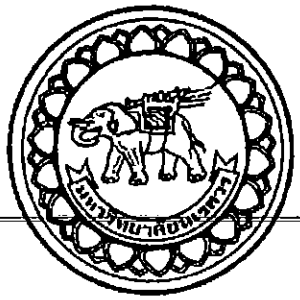


การพัฒนาชุดคำสั่งเชิงวัตถุทางการประมวลผลภาพดิจิทัล

DEVELOPMENT OF DIGITAL IMAGE PROCESSING OBJECT LIBRARY

นาย ฐิติ เสาวลักษณ์ รหัส 43370352

นางสาวบุญศิริ จั่นแก้ว รหัส 43370501

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2546

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน การพัฒนาชุดคำสั่งเชิงวัตถุทางการประมวลผลภาพดิจิทัล

ผู้ดำเนินโครงงาน นาย ฐิติ เสาวลักษณ์ รหัส 43370352

นางสาวบุญศิริ จั่นแก้ว รหัส 43370501

อาจารย์ที่ปรึกษา อาจารย์ รัฐภูมิ วรานุสาสน์

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2546

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงการวิศวกรรม

.....ประธานกรรมการ

(อาจารย์ รัฐภูมิ วรานุสาสน์)

.....กรรมการ

(ดร. พนมขวัญ ริยะมงคล)

.....กรรมการ

(ดร. สุชาติ แย้มเม่น)

หัวข้อโครงงาน	การพัฒนาชุดคำสั่งเชิงวัตถุทางการประมวลผลภาพดิจิทัล
ผู้ดำเนินโครงงาน	นาย ฐิติ เสาวลักษณ์ รหัส 43370352
	นางสาว บุญศิริ จั่นแก้ว รหัส 43370501
อาจารย์ที่ปรึกษา	อาจารย์ รัฐภูมิ วรานุสาสน์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2546

บทคัดย่อ

ภาพเป็นสิ่งสำคัญในการติดต่อสื่อสารและใช้ประโยชน์ในด้านต่างๆ เทคนิคการประมวลผลภาพดิจิทัลจึงได้ถูกนำมาประยุกต์ใช้อย่างกว้างขวาง การพัฒนาซอฟท์แวร์ที่สามารถนำส่วนของโปรแกรมกลับมาใช้ได้ใหม่นั้น ในยุคต้นใช้ แนวคิดการจัดซอฟท์แวร์เป็นฟังก์ชัน แต่เนื่องจากวิธีนี้มีข้อเสีย คือ การแก้ไข และการเพิ่มเติมฟังก์ชันทำได้ไม่สะดวก ผู้จัดทำโครงงานจึงนำเทคนิคการประมวลผลภาพดิจิทัลมาพัฒนา โดยได้นำเอาแนวทางของการเขียนโปรแกรมเชิงวัตถุเข้ามาพัฒนาร่วม เพื่อลดการเขียนโปรแกรมที่ซ้ำซ้อน เพิ่มประสิทธิภาพ และความสะดวกในการเขียนโปรแกรมทางด้านการประมวลผลภาพดิจิทัล โดยที่แนวทางของการเขียนโปรแกรมเชิงวัตถุสามารถที่จะเก็บชุดคำสั่งและนำมาเรียกใช้ได้ใหม่ที่สามารถประหยัดเวลาในการเขียนโปรแกรมได้

ผลที่ได้จากการทำโครงงาน คือ ชุดคำสั่งนี้สามารถลดการเขียนโปรแกรมที่ซ้ำซ้อน เพิ่มประสิทธิภาพ และความสะดวกในการเขียนโปรแกรมทางด้านการประมวลผลภาพดิจิทัลได้

Project Title	Development of Digital Image Processing Object Library		
Name	Mr. Thiti	Solvaluk	ID 43370352
	Miss Boonsiri	Junkeaw	ID 43370501
Project Advisor	Mr. Rattapoom	Waranusart	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2003		

ABSTRACT

Digital image plays very important role in many fields, which makes digital image processing also being applied largely. Developing an image processing application is not a trivial task. Reusable functions were applied, but this method is inconvenient. Object-Oriented programming is applied to develop an object library in order to increase efficiency in developing image processing application.

The result of this project is an object library which provides reusability and efficiency in developing image processing application.

กิตติกรรมประกาศ

โครงงานวิจัยวิศวกรรมคอมพิวเตอร์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดี ทางผู้ดำเนินงานต้องขอขอบพระคุณ อาจารย์ รัฐภูมิ วรานุสาสน์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานที่กรุณาให้คำปรึกษาและชี้แนวทางการแก้ไขปัญหาต่างๆ ที่เกิดขึ้นในระหว่างการทำโครงงาน ตลอดจนติดตามประเมินผลการทำโครงงานมาโดยตลอด และทางผู้จัดทำใคร่ขอขอบพระคุณท่านอาจารย์เป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณฝ่ายการเงินของคณะวิศวกรรมศาสตร์มหาวิทยาลัยนเรศวรที่ช่วยอำนวยความสะดวกในการเบิกจ่ายในการดำเนินโครงงาน

ขอขอบพระคุณบิดามารดา ที่ให้อุปการะเลี้ยงดูและสั่งสอนจนสามารถเติบโตมาจนถึงปัจจุบัน ตลอดจนช่วยอุปการะทางการเงินและคอยให้กำลังใจ จนกระทั่งโครงงานนี้เสร็จสมบูรณ์

ขอขอบพระคุณทุกๆ ท่านที่มิได้เอ่ยนามในที่นี้ ที่มีส่วนร่วมช่วยให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้คณะผู้ดำเนินงานขอขอบคุณ คุณงามความดีที่เกิดจากโครงงานนี้ แด่ผู้มีพระคุณทุกท่านที่มีส่วนร่วมในการทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี และถ้าเกิดผิดพลาดประการใดจากโครงงานนี้ คณะผู้ดำเนินงานต้องกราบขออภัยไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำโครงงานวิศวกรรมคอมพิวเตอร์
นาย ฐิติ เสาวลักษณ์
นางสาว บุญศิริ จั่นแก้ว

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ก

บทคัดย่อภาษาอังกฤษ..... ข

กิตติกรรมประกาศ..... ค

สารบัญ..... ง

สารบัญรูป..... ฉ

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน..... 1

1.2 วัตถุประสงค์ของโครงงาน..... 4

1.3 ขอบข่ายของโครงงาน..... 4

1.4 ขั้นตอนการดำเนินงาน..... 5

1.5 ผลที่คาดว่าจะได้รับ..... 5

1.6 งบประมาณที่ใช้..... 6

บทที่ 2 งานวิจัยที่เกี่ยวข้อง

2.1 หลักการพัฒนาโปรแกรมเบื้องต้น..... 7

2.2 ทฤษฎีการประมวลผลภาพดิจิทัล..... 10

บทที่ 3 วิธีดำเนินการวิจัย

3.1 ศึกษาและออกแบบการสร้างคอมโพเนนต์ บน Borland C++ Builder..... 35

3.2 การพัฒนาโปรแกรม..... 36

บทที่ 4 ผลการทดลอง

4.1 ชุดคำสั่งทำการอินเวิร์ทภาพ (Invert)..... 56

4.2 ชุดคำสั่งสร้างภาพไบนารี (Binary)..... 56

4.3 ชุดคำสั่งคำนวณหาฮิสโตแกรม (Histogram)..... 57

4.4 ชุดคำสั่งการประมวลผลภาพเชิงเรขาคณิต (Geometric Operations)..... 58

สารบัญ (ต่อ)

หน้า

4.5 ชุดคำสั่ง TImageProcessing (Flip) 59
4.6 ชุดคำสั่งหาขอบภาพ (Edge Detection) 59
4.7 ชุดคำสั่ง TImageProcessing (Robert) 60
4.8 ชุดคำสั่งทำให้ภาพเรียบ (Smoothing) 60
4.9 ชุดคำสั่ง TImageProcessing (Median) 61
4.10 ชุดคำสั่งทำให้ภาพคมชัดขึ้น (Sharpening) 61
4.11 ชุดคำสั่งมอโฟโลยี (Morphological Operation) 62
4.12 ชุดคำสั่ง TImageProcessing (Dilation) 63
4.13 ชุดคำสั่ง TImageProcessing (Erotion) 63
4.14 ชุดคำสั่ง TImageProcessing (Opening) 64
4.15 ชุดคำสั่ง TImageProcessing (Closing) 64

บทที่ 5 บทสรุป
5.1 สรุปผล 65
5.2 ปัญหาที่พบในการทำโครงงานวิจัย 65
5.3 ข้อเสนอแนะ 65

เอกสารอ้างอิง 66

ภาคผนวก
ก คู่มือการใช้งานชุดคำสั่งเชิงวัตถุทางการประมวลผลภาพดิจิทัล 67

ประวัติผู้เขียนโครงงาน 73

สารบัญรูป

รูปที่		หน้า
1.1	ภาพที่ได้จากการส่งผ่านทางสายเคเบิลในปี ค.ศ. 1921	1
1.2	ภาพดวงจันทร์ที่ได้จากยานเรนเจอร์ 7 ในปี ค.ศ. 1964	2
1.3	ภาพตัวอย่างการประยุกต์ใช้การประมวลผลภาพดิจิทัลในด้านต่างๆ 1) ภาพศีรษะมนุษย์จากการทำ MRI 2) ภาพลายนิ้วมือ 3) ภาพแผงวงจร 4) ภาพถ่ายดาวเทียมบริเวณวอชิงตัน ดี.ซี. ที่ได้จากดาวเทียมแลนด์แซท (LANDSAT)	3
2.1	วัตถุของรูปทรงเรขาคณิต	8
2.2	ไบโมดัล อิมเมจ ฮีสโตแกรม (Bimodal image histogram)	11
2.3	(ก) ภาพต้นฉบับ (ข) แผนภูมิต้นไม้ (Quad tree representation)	15
2.4	แสดงการหมุนตำแหน่งของภาพรอบจุดหมุน	16
2.5	แสดงการย้ายตำแหน่งของจุด P	17
2.6	แสดงการสะท้อนของภาพตามแกน x	19
2.7	แสดงการสะท้อนของภาพตามแกน y	20
2.8	แสดงการสะท้อนของภาพรอบแกนใด ๆ	20
2.9	แสดงเมทริกซ์สำหรับหมุนภาพไปยังระนาบแกน x	21
2.10	แสดงการสะท้อนของภาพกับแกน x	21
2.11	แสดงการหมุนภาพกลับไปยังระนาบเดิม	21
2.12	แสดงลักษณะข้อมูลภาพ	22
2.13	แสดงการกระทำการเบื้องต้น	22
2.14	แสดงถึงจุดภาพที่ทราบ	23
2.15	แสดงแม่แบบข้อมูลภาพ	23
2.16	แสดงข้อมูลภาพแถวแรก	24
2.17	แสดงการยูเนียนจุดภาพเท่ากับ 1 ในแถวแรก	24
2.18	แสดงการยูเนียนจุดภาพเท่ากับ 1 ณ. ตำแหน่งจุดภาพที่สองในแถวแรก	24
2.19	แสดงภาพผลลัพธ์จากการยูเนียน	24
2.20	การแสดงแม่แบบกร่อนภาพกับข้อมูลต้นฉบับ	25
2.21	แสดงภาพผลลัพธ์ของการกร่อน	25
2.22	ผลลัพธ์เมื่อแม่แบบเป็น 1 ทั้งหมด	25

สารบัญรูป (ต่อ)

รูปที่		หน้า
2.23	แสดงการเปิดด้วย 1 ทั้งหมด	26
2.24	แสดงการปิดด้วย 1 ทั้งหมด	27
2.25	การกรองข้อมูลภาพ	27
2.26	เป็นวิธีการกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน	29
2.27	การกรองข้อมูลภาพโดยใช้ค่าฐานนิยม	30
2.28	การกรองโดยใช้หน้าต่าง	30
2.29	การกรองข้อมูลภาพโดยใช้หน้าต่าง	31
2.30	การกรองภาพด้วยแม่แบบ	33
2.31	ฟังก์ชันการแปลงระดับเทาเพื่อปรับปรุงความแตกต่างของภาพ	34
4.1	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับอินเวิร์ทภาพ	56
4.2	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับสร้างเส้นขีดแบ่งของภาพ	57
4.3	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับคำนวณหาฮิสโตแกรมของภาพ	57
4.4	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการหมุนภาพ 90 องศา (ขวา)	58
4.5	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการหมุนภาพ 90 องศา (ซ้าย)	58
4.6	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการกลับภาพ	59
4.7	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับหาขอบภาพแบบโซเบล	59
4.8	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับหาขอบภาพแบบ	60
4.9	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการทำให้ภาพเรียบแบบการกระทำเลขคณิตหน้ากาก (Mask) ขนาด 6x6	60
4.10	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการทำให้ภาพเรียบแบบการกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน(Median Filtering)หน้ากาก (Mask)ขนาด6x6	61
4.11	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับทำให้ภาพคมชัดขึ้น (Sharpening) โดยใช้ ตัวกรองความถี่สูงผ่าน (High-pass Filtering) ขนาด 3x3	62
4.12	แสดงภาพที่ใช้ในการทดลองเพื่อนำไปเป็นผลลัพธ์	62

สารบัญรูป (ต่อ)

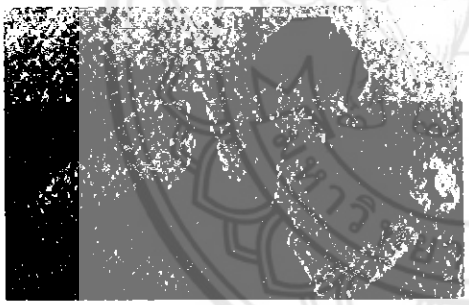
รูปที่		หน้า
4.13	แสดงภาพที่ใช้ในการทดลองเพื่อนำไปหน้ากาก (Mask)	62
4.14	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการขยายภาพ	63
4.15	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการกร่อนภาพ	63
4.16	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการเปิด	64
4.17	แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการปิด	64
ก-1	แสดงคอมโพเนนต์แพเลตต์บน C++ Builder	67
ก-2	แสดงการนำคอมโพเนนต์วางไว้บนฟอร์มเพื่อใช้งาน	67

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

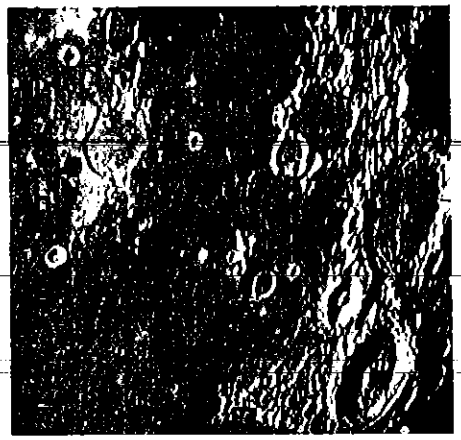
ในสมัยก่อนมนุษย์มีการมองเห็นแต่ไม่สามารถที่จะบันทึกภาพที่มองเห็นได้จริงนั้นเก็บไว้ได้ มีเพียงแต่การวาดภาพเสมือนเก็บไว้ และภาพวาดนั้นอาจจะไม่สมบูรณ์แบบ จึงมีการตกแต่งภาพเพื่อให้สมจริงมากที่สุด

ในเวลาต่อมามนุษย์ได้เริ่มคิดสิ่งประดิษฐ์ที่สามารถบันทึกภาพเก็บไว้ให้เหมือนจริง คือ กล้องถ่ายภาพ และได้มีการส่งรูปภาพนั้นไปยังที่ต่างๆ เพื่อต้องการแจ้งข่าวสารนั้นๆ โดยนำรูปภาพส่งติดไปกับเอกสารนั้น และได้กลายเป็นจุดเริ่มต้นของการพัฒนาขบวนการประมวลผลภาพ (Image Processing) เพื่อให้ภาพคมชัดในการที่จะนำรูปภาพนั้นไปช่วยทางด้านต่างๆ ในทศวรรษที่ 1920 ได้มีการส่งข้อมูลภาพผ่านทางสายเคเบิลใต้น้ำระหว่างลอนดอนกับนิวยอร์กโดยอาศัยอุปกรณ์การพิมพ์ที่ออกแบบมาโดยเฉพาะแปลงข้อมูลภาพเป็นรหัสส่งผ่านทางสายเคเบิลแล้วใช้เครื่องอีกชนิดหนึ่งแปลงกลับเมื่อถึงปลายทาง ดังรูปที่ 1.1



รูปที่ 1.1 ภาพที่ได้จากการส่งผ่านทางสายเคเบิลในปี ค.ศ. 1921

การใช้คอมพิวเตอร์ในการประมวลผลภาพครั้งแรกเกิดขึ้นในช่วงต้นของทศวรรษที่ 1960 ในการสำรวจอวกาศ เมื่อภาพดวงจันทร์ที่ถูกส่งมาจากยานเรนเจอร์ 7 (Ranger 7) ถูกปรับปรุงคุณภาพด้วย การใช้คอมพิวเตอร์ในห้องปฏิบัติการพลังขับเคลื่อนเจ็ต (Jet Propulsion Laboratory) นับว่าเป็นภาพแรกที่ใช้คอมพิวเตอร์ในการปรับปรุง ดังแสดงในรูปที่ 1.2

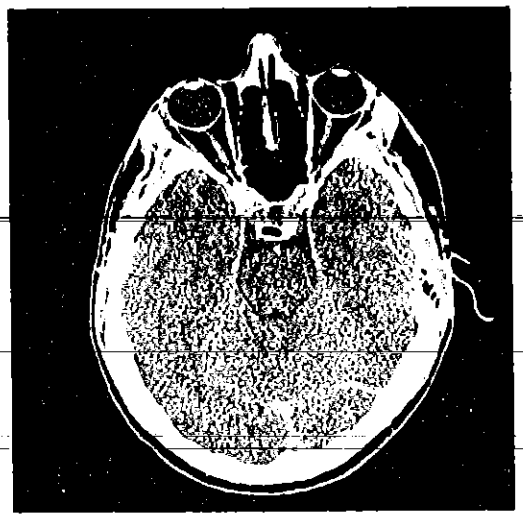


รูปที่ 1.2 ภาพดวงจันทร์ที่ได้จากยานเรนเจอร์ 7 ในปี ค.ศ. 1964

หลังจากนั้นมาการประมวลผลภาพดิจิทัล (Digital Image Processing) ได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์ และได้ถูกนำไปประยุกต์ใช้ให้เกิดประโยชน์ทางด้านต่างๆ เช่น

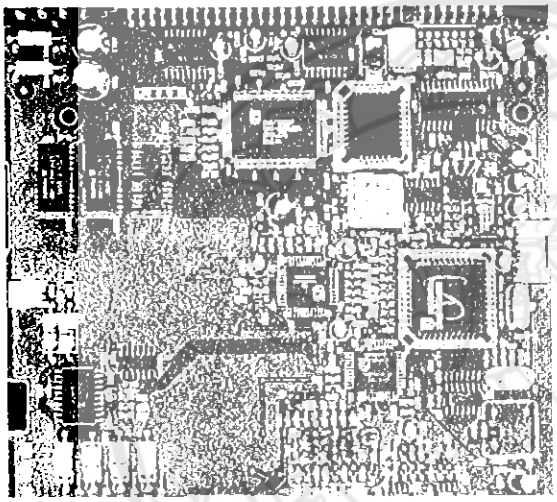
1. ทางด้านการแพทย์ เช่น นำไปใช้ปรับปรุงภาพถ่ายรังสีเอ็กซ์ (X- ray) ใช้แยกส่วนของเนื้อสมองในภาพ MRI เป็นต้น ดังตัวอย่างในรูปที่ 1.3.1
2. ทางด้านนิติเวชและการบังคับใช้กฎหมาย เช่น การตรวจจับลายนิ้วมือ การตรวจจับแผ่นป้ายทะเบียนรถ เป็นต้น ดังตัวอย่างในรูปที่ 1.3.2
3. ทางด้านการทหาร เช่น การปรับปรุงภาพถ่ายทางอากาศ การนำร่องขีปนาวุธ การประมวลผลภาพที่ได้จากเรดาร์ เป็นต้น
4. ทางด้านทางอุตสาหกรรม เช่น การตรวจสอบคุณภาพผลิตภัณฑ์ หุ่นยนต์ เป็นต้น ดังตัวอย่างในรูปที่ 1.3.3
5. ทางด้านภูมิศาสตร์และอุตุนิยมวิทยา เช่น การศึกษาภาพถ่ายจากดาวเทียม การสำรวจทรัพยากรธรรมชาติ เป็นต้น ดังตัวอย่างในรูปที่ 1.3.4

นอกจากประโยชน์ในด้านต่างๆ ข้างต้นแล้ว การประมวลผลภาพดิจิทัลยังใช้ในการสำรวจอวกาศ หนังสือพิมพ์ ภาพยนตร์ เป็นต้น

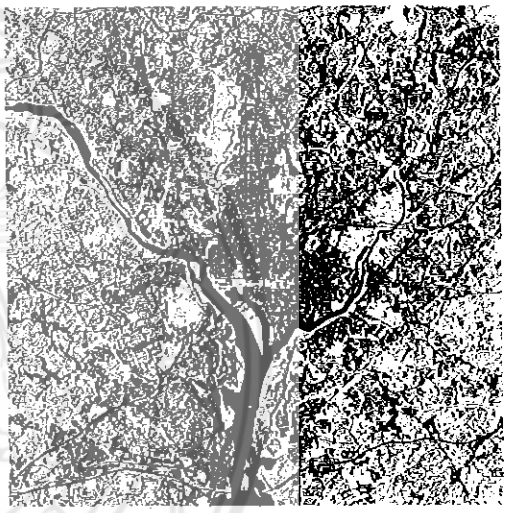


1)

2)



3)



4)

รูปที่ 1.3 ภาพตัวอย่างการประยุกต์ใช้การประมวลผลภาพดิจิทัลในด้านต่างๆ 1) ภาพศีรษะมนุษย์จากการทำ MRI 2) ภาพลายนิ้วมือ 3) ภาพแผงวงจร 4) ภาพถ่ายดาวเทียมบริเวณอชิงตัน ดี.ซี. ที่ได้จากดาวเทียมแลนด์แซท (LANDSAT)

จะเห็นได้ว่าเทคนิคการประมวลผลภาพดิจิทัลได้ถูกนำไปประยุกต์ใช้อย่างกว้างขวาง มีบริษัทเอกชนจำนวนมากพัฒนาซอฟท์แวร์ขึ้นมาเพื่อใช้กับงานทางด้านการประมวลผลภาพดิจิทัล แต่การพัฒนาซอฟท์แวร์ด้านนี้นั้น ใช้เวลาในการพัฒนามาก และขั้นตอนวิธีทางการประมวลผลภาพดิจิทัลบางวิธีเป็นที่นิยม และมีใช้ในซอฟท์แวร์ส่วนมาก เช่น ขั้นตอนวิธีการหาขอบภาพ (Edge Detection) ขั้นตอนวิธีการลดสัญญาณรบกวน (Noise Reduction) เป็นต้น ซึ่งขั้นตอนวิธีที่ใช้ซ้ำบ่อยครั้งดังกล่าว หากนำรหัสโปรแกรม (Program Code) นั้นๆ มาใช้ใหม่ได้จะทำให้ขั้นตอนการพัฒนาซอฟท์แวร์ทำได้ด้วยความสะดวกและรวดเร็วมากขึ้น

การพัฒนาซอฟท์แวร์ที่สามารถนำส่วนของโปรแกรมกลับมาใช้ได้ใหม่นั้น ในยุคต้นใช้แนวคิดการจัดซอฟท์แวร์เป็นฟังก์ชั่น (Function) แต่เนื่องจากวิธีนี้มีข้อเสียคือ การแก้ไข และการเพิ่มเติมฟังก์ชั่นทำได้ไม่สะดวกเท่าที่ควร ต่อมาเมื่อมีการพัฒนาแนวคิดการโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP) ทำให้การพัฒนาโปรแกรมที่สามารถนำมาใช้ได้อีกสะดวกมากขึ้น

ดังนั้น ผู้จัดทำโครงงานจึงมีแนวคิดที่จะนำเอาหลักการเขียนโปรแกรมเชิงวัตถุมาประยุกต์เพื่อเพิ่มประสิทธิภาพในการเขียนโปรแกรมประเภทการประมวลผลภาพดิจิทัลนี้ ให้ใช้งานได้บนไมโครซอฟท์วินโดวส์ ซึ่งเป็นการอำนวยความสะดวกให้กับผู้พัฒนาซอฟท์แวร์ให้มีประสิทธิภาพมากขึ้น ลดการทำงานที่ซ้ำซ้อน และลดเวลาในการพัฒนา

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาทฤษฎีแนวทางของการเขียนโปรแกรมเชิงวัตถุและทฤษฎีด้านการประมวลผลภาพดิจิทัล

1.2.2 เพื่อพัฒนาชุดคำสั่งทางด้านการประมวลผลภาพดิจิทัล

1.2.3 เพื่อสามารถนำชุดคำสั่งทางด้านการประมวลผลภาพดิจิทัลมาใช้งานได้จริง

1.2.4 เพื่อลดการเขียนโปรแกรมแบบซ้ำซ้อน เพิ่มประสิทธิภาพและความสะดวกในการเขียนโปรแกรมทางด้านการประมวลผลภาพดิจิทัลมา

1.3 ขอบข่ายของโครงงาน

1.3.1 พัฒนาชุดคำสั่งทางการประมวลผลภาพดิจิทัลโดยวิธีโปรแกรมเชิงวัตถุ เพื่อให้นำกับ มาใช้ได้อีกบน สภาพแวดล้อมของการเขียนโปรแกรมของ Borland C++ Builder

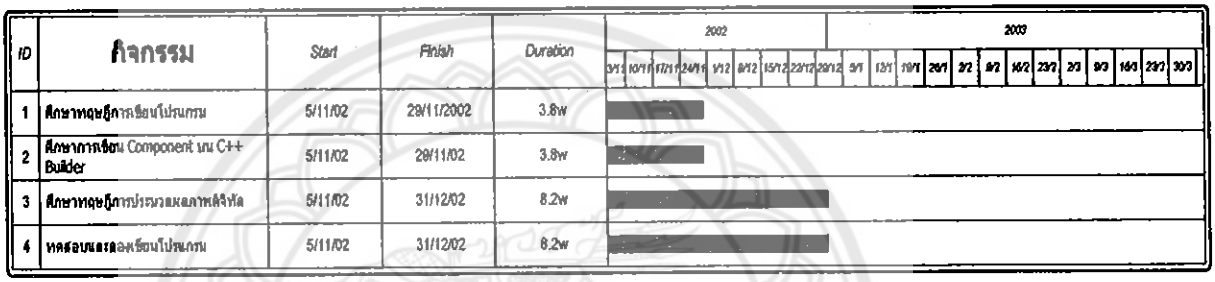
1.3.2 ชุดคำสั่งครอบคลุมทฤษฎีทางการประมวลผลภาพดิจิทัลเบื้องต้น ดังนี้

- ชุดคำสั่งทำงานกับภาพระดับเทา (Gray Scale)
- ชุดคำสั่งสามารถทำการประมวลผลภาพเชิงเรขาคณิต (Geometric Operations) ได้ เช่น การหมุนภาพ (Rotation) การพลิกภาพ (Flip)
- ชุดคำสั่งสามารถคำนวณหาฮิสโตแกรม (Histogram) ของภาพได้
- ชุดคำสั่งสามารถทำการอินเวิร์ทภาพ (Invert) ได้
- ชุดคำสั่งสามารถสร้างภาพไบนารี (Binary) ได้จากการใช้ค่าขีดแบ่ง (Threshold Value)
- ชุดคำสั่งสามารถหาขอบภาพ (Edge Detection) ได้โดยขั้นตอนวิธีพื้นฐาน เช่น การหาขอบภาพแบบโซเบล (Sobel's Edge Detection) การหาขอบภาพแบบโรเบร์ต (Robert's Edge Detection)
- ชุดคำสั่งสามารถทำให้ภาพเรียบ (Smoothing) ได้ เช่น การกระทำเลขคณิต(Arithmetic)

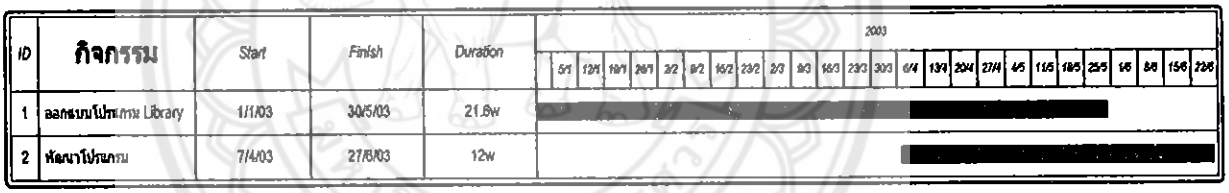
การกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน (Median Filtering)

- ชุดคำสั่งสามารถทำให้ภาพคมชัดขึ้น (Sharpening) ได้ เช่น ตัวกรองความถี่สูงผ่าน (High-pass Filtering)
- ชุดคำสั่งสามารถทำการกระทำแบบมอโฟโลยี (Morphological Operation) ได้ เช่น การขยายภาพ (Dilation) การกร่อนภาพ (Erosion) การเปิด (Opening) การปิด (Closing)

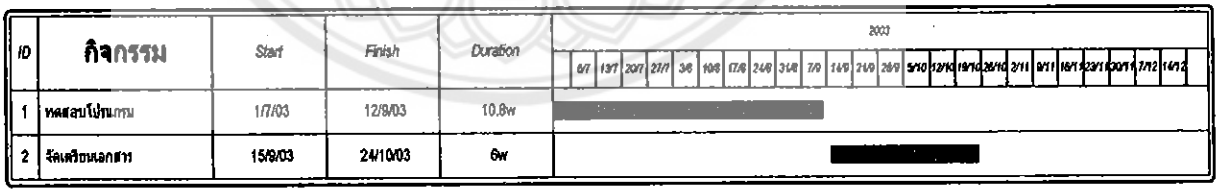
1.4 ขั้นตอนการดำเนินงาน



ID	กิจกรรม	Start	Finish	Duration
1	ศึกษาทฤษฎีการเขียนโปรแกรม	5/11/02	29/11/2002	3.8w
2	ศึกษาการเขียน Component บน C++ Builder	5/11/02	29/11/02	3.8w
3	ศึกษาทฤษฎีการประมวลผลภาพดิจิทัล	5/11/02	31/12/02	8.2w
4	ทดสอบและเตรียมโปรแกรม	5/11/02	31/12/02	8.2w



ID	กิจกรรม	Start	Finish	Duration
1	ออกแบบโปรแกรม Library	1/1/03	30/5/03	21.6w
2	พัฒนาโปรแกรม	7/4/03	27/6/03	12w



ID	กิจกรรม	Start	Finish	Duration
1	ทดสอบโปรแกรม	1/7/03	12/9/03	10.8w
2	จัดเตรียมเอกสาร	15/9/03	24/10/03	6w

1.5 ผลที่คาดว่าจะได้รับ

-ได้ความรู้ทางด้านการประมวลผลภาพ

-ได้ความรู้การเขียนโปรแกรมเชิงวัตถุ

-ได้ชุดคำสั่งการประมวลผลภาพดิจิทัล

1.6 งบประมาณ

1. ค่าหนังสือและเอกสาร	800	บาท
2. ค่าหมึกเครื่องพิมพ์	600	บาท
3. ค่าซอฟต์แวร์ที่ใช้ในการพัฒนา	500	บาท
4. ค่ากระดาษ	100	บาท
รวมทั้งสิ้น	2,000	บาท (สองพันบาทถ้วน)

บทที่ 2
งานวิจัยที่เกี่ยวข้อง

2.1 หลักการพัฒนาโปรแกรมเบื้องต้น

ในการเขียนชุดคำสั่งงานด้วยภาษาคอมพิวเตอร์จะมีแนววิธีการอยู่สองรูปแบบด้วยกันคือ แบบที่เรียกว่าภาษาเชิงกระบวนการคำสั่ง (Procedural Programming) และแนวเชิงวัตถุ (Objected Oriented Programming)

การเขียนโปรแกรมภาษาเชิงกระบวนคำสั่งจะเริ่มต้นจากการตั้งตัวแปร (Variable) เพื่อทำงานของเนื้อที่ในหน่วยความจำและจัดเตรียมเนื้อที่ในการจัดเก็บข้อมูล จากนั้นก็เขียนขั้นตอนหรือลำดับในการที่จะสั่งให้เครื่องคอมพิวเตอร์ทำงานตามที่กำหนด

2.1.1. การเขียนโปรแกรมเชิงวัตถุ

การเขียนโปรแกรมเชิงวัตถุจะแตกต่างจากภาษาเชิงกระบวนคำสั่งผู้พัฒนาโปรแกรมเชิงวัตถุจะต้องมีวิสัยทัศน์ที่เรียกว่าเชิงวัตถุ เช่น ในการคำนวณหาค่าแรงของพนักงานแต่ละคนจะมีตัวแปรที่กำหนดความแตกต่างของพนักงาน คือ อัตราค่าจ้างต่อชั่วโมงอาจแตกต่างกันและจำนวนชั่วโมงในการทำงานของพนักงานแต่ละคนก็แตกต่างกันเช่นกัน ดังนั้นตัวแปรเหล่านี้เราเรียกว่าเป็นคุณลักษณะประจำ (Attribute) หรือเป็นสถานการณ์ (State) แต่กระบวนการหลักในการทำงานข้างต้นจะประกอบด้วยการคำนวณค่าจ้างที่ต้องจ่าย นอกจากนี้อาจมีรายการอื่นๆอีกเช่น การคำนวณหาภาษีหัก ณ ที่จ่าย การพิมพ์ สลิปเงินเดือนเป็นต้น

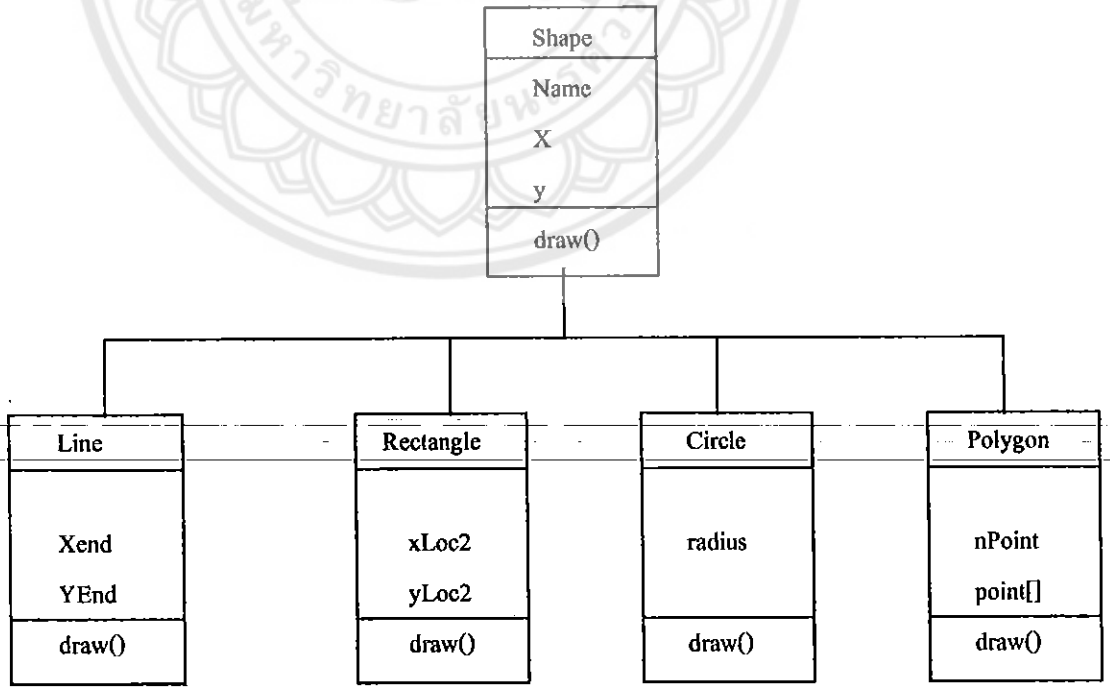
ดังนั้นการมองแบบเชิงวัตถุจะต้องพยายามในการค้นหาตัวแปรที่ใช้ในการจำแนกความแตกต่างของวัตถุแต่ละอย่างออกมา ขณะเดียวกันก็ต้องพิจารณาพฤติกรรมของวัตถุดังกล่าว และรวบรวมนำมาเขียนเป็นคลาส (Class) ซึ่งคลาสจะเป็นเหมือนแม่แบบ หรือ พิมพ์เขียวในการที่จะไปสร้างวัตถุต่างๆให้เกิดขึ้นจริงและมีชีวิตในระบบ ตัวอย่างที่จะสร้างจากคลาสดังกล่าวก็จะเป็นการคำนวณอัตราค่าแรงของพนักงานแต่ละคนเป็นต้น

โดยปกติคลาสจะเป็นเสมือนพิมพ์เขียวหรือเป็นแม่แบบที่จะทำตาม เช่นเมื่อมีการรับพนักงานใหม่ และต้องมีการจ่ายค่าแรงให้พนักงาน ดังนั้นคลาสแม่แบบจะนำมาสร้างเป็นออบเจ็กต์หรืออินสแทนซ์ (Instance) ซึ่งออบเจ็กต์ที่สร้างจะมีการกำหนดรายละเอียดที่แตกต่างกันไป เช่น อัตราค่าจ้าง จำนวนชั่วโมงการทำงาน แต่ออบเจ็กต์จะรวมข้อมูลและขั้นตอนหรือฟังก์ชันในการทำงานเข้าด้วยกัน

อีกตัวอย่างที่นิยมมาให้เห็นทางการเขียนโปรแกรมเชิงวัตถุคือ คลาสของรถ ตัวแปร หรือคุณลักษณะที่ใช้ในการอธิบายความแตกต่างของรถแต่ละคัน คือ ยี่ห้อของรถ เช่น อาจเป็น Toyota BMW สีของรถก็เป็นอีกตัวแปรที่ใช้อธิบายความแตกต่างของตัวรถจำนวนผู้โดยสารที่สามารถ

โดยสารบนรถ เป็นต้น ส่วนพฤติกรรมของรถ เช่น สามารถขับเคลื่อนไปข้างหน้า หรือ ถอยหลัง หรือหยุด เป็นต้น ซึ่งข้อมูลเหล่านี้จะถูกรวบรวมมาเป็นเสมือนพิมพ์เขียวที่ใช้ในการผลิตรถ และรถแต่ละคันที่ผลิตออกมาเราจะเรียกว่าเป็นอินสแทนซ์ซึ่งอินสแทนซ์จะมีลักษณะคุณสมบัติ (attribute) และเมทอด (method) ของมันเอง พิจารณารูปทรง สี่เหลี่ยมจัตุรัส สามเหลี่ยม วงกลม เส้น รูปทรงเหล่านี้ถือเป็นรูปทรงทางเรขาคณิตเกือบทั้งหมด คุณสมบัติทั่วไปที่ใช้ในการอธิบายรูปทรงต่างๆ คือ จุด ทั้งนี้เพราะในการวาดรูปทรงเรขาคณิตใดๆก็ตามพิกัดของจุดเป็นสิ่งที่สำคัญ และพฤติกรรมที่สำคัญคือการวาดรูป หรือการเปลี่ยนแปลงรูปแบบของรูปทรง อย่างไรก็ตามรูปทรงเรขาคณิตทั้งหมดก็ยังมีความแตกต่างกัน ไม่ว่าจะเป็นกรรมวิธีในการวาดรูป หรือ จำนวนพิกัดของจุด ในการพิจารณาเชิงวัตถุให้พยายามหาสิ่งที่มีคุณสมบัติร่วมกันก่อน แล้วจึงแยกความแตกต่างย่อย ซึ่งเรียกว่าเป็นสับคลาส (Sub Class) โดยที่สับคลาสจะมีคุณสมบัติเหมือนวัตถุต้นแบบ ที่เรียกว่าซูเปอร์คลาส (Super Class)

ดังนั้นรูปทรงเรขาคณิต (Shapes) โดยทั่วไปคือต้องมีการกำหนดจุดพิกัด ซึ่งเป็นจุดเริ่มต้นของการที่จะวาดรูป และอาจมีชื่อกำกับเพื่อใช้ในการอ้างอิง ส่วนรายละเอียดที่ทำให้แต่ละรูปทรงที่แตกต่างกันก็จะขยายเพิ่มจากคลาสพื้นฐาน เช่นของรูปทรงเส้นตรง ต้องการเพิ่มจุดจบของเส้น หรือกรณีที่เป็นวงกลมต้องการความยาวรัศมีในรูปที่ 2.1 จะเห็นว่าพฤติกรรม draw ซึ่งเป็นชุดคำสั่งในการวาดรูปทั้งในส่วนของ shape และ ส่วนที่เป็นคลาสขยายทั้งนี้เพราะในการวาดรูปทรงแต่ละแบบไม่เหมือนกัน หลักการในการวาดจึงจำเป็นต้องใส่ไว้ที่คลาสย่อยแต่ละคลาส



รูปที่ 2.1 วัตถุของรูปทรงเรขาคณิต

ตัวอย่างเรื่องรูปทรงเรขาคณิตนี้เป็นตัวอย่างของการสืบทอด ซึ่งถือว่าเป็นคุณสมบัติที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ ที่เรียกว่า การสืบทอด (Inheritance)

ในการออกแบบเชิงวัตถุที่สำคัญ คือ ต้องรวบรวมคุณลักษณะประจำและพฤติกรรมซึ่งคุณลักษณะประจำจะแทนด้วยตัวแปรคือสิ่งที่ใช้ในการอธิบายความแตกต่างระหว่างวัตถุต่างๆ เช่น ตัวอย่างเรื่อง รถ โครงของตัวถังของรถอาจเหมือนกัน แต่การตกแต่งจะทำให้รถแต่ละคันแตกต่างกัน เช่นเครื่องยนต์ที่ใส่ในรถแต่ละคัน ก็ทำให้ราคาขายแตกต่างกัน หรือ อย่างกรณีของรูปทรงเรขาคณิต เส้นตรง สี่เหลี่ยม วงกลม ทั้งหมดถือเป็นรูปทรงทางเรขาคณิตแต่จำนวนจุดที่ใช้ในการสร้างรูปทรงแตกต่างกันทำให้รูปทรงที่เห็นแตกต่างกัน

ส่วนพฤติกรรมคือ บริการหรืองานที่วัตถุต้องทำ เช่นกรณีของรูปทรงเรขาคณิต เราต้องการบริการคือการวาดรูปทรงนั้นๆ หรือ ของรถอาจเป็นการขับเคลื่อนของรถ เป็นต้น

ดังนั้นจะเห็นว่าการพัฒนาเชิงวัตถุสอดคล้องกับความเป็นจริงในโลกมนุษย์มากขึ้น คือ ต้องการฟังก์ชันงานใดเพิ่มก็เพียงแต่ซื้อฟังก์ชันมาใส่เพิ่มโดยที่เราไม่จำเป็นต้องทราบว่าฟังก์ชันนั้นมีกรรมวิธีมีกรรมวิธีในการใช้งานอย่างไรหรือไม่ต้องทราบว่าภายในฟังก์ชันมีรายละเอียดอะไรบ้าง แต่สิ่งที่ต้องทราบคือส่วนเชื่อมประสาน หรือส่วนที่แต่ละออบเจ็กต์จะติดต่อกัน

ดังนั้นการออกแบบหรือเขียนโปรแกรมเชิงวัตถุ ภาษาคอมพิวเตอร์ ต้องมีคุณสมบัติที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุซึ่งต้องประกอบด้วยหลักการดังนี้

2.1.1.1. ความสามารถในการสืบทอด (Inheritance) ตัวอย่างเรื่อง Shape เป็นตัวอย่างที่ชัดเจนในการสืบทอด กล่าวคือชื่อของรูปทรงสี่เหลี่ยม และพิกัดจุดจะสืบทอดไปยังรูปทรงอื่น เพราะเป็นคุณสมบัติที่ทุกรูปทรงต้องมี ซึ่งหลักการของการสืบทอดนี้ทำให้เราสามารถสร้างวัตถุอื่นๆบนพื้นฐานของวัตถุที่มีอยู่ และสามารถเรียกใช้ฟังก์ชันหรือ ชุดคำสั่งที่มีอยู่แล้วมาใช้งานได้อย่างประสิทธิภาพลดเวลาในการโปรแกรมใหม่ทุกครั้ง

2.1.1.2. ความสามารถในการเก็บซ่อน (Encapsulation) ข้อมูล โดยทั่วไปของการเขียนชุดคำสั่ง ข้อมูลบางประเภทเป็นข้อมูลที่ไม่ต้องการให้มีการแก้ไข ทั้งนี้เพื่อให้การประมวลผลเป็นไปตามวัตถุประสงค์ที่ต้องการ ภาษาที่ใช้ในการเขียนชุดคำสั่งงานที่ดีควรมีคุณสมบัติที่จะเก็บซ่อนข้อมูลที่ไม่ต้องการให้ผู้ใช้เปลี่ยนแปลง

2.1.1.3. ความสามารถในการแปลงร่าง (Polymorphism) ในการสืบทอดคุณสมบัติ ไม่จำเป็นเสมอไปที่จะต้องสืบทอดคุณสมบัติทุกอย่าง ในบางครั้งอาจต้องมีการแก้ไขลำดับในการทำงานของฟังก์ชันเดิม เพื่อให้การพัฒนาไม่ต้องเสียเวลาไปแก้ไขชุดคำสั่งเดิม ซึ่งถ้ามีการแก้ไขชุดคำสั่งเดิมจะทำให้ต้องเสียเวลาในการตามแก้ไขโปรแกรมต่างๆ ที่ใช้ฟังก์ชันที่ถูกแก้ไข วิธีการที่ดีคืออนุญาตให้มีการเขียนชุดคำสั่งใหม่ภายใต้ชื่อฟังก์ชันเดิมที่มีอยู่ เช่นในกรณีฟังก์ชัน draw() ในคลาส Shape จะเห็นว่าการวาดรูปทรงต่างๆจะมีกรรมวิธีในการวาดที่แตกต่างกันดังนั้นคลาสของ Line, Polygon,

Circle และ Rectangle สามารถที่จะเขียนชุดคำสั่ง draw() คือมีฟังก์ชันซ้ำกันแต่ทำงานภายในที่แตกต่างกัน

2.1.1.4. ความสามารถในการจัดการโครงสร้างข้อมูลแบบเชิงนาม หรือ แอบแสตก (Abstract) ความหมายของข้อมูลแบบเชิงนามคือข้อมูลหรือกระบวนการยังไม่เกิดขึ้นจริง ซึ่งในการพัฒนาระบบงานข้อมูลอาจจะยังไม่เกิด หรือยังไม่สามารถที่จะอธิบายได้ว่าขั้นตอนในการทำงานจะเป็นอย่างไร แต่เพื่อไม่ให้งานต้องหยุดชะงัก การสร้างคลาสเป็นแบบเชิงนามก็จะเป็นอีกวิธีหนึ่ง

2.2 ทฤษฎีการประมวลผลภาพดิจิทัล

2.2.1 การแยกข้อมูลภาพ (Image Segmentation)

การแยกข้อมูลภาพ คือ การแยกข้อมูลภาพออกเป็นส่วนๆซึ่งจะทำให้สามารถแยกข้อมูลภาพของส่วนที่ต้องการออกมาได้

วิธีการพื้นฐานสำหรับการแยกข้อมูลภาพออกเป็นส่วนๆ คือ การพิจารณาภาพแอปปริจูด (Image amplitude) (ได้แก่การพิจารณาความสว่างของภาพสำหรับภาพแบบระดับเทา และความแตกต่างของสีสำหรับภาพสี) นอกจากนี้ขอบของภาพและลักษณะของพื้นผิว ก็เป็นองค์ประกอบหนึ่งที่จะทำให้สามารถทำการเซคเมนเตชันได้สะดวกยิ่งขึ้น วิธีการแยกข้อมูลภาพ มีวิธีดังนี้

2.2.1.1 วิธีการแยกส่วนภาพโดยอาศัยความเข้ม (Amplitude segmentation methods)

สำหรับการแยกข้อมูลภาพออกเป็นส่วน ๆ ในหัวข้อนี้จะเป็นการพิจารณาความเข้มของจุดภาพต่าง ๆ ภายในภาพ ซึ่งผลของการแยกข้อมูลภาพจะขึ้นอยู่กับวิธีการหาค่าขีดแบ่งของส่วนประกอบที่เป็นความเข้มหรือสีของภาพ ซึ่งมีอยู่หลายวิธีด้วยกันดังนี้คือ

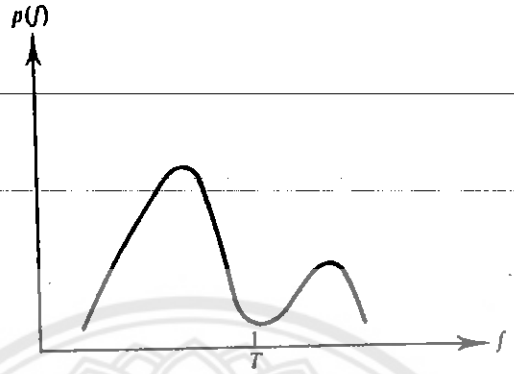
2.2.1.1.1 ค่าขีดแบ่งความเข้มสว่างระดับสอง (Bilevel Luminance Thresholding)

สำหรับภาพบางชนิดจะมีลักษณะวัตถุที่เราสนใจ ซึ่งมีความเข้มคงที่เมื่อเทียบกับพื้นหลังตัวอย่างได้แก่ ภาพของตัวอักษร (Text) เป็นต้น ภาพเหล่านี้จะมีความเข้มของวัตถุที่เราสามารถแยกออกพื้นหลังได้อย่างชัดเจน (มีความเข้มสองระดับได้แก่ความเข้มของวัตถุและความเข้มของพื้นหลัง)

การแยกข้อมูลภาพออกเป็นส่วนๆ สามารถทำได้โดยกำหนดค่าขีดแบ่งซึ่งเป็นค่าความเข้มให้มีค่าที่สามารถแยกความแตกต่างของวัตถุและพื้นหลังได้ตัวอย่างเช่น ภาพของตัวอักษรที่มีความเข้มของตัวอักษรเป็น 0 (สีดำ) และมีความเข้มของพื้นหลังเป็น 255 (สีขาว) ดังนั้นค่าขัดแย้งจึงควรจะมีค่าเท่ากับ 128 เพื่อที่จะให้สามารถแยกวัตถุออกจากพื้นหลังได้ โดยปกติแล้วการเลือกค่าขีดแบ่งจะขึ้นอยู่กับ ฮิสโตแกรมของภาพ ตามรูปที่ 2.2 แสดงการหาค่าขีดแบ่งโดยค่าขีดแบ่งควรที่จะเลือกค่าในฮิสโตแกรมที่อยู่จุดต่ำสุดระหว่างจุดยอด (peaks)

เมื่อ $$g(x,y)=\begin{cases} 1 \text{เมื่อ} (x,y) > T \\ 0 \text{เมื่อ} (x,y) \text{เป็นค่าอื่น} \end{cases} \quad (2.1)$$

$g(x,y)$ เป็นข้อมูลภาพ ณ ตำแหน่งที่ x,y T เป็นค่าขีดแบ่ง



รูปที่ 2.2 ไบโมดัล อิมเมจ ฮีสโตแกรม (Bimodal image histogram)

ตัวอย่าง โปรแกรมสำหรับการแยกข้อมูลภาพออกเป็นส่วนๆ มีลักษณะเป็นดังนี้คือ

```
int thres(a,b,t,N1,M1,N2,M2)
image a,b;
int t;
int N1,M1,N2,M2;
/*Subroutine to threshold an image
     a,b: buffers
     t: threshold (integer)
     N1, M1: start coordinates
     N2, M2: end  coordinates */
{ int I,j;
  for(i=N1; i<N2; i++)
      for(j=M1; j<M2; j++)
           if(a[i][j]<t)  b[i][j]=0;  else  b[i][j]=1;

      return(0);
}
```

ขั้นตอนวิธีที่ 2.1 การแยกข้อมูลภาพออกเป็นส่วนๆ

2.2.1.1.2 ค่าขีดแบ่งความเข้มสว่างหลายระดับ (Multilevel Luminance Thresholding)

สำหรับภาพที่จะประกอบด้วยหลาย ๆ วัตถุสามารถทำการแยกข้อมูลภาพออกเป็นส่วนๆ ได้โดยการใช้ค่าขีดแบ่งหลาย ๆ ค่า สำหรับภาพที่มี N วัตถุโดยที่แต่ละวัตถุจะมีช่วงกว้างของความเข้มเท่ากับ R_i (กำหนดได้ด้วยค่าขีดแบ่ง 2 ค่าคือ T_{i-1}, T_i) สามารถทำการแยกได้ดังนี้

$$g(x,y) = R_i \qquad \text{เมื่อ } T_{i-1} \le f(x,y) \le T_i),\ \ i = 1,...,N \qquad (2.2)$$

ค่าขีดแบ่งสามารถหาได้จากฮิสโตแกรมของภาพ แต่ในหลาย ๆ กรณีที่การเปลี่ยนแปลงของฮิสโตแกรมไม่สามารถบอกการเปลี่ยนแปลงระหว่างวัตถุได้อย่างชัดเจน วิธีการที่ง่ายที่สุดที่จะทำให้ฮิสโตแกรมสามารถหาค่าขีดแบ่งได้ง่ายขึ้นก็คือการใช้วิธี การหาขอบภาพ (Edge Detection) เพื่อพิจารณาจุดภาพต่าง ๆ ของภาพว่าเป็นขอบของวัตถุหรือไม่

ตัวอย่างโปรแกรมสำหรับขั้นตอนวิธีที่ 2

```
int segm(a,b,nr,N1,M1,N2,M2)
image a,b;
int nr;
int N1,M1,N2,M2;
/* Subroutine to segment an image in n regions
a,b: buffers
nr: number of regions (integer)
N1,M1: start coordinates
N2,M2: end  coordinates  */
{ int i,j;
 nr=256/nr;
 for(i=N1; i<N2; i++)
    for(j=M1; j<M2; j++)
       b[i][j]=((int)a[i][j]/nr)*nr;
    return 0;
          }
```

ขั้นตอนวิธีที่ 2 การแยกข้อมูลภาพออกเป็นส่วนๆ

2.2.1.2 วิธีการแยกส่วนภาพโดยอาศัยพื้นที่ (Region segmentation methods)

สำหรับในหัวข้อนี้จะเป็นการแยกข้อมูลภาพออกเป็นส่วนๆ โดยการพิจารณาเป็นกลุ่มของข้อมูลภาพ

2.2.1.2.1 รีเจียน โกอิ้ง (Region Growing)

วิธีการรีเจียน โกอิ้ง (Region Growing) จะนำจุดภาพข้างเคียงมาพิจารณาและจัดบริเวณ (Region) ของจุดภาพเหล่านี้เข้าไว้ด้วยกันโดยการพิจารณาถึงความเข้มของจุดภาพ (ค่าของจุดภาพมีค่าใกล้เคียงกัน) ในการแยกข้อมูลภาพออกเป็นส่วนๆ จะต้องมีการกำหนดบริเวณ (Region) ที่ต้องการจะแบ่ง R_i โดยที่แต่ละบริเวณจะต้องมีการกำหนดค่าความเข้มของจุดภาพเริ่มต้น ซึ่งมีไว้เพื่อใช้สำหรับบริเวณ (growth) ในการขยายบริเวณจำเป็นจะต้องมีกฎเพื่อใช้เป็นวิธีการขยายบริเวณรวมทั้งกฎของการตรวจสอบความเป็นอันหนึ่งอันเดียวกันกับบริเวณ (homogeneity) ของทุกระยะของการขยายตัว ในแต่ละระยะของการขยายบริเวณ $R_i^{(K)}, i = 1,...,N$ จะมีการตรวจสอบว่ามีจุดภาพที่ยังไม่ได้จัดกลุ่มหรือไม่ (เป็นจุดภาพทั้ง 8 ที่เชื่อมต่อ (8-

neighbourbood) อยู่กับจุดภาพที่ขอบของบริเวณ) และก่อนที่จะมีการกำหนดจุดภาพใด ๆ (x) เข้าสู่บริเวณใดบริเวณหนึ่งจะต้องมีการตรวจสอบอีกว่าบริเวณที่จะขยายนั้นยังคงมีความเป็นอันหนึ่งอันเดียวกันอีกหรือไม่ ดังสมการที่ (2.3)

$$P(R_i^{(k} \cup \{x\}) = TRUE \tag{2.3}$$

ประสิทธิภาพของการแยกข้อมูลภาพ วิธีนี้จะขึ้นอยู่กับการเลือกค่าความเข้มของจุดภาพเริ่มต้นในแต่ละกลุ่มซึ่งกำหนดโดยผู้ใช้ ซึ่งกระบวนการนี้สามารถหาได้อย่างอัตโนมัติจากฮิสโตแกรมของภาพโดยการพิจารณาค่าสูงสุดของฮิสโตแกรม (จุดยอด) มาใช้เป็นค่าความเข้มของจุดภาพเริ่มต้นของบริเวณได้ โดยปกติแล้วจะมีค่าความเข้มเริ่มต้นมากกว่าหนึ่งค่าต่อหนึ่งบริเวณ ดังนั้นจึงต้องมีวิธีการรวมบริเวณ (merging) เพื่อที่จะใช้ในการรวมบริเวณที่มีคุณสมบัติทาง สถิติ (พิจารณาค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน) ใกล้เคียงกัน

$$m_i = \frac{1}{n} \sum_{(k,l) \in R_i} f(k,l) \tag{2.4}$$

$$\sigma_i = \sqrt{\frac{1}{n} \sum_{(k,l) \in R_i} (f(k,l) - m_i)^2} \tag{2.5}$$

เมื่อ m_i เป็นค่าเฉลี่ยของค่าความเข้มในกลุ่มที่ i ที่มีจำนวนจุดภาพเท่ากับ n

ค่าเหล่านี้นำไปใช้ในการตัดสินใจว่าจะสามารถรวมบริเวณ R_1, R_2 เข้าไว้ด้วยกันได้หรือไม่ ถ้าค่าเฉลี่ยของบริเวณ m_1, m_2 มีค่าใกล้เคียงกันก็สามารถรวมกันได้ดังนี้คือ

$$|m_1 - m_2| < k\sigma_i \qquad i = 1,2 \tag{2.6}$$

2.2.1.2.2 การแยกและการรวม (Split and Merge)

การรวมบริเวณ (Merge region) การรวมบริเวณเป็นวิธีการแยกข้อมูลภาพวิธีหนึ่งซึ่งจะมีขั้นตอนวิธีดังนี้ ถ้าสมมุติว่ามีการสแกนไปยังจุดภาพต่าง ๆ ภายในภาพจากจุดภาพแรกไปยังจุดภาพสุดท้ายของภาพไปตามแถวและหลักตามลำดับ ในช่วงระหว่างการสแกนก็จะมีการกำหนดจุดภาพนั้นไปยังบริเวณต่าง ๆ สมมติปัจจุบันอยู่ที่จุดภาพ (k,l) ดังนั้นจุดภาพแรกจนถึงจุดภาพ $(k-1,l)$ ได้ถูกกำหนดให้อยู่ในบริเวณต่าง ๆ หมดแล้ว ดังนั้นจุดภาพที่ (k,l) จึงเปรียบเสมือนเป็นอีกบริเวณหนึ่งที่พยายามที่จะรวมเข้ากับบริเวณที่มีอยู่ก่อนหน้านั้น (บริเวณที่มีทั้งหมด R_i แต่จะเลือกเอาเฉพาะบริเวณที่อยู่ข้างเคียงเท่านั้น ได้แก่กลุ่มที่มีจุดภาพ ณ ตำแหน่ง

$(k-1,l)$, $(k+1,l)$, $(k,l-1)$ และ $(k,l+1)$ เป็นสมาชิกอยู่) หากพบว่าไม่สามารถทำการรวมเข้ากับบริเวณใดได้ก็ให้สร้างบริเวณใหม่ขึ้นมา

ประสิทธิภาพของวิธีนี้จะขึ้นอยู่กับกฎของการรวมบริเวณ ($P(R_i \cup (k,l))$) ของจุดภาพ (k,l) เข้ากับบริเวณ R_i กฎของการรวมจะขึ้นอยู่กับค่าเฉลี่ยของกลุ่ม m_i และค่าเบี่ยงเบนมาตรฐาน s_i ตามที่อธิบายไว้ในสมการที่ (2.4) และ (2.5) ตามลำดับ แต่สำหรับการรวมกันของ $R \cup (k,l)$ จะมีการเปลี่ยนแปลงการหาค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานเป็นดังนี้คือ

$$m_i^{'} = \frac{1}{n+1}(f(k,l) + nm_i) \tag{2.7}$$

$$\sigma_i^{'} = \sqrt{\frac{1}{n+1}\left(n\sigma_i^2 + \frac{n}{n+1}[f(k,l) - m_i]^2\right)} \tag{2.8}$$

การรวมกันสามารถทำได้เมื่อค่าความเข้มของจุดภาพ $f(k,l)$ มีค่าใกล้เคียงกับค่าเฉลี่ย $m_i^{'}$ ดังต่อไปนี้คือ

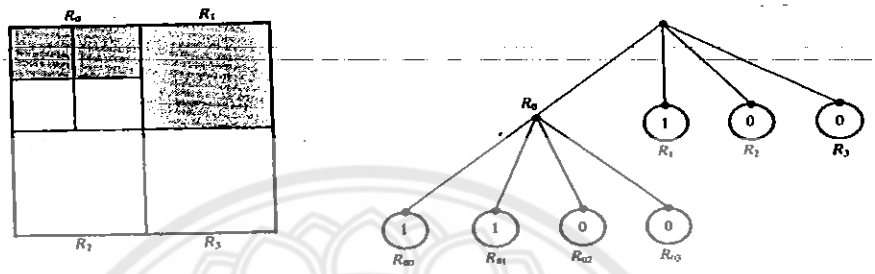
$$|f(k,l) - m_i| <= T_i(k,l) \tag{2.9}$$

เมื่อ T_i เป็นค่าขีดแบ่ง ที่มีค่าเปลี่ยนแปลงอยู่ระหว่างบริเวณ R_i กับความเข้มของจุดภาพ $f(k,l)$ ซึ่งสามารถหาได้จาก

$$T_i(k,l) = \left(1 - \frac{\sigma_i^{'}}{m_i^{'}}\right)T \tag{2.10}$$

ถ้าหากไม่มีบริเวณใดเลยที่จุดภาพที่ (k,l) สามารถรวมเข้าด้วยกันได้ก็ให้สร้างบริเวณใหม่ขึ้นมา ถ้าหากมีมากกว่าหนึ่งบริเวณที่จุดภาพที่ (k,l) สามารถรวมเข้าด้วยกันได้ก็ให้รวมเข้าด้วยกันกับบริเวณที่มีค่าความแตกต่าง $|f(k,l) - m_i|$ ที่มีค่าน้อยที่สุด การขยายตัวของบริเวณจะขึ้นอยู่กับค่าขีดแบ่ง (T) ซึ่งถ้ามีค่าน้อย ๆ ก็จะทำให้ค่า $T_i(k,l)$ มีค่าน้อยด้วย (สำหรับทุก ๆ บริเวณ) และการรวมกันก็จะทำได้ยากยิ่งขึ้น แต่ถ้าค่าขีดแบ่งมีค่ามาก ๆ ก็จะทำให้ในแต่ละบริเวณมีค่าความแตกต่างของสมาชิกภายในบริเวณมากขึ้น (ค่าเบี่ยงเบนมาตรฐานมีค่ามากขึ้น) นอกจากนี้ค่า ขีดแบ่ง $T_i(k,l)$ ยังจะขึ้นอยู่กับค่าอัตราส่วน $\sigma_i^{'}/m_i^{'}$ ถ้าในบริเวณสมาชิกที่มีค่าความเข้มที่แตกต่างกันน้อยก็จะทำให้ค่าอัตราส่วนนี้มีค่าเข้าใกล้ศูนย์และค่า $T_i(k,l)$ จะมีค่าเข้าใกล้ T ดังนั้น T จึงเป็นค่าความแตกต่างของ $|f(k,l) - m_i|$ ที่มากที่สุดที่สามารถยอมรับได้ และถ้าความแตกต่างความเข้มของสมาชิกในบริเวณยิ่งมีค่าสูงขึ้น (less homogeneous) ค่าอัตราส่วน $\sigma_i^{'}/m_i^{'}$ ก็จะมีค่าสูงขึ้นด้วย

การแยกบริเวณ (Split region) เป็นลักษณะของการแยกข้อมูลภาพอีกวิธีหนึ่งที่มีลักษณะตรงข้ามกับ การรวมบริเวณ (เป็นลักษณะการทำงานจากบนลงล่าง) โดยเริ่มต้นจะมีการสมมุติว่าทั้งภาพจะมีเพียงหนึ่งบริเวณเท่านั้นโดยถ้าหากว่าไม่เป็นความจริงก็ให้ทำการแยกบริเวณนี้ออกเป็นสี่บริเวณย่อยและจะมีการพิจารณาลักษณะนี้เรื่อยๆจนกระทั่งได้บริเวณของภาพที่มีสมาชิกของบริเวณที่มีค่าใกล้เคียงกันในระดับที่สามารถยอมรับได้ (homogeneous) ขั้นตอนวิธีการนี้แสดงดังรูปที่ 2.3



(ก) (ข)

รูปที่ 2.3 (ก) ภาพต้นฉบับ (ข) แผนภูมิต้นไม้ (Quad tree representation)

การตรวจสอบว่าบริเวณใด สามารถยอมรับได้ หรือไม่ทำได้โดยการคำนวณผลต่าง ของค่าความเข้มของจุดภาพที่ได้จากค่าความเข้มสูงสุดลบกับ จุดภาพที่มีความเข้มน้อยสุดแล้วนำผลที่ได้ไปเปรียบเทียบกับค่าขีดแบ่งว่ามีค่าน้อยกว่าหรือไม่

คุณสมบัติของการแยกบริเวณที่น่าสนใจคือถ้าภาพเริ่มต้นมีขนาดเป็น NxN ก็จะทำให้ขนาดเท่ากับเป็นกำลังของ 2 ($N = 2^n$) ทุกกลุ่มที่ถูกสร้างด้วยขั้นตอนวิธีการแยกบริเวณ จะเป็นสี่เหลี่ยมขนาดเท่ากับ MxM เมื่อ M เป็นกำลังของ 2 ($M = 2^m$), m≤n และเนื่องจากขั้นตอนวิธีนี้จะเป็นลักษณะของการเรียกตัวเอง (recursion) ดังนั้นจึงสามารถแสดงเป็นลักษณะของรูปต้นไม้ (tree) ได้ซึ่งแต่ละโหนดจะมีการแตกออกมาเป็น 4 กลุ่มซึ่งเรียกว่า แผนภูมิต้นไม้ (Quad tree) ดังแสดงดังรูปที่ 2.3 (ข)

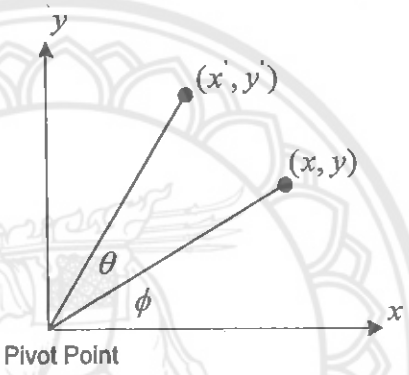
การแยกและการรวม (Split and merge) เป็นการใช้เฉพาะขั้นตอนวิธีการแยกบริเวณจะมีข้อเสียคือวิธีการนี้จะทำให้มีการสร้างบริเวณขึ้นมาใหม่ R_i, R_j ซึ่งบริเวณเหล่านี้อาจจะสามารถรวมเป็นบริเวณเดียวกัน $P(R_i \cup R_j) = TRUE$ ซึ่งในความเป็นจริงแล้วควรที่จะมีการรวมทั้งสองบริเวณเข้าด้วยกันดังนั้นจึงได้มีการนำเอาการรวมบริเวณเข้ามาใช้ด้วยและเรียกว่าขั้นตอนวิธีการแยกและการรวม (Split and merge) ซึ่งจะมีลักษณะการทำงานดังนี้ คือ ถ้าบริเวณ R เป็นบริเวณที่ไม่เข้ากัน (inhomogeneous) ก็ให้ทำการแยกออกเป็น 4 บริเวณย่อยเรื่อย ๆ ถ้าหากบริเวณสองบริเวณ R_i, R_j สามารถรวมเข้าด้วยกันได้ (homogeneous) ($P(R_i \cup R_j) = TRUE$) ก็ให้ทำการรวมเข้าด้วยกัน

ขั้นตอนวิธีนี้จะหยุดก็ต่อเมื่อไม่สามารถที่จะทำการแยกเป็นบริเวณย่อย ๆ ได้อีกรวมทั้งไม่สามารถรวมบริเวณต่าง ๆ เข้าไว้ด้วยกันได้อีก

2.2.2 การแปลงข้อมูลภาพในสองมิติ (Two Dimensional Geometric Transformation)

การแปลงข้อมูลภาพสำหรับสองมิตินั้น การแปลงภาพเป็นสิ่งจำเป็นมากสำหรับการประมวลผลภาพดิจิทัลเนื่องจากเป็นกระบวนการที่จะนำไปสู่การวิเคราะห์ภาพ (Digital Image Analysis) การแปลงข้อมูลภาพแบบพื้นฐาน (Basic Transformation) ประกอบด้วย

การหมุนภาพ (Rotation) เป็นการหมุนตำแหน่งของภาพในระนาบ xy รอบจุดหมุน (Pivot Point)



รูปที่ 2.4 แสดงการหมุนตำแหน่งของภาพรอบจุดหมุน

จากรูปที่ 2.4 จะได้ว่า

$$x = r\cos(\phi)$$
$$y = r\sin(\phi) \qquad (2.11)$$

และ

$$x' = r\cos(\phi+\theta) = r(\cos\phi\cos\theta - \sin\phi\sin\theta)$$
$$y' = r\sin(\phi+\theta) = r(\sin\phi\cos\theta - \cos\phi\sin\theta) \qquad (2.12)$$

เพราะฉะนั้นจากสมการ

$$\text{P1: } (2 \le N(p_0) \le 6) \,\&\, \&(T(p_0) = 1) \,\&\, \&(p1.p3.p5 = 0) \,\&\, \&(p3.p5.p7 = 0)$$
$$\text{P2: } (2 \le N(p_0) \le 6) \,\&\, \&(T(p_0) = 1) \,\&\, \&(p1.p3.p7 = 0) \,\&\, \&(p1.p5.p7 = 0)$$

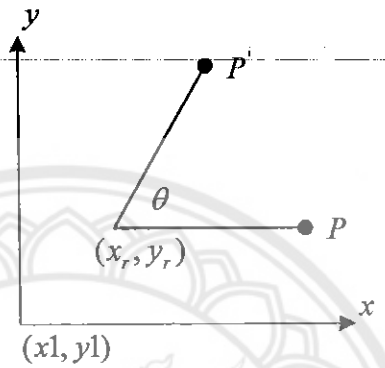
จะได้สมการของการหมุนรอบจุดหมุน ดังนี้คือ

$$x' = x\cos(\theta) - y\sin\theta$$
$$y' = x\sin(\theta) - y\cos\theta \qquad (2.13)$$

ซึ่งสามารถเขียนให้อยู่ในรูปแบบของเมทริกซ์ได้ มีลักษณะดังนี้คือ $P' = R.P$ เมื่อ

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad P = \begin{bmatrix} x \\ y \end{bmatrix} \text{ และ } R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.14}$$

การหมุนภาพเมื่อจุดหมุนไม่ได้อยู่ที่จุดกำเนิด พิจารณาเมื่อจุดหมุนไม่ได้อยู่ในตำแหน่ง (0,0) (ย้ายไปอยู่ที่ตำแหน่ง (x_r, y_r))



รูปที่ 2.5 แสดงการย้ายตำแหน่งของจุด P

วิธีการในการหมุนภาพเมื่อจุดหมุนไม่ได้อยู่ที่จุดกำเนิด สามารถทำได้ดังนี้คือ

1. ทำการเปลี่ยนจุด s,6o ไปยังจุดกำเนิด

$$x1 = x - x_r$$
$$y1 = y - y_r \tag{2.15}$$

2. ทำการหมุนรอบจุด กำเนิด
3. ย้ายกลับไปยังจุดเดิม โดยการบวกด้วย x_r และ y_r
4. สมการการหมุนรอบจุดหมุน ใด ๆ ที่ไม่ใช้จุดกำเนิด มีลักษณะดังนี้คือ

$$x' = (x - x_r)\cos\theta - (y - y_r)\sin\theta + x_r$$
$$y' = (x - x_r)\sin\theta + (y - y_r)\cos\theta + y_r \tag{2.16}$$

ดังนั้นเมทริกซ์ของการแปลงแบบต่าง ๆ จะมีลักษณะเป็นดังนี้

การหมุนภาพ

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.17}$$

การย่อและขยายภาพ

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.18}$$

ตัวอย่างของการหมุนภาพ ถ้ามีการแปลงภาพ 2 ครั้ง

หมุนภาพโดยใช้ $R(\theta_1)$

หมุนภาพโดยใช้ $R(\theta_2)$

สมการของการหมุนภาพจะมีลักษณะดังนี้

$$P' = R(\theta_2).R(\theta_1).P \tag{2.19}$$

$$= \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_{21} & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_2+\theta_1) & -\sin(\theta_2+\theta_1) & 0 \\ \sin(\theta_2+\theta_1) & .\cos(\theta_2+\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.20}$$

ซึ่งจากการทดลองพบว่าการหมุนภาพจะมีคุณสมบัติของการสลับที่ดังนี้

$$R(\theta_2).R(\theta_1) = R(\theta_1)R(\theta_2) \tag{2.21}$$

ถ้ามีการแปลงภาพ 2 ครั้ง

หมุนภาพโดยใช้ $S(S_{x1}, S_{y1})$

หมุนภาพโดยใช้ $S(S_{x2}, S_{y2})$

สมการของการย่อขยายภาพจะมีลักษณะดังนี้

$$P' = S(S_{x2}, S_{y2}).S(S_{x1}, S_{y1}).P \tag{2.22}$$

$$= \begin{bmatrix} S_{x2} & 0 & 0 \\ 0 & S_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_{x1} & 0 & 0 \\ 0 & S_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

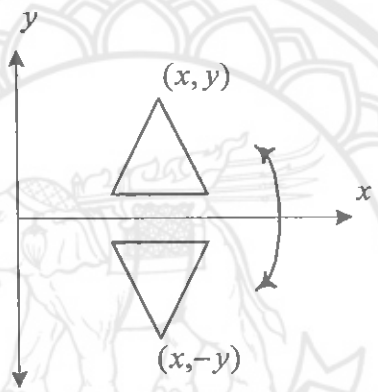
$$= \begin{bmatrix} S_{x1}S_{x2} & 0 & 0 \\ 0 & S_{y1}S_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.23)$$

ซึ่งจากการทดลองพบว่าการย่อขยายภาพจะมีคุณสมบัติของการสลับที่ดังนี้

$$S(S_{x2}, S_{y2}).S(S_{x1}, S_{y1}) = S(S_{x1}, S_{y1}).S(S_{x2}, S_{y2}) \quad (2.24)$$

การแปลงภาพแบบอื่น ๆ

การสะท้อนของภาพสามารถเกิดขึ้นได้ดังนี้ คือ การสะท้อนตามแกน x การสะท้อนตามแกน x จะมีลักษณะตัวอย่างดังภาพรูปที่ 2.6



รูปที่ 2.6 แสดงการสะท้อนของภาพตามแกน X

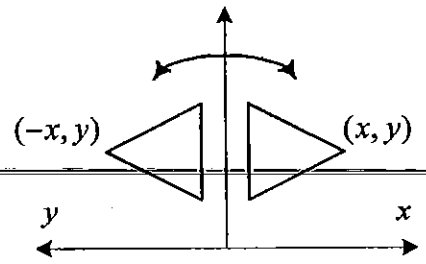
สามารถเขียนให้อยู่ในรูปแบบสมการได้ดังนี้

$$x' = x$$
$$y' = -y \quad (2.25)$$

สามารถใช้เมทริกซ์ในการแปลงได้ดังนี้

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.26)$$

การสะท้อนตามแกน y การสะท้อนตามแกน x จะมีลักษณะตัวอย่างดังรูปที่ 2.7



รูปที่ 2.7 แสดงการสะท้อนของภาพตามแกน y

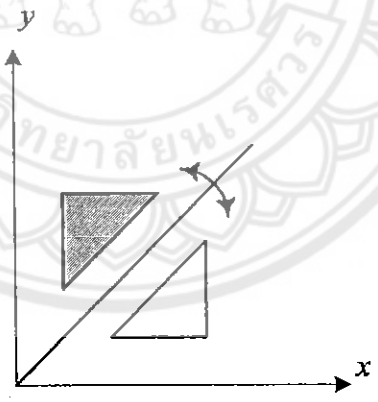
สามารถเขียนให้อยู่ในรูปแบบสมการได้ดังนี้

$$x' = -x$$
$$y' = y \tag{2.27}$$

สามารถใช้เมตริกในการแปลงได้ดังนี้

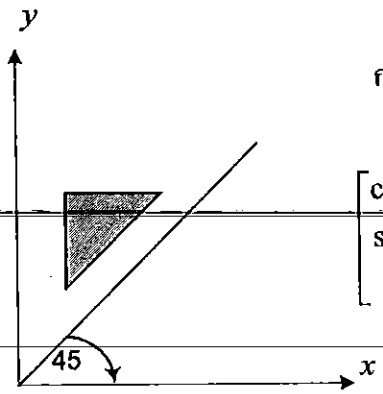
$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

การสะท้อนรอบแกนใดๆ สำหรับการสะท้อนรอบเส้นตรงใดๆ สามารถทำได้ดังนี้ (สมมุติเป็นการสะท้อนรอบแกน y=x)



รูปที่ 2.8 แสดงการสะท้อนของภาพรอบแกนใด ๆ

หมุนภาพไปที่ระนาบแกน x

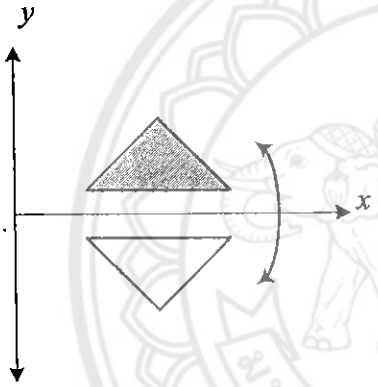


การหมุนภาพไปยังระนาบแกน x ใช้เมทริกซ์ดังนี้

$$\begin{bmatrix} \cos(45) & -\sin(45) & 0 \\ \sin(45) & \cos(45) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

รูปที่ 2.9 แสดงเมทริกซ์สำหรับหมุนภาพไปยังระนาบแกน x

ทำการสะท้อนภาพกับแกน x

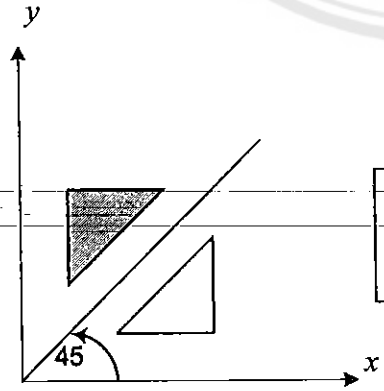


การสะท้อนของภาพกับแกน x ใช้เมทริกซ์ดังนี้

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

รูปที่ 2.10 แสดงการสะท้อนของภาพกับแกน x

หมุนภาพกลับ



การหมุนภาพไปยังระนาบเดิม ใช้เมทริกซ์ดังนี้

$$\begin{bmatrix} \cos(-45) & -\sin(-45) & 0 \\ \sin(-45) & \cos(-45) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

รูปที่ 2.11 แสดงการหมุนภาพกลับไปยังระนาบเดิม

2.2.3 การประมวลผลภาพกับรูปร่างและโครงร่างของภาพ (Morphological Image Processing)

การประมวลผลภาพโดยการเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพ การกระทำพื้นฐานโดยทั่วไปได้แก่ การขยายภาพ การกร่อนภาพ และ สเกเลตัน (Skeleton) โดยการขยายภาพ คือ การขยายภาพโดยมีสัดส่วนเท่ากันทั่วทั้งภาพ การกร่อนภาพ (Erosion) ส่วนการทำสเกเลตันเป็นการหาโครงสร้างหลักของวัตถุซึ่งจะกล่าวโดยละเอียดต่อจากนี้

นอกจากการกระทำพื้นฐานดังที่ได้กล่าวข้างต้นแล้วยังมีการกระทำอื่น ๆ อีกที่ไม่ได้กล่าว ได้แก่การเปิด และ การปิด เป็นต้น

2.2.3.1 เทคนิคของฮิต-ออร์-มิส (Hit-or- Miss)

การกระทำพื้นฐานสำหรับการกระทำกับรูปร่างหรือโครงสร้างของภาพ ไม่ว่าจะเป็นการย่อหรือการขยายภาพ จำเป็นที่จะต้องมีการนำเอาเทคนิคของฮิต-ออร์-มิส มาใช้ แนวคิดคือการกำหนดให้มีแม่แบบที่มีขนาดเล็ก ๆ และเป็นจำนวนคี่ (โดยทั่วไปจะมีค่าเท่ากับ 3x3) ไปเปรียบเทียบกับข้อมูลภาพ โดยการเปรียบเทียบจะทำตลอดทั้งภาพตั้งแต่ต้นภาพจนถึงท้ายภาพ ถ้าข้อมูลของภาพมีลักษณะเหมือนกับแมตริกดังกล่าวผลลัพธ์ที่ได้จะขึ้นอยู่กับจุดภาพ ที่เป็นศูนย์กลางของแม่แบบซึ่งจะถูกกำหนดให้เป็นค่าตามต้องการ (1 หรือ 0) แต่ถ้าข้อมูลในแม่แบบไม่เหมือนกับข้อมูลภาพข้อมูลผลลัพธ์ที่ได้จะมีค่าตรงกันข้าม การกระทำพื้นฐานสำหรับรูปร่างหรือโครงสร้างพื้นฐาน พิจารณาข้อมูลภาพจะเป็นลักษณะดังนี้

$$\begin{matrix} 1 & * & 1 & * & 1 \\ * & 1 & * & 1 & * \\ 1 & * & 1 & * & 1 \end{matrix}$$

รูปที่ 2.12 แสดงลักษณะข้อมูลภาพ

เนื่องจากสามารถแทนลักษณะภาพได้ดังรูปที่ 2.12 ดังนั้นสามารถกำหนดให้มี ข้อมูลภาพสำหรับการกระทำได้ ดังนี้คือ

$$A = \begin{matrix} 1 & * & 1 & * & 1 \\ * & 1 & * & 1 & * \\ 1 & * & 1 & * & 1 \end{matrix} \qquad B = \begin{matrix} * & * & * & 1 & 1 \\ * & * & * & 1 & 1 \\ * & * & * & 1 & 1 \end{matrix}$$

$$A \cup B = \begin{matrix} 1 & * & 1 & 1 & 1 \\ * & 1 & * & 1 & 1 \\ 1 & * & 1 & 1 & 1 \end{matrix} \qquad A \cap B = \begin{matrix} * & * & * & * & 1 \\ * & * & * & 1 & * \\ * & * & * & * & 1 \end{matrix}$$

รูปที่ 2.13 แสดงการกระทำการเบื้องต้น

หมายเหตุ ข้อมูลภาพตามรูปที่ 2.13 แสดงถึงจุดภาพที่ทราบ (ค่าเท่ากับ 1) และค่าที่ไม่ทราบ (แสดงด้วย *) ชุดของข้อมูลภาพจะขยายออกไปทางด้านบน ล่าง ซ้าย ขวาแบบไม่จำกัด

```
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * *
* * * 1 * 1 * 1 * * *
* * * * 1 * 1 * * * *
* * * 1 * 1 * 1 * * *
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * *
```

รูปที่ 2.14 แสดงถึงจุดภาพที่ทราบ

วงกลมที่ล้อมรอบจุดภาพ 1 ตามรูปที่ 2.14 แสดงตำแหน่งเริ่มต้นของภาพ (กำเนิด)

2.2.3.2 การขยายภาพ (Dilation)

การขยายภาพในที่นี้จะพิจารณาสำหรับข้อมูลภาพที่เป็นแบบไบนารี โดยการใช้เทคนิคการ ฮิต-ออร์-มิส ตามที่ได้กล่าวไว้ในหัวข้อ 2.2.3.1 การขยายภาพจะทำได้โดยกำหนดแม่แบบ (ซึ่งสามารถสร้างได้จาก * และ 1 โดยมีจุดเริ่มต้นที่กำหนดโดยวงกลม) และนำแม่แบบ นี้สแกนไปบนข้อมูลภาพตามลำดับตลอดทั้งภาพซึ่งใน ขณะที่จุดเริ่มของแม่แบบตรงกับตำแหน่งข้อมูลภาพที่จุดภาพมีค่าเท่ากับ 1 นั้นก็จะทำการยูเนียนแม่แบบ นี้เข้ากับข้อมูลภาพดังรูปที่ 2.15

ข้อมูลภาพ

```
* * * * * * 1 * * 1 *
* * * * * * 1 * * * 1
* * * * * 1 1 * 1 1 *
* * * * 1 1 1 1 1 1 1
* * * * 1 1 1 1 1 * 1
* * * * 1 1 1 1 1 1 1
* * * * 1 1 1 1 1 1 1
```

แม่แบบ

```
1 *
1 1
```

รูปที่ 2.15 แสดงแม่แบบข้อมูลภาพ

ข้อมูลแถวแรกของภาพเป็นดังนี้

* * * * * * 1 * * 1 *

รูปที่ 2.16 แสดงข้อมูลภาพแถวแรก

เมื่อทำการยูเนียนกับแม่แบบ ณ. ตำแหน่งข้อมูลภาพที่จุดภาพเท่ากับ 1 ในแถวแรก

* * * * * * 1 * * * *
* * * * * * 1 1 * 1 *

รูปที่ 2.17 แสดงการยูเนียนจุดภาพเท่ากับ 1 ในแถวแรก

และเมื่อยูเนียนกับแม่แบบเข้ากับจุดภาพที่มีค่าเท่ากับ 1 ณ. ตำแหน่งจุดภาพที่สองในแถวแรก

* * * * * * 1 * * * *
* * * * * * 1 1 * 1 1

รูปที่ 2.18 แสดงการยูเนียนจุดภาพเท่ากับ 1 ณ. ตำแหน่งจุดภาพที่สองในแถวแรก

และเมื่อทำการยูเนียนทั้งภาพจะได้ภาพสุดท้ายดังนี้

* * * * * * 1 * * 1 * *
* * * * * * 1 1 * 1 1 *
* * * * * 1 1 1 1 1 1 1
* * * * 1 1 1 1 1 1 1 1
* * * * 1 1 1 1 1 1 1 1
* * * * 1 1 1 1 1 1 1 1
* * * * 1 1 1 1 1 1 1 1
* * * * 1 1 1 1 1 1 1 1

รูปที่ 2.19 แสดงภาพผลลัทธ์จากการยูเนียน

2.2.3.3 การกร่อนภาพ (Erosion)

การกร่อนภาพเป็นลักษณะของการลบข้อมูลภาพบริเวณขอบของภาพ การกร่อนภาพที่สามารถทำได้มีลักษณะคล้ายกับการขยายภาพโดยการสร้างแม่แบบขึ้นแล้วนำแม่แบบไปสแกนตามข้อมูลภาพ สำหรับทุกตำแหน่งที่เลื่อนแม่แบบไปบนภาพก็จะมีการเปรียบเทียบกับข้อมูลภาพ ถ้าข้อมูลภาพมีค่าเหมือนกับแม่แบบ จะทำการกำหนดค่าข้อมูลภาพในตำแหน่งที่ตรงกับจุดเริ่มต้นของแม่แบบ ถูกกำหนดให้มีค่าเท่ากับ 1

i50091960
ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปร.
จ332ก
2546

ข้อมูลภาพ

```
* * * * * * 1 * * 1 *
* * * * * * 1 * * * 1
* * * * * 1 1 * 1 1 *
* * * * 1 1 1 1 1 1 1
* * * * 1 1 1 1 1 * 1
* * * * 1 1 1 1 1 1 1
* * * * 1 1 1 1 1 1 1
```

แม่แบบ

```
1 *
1 1
```

รูปที่ 2.20 การแสดงแม่แบบกร่อนภาพกับข้อมูลต้นฉบับ

ผลที่ได้จะมีเพียง 3 ตำแหน่งเท่านั้นที่มีค่าเหมือนกับ แม่แบบ

```
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * * 1 * * 1 *
* * * * * * * * * * *
* * * * * * * * 1 * *
* * * * * * * * * * *
```

รูปที่ 2.21 แสดงภาพผลลัพธ์ของการกร่อน

ผลที่ได้ตามรูปที่ 2.21 ข้อมูลภาพที่ผ่านการกระทำกับแม่แบบ แล้วพบว่ามีข้อ**มูลของภาพเพียง** 3 ตำแหน่งเท่านั้นที่เหมือนกับ แม่แบบ ถ้ามีการเปลี่ยน แม่แบบ เป็น $\begin{matrix}1 & 1\\ 1 & 1\end{matrix}$ ผลที่**ได้มีลักษณะดังนี้ คือ**

```
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * *
* * * * * 1 * * 1 * *
* * * * 1 1 1 1 1 * *
* * * * 1 1 1 1 * * *
* * * * 1 1 1 1 1 1 *
```

รูปที่ 2.22 ผลลัพธ์เมื่อแม่แบบเป็น 1 ทั้งหมด

ผลที่ได้ตามรูปที่ 2.22 จะเห็นว่าจะเป็นการย่อขนาดของภาพแต่สามารถย่อขนาดได้น้อยกว่าเมื่อใช้แม่แบบ $\begin{matrix} 1 & * \\ 1 & 1 \end{matrix}$ ซึ่งได้ผลเป็นที่น่ายอมรับมากกว่าดังนั้นในการเลือกแม่แบบ เป็นสิ่งที่สำคัญอย่างหนึ่งในการย่อและขยายภาพ

2.2.3.4 การปิดและการเปิด (Opening and Closing)

การเปิด (Opening) กำหนดให้ OPEN (I, T) เป็นการเปิดของภาพ I โดยใช้ แม่แบบ T ซึ่งมีลักษณะดังสมการต่อไปนี้

$$OPEN (I, T) = D(E(I)) \qquad (2.29)$$

จากสมการจะได้ว่าการกระทำ OPEN คือการนำข้อมูลภาพ I ผ่านการทำกร่อนภาพแล้วตามด้วยการขยายภาพโดยใช้แม่แบบชุดเดียวกันคือ T ดังรูปที่ 2.23

ภาพต้นแบบ เปิดโดย

0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	4		0	0	0	0	0	0	1	4
0	0	0	0	1	2	4	5		0	0	0	0	1	2	4	4
0	0	1	1	2	5	5	4		0	0	0	0	1	2	4	4
0	0	0	0	0	1	2	3		0	0	0	0	0	1	2	3
0	0	0	0	0	0	1	3		0	0	0	0	0	0	1	3
0	0	0	0	0	1	2	3		0	0	0	0	0	1	2	3
0	0	0	0	1	2	2	3		0	0	0	0	0	1	2	3

1
1

รูปที่ 2.23 แสดงการเปิดด้วย 1 ทั้งหมด

การปิด (Closing) กำหนดให้ CLOSE (I, T) เป็นการปิดของภาพ I โดยใช้ แม่แบบ T ซึ่งมีลักษณะดังสมการต่อไปนี้

$$CLOSE (I, T) = E (D(I)) \qquad (2.30)$$

จากสมการจะได้ว่าการทำการกระทำ CLOSE คือการนำข้อมูลภาพ I ผ่านการทำการขยายภาพแล้วตามด้วยการกร่อนภาพโดยใช้ แม่แบบ ชุดเดียวกันคือ T ได้ดังรูปที่ 2.24

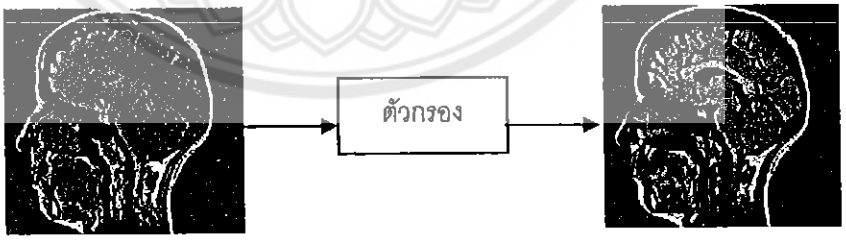
ภาพต้นแบบ	ปิดโดย 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 4 0 0 0 0 1 2 4 5 0 0 1 1 2 5 5 4 0 0 0 0 0 1 2 3 0 0 0 0 0 0 1 3 0 0 0 0 0 1 2 3 0 0 0 0 1 2 2 3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 4 0 0 0 0 1 2 4 5 0 0 1 1 2 5 5 4 0 0 0 0 0 1 2 3 0 0 0 0 0 1 2 3 0 0 0 0 0 1 2 3 0 0 0 0 1 2 2 3

รูปที่ 2.24 แสดงการปิดด้วย 1 ทั้งหมด

2.2.4 การกรองข้อมูลภาพ (Image Filtering)

การนำภาพไปผ่านตัวกรองสัญญาณเพื่อให้ได้ภาพผลลัพธ์ออกมา มีคุณสมบัติแตกต่างจากภาพเริ่มต้น วัตถุประสงค์หลักของการกรองข้อมูลภาพคือการเน้น (Enhance) หรือลดทอน (attenuate) คุณสมบัติบางประการของภาพ เพื่อให้ได้ภาพที่มีคุณสมบัติตามต้องการ

การกรองข้อมูลภาพ คือ การประมวลผลภาพอย่างหนึ่งที่จำเป็นมาก เนื่องจากในการใช้งานจริง ภาพที่ได้มามักมีสัญญาณรบกวน หรือสัญญาณไม่พึงประสงค์อื่นๆ ปะปนอยู่ด้วย การกรองข้อมูลภาพสามารถปรับปรุงให้ภาพมีคุณสมบัติที่ดีขึ้น เหมาะแก่การประมวลผลในขั้นต่อไป การกรองข้อมูลภาพสามารถแสดงได้ดังรูปที่ 2.25



รูปที่ 2.25 การกรองข้อมูลภาพ

องค์ประกอบสำคัญของการกรองข้อมูลภาพคือ ตัวกรอง หากเปรียบภาพเป็นสัญญาณไฟฟ้าที่มีความถี่ต่างๆ ผสมกันอยู่ ตัวกรองก็คือวงจรไฟฟ้าที่ทำหน้าที่เลือกหรือกรองให้สัญญาณไฟฟ้าที่มีความถี่ในช่วงที่ต้องการผ่านออกไปได้ คุณสมบัติของตัวกรองคือตัวกำหนดคุณสมบัติของภาพผลลัพธ์ อาจมองข้อมูลของภาพๆ หนึ่งให้เป็นสัญญาณๆ หนึ่งได้ ด้วยการ

กำหนดให้ระดับความเข้มแสงของแต่ละจุดคือขนาด (Amplitude) ของสัญญาณ ณ. ตำแหน่งนั้นๆ ข้อแตกต่างระหว่างสัญญาณไฟฟ้ากับภาพ คือ

1. ขนาดของสัญญาณไฟฟ้า คือ ค่าแรงดันหรือกระแส แต่ขนาดของข้อมูลภาพคือระดับความเข้มแสงของจุดภาพ

2. การเปลี่ยนแปลงของสัญญาณไฟฟ้าเป็นการเปลี่ยนแปลงเทียบกับเวลา ความถี่ของสัญญาณไฟฟ้าถูกกำหนดโดยอัตราการเปลี่ยนแปลงของขนาดของสัญญาณในหนึ่งช่วงเวลา แต่การเปลี่ยนแปลงของข้อมูลภาพเป็นการเปลี่ยนแปลงเทียบกับตำแหน่งของจุดภาพ ความถี่ของการเปลี่ยนแปลงขึ้นอยู่กับอัตราการเปลี่ยนระดับความเข้มแสงของจุดที่อยู่ถัดกันไป

3. สัญญาณไฟฟ้าเป็นสัญญาณมิติเดียว (amplitude vs time) แต่ภาพเป็นสัญญาณ 2 มิติ (intensity vs X & Y)

ตัวกรอง คือ ระบบๆ หนึ่งซึ่งรับสัญญาณเข้า (input) ประมวลผลสัญญาณ และส่งสัญญาณออก (output) โดยทั่วไปตัวกรองจะถูกสร้างให้เป็นระบบเชิงเส้น (linear system) เนื่องจากออกแบบได้ง่าย และมีประสิทธิภาพดี รวมทั้งปัจจุบันมีทฤษฎี และเทคนิคมากมายเกี่ยวกับการออกแบบตัวกรองสัญญาณแบบเชิงเส้นในการกรองข้อมูลภาพ มักพิจารณาว่าภาพคือ สัญญาณ 2 มิติที่ประกอบขึ้นจากสัญญาณความถี่ต่างๆ ผสมกันอยู่ในสัดส่วนที่ต่างกัน การออกแบบตัวกรองจึงเป็นการกำหนดว่าต้องการกำจัดสัญญาณความถี่ใดออกไป หรือต้องการเลือกสัญญาณความถี่ใดบ้าง

ประเภทของตัวกรอง แบ่งออกได้เป็น 3 ประเภทตามลักษณะการเลือกความถี่ คือ

1. ตัวกรองความถี่ต่ำผ่าน (Low-pass Filter)
2. ตัวกรองความถี่สูงผ่าน (High-pass Filter)

ค่าพารามิเตอร์หลักในการกำหนดคุณสมบัติของตัวกรองคือ ค่าความถี่คัตออฟ (cut-off frequency) ความถี่คัตออฟ คือ ความถี่ที่ระบุจุดตัดของสัญญาณว่าจะให้ผ่าน หรือไม่ผ่าน ตัวอย่างเช่น ตัวกรองความถี่ต่ำผ่านที่มีค่าความถี่คัตออฟเท่ากับ 1,000 เฮิรตซ์จะยอมให้สัญญาณที่มีความถี่ต่ำกว่า 1,000 เฮิรตซ์ผ่านไปได้ แต่จะไม่ยอมให้สัญญาณที่มีความถี่สูงกว่า 1,000 เฮิรตซ์ผ่าน สำหรับตัวกรองความถี่สูงผ่านจะทำงานตรงข้ามกับตัวกรองความถี่ต่ำผ่าน คือไม่ยอมให้สัญญาณที่มีความถี่ต่ำกว่าความถี่คัตออฟผ่านไปได้ แต่จะยอมให้ความถี่ที่สูงกว่าความถี่คัตออฟผ่านได้ วงจรกรองแถบความถี่ผ่านยอมให้สัญญาณในช่วงความถี่หนึ่งผ่านไปได้ หากสัญญาณมีความถี่อยู่นอกช่วงจะถูกลดทอนหรือไม่ยอมให้ผ่านไป สำหรับวงจรหยุดแถบความถี่จะมีลักษณะการทำงานที่ตรงข้ามกัน คือ จะลดทอนสัญญาณที่มีความถี่ในช่วงที่กำหนดลง และจะผ่านความถี่ที่อยู่นอกช่วง ในการกรองสัญญาณใดๆ จะต้องทราบความถี่ หรือช่วงความถี่ของสัญญาณที่ต้องการและสัญญาณที่ไม่ต้องการ จากนั้นจะเลือกตัวกรองที่เหมาะสมมาใช้เพื่อกำจัดสัญญาณที่ไม่ต้องการออก และหรือ เน้นสัญญาณที่ต้องการให้เด่นชัดยิ่งขึ้น ตัวอย่างเช่น สัญญาณรบกวน

3. การกรองโดยการเฉลี่ยจากหลายภาพ

หากมีชุดของภาพคุณภาพต่ำหลาย ๆ ภาพซึ่งถ่ายจากมุมกล้องเดียวกัน สามารถสร้างภาพใหม่ที่มีคุณภาพสูงกว่าจากชุดภาพนั้นได้ หากสัญญาณรบกวนเกิดขึ้นแบบสุ่ม ภาพที่เก็บแต่ละครั้งย่อมมีลักษณะแตกต่างกัน หากความเข้มแสงของจุดในภาพหนึ่งถูกรบกวน สามารถนำข้อมูลความเข้มแสงของจุด จากภาพอื่น ณ ตำแหน่งเดียวกันมาแทน แต่ละจุดในภาพผลลัพธ์ที่ได้จะเกิดจากการเฉลี่ย (หรือเลือก) จากจุดที่ตรงกันของภาพต่าง ๆ ในชุดภาพ

2.2.4.1 การกรองข้อมูลภาพโดยใช้ค่าเฉลี่ยแบบคณิตศาสตร์ (Mean filtering)

วิธีการนี้จะใช้ค่าเฉลี่ยแบบคณิตศาสตร์ของจุดทั้งหมด หากมีภาพขนาด NxM ทั้งหมด K ภาพสามารถคำนวณหาภาพใหม่ได้ดังนี้

$$\hat{I}(x,y)=\frac{1}{K}\sum_{j=1}^{K} I_j(x,y) \qquad (2.31)$$

$\hat{I}(x,y)$ คือความเข้มแสงของจุด ณ ตำแหน่ง (x,y) ในภาพผลลัพธ์

$\hat{I}(x,y)$ คือความเข้มแสงของจุด ณ ตำแหน่ง (x,y) ในภาพที่ j

วิธีนี้เป็นการลดทอนสัญญาณรบกวน ภาพที่ได้จะมีสัญญาณรบกวนลดลง

2.2.4.2 การกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน (Median filtering)

วิธีการนี้จะนำเอาความเข้มแสงของจุดที่ตรงกันในภาพต่างๆ มาเรียงลำดับ (Sort) จากน้อยไปหามาก จากนั้นจะเลือกค่าที่อยู่ตรงกลางไปใช้ หากจำนวนภาพทั้งหมดเป็นจำนวนคู่ ค่าทั้งสองที่อยู่ตรงกลางจะนำมาหาค่าเฉลี่ย วิธีการนี้จะต้องใช้การเรียงลำดับซึ่งเป็นกระบวนการที่ใช้เวลาในการคำนวณสูง แต่ข้อดีคือไม่สูญเสียความคมชัด ได้ดังรูปที่ 2.26

ภาพที่ 1				ภาพที่ 2				ภาพที่ 3					ผลลัพธ์			
1	2	1	3	2	3	4	3	3	2	1	4		2	2	1	3
4	2	2	1	5	3	4	1	2	1	4	0		4	2	4	1
0	1	1	3	3	2	4	2	1	4	2	0	=	1	2	2	2
2	2	1	1	1	3	1	2	2	4	0	2		2	3	1	2

รูปที่ 2.26 เป็นวิธีการกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน

2.2.4.3 การกรองข้อมูลภาพโดยใช้ค่าฐานนิยม (Modal filtering)

วิธีการนี้คล้ายกับวิธีใช้ค่ามัธยฐาน แต่ไม่ใช้การเรียงลำดับข้อมูล ระดับความเข้มแสงที่ใช้บ่อยที่สุดจะถูกเลือกไปใช้ วิธีนี้เสมือนการลงคะแนนเสียง ผู้ที่ได้คะแนนเสียงสูงที่สุดคือผู้ชนะ วิธีนี้เหมาะสำหรับการลดสัญญาณรบกวนที่เกิดขึ้นไม่บ่อย ได้ดังรูปที่ 2.27

ภาพที่ 1		ภาพที่ 2		ภาพที่ 3		ผลลัพธ์
1 2 1 3		2 3 4 3		3 2 1 4		2 2 1 3
4 2 2 1		5 3 4 1		2 1 4 0		4 2 4 1
0 1 1 3	,	3 2 4 2	,	1 4 2 0	=	1 2 2 2
2 2 1 1		1 3 1 2		2 4 0 2		2 3 1 2

รูปที่ 2.27 การกรองข้อมูลภาพโดยใช้ค่าฐานนิยม

2.2.4.4 การกรองโดยใช้หน้าต่าง (Mask filtering)

การกรองข้อมูลภาพวิธีนี้จะใช้หน้าต่างในการกำหนดขอบเขตของการพิจารณาเพื่อหาระดับความเข้มแสงของจุดต่างๆ ในภาพผลลัพธ์ ความเข้มแสงของจุดที่อยู่รอบๆ จุดกึ่งกลางของหน้าต่างจะถูกนำมาหาค่าเฉลี่ย ค่าเฉลี่ยที่ได้คือค่าความเข้มแสงของจุดในภาพผลลัพธ์ หน้าต่างจะถูกเลื่อนไปยังตำแหน่งต่าง ๆ ในภาพจนครบทุกจุดดังรูปที่ 2.28

รูปที่ 2.28 การกรองโดยใช้หน้าต่าง

จากรูปที่ 2.28 จะเห็นว่ามีหน้าต่างขนาด 3x3 ครอบอยู่ที่มุมบนด้านซ้ายของภาพเริ่มต้น ความเข้มแสง ณ. จุดกึ่งกลางของหน้าต่างมีค่าเท่ากับ 1 ความเข้มแสงของจุดภาพในภาพผลลัพธ์ ณ. ตำแหน่งที่ตรงกับกึ่งกลางของหน้าต่างที่ครอบอยู่บนภาพเริ่มต้น (จุด A) สามารถคำนวณได้จากค่าเฉลี่ยความเข้มแสงของทุกจุดในหน้าต่าง การหาค่าเฉลี่ยสามารถทำได้ 3 แบบคือ การหาค่าเฉลี่ยแบบคณิตศาสตร์ การหาค่าเฉลี่ยแบบมัธยฐาน และการหาค่าเฉลี่ยแบบฐานนิยม

การหาค่าเฉลี่ยแบบคณิตศาสตร์ทำได้โดยการหาผลรวมของค่าความเข้มแสงของจุดทุกจุดในหน้าต่าง แล้วหารด้วยจำนวนจุดทั้งหมดในหน้าต่าง จากตัวอย่างในรูปที่ 2.29 (ก) ความเข้มแสงที่จุด A มีค่าเท่ากับ (0+0+0+0+1+2+0+2+3) / 9 = 8/9 ค่าความเข้มแสงที่จุดอื่นๆ สามารถคำนวณได้โดยการเลื่อนหน้าต่างให้จุดกึ่งกลางตรงกับจุดที่ต้องการหาค่า

การหาค่าเฉลี่ยแบบมัธยฐานทำได้โดยการนำค่าทั้งหมดในตารางมาเรียงลำดับ (Sort) จากน้อยไปหามาก (หรือจากมากไปหาน้อยก็ได้) จากนั้นจะเลือกค่าที่อยู่ตรงกลางของลำดับเป็นค่าความเข้มแสงของจุดในภาพผลลัพธ์ หากจำนวนจุดในหน้าต่างเป็นจำนวนคู่ ผลลัพธ์จะคำนวณได้จากการเฉลี่ยค่าระหว่างจุดกึ่งกลางทั้งสอง จากตัวอย่างในรูปที่ 2.29 (ข) เมื่อเรียงลำดับความเข้มแสงจะได้ลำดับดังนี้ (0 0 0 0 0 1 2 2 3) ค่าที่อยู่ตรงกลางคือ 0 ดังนั้นความเข้มแสงที่จุด A มีค่าเท่ากับ 0

การหาค่าเฉลี่ยแบบฐานนิยมทำได้โดยการเลือกระดับความเข้มแสงที่ใช้บ่อยที่สุดในหน้าต่างมาเป็นคำตอบ ปัญหาที่อาจเกิดจากการใช้วิธีนี้คือ มีระดับความเข้มแสงที่ใช้บ่อยที่สุดมากกว่า (มีหลายคำตอบ) วิธีการแก้ไขคือการหาค่าเฉลี่ย หรือเปลี่ยนไปใช้ การหาค่าเฉลี่ยแบบมัธยฐาน จากตัวอย่างในรูป 2.29 (ค) ค่าความเข้มแสงที่ใช้บ่อยที่สุดคือ 0 ดังนั้นความเข้มแสงที่จุด A มีค่าเท่ากับ 0

นอกจากการหาค่าผลลัพธ์โดยวิธีหาค่าเฉลี่ยทั้ง 3 ดังที่ได้กล่าวมาแล้ว ยังมีวิธีการหาผลลัพธ์อีกวิธีหนึ่งคือการหาค่าเฉลี่ยจากจุด k จุดที่มีค่าความเข้มใกล้กับค่าความเข้มแสงของจุดกึ่งกลางของหน้าต่าง วิธีนี้เรียกว่า k-closest averaging การคำนวณหาผลลัพธ์เริ่มจากการนำค่าความเข้มแสงของทุกจุดในหน้าต่างมาเรียงลำดับจากน้อยไปหามาก จากนั้นค่าที่อยู่รอบๆ ค่าของจุดกึ่งกลางหน้าต่างจำนวน k ค่าจะถูกเลือกมาเพื่อหาค่าเฉลี่ย ค่าเฉลี่ยนี้คือความเข้มแสงของจุดในภาพผลลัพธ์ ในการหาค่าเฉลี่ย อาจนำค่าของความเข้มสีที่จุดกึ่งกลางมาคิดด้วยก็ได้ จากตัวอย่างในรูปที่ 2.29 (ง) เมื่อเรียงลำดับความเข้มแสงจะได้ลำดับดังนี้ (0 0 0 0 0 1 2 2 3) หากกำหนดให้ k = 4 และไม่นำค่าที่จุดกึ่งกลาง (1) มาคิด ค่าความเข้มแสงที่จุด A มีค่าเท่ากับ (0+0+2+2)/4 = 1 หากนำค่าที่จุดกึ่งกลางมาคิด ค่าความเข้มแสงที่จุด A จะมีค่าเท่ากับ (0+0+1+2+2)/5 = 1

1 2 2 1	0 1 1 0	0 0 0 0	1 2 1 3
1 2 2 2	1 2 2 1	0 2 2 1	2 2 3 1
1 2 2 1	0 2 1 0	0 0 0 0	1 4 2 1
(ก)	(ข)	(ค)	(ง)

รูปที่ 2.29 การกรองข้อมูลภาพโดยใช้หน้าต่าง

(ก) ใช้การเฉลี่ยแบบคณิตศาสตร์ (แสดงผลลัพธ์หลังการปัดเศษ)

(ข) ใช้มัธยฐาน

(ค) ใช้ฐานนิยม (ใช้มัธยฐานแทนสำหรับจุดที่มีปัญหา)

(ง) ใช้ k-closest averaging (แสดงผลลัพธ์หลังการปัดเศษ)

2.2.4.5 แม่แบบสำหรับกรองความถี่ต่ำผ่าน (Low Pass Filter)

แม่แบบขนาด 3 x 3 ต่อไปนี้สามารถใช้เพื่อลดการเปลี่ยนแปลงอย่างฉับพลันของค่าความเข้มแสงในภาพ

1	1	1
1	1	1
1	1	1

ผลของการคอนโวลูชันกับแม่แบบนี้ จะเหมือนกับการหาผลรวมของจุดภาพทั้ง 9 จุดที่แม่แบบซ้อนทับอยู่ การเปลี่ยนแปลงใดๆ ในบริเวณดังกล่าวจะถูกเฉลี่ยให้มีความราบเรียบ ผลที่ได้คือสัญญาณที่มีการเปลี่ยนแปลง (เช่นสัญญาณรบกวนความถี่สูง) จะถูกลดทอน ภาพที่ได้จะมีความคมลดลง คุณสมบัติเช่นนี้เปรียบได้กับการกรองสัญญาณความถี่ต่ำผ่าน

แม่แบบการกรองความถี่ต่ำผ่านที่นิยมใช้อีกแบบหนึ่งคือ

1	3	1
3	16	3
1	3	1

แม่แบบนี้จะเน้นความสำคัญของจุดที่อยู่ตรงกลางแม่แบบเป็นพิเศษ โดยจะให้จุดกลางมีน้ำหนัก 50% ของทั้งหมด และให้น้ำหนักรวมของจุดทั้ง 4 ที่อยู่ด้านบน ด้านล่าง ด้านซ้าย และด้านขวาของจุดกลางมีค่าเท่ากับ 40% ส่วนจุดมุมทั้ง 4 มีน้ำหนักเพียง 10% โดยจุดที่อยู่ใกล้จุดศูนย์กลางจะมีน้ำหนักมากกว่าจุดที่อยู่ห่างออกไป

2.2.4.6 แม่แบบสำหรับกรองความถี่สูงผ่าน (High Pass Filter)

สัญญาณความถี่สูงคือสัญญาณที่มีการเปลี่ยนแปลงค่าไปมาอย่างรวดเร็ว ต่างกับสัญญาณความถี่ต่ำ ซึ่งมีการเปลี่ยนค่าอย่างช้าๆ หรือไม่เปลี่ยนแปลงเลย การกรองความถี่สูงผ่าน(High pass filter) ก็คือการกรองสัญญาณที่เพิ่มความแรงของสัญญาณที่มีความถี่สูงและลดความแรงของสัญญาณที่มีความถี่ต่ำ แม่แบบต่อไปนี้ใช้สำหรับการกรองความถี่สูงผ่าน

0	-1	0
-1	4	-1
0	-1	0

จะเห็นว่าผลรวมของทุกค่าในแม่แบบมีค่าเท่ากับศูนย์ ซึ่งหมายความว่า ถ้าวางแม่แบบนี้ลงบนบริเวณของภาพที่มีค่าความเข้มแสงคงที่ ผลลัพธ์ที่ได้จะมีค่าเป็นศูนย์ อย่างไรก็ตาม ถ้าค่าที่บริเวณตรงกลางแตกต่างกับค่ารอบๆ ผลลัพธ์ที่ได้จะแสดงค่าความแตกต่างยิ่งขึ้น

ภาพหลังจากการกรองความถี่สูงผ่าน หลังจากการกรองความถี่ต่ำผ่าน

0 0 0 0 0		
0 1 1 1 0	2 1 2	4 6 4
0 1 1 1 0	1 0 1	6 9 6
0 1 1 1 0	1 0 1	6 9 6
0 1 1 1 0	1 –5 1	11 14 11
0 1 6 1 0	-4 20 -4	11 14 11
0 1 1 1 0	2 –4 2	9 11 9
0 0 0 0 0		

รูปที่ 2.30 การกรองภาพด้วยแม่แบบ

จากรูปที่ 2.30 จะเห็นว่าหลังการกรองความถี่สูงผ่านขอบภาพจะเด่นชัด ส่วนที่เป็นค่าคงที่จะกลายเป็นศูนย์ และส่วนที่มีการเปลี่ยนแปลงจาก 1 เป็น 6 ถูกขยายเป็นจาก –4 ไป 20 สำหรับการกรองความถี่ต่ำผ่าน การเปลี่ยนถูกลดทอนลง ภาพผลลัพธ์มีความราบเรียบขึ้น

2.2.5 การทำภาพเนกาทีฟ (Image Negative)

การทำภาพเนกาทีฟนั้นจะเป็นการกลับค่าระดับเทาของภาพ ให้เป็นค่าที่ตรงกันข้ามกับค่าระดับเทาของภาพต้นฉบับซึ่งจะมีประโยชน์ในการปรับปรุงรายละเอียดของภาพ ที่มีสีอ่อน (สีเทา หรือขาว) ทำให้ง่ายต่อการวิเคราะห์ภาพ (Digital Image Analysis) ซึ่งในภาพระดับเทาจะหาค่าได้จากสมการ

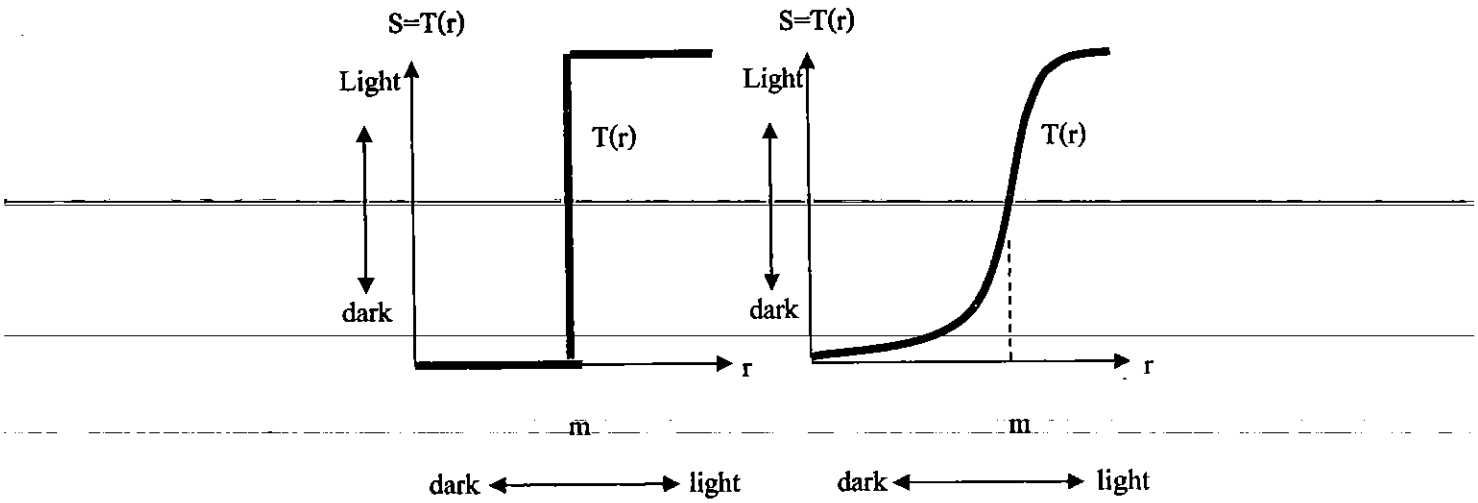
$$A[x,y] = 255 - A[x,y] \tag{2.32}$$

2.2.6 การกระทำเลขคณิต (Arithmetic Operation)

การกระทำเลขคณิตประกอบไปด้วยการบวก (Addition) การลบ (subtraction) การคูณ (multiplication) และการหาร (division) ค่าของจุดภาพด้วยค่าคงที่ ดังนั้นการบวก หรือการลบจะเป็นการเพิ่มและลดค่าความสว่าง (brightness) ของภาพ ตามลำดับ

2.2.7 การหาค่าขีดแบ่ง (thresholding)

การแปลงภาพระดับเทาในลักษณะนี้นั้นจะทำให้เกิดภาพสองระดับ หรือเรียกว่าภาพไบนารี (Binary image) ซึ่งการปรับปรุงภาพลักษณะนี้นั้นผลลัพธ์ที่ได้จะขึ้นอยู่กับค่าของแต่ละตำแหน่งนั้น



รูปที่ 2.31 ฟังก์ชันการแปลงระดับเทาเพื่อปรับปรุงความแตกต่างของภาพ

2.2.8 การฮิสโตแกรม (histrogram)

ฮิสโตแกรมเป็นการนับจำนวนของจุดภาพภายในรูปภาพที่ต้องการหาและจึงนำไปสร้างเป็นเส้นกราฟขึ้นมาซึ่งในการฮิสโตแกรมนั้นอาจจะนำไปช่วยในการหาค่าขีดแบ่งของภาพได้ดังแสดงในตัวอย่างของโปรแกรมดังนี้คือ

```
for(x=0; x<256; x++){ hist1[x] = 0; }
    for(y=0; y<iH; y++){
        for(x=0; x<iW; x++){ptr1=arrImage[x][y];
              hist1[ptr1]=hist1[ptr1]+1;}}
          for(x=0; x<256; x++){
              Series1->Add( hist1[x],"", clRed);}
```

2.2.9 การหาขอบภาพ (Edge Detection)

ในการหาขอบภาพเป็นการหาเส้นขอบของรูปภาพ ซึ่งการหาขอบภาพนั้นจะช่วยให้สามารถค้นหาเส้นขอบของรูปภาพที่ต้องการค้นหาได้รวดเร็วขึ้นและสะดวกขึ้น

2.2.9.1 การหาขอบภาพด้วยวิธีโรเบิร์ต (Roberts Edge Detection) การหาขอบภาพด้วยวิธีนี้นั้นจะมีสมการดังนี้คือ

$$\sqrt{[I(r,c)-I(r-1,c-1)]^2+[I(r,c-1)-I(r-1,c)]^2} \qquad (2.33)$$

และจะได้

$$|I(r,c)-I(r-1,c-1)|+|I(r,c-1)-I(r-1,c)| \qquad (2.34)$$

บทที่ 3
วิธีดำเนินการวิจัย

3.1 ศึกษาและออกแบบการสร้างคอมโพเนนต์ บน Borland C++ Builder

สำหรับการสร้างคอมโพเนนต์จะประกอบไปด้วย 3 ขั้นตอน คือ

1. เลือกชนิดต้นแบบ (Ancestor Type)
2. เขียนโปรแกรมสร้างคอมโพเนนต์ (ใช้ Component Wizard ช่วยเบื้องต้น)
3. ทดลองใช้คอมโพเนนต์

3.1.1 การเลือกชนิดต้นแบบ (Ancestor Type)

เมื่อคิดที่จะสร้างคอมโพเนนต์ขึ้นมา การเลือกชนิด ที่ต้องการเลียนแบบเป็นสิ่งที่สำคัญที่สุด โดยที่ให้เลือกชนิดทุกแบบที่มีใน C++ Builder โดยสามารถแบ่งชนิดออกเป็นส่วนใหญ่ๆ ดังนี้ ชนิดพื้นฐานของคอมโพเนนต์ใน C++ Builder มีชนิดพื้นฐานอยู่ 5 แบบคือ

1. TOject เป็นแบบที่พื้นฐานที่สุด ดังนั้นวัตถุใน C++ Builder ต้องมีชนิดนี้เป็นพื้นฐาน แม้วัตถุนั้นไม่ใช่คอมโพเนนต์ก็ตาม
2. TComponent เป็นพื้นฐานของคอมโพเนนต์ทุกตัว
3. TGaphicControl เป็นพื้นฐานของคอมโพเนนต์ที่ใช้ในการแสดงรูปต่างๆ โดยไม่อาศัยการทำงานของ วินโดวส์
4. TWin Control เป็นคอมโพเนนต์ที่อาศัยวินโดว์เป็นตัวช่วยในการควบคุมการทำงาน เช่น Button และ Radio Button หรือ Edit เป็นต้น
5. TCustomControl เป็นการรวบรวมความสามารถของ Twin Control และใช้ TGraphicControl เข้าด้วยกัน จะใช้ในกรณีที่นำคอมโพเนนต์ที่อยู่ใน TWin Control มาแก้ไข

3.2 การพัฒนาโปรแกรม

การออกแบบโปรแกรม

3.2.1 TImageProcessing (Arithmetic)

TImageProcessing (Arithmetic)		
- setarth	:	unsigned int
- Arithmetic	:	TBitmap
- ArithmeticToImage	:	TImage
+ InitializeData2	:	Tbitmap
+ SetWidth2	:	unsigned int
+ SetHeight2	:	unsigned int
+ SetData2	:	unsigned int
+ GetData2	:	unsigned int
+ CreateGreyPalette	:	Void
+ GetWidth2	:	Void
+ GetHeight2	:	Void
+ Getarth	:	Void
+ AssignToBitmap2	:	TBitmap

InitializeData2 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการประมวลตามคำสั่งของการใช้งานในการทำให้ภาพเรียบ

SetWidth2 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight2 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData2 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปในอเรย์ขนาด 2 มิติ

GetData2 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPaletteเป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth2 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไป เพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight2 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

Getarth เป็นเมตธอดที่ใช้ในการส่งค่าของหน้าต่าง (mask) ที่ใช้ในการประมวลผลร่วมกับ อเรย์เพื่อทำให้ภาพเรียบขึ้น

AssignToBitmap2 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

Setarth เป็นเมตธอดที่ใช้ในการกำหนดขนาดของหน้าต่าง (mask) ที่ใช้ในการประมวลผลร่วมกับอเรย์เพื่อทำให้ภาพเรียบขึ้น

Arithmetic เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการทำให้ภาพเรียบขึ้น

ArithmeticToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.2 TImageProcessing (Median)

TImageProcessing (Median)		
- Setmask	:	unsigned int
- Median	:	Tbitmap
- MedianToImage	:	TImage
+ InitializeData6	:	TBitmap
+ SetWidth6	:	unsigned int
+ SetHeight6	:	unsigned int
+ SetData6	:	unsigned int
+ GetData6	:	unsigned int
+ Getmask	:	unsigned int
+ CreateGreyPalette	:	Void
+ GetWidth6	:	Void
+ GetHeight6	:	Void
+ AssignToBitmap6	:	TImage

InitializeData6 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการประมวลตามคำสั่งของการใช้งานในการทำให้ภาพเรียบด้วยวิธีเรียงค่าและกำหนดค่ากลางของหน้าต่าง (mask)

SetWidth6 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight6 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData6 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปในอเรย์ขนาด 2 มิติ

GetData6 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth6 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight6 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

Getmask เป็นเมตธอดที่ใช้ในการส่งค่าของหน้าต่าง (mask) ที่ใช้ในการประมวลผลร่วมกับอเรย์เพื่อทำให้ภาพเรียบขึ้นด้วยวิธีเรียงค่าและกำหนดค่ากลางของหน้าต่าง (mask)

AssignToBitmap6 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

Setmask เป็นเมตธอดที่ใช้ในการกำหนดขนาดของหน้าต่าง (mask) ที่ใช้ในการประมวลผลร่วมกับอเรย์เพื่อทำให้ภาพเรียบขึ้นด้วยวิธีเรียงค่าและกำหนดค่ากลางของหน้าต่าง (mask)

Median เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการทำให้ภาพเรียบขึ้น

MedianToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.3 TIMageProcessing (Invert)

TImageProcessing (Invert)		
- Invert	:	TBitmap
- InvertToImage	:	TImage
+ InitializeData	:	Tbitmap
+ SetWidth	:	unsigned int
+ SetHeight	:	unsigned int
+ SetData	:	unsigned int
+ GetData	:	unsigned int
+ CreateGreyPalette	:	Void
+ GetWidth	:	Void
+ GetHeight	:	Void
+ Getarth	:	Void
+ AssignToBitmap	:	TBitmap

InitializeData เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการประมวลตามคำสั่งในการทำการอินเวิร์ทภาพ

SetWidth เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปใน อเรย์ ขนาด 2 มิติ

GetData เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

Invert เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการอินเวิร์ทภาพ

InvertToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.4 TImageProcessing(Histrogram)

TImageProcessing(Histrogram)		
- His	:	Const
- Histrogram	:	TBitmap
+ InitialData1	:	unsigned int
+ InitializeData1	:	TBitmap
+ SetWidth1	:	unsigned int
+ SetHeight1	:	unsigned int
+ SetData1	:	unsigned int
+ GetData1	:	unsigned int
+ SetHis1	:	unsigned int
+ GetWidth1	:	Void
+ GetHeight1	:	Void
+ SetRed1	:	unsigned int
+ SetGreen1	:	unsigned int
+ SetBlue1	:	unsigned int
+ SetIntensity1	:	unsigned int
+ GetRed1	:	unsigned int
+ GetGreen1	:	unsigned int
+ GetBlue1	:	unsigned int
+ GetIntensity1	:	unsigned int
+ CreateGreyPalette	:	Void

InitializeData1เป็นเมตธอดที่ใช้ในการเตรียมพร้อม อเรย์เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการประมวลตามคำสั่งในการหาฮิสโตแกรม (Histogram) ของภาพ

SetHis1 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปในอเรย์เพื่อรวบรวมสถิติเพื่อหาฮิสโต-แกรม (Histogram) ของภาพ

SetWidth1 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight1 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData1 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปในอเรย์ ขนาด 2 มิติ

GetData1 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

His เป็นเมตธอดที่ใช้ในการส่งค่าออกไปแสดงผลเพื่อใช้ในการหาฮิสโตแกรม (Histogram) ของภาพ

GetWidth1 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight1 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

Histrogram เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการหาฮิสโตแกรม (Histogram) ของภาพ

3.2.5 TImageProcessing (Threshold)

TImageProcessing(Threshold)		
- Setths	:	unsigned int
- Threshold	:	TBitmap
- ThresholdToImage	:	TImage
+ InitializeData3	:	Tbitmap
+ SetWidth3	:	unsigned int
+ SetHeight3	:	unsigned int
+ SetData3	:	unsigned int
+ GetData3	:	unsigned int
+ CreateGreyPalette	:	Void
+ GetWidth3	:	Void
+ GetHeight3	:	Void
+ Getarth	:	Void
+ AssignToBitmap3	:	Tbitmap

InitializeData3 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บอเรย์ภายในแบบ 2 มิติ และทำการประมวลตามคำสั่งในการสร้างภาพไบนารี (Binary) ได้จากการใช้ค่าขีดแบ่ง (Threshold Value)

Setths เป็นเมตธอดที่ใช้ในการรับค่าเพื่อเป็นตัวกลางในการ สร้างภาพไบนารี (Binary) โดยการใช้ค่าขีดแบ่ง (Threshold Value)

SetWidth3 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight3 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData3 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปใน อเรย์ ขนาด 2 มิติ

GetData3 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

Getths เป็นเมตธอดที่ใช้ในการส่งค่าออกไปเพื่อกำหนดค่าขีดแบ่งของภาพในการประมวลผล

GetWidth3 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight3 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap3 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

Threshold เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการอินเวิร์ทภาพ

ThresholdToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.6 TImageProcessing (highpass)

TImageProcessing(Highpass)		
- Highpass	:	Tbitmap
- HighpassToImage	:	TImage
+ InitializeData7	:	TBitmap
+ SetWidth7	:	unsigned int
+ SetHeight7	:	unsigned int
+ SetData7	:	unsigned int
+ GetData7	:	unsigned int
+ CreateGreyPalette	:	Void
+ GetWidth7	:	Void
+ GetHeight7	:	Void
+ AssignToBitmap7	:	TImage

InitializeData7 เป็นเมตธอดที่ใช้ในการเตรียมพร้อม อเรย์ เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการประมวลตามคำสั่งในการทำให้ภาพคมชัดขึ้น

SetWidth7 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight7 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData7 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปใน อเรย์ ขนาด 2 มิติ

GetData7 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth7 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight7 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap7 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

Highpass เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการทำให้ภาพคมชัดขึ้น

HighpassToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.7 TImageProcessing (Robert)

TImageProcessing(Robert)		
- Robert	:	Tbitmap
- RobertToImage	:	TImage
+ InitializeData4	:	TBitmap
+ SetWidth4	:	unsigned int
+ SetHeight4	:	unsigned int
+ SetData4	:	unsigned int
+ GetData4	:	unsigned int
+ CreateGreyPalette	:	Void
+ GetWidth4	:	Void
+ GetHeight4	:	Void
+ AssignToBitmap4	:	TImage

InitializeData4 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์ เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการการหาขอบภาพแบบโรเบริต

SetWidth4 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight4 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData4 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปใน อเรย์ ขนาด 2 มิติ

GetData4 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth4 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight4 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap4 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

Robert เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการหาขอบภาพแบบโรเบริต

RobertToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.8 TImageProcessing (Sobel)

TImageProcessing(Sobel)		
- Sobel	:	TBitmap
- SobelToImage	:	TImage
+ InitializeData5	:	TBitmap
+ SetWidth5	:	unsigned int
+ SetHeight5	:	unsigned int
+ SetData5	:	unsigned int
+ GetData5	:	unsigned int
+ CreateGreyPalette	:	Void
+ GetWidth5	:	Void
+ GetHeight5	:	Void
+ AssignToBitmap5	:	TBitmap

InitializeData5 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์ เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการการหาขอบภาพแบบโซเบล

SetWidth5 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight5 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData5 เป็นเมตธอตที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปในอเรย์ ขนาด 2 มิติ

GetData5 เป็นเมตธอตที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอตที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth5 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight5 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap5 เป็นเมตธอตที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

Sobel เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการหาขอบภาพแบบโซเบล

SobelToImage เป็นเมตธอตที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.9 TImagrProcessing(Dilation,Erotion)

TImageProcessing(Dilation,Erotion)		
- InPutEroDila_tion :	TBitmap	
- InPutEroDila_tionToImage :	TImage	
- MaskEroDila_tion :	TBitmap	
- OutPutEroDila_tionToImage :	TImage	
+ InitializeData18 :	TBitmap	
+ SetWidth18 :	unsigned int	
+ SetHeight18 :	unsigned int	
+ SetData18 :	unsigned int	
+ GetData18 :	unsigned int	
+ Getmask18 :	Void	
+ CreateGreyPalette :	Void	
+ InitializeData19 :	TBitmap	
+ SetWidth19 :	unsigned int	
+ SetHeight19 :	unsigned int	
+ SetData1 :	unsigned int	
+ GetData19 :	unsigned int	
+ Getmask19 :	Void	
+ MorpImg18 :	int	
+ Setmask18 :	unsigned int	
+ GetWidth18 :	Void	
+ GetHeight18 :	Void	
+ AssignToBitmap18 :	Tbitmap	
+ Setmask19 :	unsigned int	
+ GetWidth19 :	Void	
+ GetHeight19 :	Void	
+ AssignToBitmap19 :	Tbitmap	

InitializeData18 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการขยายภาพ (Dilation) และการกร่อนภาพ(Erotion)

SetWidth18 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight18 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData18 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปใน อเรย์ ขนาด 2 มิติ

GetData18 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth18 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight18 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap18 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

InPutEroDila_tion เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการขยายภาพ (Dilation) และการกร่อนภาพ(Erotion)

AssignToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

MorpImg18 เป็นเมตธอดที่ใช้ในการประมวลผลระหว่างภาพต้นฉบับและภาพที่เป็นหน้าต่าง (mask) และให้ได้ภาพที่ขยาย (Dilation) แล้ว

InitializeData19

เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการขยายภาพ (Dilation)

SetWidth19 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight19 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData19 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปใน อเรย์ ขนาด 2 มิติ

GetData19 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

GetWidth19 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight19 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap19 เป็นเมตธอตที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

MaskEroDila_tion เป็นเมตธอตที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปเป็นหน้าต่าง (mask) เพื่อประมวลผลร่วมกับภาพต้นแบบในการขยายภาพ (Dilation)

OutPutEroDila_tionToImage เป็นเมตธอตที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.10 TImageProcessing (Opening,Closing)

TImageProcessing (Opening,Closing)		
- InPutOpenClos_ing	:	TBitmap
- InPutOpenClos_ingToImage	:	TImage
- MaskOpenClos_ing	:	TBitmap
- OutPutOpenClos_ingToImage	:	TImage
+ InitializeData14	:	TBitmap
+ SetWidth14	:	unsigned int
+ SetHeight14	:	unsigned int
+ SetData14	:	unsigned int
+ GetData14	:	unsigned int
+ Getmask14	:	Void
+ CreateGreyPalette	:	Void
+ InitializeData15	:	TBitmap
+ SetWidth15	:	unsigned int
+ SetHeight15	:	unsigned int
+ SetData15	:	unsigned int
+ GetData15	:	unsigned int
+ Getmask15	:	Void
+ MorpImg14	:	int
+ MorpImg15	:	int
+ Setmask14	:	unsigned int
+ GetWidth14	:	Void
+ GetHeight14	:	Void
+ AssignToBitmap14	:	Tbitmap
+ Setmask15	:	unsigned int
+ GetWidth15	:	Void
+ GetHeight15	:	Void
+ AssignToBitmap15	:	Tbitmap

InitializeData14 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการเปิด (Opening) และการปิด (Closing)

SetWidth14 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight14 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData14 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปในอเรย์ ขนาด 2 มิติ

GetData14 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth14 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight14 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap14 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

InPutOpenClos_ing เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการทำการเปิด (Opening) และการปิด (Closing)

InPutOpenClos_ingToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

MorpImg14 เป็นเมตธอดที่ใช้ในการประมวลผลระหว่างภาพต้นฉบับและภาพที่เป็นหน้าต่าง (mask) และให้ได้ภาพที่กร่อน (Erotion) แล้ว

MorpImg15 เป็นเมตธอดที่ใช้ในการประมวลผลระหว่างภาพต้นฉบับที่กระทำการกรกร่อน (Erotion) แล้วและภาพที่เป็นหน้าต่าง (mask) นำมาผ่านการขยาย (Dilaion) เพื่อให้ได้ภาพแบบเปิด (Opening)

InitializeData15

เป็นเมตธอดที่ใช้ในการเตรียมพร้อมอเรย์เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการเปิด (Opening) และการปิด (Closing)

SetWidth15 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight15 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData15 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปใน อเรย์ ขนาด 2 มิติ

GetData15 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

GetWidth15 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight15 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap15 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

MaskOpenClos_ing เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปเป็นหน้าต่าง (mask) เพื่อประมวลผลร่วมกับภาพต้นแบบในทำการเปิด (Opening) และการปิด(Closing)

OutPutOpenClos_ingToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.11 TImageProcessing (Flip)

TImageProcessing (Flip)		
- Flip	:	TBitmap
- FlipToImage	:	TImage
+ InitializeData8	:	TBitmap
+ SetWidth8	:	unsigned int
+ SetHeight8	:	unsigned int
+ SetData8	:	unsigned int
+ GetData8	:	unsigned int
+ Getmask8	:	Void
+ CreateGreyPalette	:	Void
+ Setmask8	:	unsigned int
+ GetWidth8	:	Void
+ GetHeight8	:	Void
+ AssignToBitmap8	:	TImage

InitializeData8 เป็นเมตธอดที่ใช้ในการเตรียมพร้อม อเรย์ เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการประมวลตามคำสั่งในการทำการพลิกภาพ

SetWidth8 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight8 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData8 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปใน อเรย์ ขนาด 2 มิติ

GetData8 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth8 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight8 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap8 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

Flip เป็นเมตธอดที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการพลิกภาพ

FlipToImage เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

3.2.12 TImageProcessing (Rotate90)

TImageProcessing (Rotate90)		
- Rotate	:	TBitmap
- RotateToImage	:	TImage
+ InitializeData10	:	TBitmap
+ SetWidth10	:	unsigned int
+ SetHeight10	:	unsigned int
+ SetData10	:	unsigned int
+ GetData10	:	unsigned int
+ Getmask10	:	Void
+ CreateGreyPalette	:	Void
+ Setmask10	:	unsigned int
+ GetWidth10	:	Void
+ GetHeight10	:	Void
+ AssignToBitmap10	:	TImage

InitializeData10 เป็นเมตธอดที่ใช้ในการเตรียมพร้อม อเรย์ เพื่อใช้งาน และส่งข้อมูลเข้าไปเก็บภายในอเรย์แบบ 2 มิติ และทำการประมวลตามคำสั่งในการทำการกลับภาพ

SetWidth10 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความกว้างของรูปภาพที่รับเข้ามา

SetHeight10 เป็นเมตธอดที่ใช้การเตรียมพร้อมค่าความยาวของรูปภาพที่รับเข้ามา

SetData10 เป็นเมตธอดที่ใช้ในการเตรียมพร้อมค่าเพื่อเก็บลงไปใน อเรย์ ขนาด 2 มิติ

GetData10 เป็นเมตธอดที่ใช้ในการส่งค่าที่เก็บไว้ในอเรย์ 2 มิติ เพื่อนำไปประมวลผล

CreateGrayPalette เป็นเมตธอดที่ใช้ในการกำหนดค่าสีของรูปภาพที่เตรียมจะส่งออกไปแสดงผล

GetWidth10 เมตธอดที่ใช้การส่งค่าความกว้างของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

GetHeight10 เมตธอดที่ใช้การส่งค่าความยาวของรูปภาพที่รับเข้ามาออกไปเพื่อเป็นส่วนประกอบในการประมวลผลรวมกับอเรย์ 2 มิติ

AssignToBitmap10 เป็นเมตธอดที่ใช้ในการกำหนดค่าลงไปใน TBitmap เพื่อที่จะเตรียมส่งออกไปยัง TImage

Rotate เป็นเมตธอตที่ใช้ในการรับค่าจากรูปภาพที่ต้องการจะนำไปประมวลผลในการกลับภาพ

RotateToImage เป็นเมตธอตที่ใช้ในการกำหนดค่าลงไปใน TImage เพื่อที่จะเตรียมส่งออกไปประมวลผล

บทที่ 4

ผลการทดลอง

4.1 ชุดคำสั่งทำการอินเวิร์ทภาพ (Invert)

ชุดคำสั่ง TImageProcessing (Invert) ใช้ในการอินเวิร์ทภาพ (Invert) บิตแม็บ (Bitmap) ขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.1

รูปที่ 4.1 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับอินเวิร์ทภาพ

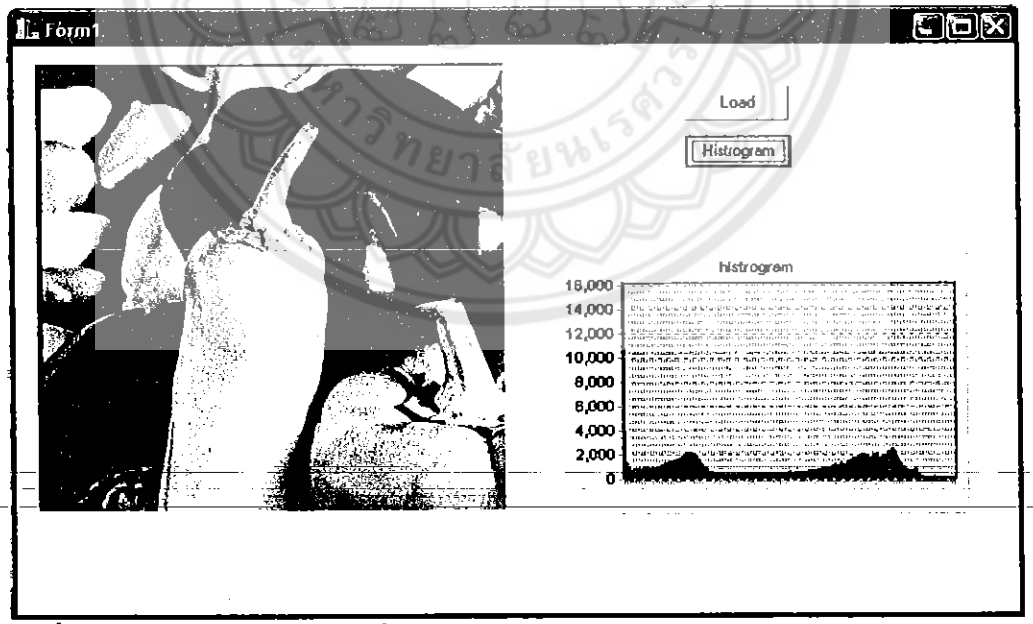
4.2 ชุดคำสั่งสร้างภาพไบนารี (Binary)

ชุดคำสั่ง TImageProcessing (Threshold) ใช้ในการสร้างเส้นขีดแบ่งของ ภาพบิตแม็บ ขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.2

รูปที่ 4.2 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับสร้างเส้นขีดแบ่งของภาพ

4.3 ชุดคำสั่งคำนวณหาฮิสโตแกรม (Histogram)

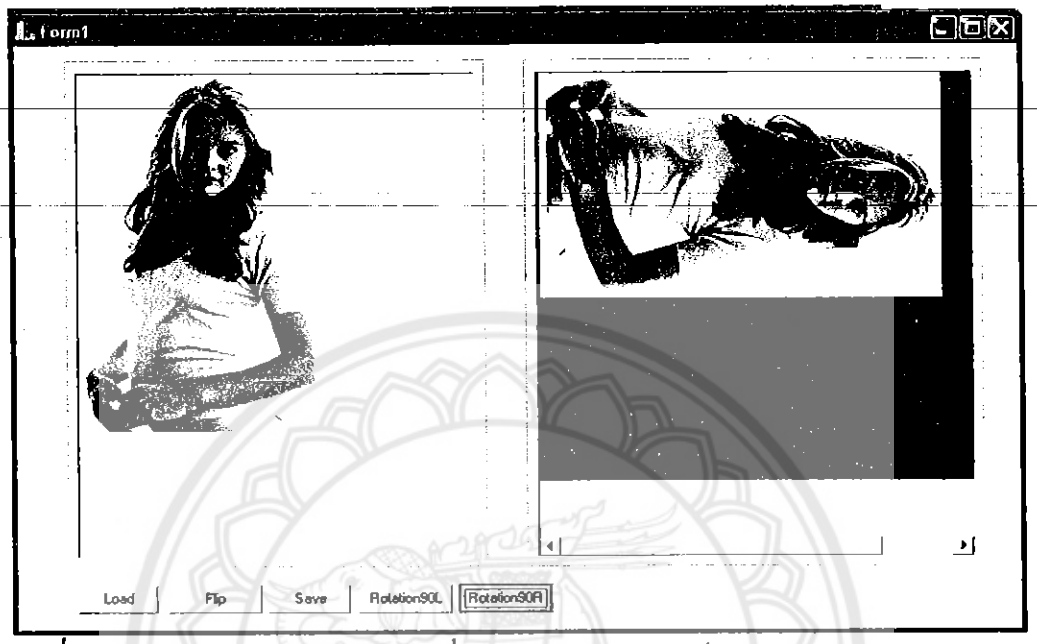
ชุดคำสั่ง TImageProcessing (Histogram) ใช้ในการคำนวณหาฮิสโตแกรม ของภาพบิตแม็บ ขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.3



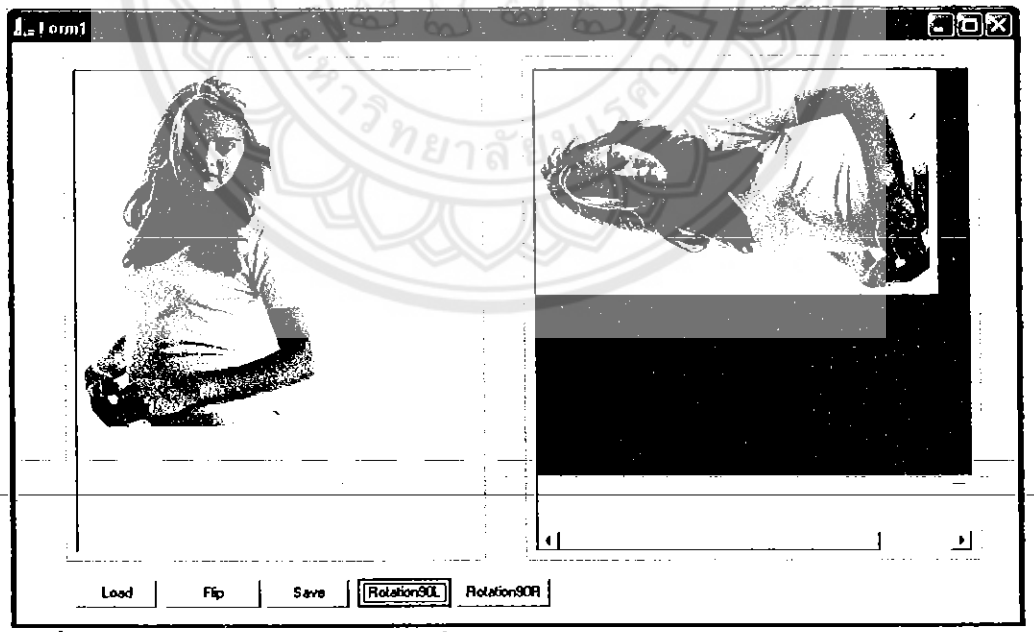
รูปที่ 4.3 ตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับคำนวณหาฮิสโตแกรมของภาพ

4.4 ชุดคำสั่งการประมวลผลภาพเชิงเรขาคณิต (Geometric Operations)

ชุดคำสั่ง TImageProcessing (Rotate90)ใช้ในการหมุนภาพบิตแม็บ แบบ 90 องศา แบบหมุนขวา และ หมุนซ้าย ได้ผลลัพธ์ดังรูปที่ 4.4



รูปที่ 4.4 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการหมุนภาพ 90 องศา (ขวา)



รูปที่ 4.5 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์ สำหรับการหมุนภาพ 90 องศา (ซ้าย)

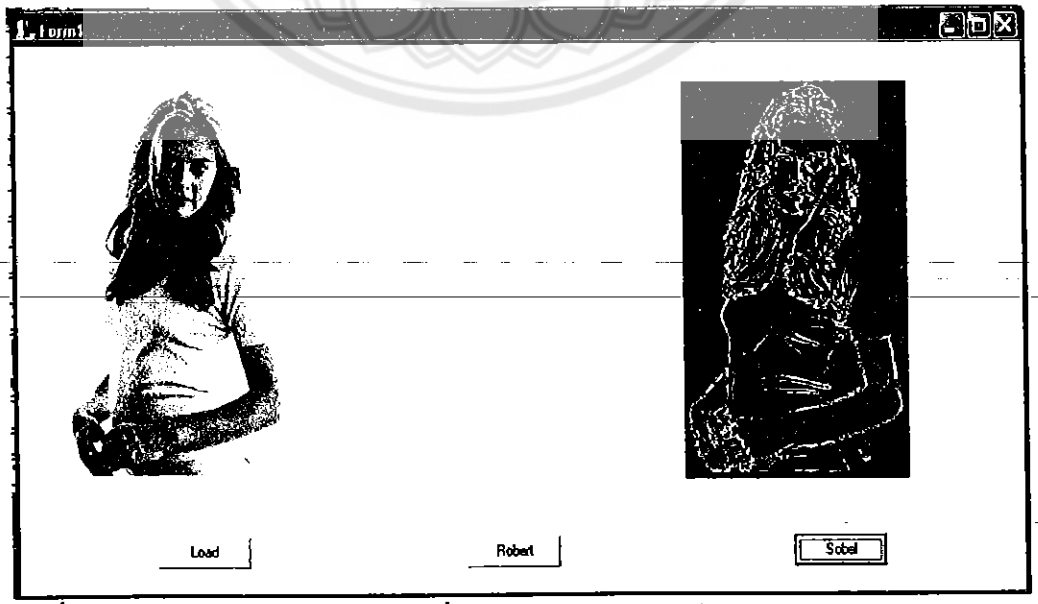
4.5 ชุดคำสั่ง TImageProcessing (Flip)

ใช้ในการกลับภาพบิตแม็บเป็นตรงข้าม ได้ผลลัพธ์ดังรูปที่ 4.6

รูปที่ 4.6 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการกลับภาพ

4.6 ชุดคำสั่งหาขอบภาพ (Edge Detection)

ชุดคำสั่ง TImageProcessing (Soble) ใช้ในการหาขอบภาพแบบโซเบล ซึ่งเป็นภาพบิตแม็บขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.7



รูปที่ 4.7 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับหาขอบภาพแบบโซเบล

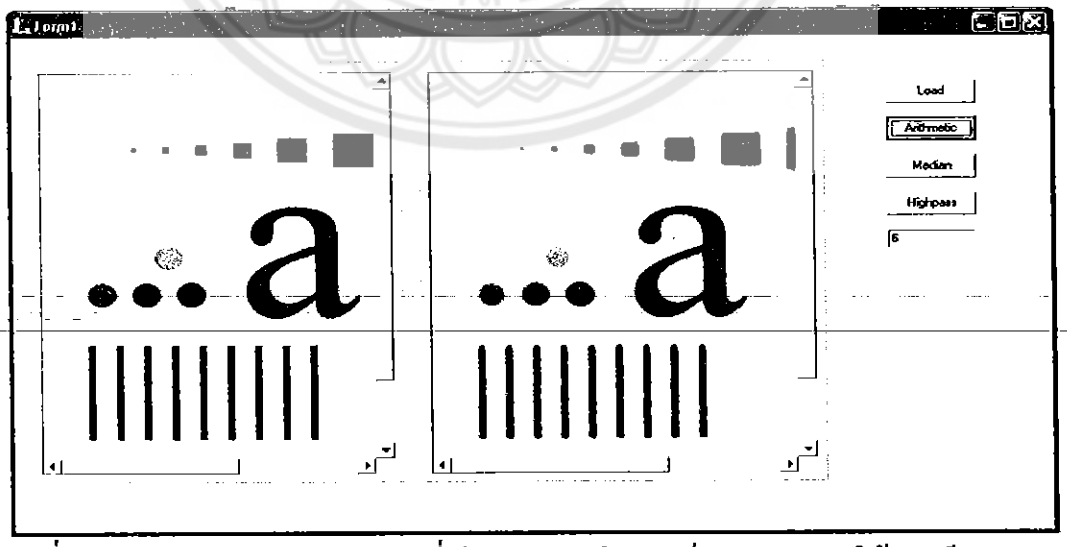
4.7 ชุดคำสั่ง TImageProcessing (Robert)

ใช้ในการหาขอบภาพแบบโรเบร์ต ซึ่งเป็นภาพบิตแม็บ ขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.8

รูปที่ 4.8 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับหาขอบภาพแบบ

4.8 ชุดคำสั่งทำให้ภาพเรียบ (Smoothing)

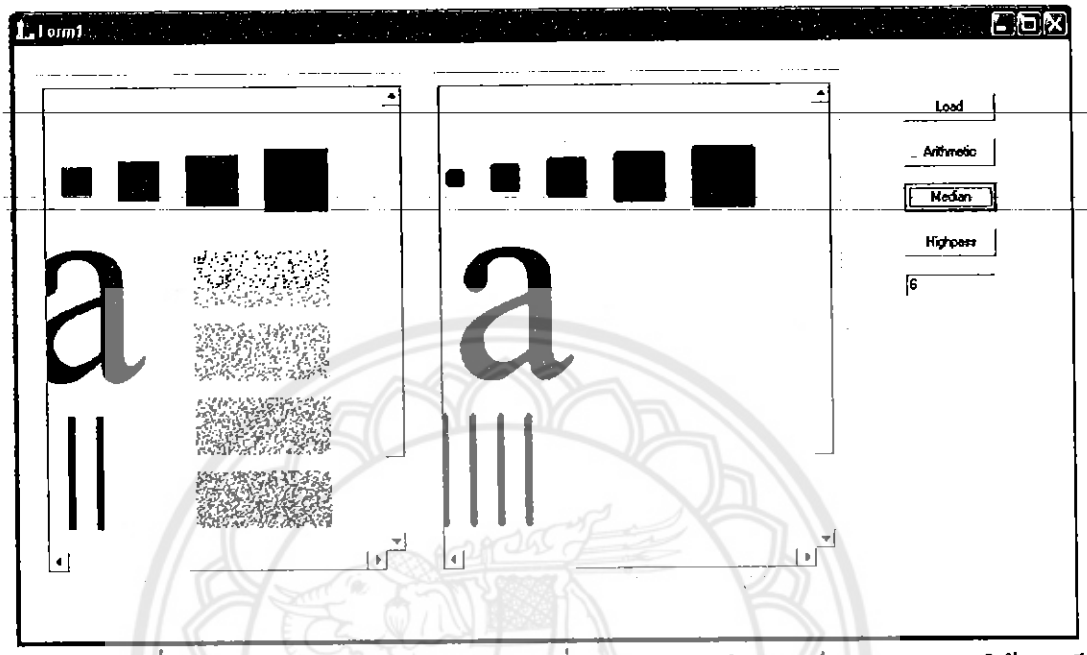
ชุดคำสั่ง TImageProcessing (Arithmetic)ใช้ในการทำให้ภาพเรียบแบบการกระทำเลขคณิต ซึ่งเป็นภาพบิตแม็บ ขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.9



รูปที่ 4.9 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการทำให้ภาพเรียบแบบการกระทำเลขคณิตหน้ากาก (Mask) ขนาด 6x6

4.9 ชุดคำสั่ง TImageProcessing (Median)

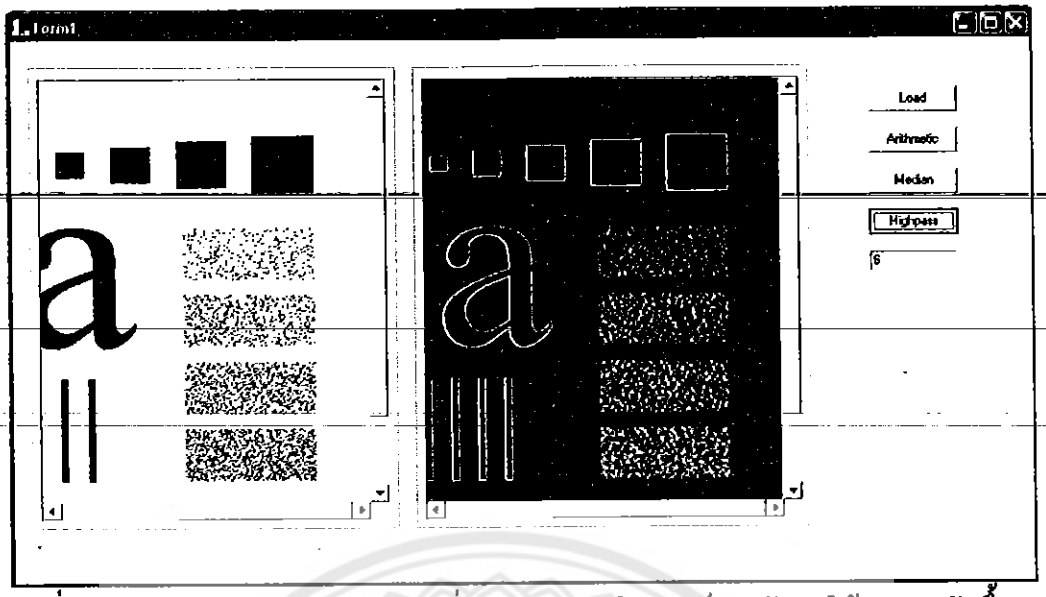
ใช้ในการทำให้ภาพเรียบแบบการกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน(Median Filtering) ซึ่งเป็นภาพบิตแม็บ ขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.10



รูปที่ 4.10 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการทำให้ภาพเรียบแบบการกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน(Median Filtering) หน้ากาก (Mask) ขนาด 6x6

4.10 ชุดคำสั่งทำให้ภาพคมชัดขึ้น (Sharpening)

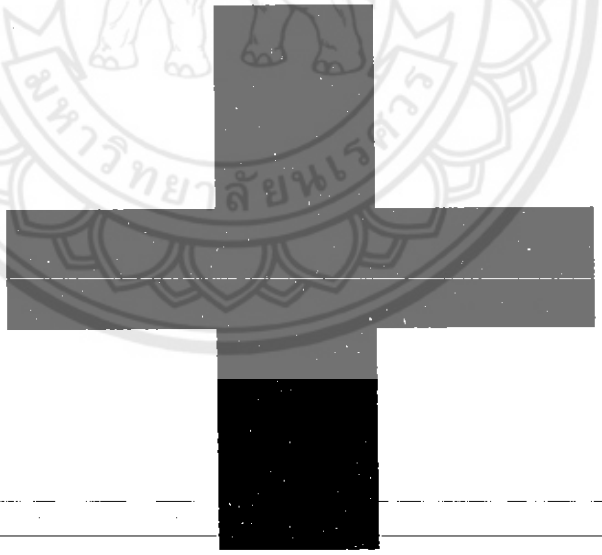
ชุดคำสั่ง TImageProcessing (Highpass)ใช้ในการทำให้ภาพคมชัดขึ้น (Sharpening) โดยใช้ตัวกรองความถี่สูงผ่าน (High-pass Filtering) ซึ่งเป็นภาพบิตแม็บ ขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.11



รูปที่ 4.11 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับทำให้ภาพคมชัดขึ้น (Sharpening) โดยใช้ ตัวกรองความถี่สูงผ่าน (High-pass Filtering) หน้ากาก (Mask) ขนาด 3x3

4.11 ชุดคำสั่งมอโฟโลยี (Morphological Operation)

ใช้ภาพ 2 ภาพในการทดลองและ 1 ภาพเป็นภาพทดลองและอีก 1 ภาพเป็นภาพหน้ากาก

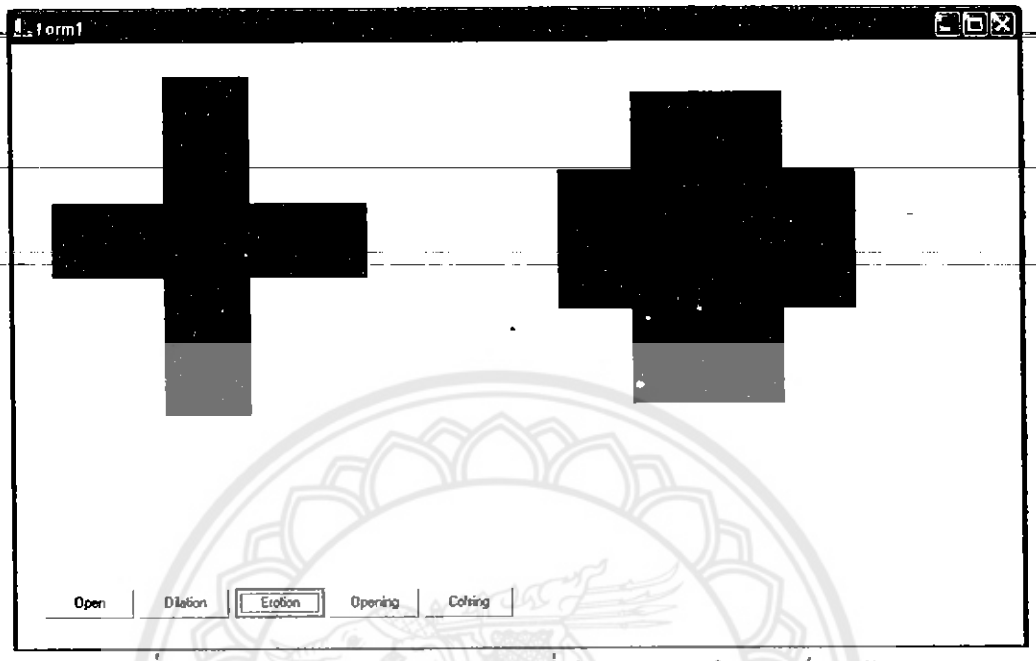


รูปที่ 4.12 แสดงภาพที่ใช้ในการทดลองเพื่อนำไปเป็นผลลัพธ์

รูปที่ 4.13 แสดงภาพที่ใช้ในการทดลองเพื่อนำไปหน้ากาก (Mask)

4.12 ชุดคำสั่ง TImageProcessing (Dilation)

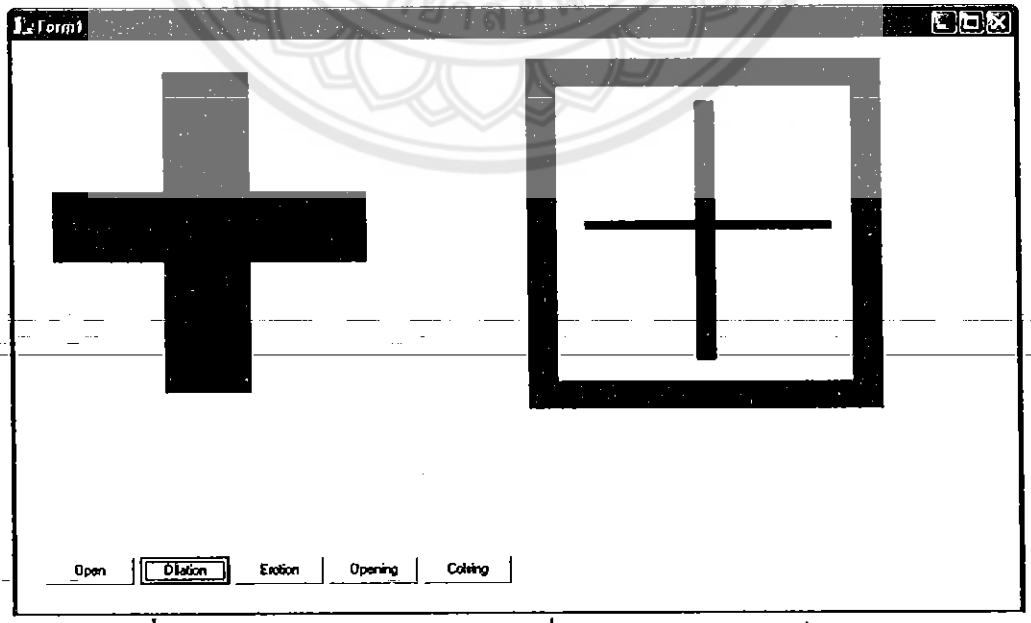
ใช้ในการขยายภาพซึ่งเป็นภาพบิตแม็บ ขนาด 8 ได้ผลลัพธ์ดังรูป 4.14



รูปที่ 4.14 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการขยายภาพ

4.13 ชุดคำสั่ง TImageProcessing (Erotion)

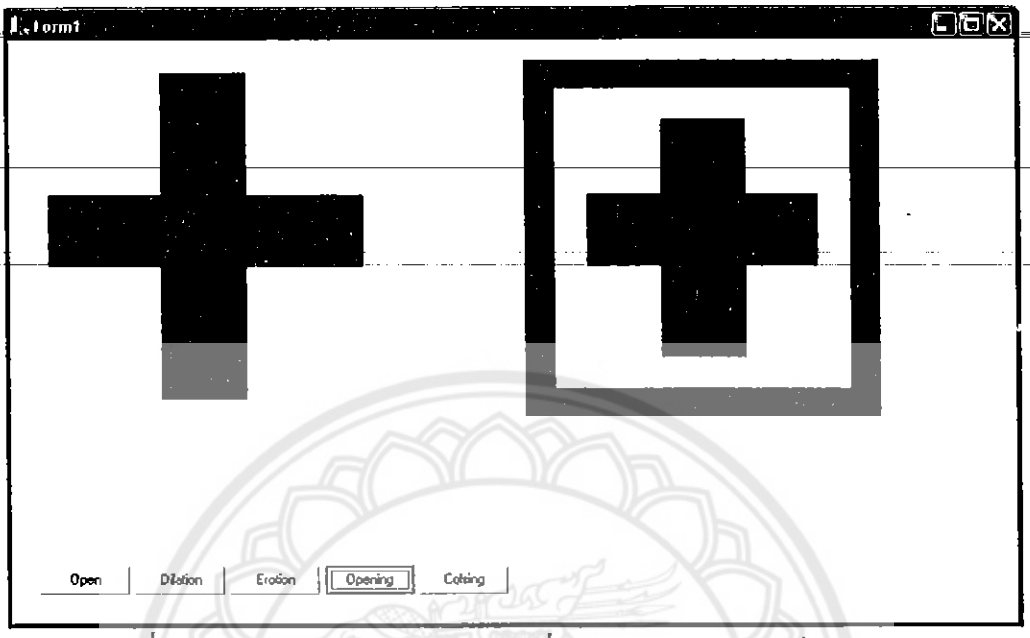
ใช้ในการกร่อนภาพ ซึ่งเป็นภาพบิตแม็บ ขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.15



รูปที่ 4.15 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการกร่อนภาพ

4.14 ชุดคำสั่ง TImageProcessing (Opening)

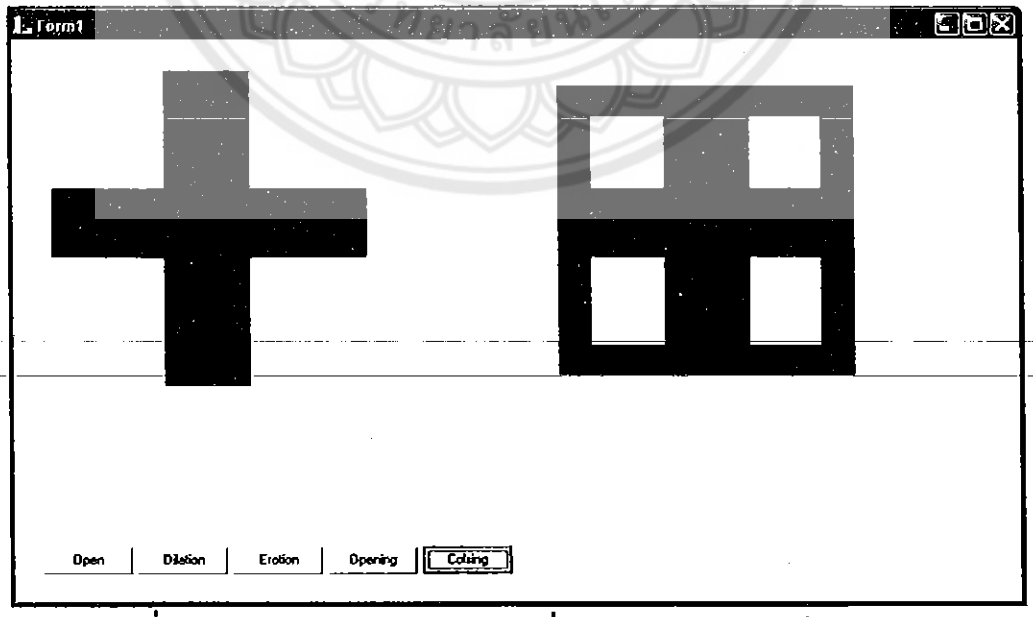
ใช้ในการเปิด ซึ่งเป็นภาพบิตแม็บขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.16



รูปที่ 4.16 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการเปิด

4.15 ชุดคำสั่ง TImageProcessing (Closing)

ใช้ในการปิด ซึ่งเป็นภาพบิตแม็บขนาด 8 บิต ได้ผลลัพธ์ดังรูปที่ 4.17



รูปที่ 4.17 แสดงตัวอย่างของโปรแกรมที่เขียนจากคอมโพเนนต์สำหรับการปิด

บทที่ 5
บทสรุป

5.1 สรุปผล

ภาพเป็นสิ่งสำคัญในการติดต่อสื่อสารและใช้ประโยชน์ทางด้านการแพทย์ หนังสือพิมพ์ ทหาร และทางด้านอุตสาหกรรม เป็นต้น จึงได้ว่าเทคนิคการประมวลผลภาพดิจิทัลได้นำมาประยุกต์ใช้อย่างกว้างขวาง การพัฒนาซอฟท์แวร์ที่สามารถนำส่วนของโปรแกรมกลับมาใช้ได้ใหม่นั้น ในยุคต้นใช้แนวคิดการจัดซอฟท์แวร์เป็นฟังก์ชั่น แต่เนื่องจากว่าวิธีนี้มีข้อเสีย คือ การแก้ไขและการเพิ่มเติมฟังก์ชั่นทำได้ไม่สะดวกเท่าที่ควร จึงนำเทคนิคการประมวลผล ภาพดิจิทัลมาพัฒนาใหม่ โดยได้นำเอาแนวทางของการเขียนโปรแกรมเชิงวัตถุเข้ามาพัฒนาร่วม เพื่อลดการเขียนโปรแกรมแบบซ้ำซ้อน เพิ่มประสิทธิภาพ และความสะดวกในการเขียนโปรแกรมทางด้านการประมวลผลภาพ ดิจิทัล โดยที่แนวทางของการเขียนโปรแกรมเชิงวัตถุสามารถที่จะเก็บชุดคำสั่งและนำมาเรียกใช้ได้ใหม่ที่สามารถประหยัดเวลาในการเขียนโปรแกรมได้

5.2 ปัญหาที่พบในการทำโครงงานวิจัย

5.2.1 ในขณะที่ต้องการเพิ่มส่วนประกอบ (Add Component) ลงไปใน C++ Builder ควรที่จะเพิ่ม (Add) เข้าไปทีเดียวพร้อมๆ กัน ไม่ควรเพิ่ม (Add) ตัวเดิมซ้ำไปซ้ำมา เพราะจะทำให้ส่วนประกอบผิดพลาด (Component error) ได้

5.2.2 การทำงานที่มีการขยายหน้ากาก (Mask) ออกกว้างมากๆ จะทำให้การประมวลผลช้าลง

5.2.3 ในขณะทำงานไม่มีส่วนประกอบ (Component) ที่เราสร้างขึ้นอยู่ในเครื่องจะไม่ทำงานตามต้องการได้ ควรทำการเพิ่มส่วนประกอบ (Add Component) เข้าไปให้เป็นที่เรียบร้อยก่อน

5.3 ข้อเสนอแนะ

ในโครงงานนี้สามารถที่จะทำการเขียนโปรแกรมเชิงวัตถุทางด้านการประมวลผลภาพดิจิทัลไปพัฒนาทางด้านคุณสมบัติ (Properties) ให้หลากหลายเพิ่มขึ้นได้

เอกสารอ้างอิง

Loannis Pitas. **DIGITAL IMAGE PROCESSING ALGORITHMS.** Cornwall : Great Britain, 1995.

Randy Crane. **A SIMPLIFIED APPROACH TO IMAGE PROCESSING.** New Jersey : Hewlett-Packard Company Published by Prentice Hall PTR Prentice-Hall, Inc, 1997.

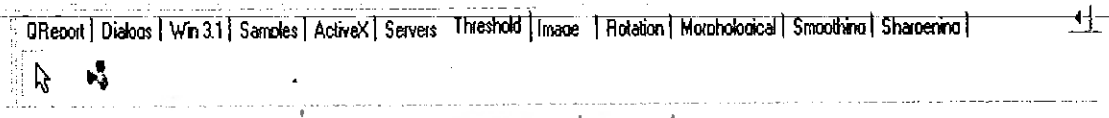
Ramesh Jain, Rangachar Kasturi and Brian G. Schunck. **MACHINE VISION.** Singapore : McGraw-Hill, 1995.

ดร.ชุลีรัตน์ จรัสกุลชัย. **การเขียนโปรแกรมเชิงวัตถุภาษาจาวา.** ครั้งที่ 1. กรุงเทพมหานคร : มหาวิทยาลัยเกษตรศาสตร์, 2543.

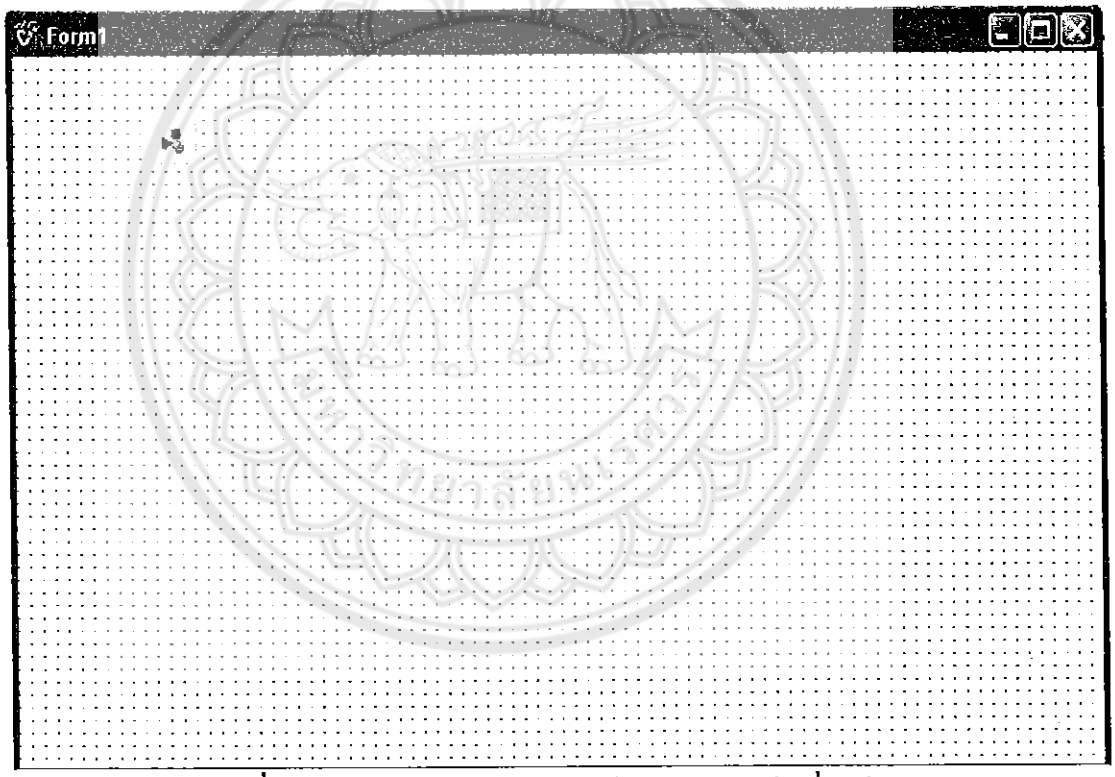
กนก กุสุมาลย์นุกูล และ ไกรวุฒิ มั่นเสถียรสิน. **คู่มือการเขียนโปรแกรม Delphi 4.** ครั้งที่ 1. กรุงเทพมหานคร : บริษัท ส.เอเชียเพลส (1989) จำกัด.

ภาคผนวก ก
คู่มือการใช้งานชุดคำสั่งเชิงวัตถุทางการประมวลผลภาพดิจิทัล

ในการใช้งานคอมโพเนนต์ทุกๆ ตัวถ้าหากต้องการนำคอมโพเนนต์ตัวใดมาใช้ให้เลือกที่คอมโพเนนต์แพเลตต์ แล้วนำไปวางไว้บนฟอร์มที่ต้องการจะใช้ ดังรูปที่ ก-1 และ ก-2



รูปที่ ก-1 แสดงคอมโพเนนต์แพเลตต์บน C++ Builder



รูปที่ ก-2 แสดงการนำคอมโพเนนต์วางไว้บนฟอร์มเพื่อใช้งาน

ก-1 ชุดคำสั่งทำการอินเวิร์ทภาพ (Invert)

ชุดคำสั่ง TImageProcessing (Invert) ทำการรับค่าจากรูปภาพต้นฉบับ ImageProcessing1->Invert (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); ทำการส่งค่าเพื่อไปแสดงผลที่ รูปภาพผลลัพธ์ ImageProcessing1->InvertToImage (*ชื่อรูปภาพผลลัพธ์);

ตัวอย่างเช่น

```
ImageProcessing1->Invert (*Image1->Picture->Bitmap) ;
ImageProcessing1->InvertToImage (*Image2) ;
```

ก-2 ชุดคำสั่งสร้างภาพไบนารี (Binary)

ชุดคำสั่ง TImageProcessing (Threshold) ใช้ในการกำหนดค่าขีดแบ่งของรูปภาพ ImageProcessing1->Setths (ค่าคงที่); ทำการรับค่าจาก รูปภาพต้นฉบับ ImageProcessing1->Threshold (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); ทำการส่งค่าเพื่อไปแสดงผลที่ รูปภาพผลลัพธ์ ImageProcessing1->ThresholdToImage (*ชื่อรูปภาพผลลัพธ์);

ตัวอย่างเช่น

```
int  a;
a = Edit1->Text.ToInt ();
 ImageProcessing1->Setths (a);
 ImageProcessing1->Threshold (*Image1->Picture->Bitmap);
 ImageProcessing1->ThresholdToImage (*Image2);
```

ก-3 ชุดคำสั่งคำนวณหาฮิสโตแกรม (Histrogram)

ชุดคำสั่ง TImageProcessing(Histrogram) ทำการรับค่าจากรูปภาพต้นฉบับImageProcessing1->Histrogram (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); ให้ทำการแสดงค่าที่เก็บไว้โดยให้เริ่มจาก 0 - 255 ImageProcessing1->His (ค่าตำแหน่งพิกเซล)

ตัวอย่างเช่น

```
int x;
 ImageProcessing1->Histrogram(*Image1->Picture->Bitmap);
 for(x=0; x<256; x++){
 Series1->Add( ImageProcessing1->His(x)," ", clRed);}
 Image1->Refresh();
```

ก-4 ชุดคำสั่งการประมวลผลภาพเชิงเรขาคณิต (Geometric Operations)

ชุดคำสั่ง TImageProcessing (Rotate90) ทำการรับค่าจากรูปภาพต้นฉบับ

ImageProcessing1->Rotate (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap, องศาในการหมุน (1-4), หมุนขวาหรือซ้าย (1-2)); ทำการส่งค่าเพื่อไปแสดงผลที่รูปภาพผลลัพธ์ ImageProcessing1->RotateToImage (*ชื่อรูปภาพผลลัพธ์);

ตัวอย่างเช่น

```
ImageProcessing1->Rotate(*Image1->Picture->Bitmap,l,2);
ImageProcessing1->RotateToImage(*Image2);
l++;
if(l>4){l=1;}}
```

ชุดคำสั่ง TImageProcessing (Flip) ทำการรับค่าจากรูปภาพต้นฉบับ ImageProcessing1->Flip (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap, f); ค่าคงที่ (1-4)); ทำการส่งค่าเพื่อไปแสดงผลที่รูปภาพผลลัพธ์ ImageProcessing1->FlipToImage (*ชื่อรูปภาพผลลัพธ์);

ตัวอย่างเช่น

```
ImageProcessing1->Flip(*Image1->Picture->Bitmap,f);
ImageProcessing1->FlipToImage(*Image2);
f++;
if(f>4){f=1;}}
```

ก-5 ชุดคำสั่งหาขอบภาพ (Edge Detection)

ชุดคำสั่ง TImageProcessing (Soble) ทำการรับค่าจากรูปภาพต้นฉบับ ImageProcessing1->Sobel (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); ทำการส่งค่าเพื่อไปแสดงผลที่รูปภาพผลลัพธ์ ImageProcessing1-> SobelToImage (*ชื่อรูปภาพผลลัพธ์);

ตัวอย่างเช่น

```
ImageProcessing1->Sobel(*Image1->Picture->Bitmap);
ImageProcessing1->SobelToImage(*Image2);
```

ชุดคำสั่งTImageProcessing (Robert) ทำการรับค่าจากรูปภาพต้นฉบับ ImageProcessing1-> Robert (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); ทำการส่งค่าเพื่อไปแสดงผลที่ รูปภาพผลลัพธ์ ImageProcessing1-> RobertToImage (*ชื่อรูปภาพผลลัพธ์);

ตัวอย่างเช่น

```
ImageProcessing1->Robert(*Image1->Picture->Bitmap);
ImageProcessing1->RobertToImage(*Image2);
```

ก-6 ชุดคำสั่งทำให้ภาพเรียบ (Smoothing)

ชุดคำสั่ง TImageProcessing (Arithmetic) กำหนดค่าของ mask ImageProcessing1->Setarth (ค่าคงที่); ทำการรับค่าจากรูปภาพต้นฉบับ ImageProcessing1-> Arithmetic (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); ทำการส่งค่าเพื่อไปแสดงผลที่รูปภาพผลลัพธ์ ImageProcessing1-> ArithmeticToImage (*ชื่อรูปภาพผลลัพธ์);

ตัวอย่างเช่น

```
int a;
a = Edit1->Text.ToInt();
ImageProcessing1->Setarth(a);
ImageProcessing1->Arithmetic(*Image1->Picture->Bitmap);
ImageProcessing1->ArithmeticToImage(*Image2);
```

ชุดคำสั่ง TImageProcessing (Median) กำหนดค่าของ mask ImageProcessing1->Setmask(b); ทำการรับค่าจากรูปภาพต้นฉบับ ImageProcessing1->Median (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); ทำการส่งค่าเพื่อไปแสดงผลที่รูปภาพผลลัพธ์ ImageProcessing1->MedianToImage (*ชื่อรูปภาพผลลัพธ์);

ตัวอย่างเช่น

```
int a;
a = Edit1->Text.ToInt();
ImageProcessing1->Setmask(a);
ImageProcessing1->Median(*Image1->Picture->Bitmap);
ImageProcessing1->MedianToImage(*Image2);
```

ก-7 ชุดคำสั่งทำให้ภาพคมชัดขึ้น (Sharpening)

ชุดคำสั่ง TImageProcessing (Highpass) ทำการรับค่าจาก รูปภาพต้นฉบับ ImageProcessing1->Highpass (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); ทำการส่งค่าเพื่อไป แสดงผลที่รูปภาพผลลัพธ์ ImageProcessing1->HighpassToImage (*ชื่อรูปภาพผลลัพธ์);

ตัวอย่างเช่น

```
ImageProcessing1->Highpass(*Image1->Picture->Bitmap);
ImageProcessing1->HighpassToImage(*Image2);
```

ก-8 ชุดคำสั่งมอโฟโลยี (Morphological Operation)

ใช้ภาพ 2 ภาพในการทดลองและ 1 ภาพเป็นภาพทดลองและอีก 1 ภาพเป็นภาพ Mask

ชุดคำสั่ง TImageProcessing (Dilation) ทำการรับค่าจากรูปภาพต้นฉบับImageProcessing1->InPutEroDila_tion (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); แสดงภาพต้นฉบับออกมา ImageProcessing1->InPutEroDila_tionToImage (*ชื่อรูปภาพต้นฉบับ); รับค่าจากรูปภาพที่เป็น Mask ImageProcessing1->MaskEroDila_tion (*ชื่อรูปภาพที่เป็นMask ->Picture->Bitmap, 2); แสดงรูปภาพผลลัพธ์ออกมา ImageProcessing1->OutPutEroDila_tionToImage (*ชื่อภาพผลลัพธ์);

ตัวอย่างเช่น

```
if(OpenPictureDialog1->Execute()){
Image2->Picture->LoadFromFile(OpenPictureDialog1->FileName);
ImageProcessing1->InPutEroDila_tion(*Image1->Picture->Bitmap);
ImageProcessing1->InPutEroDila_tionToImage(*Image1);
ImageProcessing1->MaskEroDila_tion(*Image2->Picture->Bitmap,2);
ImageProcessing1->OutPutEroDila_tionToImage(*Image2);}
```

ชุดคำสั่ง TImageProcessing (Erotion) ทำการรับค่าจากรูปภาพต้นฉบับ ImageProcessing1->InPutEroDila_tion (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); แสดงภาพต้นฉบับออกมา ImageProcessing1->InPutEroDila_tionToImage (*ชื่อรูปภาพต้นฉบับ); รับค่าจากรูปภาพที่เป็น Mask ImageProcessing1->MaskEroDila_tion (*ชื่อรูปภาพที่เป็นMask ->Picture->Bitmap,1); แสดงรูปภาพผลลัพธ์ออกมา ImageProcessing1->OutPutEroDila_tionToImage (*ชื่อภาพผลลัพธ์);

ตัวอย่างเช่น

```
if(OpenPictureDialog1->Execute()){
Image2->Picture->LoadFromFile(OpenPictureDialog1->FileName);
ImageProcessing1->InPutEroDila_tion(*Image1->Picture->Bitmap);
ImageProcessing1->InPutEroDila_tionToImage(*Image1);
ImageProcessing1->MaskEroDila_tion(*Image2->Picture->Bitmap,1);
ImageProcessing1->OutPutEroDila_tionToImage(*Image2);}
```

ชุดคำสั่ง TImageProcessing (Opening) ทำการรับค่าจาก รูปภาพต้นฉบับ ImageProcessing1->InPutOpenClos_ing (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); แสดงภาพต้นฉบับออกมา ImageProcessing1-> InPutOpenClos_ingToImage (*ชื่อรูปภาพต้นฉบับ); รับค่าจากรูปภาพที่เป็น Mask ImageProcessing1->MaskOpenClos_ing (*ชื่อรูปภาพที่เป็น Mask->Picture ->Bitmap, 1); แสดงรูปภาพผลลัพธ์ออกมา ImageProcessing1-> OutPutOpenClos_ingToImage (*ชื่อภาพผลลัพธ์);

ตัวอย่างเช่น

```
if(OpenPictureDialog1->Execute()){
Image2->Picture->LoadFromFile(OpenPictureDialog1->FileName);
ImageProcessing1->InPutOpenClos_ing(*Image1->Picture->Bitmap);
ImageProcessing1->InPutOpenClos_ingToImage(*Image1);
ImageProcessing1->MaskOpenClos_ing(*Image2->Picture->Bitmap,1);
ImageProcessing1->OutPutOpenClos_ingToImage(*Image2);}
```

ชุดคำสั่ง TImageProcessing (Closing) ทำการรับค่าจากรูปภาพต้นฉบับ ImageProcessing1->InPutOpenClos_ing (*ชื่อรูปภาพต้นฉบับ->Picture->Bitmap); แสดงภาพต้นฉบับออกมา ImageProcessing1-> InPutOpenClos_ingToImage (*ชื่อรูปภาพต้นฉบับ); รับค่าจากรูปภาพที่เป็น Mask ImageProcessing1->MaskOpenClos_ing (*ชื่อรูปภาพที่เป็น Mask->Picture ->Bitmap, 2); แสดงรูปภาพผลลัพธ์ออกมา ImageProcessing1-> OutPutOpenClos_ingToImage (*ชื่อภาพผลลัพธ์);

ตัวอย่างเช่น

```
if(OpenPictureDialog1->Execute()){
Image2->Picture->LoadFromFile(OpenPictureDialog1->FileName);
ImageProcessing1->InPutOpenClos_ing(*Image1->Picture->Bitmap);
ImageProcessing1->InPutOpenClos_ingToImage(*Image1);
ImageProcessing1->MaskOpenClos_ing(*Image2->Picture->Bitmap,2);
ImageProcessing1->OutPutOpenClos_ingToImage(*Image2);}
```

ประวัติผู้เขียนโครงงาน

ชื่อ นายฐิติ เสาวลักษณ์
ภูมิลำเนา 77/140 ถ.ศรีธรรมไตรปิฎก อ.เมือง จ. พิษณุโลก
ประวัติการศึกษา
- จบประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)
วิทยาลัยเทคนิคพิษณุโลก แห่งที่ 2 จ. พิษณุโลก
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยนเรศวร
E-mail : thx_s@hotmail.com

ชื่อ นางสาวบุญศิริ จั่นแก้ว
ภูมิลำเนา 18 หมู่ 1 ต. บ้านโสก อ.หล่มสัก จ. เพชรบูรณ์
ประวัติการศึกษา
- จบระดับมัธยมศึกษาจากโรงเรียนหล่มสักวิทยาคม
- ปัจจุบันศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยนเรศวร
E-mail : kwant501@hotmail.com