

การหาพิกเซลภาพที่ติดกันในภาพ 3 มิติ

Connected Components Labeling in 3-Dimension

ไพศาล สุธีบรรเจ็ด¹ และ วิเชียร เปรมชัยสวัสดิ์²

บัณฑิตวิทยาลัย สาขาวิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยสยาม

38 ถ. เพชรเกษม เขตภาษีเจริญ กรุงเทพฯ 10160 โทรศัพท์: 0-2868-6656

E-mail: mr.phaisam@gmail.com¹, wichian@siam.edu²

บทคัดย่อ

การค้นหาคอมโพเนนต์ที่ติดกัน (Connected components) เป็นกระบวนการทำงานขั้นพื้นฐานในการประมวลผลภาพดิจิทัล (Digital Image Processing) งานวิจัยนี้นำเสนอ Scan mask ที่ใช้ในการตรวจสอบหาพิกเซลที่ติดกันในภาพ 3 มิติ ที่มีจำนวนจุดตรวจสอบลดลงจากเดิม 14 ตำแหน่ง เหลือเพียง 8 ตำแหน่ง

คำสำคัญ: พิกเซลภาพที่ติดกัน, หน้ากาก

Abstract

The connected component labeling is the underlying technique that typically used in today's digital image processing. In the digital world today, processing speed is crucial in particular, working with large images. This research proposes a new scan mask for connected component labeling operation in 3-dimension image that the number of pixels to be inspected is reduced from 14 to 8 positions.

Keywords: connected component, scan mask

1. คำนำ

การค้นหาคอมโพเนนต์ที่ติดกัน (Connected components) เป็นกระบวนการทำงานขั้นพื้นฐานในการประมวลผลภาพดิจิทัล (Digital Image Processing) ทั้งภาพแบบ 2 มิติ เช่น การแปลงไฟล์ภาพเป็นไฟล์อักขร (Optical character recognition หรือ OCR) การค้นคืนรูปภาพ (Image Retrieval) และ 3 มิติ เช่น การประมวลผลภาพทางการแพทย์ที่ได้จากเครื่อง MRI

2. การค้นหาคอมโพเนนต์ที่ติดกัน

การค้นหาคอมโพเนนต์ที่ติดกัน โดยทั่วไปจะทำการค้นหาในภาพขาวดำ หรือเรียกว่าภาพไบนารี (Binary image) โดยแบ่งพิกเซลภาพออกเป็น 2 ส่วนคือ ส่วนที่เป็นวัตถุ (Foreground) และส่วนที่เป็นพื้นหลัง (Background)

2.1 การค้นหาคอมโพเนนต์ที่ติดกันในภาพ 2 มิติ

การนิยามพิกเซลภาพที่ติดกันในภาพ 2 มิติ นิยมใช้ 8-connectivity $N_8(p)$ ดังสมการที่ 1 และในรูปที่ 1

$$N_8(p) = \{P_{(x+1,y)}, P_{(x-1,y)}, P_{(x,y+1)}, P_{(x,y-1)}, P_{(x+1,y+1)}, P_{(x+1,y-1)}, P_{(x-1,y+1)}, P_{(x-1,y-1)}\} \quad (1)$$

$P_{(x-1,y+1)}$	$P_{(x,y+1)}$	$P_{(x+1,y+1)}$
$P_{(x-1,y)}$	$P_{(x,y)}$	$P_{(x+1,y)}$
$P_{(x-1,y-1)}$	$P_{(x,y-1)}$	$P_{(x+1,y-1)}$

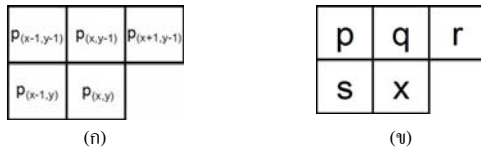
รูปที่ 1. 8-connectivity

วิธีการค้นหาคอมโพเนนต์ของวัตถุที่ติดกันในภาพ 2 มิติ ได้มีการนำเสนอครั้งแรกโดย Rosenfeld และ Pfaltz. ในปี 1966 [1] โดยทำการสแกนแบบ Raster scan จำนวน 2 ครั้ง หรือที่เรียกว่า Two-scans labeling โดยการทำการ Two-scans labeling จะแบ่งการทำงานออกเป็น 3 ขั้นตอนดังนี้

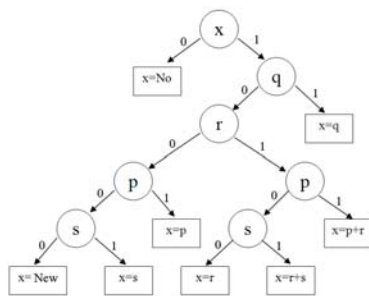
1. *First image scan*: เป็นการนำ scan mask ดังแสดงในรูปที่ 2 มาทำการค้นหาคอมโพเนนต์ที่เป็นวัตถุ Foreground จากนั้นจะกำหนดค่า Label แบบชั่วคราว (Provisional label) ให้กับพิกเซลภาพที่เป็นวัตถุ และทำการบันทึกค่าของ Provisional Label ลงใน Equivalences Table
2. *Equivalences Resolution*: เป็นการนำ Provisional Label ใน Equivalences Table มาทำการจัดกลุ่ม และเลือกค่าน้อยที่สุดของแต่ละกลุ่ม เพื่อนำมาใช้เป็นตัวแทนของกลุ่ม
3. *Second image scan*: ทำการสแกนในรอบที่สอง เพื่อนำตัวแทนของแต่ละกลุ่มในตาราง Equivalences Table มาทำการ Label ลงที่ภาพคำตอบ

ในปี 2005 Wu *et al.* [2] ได้เสนอวิธีการค้นหาคอมโพเนนต์ที่ติดกันในขั้นตอน First image scan ได้รวดเร็วขึ้น โดยการลดจำนวนครั้งของการตรวจสอบพิกเซลภาพในรอบแรกของ Two-scans labeling ลง กล่าวคือ โดยทั่วไปการค้นหาคอมโพเนนต์ที่ติดกัน ด้วยวิธี 8-connectivity จะทำการตรวจสอบพิกเซลภาพปัจจุบัน "x" ถ้าเป็น Foreground ก็จะทำการ

ตรวจสอบตำแหน่งข้างเคียงที่เป็นสมาชิกของ 8-connectivity ของตำแหน่งปัจจุบัน “x” ทั้ง 4 ตำแหน่ง ดังรูปที่ 2 ซึ่งการตรวจสอบทั้ง 4 ตำแหน่ง ทำให้การประมวลผลใช้เวลานาน ดังนั้น Wu *et al.* [2] จึงนำเสนอต้นไม้ตัดสินใจ (Decision Tree) เพื่อลดจำนวนครั้งของการตรวจสอบลง ดังแสดงในรูปที่ 3



รูปที่ 2. Scan mask ที่ใช้ตรวจสอบหา 8-connectivity (ก) ตำแหน่งของพิกเซล (ข) ตัวอักษรที่ใช้เรียกแทนตำแหน่งต่างๆ



รูปที่ 3. ต้นไม้ตัดสินใจที่ใช้ตรวจสอบหา 8-connectivity

จากรูปที่ 3 มีการทำงานอยู่ 4 ประเภทดังนี้

1. *No action*: พิกเซล “x” เป็น Background ไม่มีการกำหนดค่าให้กับพิกเซล “x”
2. *New label*: พิกเซล “x” เป็น Foreground และพิกเซลข้างเคียงเป็น Background ให้กำหนดค่า Provisional Label ค่าใหม่ ให้กับพิกเซล “x”
3. *Assign action*: พิกเซล “x” เป็น Foreground และพิกเซลข้างเคียงทุกพิกเซลเป็น Foreground ที่มี Provisional Label ค่าเดียวกัน
4. *Merge action*: พิกเซล “x” เป็น Foreground และพิกเซลข้างเคียงมากกว่า 1 พิกเซลเป็น Foreground และมี Provisional Label ที่ไม่ซ้ำกัน

ในปี 2010, Grana *et al.* [3] ได้ทำการวิเคราะห์กรณีที่นำไปได้ทั้งหมดที่เกิดขึ้นจากการใช้ Scan mask ในรูปที่ 2 ให้มาอยู่ในรูปของ OR-decision table ดังแสดงในตารางที่ 1

ในปี 2011, Suthetbanjard และ Premchaiswadi [4] ได้เสนอ P-Mask ที่ใช้ตรวจสอบหา 8-connectivity ดังแสดงในรูปที่ 4 ซึ่งลดจำนวนจุดตรวจสอบลงจากเดิม 5 พิกเซล เหลือเพียง 4 พิกเซล โดยไม่จำเป็นต้องตรวจสอบตำแหน่ง “r” ซึ่งตำแหน่งที่ตรวจสอบยังคงมีความสัมพันธ์กับตำแหน่งข้างเคียงแบบ 8-connectivity ด้วยการเพิ่มการประมวลผลการทำงานแบบ *Merge only action* ขึ้นมา (เช่น Merge ตำแหน่ง q และ s ซึ่ง

เขียนแทนด้วย q+s) ซึ่งการทำงานแบบ Merge only action จะทำการ Merge ค่า Label แบบชั่วคราวของตำแหน่งข้างเคียงเข้าด้วยกัน แต่ไม่ต้องกำหนดค่าที่ได้ให้กับตำแหน่ง “x” การใช้หน้ากากแบบ P-Mask นี้มีกรณีที่เป็นไปได้ทั้งหมดเท่ากับ $2^4=16$ กรณี ดังแสดงในตารางที่ 2

ตารางที่ 1. OR-decision table จาก Scan mask ในรูปที่ 2

Condition	Action										
	no action	new label	Assign				Merge				
			x=p	x=q	x=r	x=s	x=p+r	x=r+s			
x	p	q	r	s							
0	-	-	-	-	1						
1	0	0	0	0		1					
1	1	0	0	0			1				
1	0	1	0	0				1			
1	0	0	1	0					1		
1	0	0	0	1						1	
1	1	1	0	0		1	1				
1	1	0	1	0							1
1	1	0	0	1		1			1		
1	0	1	1	0			1	1			
1	0	1	0	1			1				1
1	1	1	1	0		1	1	1			
1	1	1	0	1		1	1			1	
1	1	0	1	1			1	1	1		1
1	1	1	1	1		1	1	1	1		



รูปที่ 4. Pixel-based scan mask (P-Mask) ใช้ตรวจสอบหา 8-connectivity (ก) ตำแหน่งของพิกเซล (ข) ตัวอักษรที่ใช้เรียกแทนตำแหน่งต่างๆ

ตารางที่ 2. OR-decision table จาก Scan mask ในรูปที่ 4

Condition	Action						
	no action	new label	Assign			Merge	Merge Only
			x=p	x=q	x=s	x=q+s	q+s
x	p	q	s				
0	0	0	0	1			
0	0	0	1	1			
0	0	1	0	1			
0	0	1	1				1
0	1	0	0	1			
0	1	0	1	1			
0	1	1	0	1			
0	1	1	1	1			
1	0	0	0		1		
1	0	0	1			1	
1	0	1	0		1		
1	0	1	1				1
1	1	0	0		1		
1	1	0	1			1	
1	1	1	0		1	1	
1	1	1	1		1	1	1

ตัวอย่างการนำ P-Mask ไปใช้ที่เวลา t_1 และ t_2 แสดงในรูปที่ 5 ซึ่ง จะเห็นได้ว่าที่เวลา t_1 นั้น P-Mask ไม่ได้ทำการตรวจสอบที่ตำแหน่ง “r” แต่ที่เวลา t_2 มีการตรวจสอบตำแหน่ง “q” ซึ่งตำแหน่ง “q” ที่เวลา t_2 ก็คือ ตำแหน่ง “r” ที่เวลา t_1 นั่นเอง ดังนั้น Merge only action (q+s) ที่เวลา $t+1$ จึงสามารถตรวจสอบตำแหน่ง “r” ที่เวลา t ได้ ดังนั้นจึงไม่มีความจำเป็น ที่จะต้องตรวจสอบตำแหน่ง “r” โดยตรง ทำให้ลดจำนวนตำแหน่งที่ต้อง ทำการตรวจสอบภายใน Scan mask ลงได้



รูปที่ 5. แสดงการใช้ P-Mask ที่เวลา t_1 และ t_2

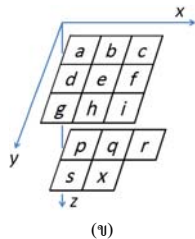
2.2 การค้นหาพิกเซลภาพที่ติดกันในภาพ 3 มิติ

การนิยามพิกเซลภาพติดกันในภาพ 3 มิติ นิยมใช้ 26-connectivity $N_{26}(p)$ ซึ่ง 26-connectivity ก็คือ 8-connectivity ในภาพ 3 มิตินั่นเอง วิธีการค้นหาพิกเซลของวัตถุที่ติดกันในภาพ 3 มิติ ได้มีการนำเสนอโดย Lumia *et al.* ในปี 1983 [5] และ Shirai ในปี 1987 [6] โดยทำการสแกน แบบ Raster scan จำนวน 2 ครั้ง หรือที่เรียกว่า Two-scans labeling โดยใช้ Scan mask ที่มีการตรวจสอบทั้งหมด 14 ตำแหน่ง ดังแสดงในรูปที่ 6

a:P _(x+1,y+1,z-1)	b:P _(x,y+1,z-1)	c:P _(x+1,y+1,z)
d:P _(x+1,y,z-1)	e:P _(x,y,z-1)	f:P _(x+1,y,z)
g:P _(x-1,y+1,z-1)	h:P _(x,y+1,z-1)	i:P _(x+1,y+1,z)

p:P _(x+1,y-1,z)	q:P _(x,y-1,z)	r:P _(x+1,y-1,z)
s:P _(x-1,y,z)	x:P _(x,y,z)	

(ก)



(ข)

รูปที่ 6. 3-D scan mask ที่ใช้ตรวจสอบหา 26-connectivity

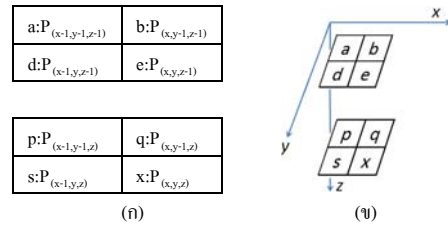
(ก) ตำแหน่งของพิกเซล (ข) ตัวอักษรที่ใช้เรียกแทนตำแหน่งต่างๆ

ในปี 2010, He *et al.* [7] ได้เสนอวิธีการค้นหาพิกเซลที่ติดกันใน ขั้นตอน First image scan ได้รวดเร็วขึ้น โดยการลดจำนวนครั้งของการ ตรวจสอบพิกเซลภาพในรอบแรกของ Two-scans labeling ลง กล่าวคือ โดยทั่วไปการค้นหาพิกเซลภาพที่ติดกัน ด้วยวิธี 26-connectivity จะทำ การตรวจสอบพิกเซลภาพปัจจุบัน “x” โดยถ้าพิกเซลภาพปัจจุบันเป็น Foreground ก็จะทำการตรวจสอบตำแหน่งข้างเคียงที่เป็นสมาชิกของ 26-connectivity ของตำแหน่งปัจจุบัน “x” ทั้งหมด 13 ตำแหน่ง ดังรูปที่ 6 ซึ่ง

การที่ต้องตรวจสอบพิกเซลข้างเคียงทั้ง 13 ตำแหน่ง ทำให้การประมวลผล ใช้เวลานาน ดังนั้น He *et al.* [7] จึงนำเสนอวิธีเพื่อลดจำนวนครั้งของการ ตรวจสอบลง โดยการวิเคราะห์จำนวนของตำแหน่งข้างเคียงของแต่ละ ตำแหน่ง ถ้าตำแหน่งใดมีจำนวนข้างเคียงมากก็ให้ทำการตรวจสอบ ตำแหน่งดังกล่าวก่อน ทำให้ในบางกรณีไม่จำเป็นต้องทำการตรวจสอบ ตำแหน่งข้างเคียงครบทั้ง 13 ตำแหน่ง ทำให้การทำงานโดยรวมทำได้ รวดเร็วขึ้น

3. วิธีการที่นำเสนอ

บทความนี้นำเสนอ Scan mask ในขั้นตอน First image scan ที่ใช้ ในการตรวจสอบหาพิกเซลที่ติดกันในภาพ 3 มิติแบบ 26-connectivity ที่มีจำนวนจุดตรวจสอบลดลงจาก 14 ตำแหน่ง เหลือเพียง 8 ตำแหน่ง ดัง แสดงในรูปที่ 7



(ก)

(ข)

รูปที่ 7. นำเสนอ 3-D scan mask ที่ใช้ตรวจสอบหา 26-connectivity

(ก) ตำแหน่งของพิกเซล (ข) ตัวอักษรที่ใช้เรียกแทนตำแหน่งต่างๆ

3-D scan mask ที่นำเสนอ ไม่ได้ทำการตรวจสอบตำแหน่ง “r, c, f, g, h, i” ที่เวลา t โดยตรง แต่จะทำการตรวจสอบตำแหน่งดังกล่าวนี้ ที่ เวลา $t+1$ ดังแสดงในตารางที่ 3

ตารางที่ 3. แสดงความสัมพันธ์ของตำแหน่ง “r, c, f, g, h, i”

กับการทำ Merge และ Merge only actions

		Merge	Merge only
	x	1	0
1	r	$x=q+s$	$q+s$
2	c	$x=s+b$	$s+b$
3	f	$x=s+e$	$s+e$
4	g	$x=q+d$	$q+d$
5	h	$x=q+e$	$q+e$
6	i	$x=p+e$	$p+e$
7	r, f, h	$x=q+s+e$	$q+s+e$

การนำ 3-D scan mask ที่นำเสนอไปใช้ สามารถทำได้โดยสร้าง OR-decision table ที่มี $2^8=256$ กรณี ในลักษณะเดียวกับ OR-decision table ที่แสดงในตารางที่ 2 จากนั้นทำการแปลง OR-decision table ที่ได้ ให้เป็น Decision tree ด้วยวิธีการของ Sutheebanjard และ Premchaiswadi

[8] จะได้ Decision tree ที่มีความสูง 8 ระดับ และมีจำนวนโหนดใบ 36 โหนด เพื่อนำไปใช้ค้นหาพิกเซลที่ติดกันในภาพ 3 มิติ

4. ผลการทดลอง

ผู้วิจัยได้ทำการพัฒนา Decision Tree เพื่อใช้ในการค้นหาพิกเซลภาพที่ติดกันแบบ 3 มิติ ด้วยภาษา C++ โดยใช้ไลบรารี OpenCV และทำการคอมไพล์ด้วย gcc เวอร์ชัน 4.4.3 บนระบบปฏิบัติการ Ubuntu 10.04

ภาพที่ใช้ในการทดลองเป็นภาพไบนารี 2 มิติขนาด 32x32 พิกเซลจำนวน 40 ภาพจาก [9] แต่ละภาพมีความหนาแน่น (Density) ของ Foreground เป็น 0.1 0.2 0.3 และ 0.4 ดังแสดงในตารางที่ 4.

การทดลอง ได้นำภาพ 2 มิติ มาวางซ้อนทับกัน ได้เป็นภาพ 3 มิติ และทำการทดลองนับจำนวน Connected components ที่ความลึก (Depth) ของภาพ (แกน Z) ต่างๆกัน จำนวน Connected components ที่ได้ถูกต้องทั้งหมด ดังแสดงในตารางที่ 4 เช่น ภาพทดสอบที่มี Density 0.2 ความลึก 5 มีจำนวน Connected components เท่ากับ 19 และที่ความลึก 10 มีจำนวน Connected components เท่ากับ 22

ตารางที่ 4. ภาพที่ใช้ในการทดลองขนาด 32x32 พิกเซล และจำนวน Connected components ที่ระดับความลึกของภาพตั้งแต่ 1 - 10

Depth	Density							
	0.1		0.2		0.3		0.4	
		cc		cc		cc		cc
1		63		74		62		26
2		79		30		10		2
3		91		15		6		1
4		113		20		4		1
5		135		19		2		1
6		155		23		4		3
7		156		22		2		1
8		169		24		3		1
9		178		20		4		1
10		196		22		4		1

5. สรุป

บทความนี้นำเสนอวิธีการตรวจสอบหาพิกเซลภาพที่ติดกันในภาพ 3 มิติ ด้วยการนำเสนอ Scan mask แบบใหม่ที่ลดจำนวนตำแหน่งที่ทำการตรวจสอบจาก 14 ตำแหน่ง เหลือเพียง 8 ตำแหน่ง ทำให้เมื่อแปลง

เป็น Decision tree ความสูงของ Decision tree จะลดลงจาก 14 ระดับ เหลือเพียง 8 ระดับ เท่านั้น

เอกสารอ้างอิง

- [1] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," J. ACM, vol. 13, no. 4, pp. 471-494, 1966.
- [2] K. Wu, E. Otoo and A. Shoshani, "Optimizing Connected Component Labeling Algorithms," Proceedings of the SPIE, Volume 5747, pp. 1965-1976, 2005.
- [3] C. Grana, D. Borghesani and R. Cucchiara, "Optimized block-based connected components labeling with decision trees," IEEE Transactions on Image Processing, vol 19, issue 6, pp. 1596-1609, June 2010.
- [4] P. Sutheebanjard and W. Premchaiswadi, "Efficient Scan Mask Techniques for Connected Components Labeling Algorithm," EURASIP Journal on Image and Video Processing, (In press), 2011.
- [5] R. Lumia, L. Shapiro, and O. Zungia, "A new connected components algorithm for virtual memory computers," Computer Vision, Graphics, and Image Processing, 22(2):287-300, 1983.
- [6] Y. Shirai. "Labeling connected regions. In Three-Dimensional," Computer Vision, pp. 86-89. Springer-Verlag, 1987.
- [7] L. He, Y. Chao, and K. Suzuki, "Two Efficient Label-Equivalence-Based Connected-Component Labeling Algorithms for Three-Dimensional Binary Images," IEEE Transection on Image Processing, vol. 20 issue 8, pp. 2122-2134, 2010.
- [8] P. Sutheebanjard and W. Premchaiswadi, "Fast Convert OR-Decision Table to Decision Tree," IEEE ICT&KE 8th, pp. 37-40, Nov. 2010.
- [9] University of Modena and Reggio Emilia, Modena, Italy, "Labeling Image Lab: an impressively fast labeling routine for Open," 2010. Available: <http://imagelab.ing.unimore.it/imagelab/labeling.asp>



ไพศาล สุธีบรรเจิด อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ และกำลังศึกษาระดับปริญญาเอก บัณฑิตวิทยาลัย สาขาวิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยสยาม งานวิจัยที่เกี่ยวข้อง Image Processing, Natural Language Processing, Stock Forecasting, Evolution Algorithm



รศ.ดร.วิเชียร เปรมชัยสวัสดิ์ คณบดีบัณฑิตวิทยาลัยสาขาเทคโนโลยีสารสนเทศและ ผู้ช่วยอธิการบดี มหาวิทยาลัยสยาม งานวิจัยที่เกี่ยวข้อง Database Management System, Knowledge Management, Image Processing, Parallel Computing