

วิธีการหากลุ่มข้อมูลของเอกสารอย่างรวดเร็ว

A Fast Block Extraction Method for Document Segmentation

ไพศาล สุทธิบรรเจิด*, วิเชียร เปรมชัยสวัสดิ์**, และ นุชรี เปรมชัยสวัสดิ์***

*คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ถนนฉลองกรุง เขตลาดกระบัง กรุงเทพฯ 10520 โทร. 0-2860-8886 E-mail: mr_phaisarn@yahoo.com

**ผู้ช่วยอธิการบดี มหาวิทยาลัยสยาม

235 ถ.เพชรเกษม เขตภาษีเจริญ กรุงเทพฯ 10163 wichian@siam.edu

***คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยธุรกิจบัณฑิตย์

ถนนประชาชื่น หลักสี่ กรุงเทพฯ 10210 nucharee@dpu.ac.th

บทคัดย่อ

ปัจจุบันมีวิธีที่ใช้ในการแบ่งโครงสร้างของเอกสารอย่างมีประสิทธิภาพและรวดเร็ว เช่น การทำ Block Extraction โดยใช้กรอบหน้าต่างขนาด 32*32 พิกเซล[1] วิธีนี้จะหาจุดดำจุดแรกเพื่อใช้ในการเริ่มต้นวางกรอบหน้าต่างขนาด 32*32 พิกเซล จากนั้นจะใช้กรอบหน้าต่างนี้เดินรอบขอบของกลุ่มข้อมูล ก็จะสามารถแบ่งเอกสารออกเป็นกลุ่มข้อมูลได้อย่างรวดเร็ว แต่วิธีที่กล่าวถึงนี้ไม่ได้อธิบายในส่วนของ การหาจุดดำจุดแรก ซึ่งใช้เป็นจุดเริ่มต้นในการประมวลผล ในบทความนี้จะได้นำเสนอวิธีที่ใช้ในการหาจุดดำจุดแรกอย่างมีประสิทธิภาพ และรวดเร็ว ด้วยการใช้น้ำต่าง แล้วตรวจสอบเพียงบางจุดในหน้าต่าง ไม่ต้องตรวจสอบทุกจุด ทำให้สามารถทำ Block Extraction ได้รวดเร็วขึ้น ในบทความนี้จะได้นำเสนอหน้าต่างที่ออกแบบขึ้นทั้งหมด 6 แบบ นำมาเปรียบเทียบ และเลือกหน้าต่างที่เหมาะสมที่สุดในการนำไปใช้ จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถทำงานได้เร็วขึ้น

คำสำคัญ: การแบ่งโครงสร้างของเอกสาร, หน้าต่าง

Abstract

In the document segmentation process, the first black pixel must be found and used as the starting point of the process. This paper presents a new method for finding the starting point in an efficient way. To speed up the process. Six types of windows are proposed and their efficiency is also compared. The experimental results show that the proposed method can speed up the process for finding the starting point significantly.

Key words: document segmentation, window

1. คำนำ

การทำ Document Segmentation มีอัลกอริทึมที่ใช้ในการทำอยู่หลายอัลกอริทึม โดยจุดเด่นของแต่ละอัลกอริทึมจะอยู่ที่ประสิทธิภาพในด้านความถูกต้องแม่นยำ และความรวดเร็วในการคำนวณ โดยอัลกอริทึมต่างๆเหล่านี้สามารถแบ่งได้เป็น 3 ประเภท ได้แก่

1.1 วิธีการบนลงล่าง (Top-Down Approach) [2]

วิธีการนี้ในเบื้องต้นจะทำการแบ่งภาพอย่างหยาบๆก่อน โดยแบ่งภาพเอกสารให้เป็นกลุ่มข้อมูลที่มีขอบเขตใหญ่ก่อน จากนั้นจึงนำแต่ละกลุ่มข้อมูลที่มีขอบเขตใหญ่ มาแบ่งอย่างละเอียดอีกที ให้เป็นกลุ่มข้อมูลที่มีขอบเขตเล็กกลง

1.2 วิธีการล่างขึ้นบน (Bottom-Up Approach) [3]

เป็นการวิเคราะห์ส่วนประกอบย่อย แล้วนำส่วนประกอบย่อยที่ได้มาประกอบกันเป็นบรรทัด และกลุ่มข้อมูล(block) โดยพิจารณาจากความใกล้ชิด และขนาดของแต่ละส่วนประกอบ

1.3 วิธีการแบบผสม (Mixed Approach) [4-5]

เป็นการนำวิธี top-down มาใช้ร่วมกับวิธี bottom-up เช่น อัลกอริทึมของ B. Kruatrachue และ P. Suthaphan [1] ซึ่งทำงานดังนี้

1. Block Extraction (top-down) [1] โดยใช้กรอบหน้าต่างขนาด 32*32 พิกเซลเดินรอบขอบของกลุ่มข้อมูลเพื่อใช้ในการแบ่งข้อมูลออกเป็นส่วนประกอบใหญ่

2. Multi-Column Block Detection and Segmentation (Bottom-up) [1] เดินรอบขอบทีละ 1 พิกเซล เพื่อหาขอบเขตย่อย แล้วนำขอบเขตย่อยมาหาแนวบรรทัด เพื่อใช้ในการตัดแบ่งคอลัมน์

จุดเด่นของวิธีนี้คือ การใช้กรอบหน้าต่างขนาด 32*32 พิกเซล [1] ในการเดินรอบขอบของกลุ่มข้อมูล โดยไม่ต้องตรวจสอบทุกจุด ทำให้การประมวลผลเป็นไปอย่างรวดเร็ว แต่วิธีนี้ยังไม่สมบูรณ์ เพราะไม่ได้กล่าวถึงขั้นตอนการหาจุดดำจุดแรก ที่ใช้เป็นจุดเริ่มต้นในการวางกรอบหน้าต่างขนาด 32*32 พิกเซล

บทความนี้จึงนำเสนอวิธีการหาจุดค่าจุดแรก เพื่อใช้เป็นจุดเริ่มต้น ในการวางกรอบหน้าต่างขนาด 32*32 พิกเซล อย่างมีประสิทธิภาพ และรวดเร็ว ด้วยการใช้น้ำต่างแล้วตรวจสอบเพียงบางตำแหน่งในหน้าต่าง

ในบทความนี้ได้พัฒนาและออกแบบหน้าต่างทั้งหมด 6 แบบ แล้วทำการทดลอง และนำผลมาเปรียบเทียบหาขนาด และรูปแบบหน้าต่างที่ให้ผลการทำงานดีที่สุด โดยวัดจากประสิทธิภาพในด้านความถูกต้องและความรวดเร็วในการคำนวณ

2. การแบ่งส่วนประกอบของภาพเอกสาร

การแบ่งส่วนประกอบของภาพเอกสาร คือการแบ่งขอบเขตของภาพออกเป็นส่วนๆ โดยในแต่ละส่วนจะมีรายละเอียดที่เป็นชนิดเดียวกัน เช่น ขอบเขตข้อมูลที่เป็นข้อความ, ขอบเขตข้อมูลที่เป็นภาพ ซึ่งแบ่งขั้นตอนการทำงานได้เป็น 2 ขั้นตอนดังนี้

2.1 Block Extraction (top-down)

การทำ Block Extraction คือการแบ่งส่วนประกอบของเอกสารออกเป็นส่วนใหญ่ๆ เหมือนกับการใช้สายตามองภาพในระยะไกลแล้วแบ่งภาพออกเป็นส่วนๆ ดังในรูปที่ 2.

การทำ Block Extraction สามารถแบ่งการทำงานออกได้เป็น 2 ขั้นตอน ดังนี้



รูปที่ 1. ภาพต้นฉบับ



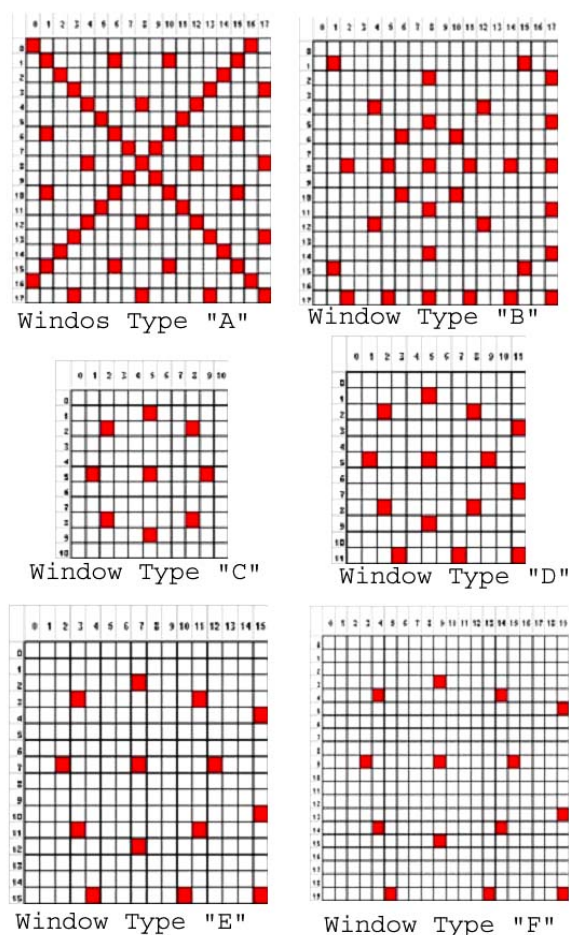
รูปที่ 2. ภาพที่ผ่านขั้นตอนการทำ Block Extraction

2.1.1 การหาจุดเริ่มต้นของกลุ่มข้อมูล

ในขั้นตอนนี้ จะทำการค้นหา และกำหนดตำแหน่งเริ่มต้นในการวางกรอบหน้าต่างอันแรก (กรอบหน้าต่างขนาด 32*32พิกเซล) ของขั้นตอนการหาขอบเขตของกลุ่มข้อมูล

การหาตำแหน่งเริ่มต้นของกลุ่มข้อมูล ก็คือการตรวจสอบหาจุดค่าจุดแรก ซึ่งถ้าสามารถตรวจสอบหาจุดค่าจุดแรกได้อย่างรวดเร็ว ก็จะ สามารถทำขั้นตอน Block Extraction ได้อย่างรวดเร็วตามไปด้วย ในบทความนี้จึงได้เสนอวิธีการหาจุดค่าจุดแรกอย่างรวดเร็วและมีประสิทธิภาพ ด้วยการใช้น้ำต่าง ในการทำงานแล้วตรวจสอบเพียงบางตำแหน่งในหน้าต่าง ทำให้การประมวลผลเป็นไปได้อย่างรวดเร็ว

ในบทความนี้ ได้นำเสนอหน้าต่างที่ได้พัฒนา และออกแบบทั้งหมด 6 แบบ ดังแสดงในรูปที่ 3.



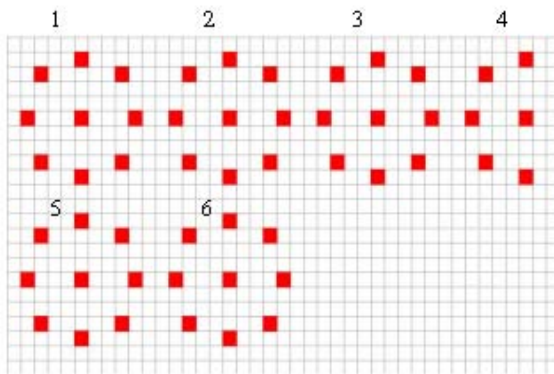
รูปที่ 3. หน้าต่างแบบ “A” – “F”

หน้าต่างแต่ละแบบที่ได้พัฒนาและออกแบบมา มีข้อแตกต่างกันดังนี้

- ขนาดหน้าต่าง
- จำนวนจุดที่ใช้ตรวจสอบ (จุดในรูปคือตำแหน่งที่ใช้ตรวจสอบ)
- การกระจายตำแหน่งของจุดที่ใช้ตรวจสอบ

แนวคิดที่ใช้ในการออกแบบหน้าต่าง คือ หน้าต่างต้องมีขนาดใหญ่ แต่จำนวนจุดที่ใช้ในการตรวจสอบต้องมีจำนวนน้อย (เพื่อให้การตรวจสอบหน้าต่างเป็นไปอย่างรวดเร็ว) และการกระจายตำแหน่งจุดที่ใช้ตรวจสอบต้องเป็นไปอย่างสมดุล เพื่อให้แต่ละจุดครอบคลุมพื้นที่ให้มากที่สุด

การหาจุดเริ่มต้นของกลุ่มข้อมูล(การหาจุดค่าจุดแรก) ทำได้โดยนำหน้าต่างแรกวางที่ตำแหน่งมุมบนซ้ายของภาพเอกสาร แล้วตรวจสอบภายในหน้าต่างว่ามีจุดค่าหรือไม่ โดยตำแหน่งที่ทำการตรวจสอบหาจุดค่าขึ้นอยู่กับชนิดของหน้าต่างที่นำมาใช้ (ดูรูปที่ 3. ประกอบ) ถ้าตรวจพบจุดค่าภายในหน้าต่าง จะตรวจสอบว่าจุดค่าที่พบเป็นสัญญาณรบกวน (noise) หรือไม่โดยการเดินรอบขอบ ณ. ตำแหน่งที่พบจุดค่า โดยเดินทีละ 1 พิกเซล ถ้าความกว้าง หรือความยาวที่ได้มากกว่า 5 พิกเซล ให้ถือว่าสิ่งที่อยู่ภายในหน้าต่างนี้คือข้อมูลให้จบการทำงานในขั้นตอนนี้ และส่งตำแหน่งมุมบนซ้ายของหน้าต่างนี้ ให้เป็นจุดเริ่มต้นของขั้นตอน “การหาขอบเขตของกลุ่มข้อมูล” ต่อไป แต่ถ้าความกว้างและความยาวน้อยกว่า 5 พิกเซล ถือว่ากลุ่มจุดค่าที่พบเป็นสัญญาณรบกวน ให้ลบสัญญาณรบกวนนี้ออกจากภาพเอกสาร แล้วตรวจสอบตำแหน่งต่อไปภายในหน้าต่าง ถ้าตรวจสอบทุกตำแหน่งในหน้าต่างแล้ว ยังไม่พบจุดค่า ให้ถือว่าภายในหน้าต่างนี้ไม่มีข้อมูล จากนั้นให้วางหน้าต่างต่อไป การวางหน้าต่างจะวางจากซ้ายไปขวา และจากบนลงล่างดังในรูปที่ 4.

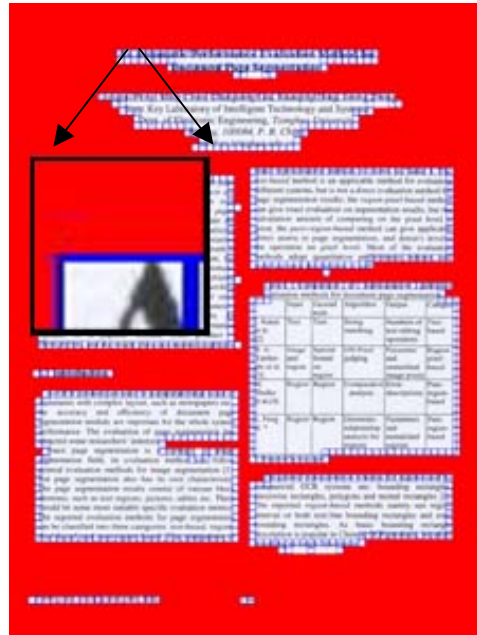


รูปที่ 4. ลำดับการวางหน้าต่างในขั้นตอน การหาจุดเริ่มต้นของกลุ่มข้อมูล

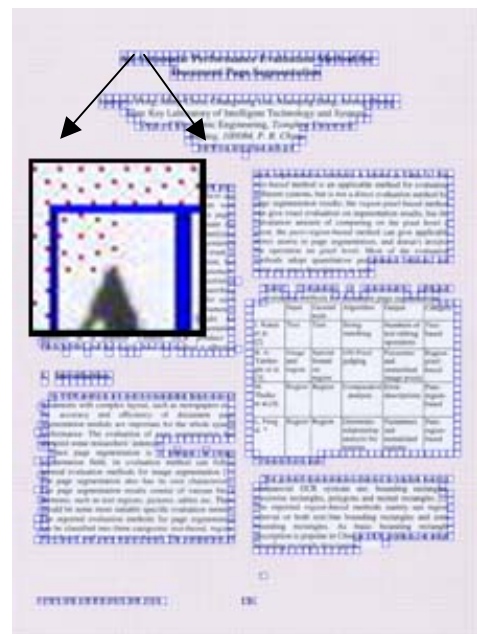
2.1.2 การหาขอบเขตของกลุ่มข้อมูล

เมื่อได้ตำแหน่งเริ่มต้นของกลุ่มข้อมูลแล้ว จะทำการวางกรอบหน้าต่างขนาด 32*32 พิกเซล รอบขอบของกลุ่มข้อมูลตามทิสทาร์รหัสแนว (Chain Codes) จนกระทั่ง หน้าต่างกลับมาที่ตำแหน่งเริ่มต้นจึงหยุด โดยกำหนดว่า ในกรอบหน้าต่างขนาด 32*32 พิกเซลนี้ ต้องมีพิกเซลค่าอยู่อย่างน้อย 10 พิกเซล จึงจะถือว่าเป็นหน้าต่างที่มีข้อมูล การเดินรอบขอบข้อมูลด้วยกรอบหน้าต่างนี้ ทำให้รู้ขอบเขตของกลุ่มข้อมูลได้ ซึ่งขอบเขตของกลุ่มข้อมูลที่ได้นี้ สามารถมีรูปร่างต่างกันได้ ขึ้นอยู่กับข้อมูลที่มี ไม่จำเป็นที่จะต้องเป็นสี่เหลี่ยม จากนั้นกลับไปทำขั้นตอนที่ 2.1.1 เพื่อหาจุดเริ่มต้นของกลุ่มข้อมูลอื่นๆต่อไป

หลังจากผ่านขั้นตอน Block Extraction แล้ว ข้อมูลที่ได้จะถูกแบ่งออกเป็นกลุ่มข้อมูล(block) ซึ่งในแต่ละกลุ่มข้อมูลอาจมีมากกว่า 1 คอลัมน์ได้ เนื่องจากระยะห่างระหว่างคอลัมน์น้อยกว่า 32 พิกเซล ซึ่งต้องมีการแยกคอลัมน์ออกจากกันโดยใช้ Multi-Column Block Detection and Segmentation ซึ่งจะกล่าวถึงในขั้นตอนต่อไป



รูปที่ 5. ภาพที่ผ่านขั้นตอนการทำ Block Extraction โดยการหาตำแหน่งเริ่มต้นของกลุ่มข้อมูลด้วยการสแกนแบบ raster



รูปที่ 6. ภาพที่ผ่านขั้นตอนการทำ Block Extraction โดยการหาตำแหน่งเริ่มต้นของกลุ่มข้อมูลด้วยการใช้หน้าต่าง

รูปที่5-6 เปรียบเทียบ การหาตำแหน่งเริ่มต้นของกลุ่มข้อมูลด้วยวิธีการสแกนแบบ raster ดังรูปที่5. และการหาตำแหน่งเริ่มต้นของกลุ่มข้อมูลด้วยการใช้หน้าต่าดังรูปที่6. โดยจุดสีแดงในรูปแสดงตำแหน่งที่ใช้ในการตรวจสอบหาจุดค่าจุดแรก ซึ่งสามารถเห็นได้ว่าจุดสีแดงในรูปที่6. น้อยกว่าจุดสีแดงในรูปที่5. ซึ่งหมายถึงว่า รูปที่6. สามารถประมวลผลได้รวดเร็วกว่ารูปที่5. เป็นอย่างมาก (เส้นสีน้ำเงินในรูปคือ กรอบหน้าต่าขนาด 32*32 พิกเซล)

2.2 Multi-Column Block Detection and Segmentation (bottom - up)

จากขั้นตอน Block Extraction ที่ผ่านมา เป็นไปได้ที่ใน 1 กลุ่มข้อมูล (block) จะมีมากกว่า 1 คอลัมน์ จึงใช้วิธี bottom-up ในการแบ่งคอลัมน์ให้ 1 กลุ่มข้อมูลมีเพียง 1 คอลัมน์ โดยแยกข้อมูลออกเป็นหน่วยย่อยที่สุดด้วยการเดินรอบขอบทีละ 1 พิกเซล จะทำให้สามารถหากรอบสี่เหลี่ยม รอบส่วนประกอบย่อยได้ (bounding box) ถ้ากรอบสี่เหลี่ยมของส่วนประกอบย่อยที่ได้ มีความกว้าง หรือความสูงมากกว่าที่กำหนดไว้ ก็จะกำหนดให้เป็นข้อมูลชนิดรูปภาพ แต่ถ้าความกว้าง และความสูงน้อยกว่าค่าที่กำหนดไว้ ก็จะให้เป็นข้อมูลชนิดตัวอักษร จากนั้นนำกรอบสี่เหลี่ยมของส่วนประกอบย่อยมาหาแนวระดับบรรทัด จากนั้นวิเคราะห์การตัดแบ่งคอลัมน์ ด้วยการฉายเงาภาพจุดขาวที่บรรทัดแรกของกลุ่มข้อมูล การฉายเงาภาพจุดขาวจะทำเฉพาะจุดขาวที่อยู่ระหว่าง กรอบสี่เหลี่ยมของส่วนประกอบย่อย ตามแนวตั้ง ถ้าความกว้าง และความสูงของการฉายเงาภาพจุดขาวนี้มีค่ามากกว่าค่าที่กำหนดไว้ จะกำหนดให้เป็นแนวแบ่งคอลัมน์ แนวแบ่งคอลัมน์สิ้นสุดที่แถว(y)ใด ก็จะเริ่มต้นทำการฉายเงาภาพจุดขาวที่แถวนั้นต่อ จนกระทั่งสิ้นสุดภาพเอกสาร หรือจบเมื่อไม่สามารถตัดแบ่งคอลัมน์ได้อีก

3. ผลการทดลอง

จากหน้าต่าที่ได้พัฒนา และออกแบบมาทั้ง 6 แบบ จะเลือกหน้าต่าที่ดีที่สุด มาใช้หาจุดเริ่มต้นของกลุ่มข้อมูล โดยเลือกจากเงื่อนไขในด้านของประสิทธิภาพความถูกต้อง และเวลาที่ใช้ในการคำนวณ

3.1 ความถูกต้อง

นำหน้าต่าทั้ง 6 แบบ มาทดสอบกับภาพอักษรภาษาไทยและภาษาอังกฤษ หน้าต่าแบบ “A”-“D” สามารถทำงานได้อย่างถูกต้อง ส่วนหน้าต่าแบบ “E”-“F” ยังมีการทำงานที่ผิดพลาดอยู่บ้าง คือ ไม่สามารถตรวจจับสัญลักษณ์บางตัวได้ ดังในรูปที่7.

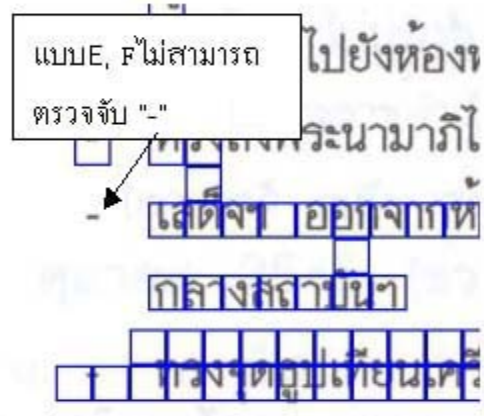
3.2 เวลาที่ใช้ในการคำนวณ

นำหน้าต่าทั้ง 6 แบบ มาเปรียบเทียบเวลาการทำงาน เพื่อแสดงให้เห็นถึงประสิทธิภาพในการคำนวณที่รวดเร็วของหน้าต่าแต่ละแบบ โดยการทดลองนี้ จับเวลาเฉพาะขั้นตอนการทำ Block Extraction เท่านั้น

เพื่อให้สามารถเห็นถึงเวลาที่แตกต่าง ของหน้าต่าแต่ละแบบได้อย่างชัดเจน



(ก)



(ข)

รูปที่ 7. (ก) แสดงความผิดพลาดจากการใช้หน้าต่า “E”, “F” (ข) จากรูป (ก) ขยายภาพให้เห็นชัดเจนขึ้น

การทดลองนี้ ทำบนเครื่อง Pentium4 1.6 GHz แรม 256 MB ภายใต้ระบบปฏิบัติการ WindowXP ใช้โปรแกรมภาษา Visual Basic.Net 2002 เอกสารขนาด A4 สแกนที่ความละเอียด 300dpi ภาพเอกสารไม่มีกรอบ ภาพเอกสารที่ใช้ในการจับเวลาแสดงในรูปที่9. เวลาที่ใช้ในการทดลองแสดงในตารางที่1.

ตารางที่ 1. เปรียบเทียบเวลาการทำงานของหน้าต่างแบบต่างๆ

แบบหน้าต่าง	ขนาดหน้าต่าง	จุดทั้งหมด	จุดที่ใช้ตรวจ	จุดที่ใช้ตรวจ สอดคิดเป็น %	เวลา (sec)
A	18x18	324	52	16.049	7.801
B	18x18	324	32	9.877	7.39
C	11x11	121	9	7.438	7.37
D	12x12	144	14	9.72	7.43
E	16x16	256	14	5.469	7.06
F	20x20	400	14	3.5	6.77

จากตารางที่ 1 แสดงเวลาการทำงานในขั้นตอน Block Extraction โดยใช้การหาจุดเริ่มต้นของกลุ่มข้อมูล ด้วยหน้าต่างแบบต่างๆ หน้าต่างทั้ง 6 แบบใช้เวลาประมาณ 6 - 8 วินาที ซึ่งรวดเร็วกว่าการหาจุดเริ่มต้นของกลุ่มข้อมูลด้วยการสแกนแบบ raster ซึ่งใช้เวลา 24.986 วินาที

4. สรุปผลการทดลอง

ในบทความนี้นำเสนอ วิธีการหาจุดเริ่มต้นของกลุ่มข้อมูล อย่างมีประสิทธิภาพ และรวดเร็วด้วยหน้าต่างแบบ “C” ซึ่งเหมาะกับงานคุณภาพ ที่ต้องการความถูกต้องสูง และการประมวลผลที่รวดเร็ว

แต่ในกรณีที่ต้องการนำอัลกอริทึมนี้ ไปใช้กับภาพเอกสารที่สแกนด้วยความละเอียดที่มากกว่า 300dpi (มีผลทำให้ตัวอักษรในภาพเอกสารมีขนาดใหญ่ขึ้น) หรือนำไปใช้กับภาพเอกสารที่ตัวอักษรมีขนาดใหญ่ ขอแนะนำหน้าต่างแบบ “E” (ซึ่งสามารถทำงานได้อย่างถูกต้องเมื่อตัวอักษรมีขนาดใหญ่ขึ้นและยังทำงานได้รวดเร็วกว่าหน้าต่างแบบ “C”) แต่ถ้าตัวอักษรมีขนาดใหญ่มาก ขอแนะนำหน้าต่างแบบ “F” หรือจะสร้างหน้าต่างขึ้นมาใหม่โดยนำหน้าต่างแบบ “F” มาทำการขยายขนาด (ซึ่งจริงๆแล้วหน้าต่างแบบ “F” ก็คือการนำหน้าต่างแบบ “E” มาขยายขนาดนั่นเอง)



รูปที่ 8. ตัวอย่างภาพที่ผ่านขั้นตอน Block Extraction



รูปที่ 9. ตัวอย่างภาพที่ผ่านขั้นตอน Block Extraction

กิตติกรรมประกาศ

ขอขอบคุณ อาจารย์ วิเชียร เปรมชัยสวัสดิ์ ที่ให้คำแนะนำปรึกษาจนบทความนี้สำเร็จลุล่วงไปด้วยดี
ขอขอบคุณ อาจารย์ ธนารัตน์ ชลิดาพงศ์ ที่ได้ส่งสอนความรู้พื้นฐานเกี่ยวกับ Digital Image Processing

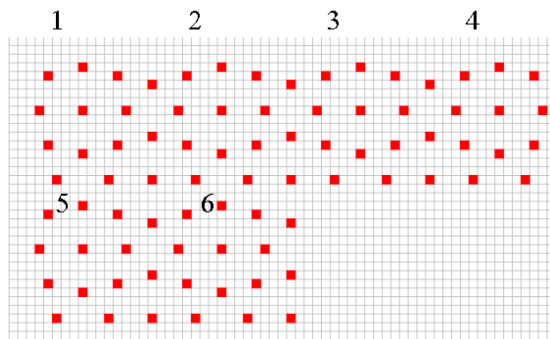
เอกสารอ้างอิง

- [1] B. Kruatrachue, P. Suthaphan, “A Fast and Efficient Method for Document Segmentation for OCR,” *Electrical and Electronic Technology*, 2001. Vol. 1, 19-22 Aug. 2001 pp. 381 -383
- [2] T. Akiyama and N. Hagita, “Automated entry system for printed documents,” *Pattern Recognition* 23, 1990, pp. 1141-1154.
- [3] K. C. Fan, C. H. Liu, and Y. K. Wang, “Segmentation and classification of mixed text/graphics/image documents,” *Pattern Recognition Letters* 15, 1994, pp. 1201-1209.
- [4] S. N. Srihari, T. Hong, and G. Srikantan, “Machine printed Japanese document recognition,” *Pattern recognition* 30, 1997, pp. 1301-1313.
- [5] D. Wang, S.N. Srihari. “Classification of Newspaper Image Blocks using Texture Analysis,” *Computer Vision, Graphics, and Image Processing*, Vol. 47, 1989, pp. 327-352.

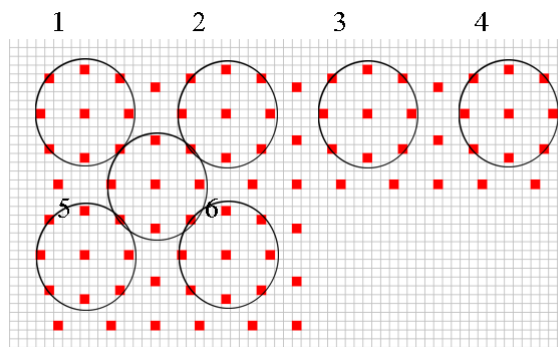
ภาคผนวก แนวคิดในการออกแบบหน้าต่าง

ในขั้นตอนการออกแบบหน้าต่างได้ค้นพบว่า หน้าต่างแบบ “C” มีความสมมาตร เหมาะที่จะนำมาใช้ในการตรวจหาจุดดำจุดแรก แต่การนำหน้าต่างมาใช้ โดยการวางหน้าต่างจากซ้ายไปขวา และจากบนลงล่างดังรูปที่ 4. ทำให้เกิดข้อจำกัด คือ ไม่สามารถขยายขนาดของหน้าต่างให้มีขนาดใหญ่ขึ้นได้ เพราะการขยายขนาดหน้าต่างจะทำให้ช่องว่างระหว่างหน้าต่างขนาดใหญ่ขึ้นตามไปด้วย ซึ่งพื้นที่บริเวณนี้จะไม่ได้รับการตรวจหาจุดดำ

ทางออกของปัญหานี้คือต้องวางหน้าต่างแบบ “C” เชื่อกัน ซึ่งช่องว่างที่เกิดขึ้นจะมีขนาดเล็กกว่าการวางเรียงแบบเดิม จึงได้มีการพัฒนาหน้าต่างแบบ “D” แล้วทำการขยายขนาดเป็นหน้าต่างแบบ “E” และหน้าต่างแบบ “F” ตามลำดับ



(ก)



(ข)

รูปที่ 10. ลำดับการวางหน้าต่างแบบ “E”

รูปที่ 10. (ก) เป็นการวางหน้าต่างแบบ “E” จำนวน 6 ครั้ง
(ข) เน้นให้เห็นถึงหน้าต่างแบบ “C” ที่ซ่อนอยู่ภายในหน้าต่างแบบ “E”



นาย ไพศาล สุธีบรรณเจิด
คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยี
พระจอมเกล้าเจ้าคุณทหารลาดกระบัง
สนใจงานในด้าน Digital Image Processing



รศ.ดร. วิเชียร เปรมชัยสวัสดิ์
จบปริญญาเอกจาก Waseda University, Tokyo
ประเทศญี่ปุ่น



รศ.ดร.นุชรี เปรมชัยสวัสดิ์
จบปริญญาเอกจาก Waseda University, Tokyo
ประเทศญี่ปุ่น