บทที่ 4

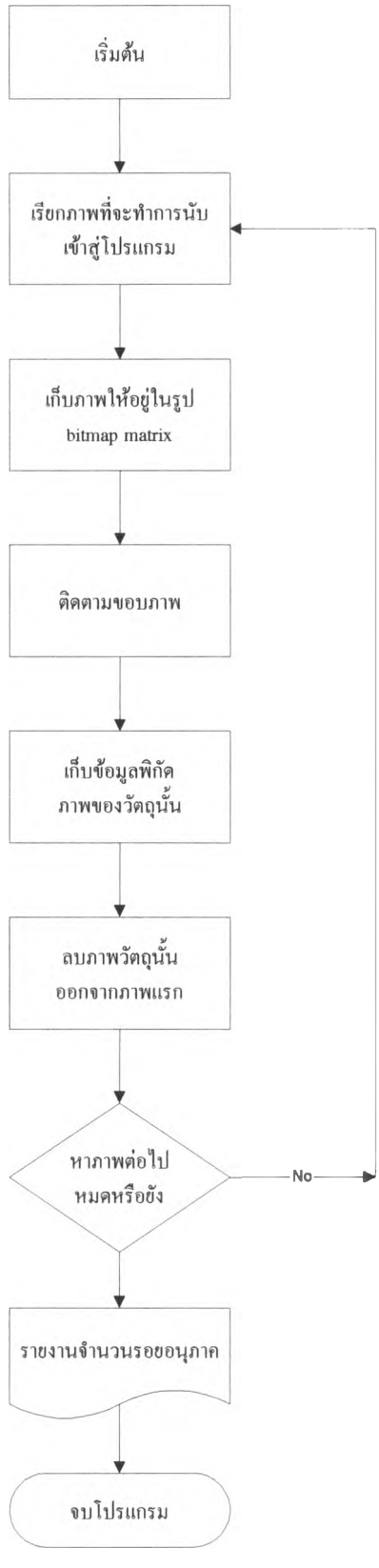
การสร้างโปรแกรมนับรอยอนุภาคแอลฟา

4.1 โปรแกรมนับรอยอนุภาคแอลฟาบนฟิล์มเซลลูโลสไนเตรต

โปรแกรมนับภาพรอยอนุภาคแอลฟาถูกพัฒนาขึ้นโดยใช้ภาษา Visual Basic version 4 โดย พัฒนาขึ้นบนระบบปฏิบัติงาน Microsoft Windows version 3.11 แต่เมื่อทำการ คอมไพล์เป็น Excecute File แล้วสามารถทำงานได้บนระบบปฏิบัติการ Microsoft Windowsทั้ง Windows 3.11 และ Windows 95 โปรแกรมหลักถูกเขียนด้วย Visual Basic แต่ในส่วนการแปลงภาพให้อยู่ในรูป bitmap matrix และการติดตามขอบภาพ (edge contour) จะเขียนด้วยภาษาซี เพื่อให้การทำงานรวดเร็ว โดยสร้างเป็น Application Programming Interface (API) โดยมีชื่อไฟล์ dll1.dll ในส่วนของภาษาซีใช้ Borland C++ version 4.5 เป็นตัวคอมไพล์

การสร้างโปรแกรมนับภาพสามารถที่จะแบ่งเป็นขั้นตอนที่สำคัญดังนี้

1. นำภาพที่ผ่านการประมวลผลภาพในบทที่3 ทำการจัดเก็บข้อมูลของภาพให้อยู่ในรูปแบบ bit image matrix เพื่อติดตามขอบภาพ
2. ทำการติดตามขอบภาพ (edge contour following)
3. เก็บข้อมูลรูปวัตถุรูปแรกจากการติดตามขอบภาพไปแล้ว
4. นับรูปวัตถุนั้นเป็นหนึ่งรูป
5. ลบรูปของวัตถุที่ผ่านการเก็บข้อมูลไปแล้ว ออกจากภาพเริ่มต้น
6. นำข้อมูลของวัตถุแรกในข้อ 3 ไปวิเคราะห์เพื่อหาขนาดของรอยอนุภาค
7. เริ่มขั้นตอนที่ 2. ใหม่อีกครั้งจนกว่ารูปของวัตถุในภาพทั้งหมดจะหมดไป
8. รายงานจำนวนวัตถุทั้งหมดในภาพที่นับได้ และ จำนวนรอยอนุภาคที่ต้องการนับ
9. เสร็จสิ้นขบวนการนับภาพ



รูปที่ 4.1 แสดงแผนผังการทำงานของโปรแกรมนับรอยอนุภาคแอลฟา

ในขั้นตอนที่กล่าวมาทั้งหมด ได้นำมาสร้างโปรแกรมโดยมีชื่อของ project คือ project.vbp ซึ่งประกอบด้วย 7 ฟอร์มที่สำคัญคือ

ฟอร์ม Form1 เป็นฟอร์มเริ่มต้นโปรแกรม

ฟอร์ม Bincont1 เป็นฟอร์ม Binary contrast ทำการแปลงภาพเป็นขาว-ดำ

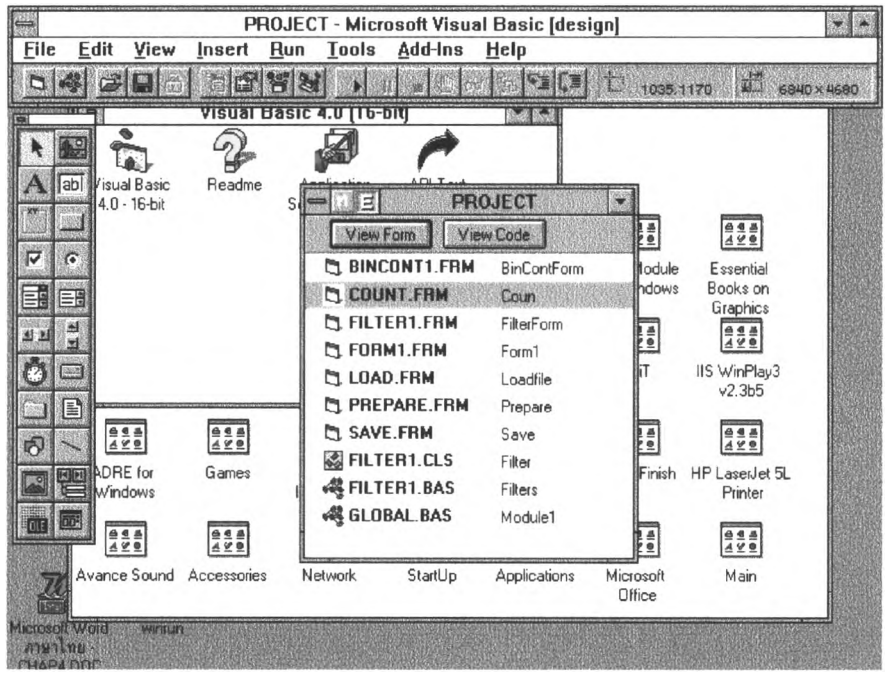
ฟอร์ม Filter1 เป็นฟอร์มตกแต่งภาพโดยใช้การ dilation และ dilation outlinning

ฟอร์ม prepare เป็นฟอร์มเตรียมการนับภาพ

ฟอร์ม load มีหน้าที่เรียกภาพที่จะทำการนับเข้าสู่โปรแกรม

ฟอร์ม save มีหน้าที่จัดเก็บภาพที่จะทำการนับ

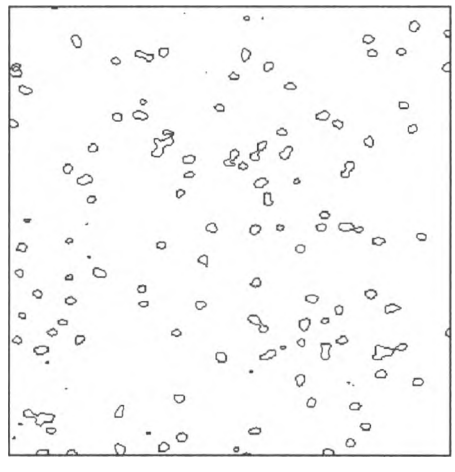
ฟอร์ม count มีหน้าที่นับภาพ และ แสดงผลการนับภาพ



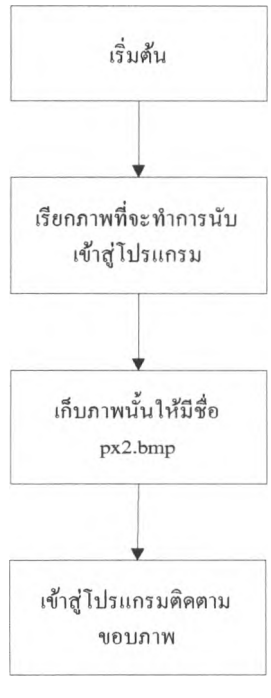
รูปที่ 4.2 รูปแสดง project.vbp

ฟอร์มที่เกี่ยวข้องกับการนับภาพคือฟอร์ม prepare, load, save และ count ฟอร์ม prepare เป็นการเตรียมการนับภาพโดยมีการทำงานดังนี้ ฟอร์ม load จะนำข้อมูลภาพที่จัดเก็บไว้หลังจากการทำ image processing แล้ว เข้าสู่โปรแกรมนับรอยอนุภาค ภาพที่จะถูกนำเข้ามาควรเป็นภาพแบบสองระดับ (ภาพขาวและดำ) เป็นภาพที่มีขอบภาพที่ต่อเนื่องและมีเฉพาะขอบภาพเท่านั้น ตามรูปที่ 4.3 ซึ่งภาพที่มีลักษณะดังนี้จะถูกเรียกว่า "ภาพสมบูรณ์" เมื่อนำภาพเข้ามาใน

โปรแกรมแล้ว ในโปรแกรมจะมีอุปกรณ์ช่วยในการตกแต่งภาพเล็กน้อย คือสามารถวาด หรือลบรูปได้ สามารถวัดขนาดของเส้นผ่าศูนย์กลางและความยาวของเส้นรอบวง ซึ่งจะมีประโยชน์มากในการกำหนดของขนาดของรอยอนุภาคที่ต้องการให้โปรแกรมนับ หลังจากที่เราได้ภาพที่ต้องการแล้ว ก็จะเข้าสู่ขั้นตอนของการนับภาพ โดยที่ฟอร์ม prepare จะเก็บภาพแบบอัตโนมัติให้มีชื่อเป็น px2.bmp เพื่อจะได้นำไปทำการ ติดตามขอบภาพต่อไปได้

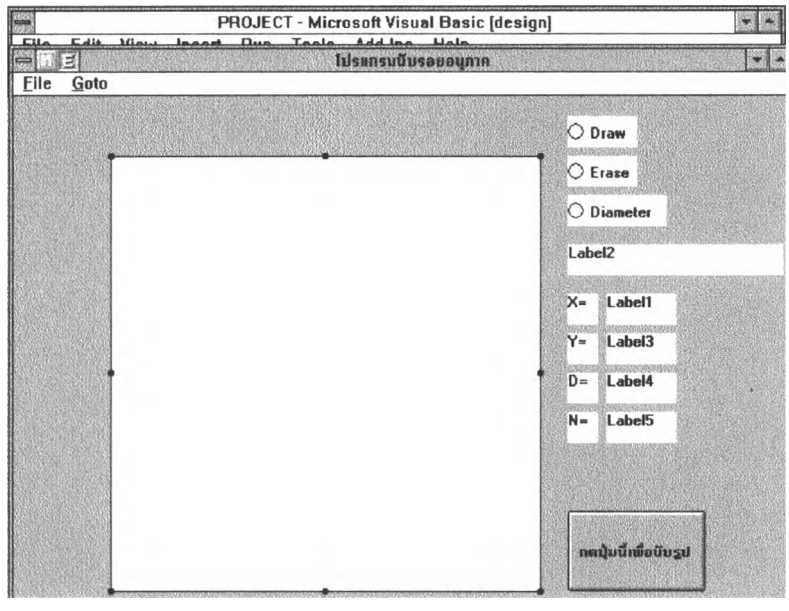


รูปที่ 4.3 รูปรอยอนุภาคแอลฟาที่ขอบภาพต่อเนื่องและมีเฉพาะขอบภาพเท่านั้น

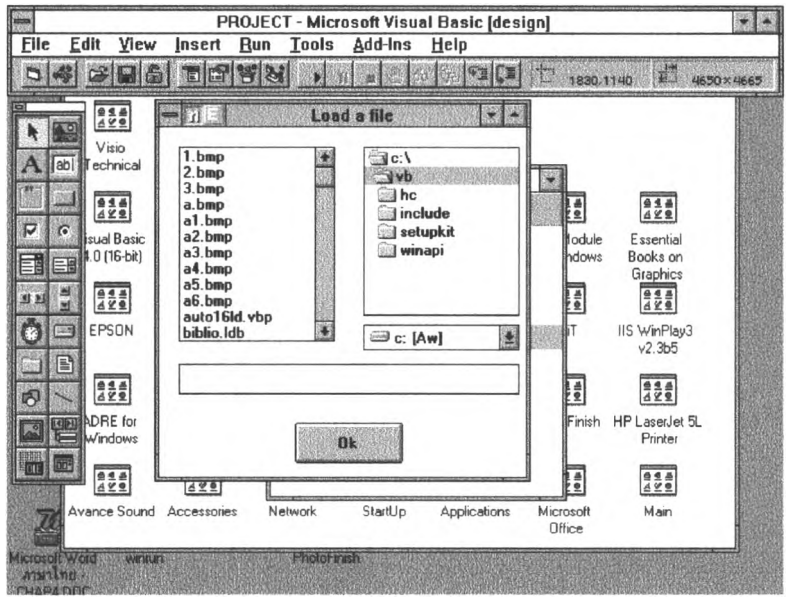


14005044

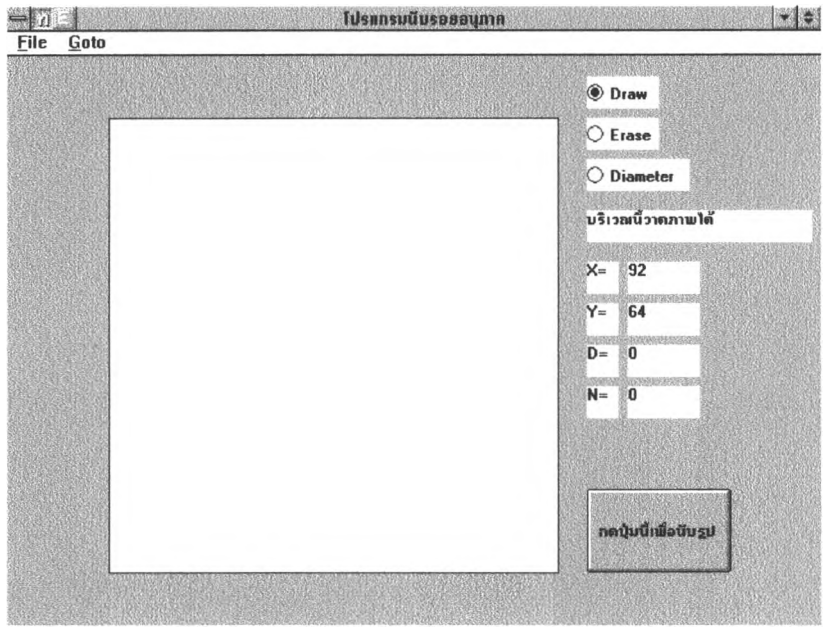
รูปที่ 4.4 แผนผังการทำงานของฟอร์ม prepare



รูปที่ 4.5 ฟอร์ม prepare ระหว่างการพัฒนา



รูปที่ 4.6 ฟอร์ม load ระหว่างการพัฒนา



รูปที่ 4.7 ฟอร์ม prepare เมื่อถูกเรียกเข้าใช้งาน

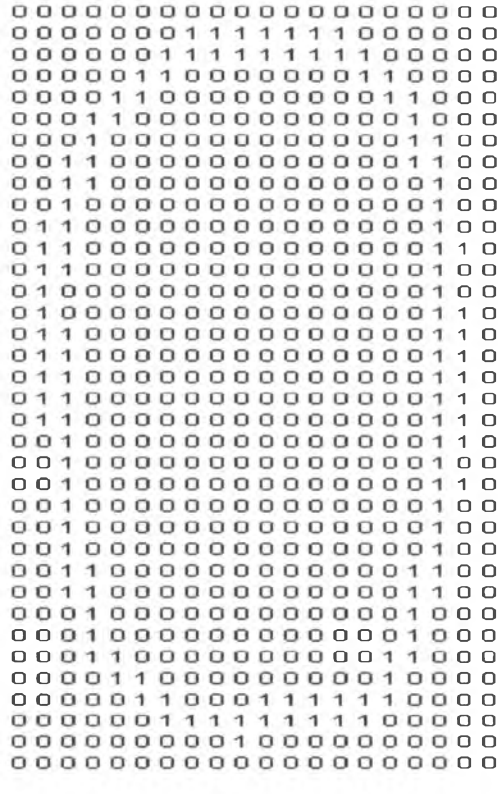
ฟอร์ม count จะทำการติดตามขอบภาพของวัตถุ (edge contour following) ซึ่งเป็นขั้นตอนที่สำคัญขั้นตอนหนึ่งของโปรแกรมนี้ โดยจะเริ่มจากการอ่านภาพที่ถูกเก็บแบบอัตโนมัติจากฟอร์ม prepare โดยภาพที่ถูกเรียกเข้ามาจะมีชื่อไฟล์ว่า px2.bmp หลังจากนั้นฟอร์ม count จะเรียก Application Programming Inteface dll1.dll เข้าใช้งาน ซึ่งใน dynamic link libraly dll1.dll จะมีโปรแกรมย่อยที่สำคัญสองโปรแกรมคือ readbmp.c และ edge.cโปรแกรม readbmp.c จะทำการเรียกภาพที่ชื่อ px2.bmp แล้วทำการเปลี่ยนให้กลายเป็นภาพแบบ bitmap matrix โดยข้อมูลของจุดภาพจะถูกเก็บอยู่ในตัวแปรที่ชื่อ mat_in[i][j] และข้อมูลภายในmat_in[i][j] จะเป็นค่าเป็น 0 หรือ 1 โดย 0 หมายถึงจุดสีขาวในภาพ, 1 หมายถึงจุดสีดำในภาพ ดังรูปที่ 4.8

เมื่อเข้าสู่ขั้นตอนติดตามขอบภาพ โปรแกรมย่อย edge.c ทำการตรวจสอบหาจุดสีดำจุดแรก ของภาพ โดยเริ่มต้นจากขอบล่างของภาพ จากซ้ายไปขวา เมื่อเจอก็จะทำการเก็บค่าพิกัดของจุดนั้นโดยโปรแกรมย่อย def_edge(i,j) ทำการหาจุดที่อยู่ถัดไปทวนเข็มนาฬิกาโดยเริ่มที่จุดด้านซ้ายก่อนแล้ว วนไป ล่างซ้าย-> ล่าง-> ล่างขวา-> ขวา-> บนขวา-> บน-> บนซ้าย จนล้อมรอบจุดแรกซึ่งจุดที่ล้อมรอบจุดแรกจะมี 8 จุดด้วยกันตามรูปที่ 4.9 เมื่อพบจุดสีดำก็จะเก็บค่าพิกัด

ของจุดนั้นแล้วหาจุดต่อไปเรื่อยๆจนวนมาที่จุดเดิมก็จะเสร็จสิ้นการหาขอบภาพของรูปแรก ขั้นตอนการทำการของโปรแกรมย่อย edge.c สามารถดูได้จากรูปที่ 4.12 ถึงรูปที่ 4.21

เมื่อทำการลบรูปแรกทิ้งไปแล้วเก็บภาพทั้งหมดใหม่ ให้มีชื่อเดิมคือ px2.bmp เรียกภาพเข้าสู่ โปรแกรมอีกครั้ง จัดเก็บภาพให้อยู่ในรูป bitmap matrix อีกครั้ง ทำการหาขอบรูปของรูปที่สอง เมื่อครบก็ลบรูปทิ้งจากรูปตั้งต้น เก็บภาพใหม่ให้ชื่อ px2.bmp ทำภาพให้อยู่ในรูปของ bitmap matrix อีกครั้ง แล้วเริ่มหาขอบภาพของรูปที่สาม ทำเช่นนี้ไปเรื่อยๆจากล่างขึ้นบน จนกระทั่งทั้งภาพไม่มีจุดสีดำหลงเหลืออยู่เลยก็จะหยุด

ค่าพิกัดของรูปแต่ละรูปจะเก็บอยู่ในไฟล์ data.txt เมื่อเสร็จการหาขอบภาพของรูปแต่ละรูปไฟล์ค่าพิกัดของรูปนั้นก็จะเปลี่ยนแปลงไป จนกระทั่งไม่รูปเหลืออยู่เลย ไฟล์พิกัดจะมีค่าเป็น 0 ซึ่งจะใช้เป็นตัวแปรให้โปรแกรมหยุดนับ ไฟล์พิกัดจะมีความสำคัญต่อการวิเคราะห์รูปแบบของรูปนั้นๆ เพื่อนำไปแยกแยะรูปที่ต้องการนับออกจากรูปที่ไม่ต้องการนับ

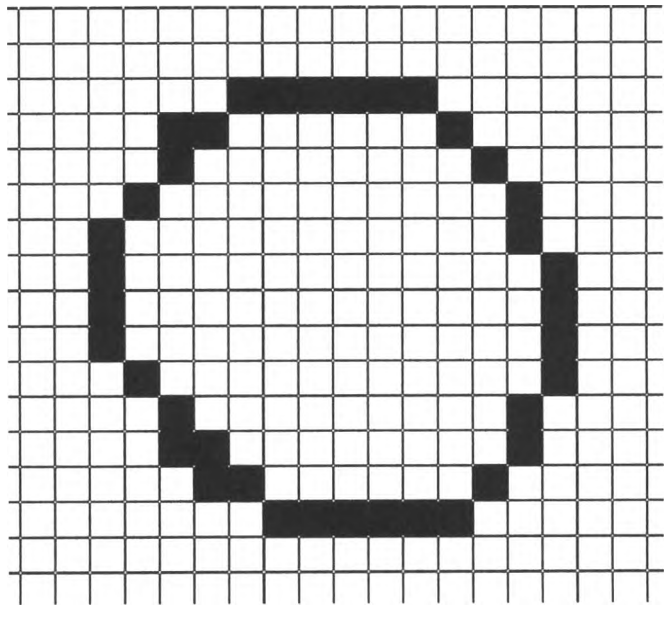


รูปที่ 4.8 แสดงการเก็บข้อมูลภาพแบบ bitmap matrix

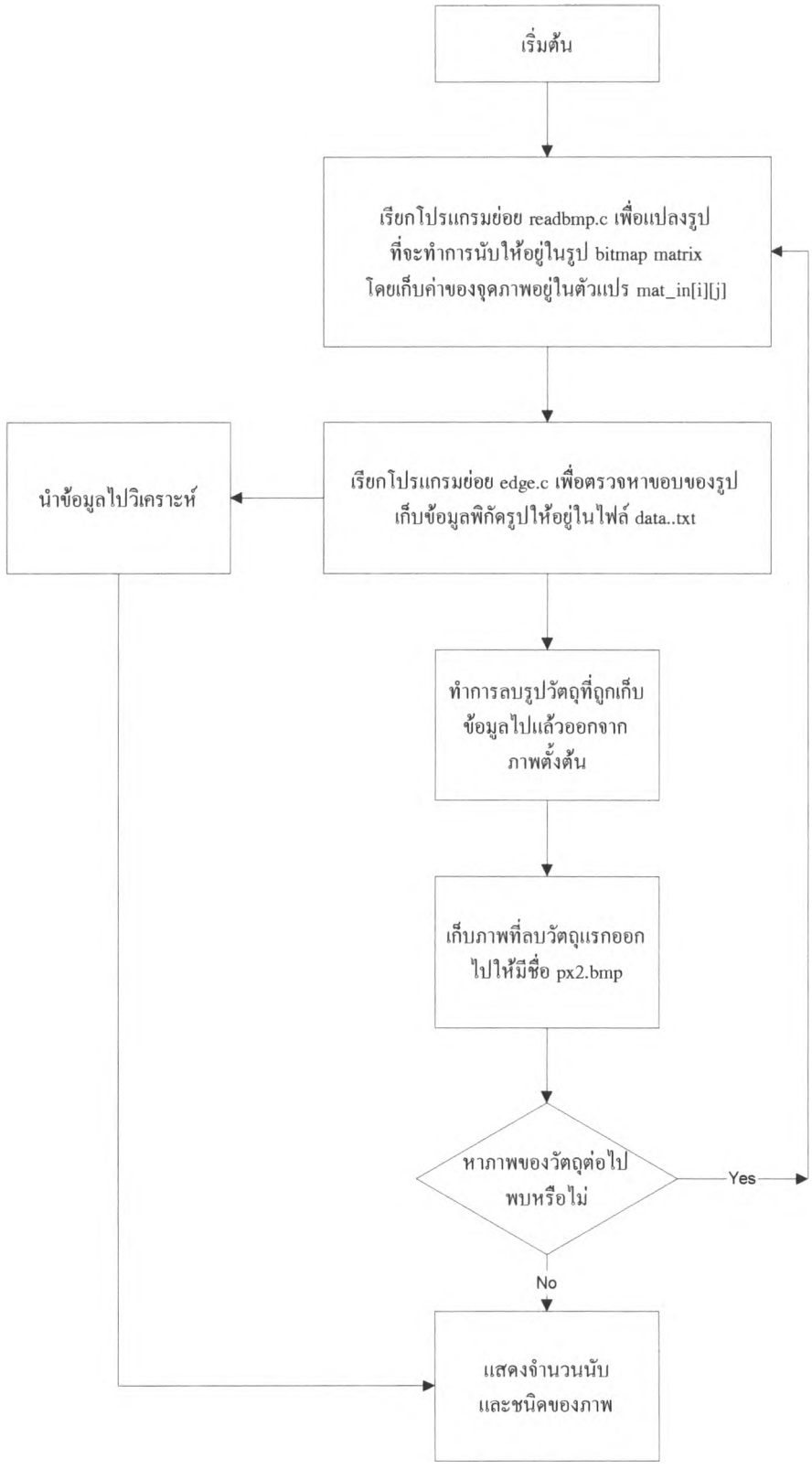
ตัวเลข 1 จะแทนจุดสีดำ เลข 0 จะแทนจุดสีขาว

i+1,j-1	i+1,j	i+1,j+1
i,j-1	i,j	i,j+1
i-1,j-1	i-1,j	i-1,j+1

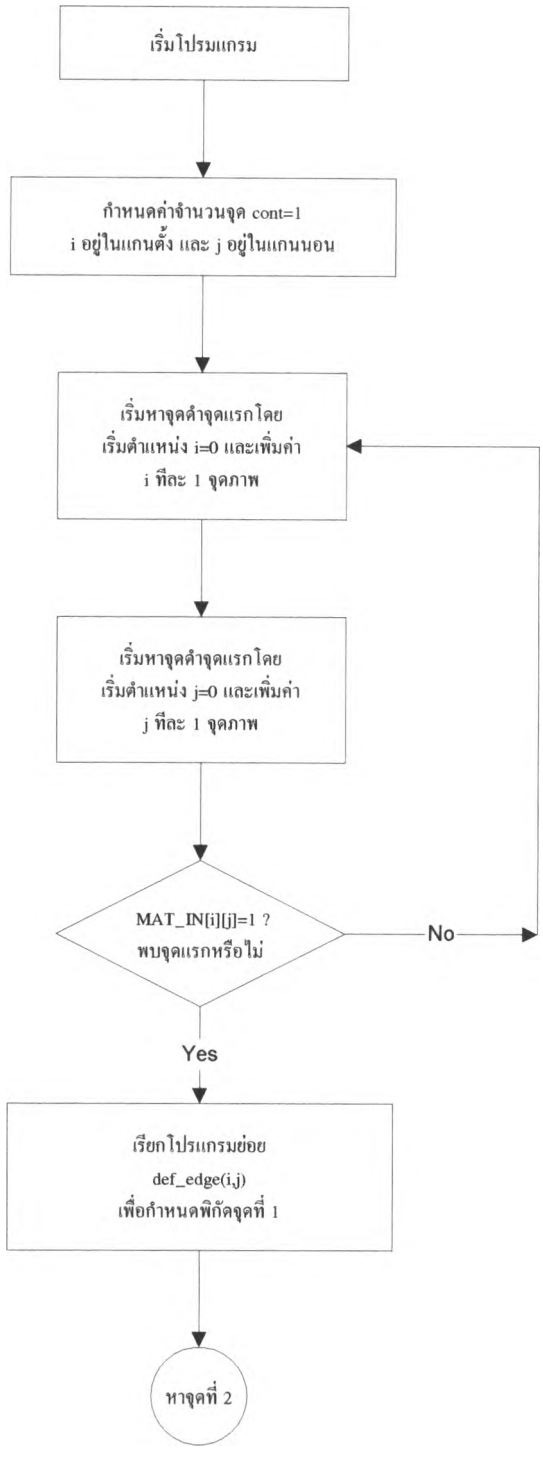
รูปที่ 4.9 แสดงตำแหน่งจุดภาพที่อยู่ติดกัน



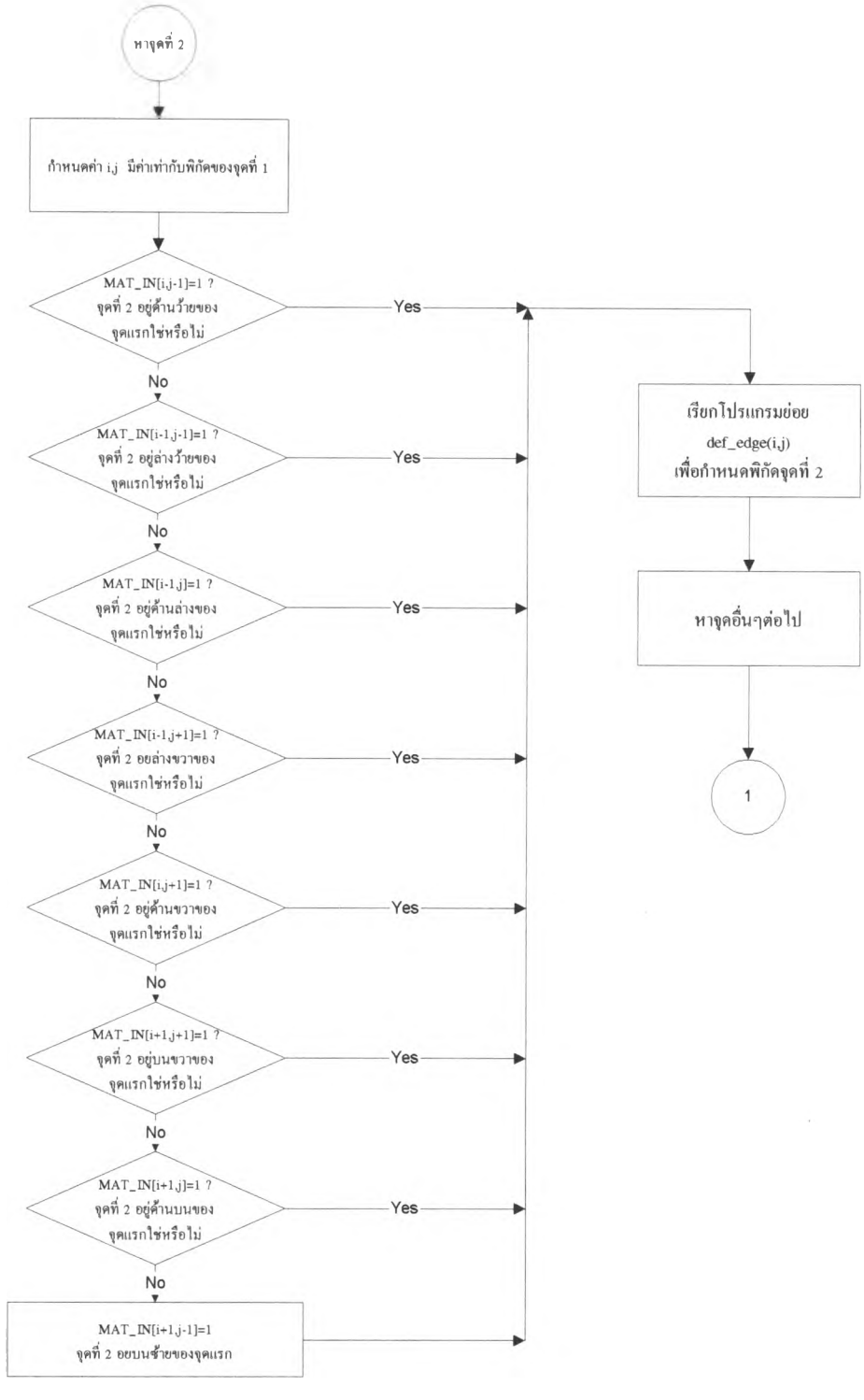
รูปที่ 4.10 รูปภาพแบบสองระดับ(ขาว-ดำ) ที่มีขอบภาพต่อเนื่อง



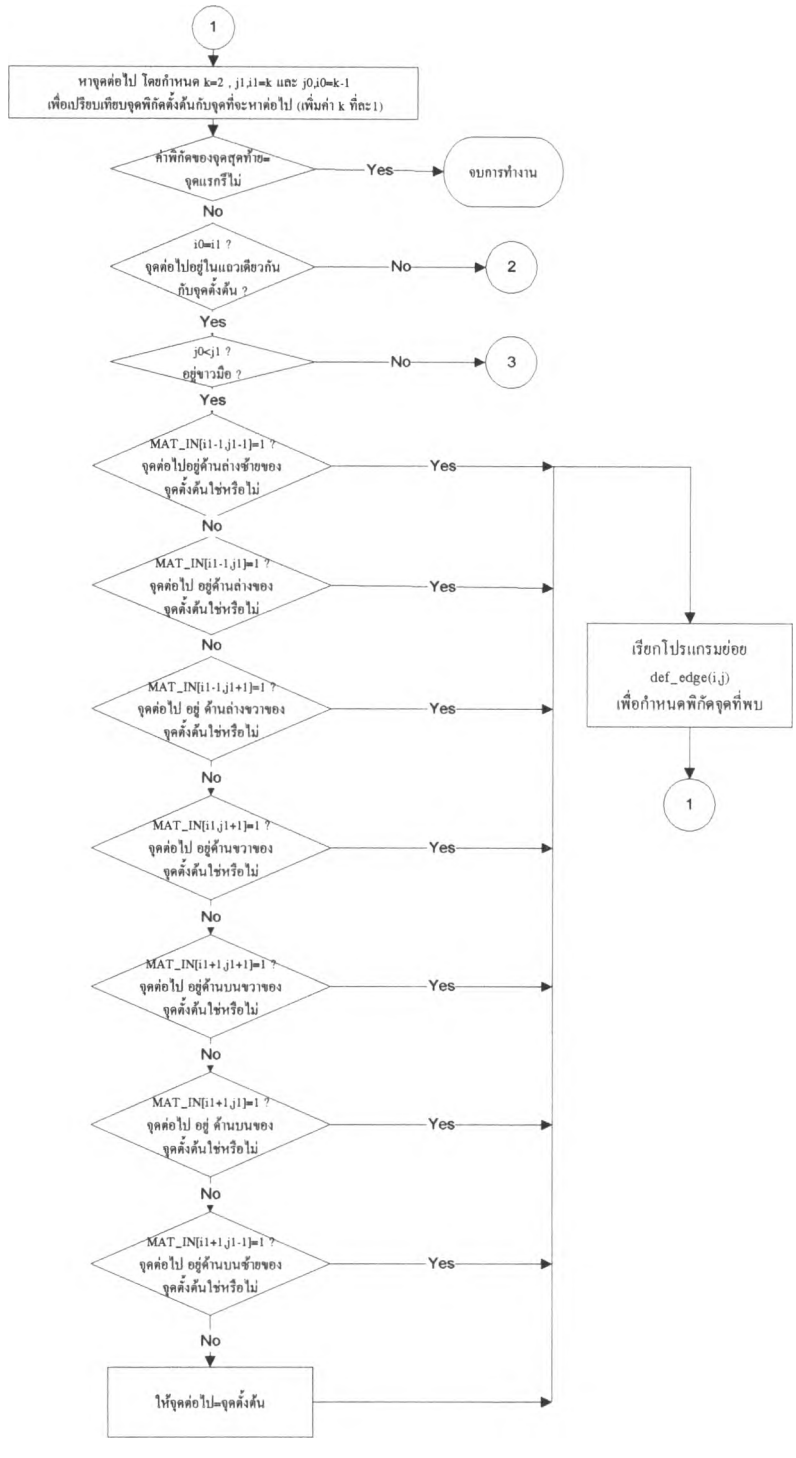
รูปที่ 4.11 แผนผังการทำงานของโปรแกรมหาขอบภาพและนับภาพ



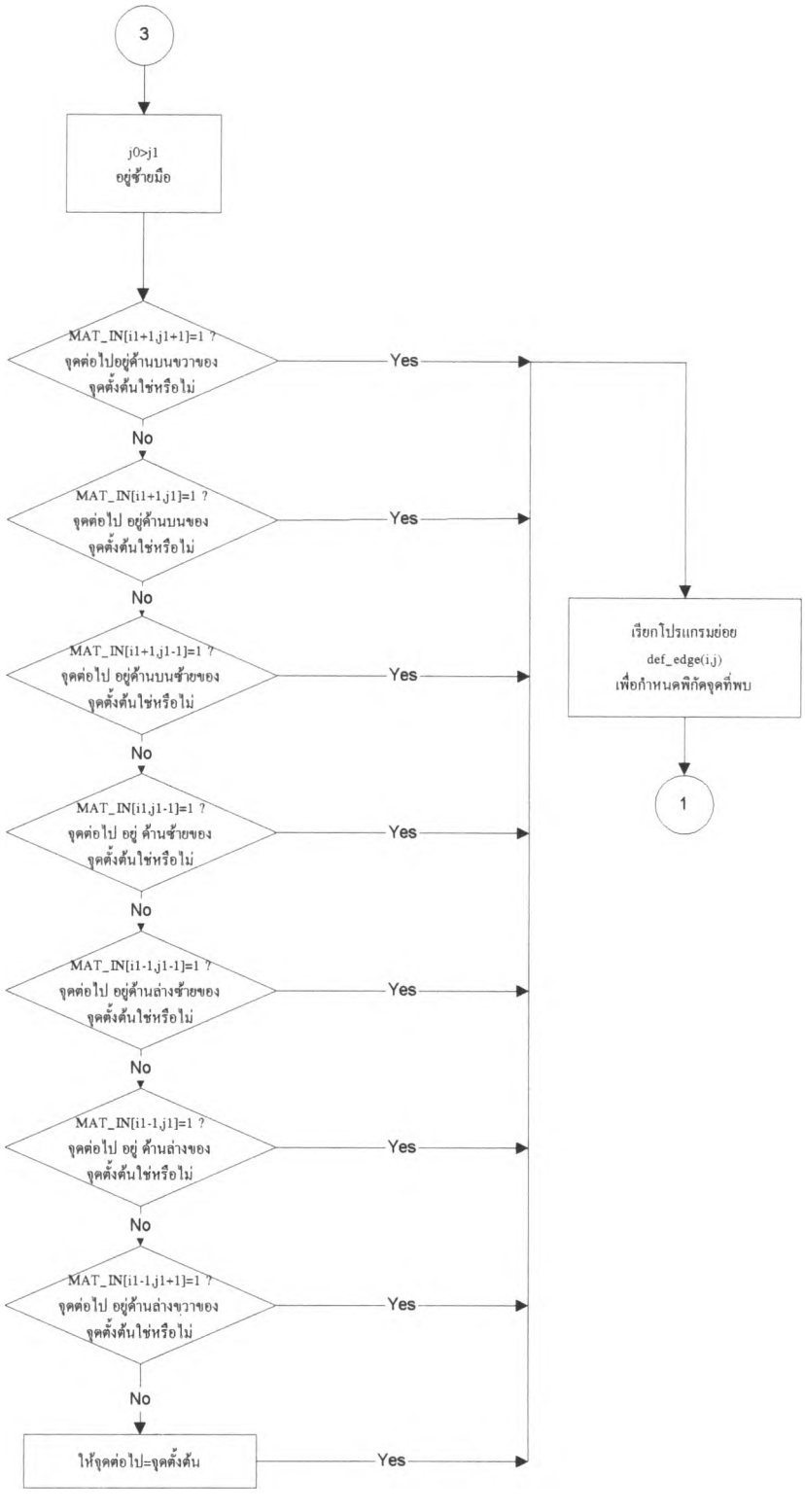
รูปที่ 4.12 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดแรก



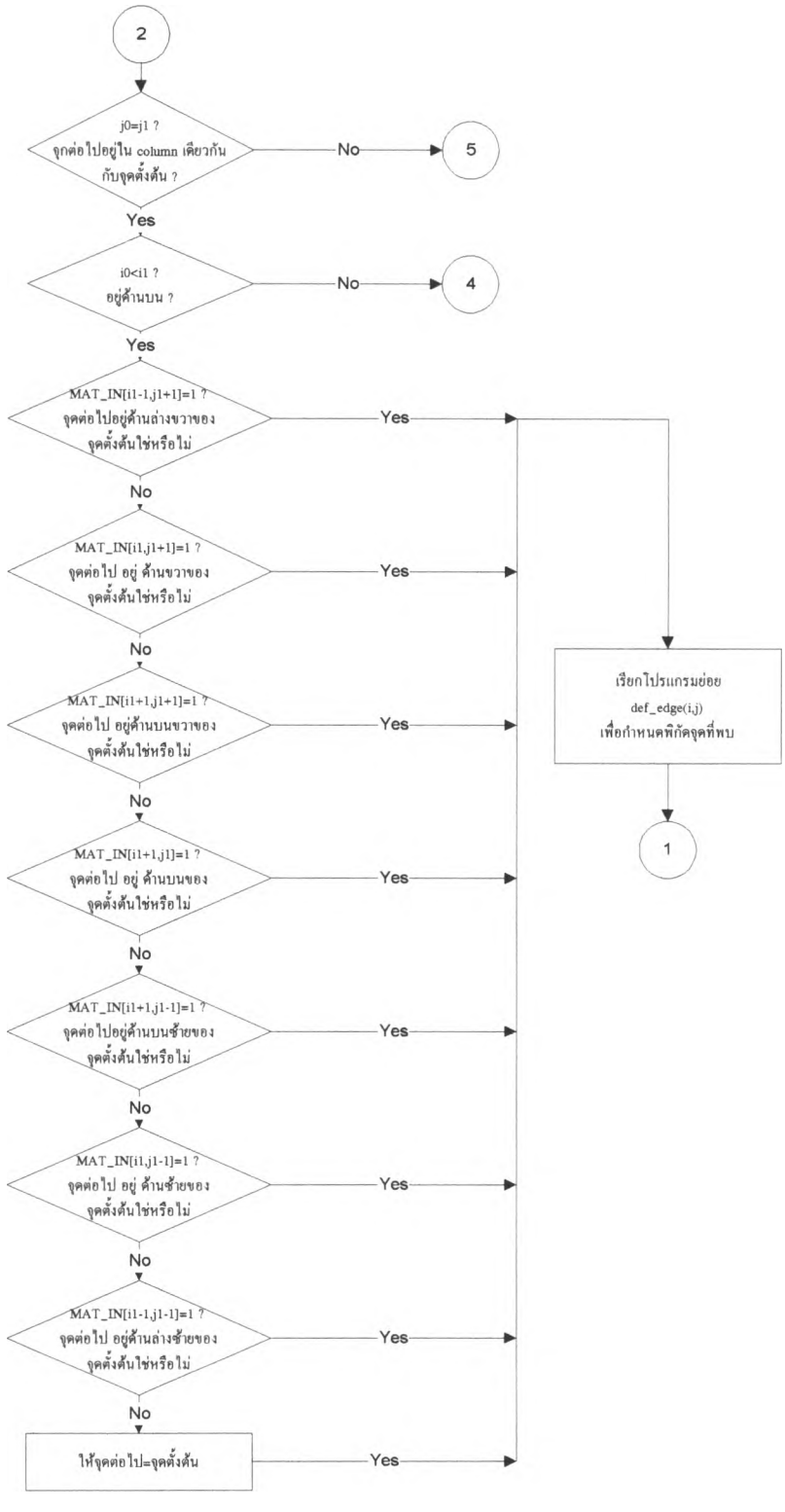
รูปที่ 4.13 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดที่สอง



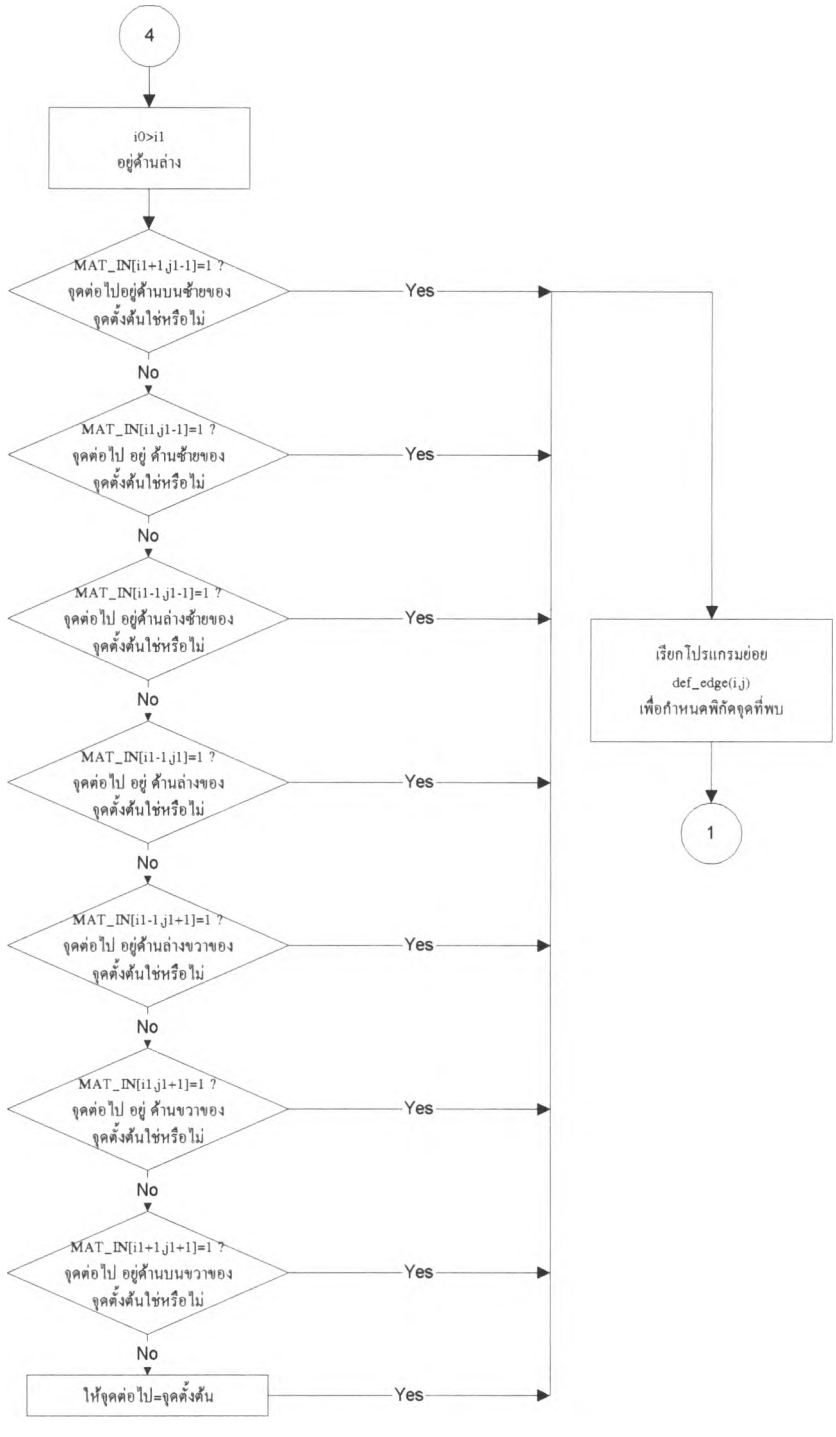
รูปที่ 4.14 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดที่สามและจุดต่อไป ในกรณีที่จุดที่สองอยู่ด้านขวาของจุดแรก



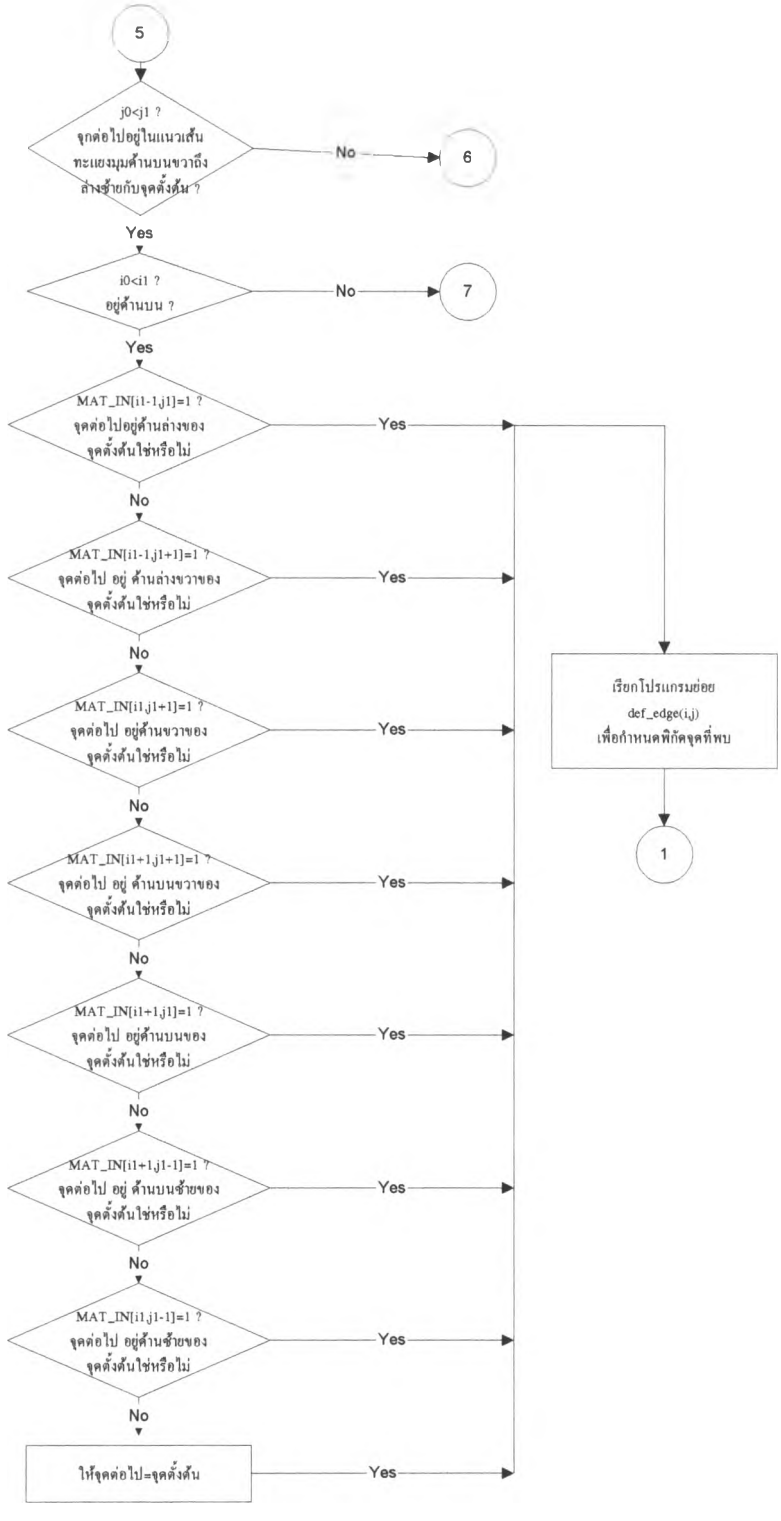
รูปที่ 4.15 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดที่สามและจุดต่อไป ในกรณีที่จุดที่สองอยู่ด้านซ้ายของจุดแรก



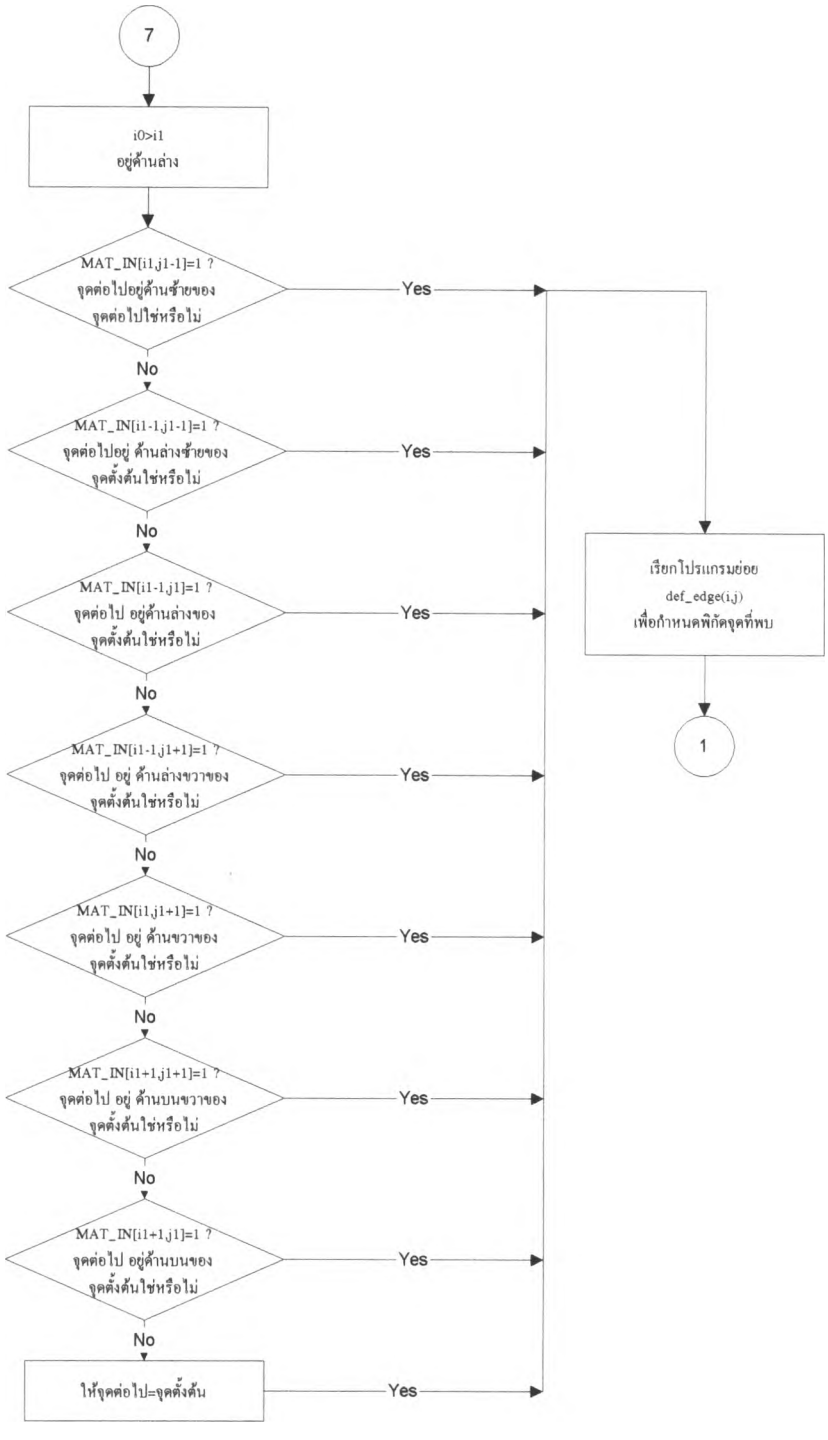
รูปที่ 4.16 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดที่สามและจุดต่อไป ในกรณีที่จุดที่สองอยู่ด้านบนของจุดแรก



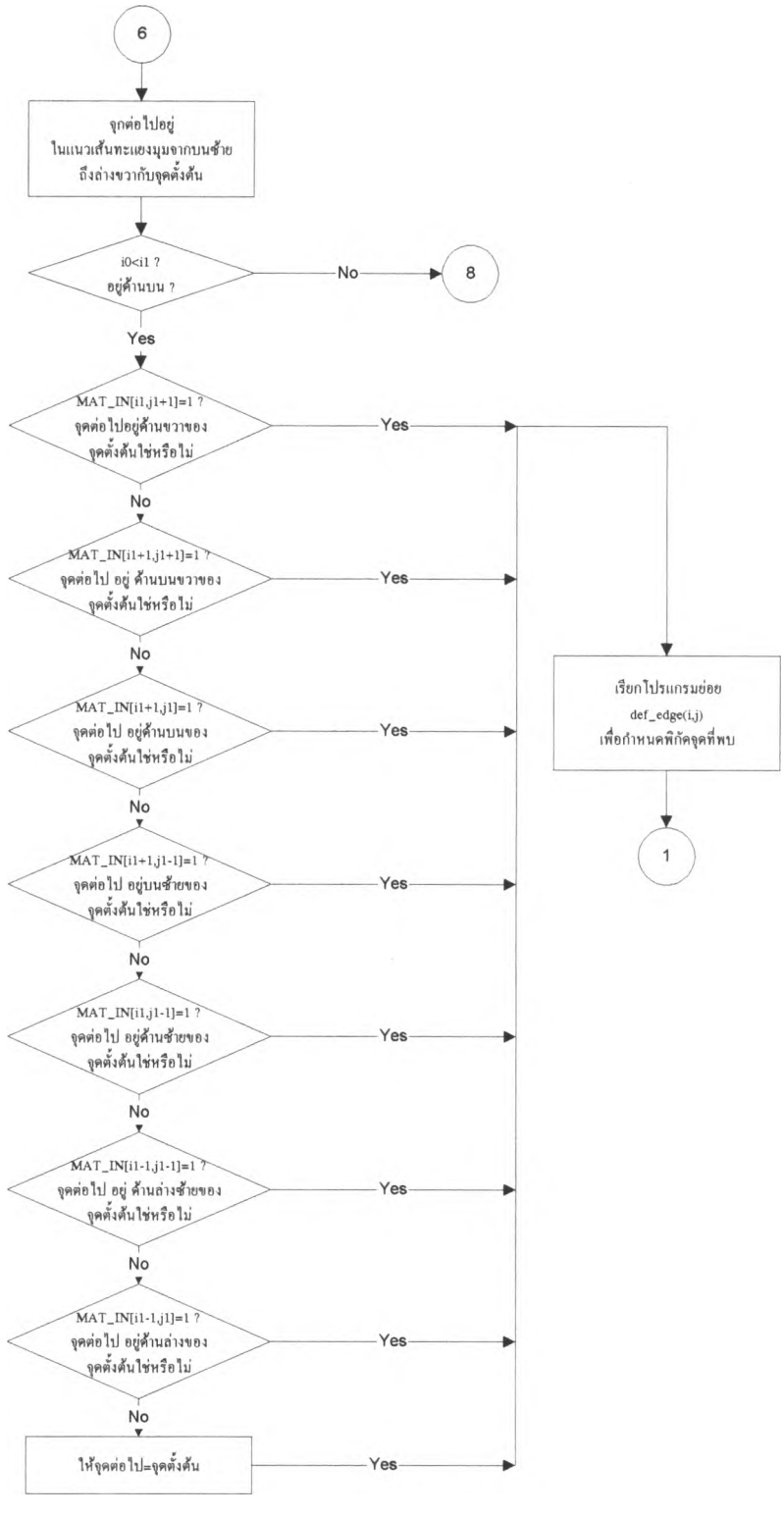
รูปที่ 4.17 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดที่สามและจุดต่อไป ในกรณีที่จุดที่สองอยู่ด้านล่างของจุดแรก



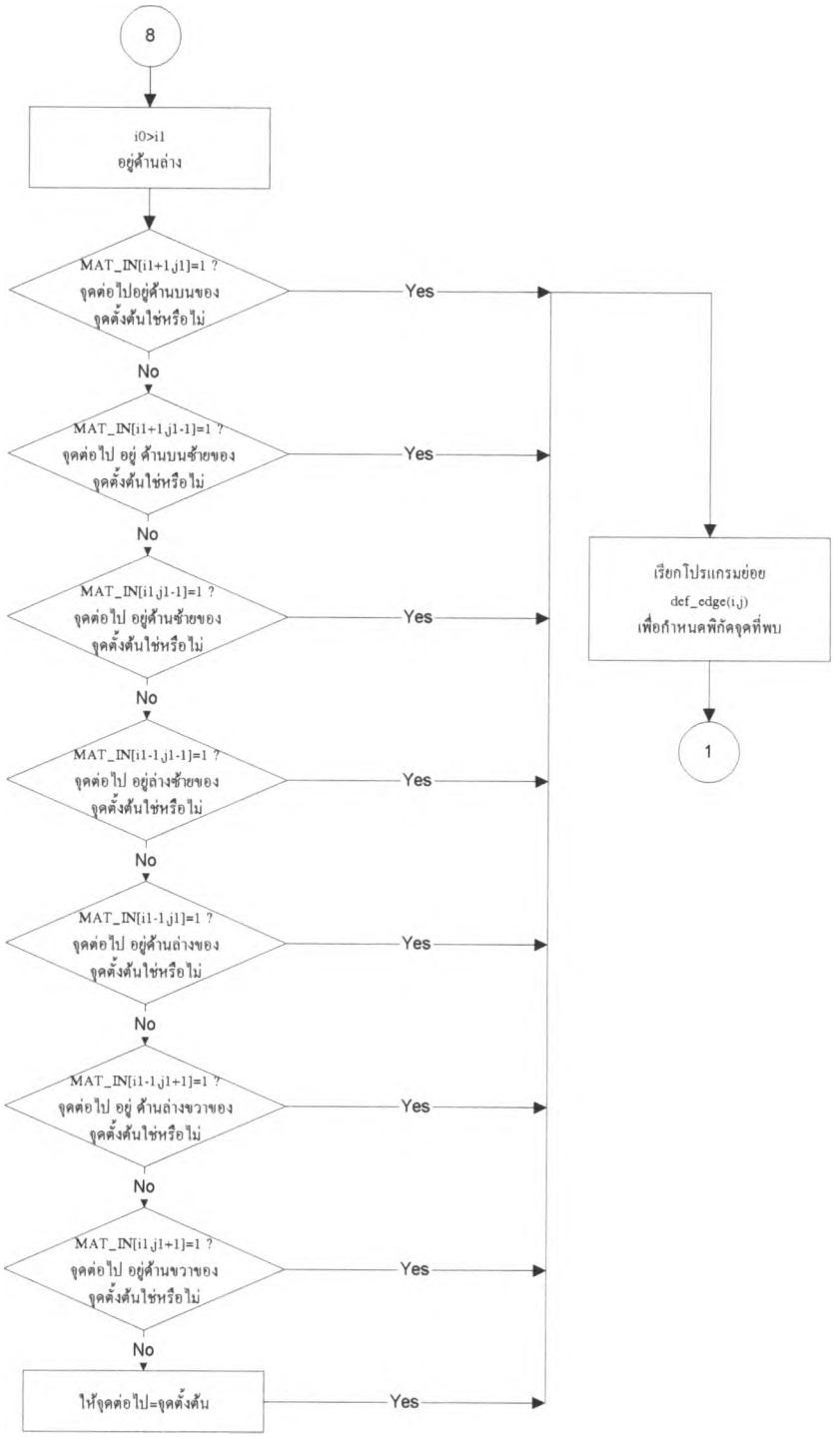
รูปที่ 4.18 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดที่สามและจุดต่อไป ในกรณีที่จุดที่สองอยู่ด้านบนขวาของจุดแรก



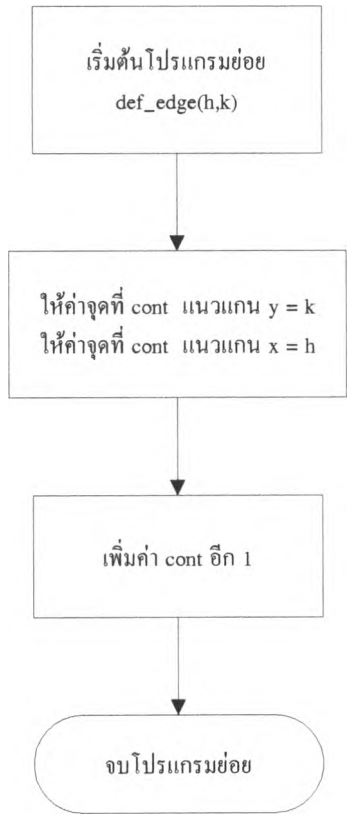
รูปที่ 4.19 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดที่สามและจุดต่อไป ในกรณีแรกจุดที่สองอยู่ด้านล่างขวาของจุดแรก



รูปที่ 4.20 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดที่สามและจุดต่อไป ในกรณีที่จุดที่สองอยู่ด้านบนซ้ายของจุดแรก

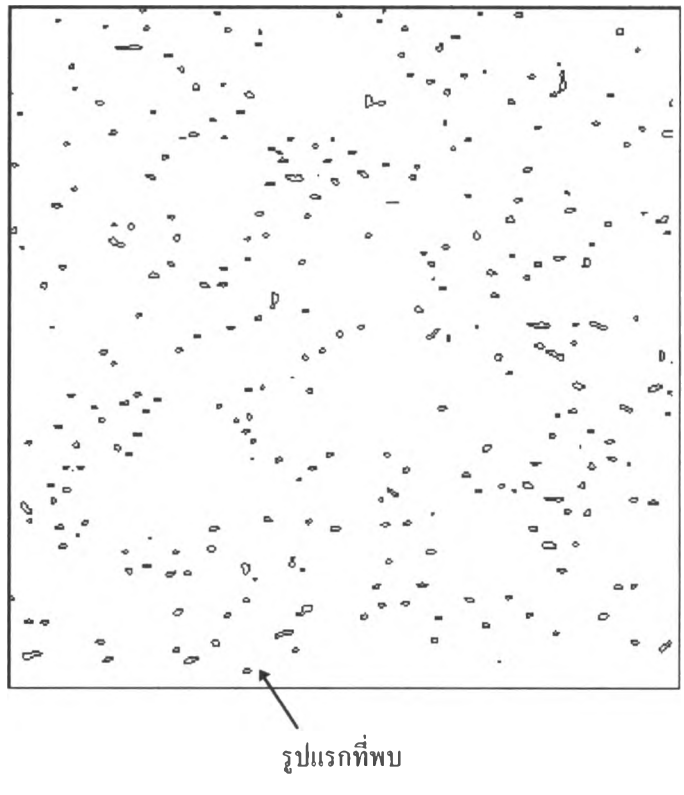


รูปที่ 4.21 แสดงผังการทำงานของโปรแกรมติดตามขอบภาพเพื่อหาจุดที่สามและจุดต่อไป ในกรณีที่จุดที่สองอยู่ด้านล่างซ้ายของจุดแรก



รูปที่ 4.22 แผนผังการทำงานของโปรแกรมย่อย def_edge(i,j)

การวิเคราะห์รูปรอยอนุภาคเพื่อแยกแยะความแตกต่างระหว่างรอยอนุภาคกับจุดภาพเล็กๆ สามารถทำได้โดยการใช้พิกัดของจุดภาพของรูปนั้นๆ ข้อมูลของจุดภาพจะถูกเก็บอยู่ในไฟล์ที่ชื่อ data.txt โดยที่ฟิลด์แรกจะเก็บข้อมูลของจำนวนจุดภาพของรูป ฟิลด์ที่สองจะเก็บพิกัดภาพในแนวตั้งของจุดแรกที่พบ ฟิลด์ที่สามจะเก็บพิกัดภาพในแนวนอนของจุดแรกที่พบ ฟิลด์ที่สี่จะเก็บพิกัดภาพในแนวตั้งของจุดที่สอง เช่นนี้ไปเรื่อยๆจนฟิลด์สุดท้ายจะเก็บข้อมูลของพิกัดภาพในแนวนอนของจุดสุดท้าย ตัวอย่างพิกัดภาพของรูปรอยอนุภาครูปหนึ่งจะเป็นดังรูปที่ 4.23



รูปที่ 4.23 รูปรอยอนุภาคแรกที่โปรแกรมนับรอยพบและค่าพิกัดที่ถูกเก็บ

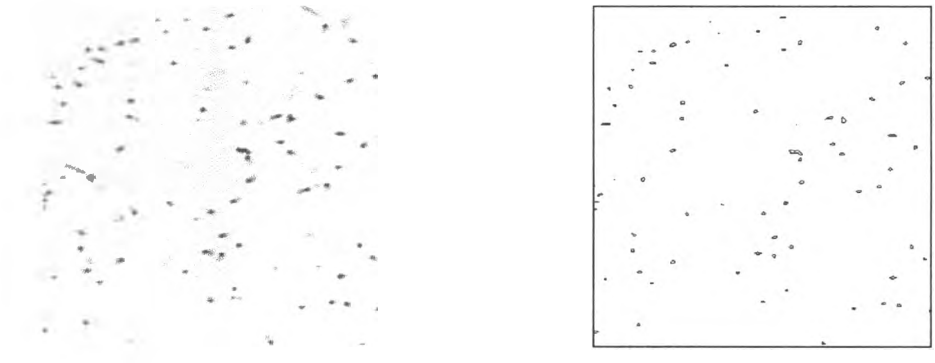
ข้อมูลของพิกัดภาพสามารถนำไปหาขนาดของรอยขนาดเล็กและใหญ่ได้ เช่นถ้ารอยขนาดใหญ่จะมีจำนวนจุดภาพมากกว่ารอยขนาดเล็ก ดังนั้นจึงหาความแตกต่างระหว่างรอยอนุภาคใหญ่ กับ เล็กได้

เมื่อจบการนับรูปวัตถุอันแรกไปแล้วก็จะทำการนับรูปวัตถุต่อไปไปเรื่อยจนในภาพไม่มีจุดสีดำเหลืออยู่เลย หลังจากนั้นก็จะทำการรวบรวมข้อมูลจำนวนรอยอนุภาคที่นับได้ พร้อมทั้งบอกจำนวนของรอยตามขนาดที่กำหนด ก็จะจบการทำงานของโปรแกรม

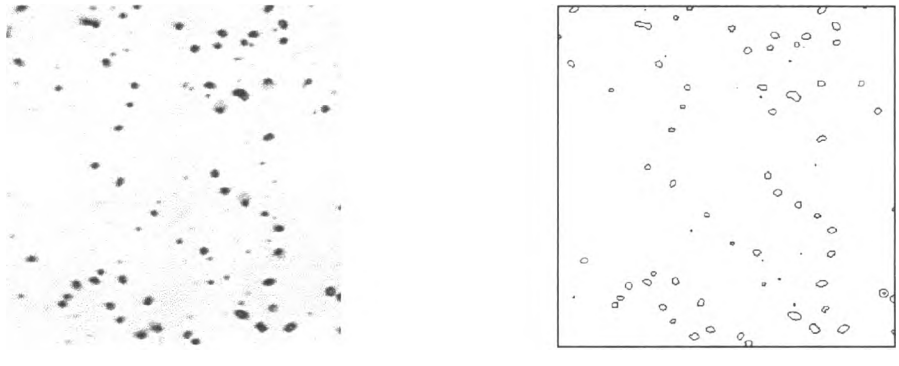
4.2 การหาเงื่อนไขการล้างกัดรอยที่เหมาะสม

โปรแกรมนับรอยอนุภาคแอลฟาบนฟิล์มเซลลูโลสไนเตรตที่พัฒนาขึ้น ยังมีข้อจำกัดในการนับอยู่ เช่นกรณีรอยอนุภาคที่มีการซ้อนทับกันเป็นกลุ่ม โปรแกรมจะไม่สามารถแยกแยะรอยดังกล่าวได้ จำนวนรอยที่นับได้จะผิดไปจากความเป็นจริง ดังนั้นเพื่อให้โปรแกรมนับรอยอนุภาคที่พัฒนาขึ้นทำงานได้ถูกต้องมากที่สุดจำเป็นต้องหาเงื่อนไขในการ ถ่ายภาพ และการล้างกัดขยายรอยที่เหมาะสม

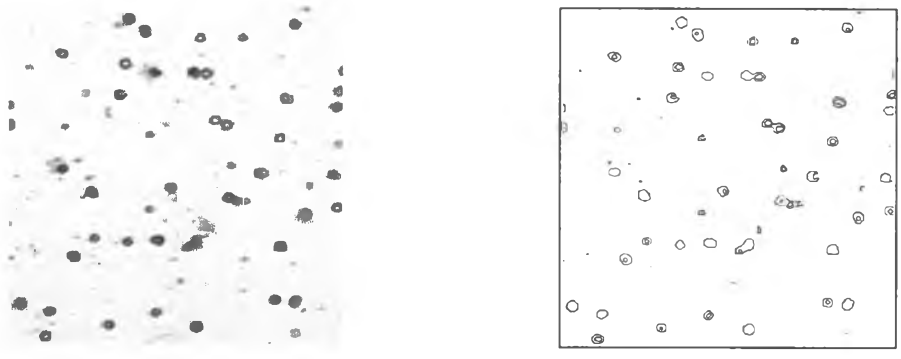
จากภาพถ่ายด้วยนิวตรอนโดยใช้เวลาถ่ายภาพต่างกัน และ เวลาล้างกัดขยายรอยต่างกัน จะได้ภาพดังนี้ (ภาพซ้ายคือภาพที่ได้จากกล้องจุลทรรศน์ ภาพขวาคือภาพที่เข้าโปรแกรมนับ)



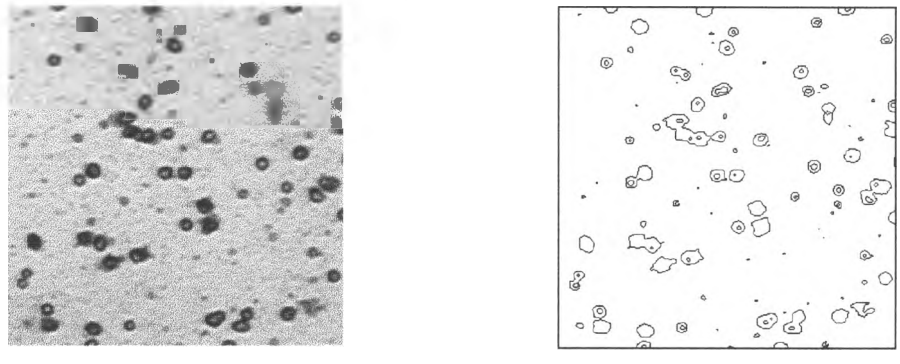
รูปที่ 4.24 เวลาถ่ายภาพ 5 นาที เวลาล้างกัดขยายรอย 10 นาที



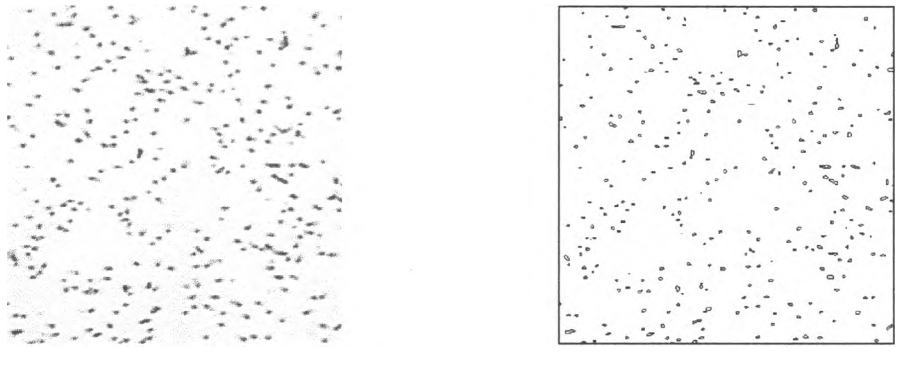
รูปที่ 4.25 เวลาถ่ายภาพ 5 นาที เวลาล้างกัดขยายรอย 20 นาที



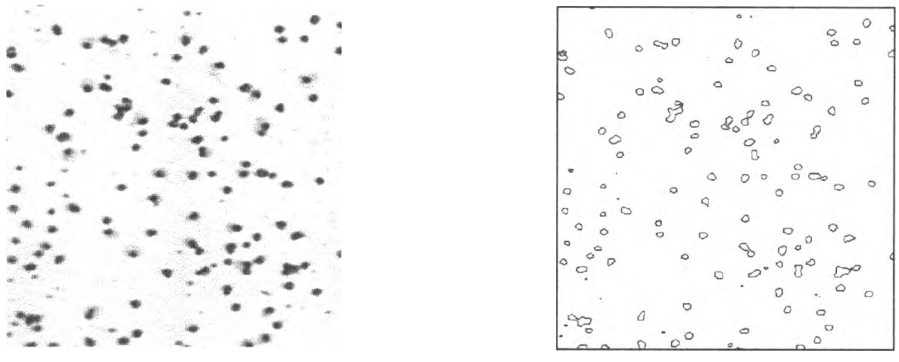
รูปที่ 4.26 เวลาถ่ายภาพ 5 นาที เวลาล้างกัดขยายรอย 30 นาที



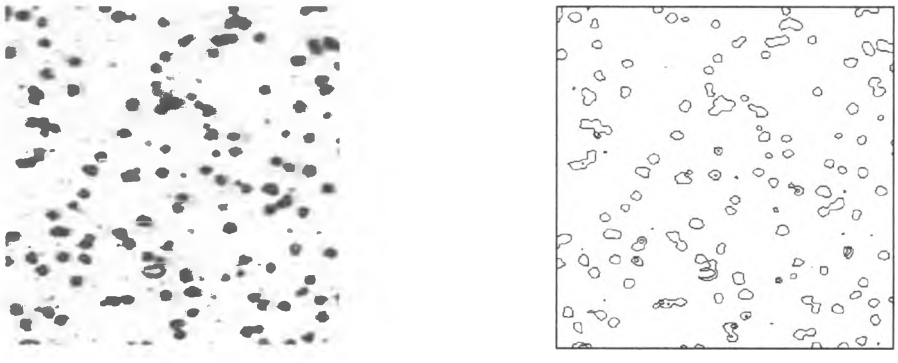
รูปที่ 4.27 เวลาถ่ายภาพ 5 นาที เวลาล้างกัดขยายรอย 40 นาที



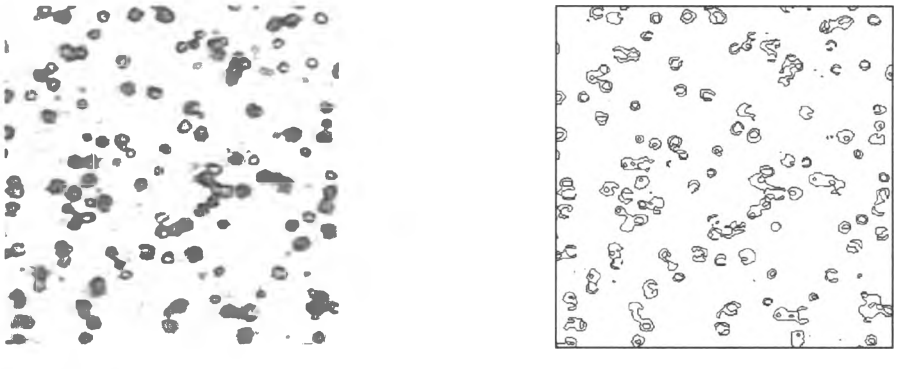
รูปที่ 4.28 เวลาถ่ายภาพ 10 นาที เวลาล้างกัดขยายรอย 10 นาที



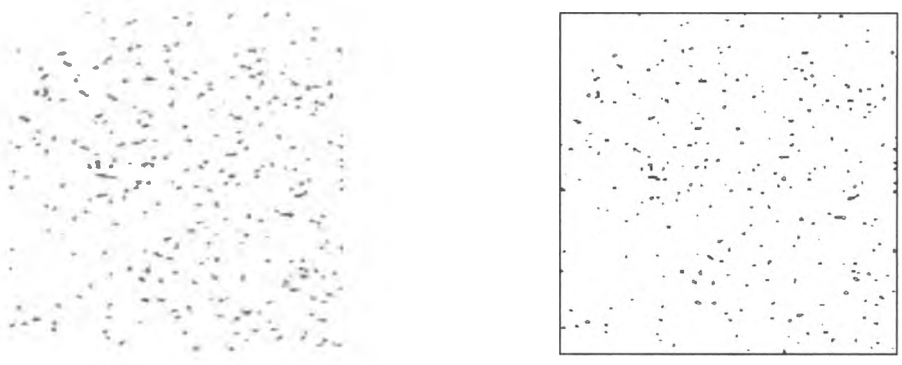
รูปที่ 4.29 เวลาถ่ายภาพ 10 นาที เวลาล้างกัดขยายรอย 20 นาที



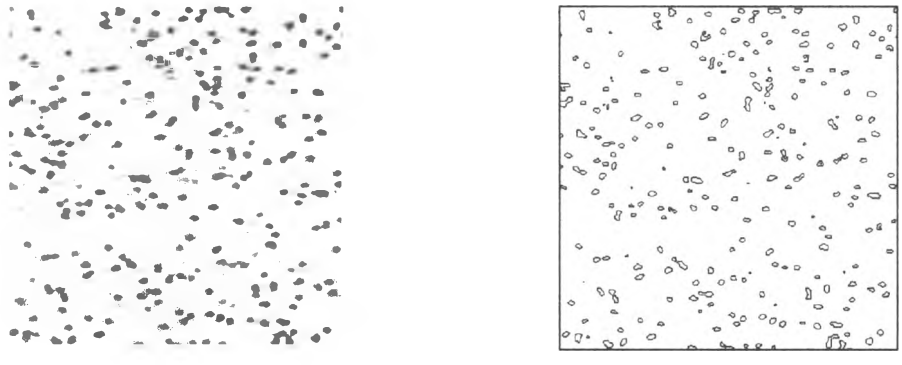
รูปที่ 4.30 เวลาถ่ายภาพ 10 นาที เวลาล้างกัดขยายรอย 30 นาที



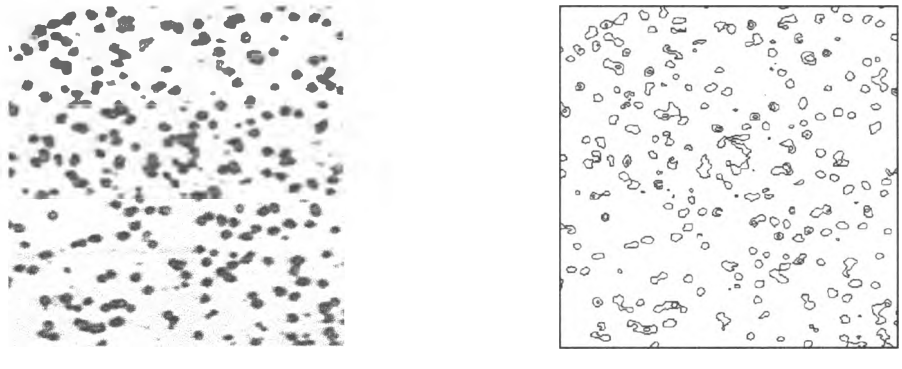
รูปที่ 4.31 เวลาถ่ายภาพ 10 นาที เวลาล้างกัดขยายรอย 40 นาที



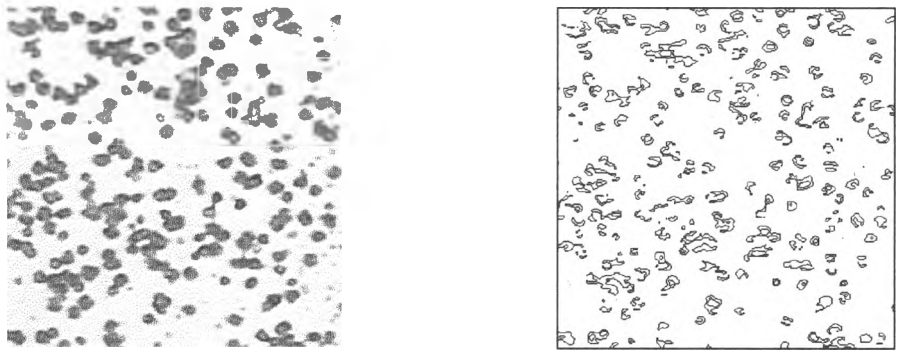
รูปที่ 4.32 เวลาถ่ายภาพ 20 นาที เวลาล้างกัดขยายรอย 10 นาที



รูปที่ 4.33 เวลาถ่ายภาพ 20 นาที เวลาล้างกัดขยายรอย 20 นาที



รูปที่ 4.34 เวลาถ่ายภาพ 20 นาที เวลาล้างกัดขยายรอย 30 นาที



รูปที่ 4.35 เวลาถ่ายภาพ 20 นาที เวลาล้างกัดขยายรอย 40 นาที

จากรูปที่ได้ที่เวลาการถ่ายภาพต่างๆ และ เวลาในการกัดขยายรอยต่างๆ ทดลองนำไปนับด้วยตา และ ใช้โปรแกรมนับภาพ จะปรากฏผลตามตารางดังนี้

เวลาถ่ายภาพ (นาที)	เวลาล้างกัดรอย (นาที)	จำนวนรอยนับด้วยตา	เวลาที่ใช้ในการนับด้วยตา (วินาที)	จำนวนรอยนับด้วยโปรแกรม	เวลาที่ใช้ในการนับด้วยโปรแกรม (วินาที)
5	10	64	35	65	26
5	20	61	34	62	33
5	30	44	28	46	28
5	40	57	32	50	52
10	10	210	113	273	122
10	20	110	55	111	52
10	30	115	56	119	58
10	40	118	46	113	64
20	10	186	104	195	110
20	20	121	120	225	1.43
20	30	214	130	188	91

ตารางที่ 4.1 แสดงการเปรียบเทียบการนับรอยอนุภาคด้วยตากับการใช้โปรแกรมนับ

จากการทดลองนับด้วยตาและนับด้วยโปรแกรมสามารถสรุปได้ดังนี้

ที่เวลาถ่ายภาพ 5-10 นาทีทุกเวลาการล้างกัดขยายรอย จะได้ผลการนับภาพรอยนุภาคใกล้เคียงกันระหว่างนับด้วยตาและนับด้วยโปรแกรมทั้งนี้เป็นเพราะรอยอนุภาคยังมีการซ้อนทับกันน้อย ยกเว้นที่เวลาถ่ายภาพ 10 นาทีเวลากัดรอย 10 นาทีจะได้ค่าจำนวนรอยต่างกันมาก ทั้งนี้เป็นเพราะขนาดของรอยเล็กและมีจำนวนรอยมากทำให้การนับด้วยตาอาจผิดพลาดได้ การถ่ายภาพตั้งแต่ 20 นาทีขึ้นไปจะมีจำนวนรอยอนุภาคมาก และมีการซ้อนทับกัน ทำให้การนับผิดพลาดมาก จึงไม่ควรใช้เวลาในการถ่ายภาพนานกว่า 20 นาที

การล้างกัดขยายรอย ยิ่งนานจะทำให้รอยมีขนาดใหญ่เกินไป เมื่อแปลงภาพให้เหลือแต่ขอบภาพจะได้ภาพที่มีจุดที่ไม่ต้องการอยู่ในรอยอนุภาค ดังนั้นเวลากัดรอยที่เหมาะสม จะอยู่ระหว่าง 10 -20 นาที

และที่สำคัญคือการนับด้วยโปรแกรมต้องกำหนดขนาดของรูปรอยที่สนใจ ให้เหมาะสมด้วย เพราะจะทำให้ได้จำนวนรอยที่ถูกต้อง

เวลาถ่ายภาพและเวลาล้างกัดรอยที่เหมาะสมที่จะใช้โปรแกรมนับภาพในการทดลองนี้พบว่า เวลาถ่ายภาพ 10 นาทีและเวลาการล้างกัดรอย 20 นาที คือสภาวะที่เหมาะสมที่สุดต่อการใช้โปรแกรมนับรอยอนุภาค