่า look ahead cost จากวิธี Shortest Path ทางเฉียง

จากผลการทดลอง พบว่ายังมีปัญหาที่ริมด้านหน้า หรือริมด้านหลังบางกว่าบริเวณที่ติดและเป็นปัญหาที่พบมากที่สุด ซึ่งจะแก้ได้จากการกำหนดขอบเขตในการหาเส้นทางเดิน ขอบเขตที่ใช้จะใช้ค่าความกว้างของตัวอักษรซึ่งได้จากการพิจารณาหาตัวอักษรที่มีความกว้างน้อยที่สุดเป็นขอบเขตด้านหน้าซึ่งก็คือ เ และมีความกว้างมากที่สุดเป็นขอบเขตด้านหลังคือ ฌ เมื่อได้ขอบเขตแล้วจะทำการหาเส้นทางเดินภายในขอบเขต ดังแสดงในรูปที่ 13



รูปที่ 13 การใช้ขอบเขตช่วยในการหาเส้นตัดตัวอักษร

3. ผลการทดลอง

จากการทดลองตัวอย่างทั้ง 383 ตัวอย่างพบว่าการแยกตัวอักษรจะได้เปอร์เซ็นต์ความถูกต้องดังตารางที่ 3 จากตาราง Shortest Path ทางตรงมีข้อผิดพลาดเกิดขึ้น ซึ่งเมื่อแก้ไขเป็นทางเฉียง ก็สามารถแก้ไขปัญหาได้เล็กน้อย จากผลของการตัดทั้งหมดพบว่ามีปัญหาการตัดผิดทางริมด้านข้างของตัวอักษรอยู่ถึง 51.875 % ของตัวที่ตัดผิด ปัญหาตัดผิดบริเวณกลาง 26.25 % ปัญหาค่า cost เท่ากัน 12.50 % และปัญหาทางเฉียง 9.38 %

วิธี	เปอร์เซ็นต์ความถูกต้อง (%)
Shortest Path ทางตรง	57.70
Shortest Path ทางเฉียง	58.22
Shortest Path ทางเฉียงและใช้ขอบเขต	76.24

ตารางที่ 3 ตารางเปรียบเทียบผลการทดลอง

4. สรุป

ข้อผิดพลาดที่เกิดขึ้นเนื่องจากการตัดที่ผ่านเนื้อตัวอักษรน้อยที่สุดอาจไม่ใช่เส้นตัดที่ถูกต้อง จึงทำให้การตัดที่ใช้ค่า cost น้อยที่สุดไม่ว่าทางตรงหรือทางเฉียงผิด แต่เปอร์เซ็นต์ที่ตัดได้ก็ถือว่าสูงพอควร ถ้าได้นำการทำ recognition มารวมด้วยก็จะทำให้เปอร์เซ็นต์การตัดถูกต้องสูงขึ้นมาก

เอกสารอ้างอิง

[1] W. Panich, S. Jitapunkul, P. Choruengwiwat, "Segmentation of Connected Characters using Distinctive Features of Thai Characters in Thai Character Recognition System", The 20th Electrical Engineering Conference, Bangkok,Thailand, pp.338-342, Nov 1997.

[2] J. Wang, J. Jean, "Segmentation of Merged Characters by Neural Networks and Shortest Path",Proceedings of The IEEE, Vol. No. , pp. 649-658, Nov 1993.

[3] Su Liang, M. Ahmadi, M. Shridhar, "Efficient Algorithms for Segmentation and Recognition of Printed Characters in Document Processing", Proceedings of The IEEE, pp. 240-244, 1993.

ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาวจรรยา เกียรติศิริอนันต์
วันเดือนปีเกิด	5 มกราคม 2513
สถานที่เกิด	จังหวัดราชบุรี
วุติการศึกษาระดับปริญญาตรี	วท.บ. (ฟิสิกส์)
สถานที่สำเร็จการศึกษา	คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร ปีการศึกษา 2534