สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การตกแต่งภาพด้วยคอมพิวเตอร์

นายจรูญ โมราวิจิตร
นายไพศาล กิตติปรีชากุล
นายวิษณุ สุทธานุรักษ์

ส.ม.
จ174ก
2538

เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

612558990

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Image Processing

Mr. Jaroon Morawichit

Mr. Phaisarn Kittipreechakun

Mr. Wisanu Suthanurak

A Special Project Submitted in Partial Fulfillment of the

Requirement for the Degree of Bachelor of Science

Department of Mathematics and Computer Sciences

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

1995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การตกแต่งภาพด้วยคอมพิวเตอร์
	Image Processing
โดย	นายจรูญ โมราวิจิตร
	นายไพศาล กิตติปรีชากุล
	นายวิษณุ สุทธานุรักษ์
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ไพรบูลย์ พันธรักษ์พงษ์

ภาควิชา คณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นับปัญหาพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต

(รศ. ภัคคินี ชิตสกุล) หัวหน้าภาควิชาฯ

คณะกรรมการปัญหาพิเศษ

(อ. กฤษฎา ไตรสุรัตน์) ประธานกรรมการ

(รศ. วิเชียร ศรีเสือขาม) กรรมการ

(อ.ไพรบูลย์ พันธรักษ์พงษ์) กรรมการ

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การตกแต่งภาพด้วยคอมพิวเตอร์	
นักศึกษา	นายจรูญ	โมราวิจิตร
	นายไพศาล	กิตติปรีชากุล
	นายวิษณุ	สุทธานุรักษ์
อาจารย์ที่ปรึกษา	อาจารย์ไพรบูลย์ พันธรักษ์พงษ์	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
ปีการศึกษา	2538	

บทคัดย่อ

การทำปัญหาพิเศษนี้มีจุดมุ่งหมายเพื่อประยุกต์ใช้ทฤษฎีในวิชาคอมพิวเตอร์กราฟิค , ศึกษาปัญหาและทำการค้นคว้าเพิ่มเติมถึงเรื่องเทคนิคที่ใช้ในกระบวนการแปลงภาพ มาทำการจัดสร้างในรูปแบบของซอฟแวร์ รวมทั้งศึกษาการทำงานและการเขียนโปรแกรมภายใต้ระบบปฏิบัติการ WINDOWS 95 ซึ่งหลังจากได้ออกแบบระบบและโปรแกรมลงบนไมโครคอมพิวเตอร์ระดับ 32 บิตแล้ว พบว่าสามารถใช้งานได้ดีเป็นที่น่าพอใจในระดับหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title Image Processing

Name Mr.Charoon Morawichit
Mr.Phaisarn Kittipreechakun
Mr.Wisanu Suthanurak

Special Project Advisor Mr.Phaiboon Pantarakpong

Department Mathematics and Computer Sciences

Academic Year 1995

Abstract

This special project aims at image processing software implementation which is used for applying in computer graphics theorm. In addition, we have studied the problem and researched about technique of image processing to create a software. After we designed and programmed on microcomputer 32 bits system. As the result, it can be used and we satisfied.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

คณะผู้จัดทำปัญหาพิเศษฉบับนี้ ขอขอบพระคุณอาจารย์ทุกท่านที่ช่วยให้ปัญหาพิเศษนี้สำเร็จลุล่วงด้วยดี โดยเฉพาะอย่างยิ่งอาจารย์ไพบูลย์ พันธรักษ์พงษ์ อาจารย์ที่ปรึกษาปัญหาพิเศษ ที่ได้กรุณาช่วยให้คำแนะนำในการออกแบบ และอาจารย์ศรัณย์ อินทโกสุม ที่ได้กรุณาเอื้อเฟื้อหนังสือเพื่อใช้ประกอบในการทำปัญหาพิเศษ

ขอขอบพระคุณคณะกรรมการทุกท่านที่ได้กรุณาสละเวลาอันมีค่าของท่าน เพื่อตรวจสอบปัญหาพิเศษนี้ จนกระทั่งสำเร็จลงด้วยดี

ท้ายสุดนี้ขอกราบขอบพระคุณบิดา มารดา ครู อาจารย์ และผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 ตัวอย่างฮีสโตแกรม	3
รูปที่ 2.2 ตารางเปรียบเทียบสี	4
รูปที่ 2.3 ตัวอย่างการปรับความสว่าง	6
รูปที่ 2.4 ตัวอย่างการกลับสีภาพ	8
รูปที่ 2.5 ตัวอย่างเทรสโฮล	9
รูปที่ 2.6 ชนิดของคอนทราส	11
รูปที่ 2.7 ตัวอย่างการปรับความคมชัด	12
รูปที่ 2.8 คอนโวลูชัน	14
รูปที่ 2.9 ตัวอย่างคอนโวลูชันเคอร์เนล	15
รูปที่ 2.10 ตัวอย่างการประมวลผลแบบบริเวณ	18
รูปที่ 2.10 (ต่อ) ตัวอย่างการประมวลผลแบบบริเวณ	19
รูปที่ 2.11 ตัวอย่างการหาขอบด้วยวิธีโซเบล	24
รูปที่ 2.12 ตัวอย่างการใช้ค่ากลาง	25
รูปที่ 2.13 การใช้ค่ากลาง	26
รูปที่ 1 แสดงหน้าจอเมื่อเข้าสู่ระบบ	41
รูปที่ 2 แสดงหน้าจอเมื่อทำการเปิดไฟล์	42
รูปที่ 3 แสดงหน้าจอเมื่อทำการประมวลผล	43
รูปที่ 4 แสดงหน้าจอฮีสโตแกรม	43
รูปที่ 5 แสดงหน้าจอรายละเอียดข้อมูลรูปภาพ	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อปัญหาพิเศษภาษาไทย	ก
บทคัดย่อปัญหาพิเศษภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญรูป	ง
บทที่ 1 บทนำ	1
ความสำคัญและที่มาของปัญหา	1
วัตถุประสงค์ของปัญหาพิเศษ	1
ขอบเขตของปัญหาพิเศษ	1
ขั้นตอนในการดำเนินงาน	1
ประโยชน์ที่ได้รับ	2
บทที่ 2 ทฤษฎีและหลักเกณฑ์ในการตกแต่งภาพ	3
การประมวลผลภาพแบบจุด (Point Process)	5
การประมวลผลภาพแบบบริเวณ (Area Process)	13
การหมุนและการกลับภาพ (Fliping and Rotating)	27
บทที่ 3 การออกแบบและสร้างระบบ	28
การออกแบบระบบ	28
ผังการทำงานของระบบ	28
เครื่องมือที่ใช้ในการสร้างระบบ	28
การสร้างระบบ	29
บทที่ 4 ผลการทำงานและข้อจำกัดของระบบ	38
ข้อจำกัดการใช้งานของระบบ	38
ผลการทำงานของระบบ	38
บทที่ 5 สรุปปัญหาและข้อเสนอแนะ	39
สรุปผลการทำปัญหาพิเศษ	39
ปัญหาและข้อเสนอแนะ	39
ภาคผนวก	
คู่มือการใช้งานระบบ	
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ความสำคัญและที่มาของปัญหาพิเศษ

ปัจจุบันมีการใช้คอมพิวเตอร์มาช่วยในการจัดเก็บรูปภาพกันอย่างแพร่หลาย เนื่องจากสามารถเก็บภาพได้จำนวนมหาศาล และรูปที่เก็บก็ไม่มีการเปลี่ยนแปลงไปจากเดิมได้ ในบางครั้งรูปที่ ต้องการจัดเก็บอาจเสียรายละเอียดด้วยสาเหตุต่างๆ เช่น ภาพอาจมีโทนสีที่เข้มมากหรืออ่อนมากเกินไป มืดมากหรือสว่างมากเกินไป เป็นต้น จึงต้องมีการปรับแต่งก่อนที่จะทำการจัดเก็บ หรือในบางครั้งผู้ใช้อาจต้องการปรับแต่งรูปภาพด้วยเทคนิคพิเศษก่อนจัดเก็บ เช่น การทำภาพเบลอ , การปรับความคมชัดของภาพ , การหาขอบรูป , การทำรูปเนกาทีฟ , การหมุนภาพ และ การกลับภาพ เป็นต้น ซึ่งศาสตร์แขนงนี้เป็นที่รู้จักกันว่า การจัดการรูปภาพ (Image Processing) ที่ต้องอาศัยทฤษฎีทางคณิตศาสตร์มาใช้อย่างมาก และต้องอาศัยคอมพิวเตอร์มาเป็นส่วนช่วยในการคำนวณและโปรแกรมระบบ

วัตถุประสงค์ของปัญหาพิเศษ

เพื่อประยุกต์ใช้ทฤษฎีในวิชาคอมพิวเตอร์กราฟิค รวมทั้งศึกษาค้นคว้าเพิ่มเติมถึงทฤษฎีต่างๆ ของ การจัดการรูปภาพ (Image Processing) มาทำการจัดสร้างโปรแกรมประยุกต์ในรูปแบบของซอฟต์แวร์

ขอบเขตของปัญหาพิเศษ

ในปัญหาพิเศษนี้ การปรับแต่งรูปภาพจะกระทำโดยรวมทั้งรูป มิได้เจาะจงเฉพาะบริเวณใดบริเวณหนึ่งในรูปภาพ

ขั้นตอนในการดำเนินงาน

1. ศึกษาโครงสร้างและการจัดเก็บข้อมูลภาพของไฟล์กราฟิค
2. ศึกษาทฤษฎีต่างๆ ของคอมพิวเตอร์กราฟิค
3. ศึกษาทฤษฎีต่างๆ ของ การจัดการรูปภาพ (Image Processing)
4. ออกแบบระบบของโปรแกรมประยุกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. โปรแกรมระบบ

6. ทดสอบการทำงานและปรับปรุงแก้ไข

7. จัดทำรายงานและคู่มือประกอบสำหรับโปรแกรมประยุกต์

ประโยชน์ที่ได้รับ

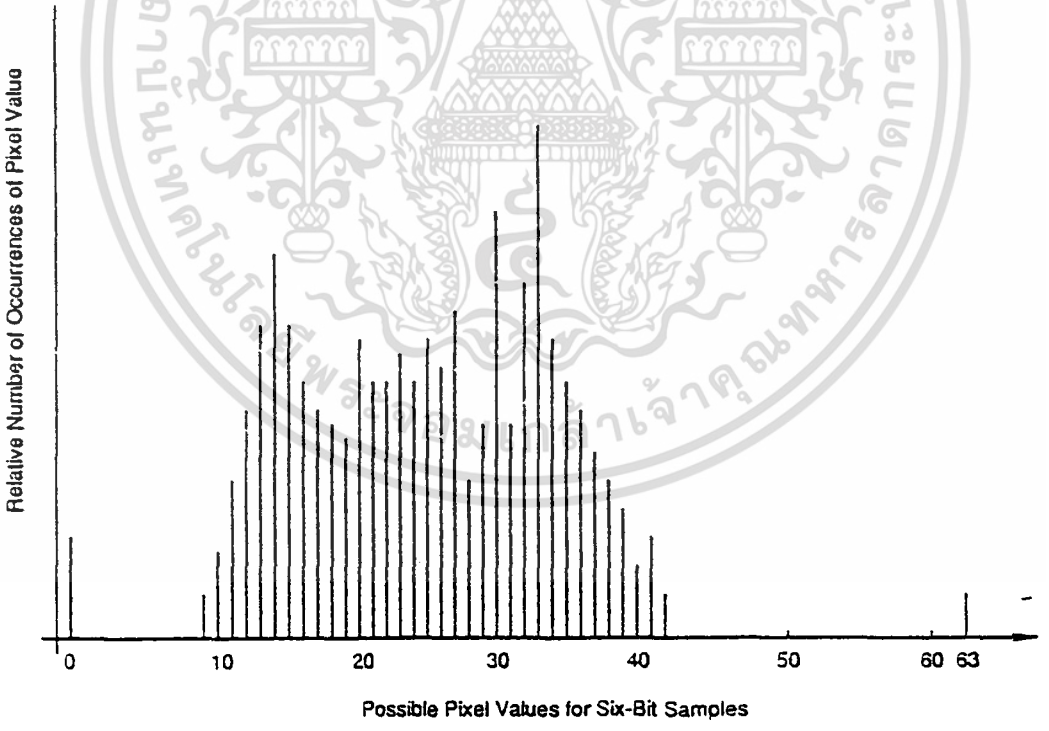
โปรแกรมที่ได้นั้นสามารถนำไปใช้ในการแก้ไข - ตกแต่ง ได้ในระดับหนึ่งและสามารถทำการจัดเก็บให้อยู่ในรูปแบบ (format) ที่ผู้ใช้ต้องการได้ อีกทั้งยังเป็นต้นแบบเพื่อใช้ในการศึกษาการทำงานของเทคนิคการประมวลผลภาพและเพื่อการพัฒนาให้ใช้งานได้ดียิ่งขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักเกณฑ์ในการตกแต่งภาพ

ฮีสโตแกรม (Histograms) เป็นกราฟ 2 มิติ ที่แสดงความสัมพันธ์ระหว่างจำนวนที่ปรากฏของแต่ละพิกเซลกับระดับความสว่างของพิกเซลส่วนในกรณีที่เป็นรูปสีก็จะแสดงถึงความสัมพันธ์ระหว่างจำนวนของพิกเซลกับค่าความสว่างของแม่สีแต่ละสี (แดง,เขียว,น้ำเงิน)

รูป 2.1 แสดงถึงการกราฟฮีสโตแกรม



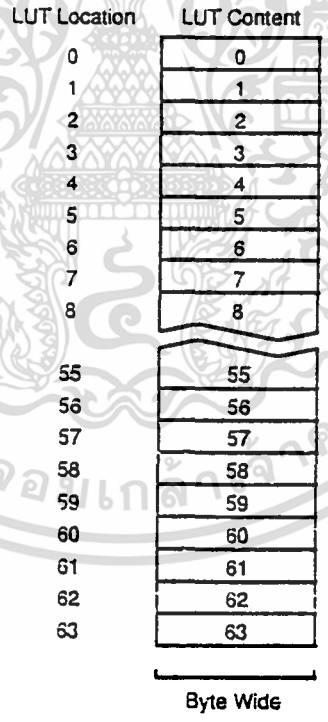
รูปที่ 2.1 ตัวอย่างฮีสโตแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางเปรียบเทียบสี (Look-Up Tables) ทำให้การทำการประมวลผลภาพแบบจุดเป็นไปได้ง่ายขึ้น อัลกอริทึม LUTs ทำงานโดยเมื่อต้องการเปลี่ยนความสว่างของพิกเซลๆหนึ่งก็ให้นำค่าในพิกเซลนั้นๆมาใช้เป็นตัวชี้ (index) ใน ตาราง LUTs แล้วนำค่าในตาราง (array) ที่ถูกเก็บอยู่ในตำแหน่งตัวชี้นั้นแทนค่านั้นกลับไปที่พิกเซลเดิม เราสามารถประกาศ LUTs ได้ดังนี้

```
BYTE  LookUpTable [ Maxquantlevels ];
```

ก่อนจะใช้ LUTs เราจะต้องทำการตั้งค่าเริ่มต้นในตอนแรกให้กับ LUTs ก่อนที่จะถูกใช้โดย มิฉะนั้นค่าที่ได้จาก LUTs จะไม่สัมพันธ์กับค่าที่ส่งเข้าไปใน LUTs การตั้งค่าเริ่มต้นนั้นหาได้จากการประมวลผลภาพแบบจุด ส่วนในกรณีอย่างง่ายนั้นทำโดยการเก็บค่า ตัวชี้นั้นเอาไว้ในตำแหน่งของมันเองใน LUTs นั่นคือ ตำแหน่ง 0 จะเก็บ 0, ตำแหน่ง 1 จะเก็บ 1 ไปเรื่อยๆ จนถึงตำแหน่ง 255 จะเก็บ 255 ดังรูป 2.2



รูปที่ 2.2 ตารางเปรียบเทียบสี

จะพบว่าข้อมูลที่ได้จาก LUT จะมีค่าเดียวกับ ข้อมูลที่เข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคนิคการแต่งภาพในปัญหาพิเศษนี้เรามองเป็นหัวข้อหลักๆ ได้ 2 หัวข้อดังนี้

1. การตกแต่งภาพแบบจุด (Point process)
2. การตกแต่งภาพแบบบริเวณ (Area process)

การประมวลผลภาพแบบจุด (Point process)

การประมวลผลภาพแบบจุด เป็นอัลกอริทึม (algorithm) ที่ใช้ในการเปลี่ยนแปลงค่าภายในพิกเซล (pixel) ในภาพ ในการประมวลผลนั้นพิกเซลแต่ละพิกเซลจะถูกแทนที่โดยค่าใหม่ซึ่งใช้ได้มาจากอัลกอริทึมในการแปลงโดยมีความสัมพันธ์กับค่าเดิมของพิกเซลนั้นๆ การประมวลผลภาพแบบจุดนั้นจะไม่เปลี่ยนแปลงความสัมพันธ์เกี่ยวกับระยะภายในภาพ และ การประมวลผลภาพแบบจุดจะไม่เปลี่ยนแปลงรายละเอียดภายในภาพ

สังเกตว่า อัลกอริทึมการประมวลผลภาพแบบจุด ทำให้ภาพดูดีขึ้นสำหรับ ภาพที่มีความต่อเนื่องของโทนสี (continuous-tone image) ซึ่งค่าความเข้มของสีถูกแสดงโดยค่าใน พิกเซลนั้นๆ ในการเปลี่ยนแปลงภาพสีนั้นจะแตกต่างออกไปเล็กน้อยคือแทนที่จะกระทำกับค่าในพิกเซลนั้นโดยตรง เราจำเป็นต้องเปลี่ยนค่าในระนาบสี (Color plane) ของแต่ละสี แทน โดยเราจะเข้าถึงค่าเหล่านั้นโดยใช้ค่าของพิกเซลนั้นเป็น index ที่ชี้ไปยังแผ่นสี (Palette) ของ VGA หรือค่า RGB

การประมวลผลภาพแบบจุด ที่พิจารณามีดังนี้ คือ

1) การปรับความสว่างของภาพ (Image Brightening)
2) การกลับสีของภาพ (Negative Image)
3) เทรสโฮล (Image Thresholding)
4) การปรับความคมชัดของภาพ (Image Contrast Stretching)

1.การปรับความสว่างของภาพ (Image Brightening)

การปรับความสว่างของภาพ (Image Brightening) เป็น การประมวลผลภาพแบบจุดซึ่งทำงานโดยการเพิ่มหรือลดความสว่างของพิกเซล ดังสมการ

$$V = V + b$$

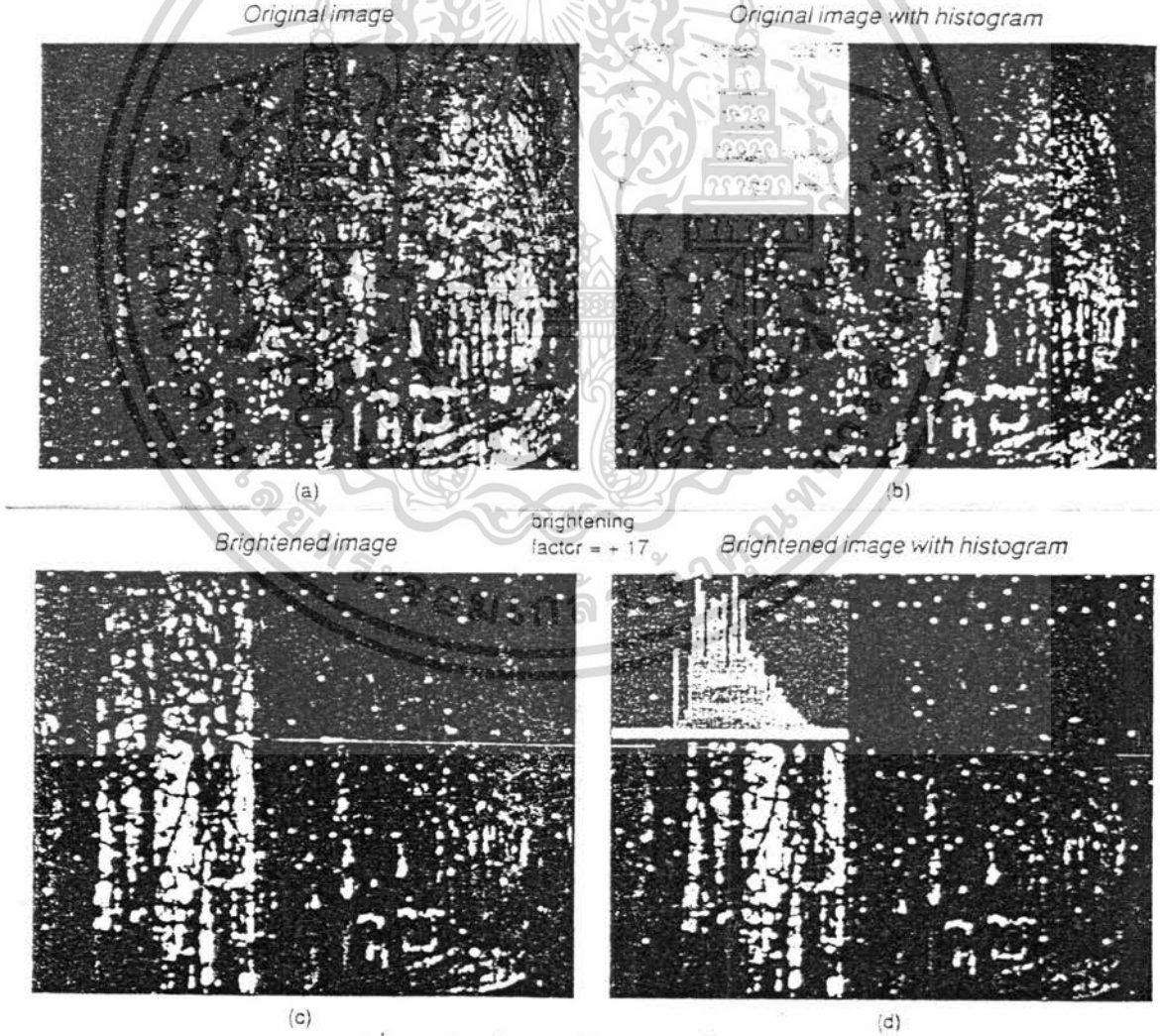
โดยที่ V คือ ค่าความสว่างของพิกเซลเดิมก่อนการเปลี่ยนแปลง

b คือ ค่าความสว่างซึ่งสามารถเป็นบวกหรือลบก็ได้ ถ้า b เป็นบวก ความสว่างของพิกเซลนั้นก็จะมีค่าเพิ่มขึ้น ถ้า b เป็นลบความสว่างของพิกเซลนั้นก็จะลดลง รูปที่ 2.3 แสดงถึงผลที่ได้จากการปรับความสว่างของภาพด้านซ้ายมือของรูป โดยค่า brightenness factor = +17 และแสดงถึงฮิสโตแกรมของภาพต้นฉบับ และ ภาพที่ผ่านการปรับความสว่างแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะพบว่าฮีสโตแกรม ของภาพที่ผ่านการปรับความสว่าง ใน ภาพ 2.3(d) มีความสัมพันธ์กับ (b) คือ ฮีสโตแกรมจะเลื่อนค่าไปทางขวาซึ่งหมายถึงภาพนั้นมีความสว่างมากขึ้น โดยระยะที่เลื่อนเท่ากับ brigtheness factor = +17 ดังนั้นเราอาจรู้จักการปรับความสว่างของภาพ (Image Brightening) ในชื่ออีกอย่างว่า การเลื่อนฮีสโตแกรม (Histogram Sliding)

การปรับค่าความสว่างของภาพ สามารถใช้สูตรที่กล่าวมาแล้ว สิ่งที่น่าสนใจในการแปลงนี้คือ ความสว่างของพิกเซลนั้นเมื่อถูกบวกหรือลบด้วย brightenness แล้วอาจเกิดการโอเวอร์โฟล์ว (overflow) หรือ อันเดอร์โฟล์ว (underflow) ได้ เช่น ถ้า brighteness factor มาบวกกับพิกเซลแล้วได้พิกเซลที่สว่างมากกว่า 255 เราจะเรียกว่าเกิด overflow ในทำนองเดียวกัน ถ้า brighteness factor มีค่าเป็นลบเมื่อนำมาคำนวนแล้วได้พิกเซลที่มีค่าติดลบ เราจะเรียกว่าเกิดอันเดอร์โฟล์วขึ้น ดังนั้นในการแปลงค่านี้เราก็จะตรวจสอบค่าใน LUTs ถ้าค่าใน LUTs มีค่าเกิน 255 จะให้ค่าเป็น 255 และ ถ้าค่าเป็นลบจะให้ค่าเป็น 0



รูปที่ 2.3 ตัวอย่างการปรับความสว่างของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.การกลับสีของภาพ (Negative Image)

วิธีนี้จะเปลี่ยนภาพจากสว่างเป็นมืดหรือจากมืดเป็นสว่าง หรือพูดอีกอย่างหนึ่งว่า เปลี่ยนเป็นตรงกันข้ามหมด คือ ถ้าจุดของภาพมืดเพียงเล็กน้อย วิธีนี้ก็จะเปลี่ยนจุดนั้นเป็นสว่างเล็กน้อย ถ้าในจุดนั้นสว่างมาก จุดนั้นก็จะเปลี่ยนเป็นมืดมาก อัลกอริทึมการกลับสีภาพ ก็จะทำงานโดยการลบค่าที่มากที่สุดคือ 255 ด้วยความสว่างของพิกเซลนั้นๆ ดังนั้นพื้นที่ของภาพที่มีค่าพิกเซลเป็น 0 ก็จะเปลี่ยนค่าพิกเซลเป็น 255, พื้นที่ของภาพที่มีค่าพิกเซลเป็น 1 จะเปลี่ยนค่าพิกเซลเป็น 254 , พื้นที่ของภาพที่มีค่าพิกเซลเป็น 3 จะเปลี่ยนค่า พิกเซลเป็น 253 , ไปเรื่อยๆ จนถึง พื้นที่ของภาพ ที่มีค่า พิกเซลเป็น 127 จะเปลี่ยนค่า พิกเซลเป็น 128

การแปลงนี้จะใช้ LUTs ด้วย และใช้ค่าเทรสโฮลในการบอกค่าเริ่มต้นของฟังก์ชันนี้ใน LUTs ถ้าเทรสโฮลเป็น 0 จะได้ทุก พิกเซลเปลี่ยนแบบตรงข้ามหมดดังที่ได้กล่าวมาแล้ว ถ้าเทรสโฮลมีค่าเป็นบวก จะได้ค่าพิกเซลที่เทรสโฮลนั้นชี้อยู่เป็นค่าเริ่มต้นที่ใช้ในอัลกอริทึมของการกลับสีของภาพ

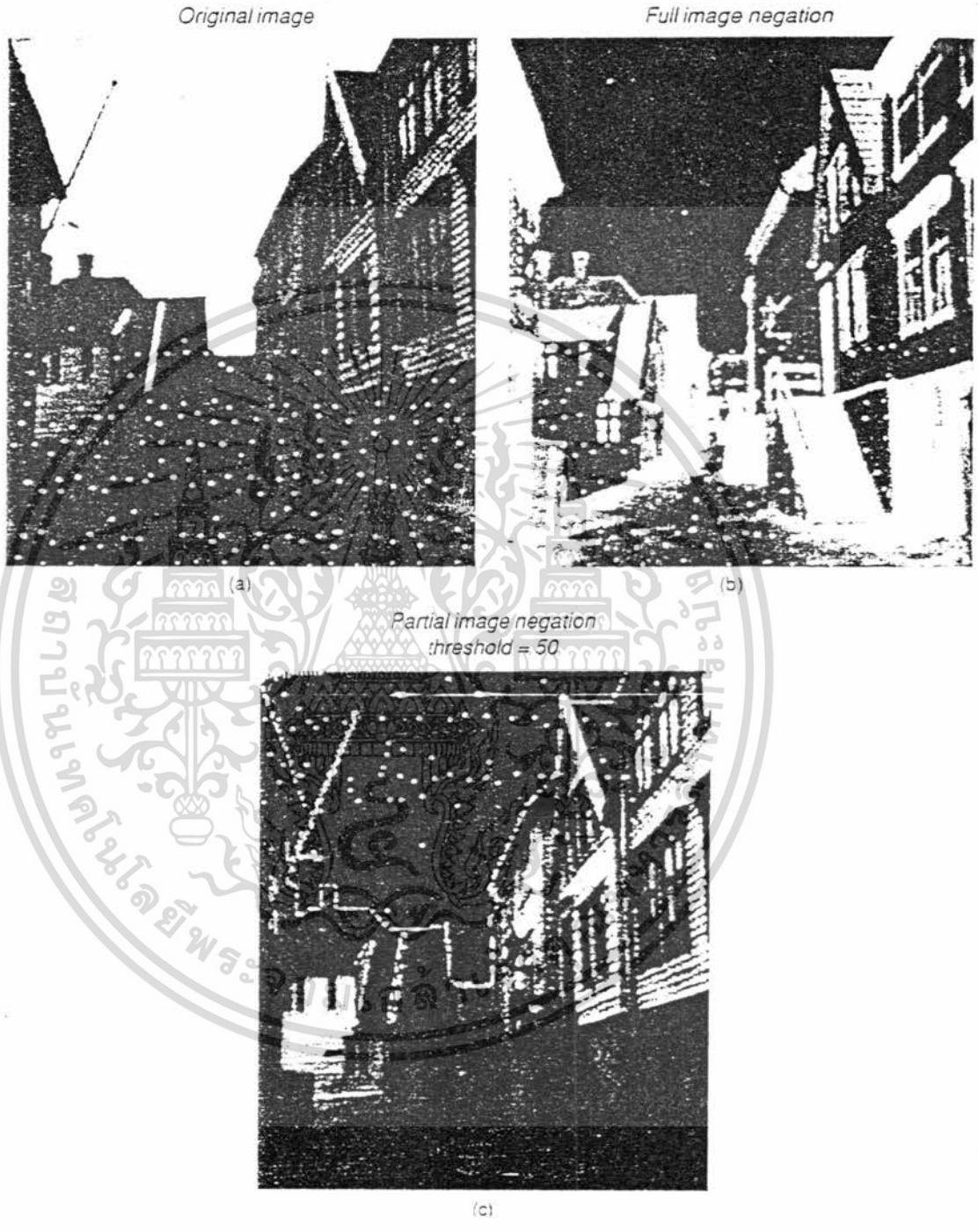
การกลับสีของภาพ มีประโยชน์ในการมองหารายละเอียดของภาพในส่วนที่มีความสว่างมาก ตาของมนุษย์จะสามารถมองเห็นรายละเอียดในบริเวณที่มืดของภาพได้มากกว่าบริเวณที่สว่าง รูปที่ 2.4 แสดงถึงผลของการกลับสีภาพ ภาพ (a) คือ ภาพดั้งเดิมก่อนการเปลี่ยนแปลงภาพ (b) เป็น การกลับสีของภาพแบบทำทั้งภาพ (c) เป็น การกลับสีของภาพเพียงบางส่วน คือในบริเวณที่สว่างค่าเทรสโฮลที่ตั้งสำหรับภาพนี้คือ 50

3.เทรสโฮล (Thresholding Image)

เป็นเทคนิคสำหรับเปลี่ยนความต่อเนื่องของโทนสีของภาพ (continuous-tone image) ให้เป็นภาพขาวดำ โดยพิกเซลที่สว่างน้อยกว่าค่าเทรสโฮล จะถูกเปลี่ยนเป็น 0 และ พิกเซลที่สว่างมากกว่าค่าเทรสโฮลจะถูกเปลี่ยนเป็น 255

การแปลงนี้เราใช้ LUTs เข้าช่วย โดยค่าใน LUTs ที่น้อยกว่า ค่าเทรสโฮลจะเปลี่ยนเป็น 0 และ ค่าที่มากว่าหรือเท่ากับค่าเทรสโฮลจะเปลี่ยนเป็น 255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ตัวอย่างการกลับสีภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 ตัวอย่างเทรสโฮล

4.การปรับความคมชัดของภาพ (Contrast Stretching)

คือการแยกความสว่างและมืดของภาพให้มีความเด่นชัดขึ้น ประเภทของ contrast แบ่งเป็น 3 ประเภท คือ low-contrast image, good-contrast image และ high-contrast image

- *low-contrast image* คือ ภาพที่ประกอบด้วยช่วงของความสว่างเพียงช่วงใดช่วงหนึ่งเพียงช่วงเดียว ส่วนใหญ่แล้วภาพที่มีลักษณะ low-contrast image จะมีสีที่ค่อนไปทางมืดหรือไม่ก็สว่างเพียงทางเดียว สำหรับฮีสโตแกรม ถ้ากลุ่มของค่าพิกเซลอยู่ ทางซ้ายของฮีสโตแกรม (ค่า พิกเซลมีค่าน้อย) จะได้ภาพค่อนข้างมืด ถ้ากลุ่มของค่าพิกเซลมีค่ามากจะได้ภาพที่สว่าง

- *good-contrast image* จะมีสีที่เฉลี่ยเท่าๆกันไปทั้งภาพไม่มีสีใดๆสีหนึ่งที่เด่นขึ้นมา ฮีสโตแกรมของ good-contrast image จะเป็นการกระจายแบบยูนิฟอร์ม (uniform) ของ ค่าพิกเซล ซึ่งจะไม่มีจุดสูงสุดหรือจุดต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

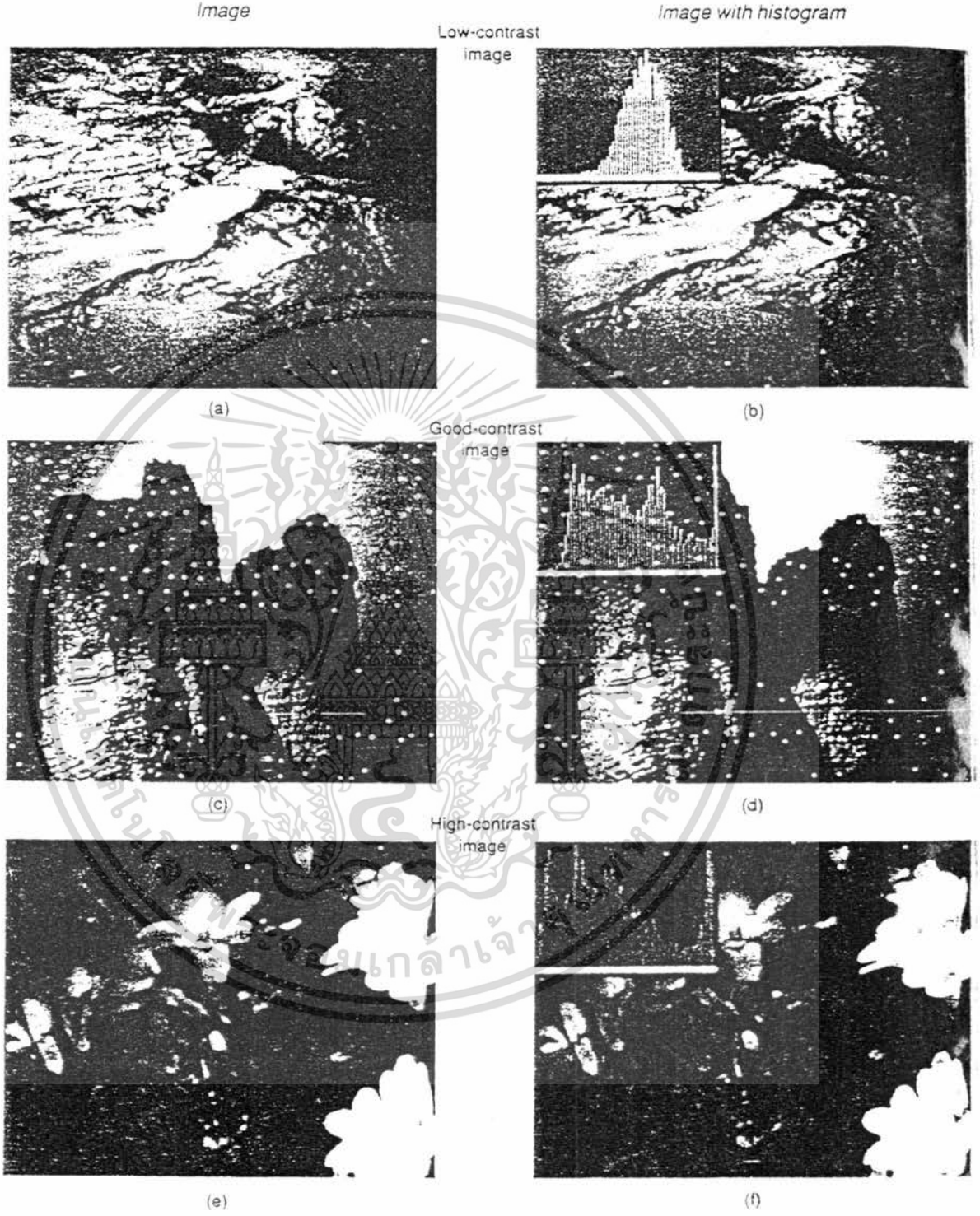
- *high-contrast image* จะมีสีที่เฉลี่ยเท่ากันทั้งภาพ แต่อย่างไรก็ตาม high-contrast image จะมีสีบางช่วงที่โดดเด่นขึ้นมา และมีสีบางช่วงที่มีเบาบางฮีสโตแกรมจะเป็นกราฟแบบไบโมดอล (bimodal) ซึ่งมี 2 ค่าสุดขีด คือจุดหนึ่งอยู่ในบริเวณค่าพิกเซลต่ำสุดและจุดหนึ่งอยู่ในบริเวณค่าพิกเซลสูงสุด

อัลกอริทึมคอนทราสสเตรชชิ่ง (contrast stretching) คือ จะใช้จุดที่เราเรียกว่าเทรสโฮล 2 จุด ทำการตั้งค่าความสว่างให้เป็น 0 ทั้งหมดสำหรับความสว่างที่น้อยกว่า ค่าของเทรสโฮล ส่วนค่าพิกเซลที่สว่างกว่าค่าเทรสโฮลจะถูกตั้งค่าให้เป็น 255 ค่าพิกเซลที่อยู่ระหว่าง 2 ค่า thresthold จะถูกคำนวนใหม่เพื่อให้ได้ค่าระหว่าง 0 - 255 รูปที่ 2.5 แสดงถึงผลของวิธีนี้

ประเภทอื่นๆ ของการแปลงภาพ แบบ good หรือ high-contrast ทำได้โดยเลือกค่าที่เหมาะสมใน LUT เราก็จะสามารถเปลี่ยนจาก good-contrast image เป็น high-contrast iamge ได้

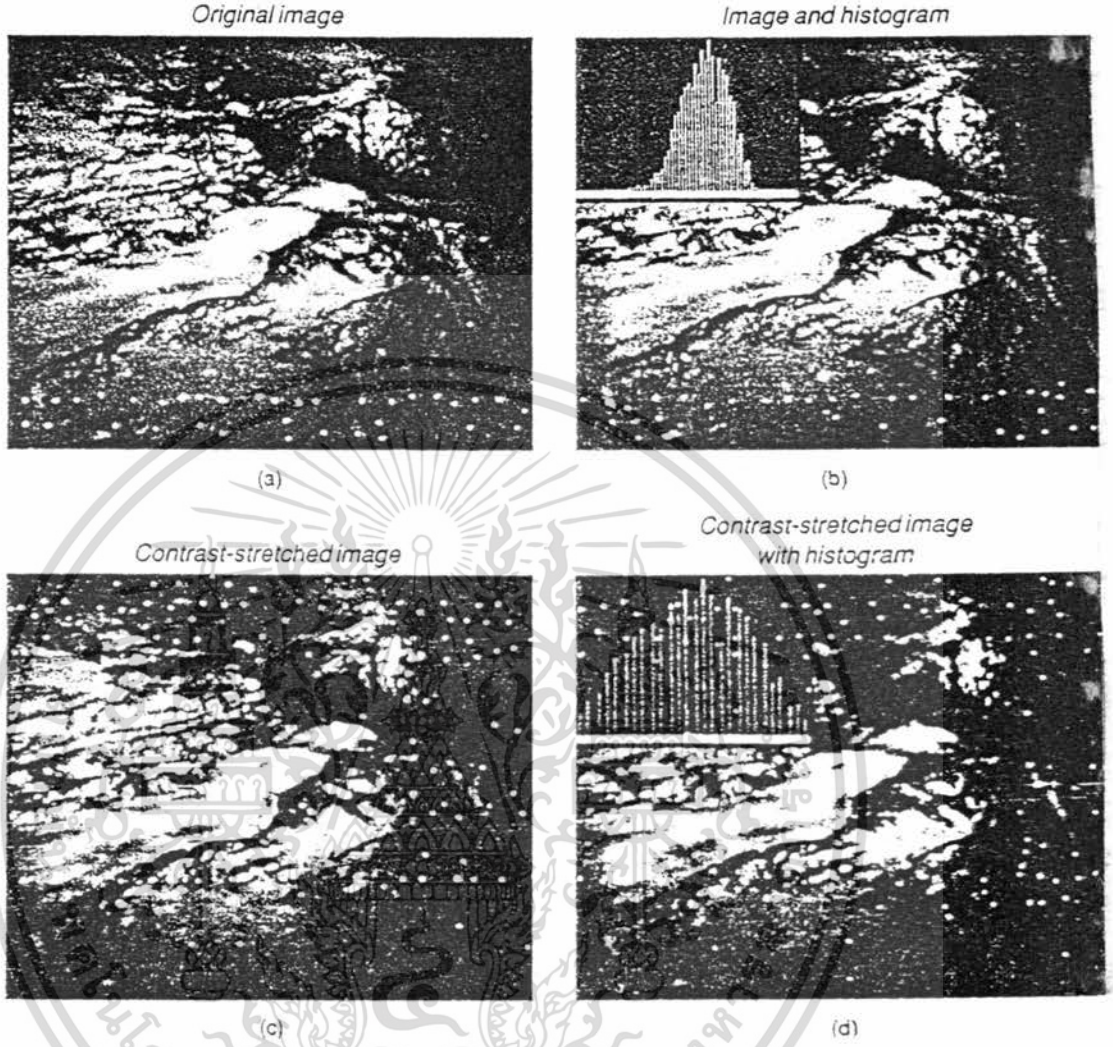
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 ชนิดของคอนทราส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ตัวอย่างการปรับความคมชัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตกแต่งภาพแบบบริเวณ (Area process)

คอนโวลูชัน (Convolution) เป็น วิธีคำนวนที่ใช้ผลรวมถ่วงน้ำหนักของความสว่างของแต่ละพิกเซลในเนเบอร์ฮูด (รูป 10.2 ขนาด 3 * 3) มาคูณกับ คอนโวลูชัน เคอร์เนล (convolution kernel) และผลรวมที่ได้ก็จะถูกใส่กลับไปที่ตำแหน่งศูนย์กลางสมาชิกของ เนเบอร์ฮูด เหมือนเดิม คอนโวลูชัน เคอร์เนล ก็คือ ค่าที่ใช้ถ่วงน้ำหนัก (หรือเรียกว่า convolution coefficient) ขนาดและสมาชิกของ คอนโวลูชัน เคอร์เนล ขึ้นจะอยู่กับประเภทการแปลงของภาพ การเปลี่ยนค่าถ่วงน้ำหนักใน คอนโวลูชัน เคอร์เนล มีผลกับขนาดและเครื่องหมายของผลรวมที่ได้ ซึ่งเป็นจะส่งผลต่อค่าของพิกเซลใหม่ รูป 10.3 แสดงถึง คอนโวลูชัน เคอร์เนล ต่างๆ ในรูป 10.3 จะพบว่าเคอร์เนล ส่วนใหญ่มีขนาด 3 *3 ซึ่งมีจำนวนแถวและหลักทั้งหมดเป็นจำนวนคี่ ขนาดของเคอร์เนล ส่วนใหญ่จะช่วยเพิ่มความยืดหยุ่นของคอนโวลูชันโปรเซส

ปัญหาของ คอนโวลูชันโปรเซส มีดังนี้

ข้อแรก คือ การประมวลผลที่ขอบของภาพนั้น เมื่อเอา คอนโวลูชัน เคอร์เนล (ซึ่งพิกเซลที่สนใจ เอา เคอร์เนล มาวางตรงกลาง) วางลงบนภาพ จะพบปัญหาในการคำนวณเมื่อเคอร์เนล อยู่บนขอบภาพ เพราะว่า ค่าถ่วงน้ำหนัก ใน เคอร์เนล มีตำแหน่งไม่ครบ 9 พิกเซล หรือพูดอีกอย่างว่า คอนโวลูชัน เคอร์เนล เกิดการ "HANGING" เมื่ออยู่ขอบที่ขอบของภาพ คือ ด้านบน, ซ้าย, ขวา และ ด้านล่างของภาพ วิธีที่จะแก้ปัญหานี้คือ

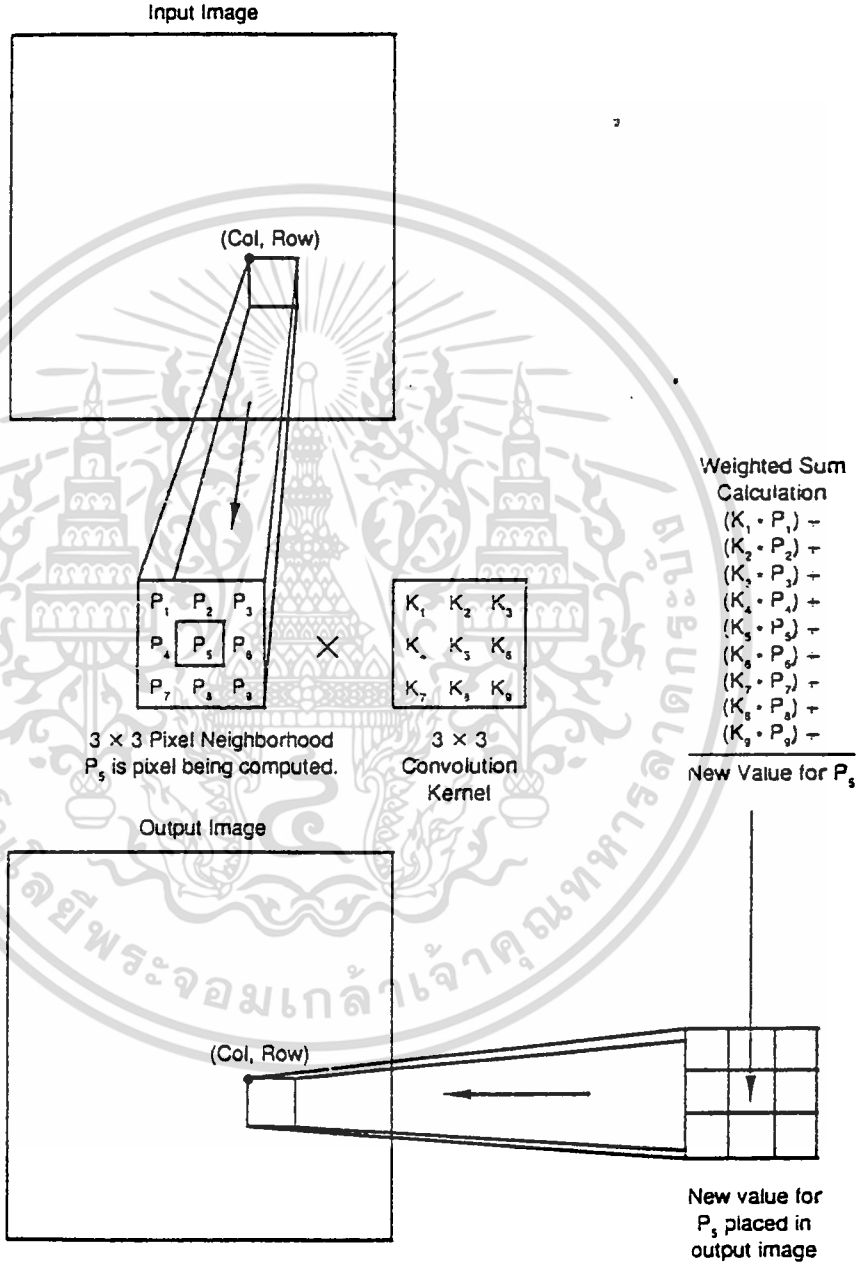
1) ข้อมูลที่ขอบของภาพ จะไม่ถูกใช้
2) ข้อมูลที่ขอบอาจถูกสังเคราะห์ (synthesize) แล้วนำไปเพิ่มลงในภาพเพื่อช่วยให้การคำนวนสมบูรณ์

ข้อสอง คำนวณค่า พิกเซลเก่าเพื่อให้ได้ค่าใหม่ โดยส่วนใหญ่ค่าใหม่ที่ได้จะอยู่ในช่วง 0 - 255 แต่ จะเห็นได้ว่าเมื่อนำเบลอริง เคอร์เนล (blurring kernel) มาคำนวณแล้วโอกาสที่ค่าที่ได้จะอยู่นอกช่วงนี้มีมาก จึงต้องมีการปรับขนาดความสว่าง (scaling) ที่ได้ใน อัลกอริทึมก่อน โดยการนำค่าที่ได้จากการคำนวนมาหารด้วย ผลรวมของสมาชิกใน คอนโวลูชัน เคอร์เนล

ต่อไปพิจารณาว่าทำไม เบอร์ริง เคอร์เนล ให้ค่ามากกว่า เคอร์เนล อื่นๆ ในรูป10.3 สังเกตว่า เคอร์เนล ทั้งหมดยกเว้น เบอร์ริง เคอร์เนล ผลรวมของสมาชิกในคอนโวลูชัน เคอร์เนล เป็น 0 หรือ 1 ส่วนผลรวมของ เบอร์ริง เคอร์เนล เป็น 25 ซึ่งหมายความว่าผลรวมถ่วงน้ำหนัก จะรวมค่าของ 25 พิกเซลที่อยู่ใน เนเบอร์ฮูด ขนาด 5 * 5 ตัวอย่างเช่น ถ้า พิกเซลแต่ละอันของ 25 พิกเซลมีค่าครึ่งหนึ่งของค่ามากสุด หรือ 128 จะได้ผลรวมเท่ากับ 25 * 128 หรือ 2200 ซึ่งเกินช่วง 0 -255 จึงต้องมีการนำผลที่ได้ไปทำให้อยู่ในช่วง 0 - 255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่พิจารณาสุดท้าย คือ เครื่องหมายของ พิกเซลที่คำนวณได้ เมื่อ คอนโวลูชัน เคอร์เนล มี ค่าถ่วงน้ำหนัก เป็นลบ ก็เป็นไปได้ที่จะได้ค่าพิกเซลเป็นลบ ค่าพิกเซลที่เป็นลบเราไม่สามารถแสดงได้ ด้วยเหตุนี้เราต้องแทนค่าพิกเซลที่เป็นลบด้วย 0



รูปที่ 2.8 คอนโวลูชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Pass Spatial Filters

LP1	LP2	LP3
$\frac{1}{9}$ $\frac{1}{9}$ $\frac{1}{9}$	$\frac{1}{10}$ $\frac{1}{10}$ $\frac{1}{10}$	$\frac{1}{16}$ $\frac{1}{8}$ $\frac{1}{16}$
$\frac{1}{9}$ $\frac{1}{9}$ $\frac{1}{9}$	$\frac{1}{10}$ $\frac{1}{5}$ $\frac{1}{10}$	$\frac{1}{8}$ $\frac{1}{4}$ $\frac{1}{8}$
$\frac{1}{9}$ $\frac{1}{9}$ $\frac{1}{9}$	$\frac{1}{10}$ $\frac{1}{10}$ $\frac{1}{10}$	$\frac{1}{16}$ $\frac{1}{8}$ $\frac{1}{16}$

$\Sigma = 1$

High-Pass Spatial Filters

HP1	HP2	HP3
−1 −1 −1	0 −1 0	1 −2 1
−1 9 −1	−1 5 −1	−2 5 −2
−1 −1 −1	0 −1 0	1 −2 1

$\Sigma = 1$

Shift and Difference Edge Enhancements

Vertical Edges	Horizontal Edges	Horizontal and Vertical Edges
0 0 0	0 −1 0	−1 0 0
−1 1 0	0 1 0	0 1 0
0 0 0	0 0 0	0 0 0

$\Sigma = 0$

Matched Filter Edge Enhancements[1]

Vertical Edges	Horizontal Edges
−1 0 1	−1 −1 −1 −1 −1
−1 0 1	0 0 0 0 0
−1 0 1	1 1 1 1 1
−1 0 1	
−1 0 1	

$\Sigma = 0$

Gradient Directional Edge Enhancements

North	Northeast	East	Southeast
1 1 1	1 1 1	−1 1 1	−1 −1 1
1 −2 1	−1 −2 1	−1 −2 1	−1 −2 1
−1 −1 −1	−1 −1 1	−1 1 1	1 1 1

South	Southwest	West	Northwest
−1 −1 −1	1 −1 −1	1 1 −1	1 1 1
1 −2 1	1 −2 −1	1 −2 −1	1 −2 −1
1 1 1	1 1 1	1 1 −1	1 −1 −1

$\Sigma = 0$

Blurring Kernel

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

$\Sigma = 25$

Laplace Edge Enhancements

LAP1[2,3]	LAP2	LAP3[2]	LAP4
0 1 0	−1 −1 −1	−1 −1 −1	1 −2 1
1 −4 1	−1 8 −1	−1 9 −1	−2 4 −2
0 1 0	−1 −1 −1	−1 −1 −1	1 −2 1

Σ normally = 0 except LAP3

Notes

1. All kernels from Baxes except these from Dawson.
2. Equivalent to adding original image to output of convolution using kernel LAP2.
3. Proof can be found in Gonzalez and Wintz.

รูปที่ 2.9 ตัวอย่างคอนโวลูชันเคอร์เนล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคนิคที่จะกล่าวถึงมีดังนี้

1 โลว์-พาส สปาเชียล ฟิลเตอร์ (Low-pass spatial filtering)

2 ไฮ-พาส สปาเชียล ฟิลเตอร์ (High-pass spatial filtering)

3 การเน้นขอบรูป (Edge Enhancement)

3.1 แบบลาปลาเชียล (Laplacian Edge Enhancement)

3.2 แบบการเลื่อนและผลต่าง (Shift and DifferenceEdge Enhancement)

3.3 แบบกราเดียน ไดเรกชันนอล (Gradient Directional Edge Enhancement)

4 การทำภาพให้มัว (Image Blurring)

5 การหาขอบด้วยอัลกอริทึมของโซเบล (Edge Detection with Sobel's Algorithm)

6 แบบใช้ค่ากลาง (Median Filtering)

การตกแต่งภาพแบบบริเวณ (Area process) จะใช้กลุ่มของพิกเซลเพื่อใช้ในการประมวล ซึ่งต่างจากการประมวลผลภาพแบบจุดที่ใช้ข้อมูลจุดเดียว กลุ่มของพิกเซลในการประมวลผลภาพแบบจุด เรียกว่าเนเบอร์ฮูด (neighborhood) โดยทั่วไปเนเบอร์ฮูดจะเป็นเมตริกซ์ 2 มิติ ของความสว่างในพิกเซลนั้น ซึ่งเมตริกซ์จะมีมิติเป็นเลขคี่ ความสว่างของพิกเซลใหม่จะเกิดจากการคำนวณซึ่งหาได้จากอัลกอริทึมของการตกแต่งภาพแบบบริเวณ ซึ่งค่าที่ได้ใหม่นี้จะวางในตำแหน่งศูนย์กลางของ เนเบอร์ฮูด ที่นำมาคำนวณ

สปาเชียล ฟรีเควนซี (spatial frequency) คือ ค่าที่เกิดจากการหารอัตราการเปลี่ยนแปลงความสว่างของ พิกเซลหรือความเข้มด้วยระยะของการเปลี่ยนแปลงที่เกิดขึ้น สปาเชียล ฟรีเควนซี มีทิศทางทั้งแนวตั้งและแนวนอนในภาพ ภาพแบบ ไฮ สปาเชียล ฟรีเควนซี (high spatial freuency) จะมีความแตกต่างของความสว่างพิกเซลอย่างรวดเร็วในระยะใกล้ เช่น ภาพของตารางหมากรุกจะเป็นแบบ ไฮ สปาเชียล ฟรีเควนซี ภาพแบบโลว์ สปาเชียล ฟรีเควนซี (low spatial frequency) จะมีความต่างของความสว่างของพิกเซลจะเป็นอย่างค่อยเป็นค่อยไป ไม่เปลี่ยนแปลงมากนัก ตัวอย่างเช่น ภาพของเมฆจะเป็นแบบ โลว์ สปาเชียล ฟรีเควนซี

สปาเชียล ฟิลเตอร์ริง (spatial filtering) ได้มีการประยุกต์ใช้อย่างมากมาย เช่น การตรวจขอบของรูป (edge detection), การทำให้ภาพคมชัดขึ้น (sharpening) , การทำให้ภาพเรียบขึ้น (smooting) และการทำให้ภาพไม่ชัดเจน (blurring)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อัลกอริทึมของการประมวลผลภาพแบบบริเวณคล้ายกับการประมวลผลภาพแบบจุด คือ

1. การประมวลผลสามารถกระทำได้ครั้งละหนึ่งพิกเซล

2. พิกเซลแต่ละพิกเซลในภาพที่ต้องการประมวลผลจะถูกดำเนินการผ่านการคำนวณเพื่อให้ได้ค่าใหม่

3. ค่าใหม่ของพิกเซลถูกวางกลับลงไปในตำแหน่งเดิม ไม่มีการเปลี่ยนตำแหน่ง

สิ่งที่แตกต่างระหว่าง การประมวลผลภาพแบบจุด และ การประมวลผลภาพแบบจุด คือ ข้อ (2) การประมวลผลภาพแบบจุด ใช้ระดับความสว่างของพิกเซลเพียงพิกเซลเดียวเป็นค่าที่นำไปคำนวณเพื่อหาความสว่างของจุดๆนั้นใหม่ แต่ การประมวลผลภาพแบบบริเวณ จะใช้ความสว่างของ เนเบอร์ฮูดเมตริกซ์ (neighbourhood metrix) ของจุดๆนั้นในการหาค่าความสว่างของจุดๆนั้นใหม่

สำหรับการประมวลผลภาพแบบบริเวณ มีจำนวนพิกเซลที่ใช้คำนวนการแปลงมากกว่า การประมวลผลภาพแบบจุด พิจารณาเมตริกซ์ 3 * 3 พิกเซลก็หมายความว่าเราต้องใช้ความสว่างของพิกเซล 9 พิกเซล ดังนั้นการประมวลผลภาพแบบบริเวณจะต้องคำนวณถึง 9 ครั้ง ซึ่งมากกว่า การประมวลผลภาพแบบจุด ซึ่งคำนวณครั้งเดียว จึงจะพบว่า การประมวลผลภาพแบบบริเวณนั้น ใช้เวลาในการประมวลผลนานโดยที่เวลาที่ใช้จะขึ้นอยู่กับขนาดของ เนเบอร์ฮูด ที่ใช้

1.โลว์-พาส สปาเชียล ฟิลเตอร์ (Low-Pass Spatial Filters)

โลว์-พาส สปาเชียล ฟิลเตอร์ (Low-Pass Spatial Filters) จะปล่อยให้ส่วนของภาพที่เป็นแบบ โลว์ สปาเชียล ฟรีเควนซี (low spatial frequency) ผ่านไปใน แต่ส่วนที่เป็นแบบ ไฮ สปาเชียล ฟรีเควนซี (high spatial frequency) ก็จะถูกกรองให้มีความแตกต่างน้อยลง โลว์-พาส สปาเชียล ฟิลเตอร์ ถูกใช้ในการลดความบกพร่องในภาพ รูป 2.9 แสดงถึง โลว์-พาส สปาเชียล เคอร์เนล ทั้ง 3 แบบ สังเกตว่า ผลรวมของค่า เคอร์เนล ของทั้ง 3 แบบเป็น 1

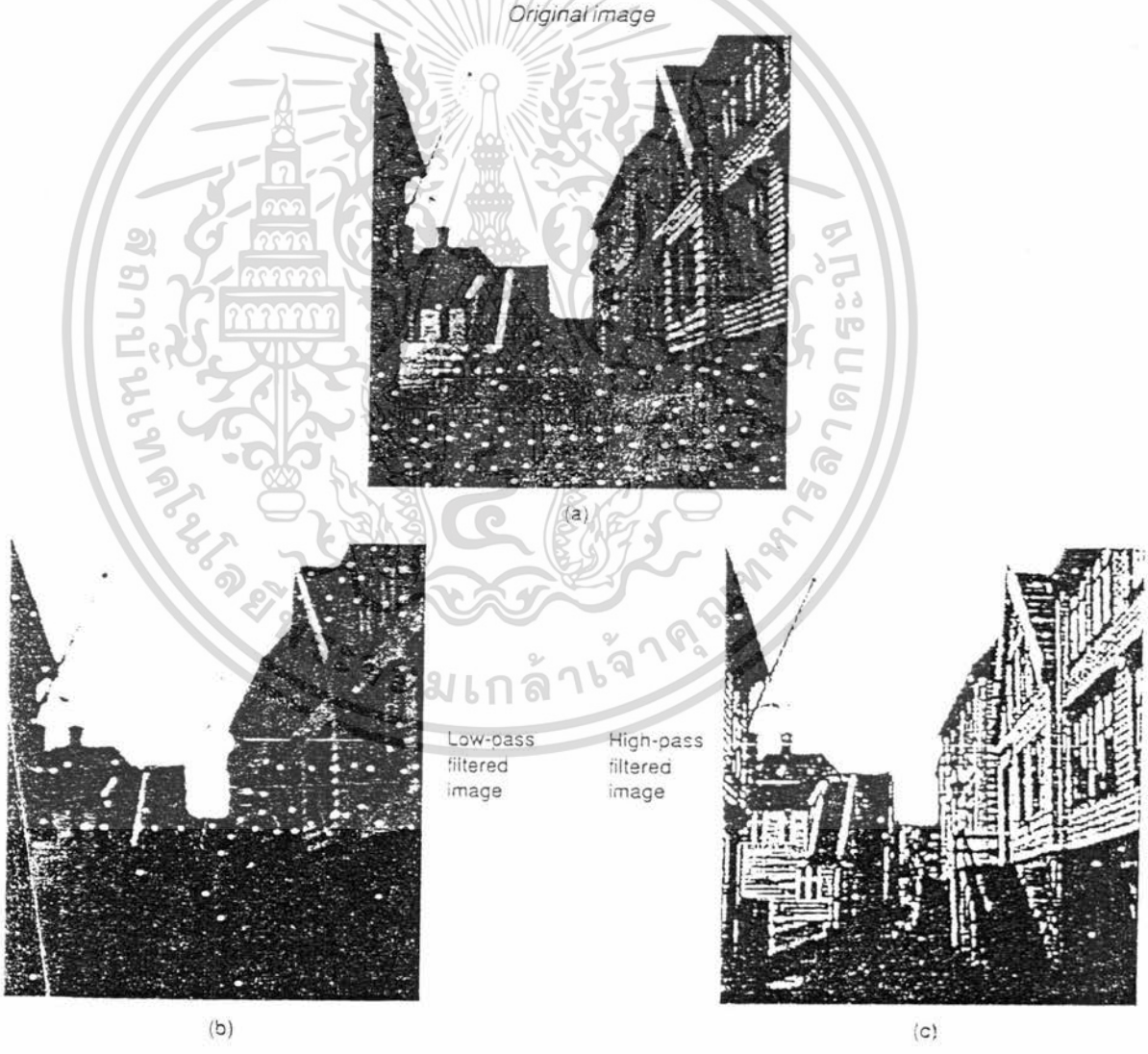
พิจารณาบางส่วนของภาพ ซึ่งไม่มี ไฮ สปาเชียล ฟรีเควนซี ซึ่งก็หมายความว่า ความสว่างของพิกเซลคงที่หรือมีการเปลี่ยนแปลงอย่างช้าๆ เมื่อ โลว์-พาส สปาเชียล เคอร์เนล ถูกใช้คำนวณกับบางส่วนของภาพ ค่าใหม่ของพิกเซลจะถูกคำนวณจากผลรวมของ ค่าถ่วงน้ำหนัก คูณกับค่าเนเบอร์ฮูด พิกเซลถ้าค่า เนเบอร์ฮูด พิกเซลมีค่าเหมือนกัน (คงที่) ค่าใหม่ที่ได้จะเหมือนกับค่าเดิมเพราะว่าผลรวมของ ค่าถ่วงน้ำหนัก เป็น 1 เมื่อ เคอร์เนล ใช้กับส่วนของภาพแบบ ไฮ สปาเชียล ฟรีเควนซี การเปลี่ยนแปลงอย่างรวดเร็วของความเข้มของพิกเซลต้องเฉลี่ยกับ เนเบอร์ฮูด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิกเซลที่เหลือ ทำให้ลดความเป็น ไฮ สปาเชียล ฟรีเควนซี ลง ผลที่ได้ของ โลว์-พาส สปาเชียล ฟิลเตอร์ คือ การ เบลอ เล็กน้อยของภาพ

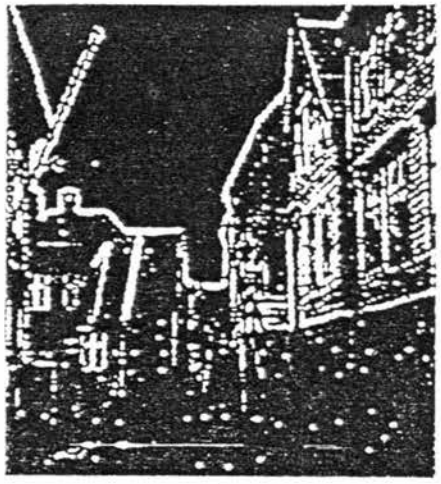
การ เบลอ นี้เกิดจาก พิกเซลที่คมชัดต้องเฉลี่ยกับ เนเบอร์ฮูด พิกเซลทำให้ ไฮ สปาเชียล ฟรีเควนซี ลดลง รูปที่ 2.10 (a) แสดงภาพดั้งเดิม 2.10 (b) low-pass filtered image

โลว์-พาส สปาเชียล ฟิลเตอร์ อาจถูกใช้ทำให้ภาพคมชัดขึ้น ถ้า ภาพที่ผ่านโลว์-พาส สปาเชียล ฟิลเตอร์ ถูกแยกออกจากภาพดั้งเดิม ทำให้ ไฮ สปาเชียล ฟรีเควนซี เพิ่มขึ้น แต่ไม่ทำให้ความบกพร่องของภาพ เพิ่มขึ้น ฉะนั้นภาพที่ได้คมชัดกว่าภาพเดิม ภาพที่ใช้ส่วนใหญ่เป็นภาพที่มีเมฆหรือหมอก เทคนิคนี้ถูกใช้เพื่อให้ภาพของ Los Angelis ดูดีในวันที่มีหมอกจัด



รูปที่ 2.10 ตัวอย่างการประมวลผลแบบบริเวณ

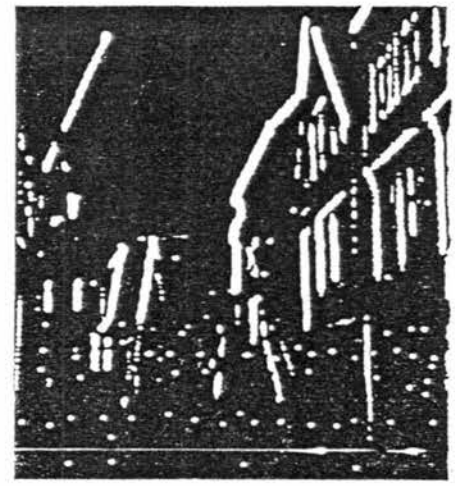
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(d)

Laplace
edge
enhancement

Gradient
edge
enhancement
north



(e)

(f)

Gradient
edge
enhancement
east

Blurred
image

(g)

รูปที่ 2.10 (ต่อ) ตัวอย่างการประมวลผลภาพแบบบริเวณ

2.ไฮ-พาส สปาเชียล ฟิลเตอร์ (High-Pass Spatial Filters)

ไฮ-พาส สปาเชียล ฟิลเตอร์ (high-pass spatial filters) จะเน้นข้อมูลที่ ไฮ สปาเชียล ฟรีเควนซี ของภาพ ในขณะที่ไม่สนใจ โลว์ สปาเชียล ฟรีเควนซี ไฮ-พาส สปาเชียล ฟิลเตอร์ ถูกใช้เมื่อวัตถุที่มี ไฮ สปาเชียล ฟรีเควนซี จะถูกเน้น ส่วนของภาพที่เป็นแบบ ไฮ สปาเชียล ฟรีเควนซีมากๆ จะถูกทำให้สว่างขึ้น ขณะที่ส่วน โลว์ สปาเชียล ฟรีเควนซี จะกลายเป็นสีดำ ความคมชัดของภาพจะถูกทำให้เพิ่มขึ้นด้วย ไฮ-พาส สปาเชียล ฟิลเตอร์ ซึ่งจะเพิ่มความบกพร่องของภาพด้วย การเน้นขอบรูป (Edge enhancement) ของภาพอาจใช้ ไฮ-พาส สปาเชียล ฟิลเตอร์ รูปที่ 2.9 แสดง ไฮ-พาส สปาเชียล ฟิลเตอร์ เคอร์เนล ทั้ง 3 แบบและ รูปที่ 2.10 (c) แสดงถึงผลของ ไฮ-พาส สปาเชียล ฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.การเน้นขอบรูป (Edge Enhancement)

การเน้นขอบรูป (Edge Enhancement) เป็นวิธีหนึ่งของ การประมวลผลภาพแบบจุด โดยที่ อัลกอริทึมนี้จะลดภาพ ให้เหลือเพียงขอบรูป ภาพใหม่ที่ได้อาจไม่คล้ายรูปเดิม ความสว่างของขอบรูปที่ได้จะขึ้นกับการเปลี่ยนแปลงความสว่างรอบขอบรูปเดิม เช่น ข้อมูลของขอบรูปที่ได้จากวิธี การเน้นขอบรูป เอาไปแต่งเติมกับรูปเดิม ทำให้รูปคมชัดขึ้น การเน้นขอบรูป อาจเป็นเครื่องมือสำหรับวาดโครงรูป

3.1 การหาขอบแบบลาปลาเชียล (Laplacian Edge Enhancement)

การหาขอบแบบลาปลาเชียล (Laplacian edge enhancement) ต่างจากวิธีการเน้นอื่นๆ โดยไม่คำนึงถึงทิศทางนั่นคือ จะได้ขอบรูปสว่างขึ้น เรียกวิธีนี้ว่า การหาขอบแบบลาปลาเชียล เพราะว่าการประมาณนี้แปลงจาก ลาปลาเชียล ทางคณิตศาสตร์

ฟังก์ชันของ laplacian f(x, y)

$$L(f(x, y)) = d^2f / dx^2 + d^2f / dy^2$$

เมื่อ d^2f / dx^2 คือ อนุพันธ์อันดับสองของ f ซึ่งขึ้นกับ x

d^2f / dy^2 คือ อนุพันธ์อันดับสองของ f ซึ่งขึ้นกับ y

สำหรับฟังก์ชัน ดีสครีต ประมาณอนุพันธ์อันดับสองได้

$$d^2f / dx^2 = f(x+1) - 2f(x) + f(x-1)$$

และ $d^2f / dy^2 = f(y+1) - 2f(y) + f(y-1)$

ดังนั้น ลาปลาเชียล ถูกประมาณโดย

$$L(f(x, y)) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

ดังนั้นเราจะได้ คอนโวลูชัน เคอร์เนล ซึ่งมีความสัมพันธ์กับ f(x, y) คือ

$$(x, y) = \begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

เคอร์เนล นี้ คือ LAP 1 ในรูป 2.9

วิธีการเน้นขอบรูปทั้งหมดรวมทั้งลาปลาเชียลจะลดโลว์ สปาเชียล ฟรีเควนซี ของภาพลง บริเวณที่ความเข้มคงที่หรือความเข้มเพิ่มขึ้นคงที่จะกลายเป็นสีดำผ่านการแปลงและบริเวณที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเข้มเปลี่ยนแปลงอย่างรวดเร็วจะสว่างขึ้น คอนโวลูชัน เคอร์เนล ลด โลว์ ฟรีเควนซี ลงอย่าง มีสัมประสิทธิ์ที่ผลรวมเป็น 0

ลาปลาเชียล คอนโวลูชัน เคอร์เนล สามารถให้ภาพแตกต่างหลายอย่าง ตัวอย่างเช่น ภาพ LAP3 ที่ได้จะเอาไปรวมกับภาพเดิม

3.2 แบบการเลื่อนและผลต่าง (The Shift and Difference Edge Enhancement)

อัลกอริทึมของการเน้นขอบรูปแบบนี้ คือ การเลื่อน (shift) ภาพไป 1 พิกเซลแล้วลบภาพที่เลื่อน (shift) แล้ว ออกจากภาพเดิม ผลที่ได้คือความแตกต่างของความสว่าง ในพื้นที่ของความเข้มของ พิกเซลคงที่ ผลของความแตกต่างที่ได้จะเป็น 0 ในพื้นที่เปลี่ยนความเข้มสูง เช่น ขอบรูป ผลที่ได้คือ มีความแตกต่างมาก ทำให้ พิกเซลสว่างขึ้น สิ่งที่ควรระวัง คือความแตกต่างมีค่าลบ ทำให้ค่า พิกเซลเปลี่ยนจาก 255 เป็น 0 ดังนั้น จะต้องมีค่าสัมบูรณ์ ที่ใช้ในแบบการเลื่อนและผลต่าง เพื่อป้องกันการเปลี่ยน ค่า พิกเซลจาก 255 เป็น 0 หรือ จาก 0 เป็น 255

การเน้นขอบตามแนวตั้ง คือ ภาพถูกเลื่อน(shift)ทางซ้าย 1 พิกเซลแล้วลบออกจากรูปเดิม

การเน้นขอบตามแนวนอน คือ ภาพถูกเลื่อน(shift)ขึ้นข้างบน 1 พิกเซลแล้วลบออกจากรูปเดิม

การเน้นขอบทั้งแนวนอนและแนวตั้ง คือ ภาพถูกเลื่อน(shift) ทางซ้าย 1 พิกเซลและขึ้นข้างบน 1 พิกเซลแล้วลบออกจากรูปเดิม

รูป 2.9 แสดงถึง คอนโวลูชัน เคอร์เนล ต่างๆ ของ แบบการเลื่อนและผลต่าง เคอร์เนลเหล่านี้ มาจากอัลกอริทึมการเน้นขอบ แบบการเลื่อนและผลต่าง ตัวอย่างเช่น การเน้นขอบตามแนวนอน คือ การเลื่อนไปทางซ้าย 1 พิกเซลแล้วลบออก เคอร์เนล ที่ได้ เหมือนอัลกอริทึม ถ้าเคอร์เนล เหล่านี้ผ่านพื้นที่ที่มีความเข้มคงที่ ผลที่ได้จาก คอนโวลูชัน คือ 0 เพราะว่า

-1 * ความเข้ม + 1* ความเข้ม = 0

ถ้ามีการเปลี่ยนความเข้มมาก ผลทีได้ จะมีทั้งตัวเลขที่เป็นบวก (การเปลี่ยน 0 เป็น 255) และตัวเลขที่ติดลบ (เปลี่ยนจาก 255 เป็น 0) ความเข้มของ พิกเซลที่ได้ขึ้นอยู่กับ ความชันของความเข้ม

3.3แบบกราเดียน ไดเรกชันนอล (Gradient Directional Edge Enhancement)

วิธี แบบการเลื่อนและผลต่าง แสดงวิธีที่ทำให้ขอบตามแนวตั้งและแนวนอนในภาพชัดเจนขึ้น อัลกอริทึม ของ การเน้นขอบ ส่วนใหญ่ใช้ เคอร์เนล ขนาด 3 * 3 มีการใช้ คอนโวลู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชัน เคอร์เนล ที่แตกต่าง 8 เคอร์เนล ซึ่งทิศทางเหมือนเข็มทิศคือ "North", "northeast","East", " Southeast", "South", "Southwest", "West" และ "Northwest" ซึ่งแสดงในรูป 2.9

ถ้าความชันในทิศทางของ เคอร์เนล เป็นบวก พิกเซลที่มีสีอ่อนจะถูกวางใน รูปผลลัพธ์ ความเข้มของพิกเซลรูปจะขึ้นอยู่กับความชันของความสว่าง ความชันยิ่งมาก ความสว่างของพิกเซลก็ยิ่งเพิ่มขึ้น ตัวอย่างของ "East" gradient เคอร์เนล จะทำให้ขอบชัดเจนขึ้น ซึ่งเปลี่ยนจากดำเป็นขาวจากซ้ายเป็นขวา รูป Image Sequence 10.1 แสดงถึง "North" และ "West" กราเดียน โอเปอร์เรชัน เราจะพบว่า ถ้าผลรวมค่าสัมประสิทธ์ เคอร์เนล เป็น 0 จะได้ว่าบริเวณของความสว่างคงที่จะได้ ผลลัพธ์ เป็น 0

4. การทำภาพให้มัว (Image Blurring)

การทำภาพมัว เป็นเทคนิคให้ฉากหลังมัว และวางวัตถุไว้ฉากหน้า ตัวอย่าง เช่น ภาพงานแต่งงานที่เจ้าสาวและเจ้าบ่าวชัดเจน แต่ฉากหลังมัว ความคมชัดและความมัว มีผลต่อสายตา

การทำภาพมัว ใช้ คอนโวลูชัน ซึ่ง เคอร์เนล ของ ภาพมัว ถูกแสดงในรูป 2.9 ซึ่งมีขนาด 5 * 5 มิติของ เคอร์เนล ยิ่งใหญ่ขึ้น ภาพที่ได้ยิ่งมัว ขึ้น คอนโวลูชัน ของเคอร์เนลการทำภาพมัว จะเฉลี่ยค่า พิกเซลของ เนเบอร์ฮูด ทุก พิกเซลการเฉลี่ยทำให้รายละเอียดของภาพ ลดลงมีผลทำให้ภาพมัว

การใช้ค่าสุดยอด เพื่อป้องกันไม่ให้ ภาพที่ได้มีค่าความสว่างเกินช่วง เพราะว่าไม่มี คอนโวลูชัน เคอร์เนล ที่เป็นค่าลบ เพื่อลดค่าผลรวม คอนโวลูชัน ค่าสุดยอด ที่ถูกเลือกขึ้นอยู่กับภาพที่ได้

5. การหาขอบด้วยอัลกอริทึมของโซเบล (Edge Detection with Sobel's Algorithm)

วิธีสำหรับอัลกอริทึมของโซเบล มี 2 วิธี โดยวิธีแรกคำนวณจาก คอนโวลูชัน 2 ชุด คือ

$$\text{kernel } x = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \qquad \text{kernel } y = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก คอนโวลูชัน คำนวณขนาด และทิศทางของการหาขอบ จาก

ขนาด = (x2 + y2)1/2

ทิศทาง = arctan (y/x)

วิธีที่ให้เข้าใจง่าย ขั้นแรก พิจารณา เนเบอร์ฮูด พิกเซลที่มีขนาด 3*3 ซึ่งมีสมาชิกดังนี้

a b c

d e f

g h i

เส้น 4 เส้น ลากผ่าน เนเบอร์ฮูด เหล่านี้ซึ่งผ่าน พิกเซลตรงกลาง (e) ซึ่งเส้นเหล่านี้คือ

เส้นที่ 1 : a - e - i

เส้นที่ 2 : b - e - h

เส้นที่ 3 : c - e - g

เส้นที่ 4 : d - e - f

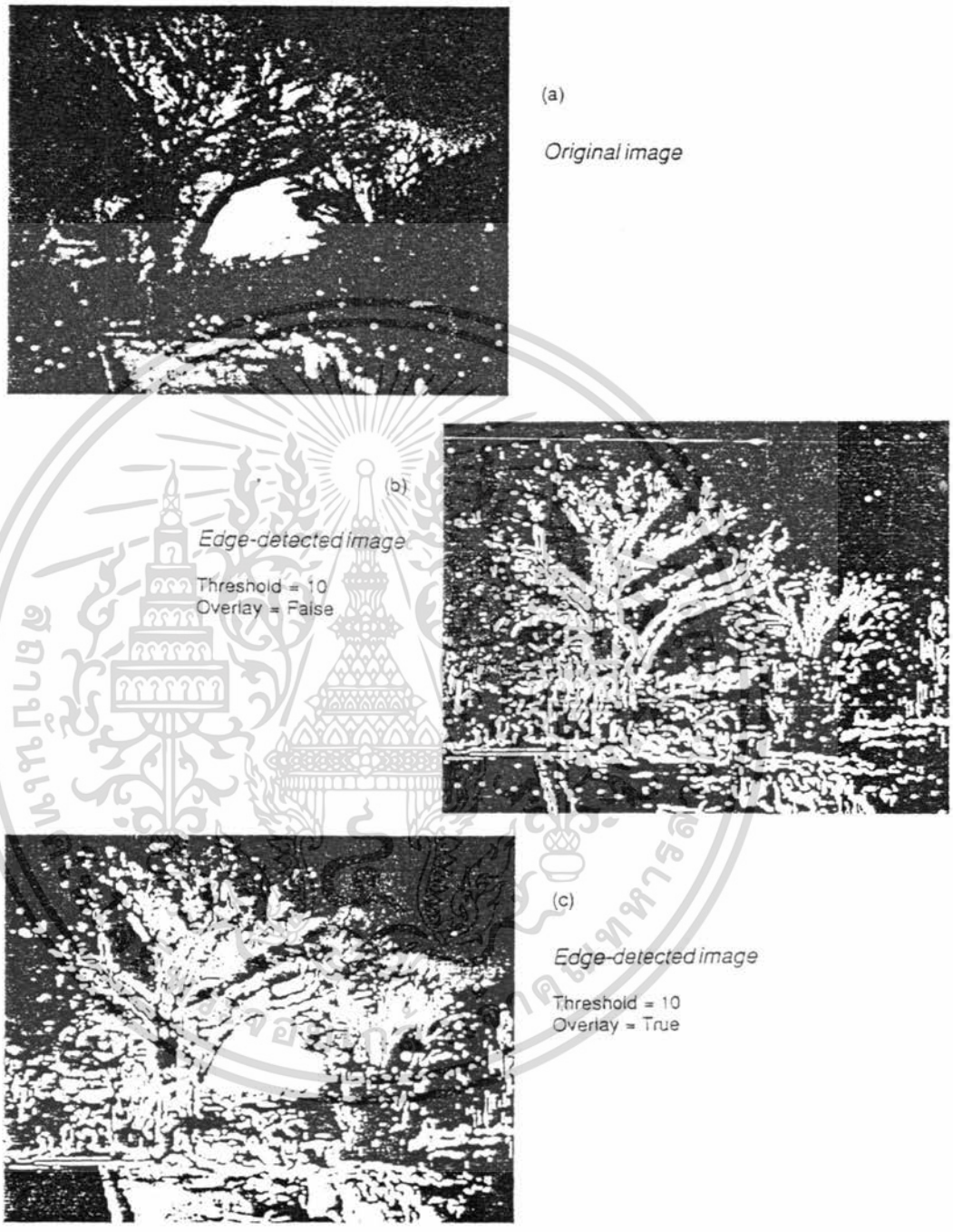
เส้นแต่ละเส้นผ่าน เนเบอร์ฮูด พิกเซลแบ่ง พิกเซลเป็น 2 ชุด ชุดละ 3 พิกเซล ตัวอย่าง เช่น เส้นที่ 1 แบ่ง พิกเซล เป็น เนเบอร์ฮูด d, g และ เนเบอร์ฮูด b, c และ f

สำหรับแต่ละสี่เส้น ค่าสัมบูรณ์ของผลต่าง ในการเฉลี่ย 2 เนเบอร์ฮูด ได้ถูก คำนวณ ค่า พิกเซลที่ได้คือ ค่าที่ใหญ่ที่สุดของค่าสัมบูรณ์ของ 4 เส้น

อัลกอริทึมของโซเบล จะใช้พิกเซลทีละพิกเซลของภาพ แล้วภาพที่ได้จะขึ้นอยู่กับจุดเทรสโฮล ค่าพิกเซลใหม่ ถ้าเกินค่า
เทรสโฮล สีของ พิกเซลที่ได้จะเป็น 255 ถ้าต่ำกว่าค่าเทรสโฮล สีของ พิกเซลที่ได้จะเป็น 0 ดังนั้น ผลลัพธ์ที่ได้จาก อัลกอริทึมของโซเบล ขึ้นกับจุดเทรสโฮล จะได้ภาพที่มีค่าพิกเซลเป็น 255 และ 0 ยกเว้น ที่ขอบ รูปที่ 2.11 แสดงถึงผลจากค่าเทรสโฮล

ค่าเทรสโฮล ที่ถูกเลือก ทำให้ได้รูปผลลัพธ์ ที่แตกต่างกัน รูปที่ 2.11 แสดงวิธีการหาขอบของโซเบล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ตัวอย่างการหาขอบด้วยวิธีโซเบล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. การใช้ค่ากลาง (Median Filtering)

วิธีการใช้ค่ากลาง ใช้ค่าของพิกเซลในเนเบอร์ฮูดพิกเซลเพื่อให้ได้ค่าพิกเซลใหม่ โดยมีอัลกอริทึมในการคำนวณ ให้ได้ค่าใหม่จากเนเบอร์ฮูดพิกเซล วิธีการ คือ เรียงค่าพิกเซลจากน้อยไปมาก แล้วเอาค่าตรงกลางขึ้นมาเป็นพิกเซลผลลัพธ์ รูป 2.13 แสดงถึง อัลกอริทึมการใช้ค่ากลาง และ รูปที่ 2.12 แสดงถึงรูปที่ใช้วิธีการใช้ค่ากลาง

(a)

Original image corrupted with random noise

(b)

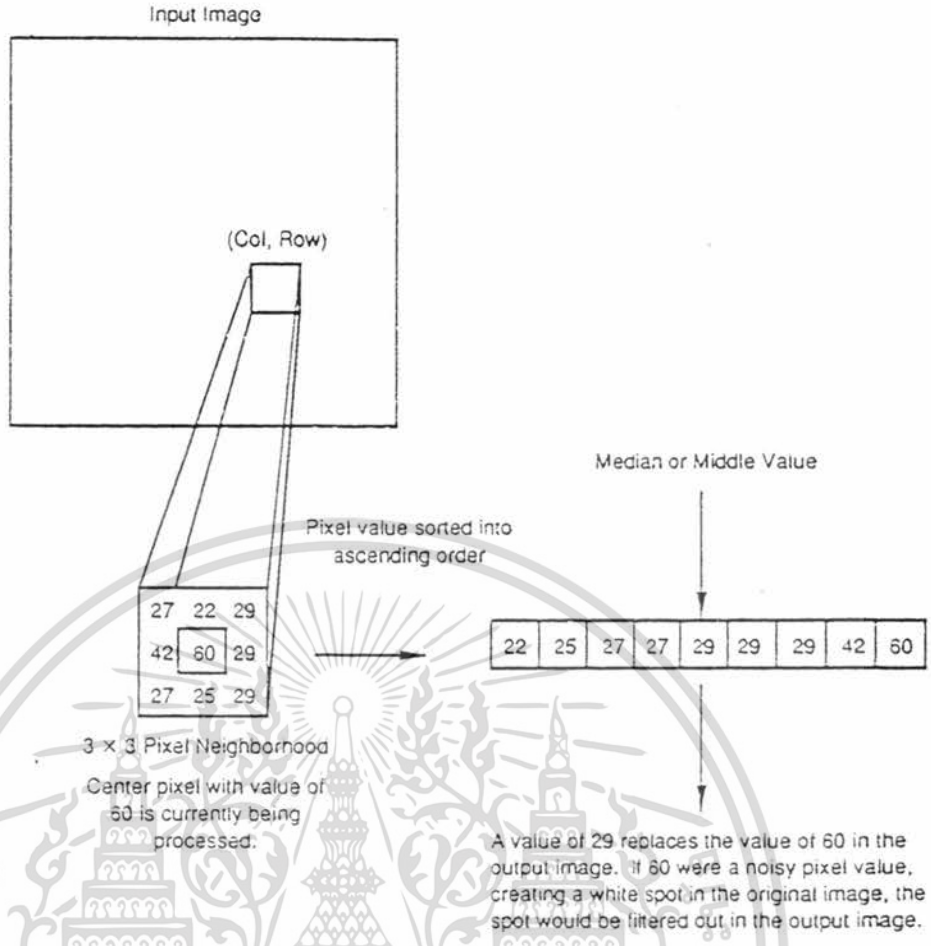
Median -filtered image. Notice absences of noise.

(c)

Median-filtered image 2nd application

รูปที่ 2.12 ตัวอย่างการใช้ค่ากลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การใช้ค่ากลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>การหมุนและการกลับภาพ (Fliping and Rotating)</u>

การกลับภาพ (Fliping) ใช้ในการเปลี่ยนรูปแบบการมองของภาพ การกลับภาพมี 2 แบบ

1. การกลับภาพตามแนวนอน (Flip Horizontal)

ทำงานโดยการสลับค่าในพิกเซลที่อยู่ในคอลัมน์ที่ 0 กับ คอลัมน์ที่อยู่ซ้ายสุด , คอลัมน์ที่ 1 กับ คอลัมน์ที่อยู่ซ้ายสุด - 1 , คอลัมน์ที่ 2 กับ คอลัมน์ที่อยู่ซ้ายสุด - 2 , ... เป็นอย่างนี้ไปเรื่อยๆจนถึงคอลัมน์ที่อยู่กลางภาพ

2 .การกลับภาพตามแนวตั้ง (Flip Vertical)

ทำงานโดยการสลับค่าในพิกเซลที่อยู่ในแถวที่ 0 กับ แถวที่อยู่ล่างสุด , แถวที่ 1 กับ คอลัมน์ที่อยู่ล่างสุด - 1 , แถวที่ 2 กับ แถวที่อยู่ล่างสุด - 2 , ... เป็นอย่างนี้ไปเรื่อยๆจนถึงแถวที่อยู่กลางภาพ

การหมุนภาพ (Rotating) มุมที่หมุนภาพจะเป็นจำนวนเท่าของ 90 องศาทั้งในทิศทางตามและทวนเข็มนาฬิกา

ทำงานโดยการสลับค่าของพิกเซลจากคอลัมน์ไปเป็นแถวและจากแถวเป็นคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและสร้างระบบ

การออกแบบระบบ

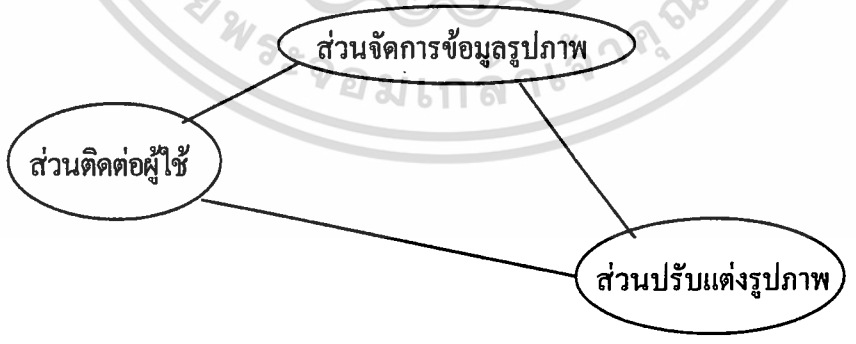
โปรแกรมประยุกต์นี้ถูกออกแบบให้ใช้กับไมโครคอมพิวเตอร์ 32 บิต และทำงานบนวินโดวส์ แบ่งการทำงานของระบบออกเป็น 3 ส่วนใหญ่ ๆ คือ ส่วนติดต่อผู้ใช้ ส่วนจัดการข้อมูลรูปภาพ และส่วนปรับแต่งรูปภาพ

ส่วนติดต่อผู้ใช้ ถูกออกแบบเป็นระบบพูลดาวน์เมนู (Pulldown menu) ฮ็อตคีย์(Hot key) ทูลบาร์ (Toolbars) และสเตตัสบาร์ (Status bar)

ส่วนจัดการข้อมูลรูปภาพ เป็นส่วนที่ทำการอ่าน ซึ่งสามารถอ่านไฟล์กราฟิคได้ 3 แบบด้วยกันคือ BMP, GIF และ PCX. เป็นส่วนที่ทำการแสดง สามารถแสดงได้หลายหน้าต่าง (MDI) และในกรณีรูปใหญ่เกินที่จะแสดงได้ในหน้าต่างนั้น ก็สามารถสกรอล์ภาพได้. อีกทั้งเป็นส่วนที่จัดเก็บรูปภาพด้วย

ส่วนปรับแต่งรูปภาพ เป็นส่วนที่นำข้อมูลรูปภาพที่อ่าน มาทำการปรับแต่งตามเทคนิคต่างๆ ที่ผู้ใช้ต้องการ

ผังการทำงานของระบบ



เครื่องมือที่ใช้ในการสร้างระบบ

ในปัญหาพิเศษนี้ คณะผู้จัดทำได้พัฒนาโปรแกรมให้ทำงานบนระบบปฏิบัติการ windows แบบ 32 บิตเท่านั้น ดังนั้นผู้จัดทำจึงได้ใช้ Visual C++ 2.0 ในการสร้างระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างระบบ

การปรับแต่งภาพมีอัลกอริทึมที่ใช้ดังนี้

อัลกอริทึมการหมุนภาพ 90 องศาในทิศทวนเข็ม

อัลกอริทึมการหมุนภาพ 90 องศาในทิศตามเข็ม

อัลกอริทึมการกลับภาพตามแนวตั้ง

อัลกอริทึมการกลับภาพตามแนวนอน

อัลกอริทึมการทำภาพเบลอ

อัลกอริทึมการทำโล-พาส สปาเชียล ฟิลเตอร์

อัลกอริทึมการทำไฮ-พาส สปาเชียล ฟิลเตอร์

อัลกอริทึมการเน้นขอบรูป

อัลกอริทึมการปรับความสว่างของภาพ

อัลกอริทึมการกลับสีของภาพ

อัลกอริทึมการทำเทรสโฮล

อัลกอริทึมการปรับความคมชัดของภาพ

อัลกอริทึมการใช้ค่ากลาง

อัลกอริทึมการหาขอบด้วยวิธีของโซเบล

แต่ละอัลกอริทึมแสดงรายละเอียดได้ดังนี้

1. อัลกอริทึมการหมุนภาพ 90 องศาในทิศทวนเข็ม

```
for (y = 0; y < Height;  y++)  {
        k = Width - 1;
        for (x  =  0; x  <  Width; x++)
                Destination[ k-- * Height + y ] = Source[ y * Width + x ];
}
```

2. อัลกอริทึมการหมุนภาพ 90 องศาในทิศตามเข็ม

```
for (y = 0; y < Height; y++)  {
        k = Height - 1;
        for (x = 0; x < Width; x++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Destination[ x * Height + (k-y) ] = Source[ y * Width + x ];
}
```

3. อัลกอริทึมการกลับภาพตามแนวตั้ง

```
for (y = 0; y < Height; y++)
        for (x = 0; x < Width; x++)
                Destination[ y * Width + (Width-x-1) ] = Source[ y * Width + x ];
```

4. อัลกอริทึมการกลับภาพตามแนวนอน

```
k = Height - 1;
for (y = 0; y < Height; y++) {
        for (x = 0; x < Width; x++)
                Destination[ k * Width + x ] = Source[ y * Width + x ];
        --k;
}
```

5. อัลกอริทึมการทำภาพเบลอ

```
for (y = 0; y < Height; y++)
        for (x = 0; x < Width; x++) {
                GetRGB(x ,y ,tmp, Source);
                r =(tmp[0].peRed + tmp[1].peRed + tmp[2].peRed +
                        tmp[3].peRed + tmp[4].peRed + tmp[5].peRed +
                        tmp[6].peRed + tmp[7].peRed + tmp[8].peRed) / 9;
                g =(tmp[0].peGreen + tmp[1].peGreen + tmp[2].peGreen +
                        tmp[3].peGreen + tmp[4].peGreen + tmp[5].peGreen +
                        tmp[6].peGreen +  tmp[7].peGreen + tmp[8].peGreen) / 9;
                b =(tmp[0].peBlue + tmp[1].peBlue + tmp[2].peBlue +
                        tmp[3].peBlue +tmp[4].peBlue + tmp[5].peBlue +          tmp
[6].peBlue + tmp[7].peBlue + tmp[8].peBlue) / 9;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                data = MatchColor(r, g, b);
                Putdata(x, y, Destination, data);
        }
```

6. อัลกอริทึมการทำโล-พาส สปาเชียล ฟิลเตอร์

```
for (y = 0; y < Height; y++)
        for (x = 0; x < Width; x++) {
                GetRGB(x ,y ,tmp, Source);
                r = (tmp[0].peRed/9) + (tmp[1].peRed/9) + (tmp[2].peRed/9) +
                        (tmp[3].peRed/9) + (tmp[4].peRed/9) + (tmp[5].peRed/9) +
                        (tmp[6].peRed/9) + (tmp[7].peRed/9) + (tmp[8].peRed/9);
                g = (tmp[0].peGreen/9) + (tmp[1].peGreen/9) + (tmp[2].peGreen/9) +
                        (tmp[3].peGreen/9) + (tmp[4].peGreen/9) + (tmp[5].
peGreen/9)
                        +(tmp[6].peGreen/9) + (tmp[7].peGreen/9) + p[8].peGreen/9);
                b = (tmp[0].peBlue/9) + (tmp[1].peBlue/9) + (tmp[2].peBlue/9) +
                        (tmp[3].peBlue/9) + (tmp[4].peBlue/9) + (tmp[5].peBlue/9) +
                        (tmp[6].peBlue/9) + (tmp[7].peBlue/9) + (tmp[8].peBlue/9);
                data = MatchColor(r, g, b);
                Putdata(x, y, Destination, data);
        }
```

7. อัลกอริทึมการทำไฮ-พาส สปาเชียล ฟิลเตอร์

```
for (y = 0; y < Height; y++)
        for (x = 0; x < Width; x++) {
          GetRGB(x ,y ,tmp, Source);
          r = (-1*tmp[0].peRed) + (-1*tmp[1].peRed) + (-1*tmp[2].peRed) +
                (-1*tmp[3].peRed) + ( 9*tmp[4].peRed) + (-1*tmp[5].peRed) +
                (-1*tmp[6].peRed) + (-1*tmp[7].peRed) + (-1*tmp[8].peRed);
          g = (-1*tmp[0].peGreen) + (-1*tmp[1].peGreen) + (-1*tmp[2].peGreen) +
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            (-1*tmp[3].peGreen) + ( 9*tmp[4].peGreen) + (-1*tmp[5].peGreen) +
            (-1*tmp[6].peGreen) + (-1*tmp[7].peGreen) + (-1*tmp[8].peGreen);
        b = (-1*tmp[0].peBlue) + (-1*tmp[1].peBlue) + (-1*tmp[2].peBlue) +
            (-1*tmp[3].peBlue) + ( 9*tmp[4].peBlue) + (-1*tmp[5].peBlue) +
            (-1*tmp[6].peBlue) + (-1*tmp[7].peBlue) + (-1*tmp[8].peBlue);
        data = MatchColor(r, g, b);
        Putdata(x, y, Destination, data);
    }
```

8. อัลกอริทึมการเน้นขอบรูป

```
for (y = 0; y < Height; y++)
    for (x = 0; x < Width; x++) {
        GetRGB(x ,y ,tmp, Source);
        r = tmp[1].peRed + tmp[3].peRed + (-4*tmp[4].peRed) +
            tmp[5].peRed + tmp[7].peRed;
        g = tmp[1].peGreen + tmp[3].peGreen + (-4*tmp[4].peGreen) +
            tmp[5].peGreen + tmp[7].peGreen;
        b = tmp[1].peBlue + tmp[3].peBlue + (-4*tmp[4].peBlue) +
            tmp[5].peBlue + tmp[7].peBlue;
        data = MatchColor(r, g, b);
        Putdata(x, y, Destination, data);
    }
```

9. อัลกอริทึมการปรับความสว่างของภาพ

```
for (i = 0; i < NumberOfColors; i++) {
    temp = Palette[i*3];
    temp += red;
    if ( temp > MAX)
        temp = MAX;
    else if (temp < MIN)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            temp = MIN;
        Palette[i*3] = temp;

        temp = Palette[i*3+1];
        temp += green;
        if ( temp > MAX)
            temp = MAX;
        else if (temp < MIN)
            temp = MIN;
        Palette[i*3+1] = temp;

        temp = Palette[i*3+2];
        temp += blue;
        if ( temp > MAX)
            temp = MAX;
        else if (temp < MIN)
            temp = MIN;
        Palette[i*3+2] = temp;
}
```

10. อัลกอริทึมการกลับสีของภาพ

```
cut = 255;
for (i = 0; i < NumberOfColors * 3; i++)
        Palette[ i ] = cut - Palette[ i ];
```

11. อัลกอริทึมการทำเทรสโฮล

```
for (i = 0; i < NumberOfColors; i++) {
        counter = 3;
        if (Palette[i*3] > Threshold)
            counter++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else    counter--;
        if (Palette[i*3+1] > Threshold)
                counter++;
        else    counter--;
        if (Palette[i*3+2] > Threshold)
                counter++;
        else    counter--;
        if (counter > 3) {
                Palette[i*3]  = 255;
                Palette[i*3+1] =  255;
                Palette[i*3+2] =  255;
        } else {
                Palette[i*3]  = 0;
                Palette[i*3+1] = 0;
                Palette[i*3+2] =  0;
        }
}
```

12. อัลกอริทึมการปรับความคมชัดของภาพ

```
top_cut = 255 - cut;
bottom_cut = cut;
for (i = 0; i < NumberOfColors * 3; i++)
        if (Palette[ i ] > top_cut)
                Palette[ i ] = 255;
        else if (Palette[ i ] < bottom_cut)
                Palette[ i ] = 0;
        else
                Palette[i] = ((top_cut - bottom_cut) * Palette[i]/255) + bottom_cut;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. อัลกอริทึมการใช้ค่ากลาง

```
for (y = 0; y < Height; y++)
        for (x = 0; x < Width; x++) {
                GetPixel(x ,y ,tmp, Source);
                data = GetMedian( tmp );
                Putdata(x, y, Destination, data);
        }
```

14. อัลกอริทึมการหาขอบด้วยวิธีของโซเบล

```
for (y = 0; y < Height; y++)
        for (x = 0; x < Width; x++) {
                GetRGB(x ,y ,tmp, Source);
                r1 = (tmp[1].peRed + tmp[2].peRed + tmp[5].peRed)/3;
                r2 = (tmp[3].peRed + tmp[6].peRed + tmp[7].peRed)/3;
                rTmp[ 0 ] = abs(r1-r2);
                g1 = (tmp[1].peGreen + tmp[2].peGreen + tmp[5].peGreen)/3;
                g2 = (tmp[3].peGreen + tmp[6].peGreen + tmp[7].peGreen)/3;
                gTmp[ 0 ] = abs(g1-g2);
                b1 = (tmp[1].peBlue + tmp[2].peBlue + tmp[5].peBlue)/3;
                b2 = (tmp[3].peBlue + tmp[6].peBlue + tmp[7].peBlue)/3;
                bTmp[ 0 ] = abs(b1-b2);

                r1 = (tmp[2].peRed + tmp[5].peRed + tmp[8].peRed)/3;
                r2 = (tmp[0].peRed + tmp[3].peRed + tmp[6].peRed)/3;
                rTmp[ 1 ] = abs(r1-r2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
g1 = (tmp[2].peGreen + tmp[5].peGreen + tmp[8].peGreen)/3;
g2 = (tmp[0].peGreen + tmp[3].peGreen + tmp[6].peGreen)/3;
gTmp[ 1 ] = abs(g1-g2);
b1 = (tmp[2].peBlue + tmp[5].peBlue + tmp[8].peBlue)/3;
b2 = (tmp[0].peBlue + tmp[3].peBlue + tmp[6].peBlue)/3;
bTmp[ 1 ] = abs(b1-b2);
r1 = (tmp[0].peRed + tmp[1].peRed + tmp[3].peRed)/3;
r2 = (tmp[5].peRed + tmp[7].peRed + tmp[8].peRed)/3;
rTmp[ 2 ] = abs(r1-r2);
g1 = (tmp[0].peGreen + tmp[1].peGreen + tmp[3].peGreen)/3;
g2 = (tmp[5].peGreen + tmp[7].peGreen + tmp[8].peGreen)/3;
gTmp[ 2 ] = abs(g1-g2);
b1 = (tmp[0].peBlue + tmp[1].peBlue + tmp[3].peBlue)/3;
b2 = (tmp[5].peBlue + tmp[7].peBlue + tmp[8].peBlue)/3;
bTmp[ 2 ] = abs(b1-b2);

r1 = (tmp[0].peRed + tmp[1].peRed + tmp[2].peRed)/3;
r2 = (tmp[6].peRed + tmp[7].peRed + tmp[8].peRed)/3;
rTmp[ 3 ] = abs(r1-r2);
g1 = (tmp[0].peGreen + tmp[1].peGreen + tmp[2].peGreen)/3;
g2 = (tmp[6].peGreen + tmp[7].peGreen + tmp[8].peGreen)/3;
gTmp[ 3 ] = abs(g1-g2);
b1 = (tmp[0].peBlue + tmp[1].peBlue + tmp[2].peBlue)/3;
b2 = (tmp[6].peBlue + tmp[7].peBlue + tmp[8].peBlue)/3;
bTmp[ 3 ] = abs(b1-b2);


r1 = 0; g1 = 0; b1 = 0;
for(j = 0; j < 4; j++) {
        if( rTmp[ j ] > r1 )  r1 = rTmp[ j ];
        if( gTmp[ j ] > g1 )  g1 = gTmp[ j ];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if( bTmp[ j ] > b1 )  b1 = bTmp[ j ];
            }
            data = MatchColor(r1, g1, b1);
            Putdata(x, y, Destination, data);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทำงานและข้อจำกัดของระบบ

ข้อจำกัดการใช้งานของระบบ

1. ฮาร์ดแวรที่ใช้ ระบบนี้ได้ขั้นต่ำต้องเป็นรุ่น 386 หน่วยความจำ 4 เมกะไบต์จอวีจี-เอ แต่ทางผู้จัดทำแนะนำให้ใช้ CPU 486 ขึ้นไปและหน่วยความจำ 8 เมกะไบต์ขึ้นไป
2. ระบบปฏิบัติการแบบ 32 บิตแบบวินโดวส์ เช่น Windows 95, Windows NT เป็นต้น
3. ไฟล์รูปภาพแบบ BMP, GIF หรือ PCX ที่นำมาทำการประมวลผลต้องเป็นแบบ 16 หรือ 256 สีเท่านั้น

ผลการทำงานของระบบ

สามารถแสดงรูป ปรับแต่งรูป และทำการจัดเก็บรูปได้ จากการทดลองพบว่า

1. การตกแต่งภาพด้วยคอมพิวเตอร์นี้ จะทำงานได้ช้าเนื่องจากมีการคำนวณทางคณิตศาสตร์มาก
2. การทำงานของระบบมีการคำนวณตัวเลขทางคณิตศาสตร์สูง โดยเฉพาะการประมวลผลภาพแบบบริเวณ จึงต้องใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูง จึงจะทำให้ระบบทำงานได้อย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปปัญหาและข้อเสนอแนะ

สรุปผลการทำปัญหาพิเศษ

ปัญหาพิเศษนี้มีจุดมุ่งหมายเพื่อจัดสร้างโปรแกรมจัดการรูปภาพในรูปแบบของซอฟแวร์ ซึ่งหลังจากได้ออกแบบระบบและโปรแกรมลงบนไมโครคอมพิวเตอร์ระดับ 32 บิตแล้ว พบว่าสามารถใช้งานได้ดี โดยภาพที่ได้จากการตกแต่งจะมีลักษณะเปลี่ยนไปตามเทคนิคที่เลือกใช้

ปัญหาและข้อเสนอแนะ

การโปรแกรมระบบพบปัญหาต่างๆ ดังนี้

1. อัลกอริทึมที่ใช้ในการแม็ปสี (Match color) ยังไม่ดีพอ เนื่องจากเวลาทำงานโดยเฉลี่ยของระบบ ใช้เวลาอยู่ที่ตรงนี้มากที่สุด
2. การอ่านข้อมูลของไฟล์รูปภาพยังใช้เวลามาก
3. การพิมพ์ออกทางเครื่องพิมพ์ขาวดำยังไม่สามารถทำงานได้ แต่สามารถดูผลก่อนทำการพิมพ์ได้

และถ้าต้องการให้ระบบนี้มีประสิทธิภาพมากขึ้น ควรปรับปรุงและแก้ไขดังนี้

1. หาอัลกอริทึมการแม็ปสีที่รวดเร็วกว่านี้มาใช้
2. ปรับปรุงให้สามารถอ่านไฟล์ได้รวดเร็วขึ้น
3. สร้างส่วนจัดการแปลงภาพสีให้เป็นภาพแบบเกรย์สเกล (gray scale) เพื่อให้สามารถพิมพ์ออกทางเครื่องพิมพ์ขาวดำได้
4. ปรับปรุงให้สามารถเลือกแก้ไขภาพเป็นบางส่วนได้
5. ปรับปรุงให้สามารถอ่านภาพในรูปแบบอื่นนอกเหนือจากนี้
6. ปรับปรุงอัลกอริทึมในการประมวลผลให้มีความรวดเร็วขึ้น

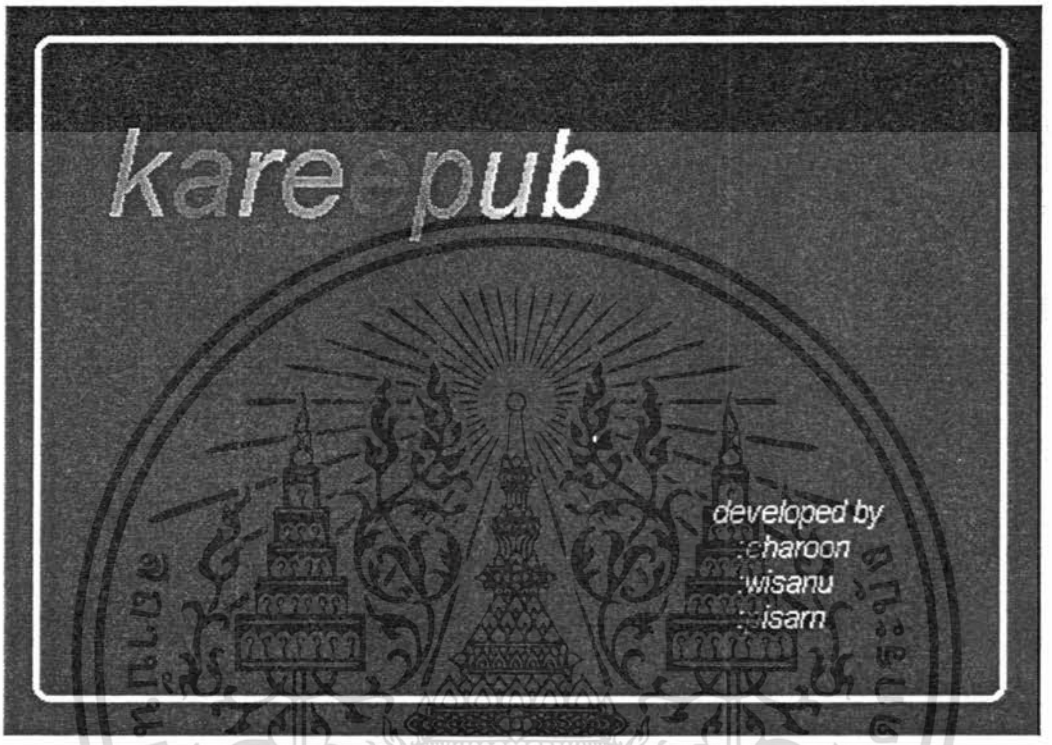
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>คู่มือการใช้งานระบบ</u>

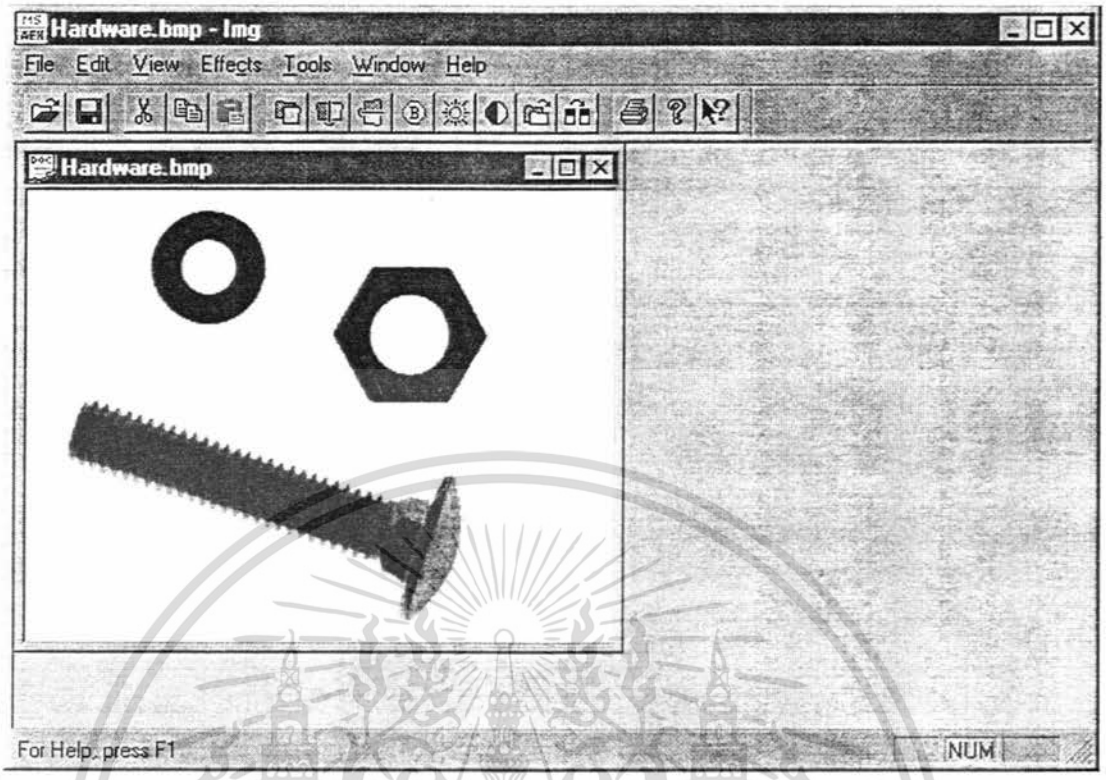
เมื่อเข้าสู่ระบบ Windows 95 หรือ Windows NT แล้วให้ทำการ RUN โดยการใช้เมาส์ดับเบิลคลิ๊ก (double click) ที่ไอคอน (icon) ของโปรแกรมก็จะปรากฏดังรูปที่ 1



รูปที่ 1 แสดงหน้าจอเมื่อเข้าสู่ระบบ

จากนั้นให้เลือก OPEN จากเมนู FILE เพื่อทำการเปิดไฟล์ที่ต้องการ ดังรูปที่ 2

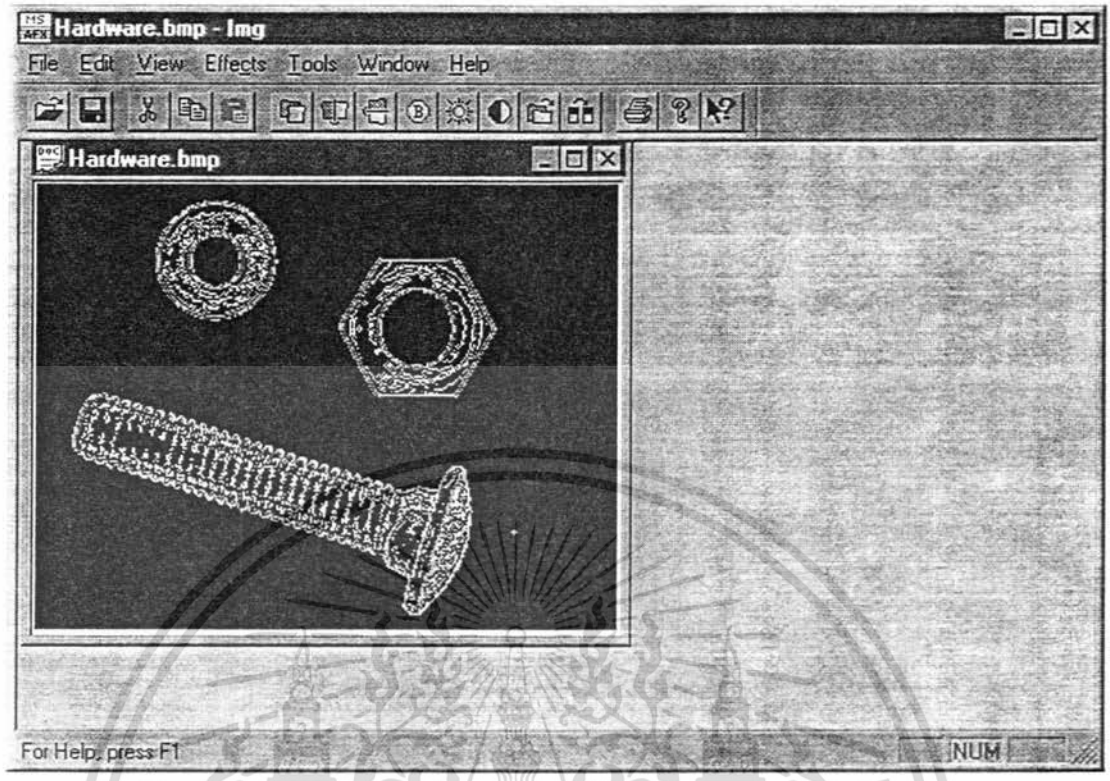
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 แสดงหน้าจอเมื่อทำการเปิดไฟล์

จากนั้นให้เลือกเทคนิคที่ต้องการประมวลผลจากเมนู EFFECTS ซึ่งจะปรากฏพูลดาวน์เมนูมาให้เลือก เช่นผู้ใช้เลือกเทคนิค Laplacian จะได้ผลการประมวลดังรูปที่ 3

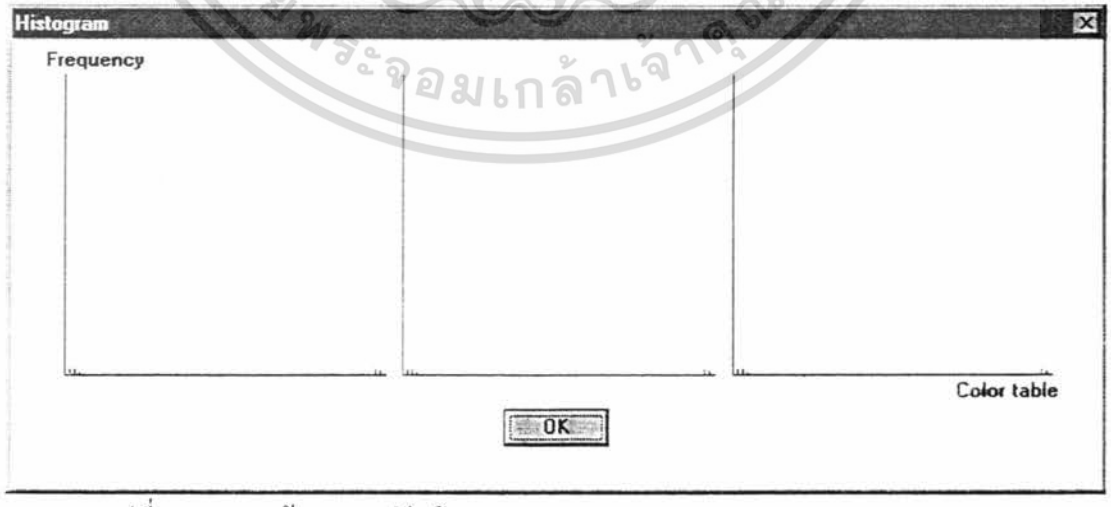
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 แสดงหน้าจอเมื่อทำการประมวลผล

หากต้องการกลับไปสู่รูปต้นฉบับให้เลือกเมนู Tools ซึ่งจะปรากฏพูลดาวน์เมนูมา ให้เลือก Original Image ก็จะได้รูปเดิมขึ้นมาอีกหนึ่งหน้าต่าง

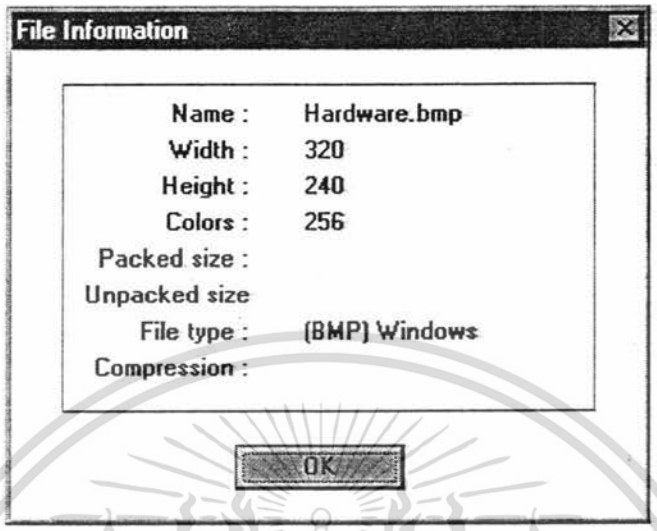
หากต้องการดูกราฟฮีสโตแกรมของรูปให้ทำการเลือก Histogram จากเมนู Tools จะได้ดังรูปที่ 4



รูปที่ 4 แสดงหน้าจอกราฟฮีสโตแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากต้องการดูรายละเอียดข้อมูลรูปภาพให้เลือก Image info จากเมนู Tools จะได้ดังรูปที่ 5



รูปที่ 5 แสดงหน้าจอรายละเอียดข้อมูลรูปภาพ

เมื่อได้ภาพที่พอใจแล้วให้ทำการเลือกเมนู Save เพื่อจัดเก็บข้อมูลในไฟล์เดิม หรือ Save As เพื่อทำการจัดเก็บในไฟล์ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- CRAIG A. LINDLEY, *Practical Image Processing in C*, John Wiley, New York, 1991
- DAVID J.KRUGLINSKI, *Inside Visual C++*, Second edition, Microsoft Press, Redmond, 1994
- LAMOTHE, RATCLIFF, SEMINATORE & TYLER, *Tricks of the Game Programming Gurus*, SAMS Publiching, Indianapolis, 1994
- ROBERT LAFORE, *Object Oriented Programming In Turbo C++*, The Waite Group Inc, 1991

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้