การสร้างโมเดลและการจำแนกแบบฟอร์มสำเร็จรูป

Model Construction and Classification of Preprinted Form

ยุทธพงษ์ รังสรรค์เสรี ปัญญา ฐิติมัชฌิมา กฤษณะ ชินสาร และ เสกสรรค์ พลศรี
ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ถนนฉลองกรุง เขตลาดกระบัง กรุงเทพฯ 10520
โทร.(02) 326-6100 ต่อ 2705 E-Mail: kryuttha@kmitl.ac.th

บทคัดย่อ

ในระบบประมวลผลแบบฟอร์มที่สามารถรองรับการทำงานกับแบบฟอร์มหลายประเภท จำเป็นที่จะต้องมีความสามารถจำแนกประเภทแบบฟอร์มได้เองโดยอัตโนมัติ และสามารถค้นหาขอบเขตของฟิลด์ข้อมูลได้อย่างถูกต้องจากโมเดลของประเภทแบบฟอร์มที่ค้นพบ บทความนี้จะเสนอขั้นตอนการประมวลผลที่จำเป็นสำหรับภาระงานทั้งสองนี้ เริ่มจากการกำจัดสัญญาณรบกวนในภาพที่สแกนเข้ามาโดยใช้อัล-กอริธึม kFill การแก้ความเอียงของเอกสารอันเนื่องมาจากการสแกน การค้นหาเส้นตรงในภาพ และการคำนวณหาจุดตัดของเส้นตรงเหล่านั้นซึ่งมีรูปแบบที่เป็นไปได้ 9 รูปแบบ ในที่สุดโมเดลของแบบฟอร์มสามารถสร้างขึ้นจากรูปลักษณะการตัดกันของเส้นตรง และการจำแนกประเภทแบบฟอร์มก็ทำได้โดยการเปรียบเทียบเส้นตรงที่ปรากฏในภาพ

ABSTRACT

Automatic form processing systems that support various types of preprinted forms have to be able to automatically identify the type of each input form, and also locate the boundary of all data fields using its corresponding constructed model. This paper will present a procedure necessary for these two tasks. Each input form is scanned and preliminary noise removed using kFill algorithm. The skew correction, straight line extraction, and line crossing detection are then applied successively. The resulting description of straight lines and line-crossing types is useful for the model construction and classification.

1. บทนำ

การประมวลผลแบบฟอร์มเอกสาร หรือการจัดการกับข้อมูลในฟอร์มเอกสารเป็นการทำงานที่มีความจำเป็นสำหรับทุกหน่วยงานทั้งในงานภาครัฐและเอกชน ทั้งนี้เพราะฟอร์มเอกสารเป็นแหล่งที่มาซึ่งสารสนเทศสำหรับการทำงานในองค์กร แต่การประมวลผลแบบฟอร์มในองค์กรทุกวันนี้ก็มีขีดจำกัดเนื่องมาจากการขาดแคลนจำนวนบุคลากร อีกทั้งยังต้องใช้ค่าใช้จ่ายจำนวนมากในการที่จะดึงเอาข้อมูลหรือสารสนเทศออกมาจากแบบฟอร์มนั้นๆ เพราะการที่เราจะเอาข้อมูลจากแบบฟอร์มไปใช้งาน ต้องให้เจ้าหน้าที่อ่านสารสนเทศที่กรอกไว้บนแบบฟอร์ม และคีย์ข้อมูลกลับสู่ระบบคอมพิวเตอร์อีกครั้งจึงจะประมวลผลต่อไปได้ นอกจากจะใช้เจ้าหน้าที่จำนวนมากในการอ่านและคีย์ข้อมูลแล้วยังเสี่ยงต่อการสูญเสียข้อมูลสำคัญ อันเนื่องมาจากความผิดพลาดที่อาจเกิดขึ้นจากการอ่านข้อมูลจากแบบฟอร์ม หรือการคีย์ข้อมูลที่อ่านได้เข้าเครื่องคอมพิวเตอร์โดยการปราศจากการตรวจสอบ ซึ่งถ้าข้อมูลตรงนี้ผิดพลาดย่อมส่งผลให้องค์กรได้รับสารสนเทศที่ผิดไปเช่นกัน การใช้คอมพิวเตอร์ในงานประมวลผลแบบฟอร์มสามารถกำหนดขั้นตอน ดังนี้ 1) การ สแกนภาพเอกสาร 2) การปรับปรุงคุณภาพเอกสาร 3) การจำแนกประเภทแบบฟอร์ม 4) การกำหนดตำแหน่งฟิลด์ข้อมูล และ 5) สร้างฟอร์มไลบรารีจากสารสนเทศแบบฟอร์ม

R. Casey et al. [1] ได้ศึกษาการประมวลผลแบบฟอร์มไว้เป็น 5 ขั้นตอนที่น่าสนใจดังนี้ ขั้นตอนที่ 1) วิเคราะห์หาองค์ประกอบของแบบฟอร์มเปล่า ๆ บนภาพเอกสารโดยใช้การติดต่อระหว่างโปรแกรมกับผู้ใช้ 2) จดจำ

องค์ประกอบของแบบฟอร์มที่ได้จากขั้นตอนที่ 1 เป็นฟอร์มไลบรารี 3) ดึงข้อมูลภาพจากอินพุตฟอร์มตามตำแหน่งที่กำหนด 4) เปลี่ยนข้อมูลภาพในขั้นตอนที่ 3 เป็นสัญลักษณ์ที่คอมพิวเตอร์เข้าใจ และ 5) เขาได้สร้างโมดูลสำหรับขจัดส่วนของแบบฟอร์มที่ติดมากับข้อมูลภาพเพื่อจะให้เหลือเฉพาะตัวข้อมูลจริงๆ วิธีที่เขานำเสนอจัดได้ว่าเป็นงานวิจัยที่เหมาะจะใช้เป็นพื้นฐานในการศึกษาและมีการอ้างถึงเสมอในงานวิจัยทางด้านการประมวลผลแบบฟอร์ม

S.L. Taylor et al. [2] ศึกษาการประมวลผลแบบฟอร์มโดยประยุกต์ใช้สำหรับงานประมวลผลแบบฟอร์มเสียภาษีของกรมสรรพากรประเทศสหรัฐอเมริกา ในรายงานวิจัยของเขาได้อธิบายเทคนิคในการจัดการกับข้อมูลภาพให้มีประสิทธิภาพรวมทั้งขั้นตอนในการแปลงข้อมูลภาพนั้น

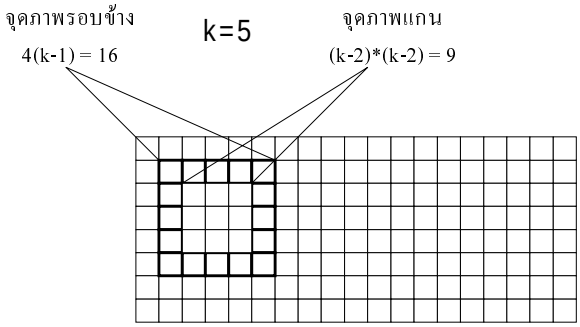
L.O'Gorman และ R.Kasturi [3] กล่าวถึงหลักการพื้นฐานของการโปรเจคชั่น เขาแนะนำการโปรเจคชั่นในแนวตั้งเพื่อสร้างฮิสโตแกรมสำหรับแบ่งภาพเอกสารออกเป็นคอลัมน์ และการโปรเจคชั่นในแนวนอนเพื่อสร้างฮิสโตแกรมสำหรับแบ่งภาพเอกสารออกเป็นบรรทัดของตัวหนังสือ (Line of Characters)

2. การปรับปรุงคุณภาพเอกสาร

2.1 การกำจัดสัญญาณรบกวน

การกำจัดสัญญาณรบกวนเป็นกระบวนการที่สำคัญในการประมวลผลข้อมูลภาพ ทั้งนี้เพราะการจะสร้างต้นแบบหรือโมเดลเพื่อการประมวลผลจำเป็นที่เราจะต้องใช้ภาพเอกสารที่มีคุณภาพดี จึงจะทำให้เราได้ต้นแบบที่ดี และมีความสมบูรณ์ สัญญาณรบกวนจะมีอยู่ 2 ลักษณะ คือ กลุ่มของจุดภาพดำปรากฏอยู่บนภาพพื้นหลังขาว และกลุ่มของภาพพื้นหลังขาวปรากฏเป็นหลุมอยู่บนกลุ่มของข้อมูลภาพ ดังนั้นในการกำจัดสัญญาณรบกวนจะต้องพิจารณาทั้งสองกรณีไปพร้อมกัน

อัลกอริธึม kFill [3] เป็นวิธีที่นำเสนอมาเพื่อการกำจัดสัญญาณรบกวนโดยการพิจารณาเติมค่าตำแหน่งของแต่ละกลุ่มจุดภาพที่กำลังสนใจกับกลุ่มจุดภาพรอบข้างโดยให้เป็นไปตามเงื่อนไข โดยพิจารณาทั้ง 2 กรณีของสัญญาณรบกวนไปพร้อมกัน กำหนดหน้าต่าง k x k บนข้อมูลภาพจะพบว่ามีกลุ่มของจุดภาพที่เราเรียกว่า "จุดภาพแกน" (core) เป็นกลุ่มของจุดภาพที่เราสนใจพิจารณาเติมค่า (k-2)x(k-2) จุดภาพ และอีก 4(k-1) เป็นกลุ่มจุดภาพที่อยู่รอบนอกของกลุ่มจุดภาพที่เราสนใจพิจารณาเติมค่าเราจะเรียกว่า "จุดภาพรอบข้าง" (neighborhood)



ภาพที่ 1 จุดภาพที่ใช้ในการขจัดสัญญาณรบกวน

ในขั้นตอนของการพิจารณาเติมค่ากลุ่มจุดภาพที่เป็นกลุ่มจุดภาพแกนของแต่ละกรอบหน้าต่างว่าจะเติมเป็นกลุ่มของจุดภาพดำ (ON) หรือขาว (OFF) จะเป็นการพิจารณากลุ่มของจุดภาพแกนกับเงื่อนไขของกลุ่มของจุดภาพรอบข้าง ดังนี้

การพิจารณาเติมค่ากลุ่มจุดภาพแกนให้เป็นกลุ่มจุดภาพขาว ต้องเป็นไปตามเงื่อนไขดังนี้

1. กลุ่มจุดภาพแกนทั้งหมดต้องเป็นจุดภาพดำ และ
2. กลุ่มจุดภาพรอบข้าง พิจารณาตัวแปร 3 ตัวต่อไปนี้

n : จำนวนของจุดภาพขาว

c : จำนวนกลุ่มจุดภาพขาวที่เรียงต่อเนื่องกัน

r : จำนวนจุดภาพขาวบนมุมทั้ง 4 ของหน้าต่าง

โดยค่าตัวแปรทั้ง 3 ต้องเป็นไปตามสมการ

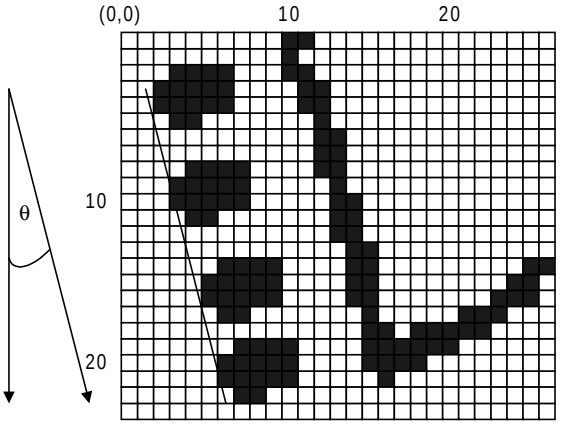
$$(c = 1)\, and\, [(n > 3k - 4)\, or\, ((n = 3k - 4)\, and\, (r = 2))]$$

ถ้าเราพบว่ากลุ่มของจุดแกนและจุดภาพรอบข้างเป็นไปตามเงื่อนไขทั้ง 2 ก็ให้เติมค่าของจุดภาพแกนเป็นกลุ่มของจุดภาพขาวแต่ถ้าหากว่าไม่เป็นไปตามเงื่อนไขที่กำหนดไว้ก็ให้เติมกลุ่มของจุดภาพแกนเป็นกลุ่มของจุดภาพดำ สำหรับการ

เลือกเติมกลุ่มจุดภาพแกนให้เป็นกลุ่มจุดภาพดำก็ทำในทางตรงกันข้าม การทดลองกำหนดหน้าต่างขนาด 3x3 หรือ k=3

2.2 การประมาณค่าความเอียงของหน้าเอกสาร

การประมาณความเอียงด้วยวิธีการที่นำเสนอในงานวิจัยนี้ จะหาแนวเส้นตรงของตำแหน่งจุดภาพดำทางซ้ายสุดของแต่ละบรรทัดข้อมูลภาพ (Image Line) ดูเพิ่มเติม [4] หาเส้นตรงเพื่อเป็นตัวแทนของตำแหน่งจุดภาพดำเหล่านั้นโดยวิธีกำลังสองน้อยที่สุด (Least Square Method : LSM) [5] จะได้สมการเส้นตรงดังนี้ $Y_i^{'} = aX_i + b$ เมื่อ a=Slope และ b=Y-intercept จากสมการเส้นตรงนำไปประมาณค่ามุมเอียงของหน้าเอกสาร โดย $\theta = \tan^{-1} a$ ดังแสดงในภาพที่ 2



ภาพที่ 2 แนวเอียงของภาพเอกสารเป็นมุม θ ต้องหมุนกลับในทิศตามเข็มนาฬิกา

3. การสร้างโมเดล

3.1 การค้นหาเส้นตรงในภาพ

การสร้างโมเดลของแบบฟอร์มในงานวิจัยต้องการข้อมูลที่สามารถบรรยายถึงตำแหน่งและความยาวของเส้นตรงในแนวนอนและแนวตั้งที่ปรากฏในภาพ การค้นหาเส้นตรงนี้อาศัยการประมวลผล 2 ขั้นตอน ขั้นตอนแรกใช้วิธี Runlength โดยสแกนทีละแถว (หรือทีละคอลัมน์) เพื่อสร้าง List ของเส้นตรงที่เป็นไปได้ทั้งหมด โดยเส้นตรงแต่ละเส้นจะถูกบรรยายโดยพิกัดจุดเริ่มต้น (BegX, BegY) และความยาวของเส้นตรงนั้น (LRun) ภาพที่ 3 แสดงตัวอย่างของภาพเริ่มต้นที่ผ่านการปรับปรุงคุณภาพเอกสารมาแล้ว พร้อมทั้งเส้นตรงในแนวนอนและแนวตั้งทั้งหมดที่คำนวณได้จากการประมวลผลในขั้นแรก พิจารณาเส้นตรงในแนวนอนจะเห็นว่า เส้นตรงที่ 1 และ 2 เป็นเส้นเดียวกัน ทำนองเดียวกับเส้นที่ 3 และ 4 ซึ่งมีตำแหน่งเริ่มต้นในแนวตั้ง (BegY) อยู่ติดกัน

การประมวลผลในขั้นตอนที่สองจะเป็นการทำงานกับข้อมูลที่อยู่ใน List ของเส้นตรงเพื่อรวม (Merge) เส้นตรงที่อยู่ติดกับเข้าด้วยกัน ตามอัลกอริธึมที่ 1 และผลการประยุกต์ใช้กับข้อมูลในภาพที่ 3 แสดงได้ในภาพที่ 4

อัลกอริธึม 1 อัลกอริธึมสำหรับการจัดการเส้นตรงที่อยู่ติดกัน

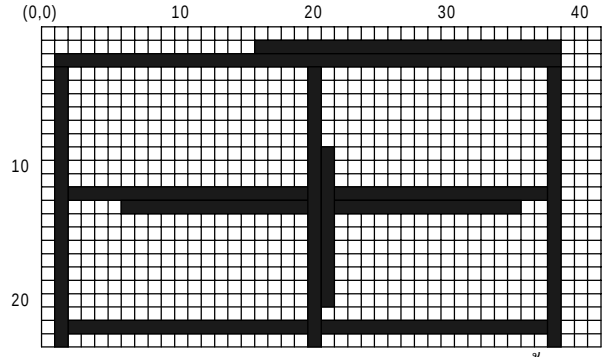
เริ่มต้น

กำหนด q และ walk เป็นตัวชี้โหนดของเส้นตรงโดย q เป็นโหนดนำ

1) พิจารณาเส้นตรงสองเส้นถ้ามีจุดเริ่มในแกน Y เท่ากัน และเส้นตรงสองเส้นมีจุดภาพขาวคั่นกลางไม่เกิน 30 จุด ให้ต่อเส้นตรงสองเส้นเป็นเส้นตรงเดียวกัน โดยจุดเริ่มต้นของเส้นตรงให้อยู่ที่โหนด q ส่วนความยาวก็ให้ยาวถึงตำแหน่งสุดท้ายของเส้นตรงของโหนด walk
2) พิจารณาเส้นตรงสองเส้นมีจุดเริ่มในแกน Y ติดกัน และตำแหน่งสุดท้ายของเส้นตรงที่ 1 ห่างจากจุดเริ่มต้นของเส้นตรงที่ 2 น้อยกว่า 30 จุด พิจารณาต่อไปว่าระหว่างสองโหนดที่ติดกันนั้น โหนดใดมีจุดเริ่มต้นในแนวแกน X น้อยกว่าก็ให้เลือกโหนดนั้นเป็นหลัก ส่วนความยาวนั้นก็ให้ยาวถึงตำแหน่งสุดท้ายที่มากที่สุดระหว่างเส้นตรงสองเส้น
3) วนกลับไปทำงานขั้นตอนที่ 1 จนกว่าจะเปรียบเทียบถึงโหนดสุดท้าย

จบงาน;

สำหรับการเลือกโหนดของเส้นตรงในแนวตั้งก็ทำในลักษณะคล้ายคลึงกันกับการขจัดโหนดในแนวนอน ซึ่งจากการใช้อัลกอริธึมดังกล่าวทำให้เส้นตรงในแกนนอนเหลือเพียงเส้นที่ 1, 3 และ 5 ส่วนเส้นตรงในแนวตั้งก็จะเหลือเพียงเส้นที่ 1, 2 และ 4 เท่านั้น โดยความยาวของเส้นตรงยังคงเท่าเดิม



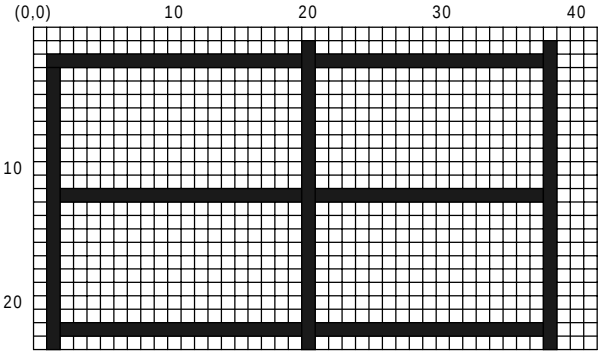
<u>เส้นตรงในแนวนอน</u>

เส้นที่	(BegX	BegY)	
LRun			
1	16	1	22
2	1	2	37
3	1	12	37
4	6	13	29
5	1	22	37

<u>เส้นตรงในแนวตั้ง</u>

เส้นที่	(BegX	BegY)	
LRun			
1	1	2	21
2	20	1	22
3	21	9	11
4	38	1	22

ภาพที่ 3 ภาพเอกสารที่เส้นตรงหนามากกว่า 1 จุดภาพ



<u>เส้นตรงในแนวนอน</u>

เส้นที่	(BegX	BegY)	
LRun			
1	1	2	37
2	1	12	37
3	1	22	37

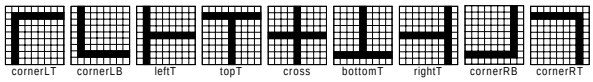
<u>เส้นตรงในแนวตั้ง</u>

เส้นที่	(BegX	BegY)	
LRun			
1	1	2	21
2	20	1	22
3	38	1	22

ภาพที่ 4 ภาพเอกสารหลังการรวมเส้นตรงที่อยู่ติดกัน

3.2 การคำนวณรูปแบบการตัดกันของเส้นตรง

เมื่อเราได้เส้นตรงที่วางตัวอยู่ในสองแนว คือ แนวตั้งและแนวนอน เส้นตรงทั้งสองต้องตัดกันแน่นอน เราจะทำการคำนวณหาตำแหน่งและรูปแบบการตัดกันของเส้นตรงเหล่านั้น ซึ่งการตัดกันของเส้นตรงเราจะนำไปใช้ในการกำหนดตำแหน่งฟิลด์ข้อมูลในการประมวลผลแบบฟอร์ม กำหนดหน้าต่างขนาด 9x9 สำหรับการหาการตัดกันของเส้นตรงสองแนว รูปแบบการตัดกันจะเป็นไปได้ 9 รูปแบบดังภาพที่ 5



ภาพที่ 5 รูปแบบของการตัดกันของเส้นตรง

ถ้าเราพิจารณาแบบตรงไปตรงมาเราจะเห็นว่าในแต่ละตำแหน่งของการตัดกันนั้น จะต้องพิจารณาถึง 9 รูปแบบซึ่งจะทำให้เสียเวลามาก (เพราะในแต่ละรูปแบบใช้หน้าต่างขนาด 9x9 แล้วแต่ละจุดพิจารณาอีก 9 รูปแบบ จำนวนครั้งในการเปรียบเทียบก็จะสูงมาก) ซึ่งถ้าเราพิจารณาให้ดีเราจะคำนวณหาการตัดกันของเส้นตรงเพียง 4 รูปแบบ คือ cornerLT, cornerRT, cornerLB และ cornerRB เมื่อได้ตำแหน่งการตัดกันทั้งหมดของเส้นตรงทั้ง 4 รูปแบบ พิจารณาต่อไปว่า ตำแหน่งจุดตัดใดๆ จะมีรูปแบบการตัดกันในอีก 5 รูปแบบที่เหลือตำแหน่งจุดตัดนั้นต้องมีรูปแบบการตัดกันใน 4 รูปแบบแรกที่กล่าวมามากกว่า 1 ครั้ง หรือกล่าวอีกนัยหนึ่งก็คือ มีโหนดของการตัดกันของเส้นตรงสองเส้นที่มีจุดตัด (X, Y) เดียวกันมากกว่า 1 โหนด โดยแต่ละโหนดนั้นมีรูปแบบของการตัดกันที่แตกต่างกัน โดยเงื่อนไขในการสร้างจุดตัด 5 รูปแบบเหลือจะกำหนดไว้ในอัลกอริธึม 2

อัลกอริธึม 2 เงื่อนไขการตัดกันของเส้นตรง 5 ประเภท

For the same Coordinate (X, Y) Of LineCrossing Position

1) IF (cornerLB and cornerRB) THEN bottomT.
2) IF (cornerLT and cornerRT) THEN topT.
3) IF (cornerRB and cornerRT) THEN rightT.
4) IF (cornerLB and cornerLT) THEN leftT.
5) IF (cornerLB and cornerRT) or (cornerRB and cornerLT) THEN cross.

ตารางที่ 1 ตำแหน่งและรูปแบบการตัดกันของเส้นตรงสองเส้นโดยการหารูปแบบการตัดกันเพียง 4 รูปแบบแรก

Point	X-Position	Y-Position	CornerType
1	1	2	cornerLT
2	20	2	cornerRT
2	20	2	cornerLT
4	38	2	cornerRT
:	:	:	:
7	2	22	cornerLB
8	20	22	cornerRB
8	20	22	cornerLB
9	38	22	cornerRB

ตารางที่ 2 ตำแหน่งและรูปแบบการตัดกันของเส้นตรงสองเส้นเมื่อผ่านความสัมพันธ์ในอัลกอริธึมที่ 2

Point	X-Position	Y-Position	CornerType
1	1	2	cornerLT
2	20	2	topT
3	38	2	cornerRT
4	2	12	leftT
5	20	12	cross
6	38	12	rightT
7	2	22	cornerLB
8	20	22	bottomT
9	38	22	cornerRB

3.3 โมเดลของแบบฟอร์ม

จากตำแหน่ง และรูปแบบการตัดกันของของเส้นตรงในสองแนว นำมาหาความสัมพันธ์เพื่อสร้างฟิลด์ข้อมูลรูปสี่เหลี่ยมผืนผ้าโดยแต่ละฟิลด์ข้อมูลนั้นจะต้องถูกล้อมรอบด้วยเส้นตรงทั้ง 4 ด้าน โดยแต่ละฟิลด์มีข้อมูลที่จำเป็นต้องใช้คือ มุมบนซ้ายของฟิลด์ (TopX, TopY), มุมล่างขวาของฟิลด์ (BottomX, BottomY) และประเภทของฟิลด์ (FieldType) การกำหนดประเภทของฟิลด์ก็เพื่อประโยชน์สูงสุดในการทำงานกับข้อมูลในแต่ละฟิลด์นั้นๆ

ในการสร้างตำแหน่งของฟิลด์รูปสี่เหลี่ยมผืนผ้า ต้องคำนึงเสมอว่าขอบเขตของแต่ละฟิลด์นั้นต้องไม่คาบเกี่ยวกัน (Overlap) กล่าวคือ สมาชิกของฟิลด์สองฟิลด์ที่ติดกันต้องไม่เกิดการอินเตอร์เซคชั่น แต่มันสามารถใช้ตำแหน่งจุดตัดกันเป็นจุดมุมในการสร้างฟิลด์ร่วมกันได้ เช่น จากภาพที่ 4 จุดตัดที่ (20, 12, cross) ซึ่งเป็นมุมล่างขวาของฟิลด์ที่ 1 และในขณะเดียวกันจุดดังกล่าวนั้นก็เป็นจุดมุมบนซ้ายของฟิลด์ที่ 4 ด้วย

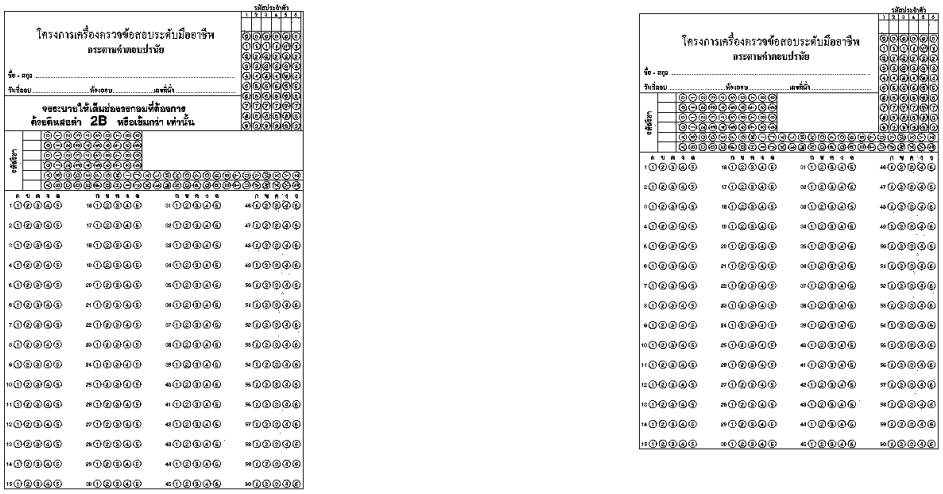
นอกจากการหาตำแหน่งจุดมุมของฟิลด์ข้อมูลแต่ละฟิลด์แล้ว ในการทำงานจริงเรายังต้องเพิ่มการทำงานในส่วนการติดต่อกับผู้ใช้ (User Interface) เพื่อให้ผู้ใช้ช่วยในการกำหนดว่าแต่ละฟิลด์นั้นจะใช้ทำงานอะไร มีข้อมูลที่สำคัญอย่างไร การประมวลผลกับข้อมูลแต่ละฟิลด์จะทำอย่างไร ผู้ใช้มีความจำเป็นต่อระบบการประมวลผลแบบฟอร์มมาก ทั้งนี้เนื่องจากคอมพิวเตอร์ไม่สามารถกำหนดความสำคัญและรูปแบบการประมวลผลของแต่ละฟิลด์ได้ด้วยตัวเอง

4. การจำแนกประเภทแบบฟอร์ม

ในงานวิจัยครั้งนี้ได้เลือกใช้แนวเส้นตรงที่เกิดขึ้นในภาพเอกสาร โดยพารามิเตอร์ที่ต้องการคือ จำนวน ความยาว และระยะห่างระหว่างเส้นตรง ทั้งนี้เพราะเส้นตรงที่ได้จากภาพนั้นแม้จะมีการย้ายตำแหน่งของเส้นตรงไปมากน้อยเพียงใด จำนวน ความยาว และระยะห่างระหว่างเส้นตรงก็ยังเท่ากันเสมอ (เมื่อสแกนด้วยความละเอียดการสแกนเท่ากัน) ซึ่งจากการทดลองกำหนดให้ใช้เส้นตรงในแนวนอนเป็นตัวแปรสำหรับจำแนกประเภทแบบฟอร์ม

5. ผลการทดลอง

อัลกอริธึมสำหรับการสร้างโมเดล และการจำแนกแบบฟอร์มสำเร็จรูปที่นำเสนอในบทความนี้ ได้ถูกนำไปประยุกต์ใช้ในระบบตรวจข้อสอบปรนัยอัตโนมัติ [6] อย่างได้ผล โดยการทดลองจำแนกแบบฟอร์มกระดาษคำตอบข้อสอบปรนัยที่มีโครงสร้าง และส่วนประกอบที่คล้ายคลึงกัน 2 รูปแบบ ดังแสดงในภาพที่ 6 ซึ่งเป็นภาพไบนารี สแกนด้วยความละเอียด 75 จุดต่อนิ้ว การทำงานแบ่งออกเป็น 2 ขั้นตอนดังนี้ ขั้นตอนแรกจะทำการสแกนเข้ามาเพื่อสร้างโมเดลของแบบฟอร์มข้อสอบแต่ละประเภท และ ขั้นตอนที่สองทำการสแกนภาพของกระดาษคำตอบที่มีการเลือกทำข้อสอบในรูปแบบที่แตกต่างกัน และเลือกทำในตำแหน่งข้อสอบต่างๆ กัน โดยใช้จำนวนภาพที่ใช้ในการทดลองรูปแบบละ 30 ภาพ ซึ่งผลการทดลองปรากฏว่าสามารถจำแนกประเภทของแบบฟอร์มกระดาษคำตอบทั้ง 2 รูปแบบได้ถูกต้องทุกครั้ง



(ก) (ข)

ภาพที่ 6 ตัวอย่างภาพแบบฟอร์มกระดาษคำตอบข้อสอบปรนัยที่ใช้ในการทดลอง (ก) รูปแบบที่ 1 และ (ข) รูปแบบที่ 2

6. สรุป

การสร้างโมเดลของแบบฟอร์มเป็นงานที่สำคัญต่อระบบการประมวลผลแบบฟอร์ม เพราะโมเดลของแบบฟอร์มเป็นฐานความรู้ที่สำคัญสำหรับการทำงานของคอมพิวเตอร์กับแบบฟอร์มแต่ละประเภทเพื่อใช้ในการค้นหาขอบเขตของฟิลด์ข้อมูลที่กรอกอยู่ในแต่ละช่องของแบบฟอร์ม การจำแนกประเภทแบบฟอร์มก็เป็นอีกขั้นตอนหนึ่งที่สำคัญมาก ในบทความนี้ได้เสนอวิธีการที่มีขั้นตอนการทำงานร่วมกันเพื่อให้บรรลุถึงเป้าหมายของงานทั้งสอง โดยอยู่บนพื้นฐานของการค้นหาเส้นตรงในแนวนอนและแนวตั้ง ซึ่งสามารถนำรูปแบบการตัดกันของเส้นตรงมาประกอบกันเป็นฟิลด์รูปสี่เหลี่ยมผืนผ้า อัลกอริธึมที่นำเสนอนี้นอกจากการประยุกต์ใช้ในระบบตรวจข้อสอบปรนัยอัตโนมัติแล้ว ยังสามารถนำไปประยุกต์ใช้กับระบบประมวลผลแบบฟอร์มประเภทอื่นๆ ได้อีกเช่นกัน

7. เอกสารอ้างอิง

[1] R. Casey, D. Ferguson, K. Mohiuddin and E. Walach, “Intellegent Form Processing Systems,” *Machine Vision and Application,* Vol. 5, pp.143-155, 1992.

[2] S.L. Taylor, R. Fritzon and J.A. Pastor, “Extraction of Data from Preprinted Form,” *Machine Vision Application.* Vol. 5, pp. 211-222, 1992.

[3] L.O’Gormam and R.Kasturi. *Document Image Analysis.* IEEE Computer Society Press, 1995.

[4] A. Dengel, “ANASTASIL : A System for Low-level and High-level Geometric Analysis of printed documents,” IN *Structured Document Image Analysis,* H.S.Baird, H.Bunke, and K. Yamamoto, eds., Springer-Verlag, pp. 70-98, 1992.

[5] ทัศนีย์ ชังเทศ และ สมภพ ถาวรยิ่ง. *การวิเคราะห์การถดถอยและสหสัมพันธ์,* โรงพิมพ์มหาวิทยาลัยธรรมศาสตร์, กรุงเทพฯ, 2530.

[6] กฤษณะ ชินสาร. *ระบบตรวจข้อสอบปรนัยอัตโนมัติ,* วิทยานิพนธ์สำหรับปริญญาวิทยาศาสตรมหาบัณฑิต, สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2540