สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมดูภาพและปรับปรุงคุณภาพของภาพ

IMAGE VIEWING AND ENHANCEMENT

สิทธิพัฒน์ มโนจุรีหกุล

เลขหมู่.....
เลขทะเบียน 51791
วัน,เดือน,ปี 29 ก.ค. 2547

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.b.....
.i.....

IMAGE VIEWING AND ENHANCEMENT

SITTIPAT MANOJUREEHAKUL

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FALCULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ โปรแกรมดูภาพและปรับปรุงคุณภาพของภาพ
IMAGE VIEWING AND ENHANCEMENT

ชื่อนักศึกษา นายสิทธิพัฒน์ มโนจุรีหกุล 43050427

ภาควิชา ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา รศ.ธีรวัฒน์ ประกอบผล

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2546

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	ผศ.ดร.ศรัณย์ อินทโกสุม	
กรรมการ	อ.วีรชัย ตันยะสิทธิ์	
กรรมการและอาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล	

(ผู้ช่วยศาสตราจารย์ ดร.วีระ บุญจริง)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	โปรแกรมดูภาพและปรับปรุงคุณภาพของภาพ
ชื่อนักศึกษา	นายสิทธิพัฒน์ มโนจุรีหกุล 43050427
ปริญญา	วิทยาศาสตรบัณฑิต
ภาควิชา	ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2546
อาจารย์ที่ปรึกษา	รศ.ธีรวัฒน์ ประกอบผล

บทคัดย่อ

ปัจจุบันเป็นที่ยอมรับกันอย่างกว้างขวางทั้งในภาครัฐและเอกชนว่า ระบบคอมพิวเตอร์และการสื่อสารข้อมูล นั้นมีความสำคัญในการเพิ่มศักยภาพและประสิทธิภาพต่อการพัฒนาระบบภายในองค์กร โปรแกรมดูภาพและปรบปรุงคุณภาพของภาพ เป็นโปรแกรมประยุกต์ที่ถูกนำมาใช้กับระบบคอมพิวเตอร์พีซี โดยอาศัยหลักการของการประมวลผลภาพ(Image Processing) โดยจะช่วยอำนวยความสะดวกให้แก่ผู้ใช้ ซึ่งในปัจจุบันเทคโนโลยีการถ่ายภาพด้วยกล้องดิจิตอล และโทรศัพท์มือถือ กำลังเป็นที่นิยมเพราะสามารถทำได้อย่างสะดวก รวดเร็ว แต่เนื่องจากความละเอียดของกล้องทำให้มีผู้ใช้ที่ประสบปัญหาต่อการถ่ายภาพ อาจจะเกิดความผิดเพี้ยนของภาพ ความมืด เบลอ ซึ่งสิ่งเหล่านี้อาจเกิดจากความไม่ตั้งใจของผู้ถ่าย หรือ สภาพแวดล้อมขณะถ่ายภาพ ทางแก้อาจจะทำโดยการถ่ายภาพนั้นใหม่ แต่ภาพบางเหตุการณ์ซึ่งไม่อาจจะทำการถ่ายภาพนั้นซ้ำอีกครั้งได้ ทำให้เราต้องสูญเสียภาพนั้นๆ ที่เราต้องการได้

ดังนั้น แนวคิดการพัฒนาโปรแกรมดูภาพและปรับปรุงคุณภาพของภาพจึงถูกพัฒนาขึ้นเพื่อช่วยแก้ไขปัญหาและช่วยเพิ่มความสะดวกให้ภาพนั้นมีความสมบูรณ์มากขึ้น ซึ่งพัฒนาด้วย Visual C++.NET ซึ่งเป็นเครื่องมือที่ช่วยในการพัฒนาโปรแกรมบนวินโดวส์ที่เป็นที่นิยมกันอย่างแพร่หลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	IMAGE VIEWING AND ENHANCEMENT
Student	Mr. Sittipat Manojureehakul 43050427
Degree	Bachelor of Science
Department	Mathematics and Computer Science, Faculty of Science
Programme	Computer Science
Academic Year	2003
Special Project Adviser	Assoc.Prof.Teerawat Prakorbphon

ABSTRACT

Today, it is widely accepted both of Government and Private company that Computer and Communication system is one of the most important and high potential resource in organization for developing system.

Most of people may have problem when they took a photo by digital camera or mobile phone with attached camera which may be losing sharpness, having blur, and lack of brightness. These entire problems come from external environment or mistake from people. According to problems, it may annoy you to take same photo again. It creates a lot of expenditure and some picture can not be taken as in past.

Therefore, this application that is called Image Viewing and Enhancement was created to adjust the source photo for making it better by base on Image processing idea. It was developed by using Visual C++.NET that is popular tool for develop program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่อง โปรแกรมดูภาพและปรับปรุงคุณภาพของภาพ สามารถสำเร็จลุล่วงไปด้วยดี กระผมผู้จัดทำ ขอขอบพระคุณ ท่านอาจารย์ ธีรวัฒน์ ประกอบผล ซึ่งเป็นอาจารย์ผู้รับผิดชอบปัญหา และ หัวข้อพิเศษเรื่องนี้ ที่กรุณาสละเวลาให้คำแนะนำ และ เป็นที่ปรึกษาในการแนะนำ และ แนวทางการแก้ไขปัญหาต่างๆ รวมทั้งกำหนดทิศทางและความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้กระผมขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทั้งกำลังใจ แรงใจ ทุนทรัพย์ และ อาจารย์ เจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ที่ได้ช่วยเหลือด้านต่างๆ และอาจารย์ทุกๆ ท่าน ที่ได้ประสาทวิชาความรู้ ประสบการณ์ ทั้งในด้านทฤษฎี และ ด้านปฏิบัติแก่ผู้จัดทำ จนการจัดทำปัญหาพิเศษในครั้งนี้สำเร็จลุล่วงไปด้วยดีทุกประการ รวมทั้งเพื่อนๆ และ พี่ๆ ทุกๆท่านที่ได้ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

ผู้จัดทำ

มีนาคม 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	IX
สารบัญตาราง	XVI
บทที่ 1 บทนำ	
1.1 ความสำคัญและที่มาของปัญหาพิเศษ	1
1.2 วัตถุประสงค์ของการพัฒนาระบบ	1
1.3 ขอบเขตของปัญหาโครงงานปัญหาพิเศษ	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 การวางแผนงานการทำโครงการ	3
บทที่ 2 กฎเกณฑ์ ทฤษฎี หลักการ	
2.1 บทนำ	4
2.2 กฎเกณฑ์ต่างๆ	4
2.3 การนำเสนอข้อมูลภาพดิจิตอล (Digital Image Representation)	4
2.4 หลักการเบื้องต้นของการประมวลผลภาพ	5
2.4.1 พิกเซล (Pixel)	5
2.4.2 เอสเป็กเรโชของภาพ (Image Aspect Ratio)	8
2.4.3 ตำแหน่งของพิกเซล (Pixel Position)	8
2.5 รีโซรูชัน(Resolution)	9
2.6 การประมวลผลสีของภาพ(Color Image Processing)	12
2.6.1 Color model	12
2.6.2 The RGB color model	12
2.6.3 Pseudo-color Image Processing	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.7 ข้อมูลรูปภาพแบบไบนารี (Binary Image Data)	16
2.8 บิตแมบ (Bitmap) และการใช้งาน	19
2.8.1 บิตแมบ	19
2.8.2 ตัวภาพ	23
2.9 การทำกระบวนการอิมเมจดิจิตอล (Digital Image Processing)	23
2.9.1 อิมเมจดิจิตอล	24
2.9.2 การทำกระบวนการอิมเมจดิจิตอล	24
2.10 วิธีการอ่านข้อมูลพิกเซลของอิมเมจ	25
2.11 โมเดลสี (Color Model)	26
2.12 การอ่านข้อมูลอิมเมจในหน่วยความจำ	27
2.12.1 การหาพอยน์เตอร์ที่ชี้ไปยังจุดเริ่มต้นหรือเกือบท้ายสุด	27
2.13 การทำกระบวนการจุดพิกเซล (Point Processing)	29
2.13.1 การแปลงที่เกี่ยวกับความสว่าง (Luminance Transformation)	30
2.13.2 การทำ Segment	30
2.13.3 การทำ Symmetry	30
2.13.4 การกลับสี (Invert Color)	31
2.13.5 การแก้ไขพิกเซลด้วย Gamma	32
2.13.6 การสร้างฮิสโทแกรม (Histogram)	33
2.13.7 การทำ Histogram Equalized	34
2.13.8 การบวกลบอิมเมจและการใช้ลอจิก And, Or และ Xor	35
2.14 การทำกระบวนการรอบๆ จุดพิกเซล (Area Processing)	36
2.14.1 ฟิลเตอร์ Low Pass	37
2.14.2 ฟิลเตอร์ High Pass	37
2.14.3 การทำอิมเมจให้นูน (Emboss)	38
2.14.4 การตรวจหาขอบอิมเมจ (Edge Detector)	38
2.14.4.1 อนุพันธ์อันดับหนึ่ง	38
2.14.4.2 โอเปอเรเตอร์ Sobel	39
2.14.4.3 โอเปอเรเตอร์ Prewitt	39
2.14.4.4 โอเปอเรเตอร์ Fri-Chen	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.14.4.5 อนุพันธ์อันดับสอง 40

2.15 การทำกระบวนรูปร่างลักษณะ (Morphology Processing) 40

2.15.1 การขยายพิกเซล (Dilation) 40

2.15.2 การลดขนาดพิกเซล (Erosion) 41

2.15.3 การทำอิมเมจในพื้นที่ว่างให้เปิดมากขึ้น (Opening) 41

2.15.4 การทำอิมเมจในพื้นที่ว่างให้เปิดมากขึ้น (Closing) 41

2.16 การตัดสิ่งที่ไม่ต้องการออกจากภาพ(Removing Noise) 41

2.16.1 วิธี Smoothing Spatial Filter 41

2.16.2 วิธี Order-Statistics Filters 44

2.16.3 วิธี Sharpening Spatial Filters 45

2.17 การหาขอบภาพ (Edge Detection) 46

2.17.1 ขอบภาพ 46

2.17.2 ตัวดำเนินการอนุพันธ์ (Derivative Operators) 48

2.18 แบบจำลองของขอบภาพ 50

2.19 อัลกอริทึมของ Marr และ Hildreth 53

2.20 อัลกอริทึมของ Canny 56

2.21 อัลกอริทึมของ Shen และ Castan 58

2.22 สรุปการทำงานของอัลกอริทึมต่างๆ 59

บทที่ 3 ลักษณะทั่วไปของโปรแกรม

3.1คุณสมบัติต่าง ๆ ของ โปรแกรม 60

3.1.1 กระบวนการอิมเมจเกี่ยวกับจุด Point Processing 60

3.1.2 กระบวนการอิมเมจที่เกี่ยวกับบริเวณรอบ ๆ จุดพิกเซล 60

3.1.3 กระบวนการทำอิมเมจที่เกี่ยวกับรูปร่างลักษณะ 60

3.2 ลักษณะทั่วไปของโปรแกรม 61

3.2.1 ลักษณะงานของโปรแกรม 61

3.3 การออกแบบตัวโปรแกรม 66

3.4 รายละเอียดของระบบงาน 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.4.1 การรับภาพ	68
3.4.2 การสร้างคลาสอิมเมจใหม่	68
3.4.3 การเลือกรูปแบบการทำงานต่างๆ	69
3.4.3.1 การอ่านข้อมูลพิกเซลในหน่วยความจำ	69
3.4.3.2 การทำ Grayscale	70
3.4.3.3 การทำการกลับสี (Invert Color)	71
3.4.3.4 การแก้ไขพิกเซลด้วยแกมมา(Gamma Correction)	71
3.4.3.5 การทำ Histogram Equalized	72
3.4.3.6 การกลับภาพในแนวตั้ง (Flip)	73
3.4.3.7 การกลับภาพในแนวนอน (Mirror)	75
3.4.3.8 การทำการบวกลบอิมเมจและการใช้ลอจิคต่างๆ	76
3.4.3.8.2 การทำลอจิค Or	76
3.4.3.8.1 การทำลอจิค Xor	77
3.4.3.8 การทำ Composite Image	77
3.4.3.9 การทำการแมบเข้ากับหน้ากาก (Masking Convolution)	78
3.4.3.10 การขยายพิกเซล (Dilation)	80
3.4.3.11 การลดพิกเซล (Erosion)	82
3.4.4 ตัวอย่างการประยุกต์ใช้ Erosion และ Dilation	84
3.4.5 รูปแบบของการหมุนที่ใช้ในRotate(Angle)	84
3.4.5.1 การหมุนตามเข็มนาฬิกา	85
3.4.5.2 การหมุนทวนเข็มนาฬิกา	86
3.4.6 การพิมพ์และดูเอกสารก่อนการพิมพ์	87
3.4.7 การสร้างสเตตัส บาร์ และ การซูมอิมเมจ	87
3.4.8 การทำUndo	88
3.4.9 การสร้าง Histogram และการคำนวณสถิติของอิมเมจ	88
3.4.10 วิธีการหาค่าเฉลี่ย (Mean) ของอิมเมจสีต่างๆ	89
3.4.11 วิธีการหาค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของอิมเมจ	90
3.5 ลักษณะการใช้งานของโปรแกรม	91

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

บทที่4 ประเมินผลระบบ

การประเมินผล 123

บทที่5 สรุปการดำเนินการและข้อเสนอแนะ

สรุปผล 125

ข้อเสนอแนะและแนวทางการพัฒนาต่อ 125

ภาคผนวก ก ตัวอย่างชุดภาพ 127

บรรณานุกรม 155

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

ภาพที่	หน้า
2.1 ข้อมูลภาพดิจิตอลแสดงถึงแนวแกน x และ y และ ฟังก์ชันแสดงความเข้มของแสง	5
2.2 ความสัมพันธ์ของภาพโดยทั่วไปกับพิกเซลเมตริกซ์	6
2.3 ดัชนีพิกเซลในเมตริกซ์	7
2.4 การแปลงภาพให้เป็นพิกเซลเมตริกซ์	8
2.5 แสดงการทำ Resolution ของภาพในแต่ละแบบ	10
2.5 แสดงการทำ Resolution ของภาพในแต่ละแบบ	11
2.6 สี่เหลี่ยมลูกบาศก์ RGB	13
2.7 ทางเลือกการนำเข้าของ Gray Level	15
2.8 การแปลง Gray level ไปเป็น RGB	15
2.9 การแปลงรูปไบนารี่ไปเป็นไบนารี่โค้ด	17
2.10 การแปลงโค้ดฐานสองไปเป็นฐานสิบหก	17
2.11 การเก็บโค้ดในรูปอาร์เรย์	17
2.12 การแปลงโค้ดฐานสองไปเป็นฐานสิบหกของรูปแบบWin32	18
2.13 การเก็บโค้ดในรูปอาร์เรย์ของรูปแบบWin32	18
2.14 โครงสร้างของบิตแมบไฟล์	21
2.15 ภาพของการ ScanLine	23
2.16 ระบบพิกัด Space	24
2.17 การทำกระบวนการอิมเมจดิจิตอลหรืออิมเมจโปรเซสซิงก์	25
2.18 โมเดลสีและการผสมสีทางแสง	26
2.19 การอ่านข้อมูลพิกเซล เมื่อฟังก์ชัน GetPitch คืนค่าเป็นบวก	28
2.20 การอ่านข้อมูลพิกเซล เมื่อฟังก์ชัน GetPitch คืนค่าเป็นลบ	28
2.21 การอ่านข้อมูลพิกเซลโดยใช้พอยน์เตอร์ pSrc	29
2.22 การแปลงแบบ Negative	31
2.23 Gamma Correction Transformation	32
2.24 ภาพที่แสดงโดยกราฟHistogram	33
2.25 การทำ Histogram Equalized	34
2.26 วิธีการคำนวณ Convolution	36
2.27 หลักการของ Spatial Filter	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

2.28 การแทนหน้ากาก Spatial Filter มิติ 3*3 43

2.29 หน้ากากขนาด 3*3 43

2.30 ข้อมูลภาพ 44

2.31 ข้อมูลที่ผ่านการกรอง 45

2.32 การเปลี่ยนแปลงค่าระดับเทา 46

2.33 (ก) ขอบภาพแบบขั้นบันได (ข) ขอบภาพแบบลาด 47

2.34 ตัวอย่างการหาขอบภาพ 47

2.35 (ก) ภาพตัดขวางของต้นไม้ (ข) ขอบภาพจากรูป ก. 48

2.36 การหาขอบภาพโดยใช้แบบจำลองของ Sobel และ Kirsch 51

2.37 ตำแหน่งของขอบภาพและการหาขอบภาพจากอนุพันธ์ที่1 และ 2 52

2.38 ลักษณะของลาปาเซี่ยน ออฟ เกาส์เซี่ยน 54

2.39 ผังการทำงานของการหาขอบภาพแบบ Marr 55

2.40 การหาขอบภาพแบบ Marr 55

2.41 แสดงผังการทำงานของ Canny 57

2.42 ผังการทำงานของการหาขอบภาพแบบ Shen และ Castan 58

3.1 แสดงFlow Chart ของลักษณะภาพรวมของโปรแกรม 61

3.2 แสดงFlow Chart ของลักษณะทั่วไปของโปรแกรม 62

3.3 แสดงFlow Chart ของกระบวนการ Point Processing 63

3.4 แสดงFlow Chart ของกระบวนการ Area Processing 64

3.5 แสดงFlow Chart ของกระบวนการ Morphology Processing 65

3.6 โมเดลคลาส CPImage 69

3.7 อัลกอริทึมการอ่านข้อมูลพิกเซล 70

3.8 อัลกอริทึมการทำภาพGrayscale 70

3.9 อัลกอริทึมการทำภาพInvert 71

3.10 อัลกอริทึมการทำภาพGamma Correction Tranformation 72

3.11 อัลกอริทึมการทำHistogram Equalized 73

3.12 การทำการ Flip ในบิตแมบ 74

3.13 อัลกอริทึมการFlip 74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

3.14 การทำการ Mirrorในบิตแมบ 75

3.15 อัลกอริทึมการMirror 75

3.16 อัลกอริทึมการAnd 76

3.17 อัลกอริทึมการOr 77

3.18 อัลกอริทึมการAnd 77

3.19 อัลกอริทึมการทำComposite 78

3.20 อัลกอริทึมการทำConvolution 80

3.21 อัลกอริทึมการทำDilation 81

3.22 อัลกอริทึมการทำErosion 83

3.23 การทำ Morphological Gradient 84

3.24 การหมุนตามเข็มนาฬิกา 85

3.25 อัลกอริทึมการหมุนตามเข็มนาฬิกา 85

3.26 การหมุนทวนเข็มนาฬิกา 86

3.27 อัลกอริทึมการหมุนทวนเข็มนาฬิกา 86

3.28 การพลอตกราฟ Histogram 89

3.29 การหาค่าเฉลี่ย 89

3.30 การหาค่าเบี่ยงเบนมาตรฐาน 90

3.35 แสดงขั้นตอนการแสดงภาพ 91

3.36 แสดงขั้นตอนการแสดงการทำภาพInvert 92

3.37 แสดงขั้นตอนการแสดงภาพก่อนทำGrayScale 93

3.38 แสดงขั้นตอนการแสดงภาพหลังจากทำGrayScale 93

3.39 แสดงขั้นตอนการแสดงภาพก่อนทำMirror 94

3.40 แสดงขั้นตอนการแสดงภาพหลังทำMirror 94

3.41 แสดงขั้นตอนการแสดงภาพก่อนทำ Gamma 95

3.42 แสดงขั้นตอนการทำ Gamma โดยค่า Gamma เท่ากับ 0.5 95

3.43 แสดงขั้นตอนการทำ Gamma โดยค่า Gamma เท่ากับ 2 96

3.44 แสดงภาพก่อนการทำขั้นตอนการFlip 97

3.45 แสดงภาพก่อนการทำขั้นตอนการFlip 97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

3.46 แสดงภาพก่อนการทำHistogram Equalized 98

3.47 แสดงภาพหลังการทำHistogram Equalized 98

3.48 แสดงการสร้างภาพใหม่ด้วยการAnd 99

3.49 แสดงการสร้างภาพใหม่ด้วยการOr 100

3.50 แสดงการสร้างภาพใหม่ด้วยการXor 100

3.51 แสดงการสร้างภาพใหม่ด้วยการบวก 101

3.52 แสดงการสร้างภาพใหม่ด้วยการลบ 101

3.53 แสดงการสร้างภาพใหม่ด้วยการเฉลี่ยของ 2 ภาพ 102

3.54แสดงการสร้างภาพใหม่ด้วยการรวมกันของ 3 ภาพโดยภาพผลลัพธ์อยู่ทางซ้ายล่าง 102

3.55 การแสดงภาพก่อนการหมุนตามเข็มนาฬิกา 103

3.56 การแสดงภาพหลังการหมุนตามเข็มนาฬิกา 103

3.57 การแสดงภาพก่อนการหมุนทวนเข็มนาฬิกา 104

3.58 การแสดงภาพหลังการหมุนทวนเข็มนาฬิกา 104

3.59 การแสดงภาพก่อนการหมุน ณ องศาใดๆ 105

3.60 การแสดงภาพหลังการหมุนไป 45 องศา 105

3.61 การแสดงภาพก่อนการใช้ฟิลเตอร์ Low Pass 106

3.62 การแสดงภาพหลังการใช้ฟิลเตอร์ Low Pass 106

3.63 การแสดงภาพก่อนการใช้ฟิลเตอร์ High Pass 107

3.64 การแสดงภาพหลังการใช้ฟิลเตอร์ High Pass 107

3.65 การแสดงภาพก่อนการหาขอบอิมเมจโดยวิธีลาปลาส 108

3.66 การแสดงภาพหลังการหาขอบอิมเมจโดยวิธีลาปลาส 108

3.67 การแสดงภาพก่อนการทำให้อิมเมจนูน 109

3.68 การแสดงภาพหลังการทำให้อิมเมจนูน 109

3.69 การแสดงภาพก่อนการทำให้เป็นภาพวาดด้วยดินสอ 110

3.70 การแสดงภาพหลังการทำให้เป็นภาพวาดด้วยดินสอ 110

3.71 การแสดงภาพก่อนการหาลายเส้นตามวิธีของSobel 111

3.72 การแสดงภาพหลังการหาลายเส้นตามแนวนอน 111

3.73 การแสดงภาพหลังการหาลายเส้นตามแนวตั้ง 112

3.74 การแสดงภาพหลังการหาลายเส้นตามแนวนอนและแนวตั้ง 112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

3.75 การแสดงภาพก่อนการหาลายเส้นตามวิธีของPrewitt 113
3.76 การแสดงภาพหลังการหาลายเส้นตามแนวนอน 113
3.77 การแสดงภาพหลังการหาลายเส้นตามแนวตั้ง 114
3.78 การแสดงภาพหลังการหาลายเส้นตามแนวนอนและแนวตั้ง 114
3.79 การแสดงภาพก่อนการหาลายเส้นตามวิธีของFrei-Chen 115
3.80 การแสดงภาพหลังการหาลายเส้นตามแนวนอน 115
3.81 การแสดงภาพหลังการหาลายเส้นตามแนวตั้ง 116
3.82 การแสดงภาพหลังการหาลายเส้นตามแนวนอนและแนวตั้ง 116
3.83 การแสดงภาพก่อนการขยายพิกเซล 117
3.84 การแสดงภาพหลังการขยายพิกเซล 117
3.85 การแสดงภาพก่อนการลดขนาดพิกเซล 118
3.86 การแสดงภาพหลังการลดขนาดพิกเซล 118
3.87 การแสดงภาพก่อนการOpening 119
3.88 การแสดงภาพหลังการ Opening สังเกตว่าพิกเซลถูกเปิดเชื่อมต่อกันมากขึ้น 119
3.89 การแสดงภาพก่อนการClosing 120
3.90 การแสดงภาพหลังการ Closing สังเกตว่าพิกเซลถูกปิดเชื่อมต่อกันมากขึ้น 120
3.91 การแสดงภาพก่อนการแสดงส่วนประกอบทางสีแดง 121
3.92 การแสดงภาพหลังการแสดงส่วนประกอบทางสีแดง 121
3.93 การแสดงภาพหลังการแสดงส่วนประกอบทางสีเขียว 122
3.94 การแสดงภาพหลังการแสดงส่วนประกอบทางสีน้ำเงิน 122
ก-1 แสดงการติดตั้งโปรแกรม Mdip1 127
ก-2 แสดงรายละเอียดของการติดตั้งโปรแกรม Mdip1 127
ก-3 แสดงการเลือกพื้นที่ๆ เราต้องการติดตั้งลงไป 128
ก-4 แสดงขั้นตอนที่ตัวโปรแกรมกำลังดำเนินการติดตั้ง 128
ก-5 แสดงขั้นตอนการเข้าสู่โปรแกรมที่ Start Menu ของวินโดวส์ 129
ก-6 ภาพทะเลทรายก่อนจากการทำ Invert 130
ก-7 ภาพทะเลทรายที่ได้หลังจากการทำ Invert 130
ก-8 ภาพสะพานก่อนจากการทำ Invert 131
ก-9 ภาพสะพานที่ได้หลังจากการทำ Invert 131

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

ก-10 ภาพวิวก่อนการทำGray Scale 132

ก-11 ภาพวิวหลังการทำGray Scale 132

ก-12 ภาพยักษ์ก่อนการทำGray Scale 133

ก-13 ภาพยักษ์หลังการทำGray Scale 133

ก-14 ภาพน้ำตกก่อนการทำHistogram Equalized 134

ก-15 ภาพน้ำตกหลังการทำHistogram Equalized 134

ก-16 ภาพต้นไม้ก่อนการทำHistogram Equalized 135

ก-17 ภาพต้นไม้หลังการทำHistogram Equalized 135

ก-18 ภาพทะเลก่อนการเพิ่มความสว่างด้วยแกมมา 136

ก-19 ภาพทะเลหลังการเพิ่มความสว่างด้วยแกมมา 136

ก-20 ภาพตุ๊กตาก่อนการลดความสว่างด้วยแกมมา 137

ก-21 ภาพตุ๊กตาหลังการลดความสว่างด้วยแกมมา 137

ก-22 ภาพดวงจันทร์ที่ถูกสร้างใหม่โดยการใช้ลอจิคAvg 138

ก-23 ภาพก้อนเมฆที่ถูกสร้างใหม่โดยการใช้ลอจิคDiff 138

ก-24 ภาพสุริยุปราคาที่ถูกสร้างใหม่โดยการใช้ลอจิคXor 139

ก-25 ภาพตึกก่อนการทำให้มัวหรือเบลอ 140

ก-26 ภาพตึกหลังการทำให้มัวหรือเบลอ 140

ก-27 ภาพตึกก่อนการทำให้คมชัด 141

ก-28 ภาพต้นตึกหลังการทำให้คมชัด 141

ก-29 รูปทะเลก่อนการหาขอบภาพด้วยวิธีลาปลาส 142

ก-30 รูปทะเลหลังการหาขอบภาพด้วยวิธีลาปลาส 142

ก-31 รูปทะเลก่อนการหาขอบภาพด้วยวิธีลาปลาส 143

ก-31 รูปทะเลหลังการหาขอบภาพด้วยวิธีลาปลาส 143

ก-32 รูปร้านก่อนการทำให้นูน 144

ก-33 รูปร้านหลังการทำให้นูน 144

ก-34 รูปการ์ตูนก่อนการทำให้เป็นภาพวาดดินสอ 145

ก-35 รูปการ์ตูนหลังการทำให้เป็นภาพวาดดินสอ 145

ก-36 รูปดวงจันทร์ก่อนการหาขอบภาพ 146

ก-37 รูปดวงจันทร์หลังการหาขอบภาพ 146

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

ก-38 รูปตึกก่อนการหาขอบภาพ 147
ก-39 รูปตึกหลังการหาขอบภาพ 147
ก-40 รูปก้อนเมฆก่อนการหาขอบภาพ 148
ก-41 รูปก้อนเมฆหลังการหาขอบภาพ 148
ก-42 รูปวิวก่อนการลดขนาดพิกเซล 149
ก-43 รูปวิวหลังการลดขนาดพิกเซล 149
ก-44 รูปไอเฟลก่อนการเพิ่มขนาดพิกเซล 150
ก-45 รูปไอเฟลหลังการเพิ่มขนาดพิกเซล 150
ก-46 รูปตึกก่อนการOpening 151
ก-47 รูปตึกหลังการOpening 151
ก-48 รูปตึกก่อนการClosing 152
ก-49 รูปตึกหลังการClosing 152
ก-50 รูปเมืองก่อนการซูม 153
ก-51 รูปเมืองหลังการซูมออก 154
ก-52 รูปเมืองหลังการซูมเข้า 154

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
1 แสดงการวางแผนงานการทำโครงการ	3
2 แสดง Truth Table ของลอจิก And,Or และ Xor	35
3 แสดงการสรุปการทำงานของอัลกอริทึมต่างๆ	59
4 แสดงรายละเอียดเมนูหลักและเมนูย่อย	66
5 การกำหนดรูปแบบการหมุนในฟังก์ชัน PlgBlt	84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหาพิเศษ

ปัจจุบันนี้ คอมพิวเตอร์เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์ คอมพิวเตอร์ช่วยทำให้การดำเนินการต่างๆ สะดวกยิ่งขึ้น ปัจจุบันข้อมูลจากสื่อต่างๆ ส่วนใหญ่ล้วนนำเสนอออกมาเป็นภาพทั้งจากกล้องดิจิตอล โทรศัพท์มือถือ และ บางภาพที่ถ่ายมาอาจจะประสบปัญหาจากความไม่สมบูรณ์ของภาพ ทำให้ไม่ตรงต่อความต้องการของผู้ถ่าย ซึ่งการแก้ไขอาจจะต้องเสียเวลาในการถ่ายภาพนั้นใหม่ซ้ำอีกครั้ง ซึ่งการกลับไปบันทึกถ่ายภาพนั้นอีกครั้งซึ่งคงจะไม่สามารถทำได้เสมอไป เมื่อต้องคำนึงถึงผลกระทบทางด้านต่างๆที่จะมากหรือน้อย อาทิเช่น เวลา งบประมาณ อย่างไรก็ดีมันก็คือผลกระทบ หรือบางที ผู้ถ่ายภาพอาจต้องการสิ่งต่อเติมเพื่อที่จะให้ภาพนั้นมีความน่าสนใจและดูแปลกตามากยิ่งขึ้น ด้วยเทคโนโลยีที่มีอยู่ในปัจจุบัน ซึ่งอุปกรณ์หลักนั้นก็คือ คอมพิวเตอร์นั่นเอง

ดังนั้นโปรแกรมนี้มีคุณสมบัติช่วยในการแสดงผลภาพ และ ยังสามารถปรับปรุงคุณภาพให้ดีขึ้นในระดับหนึ่ง โดยใช้ความรู้ด้านเทคนิคการประมวลผลภาพ โดยเป็นแอปพลิเคชั่นของวินโดวส์ ซึ่งพัฒนาโดยภาษา Visual C++ เพื่อประโยชน์ในการช่วยลดค่าใช้จ่ายและเวลา ในการที่ต้องถ่ายภาพใหม่อีกครั้ง อีกทั้งช่วยในการเพิ่มเติมให้ภาพนั้นมีความน่าสนใจมากขึ้น

1.2 วัตถุประสงค์

-ศึกษาการทำงานของโปรแกรมและแก้ไขปัญหาภาพด้วยเทคนิคของการประมวลผลภาพต่างๆ

-สามารถปรับแต่งรูปภาพ ด้วยระบบงานทางคอมพิวเตอร์ กราฟิก (Computer Graphics)

-สร้างโปรแกรมประยุกต์เพื่อนำมาใช้กับภาพเพื่อจุดประสงค์ในการใช้ตกแต่งรูปภาพ โดยรับ input จากไฟล์ภาพ

-เสริมสร้างความรู้ทาง Computer Graphics เพื่อที่สามารถนำไปประยุกต์ใช้ได้ในอนาคต

-นำเทคนิคทางคณิตศาสตร์และสถิติมาใช้กับภาพมาใช้กับภาพที่ถูกแสดงในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปัญหา

ในหัวข้อปัญหาพิเศษนี้ได้นำความรู้ทางคอมพิวเตอร์ที่ได้รับจากการศึกษามาใช้ในการพัฒนา โปรแกรมคอมพิวเตอร์ซึ่งนำเสนอเป็นโปรแกรมแสดงภาพและปรับแต่ง ต่อเติมภาพ โดยนำความรู้จากคอมพิวเตอร์ กราฟิก (Computer graphic) และ เทคนิคการประมวลผลภาพ (Image Processing) โดยรูปแบบฟอร์แมทของภาพเป็นแบบ BMP สำหรับโปรแกรมที่ใช้พัฒนา คือ Microsoft Visual C++.NET

1.4 ขั้นตอนและการดำเนินงาน

-ศึกษาโปรแกรมที่ใช้พัฒนา เป็นขั้นตอนในการศึกษาถึงซอฟต์แวร์ที่นำมาใช้ในการเขียนโปรแกรม โดยจะเน้นการเขียนโปรแกรมเชิงวัตถุ (OOP) ด้วยภาษา Visual C++

-ศึกษาลักษณะของภาพ เป็นขั้นตอนในการศึกษาประเภทลักษณะ format ภาพ การเก็บเป็น file

-ศึกษา Image Processing และ เทคนิคต่างๆ เป็นขั้นตอนในการศึกษาขั้นตอน วิธีการ และ อัลกอริทึมของเทคนิคการประมวลผลภาพ

-วิเคราะห์และออกแบบการทำงานของโปรแกรม เป็นขั้นตอนในการศึกษาและออกแบบส่วนของการติดต่อกับผู้ใช้โดยออกแบบระบบออกเป็นส่วนๆ

-การพัฒนาโปรแกรม โดยการนำเอาวิธีการทางคอมพิวเตอร์ที่ได้เรียนและศึกษามาใช้ โดยทำการเขียนโปรแกรมตามแผนงานและกระบวนการที่ได้กำหนดไว้ในขั้นตอนการวิเคราะห์และออกแบบการทำงานของโปรแกรม

-การทดสอบและปรับปรุงโปรแกรม เป็นขั้นตอนในการทดสอบความผิดพลาดและการทำงานของโปรแกรม เพื่อที่ได้นำปัญหามาปรับปรุงแก้ไขข้อผิดพลาดที่เกิดขึ้น และ ความเป็นไปได้ของความสามารถของตัวโปรแกรม

-ทำเอกสารประกอบและอ้างอิง เป็นขั้นตอนในการจัดทำเอกสารประกอบโปรแกรม รวมทั้งเอกสารอ้างอิงในการศึกษาปัญหาพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 การวางแผนงานการทำโครงการ

1 ก.ค. - 31 ก.ค. ศึกษาและรวบรวมข้อมูล

1 ส.ค. - 31 ส.ค. ศึกษาการใช้ซอฟต์แวร์

1 ก.ย. – 15 ก.ย. ศึกษาโปรแกรมเชิงวัตถุ

16 ก.ย. – 10 ต.ค. วิเคราะห์และออกแบบการทำงานของโปรแกรม
เขียนแผนผังการทำงานโปรแกรม

11 ต.ค. – 10 ม.ค. พัฒนาโปรแกรม

11 ม.ค. – 26. ก.พ. ทดสอบและแก้ไขโปรแกรม

27 ก.พ. – 5 มี.ค. สรุปโครงงานปัญหาพิเศษ

ตารางที่ 1 ตารางแสดงการวางแผนงานการทำโครงการ

งาน	July	August	September	October	November	December	January	February	March
ศึกษาและรวบรวมข้อมูล									
ศึกษาการใช้ซอฟต์แวร์									
ศึกษาโปรแกรมเชิงวัตถุ									
วิเคราะห์และออกแบบการทำงานของโปรแกรม									
พัฒนาโปรแกรม									
ทดสอบและแก้ไขโปรแกรม									
สรุปโครงงานปัญหาพิเศษ									

Task

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

กฎเกณฑ์ ทฤษฎี หลักการ

2.1 บทนำ

การประมวลผลภาพ (Image Processing) ซึ่งเป็นสัญญาณภาพสองมิตินั้น จำเป็นต้องใช้การคำนวณแบบ 2 มิติ ในบทนี้จะกล่าวถึงข้อมูล คำจำกัดความ และ ความรู้พื้นฐานเกี่ยวกับทฤษฎีการประมวลผลภาพ เพื่อเป็นพื้นฐานในการทำความเข้าใจในบทต่อๆ ไป

2.2 กฎเกณฑ์ต่างๆ

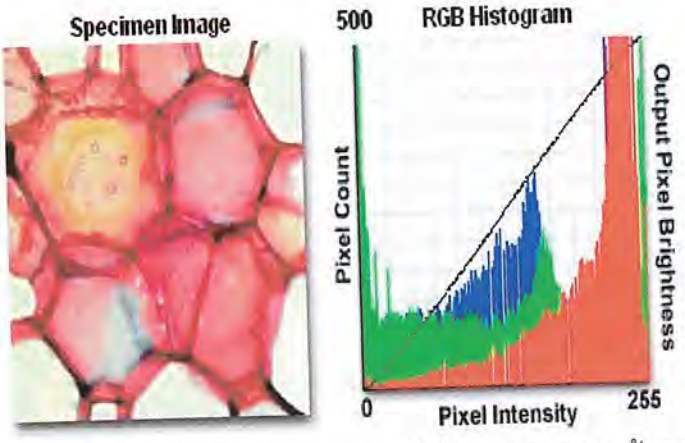
อาจจะกล่าวได้ว่า การประมวลผลภาพ หมายถึง การแสดงภาพที่เกิดจากการถ่ายรูปหรือจากการสแกนภาพให้ปรากฏบนจอคอมพิวเตอร์ วิธีการทางอิมเมจโปรเซสซิงก์ จะต่างกับวิธีการของ คอมพิวเตอร์กราฟิก กล่าวคือ ในระบบของคอมพิวเตอร์กราฟิก ตัวคอมพิวเตอร์เองจะเป็นตัวสร้างภาพ แต่เทคนิคทางอิมเมจโปรเซสซิงก์ นั้นจะใช้คอมพิวเตอร์สำหรับการจัดรูปแบบของสีและ แสงเงาที่มีอยู่แล้วในภาพให้เป็นข้อมูลทางดิจิตอล แล้วอาจจะมีวิธีการทำให้ภาพที่รับเข้ามานั้น มีความชัดเจนมากขึ้นก่อน จากนั้นก็จัดการกับข้อมูลดิจิตอลนี้ให้เป็นภาพส่งออกไปที่จอภาพของคอมพิวเตอร์อีกที วิธีการนี้มีประโยชน์ในการแสดงภาพของวัตถุที่เราไม่สามารถจะเห็นได้โดยตรง เช่น ภาพถ่ายทางดาวเทียม ภาพจากทีวีสแกนของหุ่นยนต์ในระบบอุตสาหกรรม ภาพที่ถ่ายด้วยมือถือ เป็นต้น

เมื่อภาพถ่ายถูกทำให้เป็นข้อมูลดิจิตอลแล้ว เราก็จะสามารถจัดการแก้ไขเปลี่ยนแปลงภาพนั้นได้โดยการจัดการกับข้อมูลดิจิตอลของภาพนั่นเอง ซึ่งเราก็จะใช้หลักการทางคอมพิวเตอร์กราฟิกมาใช้กับข้อมูลเหล่านี้ได้ เช่น ในการถ่ายภาพสำหรับโฆษณา เราสามารถจะทำให้ภาพที่เห็นเหมือนภาพถ่ายนั้นแปลกตาออกไปจากเดิมได้โดยการมีการเพิ่มภาพบางอย่าง หรือ บางส่วนของภาพนั้นหายไปทำให้เกิดภาพที่ไม่น่าเป็นจริงแต่ดูแล้วเหมือนเกิดขึ้นจริงได้ เป็นต้น

2.3 การนำเสนอข้อมูลภาพดิจิตอล (Digital Image Representation)

คำว่า อิมเมจ (Image) หมายถึง ความเข้มของแสงซึ่งแสดงได้ด้วยฟังก์ชันความเข้มของแสงในระนาบ 2 มิติ f(x, y) โดยโคออร์ดิเนตที่เกิดขึ้นที่ภาพจริง ณ จุดต่างๆ และ ค่าของฟังก์ชัน f ณ จุด (x, y) ใดๆ จะเป็นสัดส่วนโดยตรงกับความสว่างหรือระดับ gray level ของภาพที่จุดนั้นๆ ซึ่งแสดงได้ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.1 ข้อมูลภาพดิจิตอลแสดงถึงแนวแกน x และ y และ ฟังก์ชันแสดงความเข้มของแสง ณ จุด (x,y)

ในการพิจารณาข้อมูลภาพแบบดิจิตอลจะแทนด้วยเมตริกซ์หนึ่งซึ่งมีแถวและหลักที่มีลักษณะเป็นเอกลักษณ์ ก็จะได้ค่าระดับ Gray level ณ จุดต่างๆ โคออร์ดิเนตหรือจุดต่างๆ ที่เกิดขึ้นในเมตริกซ์ เรียกว่า พิกเซล (Pixel) หรือจุดย่อยของภาพ (Picture element) โดยทั่วไปขนาดของข้อมูลภาพสามารถเปลี่ยนแปลงได้แล้วแต่การใช้งาน โดยมากจะเลือกเป็นสี่เหลี่ยมจัตุรัสและแบ่งระดับ Gray level ด้วยสองยกกำลังด้วยเลขจำนวนเต็ม เช่น พื้นที่ 512 x 512 และมีระดับ Gray level 128 ระดับ

2.4 หลักการเบื้องต้นของการประมวลผลภาพ

2.4.1 พิกเซล (Pixel)

พิกเซล หรือ จุดภาพ เป็นหน่วยที่เล็กที่สุดของภาพกราฟิกที่สามารถแสดงออกทางจอภาพและอุปกรณ์แสดงผลประเภทกราฟิกอื่นๆของระบบคอมพิวเตอร์ พิกเซลมีความสำคัญต่อการสร้างกราฟิกในระบบคอมพิวเตอร์มาก เพราะในทุกๆส่วน เช่น จุด เส้น แบบลาย และ สีของภาพ ล้วนประกอบมาจากพิกเซลหนึ่งๆ อาจจะมีความเข้มและสีแตกต่างกันได้ ในอุปกรณ์แสดงกราฟิกอย่างง่ายอาจทำได้แค่เพียงแสดงหรือไม่แสดงพิกเซล พิกเซลแสดงผลบนตารางเมตริกซ์สี่เหลี่ยมจัตุรัสเช่นเดียวกับ ตารางกริดหรือกระดาษกราฟ แต่ละพิกเซลที่ปรากฏอาจเหมือนกับสีที่ระบายลงบนช่องเล็กๆ ในกริด ซึ่งแต่ละช่องของกริดมีค่าเป็นจำนวนเต็ม กลุ่มของพิกเซลที่เรียงกันอยู่บนตารางกริด เรียกว่า แผนที่บิต หรือ บิตแมบ (Bitmap) ซึ่งเป็นเมตริกซ์ของพิกเซล

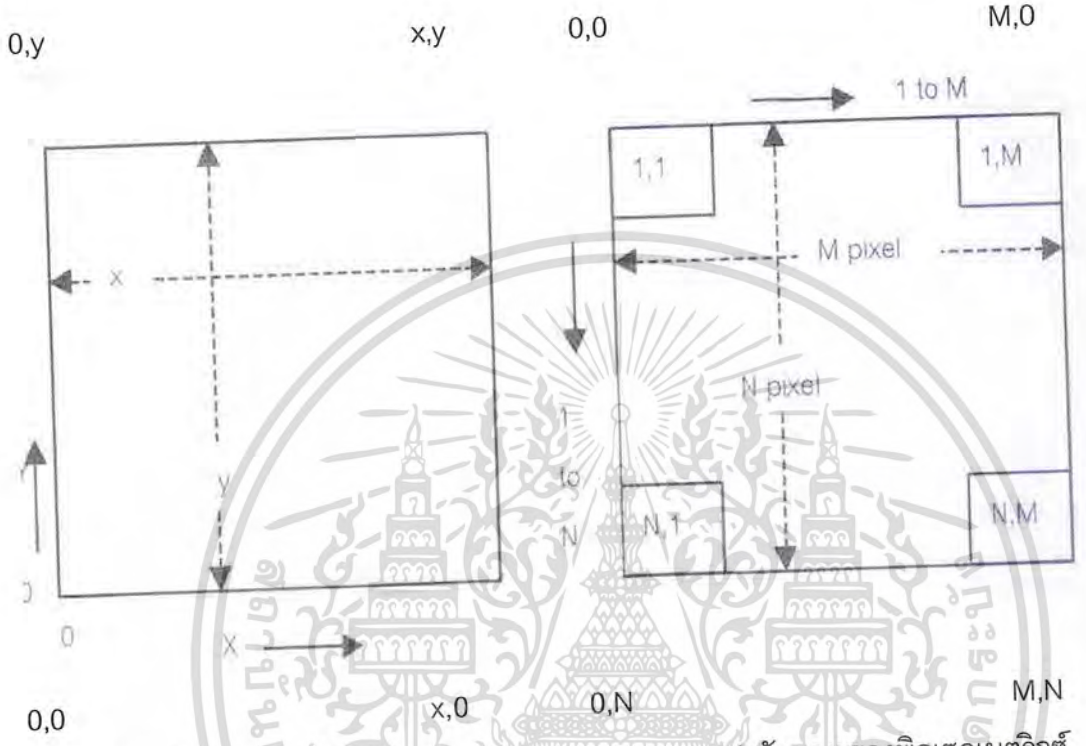
การแสดงข้อมูลของภาพดิจิตอลสามารถอธิบายได้ด้วยเมตริกซ์ MxN และให้จุดต่างๆ ที่อยู่ในเมตริกซ์เป็นจุด x, y ใดๆ ซึ่งเป็นส่วนประกอบของภาพ ในแต่ละจุด x, y ใดๆ เรียกว่า พิกเซล หรือ จุดภาพในแต่ละพิกเซลจะแสดงให้เราเห็นได้ด้วยฟังก์ชันของความเข้มของแสง เมื่อเราเปรียบเทียบระหว่างภาพและพิกเซลเมตริกซ์ (Pixel matrix) ดังรูป 2.2 จะเห็นได้ว่าจุดกำเนิดของภาพจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ที่มุมล่างซ้าย แต่จุดกำเนิดของพิกเซลจะอยู่มุมบนซ้าย ซึ่งเป็นลักษณะการประมวลผลภาพในระบบกราฟิกคอมพิวเตอร์

กล่าวคือ $i = x$ เมื่อ $1 \leq i \leq M$

$j = (N-y)$ เมื่อ $1 \leq j \leq N$



a) ลักษณะของรูปภาพ b) ลักษณะของพิกเซลเมตริกซ์

รูปที่ 2.2 ความสัมพันธ์ของภาพโดยทั่วไปกับพิกเซลเมตริกซ์

เมื่อกำหนด

I = พิกัดหลังจากแปลงเป็นพิกเซลเมตริกซ์ทางแกนนอน

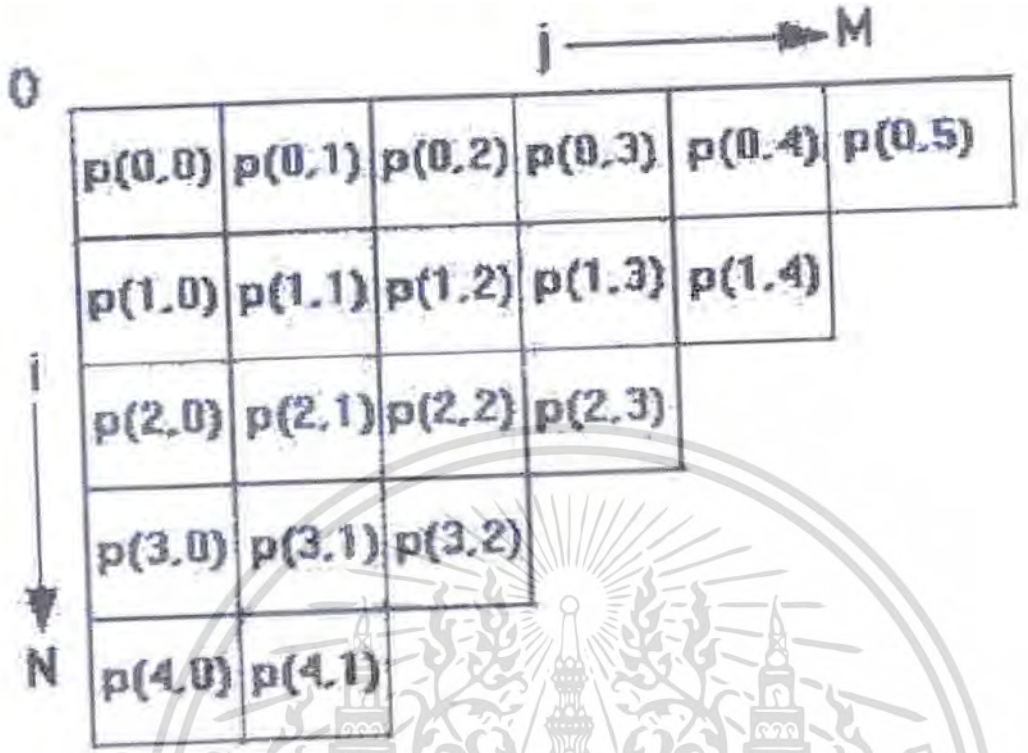
J = พิกัดหลังจากแปลงเป็นพิกเซลเมตริกซ์ทางแกนตั้ง

M = จำนวนสูงสุดของพิกเซลในแนวนอน

N = จำนวนสูงสุดของพิกเซลในแนวตั้ง

เมื่อเราให้จุดต่างๆ บนเมตริกซ์เป็น P(i, j) ใดๆ จะแสดงดัชนีพิกเซลในเมตริกซ์ได้ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.3 ดัชนีพิกเซลในเมตริกซ์

ค่าของพิกเซลหรือฟังก์ชัน P(i, j) ณ จุดใดๆ จะแสดงได้ด้วยค่าความเข้มของแสงซึ่งอาจจะแบ่งได้หลายระดับ ถ้ามีแค่ 2 ระดับก็จะมีแค่ 0 กับ 1

ลักษณะข้อมูลภาพ

1. ภาพ 2 ระดับ คือ มีแค่จุดขาวกับดำเท่านั้น โดยแต่ละจุดเป็นข้อมูล 1 บิต
2. ภาพ 16 ระดับ ซึ่งในแต่ละภาพจะเป็นข้อมูล 4 บิต ซึ่งทำให้สามารถแสดงผลได้ 16 ระดับสี หรือ 16 ระดับสีเทา ซึ่งขึ้นกับว่าเป็นข้อมูลภาพสีหรือขาว-ดำ
3. ภาพ 256 ระดับ ซึ่งในแต่ละจุดภาพจะเป็นข้อมูล 8 บิต ซึ่งทำสามารถแสดงภาพได้ 256 ระดับสี หรือ ระดับสีเทา ขึ้นอยู่กับว่า ภาพนั้นเป็นภาพสี หรือ ขาว-ดำ
4. ภาพ True Color ซึ่งในแต่ละจุดจะเป็นข้อมูลขนาด 24 บิต ทำให้สามารถแสงผลภาพได้เหมือนภาพจริงที่สุด เพราะ สามารถแสดงสีได้ถึง 16,777,216 สี ภาพ true color สามารถแสดงได้เฉพาะภาพสีเท่านั้น ไม่สามารถแสดงภาพขาว-ดำได้

การแสดงภาพนี้ใช้วิธีตั้งค่าของแม่สีในตารางสี โดยอาจจะเลือกสีแบบ 16 สี จาก 64 สี หรือ 16 สี จาก 262, 144 สี ขึ้นอยู่กับโหมดของการแสดงผล สำหรับ true color ไม่มีการเลือกสีแสดงผลโดย

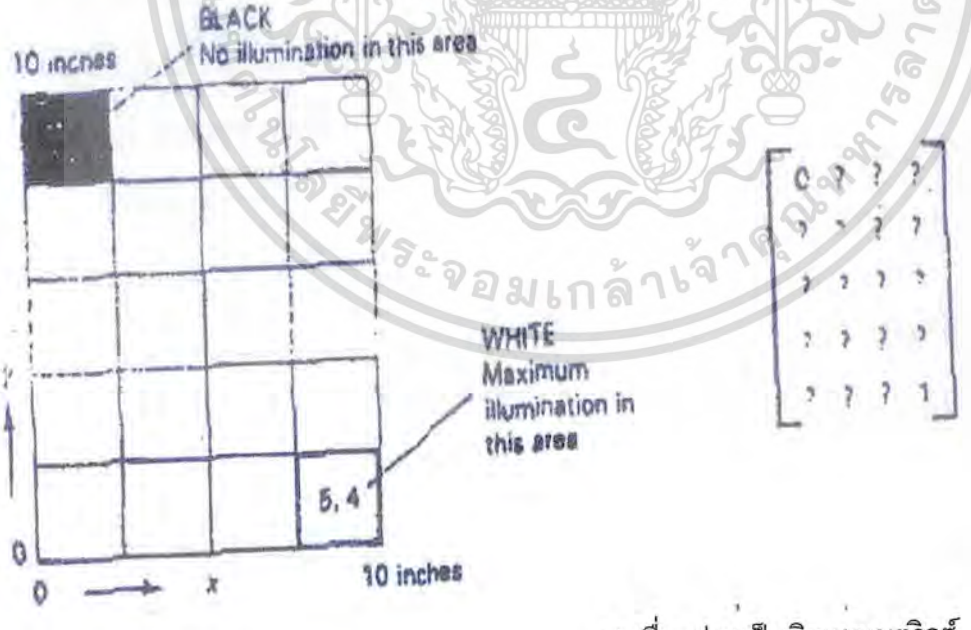
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งค่าสี RGB ผ่าน D/A สีละ 8 บิต ออกไปเลย ความแตกต่างของการแสดงผลสีและภาพขาว-ดำ คือ ภาพขาว-ดำจะต้องตั้งให้แม่สีทั้ง 3 สีมีค่าเท่ากัน เนื่องจาก VGA กำหนดให้แม่สีแต่ละสีใช้รีจิสเตอร์ (register) 6 บิต ทำให้แม่สีแต่ละแม่สีแสดงผลได้เพียง 34 ระดับ เท่านั้น ยังผลเราแสดงผลภาพ 256 ระดับเท่านั้น หากต้องการให้เห็นจริงทั้ง 256 ระดับต้องแสดงใน true color mode แล้วให้ RGB มีค่าเท่ากัน ซึ่งโหมดนี้ใช้รีจิสเตอร์ 8 บิต สำหรับแม่สีแต่ละสี

2.4.2 **เอสเป็กเรโชของภาพ** (Image Aspect Ratio)

เอสเป็กเรโชของภาพ คือ อัตราส่วนระหว่างพิกเซลทางแนวขวางและจำนวนพิกเซลทางแนวดิ่งที่ใช้ในการสร้างภาพ หากจะยกตัวอย่างเทียบกับกระดาษกราฟ จะเห็นได้ว่าภาพบิตแมปใดๆ ก็ตาม จะมีจำนวนพิกเซลคงที่ในมิติแนวขวาง และ แนวดิ่ง ซึ่งอัตราส่วนมีไว้อ้างถึงขนาดของภาพ มักจะเขียนในรูปของ 800 × 600 (ซึ่งหมายถึง รูปภาพที่มี 800 พิกเซลในแนวขวาง และ 600 บรรทัดของพิกเซลในแนวดิ่ง) เราสามารถคำนวณหาจำนวนพิกเซลทั้งหมดในรูปภาพได้ โดยการคูณเลขทั้งสองนี้เข้าด้วยกัน นั้นคือ รูปภาพที่มีเอสเป็กเรโช 800 × 600 จะมีทั้งหมด 480,000 พิกเซล ซึ่งจำนวนดังกล่าวนี้ ไม่ได้หมายถึงขนาดไฟล์ของภาพนั้นๆ

2.4.3 **ตำแหน่งของพิกเซล** (Pixel Position)



a) ภาพ b) เมื่อแปลงเป็นพิกเซลเมตริกซ์

รูปที่ 2.4 การแปลงภาพให้เป็นพิกเซลเมตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของจุดภาพหรือพิกเซลทุกจุดจะต้องอยู่ภายในพื้นที่ MxN เมื่อเราพิจารณาจากรูป 2.4 (a) จะเห็นว่าไม่มีแสง ณ จุดบริเวณมุมบนซ้าย และ บริเวณที่สว่างที่สุดอยู่ที่มุมล่างขวาของภาพซึ่งมีขนาด 10 x 10 นิ้ว พื้นที่ที่ไม่มีแสงแสดงได้ด้วยศูนย์ ส่วนพื้นที่ที่สว่างที่สุดแสดงได้ด้วยหนึ่งและภาพที่เห็นได้คือ เมตริกซ์ 5 x 4 (5 แถว, 4 คอลัมน์) แต่ละส่วนของภาพจะกว้าง 2.5 นิ้ว บริเวณมุมบนซ้ายจะเป็น 0 และ พื้นที่ 2.5 x 2 นิ้ว จะเป็นหนึ่งซึ่งนำมาเขียนเป็นเมตริกซ์ได้ ดังรูปที่ 2.4 (b)

2.5 รีโซรูชัน (Resolution)

รีโซรูชั่น (Resolution) หมายถึง รายละเอียดที่อุปกรณ์แสดงกราฟิกชนิดหนึ่งมีอยู่ ค่ารีโซรูชั่น มักระบุเป็นจำนวนพิกเซลในแนวนอนคือแนวแกน X และจำนวนพิกเซลในแนวตั้งคือแนวแกน Y ดังนั้น รีโซรูชั่น 720 x 384 จึงหมายความว่า อุปกรณ์กราฟิกชนิดนี้สามารถแสดงพิกเซลในแนวนอนได้ไม่เกิน 720 พิกเซล และ แสดงพิกเซลในแนวตั้งได้ไม่เกิน 348 พิกเซล ผู้ผลิตอุปกรณ์กราฟิกแสดงพิกเซลบางรายจะระบุค่ารีโซรูชั่นเป็น 3 ระดับ คือ ระดับสูง (high resolution) ระดับปานกลาง (medium resolution) และ ระดับต่ำ (low resolution) โดยพิจารณาจากจำนวนพิกเซลในแนวนอนอย่างเดียว ซึ่งมีหลักว่า ถ้าค่าน้อยกว่า 128 ถือว่าเป็นระดับต่ำ ถ้าค่าระหว่าง 128 ถึง 512 เป็นระดับปานกลาง และถ้าค่าสูงกว่า 512 ขึ้นไปถือว่าเป็นระดับสูง สำหรับจอภาพปกติ ถ้ามีค่ารีโซรูชั่นมากกว่า 1500 ตาเราจะมองเกือบไม่เห็นแต่ละพิกเซล คือจะมองเห็นเป็นภาพที่มีความละเอียดคมชัดสูงมาก คอมพิวเตอร์กราฟิกที่ใช้กับฟิลม์ถ่ายรูปในระดับมืออาชีพจะต้องใช้ค่ารีโซรูชั่นสูงถึง 3000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Original image of Golden Gate Bridge:

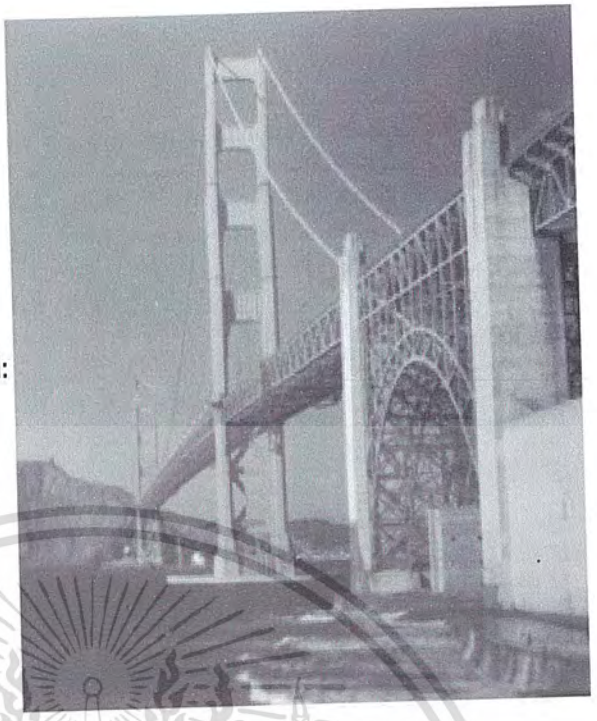
Image resampled at half the original resolution:

With image size adjusted:

Without image size adjusted:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Image resampled at twice the original resolution:



With image size adjusted:

Without image size adjusted:

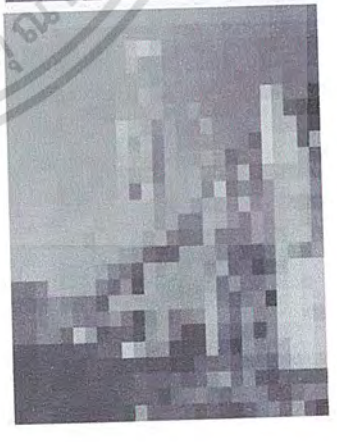


Image which has been downsampled at one-tenth resolution, and upsampled at 10x resolution:

รูปที่ 2.5 แสดงการทำ Resolution ของภาพในแต่ละแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การประมวลผลสีของภาพ (Color Image Processing)

การประมวลผลสีของภาพ แบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ Full Color กับ Pseudo-color ในขั้นแรกถามว่าภาพที่ได้มาโดยปกตินั้นจะเป็นภาพที่เต็มไปด้วยสีสัน เช่น สีของกล้องวีดีโอ หรือสีของสแกนเนอร์ ในขั้นที่สอง คือ มีปัญหาในการกำหนดเขตสีว่าจะเป็นสีเข้มสีเดียวหรือกำหนดขอบเขตความเข้มของสี เมื่อไม่นานมานี้เมื่อไม่นานมานี้มีการเทียบกระบวนการประมวลผลสีของภาพกับกระบวนการอื่นๆ ซึ่งส่วนใหญ่การประมวลผลสีของภาพจะทำเสร็จในขั้น Pseudo-color เป็นที่น่าสังเกตว่า การประมวลผลสีของภาพมีเพิ่มขึ้นในปี 1980 นั้นมีการจัดทำตัว color sensors และทำตัวในการประมวลผลที่เกี่ยวกับสีภาพ

2.6.1 Color model

จุดประสงค์ของ Color model คือ เป็นสิ่งที่ช่วยในการระบุรายละเอียดจากสีในบางมาตรฐาน ทั่วๆไป Color model เป็นรายละเอียดจากระบบ 3-D coordinate และ subspace ของระบบ ซึ่งแต่ละสีจะแสดงในรูปแบบจุดๆเดียว (Single point) color model ที่เราจะกล่าวถึงนั้นจะกล่าวถึง RGB color model เพียงเท่านั้น

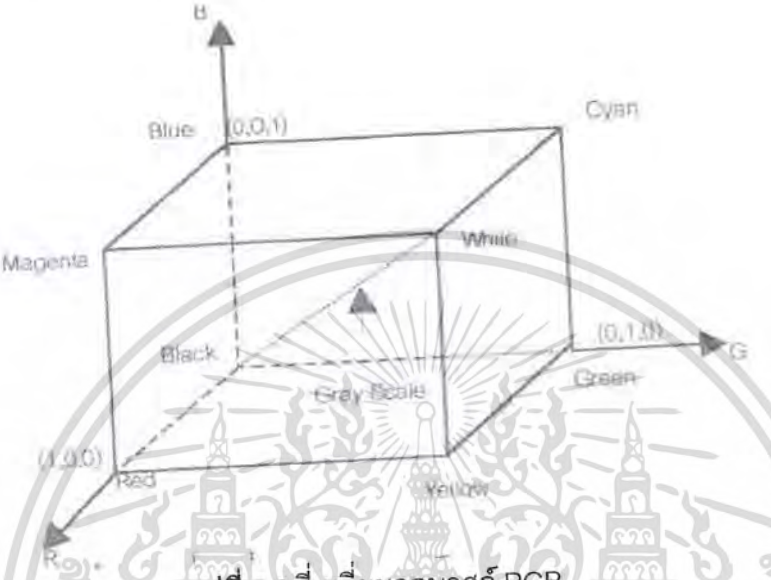
2.6.2 The RGB color model

RGB model ในแต่ละตัวโมเดลนั้นประกอบไปด้วยสีแดง เขียว น้ำเงิน เป็นพื้นฐาน ตัวโมเดลจะอยู่บนพื้นฐานของระบบ Cartesian coordinate ช่วงต่อของสีแสดงได้ ซึ่งค่าของ RGB จะอยู่ที่ 3 มุม คือ Cyan, Magenta, Yellow ซึ่งแต่ละมุมจะมีสีดำอยู่ ณ จุดกำเนิด และ สีขาวอยู่ห่างจากจุดกำเนิด โมเดลตัวนี้ gray scale จะขยายไปจากสีดำไปถึงสีขาวตามแนวเส้นระหว่างจุดทั้งสองและสีต่างๆ จะเป็นจุดอยู่บน หรือ ในสี่เหลี่ยมลูกบาศก์นั้น โดยกำหนดจากเส้นเวกเตอร์ที่ขยายจากจุดกำเนิด เพื่อความสะดวกให้สันนิษฐานว่าค่าของสีทั้งหมดนั้นปกติอยู่ในสี่เหลี่ยมลูกบาศก์ เพื่อแสดงว่าค่าทั้งหมดของ R,G,B นั้นถูกสมมติให้อยู่ใน [0,1] ภาพใน RGB color model นั้นประกอบด้วยภาพที่อยู่ในแนวราบ 3 แนว โดยจะมี 1 สีในแต่ละแนว เมื่อป้อนเข้าสู่ RGB monitor แล้วภาพ 3 ภาพเหล่านี้จะอยู่รวมกันบนฉากแสง ที่จะผลิตภาพที่ประกอบด้วยสีต่างๆ ดังนั้น การใช้ RGB model สำหรับการประมวลผลภาพเป็นการทำความเข้าใจภาพต่างๆ เหล่านั้นเมื่อแสดงอยู่ในข้อกำหนดของแนวราบของสีทั้ง 3 อีกอย่างหนึ่ง คือ สีของกล้องวีดีโอได้ใช้ประโยชน์จาก RGB format ในการทำ digital ภาพ ซึ่งการกระทำอันนี้สำคัญต่อตัว โมเดลในการประมวลผลภาพ

ตัวอย่างหนึ่งที่ดีที่สุดของประโยชน์ของ RGB model อยู่ในการประมวลผลจากข้อมูลทางอากาศและการแยกเงาของภาพถ่ายจากดาวเทียม ยกตัวอย่างเช่น รูปจำลองของ LANDSAT ที่ประกอบด้วยภาพดิจิตอล 4 ภาพ แต่ละภาพมีฉากอย่างเดียวกัน แต่จะนำผ่านแนวเงาสีที่ต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ window (ช่อง) 2 ช่องจากช่องทั้งหมดอยู่ในส่วนของสเปกตรัมที่สามารถมองเห็นได้ จะเห็นคร่าวๆเป็นสีแดงและสีเขียว และ 2 ช่องอื่นๆ จะอยู่ในส่วนของสเปกตรัมที่ต่ำกว่าสีแดง ดังนั้น ภาพแนวราบแต่ละอันมีความหมายตามหลักฟิสิกส์ และ การรวมกันของสีที่ใช้โมเดล RGB และ แสดงความเข้าใจเมื่อเห็นสีฉาก เมื่อการแบ่งเส้นสีของภาพอยู่บนพื้นฐานของส่วนประกอบเงาสีของวัตถุ



รูปที่ 2.6 สี่เหลี่ยมลูกบาศก์ RGB

สมมติว่าปัญหาหนึ่งที่เพิ่มขึ้นมาจากสีของภาพหน้าคน เป็นส่วนของเงาที่ถูกบดบังความเท่ากันของฮิสโทแกรมจะปฏิบัติกับความเข้มของสีเท่านั้น วิธีที่ชัดเจน คือ ภาพในแนวราบแต่ละอันมีความเท่ากัน ของฮิสโทแกรมที่ต่างหากออกมา ส่วนของภาพที่ถูกซ่อนโดยแสงเงาทั้งหมดมีท่าทางว่าจะเพิ่มขึ้น อย่างไรก็ตามความเข้มของภาพแนวราบทั้ง 3 จะถูกเปลี่ยนแตกต่างกัน ผลลัพธ์ในการเปลี่ยนแปลงของสีที่สัมพันธ์กันระหว่างตัวมัน ผลจะเป็นส่วนสำคัญที่แสดงถึงคุณสมบัติของสี เช่น สีโทนเนื้อ ซึ่งจะไม่ปรากฏตามธรรมชาติเมื่อมองบน RGB มอนิเตอร์ โมเดลสีบางอันจะโต้ตอบทางทิศทางที่ดีกว่าที่เหมาะสมสำหรับบทสร้างในเรขาคณิต

2.6.3 Pseudo-color Image Processing

ตอนนี้จะพูดถึงการกำหนดสีในภาพเป็นสีเดียวบนพื้นฐานของคุณสมบัติต่างๆของ Gray-level

Intensity slicing (ส่วนแบ่งของความเข้มสี)

เทคนิคของ Intensity slicing (บางครั้งเรียกว่า density หรือ น้ำหนักของวัตถุ) และ เส้นสีเป็นหนึ่งในตัวอย่างที่ง่ายที่สุดของกระบวนการทำภาพแบบ Pseudo-color Image Processing ถ้าภาพที่เราเห็นเป็นภาพที่มีความเข้มเป็นภาพแบบ 2 มิติ ขั้นตอนสามารถแสดงให้เห็นว่า หนึ่งใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวางเส้นตามแนวขนานให้เท่ากันกับระดับ (Plane) ของภาพแต่ละระดับที่แบ่งนั้นมีหน้าที่พื้นที่เส้นตัด ตัวอย่างของการใช้ plane ที่ f(x, y) = l ให้แบ่งเส้นเขตออกเป็น 2 ระดับ

ความแตกต่างของสีถูกกำหนดให้อยู่ในแต่ละข้างของ Plane ดังในรูป ภาพใดๆก็ตามที่มีระดับสูงกว่าสีเทาสูงกว่า plane จะถูกกำหนดรหัสกับตัวอื่นๆ ระดับหลายระดับที่วางอยู่บน plane ตัวมันเองอาจถูกตัดสินโดยใช้สี 1 ใน 2 สี ผลลัพธ์ภาพที่มี 2 สีนั้นที่เกี่ยวข้องกับรูปร่างที่สามารถถูกควบคุมโดยการเคลื่อนไหวของส่วนแบ่งขึ้นและลงในแกนระดับสีเทาโดยทั่วๆ ไป เทคนิคอาจจะถูกสรุปได้ตามนี้ สมมติว่า ระดับ M ถูกกำหนดให้อยู่ในระดับ $I_1, I2, \ldots, I_m$ และนำไปสู่ I_o ที่แสดงสีดำที่ [f(x, y) =O] และ แสดงสีขาว [f(x, y) =L] ดังนั้นสมมติว่า O<M<L ระดับ M แบ่งสเกลสีเทาเป็นแนว M+1 และการกำหนดสีเป็นการทำตามความเกี่ยวข้องกัน

$$f(x, y) = c_k \text{ if } f(x, y) \in R_k$$

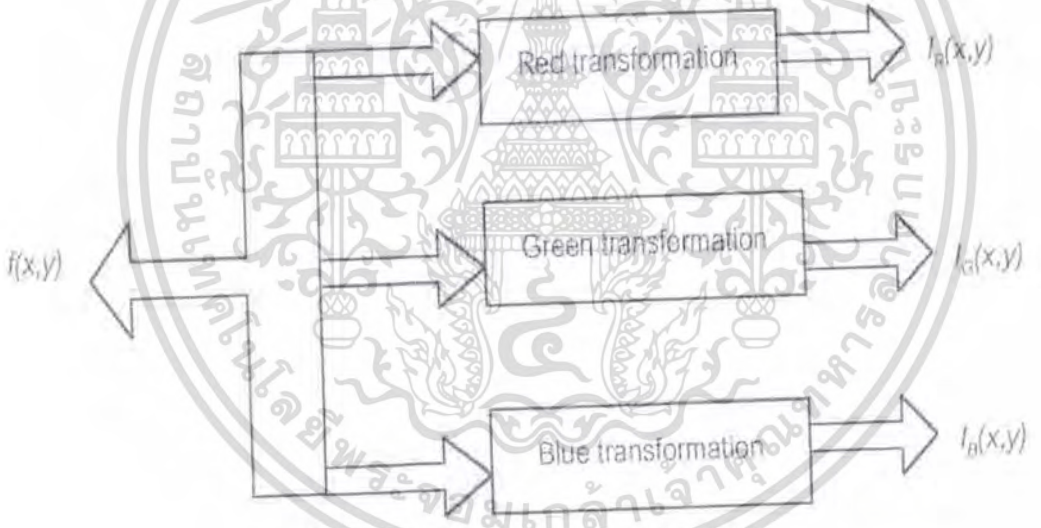
จุดมุ่งหมายของ Plane เป็นประโยชน์เดิมๆ สำหรับการแสดงทางคณิตศาสตร์ของเทคนิคการแบ่งความเข้มสี รูปที่ 2.6 แสดงการเลือกทางว่าจำกัดแผนภาพที่เหมือนกัน ในรูปที่ 2.6 การกำหนดระดับการนำเข้าสีเทานั้นๆ ถูกกำหนดโดย 1 ใน 2 สี ขึ้นอยู่กับว่ามันอยู่ในระดับที่สูงกว่าหรือต่ำกว่าค่าของ I_i เมื่อระดับที่มากกว่าถูกใช้ หน้าที่ของแผนภาพก็จะถูกนำไปในอีกฟอร์มหนึ่ง

Gray level to Color Transformation (ระดับสีเทาที่นำไปสู่กระบวนการแปลงสี)

การเปลี่ยนแปลงชนิดอื่นๆ เป็นเรื่องทั่วไปและการทำเช่นนี้อาจจะได้รับการแพร่หลายของ Pseudo-color ที่สูงขึ้นมากกว่าเทคนิคการแบ่งอย่างง่ายที่กล่าวมาก่อนหน้านี้วิธีที่ให้แสดงได้ดังรูป 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 ทางเลือกการนำเข้าของ Gray Level



รูปที่ 2.8 การแปลง Gray level ไปเป็น RGB

โดยพื้นฐานทั่ว ๆ ไป ความคิดรองๆ ลงมาที่ใกล้กับการเปลี่ยนแปลง 3 แบบ ในระดับสีเทาของการนำภาพเข้า ผลลัพธ์ทั้ง 3 อย่างนี้ เป็นการแยกกระบอกฉีดสีแดง สีเขียว และ สีน้ำเงิน ของสีบนจอโทรทัศน์ กระบวนการนี้ก่อให้เกิดภาพที่ประกอบไปด้วยสีต่างๆ ซึ่งประกอบไปด้วยคลื่นความถี่ที่เปลี่ยนแปลงไปตามธรรมชาติ แสดงให้เห็นว่า การเปลี่ยนแปลงต่างๆ เหล่านี้บนค่าระดับเทาของภาพไม่ใช่หน้าที่ที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงในก่อนหน้านี้ ขั้นตอนที่แสดงอยู่ในรูปที่ 2.3 เป็นเทคนิคเป็นเทคนิคพิเศษที่มีการอธิบายไว้ Piecewise linear of the gray levels ก่อให้เกิดสีต่างๆ อย่างไรก็ตาม วิธีการที่ใช้พิจารณาในตอนนี้ สามารถทำให้อยู่บนพื้นฐานของความเรียบ nonlinear function ถูกใช้ก่อให้เกิดเทคนิคต่างๆ

บิตแมบ เป็นวัตถุเกี่ยวกับรูปภาพที่ถูกสร้างขึ้น เคลื่อนย้าย ย่อขนาด หมุน และ เก็บข้อมูลรูปภาพ

Device Context เป็นรูปร่างของพวกนิยามกลุ่มวัตถุเกี่ยวกับรูปภาพ คุณสมบัติของรูปภาพ และ วิธีเกี่ยวกับรูปภาพที่มีผลต่อการแสดงภาพออกมา

Device Independent เป็นลักษณะเด่นของ Microsoft Operating Systems ใน Win 32 base แอปพลิเคชันสามารถที่จะวาดและเขียนรูปออกมาบนอุปกรณ์ได้หลายชนิดซึ่ง software นั้นจะรองรับอุปกรณ์อิสระต้องบรรจุด้วย two dynamic link libraries ได้แก่ GDI.DLL ซึ่งจะอ้างอิงถึงส่วนติดต่อกับอุปกรณ์กราฟฟิก และ ส่วนที่อ้างอิง device driver เช่น VGA.DLL อ้างอิงถึง VGA Display, EPSON9.DLL อ้างถึง Epson FX-80 Printer

2.7 ข้อมูลรูปภาพแบบไบนารี (Binary Image Data)

ข้อมูลรูปภาพแบบไบนารีถูกเรียกว่า "รูปภาพ ขาว และ ดำ" มันประกอบไปด้วยจุด พิกเซล แต่ละจุดสีของพิกเซลเป็นสีดำหรือสีขาวและหนึ่งพิกเซลแทนด้วยหนึ่งบิต ดังรูปซึ่งเป็น ตัวอย่างของข้อมูลรูปภาพแบบไบนารี รูปนี้มีขนาด 12*8 พิกเซล ทางด้านซ้ายของรูปแสดง ภาพอิมเมจ และด้านขวาแสดงข้อมูลที่เป็นบิต ในข้อมูลนี้จุดพิกเซลสีขาวแสดงด้วยบิต 0 และ จุดดำแสดงด้วยบิต 1 ในทางคอมพิวเตอร์ข้อมูลส่วนใหญ่จะถูกประมวลผลโดยการใช้หน่วยไบต์ โดย 1 ไบต์ประกอบไปด้วย 8 บิต ดังนั้นข้อมูลรูปภาพแบบไบนารีจะถูกแปลงเป็นข้อมูลแบบไบต์ ใน กรณีนี้ข้อมูลแบบบิตจะถูกแบ่งไปเป็นข้อมูลไบต์ โดยการแบ่งกลุ่มทีละ 8 บิต ถ้าความกว้างของ อิมเมจไม่สามารถแบ่งทีละ 8 บิตได้ลงตัว บิตเสริมจะถูกใส่ลงไปในส่วนท้ายสุดของแต่ละแถว ของข้อมูลอิมเมจ ข้อมูลรูปภาพจะถูกเก็บในอาร์เรย์ ชนิดข้อมูลอักขระที่จะประกาศได้ดังนี้

*unsigned char *buf;*

buf[0] จะบรรจุข้อมูลไบต์แรกของข้อมูลอิมเมจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

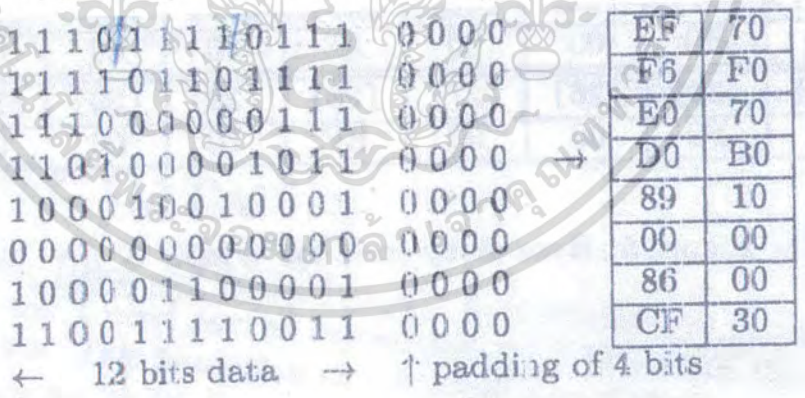
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ข้อควรระวังสำหรับภาพบิตแมบบนWin32

ใน Win32 DIB(Device independent Bitmap) ข้อมูลรูปภาพจะถูกเก็บจากด้านซ้ายล่างไปยังด้านขวาบน และแต่ละแถวข้อมูลต้องถูกแบ่งอกเป็น 4 ไบต์ ถ้าจำนวนของข้อมูลแต่ละแถวไม่สามารถแบ่งออกเป็น 4 ไบต์ บิตเสริมจะถูกใส่ลงไปในส่วนท้ายสุดของแต่ละแถว ดังในรูป พิกเซลรูปภาพเป็น 12*8=96 พิกเซล แต่ขนาดของข้อมูลจะถูกแปลงเป็น 32 ไบต์

Figure 1: Binary image to binary code

รูปที่ 2.9 การแปลงรูปไบนารี่ไปเป็นไบนารี่โค้ด

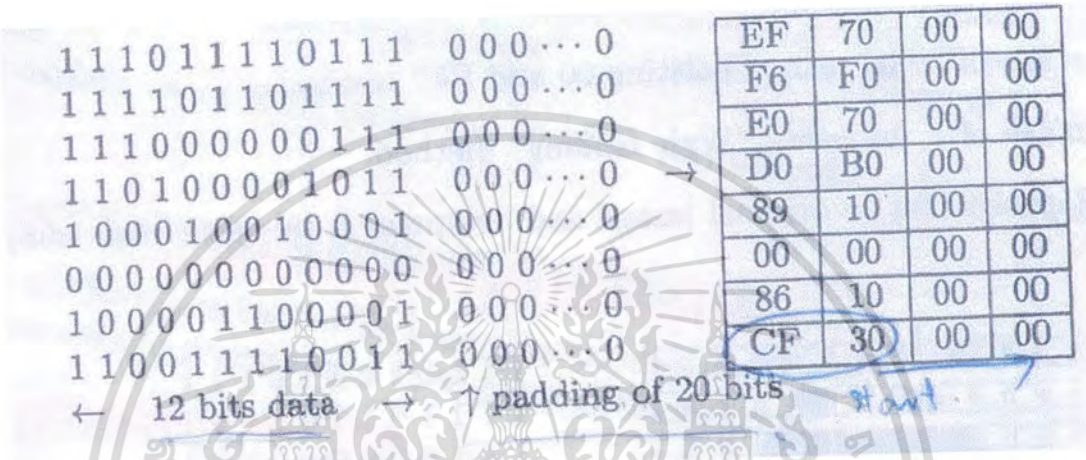


รูปที่ 2.10 การแปลงโค้ดฐานสองไปเป็นฐานสิบหก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
51791

buf	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
	EF	70	F6	F0	E0	70	D0	B0
buf	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
	89	10	00	00	86	10	CF	30

รูปที่ 2.11 การเก็บโค้ดในรูปอาร์เรย์



รูปที่ 2.12 การแปลงโค้ดฐานสองไปเป็นฐานสิบหกของรูปแบบWin32

buf	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
	CF	30	00	00	86	10	00	00
buf	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
	00	00	00	00	89	10	00	00
buf	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]
	D0	B0	00	00	E0	70	00	00
buf	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]
	F6	F0	00	00	EF	70	00	00

รูปที่ 2.13 การเก็บโค้ดในรูปอาร์เรย์ของรูปแบบWin32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 บิตแมบ (Bitmap) และการใช้งาน

ไฟล์อิมเมจหรือรูปภาพบิตแมบเป็นส่วนสำคัญในการเขียนโปรแกรมแบบ GUI หรือ Graphic User Interface ซึ่งช่วยในการอำนวยความสะดวกแก่ผู้ใช้โปรแกรม เช่น การเอารูปบิตแมบวางลงบนปุ่มกด การนำรูปบิตแมบมาเป็นสีรองพื้น การนำบิตแมบมาสร้างเมนู เป็นต้น

2.8.1 บิตแมบ

ในที่นี้หมายถึงจุดข้อมูลพิกเซลของอิมเมจที่เก็บไว้ในอาร์เรย์ หรือ บัฟเฟอร์บิตแมบ การเก็บข้อมูลจุดพิกเซลในบัฟเฟอร์บิตแมบจะมากหรือน้อยขึ้นอยู่กับความลึกของสี หรือ เรียกว่า จำนวนบิตต่อพิกเซล ยิ่งมีจำนวนบิตต่อพิกเซลมากๆ ก็จะใช้หน่วยความจำเครื่องคอมพิวเตอร์มากขึ้นไปด้วย และ ทำให้คุณภาพของกราฟิกดีตามไปด้วย ในทางตรงข้าม ถ้าจำนวนบิตต่อพิกเซลน้อย คุณภาพของกราฟิกก็จะแย่ แต่จะใช้หน่วยความจำน้อยไปด้วย การจะใช้ขนาดจำนวนบิตต่อพิกเซลมากหรือน้อยขึ้นอยู่กับงานนั้นโดยเฉพาะ ถ้างานนั้นต้องการกราฟิกคุณภาพสูง ก็จะใช้จำนวนบิตต่อพิกเซลมาก รวมทั้งมอนิเตอร์ต้องรองรับจำนวนสีได้ด้วย ไม่เช่นนั้นภาพกราฟิกก็จะไม่สามารถแสดงสีได้ครบ ในระบบวินโดวส์ ถ้าอิมเมจนั้นมีจำนวนสีมากกว่าจำนวนสีของมอนิเตอร์ระบบวินโดวส์ก็จะใช้การแม็บสี ด้วยการทำ Palette ซึ่งจะพบได้ในคอมพิวเตอร์รุ่นแรกๆ ตัวอย่างเช่น มีภาพอิมเมจหนึ่งที่มีจำนวนสีเท่ากับ 65536 สี และนำมาใช้กับเครื่องคอมพิวเตอร์ที่มีการ์ดจอสีที่แสดงได้เพียง 256 สี ฉะนั้นเอาท์พุตหน้าจอคอมพิวเตอร์ จะแสดงภาพอิมเมจได้เพียง 256 สี

บิตแมบในวินโดวส์จะถูกแบ่งออกเป็น 2 ชนิด คือ

- ชนิด Device Dependent Bitmap (DDB)
- ชนิด Device Independent Bitmap (DIB)

ชนิด DDB บิตแมบชนิดนี้ใช้กับวินโดวส์รุ่นแรกๆ ก่อนที่จะออกจากเวอร์ชัน 3 บิตแมบชนิด DDB จะขึ้นอยู่กับอุปกรณ์ฮาร์ดแวร์ที่ใช้อยู่ หรือ การ์ดจอมอนิเตอร์ เช่น ความลึกของสีและพิกเซลต่อนิ้ว ทำให้คุณภาพบิตแมบจะเปลี่ยนแปลง เมื่อมีการโอนย้ายไฟล์ระหว่างเครื่องคอมพิวเตอร์ บิตแมบชนิด DDB จะเก็บรายละเอียดโครงสร้าง BITMAP ไว้ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Typedef struct tagBITMAP
{
        LONG bmType;
        LONG bmWidth;
        LONG bmHeight;
        LONG bmWidthBytes;
        WORD bmPlanes;
        WORD bmBitsPixel;
        LPVOID bmBits;
} BITMAP;
```

โดยที่ bmType กำหนดชนิดของไฟล์บิตแมบ โดยปกติจะกำหนดเป็นค่า 0

bmWidth กำหนดความกว้างของบิตแมบในหน่วยพิกเซล โดยปกติจะต้องมากกว่า 0

bmHeight กำหนดความสูงของบิตแมบในหน่วยพิกเซล โดยปกติจะต้องมากกว่า 0

bmWidthBytes กำหนดจำนวนไบท์ต่อการสแกนหนึ่งแถว

bmPlanes กำหนดจำนวนเพลนสีของบิตแมบ

bmBitsPixel กำหนดจำนวนบิตที่ใช้อธิบายถึงสีของพิกเซล

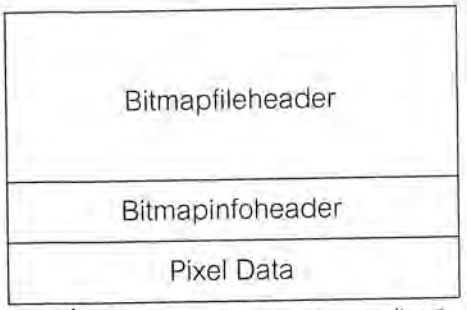
bmBits ชี้ไปยังที่เก็บข้อมูลภาพพิกเซลในอาร์เรย์

ชนิด DIB บิตแมบชนิดนี้ใช้กับวินโดวส์ตั้งแต่เวอร์ชัน 3 ขึ้นไป ซึ่งได้ปรับปรุงจากบิตแมบชนิด DDB ให้ดีขึ้น บิตแมบชนิด DIB จะไม่ขึ้นอยู่กับอุปกรณ์ฮาร์ดแวร์ ทำให้มีคุณภาพที่ดีกว่า DDB บิตแมบชนิดนี้จะประกอบด้วย ข้อมูลต่างๆ ดังนี้

- ส่วนหัว (Header) ซึ่งจะเก็บไว้ในโครงสร้าง BITMAPINFOHEADER และ BITMAPFILEHEADER
- ตารางสี (Color Palette) ซึ่งจะเก็บไว้ในโครงสร้าง RGBQUAD
- ข้อมูลพิกเซล (Pixel Data) ซึ่งจะเก็บไว้ในอาร์เรย์ รวมทั้งข้อมูลเกี่ยวกับการบีบอัดข้อมูล

นอกจากนี้แล้วยังมีชนิด DIB Section ซึ่งใช้กับวินโดวส์ NT และมีประสิทธิภาพในการแสดงผลที่รวดเร็วกว่าชนิดแรกๆ ทั้งนี้เพราะ DIB Section สามารถถูกเลือกเข้าอุปกรณ์ดีไวส์คอนเท็กซ์ หรือเรียกย่อๆ ว่า ดีซี ได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 โครงสร้างของบิตแมบไฟล์

โดยจะขอชี้แจงเป็นส่วนๆ 3 ส่วนจากที่กล่าวไปข้างต้น ได้ดังนี้

1. Bitmap File Header จะเป็นข้อมูลที่อยู่ส่วนแรกสุดของไฟล์บิตแมบ ซึ่งมีลักษณะ ดังนี้

```
Typedef struct tagBITMAPFILEHEADER
{
WORD bfType;
DWORD bfSize;
WORD bfReserved1;
WORD bfReserved2;
DWORD bfOffBits;
} BITMAPFILEHEADER, FAR *LPBITMAPFILEHEADER,
*PBITMAPFILEHEADER;
```

bfType ใช้บอกชนิดของภาพซึ่งจะต้องเป็นค่า 0x4D42

bfSize ใช้บอกขนาดของภาพทั้งหมด มีหน่วยเป็นไบต์

bfReserved1 ต้องเป็น 0

bfReserved2 ต้องเป็น 0

bfOffBits ใช้บอกขนาดของheaderทั้งหมด มีหน่วยเป็นไบต์

2. Bitmap Info Header เป็นส่วนที่อยู่ถัดจาก Bitmap File Header มีลักษณะดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Typedef struct tagBITMAPINFOHEADER
        {
        DWORD biSize;
        LONG biWidth;
        LONG biHeight;
        WORD biPlanes;
        WORD biBitCount;
        DWORD biCompression;
        DWORD biSizeImage;
        LONG biXPelsPerMeter;
        LONG biYPelsPerMeter;
        DWORD biCirUsed;
        DWORD biCirImportant;
        }   BITMAPINFOHEADER,   FAR   *LPBITMAPINFOHEADER,
*PBITMAPINFOHEADER;
```

biSize ใช้บอกขนาดของheaderส่วนนี้

biWidthใช้บอกความกว้างของภาพ

biHeight ใช้บอกความสูงของภาพ

biPlanes ใช้บอกจำนวนระนาบของภาพ ค่านี้ต้องเป็น 1

biBitCount ใช้บอกจำนวนไบต์ที่ใช้บอกค่าสีต่อหนึ่งจุดสี

biCompression ใช้บอกเรื่องการย่อของภาพว่าย่อแบบใด จะใช้กับ Compressed Bottom-Up Bitmap ซึ่งมีค่าได้ ดังนี้

- BI_RGB หมายถึง ไม่ทำการย่อภาพ
- BI_RLEB หมายถึง การย่อแบบ Run Length Encoder(RLE) ซึ่งจะใช้ 8 บิต ต่อ 1 จุดสี
- BI_RLE4 หมายถึง การย่อแบบ Run Length Encoder(RLE) ซึ่งจะใช้ 4 บิต ต่อ 1 จุดสี
- BI_BITFIELDS หมายถึง จะใช้ Double Word หนึ่งตัวต่อ 1 จุดสี (32 บิต)
- BI_JPEG หมายถึง การย่อแบบมาตรฐาน JPEG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BiSizeImage ใช้บอกขนาดของภาพ เฉพาะส่วนที่เป็นภาพ มีหน่วยเป็นไบต์

BiXPelsPerMeter ใช้บอกค่าความละเอียดในแนวนอน

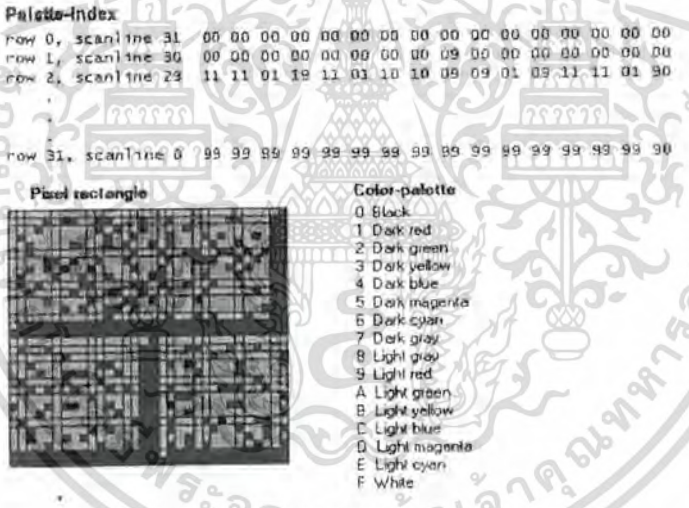
BiYPelsPerMeter ใช้บอกค่าความละเอียดในแนวตั้ง

BiCirUsed ใช้บอกค่าดัชนีสีจากตารางสี เช่น ถ้าค่านี้เป็น 0 ภาพจะใช้จำนวนสีเท่ากับ biBitCount ซึ่งจะต้องดูความสัมพันธ์กับค่า biCompression

BiCirImportant ใช้บอกค่าจำนวนสีที่ต้องใช้ ถ้าค่านี้เป็น 0 หมายถึงใช้ทุกสี

2.8.2 ตัวภาพ

ตัวภาพจะเก็บค่าสีของจุดที่ 1 จุดที่ 2 และ จุดที่ 3 เรียงลำดับไปเรื่อยๆ (โดยในตัวอย่างนี้จะใช้ 1 จุด ต่อ 4 บิต) โดยจุดแรกจะเริ่มจาก มุมล่างซ้ายของภาพ จุดที่สองคือ จุดที่อยู่ทางขวาของจุดแรก แล้วไล่ตามจุดไปเรื่อยๆ จนหมดแถวแล้วมาเริ่มใหม่ที่บรรทัดที่สองจากข้างล่าง โดยจะสแกนอย่างนี้ไปจนหมดภาพ ซึ่งเรียกว่าการ Scan Line



รูปที่ 2.15 ภาพของการ ScanLine

2.9 การทำกระบวนการอิมเมจดิจิตอล (Digital Image Processing)

โดยปกติแล้วสายตาของคนทั่วไปจะมองเห็นภาพทิวทัศน์วิวต่างๆ เป็นลักษณะแบบอนาล็อก ซึ่งสามารถอธิบายได้ด้วยคณิตศาสตร์ที่มีตัวแปรแบบนับได้อย่างต่อเนื่อง แต่คอมพิวเตอร์จะใช้เลขฐานสองเป็นหลักในการคำนวณ เมื่อนำภาพอิมเมจมาแปลงเข้าสู่เครื่องคอมพิวเตอร์ อิมเมจนั้นจะกลายเป็น อิมเมจดิจิตอล (Digital Image)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1 อิมเมจดิจิตอล

อิมเมจดิจิตอลเป็นผลมาจากการสุ่มค่าในระบบพิกัด Space หรือ (Spatial Coordinate) ดังรูปที่ 2.16 และการกระทำ Quantization ของค่าระดับความสว่าง (Brightness Value) หรือความเข้ม (Intensity) ระบบพิกัด Space นี้จะใช้กับการแสดงอิมเมจดิจิตอล ซึ่งจะมีขนาดความกว้างและความสูงของอิมเมจแสดงในแกน Y และ X ตามลำดับ ส่วนจุดใดๆ ที่วางบนระนาบ XY จะเป็นฟังก์ชัน f(x,y) และ เรียกว่า พิกเซล (Pixel) ที่แสดงถึงค่าระดับความเข้ม ซึ่งจะเป็นจำนวนที่นับได้จำกัด (Finite Number) แบบไม่ต่อเนื่อง หรือเรียกว่า Discrete Quantity ค่า Discrete Quantity เป็นผลมาจากการทำ Quantization โดยจะใช้การแปลงจากอนาล็อก (Analog) เป็นดิจิตอล (Digital)

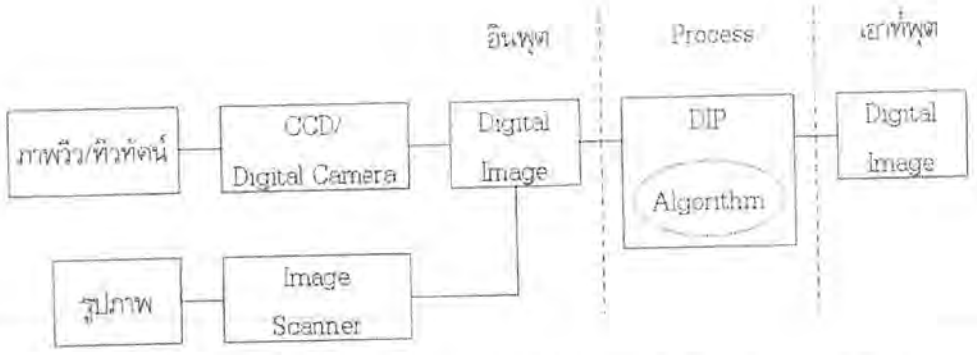
รูปที่2.16 ระบบพิกัด Space

2.9.2 การทำกระบวนการอิมเมจดิจิตอล

มีสองเหตุผลใหญ่ๆ ที่ต้องการทำกระบวนการอิมเมจดิจิตอล คือ เพื่อปรับปรุงอิมเมจดิจิตอลให้มองเห็นได้ง่ายขึ้น และเพื่อปรับปรุงอิมเมจให้หุ่นยนต์ตีความหมาย หรือเข้าใจจดจำรูปร่างลักษณะได้อย่างแม่นยำ ตัวอย่างเช่น การจดจำตัวอักษร หรือ Optical Character Recognition (OCR) ที่สามารถจดจำตัวอักขระได้ถึง 99.9% การปรับปรุงอิมเมจให้ใช้พื้นที่น้อยลง การตรวจสอบลายพิมพ์มือของแต่ละบุคคล เป็นต้น

ดังนั้นอาจกล่าวง่ายๆได้ว่าการทำกระบวนการอิมเมจดิจิตอลหรืออิมเมจโปรเซสซิงก์ คือ การนำอิมเมจดิจิตอลเข้ามาทำกระบวนการโดยการใส่ฟังก์ชันอัลกอริทึมต่างๆ เข้าไป ก็จะได้เอาท์พุตเป็นอิมเมจดิจิตอล ที่ตรงตามแนวความคิดของการทำกระบวนการนี้ เราสามารถแสดงภาพตั้งแต่การนำภาพวิว ทิวทัศน์ จนถึงเอาท์พุต ดังรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การทำกระบวนการอิมเมจดิจิตอลหรืออิมเมจโปรเซสซิงก์

2.10 วิธีการอ่านข้อมูลพิกเซลของอิมเมจ

จากสมการที่ 1 จุดที่วางอยู่ในพิกัด Space นี้ก็คือพิกเซล (Pixel) หรือ Picture Element ซึ่งก็คือ ความสว่างหรือค่า Luminance (L) ของอิมเมจถ้าอิมเมจนั้นเป็นภาพขาวดำ ขนาด 8 บิต จะมีค่า Lเท่ากับ 2^8 หรือเท่ากับ 256 ระดับ คือ ระดับตั้งแต่ 0(พิกเซลเป็นสีดำ) จนถึง 255 (พิกเซลเป็นสีขาว) ($0 \leq L \leq 255$) บางครั้งค่าความสว่าง (L) อาจมีความหมายถึงความละเอียดของอิมเมจ (Image Resolution) ถ้าพิกเซลเป็นภาพขาว-ดำ จะอ่านค่าอิมเมจดิจิตอลในรูปแบบเมตริก 2 มิติ ขนาด M*N ได้ดังนี้

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,N-1) \\ & & \\ f(M-1,0) & f(M-1,1) & f(M-1,N-1) \end{bmatrix}_{M\times N} \quad \ldots (1.)$$

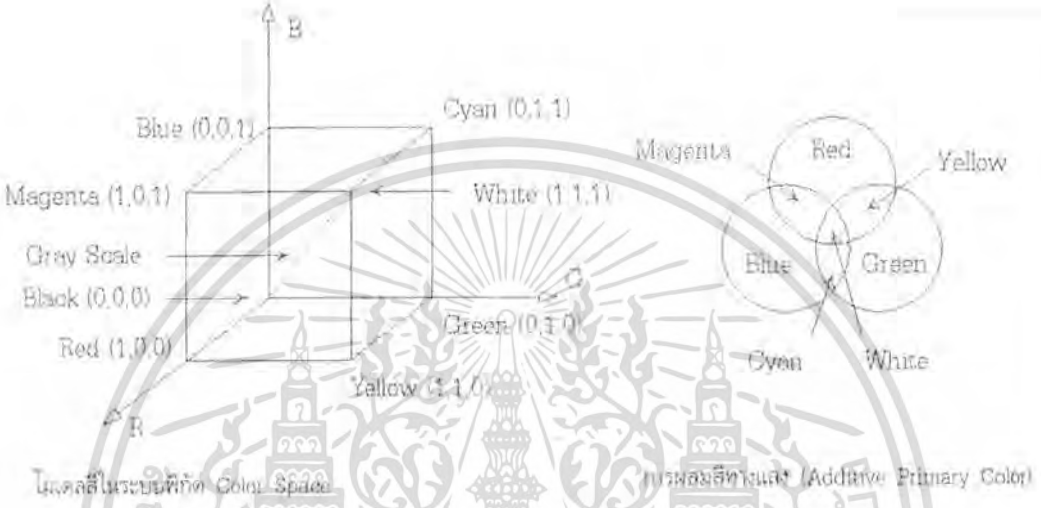
โดยที่ ค่า f(x,y) จะอยู่ในช่วง 0 ถึง 255 ($0 \leq f(x,y) \leq L$)

สมมุติว่าอ่านค่าพิกเซลจากอิมเมจหนึ่งได้ f(x,y)เท่ากับ 10 แสดงว่า จุดพิกเซลนั้นมีความสว่างน้อยมาก หรือ ค่อนข้างจะดำ ถ้าค่าที่อ่านได้เป็น 250 แสดงว่า จุดพิกเซลนั้นมีความสว่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 โมเดลสี (Color Model)

ดังที่ได้กล่าวไปแล้วว่า โมเดลสี หรือ Color Model ประกอบด้วย 3 แม่สีหลัก ได้แก่ สีแดง เขียว น้ำเงิน ถ้านำระบบสีมา plot กราฟในระบบพิกัด Color Space โดยแต่ละสีมีค่า 0 ถึง 1 (0 แสดงถึง ความมืด และ 1 แสดงถึง ความสว่าง) จะได้ภาพการผสมสีทางแสงหรือการบวกแม่สีเข้าด้วยกัน (Additive Primary Color) ดังรูปที่ 2.18



โมเดลสีในระบบพิกัด Color Space การผสมสีทางแสง (Additive Primary Color)

รูปที่ 2.18 โมเดลสีและการผสมสีทางแสง

ถ้าแต่ละแม่สีเป็นขนาด 8 บิต รวมทั้งหมดเท่ากับ เท่ากับ 24 บิต ซึ่งสามารถสร้างสีได้ถึง 256*256*256 เท่ากับ 16,777,216 สี ในที่นี้ใช้พิกเซลอิมเมจที่มีแต่ละแม่สีเท่ากับ 8 บิต หรือ เรียกว่า มีความลึกเท่ากับ 24 บิต เป็นหลัก

ถ้าพิกเซลเป็นภาพสีมีขนาด 24 บิต จะอ่านค่าอิมเมจดิจิตอลในรูปแบบ 2 มิติ ขนาด M*N เหมือนกับในสมการ (1.) แต่ค่า f(x,y) จะอยู่ในช่วงที่ประกอบด้วย

R ระดับ 0 จนถึง 255 ($0 \leq R \leq 255$)

G ระดับ 0 จนถึง 255 ($0 \leq G \leq 255$)

และ B ระดับ 0 จนถึง 255 ($0 \leq B \leq 255$)

ในบางครั้งถ้าต้องการแปลงโมเดลสี ให้เป็นขาว-ดำ ซึ่งก็คือ Gray Scale จะใช้สมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Gray Scale = 0.299*R+0.578G+0.114B ... (2.)

แต่เราสามารถใช้อีกสมการ โดยการหาค่าเฉลี่ยทั้งสามสี ดังนี้

$$\text{Gray Scale} = \frac{R+G+B}{3} \quad \ldots (3.)$$

จากรูปที่ 3 ค่า Gray Scale ก็คือค่าที่อยู่ในช่วง (0,0,0) จนถึง (1,1,1)

2.12 การอ่านข้อมูลอิมเมจในหน่วยความจำ

ในที่นี้เราจะอ่านข้อมูลจากไฟล์อิมเมจชนิด Bmp,Jpeg,Tiff และ Png ซึ่งจะต้องเป็นขนาดความลึกเท่ากับ 24 บิตเท่านั้น เพื่อที่จะเข้าใจถึงการอ่านข้อมูลอิมเมจบิตแมบในหน่วยความจำเครื่องคอมพิวเตอร์ เราจะเริ่มต้นด้วยการหาขนาดหน่วยความจำที่จะต้องใช้เมื่อโหลดอิมเมจหนึ่งขึ้นมา จากนั้นหาพอยน์เตอร์ที่ชี้ไปยังค่าพิกเซลบิตแมบในบล็อกหน่วยความจำ

ขนาดหน่วยความจำที่ต้องใช้ เมื่ออ่านข้อมูลไฟล์อิมเมจชนิดต่างๆ ขนาดบล็อกหน่วยความจำอิมเมจนี้ จะคำนวณได้จากความกว้างคูณกับความสูงและคูณกับความลึกของอิมเมจ โดยไม่สนใจว่าอินพุตอิมเมจจะเป็นชนิดใด เพราะเมื่ออ่านเข้าสู่หน่วยความจำ จะถูกแปลงเป็นชนิดของบิตแมบ (Bmp) เท่านั้น

ตัวอย่างการอ่านอิมเมจ เช่น อิมเมจชนิด Jpeg มีความกว้างและความสูงเท่ากับ 256 หน่วย และ ความลึกเท่ากับ 24 บิต เราต้องใช้หน่วยความจำเครื่องคอมพิวเตอร์อย่างน้อยที่สุดเท่ากับ $256*256*24_{10}$ = 180 KByte (1 ไบต์ = 8 บิต)

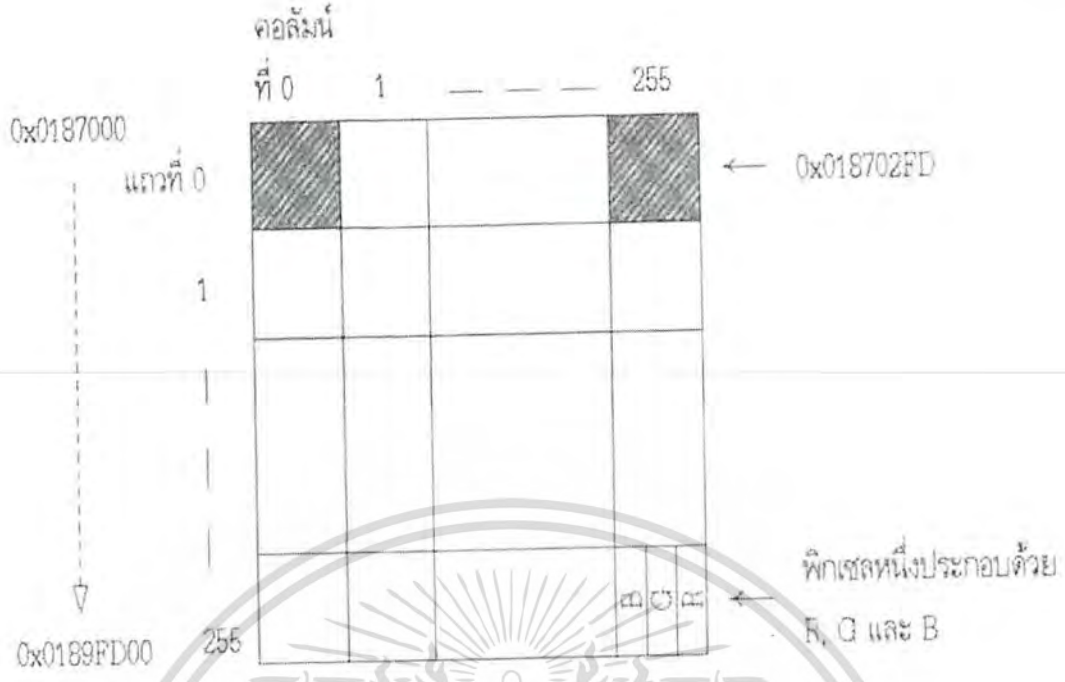
2.12.1 การหาพอยน์เตอร์ที่ชี้ไปยังจุดเริ่มต้นหรือเกือบท้ายสุด

การหาพอยน์เตอร์ที่ชี้ไปยังจุดพิกัดเริ่มต้นของบิตแมบ หรือ เกือบท้ายสุดในบล็อคหน่วยความจำ จะใช้ฟังก์ชัน GetBits ร่วมกับฟังก์ชัน GetPitch และหาพอยน์เตอร์ที่ชี้ไปยังพิกเซลเฉพาะจะใช้ฟังก์ชัน GetPixelAddress

ในการหาพอยน์เตอร์ จะเริ่มต้นด้วยการหา Pitch ของอิมเมจก่อน โดยใช้ฟังก์ชัน GetPitch ถ้าฟังก์ชันนี้ คืนค่าเป็นบวก แสดงว่า การอ่านค่าบิตแมบจะสแกนจากบนลงล่าง และเมื่อใช้ฟังก์ชัน GetBits จะคืนค่าพอยน์เตอร์ที่มุมบนซ้ายสุดของบล็อคหน่วยความจำอิมเมจ ซึ่งจะชี้ ณ จุดพิกเซลเริ่มต้นที่แถวบนสุด คอลัมน์แรก

ฟังก์ชัน GetPitch ถ้าฟังก์ชันนี้ คืนค่าเป็นลบ แสดงว่าการอ่านค่าบิตแมบจะสแกนจากแถวล่างสุดขึ้นไปด้านบน เมื่อใช้ฟังก์ชัน GetBits จะคืนค่าพอยน์เตอร์ที่มุมล่างซ้ายสุดของบล็อคหน่วยความจำอิมเมจ หรือ เกือบท้ายสุด ซึ่งจะชี้ ณ จุดพิกเซลเริ่มต้นที่แถวล่างสุด คอลัมน์แรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 การอ่านข้อมูลพิกเซล เมื่อฟังก์ชัน GetPitch คืนค่าเป็นบวก

รูปที่ 2.20 การอ่านข้อมูลพิกเซล เมื่อฟังก์ชัน GetPitch คืนค่าเป็นลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.21 การอ่านข้อมูลพิกเซลโดยใช้พอยน์เตอร์ pSrc

2.13 การทำกระบวนการจุดพิกเซล (Point Processing)

การทำรูปแบบนี้เป็นแบบที่ง่ายที่สุดในการปรับปรุงคุณภาพอิมเมจให้ดีกว่าอิมเมจต้นฉบับ ในการกระทำกับจุดพิกเซลนั้นสามารถกระทำได้หลายๆ กรณีด้วยกัน ไม่ว่าจะเป็นการนำค่าคงที่ใดๆ มาดำเนินการทางคณิตศาสตร์ เช่นการ บวก ลบ คูณ หาร รวมทั้งการดำเนินการด้วยลอจิก And,Or และ Xor การเปลี่ยนแปลงที่เกี่ยวกับความสว่าง เป็นต้น ในที่นี้เราจะพิจารณาเฉพาะการแปลงที่เกี่ยวกับความสว่าง ได้แก่ การแปลงสีให้เป็นขาวดำ การกลับสี (Invert Color) และ การแก้ไขพิกเซลด้วย Gamma การสร้างฮิสโทแกรม การทำ Histogram Equalised และ การบวกลบอิมเมจ การใช้ลอจิก And,Or และ Xor ดังรายละเอียดต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.1 **การแปลงที่เกี่ยวกับความสว่าง** (Luminance Transformation)

บางครั้งการแปลงชนิดนี้อาจเรียกว่า การแปลงความเข้ม (Intensity Transformation) โดยใช้สมการ 1/3(R+G+B) หรือ 0.299R+0.587G+0.114B ค่าที่ได้จากการแปลงจะอยู่ในระดับความสว่าง ตั้งแต่ 0 ถึง 255 โดยที่ 0 แสดงถึงความมืด ถ้าแสดงบนจอคอมพิวเตอร์จะเป็นสีดำ และ 255 แสดงถึงความสว่าง ถ้าแสดงบนจอคอมพิวเตอร์จะเป็นสีขาว

วิธีการแปลง เริ่มต้นด้วยการหาพอยน์เตอร์ที่ชี้ไปยังพิกเซล ณ ตำแหน่งพิกัดที่ต้องการแปลง โดยใช้ฟังก์ชัน GetPixelAddress จากนั้นจึงอ่านข้อมูลของแต่ละสี มาเข้าสมการข้างต้นนี้ เราก็จะได้พิกเซลใหม่ที่จะมาแทนที่พิกเซลเดิม

ในการใช้พอยน์เตอร์ที่ได้จากฟังก์ชัน GetPixelAddress จะชี้ไปยังสีน้ำเงินก่อน ตามด้วยสีเขียว และ สีแดง (ไม่ใช่ สีแดง เขียว และน้ำเงิน)

ถ้าต้องการแปลงพิกเซลสี ให้เป็นสีขาวดำตลอดทั้งทั้งอิมเมจ เราก็สแกนทุกแถวทุกคอลัมน์ด้วยใช้คำสั่ง Loop for

2.13.2 **การทำ** Segment

เป็นการแบ่งแยกภาพวัตถุกับฉากให้ชัดเจนขึ้น จะเป็นการทำงานเพื่อให้เห็นความแตกต่างของวัตถุและฉาก ซึ่งในการแบ่งแยกภาพนั้นจะสามารถทำได้แม้ภาพที่นำมาจะเป็นภาพเบลอ หรือ ภาพที่โฟกัสไม่ตรงภาพที่ขอบภาพไม่คมชัดก็ทำได้ โดยเราจะใช้วิธีการหาจุดเริ่มเปลี่ยน ซึ่งอาศัยหลักการเอนโทรปีของฮิสโทแกรมช่วยด้วยการนำภาพมาหาจุดเริ่มเปลี่ยนที่เหมาะสม โดยการนำเอาภาพนั้นมาผ่านการกรองแบบมัธยฐานเพื่อประมวลผลภาพ แล้วจึงนำมาทำฮิสโแกรม 2 มิติ ซึ่งจะมีลักษณะเป็นเมตริกที่มีขนาดเท่ากับช่วงของระดับเทา จึงค่อยทำการนำค่าเอนโทปีจากฮิสโทแกรม 2 มิติ และหาค่าของจุดเริ่มเปลี่ยนที่เหมาะสมในฮิสโทแกรม 2 มิติ หลังจากนั้นจะทำการสร้างภาพ 2 ระดับซึ่งจะได้ภาพที่แบ่งออกเป็นวัตถุและฉากเห็นได้ชัดเจน

2.13.3 **การทำ** Symmetry

ในการทำสมมาตรนั้น การกำหนดขอบเขตของภาพจะต้องขึ้นอยู่กับผู้ใช้โปรแกรมโดยผู้ใช้จต้องทราบว่าทิศทางของภาพว่าจะต้องการสมมาตรไปในทิศทางไหนซึ่งการทำงานต้องอาศัยขอบเขตที่กำหนดเป็นตัวหลัก กล่าวคือ ภายในขอบเขตที่กำหนดเป็นตัวหลักกล่าวคือ ภายในขอบเขตที่ถูกสร้างขึ้นโดยผู้ใช้นั้นภายในจะมีการกำหนดแกนหลักให้อยู่ตรงกึ่งกลางของขอบเขต จากนั้นจะทำการเปลี่ยนค่า Blt ที่พิกเซลที่อยู่ฝั่งตรงข้ามให้เหมือนกัน ซึ่งคล้ายการส่องกระจกโดยได้เขียนการทำสมมาตรไว้ 4 ทิศทาง คือ บน ล่าง ซ้าย และ ขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.4 **การกลับสี** (Invert Color)

เราจะพบได้บ่อยในฟิลม์ Negative ซึ่งจะกลับสีจากสว่างเป็นสีมืดแทน โดยปกติแต่ละสีจะมีค่าตั้งแต่ 0 ถึง 255 ถ้าเราเขียนการแปลงกลับสีในรูปสมการจะได้ ดังนี้

$$R_s = 256 - 1 - R_r$$
$$G_s = 256 - 1 - G_r \quad \text{.....(4.)}$$
$$B_s = 256 - 1 - B_r$$

โดยที่ s คือ เอาท์พุต และ r คือ อินพุต

จากสมการที่ 4 จะได้ความสัมพันธ์แสดงได้ดังรูปที่ 2.22

รูปที่ 2.22 การแปลงแบบ Negative

จากรูปที่ 2.22 ถ้าอินพุตเป็น 0 เอาท์พุต จะเป็น 255 และอินพุตเป็น 255 เอาท์พุตจะเป็น 0 แทน ประโยชน์ในการใช้การกลับสีนั้น จะเหมาะกับการดูบริเวณที่มืดๆ ของอิมเมจ ซึ่งเมื่อกลับสีแล้ว จะกลายเป็นบริเวณสว่างแทน

2.13.5 **การแก้ไขพิกเซลด้วย** Gamma

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับค่าพิกเซลของอิมเมจโดยการชดเชยด้วยแกมมา (Gamma) หรือเรียกว่า Gamma Correction Transformation ที่ชดเชยด้วยแกมมา ก็เพราะอุปกรณ์ที่เกี่ยวกับการแสดงผลอิมเมจ ตัวอย่างเช่น มอนิเตอร์ที่ประกอบด้วยชุดหลอดภาพ Cathode Ray Tube (CRT) ทำให้การทำงานแบบไม่ Linear (Non Linear) จะส่งผลให้ภาพอิมเมจแสดงบนจอมอนิเตอร์ไม่ลิเนียร์ไปด้วย เมื่อแก้ไขหรือปับค่าด้วยแกมมาแล้ว ภาพอิมเมจก็จะแสดงผลลิเนียร์ วิธีการแก้ไขก็คือ ใช้สมการแกมมา ดังนี้

$$R_s = R_r^{1/\gamma}$$
$$G_s = G_r^{1/\gamma} \qquad .....(5.)$$
$$B_s = B_r^{1/\gamma}$$

โดยที่ s คือ เอาท์พุต และ r คือ อินพุต

$0 < \gamma < 1$ จะเป็นการแปลงแบบเส้นโค้งแบบเอ็กโปเนนเชียล

$\gamma = 1$ จะเหมือนกับอินพุต (ไม่มีการแปลง)

$\gamma > 1$ จะเป็นการแปลงแบบเส้นโค้งลอการิทึม แสดงดังรูปที่ 2.23

รูปที่ 2.23 Gamma Correction Transformation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.23 จะเห็นว่า ถ้า $0 < \gamma < 1$ ภาพอิมเมจใหม่ จะมีความมืดมากกว่าอิมเมจต้นฉบับ ขณะที่ $\gamma > 1$ ภาพอิมเมจจะมีความสว่างมากกว่าปกติ

วิธีการแปลง เริ่มต้นด้วยการหาพอยน์เตอร์ชี้ไปยังพิกเซล ณ ตำแหน่งพิกัดที่ต้องการแปลงโดยใช้ ฟังก์ชัน GetPixelAddress จากนั้นจึงอ่านข้อมูลของแต่ละสี มาคำนวณค่าใหม่โดยใช้สมการ (5.)

ถ้าต้องการแปลงพิกเซลทั้งหมด เราก็สแกนทุกแถวทุกคอลัมน์ด้วยใช้คำสั่ง Loop for

2.13.6 การสร้างฮิสโทแกรม (Histogram)

ฮิสโทแกรมของอิมเมจ อธิบายถึงความมืด ความสว่าง ความคมชัด (Contrast) สูงหรือต่ำ อีกทั้งยังสามารถประยุกต์ใช้กับการทำ Equalized อิมเมจ ซึ่งเรียกว่า Histogram Equalized นอกจากนั้นแล้วยังสามารถนำมาประยุกต์ใช้กับงานด้านการบีบอัดอิมเมจ (Image Compression) และการแยกอิมเมจออกเป็นส่วนๆ (Image Segmentation) ด้วย

ฮิสโทแกรมของอิมเมจ ก็คือกราฟแท่งที่แสดงถึงความเข้มพิกเซลในแต่ละระดับ ซึ่งจะเริ่มต้นระดับ 0 จนถึง 255 ระดับของพิกเซลจะถูกวางอยู่บนแกน X ส่วนแท่งกราฟที่แสดงถึงความถี่ที่เกิดขึ้นซ้ำๆ กันในระดับนั้นๆ จะถูกแสดงบนแกน Y

รูปที่ 2.24 ภาพที่แสดงโดยกราฟHistogram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.7 **การทำ** Histogram Equalized

ในกรณีอิมเมจมีความคมชัดแย่ๆ อิมเมจนั้นสามารถนำมาปรับปรับคุณภาพใหม่ด้วยการทำ Histogram Equalized ซึ่งจะทำการกระจายความสว่างพิกเซลใหม่ ถ้าถ้าฮิสโทแกรมมีจุดยอดเป็นห้วงๆ (หลายจุดยอด) เมื่อทำด้วย Histogram Equalized แล้วจุดยอดห้วงๆ นั้น ก็ยังคงเหมือนเดิม แต่จะถูกเลื่อนกระจายตลอดระดับความสว่างของอิมเมจ

วิธีการทำ Histogram Equalized

โดยใช้สมการ $S_k = \sum_{j=0}^{k} n_j / n \; ; k=0,1,2,..,L-1$ ...(6.)

โดยที่

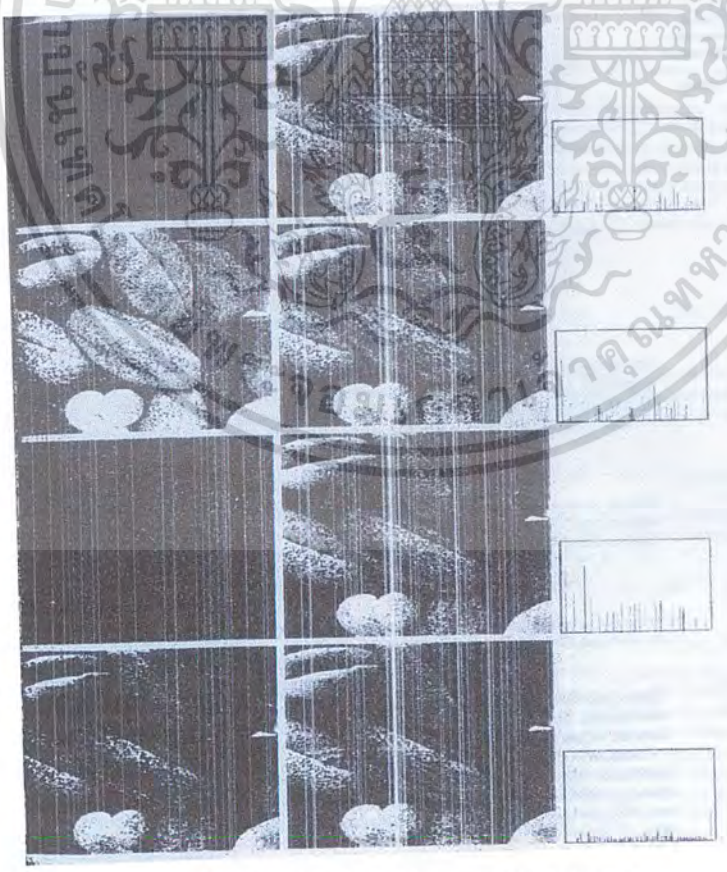
L คือ ระดับความสว่างสูงสุด

S คือ เอาท์พุต

n คือ จำนวนพิกเซลทั้งหมด

n_k คือ ผลรวมฮิสโทแกรม ณ ระดับ k

ในการใช้งานจริงๆ สมการที่ (6.) จะต้องมาแมบกับระดับความสว่างของพิกเซลอีกครั้งหนึ่ง จึงจะได้เอาท์พุตอิมเมจที่ถูกทำ H.E. สำหรับการคำนวณ H.E. ในที่นี้จะดัดแลงจากสมการที่ (6.) มาใช้



รูปที่ 2.25 การทำ Histogram Equalized

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.8 การบวกลบอิมเมจและการใช้ลอจิก And, Or และ Xor

การดำเนินการด้วยคณิตศาสตร์เหล่านี้ จะใช้ในการปรับค่าความสว่างของอิมเมจ การบวกด้วยค่าคงที่จะช่วยเพิ่มความสว่างแต่ไม่ได้เพิ่มความคมชัดของอิมเมจ การลบด้วยค่าคงที่จะช่วยเพิ่มความมืด ซึ่งจะเหมาะกับอิมเมจที่มีความสว่างมาก ในที่นี้การบวกจะเป็นการบวกระหว่างอิมเมจด้วยกัน ผลลัพธ์จะกลายเป็นอิมเมจใหม่ และ การลบก็จะเป็นการลบระหว่างอิมเมจ การบวกและการลบอิมเมจสามารถเขียนเป็นสมการง่ายๆ ดังนี้

$$s = r_a \pm r_b \quad \ldots (7.)$$

โดยที่ s คือ พิกเซลใหม่

r_a คือ พิกเซลอิมเมจแรก

r_b คือ พิกเซลอิมเมจที่สอง

การหาค่าเฉลี่ยอิมเมจนั้นจะนำแต่ละพิกเซลอิมเมจมาบวกกันแล้วหารด้วยสอง ซึ่งจะช่วยกำจัด Noise ของอิมเมจได้

การใช้ลอจิก And,Or และ Xor จะมีประโยชน์มากในการทำ Binary Image เช่น การสร้างอิมเมจจากหลายๆ อิมเมจประกอบกัน (Composite Image) โดยการใช้ลอจิก And,Or และ Xor ร่วมกัน การMorphology เป็นต้น ลอจิก And,Or และ Xor จะมีคุณสมบัติ ดังตารางที่ 2

ตารางที่ 2 Truth Table **ของลอจิก** And,Or **และ** Xor

Logic	AND	OR	XOR
อินพุต A	0 0 1 1	0 0 1 1	0 0 1 1
อินพุต B	0 1 0 1	0 1 0 1	0 1 0 1
เอาท์พุต	0 0 0 1	0 1 1 1	0 1 1 0

สมการที่ใช้ทำ Logic And,Or และ Xor แสดงได้ ดังนี้

โลจิก And : $S = r_a \& r_b$

โลจิก Or : $S = r_a | r_b$ .....(8.)

โลจิก Xor : $S = r_a \wedge r_b$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14 การทำกระบวนการรอบๆ จุดพิกเซล (Area Processing)

การทำกระบวนการนี้จะเอาค่าบริเวณรอบๆ พิกเซลมาคำนวณ แล้วนำมาแทนค่าพิกเซลเดิม บริเวณรอบๆ พิกเซลนี้เรียกได้หลายๆ แบบ ตัวอย่างเช่น Filter Window , Kernel หรือ Mask ส่วนใหญ่มักจะนิยมเรียก Kernel หรือ Mask การคำนวณบริเวณรอบๆ พิกเซล อาจเรียกว่า การทำ Convolution

การทำ Convolution คือ การหาผลรวมที่ได้ถ่วงน้ำหนักแล้วบริเวณรอบๆ จุดพิกเซล ค่าถ่วงน้ำหนักนี้เรียกว่า Mask Coefficient หรือ Kernel Coefficient โดยปกติจะกำหนดค่าถ่วงน้ำหนักเป็นเมตริกขนาด n x n โดย n จะเป็นเลขคี่ เช่น 3,5,7,9, ... เป็นต้น เพราะว่าสามารถหาค่าจุดพิกเซลตรงกลางได้ (ถ้าเป็น 1 จะเหมือนกับการทำกระบวนจุดพิกเซล) ในที่นี้จะใช้เมตริกซ์ขนาด 3 x 3 เป็นหลัก เราสามารถแสดงวิธีการคำนวณและสมการได้ดังรูป 2.26

รูปที่ 2.26 วิธีการคำนวณ Convolution

ค่าพิกเซลใหม่ f'(4) จะเท่ากับ $(c_0f_0 + c_1f_1 + c_2f_2 + c_3f_3 + c_4f_4 + c_5f_5 + c_6f_6 + c_7f_7 + c_8f_8)$

หรือ $f'(4) = \sum c_i f_i$ .....(9.)

สมการที่ (9.) เรียกว่า Linear filter ของอิมเมจ

ผลรวมค่าถ่วงน้ำหนัก (Mask Coefficient) แสดงได้ดังสมการ $\sum c_i$ .....(10.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (10.) ค่าผลรวมถ่วงน้ำหนักจะมีผลกระทบต่อระดับความสว่างตลอดทั้งอิมเมจ โดยทั่วไปแล้วผลรวมจะเป็นค่า 1 แต่ถ้าผลรวมเท่ากับหรือน้อยกว่า 0 อิมเมจจะมีความมืดมากขึ้น ซึ่งจะพบได้ในการตรวจหาขอบอิมเมจ หรือ Edge Detector

จากสมการที่ (9.) ถ้านำมาหารด้วยสมการที่ (10.) เราจะใช้สมการฟิลเตอร์ (Filter) ในรูปแบบทั่วๆ ไปดังนี้

$$f'(4) = \sum c_i f_i / \sum c_i \qquad \text{.....(11.)}$$

โดยที่สมการ (10.) ต้องไม่เท่ากับ 0 หรือ ถ้าสมการ (10.) = 0 จะเซตเป็น 1

ค่า Mask Coefficient สามารถกำหนดได้หลายรูปแบบ ในแต่ละรูปแบบก็จะมีผลตลอดทั้งอิมเมจ ทำให้ค่า Mask Coefficient มีชื่อเรียกที่แตกต่างกัน ได้แก่ ฟิลเตอร์ Low Pass ฟิลเตอร์ Low Pass ฟิลเตอร์ High Pass การตรวจหาขอบอิมเมจ (Edge Detector) ซึ่งมีได้หลายๆ วิธีด้วยกัน และการทำอิมเมจให้นูน (Emboss) ดังรายละเอียดต่อไปนี้

2.14.1 **ฟิลเตอร์** Low Pass

การปรับอิมเมจให้มัวขึ้น หรือ เบลอ โดยทั่วๆ ไปแล้วค่า Mask Coefficient จะเป็น

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/16 & 1/16 \end{bmatrix}$$

นอกจากนี้แล้ว ค่า Mask Coefficient ยังกำหนดเป็นเมตริกขนาด 5 x 5 หรือ 7x 7 และอื่นๆ ยิ่งขนาดเมตริกซ์ยิ่งใหญ่ ก็จะทำให้อิมเมจเบลอมากขึ้น

2.14.2 **ฟิลเตอร์** High Pass

การปรับอิมเมจให้คมชัดขึ้น (Sharped Image) ซึ่งตรงข้ามกับฟิลเตอร์ Low Pass ค่า Mask Coefficient มีได้หลายๆ รูปแบบ เช่น

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 9 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 15 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14.3 **การทำอิมเมจให้นูน** (Emboss)

การปรับอิมเมจให้นูนขึ้น (Emboss) โดยใช้ค่า Mask Coefficient ดังนี้

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} +127 \qquad \text{หรือ} \qquad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} +127$$

2.14.4 **การตรวจหาขอบอิมเมจ** (Edge Detector)

ขอบอิมเมจ ก็คือชุดของพิกเซลต่อๆ กันที่วางอยู่บนขอบระหว่างพื้นที่สองส่วนของอิมเมจ ขอบอิมเมจจะช่วยอธิบายถึงรูปร่าง ลักษณะ ขนาด และ อื่นๆ ของอิมเมจ การตรวจหาขอบอิมเมจเป็นก้าวแรกในการแยกอิมเมจเป็นส่วน (Image Segmentation) แล้วนำมาประกอบรวมกันเป็นออบเจ็กต์หรือพื้นที่ใหม่ (Region) ในที่นี้เราจะพิจารณาการตรวจหาขอบอิมเมจด้วยอนุพันธ์อันดับ 1 และ 2

อนุพันธ์อันดับหนึ่ง (First Order Derivative) ได้แก่ โอเปอร์เรเตอร์ Sobel ,Prewitt และ Frei Chen

ส่วนอนุพันธ์อันดับสอง (Second Order Derivative) ได้แก่ โอเปอร์เรเตอร์ Laplacian

2.14.4.1 **อนุพันธ์อันดับหนึ่ง** ถ้านำมาหาขอบอิมเมจในทิศทางแนวนอน และแนวตั้ง ก็จะหาอนุพันธ์สมการที่ 1 เราจะเรียกว่าการหา Gradient ซึ่งเป็นเวกเตอร์เมตริกซ์ แสดงได้ดังนี้

$$\nabla f = \begin{bmatrix} H_y(x,y) \\ H_c(x,y) \end{bmatrix} = \begin{bmatrix} \partial / \partial x\,(f(x,y)) \\ \partial / \partial y\,(f(x,y)) \end{bmatrix} \qquad \ldots(12.)$$

โดยที่ / x(f(x,y))จะเป็นการหาขอบอิมเมจในแนวนอน (x) และ / y(f(x,y))จะเป็นการหาขอบอิมเมจในทิศทางแนวตั้ง (y) ถ้าหาทั้งสองทิศทาง จะเป็นขนาดของเวกเตอร์ (Magnitude Vector) ของสมการ (12.) ซึ่งจะเขียนใหม่ได้ดังนี้

$$\nabla f = [(H_r)^2(x,y) + (H_c)^2(x,y)]^{1/2}$$

$$\approx |H_r(x,y)| + |H_c(x,y)| \qquad \ldots(13.)$$

ส่วนทิศทางในการตรวจหาขอบอิมเมจได้จากสมการ

$$\theta = \tan^{-1} \begin{bmatrix} \underline{Hc(x,y)} \\ H_r(x,y) \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14.4.2 **โอเปอเรเตอร์** Sobel จะใช้ค่า Mask Coefficient ดังนี้

ในทิศทางแนวนอน $H_r(x,y)$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

ในทิศทางแนวตั้ง $H_c(x,y)$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

2.14.4.3 **โอเปอเรเตอร์** Prewitt จะใช้ค่า Mask Coefficient ดังนี้

ในทิศทางแนวนอน $H_r(x,y)$

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

ในทิศทางแนวตั้ง $H_c(x,y)$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

2.14.4.4 **โอเปอเรเตอร์** Fri-Chen จะใช้ค่า Mask Coefficient ดังนี้

ในทิศทางแนวนอน $H_r(x,y)$

$$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทิศทางแนวตั้ง $H_c(x,y)$

$$\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & -1 \\ -1 & 0 & 1 \end{bmatrix}$$

2.14.4.5 **อนุพันธ์อันดับสอง** โดยการนำสมการที่ (12.) มาหาค่าอนุพันธ์อีกครั้ง จะได้สมการใหม่ ดังนี้

$$\nabla^2 f = \left| \begin{array}{l} \partial^2/\partial x\,(f(x,y)) \\ \partial^2/\partial y\,(f(x,y)) \end{array} \right|$$

อนุพันธ์อันดับสองจะมีโอเปอร์ Laplace ซึ่งมีค่า Mask Coefficient ดังนี้

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

สำหรับรายละเอียดอื่นๆ ในการหาขอบนั้นจะกล่าวเพิ่มเติมอีกต่อไปภายหลัง ถึงแนวคิดทฤษฎีต่างๆ

2.15 การทำกระบวนรูปร่างลักษณะ (Morphology Processing)

ในที่นี้จะอธิบายเฉพาะการทำกระบวนการรูปร่างลักษณะสำหรับสำหรับอิมเมจขาวดำหรือสีโดยส่วนใหญ่แล้ว จะใช้กับอิมเมจขาวดำมากกว่า ส่วนการทำกระบวนการอิมเมจดิจิตอลไบนารี(Binary Image) มีเฉพาะสีขากับดำเท่านั้น

การทำกระบวนการรูปร่างลักษณะจะคล้ายกับการทำ Convolution ซึ่งจะต้องใช้ Mask Coefficientแต่ในการทำนี้จะเรียก Mask Coefficient ใหม่ว่า Structure Element (S.E.) การทำกระบวนการรูปร่างลักษณะ มีได้หลายแบบด้วยกัน ตัวอย่างเช่น การขยายพิกเซล (Dilation) การลดขนาดพิกเซล (Erosion) การทำอิมเมจในพื้นที่ว่างให้เปิดมากขึ้น (Opening) และ การทำอิมเมจในพื้นที่ว่างให้ปิดมากขึ้น การขยายหรือลดพิกเซล การทำให้อิมเมจให้เปิดมากหรือการทำให้อิมเมจให้ปิดมาก จะขึ้นอยู่กับ S.E.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15.1 **การขยายพิกเซล** (Dilation)

การทำ Dilation การขยายพิกเซลที่สว่างให้ใหญ่ขึ้น สำหรับอิมเมจขาวดำ หรือ Gray Scale Morphology แสดงสมการได้ดังต่อไปนี้

ให้ A เป็นค่าพิกเซล ณ x,y ใดๆ และ H เป็น Structure Element ขนาด I*j ทั้ง A และ H เป็นจำนวนจริง หรือ เขียนใหม่ได้ว่า A และ H ε R

$$A \oplus H_{i,j} = \max_{i,j} \left\{ A(x-I,y-j) + H(I,j) \right\} \quad \ldots(16.)$$

2.15.2 **การลดขนาดพิกเซล** (Erosion)

วิธีการนี้จะตรงกันข้ามกับวิธีการ Dilation คือจะลดขนาดพิกเซลที่สว่างลง ฉะนั้นจากสมการที่ (16.) จะเขียนใหม่ได้ดังนี้

$$A \ominus H_{I,j} = \min_{i,j} \left\{ A(x-I,y-j) - H(I,j) \right\} \quad \ldots(17.)$$

2.15.3 **การทำอิมเมจในพื้นที่ว่างให้เปิดมากขึ้น** (Opening)

รูปแบบการทำนี้ คือ ทำ Erosion ก่อน จากนั้นจึงทำ Dilation

2.15.4 **การทำอิมเมจในพื้นที่ว่างให้เปิดมากขึ้น** (Closing)

รูปแบบการทำนี้ คือ ทำ Dilation ก่อน จากนั้นจึงทำ Erosion

2.16 การตัดสิ่งที่ไม่ต้องการออกจากภาพ(Removing Noise)

2.16.1 **วิธี** Smoothing Spatial Filter วิธีนี้ถูกใช้เพื่อการลดความไม่คมชัดและสิ่งที่ไม่ต้องการออกจากภาพ

แบบที่ 1 ใช้หน้ากากในอัตราส่วนที่หารด้วย 9

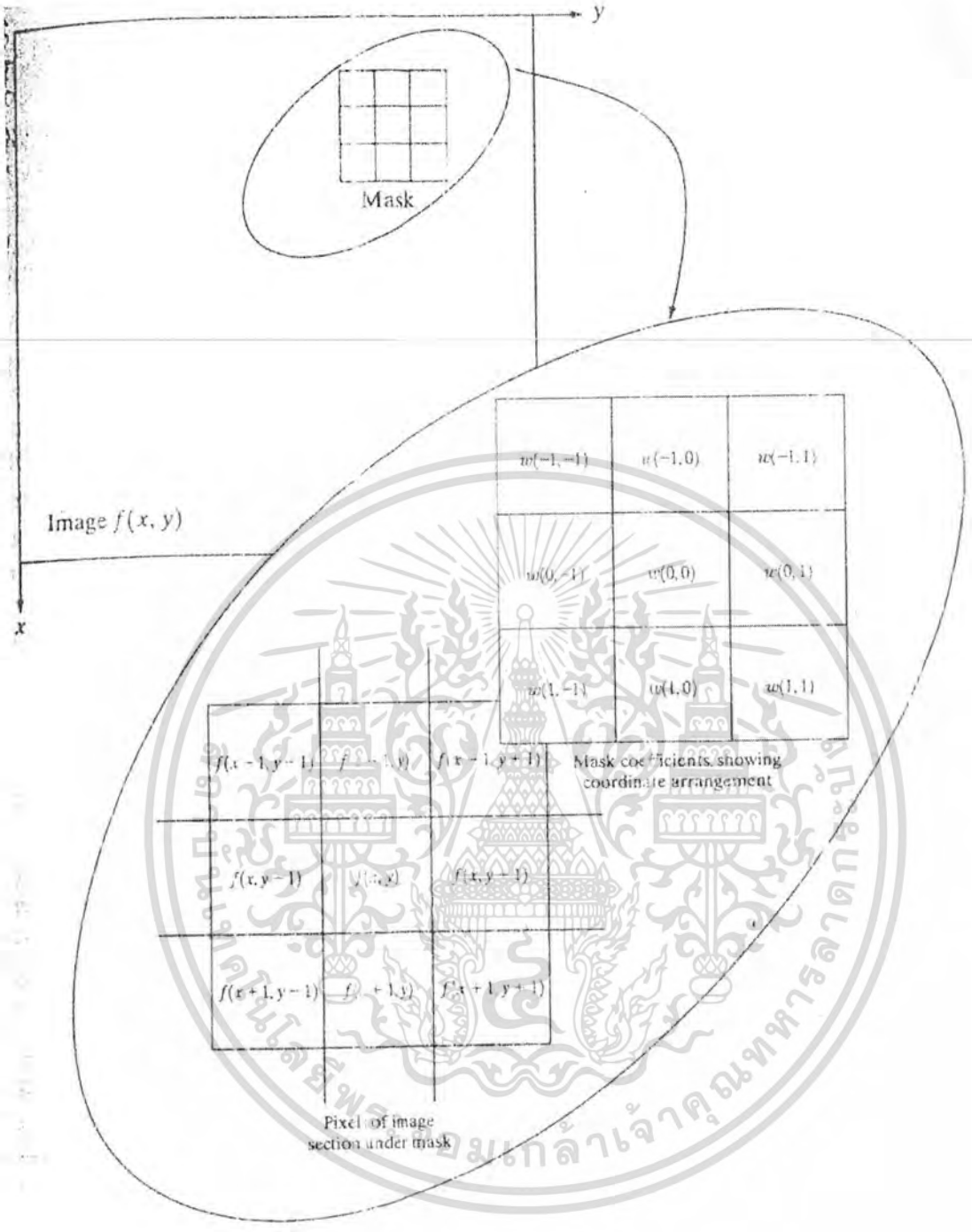
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

แบบที่ 2 ใช้หน้ากากในอัตราส่วนที่หารด้วย 16

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

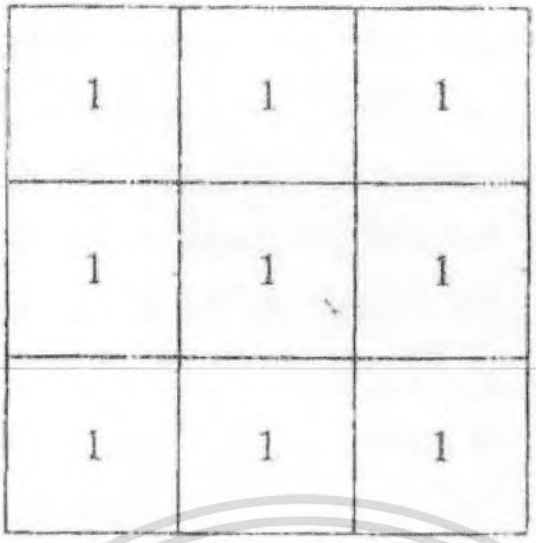
โดยผลลัพธ์ของวิธีนี้เป็นการเฉลี่ยค่าพิกเซล โดยฟิลเตอร์เหล่านี้อาจเรียกได้ว่าเป็นฟิลเตอร์ค่าเฉลี่ย (Averaging filters) แนวคิดของวิธีนี้จะตรงไปตรงมาโดยการแทนค่าของแต่ละพิกเซลในอิมเมจโดยค่าเฉลี่ยของระดับเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



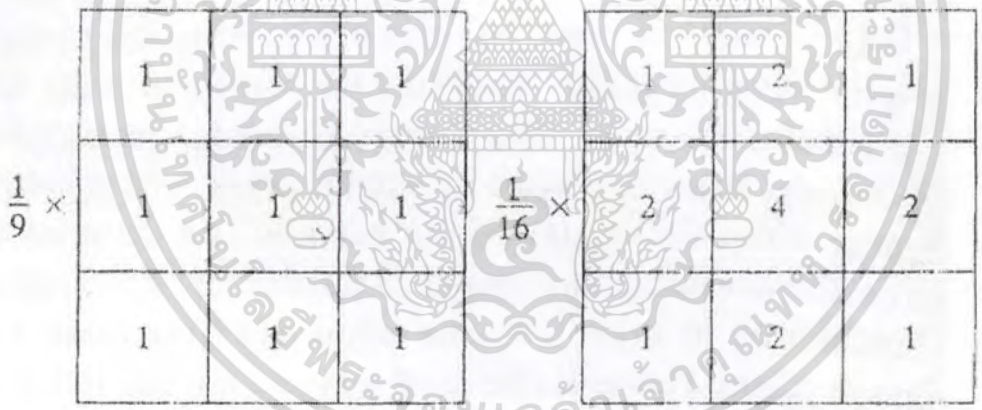
รูปที่ 2.27 หลักการของ Spatial Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 การแทนหน้ากาก Spatial Filter มิติ 3*3

จากรูป จะแสดงฟิลเตอร์ 2 มิติขนาด 3*3 ประโยชน์ของฟิลเตอร์แรกให้ค่าเฉลี่ยมาตรฐานของพิกเซล และ หน้ากากที่ 2 แสดงได้ดังรูป



รูปที่ 2.29 หน้ากากขนาด 3*3

โดยค่าคงที่ๆ เอาไปคูณด้านหน้าจะเท่ากับผลรวมของค่าสัมประสิทธิ์ทั้งหมดในหน้ากาก เพราะเป็นการหาค่าเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16.2 วิธี Order-Statistics Filters

วิธีนี้จะใช้ฟิลเตอร์แบบไม่เป็นเชิงเส้นโดยการลำดับของจุดพิกเซลในพื้นที่ฟิลเตอร์ที่กำหนด แล้วแทนค่าผลลัพธ์ของค่าพิกเซลกึ่งกลางที่เรียงลำดับแล้ว ตัวอย่างที่เป็นที่นิยมในวิธีนี้เรียกว่าการกรองแบบมัธยฐาน (Median filtering) วิธีนี้จะให้ผลลัพธ์ที่ดีมากในการลบสิ่งที่ไม่ต้องการออกจากภาพ โดยวิธีนี้จะเบลอน้อยกว่าการใช้ฟิลเตอร์เชิงเส้น (Smoothing filter) หลักการของวิธีนี้ขั้นแรกเราจะเรียงลำดับค่าจากน้อยไปมากหรือจากมากไปน้อยของพื้นที่พิกเซลที่เรากำหนด ตัวอย่างเช่น ถ้าพื้นที่ที่เรากำหนดมีขนาด 3*3 ค่ากึ่งกลางที่จะถูกดึงออกมาจะเป็นตำแหน่งที่ 5 แลถ้าพื้นที่ที่เรากำหนดมีขนาด 5*5

ตัวอย่างการทดลอง

ขั้นตอนที่ 1 ในการทดลองนี้ เราใช้ภาพมาทำการกรองแบบการกรองแบบมัธยฐาน ซึ่งเป็นการประมวลผลภาพแบบดิจิตอล โดยตัวดำเนินการแบบไม่เป็นเชิงเส้นตามที่กล่าวไว้ (Non-Linear Operate) และเป็น Low Pass Filter โดยอาศัยข้อมูลในตารางหน้าต่าง หรือ หน้ากาก (Mask) ในที่นี้เราจะใช้หน้ากากขนาด 3*3 กำหนดลงบริเวณข้อมูลของภาพ แล้วนำข้อมูลเหล่านั้นมาเรียงกันตามลำดับตามค่าของระดับความเข้มจากค่าต่ำสุดไปยังค่าสูงสุด จากนั้นเราก็จะทำการเลือกข้อมูลที่อยู่ตรงกลางของค่าทั้งหมดมา ในที่นี้เราต้องเลือกตำแหน่งที่ 5 จากรูปข้อมูลภาพ โดยการนำค่าภายในหน้ากากที่เรากำหนดมาเรียงจากน้อยไปมาก คือ 1,1,1,1,1,1,1,1,9 แล้วเลือกค่าลำดับที่ 5 มาใส่ที่ตำแหน่งกึ่งกลางของภาพผลลัพธ์ ทำเช่นนี้ไปจนตลอดข้อมูลภาพ โดยรูปที่ 2.6 เป็นข้อมูลที่ผ่านการกรองแล้ว โดยค่าที่กรองได้อาจเป็นค่าใหม่หรือค่าเดิมก็ได้ ถ้าหากมีข้อมูลตัวใดที่มีค่าแตกต่างไปจากค่าของจุดภาพใกล้เคียงกันมาก ก็จะถือว่าเป็นสัญญาณรบกวน (Noise) ที่ปนมากับข้อมูลภาพและจะถูกกำจัดออกไป

1	1	1	1	1
1	1	1	9	1
1	9	1	1	1
1	1	1	7	1
1	1	1	1	1
1	1	1	1	1

รูปที่ 2.30 ข้อมูลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

รูปที่ 2.31 ข้อมูลที่ผ่านการกรอง

2.16.3 **วิธี** Sharpening Spatial Filters

จุดประสงค์หลักของวิธีนี้เพื่อที่จะเน้นรายละเอียดที่ดีในอิมเมจ หรือเพิ่มรายละเอียดในส่วนที่เบลอ โดยวิธีนี้จะใช้หน้ากากไปใส่กับภาพต้นฉบับแล้วนำค่าทั้งหมดของทุกจุดพิกเซลหลังจากการแมบด้วยหน้ากากมาบวกกัน แล้วใส่ที่จุดกึ่งกลางของภาพผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

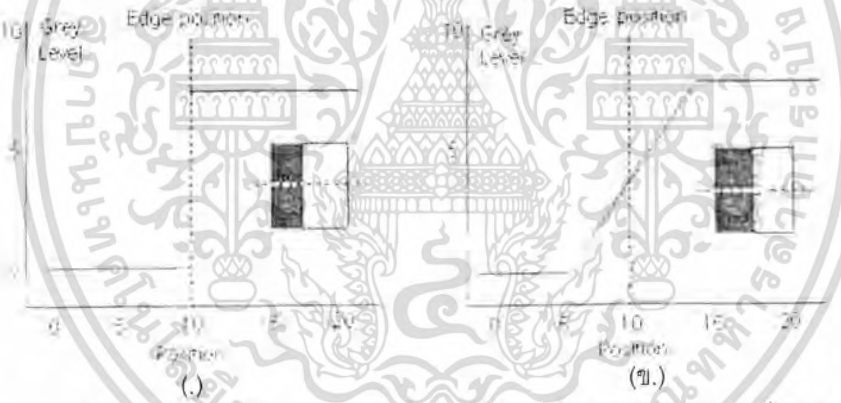
2.17 การหาขอบภาพ (Edge Detection)

จากอดีตถึงปัจจุบันได้มีผู้คิดค้นและพัฒนาอัลกอริทึมต่างๆ ที่ใช้ในการหาขอบเขตภาพมากมาย เนื้อหาส่วนนี้จะนำเสนอแนวคิดต่างๆ ในการหาขอบภาพ

2.17.1 ขอบภาพ

จากที่เคยได้กล่าวมาข้างต้นแล้ว ดังนั้น ขอบภาพ คือ การเปลี่ยนแปลงอย่างฉับพลัน ของความเข้มของค่าระดับเทา ดังนั้นคำจำกัดความของการหาขอบภาพ คือ การระบุถึงตำแหน่งของพิกเซลที่มีการเปลี่ยนแปลงของความเข้มดังกล่าว

ในการสร้างอัลกอริทึมที่ใช้ในการหาขอบภาพนั้น ต้องเริ่มจากการให้คำจำกัดความรูปแบบของขอบภาพก่อน แล้วจึงใช้คำจำกัดความนั้นในการหาอัลกอริทึมที่เหมาะสมต่อไป มีคำจัดความมากมายถูกใช้เป็นอย่างมาก แต่ที่นิยมใช้และง่ายส่วนมาก คือ ขอบภาพแบบขั้นบันไดอุดมคติ (Ideal Step Edge) จากรูป เป็นสัญญาณหนึ่งมิติที่มีการเปลี่ยนแปลงเกิดขึ้นในตำแหน่งแน่นอนเพียงตำแหน่งเดียว และ ถ้าการเปลี่ยนแปลงมีค่ามากจะทำให้ตำแหน่งที่เป็นขอบภาพสามารถถูกตรวจพบได้ง่าย



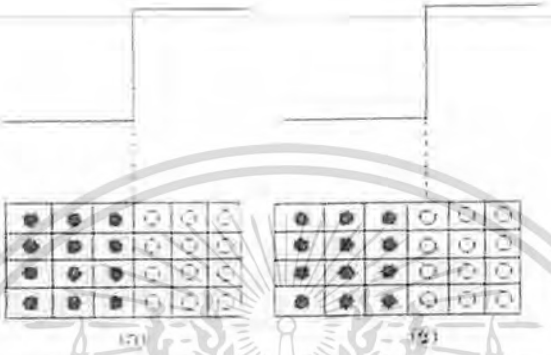
(ก.) (ข.)

รูปที่ 2.32 การเปลี่ยนแปลงค่าระดับเทา (ก) เปลี่ยนที่พิกเซลที่ 10 (ข) ตำแหน่งที่เกิดการเปลี่ยนแปลงขยายเป็น 10 พิกเซล

แต่โดยมากขอบภาพจะมีลักษณะที่เรียกว่าขอบภาพแบบลาด (Ramp Edge) ตามรูปที่ 2.32 ข. สาเหตุของการที่ขอบภาพมีลักษณะที่ค่อยๆ เปลี่ยนแปลงนี้เนื่องจากสาเหตุที่สำคัญ 2 ประการ ประการแรกเกิดจากการสุ่มตัวอย่างที่ตำแหน่งของพิกเซลในภาพของอุปกรณ์ดิจิตอลไม่พอดีกับตำแหน่งที่เป็นขอบภาพ ตามที่แสดงในรูปที่ 2.32 จะเห็นว่าเดิมขอบภาพจะปรากฏที่ด้านซ้ายสุดของพิกเซลที่มีสีขาว ถ้าเลื่อนกล้องไปทางซ้ายเป็นระยะทางครึ่งพิกเซล ตำแหน่งที่เป็นขอบภาพที่เห็นจากกล้องก็จะเลื่อนมาทางขวามือเป็นระยะทางครึ่งพิกเซลเช่นกัน ผลที่ได้ก็คือค่าความเข้มของพิกเซลในตำแหน่งที่เป็นขอบภาพจะเป็นค่าเฉลี่ยระหว่างพิกเซลที่เป็นสีขาวและสีดำ ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งที่เป็นขอบภาพมีลักษณะเบลอ และการเปลี่ยนแปลงค่าความเข้มจะเกิดขึ้นในช่วงจำกัดค่าหนึ่ง ซึ่งเราเรียกขอบภาพแบบนี้ว่า ขอบภาพแบบลาด ประการที่สอง เกิดจากสัญญาณรบกวน ซึ่งอาจเกิดได้จากสาเหตุหลายประการ เช่น ความเข้มของแสง ชนิดของกล้องและเลนส์ ความเคลื่อนไหวของวัตถุในภาพ อุณหภูมิ ฝุ่น และอื่นๆ



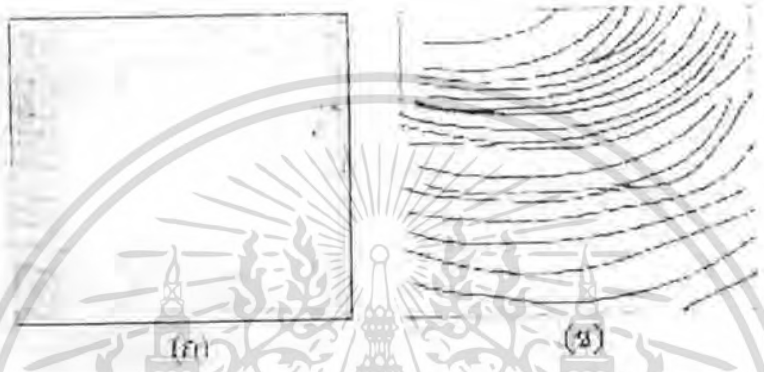
รูปที่ 2.33 (ก) ขอบภาพแบบขั้นบันได (ข) ขอบภาพแบบลาด

ตัวอย่างการหาขอบภาพแสดงในรูปที่ 2.34 โดยวัตถุ 2 ชิ้นซ้อนทับกันอยู่ โดยภาพ ก. เป็นภาพต้นฉบับส่วนภาพ ข. เป็นตำแหน่งที่เป็นขอบภาพจากภาพ ก. จากภาพ ข. จะแสดงให้เห็นถึงรูปร่างของวัตถุในภาพ สังเกตว่าในขณะนี้เรายังไม่สามารถบอกได้ว่า ส่วนไหนของภาพเป็นพื้นหลัง และส่วนไหนเป็นวัตถุ บอกได้แต่เพียงขอบเขตระหว่างวัตถุกับวัตถุหรือวัตถุกับพื้นหลัง อย่างไรก็ตามถ้าเราทราบว่าภาพต้นฉบับเป็นภาพเม็ดเลือด (คือ ทราบว่าวัตถุที่กำลังสนใจมีลักษณะกลม) เราจะสามารถบอกถึงข้อมูลต่างๆ ในภาพได้มากขึ้น ตัวอย่างเช่น สามารถบอกได้ว่า ส่วนใดของภาพเป็นเม็ดเลือด และ เม็ดเลือดหมายเลข 3 ทับหมายเลข 2 อยู่ และอยู่ใกล้มากกว่า

รูปที่ 2.34 ตัวอย่างการหาขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบภาพเป็นขั้นตอนหนึ่งของกระบวนการที่เรียกว่า เซกเมนเตชัน (Segmentation) ซึ่งจะทำการแบ่ง และ ระบุพื้นที่แต่ละพื้นที่ที่มีลักษณะหรือคุณสมบัติต่างๆ กัน การทำกระบวนการนี้ พื้นที่ต่างๆ ในภาพซึ่งอาจเป็นวัตถุต่างๆ จะถูกแยกจากกันด้วยขอบภาพและจะต้องถูกทำขั้นตอนอื่นต่อไปเพื่อจะระบุว่าแต่ละพื้นที่ในภาพเป็นวัตถุอะไร แต่ในบางครั้งการหาขอบภาพเพียงขั้นตอนเดียวก็เพียงพอต่อการประยุกต์ใช้งานสำหรับงานบางอย่าง โดยเฉพาะเมื่อวัตถุที่เราสนใจในภาพเป็นเส้นดังจะแสดงให้เห็นในตัวอย่างต่อไป



รูปที่ 2.35 (ก) ภาพตัดขวางของต้นไม้ (ข) ขอบภาพจากรูป ก.

พิจารณารูป ก. ซึ่งเป็นรูปภาพตัดขวางของต้นไม้ ในภาพเนื้อไม้จะประกอบไปด้วยวงปี ซึ่งเป็นส่วนที่เราสนใจในภาพ (ต้องการทราบจำนวนวงปี) จะเห็นได้ว่าเมื่อต้องการทราบอายุของต้นไม้ สิ่งที่เราต้องทำมีเพียงหาขอบภาพและนับจำนวนเส้นของวงปีเท่านั้น

2.17.2 **ตัวดำเนินการอนุพันธ์** (Derivative Operators)

จะกล่าวถึงตัวดำเนินการอนุพันธ์ซึ่งจะให้ผลตอบสนองเป็นการหาค่าเกรเดียนของค่าระดับเทาที่แต่ละพิกเซลในภาพ ซึ่งจะมีค่ามากในบริเวณขอบภาพ (บริเวณที่ค่าระดับเทามีการเปลี่ยนแปลง) และมีค่าที่น้อยในบริเวณที่ค่าระดับเทามีค่าคงที่ ดังนั้น ถ้าให้ภาพเป็นฟังก์ชันของตัวแปร 2 ตัว A(x,y) ค่าเกรเดียนของภาพที่ตำแหน่ง x,y ใดๆจะมีค่าตามสมการที่ 18.

$$\nabla A\,(x,y) = \left(\frac{\partial A}{\partial x}\,,\,\frac{\partial A}{\partial y}\right) \qquad \text{.....(18.)}$$

เนื่องจากภาพเป็นสัญญาณแบบไม่ต่อเนื่องจึงต้องเปลี่ยนการหาอนุพันธ์ย่อยเป็นการหาความแตกต่างของค่าระดับเทา โดยการหาค่าความแตกต่างของค่าระดับเทาของภาพ ∇_1 ในทิศทาง x,y ใดๆ จะสามารถแสดงได้ตามสมการที่ 19. และ 20. ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\nabla_{x1}A\,(x,y) = A\,(x,y) - A\,(x\text{-}1,y) \quad \text{.....(19.)}$$

$$\nabla_{y1}A\,(x,y) = A\,(x,y) - A\,(x\text{-}1,y) \quad \text{.....(20.)}$$

โดยในการประมาณค่าเกรเดียนแบบนี้ใช้สมมติฐานว่าการเปลี่ยนแปลงของค่าระดับเทาระหว่างพิกเซลเป็นแบบเชิงเส้น ดังนั้นไม่ว่าจะหาอนุพันธ์ที่ตำแหน่งใดก็จะได้ค่าความชันเท่ากัน แต่ก็เกิดปัญหาขึ้นคือการประมาณค่าอนุพันธ์ด้วยสมการที่ 19. และ 20. จะไม่ได้เป็นการประมาณค่าเกรเดียนที่ตำแหน่ง x,y แต่จะเป็นการประมาณค่าที่ตำแหน่ง x-1/2 , y-1/2 ดังนั้นตำแหน่งของขอบภาพที่ได้จะเลื่อนไปในทิศทาง –x และ –y เป็นระยะทางครึ่งพิกเซล จึงได้มีการปรับปรุงโดยใช้ตัวดำเนินการ ∇_2 แทนโดนจะมีค่า ดังนี้

$$\nabla_{y2}A\,(x,y) = A\,(x\text{+}1,y) - A\,(x\text{-}1,y) \quad \text{.....(21.)}$$

$$\nabla_{y2}A\,(x,y) = A\,(x,y\text{+}1) - A\,(x,y\text{-}1) \quad \text{.....(22.)}$$

จะเห็นว่าตัวดำเนินการ ∇_2 นี้จะสมมาตรที่ตำแหน่ง x,y และเป็นการประมาณค่าเกรเดียนที่ตำแหน่ง x,y ถึงแม้ว่าจะไม่ได้พิจารณาค่าระดับเทาที่ตำแหน่งดังกล่าวเลยก็ตาม

เมื่อใช้ตัวดำเนินการ ∇_2 ในการคำนวณค่าเกรเดียน ผลที่ได้จะเป็นเวกเตอร์ในทิศทางแกน x และ y คือ ∇_x ,∇_2 ซึ่งจะบอกถึงข้อมูลของของความแข็งแรง และทิศทางของขอบภาพที่ตำแหน่งนั้น โดยขนาดและทิศทางของเวกเตอร์รวมของ ∇_x ,∇_2 จะเป็นค่าความแข็งแรงของขอบภาพ หรือ ผลตอบสนองของขอบภาพ (Edge Response) และทิศทางของขอบภาพตามลำดับซึ่งสามารถแสดงได้ดังสมการ 23. และ 24.

$$G_{mag} = \sqrt{(\partial A/\partial x)^2 + (\partial A/\partial y)^2} \quad \text{.....(23.)}$$

$$G_{dir} = a\,tan(\partial A/\partial y/\partial A/\partial x) \quad \text{.....(24.)}$$

ในการหาขอบภาพโดยใช้ตัวดำเนินการอนุพันธ์นี้ พิกเซลที่ตำแหน่งใดมีค่าขนาดของเกรเดียนมากกว่าค่าเทรชโฮลด์ที่กำหนดไว้ ก็จะกำหนดให้เป็นพิกเซลของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.18 แบบจำลองของขอบภาพ

การหาขอบภาพแบบนี้ทำโดยใช้แบบจำลอง (Template) ขนาดเล็กของขอบภาพในทิศทางต่างๆ เป็นหน้ากากคอนโวลูชันเพื่อนำมาคอนโวลูชันกับภาพต้นฉบับ ปทนที่การใช้ตัวดำเนินการอนุพันธ์เพื่อหาเกรเดียนที่ตำแหน่งต่างๆ โดยตรงเหมือนในหัวข้อที่แล้ว อย่างไรก็ตามเป้าหมายหลัก ที่ใช้แบบจำลองของขอบภาพก็ยังเป็นค่าประมาณเกรเดียนนั่นเอง ในหัวข้อนี้จะขอยกตัวอย่าง แบบจำลอง 2 แบบ ซึ่งได้รับการยอมรับว่าให้ผลดี และ มีขนาดเล็กมาแสดง

แบบแรก คือ Sobel Edge Detector ใช้หน้ากากคอนโวลูชันขนาด 3*3 พิกเซล 2 ตัว ในการหาเกรเดียนในทิศทางแกน x และ y โดยมีค่าดังต่อไปนี้

$$S_x = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \qquad S_y = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

สังเกตได้ว่าผลลัพธ์ที่ได้จาการคอนโวลูชันจะเป็นค่าเกรเดียนนั่นเอง โดยค่าน้ำหนักในทิศทางทแยงมุมจะมีค่าน้อยกว่าค่าน้ำหนักในทิศทางแกนตั้งและแกนนอน ส่วนประกอบในทางแกน X คือ S_x และส่วนประกอบในทางแกน Y คือ S_y จะนำไปใช้ในการหาขนาดและทิศทางของเกรเดียน สามารถหาได้จากสมการ 23. และ 24.

หลังจากคอนโวลูชันแล้วทุกพิกเซลจะมีขนาดของเกรเดียน ถ้าพิกเซลใดมีค่ามากกว่าเทรชโฮลด์ ก็จะให้เป็นพิกเซลของขอบภาพ วิธีที่ดีที่สุดในการหาขนาดของเกรเดียน สามารถทำตามสมการที่ 3.6 แต่จากสมการจะเห็นว่า ต้องเกี่ยวของกับการถอดรากที่สอง และ ผลที่ได้ยังเป็นเลขทศนิยมอีกด้วย เพื่อลดความยุ่งยากในการคำนวณเราสามารถเลือกวิธีหาขนาดของเกรเดียนอีกวิธีหนึ่งได้ โดยให้เท่ากับผลรวมของค่าสัมบูรณ์ของ S_x และ S_y หรือค่าที่มากที่สุดจากสองค่า S_x และ S_yก็ได้

ตัวอย่างอย่างที่ 2 ทำโดย Kirsch หน้ากากคอนโวลูชันจะมีขนาด 3*3 เช่นเดียวกับ Sobel แต่ทำในทิศทางมากกว่า โดยหน้ากากคอนโวลูชันทั้ง 8 ตัวมีค่าดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K0 = -3 -3 5
-3 0 5
-3 -3 5

K1 = -3 5 5
-3 0 5
-3 -3 -3

K2 = 5 5 5
-3 0 -3
-3 -3 -3

K3 = 5 5 -3
5 0 -3
-3 -3 -3

K4 = 5 -3 -3
5 0 -3
5 -3 -3

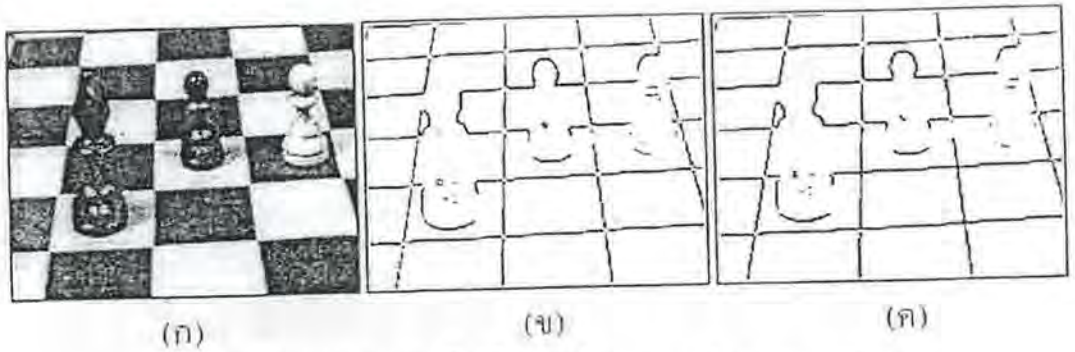
K5 = -3 -3 -3
5 0 -3
5 5 -3

K6 = -3 -3 -3
-3 0 -3
5 5 5

K7 = -3 -3 -3
-3 0 5
-3 5 5

หน้ากากคอนโวลูชันชนิดนี้พยายามที่จะจำลองชนิดของการเปลี่ยนแปลงของค่าระดับเทาให้มีหลายทิศทางเหมือนกับของภาพมากที่สุด จะเห็นว่าหน้ากากคอนโวลูชันหนึ่งตัวถูกใช้สำหรับแทนทิศทาง 1 ใน 8 ของวงกลม (45 องศา) ตัวอย่างเช่น ถ้าผลตอบสนองจาก K0 มีค่ามากจะสามารถบอกได้ว่ามีขอบในแนวตั้ง (เกรเดียนในแนวนอน) ที่ตำแหน่งพิกเซลกลางของแบบจำลองในการหาขอบภาพ ภาพต้นฉบับต้องคอนโวลูชั่นกับหน้ากากทุกตัว ขนาดของผลตอบสนองที่ตำแหน่งใดๆ ให้มีค่าเท่ากับผลตอบสนองที่มีค่ามากที่สุดของแบบจำลองทั้ง 8 และ ทิศทางขอบภาพจะมีค่าเท่ากับ ¶/4*t โดย t เป็นเลขที่ของแบบจำลองที่ให้ผลตอบสนองมากที่สุด

ตัวอย่างของการหาขอบภาพโดยใช้แบบจำลองของ Sobel และ Kirsch โดย ก. เป็นภาพต้นฉบับ ข. ภาพขอบที่ได้จากวิธี Sobel ค. ภาพขอบที่ได้จากวิธี Kirsch

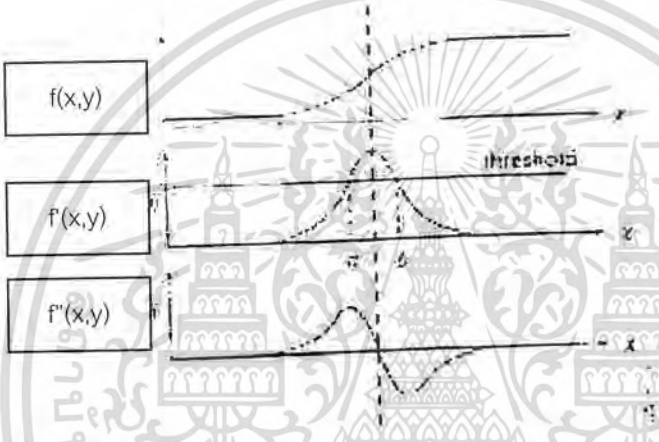


(ก) (ข) (ค)

รูปที่ 2.36 การหาขอบภาพโดยใช้แบบจำลองของ Sobel และ Kirsch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าการหาขอบภาพที่ได้จากค่าเกรเดียนในรูปแบบต่างๆ ที่ได้กล่าวไปแล้ว จะหาโดย การใช้ค่าเทรชโฮลด์ค่าหนึ่งเป็นตัวตัดสินว่าจะเป็นตำแหน่งที่พิจารณาเป็นขอบภาพหรือไม่ โดยถ้าตำแหน่งดังกล่าวมีค่าเกรเดียนสูงกว่าเทรชโฮลด์ก็จะกำหนดให้เป็นขอบภาพ ทำให้ผลที่ได้ออกมามีตำแหน่งที่เป็นขอบภาพอยู่จำนวนมาก (คือขอบภาพมีลักษณะเป็นเส้นหนา) ซึ่งจะไม่เป็นผลดี มีวิธีการที่จะแก้ไขการที่มีขอบภาพจำนวนมากได้ก็ คือ การให้เฉพาะจุดที่มีค่าเกรเดียนสูงสุดในบริเวณนั้นเป็นขอบภาพ ซึ่งในรูปหมายความว่าตำแหน่งที่เป็นขอบภาพจะต้องมีค่าอนุพันธ์อันดับ 1 ที่มากที่สุดหรือจะต้องเป็นตำแหน่งที่เกิดซีโร่ ครอสซิ่ง (Zero Crossing) ของอนุพันธ์อันดับที่ 2 ของค่าความเข้มของค่าระดับเทาในภาพ



รูปที่ 2.37 ตำแหน่งของขอบภาพและการหาขอบภาพจากอนุพันธ์ที่1 และ 2

พิจารณาตัวดำเนินการลาปาเซี่ยนที่เมื่อคอนโวลูชั่นกับสัญญาณใดๆจะให้ผลตอบสนองเท่ากับการหาอนุพันธ์อันดับ 2 ของสัญญาณนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19 อัลกอริทึมของ Marr และ Hildreth

เนื่องจากตำแหน่งขอบภาพที่หาได้จากตำแหน่งที่เกิด Zero Crossing โดยใช้ตัวดำเนินการลาปาเซี่ยนในการหาอนุพันธ์อันดับ2 ของค่าความเข้มของค่าระดับเทาของภาพ จะมีขอบภาพปลอมเกิดขึ้นเป็นจำนวนมากเนื่องจากสัญญาณรบกวนในภาพ ดังนั้นจึงต้องมีการกำจัดสัญญาณรบกวนในภาพออกไปเสียก่อน การลดผลกระทบจาก สัญญาณรบกวน ดังกล่าวทำได้โดยใช้โลว์พาส ฟิลเตอร์(Low pass Filter)คอนโวลูชันกับภาพต้นฉบับ

Marr และ Hildreth ได้สร้างหน้ากากคอนโวลูชันแบบใหม่ขึ้นมาซึ่งประกอบด้วยตัวดำเนินการลาปาเซี่ยนทำหน้าที่เน้น (Enhance) ขอบภาพ และ เกาส์เซี่ยนฟิลเตอร์ ซึ่งทำหน้าที่ในการลดสัญญาณรบกวน (Smoothing) เรียกว่า ลาปาเซี่ยน ออฟ เกาส์เซี่ยนฟิลเตอร์ (Laplacian of Gaussian Filter)

การหาขอบภาพวิธีการนี้ภาพต้นฉบับจะต้องทำการคอนโวลูชั่นกับเกาส์เซี่ยนฟิลเตอร์เพื่อกำจัดสัญญาณรบกวนออกไป หลังจากนั้นจึงนำไปหาอนุพันธ์อันดับ 2 โดยการคอนโวลูชั่นกับตัวดำเนินการลาปาเซี่ยน แล้วหาตำแหน่งที่เป็นขอบภาพจากตำแหน่งที่เกิด Zero Crossing แต่ Marr ได้สรุปว่าในภาพโดยทั่วไปนั้น ส่วนที่สนใจ(Feature of interest)จะเกิดขึ้นในหลายสเกล(Scale) และไม่มีตัวดำเนินการใดที่สามารถให้ผลดีในทุกสเกลได้ ดังนั้น อัลกอริทึมของ Marr จะต้องใช้ Smoothing filter หลายสเกล โดยใช้เกาส์เซี่ยนฟิลเตอร์ 2 ตัวที่ค่าความเบี่ยงเบนมาตรฐานต่างกัน (ใช้ความเบี่ยงเบนมาตรฐานของเกาส์เซี่ยนฟิลเตอร์จะเป็นตัวปรับสเกล)แล้วนำผลที่ได้(ตำแหน่งของขอบภาพ)มาพิจารณาประกอบกัน โดยตำแหน่งใดที่ผลตอบสนองของตัวดำเนินการลาปาเซี่ยนเกิดซีโรครอสซิ่งทั้ง 2 สเกล จึงจะให้ตำแหน่งนั้นเป็นขอบภาพที่แท้จริง ตัวดำเนินการเกาส์เซี่ยนฟิลเตอร์สองมิติจะมีค่าตามสมการ 25.

$$G_n(x, y) = \sigma^2 e^{((-(x\hat{}2+y\hat{}2))/\sigma\hat{}2)} \quad \ldots(25.)$$

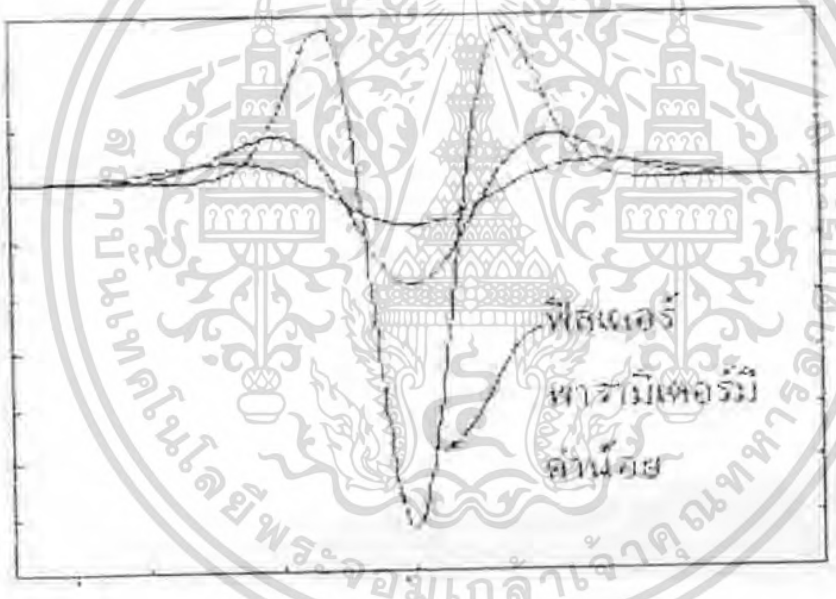
อย่างไรก็ตามในกรณีนี้ลำดับของตัวดำเนินการลาปาสเซี่ยน และ เกาส์เซี่ยนฟิลเตอร์จะไม่มีผล ดังนั้นเพื่อความสะดวกสามารถรวมตัวดำเนินการลาปาสเซี่ยน เข้ากับเกาส์เซี่ยนฟิลเตอร์เป็นฟิลเตอร์ตัวเดียวเรียกว่า ลาปาสเซี่ยน ออฟ เกาส์เซี่ยน ฟิลเตอร์ได้จะมีค่าตามสมการ 26.

$$\nabla^2 G_n = \left(\frac{r^2 - 2\sigma^2}{\sigma^4} \right) e^{(-r\hat{}2/2\sigma\hat{}2)} \quad \ldots(26.)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

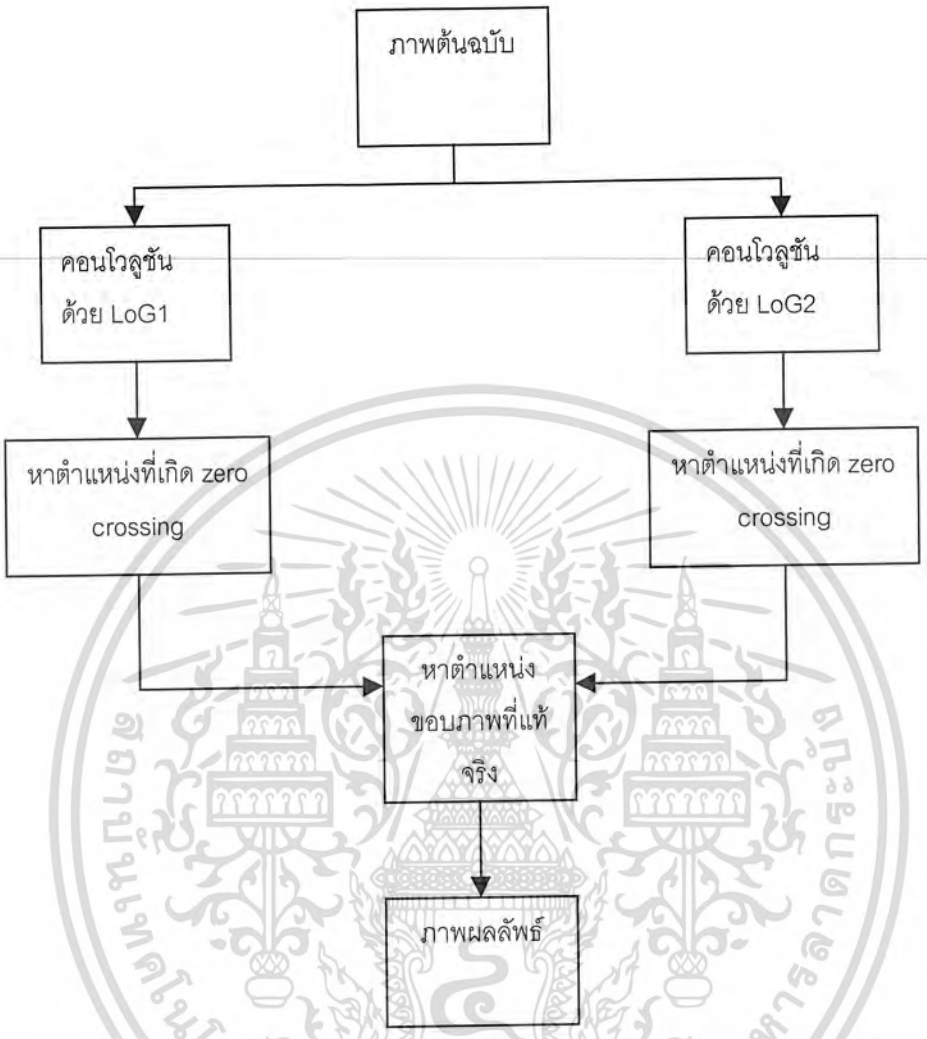
โดยที่ $r = (x^2+y^2)$ ดังนั้นการหาขอบภาพวิธีการนี้สามารถทำได้โดยใช้ลาปาเซี่ยน ออฟ เกาส์เซี่ยน ฟิลเตอร์ ไปคอนโวลูชันกับภาพแล้วหาตำแหน่งที่เกิด Zero Crossing จากนั้นเปลี่ยนสเกลที่พิจารณาโดนเปลี่ยนค่าความเบี่ยงเบนมาตรฐาน σ ในสมการที่ 26. แล้วทำขบวนการซ้ำเดิมอีกครั้ง ตำแหน่งที่เป็นขอบภาพจะได้จากตำแหน่งที่เกิด Zero Crossing ทั้ง 2 สเกล

รูป 2.38 แสดงลักษณะของลาปาเซี่ยน ออฟ เกาส์เซี่ยน ในโดเมนของระยะทางเมื่อเปลี่ยนค่าความเบี่ยงเบนมาตรฐานให้มีค่าต่างๆ กัน จะเห็นว่าถ้าความเบี่ยงเบนมาตรฐานมีค่าน้อย จะทำให้ขนาด (ความกว้าง) ของฟิลเตอร์มีค่าน้อยทำให้ระดับของการสมูทมีค่าน้อย และถ้าให้ความเบี่ยงเบนมาตรฐานมีค่ามาก จะทำให้ขนาด (ความกว้าง) ของฟิลเตอร์มีค่ามากตามไปด้วยทำให้ระดับของการสมูทมีค่ามาก และ ตำแหน่งของขอบภาพที่ได้ก็จะมีจำนวนน้อย รูปที่ 2.39 แสดงผังการทำงาน

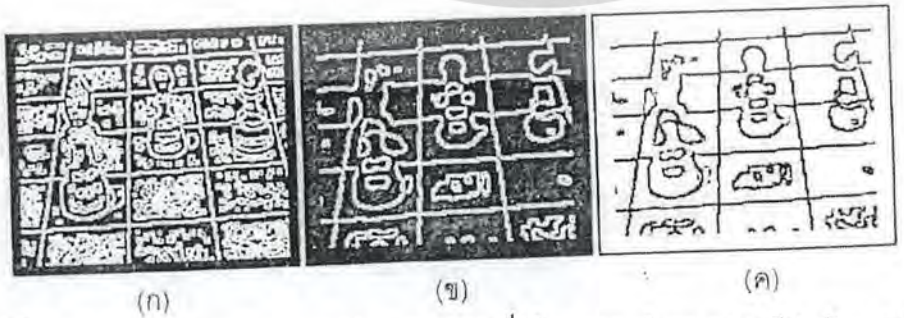


รูปที่ 2.38 ลักษณะของลาปาเซี่ยน ออฟ เกาส์เซี่ยน ในโดเมนของระยะทางเมื่อใช้ฟิลเตอร์พารามิเตอร์ค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.39 ผังการทำงานของการหาขอบภาพแบบ Marr



รูปที่ 2.40 การหาขอบภาพแบบ Marr (ก) ขอบที่ได้จากการกำหนด σ มีค่าน้อย (ข) ขอบที่ได้จากการกำหนด σ มีค่ามาก (ค) ขอบภาพเมื่อพิจารณาทั้ง 2 สเกลประกอบกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.20 อัลกอริทึมของ Canny

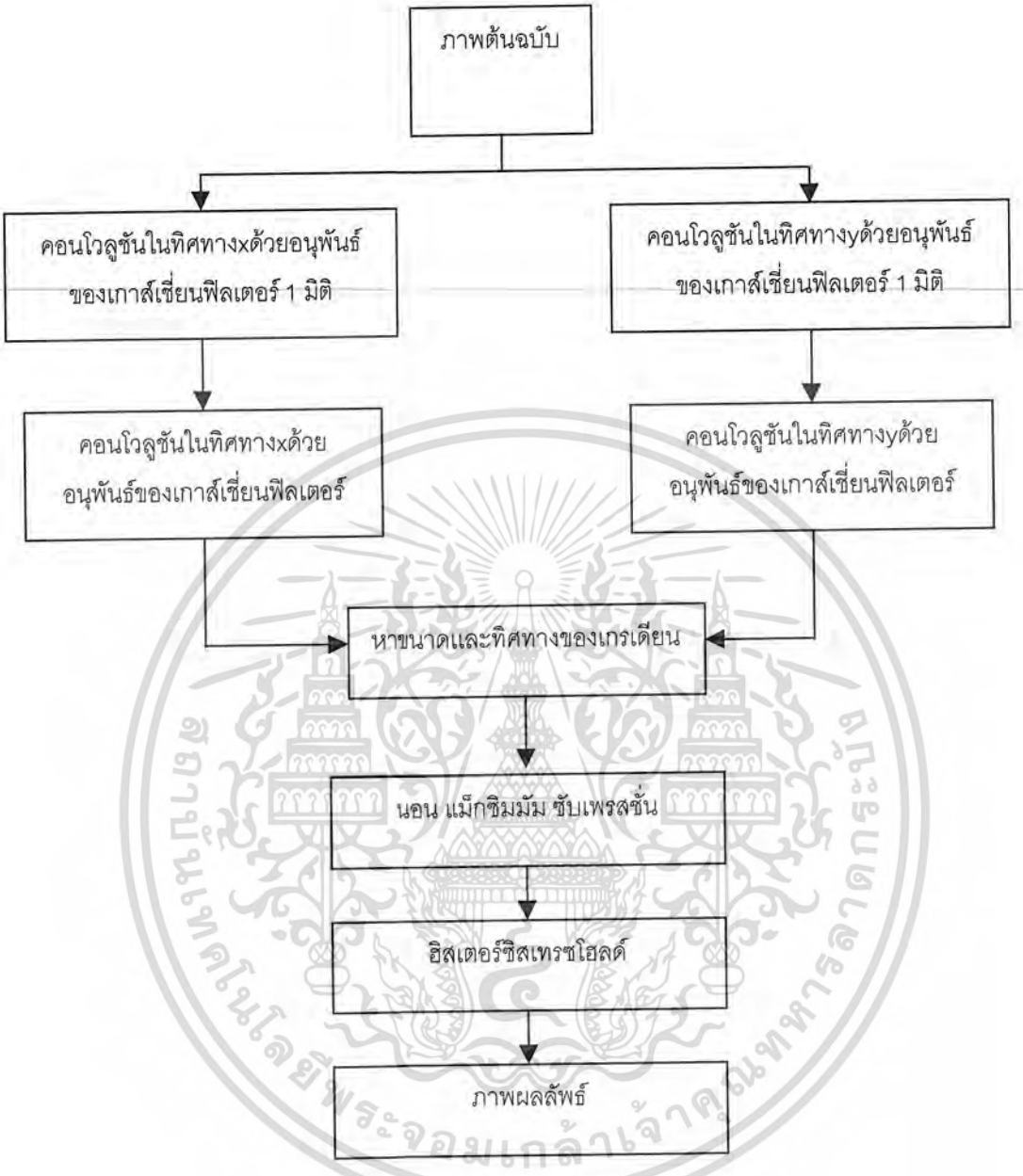
วิธีการในการหาขอบภาพของ Canny ทำโดยใช้หน้ากากคอนโวลูชั่นที่เป็นอนุพันธ์อันดับ 1 ของเกาส์เซี่ยน ฟิลเตอร์ ซึ่งมีค่าตามสมการ 27.

$$G'(X) = \left(-\frac{X}{\sigma^2} \right) e^{-((x^2)/2\sigma^2)} \quad .....(27.)$$

อัลกอริทึมของ Canny จะทำการคอนโวลูชัน 1 มิติโดยเริ่มจากการคอนโวลูชั่นรหว่างภาพต้นฉบับกับเกาส์เซี่ยนฟิลเตอร์ หลังจากนั้นคอนโวลูชั่นกับอนุพันธ์อันดับ 1 ของเกาส์เซี่ยนฟิลเตอร์อีกครั้ง โดยในการทำคอนโวลูชันจะแยกทำในแกน x และ y อย่างละครั้ง ผลที่ได้จะเป็นส่วนประกอบทางแกน x (I'_x) และ x (I'_y) ตามลำดับ ส่วนประกอบทั้ง 2 นี้ จะนำไปคำนวณหา ขนาดและทิศทางของเกรเดียนที่แต่ละพิกเซล จากนั้นนำไปทำกระบวนการ นอน-แมกซีมัม ซับเพรสชัน (Nonmaximum Suppression) โดยกระบวนการนี้ จะหาพิกเซลที่มีขนาดของเกรเดียนตามทิศทางที่ตำแหน่งนั้นมากที่สุด จากนั้นพิเซลดังกล่าวจะนำไปทำกระบวนการ Hysteresis Threshold โดยกำหนดค่าเทรชโฮลด์ขึ้นมา 2 ค่า โดยค่ามากเป็น T_h และค่าน้อยเป็น T_l พิกเซลใดมีขนาดของเกรเดียนมากกว่า T_h จะให้เป็นขอบภาพ จากนั้นพิกเซลใดที่อยู่ติดกับพิกเซลที่เป็นขอบภาพและมีค่ามากกว่า T_l จะให้เป็นขอบภาพเช่นกัน

โดยในรูปด้านล่างจะแสดงผังการทำงานของ Canny

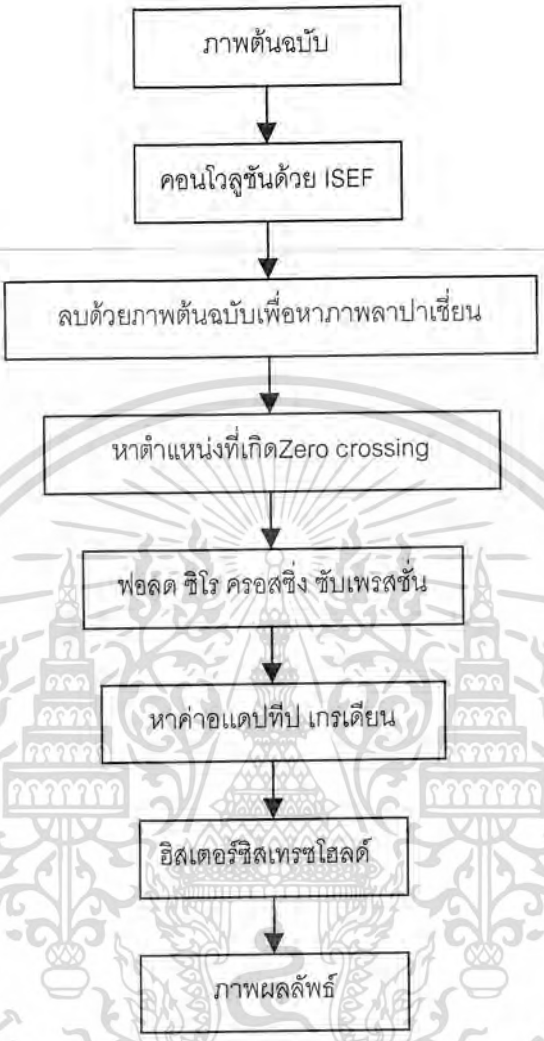
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.41 แสดงผังการทำงานของ Canny

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.21 อัลกอริทึมของ Shen และ Castan



รูปที่ 2.42 ผังการทำงานของการหาขอบภาพแบบ Shen และ Castan

โดย Shen และ Castanได้ใช้หน้ากากคอนโวลูชันหรือฟิลเตอร์ที่มีชื่อเรียกว่า Infinite Symmetric Exponential หรือ ISEF ในการคอนโวลูชันระหว่างภาพต้นฉบับกับ ISEF จะทำซ้ำแบบ Recursive ภาพที่ได้จากการฟิลเตอร์จะถูกนำไปประมาณค่าของลาปาเซี่ยน โดยการนำไปลบกับภาพต้นฉบับ แล้วหาขอบภาพจากตำแหน่งที่เกิด Zero crossing จากนั้นจะเป็นการปรับปรุงขอบภาพที่ได้โดยการตัดส่วนที่เป็นขอบภาพปลอม (False Edge) ทิ้ง โดยขั้นแรกเรียกว่า False Zero – Crossing Suppression วิธีนี้ทำโดยการตั้งสมมติฐานที่ว่า ในตำแหน่งที่เกิด Zero – Crossing จากบวกเป็นลบ (Positive Zero – Crossing) นั้นจะต้องตรงกับตำแหน่งที่มีค่าเกรเดียนที่จุดนั้นเป็นบวก และในตำแหน่งที่เกิด Zero – Crossingจากลบเป็นบวก(Negative Zero – Crossing) จะต้องตรงกับตำแหน่งที่มีค่าเกรเดียนที่จุดนั้นเป็นลบ ดังนั้นตำแหน่งใดที่ไม่ตรงตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เงื่อนไขนี้ตัดทิ้ง ขั้นที่ 2 พิกเซลที่เป็นขอบภาพจะได้รับค่าเกรเดียน โดยค่าเกรเดียนนี้จะปรับตามแต่ละพื้นที่ในภาพ (Adaptive Gradient) จากนั้นตำแหน่งที่เป็นที่เป็นขอบภาพนี้จะถูกทำการเทรชโฮลด์อีกครั้งโดยใช้วิธี ฮิสเตอร์ซิสเทรชโฮลด์(Hystersis Threshold) เช่นเดียวกับของ Canny ซึ่งการใช้ค่าเกรเดียนแบบปรับได้ร่วมกับการทำเทรชโฮลด์แบบนี้จะมีผลประโยชน์มากโดยเฉพาะกับภาพที่มีการรบกวนมาก

2.22 สรุปการทำงานของอัลกอริทึมต่างๆ

ตารางที่ 3 แสดงการสรุปการทำงานของอัลกอริทึมต่างๆ

อัลกอริทึม	หน้ากากคอนโวลูชัน	การหาตำแหน่งขอบภาพ	การปรับปรุงขอบภาพ
Marr-Hildreth	LoG. Filter	Zero-Crossing	ทำ 2 ครั้ง(2 สเกล)
Canny	Derivative of Gaussian Filter	ตำแหน่งที่เกรเดียนมีค่าสูงสุด	ฮิสเตอร์ริซิสเทรชโฮลด์
Shen-Castan	ISEF.	Zero-Crossing	ฮิสเตอร์ริซิสเทรชโฮลด์

จากเนื้อหาที่ผ่านมาจะเห็นได้ว่า อัลกอริทึมในการหาขอบภาพจะประกอบด้วยขั้นตอนต่างๆ สามขั้นตอน

ขั้นตอนแรก จะเป็นการเน้นขอบภาพ (Edge Enhancement) ซึ่งเป็นการเน้นให้ตำแหน่งขอบภาพเด่นชัด และสามารถตรวจพบได้ง่ายขึ้น ทำโดยการคอนโวลูชันระหว่างภาพต้นฉบับกับหน้ากากคอนโวลูชัน ซึ่งจะใช้หน้ากากต่างกันในแต่ละอัลกอริทึม

ขั้นตอนที่ 2 คือ การหาตำแหน่งที่เป็นขอบภาพ ซึ่งมีหลายวิธี เช่นหาจากตำแหน่งที่เกิด Zero-Crossing หาจากตำแหน่งที่มีค่าเกรเดียนมากที่สุดซึ่งขึ้นอยู่กับหน้ากากคอนโวลูชันที่ใช้

ขั้นตอนที่ 3 จะเป็นการปรับปรุงขอบภาพที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ลักษณะทั่วไปของโปรแกรม

3.1คุณสมบัติต่าง ๆ ของ โปรแกรม

คุณสมบัติต่าง ๆ ของ โปรแกรมที่สร้างมีความสามารถดังต่อไปนี้

3.1.1 กระบวนการอิมเมจเกี่ยวกับจุด Point Processing

- กลับสี (Invert Color)เช่น กลับสีจากสีขาวกลายเป็นดำ เป็นต้น
- แปลงสีของอิมเมจเป็นขาวดำ (Gray Color)
- ดูฮิสโทแกรม (Histogram) ของอิมเมจ
- ทำ Histogram Equalized
- แก้ไขค่าระดับ (Level) ความเข้มของอิมเมจโดยใช้แกมมา (Gamma)
- การสร้างอิมเมจใหม่โดยใช้ลอจิก And , Or , Xor และการใช้การบวกการลบและการหาค่าเฉลี่ย
- หมุนอิมเมจ รวมทั้งกลับรูปอิมเมจในแนวตั้ง (Mirror) และกลับรูปอิมเมจในแนวนอน (Flip)

3.1.2 กระบวนการอิมเมจที่เกี่ยวกับบริเวณรอบ ๆ จุดพิกเซล (Area Processing)

- การทำภาพอิมเมจให้มัว (Low Pass Filter)
- การทำภาพอิมเมจให้คมชัดมากขึ้น (High Pass Filter)
- การหาลายเส้นขอบอิมเมจ (Edge Detection) เช่น Laplacian , Sobel ,Prewitt และ Frei-Chen
- ทำภาพอิมเมจให้นูน (Emboss)

3.1.3 กระบวนการทำอิมเมจที่เกี่ยวกับรูปร่างลักษณะ (Morphology Processing)

- ทำการลดขนาดพิกเซลของอิมเมจ (Erosion)
- ทำการเพิ่มขนาดพิกเซลของอิมเมจ (Dilation)
- ทำอิมเมจในพื้นที่ว่างให้เปิดมากขึ้น (Opening)
- ทำอิมเมจในพื้นที่ว่างให้ปิดมากขึ้น (Closing)

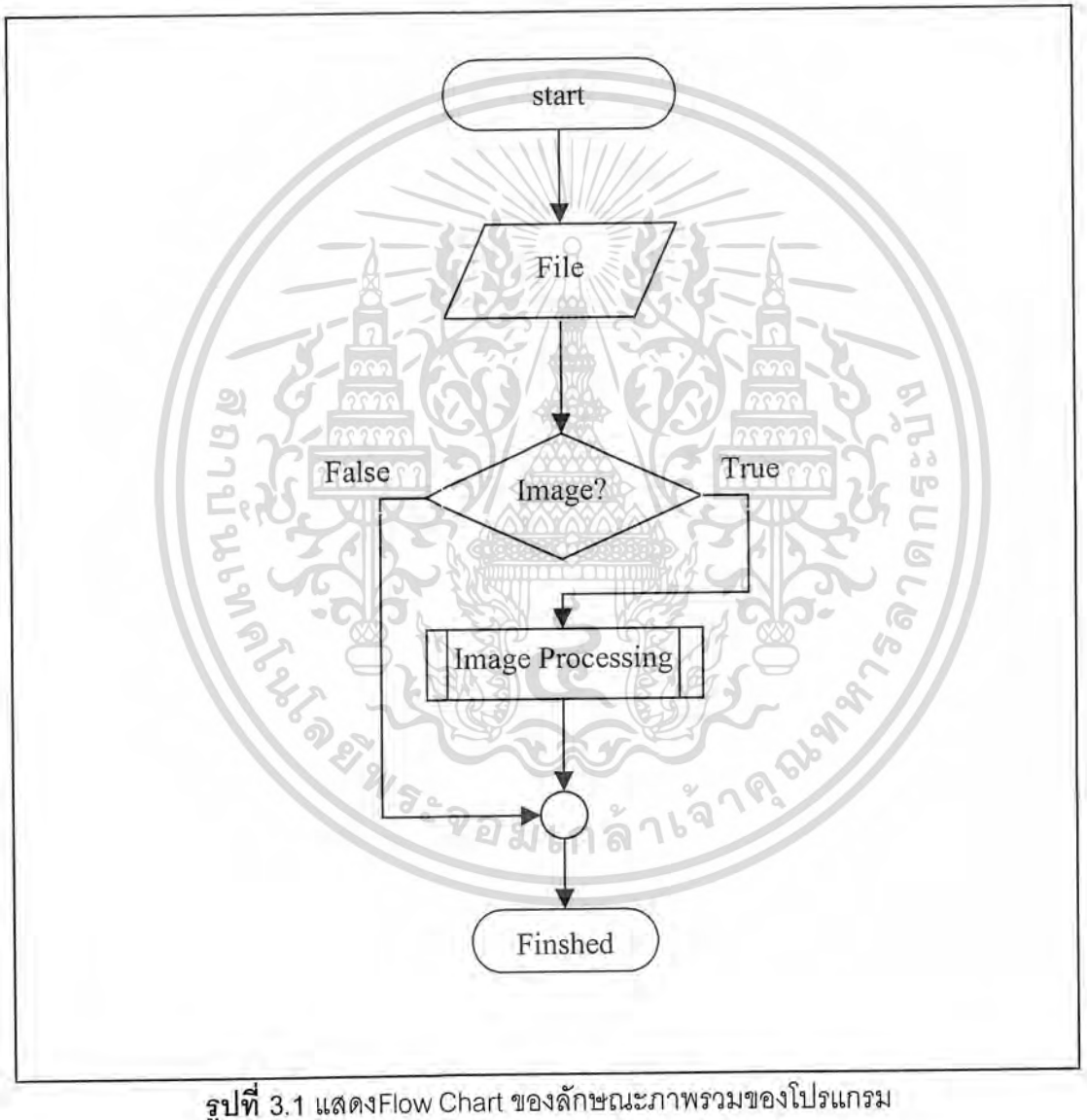
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ลักษณะทั่วไปของโปรแกรม

3.2.1 ลักษณะงานของโปรแกรม

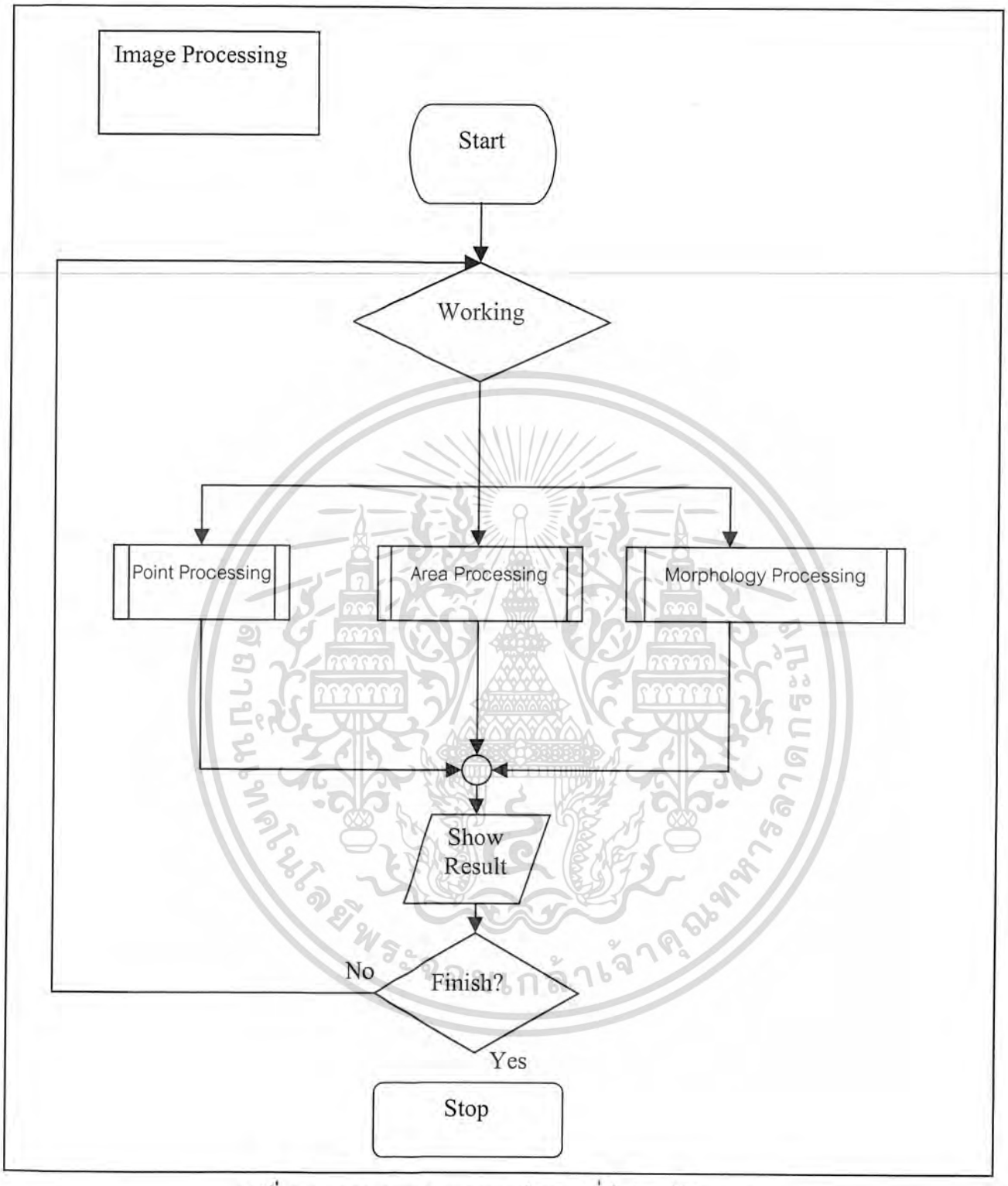
ในระบบการทำงานของโปรแกรมนั้น จะต้องทำการรับภาพต้นฉบับ สามารถที่จะเปิดไฟล์อิมเมจชนิดต่างๆ ได้แก่ BMP, DIB ,RLE , JPEG , GIF , TIFF , TIF และ PNG เพื่อที่จะสามารถนำมาปรับแต่ แก้ไข เป็นเอกสารหรือใช้เพื่อวิวภาพ และ เมื่อดำเนินการเรียบร้อยผู้ใช้สามารถบันทึกเอกสารเป็นไฟล์อิมเมจชนิดต่างๆ ได้

ผู้ใช้ยังสามารถสั่งพิมพ์ภาพออกทางเครื่องพิมพ์และก่อนพิมพ์สามารถดู เอกสารก่อนพิมพ์ได้



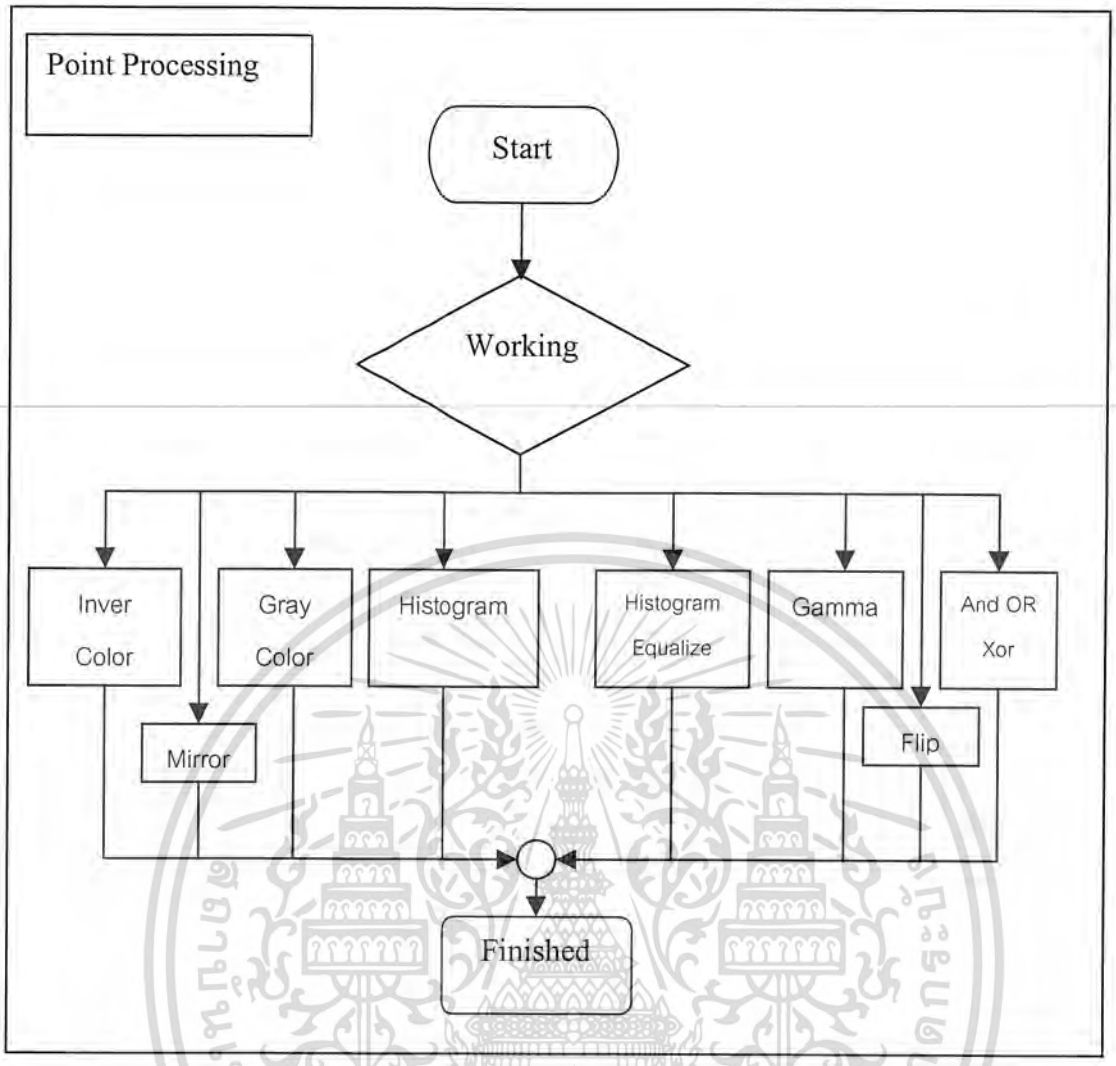
รูปที่ 3.1 แสดงFlow Chart ของลักษณะภาพรวมของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงFlow Chart ของลักษณะทั่วไปของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

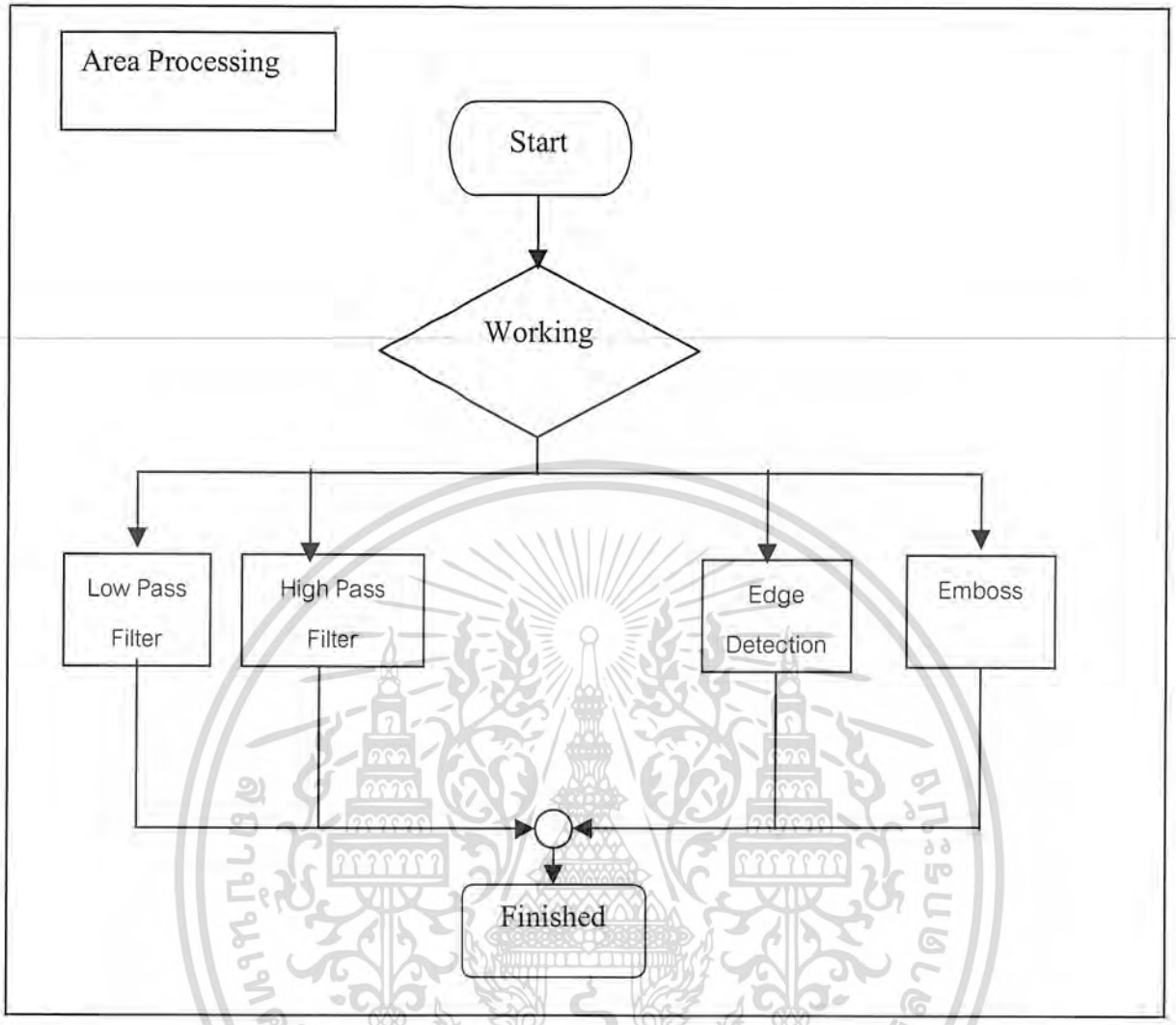


รูปที่ 3.3 แสดงFlow Chart ของกระบวนการ Point Processing

กระบวนการอิมเมจเกี่ยวกับจุด Point Processing

- กลับสี (Invert Color)เช่น กลับสีจากสีขาวกลายเป็นดำ เป็นต้น
- แปลงสีของอิมเมจเป็นขาวดำ (Gray Color)
- ดูฮิสโทแกรม (Histogram) ของอิมเมจ
- ทำ Histogram Equalized
- แก้ไขค่าระดับ (Level) ความเข้มของอิมเมจโดยใช้แกมมา (Gamma)
- การสร้างอิมเมจใหม่โดยใช้ลอจิก And , Or , Xor และการใช้การบวกการลบและการหาค่าเฉลี่ย
- หมุนอิมเมจ รวมทั้งกลับรูปอิมเมจในแนวตั้ง (Mirror) และกลับรูปอิมเมจในแนวนอน (Flip)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

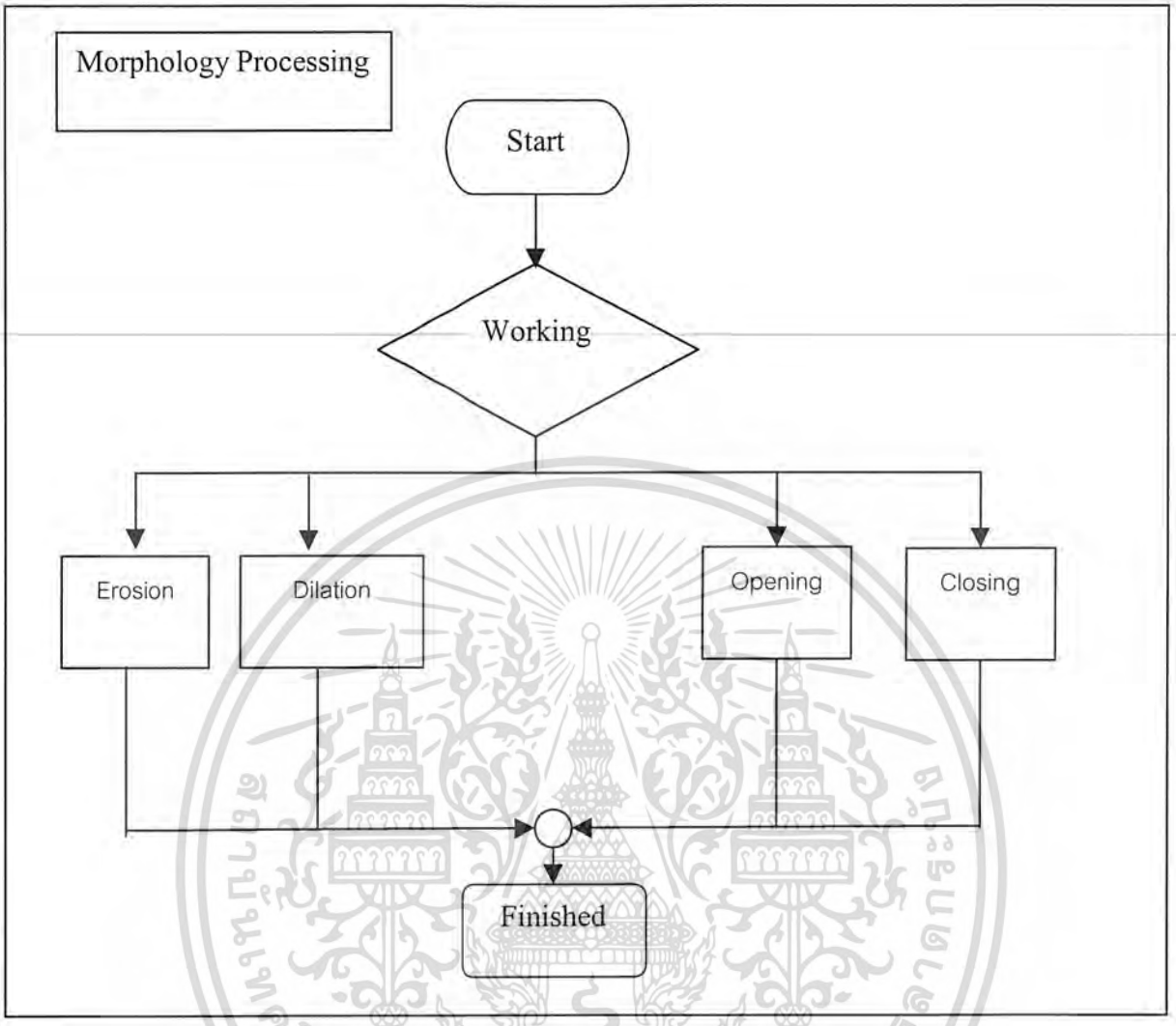


รูปที่ 3.4 แสดงFlow Chart ของกระบวนการ Area Processing

กระบวนการอิมเมจที่เกี่ยวกับบริเวณรอบ ๆ จุดพิกเซล (Area Processing)

- การทำภาพอิมเมจให้มัว (Low Pass Filter)
- การทำภาพอิมเมจให้คมชัดมากขึ้น (High Pass Filter)
- การหาลายเส้นขอบอิมเมจ (Edge Detection) เช่น Laplacian , Sobel ,Prewitt และ Frei-Chen
- ทำภาพอิมเมจให้นูน (Emboss)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงFlow Chart ของกระบวนการ Morphology Processing

กระบวนการทำอิมเมจที่เกี่ยวกับรูปร่างลักษณะ (Morphology Processing)

- ทำการลดขนาดพิกเซลของอิมเมจ (Erosion)
- ทำการเพิ่มขนาดพิกเซลของอิมเมจ (Dilation)
- ทำอิมเมจในพื้นที่ว่างให้เปิดมากขึ้น (Opening)
- ทำอิมเมจในพื้นที่ว่างให้ปิดมากขึ้น (Closing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบตัวโปรแกรม

ตารางที่ 4 รายละเอียดเมนูหลักและเมนูย่อย

เมนูหลัก	เมนูย่อย	รายละเอียด
File	Open	ใช้เพื่อเปิดไฟล์อิมเมจชนิดต่างๆ ได้แก่ Bmp,Dib,Rle,Jpeg,Gif,Tiff,Tifและ Png
	Close	ใช้เพื่อเปิดเอกสารและวิว
	Save As	ใช้เพื่อบันทึกเอกสารเป็นไฟล์อิมเมจชนิดต่างๆ
	Print	ใช้เพื่อพิมพ์เอกสารหรืออิมเมจออกทางเครื่องพิมพ์
	Print Preview	ใช้เพื่อดูเอกสารก่อนพิมพ์ออกจากเครื่องพิมพ์
	Print Setup	ใช้เพื่อset up เครื่องพิมพ์ขนาดกระดาษ ทิศทางการพิมพ์(Portrait หรือ Landscape)
	Exit	เลิกใช้โปรแกรม
Edit	Undo	ใช้เพื่อย้อนกลับการกระทำครั้งสุดท้าย
View	Toolbar	ใช้เพื่อเปิดปิดทูลบาร์อิมเมจ
	Status Bar	ใช้เพื่อเปิดปิด status bar ที่ใช้อธิบายคำสั่งเมนูหรือทูลบาร์
	Zoom In (+)	ใช้เพื่อเพิ่มขยายขนาดภาพอิมเมจให้มีขนาดใหญ่ขึ้น
	Zoom Out (-)	ใช้เพื่อลดขนาดภาพอิมเมจ
	Full Size	ใช้เพื่อ set ขนาดภาพอิมเมจให้เท่ากับภาพต้นฉบับ
Images	Invert Color	ใช้เพื่อกลับสีอิมเมจ เช่น จากขาวเป็นดำ เป็นต้น
	Gray Scale	ใช้เพื่อเปลี่ยนสีเป็นขาว-ดำ 256 ระดับ
	Histogram	ใช้เพื่อดูฮิสโทแกรม และ ข้อมูลสถิติของอิมเมจ เช่น หาค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน
	Histogram Equalized	ใช้เพื่อเพิ่มความสว่างแก่อิมเมจด้วย Histogram Equalized
	Gamma	ใช้เพื่อเพิ่มหรือลดด้วยค่า Gamma (r) สำหรับอิมเมจ
	Flip(Vertical)	ใช้เพื่อกลับรูปอิมเมจในแนวตั้ง
	Mirror(Horizon)	ใช้เพื่อกลับรูปอิมเมจในแนวนอน
	Rotate 90	ใช้เพื่อหมุนอิมเมจ 90 องศาทวนตามนาฬิกา
	Rotate -90	ใช้เพื่อหมุนอิมเมจ 90 องศาทวนเข็มนาฬิกา
	Rotate (Angle)	ใช้เพื่อหมุนอิมเมจ ณ องศาใดๆ และ ยกเลิกการหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนูหลัก	เมนูย่อย	รายละเอียด
	Basic > And	ใช้สร้างอิมเมจใหม่ที่ได้จากการใช้ลอจิก And ระหว่าง 2 อิมเมจ
	Basic > Xor	ใช้สร้างอิมเมจใหม่ที่ได้จากการใช้ลอจิก Xor ระหว่าง 2 อิมเมจ
	Basic > Add	ใช้สร้างอิมเมจใหม่ที่ได้จากการใช้การบวกระหว่าง 2 อิมเมจ
	Basic > Diff	ใช้สร้างอิมเมจใหม่ที่ได้จากการใช้การลบระหว่าง 2 อิมเมจ
	Basic > Average	ใช้สร้างอิมเมจใหม่ที่ได้จากการหาเฉลี่ยระหว่าง 2 อิมเมจ
	Image Composite	ใช้สร้างอิมเมจใหม่ที่ได้จากการใช้โลจิก And,Orและลบ
Filter Tools	Low Pass	ใช้เพื่อทำภาพให้มัว
	High Pass	ใช้เพื่อทำภาพให้คมชัด
	Laplacian Edge	ใช้เพื่อหาลายเส้นขอบอิมเมจ (Edge Detection)
	Emboss	ใช้เพื่อทำภาพอิมเมจให้นูนขึ้น
	Custom	แสดงไดอะล็อกให้ผู้ใช้กำหนดค่าการทำกระบวนการอิมเมจบริเวณรอบๆจุดพิเซล
	Pencil Drawing	ใช้เพื่อหาลายเส้นอิมเมจโดยมีพื้นที่เป็นสีขาว
Edge Detection	Sobel (X)	ใช้เพื่อหาลายเส้นอิมเมจทั้ง X
	Sobel (Y)	ใช้เพื่อหาลายเส้นอิมเมจทั้ง Y
	Sobel (XY)	ใช้เพื่อหาลายเส้นอิมเมจทั้ง X และ Y
	Prewitt (X)	ใช้เพื่อหาลายเส้นอิมเมจทั้ง X
	Prewitt (Y)	ใช้เพื่อหาลายเส้นอิมเมจทั้ง Y
	Prewitt (XY)	ใช้เพื่อหาลายเส้นอิมเมจทั้ง X และ Y
	Fri-Chen (X)	ใช้เพื่อหาลายเส้นอิมเมจทั้ง X
	Fri-Chen (Y)	ใช้เพื่อหาลายเส้นอิมเมจทั้ง Y
	Fri-Chen (XY)	ใช้เพื่อหาลายเส้นอิมเมจทั้ง X และ Y
Morphology	Erosion	ใช้เพื่อลดขนาดพิกเซลของอิมเมจ
	Dilation	ใช้เพื่อเพิ่มขนาดพิกเซลของอิมเมจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนูหลัก	เมนูย่อย	รายละเอียด
	Opening	ใช้เพื่อทำอิมเมจในพื้นที่ว่างหรือช่องว่าง(Gap) ให้มีการเปิดมากขึ้น
	Closing	ใช้เพื่อทำอิมเมจในพื้นที่ว่างหรือช่องว่าง ให้มีการปิดมากขึ้น

3.4 รายละเอียดของระบบงาน

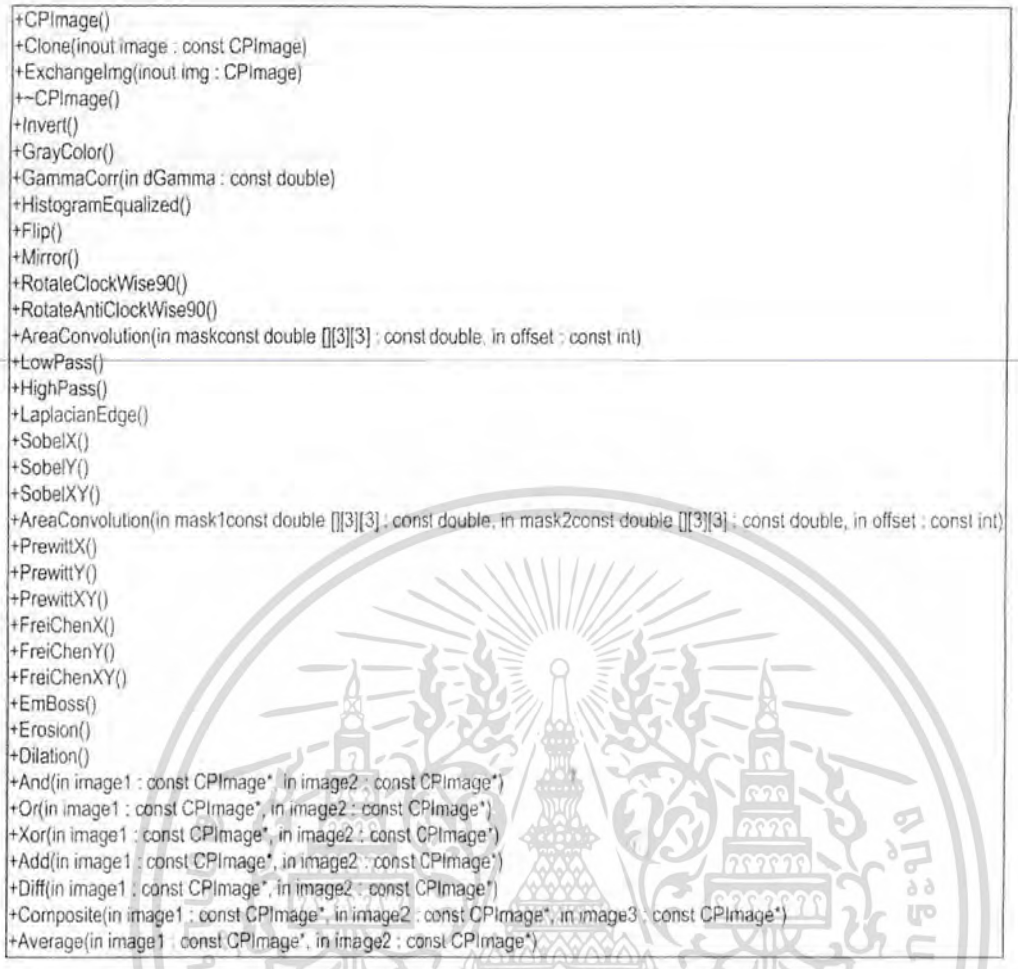
3.4.1 การรับภาพ

ภาพที่ถูกรับเข้ามาเพื่อนำมาแสดงในโปรแกรมนั้นเป็นภาพชนิด Bmp,Jpeg,Gif,Tiff,Png โดยมีความลึกสีไม่เกิน 24 บิต

3.4.2 การสร้างคลาสอิมเมจใหม่

โดยใช้คุณสมบัติของกรสืบทอด(Inheritance) ของคลาส CImage เพราะเราจะไม่ได้ดำเนินการบนบิตแมบอย่างเดียวแต่เราต้องการที่จะเปิดภาพไฟล์อิมเมจชนิด Bitmap (Bmp,Dib,Rle),Jpeg(Jpg,Jpeg,Jpe และ Jiff),Gif (Gif),Tiff (Tif,Tiff) และ Pgn (Pgn)เป็นขนาด 24 บิต(R = G= B = 8 บิต)
โดยเราจะเขียนโมเดลคลาส ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 โมเดลคลาส CPImage

3.4.3 การเลือกรูปแบบการทำงานต่าง ๆ

3.4.3.1 การอ่านข้อมูลพิกเซลในหน่วยความจำ เป็นการดึงข้อมูลแต่ละสีออกมาในหนึ่งจุดพิกเซล โดยมีอัลกอริทึม ดังนี้

```
BYTE r, g, b;
BYTE *p;
for(int y=0; y<maxHeight; y++)//Loop ตามความกว้าง
{
    for(int x=0; x<maxWidth; x++)Loop ตามความสูง
    {
        p = GetPixelAddress(x,y);
        //อ่านค่าพิกเซลที่ตำแหน่งใดๆ โดยเอาตัวแปร p มาเก็บ
        r = p[2];//เก็บค่าสีแดง
        g = p[1];//เก็บค่าสีเขียว
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            b = p[0];//เก็บค่าสีน้ำเงิน
        }//for y
    }//for x
```

รูปที่ 3.7 อัลกอริทึมการอ่านข้อมูลพิกเซล

3.4.3.2 **การทำ** Grayscale ฟังก์ชันนี้จะทำงานโดยจะเปลี่ยนภาพต้นฉบับเป็นภาพแบบขาวดำใน 256 ระดับ โดยเราจะนำค่าสีของแต่ละพิกเซลออกมาแยกค่าสีตามแม่สีในโหมด RGB แล้วนำค่าสีแดง สีเขียว และ น้ำเงินมาแทนค่าลงในสมการ

Grayscale = r*0.299 + g*0.587 + b*0.114 หรือ

Grayscale = $((r^2+g^2+b^2)/3)^{1/2}$

r = ค่าของสีแดง

g = ค่าของสีเขียง

b = ค่าของสีน้ำเงิน

โดยมีอัลกอริทึม ดังนี้

```
อ่านข้อมูลความกว้าง
อ่านข้อมูลความสูง
BYTE r,g,b,avg;
BYTE *p;
Loop ตามความกว้าง
{
    Loop ตามความสูง
    {
        อ่านค่าพิกเซลที่ตำแหน่งใดๆ โดยเอาตัวแปร p มาเก็บ
        r = p[2];//เก็บค่าสีแดง
        g = p[1];//เก็บค่าสีเขียว
        b = p[0];//เก็บค่าสีน้ำเงิน
        avg = (BYTE)(r*0.299 + g*0.587 + b*0.114);
        *p++ = avg;
        *p++ = avg;
        *p = avg;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }//for y
    }//for x
```

รูปที่ 3.8 อัลกอริทึมการทำภาพGrayscale

3.4.3.3 **การทำการกลับสี** (Invert Color) เป็นวิธีการกลับสีจากสว่างเป็นมืดและจากมืดเป็นสว่าง

โดยจะมีสมการการแปลง ดังนี้

r = 256-1-r

g = 256-1-g

b = 256-1-g

โดยมีอัลกอริทึม ดังนี้

```
อ่านข้อมูลความกว้าง
อ่านข้อมูลความสูง
for (int y = 0; y<maxHeight; y++)
{
    for (int x = 0; x<maxWidth; x++)
    {
        p = (BYTE*)GetPixelAddress(x,y);
        p[2] = 255-p[2];
        p[1] = 255-p[1];
        p[0] = 255-p[0];
    }//for x
}//for y
```

รูปที่ 3.9 อัลกอริทึมการทำภาพInvert

3.4.3.4 **การแก้ไขพิกเซลด้วยแกมมา**(Gamma Correction) โดยจะใช้สมการ $Output = Input^{1/Gamma}$ โดยเราจะรับค่าตัวแปรแกมมาจากไดอะล็อกโดยมีค่าจาก 1 ถึง 10 โดยมีอัลกอริทึม ดังนี้

```
//รับค่าแกมมา
Double gamma = 1/G;
อ่านข้อมูลความกว้าง
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
อ่านข้อมูลความสูง
for (int y = 0; y<maxHeight; y++)
{
        for (int x = 0; x<maxWidth; x++)
        {
                p = (BYTE*)GetPixelAddress(x,y);
                rGamma = pow(p[2]/255,gamma)*255;
                gGamma = pow(p[1]/255,gamma)*255;
                bGamma = pow(p[0]/255,gamma)*255;
        }//for x
}//for y
```

รูปที่ 3.10 อัลกอริทึมการทำภาพGamma Correction Tranformation

3.4.3.5 **การทำ** Histogram Equalized โดยเราจะคำนวณฮิสโทแกรมของอิมเมจก่อน คือ คำนวณความถี่ที่เกิดขึ้นซ้ำๆ ณ ระดับความเข้มของพิกเซลนั้น ตัวอย่างเช่น ที่ระดับความเข้มที่ 66 มีการเกิดขึ้นซ้ำกันทั้งหมด 1000 ครั้ง ดังนั้น ฮิสโทแกรมจะมีความถี่ 1000 ที่ระดับความเข้มที่ 66 แล้วจึงคำนวณผลรวมของฮิสโทแกรมในแต่ละระดับความเข้มก่อนโดยเริ่มจากความเข้มที่ 0 ถึง ความเข้มของจุดพิกเซลใดๆ จากนั้นทำการนำค่าผลรวมที่พิกเซลนั้นมาคูณกับค่าความเข้มสูงสุด (255) แล้วหารด้วยจำนวนพิกเซลทั้งหมดของอิมเมจ
โดยมีอัลกอริทึม ดังนี้

```
for (int y=0; y <maxHigh ; y++)
{
        for  (int x=0; x<maxWidth; x++)
        {
                pSrc = (BYTE*)GetPixelAddress( x,y );
                pRed[pSrc[2]]++;
                pGreen[pSrc[1]]++;
                pBlue[pSrc[0]]++;
        }
}
int nPixels = maxWidth*maxHigh;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

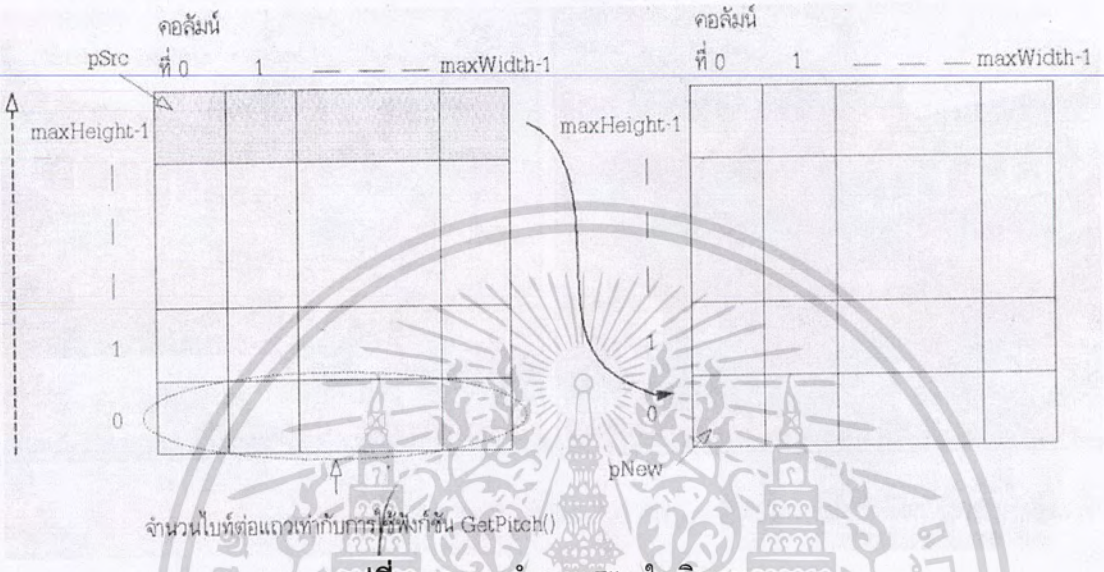
```
int sR, sGr, sB;
sR = sGr = sB = 0;
float factor = (float)(255.0/nPixels);

for (int i = 0; i < 256; i++)
{
        sR += pRed[i];
        sGr += pGreen[i];
        sB += pBlue[i];
        sHRed[i] =(int)((sR*factor)+0.5);
        sHGreen[i] =(int)((sGr*factor)+0.5);
        sHBlue[i] =(int)((sB*factor)+0.5);
}

for (int y=0; y <maxHigh ; y++)
{
        for  (int x=0; x<maxWidth; x++)
        {
                pSrc = (BYTE*)GetPixelAddress( x,y );
                pNew = (BYTE*)image.GetPixelAddress(x,y);
                *pNew++ = (BYTE)sHBlue[pSrc[0]];
                *pNew++ = (BYTE)sHGreen[pSrc[1]];
                *pNew= (BYTE)sHRed[pSrc[2]];
        }
}
```

รูปที่ 3.11 อัลกอริทึมการทำHistogram Equalized

3.4.3.6 **การกลับภาพในแนวตั้ง** (Flip) โดยเราจะใช้ฟังก์ชัน GetBits() โดยถ้าฟังก์ชันนี้จะนำพอยน์เตอร์ไปชี้ที่แถวแรก ณ ตำแหน่งท้ายสุด คอลัมน์แรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนฟังก์ชัน GetPitch() นั้นจะบอกจำนวนต่อการไบท์ต่อการสแกนใน 1 บรรทัดด้วย ตัวอย่างเช่น ถ้าอิมเมจมีความกว้างและความสูงเท่ากับ 256 หน่วยถ้าใช้ฟังก์ชัน GetPitch() จะคืนค่าเป็น 768 หรือ 256*3 (อย่าลืมว่าภาพนั้นมีความลึก 24 บิต จะประกอบด้วยแม่สีอย่างละ 8 บิต รวมกันแล้วเท่ากับ 24 บิตพอดี) จากแนวคิดนี้จะแสดงได้ดังรูป



รูปที่ 3.12 การทำการ Flip ในบิตแมบ

โดยมีอัลกอริทึมดังนี้

```
BYTE *pSrc,*pNew;
int highTotal = (maxHeight-1)*GetPitch();//ความสูงของอิมเมจ
pSrc = (BYTE*)GetBits() + highTotal;
pNew = (BYTE*)image.GetBits();
/*Retrieves a pointer to the actual bit values of a given pixel in a bitmap.
void* GetBits( ) throw( );*/
int nPitch = abs(GetPitch());
for(int y=0; y < maxHeight; y++)
{//Copies characters between buffers.
        memcpy(pNew,pSrc,nPitch);
        pSrc = pSrc - GetPitch();//move to next row
        pNew = pNew + GetPitch();//move to next row
}//for y
```

รูปที่ 3.13 อัลกอริทึมการFlip

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3.7 **การกลับภาพในแนวนอน** (Mirror) โดยใช้แนวคิดเดียวกับการ Flip โดยแทนที่เราจะบวกด้วยความสูงของภาพเราจะบวกด้วยความกว้างของภาพดังนั้นพอยน์เตอร์ที่ได้ก็จะอยู่ที่ตำแหน่งแถวล่างสุดมุมขวา

รูปที่ 3.14 การทำการ Mirrorในบิตแมบ

โดยมีอัลกอริทึมดังนี้

```
BYTE *pSrc,*pNew;
int widthTotal = (maxWidth-1)*3;//มีค่า RGB อย่างละ 1 ไบท์
pSrc = (BYTE*)GetBits();
pSrc = pSrc + widthTotal;//ชี้ที่แถวล่างหลักสุดท้าย
pNew = (BYTE*)image.GetBits();
for(int y=0; y<maxHeight;  y++)
{
        for(int x=0; x<=widthTotal;  x+=3)
        {
                *(pNew+x) = *(pSrc-x);
                *(pNew+x+1) = *(pSrc-x+1);
                *(pNew+x+2) = *(pSrc-x+2);
        }//for x
        pSrc = pSrc+GetPitch();
        pNew = pNew+GetPitch();
}//for y
```

รูปที่ 3.15 อัลกอริทึมการMirror

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3.8 **การทำการบวกลบอิมเมจและการใช้ลอจิคต่างๆ** เป็นฟังก์ชันเพื่อการสร้างอิมเมจใหม่โดยจะรับอินพุตภาพ 2 ภาพมาทำการบวก ลบ หรือ เข้าลอจิคต่างๆ กัน

3.4.3.8.1 **การทำลอจิค And** เป็นการนำพอยต์เตอร์ไปชี้ที่ภาพต้นแบบ 2 ภาพ แล้วอ่านค่าพิกเซลจากแหล่งภาพทั้ง 2 แล้วนำมาเข้าตรรกศาสตร์ด้วยการลอจิคแอนด์ โดยมีอัลกอริทึมดังนี้

```
for(int y=0; y<maxHeight; y++)
{
        for(int x=0; x<maxWidth; x++)
        {
        p1=อ่านข้อมูลอิมเมจภาพที่หนึ่ง
        p2=อ่านข้อมูลอิมเมจภาพที่สอง
        p3=นำพอยน์เตอร์ไปชี้ที่อิมเมจใหม่
        p3[0] = p1[0] & p2[0];//blue
        p3[1] = p1[1] & p2[1];//green
        p3[2] = p1[2] & p2[2];//red
        }//for x
}//for y
```

รูปที่ 3.16 อัลกอริทึมการAnd

3.4.3.8.2 **การทำลอจิค Or** เป็นการนำพอยต์เตอร์ไปชี้ที่ภาพต้นแบบ 2 ภาพ แล้วอ่านค่าพิกเซลจากแหล่งภาพทั้ง 2 แล้วนำมาเข้าตรรกศาสตร์ด้วยการลอจิคออร์ โดยมีอัลกอริทึมดังนี้

```
for(int y=0; y<maxHeight; y++)
{
        for(int x=0; x<maxWidth; x++)
        {
        p1=อ่านข้อมูลอิมเมจภาพที่หนึ่ง
        p2=อ่านข้อมูลอิมเมจภาพที่สอง
        p3=นำพอยน์เตอร์ไปชี้ที่อิมเมจใหม่
        p3[0] = p1[0] | p2[0];//blue
        p3[1] = p1[1] | p2[1];//green
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            p3[2] = p1[2] | p2[2];//red
            }//for x
        }//for y
```

รูปที่ 3.17 อัลกอริทึมการOr

3.4.3.8.1 **การทำลอจิค** Xor เป็นการนำพอยต์เตอร์ไปชี้ที่ภาพต้นแบบ 2 ภาพ แล้วอ่านค่าพิกเซลจากแหล่งภาพทั้ง 2 แล้วนำมาเข้าตรรกศาสตร์ด้วยการลอจิคXor

โดยมีอัลกอริทึมดังนี้

```
for(int y=0; y<maxHeight; y++)
{
        for(int x=0; x<maxWidth; x++)
        {
        p1=อ่านข้อมูลอิมเมจภาพที่หนึ่ง
        p2=อ่านข้อมูลอิมเมจภาพที่สอง
        p3=นำพอยน์เตอร์ไปชี้ที่อิมเมจใหม่
        p3[0] = p1[0] ^ p2[0];//blue
        p3[1] = p1[1] ^ p2[1];//green
        p3[2] = p1[2] ^ p2[2];//red
        }//for x
}//for y
```

รูปที่ 3.18 อัลกอริทึมการAnd

3.4.3.8 **การทำ** Composite Image เป็นการรวมภาพที่มีขนาดเดียวกัน โดยภาพผลลัพท์ที่ได้คือการนำเอาภาพทั้ง 3 มารวมกันโดยใช้ลอจิก And,Or และ Invert ร่วมกัน โดยมีอัลกอริทึมดังนี้

```
for(int y=0; y<maxHeight; y++)
{
        for(int x=0; x<maxWidth; x++)
        {
            p1=อ่านข้อมูลอิมเมจส่วนหน้า
            p2=อ่านข้อมูลอิมเมจmask
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            p3=อ่านข้อมูลอิมเมจส่วนหลัง
            blue = p1[0] & p2[0] | p3[0] & (255 – p2[0]);
            green = p1[1] & p2[1] | p3[1] & (255 – p2[1]);
            red = p1[2] & p2[2] | p3[2] & (255 – p2[2]);
        }//for x
    }//for y
```

รูปที่ 3.19 อัลกอริทึมการทำComposite

3.4.3.9 **การทำการแมบเข้ากับหน้ากาก** (Masking Convolution) เป็นฟังก์ชันที่ทำหน้าที่คำนวณค่าของพิกเซลบริเวณรอบๆ พิกเซลหนึ่ง ซึ่งเราเรียกว่า กระบวนการอิมเมจที่เกี่ยวกับบริเวณรอบๆ จุดพิกเซล (Area Processing) หรือ การทำ Convolution ซึ่งก็คือ การคำนวณผลรวมค่าของพิกเซลเฉลี่ยถ่วงน้ำหนักบริเวณรอบๆ พิกเซลหนึ่ง โดยใช้เมตริกซ์ที่เรียกว่า Convolution Mask หรือ Convolution Kernel ในที่นี้จะใช้ขนาดมิติ 3*3 เป็นหลักแล้วนำผลรวมหลังการถ่วงน้ำหนักมาหารด้วยผลรวมค่าน้ำหนักหรือสัมประสิทธิ์ในหน้ากาก Convolution Mask

ตัวอย่างการคำนวณ

รูปภาพต้นฉบับ

0	5	6	7
1	5	5	3
5	3	4	2
7	4	6	3

Convolution Mask

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

Result Image

	6	...	
	...	...	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าพิกเซลใหม่คำนวณได้จาก = 0-5+0-1+20-5+0-3+0 = 6

โดยมีอัลกอริทึม ดังนี้

รับค่า mask[3][3] และ offset

```
double factor = 0.0;
for (int j=0; j<3; j++)//หาค่าถ่วงน้ำหนักโดยการหาผลรวมสัมประสิทธิ์ในหน้ากาก
{
        for(int i=0; i<3; i++)
        {
                factor = factor+mask[j][i];
        }//for i
}//for j
for(int y=1; y<maxHeight; y++)
{
        for(int x=1; x<maxWidth; x++)
        {
                nRedCenter=nBlueCenter=nGreenCenter=0;
                pNew = GetPixelAddress(x,y);

                for(int j=0; j<3; j++)
                {
                        for(int i=0; i<3; i++)
                        {
                        p = GetPixelAddress(x+i-center,y+j-center);
                        nRedCenter = nRedCenter + (int)(p[2] * mask[j][i]);
                        nGreenCenter = nGreenCenter + (int)(p[1] * mask[j][i]);
                        nBlueCenter   = nBlueCenter  + (int)(p[0] * mask[j][i]);
                        }//for i
                }//for j
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            red = (nRedCenter/factor+offset);
            green =(nGreenCenter/factor+offset);
            blue =(nBlueCenter/factor+offset);

            *pNew++ = (BYTE)blue;
            *pNew++ = (BYTE)green;
            *pNew = (BYTE)red;
        }//for x
    }//for y
```

รูปที่ 3.20 อัลกอริทึมการทำConvolution

3.4.3.10 **การขยายพิกเซล (Dilation)** ฟังก์ชันนี้คล้ายกับการทำ Convolution ตามที่ได้กล่าวไปข้างต้นโดยเราจะกำหนดค่าต่ำสุดก่อน (0) แก่ทุกแม่สี ต่อไปจะคำนวณหาค่าพิกเซลบริเวณรอบๆ ของพิกเซลนั้นโดยการนำเอาค่าของพิกเซลที่อ่านมานำมาบวกกับค่าหน้ากากที่กำหนด จากนั้นหาค่ามี่มากที่สุดหลังจากการบวกกับหน้ากาก เพื่อนำไปเป็นค่าผลลัพธ์ของภาพใหม่

ตัวอย่างการคำนวณ

รูปภาพต้นฉบับ

45	30	50	30
70	50	25	100
100	60	90	25
200	0	10	5

Structure Element

25	*20*	*10*
15	*10*	*15*
5	*0*	*20*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Result Image

	110	...	
	...	...	

ค่าพิกเซลใหม่คำนวณได้จากการเลือกค่าที่มากที่สุดจาก 70,50,60,85,60,40,105,60,**110** ซึ่ง 110 เป็นค่ามากที่สุด

โดยมีอัลกอริทึม ดังนี้

```
int center = (kernel -1)/2;//kernel = 3 (มิติ)
for(int y=1; y<maxHeight-1; y++)
{
        for(int x=1; x<maxWidth-1; x++)
        {
                max=0;
                for(int j=0; j<kernel; j++)
                {
                        for(int i=0; i<kernel; i++)
                        {
                                p=GetPixelAddress(x+i-center,y+j-center);
                                redVal = p[2] + Se[ j ][ i ];
                                greenVal = p[1] + Se[ j ][ i ];
                                blueVal = p[0] + Se[ j ][ i ];
                                red = max(red,redVal);
                                green = max(green,greenVal);
                                blue = max(blue,blueVal);
                        }//for i
                }//for j
        }//for x
}//for y
```

รูปที่ 3.21 อัลกอริทึมการทำDilation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3.11 **การลดพิกเซล** (Erosion) ฟังก์ชันนี้คล้ายกับการทำการขยายพิกเซล ตามที่ได้กล่าวไปข้างต้นแต่จะเป็นกระบวนการตรงกันข้าม โดยเราจะกำหนดค่ามากสุดก่อน (255) แก่ทุกแม่สี ต่อไปจะคำนวณหาค่าพิกเซลบริเวณรอบๆ ของพิกเซลนั้นโดยการนำเอาค่าของพิกเซลที่อ่านมานำมาลบกับค่าหน้ากากที่กำหนด จากนั้นหาค่าที่น้อยที่สุดหลังจากการลบกับหน้ากาก เพื่อนำไปเป็นค่าผลลัพธ์ของภาพใหม่

ตัวอย่างการคำนวณ

รูปภาพต้นฉบับ

45	30	50	30
70	50	25	100
100	60	90	25
200	0	10	5

Structure Element

25	*20*	*10*
15	*10*	*15*
5	*0*	*20*

Result Image

	10	...	
	...	...	

ค่าพิกเซลใหม่คำนวณได้จากการเลือกค่าที่มากที่สุดจาก 20,**10**,40,55,40,**10**,95,60,70 ซึ่ง 10 เป็นค่าน้อยที่สุด

โดยมีอัลกอริทึม ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

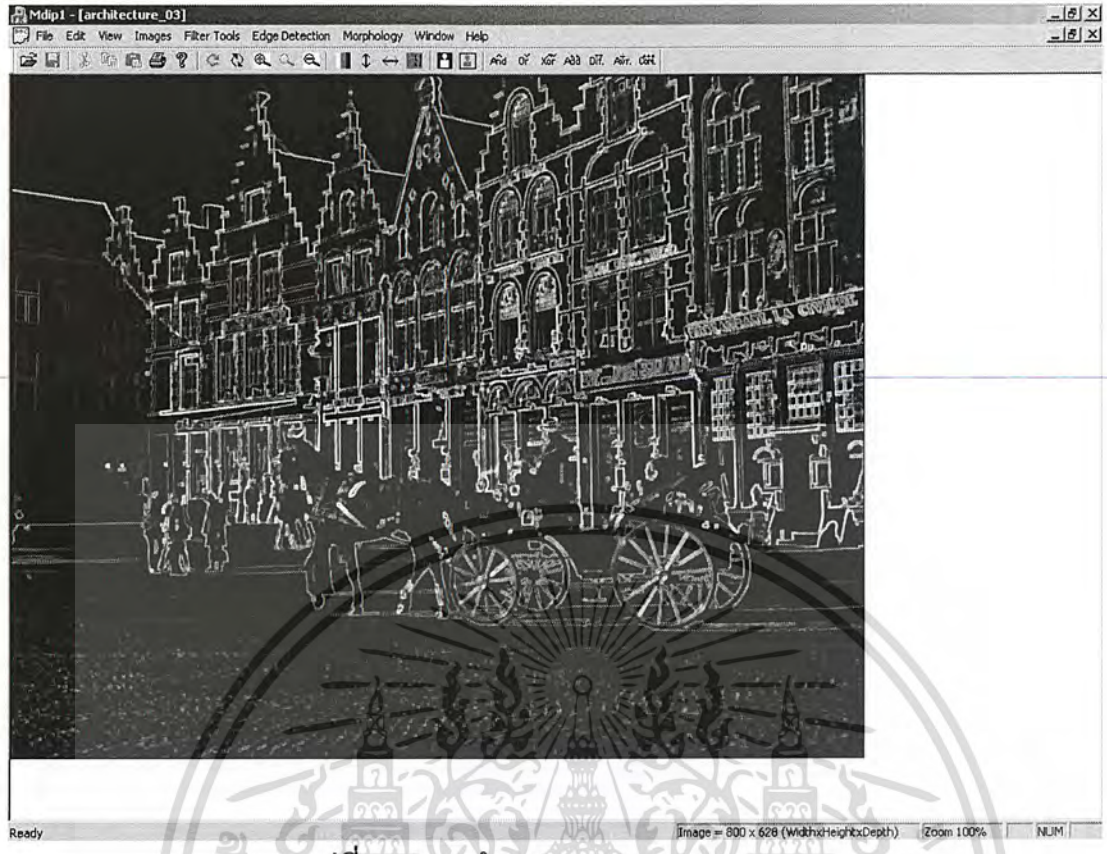
```
int center = (kernel -1)/2;//kernel = 3 (มิติ)
for(int y=1; y<maxHeight-1; y++)
{
        for(int x=1; x<maxWidth-1; x++)
        {
                max=0;
                for(int j=0; j<kernel; j++)
                {
                        for(int i=0; i<kernel; i++)
                        {
                                p=GetPixelAddress(x+i-center,y+j-center);
                                redVal = p[2] - Se[ j ][ i ];
                                greenVal = p[1] - Se[ j ][ i ];
                                blueVal = p[0] - Se[ j ][ i ];
                                red = min(red,redVal);
                                green = min(green,greenVal);
                                blue = min(blue,blueVal);
                        }//for i
                }//for j
        }//for x
}//for y
```

รูปที่ 3.22 อัลกอริทึมการทำErosion

3.4.4 ตัวอย่างการประยุกต์ใช้ Erosion และ Dilation

การประยุกต์แบบนี้เราจะเรียกว่า Morphological Gradient โดยการนำอิมเมจจากการทำ Dilation มาลบกับอิมเมจที่ได้จากการทำ Erosion ซึ่งจะได้การทำกระบวนการหาขอบภาพ(Edge Detection)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 การทำ Morphological Gradient

3.4.5 รูปแบบของการหมุนที่ใช้ในRotate(Angle)

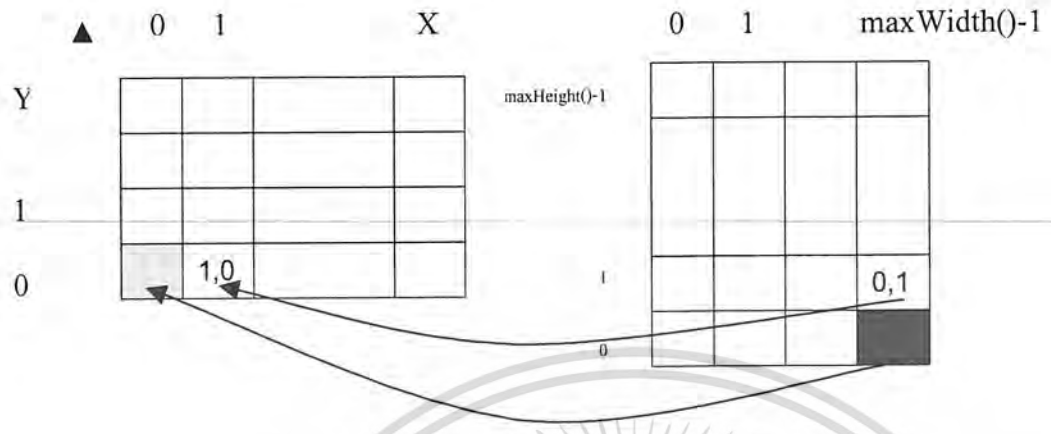
ตารางที่ 5 **การกำหนดรูปแบบการหมุนในฟังก์ชัน** PlgBlt

รูปแบบการหมุน	POINT[0]		POINT[1]		POINT[2]	
	x	y	x	y	x	y
ตามเข็มนาฬิกา	H*Sinθ	0	H*Sinθ+ W*Cosθ	W*Sinθ	0	H*Cosθ
ทวนเข็มนาฬิกา	0	W*Sinθ	W*Cosθ	0	H*Sinθ	W*Sinθ+ H*Cosθ

หมายเหตุ W=ความกว้างของอิมเมจ H = ความสูงของอิมเมจ และ θ คือ ค่าตั้งแต่ -90 ถึง 90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5.1 การหมุนตามเข็มนาฬิกา



รูปที่ 3.24 การหมุนตามเข็มนาฬิกา

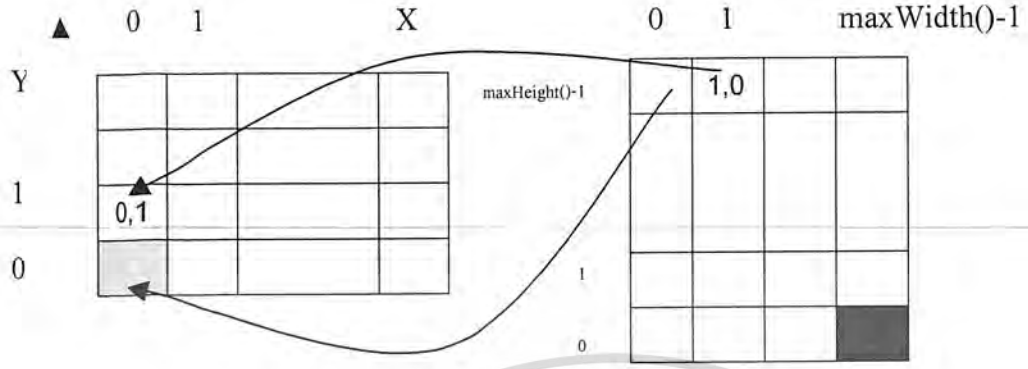
โดยมีอัลกอริทึม ดังนี้

```
BYTE *pSrc; BYTE *pNew;
        for (int x = 0; x<maxWidth; x++)
        {
                x2 = maxWidth-x-1;
                for (int y=0; y <maxHeight ; y++)
                {
                pSrc = (BYTE*)GetPixelAddress(y, x);
                pNew = (BYTE*)image.GetPixelAddress(x2,y);
                *pNew++ = *pSrc++;
                *pNew++ = *pSrc++;
                *pNew = *pSrc;
                }
        }
```

รูปที่ 3.25 อัลกอริทึมการหมุนตามเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5.2 การหมุนทวนเข็มนาฬิกา



รูปที่ 3.26 การหมุนทวนเข็มนาฬิกา

โดยมีอัลกอริทึม ดังนี้

```
BYTE *pSrc;BYTE *pNew;
        for (int y=0; y <maxHeight ; y++)
        {
                y2 = maxHeight-y-1;
                for (int x=0; x<maxWidth; x++)
                {
                pSrc = (BYTE*)GetPixelAddress(y , x);
                pNew = (BYTE*)image.GetPixelAddress(x,y2);
                *pNew++ = *pSrc++;
                *pNew++ = *pSrc++;
                *pNew = *pSrc;
                }
        }
```

รูปที่ 3.27 อัลกอริทึมการหมุนทวนเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.6 การพิมพ์และดูเอกสารก่อนการพิมพ์

การพิมพ์ออกทางเครื่องพิมพ์นั้นเราจะใช้ฟังก์ชัน OnDraw ที่เป็นฟังก์ชันในการแสดงผล โดยจะพิมพ์ออกมาจะมีขนาดเดียวกับที่แสดงผลบนจอคอมพิวเตอร์ (Monitor) หรือที่เราเรียกว่า What You See Is What You Get (WYSIWYG) แต่การพิมพ์ข้อมูลออกทางเครื่องพิมพ์ เครื่องพิมพ์บางเครื่องอาจจะไม่สนับสนุนฟังก์ชัน BitBlt หรือ StretchBlt ได้หรือไม่

เรามีข้อกำหนดว่า ขนาดอิมเมจในโหมดการแสดงผลหน้าจอมอนิเตอร์ ต้องมีขนาดเดียวกับในโหมดการพิมพ์ออกทางเครื่องพิมพ์ ดังนั้นเราต้องทราบความละเอียดของพิกเซลต่อนิ้วของจอมอนิเตอร์ในโหมดการวาดรูปปกติก่อน

เราจะต้องแสดงชื่อและนามสกุลและพาทของภาพออกทางเครื่องพิมพ์ซึ่งคล้ายกับการพิมพ์ในโหมดการดูเอกสารก่อนการพิมพ์ (Print Preview)

ต่อไปจะคำนวณความละเอียดต่อนิ้วทั้งความกว้างและความสูงของเครื่องพิมพ์ เพื่อนำมาใช้คำนวณความกว้างและความสูงของอิมเมจ ให้มีขนาดเดียวกับบนหน้าจอคอมพิวเตอร์

<u>วิธีการคำนวณหา Factor ความกว้างและความสูง</u>

- Factor ความกว้าง ได้จากค่าความกว้างต่อนิ้วของเครื่องพิมพ์ (พิกเซล/นิ้ว) หารด้วยค่าความกว้างต่อนิ้วของจอมอมิเตอร์(พิกเซล/นิ้ว)
- Factor ความสูง ได้จากค่าความสูงต่อนิ้วของเครื่องพิมพ์ (พิกเซล/นิ้ว) หารด้วยค่าความสูงต่อนิ้วของจอมอมิเตอร์(พิกเซล/นิ้ว)

3.4.7 การสร้างสเตตัส บาร์ และ การซูมอิมเมจ

สเตตัส บาร์ ในที่นี้หมายถึงการเพิ่มเพน (Pane) บนหน้าวินโดวส์ โดยเพนนี้จะแสดงรายละเอียดของอิมเมจ (ความกว้าง ความยาว และ ความลึก) ในเพนที่สอง และแสดงอัตราการซูมอยู่ในอัตราเท่าไร ในเพนที่ 3

ส่วนการซูมอิมเมจ สามารถเพิ่มอัตราขยายได้จาก 100 ถึง 1000 เปอร์เซ็นต์ หรือ เรียกว่า ซูม อิน (Zoom In) และลดอัตราขยายได้จาก 100 ถึง 25 เปอรเซ็นต์ หรือ เรียกว่า ซูม เอาท์ (Zoom Out)

การซูมอิมเมจเข้า(Zoom In) โดยฟังก์ชันนี้แยกการทำงานเป็น 2 กรณี

- กรณีแรก ถ้าค่าตัวแปรการซูมมีค่ามากกว่า 1 จะเพิ่มอัตราการขยายออกไปครั้งละ 0.5 เท่า ถ้าเพิ่มจนมีค่าเท่ากับ 10 แล้วจะไม่เพิ่มอีกต่อไป
- กรณีสอง ถ้าค่าตัวแปรการซูมน้อยกว่า 1 จะเพิ่มอัตราการซูมไปครั้งละ 0.25

การซูมอิมเมจออก(Zoom Out) โดยฟังก์ชันนี้แยกการทำงานเป็น 2 กรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กรณีแรก ถ้าค่าตัวแปรการซูมน้อยกว่า 2 จะลดอัตราการซูมไปครั้งละ 0.25
- กรณีสอง เมื่อลดไปถึง 0.25 แล้วจะไม่ลดอีกต่อไป

3.4.8 การทำUndo

การทำUndo จะช่วยอำนวยความสะดวกเมื่อต้องการข้อมูลต้นฉบับ หรือ ทำการย้อนการทำงานกลับไปเป็นเหมือนเดิมก่อนการเปลี่ยนแปลง โดยหลังจากการเรียกฟังก์ชันอื่นๆ แล้วทำให้ข้อมูลทางอิมเมจเปลี่ยนไป โดยไม่จำเป็นต้นเปิดไฟล์อิมเมจขึ้นมาใหม่

หลักการทำ Undo คือ บันทึกข้อมูลก่อนที่จะไปเรียกฟังก์ชันอื่นๆ เมื่อคลิกเมนูคำสั่ง Edit > Undo หรือ Ctrl+Z ก็จะเรียกข้อมูลที่เก็บขึ้นมาครั้งแรกมาใช้ ถ้าการทำ Undo มีหนึ่งระดับ แต่มีการทำฟังก์ชันเดิมซ้ำๆกัน ก็จำเป็นต้องเคลียร์ข้อมูลที่เก็บไว้ในครั้งแรกก่อน แล้วจึงบันทึกข้อมูลที่เก็บไว้ในครั้งแรกก่อน แล้วจึงบันทึกข้อมูลใหม่ลงไป

- สร้างตัวแปรใหม่ เพื่อใช้เก็บข้อมูลก่อนที่จะไปเรียกฟังก์ชันอื่นทำงาน
- ถ้าก่อนหน้านี้มีการบันทึก ข้อมูลไว้แล้ว ต้องลบข้อมูลเก่าออกเสีย
- เอาข้อมูลที่เก็บไว้มาแทนที่ข้อมูลเดิม เมื่อมีการเลือกคำสั่ง Undo
- อัพเดทวิวอีกครั้ง และอัพเดทเมนูคำสั่ง Undo ถ้าค่าที่เก็บไว้เป็นค่า NULL เมนูคำสั่ง Undo จะใช้ไม่ได้

3.4.9 การสร้าง Histogram และการคำนวณสถิติของอิมเมจ

การดูฮิสโทแกรมและค่าสถิติของอิมเมจนั้นจะสามารถอธิบายคุณลักษณะของอิมเมจ ยกตัวอย่างเช่น การนำมาวิเคราะห์ในการมองของเครื่องจักร (Machine Vision) เช่น หุ่นยนต์ในอุตสาหกรรม

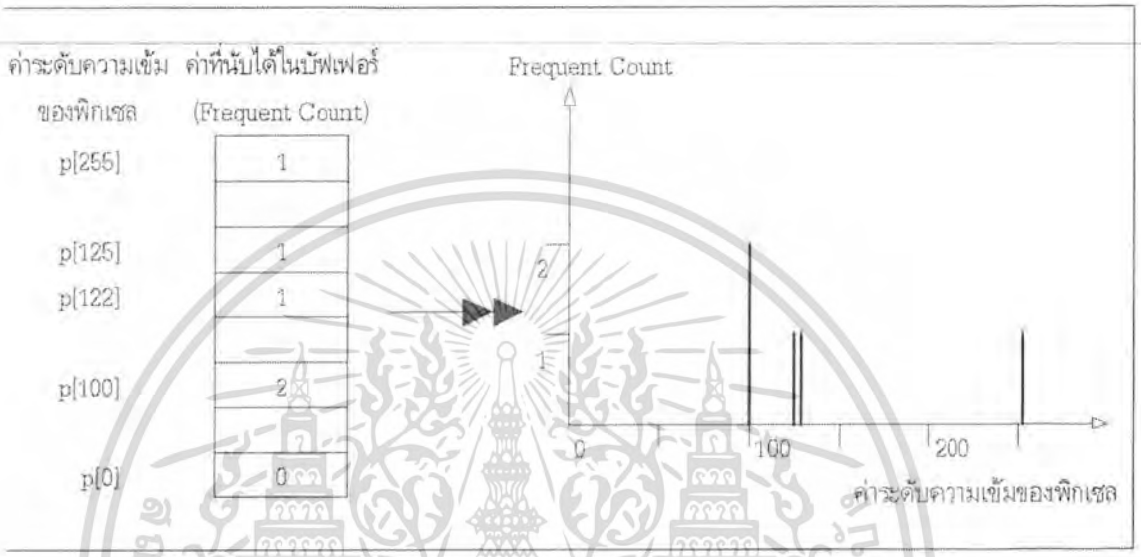
การสร้าง Histogram ซึ่งจะเป็นการพลอตกราฟความถี่ที่เกิดขึ้นซ้ำๆ กันในแต่ละระดับตั้งแต่ 0 ถึง 255 โดยแกนตั้ง (Y) จะเป็นค่าความถี่ที่เกิดขึ้น และ แกนนอน (X) จะเป็นค่าระดับความเข้มของพิกเซลอิมเมจ

เราจะนับความถี่ที่เกิดขึ้นซ้ำๆ กัน โดยมีวิธีการนับดังนี้ โดยเราจะใช้ค่าบัฟเฟอร์เก็บค่าแม่สี RGB เพื่อเก็บค่าการนับตั้งแต่ระดับ 0 ถึง 255 ในแต่ละระดับนั้นๆ ถ้ามีการเกิดขึ้นซ้ำๆ กัน ก็จะนับค่านั้นเพิ่มขึ้นไปครั้งละ 1

ตัวอย่างเช่น อิมเมจหนึ่งมี 256 ระดับ ละมีจำนวนพิกเซลเท่ากับ 5 พิกเซล ใน 5 พิกเซลนั้นมีค่าเรียงลำดับต่างๆ ดังนี้ 100,125,255,122 และ 100 ดังนั้น ค่าที่นับความถี่ที่เกิดขึ้นซ้ำๆ กัน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ระดับ 100 จะมีความถี่เท่ากับ 2
- ที่ระดับ 125,255,122 เป็นอย่างละ 1
- นอกนั้นเป็น 0 หมด



รูปที่ 3.28 การพลอตกราฟ Histogram

3.4.10 วิธีการหาค่าเฉลี่ย (Mean) ของอิมเมจสีต่าง ๆ

โดยการใช้ตัวแปรเพื่อเก็บค่าผลรวมของแต่ละแม่สี เพื่อเก็บค่าผลรวมของค่าระดับพิกเซลทั้งหมด แล้วหารด้วยจำนวนพิกเซลทั้งหมด ตัวอย่างเช่น อิมเมจหนึ่งมีจำนวนพิกเซลเป็น 100,125,255,122 และ 100 ดังนั้นค่าผลรวมระดับพิกเซลทั้งหมดเท่ากับ 100 +125 + 255 + 122 + 100 = 702 และจำนวนพิกเซลทั้งหมดเท่ากับ 5 ดังนั้นค่าเฉลี่ยของอิมเมจเท่ากับ 702/5 = 140.5

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

รูปที่ 3.29 การหาค่าเฉลี่ย

โดยที่ X bar คือ ค่าเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N คือ จำนวนพิกเซลทั้งหมด

Xi คือ ค่าระดับความเข้มพิกเซลที่ i

3.4.11 วิธีการหาค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) **ของอิมเมจ**

ทำได้โดยการหาค่า Variance ก่อนแล้วจึงถอดรากที่สอง ก็จะได้ค่าเบี่ยงเบนมาตรฐานหรือSD การหาค่า Variance ทำได้โดยการหาผลรวมการยกกำลังสองของค่าระดับพิกเซลนั้นลบด้วยค่าเฉลี่ยของอิมเมจทั้งหมด แล้วหารด้วยจำนวนพิกเซลทั้งหมด

$$Var = \frac{1}{N}\sum_{i=1}^{N}\left(X_i - \overline{X}\right)^2$$

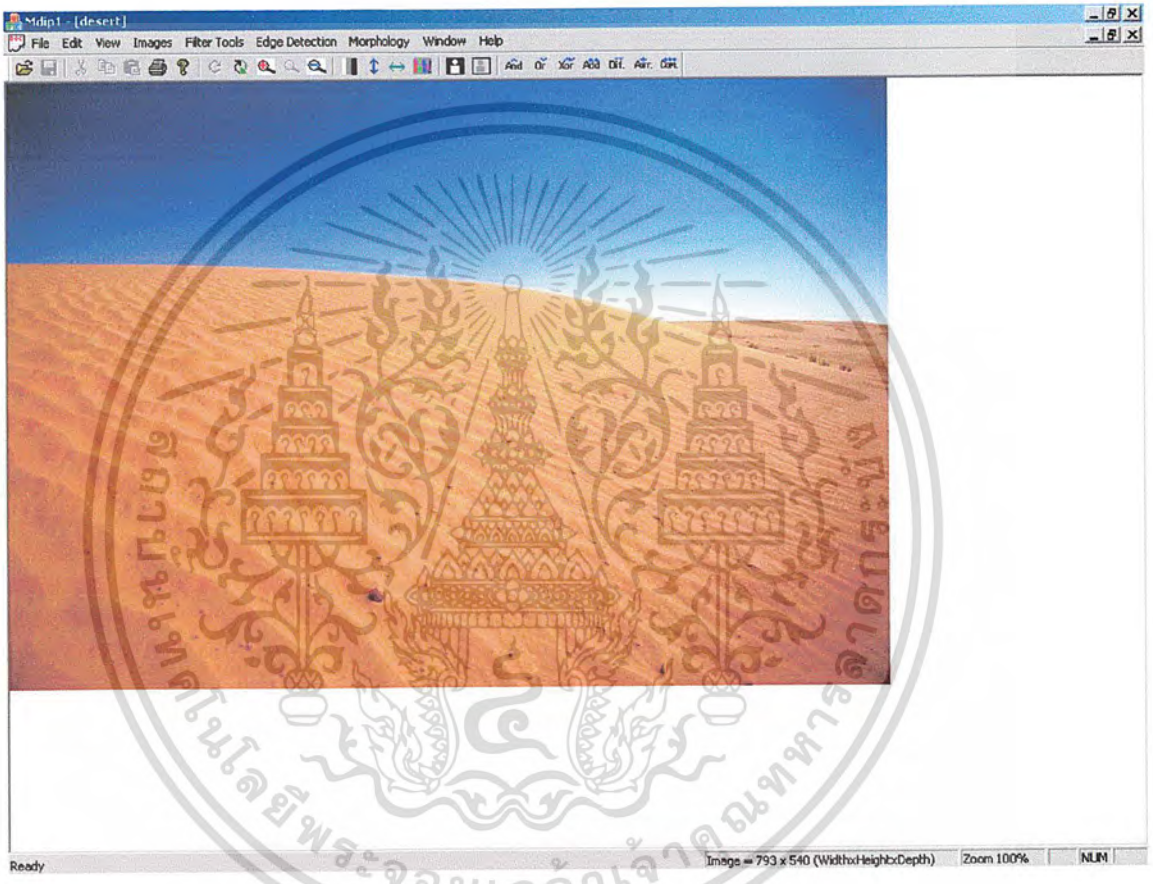
$$\sigma = \sqrt{Var}$$

รูปที่ 3.30 การหาค่าเบี่ยงเบนมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ลักษณะการใช้งานของโปรแกรม

โปรแกรม Mdip1 นั้นเป็นแอปพลิเคชั่นบนวินโดวส์เมื่อเราเปิดโปรแกรมขึ้นมาแล้วให้เราไปที่เมนูของโปรแกรมเพื่อเข้าเมนู File>Open หรือ จะทำการเลือกที่แถบทูลบาร์ที่มีรูปแฟ้มก็ได้เพื่อทำการเปิดไฟล์

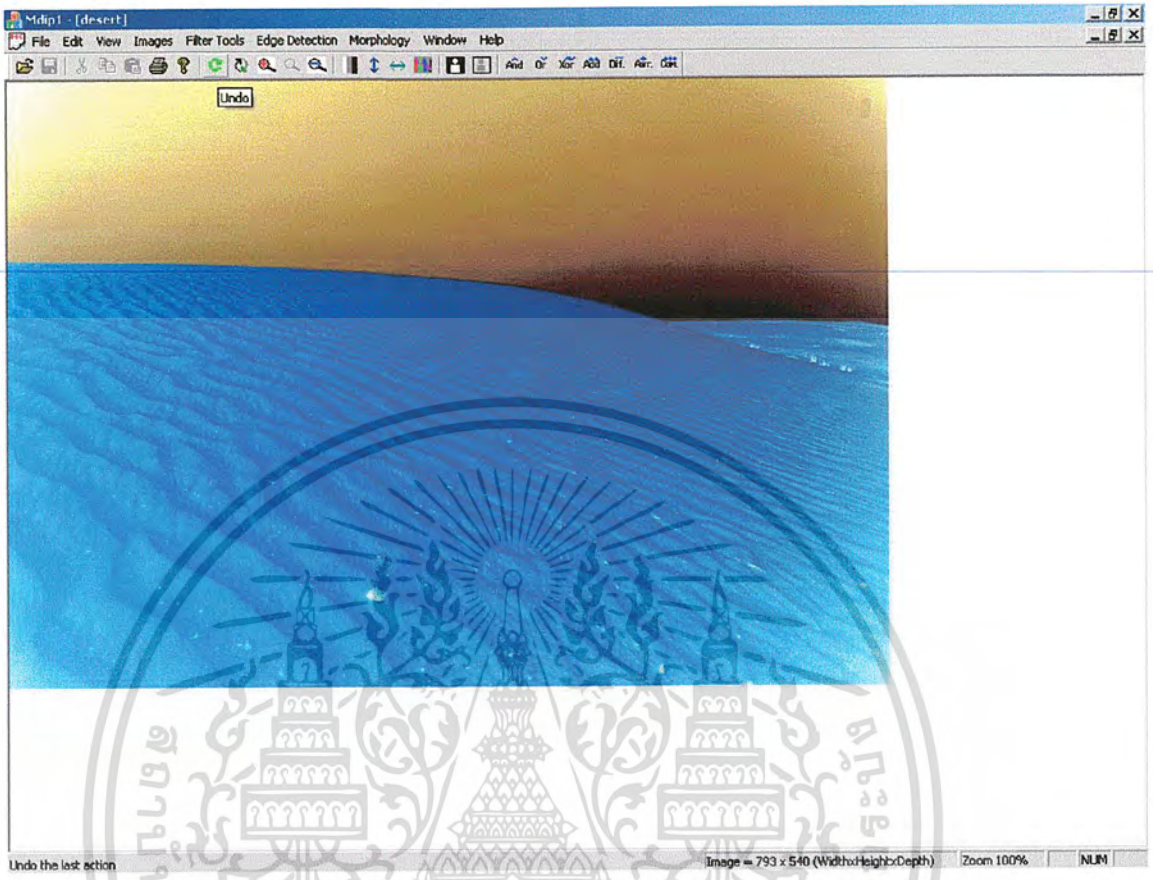


รูปที่ 3.35 แสดงขั้นตอนการแสดงภาพ

เมื่อเราเปิดโปรแกรมขึ้นมาครั้งแรกเราจะพบว่า ตัวโปรแกรมจะประกอบไปด้วยเมนูต่างๆ ตามที่ได้กล่าวไว้ในบทข้างต้น นอกจากนั้นยังมี Tool Bar ช่วยในการสะดวกต่อการทำงานซึ่ง Tool Bar นี้สามารถอยู่ติดกับเมนูหรือสามารถเคลื่อนไปภายในหน้าต่างโปรแกรมซึ่งขึ้นอยู่กับความต้องการของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำภาพInvert

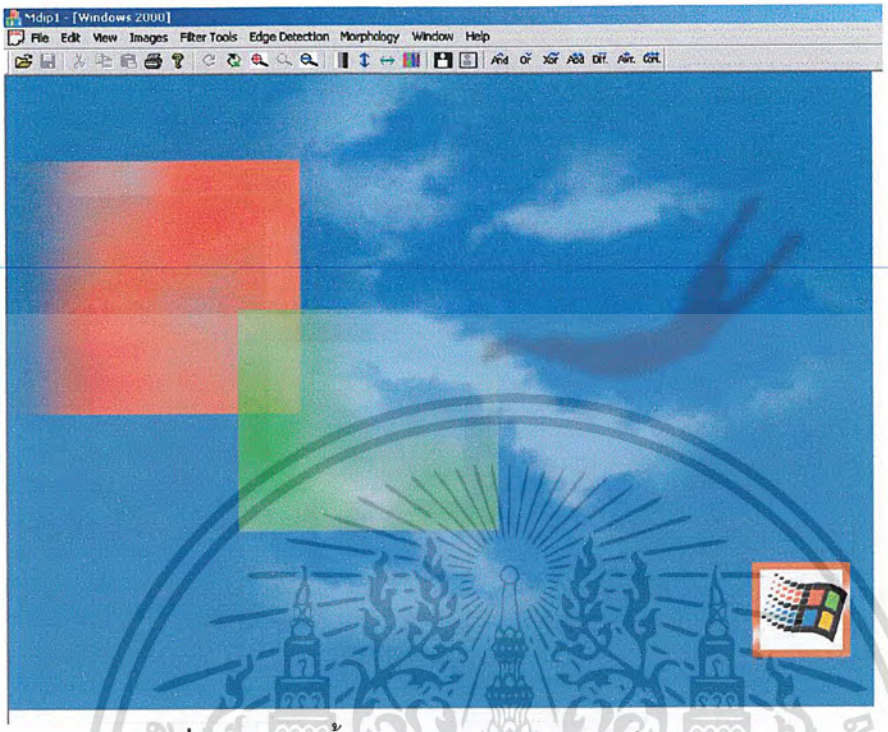


รูปที่ 3.36 แสดงขั้นตอนการแสดงการทำภาพInvert

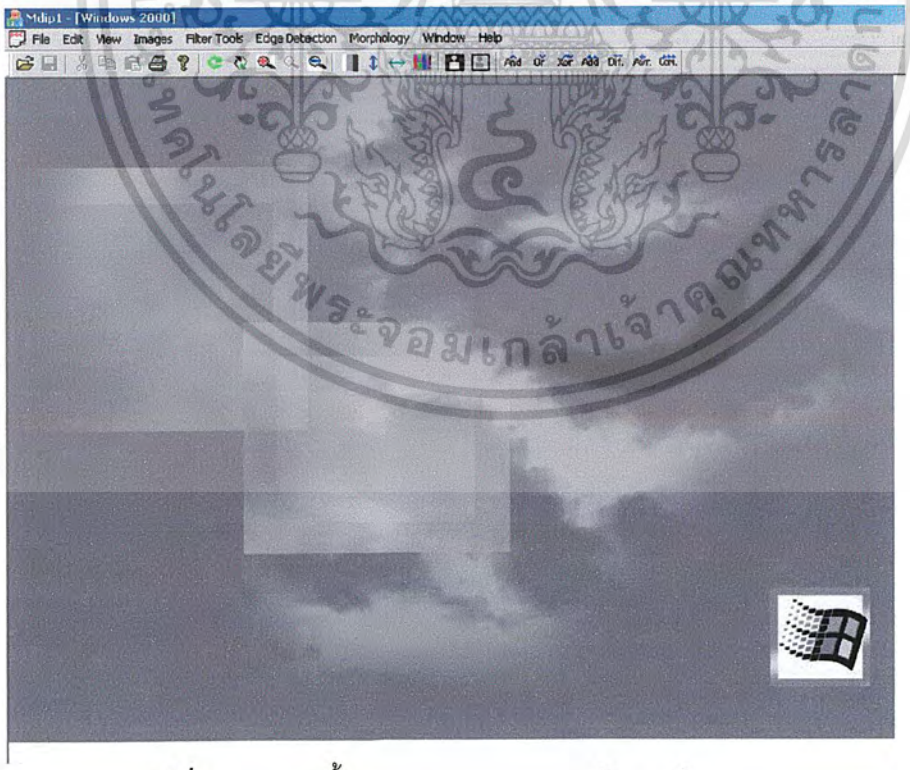
ให้เราไปที่เมนูของโปรแกรมเพื่อเข้าเมนู Images > Invert Color ซึ่งจะแสดงผลการทำงานทันทีซึ่งภาพที่ออกมาจะเป็นเหมือนภาพในฟิลม์ถ่ายรูปหรือภาพแบบNegative

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำ GrayScale



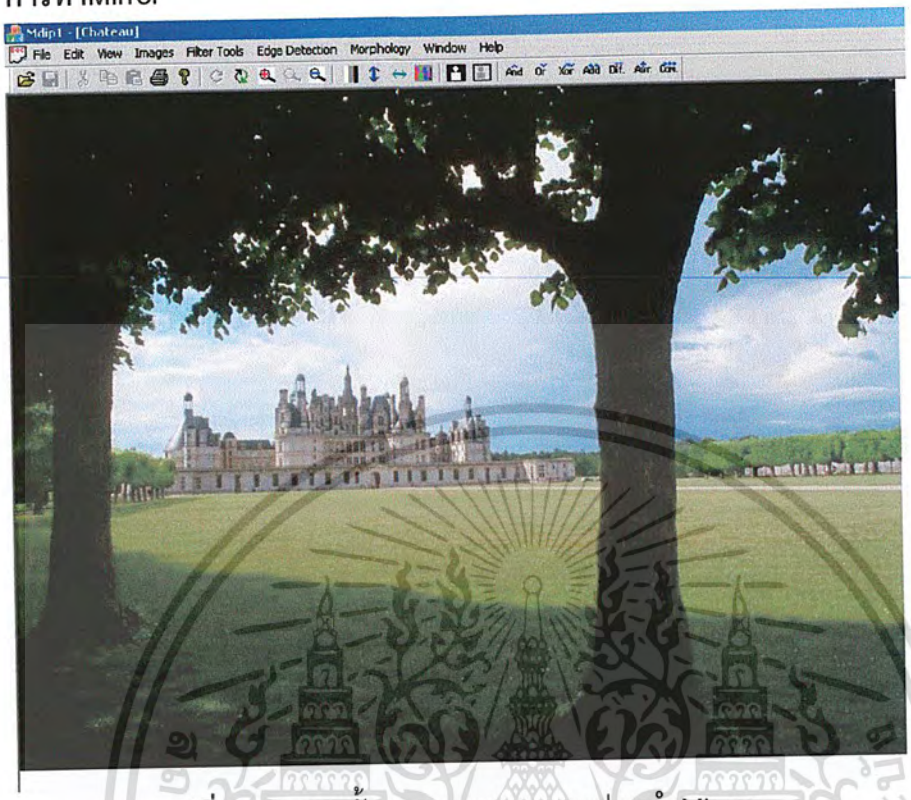
รูปที่ 3.37 แสดงขั้นตอนการแสดงภาพก่อนทำGrayScale



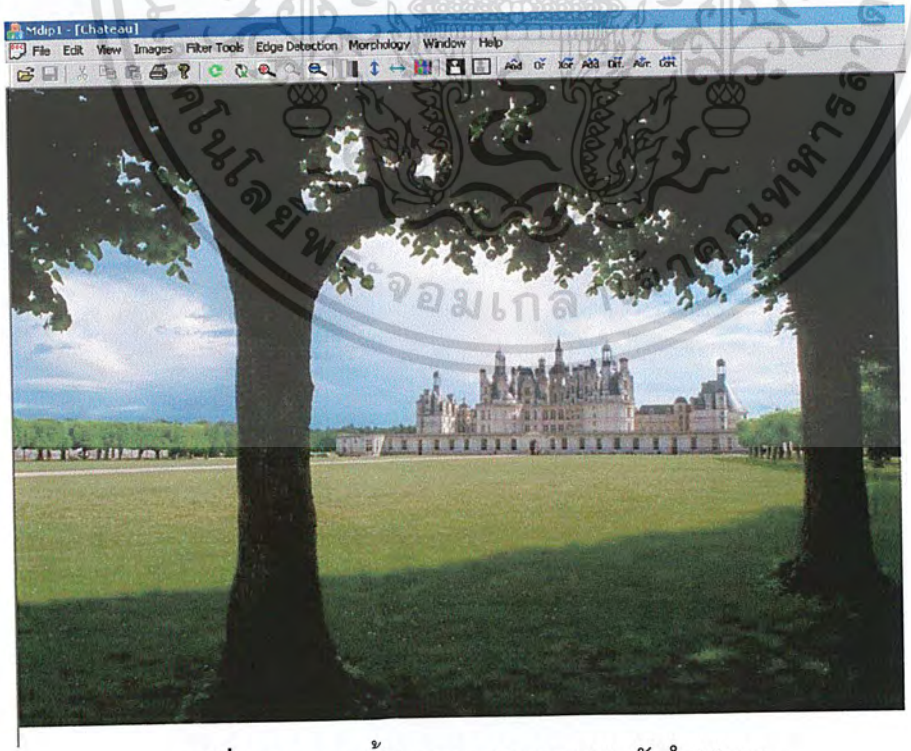
รูปที่ 3.38 แสดงขั้นตอนการแสดงภาพหลังจากทำGrayScale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำMirror



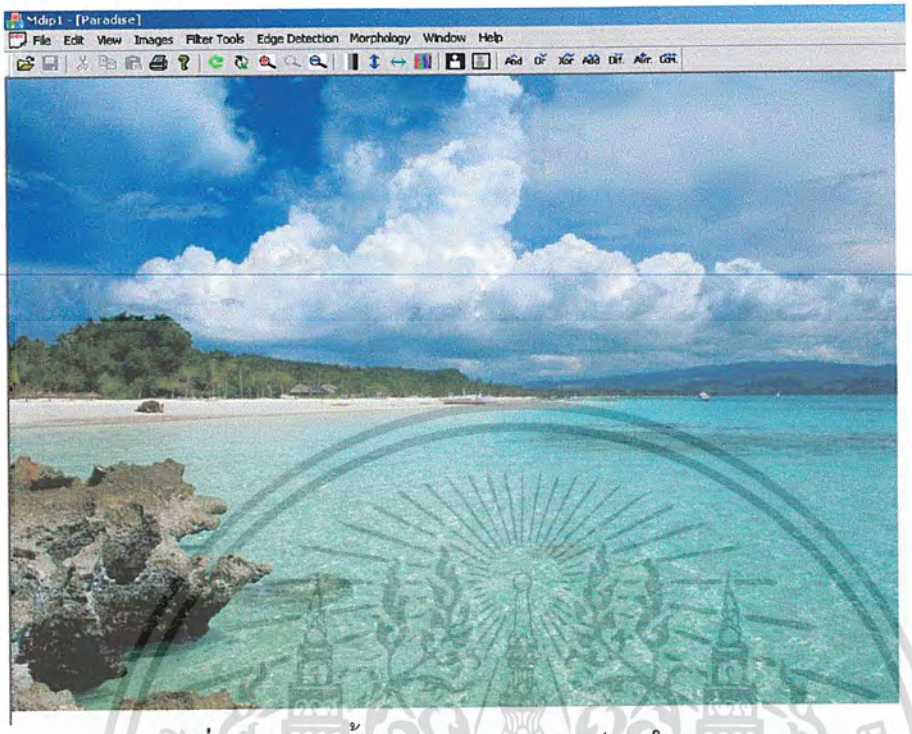
รูปที่ 3.39 แสดงขั้นตอนการแสดงภาพก่อนทำMirror



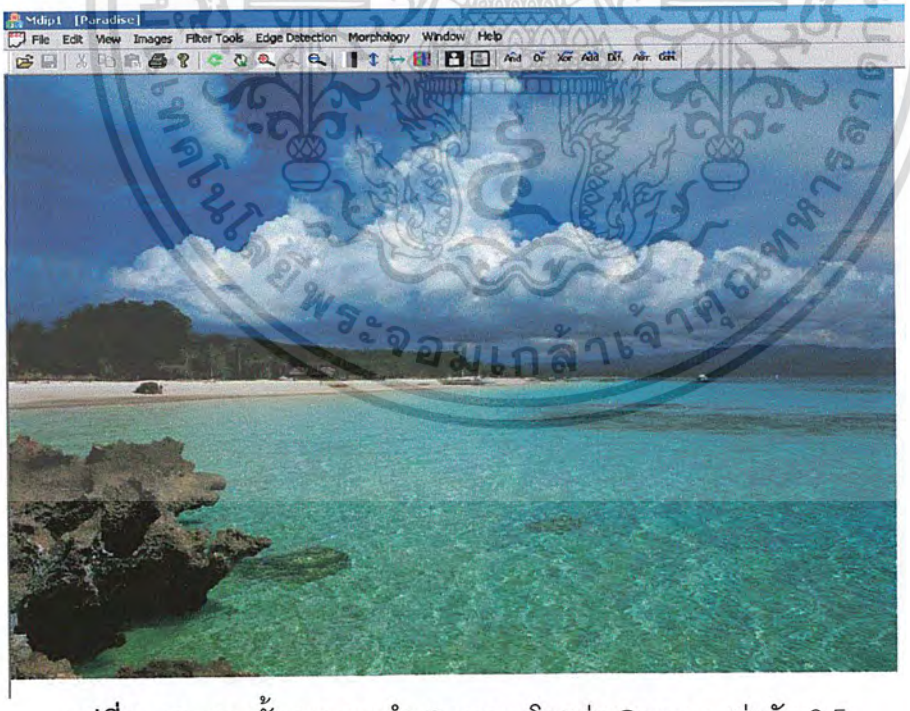
รูปที่ 3.40 แสดงขั้นตอนการแสดงภาพหลังทำMirror

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำGamma

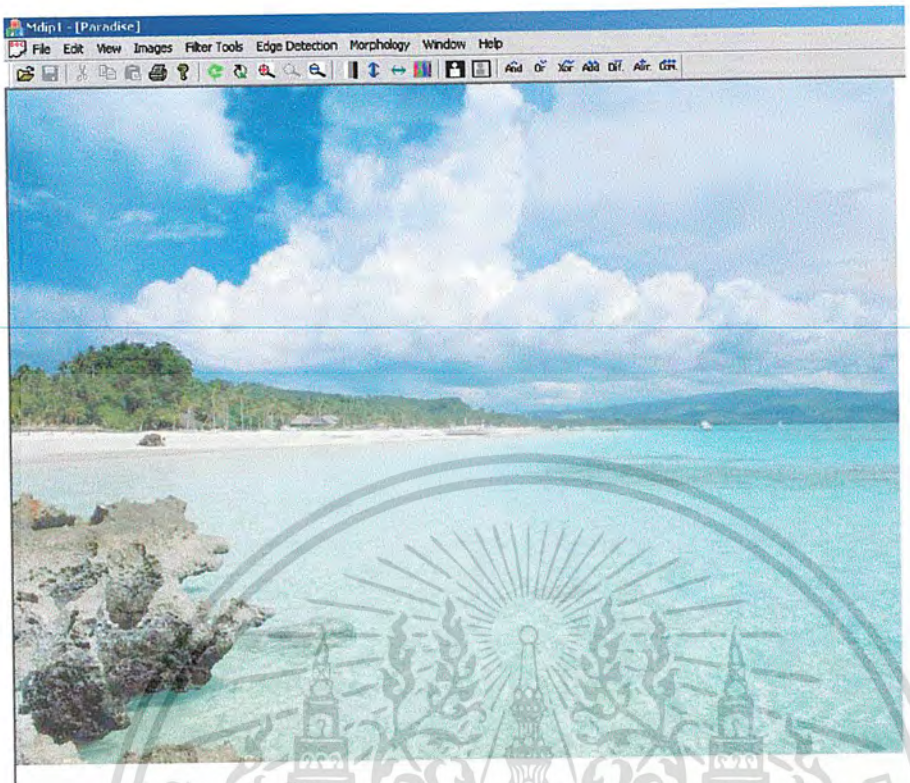


รูปที่ 3.41 แสดงขั้นตอนการแสดงภาพก่อนทำ Gamma



รูปที่ 3.42 แสดงขั้นตอนการทำ Gamma โดยค่า Gamma เท่ากับ 0.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

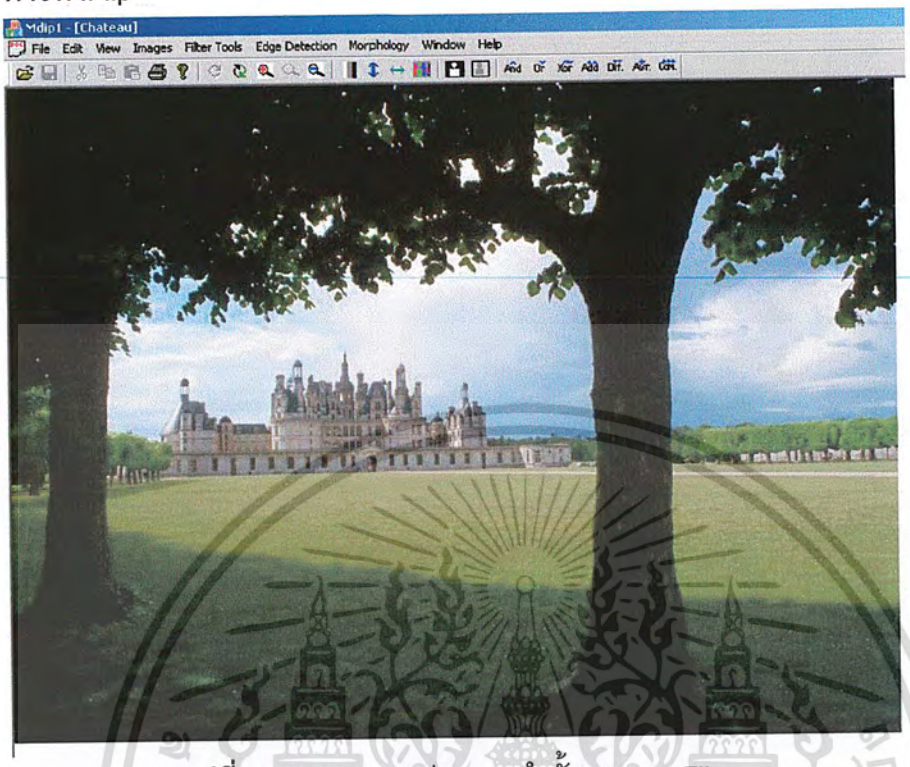


รูปที่ 3.43 แสดงขั้นตอนการทำ Gamma โดยค่า Gamma เท่ากับ 2

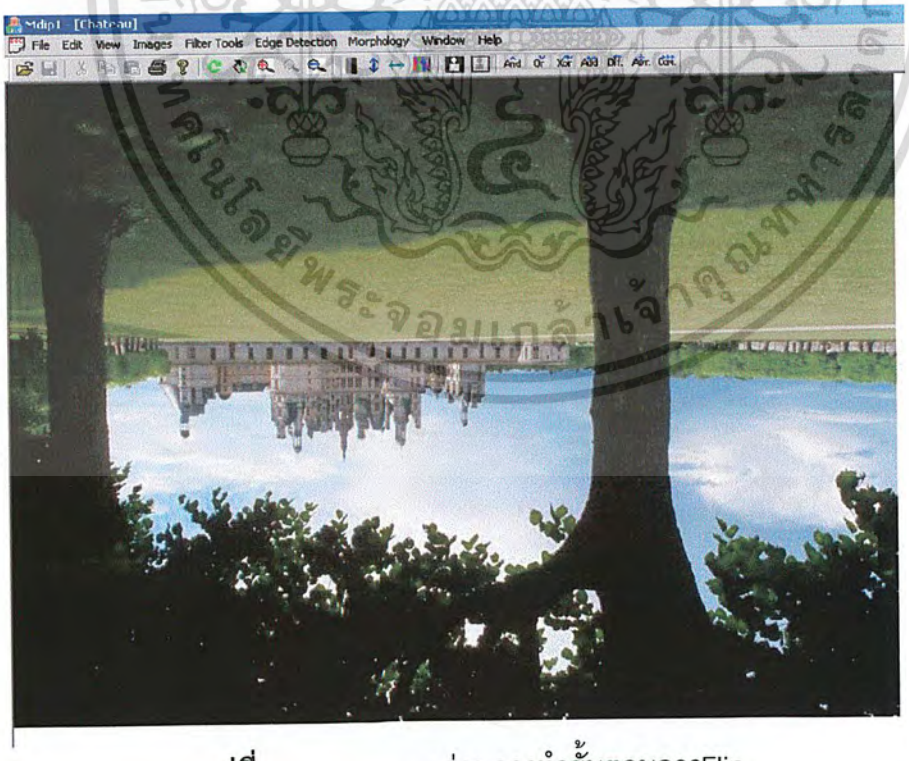
โดยผู้ใช้จะทำการอินพุตค่าของแกมมาเข้าไปในไดอะล็อค ถ้าค่าแกมมามีค่าต่ำภาพที่ออกมาจะมีความสว่างน้อยลง เหมาะกับการใช้กับภาพที่ถ่ายออกมาแล้วมีความสว่างมาก เราจึงต้องทำให้ภาพนั้นมีความสว่างมากขึ้น ในทางกลับกันถ่ภาพที่ถ่ายออกมามีความมืดมาก เราจะทำการแก้ไขด้วยการอินพุตค่าแกมมาค่ามากๆเข้าไป จะทำให้ภาพที่ได้ออกมานั้นมีความสว่างมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำFlip



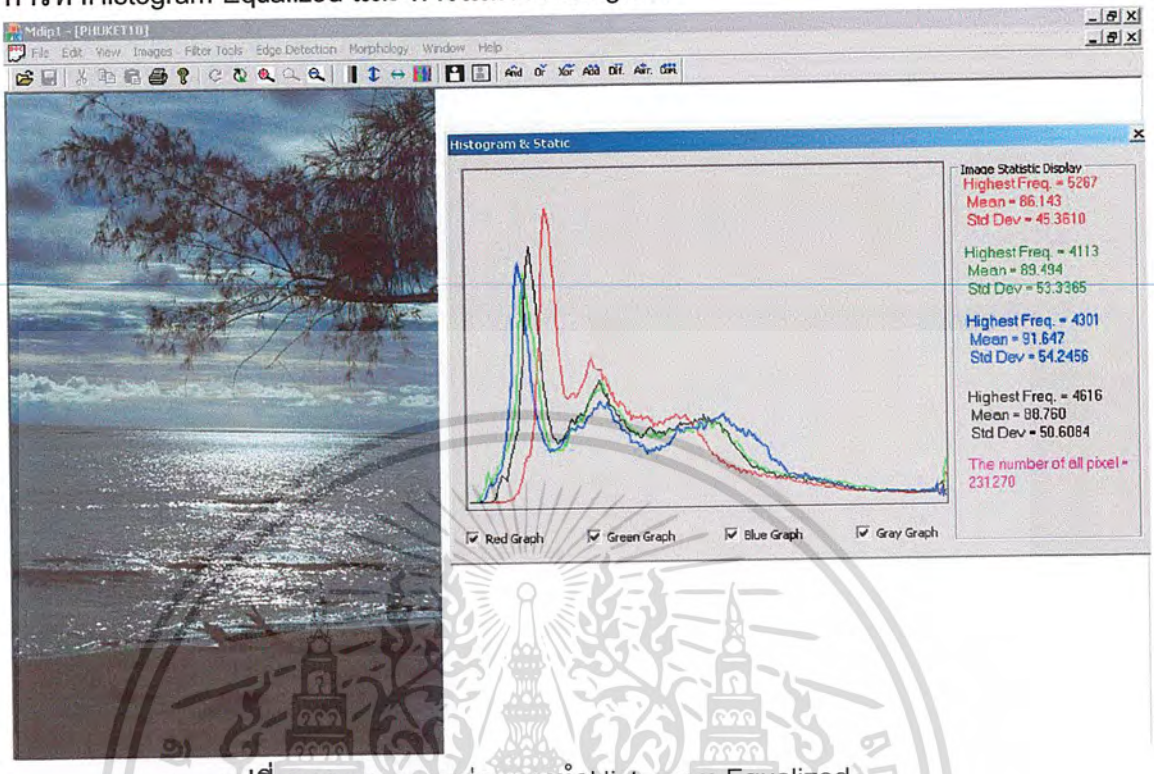
รูปที่ 3.44 แสดงภาพก่อนการทำขั้นตอนการFlip



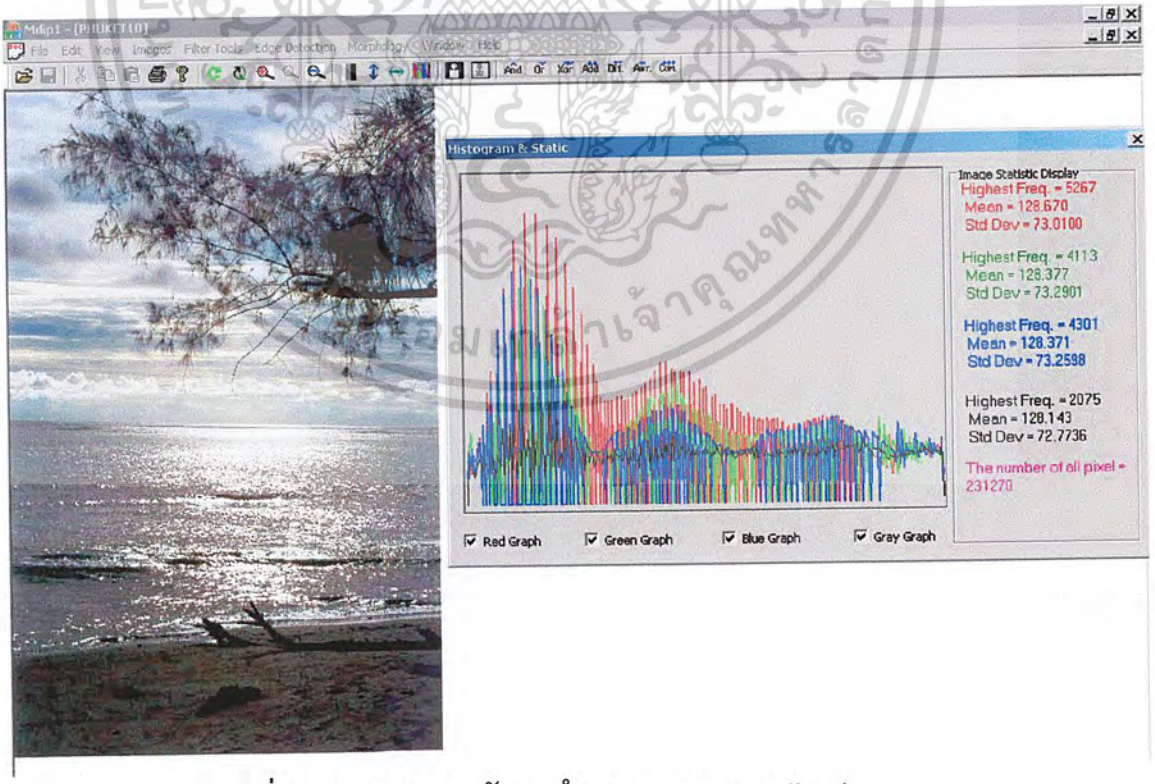
รูปที่ 3.45 แสดงภาพก่อนการทำขั้นตอนการFlip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำHistogram Equalized และ การแสดง Histogram



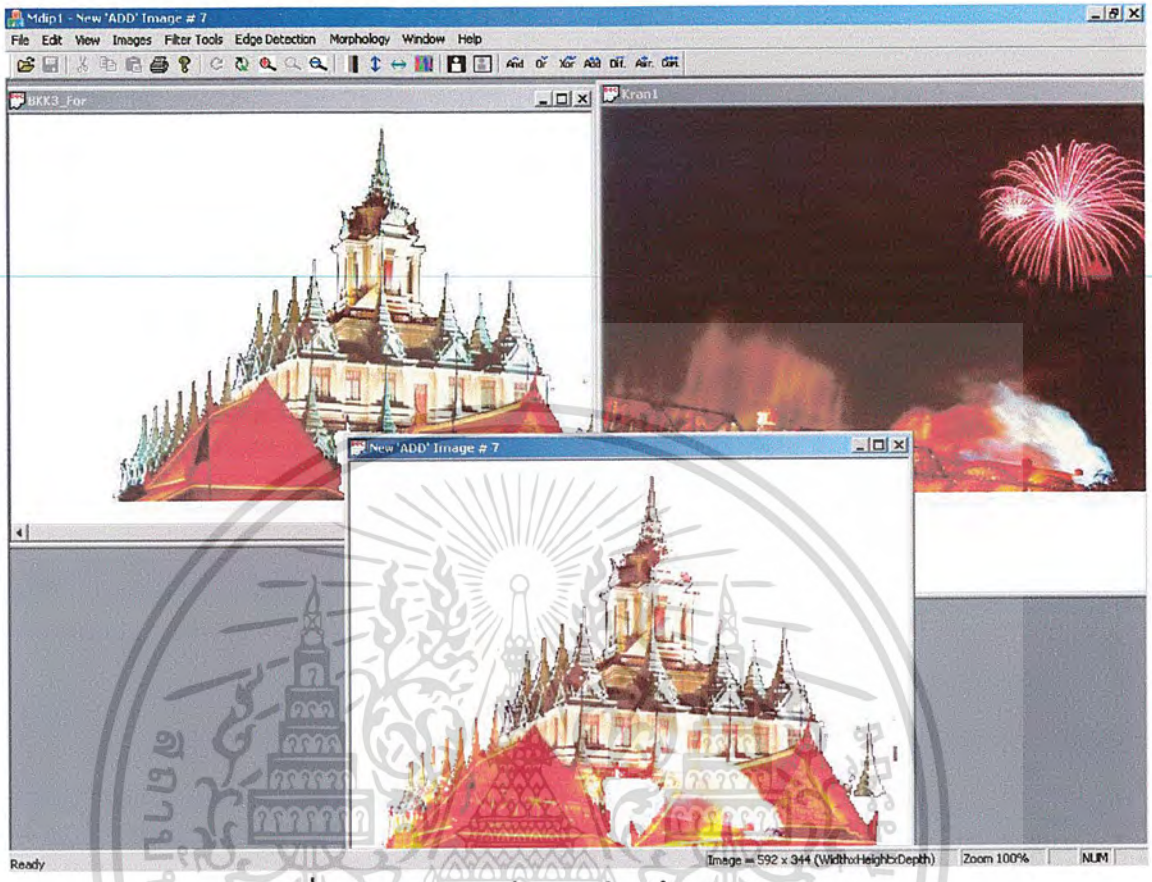
รูปที่ 3.46 แสดงภาพก่อนการทำHistogram Equalized



รูปที่ 3.47 แสดงภาพหลังการทำHistogram Equalized

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างภาพใหม่ด้วยการใช้ลอจิคAnd

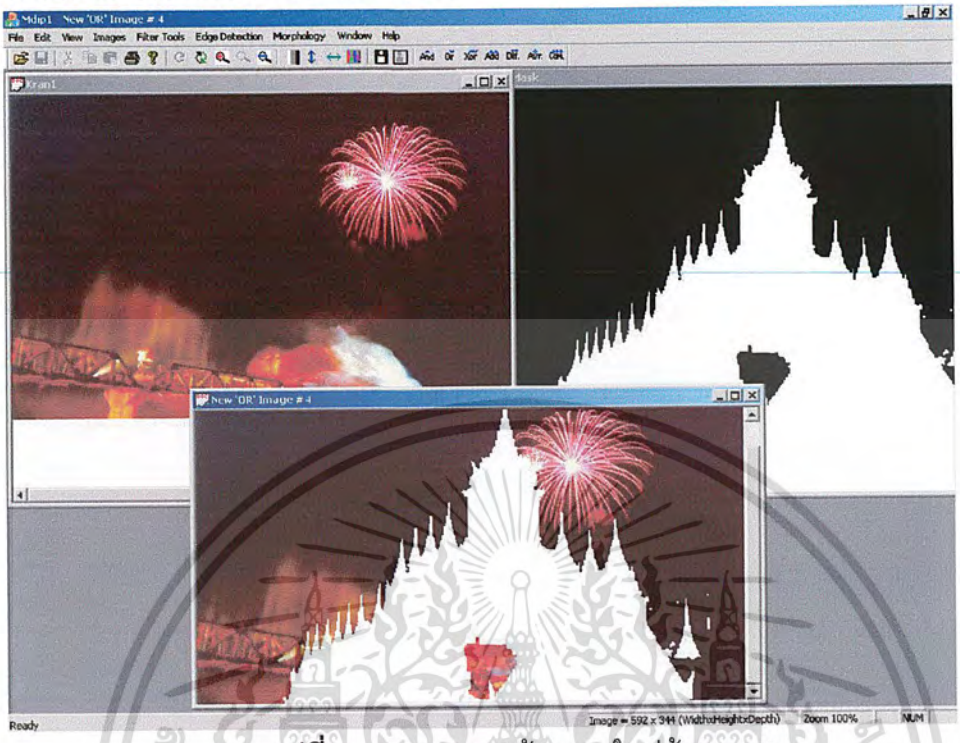


รูปที่ 3.48 แสดงการสร้างภาพใหม่ด้วยการAnd

โดย 2 ภาพบนแอนด์กันได้ภาพล่างเป็นผลลัพธ์ ซึ่งภาพใหม่ที่ได้ออกมาจะยังคงมีภาพเดิมผสมกันขึ้นกับลอจิคที่ใช้ ซึ่งผู้ใช้สามารถบันทึก (Save) ไฟล์รูปภาพใหม่ที่เกิดจากการทำลอจิคได้

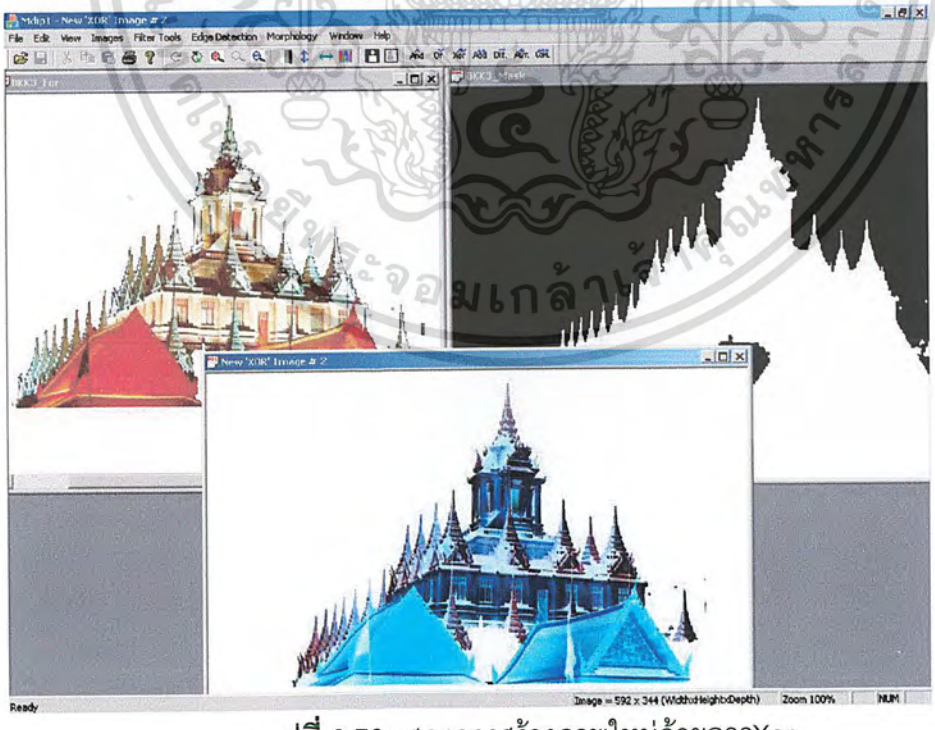
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างภาพใหม่ด้วยการใช้ลอจิคOr



รูปที่ 3.49 แสดงการสร้างภาพใหม่ด้วยการOr

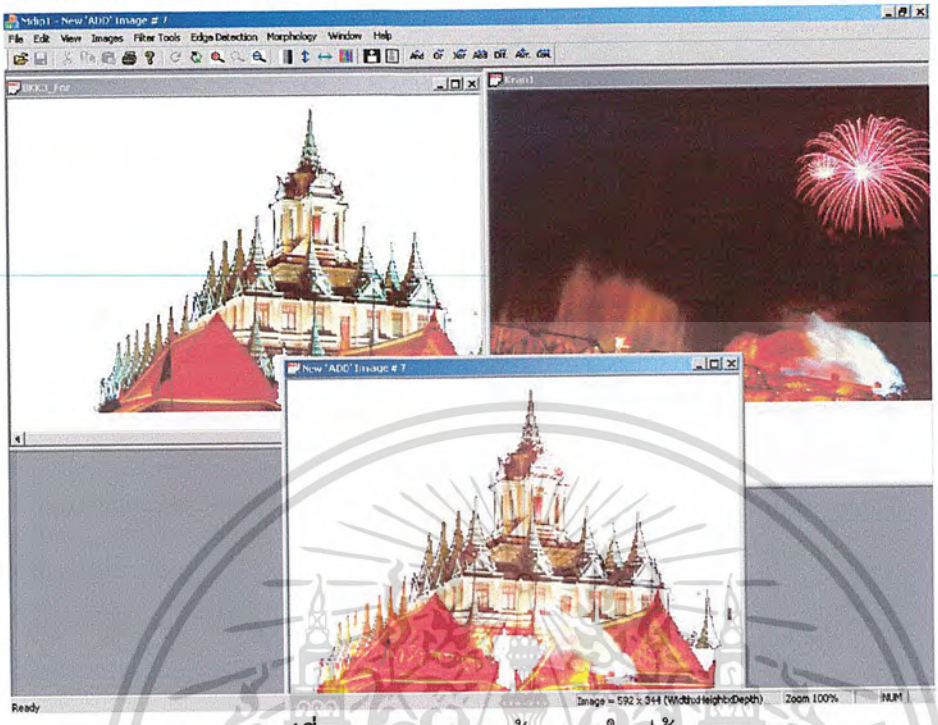
การสร้างภาพใหม่ด้วยการใช้ลอจิคXor



รูปที่ 3.50 แสดงการสร้างภาพใหม่ด้วยการXor

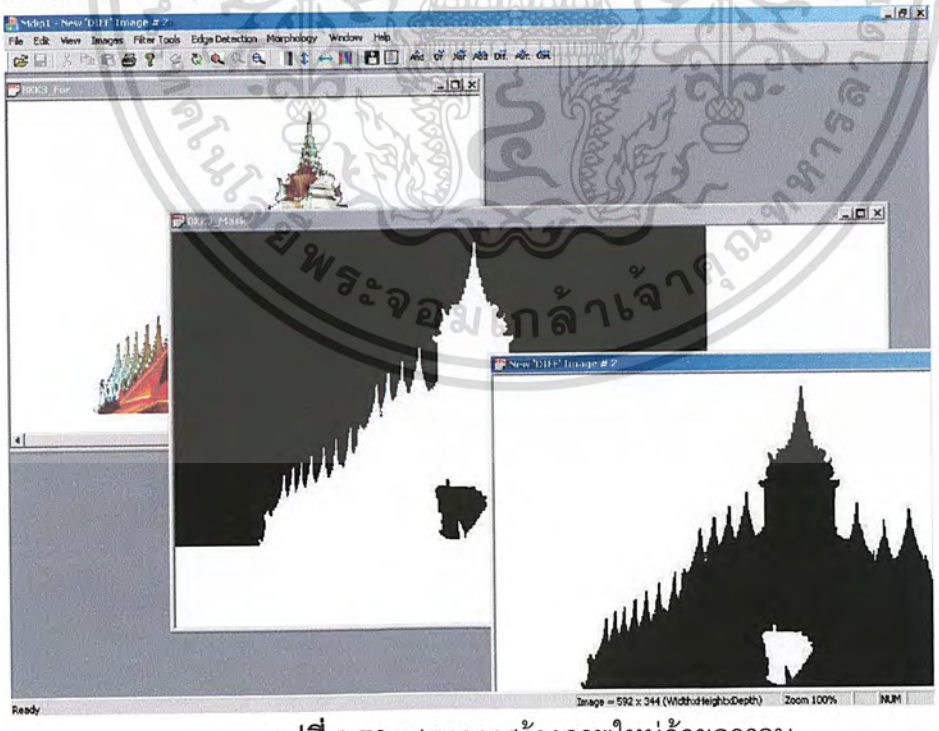
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างภาพใหม่ด้วยการใช้การบวก



รูปที่ 3.51 แสดงการสร้างภาพใหม่ด้วยการบวก

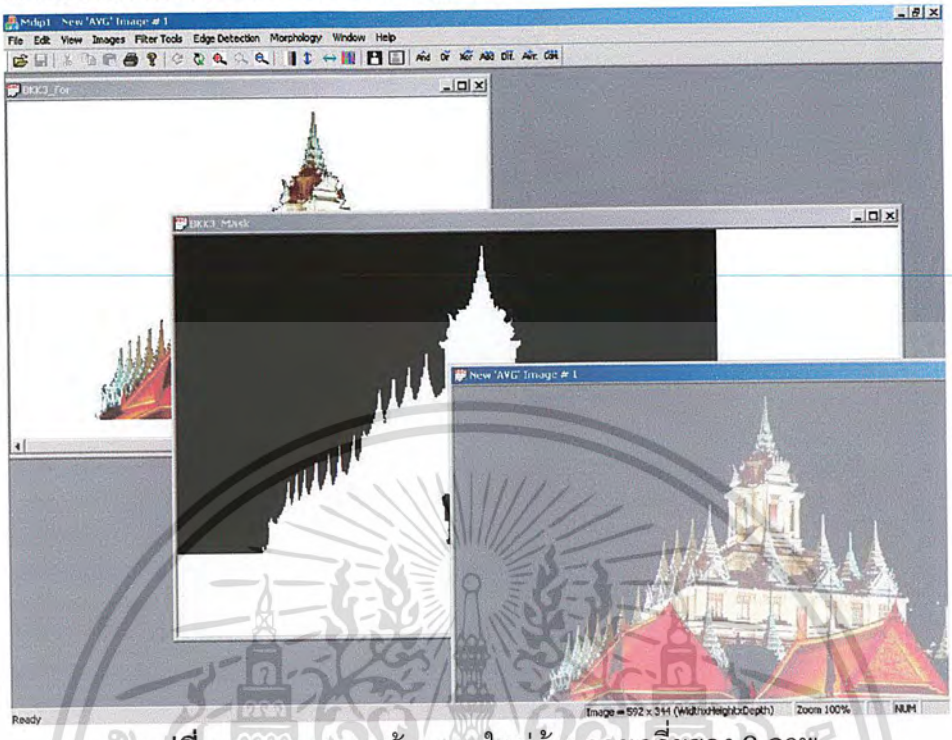
การสร้างภาพใหม่ด้วยการใช้การลบ



รูปที่ 3.52 แสดงการสร้างภาพใหม่ด้วยการลบ

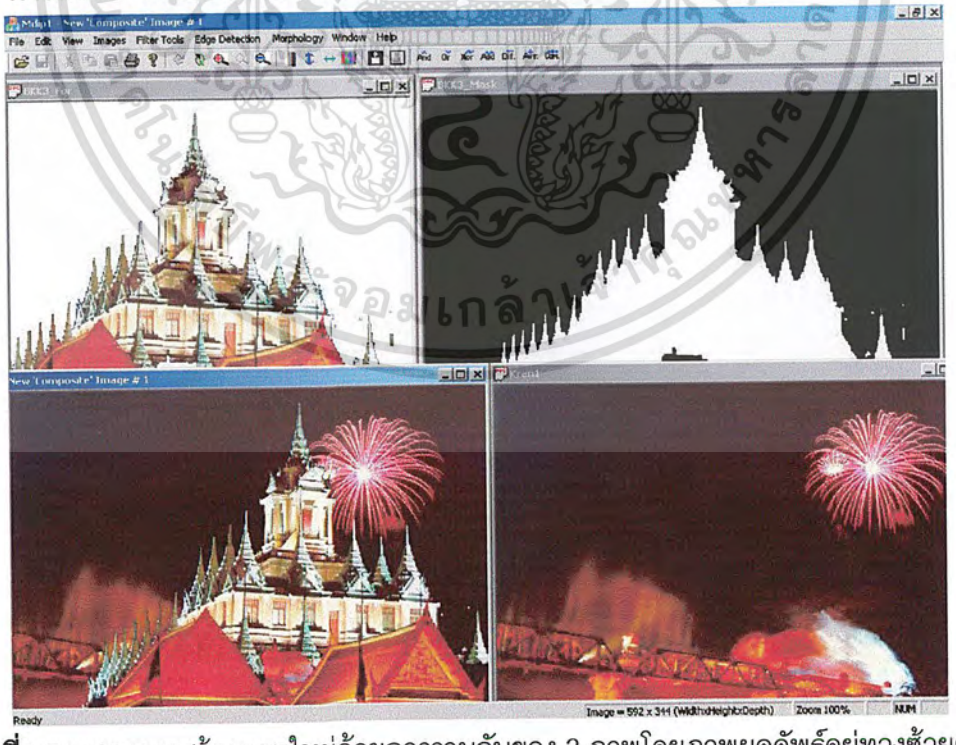
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างภาพใหม่ด้วยการใช้การเฉลี่ยภาพ



รูปที่ 3.53 แสดงการสร้างภาพใหม่ด้วยการเฉลี่ยของ 2 ภาพ

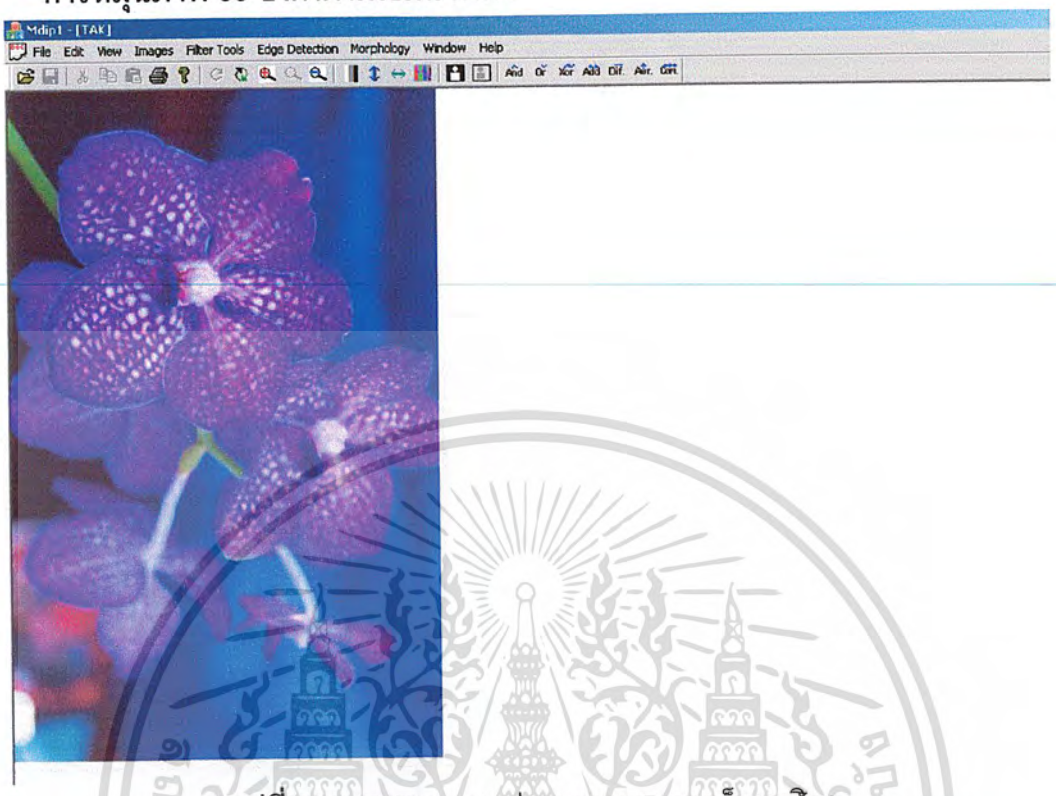
การสร้างภาพใหม่ด้วยการใช้การรวมภาพ 3 ภาพ



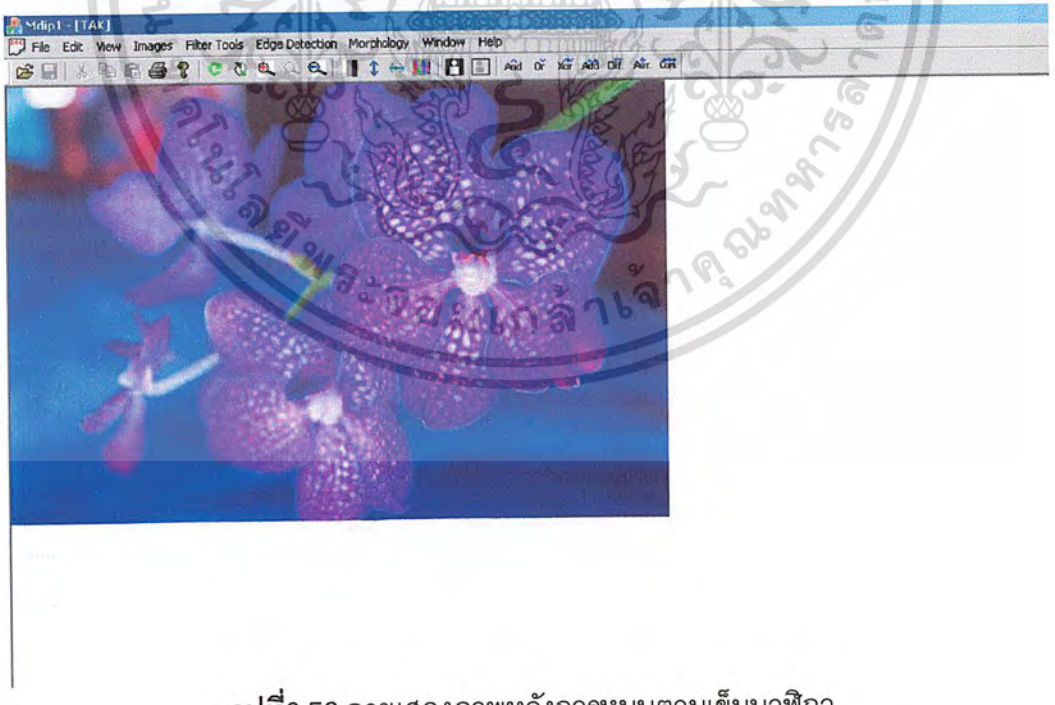
รูปที่3.54แสดงการสร้างภาพใหม่ด้วยการรวมกันของ 3 ภาพโดยภาพผลลัพธ์อยู่ทางซ้ายล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหมุนภาพ 90 องศาตามเข็มนาฬิกา



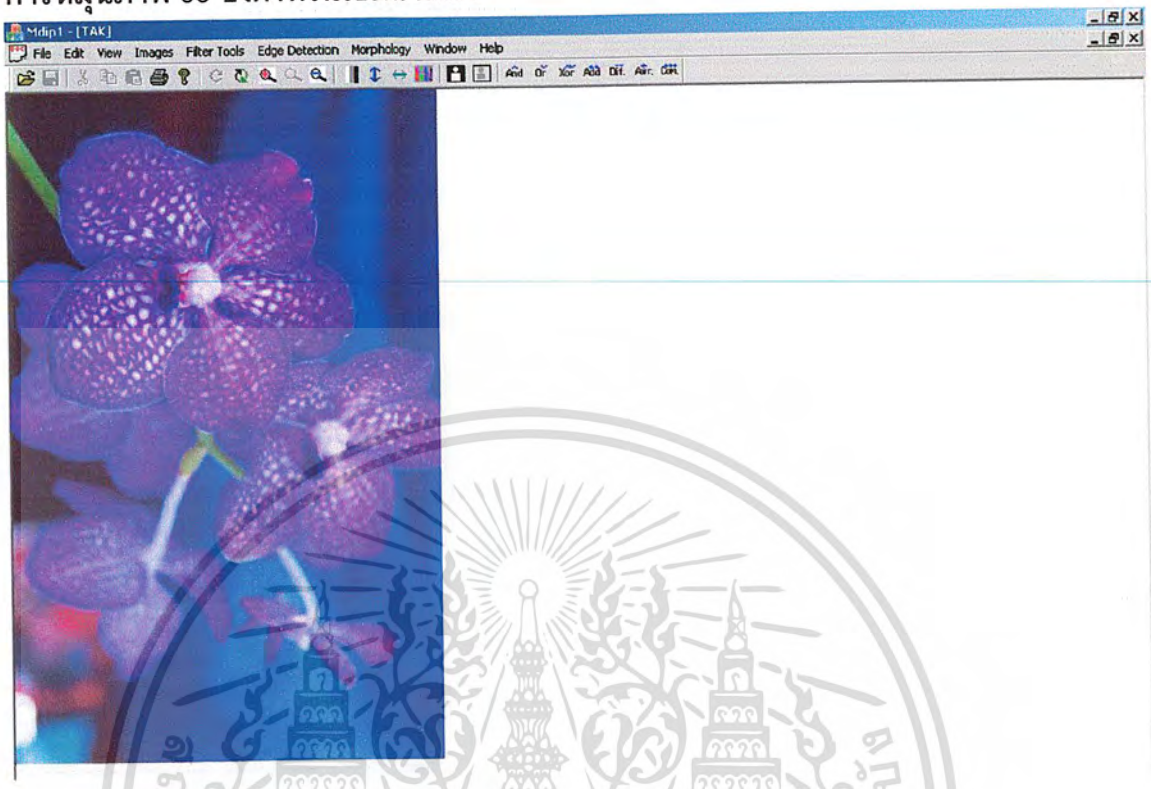
รูปที่3.55 การแสดงภาพก่อนการหมุนตามเข็มนาฬิกา



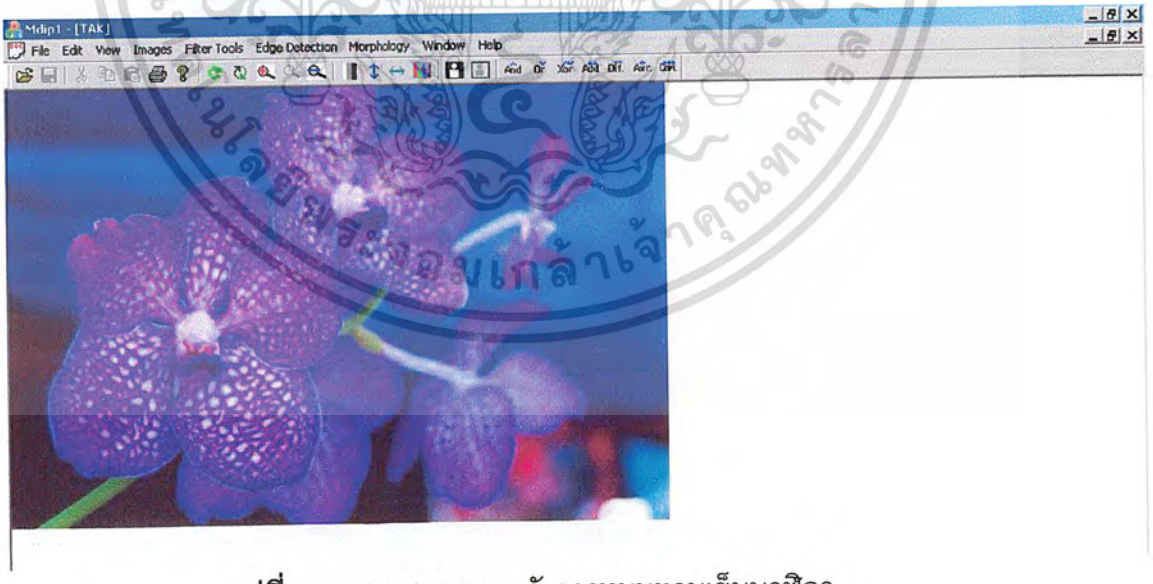
รูปที่3.56 การแสดงภาพหลังการหมุนตามเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหมุนภาพ 90 องศาทวนเข็มนาฬิกา



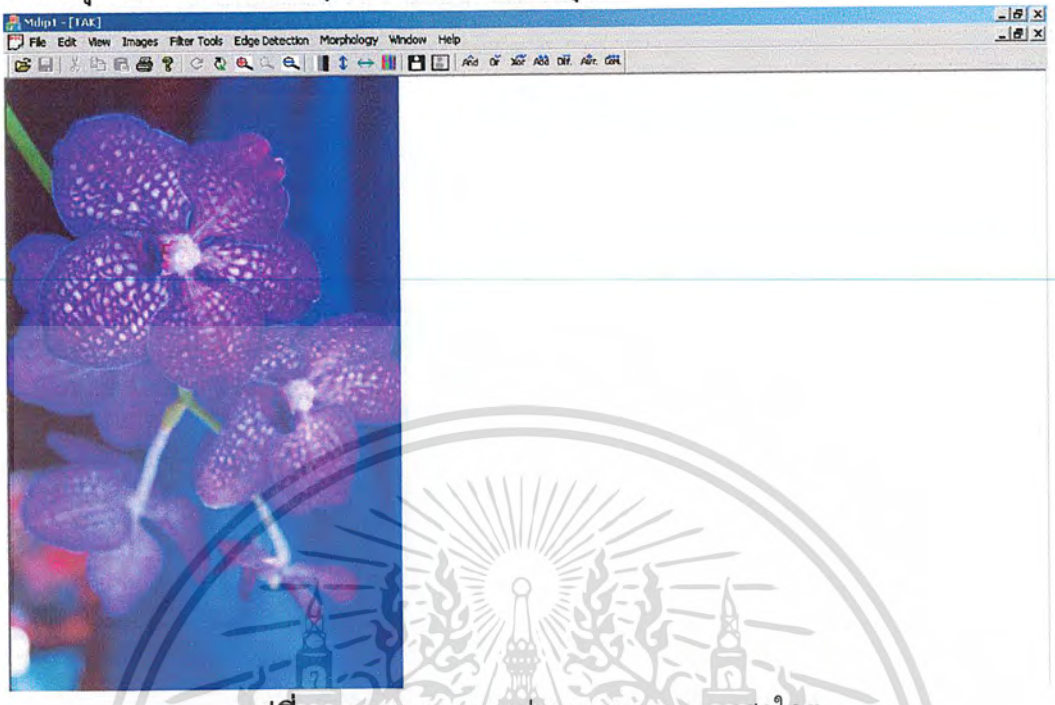
รูปที่3.57 การแสดงภาพก่อนการหมุนทวนเข็มนาฬิกา



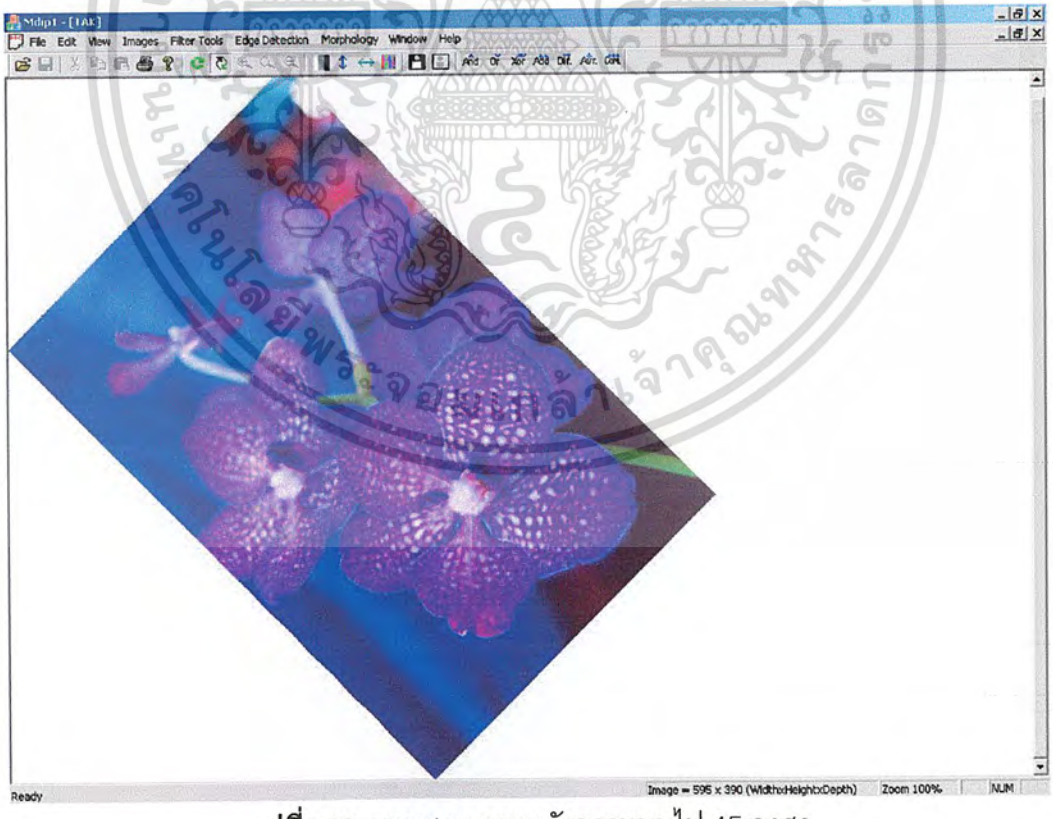
รูปที่3.58 การแสดงภาพหลังการหมุนทวนเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหมุนภาพ ณ องศาใดๆ และ ยกเลิกการหมุน



รูปที่3.59 การแสดงภาพก่อนการหมุน ณ องศาใดๆ

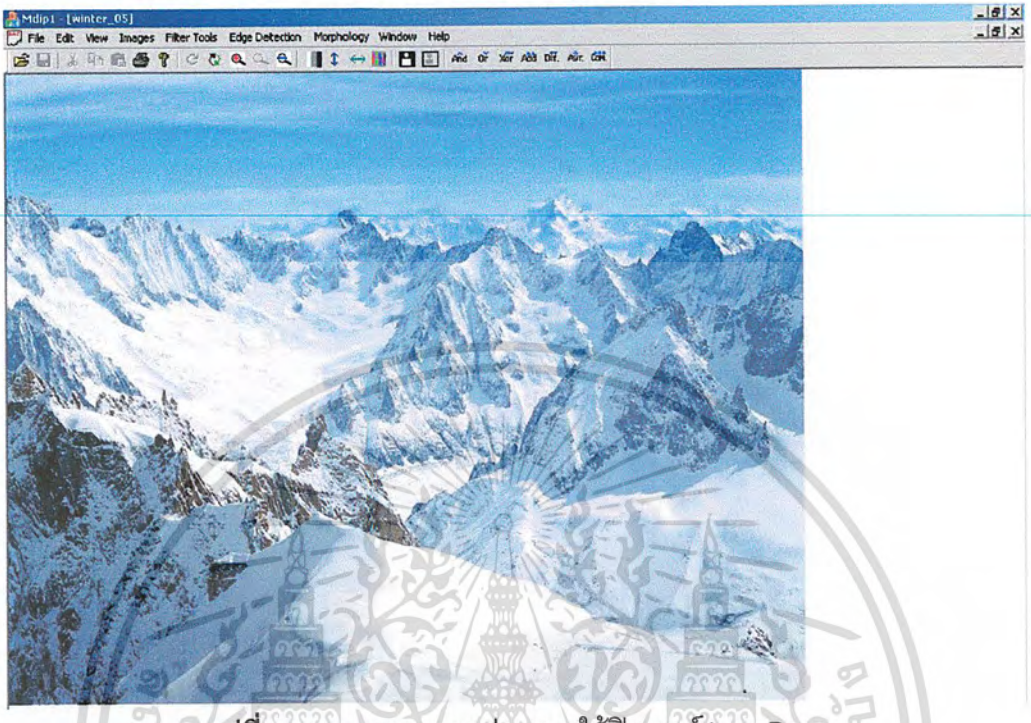


รูปที่3.60 การแสดงภาพหลังการหมุนไป 45 องศา

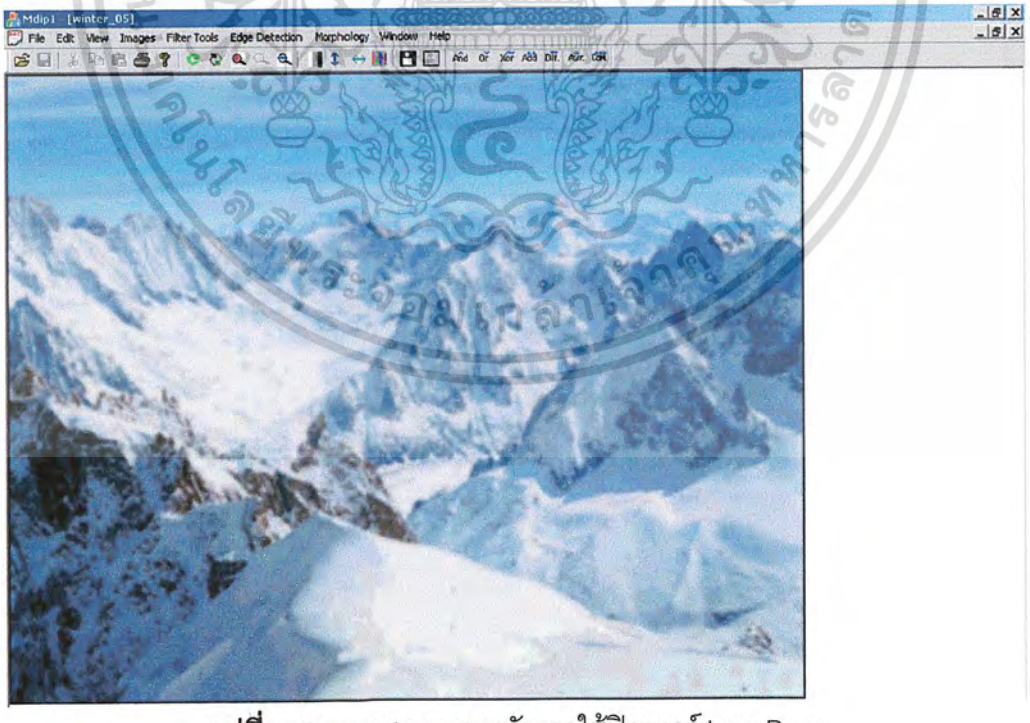
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ฟิลเตอร์ Low Pass เพื่อสร้างภาพมัว

การปรับอิมเมจให้มัวขึ้นหรือเบลอ โดยใช้เมนูคำสั่ง Filter Tools > Low Pass



รูปที่3.61 การแสดงภาพก่อนการใช้ฟิลเตอร์ Low Pass

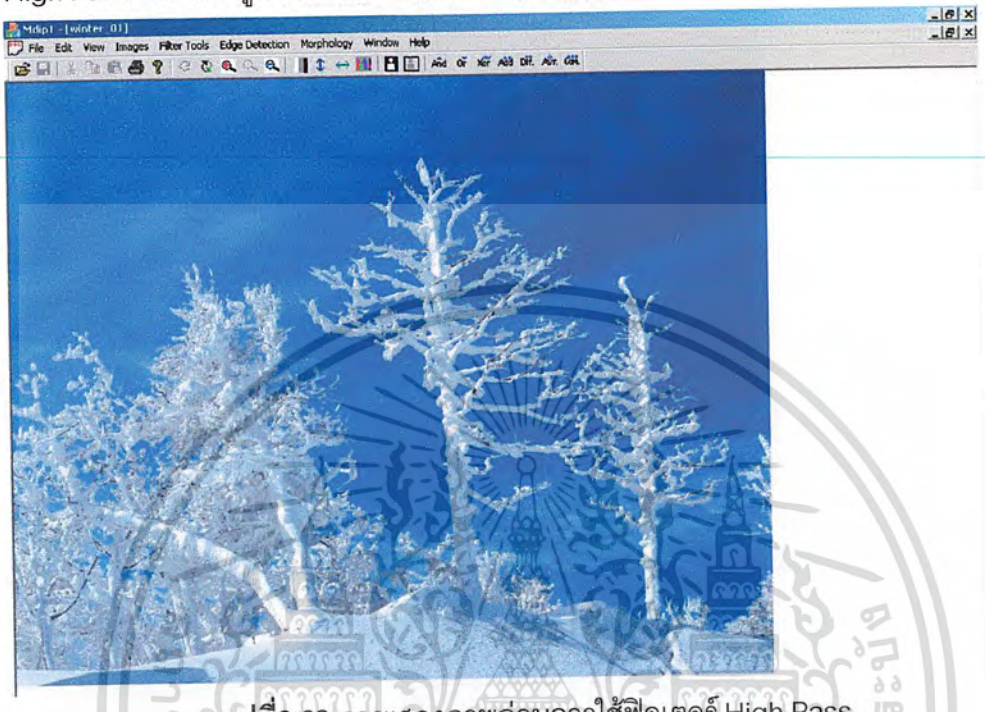


รูปที่3.62 การแสดงภาพหลังการใช้ฟิลเตอร์ Low Pass

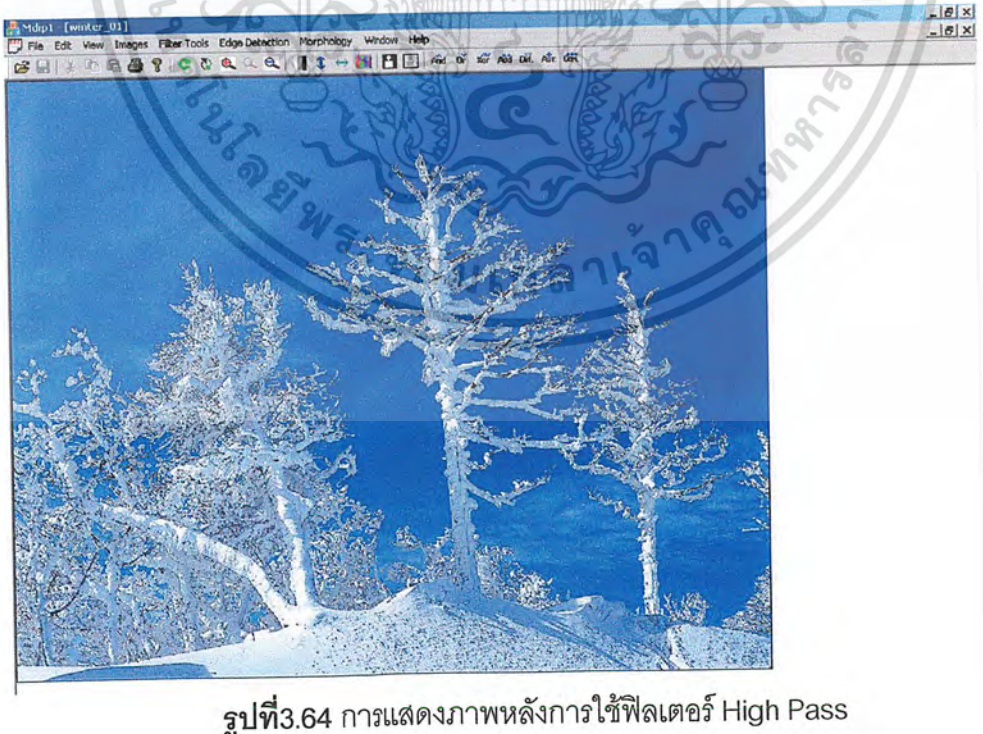
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ฟิลเตอร์ High Pass เพื่อสร้างภาพคมชัด

การปรับอิมเมจให้คมชัดขึ้น หรือ Sharpen Image ซึ่งจะตรงข้ามกับการใช้ฟิลเตอร์ High Pass โดยใช้เมนูคำสั่ง Filter Tools > High Pass



รูปที่3.63 การแสดงภาพก่อนการใช้ฟิลเตอร์ High Pass



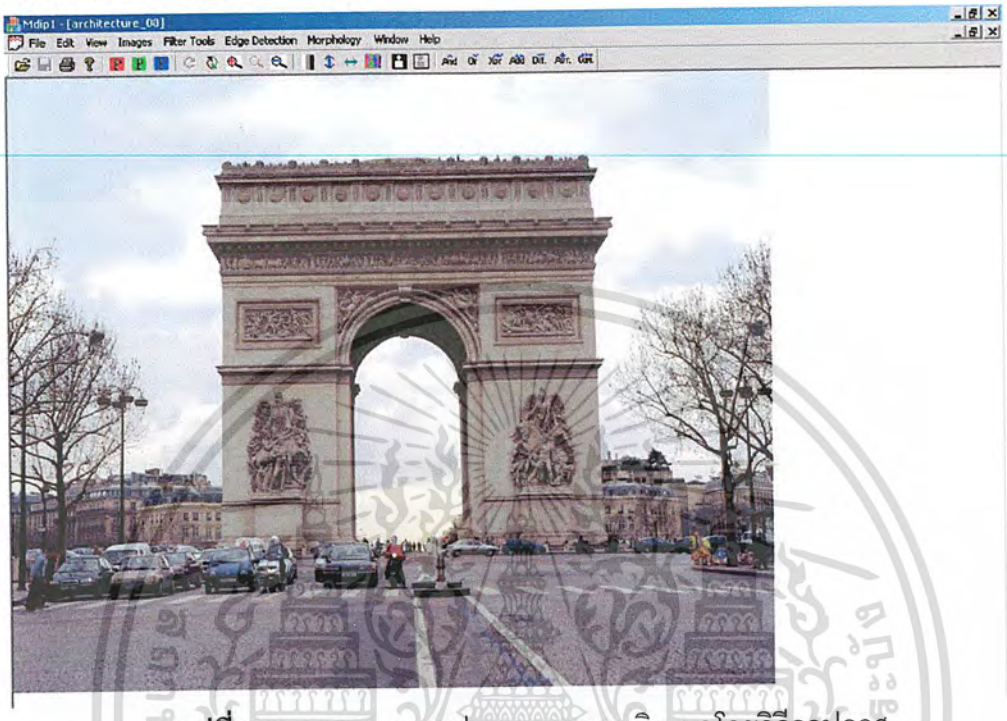
รูปที่3.64 การแสดงภาพหลังการใช้ฟิลเตอร์ High Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

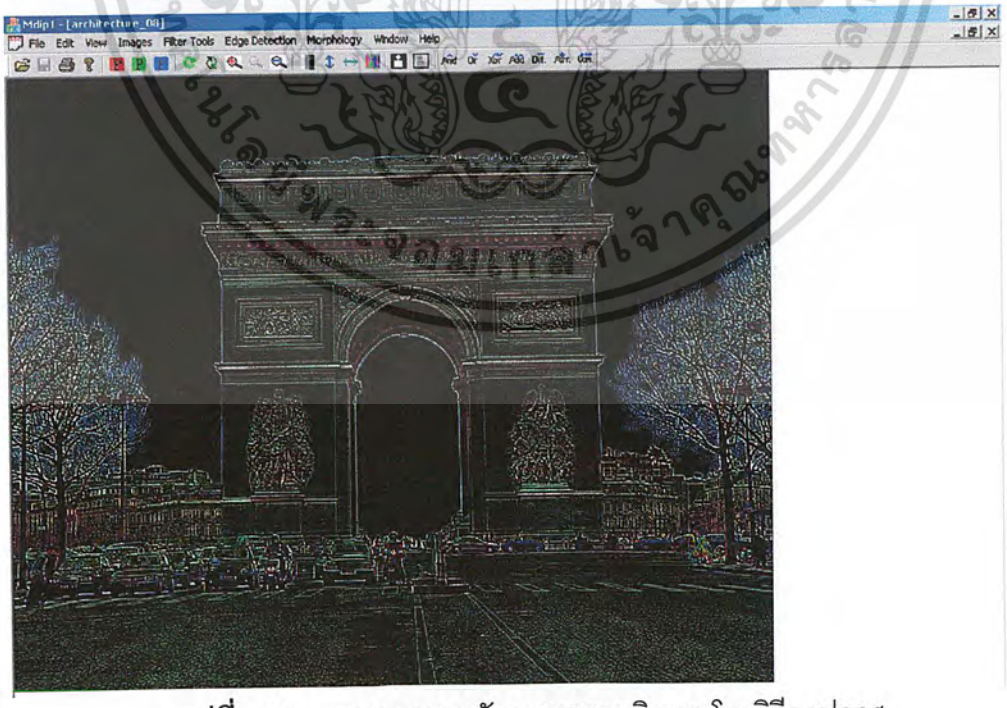
การหาขอบอิมเมจ (Edge Detection) โดยวิธีของลาปลาส

ใช้เพื่อหาลายเส้นของขอบอิมเมจโดยการใช้หน้ากากของลาปลาส โดยใช้เมนูคำสั่ง

Filter Tools > Lapalcian Edge



รูปที่3.65 การแสดงภาพก่อนการหาขอบอิมเมจโดยวิธีลาปลาส

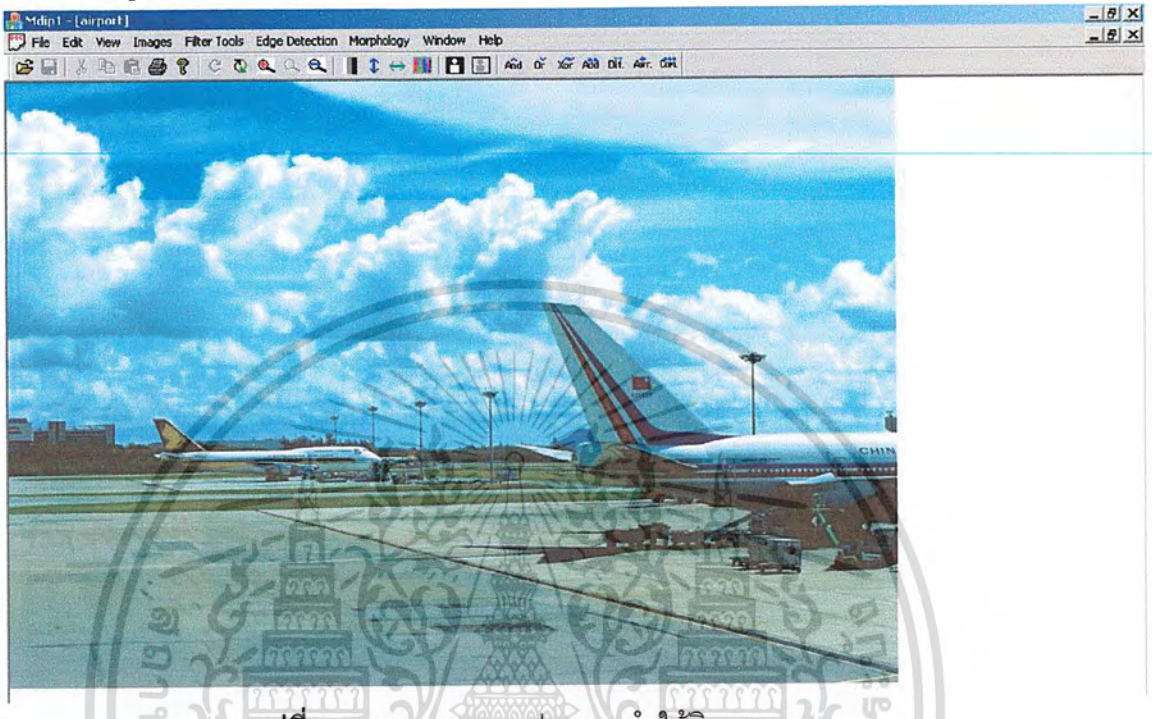


รูปที่3.66 การแสดงภาพหลังการหาขอบอิมเมจโดยวิธีลาปลาส

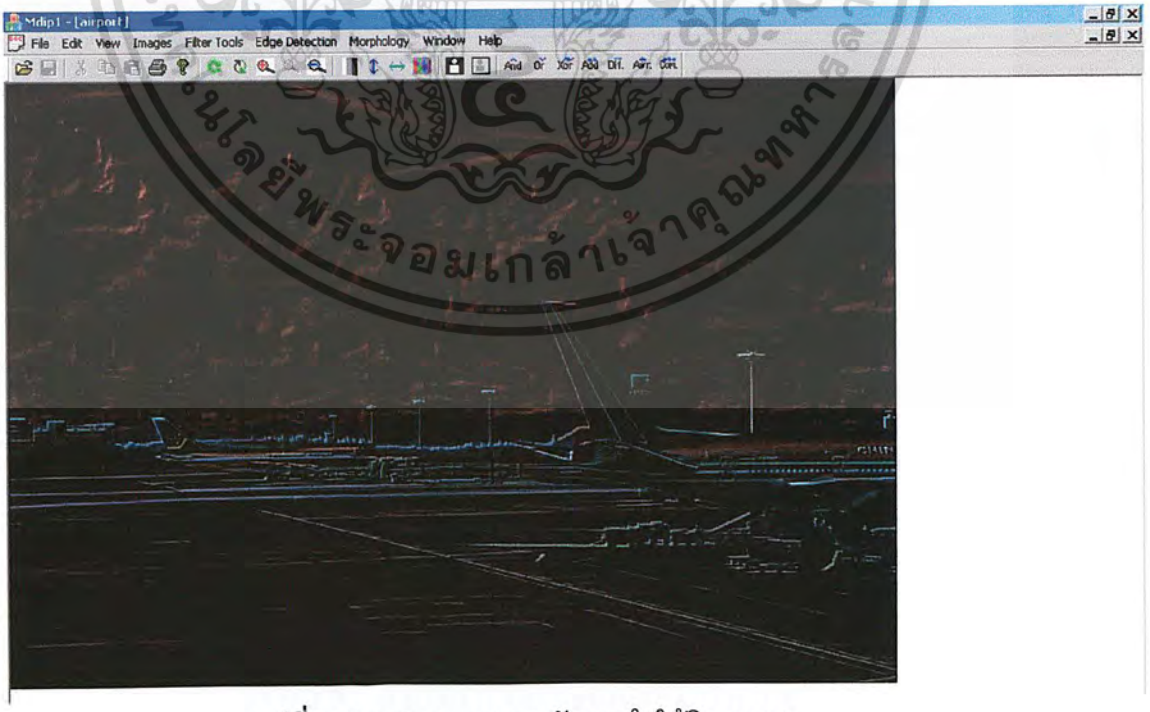
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำให้อิมเมจนูนขึ้น (Emboss)

โดยใช้เมนูคำสั่ง Filter Tools > Emboss โดยภาพที่ออกมาจะเหมือนกับการหาขอบ แต่ภาพจะดูมีมิติมากกว่า



รูปที่3.67 การแสดงภาพก่อนการทำให้อิมเมจนูน



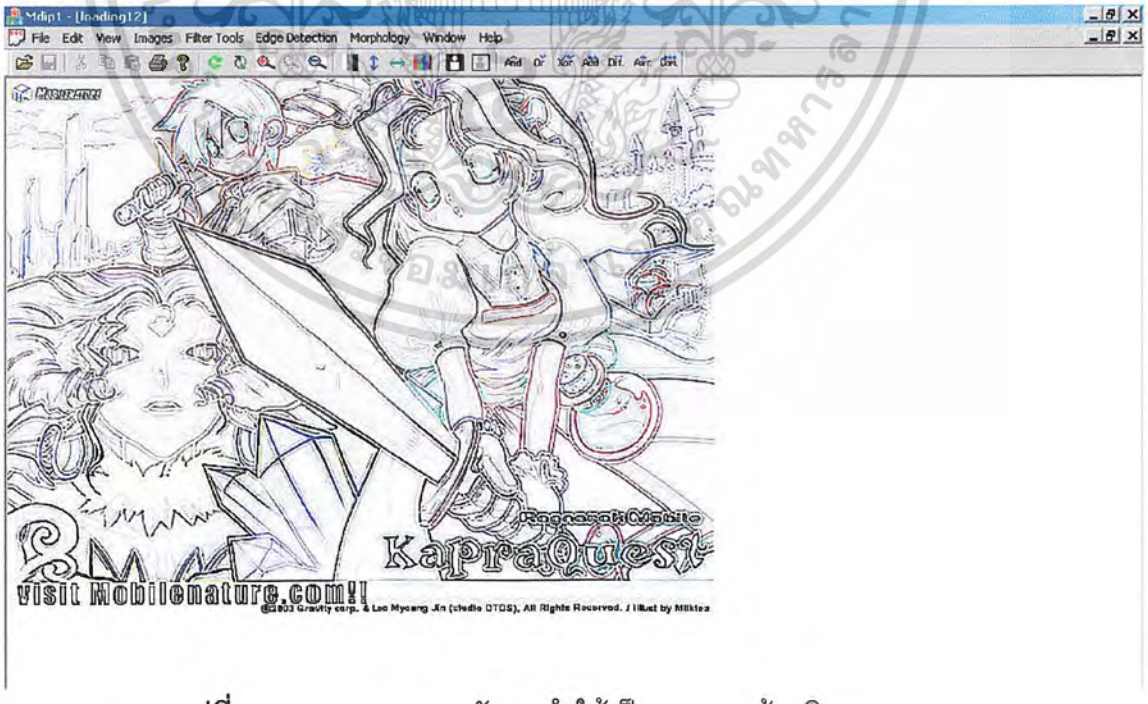
รูปที่3.68 การแสดงภาพหลังการทำให้อิมเมจนูน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำให้ภาพเป็นภาพวาดด้วยดินสอ(Pencil Drawing)

หลักการก็คือการหาลายเส้นของอิมเมจโดยให้พื้นเป็นสีขาว โดยใช้เมนูคำสั่ง Filter Tools > Pencil Drawing

รูปที่3.69 การแสดงภาพก่อนการทำให้เป็นภาพวาดด้วยดินสอ

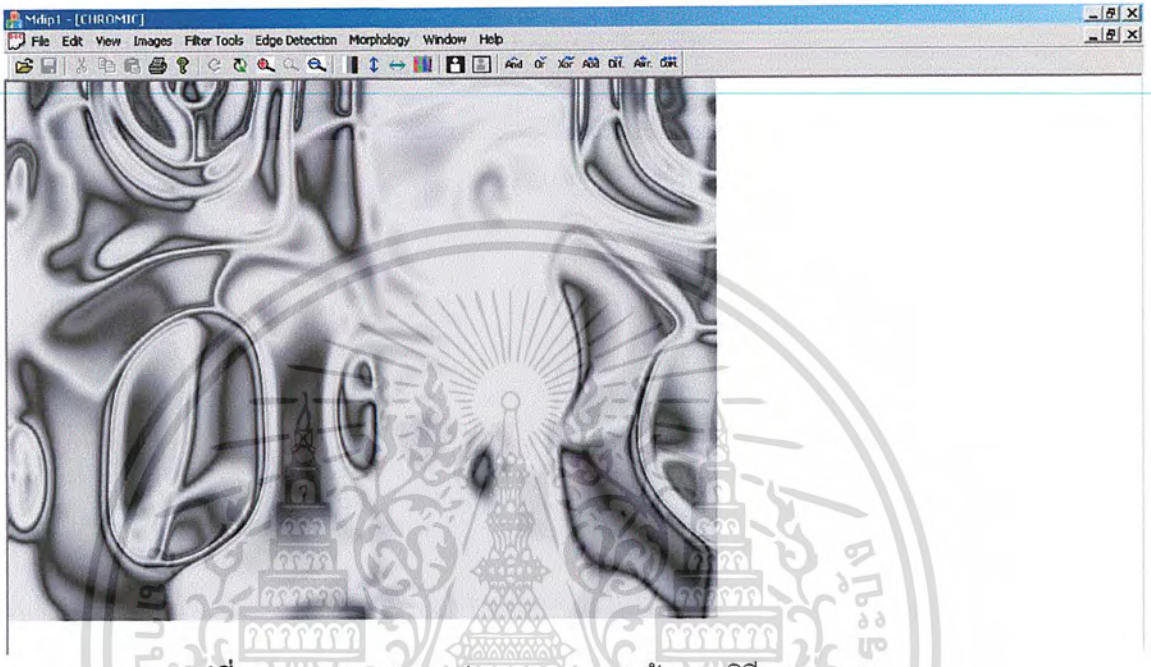


รูปที่3.70 การแสดงภาพหลังการทำให้เป็นภาพวาดด้วยดินสอ

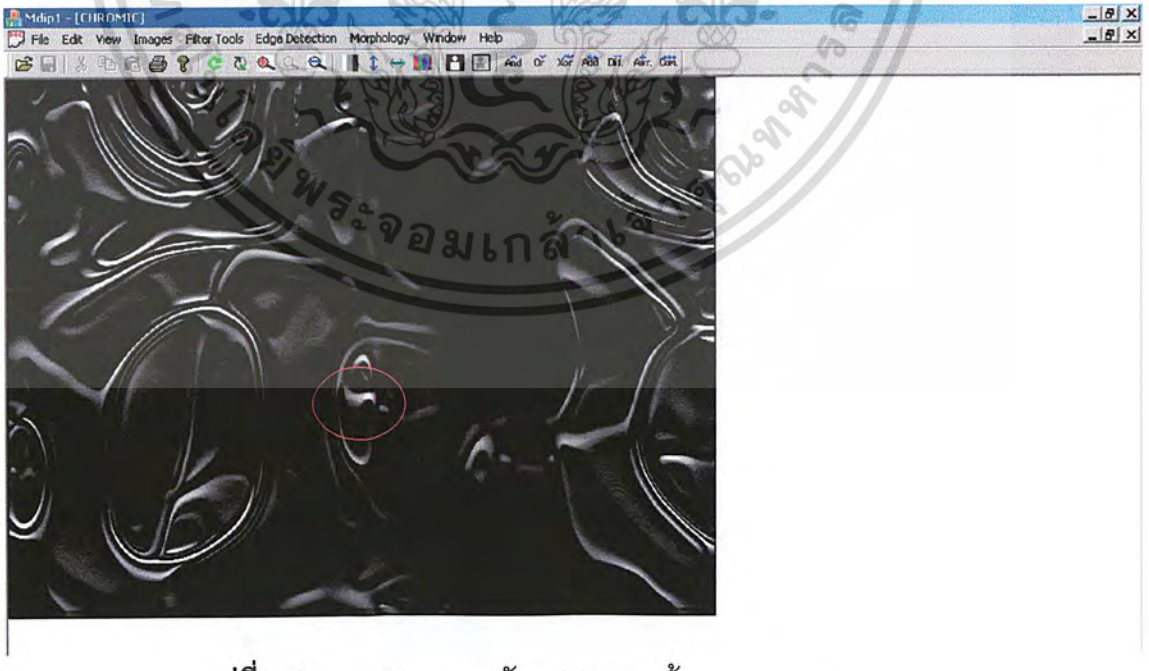
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบอิมเมจโดยวิธีของSobel

ใช้เพื่อหาลายเส้นอิมเมจในทิศทางแกนนอน หรือ แกน X แกนตั้ง หรือ แกน Y และแกนตั้งและแกนนอน หรือ แกน XY โดยวิธีของ Sobel โดยใช้เมนูคำสั่ง Edge Detection > Sobel



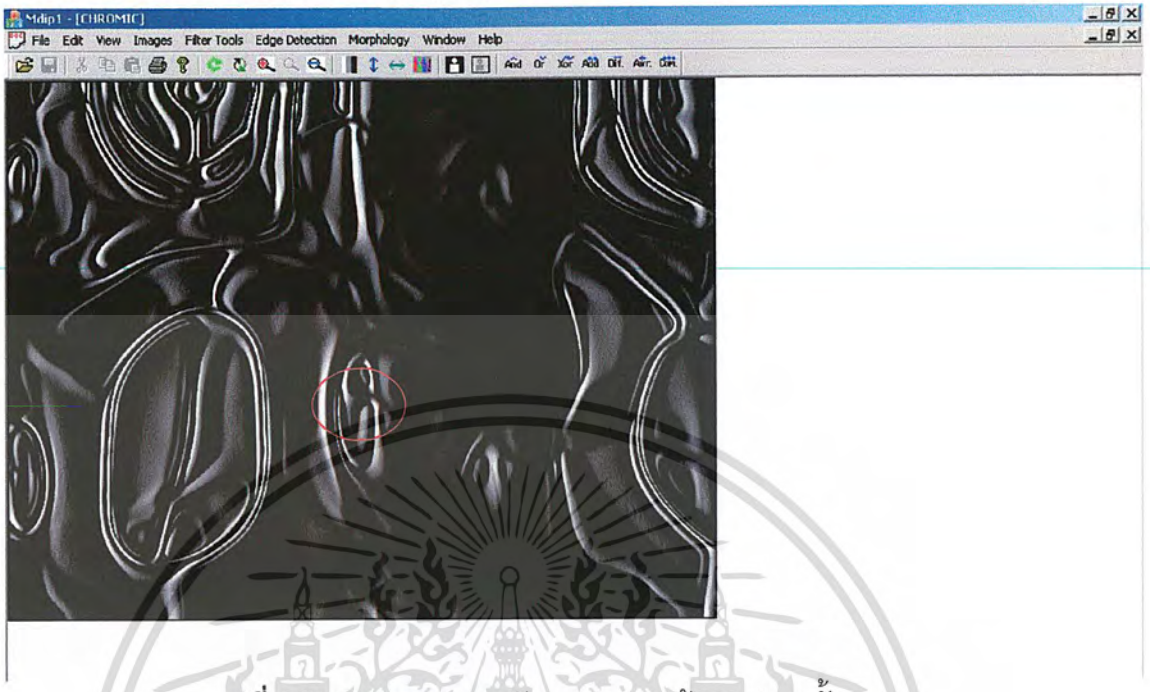
รูปที่3.71 การแสดงภาพก่อนการหาลายเส้นตามวิธีของSobel



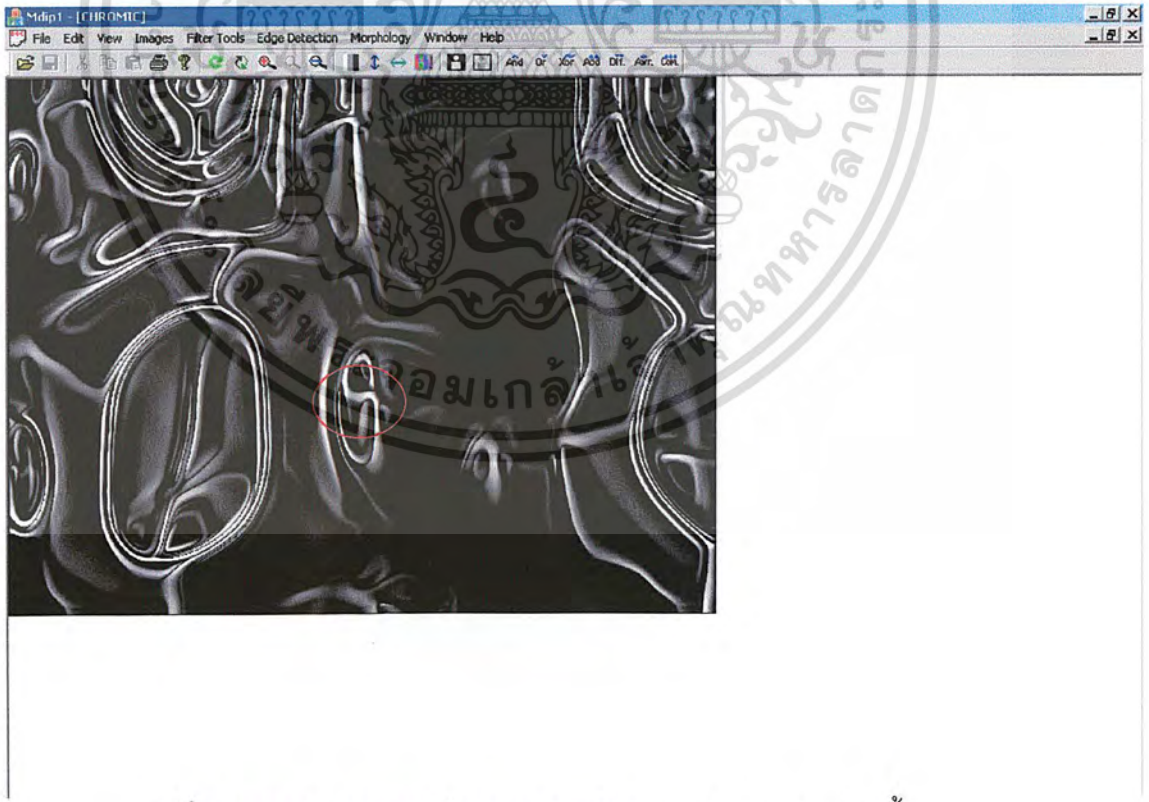
รูปที่3.72 การแสดงภาพหลังการหาลายเส้นตามแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบอิมเมจโดยวิธีของSobel(ต่อ)



รูปที่3.73 การแสดงภาพหลังการหาลายเส้นตามแนวตั้ง

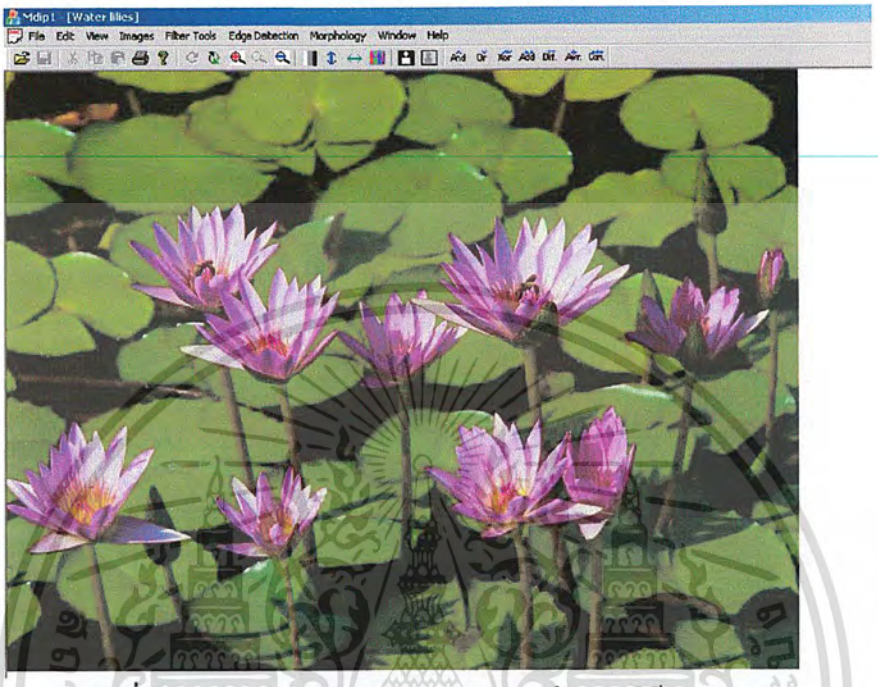


รูปที่3.74 การแสดงภาพหลังการหาลายเส้นตามแนวนอนและแนวตั้ง

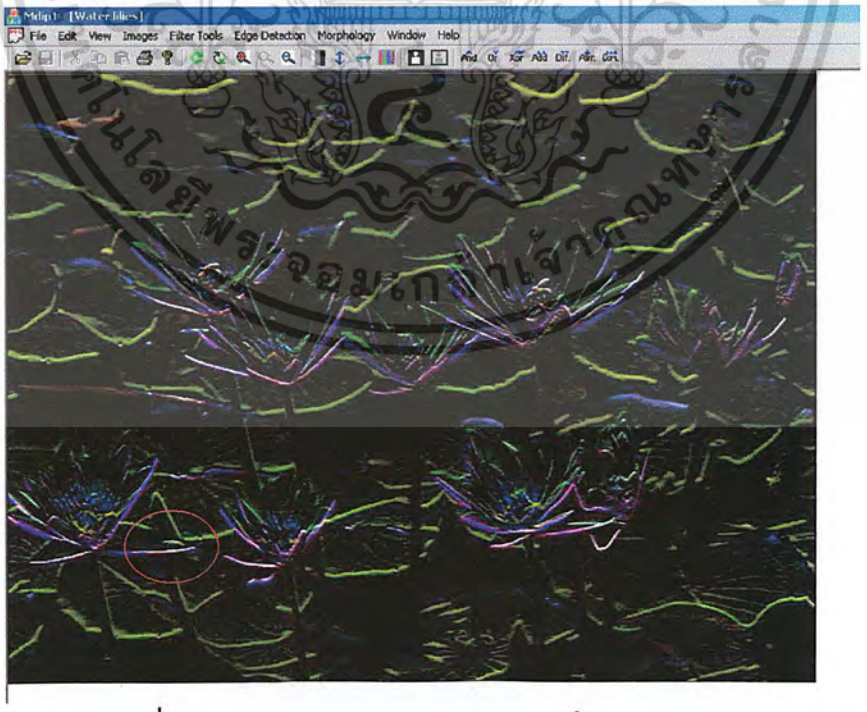
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบอิมเมจโดยวิธีของPrewitt

ใช้เพื่อหาลายเส้นอิมเมจในทิศทางแกนนอน(แกน X) แกนตั้ง(แกน Y) และแกนตั้งและแกนนอน(แกน XY) โดยวิธีของ Prewitt โดยใช้เมนูคำสั่ง Edge Detection > Prewitt



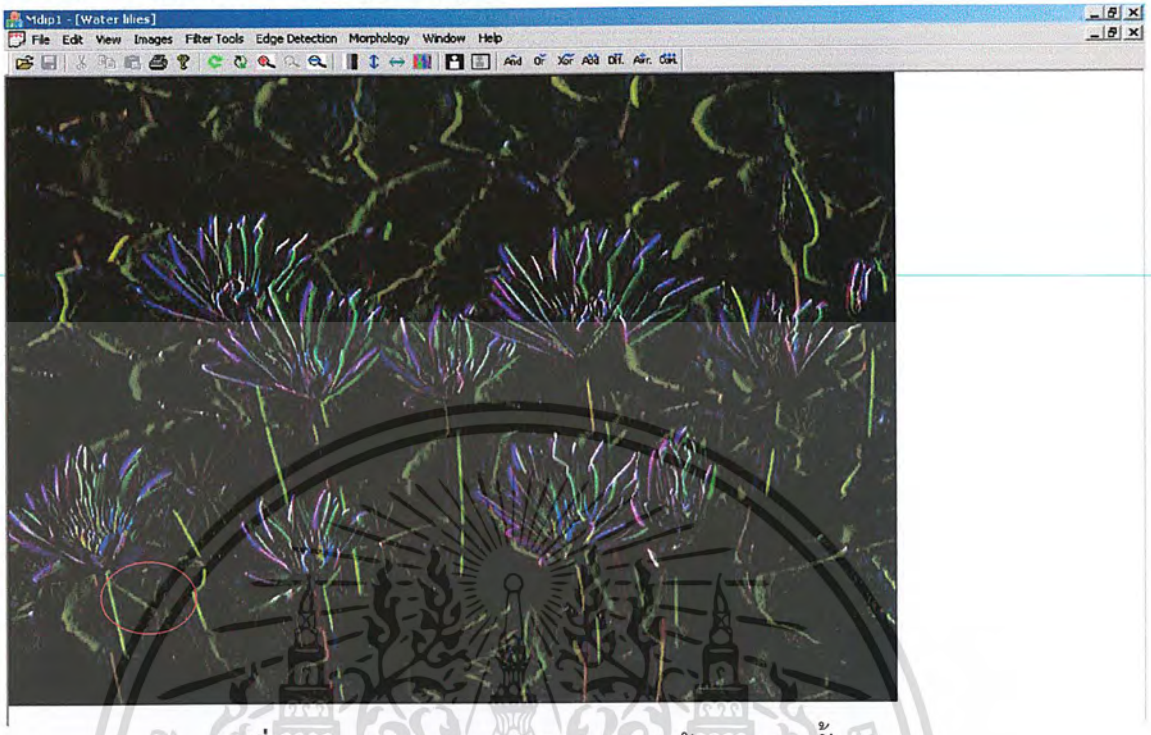
รูปที่3.75 การแสดงภาพก่อนการหาลายเส้นตามวิธีของPrewitt



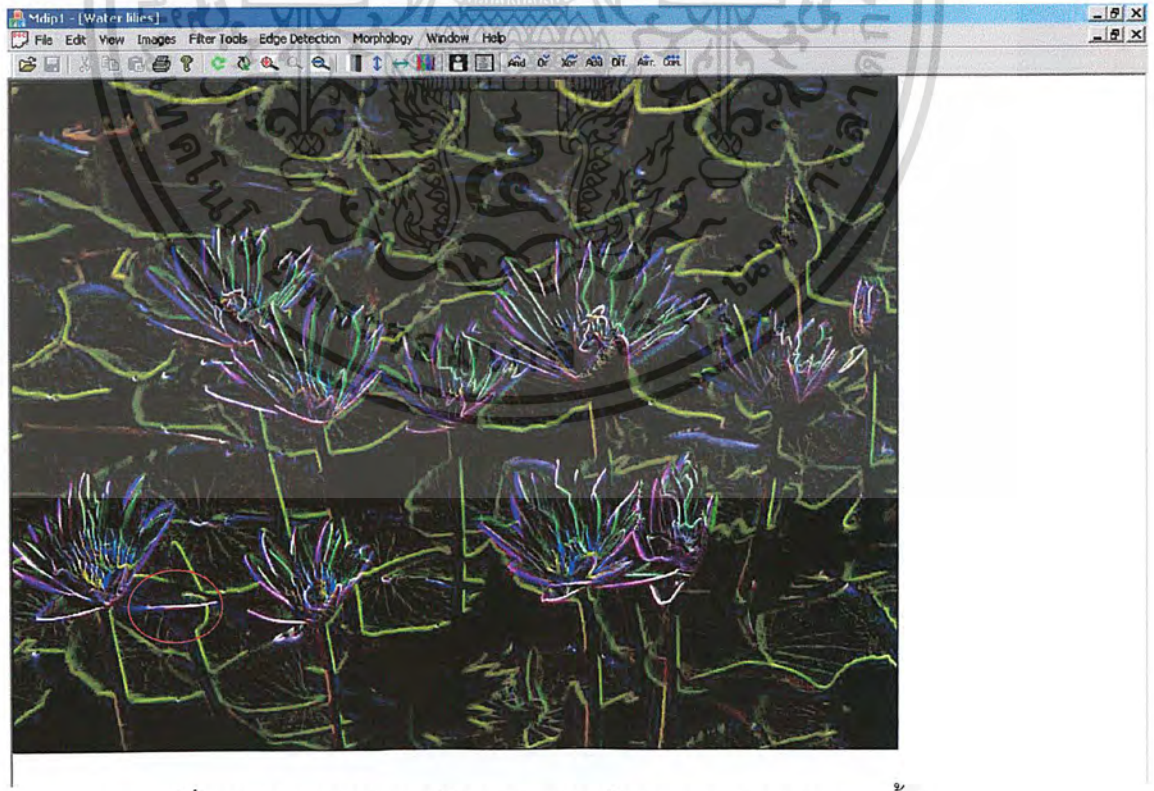
รูปที่3.76 การแสดงภาพหลังการหาลายเส้นตามแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบอิมเมจโดยวิธีของPrewitt (ต่อ)



รูปที่3.77 การแสดงภาพหลังการหาลายเส้นตามแนวตั้ง

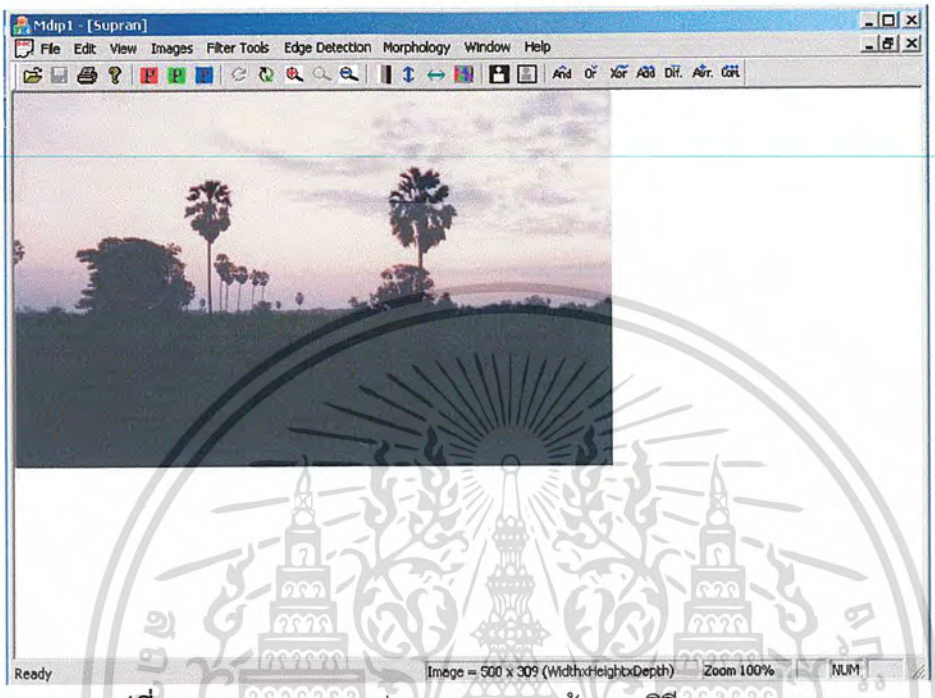


รูปที่3.78 การแสดงภาพหลังการหาลายเส้นตามแนวนอนและแนวตั้ง

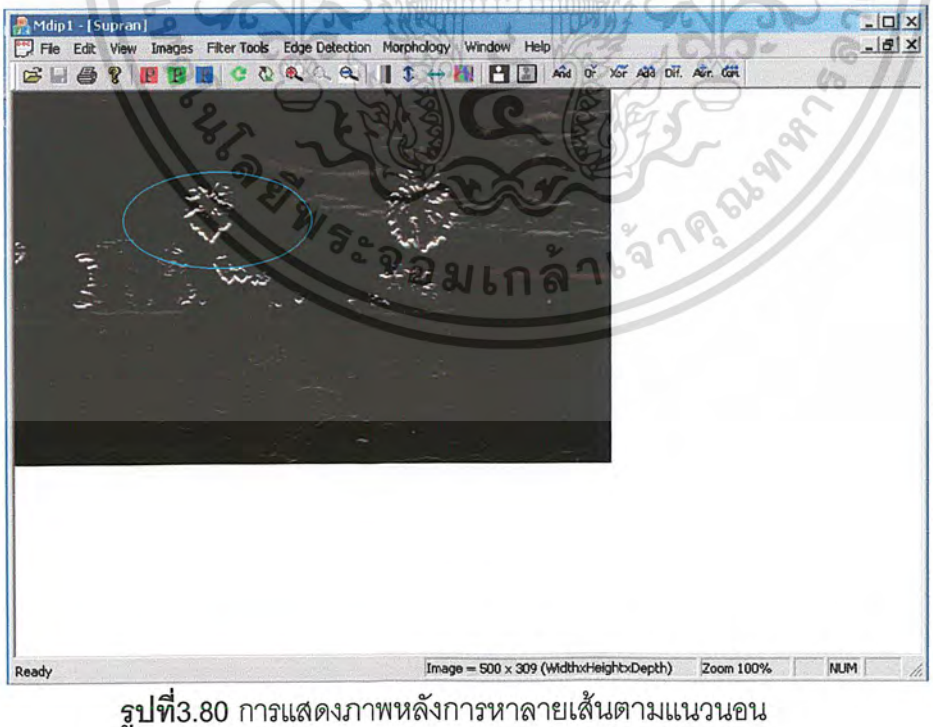
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบอิมเมจโดยวิธีของFrei-Chen

ใช้เพื่อหาลายเส้นอิมเมจในทิศทางแกนนอน(แกน X) แกนตั้ง(แกน Y)และแกนตั้งและแกนนอน(แกน XY)โดยวิธีของ Frei-Chen โดยใช้เมนูคำสั่ง Edge Detection > Frei-Chen



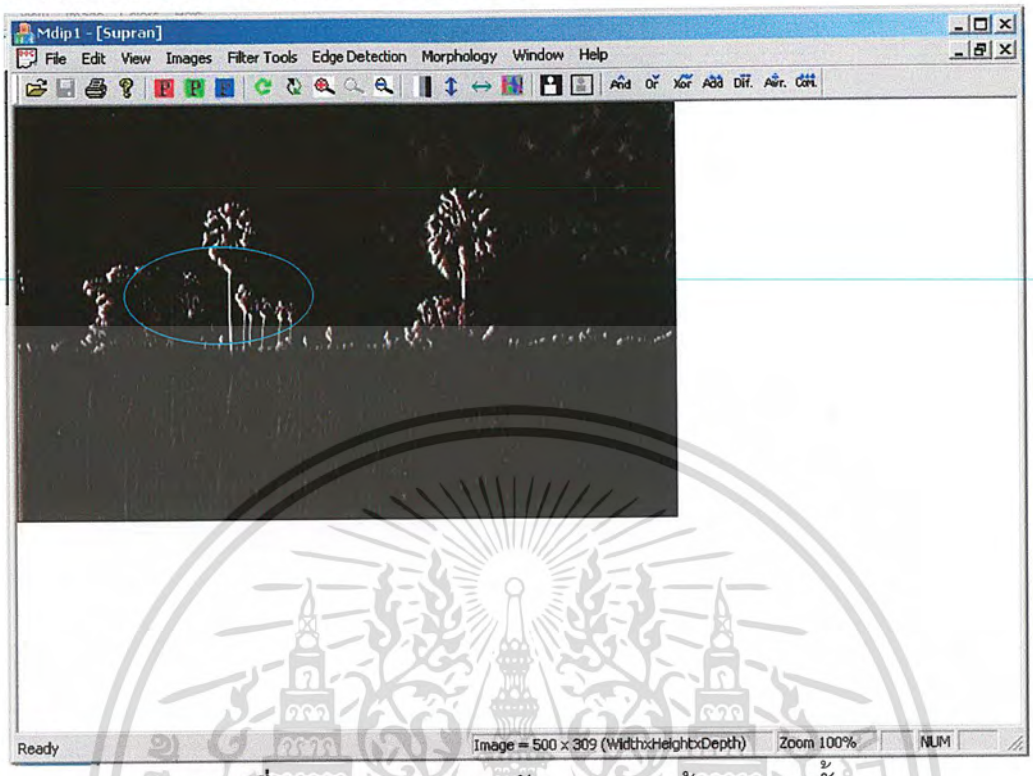
รูปที่3.79 การแสดงภาพก่อนการหาลายเส้นตามวิธีของFrei-Chen



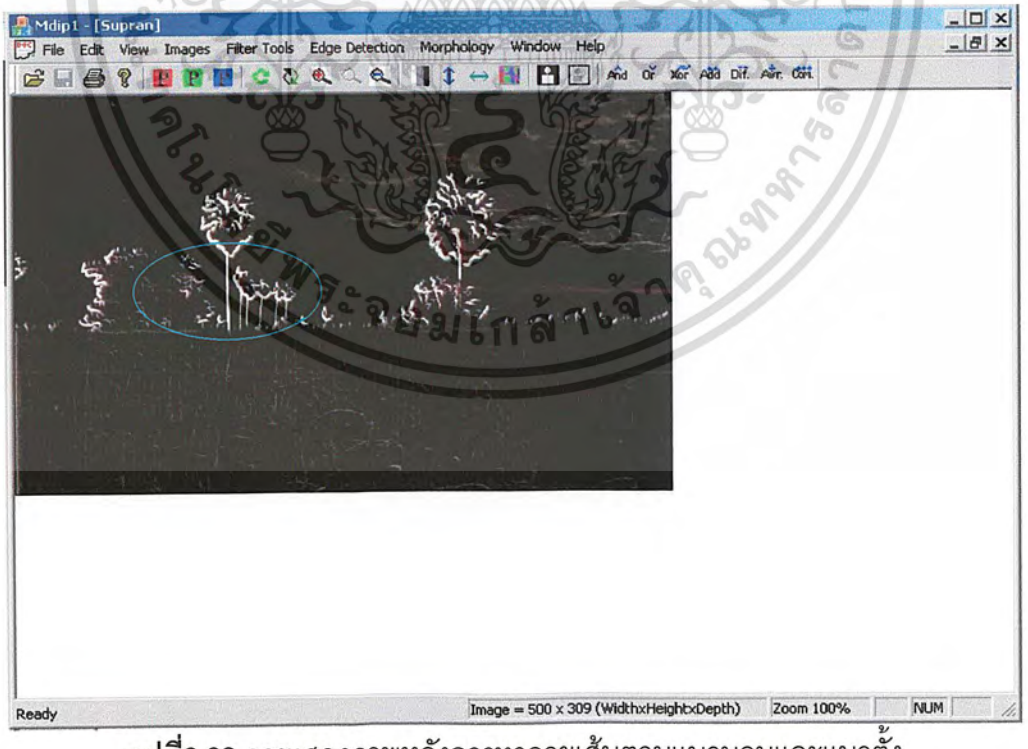
รูปที่3.80 การแสดงภาพหลังการหาลายเส้นตามแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบอิมเมจโดยวิธีของFrei-Chen(ต่อ)



รูปที่3.81 การแสดงภาพหลังการหาลายเส้นตามแนวตั้ง

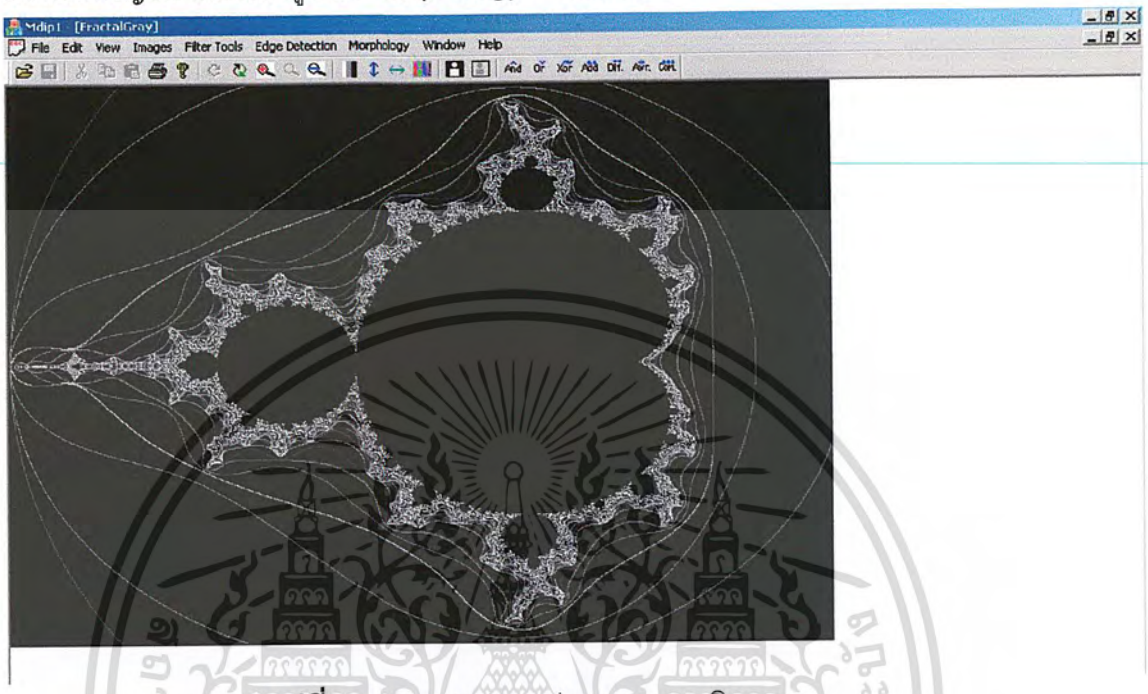


รูปที่3.82 การแสดงภาพหลังการหาลายเส้นตามแนวนอนและแนวตั้ง

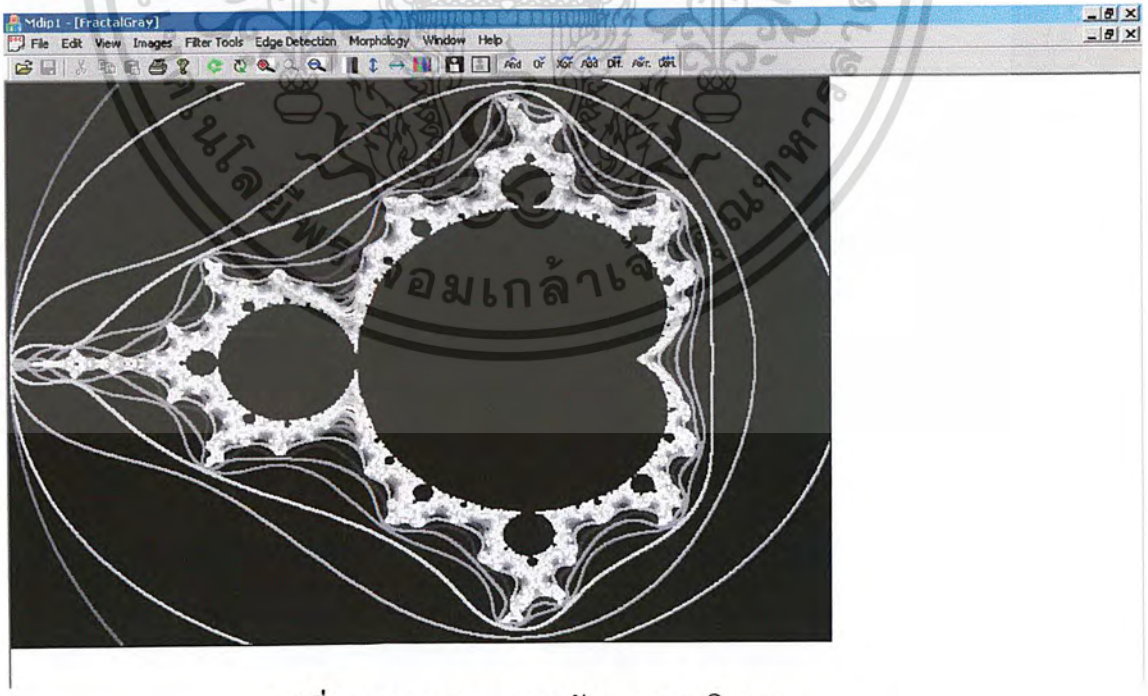
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขยายพิกเซล(Dilation)

พูดง่ายๆ ก็คือการเพิ่มจุดสว่างและขจัดจุดมืดออกไปจากภาพโดยการขยายพิกเซลที่สว่างให้ใหญ่ขึ้น โดยใช้เมนูคำสั่ง Morphology > Dilation



รูปที่3.83 การแสดงภาพก่อนการขยายพิกเซล

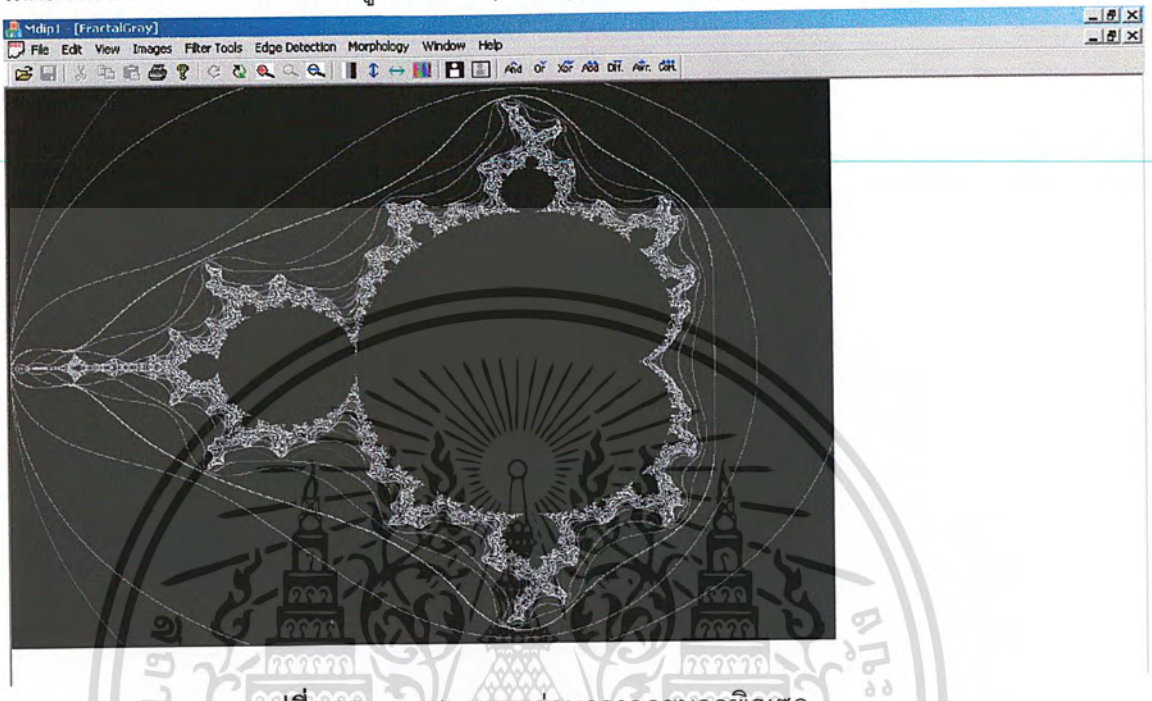


รูปที่3.84 การแสดงภาพหลังการขยายพิกเซล

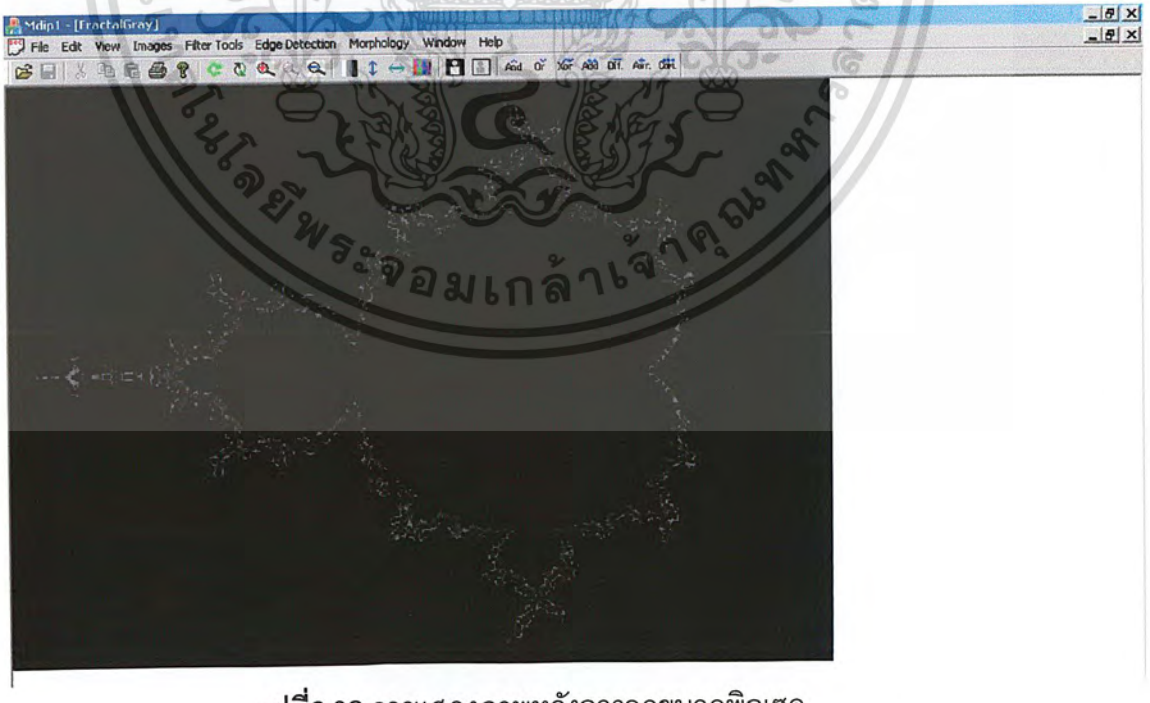
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลดพิกเซล(Erosion)

ก็คือ การเพิ่มจุดสว่างและขจัดจุดมืดออกไปจากภาพโดยการขจัดพิกเซลที่สว่างให้เล็กลงหรือขจัดออกไป โดยใช้เมนูคำสั่ง Morphology > Erosion



รูปที่3.85 การแสดงภาพก่อนการลดขนาดพิกเซล

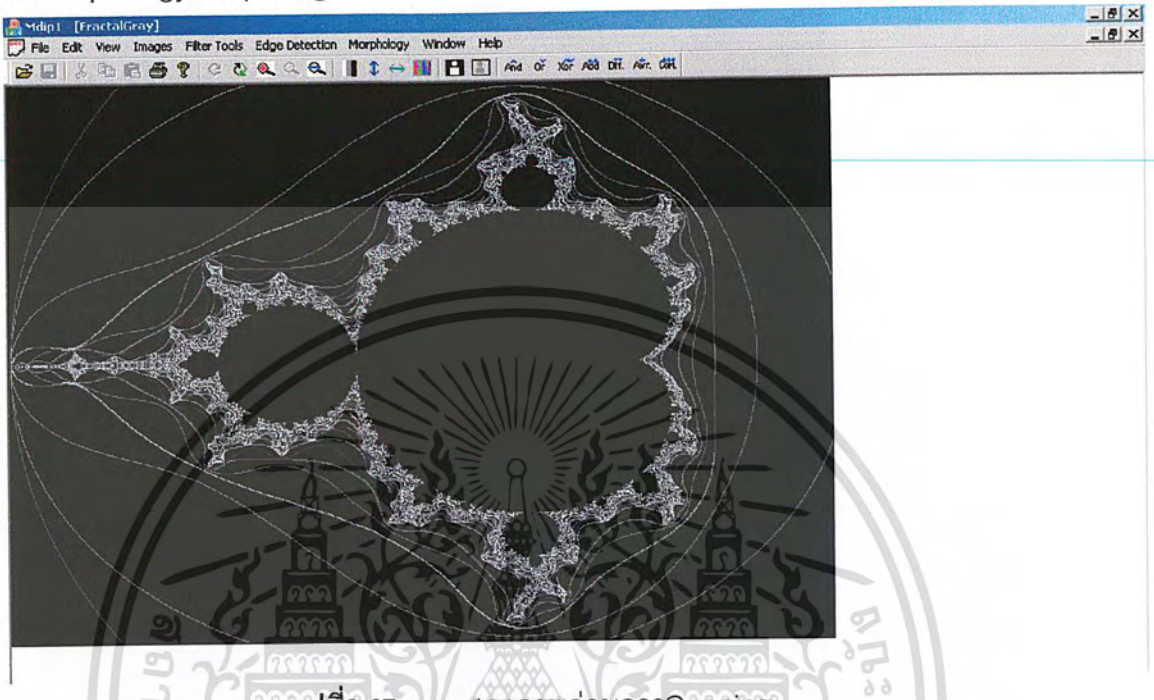


รูปที่3.86 การแสดงภาพหลังการลดขนาดพิกเซล

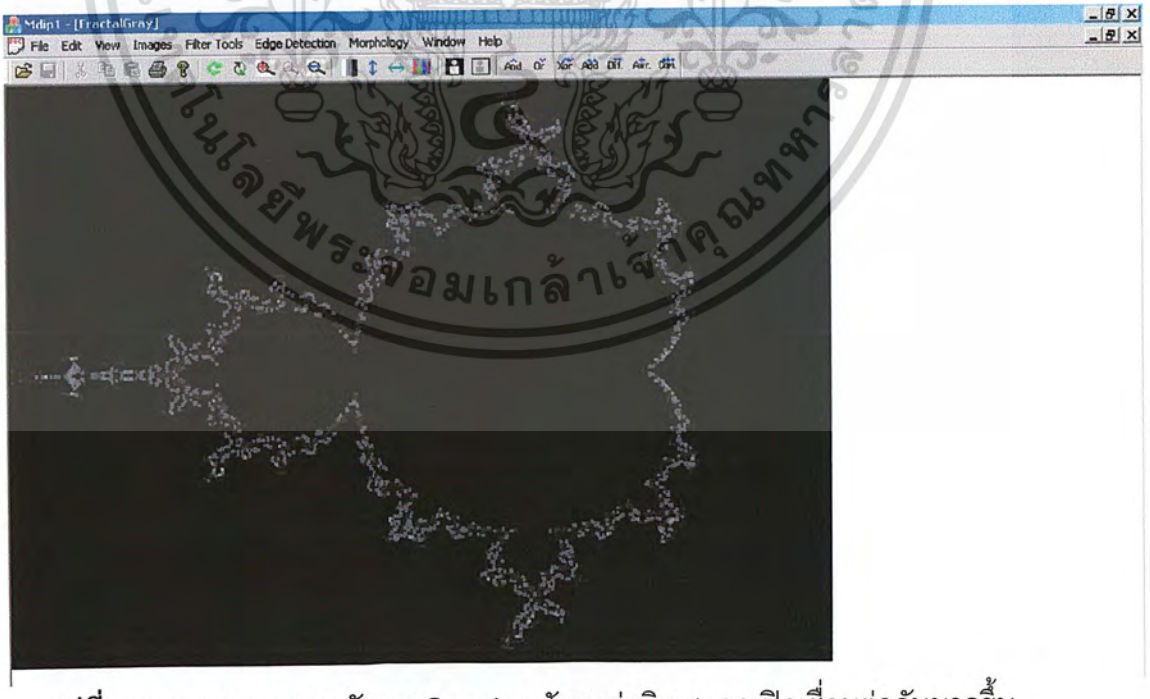
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำอิมเมจในพื้นที่ว่างให้เปิดมากขึ้น(Opening)

โดยแนวคิด คือ การลดขนาดพิกเซลก่อนแล้วทำการขยายขนาดพิกเซล โดยใช้เมนูคำสั่ง Morphology > Opening



รูปที่3.87 การแสดงภาพก่อนการOpening

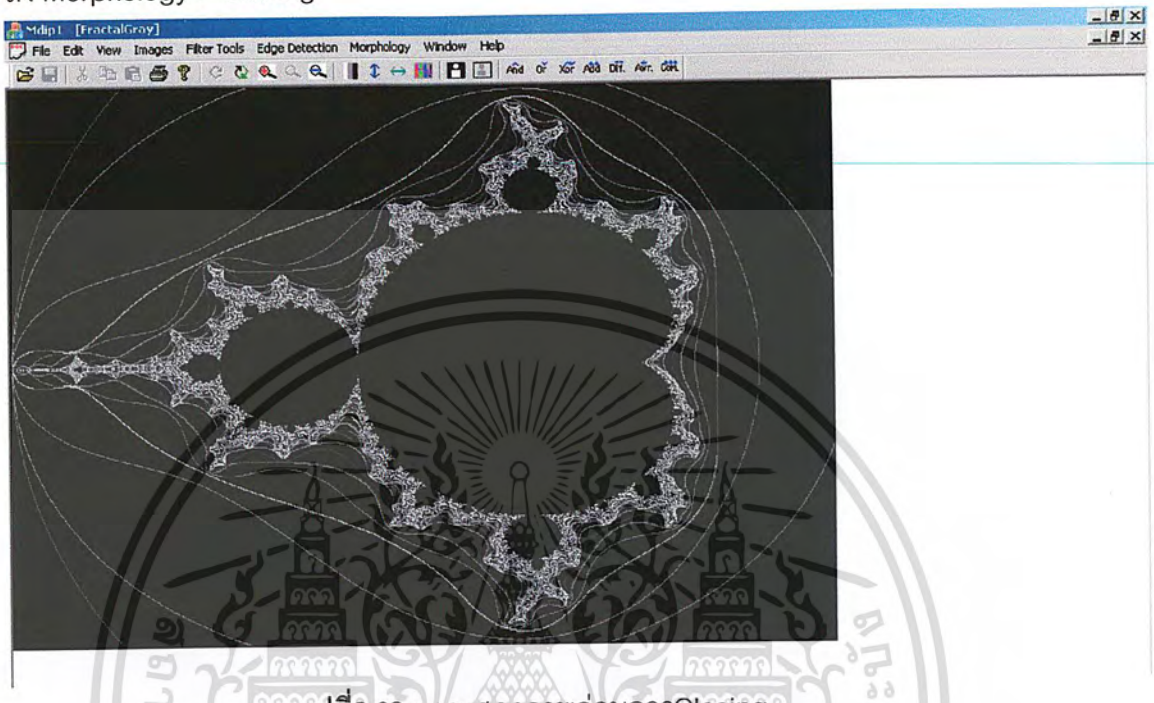


รูปที่3.88 การแสดงภาพหลังการ Opening สังเกตว่าพิกเซลถูกเปิดเชื่อมต่อกันมากขึ้น

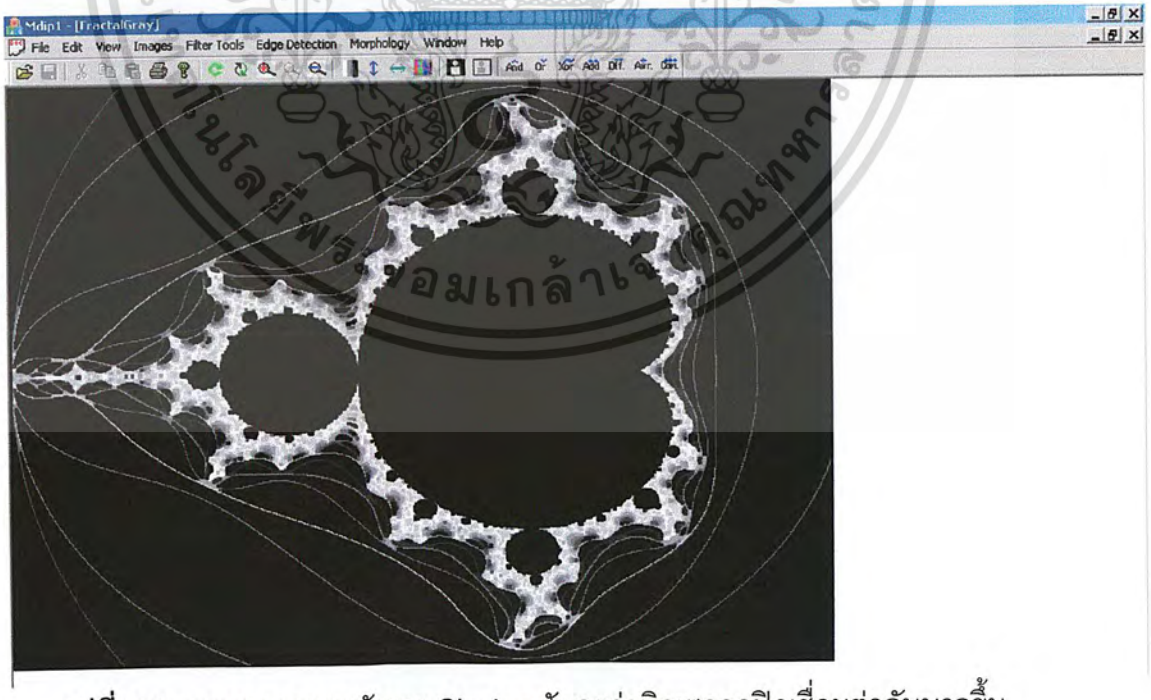
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำอิมเมจในพื้นที่ว่างให้ปิดมากขึ้น(Closing)

โดยแนวคิด คือ การขยายขนาดพิกเซลก่อนแล้วทำการลดขนาดพิกเซล โดยใช้เมนูคำสั่ง Morphology > Closing



รูปที่3.89 การแสดงภาพก่อนการClosing

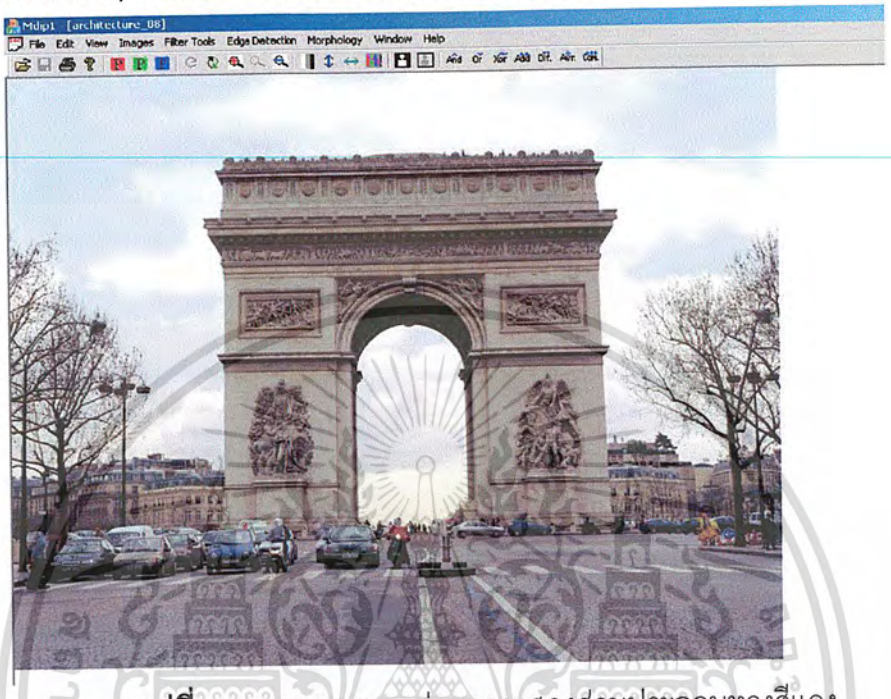


รูปที่3.90 การแสดงภาพหลังการ Closing สังเกตว่าพิกเซลถูกปิดเชื่อมต่อกันมากขึ้น

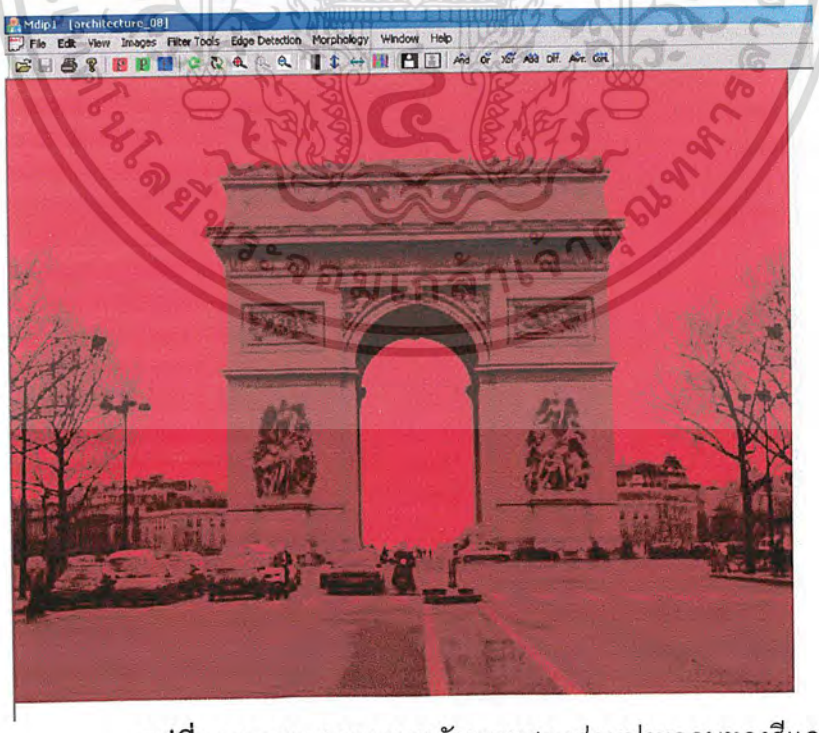
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงส่วนประกอบทางสีแดง สีเขียว และ สีน้ำเงิน

โดยแนวคิด คือ การทำให้ภาพถูกกรอง เสมือนมองผ่านกระจกสีแดง โดยใช้เมนูคำสั่ง View > Red Component หรือ Green Component และ Blue Component



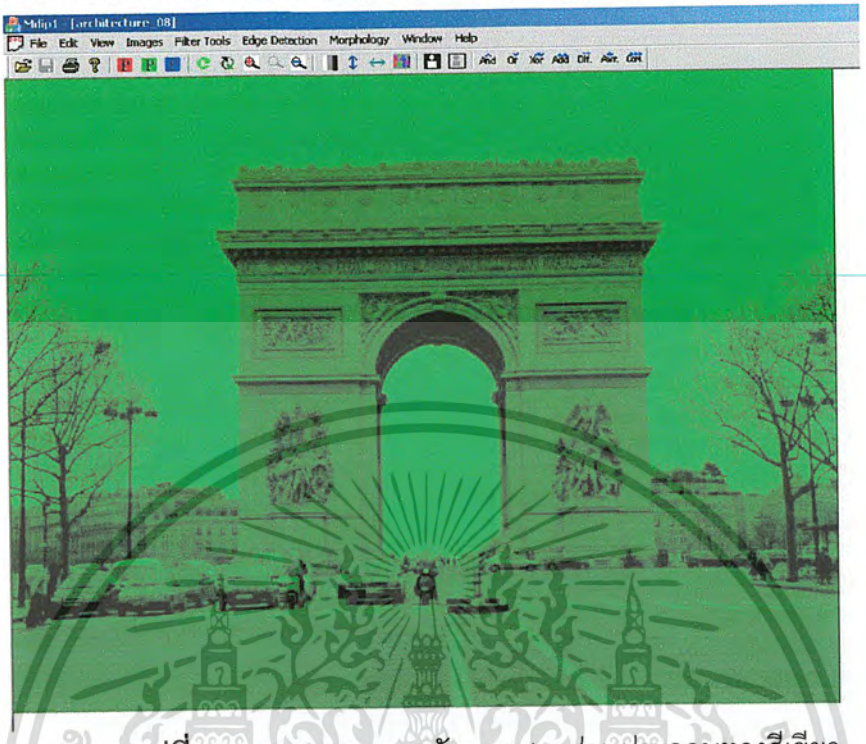
รูปที่3.91 การแสดงภาพก่อนการแสดงส่วนประกอบทางสีแดง



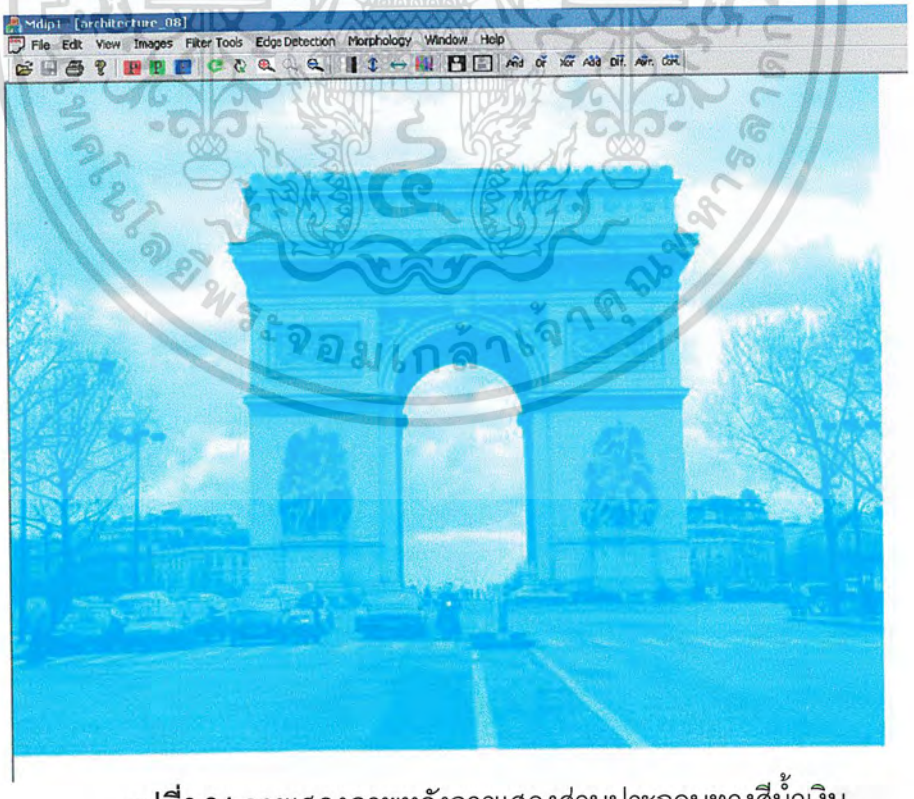
รูปที่3.92 การแสดงภาพหลังการแสดงส่วนประกอบทางสีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงส่วนประกอบทางสีแดง สีเขียว และ สีน้ำเงิน (ต่อ)



รูปที่3.93 การแสดงภาพหลังการแสดงส่วนประกอบทางสีเขียว



รูปที่3.94 การแสดงภาพหลังการแสดงส่วนประกอบทางสีน้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การประเมินผลระบบ

การประเมินผล

โปรแกรมนี้ในปัญหาพิเศษนี้ถูกพัฒนาขึ้นโดยใช้หลักการประมวลผลภาพ โดยนำแนวคิดที่สำคัญของแต่ละแนวคิด การใช้หลักของคณิตศาสตร์ และ สถิติ เช่น ระบบสมการ ระบบลอจิค ระบบการคำนวณสถิติภาพ โดยโปรแกรมนี้สามารถทำงานได้บน Windows ซึ่งสามารถประมวลผลภาพจากไฟล์รูปภาพแบบต่างๆ เช่น Bmp, Dib, Rle, Jpeg, Gif, Tiff, Tiff และ Png ซึ่งเป็นประโยชน์อย่างมากในการศึกษาหรือวิจัยการวิเคราะห์ ยกตัวอย่างเช่น การหาขอบอิมเมจแล้วนำภาพที่ได้จากการหาขอบภาพแล้วนำมารวมกับภาพต้นฉบับ ก็จะได้ภาพผลลัพธ์ที่มีขอบภาพที่ชัดเจนมากขึ้น นำไปประยุกต์กับการดูดาวขององค์การนาซา หรือ การสร้างภาพอิมเมจใหม่โดยใช้การบวกลบ และ ตรรคศาสตร์ และแนวคิดของการหาขอบภาพมีประโยชน์อย่างมากในระบบการรู้จำของเครื่องจักรที่เมื่อหาขอบภาพได้แล้วจะให้เครื่องจักรจดจำขอบภาพและนำไปเปรียบเทียบกับวัตถุที่เราต้องการ เมื่อมีวัตถุที่แปลกปลอมเข้ามาก็จะทำให้เครื่องจักรรู้ได้ หรือการจดจำลายมือ การสแกนม่านตา และอื่นๆ

จากการที่ลองใช้โปรแกรมนี้กับภาพต่างๆ ต่างฟอร์แมทภาพ ในส่วนของตัวโปรแกรมนั้นในส่วนของการทำงานที่จะต้องขึ้นอยู่กับตัวผู้ใช้โปรแกรมด้วยว่าจะใช้งานออกมาได้ตรงตามวัตถุประสงค์ ในกรณีของการทำงานกับภาพบิตแมบชนิดอินเด็กซ์นั้น ซึ่งเป็นภาพที่มีจำนวนสีไม่แน่นอน อาจจะเป็นไปได้ว่ามีจำนวนบิตต่อสี ได้ตั้งแต่ 1 บิต(2สี) 4 บิต(16สี) 8บิต(256สี) นั้นโปรแกรมจะส่วนใหญ่ไม่ทำกระบวนการซึ่งไม่ตรงตามมาตรฐานของบิตแมบ คือ 24 บิตบนวินโดวส์นั้นเป็นแบบ 32 บิตนั้นถ้าประมวลผลภาพที่มีบิตน้อยๆ จะทำการเพิ่ม Padding Bits เข้าไปซึ่งจะทำให้ภาพอิมเมจผลลัพธ์นั้นคลาดเคลื่อนนั้นเป็นไปได้สูง ส่วนการประมวลผลของโปรแกรมนั้นบางภาพนำมาเข้ากับบางฟังก์ชันก็ไม่ได้หมายความว่าคุณภาพจะต้องออกมาดีเสมอไป เช่น ภาพที่มีความมืดมาก ถ้านำมาInvert หรือ ทำ GrayScale แล้วภาพที่ออกมาจะดูไม่ค่อยรู้เรื่อง หรือบางภาพเมื่อนำมาปรับปรุงด้วยการทำ Histogram Equalized แล้วอาจจะออกมาชัดเจนขึ้น เพราะตัวฟังก์ชันนี้จะทำการเฉลี่ยค่าสีให้ออกมาเท่าๆ กัน แต่บางภาพหลังจากทำแล้วผลที่ออกมาไม่ชัดเจน หรือ อาจจะดูไม่รู้เรื่องได้ ส่วนของการสร้างภาพใหม่ถ้าเกิดภาพอิมเมจไม่มีความกว้างและความสูงและความลึกที่เท่ากันพอดีก็จะไม่สามารถสร้างภาพผลลัพธ์ได้ เนื่องจาก จะเกิดปัญหาของการซ้อนทับกันของภาพในส่วนที่ไม่เท่ากัน ถ้าเกิดนำภาพทั้งสองที่มีความกว้าง ความสูงและความลึก มาแต่ไม่มีความใกล้เคียงกันทางด้านโทนสีก็จะทำให้สีของภาพผลลัพธ์อาจจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพี้ยนได้ ดังนั้น ก่อนการสร้างภาพใหม่ผู้ใช้จะต้องนำภาพที่มีความกว้างและความสูงที่เท่ากันและมีเฉดสีใกล้เคียงกันมาเป็นอินพุตซึ่งจะทำให้ภาพผลลัพธ์ที่ออกมามีความแปลกตาและมีประสิทธิภาพตามวัตถุประสงค์ ส่วนของการหาขอบภาพนั้น ก็ขึ้นอยู่กับภาพที่นำมาเป็นอินพุตนั้นจะเป็นอย่างไรซึ่งถ้าภาพบางภาพมีขอบเขตและรายละเอียดที่ต้องการเน้นไม่มากเกินไปก็จะทำให้โปรแกรมสามาเน้นขอบและลวดลายออกมาได้อย่างชัดเจนมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อเสนอแนะ

สรุปผล

การพัฒนาโปรแกรมดูภาพและปรับปรุงคุณภาพของภาพ ได้ดำเนินตามวัตถุประสงค์ และประโยชน์ต่างๆ ตามที่ได้วางไว้ คือ สามารถนำความรู้ทางด้านคอมพิวเตอร์กราฟิก และ ความรู้ทางด้านการประมวลผลภาพมาประยุกต์ใช้ ช่วยให้ผู้ใช้ได้รับความสะดวกมากขึ้น

อนึ่งการพัฒนาของโปรแกรมนี้ยังมีข้อเสนอแนะอีกส่วนหนึ่งซึ่งจะเป็นแนวทางในการพัฒนาอีกต่อไป กล่าวคือ

ข้อจำกัดของโปรแกรมที่ต้องทำงานบน Windows XP เท่านั้นเพราะจากสถาปัตยกรรมใหม่ของดอทเนท (.NET)

การที่จะใช้ภาษา Visual C++ ในการพัฒนา จะต้องใช้เวลาในการศึกษาในระยะหนึ่งพอสมควรและจะต้องมีความเข้าในในการเขียนโปรแกรมติดต่อกับฮาร์ดแวร์ เช่น พรินท์เตอร์

ข้อเสนอแนะและแนวทางการพัฒนาต่อ

เนื่องจากโปรแกรมเป็นการเขียนโปรแกรมเชิงวัตถุ (OOP: Object Oriented Programming) ดังนั้นการที่จะนำไปพัฒนาต่อหรือนำไปใช้เป็นองค์ประกอบกับโปรแกรมอื่นเพื่อเพิ่มประสิทธิ์ภาพได้

การที่จะเพิ่มประสิทธิภาพในการพัฒนาโปรแกรมนี้ต่อๆไปนั้น ผู้ที่จะพัฒนาควรจะศึกษาในเรื่อง Advance Computer Graphics และ Image Processing และการเขียนโปรแกรมขั้นสูง

สำหรับอัลกอริทึมที่ใช้ในการประมวลผลภาพนั้น อาจมีแนวคิดหรือมีอีกหลายวิธีที่มีประสิทธิภาพมากกว่าอัลกอริทึมที่ใช้ในปัญหาพิเศษนี้ ซึ่งมีมากมายแล้วก็ให้ผลลัพธ์ออกมาเหมือนกัน ดังนั้นผู้ที่จะศึกษาต่อต้องควรเลือกอัลกอริทึมที่มีประสิทธิภาพมีความยืดหยุ่น ไม่ซับซ้อนมากในการประมวลผล ผิดพลาดน้อยที่สุด และเป็นที่ยอมรับอย่างกว้างขวาง มาใช้ศึกษา

ส่วนติดต่อกับผู้ใช้ การวางตำแหน่งของทูลบาร์ควรจัดให้ใช้ได้ง่าย และควรนำออกมาจากเมนูมาเป็นทูลบาร์เพื่อสะดวกต่อการใช้งาน และ ความรวดเร็ว

โดยโปรแกรมที่เป็นต้นแบบนี้ ยังต้องมีการพัฒนาให้ดียิ่งขึ้นและสิ่งที่ควรแก้ไขดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ควรเพิ่มในส่วนของการบีบอัดภาพและแปลงภาพให้มีขนาดและลงได้ ก็จะทำให้โปรแกรมมีความสมบูรณ์ในตัวเองมากขึ้น
2. การทำ Undo ควรมีหลายๆ ขั้นตอน และ ควรมีการ Redo ควรมีการถามผู้ใช้ว่าควรมีการบันทึกก่อนการปิดหรือไม่
3. การกระพริบของหน้าจอแสดงผลภาพอันเกิดจากการย่อและขยายของหน้าต่างแสดงผล ซึ่งผู้พัฒนาพยายามสืบค้นและหาทางแก้ไขอยู่ จึงหวังว่าถ้ามีผู้นำไปพัฒนาต่อจะสามารถบรรลุปัญหานี้ได้
4. ข้อจำกัดของระบบในการรันโปรแกรม เนื่องจากผู้พัฒนาได้ใช้ Visual C++.Net ซึ่งใช้สถาปัตยกรรมใหม่ของ MFC ดังนั้นจึงไม่สามารถเปิดรันโปรแกรมนี้ได้บน Visual C++6 หรือต่ำกว่าได้
5. จากข้อจำกัดของเวลาและความสามารถของผู้พัฒนาที่ต้องการพัฒนาโปรแกรมนี้ให้ประมวลผลเป็นแบบใช้เธรดซึ่งเป็นเรื่องของการเขียนโปรแกรมที่ซับซ้อนมาก ซึ่งถ้าทำการพัฒนาให้สำเร็จบรรลุได้ก็จะทำให้โปรแกรมทำงานได้เร็วขึ้นและทำงานบางอย่างได้พร้อมๆกันในเวลาเดียวกันได้ เพราะจะสร้างเธรดงานที่ใช้ในการประมวลผลภาพโดยเฉพาะ

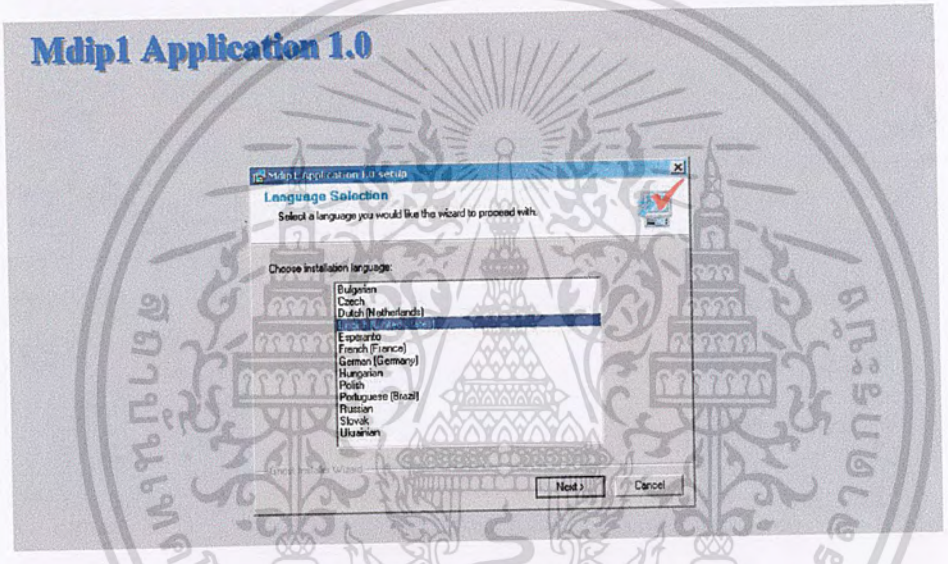
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก การติดตั้งโปรแกรมและตัวอย่างชุดภาพ

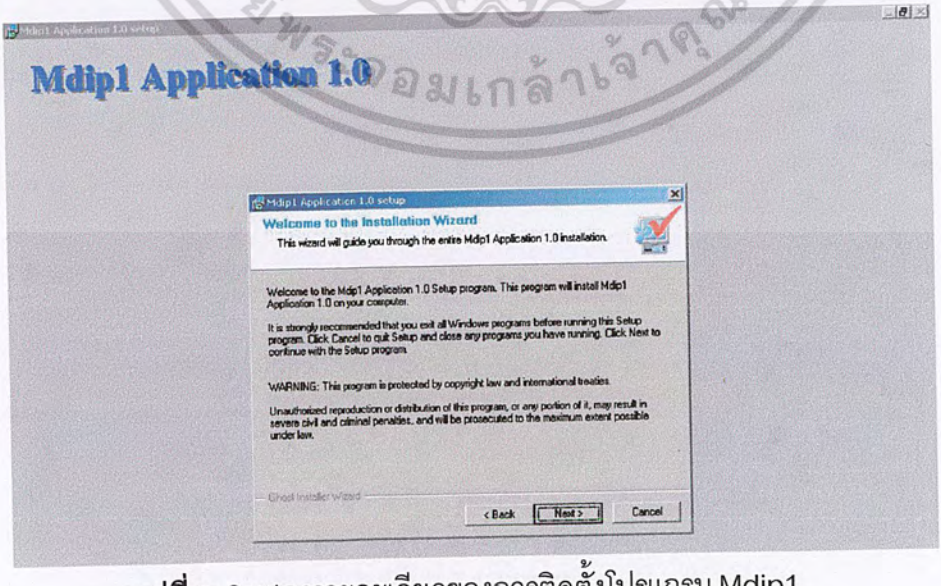
1.การติดตั้งโปรแกรม

ให้ดับเบิ้ลคลิกตัว Install และจะมีข้อความตามรูปจะปรากฏหน้าจอแสดงการติดตั้ง ให้เลือกภาษา เมื่อกด Next ก็จะปรากฏหน้าจอ intro แล้วให้ผู้ใช้เลือกพื้นที่ๆ โปรแกรมต้องใช้ทำการติดตั้ง จากนั้นให้กด Next ไปเรื่อยๆ ตามลำดับของการติดตั้ง และจะทำการเข้าสู่หน้าจอที่ทำการติดตั้ง

เมื่อติดตั้งเสร็จแล้ว เราจะเข้าสู่โปรแกรม Mdip1 ของเราได้ โดยให้ไปที่ Start Menu ของ Windows แล้วไปที่ Program แล้วเลือก คลิกที่ชื่อโปรแกรมเพื่อเข้าสู่โปรแกรม



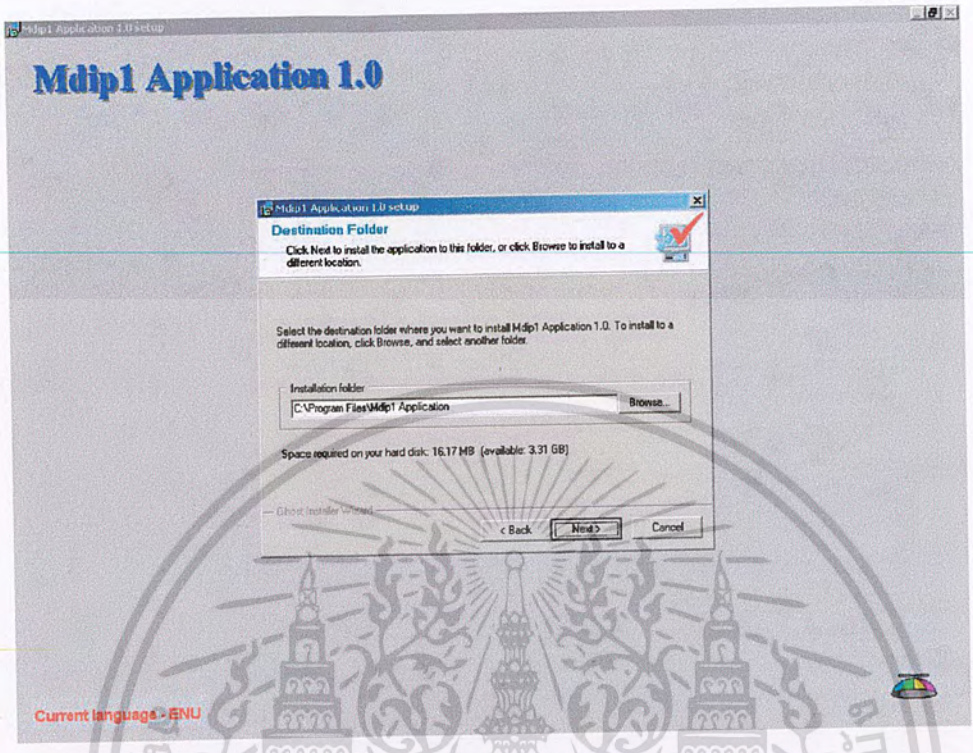
รูปที่ ก-1 แสดงการติดตั้งโปรแกรม Mdip1



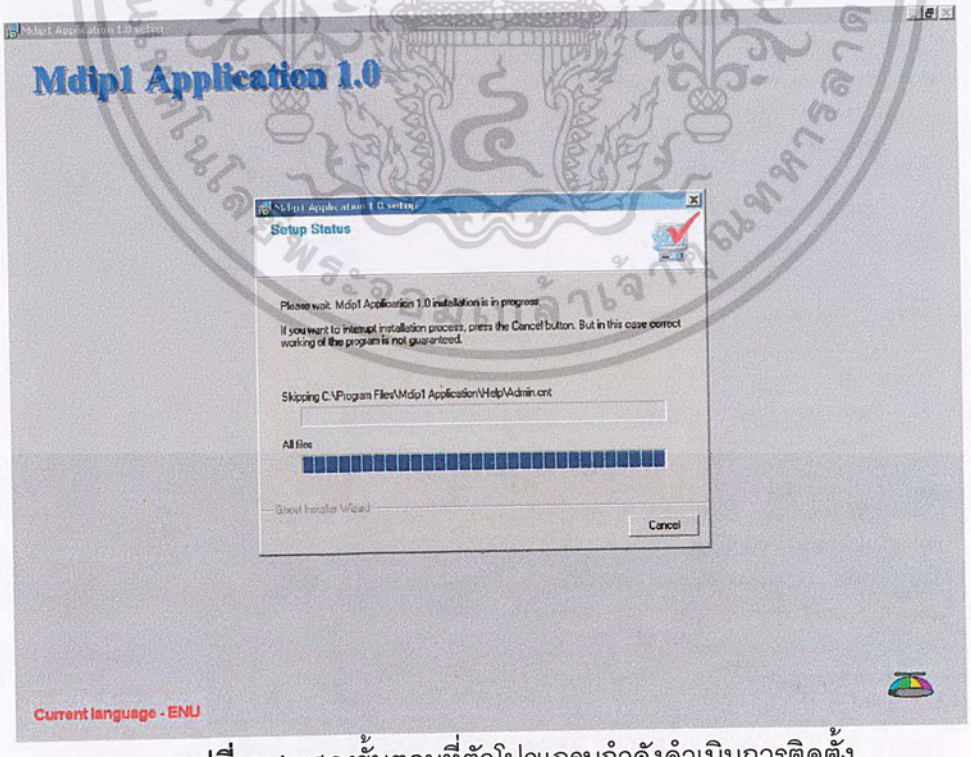
รูปที่ ก-2 แสดงรายละเอียดของการติดตั้งโปรแกรม Mdip1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้เรากด Next ไปเรื่อยๆ โปรแกรมจะแสดงหน้าจอการติดตั้ง โดยเราสามารถกด Browse เพื่อเลือกDriveที่เราต้องการลงโปรแกรมแล้วกด Next

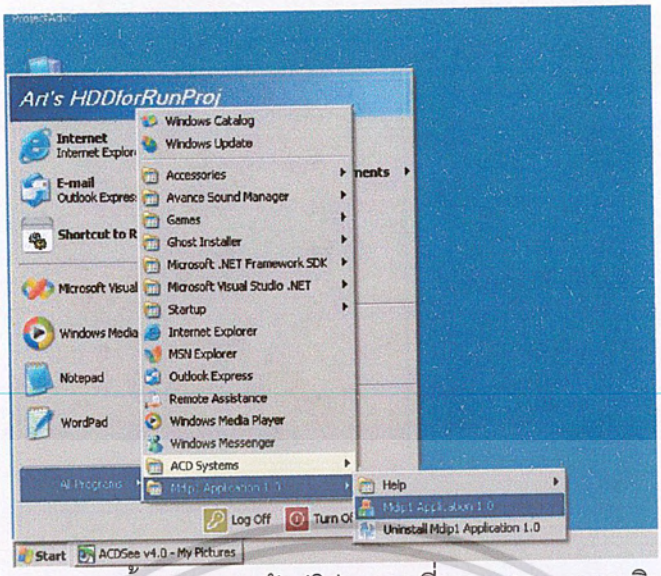


รูปที่ ก-3 แสดงการเลือกพื้นที่ๆ เราต้องการติดตั้งลงไป



รูปที่ ก-4 แสดงขั้นตอนที่ตัวโปรแกรมกำลังดำเนินการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



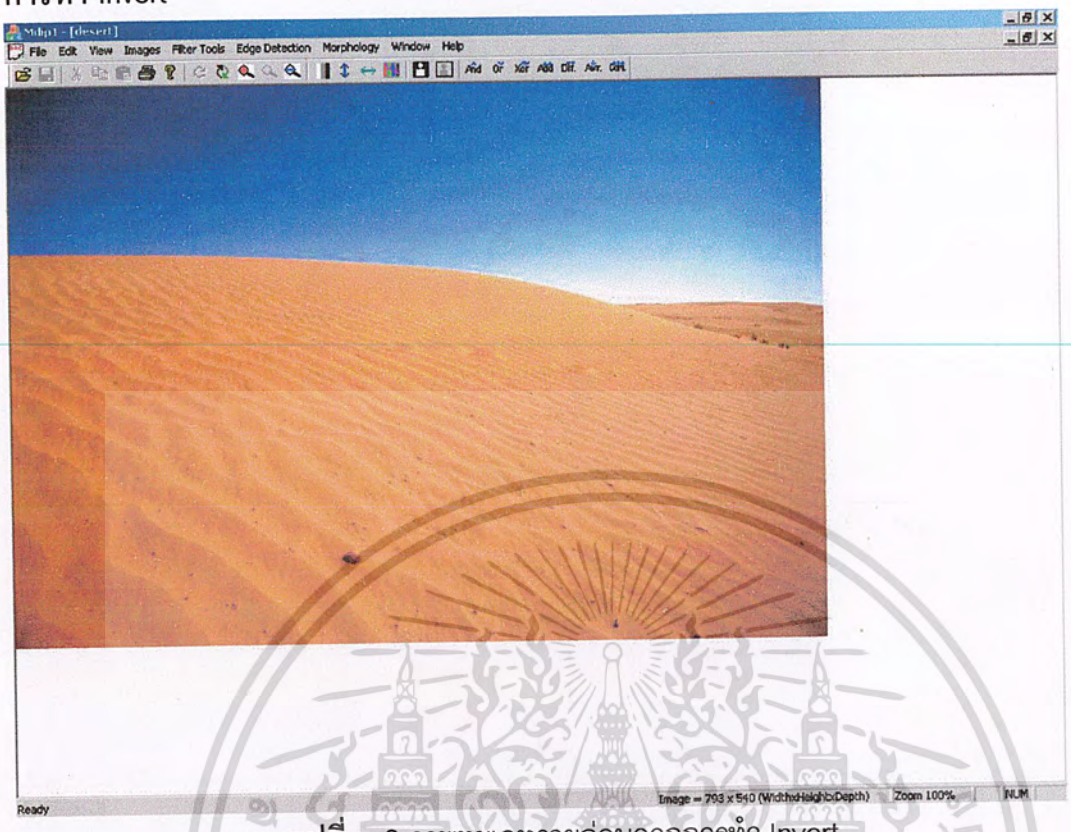
รูปที่ ก-5 แสดงขั้นตอนการเข้าสู่โปรแกรมที่ Start Menu ของวินโดวส์

2.ตัวอย่างชุดภาพ

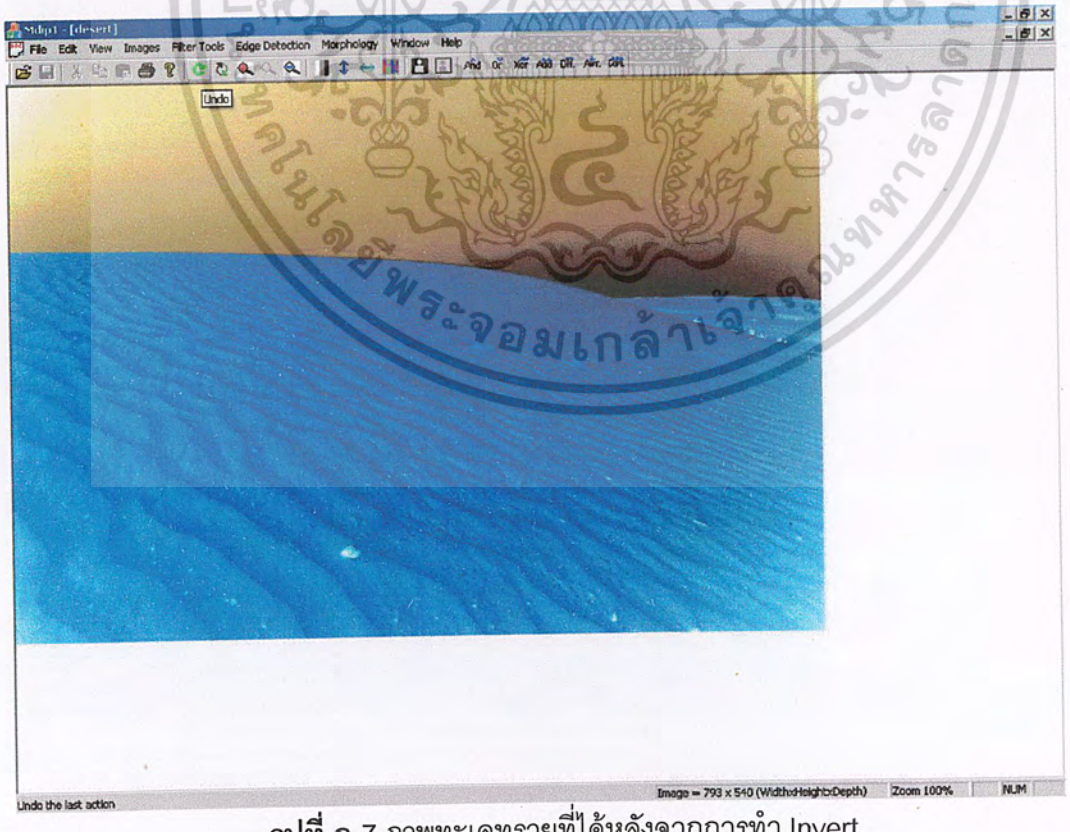
โดยภาพในภาพผนวกนี้เป็นตัวอย่างการนำภาพจากที่ต่างๆ มาปรับแต่งโดยโปรแกรม โดยจะแสดงภาพก่อนการแต่ง และ หลังการแต่งด้วยฟังก์ชันต่างๆ ภายในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำ Invert

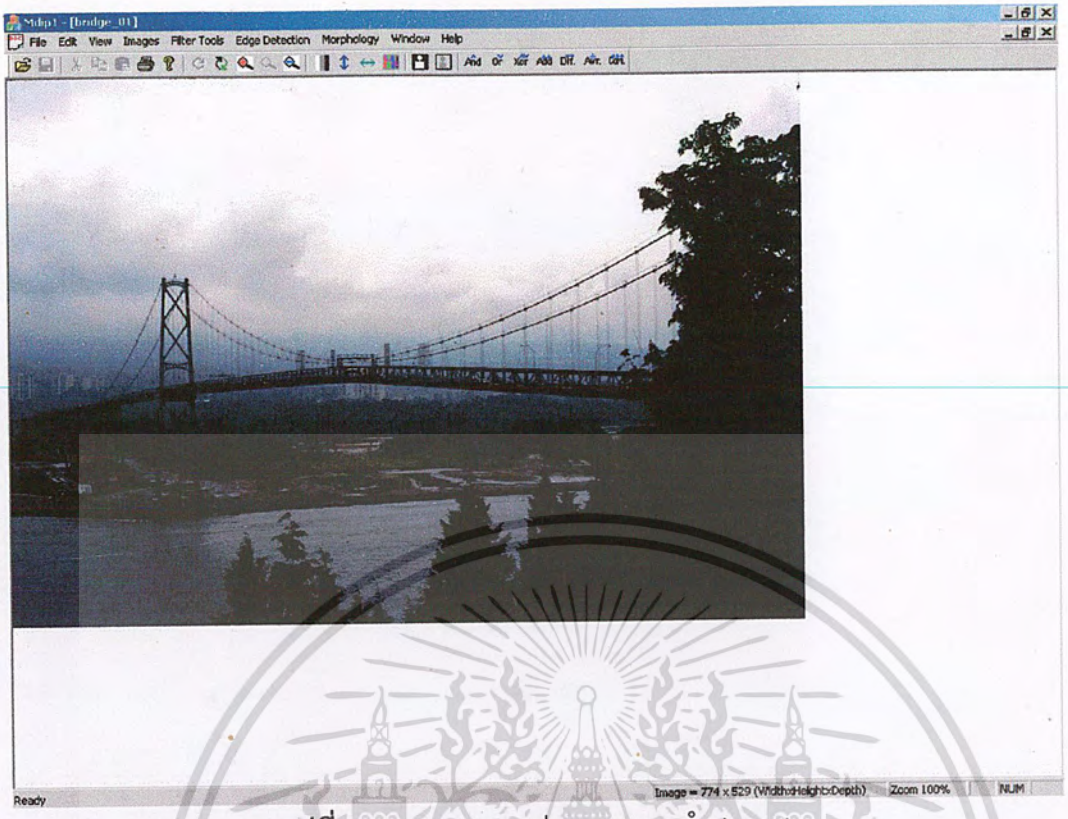


รูปที่ ก-6 ภาพทะเลทรายก่อนจากการทำ Invert



รูปที่ ก-7 ภาพทะเลทรายที่ได้หลังจากการทำ Invert

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

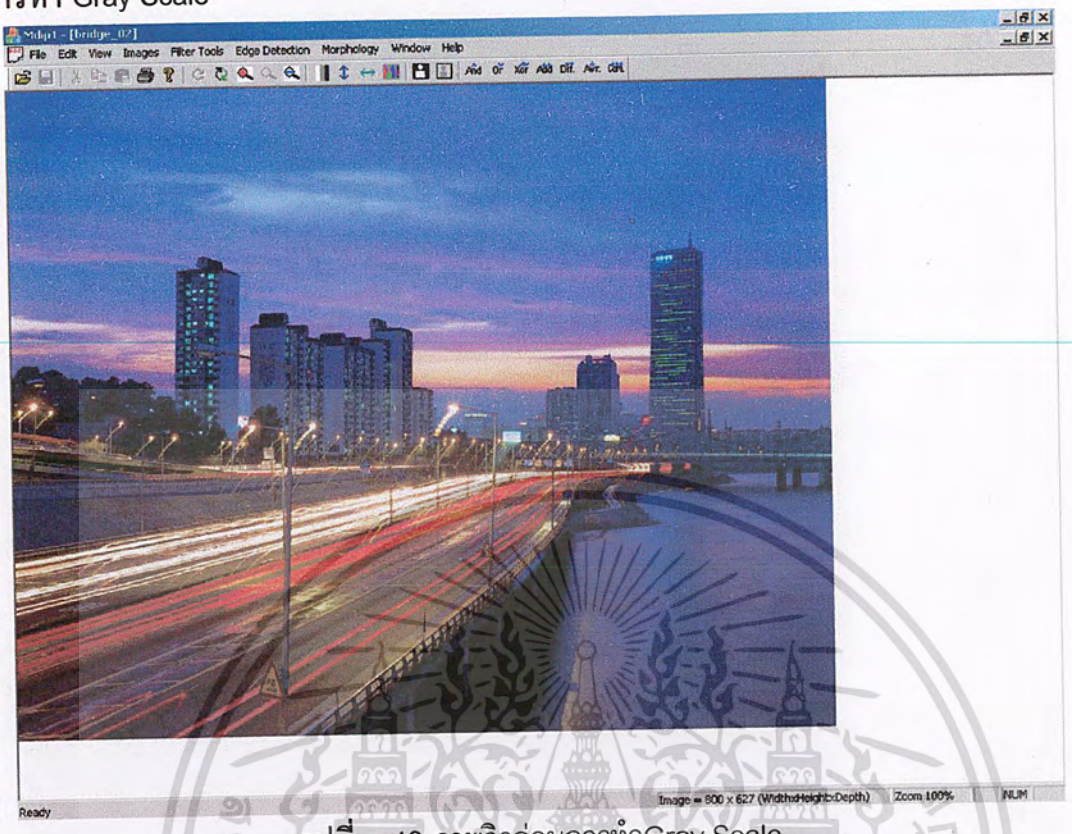


รูปที่ ก-8 ภาพสะพานก่อนจากการทำ Invert

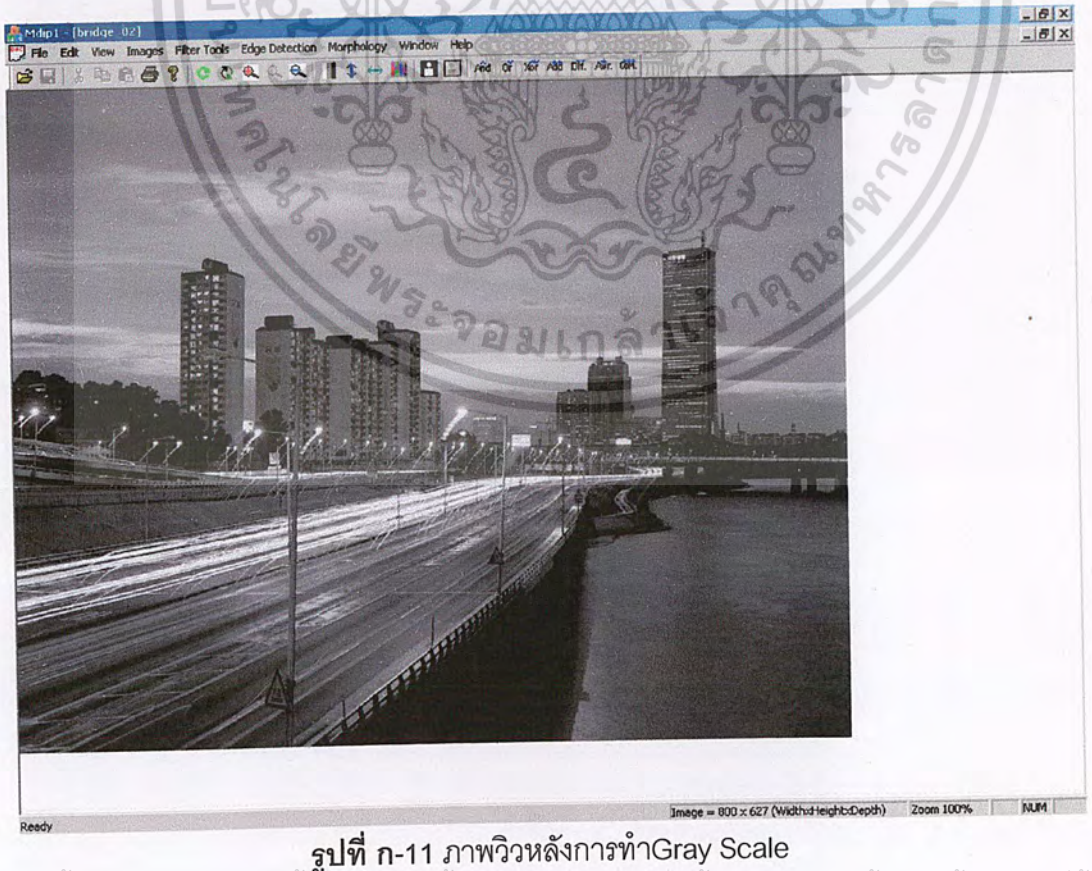
รูปที่ ก-9 ภาพสะพานที่ได้หลังจากการทำ Invert

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำ Gray Scale

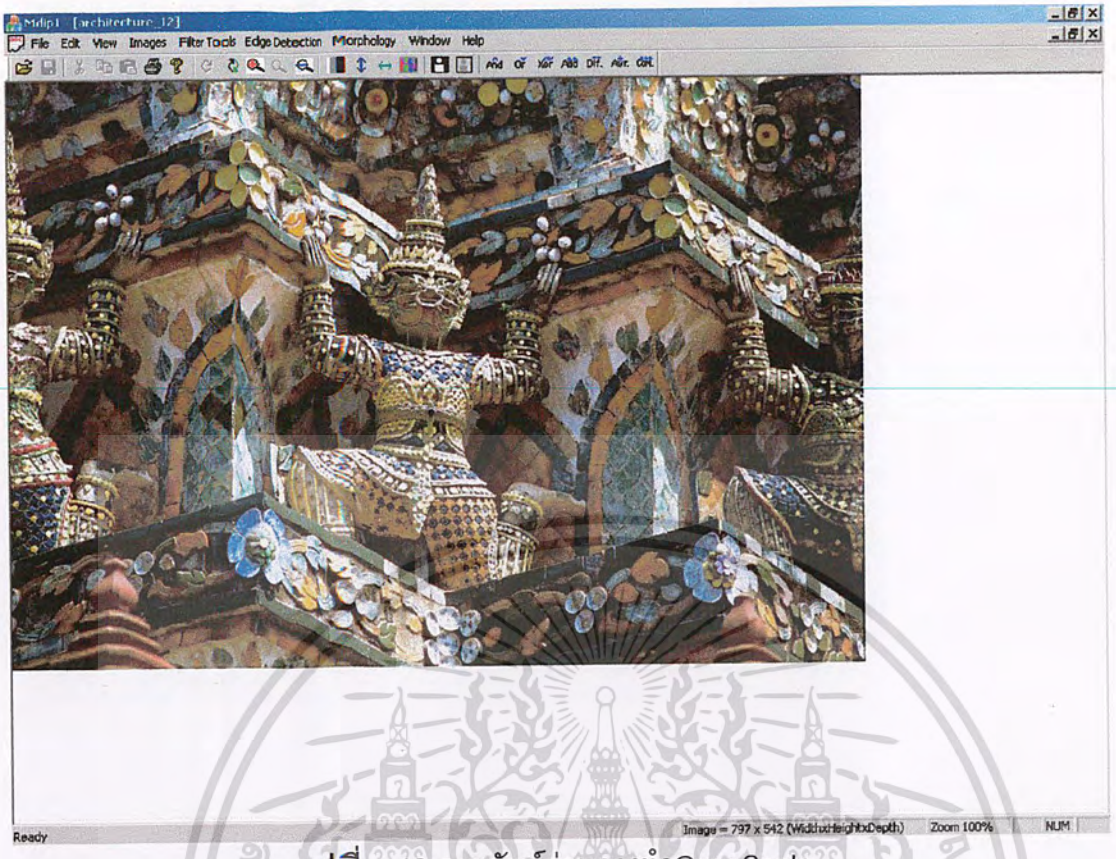


รูปที่ ก-10 ภาพวิวก่อนการทำGray Scale



รูปที่ ก-11 ภาพวิวหลังการทำGray Scale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

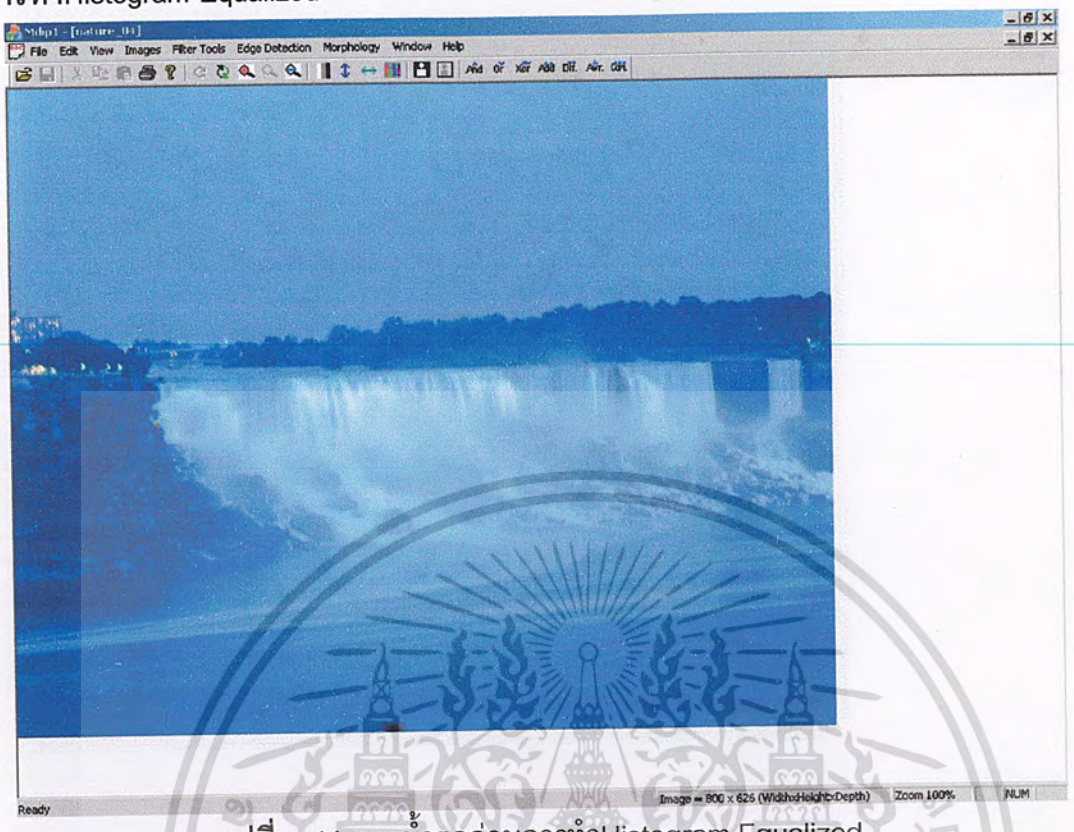


รูปที่ ก-12 ภาพยักษ์ก่อนการทำGray Scale

รูปที่ ก-13 ภาพยักษ์หลังการทำGray Scale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

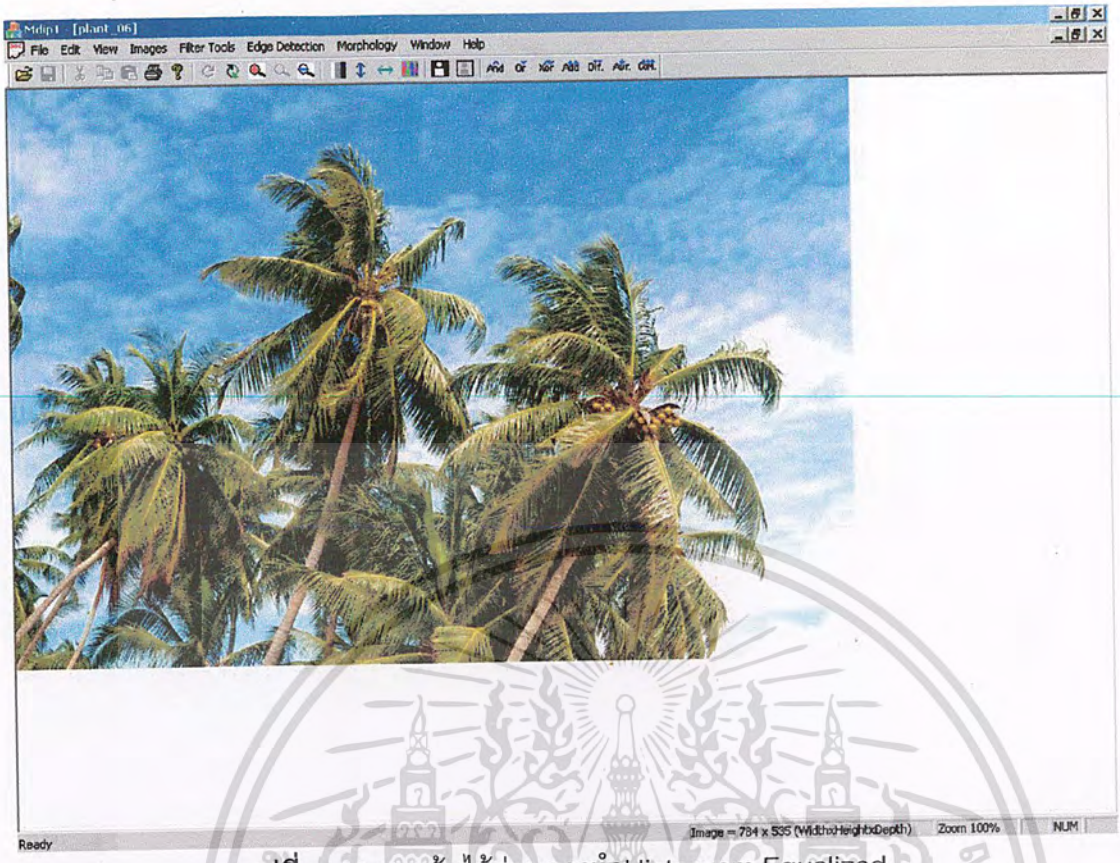
การทำHistogram Equalized



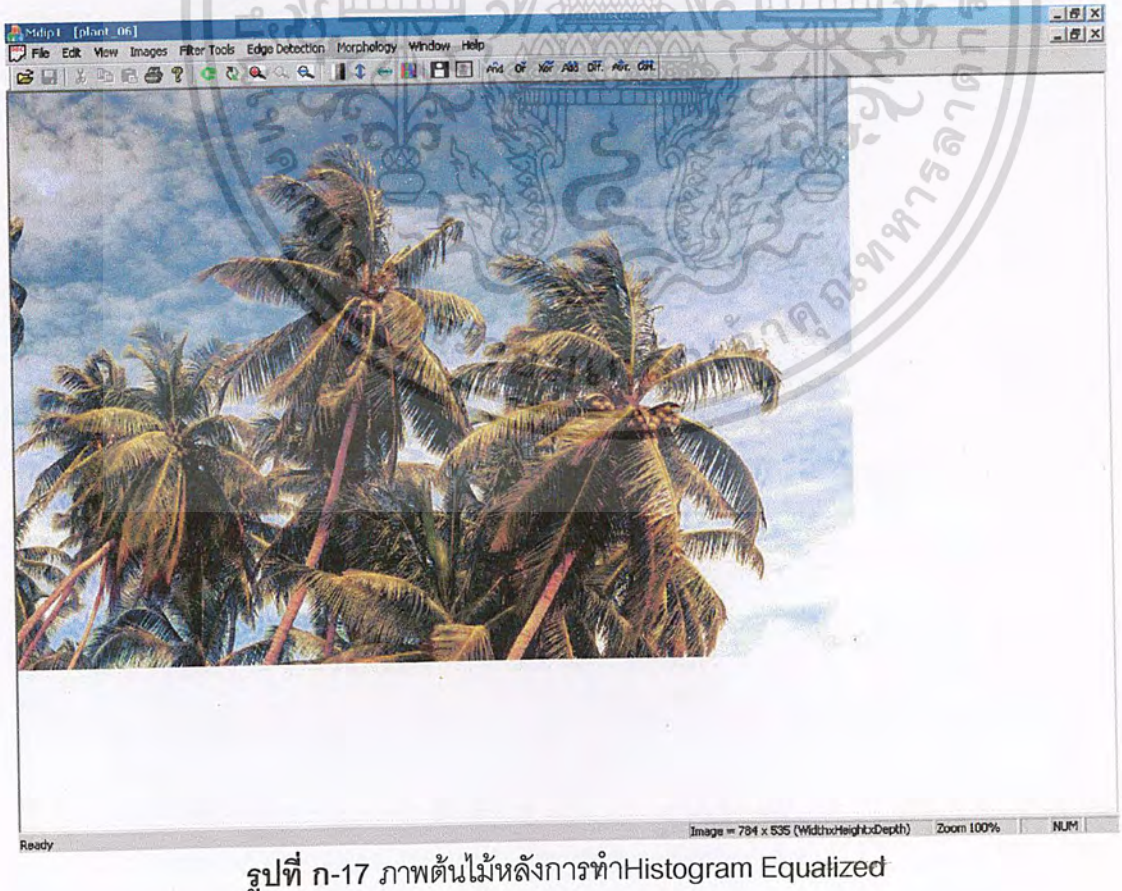
รูปที่ ก-14 ภาพน้ำตกก่อนการทำHistogram Equalized

รูปที่ ก-15 ภาพน้ำตกหลังการทำHistogram Equalized

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



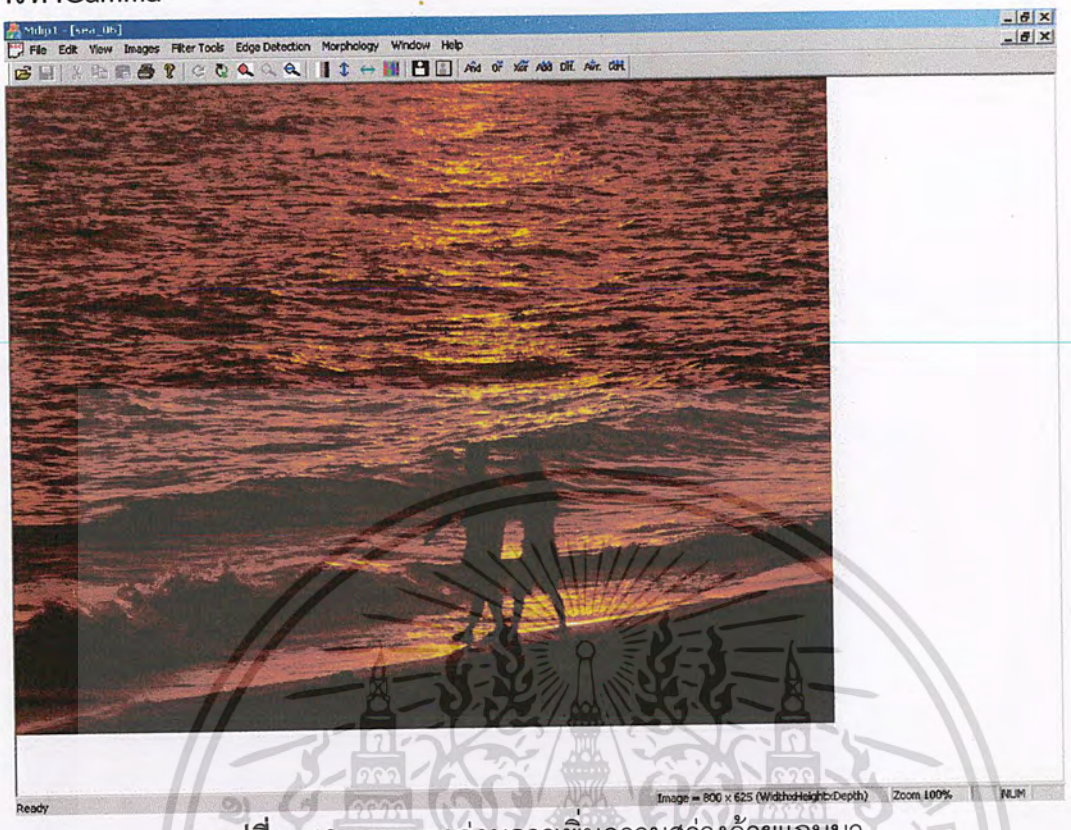
รูปที่ ก-16 ภาพต้นไม้ก่อนการทำHistogram Equalized



รูปที่ ก-17 ภาพต้นไม้หลังการทำHistogram Equalized

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำGamma

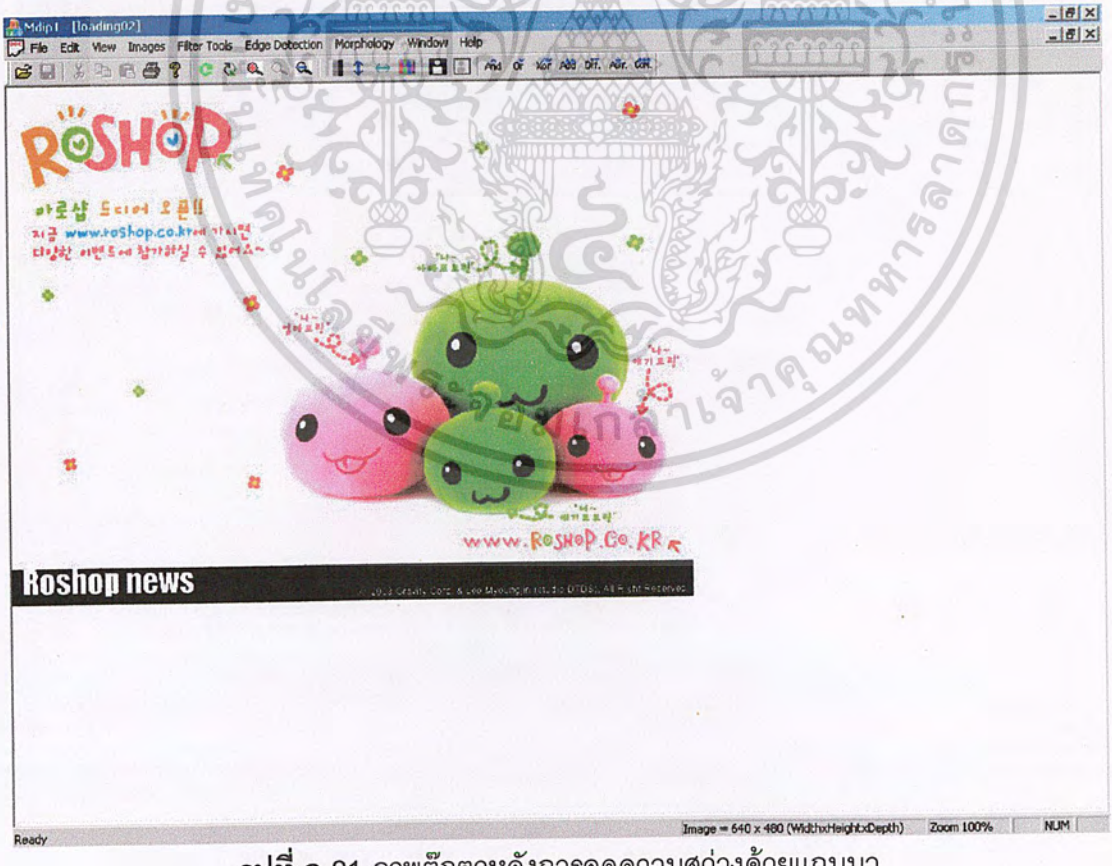


รูปที่ ก-18 ภาพทะเลก่อนการเพิ่มความสว่างด้วยแกมมา

รูปที่ ก-19 ภาพทะเลหลังการเพิ่มความสว่างด้วยแกมมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

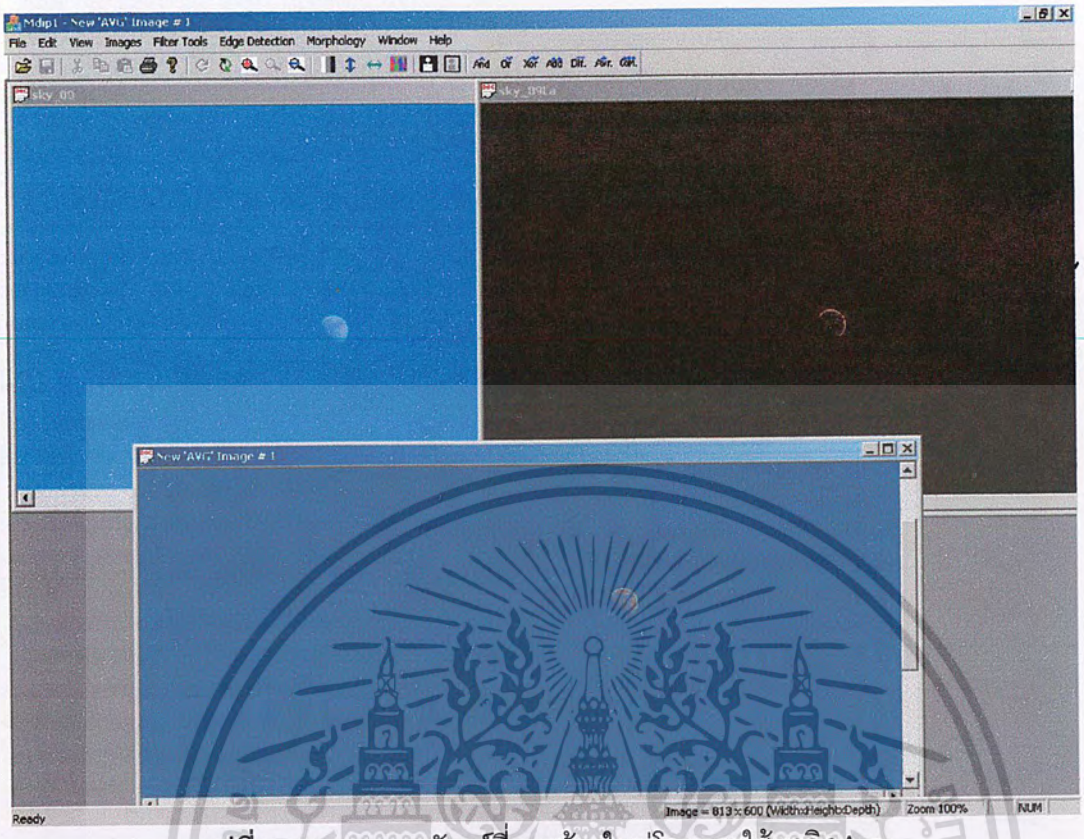
รูปที่ ก-20 ภาพตุ๊กตาก่อนการลดความสว่างด้วยแกมมา



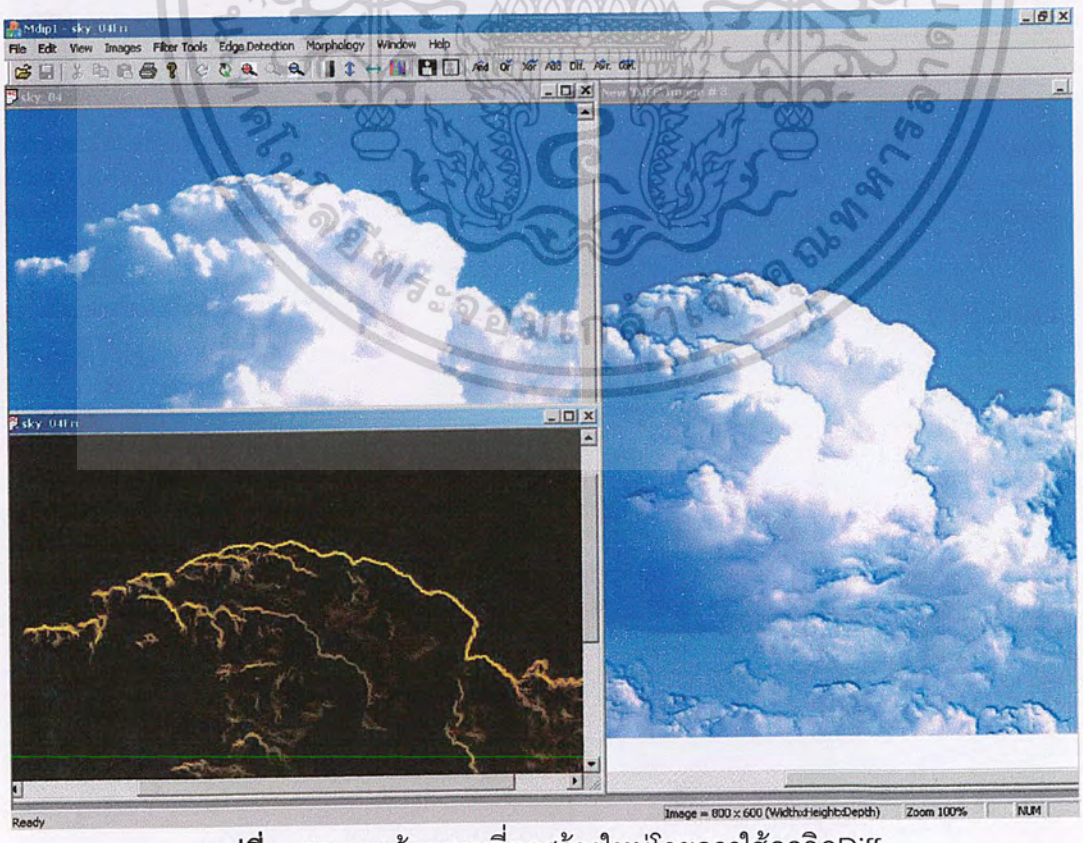
รูปที่ ก-21 ภาพตุ๊กตาหลังการลดความสว่างด้วยแกมมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างภาพใหม่โดยการใช้ลอจิค

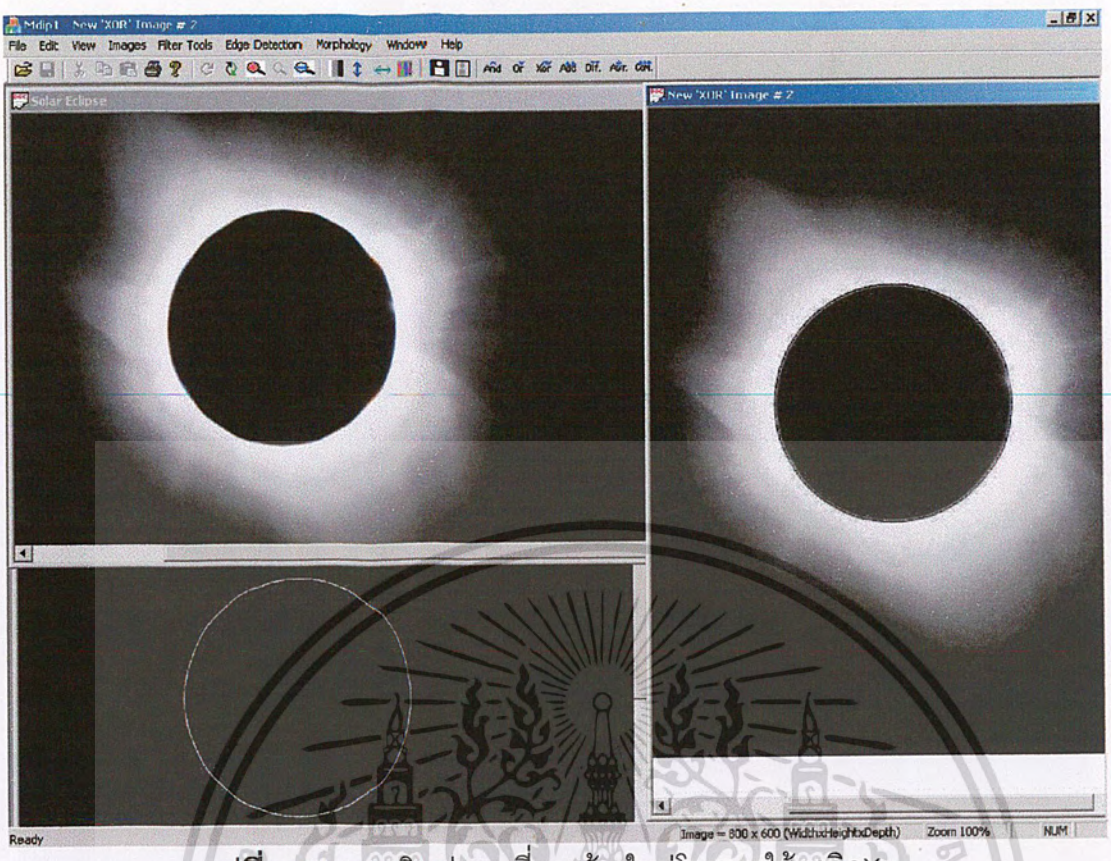


รูปที่ ก-22 ภาพดวงจันทร์ที่ถูกสร้างใหม่โดยการใช้ลอจิคAvg



รูปที่ ก-23 ภาพก้อนเมฆที่ถูกสร้างใหม่โดยการใช้ลอจิคDiff

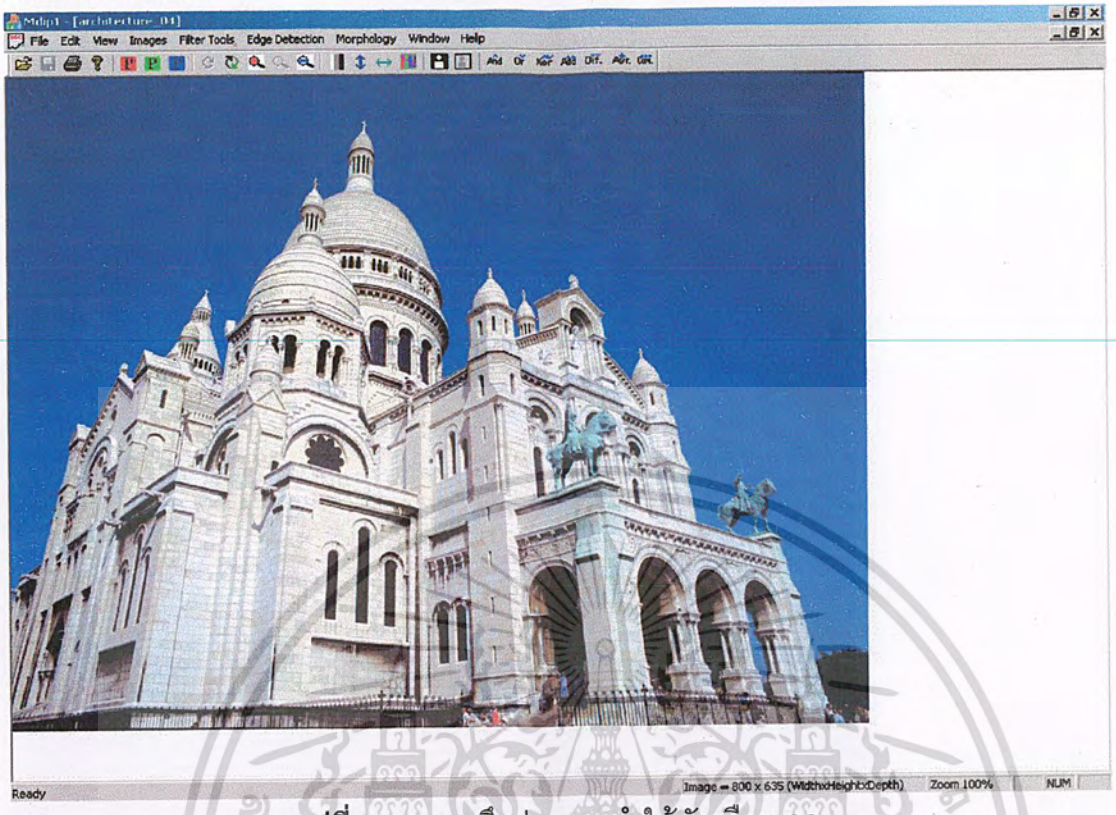
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



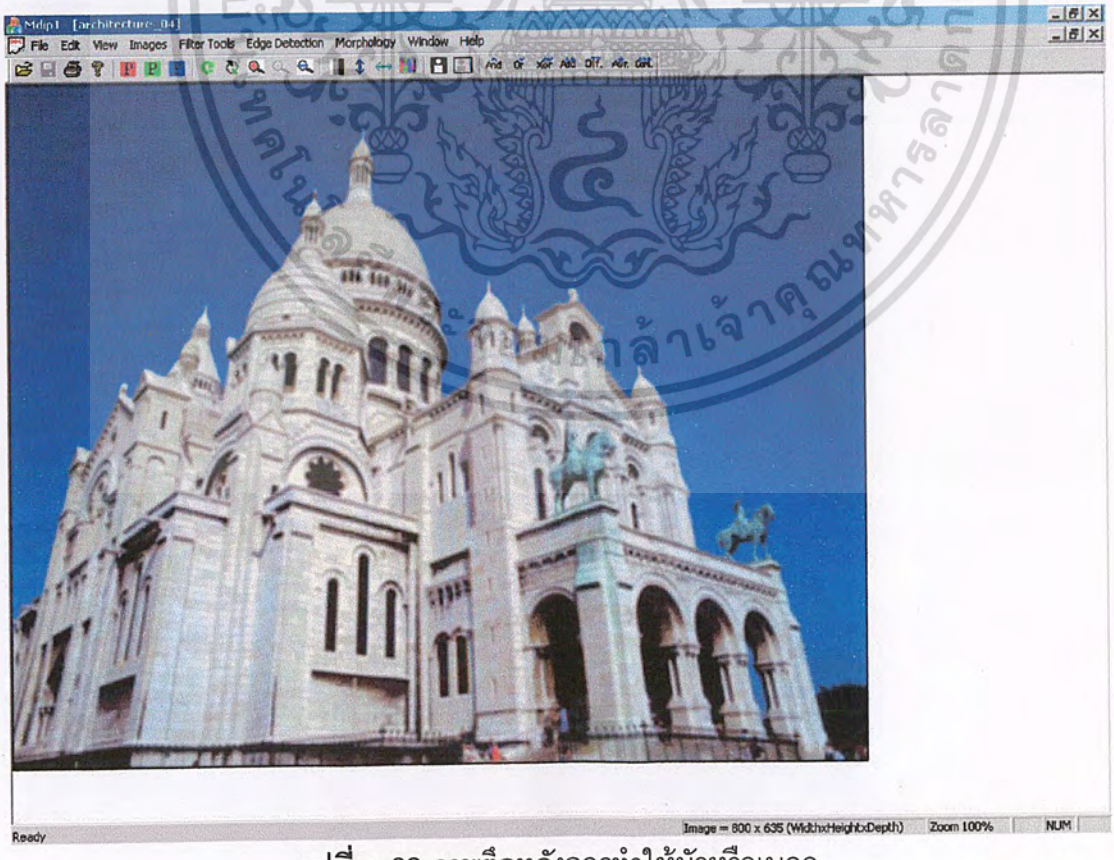
รูปที่ ก-24 ภาพสุริยุปราคาที่ถูกสร้างใหม่โดยการใช้ลอจิคXor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำLow Pass

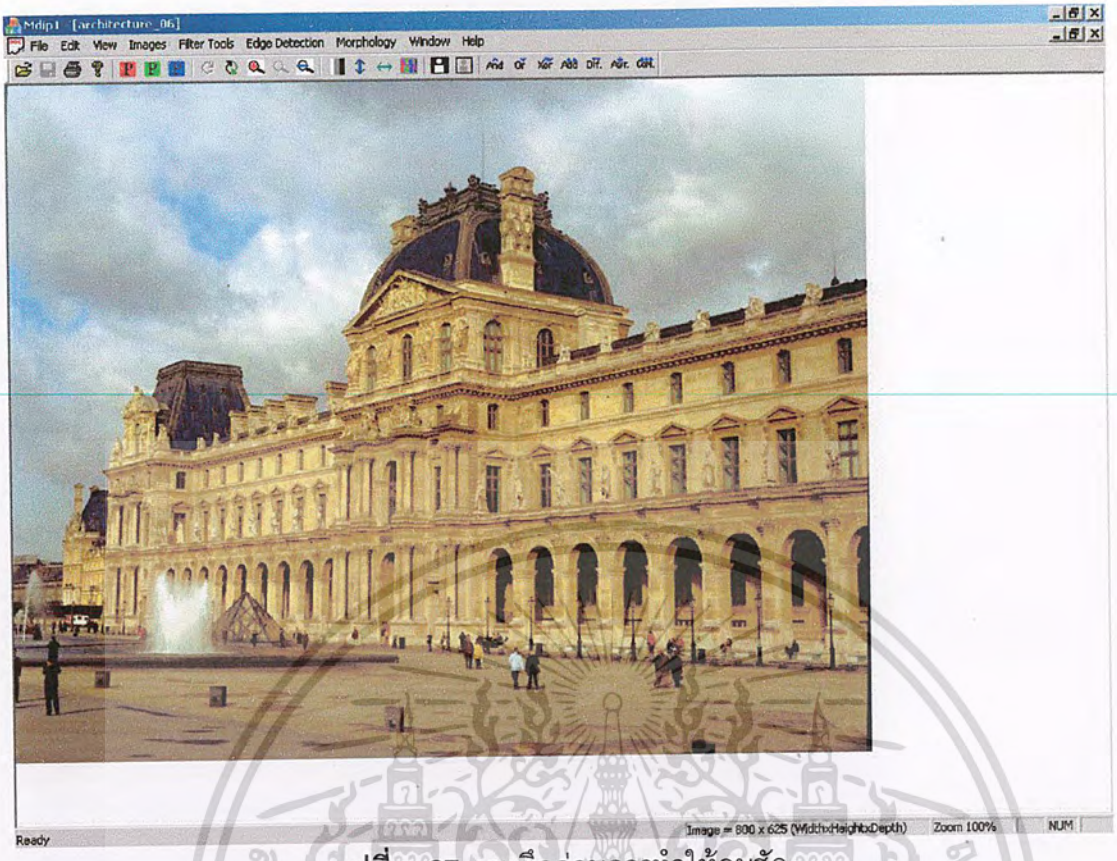


รูปที่ ก-25 ภาพตึกก่อนการทำให้มัวหรือเบลอ



รูปที่ ก-26 ภาพตึกหลังการทำให้มัวหรือเบลอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

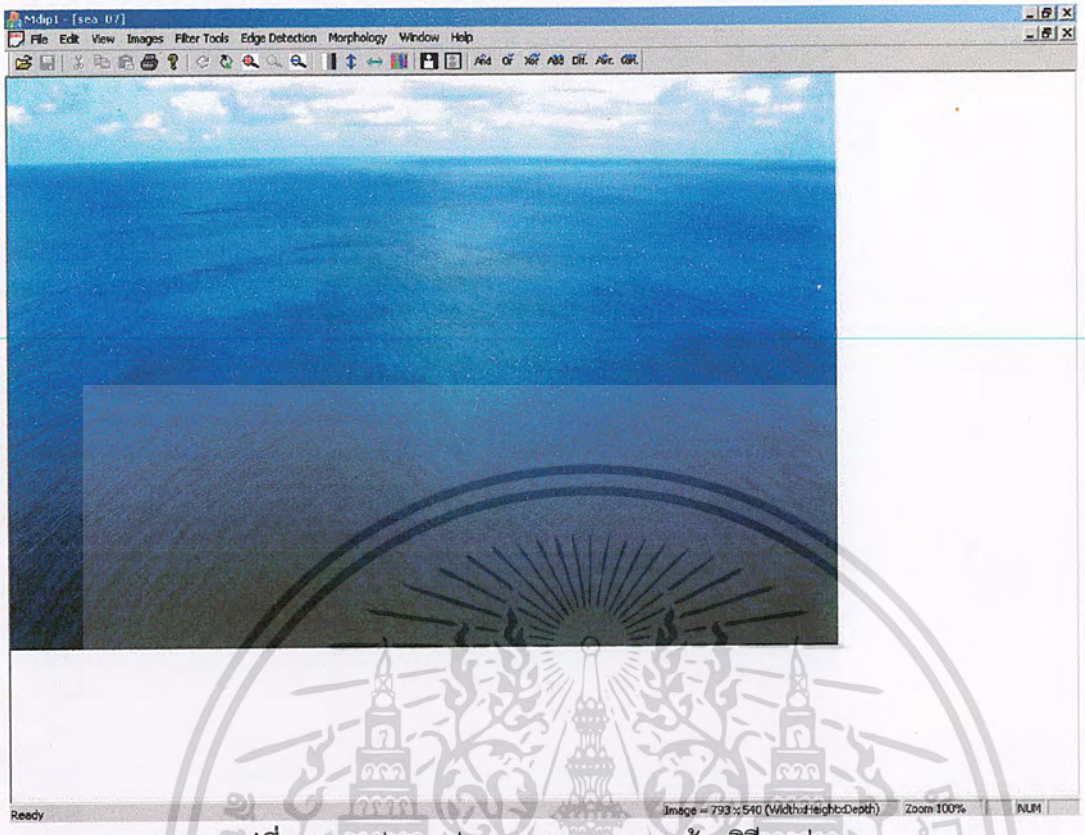


รูปที่ ก-27 ภาพตึกก่อนการทำให้คมชัด

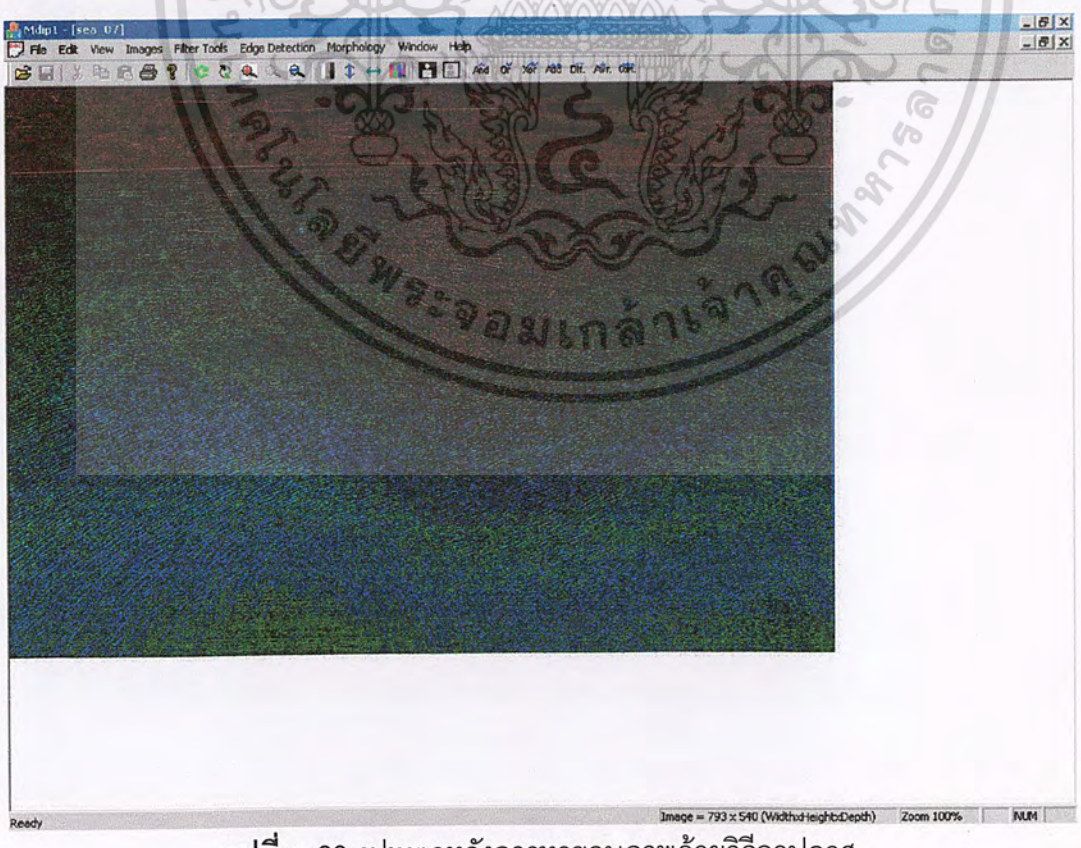
รูปที่ ก-28 ภาพตึกหลังการทำให้คมชัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบภาพด้วยวิธีลาปลาส

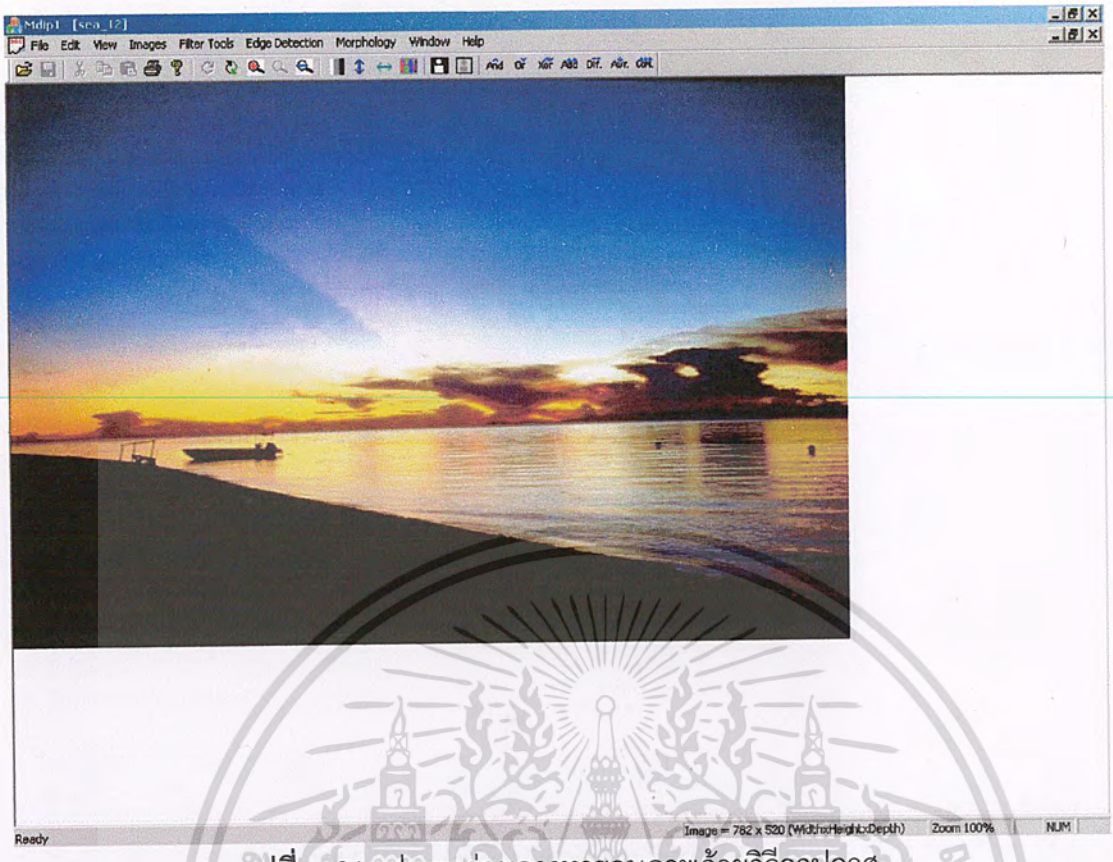


รูปที่ ก-29 รูปทะเลก่อนการหาขอบภาพด้วยวิธีลาปลาส

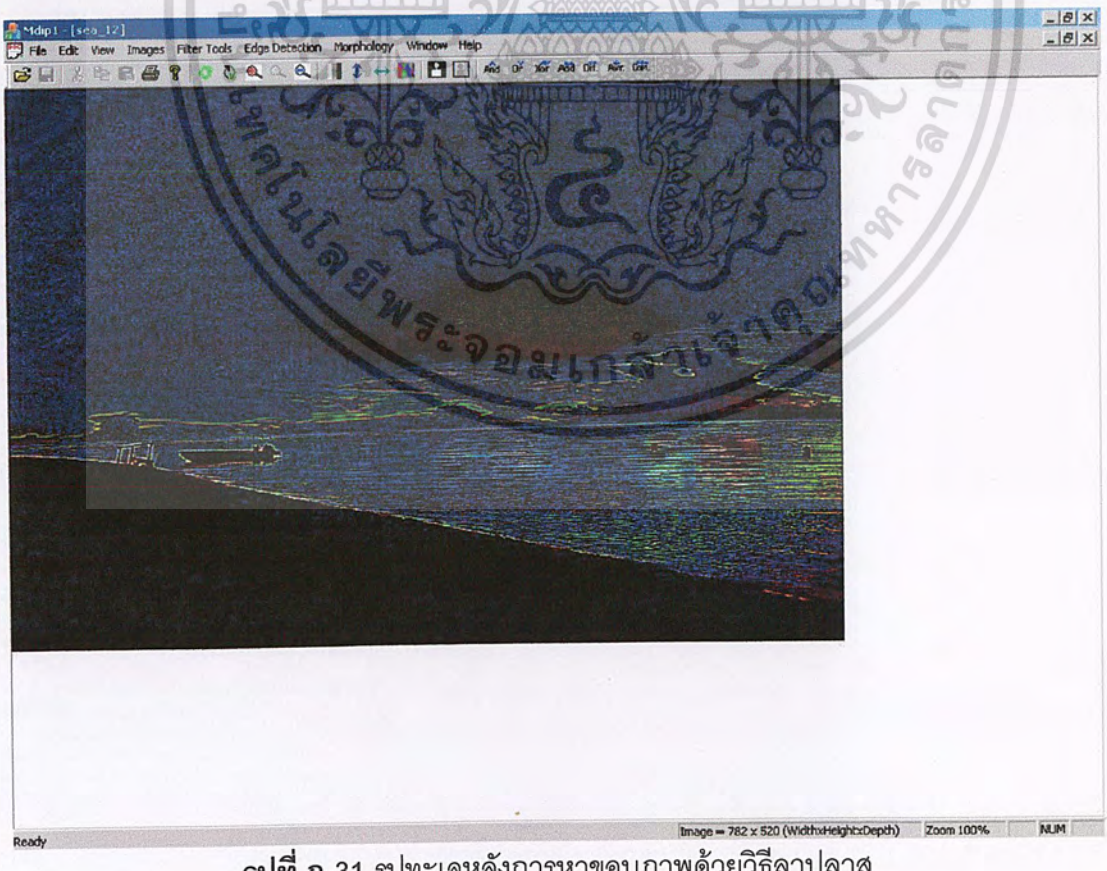


รูปที่ ก-30 รูปทะเลหลังการหาขอบภาพด้วยวิธีลาปลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



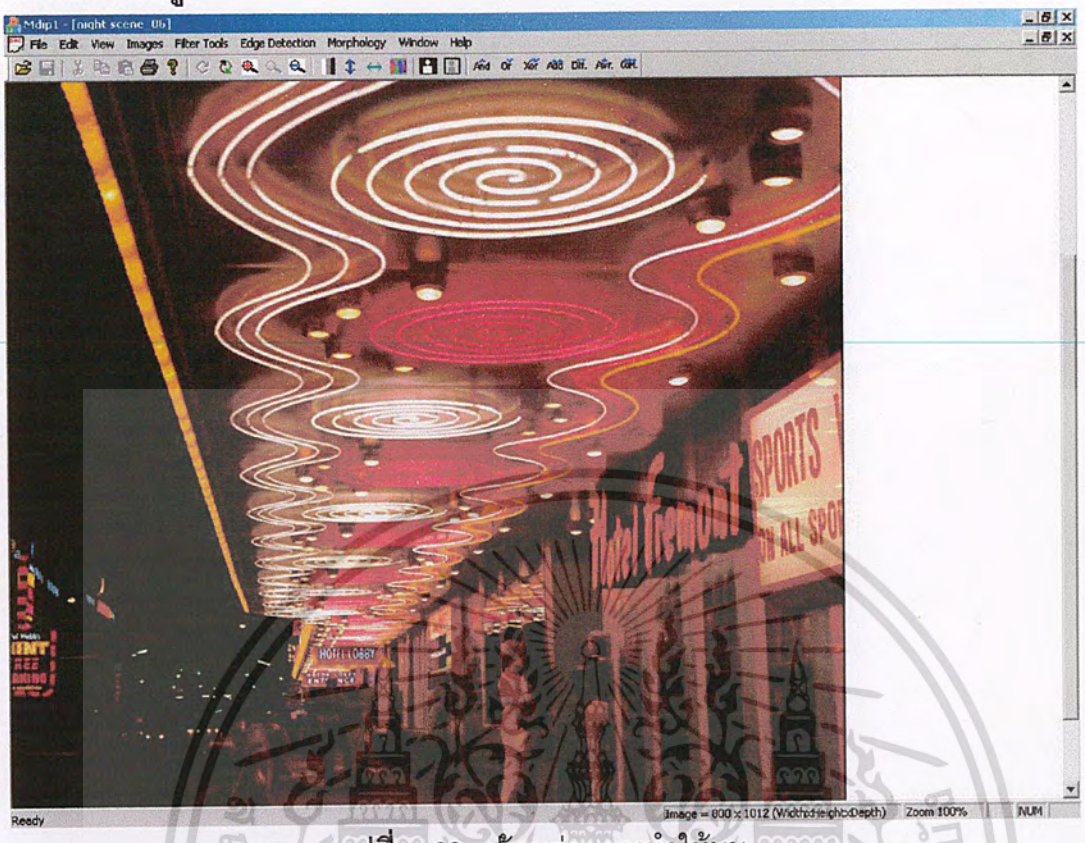
รูปที่ ก-31 รูปทะเลก่อนการหาขอบภาพด้วยวิธีลาปลาส



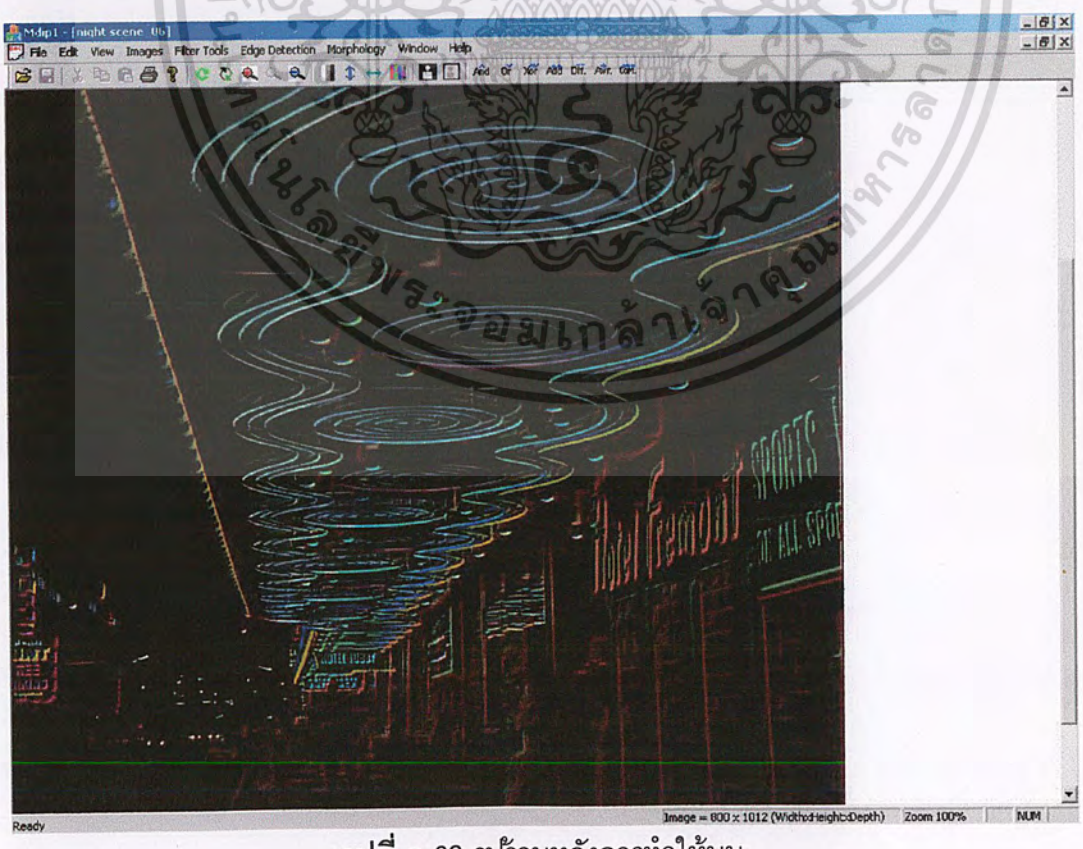
รูปที่ ก-31 รูปทะเลหลังการหาขอบภาพด้วยวิธีลาปลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำให้ภาพนูน



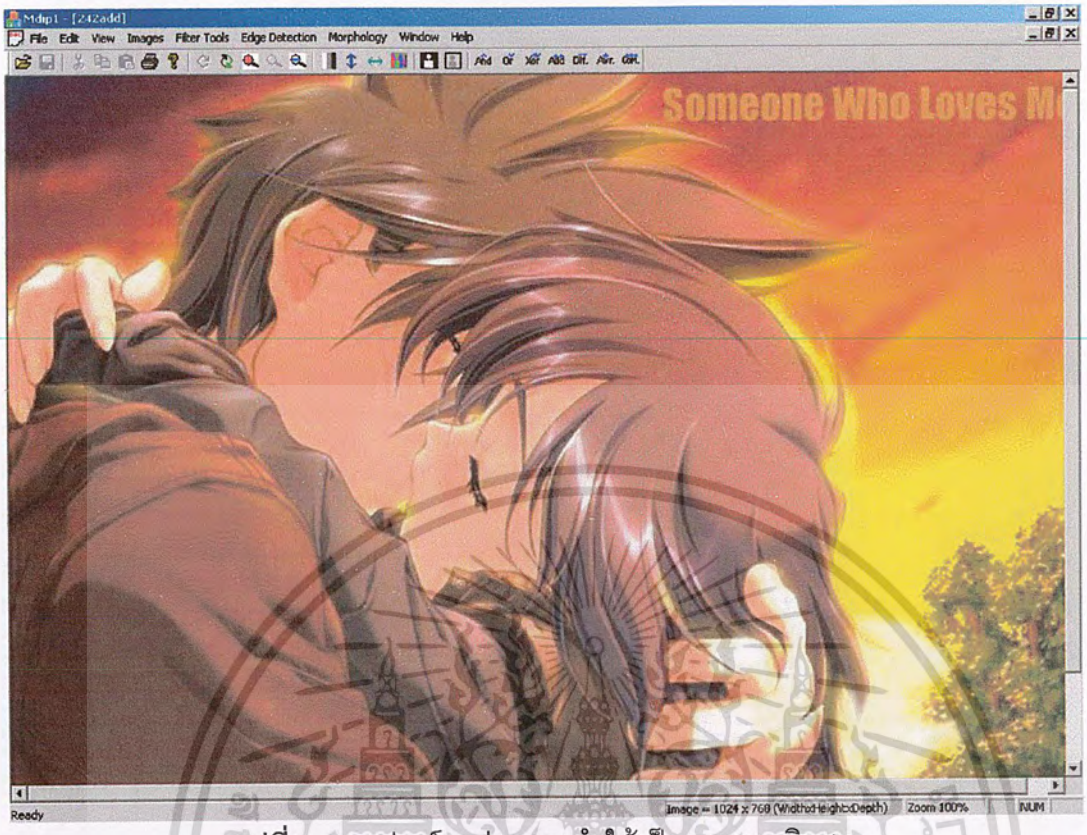
รูปที่ ก-32 รูปร้านก่อนการทำให้นูน



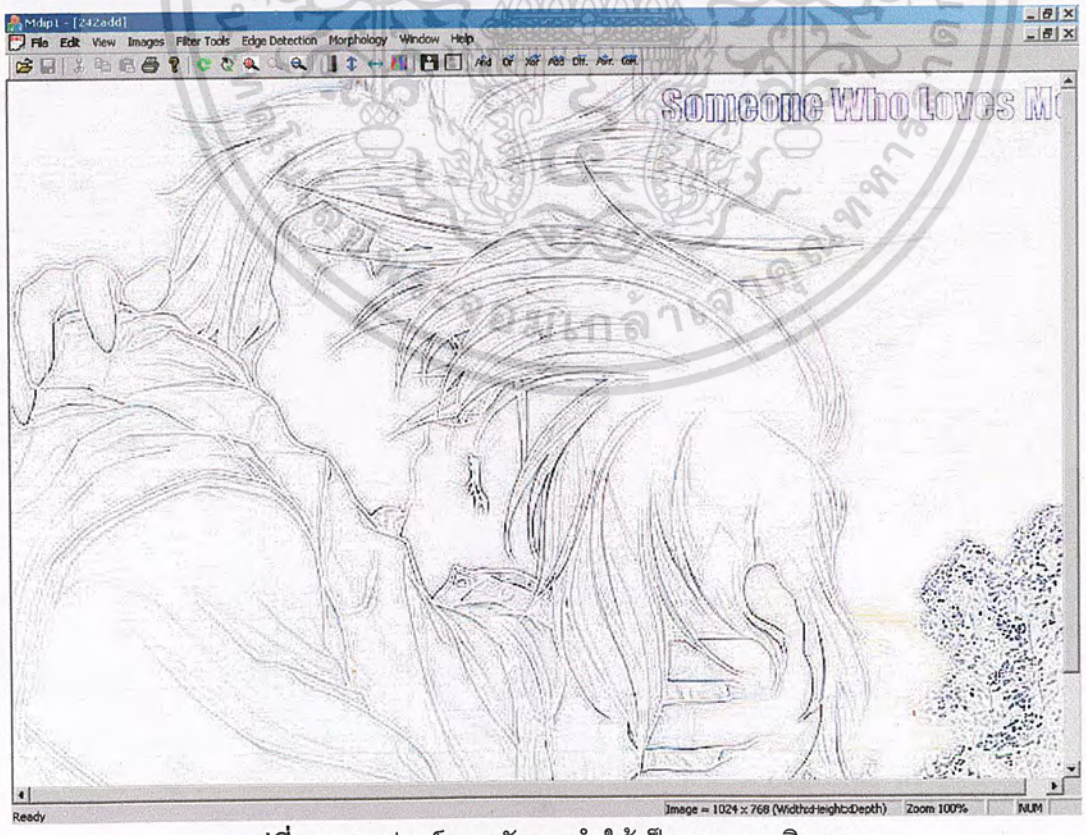
รูปที่ ก-33 รูปร้านหลังการทำให้นูน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำให้เป็นภาพวาดดินสอ



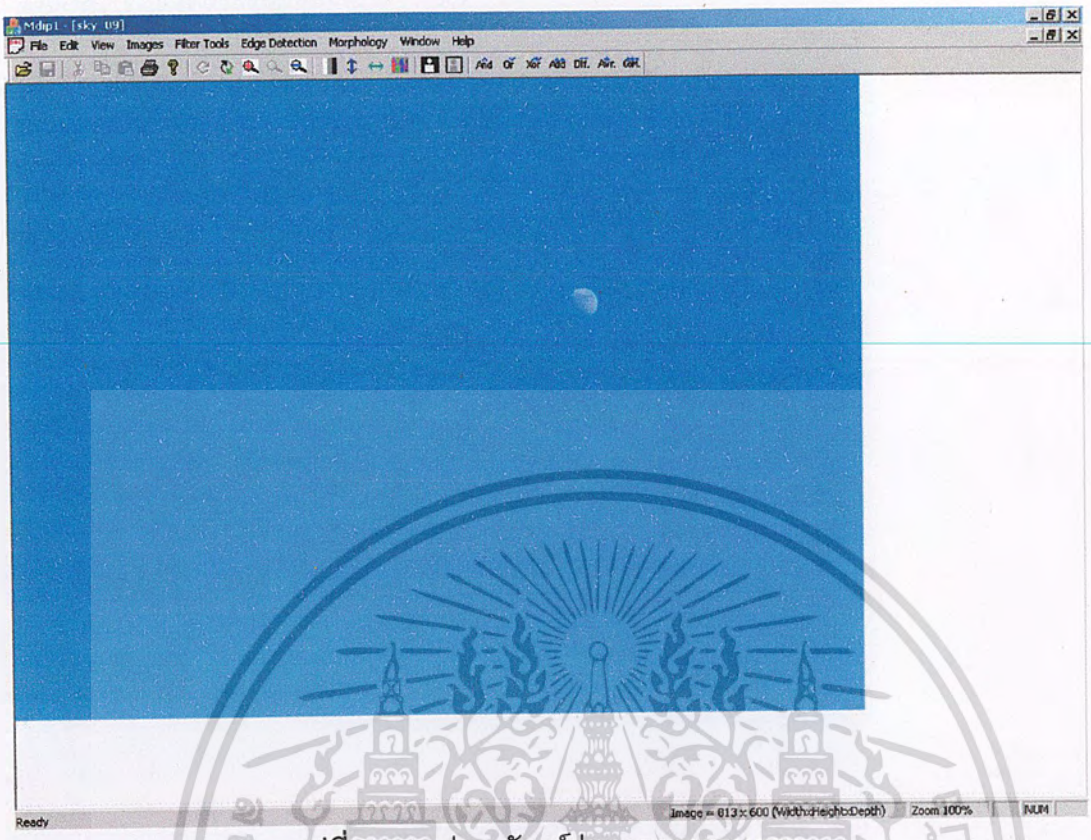
รูปที่ ก-34 รูปการ์ตูนก่อนการทำให้เป็นภาพวาดดินสอ



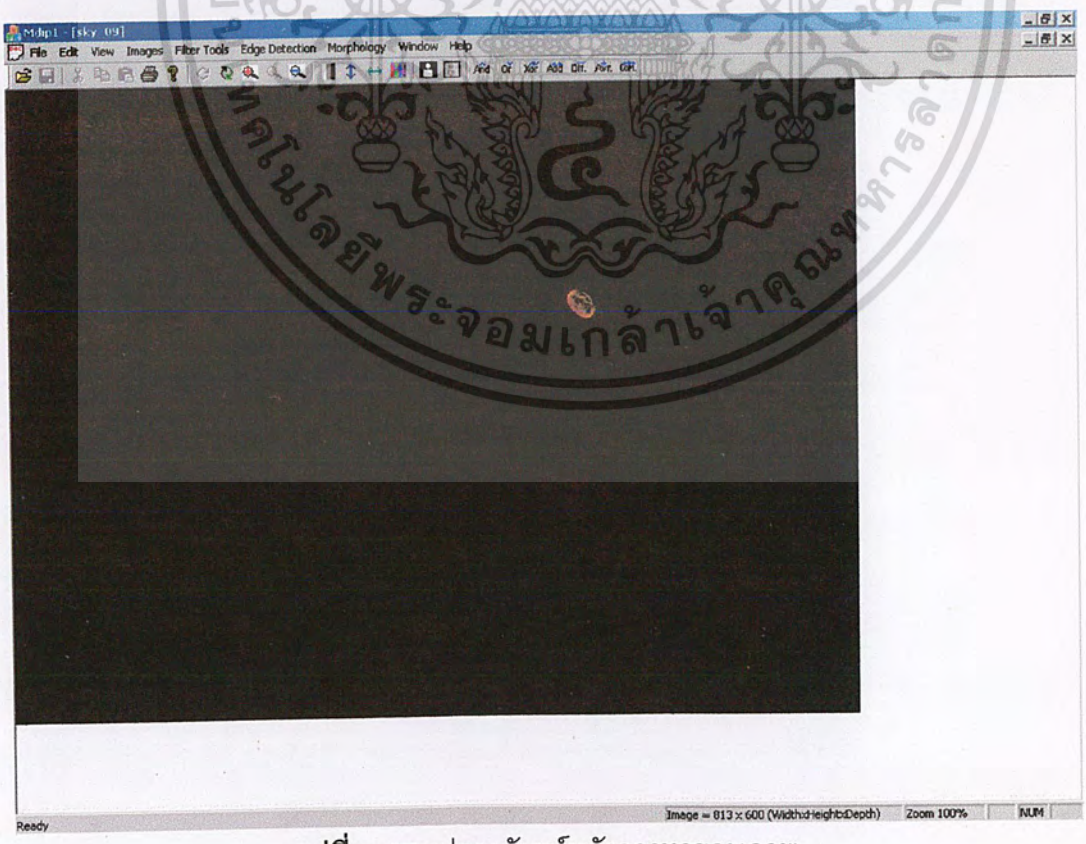
รูปที่ ก-35 รูปการ์ตูนหลังการทำให้เป็นภาพวาดดินสอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบภาพด้วยวิธีSobel



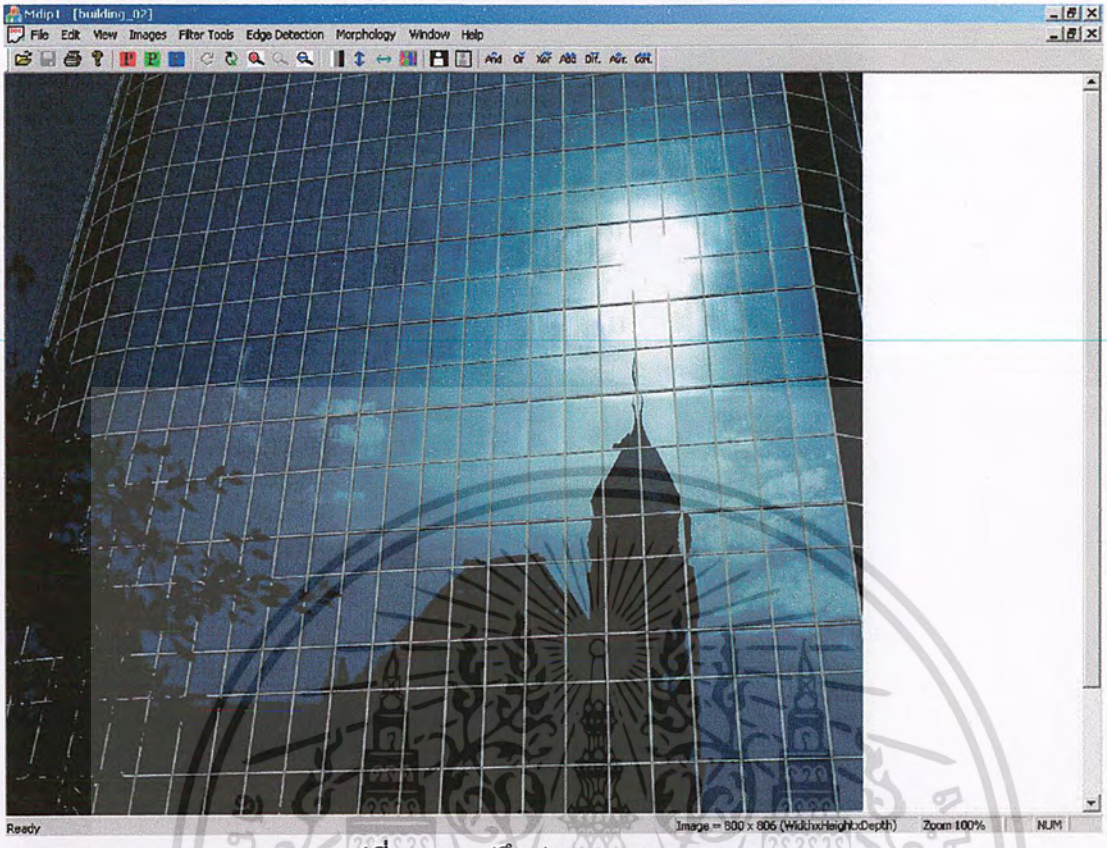
รูปที่ ก-36 รูปดวงจันทร์ก่อนการหาขอบภาพ



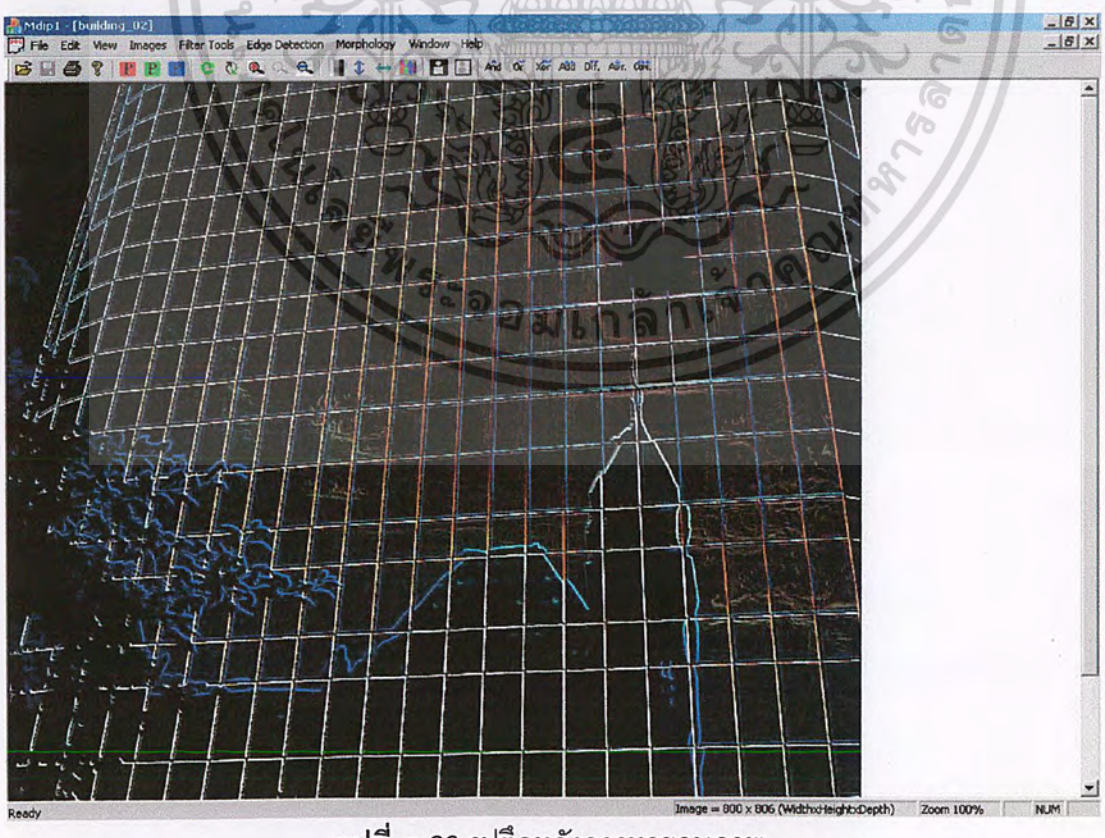
รูปที่ ก-37 รูปดวงจันทร์หลังการหาขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบภาพด้วยวิธีPrewitt



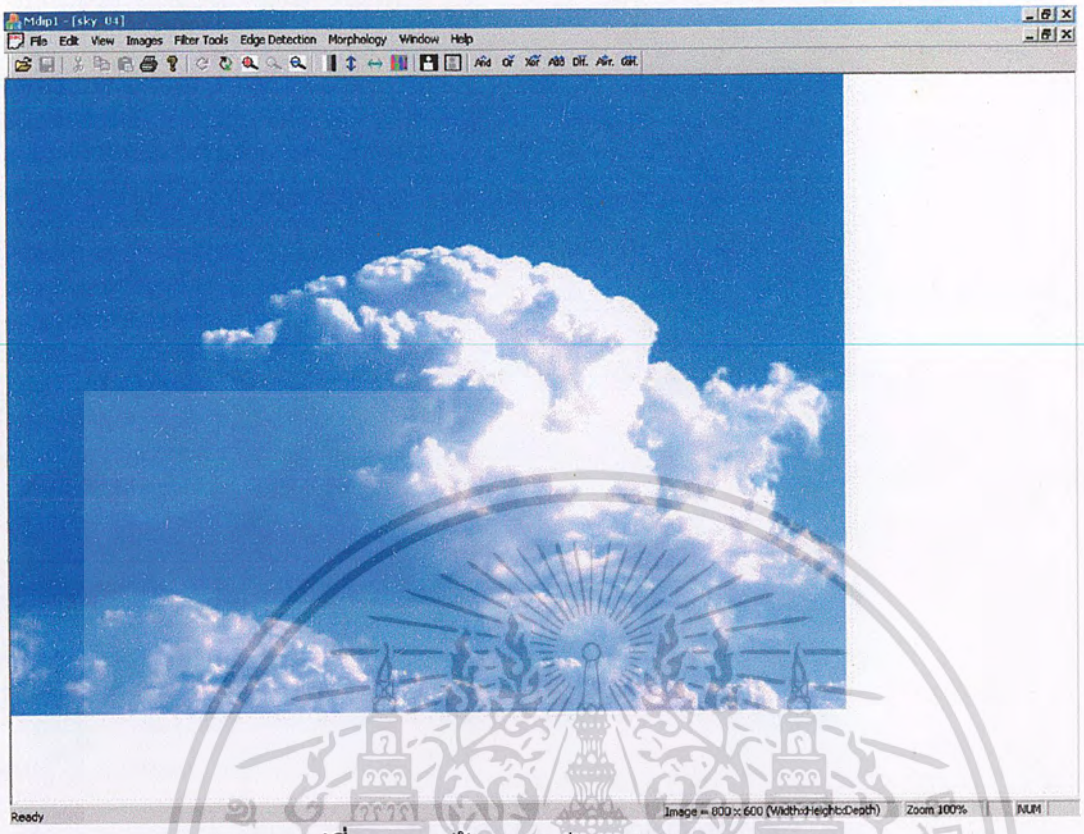
รูปที่ ก-38 รูปตึกก่อนการหาขอบภาพ



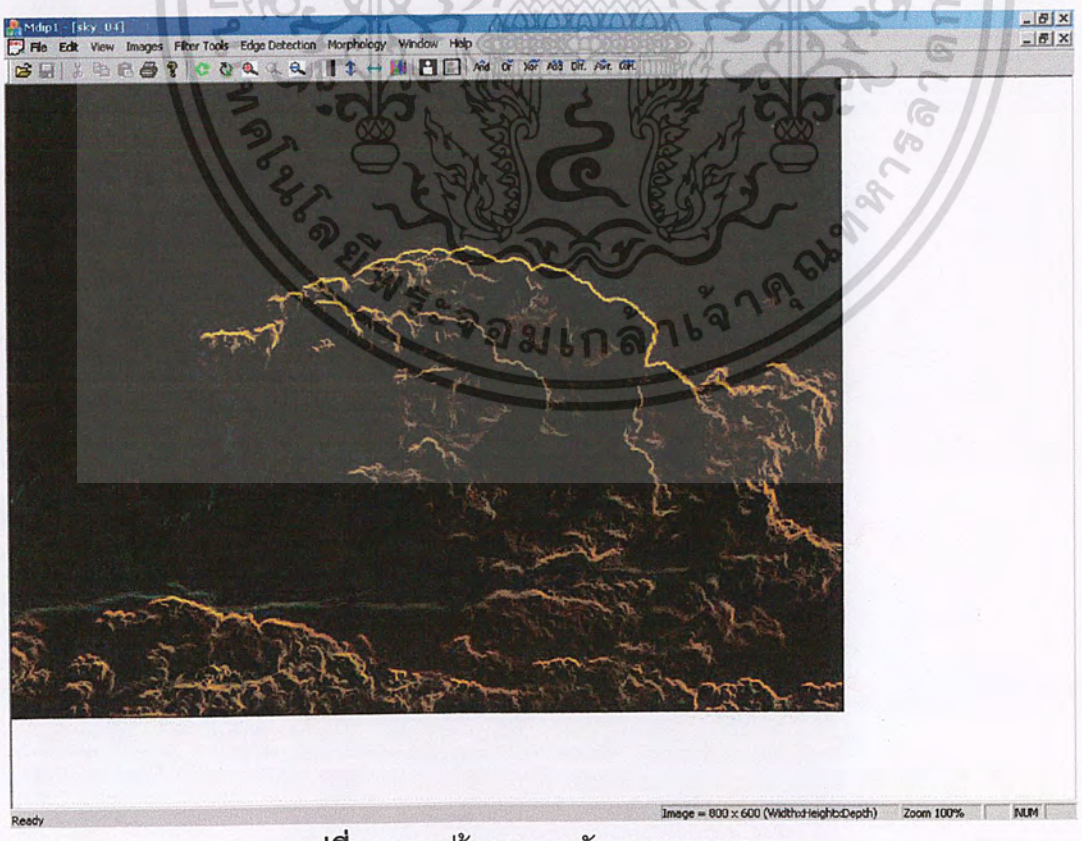
รูปที่ ก-39 รูปตึกหลังการหาขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาขอบภาพด้วยวิธีFrei-Chen



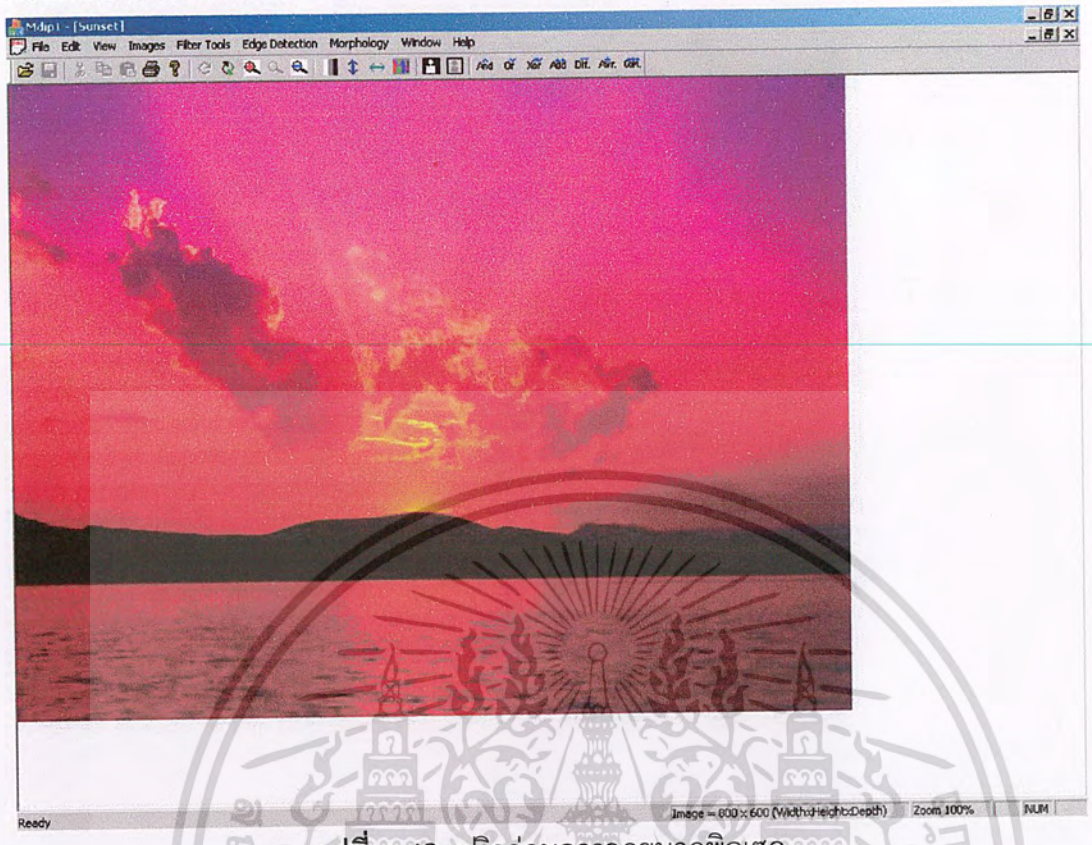
รูปที่ ก-40 รูปก้อนเมฆก่อนการหาขอบภาพ



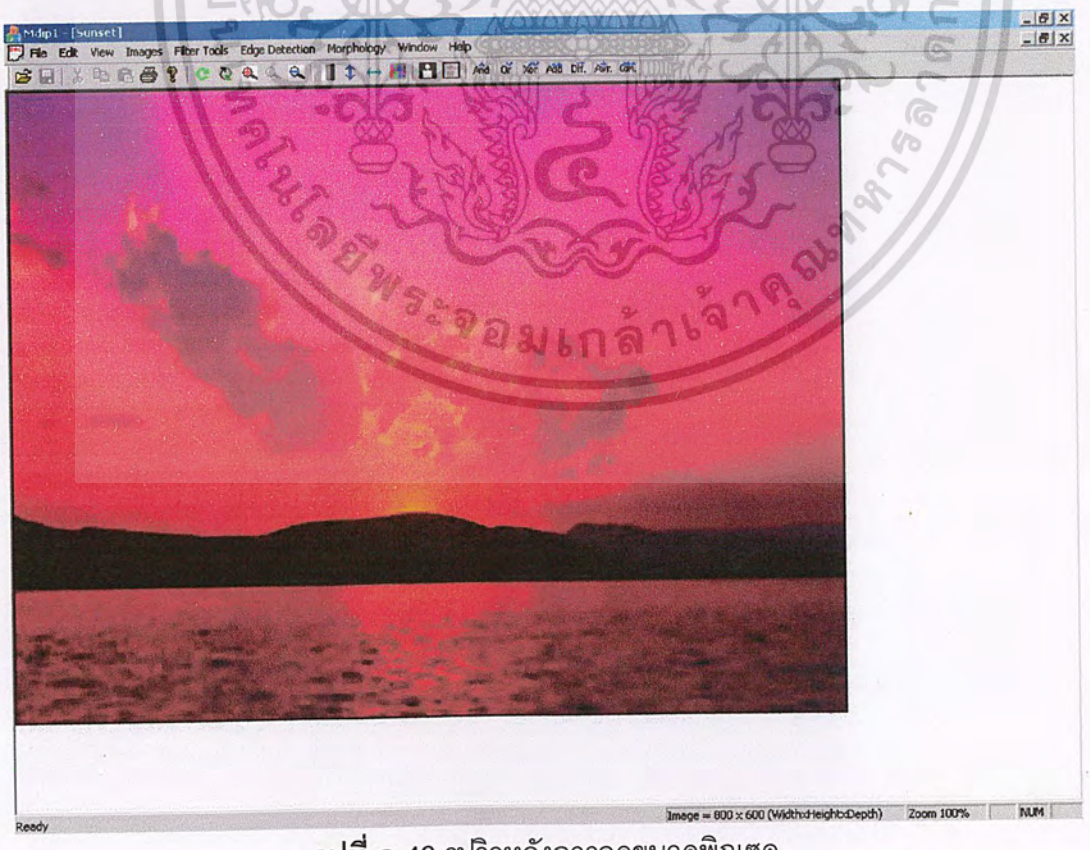
รูปที่ ก-41 รูปก้อนเมฆหลังการหาขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลดขนาดของพิกเซล



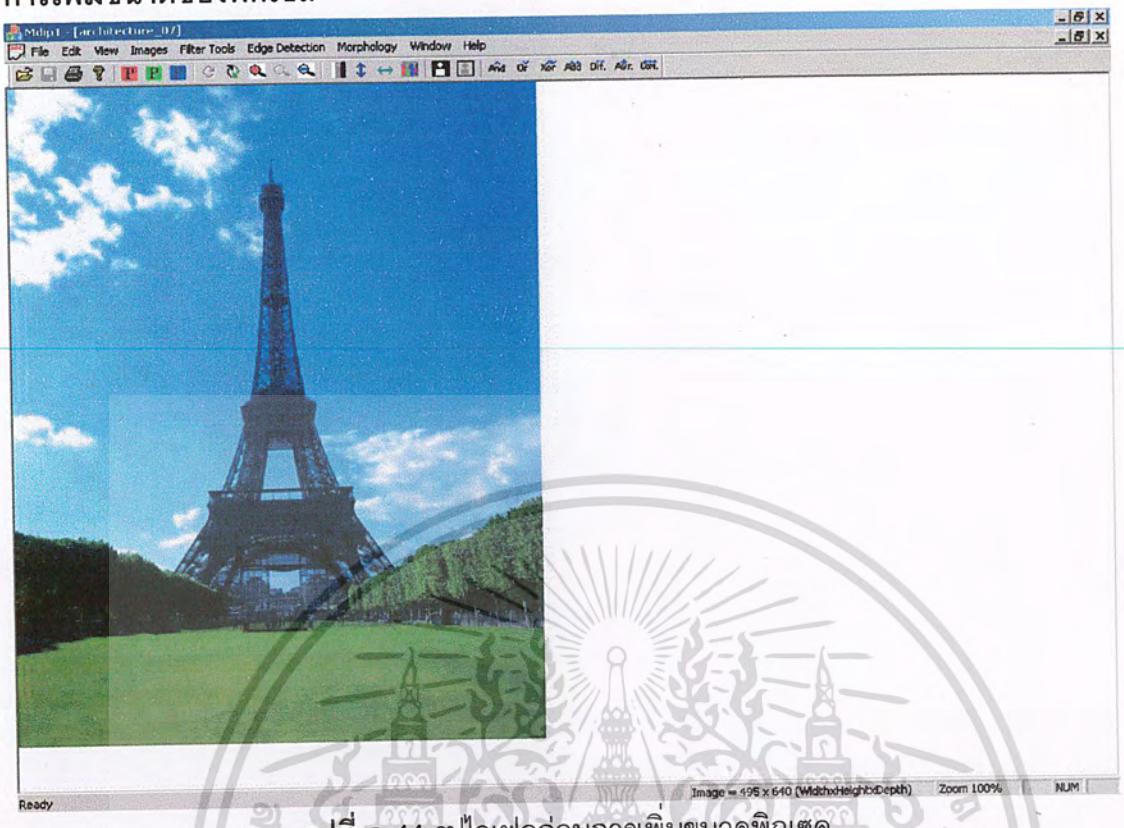
รูปที่ ก-42 รูปวิวก่อนการลดขนาดพิกเซล



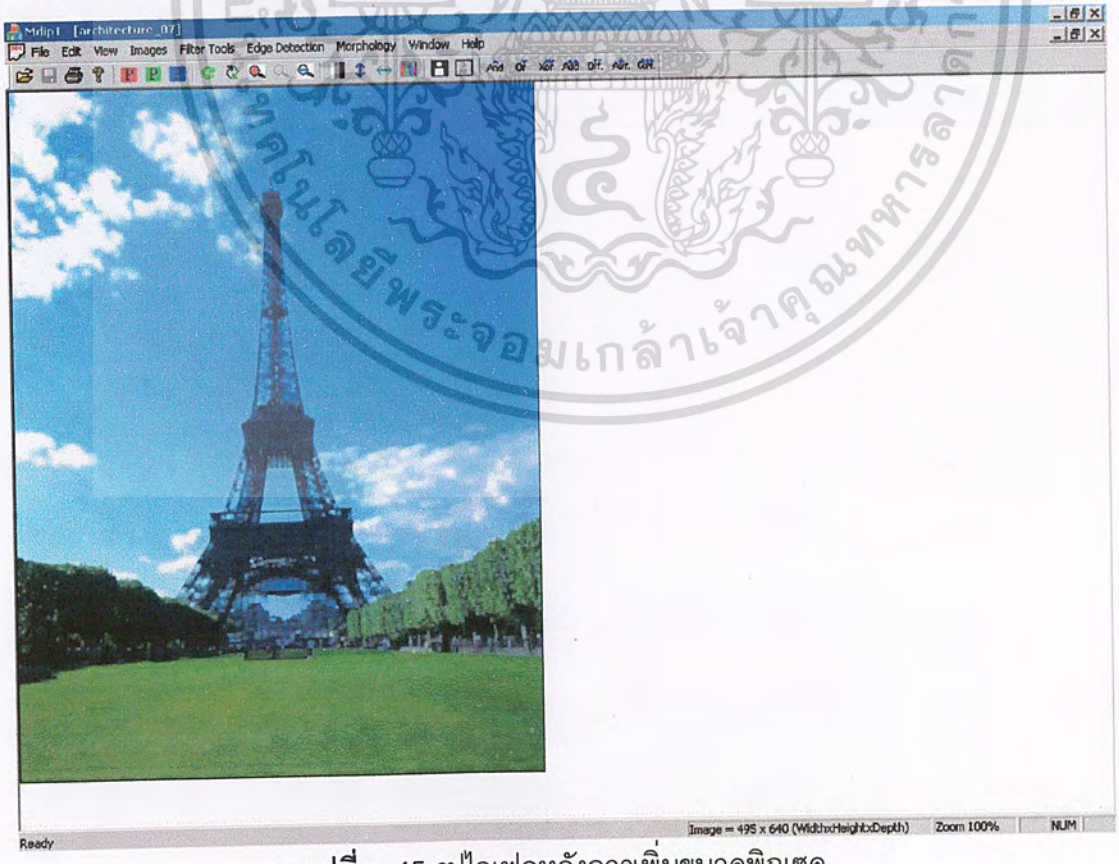
รูปที่ ก-43 รูปวิวหลังการลดขนาดพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มขนาดของพิกเซล



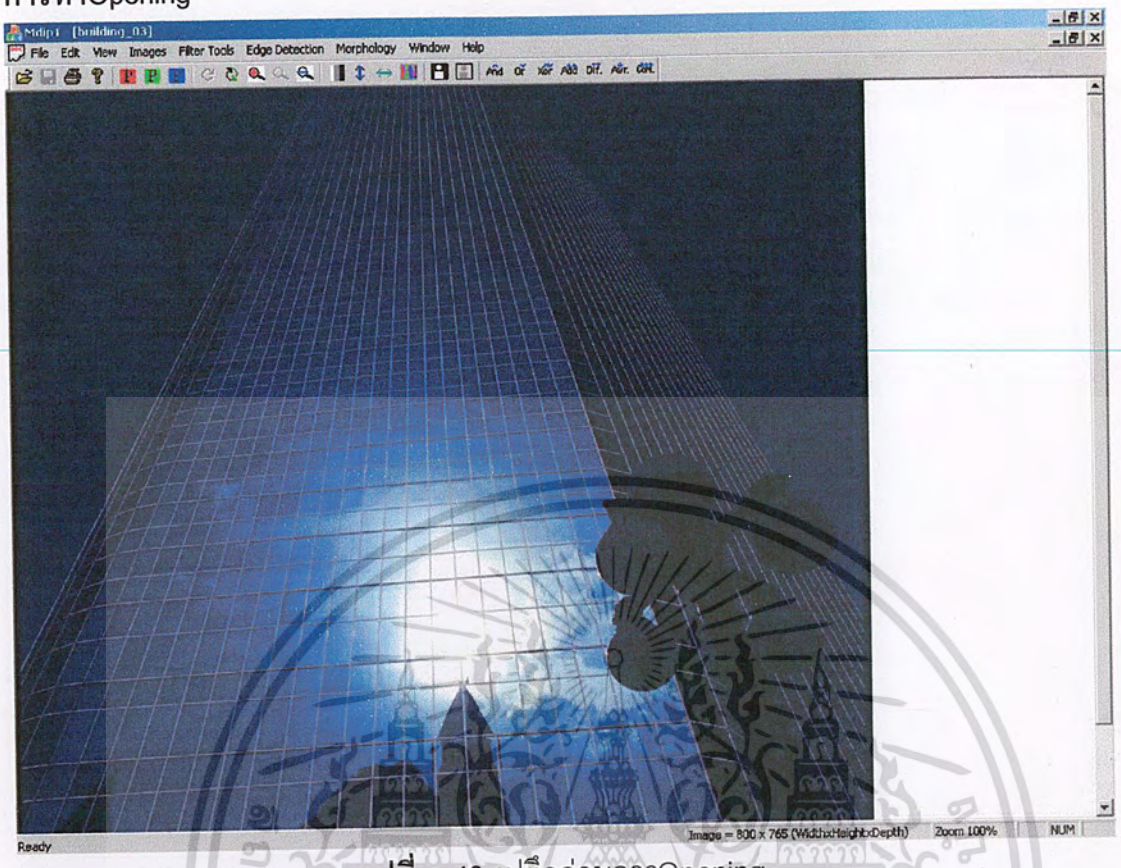
รูปที่ ก-44 รูปไอเฟลก่อนการเพิ่มขนาดพิกเซล



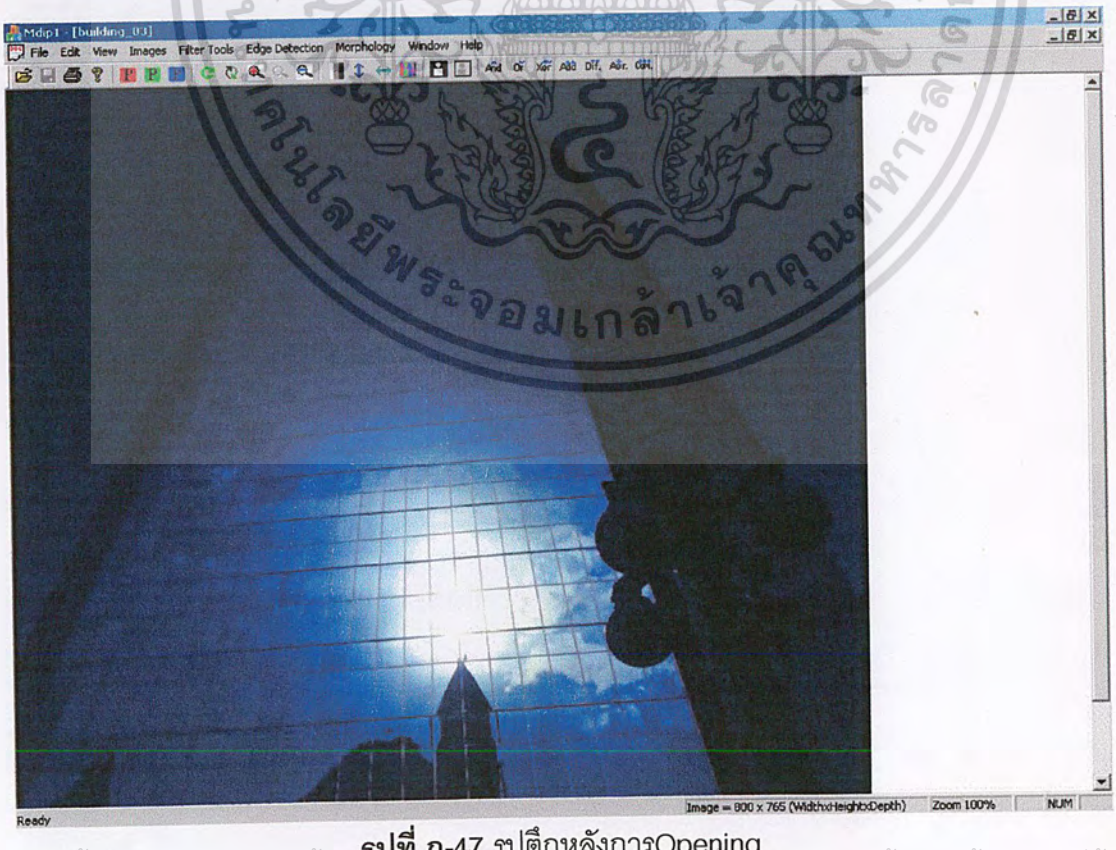
รูปที่ ก-45 รูปไอเฟลหลังการเพิ่มขนาดพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำOpening



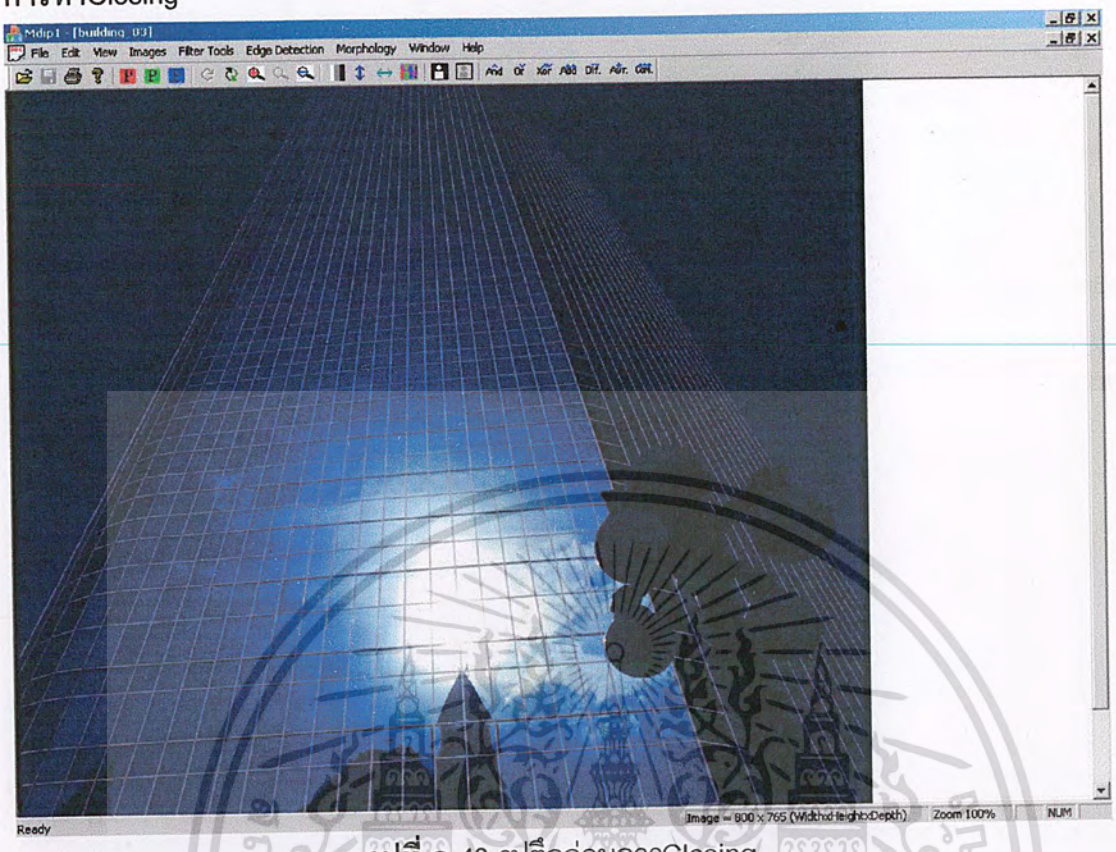
รูปที่ ก-46 รูปตึกก่อนการOpening



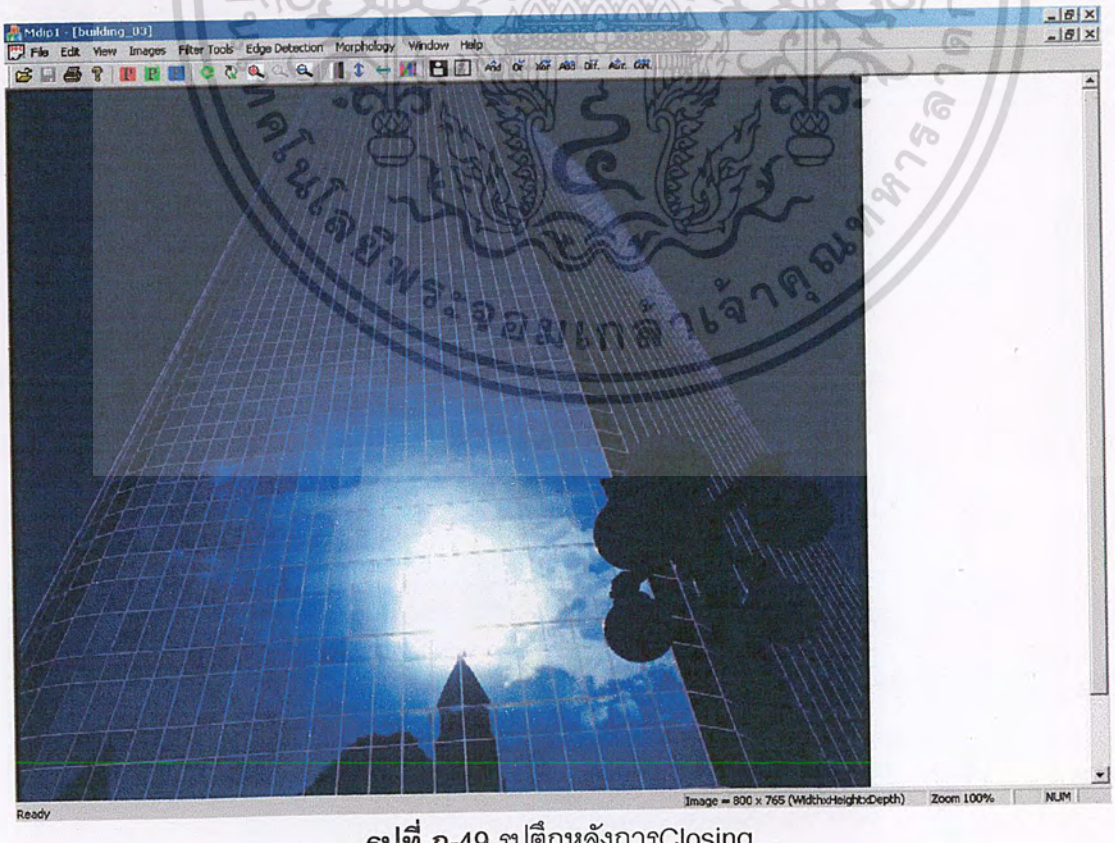
รูปที่ ก-47 รูปตึกหลังการOpening

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำClosing



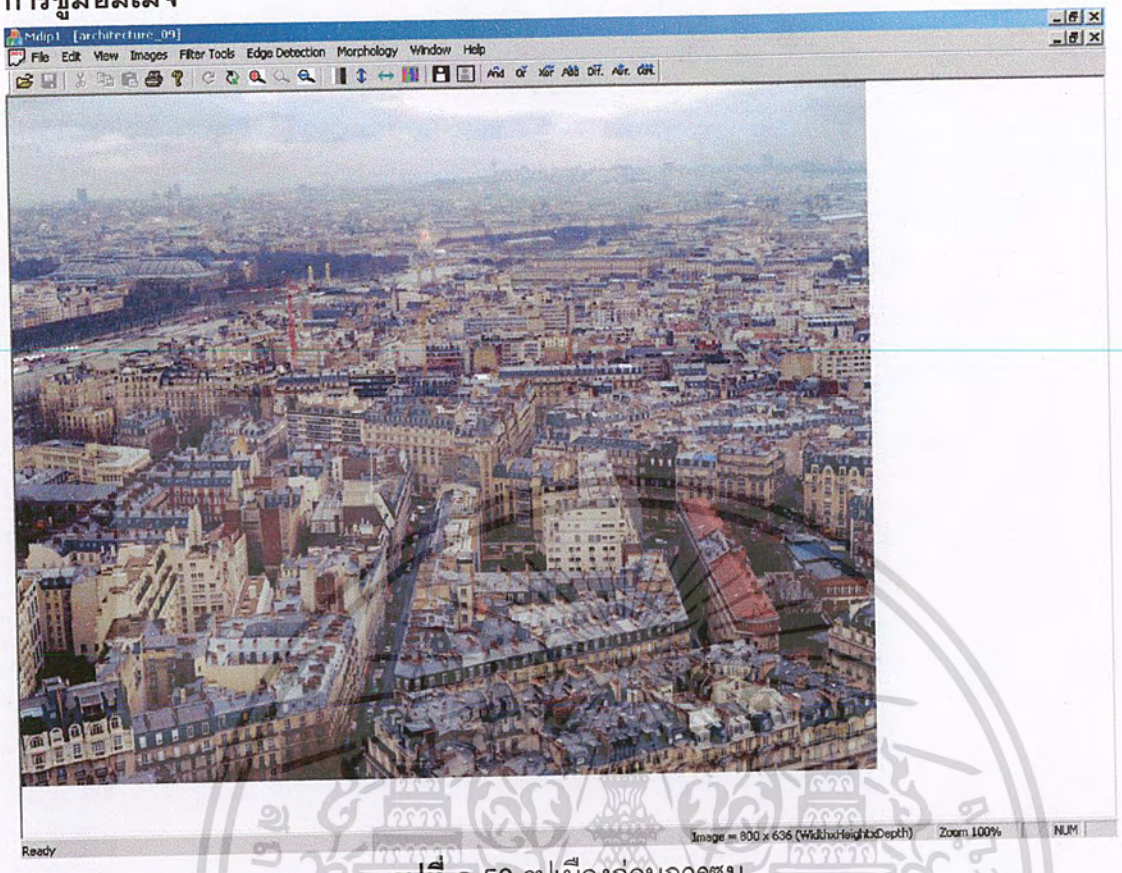
รูปที่ ก-48 รูปตึกก่อนการClosing



รูปที่ ก-49 รูปตึกหลังการClosing

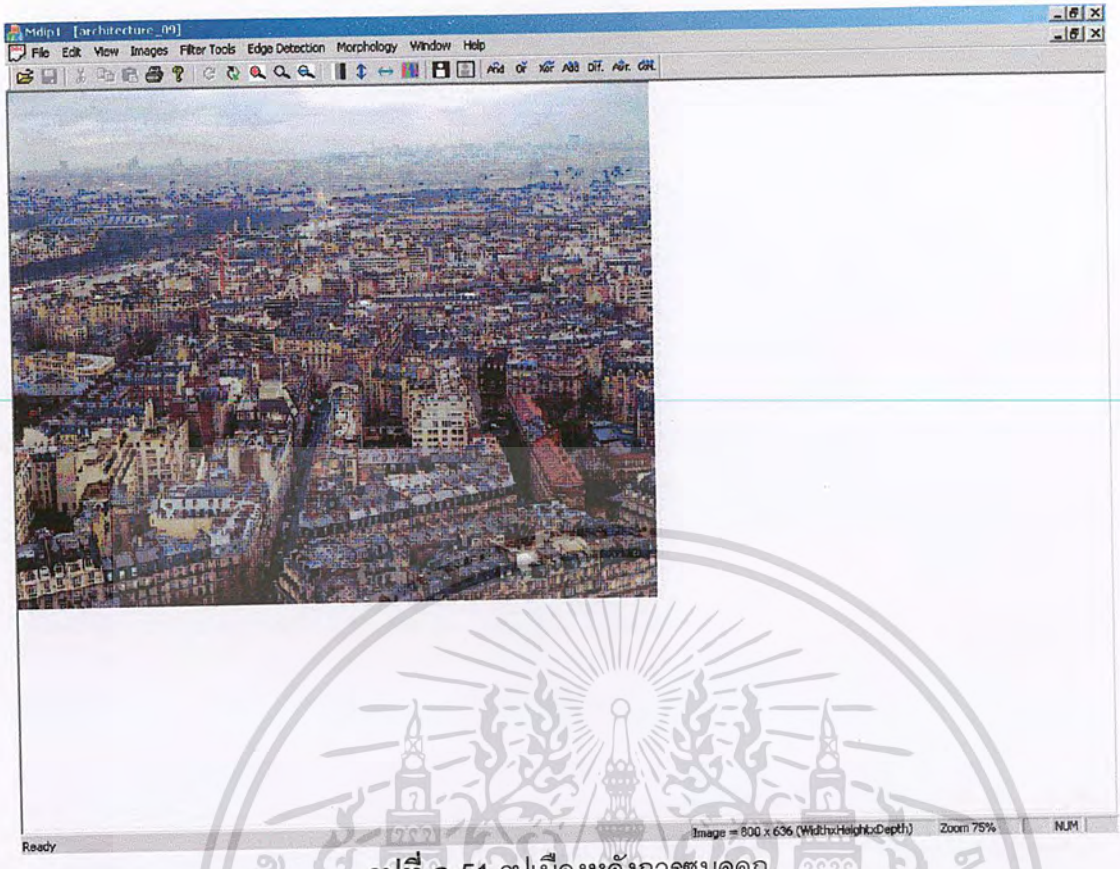
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การซูมอิมเมจ



รูปที่ ก-50 รูปเมืองก่อนการซูม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-51 รูปเมืองหลังการซูมออก

รูปที่ ก-52 รูปเมืองหลังการซูมเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ยุทธนา ลีลาศวัฒนกุล . 2544 . **คู่มือการเขียนโปรแกรมด้วย** Visual C++ 6.0 **ฉบับโปรแกรมเมอร์** . กรุงเทพฯ : อินโฟเพลส.

ยุทธนา ลีลาศวัฒนกุล . 2544 . **คู่มือการเขียนโปรแกรมวินโดวส์ขั้นสูงด้วย** Visual C++ . กรุงเทพฯ : ดวงกมลสมัย.

นิรุธ อำนวยศิลป์ . 2544 . **คู่มือการเขียนโปรแกรม** Microsoft Visual C++ Version 6.0 . กรุงเทพฯ : ซัคเซส มีเดีย.

พงษ์ศักดิ์ เกี้ยวมาศ. "โปรแกรมปรับแต่งและต่อเติมภาพ." วิทยานิพนธ์วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2542.

Don Pearson. 1991 . Image Processing . North Yorkshire : McGraw – Hill.

Awcock, G.W. and Thomas, R. 1996. Applied Image Procesing. USA : McGraw – Hill.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้