หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

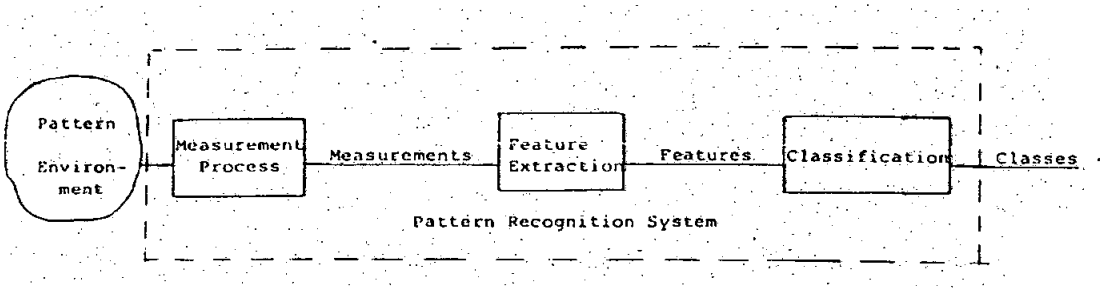
บทที่ 2

ทฤษฎีของระบบจดจำ

2.1 ทฤษฎีเบื้องต้น

วิชาเรื่องการจดจำรูปแบบ (pattern recognition) เป็นวิชาที่มีมานาน เป็นเรื่องเกี่ยวกับการทำให้เครื่องมือมีความฉลาดขึ้น โดยสามารถเรียนรู้และจำแนกความแตกต่างของรูปแบบอินพุทต่าง ๆ เช่น รูปร่าง ขนาด สี เสียง ฯลฯ การประยุกต์ใช้งานมีอย่างกว้างขวาง เช่น การสร้างเครื่องมือ (คอมพิวเตอร์) จำแนกข้อมูลรูปถ่ายดาวเทียม หรือการสร้างเครื่องอ่านตัวอักษร (OCR) ในวิทยานิพนธ์ฉบับนี้จะกล่าวทฤษฎีการจดจำรูปแบบโดยเน้นวิธีการสำหรับระบบจำแนกตัวอักษร

ระบบจดจำรูปแบบทั่ว ๆ ไป มีบล็อกการทำงานดังรูปที่ 2.1



รูปที่ 2.1 แสดงบล็อกของระบบจดจำรูปแบบ

เราสามารถแบ่งการทำงานได้เป็น 3 ส่วน ดังนี้

1. จากข้อมูลรูปแบบในธรรมชาติจะถูกนำมาผ่านขบวนการวัด (measurement

process) เพื่อวัดข้อมูลอินพุท เพื่อใช้เป็นตัวแสดงคุณสมบัติของวัตถุที่ถูกวัด ซึ่งในขบวนการนี้ต้องอาศัยตัวรับสัญญาณ (sensor) เป็นอุปกรณ์วัดข้อมูล

2. ข้อมูลที่ได้จากขบวนการในข้อ 1. จะถูกผ่านขบวนการแยกลักษณะสำคัญ (feature extraction) เพื่อเลือกเอาเฉพาะข้อมูลที่บ่งถึงลักษณะเด่นของวัตถุนั้น เพื่อใช้เป็นตัวบอกความแตกต่าง ความเหมือนของข้อมูลในกลุ่มเดียวกัน ข้อมูลที่ได้จะเรียกเป็นลักษณะสำคัญ (feature)

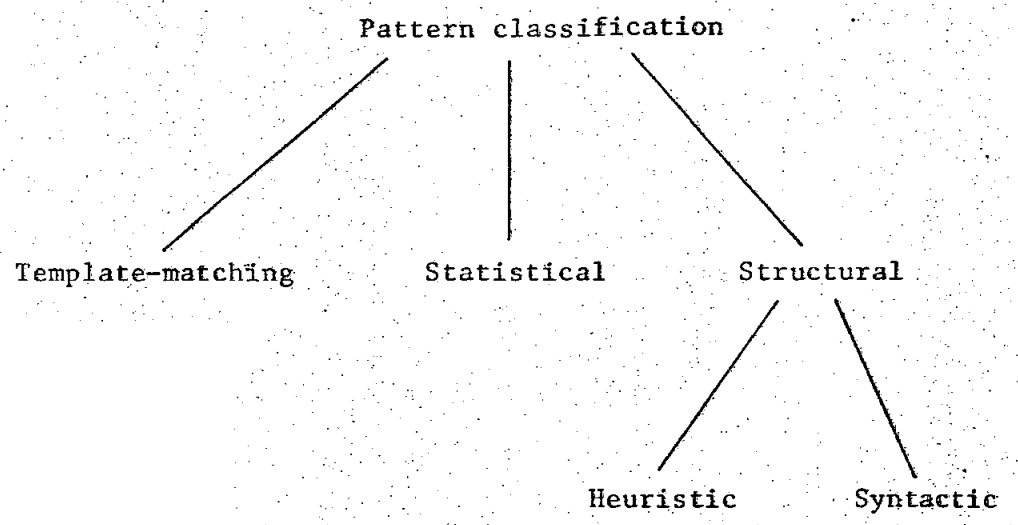
3. ข้อมูลลักษณะสำคัญจะถูกขบวนจำแนกข้อมูลเป็นตัวตัดสินว่า วัตถุที่เราได้ทำการวัดนั้น อยู่ในกลุ่มใด

จากที่ได้กล่าวมาแล้วจะเห็นได้ว่าในการสร้างระบบจดจำข้อมูลมีปัญหาที่เราต้องขบคิดอยู่ 3 ประการ คือ

1. จะใช้อะไรเป็นตัววัดข้อมูลจากรูปแบบ (วัตถุ)
2. จะเลือกลักษณะสำคัญอะไรดี
3. การจำแนกข้อมูล จะใช้อะไรเป็นตัวตัดสิน

สำหรับทางเลือกในการแก้ปัญหาทั้งสามนั้น จะได้กล่าวต่อไป

ในระบบจดจำข้อมูลสามารถแบ่งวิธีในการจำแนกข้อมูลได้ 3 วิธี ดังแสดงในแผนภูมิในรูปที่ 2.2



รูปที่ 2.2 แสดงแผนภูมิแสดงวิธีในการจำแนกข้อมูล

2.2 การตัดสินใจเลือกวิธีจำแนกข้อมูล

ก่อนอื่นขอให้พิจารณาวิธีการในแต่ละวิธีเสียก่อน

1. Template-matching ในวิธีการนี้เซตของข้อมูลต้นแบบ (templates หรือ prototypes) จะถูกเก็บไว้ รูปแบบที่ไม่ทราบ (unknown pattern) จะถูกนำไปเทียบ (match) กับข้อมูลต้นแบบ การตัดสินใจว่า แบบที่ไม่ทราบจะอยู่ในกลุ่มข้อมูลต้นแบบใดขึ้นกับ กฎเกณฑ์ความเหมือน (simlarity criterion) [12] โดยปรกติถ้าสร้างบนระบบง่าย ข้อมูลต้นแบบจะเก็บในลักษณะเป็นข้อมูลดิบ (raw data) วิธีนี้ใช้กันมากในการจดจำตัวอักษรตัวพิมพ์ [13,15,16] ข้อดีของวิธีการนี้ คือ เป็นวิธีที่ง่าย มีลำดับขั้นตอนประมวลข้อมูลน้อย การทำงานมีอัตราเร็วมาก ข้อเสีย คือ ถ้ารูปแบบตัวอักษรมีการปรวนแปรมากการเลือกข้อมูลต้นแบบที่ดีจะยากมาก

2. Statistical (decision-theoretic) เป็นวิธีที่ใช้ความรู้ทางสถิติมาช่วยในการวิเคราะห์ข้อมูล โดยเซ็ทของการวัดจะถูกเรียกว่า ฟีเชอะ ซึ่งได้มาจากการแยกคุณสมบัติที่สำคัญจากข้อมูลดิบ ส่วนของฟีเชอะนี้จะต้องมีการแปรเปลี่ยนน้อยกับการแปรปรวนของรูปแบบ กฎเกณฑ์ในการตัดสินใจใช้วิธีการดังต่อไปนี้ discriminant function, minimum - distance, maximum-likelihood เป็นต้น [12,13,17,18,19,20,21]

3. Structural มีที่มาจากบางปัญหาที่ว่าข้อมูลทางโครงสร้างเป็นส่วนสำคัญในการบอกความแตกต่างของรูปแบบ การบรรยายรูปร่างจะอยู่เทอมของ subpattern หรือเรียกอีกอย่างว่า pattern primitives ซึ่งเป็นคุณสมบัติทางโครงสร้าง โดยมีตัว primitive extractor เป็นตัวกำเนิด pattern primitives ซึ่งจะถูกเรียงเป็น string ในวิธีการนี้แบ่งเป็นวิธีการย่อยได้ 2 วิธี ดังแสดงในรูปที่ 2.2 โดยวิธี Heuristic กฎเกณฑ์การตัดสินใจได้มาจากประสบการณ์และการทดลอง วิธีการนี้มีการใช้อย่างมากมายในเรื่องการจดจำตัวอักษร เนื่องจากเป็นวิธีที่เน้นการประยุกต์ใช้งานอย่างแท้จริง ส่วนวิธี syntactic (หรือ linguistic) อาศัยทฤษฎีทาง formal language สร้างตัว parser มาจัดการกับ string [12,14]

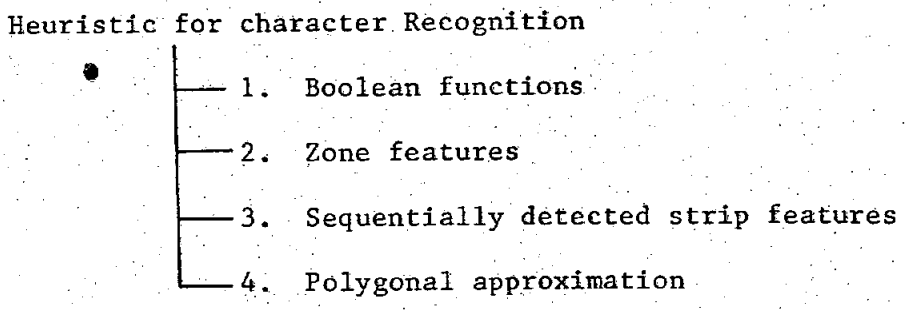
จากที่กล่าวมาแล้วจะเห็นได้ว่า วิธีที่ 1. ไม่เหมาะที่จะนำมาใช้กับการจดจำตัวเลขลายมือเขียน เนื่องจากตัวเลขลายมือเขียนมีความแปรปรวนมาก

ส่วนวิธีที่ 2. มีข้อเสีย เนื่องจากต้องใช้ข้อมูลในการคำนวณมาก ซึ่งไม่เหมาะกับการใช้งาน เนื่องจากระบบ OCR ที่เราจะสร้างจะใช้ไมโครคอมพิวเตอร์ในการโปรเซสข้อมูล ซึ่งตัวไมโครคอมพิวเตอร์จะมีข้อจำกัดในเรื่องความเร็ว ขนาดหน่วยความจำ วิธีการซึ่งมีการใช้ข้อมูลขนาดใหญ่ และมีการคำนวณมากจึงไม่เหมาะสมในการนำมาประยุกต์ใช้ ส่วนวิธี Structural นั้น มีข้อดีคือ มีการคำนวณน้อย ใช้โครงสร้างรูปแบบในการตัดสินใจซึ่งเหมาะสมกับการวิจัยมาก และในทางปฏิบัติจริง ๆ ระบบ OCR ส่วนใหญ่จะเลือกใช้วิธีการนี้

ในวิทยานิพนธ์ฉบับนี้จะเลือกวิธี Heuristic เนื่องจากวิธี Syntactic การบรรยายลักษณะต้องอาศัย การสร้าง grammar ซึ่งทำให้ขบวนการจำแนกทำได้ช้าขึ้น แต่อย่างไรก็ตามถ้ามีคอมพิวเตอร์ขนาดใหญ่ขึ้น ความเร็วสูงขึ้น วิธีการนี้จะเหมาะสมในการใช้เนื่องจากเปอร์เซนต์ความถูกต้องสูงกว่าวิธี Heuristic (4)

2.3 วิธี Heuristic

วิธี Heuristic เป็นวิธีที่ได้มาจากประสบการณ์ที่ได้จากการทดลอง ดังนั้นทฤษฎีตายตัวจึงไม่มี แต่จากการศึกษาเฉพาะส่วนที่ใช้กับการจำแนกตัวอักษรพอจะยกตัวอย่างแสดงได้ในรูปที่ 2.3



รูปที่ 2.3 แสดงตัวอย่างวิธีการจำแนกตัวอักษรของวิธี Heuristic

2.3.1 Boolean functions (13) การตัดสินใจทำโดยใช้คณิตศาสตร์ตรรกะ (boolean) มาสร้าง boolean discriminant function โดยถ้าพิจารณาอินพุทเป็น

ข้อมูลซึ่งเรียงตามลำดับโดยมีตัวชี้ คือ $j = 1, \ldots, m$ เราจะกำหนดตัวแปร Boolean x_j (o หรือ ๑) เป็นค่าข้อมูลเลขฐานสองที่ตำแหน่ง j จากนั้นจะสร้าง $f_r(x)$ - Boolean discriminant โดยตัว $f_r(x)$ จะอยู่ในเทอมความสัมพันธ์ของตัวแปร Boolean x_j โดย $j = ๑, \ldots, m$ การตัดสินใจขึ้นกับค่า $f_r(x)$ ของ class ที่ r ใดเป็นจริง (โลจิก ๑) เครื่องอ่านตัวอักษรที่สร้างขึ้นโดยใช้วิธีการนี้ได้แก่ IBM 1418, IBM 1275, IBM 1975 ในที่นี้จะได้ยกตัวอย่างที่ใช้ในเครื่อง IBM 1418

เครื่อง IBM 1418 จะทำการอ่านตัวอักษรหลังจากนั้นตัวอักษรจะถูกแปลงเป็นข้อมูลเลขฐานสอง เข้าไปยังตัวชิฟต์รีจีสเตอร์ 17 X 10 บิท ซึ่งภายในตัวชิฟต์รีจิสเตอร์มีตัวชี้ตั้งแต่ตำแหน่ง ๑ ถึง ๑๗๐ ดังแสดงในตำแหน่งในรูปที่ 2.4

ELEMENTS OF SHIFT REGISTER OF IBM 1418

1	18	35	52	69	86	103	120	137	154
2	19								155
3		37							
4			55	72					
5		39				107			
6	23	40			91	108	125		
7	24	41	58	75	92	109			
8	25	42	59	76	93	110	127		
9					94	111			
10			61	79					
11			62						
12									
13									
14									
15									168
16									169
17								153	170

รูปที่ 2.4 แสดงตำแหน่งของซิฟต์รีจีสเตอร์ของ IBM 1418

ในที่นี้ j = 1,..., 170 ตัวอย่าง Boolean discriminant function สำหรับจำแนกกลุ่มของ เลข '4' ได้แก่

$$f_4(x) = \bar{x}_{62} \cdot \bar{x}_{79} \cdot (\bar{x}_5 \vee \bar{x}_{107}) \cdot (x_{92} \vee x_{109} \vee x_{127}) \cdot [(\overline{x_{107} \vee x_{125}}) \vee (x_{91} \cdot x_{92})] \cdot (x_{92} \vee x_{93}) \cdot (x_{93} \vee x_{110}) \cdot (x_{58} \vee x_{59}) \cdot (x_{55} \vee x_{72}) \cdot (x_{75} \vee x_{76}) \cdot (x_{41} \vee x_{42}) \cdot (x_{24} \vee x_{25}) \cdot (x_{23} \vee x_{40}) \cdot (\bar{x}_{37} \vee x_7) \cdot [x_{92} \vee x_{93} \vee z_{14} \cdot z_{15}] \cdot \bar{z}_3 \cdot \bar{z}_{16} \cdot \bar{z}_{34} \cdot \bar{z}_{51}$$

Z_i คือ Boolean function ซึ่งเป็น เทอมที่มีใช้ร่วมในกลุ่มอื่น ๆ โดยใน IBM1418 ใช้ Z_i ถึง 50 เทอม ในการอ่านตัวอักษร เพียงแบบ เดียว (single font) และตัวอักขระพิเศษบางตัว

2.3.2 Zone features[13] วิธีการนี้บริษัท NEC ของญี่ปุ่น เป็นผู้คิดค้นโดยการใช้ชิฟต์รีจิสเตอร์ดังที่กล่าวใน 2.3.1 แต่แบ่งออก เป็น 9 โซน มีการ เรียงลำดับดังข้างล่าง

1	2	3
4	5	6
7	8	9

จะมีการสร้างหน้าต่างขนาด 3 X 3 โดยภายใน มีการอ้างอิงตำแหน่งดังข้างล่าง

x_1	x_4	x_7
x_2	x_5	x_8
x_3	x_6	x_9

ค่า x_i (i = 1 ถึง 9) จะ เป็น เลขฐานสอง

จากนั้นจะมีการสร้าง boolean functions

$$V = x_4.x_5.x_6(\bar{x}_2 V \bar{x}_8) \qquad I = x_3.x_5.x_7.(\bar{x}_1 V \bar{x}_9)$$

$$H = x_2.x_5.x_8.(\bar{x}_4 V \bar{x}_6) \qquad D = x_1.x_5.x_9.(\bar{x}_7 V \bar{x}_3)$$

NEC ออกแบบให้ชิฟต์รีจีสเตอร์มีขนาด 15 X 15 บิท เมื่อถูกแบ่งออกเป็น 9 โซน ในแต่ละโซนจะมีขนาด 5 x 5 บิท ภายในแต่ละโซนจะกำหนดให้มีเทอมตัวแปรสี่ตัวคือ T_{iV} , T_{iH} , T_{iI} , T_{iD} โดยที่ T_{iV} จะเป็นจริงก็ต่อเมื่อภายในโซน i เมื่อสแกนด้วยหน้าต่างขนาด 3 X 3 บิท มีเทอม V ดังที่กล่าวในข้างบนเป็นจริงอย่างน้อย 4 ตำแหน่ง เทอม T_{iH} , T_{iI} , T_{iD} ก็คิดเช่นเดียวกันดังที่กล่าวมา ข้อมูล 15 X 15 บิท จะถูกเปลี่ยนเป็นรูปแบบฐานสอง36 รูปแบบ (9 โซนในแต่ละโซนมี 4 T's) หลังจากนั้นจะมีการสร้าง boolean discriminant function คล้ายกับในหัวข้อที่ 2.3.1 ตัวอย่างสำหรับเลข '3' ดังนี้

$$f_3(X) = \bar{T}_{1D}.\bar{T}_{1V}.\bar{T}_{1I}.T_{1H}.T_{2H}.T_{3H}.\bar{T}_{4D}.\bar{T}_{4V}.\bar{T}_{4I}.\bar{T}_{4H}$$

$$.\bar{T}_{5D}.\bar{T}_{5V}.\bar{T}_{5I}.T_{5H}.\ (T_{6D} V\ T_{6V} V\ T_{6I} V\ T_{6H})$$

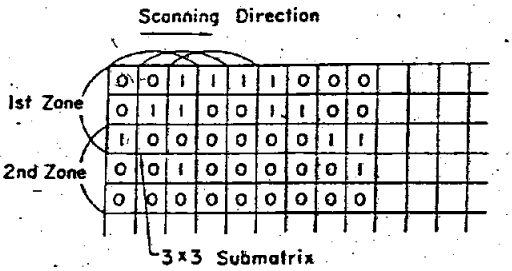
$$.\bar{T}_{7D}.\bar{T}_{7V}.\bar{T}_{7H}.\bar{T}_{8D}.\bar{T}_{8V}.T_{8H}.\bar{T}_{9D}.\bar{T}_{9H}$$

ถ้าข้อมูลอินพุทมีค่า $f_3(x)$ = True มันจะถูกจำแนกว่าเป็นข้อมูลตัวเลข '3'

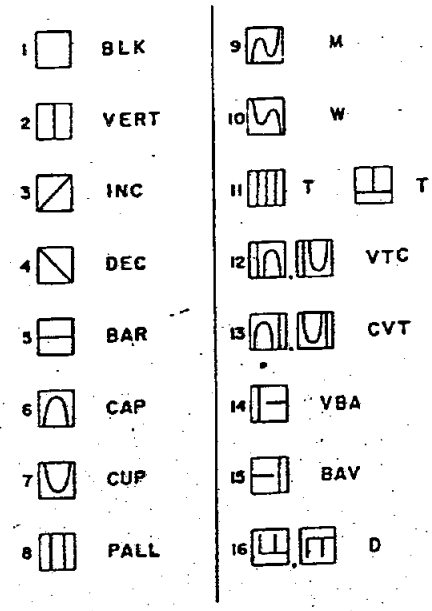
2.3.3 Sequentially detected strip features[1] ถูกพัฒนาเพื่อใช้จำแนกตัวเลขอารบิกให้แก่งานไปรษณีย์ของญี่ปุ่น โดยบริษัท Toshiba วิธีการคล้ายคลึงกับหัวข้อ 2.3.2 โดยการพิจารณาข้อมูลเป็นตาราง n X m จากนั้นสแกนข้อมูลด้วยหน้าต่างขนาด 3 X 3 กำเนิดรหัสซึ่งเรียกว่า " stroke segment direction ซึ่งมี 7 แบบดังแสดงในรูป 2.5 ซึ่งได้แก่ ช่องว่าง (blank) แนวตั้ง (vertical) แนวเอียงขึ้น (incrementally slanted) แนวเอียงลง (decrementally slanted) แนวนอนสูง (high horizontal) แนวนอนต่ำ (low horizontal) ไม่แน่ใจ (undecided) การสแกนสร้างรหัสนี้จะสแกนเลื่อนไปทางขวาที่ละบิทและเมื่อหมดเส้นจะสแกนลงที่ละ 2 บิท ดังในรูป 2.6 ในแต่ละแนวที่สแกนจะได้รหัส stroke segment direction

1 ชุด ซึ่งเราจะนำมารวมสร้างเป็นรหัสขึ้นใหม่แทนลักษณะทางแนวนอน (horizontal features) ซึ่งมีได้ 16 แบบ ดังแสดงในรูปที่ 2.7 เมื่อสแกนครบทุกตำแหน่ง เราจะได้รหัสทางแนวนอนเป็นลำดับ (sequence) จากบนลงล่าง ดังแสดงตัวอย่างการสแกนตัวเลข ในรูปที่ 2.8 การตัดสินใจทำได้โดยสร้าง แผนภูมิทรานซิชัน (transition diagram) สำหรับตัวเลขแต่ละตัว (0 - 9) ขึ้นเพื่อทดสอบรูปแบบของรหัสซีเควนทางแนวนอน ตัวอย่างทรานซิชันสำหรับเลข '8' แสดงในรูปที่ 2.9

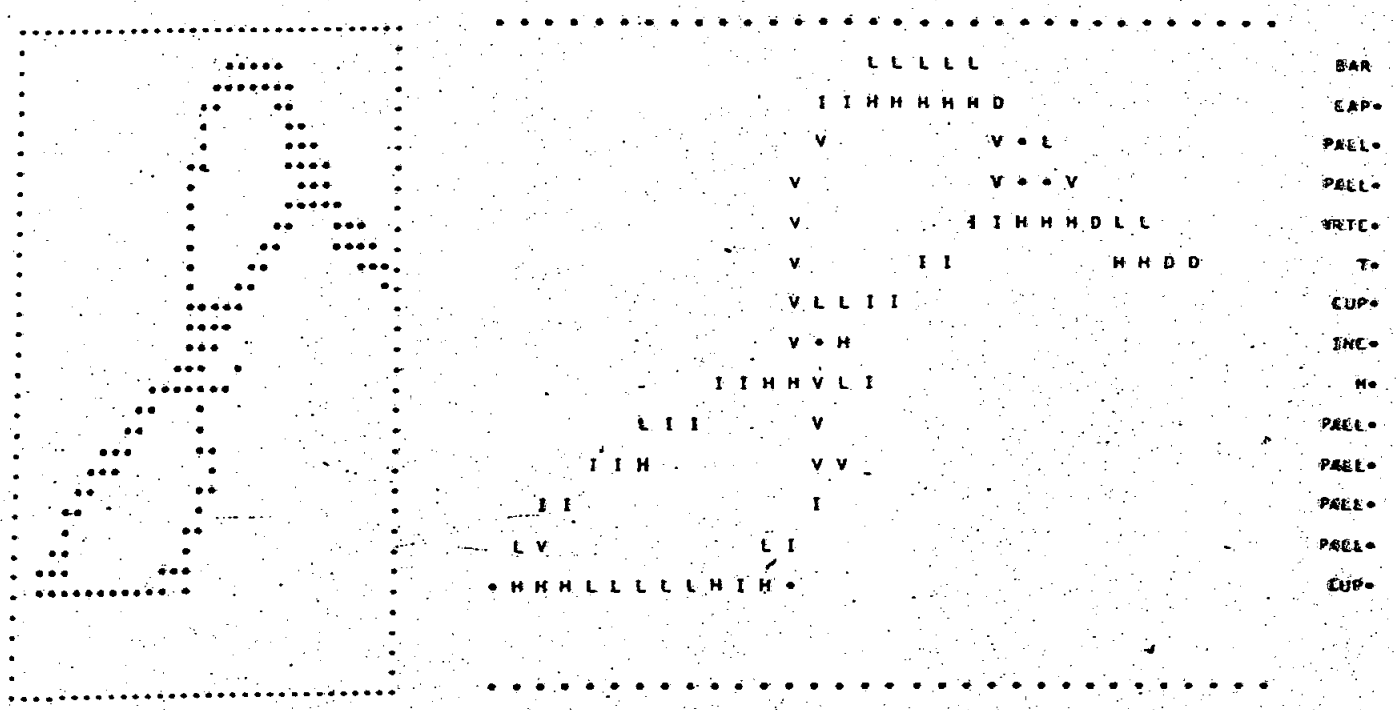
รูปที่ 2.5 แสดงรูปแบบของ stroke segment direction



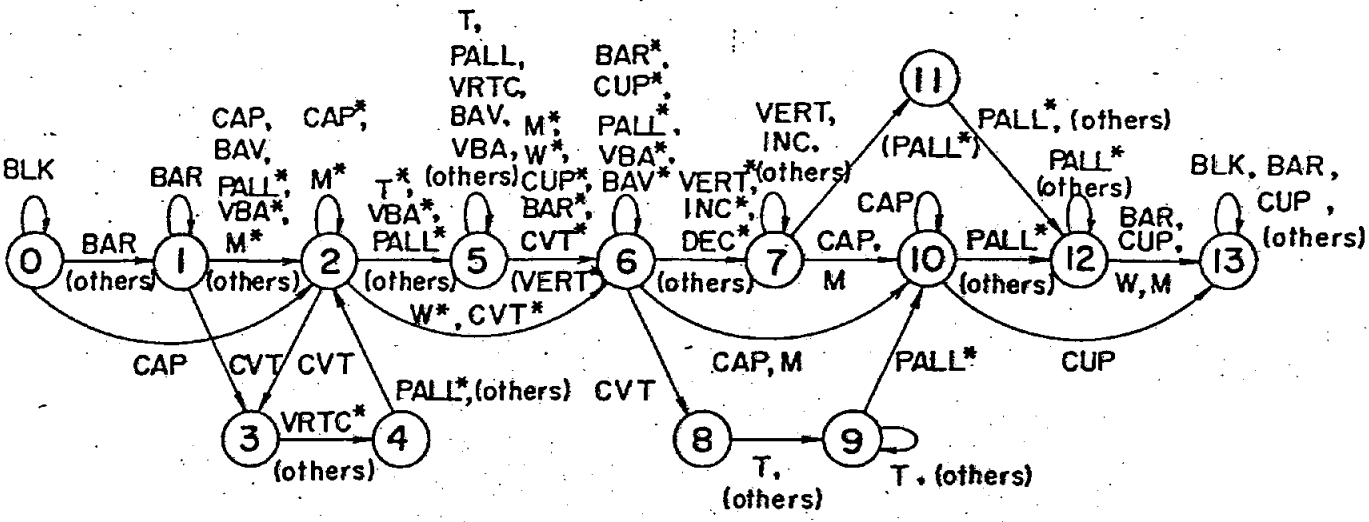
รูปที่ 2.6 แสดงการสแกนของหน้าต่าง 3 X 3



รูปที่ 2.7 แสดงรหัสแทนลักษณะทางแนวนอน



รูปที่ 2.8 แสดงตัวอย่างรหัสที่ได้จากการสแกนตัวเลข '8'



รูปที่ 2.9 แสดงตัวอย่าง แผนภูมิทรานซิชันสำหรับเลข 8

2.3.4 Polygonal approximation[3] วิธีนี้แบ่งได้เป็น 2 ขั้นตอน

1. ใช้วิธี shape analysis หาจุดกรอบนอกของตัวเลขโดยขบวนการกวาด ระหว่างการกวาดจะหาตำแหน่งรู (hole) ภายในตัวเลขเพื่อใช้เป็นลักษณะสำคัญในการตัดสินใจด้วยหลังจากนั้นจะใช้ piecewise linear approximation ทำการประมาณกรอบนอกตัวเลขให้เป็นรูปหลายเหลี่ยม (polygon)

2. ทำการกวาดกรอบนอกอีกครั้งหนึ่ง เพื่อหาตำแหน่งส่วนโค้งส่วนเว้าเพื่อใช้เป็นลักษณะสำคัญ ตัวอย่างลักษณะสำคัญที่ใช้ เช่น

U_1 - จำนวนรู (หาในขั้นตอนที่ 1)

U_2 - ตำแหน่งของรู โดยที่

$U_2 = 1$ ถ้าเป็นรูที่อยู่ด้านล่าง

$U_2 = 2$ ถ้าเป็นรูที่อยู่ด้านบน

$U_2 = 0$ เป็นรูที่อยู่ตรงกลาง

หลังจากที่ได้ลักษณะสำคัญแล้วจะทำการตัดสินใจโดยการใช้ decision tree ที่สร้างขึ้นจากการศึกษาความเป็นไปได้ของตัวเลขแต่ละตัวกับรูปแบบลักษณะสำคัญที่เกิดขึ้น โดยที่ปลายของ tree จะบอกได้ว่าตัวเลขอะไร

วิธีการทั้ง 4 ชนิดที่กล่าวมา สามารถสรุปเปรียบเทียบข้อดีข้อเสียได้ดังตารางที่ 2.1

013397

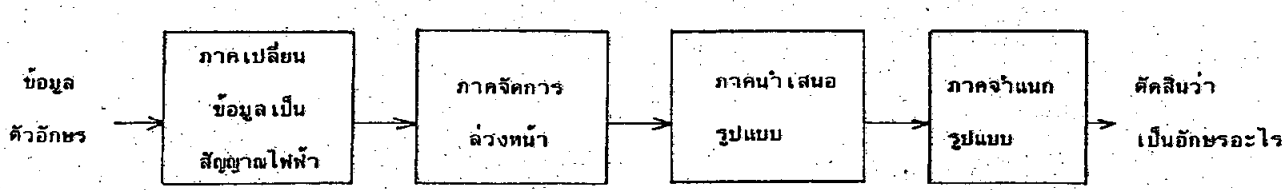
ตารางที่ 2.1 เปรียบเทียบข้อดีข้อเสียวิธีต่าง ๆ ในกลุ่ม Heuristic approach

	วิธีที่ 1	วิธีที่ 2	วิธีที่ 3	วิธีที่ 4
1. ผลกระทบจากความแปรปรวนของตัวเลข	มาก	ปานกลาง	น้อย	น้อย
2. ขั้นตอนในการหาลักษณะสำคัญ	น้อย	ปานกลาง	มาก	ปานกลาง
3. รูปแบบฟังก์ชันในการตัดสินใจ	ง่าย	ง่าย	ยาก	ง่าย
4. จำนวนลักษณะสำคัญ	มาก	มาก	ปานกลาง	น้อย
5. การเลือกลักษณะสำคัญ	ยาก	ยาก	ยาก	ง่าย

จากข้อเปรียบเทียบในตารางข้างบนจะเห็นว่าเราควรเลือกใช้วิธี polygonal approximation ในการสร้างระบบจดจำตัวอักษร เนื่องจากการเลือกลักษณะสำคัญทำได้ง่าย รูปแบบการตัดสินใจซึ่งใช้ tree decision ก็สร้างได้ง่าย

2.4 โครงสร้างของระบบจดจำตัวอักษร

ระบบจดจำตัวอักษรแบ่งแยกเป็น 4 ส่วนใหญ่ดังแสดงในรูปที่ 2.10



รูปที่ 2.10 แสดงโครงสร้างระบบจดจำตัวอักษร

ได้แก่

1. ภาคเปลี่ยนข้อมูลเป็นสัญญาณไฟฟ้า (data conversion)
2. ภาคจัดการล่วงหน้า (preprocessing)
3. ภาคนำเสนอรูปแบบ (pattern representation)
4. ภาคจำแนกรูปแบบ (classifier)

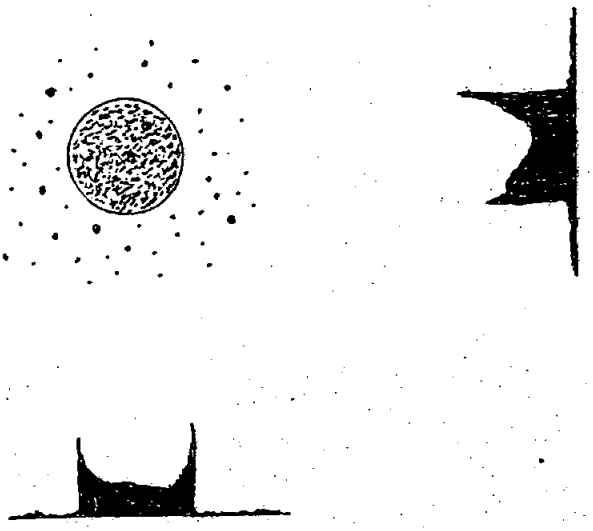
2.4.1 ภาคเปลี่ยนข้อมูลเป็นสัญญาณไฟฟ้า

เป็นขบวนการเปลี่ยนข้อมูลตัวอักษรซึ่งเขียนหรือพิมพ์ในกระดาษให้เป็นสัญญาณไฟฟ้า เพื่อส่งไปประมวลในขบวนการต่อไปโดยอาศัยตัว sensor , scanner , digitizer เป็นต้น รายละเอียดในส่วนนี้จะได้กล่าวในบทที่ 3

2.4.2 ภาคจัดการล่วงหน้า

เป็นขบวนการเตรียมข้อมูลให้พร้อมก่อนที่จะส่งต่อไปยังภาคถัดไป ซึ่งรวมขบวนการหาขอบเขต (boundary decision) การกำจัดสัญญาณรบกวน (filtering) การหาขอบตัวอักษร (edge detection) การทำให้บาง (thining)

ก. การหาขอบเขต เป็นการแยกตัวอักษรออกจากพื้นภาพ (background) โดยมีวิธีง่าย ๆ คือ การสแกนทั้งแนวนอนและแนวตั้ง เมื่อพบข้อมูลที่ตำแหน่งแถว,คอลัมน์ใด เราก็จะรู้ขอบเขตของตัวอักษรที่จะนำไปประมวลต่อไป แต่วิธีนี้จะใช้ได้ดีเฉพาะพื้นภาพกับกับตัวอักษรมีระดับสัญญาณต่างกัน (0 กับ 1) ในกรณีพื้นภาพมีสัญญาณรบกวน (noise) เราจะทำการโปรเจคค่าระดับสีเทา (gray leavel) ลงบนแกน x และ y[12]



หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

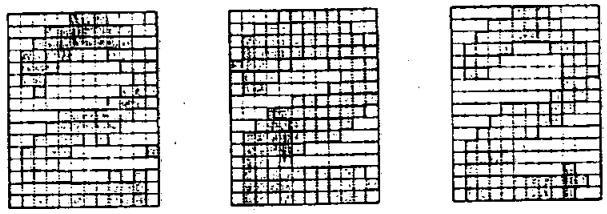
รูปที่ 2.11 แสดงการโปรเจคลงบนแกน x และ y

ถ้ากำหนดให้ $f(x,y)$ เป็นระดับสีเทาที่จุด x , y ซึ่งจะทำการโปรเจคในขอบเขต R การโปรเจคของ f บนแกน x และ y จะได้เป็น $\int_R f(x,y)dx$ และ $\int_R f(x,y)dy$ ซึ่งจะเห็นจากรูปที่ 2.11 ว่าตรงตำแหน่งวัตถุจะมีระดับสีเทาสูงกว่าพื้นภาพ ทำให้เราสามารถหาขอบเขตจากตำแหน่งซึ่งมีระดับสีเทาสูงได้ การวิจัยนี้จะใช้ทั้งสองวิธีร่วมกัน

ข. การกำจัดสัญญาณรบกวน เป็นขบวนการกำจัดข้อมูลแปลกปลอมต่าง ๆ ที่ปะปนมากับตัวเลขที่อ่าน เช่น รอยเปื้อนของหมึกหรือรอยด่างดำบนกระดาษ สัญญาณรบกวนที่เกิดจากตัว sensor ต่าง ๆ ในที่นี้จะได้แนะนำ 2 วิธี คือ

1. วิธี Dineen[13] โดยการสร้างหน้าต่างขนาด n x n วิ่งไปทุก ๆ ตำแหน่งของข้อมูลไบนารี ซึ่งในแต่ละตำแหน่งจุดดำภายในหน้าต่างจะถูกนับและจะมีการสร้างข้อมูลไบนารีชุดใหม่ ซึ่งมีตำแหน่งสัมพันธ์กับข้อมูลชุดเก่า โดยมีเงื่อนไขว่าข้อมูลชุดใหม่จะถูกกำหนดเป็นจุดดำถ้าจำนวนจุดดำในหน้าต่างมากกว่าหรือเท่ากับค่า θ ที่กำหนดค่าหนึ่ง

ตัวอย่างเช่น ถ้า $n = \theta = 3$ จากข้อมูลชุดเก่าในรูปที่ 2.12 (ก) จะได้ข้อมูลชุดใหม่ 2.12 (ข) และ (ค)



รูปที่ 2.12 แสดงผลการลดสัญญาณรบกวนโดยวิธี Dineen (ก) รูปต้นแบบ (ข) และ (ค) เป็นผลเมื่อ $\theta = 3$ และ 5 ตามลำดับ

2. วิธีขยายและหด (expansion and shrinking)[14] ซึ่งเป็นวิธีที่แพร่หลายอีกวิธีหนึ่ง

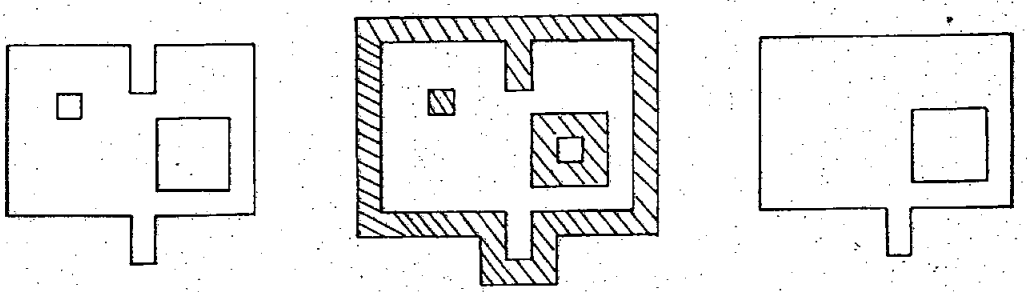
ถ้ากำหนดให้ OB(R) - จุดภายนอกของภาพ R

IB(R) - จุดภายในของภาพ R

เราจะนิยามได้ว่า

การขยาย $E(R) = R \cup OB(R)$ ($\cup$ = Union)

การหด $S(R) = R - IB(R)$ ($-$ = set subtraction)



รูปที่ 2.13 แสดงผลจากขบวนการขยายและหด

รูปที่ 2.13 แสดงผลของการลดสัญญาณรบกวนด้วยขบวนการขยายและหด โดยในรูป (ก) แสดงข้อมูลต้นแบบ รูป (ข) แสดงผลหลังจากขบวนการขยาย และในรูป (ค) แสดงผลหลังจากขบวนการหดในรูป (ข) โดยพื้นที่แรเงาในรูป (ข) แสดงเซตของ OB(R) จะเห็นได้ว่ารูบางส่วนจะถูกกำจัดไปพร้อมกับส่วนเว้าบางส่วน

ในวิทยานิพนธ์นี้จะเลือกการกำจัดสัญญาณรบกวนแบบที่ 1. เนื่องจากขบวนการของ Dineen สามารถกำจัดสัญญาณรบกวนที่เป็นจุดดำบนพื้น และยังสามารถกำจัดจุดขาวที่เกิดขึ้นในเนื้อตัวเลขได้ในขบวนการเดียวกัน ส่วนวิธีหดและขยายจะสามารถกำจัดจุดขาวในเนื้อวัตถุได้ แต่จะไม่สามารถกำจัดจุดดำบนพื้นผิวได้ ซึ่งถ้าพื้นผิวมีจุดดำเกิดขึ้นจะทำให้ขบวนการโปรเซสในขั้นตอนต่อไป เกิดข้อผิดพลาดได้

ค. การหาขอบตัวอักษร เป็นขบวนการหาขอบรูปร่างของตัวอักษรเพื่อใช้เป็นข้อมูลบอกความแตกต่างระหว่างตัวอักษรต่างแบบกัน ขบวนการนี้จะถูกใช้ต่อเมื่อเราเลือกวิธีจดจำแบบ structural ซึ่งได้กล่าวมาแล้วในหัวข้อ 2.2 การหาขอบตัวอักษรมีที่นิยมใช้อยู่ 2 วิธี

1. ค่าเกรเดียนต์โดยการใช้ค่าเกรเดียนต์เป็นตัวบอกว่าตำแหน่งใดในภาพมีขอบ

ถ้าให้ $g(x,y)$ เป็น ค่าเกรเดียนต์ที่ตำแหน่ง x , y

จะได้ว่า

$$g(x,y) = \left|f(x,y) - f(x-1,y)\right| + \left|f(x,y) - f(x,y-1)\right|$$

และถ้ากำหนดให้ BEP เป็น ขอบของภาพข้อมูลเลขฐานสอง (binary edge picture)

จะได้ว่า

$$BEP(x,y) = \begin{cases} 1 \text{ ถ้า } g(x,y) > \text{ threshold} \\ 0 \quad \text{กรณีอื่น ๆ} \end{cases}$$

หรืออาจเลือกใช้สูตรเกรเดียนต์ที่ซับซ้อนขึ้นแต่สามารถบรรยายขอบได้ชัดขึ้น เช่น Sobel's gradient โดยการพิจารณาหน้าต่างขนาด 3 x 3 ดังแสดงในรูปที่ 2.14

a	b	c
d	e	f
g	h	i

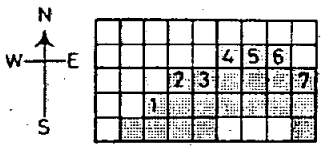
รูปที่ 2.14 แสดงสัญญลักษณ์ของจุดข้างเคียงที่ใช้ใน Sobel's gradient

จะได้ว่า

$$g(e) = \left|\left[f(a) + 2f(b) + f(c)\right] - \left[f(g) + 2f(h) + f(i)\right]\right| + \left|\left[f(a) + 2f(d) + f(g)\right] - \left[f(c) + 2f(f) + f(i)\right]\right|$$

โดย g(e) เป็นค่าเกรเดียนต์ที่ตำแหน่ง e สำหรับข้อมูลเลขฐานสองค่าเกรเดียนต์เท่ากับ 6 จะเป็นตำแหน่งมุม (corner) หรือขอบในแนวทะแยง (diagonal edge) สำหรับขอบแนวนอนและแนวตั้งจะมีค่าเป็น 4

2. การกวาดคอนทัวร์[13] (contour tracing) เป็นการหาขอบตัวอักษรโดยการกวาดตามขอบของตัวอักษรซึ่งข้อมูลได้ถูกแปลงเป็นเลขฐานสองแล้ว การทำงานเริ่มจากตำแหน่งบนสุดของตัวอักษร โดยจะทำการกวาดหาทิศทางซึ่งพบข้อมูลขาวและดำติดกันตามลำดับ ตรงตำแหน่งนั้นจะเป็นจุดขอบ หลังจากนั้นจุดเริ่มกวาดถัดไปจะเริ่มจากจุดขอบซึ่งหาได้จากกรรมวิธีที่ผ่านมาทิศทางการกวาดอยู่ในรูปแบบคล้าย chain code[25] ดูตัวอย่างการกวาดคอนทัวร์ในรูปที่ 2.15



รูปที่ 2.15 แสดงการหาขอบโดยวิธีคอนทัว

จะเห็นว่าการทำงานจะเริ่มจากตำแหน่งที่ 4 กวาดตามเข็มนาฬิกาจะเจอจุดขาวอยู่ในทิศทาง 0 และพบจุดดำอยู่ในทิศทาง 1 เราก็จะกำหนดให้จุดตำแหน่ง 5 เป็นจุดขอบ เนื่องจากมีทิศทางข้อมูลขาวติดกับดำ ตำแหน่งใหม่การกวาดจะไปสู่ตำแหน่งที่ 5 หลังจากนั้น จะทำการกวาดต่อไปเลยจนวนกลับมาครบจุดเดิม คือ ตำแหน่ง 4 เป็นอันเสร็จวิธีการ

ในวิทยานิพนธ์นี้จะเลือกใช้ร่วมกันทั้ง 2 วิธี เนื่องจากวิธีแรก สามารถแสดงตำแหน่งรูของตัวเลขได้ แต่ไม่ได้บอกความสัมพันธ์ของแต่ละจุดของขอบที่ต่อกัน ส่วนวิธีหลังสามารถเก็บข้อมูลขอบเป็นสตริงได้ ซึ่งใช้ในการบรรยายโครงสร้างของตัวเลข แต่วิธีกวาดคอนทัวร์ไม่สามารถบอกได้ว่าตัวเลขมีรู ซึ่งในการจำแนกตัวเลขนั้น รูเป็นสิ่งที่ช่วยบอกความแตกต่างของตัวเลขได้ด้วย

ง. การทำให้บาง ในบางครั้งมีความต้องการที่จะให้ตัวเลขหรือตัวอักษรมีขนาดข้อมูลลดลง แต่ยังมีโครงสร้างเหมือนเดิม เป็นการหาโครงสร้างหลักคล้าย

มักจะใช้ในการนำเสนอรูปแบบเป็นเส้นหรือสโตก[1] (stroke) วิธีที่ใช้โดยทั่ว ๆ ไป มี 2 วิธี

1. ใช้ขบวนการหด โดยการทำซ้ำ ๆ จนกว่าขนาดของตัวอักษรมีความหนา 1 ถึง 2 จุด

2. วิธี sherman โดยอาศัยหลักการของวิธีกวาดคอนทัวร์เพียงแต่จุดกึ่งกลาง ที่ทำการกวาดจะถูก เปลี่ยนจากจุดดำเป็นขาว ยกเว้นแต่จะเกิดสภาวะดังนี้ขึ้น

- จุดกึ่งกลางเชื่อมต่อกับจุดดำข้างเคียง 2 จุด ซึ่งไม่ต่อกันเลยดังแสดงในรูปที่ 2.16 (ก)

- จุดกึ่งกลางมีจุดดำข้างเคียงเพียงจุดเดียวดังแสดงรูปที่ 2.16 (ข)

ก ข

รูปที่ 2.16 แสดงจุดข้างเคียงที่ได้รับการยกเว้นในวิธีของ Sherman

ในวิทยานิพนธ์นี้ วิธีทำให้บางไม่มีความจำเป็นต้องใช้ เนื่องจากขบวนการจำแนกด้วยวิธี polygonal approximation ที่ได้เลือกแล้วนั้น ไม่ได้ใช้สโตก หรือเส้น เป็นคุณสมบัติในการบรรยายตัวเลข แต่จะใช้ตำแหน่งรูและส่วนเว้าเป็นคุณสมบัติสำคัญในการตัดสินใจ

2.4.3 ภาคนำเสนอรูปแบบ

มีหน้าที่แยกส่วนลักษณะที่สำคัญออกจากข้อมูลดิบ เพื่อส่งไปยังภาคจำแนกต่อไป เนื่องจากในวิทยานิพนธ์นี้เลือกวิธี polygonal approximation ดังนั้น การนำเสนอรูปแบบในเทอมของฟีเชอะ ซึ่งจะเป็นการทำงานในภาคนี้แบ่งออกเป็น 2 ขั้นตอน คือ

ก. เกลาเส้นโค้ง (curve fitting) จุดประสงค์เพื่อลดสัญญาณรบกวนบริเวณขอบของตัวเลข โดยใช้วิธี split and merge[25] ซึ่งจะใช้สมการเส้นตรงในการเกลาขอบตัวเลขให้ได้เป็นรูปหลายเหลี่ยม

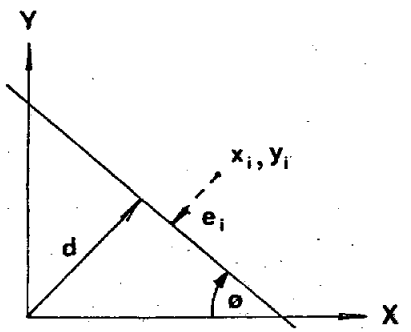
ถ้าให้สมการตรง $x\sin\theta + y\cos\theta = d$ (1)

และ (x_i, y_i) เป็น จุดที่ต้องการประมาณค่า

เราจะสามารถหาค่าผิดพลาด (error) ที่จุด (x_i, y_i) ในการแทนด้วยเส้นตรงในสมการ (1) ได้ดังนี้

$$e_i = | x_i \sin\phi + y_i \cos\phi - d | \qquad (2)$$

ซึ่งระยะ e_i แสดงให้เห็นได้ในรูปที่ 2.17



หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 2.17 แสดงพารามิเตอร์ที่ใช้ในสมการ (2)

ให้ E_2 เป็น ผลรวมกำลังสองของค่าผิดพลาด (integral square error) จะได้ว่า

$$E_2 = \sum_{i=1}^{N} e^2_i \qquad (3)$$

โดย N เป็นจำนวนจุดที่ใช้ในการประมาณ การประมาณด้วยเส้นตรงจะใช้กฎเกณฑ์ผลรวมกำลังสองน้อยที่สุดเมื่อเลือก ø และ d เป็นตัวแปรจะได้ว่า

ผลรวมค่าผิดพลาดจะน้อยที่สุดถ้าเลือก ø และ d ตามสมการข้างล่างนี้ (14)

$$\sin 2\phi (V_{xx} - V_{yy}) + 2 \cos 2\phi V_{xy} = 0 \qquad (4)$$

$$d = \sin\phi . V_x + \cos\phi . V_y \qquad (5)$$

โดย

$$V_x = \frac{1}{N} \sum_{i=1}^{N} x_i \qquad (6)$$

$$V_y = \frac{1}{N} \sum_{i=1}^{N} y_i \qquad (7)$$

$$V_{xx} = \sum_{i=1}^{N} (x_i - V_x)^2 \qquad (8)$$

$$V_{yy} = \sum_{i=1}^{N} (y_i - V_y)^2 \qquad (9)$$

$$V_{xy} = \sum_{i=1}^{N} (x_i - V_x)(y_i - V_y) \qquad (10)$$

$$E_2 = \sin^2 \phi . V_{xx} + \cos^2 \phi . V_{yy} + \sin 2\phi . V_{xy} \qquad (11)$$

ขั้นตอนการทำ Split and merge มีดังนี้

กำหนด $i = 1, 2, \ldots, N$ เป็นจำนวนเซกเมนต์เริ่มต้นของขอบตัวเลข

E_{max} เป็นค่าผิดพลาดสูงสุดที่ยอมรับได้

ขั้นที่ 1 ทำการ split เซกเมนต์ที่มี $E_i > E_{max}$ โดยการแบ่งครึ่งเซกเมนต์ที่ i ออกเป็นสองเซกเมนต์ใหม่ ในขั้นตอนนี้ จำนวนเซกเมนต์จะเพิ่มขึ้น

ขั้นที่ 2 จะทำการ merge เซกเมนต์ i กับ i + 1 ถ้าเซกเมนต์ i_n ใหม่ที่เกิดจากการต่อเซกเมนต์ i กับ i + 1 มี E_{i_n} น้อยที่สุด เมื่อเทียบกับการ merge ของเซกเมนต์อื่น ๆ และ E_{i_n} ที่ได้จะต้องน้อยกว่าหรือเท่ากับ E_{max} การ merge จะดำเนินต่อไปจนไม่สามารถกำเนิดเซกเมนต์ใหม่ที่มี $E_{i_n} \leqslant E_{max}$ ได้

ข. การสร้างลักษณะสำคัญ (feature generation) จะทำการกวาดตามขอบตัวเลขเพื่อหาส่วนเว้า โดยถ้ามุมระหว่างเซกเมนต์น้อยกว่า 180 องศา จะถือว่าตรงตำแหน่งมุมนั้นเป็นมุมเว้า นำคุณสมบัติอันนี้มากำเนิด ลักษณะสำคัญซึ่งรายละเอียดจะกล่าวในบทที่ 4

2.4.4 ภาคจำแนกรูปแบบ (classifier) จะเอาลักษณะสำคัญพื้นฐานที่รู้แล้ว (priori) ในการบรรยายตัวเลข มาสร้าง decision tree ซึ่งใช้เป็นตัวจำแนกรูปแบบของตัวเลข รายละเอียดจะกล่าวในบทที่ 4