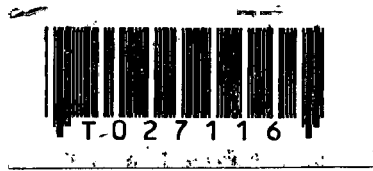


การสร้างและพัฒนาโปรแกรมการจัดฐานข้อมูลและรูปภาพทางการแพทย์
MEDICAL IMAGE AND DATABASE SOFTWARE



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน.....27116
วัน, เดือน, ปี.....11 ก.พ. 2540

พ.ศ. 2539

ISBN 974-621-603-1

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MEDICAL IMAGE AND DATABASE SOFTWARE



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF SCIENCE PROGRAM IN COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1996

ISBN 974-621-603-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การสร้างและพัฒนาโปรแกรมการจัดฐานข้อมูลและรูปภาพทางการแพทย์
นักศึกษา	นายมนัส นามะสนธิ
อาจารย์ผู้ควบคุมวิทยานิพนธ์	อาจารย์กวิน สนธิเพิ่มพูน
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2539

บทคัดย่อ

ในระบบภาพรังสีเอ็กซเรย์ของคนไข้เดิมนั้นในการค้นหาหรือการจัดเก็บข้อมูล จะต้องทำการจัดเก็บฟิล์มเอ็กซเรย์ของคนไข้พร้อมประวัติซึ่งต้องใช้พื้นที่มากในการจัดเก็บและใช้เวลาค้นหามานานซึ่งบางครั้งอาจจะหาไม่พบ ดังนั้นจึงเป็นเหตุผลทำการพัฒนาโปรแกรมข้อมูลภาพทางการแพทย์ขึ้น เนื่องจากปัจจุบันมีโปรแกรมการจัดฐานข้อมูลหลายตัวสามารถเก็บข้อมูลภาพร่วมกับข้อมูลอักขระอื่นได้ในการจัดเก็บข้อมูล จะพัฒนาโปรแกรมที่สามารถนำข้อมูลภาพมาทำการประมวลผลภาพทางดิจิทัลเพื่อเป็นเครื่องมือช่วยในการวินิจฉัยของแพทย์โดยพยายามหาวิธีที่เหมาะสมสำหรับการปรับปรุงภาพเพื่อให้ผลที่ดีที่สุด ภาพที่นำมาจัดเก็บได้มาจากกล้องถ่ายภาพทีวีถ่ายจากฟิล์มเอ็กซเรย์หรือได้จากอุปกรณ์รังสีเอ็กซเรย์โดยตรงจึงทำให้แพทย์สะดวกขึ้นในการทำงานสำหรับวิเคราะห์อาการคนไข้

Thesis Title	Medical Image and Database Software
Student	Mr. Manat Namasonthi
Thesis Advisor	Mr. Kawin Sonthipermpon
Level of Study	Master of Science Program in Computer Science and Information Technology
Department	Mathematics and Computer Science Faculty of Science King Mongkut's Institute of Technology Ladkrabang
Year	1996

ABSTRACT

In the old x-ray image of patient system was storing x-ray film and resume of patient had many area to kept and searching data had a lot of times. So in this thesis has developed the medical image and database software for storing x-ray image and other data of pateint within computer. In present it have many database software can store image and character data within database together. This thesis is developed program can using digital image processing to enhance quality of image in database as tool for diagnostic x-ray image. The x-ray image can recieved from x-ray film or x-ray camera in x-ray TV-flouroscopy directly. So doctor can easy to work and diagnostic x-ray image of patient.

กิติกรรมประกาศ

ขอขอบพระคุณอาจารย์กวิน สนธิเพิ่มพูน ที่ให้คำปรึกษาและสนับสนุนในด้านต่างๆ ตลอดจนช่วยจัดหาอุปกรณ์ที่ใช้ในการทำวิทยานิพนธ์นี้ และอาจารย์ทุกท่านที่ได้กรุณาให้คำแนะนำต่างๆจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี และขอขอบคุณบัณฑิตวิทยาลัยที่ให้ทุนสนับสนุนในการทำวิทยานิพนธ์นี้

มนัส นามะสนธิ



สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
บทที่	
1 บทนำ	
1.1 บทนำ.....	1
1.2 วัตถุประสงค์ในโครงการวิจัยและพัฒนา.....	2
2 ระบบฐานข้อมูลเดิม	
2.1 ระบบงานเดิม	4
2.2 การจัดเก็บตารางในฐานข้อมูล	5
2.3 ลักษณะการทำงานของระบบงาน	7
2.4 รูปแบบการจัดเก็บข้อมูลภาพ	10
3 พื้นฐานการประมวลผลภาพดิจิทัล	
3.1 วิธีการสเปซเขียนโดเมน	17
3.2 การทำคอนโวลูชันในสเปซเขียนโดเมน	18
3.3 การเปิดตาราง	23
4 วิธีการปรับปรุงภาพเพื่อช่วยในการวินิจฉัย	
4.1 การขยายภาพ	27
4.2 การปรับความสว่างของภาพ	30
4.3 การปรับปรุงคอนทราสต์	31
4.4 การปรับปรุงฮิสโตแกรมของภาพ	36

สารบัญ (ต่อ)

บทที่	หน้า
4.5 การทำภาพให้ราบเรียบ	42
4.6 การทำภาพให้คมชัด	44
4.7 การหาขอบภาพ	48
4.8 การหาค่าเทรสโไวค์	51
4.9 การปรับปรุงภาพด้วยวิธีฮิสโตแกรมอีควาไลเซชันแบบแยกส่วนภาพ	61
4.10 การทำอันซาร์ปมาสก์แบบปรับได้แปรตามค่าระดับเทาภาพ	62
5 วิธีการใช้งาน	65
6 บทสรุปและข้อเสนอแนะ	71
ภาคผนวก	72
ภาคผนวก ก. เอกสารอ้างอิง	73
ภาคผนวก ข. บทความที่ตีพิมพ์ในวารสาร	74
ภาคผนวก ค. โปรแกรมการประมวลผล	75
ประวัติผู้เขียน	121

สารบัญตาราง

หน้า

ตารางที่

1. ตารางเก็บประวัติส่วนตัวผู้ป่วย	5
2. ตารางเก็บรหัสและชื่อแผนกต่างๆ	5
3. ตารางการเก็บประวัติการตรวจเอ็กซเรย์ผู้ป่วย	6
4. แสดงจำนวนบิตต่อจุดภาพ biBitCount	13
5. แสดงรูปแบบการบีบอัดของ biCompression	13
6. แสดงจำนวนดัชนีสีในตารางสีของ biCirUsed	13
7. ภาระในการคำนวณโดยการทดลองใช้ตารางหน้ากากจตุรัส.....	23
ขนาดต่างๆ กับภาพต้นแบบขนาด 512x512 จุดภาพ	



สารบัญภาพ

หน้า

1. แสดงรายการทำงานระบบวินิจฉัยทางการแพทย์	7
2. แสดงการค้นหาข้อมูล	8
3. แสดงรายงานต่างๆของภาพที่ทำการตรวจ	9
4. แสดงโครงสร้างที่ใหญ่ที่สุดของภาพถ่ายในรูปบิตแมพ	10
5. แสดงข้อมูลภาพบิตแมพ 2 สี	14
6. แสดงข้อมูลภาพบิตแมพ 16 สี	15
7. แสดงข้อมูลภาพบิตแมพ 256 สี	15
8. แสดงข้อมูลภาพบิตแมพ 16.7 ล้านสี	16
9. แสดงบริเวณ 3x3 ล้อมรอบที่จุด (x,y) ในภาพ.....	17
10. แสดงฟังก์ชันการแปลงของระดับความเข้มสำหรับการปรับปรุงคอนทราส	18
11. แสดงคู่ของการทำคอนวอลูชันในสเปซเชิงโดเมน	19
12. การหาจุดข้างเคียงในบริเวณของการทำคอนวอลูชันในบริเวณกรอบภาพ	21
13. แสดงลำดับขั้นตอนวิธีการคำนวณโดยตรงกับการใช้วิธีเปิดตาราง	25
14. ภาพต้นแบบและฮิสโตแกรมของภาพ	26
15. ภาพที่ทำการขยาย	29
16. ภาพหลังปรับความสว่าง	30
17. แสดงฟังก์ชันในการดึงอย่างเชิงเส้นอย่างง่าย	31
18. แสดงการแปลงฮิสโตแกรมเดิมไปยังฮิสโตแกรมใหม่	33
19. โพลีชาร์ตลำดับขั้นตอนการดึงเชิงเส้นอย่างอัตโนมัติ	34
20. แสดงภาพและฮิสโตแกรมที่ทำการดึงอย่างอัตโนมัติ	35
21. รูปแบบฟังก์ชันแปลงของค่าระดับเทา	36
22. แสดงฟังก์ชันความหนาแน่นของระดับเทา	37
23. ตัวอย่างการใช้ฮิสโตแกรมอิกวไรเซชันด้วยระดับเทาที่ไม่เท่ากันโดยประมาณ	38
24. ตัวอย่างการแปลงฮิสโตแกรม	40
25. แสดงภาพการแปลงโดยฮิสโตแกรมอิกวไรเซชัน	41

สารบัญภาพ (ต่อ)

หน้า

26. แสดงภาพที่ทำให้เรียงด้วยวิธีหาค่าเฉลี่ย	43
27. แสดงภาพที่ปรับความคมชัด	46
28. แสดงภาพที่ทำการหาขอบภาพด้วยวิธีโซเบล	50
29. แสดงฮิสโตแกรมระดับเทา (a) เทรสโหวด์เดี่ยว (b) หลายเทรสโหวด์	51
30. ฮิสโตแกรมเทรสโหวด์	53
31. ภาพที่หาเทรสโหวด์ด้วยเอนโทรปี	59
32. แสดงวิธีหาค่าเทรสโหวด์ด้วยเอนโทรปี	60
33. แสดงภาพที่ทำการปรับปรุงภาพในส่วนปอดโดยวิธีฮิสโตแกรมอีควอไรเซชัน	61
34. รูปแบบพารามิเตอร์เน้น	62
35. ภาพที่ทำการอันซาร์ปมาสก์แบบปรับได้	63
36. จอภาพการประมวลผลภาพ	65
37. การบันทึกข้อมูลจากจอภาพลงแฟ้มข้อมูลภาพ .BMP	66
38. แสดงรายการจากเมนู 'Image'	67
39. การปรับระดับความสว่างภาพ	68
40. แสดงรายการเมนู 'Chest'	69
41. แสดงรายการเมนู 'Tool'	70

บทที่ 1

บทนำ

1. บทนำ

ในปัจจุบันการเก็บรักษาภาพถ่ายเอ็กซเรย์มีจำนวนมากตามโรงพยาบาลต่างๆรวมทั้งระบบข้อมูล ของรายละเอียดส่วนบุคคลของผู้ที่มาทำการเอ็กซเรย์มีจำนวนมากและยุ่งยากต่อการค้นหาข้อมูลของผู้ป่วยในแต่ละราย เพื่อให้แพทย์ผู้ทำการตรวจมาวินิจฉัยที่จะต้องเก็บรักษาฟิล์มเอ็กซเรย์และประวัติต่างๆ ของคนไข้ ซึ่งไม่สะดวกและล่าช้าในการทำงานของ แพทย์ที่ทำการอ่านและวินิจฉัยฟิล์มเอ็กซเรย์นั้น ตลอดจนถึงขั้นตอนและเวลาในการที่จะฉายรังสีเอ็กซเรย์ในแต่ละครั้งจะต้องผ่านขั้นตอนในการล้างภาพฟิล์มเอ็กซเรย์ดังกล่าว ดังนั้นจึงมีการคิดที่จะนำคอมพิวเตอร์มาประยุกต์ใช้รวมถึงนำทฤษฎีทางด้านการประมวลผลข้อมูลภาพ (Digital Image Processing) มาใช้ เพื่อจะลดขั้นตอนและเวลาในการทำงานให้รวดเร็วยิ่งขึ้น สำหรับอุปกรณ์ที่นำมาใช้ในการรับข้อมูลภาพนั้นจะใช้กล้องถ่ายภาพที่วีดิทัศน์จากฟิล์มเอ็กซเรย์หรือต่อโดยตรงจากกล้องถ่ายภาพที่วีดิทัศน์ที่อยู่ในระบบอุปกรณ์รังสีเอ็กซเรย์อยู่แล้วสัญญาณภาพที่ได้จากระบบอุปกรณ์รังสีเอ็กซเรย์นั้น เป็นสัญญาณในระบบ PAL ในการแปลงข้อมูลจากสัญญาณชนิดนี้เป็นสัญญาณทางดิจิทัลที่ใช้ในคอมพิวเตอร์จะต้องใช้การดิจิตัลที่สามารถแปลงข้อมูลชนิดนี้ได้เพื่อที่จะแปลงข้อมูลเป็นไบนารีและจัดรูปแบบ (format) ให้ข้อมูลภาพ เป็นชนิดบิตแมต โดยจะเก็บข้อมูลชนิดนี้ในฐานะข้อมูลร่วมกับข้อมูลอื่นสามารถเรียกดูข้อมูลเกี่ยวกับประวัติคนไข้ได้พร้อมๆ กับภาพเอ็กซเรย์นี้ สำหรับในส่วนที่จะนำมาแสดงผลภาพทางจอคอมพิวเตอร์นั้นจะนำทฤษฎีทางด้านการประมวลผลข้อมูลภาพมาใช้ในการปรับปรุงภาพ (Image Enhancement) เพื่อช่วยเหลือในการดูข้อมูลภาพที่ภาพต้นแบบมีลักษณะไม่ชัดเจนนัก ตัวอย่างสำหรับอัลกอริทึมต่างๆ ที่จะนำมาใช้นั้นเช่นการปรับปรุงภาพโดยวิธีฮิสโตแกรมอีควอลไรเซชัน (Histogram Equalization) วิธีการนี้ทำเพื่อที่จะกระจายระดับความเข้มของภาพให้มีลักษณะเท่ากัน เพื่อที่จะให้ภาพไม่ออกไปในลักษณะโทนใดโทนหนึ่งมากเกินไปจะทำให้สามารถมองเห็นรายละเอียดของภาพในส่วนที่ไม่ชัดเจนนักได้ดีขึ้น การทำภาพให้คมชัด (Image Sharpening) ใช้ในกรณีที่ข้อมูลภาพที่ได้มีลักษณะมัว (blur) จะใช้วิธีการนี้เพื่อให้ภาพคมชัดนั่นเอง การทำภาพให้ราบเรียบ (Image Smoothing) วิธีการนี้ใช้ในกรณีที่ภาพที่ได้มีสัญญาณรบกวน วิธีการนี้

เอกสารนี้เป็นเอกสารที่มอบไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จะดูขอบภาพของรูปนั้น สำหรับภาพทางการแพทย์นั้นมียู่หลายชนิดด้วยกัน เช่น ภาพอัลตราซาวด์ภาพเอ็กซเรย์สมอง ภาพเอ็กซเรย์ทรวงอก เป็นต้น กรณีของภาพเอ็กซเรย์ทรวงอกเป็นภาพที่มีรายละเอียดของภาพมาก จึงได้ทำการเสนอวิธีการปรับปรุงภาพทรวงอกโดยแยกส่วนอวัยวะของส่วนที่เป็นปกติกับอวัยวะจากหลังโดยการหาเทรสไฮวดโดยอัตโนมัติ เพื่อที่แยกการปรับปรุงภาพในแต่ละส่วน เช่นการทำฮิสโตแกรมฮิสโตแกรมโรเซชันด้วยการแยกฮิสโตแกรมในแต่ละส่วน และการปรับภาพให้คมชัดด้วยวิธีอันซาร์ปมาส์กแบบปรับได้ตามลักษณะของภาพ

โปรแกรมนี้ใช้ไมโครซอฟท์แอคเซส(Microsoft Access) ในการจัดการฐานข้อมูล ส่วนโปรแกรมที่ใช้ในการดึงข้อมูลภาพมาประมวลผลทางดิจิทัลใช้บอร์ดแลนด์ซีเวอร์ชัน 3.1 (Borland C Version 3.1) บนไมโครซอฟท์วินโดวส์ สำหรับเนื้อหาในวิทยานิพนธ์นี้จะเน้นในส่วนการนำเอาอัลกอริทึมต่าง ๆ ในการประมวลผลภาพเป็นส่วนใหญ่

1.2 วัตถุประสงค์ของโครงการวิจัย

เพื่อที่จะพัฒนาโปรแกรมที่เก็บข้อมูลภาพรังสีเอกซ์เรย์ในฐานะข้อมูล และสามารถที่จะนำเอาอัลกอริทึมการประมวลผลภาพมาช่วยในการวินิจฉัยภาพเพื่อเพิ่มประสิทธิภาพของระบบงานให้ดีกว่าเดิม อุปกรณ์ที่ใช้ในโครงงานได้แก่ คอมพิวเตอร์ รุ่น 486 หน่วยความจำ 8 เมกะไบต์ และอุปกรณ์รับภาพและแปลงข้อมูลภาพให้เป็นข้อมูลภาพ คือการ์ดแสดงผลที่สามารถแปลงสัญญาณภาพเป็นข้อมูลดิจิทัลยี่ห้อ TARGA รุ่น TARGA+ 16/32P กล้องถ่ายวีดีโอขาวดำชนิดซีซีดียี่ห้อSONY รุ่น SSC_M370E สำหรับขอบเขตของวิทยานิพนธ์ฉบับนี้ มีดังต่อไปนี้

บทที่ 1 บทนำ

บทที่ 2 เป็นการกล่าวถึงระบบฐานข้อมูลเดิมว่ามีลักษณะการทำงาน การจัดเก็บตารางในฐานข้อมูล รูปแบบการจัดเก็บข้อมูลภาพ การติดต่อเชื่อมโยงกับโปรแกรมการประมวลผลข้อมูลภาพเป็นอย่างไร

บทที่ 3 จะกล่าวความรู้พื้นฐานที่ควรทราบในการประมวลผลภาพดิจิทัล เช่นการทำงานบน
สเปเชียลโดเมนหรือโดเมนความถี่ การคำนวณทางคณิตศาสตร์โดยการทำคอนโวลูชัน
บนสเปเชียลโดเมนที่เป็นพื้นฐานของการคำนวณที่ใช้ในอัลกอริทึมต่างๆ การเพิ่ม
ความเร็วในการคำนวณผลด้วยวิธีการเปิดตาราง

บทที่ 4 กล่าวถึงอัลกอริทึมและวิธีการต่างๆ ที่ทางการประมวลผลภาพที่นำมาใช้ในการปรับปรุงภาพหรือเป็นเครื่องช่วยในการปรับปรุงภาพ เช่น การขยายภาพ การปรับความสว่างของภาพ การปรับปรุงคอนทราสต์ การปรับปรุงฮิสโตแกรมของภาพ การทำภาพให้เรียบ การทำภาพให้คมชัด การหาขอบภาพ การหาค่าเทรตไฮวด์ การปรับปรุงภาพโดยทำการแยกภาพออกเป็นสองส่วนเป็นภาพวัตถุกับฉากหลังแล้วทำการปรับปรุงภาพ แต่ละส่วนด้วยวิธีฮิสโตแกรมอีควอไลเซชัน

บทที่ 5 วิธีการใช้งาน

บทที่ 6 บทสรุปและข้อเสนอแนะจากการทำวิทยานิพนธ์นี้



บทที่ 2

ระบบฐานข้อมูลเดิม

2.1 ระบบงานเดิม

สำหรับภาพทางการแพทย์เป็นข้อมูลสำคัญที่แพทย์ใช้ประกอบวินิจฉัย ตัวอย่างภาพทางการแพทย์ที่ใช้บ่อยได้แก่ภาพจากเครื่องเอ็กซเรย์ธรรมดาเครื่องอัลตราซาวด์หรือเครื่องเอ็กซเรย์คอมพิวเตอร์ รังสีแพทย์เป็นผู้แปลผลภาพเหล่านี้เพื่อให้แพทย์ที่ส่งผู้ป่วยนำข้อมูลไปใช้โดยการถ่ายภาพลงบนแผ่น फिल्मและพิมพ์รายงานผล การเก็บภาพบนฟิล์มมีข้อเสียคือราคาแพง สิ้นเปลืองเนื้อที่ในการเก็บ และ กรณีฟิล์มหายไม่สามารถเรียกกลับมาได้ต้องนำผู้ป่วยมาทำการตรวจใหม่ นอกจากนี้การส่งข้อมูลให้แพทย์ผู้รักษาล่าช้า

ฐานข้อมูลเดิมนั้นเป็นฐานข้อมูลของโรงพยาบาลรามาริบัติที่พัฒนาขึ้นบนไมโครซอฟท์วินโดวส์ โดยใช้ซอฟต์แวร์ในการจัดการฐานข้อมูลไมโครซอฟท์แอคเซส (Microsoft Access) โปรแกรมดังกล่าวเป็นผลงานของภาควิชารังสีเอ็กซเรย์โรงพยาบาลรามาริบัติร่วมมือกับสถาบันเทคโนโลยีแห่งเอเชีย สำหรับความสามารถของโปรแกรมหาดังกล่าวมีดังนี้คือ บันทึกข้อมูลภาพเอ็กซเรย์จากอุปกรณ์รังสีเอ็กซเรย์หรือถ่ายจากฟิล์มเอ็กซเรย์ของผู้ป่วย การค้นหาข้อมูลภาพการตรวจเก่า การลบข้อมูลภาพออกฐานข้อมูล การพิมพ์รายงานผลการตรวจออกทางเครื่องพิมพ์เลเซอร์ การเก็บข้อมูลของผู้ป่วยที่ใช้นั้นจะเก็บประวัติของผู้ป่วย เช่น หมายเลขประจำตัวผู้ป่วย วันเดือนปีเกิดผู้ป่วย เพศ ในการที่ผู้ป่วยจะมาถ่ายเอ็กซเรย์นั้นจะต้องผ่านการตรวจจากแผนกอื่น ๆ ของโรงพยาบาลก่อนที่จะนำผู้ป่วยส่งมาถ่ายภาพรังสีเอ็กซเรย์ ที่แผนกรังสีวิทยา เช่นผู้ป่วยอาจจะถูกส่งมาจากแผนกอายุรกรรม ศัลยกรรม ทันตกรรมหรือ สูตินารี เป็นต้น ซึ่งจะต้องระบุอวัยวะที่ทำการตรวจว่าเป็นส่วนใด ข้อมูลที่จะต้องทราบในการตรวจแต่ละครั้งคือวันที่ทำการตรวจ ลำดับที่ในการตรวจในหนึ่งวัน จำนวนภาพที่จะต้องทำการตรวจหนึ่งคนอาจจะต้องมีการตรวจหรือถ่ายฟิล์มมากกว่าหนึ่งครั้ง ห้องที่ใช้ในการตรวจ การบันทึกจุดสำคัญในการตรวจของแพทย์และจะต้องบันทึกรายงานของ แพทย์ การจัดเก็บข้อมูลต่าง ๆ ในฐานข้อมูลนั้นได้ออกแบบไว้เป็น 3 ตาราง ตารางที่ 1จะเป็นตารางที่ใช้เก็บข้อมูลประวัติของผู้ป่วย เพื่อใช้เป็นข้อมูล อ้างอิงของผู้ป่วยจะจัดเก็บ หมายเลขประจำตัวผู้ป่วย ชื่อนามสกุลของผู้ป่วย วันเดือน ปีเกิด เพศ ตารางที่ 2 เป็นการจัดเก็บข้อมูลรายชื่อแผนกของโรงพยาบาลทั้งหมด ตารางที่ 3 จะใช้ตารางหลักในการทำงานของระบบงานนี้จะจัดเก็บประวัติการตรวจ

เอกสารประกอบคำอธิบายการทำงานของระบบงานนี้จะจัดทำเป็นเอกสารประกอบคำอธิบายการทำงานไว้ให้ดูว่ากรณีใดบ้างที่จำเป็นต้องมีการปรับปรุงแก้ไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ็กซ์เรย์ของผู้ป่วย ซึ่งภาพเอ็กซ์เรย์จะเก็บอยู่ในตารางข้อมูล สำหรับรายละเอียดของ การเก็บข้อมูลในแต่ละตารางจะกล่าวในหัวข้อต่อไป

2.2 การจัดเก็บตารางในฐานข้อมูล

ตารางที่ใช้ในระบบงานนั้นมีอยู่ 3 ตาราง ได้แก่ ตารางเก็บประวัติส่วนตัวของผู้ป่วย ตารางที่เก็บรหัสและชื่อแผนกต่างๆ ตารางเก็บประวัติการตรวจเอ็กซ์เรย์ผู้ป่วยมีรายละเอียดดังนี้

ตารางที่1

เลขที่	ชื่อ	ชนิดข้อมูล	ขนาด	คำอธิบาย
1*	HNNO	Character	8	หมายเลขประจำตัวผู้ป่วย
2	Name	Character	45	ชื่อสกุลผู้ป่วย
3	BrithDate	Date	8	วันเดือนปีเกิดผู้ป่วย
4	Sex	Numeric	1	เพศ

ตารางเก็บประวัติส่วนตัวของผู้ป่วย (Table Patient)

ตารางที่2

เลขที่	ชื่อ	ชนิดข้อมูล	ขนาด	คำอธิบาย
1*	Location	Character	3	แผนกที่ทำการส่งผู้ป่วย
2	SectionName	Character	20	ชื่อแผนก

ตารางเก็บรหัสและชื่อแผนกต่างๆ (Table Location)

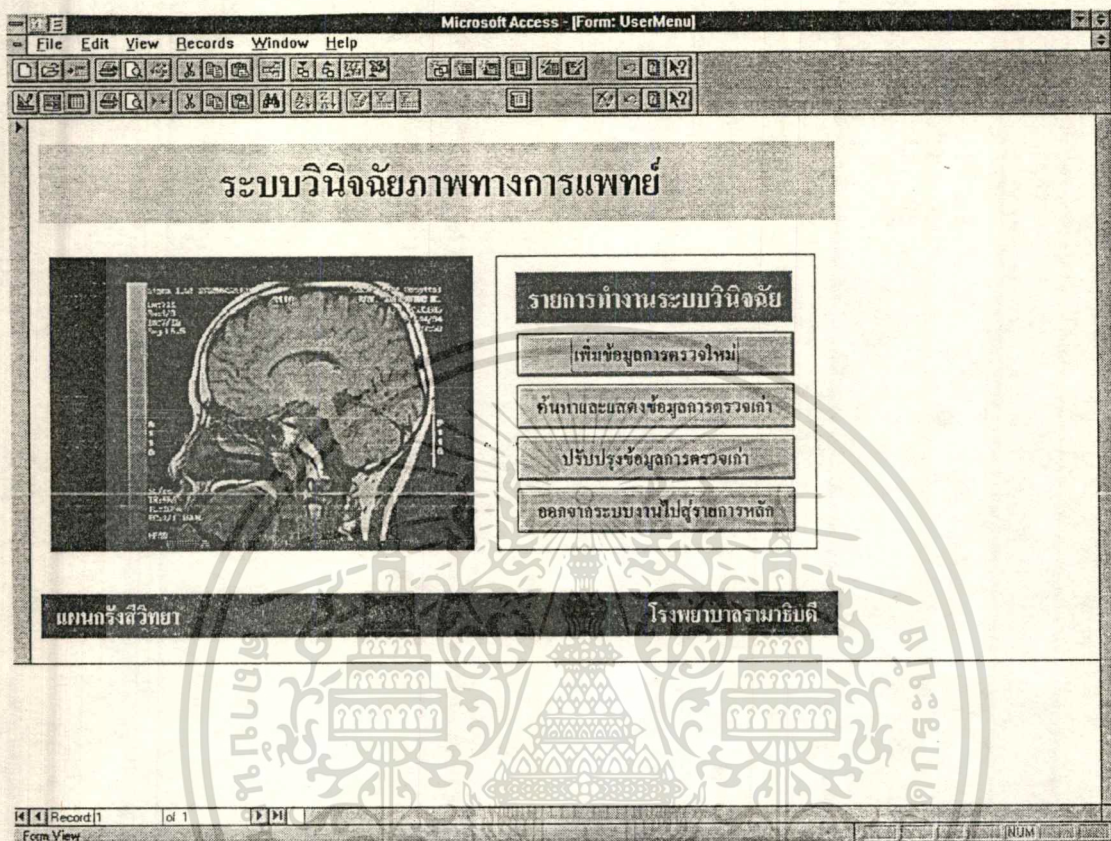
ตารางที่3

เลขที่	ชื่อ	ชนิดข้อมูล	ขนาด	คำอธิบาย
1*	HNNO	Character	8	หมายเลขประจำตัวผู้ป่วย
2*	ExamDate	Date	8	วันที่ทำการตรวจ
3*	ExamNo	Numeric	2	ลำดับที่ในการตรวจในหนึ่งวัน
4*	RunNo	Numeric	2	ลำดับที่ของรูปภาพในการตรวจ ครั้งหนึ่งๆ
5	ExamTime	Numeric	2	เวลาในการตรวจ
6	Photo	Picture	-	รูปภาพจากการตรวจ
7	Room	Character	5	ห้องที่ใช้ในการตรวจ
8	Note	Character	20	การบันทึกจุดสำคัญ
9	Report	Character	40	รายงานแพทย์
10	Location	Character	3	แผนที่ทำการส่งผู้ป่วยมาตรวจ
11	Organ	Character	20	อวัยวะที่ทำการตรวจ

หมายเหตุ * หมายถึง คอลัมน์ที่ใช้เป็นดัชนีในการค้นหาข้อมูล

ตารางการเก็บประวัติการตรวจเอ็กซเรย์ของผู้ป่วย (Table X-Rays)

ภาพที่ 1



แสดงรายการทำงานระบบวินิจฉัยทางการแพทย์

2.3 ลักษณะการทำงานของระบบ

จากภาพที่ 1 จะแสดงรายการต่างๆที่ใช้ในระบบงานได้แก่ การเพิ่มเติมการตรวจใหม่ การค้นหาข้อมูลการตรวจเก่า การปรับปรุงข้อมูล การเก็บข้อมูลจะเก็บไว้ในตารางที่ได้กล่าวไว้แล้ว ในส่วนการค้นหาข้อมูลจะใช้คอลัมน์รหัสผู้ตรวจ วันที่ตรวจ ลำดับในการตรวจเป็นดัชนีในการค้นหาข้อมูล เนื่องจากในการตรวจผู้ป่วยแต่ละครั้งจะทำการเอ็กซเรย์และเก็บภาพมากกว่าหนึ่งภาพ ดังนั้นในการแสดงผลทางจอภาพจะแสดงได้ 12 ภาพและสามารถดูรายงานแพทย์ได้ทีละหนึ่งภาพ สำหรับจอภาพรายงานแพทย์ผู้ที่ทำการตรวจ แสดงในภาพที่ 3 (a) จากจอภาพนี้ถ้าต้องการดูภาพเพียงอย่างเดียวให้เลือกปุ่มรายการ 'Tool' เมื่อแสดงภาพที่ต้องการแล้ว ถ้าต้องการปรับปรุงภาพด้วยการประมวลผลภาพทางดิจิทัลให้เลือกปุ่ม 'Process' จะเข้าสู่การทำงานที่จะกล่าวไว้ในบทที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2

Microsoft Access - [Form: Seek Data]

กรอกรหัส - วันที่ตรวจ - ลำดับที่ในการตรวจเพื่อทำการค้นหาข้อมูลที่ต้องการ

รหัสผู้ตรวจ : 00000001
 วันที่ตรวจ : 18/05/1994
 ลำดับที่ :

ชื่อ - สกุล : ศุภานดา มหาวิเศษ
 เพศ : หญิง
 วันเดือนปีเกิด : 01/01/1973
 รหัสผู้ตรวจ : 00000001

Record: 1 of 1

เลือกหัวข้อกรที่งาน

ค้นหาข้อมูล ออกรายการ

(a)

Microsoft Access - [Form: Seek Display 11]

แสดงผลข้อมูลจากการตรวจ แผนกรังสีวิทยา โรงพยาบาล สรรพสิทธิ์

1	2	3	4
Record: 1	Record: 2	Record: 3	Record: 4

TOOLS

1 2

PAGE

1

รหัสผู้ตรวจ	ชื่อ-สกุล	วันที่ตรวจ	ลำดับที่	วันเดือนปีเกิด
00000004	บพิศ นมะสมบัติ	20/10/1994	1	21/04/15

Record: 1 of 4

(b)

เอกสารแสดงการค้นหาข้อมูล (a) ค้นหาด้วยรหัส วันที่ และลำดับการตรวจ (b) แสดงข้อมูลภาพที่ได้จากการค้นหา
 ไม่ว่าการมีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3

Microsoft Access - [Form: Seek Review All]

File Edit View Records Window Help

แสดงผลการตรวจ แผนกรังสีวิทยา โรงพยาบาลรามธิบดี

รหัสตรวจ: 00000004 ชื่อ-สกุล: นพ. ปัท มานะภณ

วันที่ตรวจ: 20/10/1994 วันที่อนชีเกิด: 21/04/1970 เพศ: ชาย

ลำดับตรวจ: 1 ลำดับภาพที่: 4 คนขอ: 002 ทนักรรม

เวลาที่ตรวจ: 12:03 PM

อวัยวะ: ทรวงอก

Note:

210000

ปกติ

Tools

Record 1 of 1

(a)

Microsoft Access - [Form: Seek Review Picture]

File Edit View Records Window Help

แสดงผลการตรวจ แผนกรังสีวิทยา โรงพยาบาลรามธิบดี

Tools

Process

Record 1 of 1

(b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะการวินิจฉัยของแพทย์ ไม่ควรนำข้อมูลไปใช้ประโยชน์ด้านการค้า
 (a) แสดงรายงานต่างๆ ของภาพที่ทำการตรวจ (b) จอภาพที่เข้าสู่การประมวลผลภาพ
 ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 รูปแบบข้อมูลภาพถ่ายโดยทั่วไป

รูปแบบที่มีใช้อยู่โดยทั่วไปมีรูปแบบดังนี้คือ GIF IFF/LBM WPG BMP PIC TGA PCX และ TIFF สำหรับรูปแบบใหญ่ ๆ ที่นิยมใช้กันมากคือ PCX GIF BMP และ TIFF แต่รูปแบบหนึ่งที่ได้ได้รับความนิยมคือ BMP หรือ บิตแมพ สาเหตุก็เพราะว่าข้อมูล แบบบิตแมพนั้นไม่มีการบีบข้อมูล (COMPRESSION) จึงมีความรวดเร็วและสะดวกในการอ่านข้อมูลอย่างมากและเป็นรูปแบบข้อมูลกลางที่ใช้ในการแลกเปลี่ยนระหว่างกันของโปรแกรมประยุกต์ต่าง ๆ ใช้ในการส่งข้อมูลไปให้อุปกรณ์ต่าง ๆ เช่น อุปกรณ์การแสดงผล เป็นต้น แต่ในรูปแบบบิตแมพนี้ต้องเปลืองเนื้อที่ฮาร์ดดิส ในการจัดเก็บข้อมูลมากกว่า รูปแบบอื่นที่ใช้ในการเข้ารหัสบีบข้อมูล ซึ่งโปรแกรมเอกเซลก็ใช้รูปแบบไฟล์ชนิดนี้

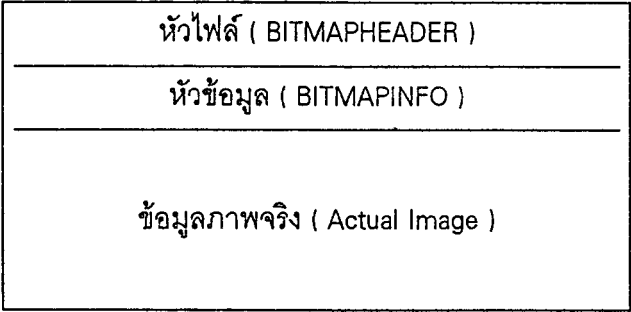
2.3.2.1 รูปแบบข้อมูลแบบบิตแมพ (BMP)

รูปแบบสำหรับใช้กับโปรแกรมระบายสีในวินโดว version 3.0 เป็นต้นมา โดยรูปแบบข้อมูลแบบนี้สามารถเป็นข้อมูลสีได้ตั้งแต่ 1 ถึง 24 บิต มีส่วนหัว(header) ในการบอกรายละเอียดต่าง ๆ ของภาพ โดยกำหนดในลักษณะโครงสร้าง (structure) ในภาษาระดับสูง ในที่นี้ใช้ภาษาซี ข้อมูลภาพในรูปแบบบิตแมพประกอบด้วยสาม ส่วนคือ

- หัวไฟล์ (BITMAPFILEHEADER)
- หัวข้อมูล (BITMAPINFO) ซึ่งรวมทั้งข้อมูลต่าง ๆ และพาเลตของสี (color palette)
- ข้อมูลภาพจริง

ในภาพที่ 4 แสดงส่วนประกอบกันของโครงสร้างที่ใหญ่ที่สุดของภาพถ่ายในรูปแบบบิตแมพ ซึ่งข้อมูลพาเลตสี และข้อมูลภาพจะเปลี่ยนโครงสร้าง ไปตามรูปแบบของจำนวนสีของภาพ และวิธีการถอดรหัส ที่ใช้ในการบีบข้อมูลภาพ ดังจะได้กล่าวต่อไป

ภาพที่ 4



2.3.2.1.1 หัวไฟล์ (BITMAPFILEHEADER)

หัวของไฟล์ (BITMAPFILEHEADER) จะประกอบด้วยข้อมูลเกี่ยวกับชนิดของ รูปแบบข้อมูล ภาพ ขนาดของภาพและโครงร่าง ดังนี้

BITMAPFILEHEADER ประกอบด้วย

- WORD bfType;
- DWORD bfSize;
- WORD bfReserved1;
- WORD bfReserved2;
- DWORD bfOffBits;

โดย bfType เป็นชนิดของรูปแบบข้อมูลภาพในที่นี้สำหรับบิตแมพคือ "BM"

bfSize เป็นขนาดของไฟล์ทั้งหมด

bfReserved1,2 ไม่ได้ใช้งาน ปกติกำหนดให้มีค่าเท่ากับศูนย์

bfOffBits เป็นขนาดของหัวไฟล์ทั้งหมดใช้เพื่อข้ามไปจุดเริ่มต้นของข้อมูลภาพจริง

2.3.2.1.2 หัวข้อมูล (BITMAPINFO)

หัวข้อมูลจะเป็นตัวกำหนดขนาดต่าง ๆ (dimensions) และข้อมูลสี (color information) ดังนี้

BITMAPINFO ประกอบด้วย

- BITMAPINFOHEADER bmiHeader;
- RGBQUAD bmiColors[1];

โดยที่ bmiHeader เป็นข้อมูลเกี่ยวกับขนาดต่าง ๆ จำนวนสี และรูปแบบของสี

bmiColors เป็นอาร์เรย์ข้อมูลสีอาร์จีบีที่เป็นตัวกำหนดสีในการแสดงผลของภาพที่เป็น รายละเอียดต่าง ๆ จะอยู่ใน BITMAPINFO HEADER ดังนี้

STRUCTURE BITMAPINFOHEADER ประกอบด้วย

-DWORD biSize;
 -DWORD biWidth;
 -DWORD biHeight;
 -WORD biPlanes;
 -WORD biBitCount;
 -DWORD biCompression;
 -DWORD biSizeImage;
 -DWORD biXPelsPerMeter;
 -DWORD biYPelsPerMeter;
 -DWORD biClrUsed;
 -DWORD biClrImportant;

โดยที่ biSize เป็นขนาดของส่วนหัวข้อมูล (BITMAPINFOHEADER) มีหน่วยเป็นไบต์

biWidth เป็นความกว้างของภาพ (มีหน่วยเป็น จุดภาพสั้น)

biHeight เป็นความสูงของภาพ (มีหน่วยเป็น เส้น)

biPlanes เป็นจำนวนหน้าของสี(color plane)สำหรับอุปกรณ์เป้าหมาย(target device)

ปกติกำหนดให้เป็น 1 เสมอ

biBitCount จำนวนบิตต่อจุดภาพ (1,4,8,24) ดูรายละเอียดได้ในตารางที่ 4

biCompression ชนิดของการบีบอัด (compression) ดูรายละเอียดได้ในตารางที่ 5

biSizeImage เป็นขนาดของภาพ (มีหน่วยเป็นไบต์)

biXPelsPerMeter ความละเอียดสำหรับอุปกรณ์เป้าหมายในแนวนอนต่อหนึ่งเมตร

biYPelsPerMeter ความละเอียดสำหรับอุปกรณ์เป้าหมายในแนวตั้งต่อหนึ่งเมตร

biClrUsed จำนวนดัชนีสี (color index) ในตารางสี (color table) มีค่าได้ 3 กรณีโดย

ดูรายละเอียดได้ในตารางที่ 6

biClrImportant จำนวนความสำคัญของดัชนีสี (color index) ในการแสดงผล ถ้าเป็น

ศูนย์ทุกสีจะมีความสำคัญเท่ากันหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4

ค่า	อธิบาย	จำนวนแอร์เรย์สี (bmiColors)
1	บิตแมพ 2 สี (monochrome bitmap)	2 แอร์เรย์
4	บิตแมพ 16 สีสูงสุด	16 แอร์เรย์
8	บิตแมพ 256 สีสูงสุด	256 แอร์เรย์
24	บิตแมพ 16.7 ล้านสีสูงสุด	NULL (ไม่มีแอร์เรย์)

แสดง จำนวนบิตต่อจุดภาพ ของ biBitCount

ตารางที่ 5

ชนิด	อธิบาย
BI_RGB	แสดงว่าข้อมูลไม่มีการบีบอัด
BI_RLE4	ทำการเข้ารหัสรันเลนส์ (run-length) ด้วยขนาด 4 บิตต่อจุดภาพ โดยในสองไบต์จะประกอบด้วยไบต์นับ(count byte) และดัชนีสีขนาด 1 ไบต์ที่แสดงผลได้ 2 จุดภาพ
BI_RLE8	ทำการเข้ารหัสรันเลนส์ (run-length) ด้วยขนาด 8 บิตต่อจุดภาพ โดยในสองไบต์จะประกอบด้วยไบต์นับ (count byte) และ ดัชนีสีขนาด 1 ไบต์ในการแสดงผล 1 จุดภาพ

แสดงรูปแบบการบีบอัด (compression formats) ของ biCompression

ตารางที่ 6

ค่า	อธิบาย
0	ใช้จำนวนของสีสูงสุดเท่ากับ 2 biBitCount
ไม่เป็นศูนย์	ถ้า biBitsCount<24 biCirUsed จะระบุ จำนวนสีที่ใช้ในการแสดงผล เช่นเมื่อ biBitsCount เท่ากับ 8 จะมีค่า biCirUsed เท่ากับ 256 สี เป็นต้น ถ้า biBitsCount=24 biCirUsed จะระบุ ขนาดของ ตารางอ้างอิง (reference table) เพื่อให้วินโดวได้รับรู้และ ใช้ในการแสดงผลพาเลตสีที่เหมาะสมที่สุด

แสดงจำนวนดัชนีสี (color index) ในตารางสี (color table) ของ biCirUsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับส่วน RGBQUAD ที่เก็บรายละเอียดของ palette สีประกอบด้วยส่วนประกอบดังนี้

STRUCTURE RGBQUAD ประกอบด้วย

BYTE rgbBlue;

BYTE rgbGreen;

BYTE rgbRed;

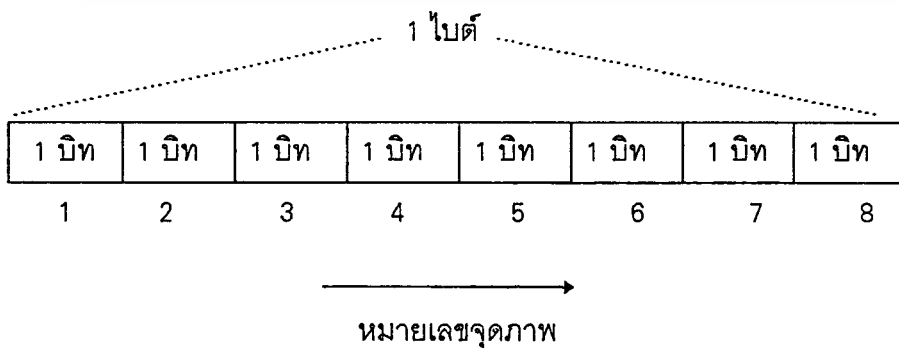
BYTE rgbReserved; ไม่ใช้งาน (ปกติตั้งให้เป็นศูนย์)

2.3.2.1.3 ข้อมูลภาพจริง (actual image information)

ส่วนของข้อมูลบิตแมพจริงจะมีการเก็บที่แตกต่างกันตามจำนวนบิตของข้อมูลภาพ (biBitcount) และลักษณะการบีบอัด biCompression แต่โดยทั่วไปแล้วข้อมูลในรูปแบบบิตแมพปกติที่นิยมในปัจจุบันในการส่งข้ามระหว่างโปรแกรมประยุกต์ต่าง ๆ นั้น จะไม่ใช้การบีบอัดข้อมูล ดังนั้น จึงแบ่งรูปแบบข้อมูลในลักษณะที่นิยมใช้กันในปัจจุบันได้เพียงสามกรณีดังนี้

กรณีที่ 1 ถ้าเป็น 1 บิตต่อจุดภาพ หมายความว่ามิตซ์นี้สี bmiColors เป็น แอร์เรย์ จำนวน 2 แอร์เรย์ คือมี 2 สี โดยข้อมูลแต่ละบิตจะแทนหนึ่งจุดภาพ ดังภาพที่ 5

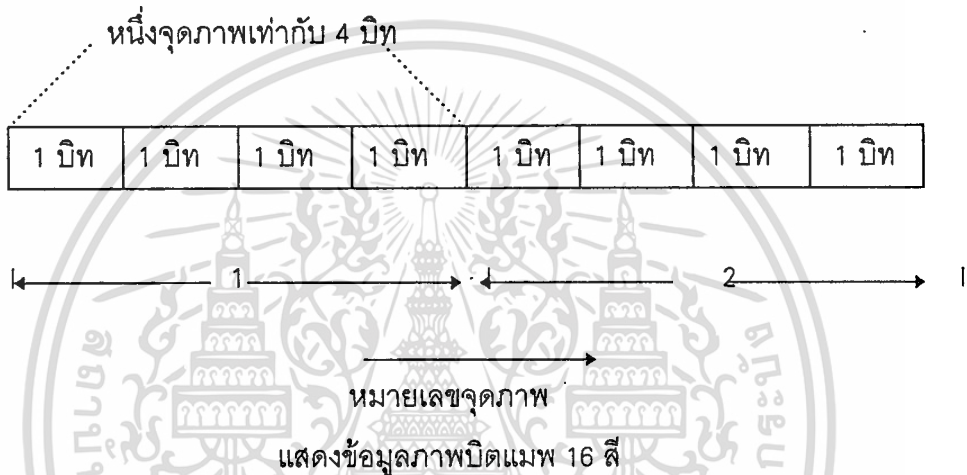
ภาพที่ 5



เอกสารนี้เป็นเอกสารต้นฉบับของโครงการวิจัย (monochrome Bitmap) ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

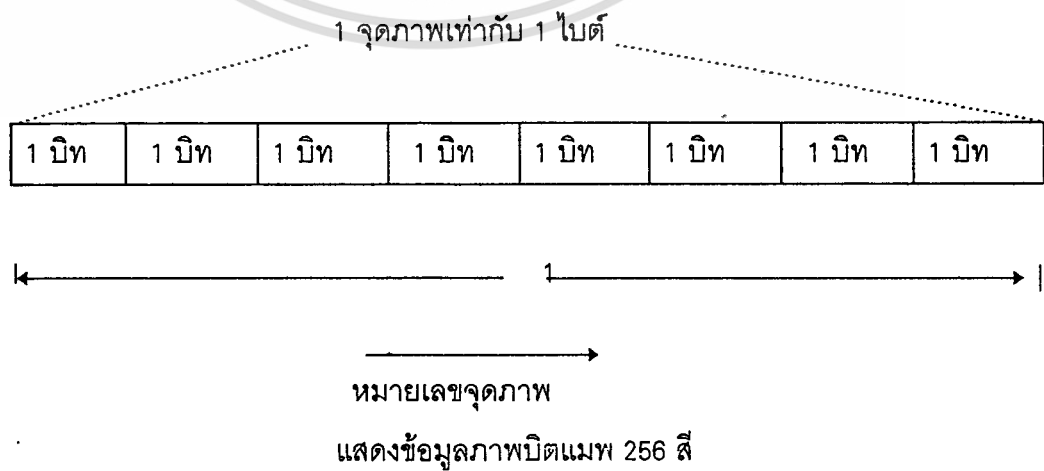
กรณีที่ 2 ถ้าเป็น 4 บิตต่อจุดภาพ หมายความว่ามิตซ์นีสี bimiColors ที่เป็น แอร์เรย์
จำนวน 16 แอร์เรย์ คือมี 16 สี โดยข้อมูลแต่ละ 4 บิต จะแทนหนึ่งจุดภาพ ดังภาพที่ 6

ภาพที่ 6



กรณีที่ 3 ถ้าเป็น 8 บิตต่อจุดภาพ หมายความว่ามิตซ์นีสี bmiColors ที่เป็นแอร์เรย์
จำนวน 256 แอร์เรย์ คือมี 256 สี โดยข้อมูลแต่ละ 8 บิตจะแทนหนึ่งจุดภาพ ดังภาพที่ 7

ภาพที่ 7



กรณีนี้ที่ 4 ถ้าเป็น 24 บิตต่อจุดภาพ หมายความว่าจะสามารถแสดงสีได้พร้อม ๆ กันเท่ากับ 2^{24} คือ 16.7 ล้านสี คือสามารถผสมสีได้โดยตรง ไม่ต้องมีการใช้ดัชนีสี ดังนั้นในกรณีนี้ bmiColors = NULL ข้อมูลแต่ละ 3 ไบต์ที่เรียงกันจะแสดง relative intensity ของสีฟ้า สีเขียว และสีแดง (RGB) ตามลำดับ โดยข้อมูลแต่ละ 24 บิตจะ แทนหนึ่งจุดภาพ ดังภาพที่ 8

ภาพที่ 8

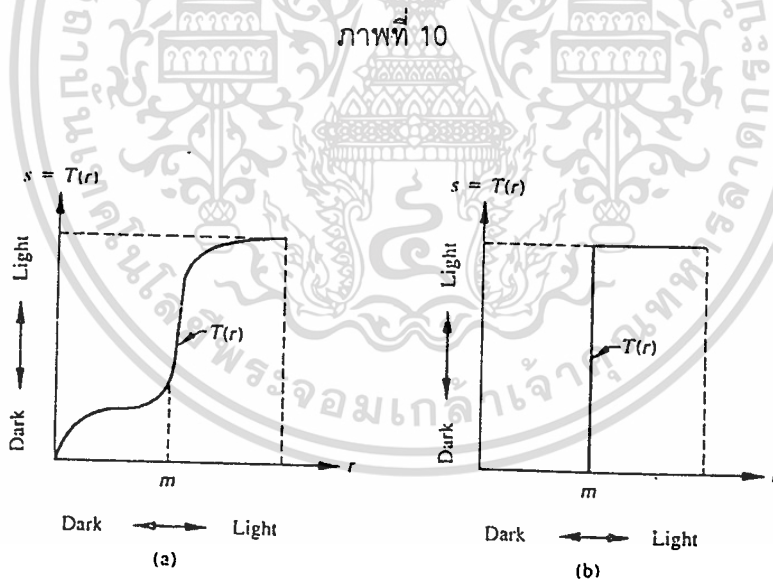


ภาพที่ใช้เก็บในฐานข้อมูลเป็นภาพขนาด 8 บิตต่อจุดภาพ เป็นภาพระดับเทา(Gray Scale)ขาวดำ 256 ระดับ การผสมสีในตารางเทียบสี ค่าสีแดง(R),ค่าสีเขียว(G),ค่าสีน้ำเงิน(B) จะมีค่าเท่ากันในทุกะดับตารางเทียบสี

แบบที่ง่ายที่สุดของตัวปฏิบัติการ T มีบริเวณ 1×1 ในกรณีที่ g จะขึ้นอยู่กับค่าของ f ที่ (x,y) เท่านั้นและ T จะกลายเป็นฟังก์ชันแปลง (Transformation Function) หรือฟังก์ชันการแมป (Mapping Function) ซึ่งอยู่ในรูป

$$s = T(r)$$

เมื่อ r และ s คือตัวแปรที่ใช้แทนค่าระดับความเข้มของ $f(x,y)$ และ $g(x,y)$ ที่จุด (x,y) ถ้า $T(r)$ มีรูปแบบในภาพที่ 10 (a) ผลของการปรับปรุงภาพจะทำให้ภาพที่มีระดับความเข้มต่ำกว่า m ถูกบีบให้อยู่ในช่วงแคบ ๆ ของ s และจะมีผลตรงกันข้ามกันในช่วงที่มีระดับความเข้มสูงกว่า m เทคนิคนี้เราเรียกว่าการปรับค่าคอนทราสต์ (Contrast Stretching) และถ้า $T(r)$ มีรูปแบบในภาพที่ 10 (b) ผลของการทำก็จะทำให้ภาพที่ใช้ฟังก์ชันนี้มีระดับความเข้ม 2 ระดับ



แสดงฟังก์ชันการแปลงของระดับความเข้มสำหรับการปรับปรุงค่าคอนทราสต์
(Contrast Enhancement)

3.2 การทำคอนโวลูชันในสเปซเอ็นโดเมน

การทำคอนโวลูชันเป็นการทำงานในรูปแบบที่เรียกว่าเป็นการทำงานที่ต้องใช้จุดข้างเคียง (Neighbourhood Operation) มาพิจารณาในการคำนวณการกระทำกับจุดข้างเคียงนั้นผลไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลัทธิได้ค่าออกมา ณ ตำแหน่งเดียวกันกับอินพุต และใช้จุดข้างเคียงที่อยู่รอบ ๆ ซึ่งการทำคอนโวลูชันเป็นการคำนวณผลรวมของผลคูณ (sum of product) ภาพที่ 11 แสดงหน้ากากของคอนโวลูชัน (convolution mask) และจุดข้างเคียงจากภาพ สมาชิกแต่ละตัวในหน้ากากเรียกว่า น้ำหนัก (weight) ค่าน้ำหนักในหน้ากากเป็นตัวคำนวณผลของการทำคอนโวลูชัน ซึ่งจะกำหนดลักษณะการกรองตามการประยุกต์ใช้ในงานต่าง ๆ

ภาพที่ 11

$$\begin{matrix} M(-1,-1) & M(0,-1) & M(1,-1) \\ M(-1, 0) & M(0, 0) & M(1, 0) \\ M(-1, 1) & M(0, 1) & M(1, 1) \end{matrix}$$

(a) หน้ากากของการทำคอนโวลูชันขนาด 3x3

$$\begin{matrix} F(i-1,j-1) & F(i,j-1) & F(i+1,j-1) \\ F(i-1,j) & F(i,j) & F(i+1,j) \\ F(i-1,j+1) & F(i,j+1) & F(i+1,j+1) \end{matrix}$$

(b) จุดข้างเคียง (neighbourhood) ต่าง ๆ ในบริเวณเมตริกซ์
ขนาด 3x3 ของภาพต้นแบบ

แสดงคู่ของการทำคอนโวลูชันในสเปเชียลโดเมน

ตัวชี้ในหน้ากากมีจุดเริ่มต้นที่จุดศูนย์กลาง โดยเริ่มที่ตำแหน่งมุมบนด้านซ้าย ของภาพ สามารถเขียนสมการคอนโวลูชันได้ดังนี้

$$C(i,j) = \sum_{k=-1}^{i+1} \sum_{l=-1}^{j+1} F(k,l)M(i-k,k-l) \quad (3.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละจุดภาพจะเป็นการวนเอามาจากจุดรอบข้างและคูณด้วยค่าที่ตรงตำแหน่งเดียวกันกับหน้ากาก นำผลคูณของแต่ละตัวนั้นมารวมกันก็จะได้ ค่าจุดภาพผลลัพธ์ ซึ่งการคำนวณที่ทำนั้นสามารถกระจายแสดงออกมาได้ดังนี้

$$\begin{aligned}
 C(i,j) = & F(i-1,j-1)M(1,1)+F(i,j-1)M(0,1)+F(i+1,j-1)M(-1,1)+ \\
 & F(i-1,j)M(1,0)+F(i,j)M(0,0)+F(i+1,j)M(-1,0)+ \\
 & F(i-1,i+1)M(1,-1)+F(i,j+1)M(0,-1)+F(i+1,j+1)M(-1,-1)
 \end{aligned} \tag{3.2}$$

จากสมการข้างบนแสดงการจัดเรียงเทอมต่าง ๆ โดยลำดับในหน้ากากมีลำดับ จากบนซ้ายไปยังด้านล่างขวา สำหรับค่าตำแหน่งที่ตรงกันในหน้ากากจะมีลำดับตรงกันข้าม คือเริ่มจากด้านล่างขวาขึ้นไปยังด้านบนซ้าย สำหรับสิ่งที่น่าสนใจของการตรงกันข้ามของ ลำดับนี้ก็คือว่าถ้าหมุนหน้ากาก 180 องศา ลำดับของเทอมภาพกับเทอมหน้ากาก จะมีลำดับในทิศทางเดียวกันโดยได้การ กระจายดังนี้

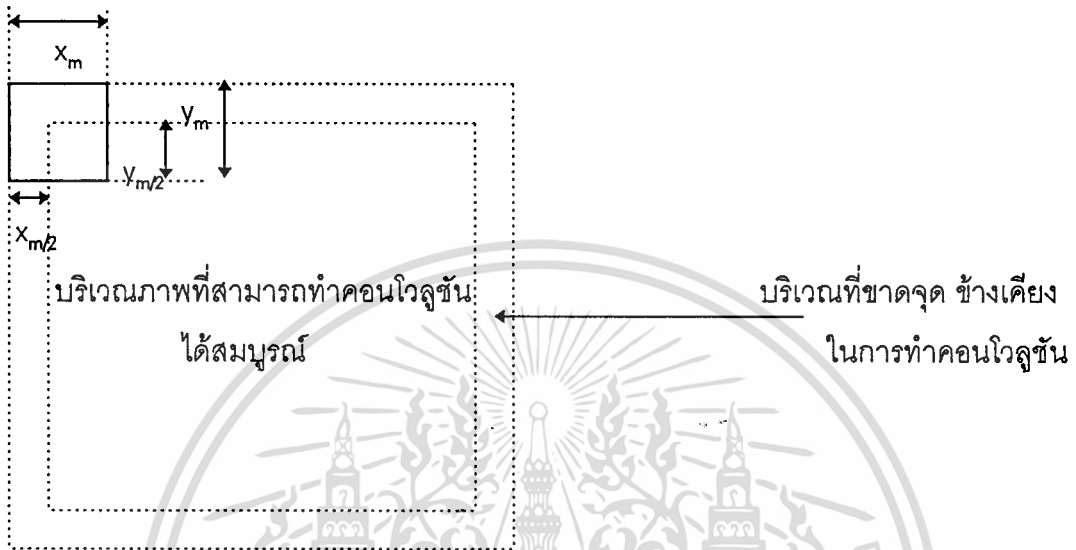
$$\begin{aligned}
 C(i,j) = & F(i-1,j-1)M(-1,-1)+F(i,j-1)M(0,-1)+F(i+1,j-1)M(1,-1)+ \\
 & F(i-1,j)M(-1,0)+F(i,j)M(0,0)+F(i+1,j)M(1,0)+ \\
 & F(i-1,i+1)M(-1,-1)+F(i,j+1)M(0,1)+F(i+1,j+1)M(1,1)
 \end{aligned} \tag{3.3}$$

ในการประยุกต์ใช้งานบางอย่างนิยมใช้การหมุนหน้ากาก 180 องศา นี้เพื่อให้่ายต่อการกำหนดตัวชี้ให้ภาพและหน้ากากมีตัวชี้ที่เหมือนกัน ในการประยุกต์บางอย่างจะมีการจัดเรียงสมาชิกของหน้ากากอย่างอัตโนมัติ โดยมีตำแหน่งของหน้ากากและภาพที่สอดคล้องกัน อีกทั้งหน้ากากก็ไม่จำเป็นต้องมีจำนวนแนวตั้งเท่ากับจำนวนแนวนอนเสมอไปด้วย แต่โดยส่วนใหญ่แล้วหน้ากากมักจะสมมาตรกัน (symmetric) และโดยทั่วไปความกว้างและความสูงจะเป็นจำนวนคี่ เพื่อให้จุดศูนย์กลางของหน้ากากเป็นจุดศูนย์กลางได้

- การขาดจุดข้างเคียงในบริเวณของการทำคอนโวลูชันในบริเวณกรอบภาพ

ในขบวนการคำนวณแบบใช้บริเวณโดยรอบของจุดภาพนี้เป็นไปไม่ได้ที่จะทำคอนโวลูชันในบริเวณขอบรอบนอกของภาพ (บริเวณกรอบภาพ) เพราะในการคำนวณจะขาดจุดภาพที่อยู่เอกสารภายนอกอาณาบริเวณของภาพ ดังแสดงในภาพที่ 12 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 12



แสดงการขาดจุดข้างเคียงในบริเวณของการทำคอนโวลูชันในบริเวณกรอบภาพ

ดังนั้นโดยทั่วไปแล้ววิธีการที่จะแก้ปัญหานี้ได้ ก็มักใช้วิธีการกำหนดแถบจุดภาพในบริเวณกรอบของภาพทั้ง 4 ด้านให้เป็นค่าเดิม หรือเป็นศูนย์แล้วแต่กรณี ความกว้างของแถบบนและด้านล่างของภาพมีค่าเท่ากับ $y_{m/2}$ และด้านซ้ายและขวาเท่ากับ $x_{m/2}$ เมื่อ x_m คือความกว้างของหน้าภาพ และ y_m คือความสูงของหน้าภาพตัวอย่างเช่น หน้าภาพขนาด 3x3 จะมีแถบที่มีความกว้าง 1 จุดภาพซึ่งไม่สามารถคำนวณได้รอบ ๆ ภาพ

ในการใช้งานจะต้องรู้ว่าจะนำภาพที่ได้ไปทำอะไรต่อไป เพราะอาจจะเป็นไปได้ที่ภาพผลลัพธ์อาจจะมีขนาดเล็กกว่าภาพต้นแบบ ตามจำนวนแถว และคอลัมน์ที่ไม่สามารถทำการคำนวณได้ ถ้าเงื่อนไขไม่เป็นเช่นนั้นและต้องการรักษาขนาดของภาพผลลัพธ์ ให้เท่ากับภาพต้นแบบก็ใช้การใส่ค่าเดิมหรือศูนย์แล้วแต่กรณี ลงไปในกรอบที่ไม่สามารถคำนวณได้ ซึ่งวิธีนี้เป็นที่นิยมใช้กันอย่างมากเพราะภาพผลลัพธ์สามารถนำไปเปรียบเทียบกับภาพต้นแบบได้ โดยตรงและในอนาคตก็สามารถนำไปคำนวณกันต่อไปได้ในขณะที่ยังรักษาแนวของจุดภาพให้ตรงกันอยู่อย่างนั้น

บางระบบต้องการการคำนวณที่สมบูรณ์แบบอย่างแท้จริงโดยการใช้ข้อมูลในบริเวณอื่น

ๆ มาใช้แทนบริเวณแถวและคอลัมน์ของจุดภาพที่อยู่เลยออกไปจากภาพต้นแบบ วิธีการหนึ่งนั้นไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำได้โดยใช้วิธีของทางด้านล่างของภาพมาใช้เป็นข้อมูลที่ต้องการ ของการคำนวณทางด้านบน และในทำนองเดียวกัน โดยใช้แถวของทางด้านบนของภาพมาใช้เป็นข้อมูลที่ต้องการของการคำนวณทางด้านล่าง ในลักษณะที่คล้ายกันของคอลัมน์ทางด้านขวาของภาพก็นำมาใช้เป็นข้อมูลที่ต้องการของการคำนวณทางด้านซ้ายเสร็จสมบูรณ์ และคอลัมน์ทางด้านซ้ายของภาพก็นำมาใช้เป็นข้อมูลที่ต้องการของการคำนวณทางด้านขวา เช่นกัน ถึงแม้ว่าวิธีการเช่นนี้ไม่เป็นการเหมาะสมนักก็ตาม แต่เพื่อให้ผลลัพธ์มีขนาดภาพเท่ากับภาพต้นแบบ และในการประยุกต์ใช้งานบางอย่าง ต้องหลีกเลี่ยงการเกิดการเพี้ยนอย่างรุนแรงของวิธีการคำนวณจากการแทนกรอบที่คำนวณไม่ได้ด้วยศูนย์เช่นในกรณีทำเกรเดียนท์เป็นต้น

บางระบบสามารถใช้วิธีแทนบริเวณกรอบที่คำนวณไม่ได้นั้น ด้วยกรอบของภาพต้นแบบในตำแหน่งที่ตรงกันได้อย่างทันที ถ้าหากเป็นการประยุกต์ที่ผลลัพธ์ค่อนข้างคล้ายกับภาพต้นแบบอย่างมากเช่นการทำให้เรียบ (smoothing) เป็นต้น ถ้าไม่มีสัญญาณรบกวนอยู่ในภาพต้นแบบในส่วนของกรอบก็จะไม่สามารถสังเกตข้อแตกต่างได้เลย

- เวลาที่ใช้

การทำคอนโวลูชันค่อนข้างจะใช้เวลาการคำนวณมากครั้ง เช่นถ้าเป็นหน้ากาก 3×3 ในการที่จะได้ผลลัพธ์ 1 จุดภาพจะต้องใช้การคูณ 9 ครั้ง การบวก 9 ครั้ง และการหารอีก 1 ครั้ง ถ้ามีการทำอิมพลไลซ์พิจารณาการทำคอนโวลูชันของหน้ากาก ที่มีไว้เท่ากับคอลัมน์แล้วจำนวนการบวกจะเท่ากับจำนวนการคูณ ตารางที่ 3.1 แสดงจำนวนครั้งที่ต้องใช้ในการคำนวณ โดยทดลองใช้หน้ากากขนาดต่าง ๆ โดยใช้ภาพต้นแบบขนาด 512×512 โดยสมมุติให้สามารถทำคอนโวลูชันที่บริเวณกรอบของภาพได้ ตัวอย่างเมื่อใช้หน้ากากขนาด 3×3 ต้องใช้จำนวนครั้งในการคำนวณเท่ากับ $2,359,296 \times 2 + 262,144 = 4,980,736$ ครั้งเป็นต้น

ภาพที่เป็นการแปลง (transformation) ของค่าระดับสีเทาแต่ละจุดภาพไปยังค่าระดับสีเทาใหม่ สำหรับการปรับปรุงภาพอย่างง่ายแล้ว ฟังก์ชันการแปลงจะเป็นฟังก์ชันเดียวกันสำหรับ ทุก ๆ จุดในภาพดังนั้นอาจเขียนได้ว่า

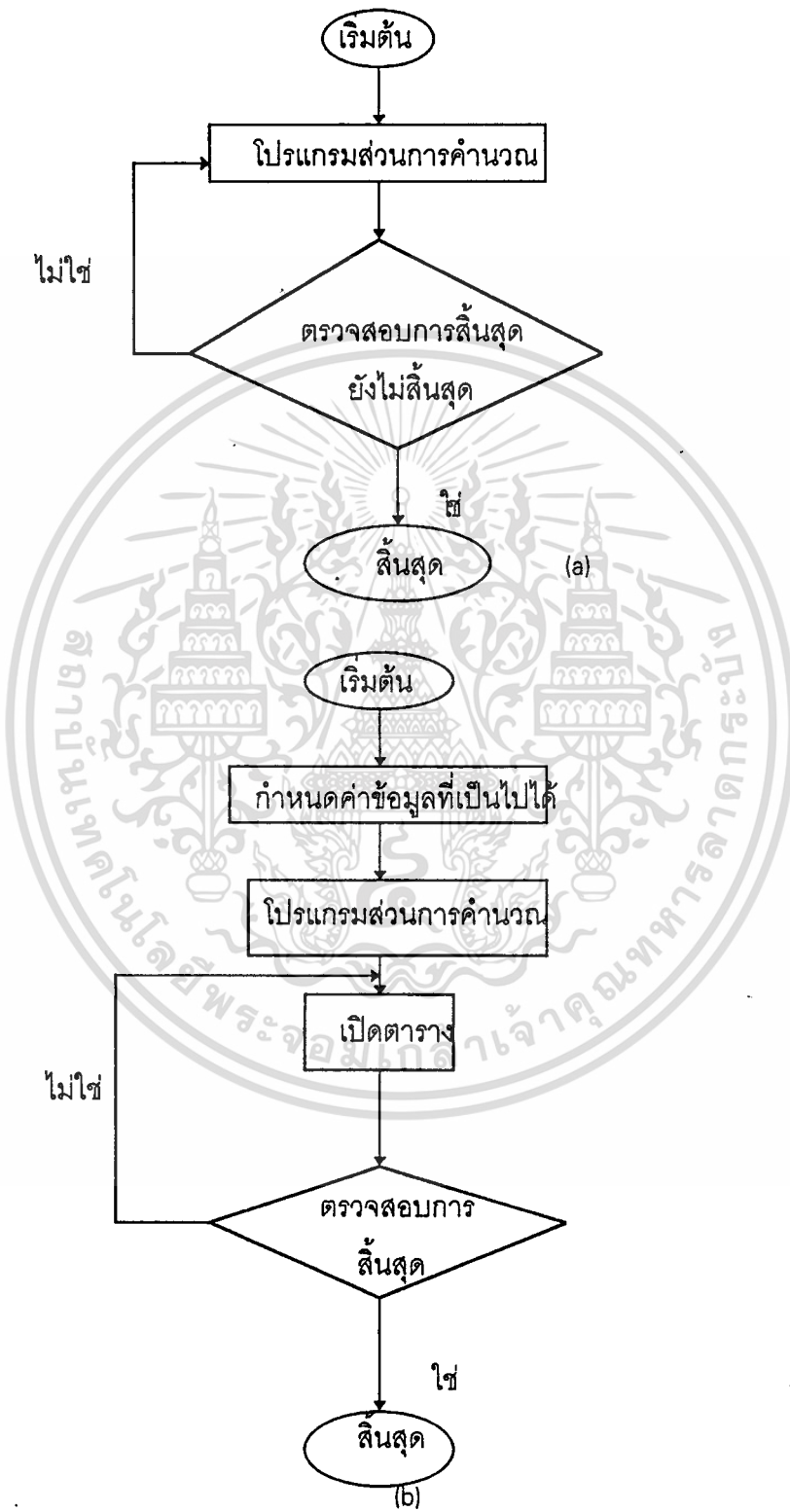
$$G'_L = T(G_L) \quad (3.5)$$

โดยที่ G_L และ G'_L เป็นค่าระดับสีเทาอินพุตและเอาต์พุตตามลำดับ และมี T คือ ฟังก์ชันการแปลง

ขั้นตอนในการทำการประมวลผลโดยตรงแสดงดังภาพที่ 13(a) ส่วนในการใช้วิธีการเปิดตารางแสดงดังภาพที่ 13(b) ซึ่งเป็นการเขียนโดยภาพารระดับสูงต่าง ๆ ฟังก์ชันการแปลง $T(G_L)$ อาจจะถูกกำหนดให้เป็นแอเรย์ (array) ที่มีจำนวนเต็ม (integer) G_L เป็นตัวชี้อยู่ ถ้ามีการใช้ ฟังก์ชัน T เดียวกันในการแปลงทุก ๆ จุดภายในภาพจึงสามารถใช้การคำนวณสำหรับทุก ๆ ค่าที่เป็นไปได้ของตัวชี้ G_L ก่อนการประมวลผล และทำการเก็บค่าระหว่างการคำนวณค่าต่าง ๆ ไว้ในแอเรย์ ภาพต้นแบบที่มีขนาด 8 บิตต่อหนึ่งจุดภาพจะต้องการตารางแอเรย์จำนวน 256 แถว ในการประมวลผลถัดมาก็เป็นการนำค่าระดับสีเทาของจุดภาพต่าง ๆ มาเป็นตัวชี้ตาราง T ค่าที่สอดคล้องของ T จะเป็นค่าระดับสีเทาเอาต์พุต จะไม่มีการคำนวณระหว่างการเปิดตาราง อีก ใช้เพียงตัวชี้ตำแหน่งที่อยู่ในตำแหน่งตารางที่ต้องการเท่านั้น

การเปิดตารางสามารถใช้สำหรับจุดประสงค์อื่น ๆ ได้อีกมากมาย ประหยัดจำนวน ครั้งในการคำนวณซึ่งหมายถึงเวลาที่ใช้ก็ลดลงนั่นเองเห็นได้ชัดเจนในการคำนวณเลขทศนิยม ที่ใช้เวลาในการคำนวณสูง

ภาพที่ 13



แสดงลำดับขั้นตอนวิธีการคำนวณโดยตรงเทียบกับการใช้วิธีการเปิดตาราง

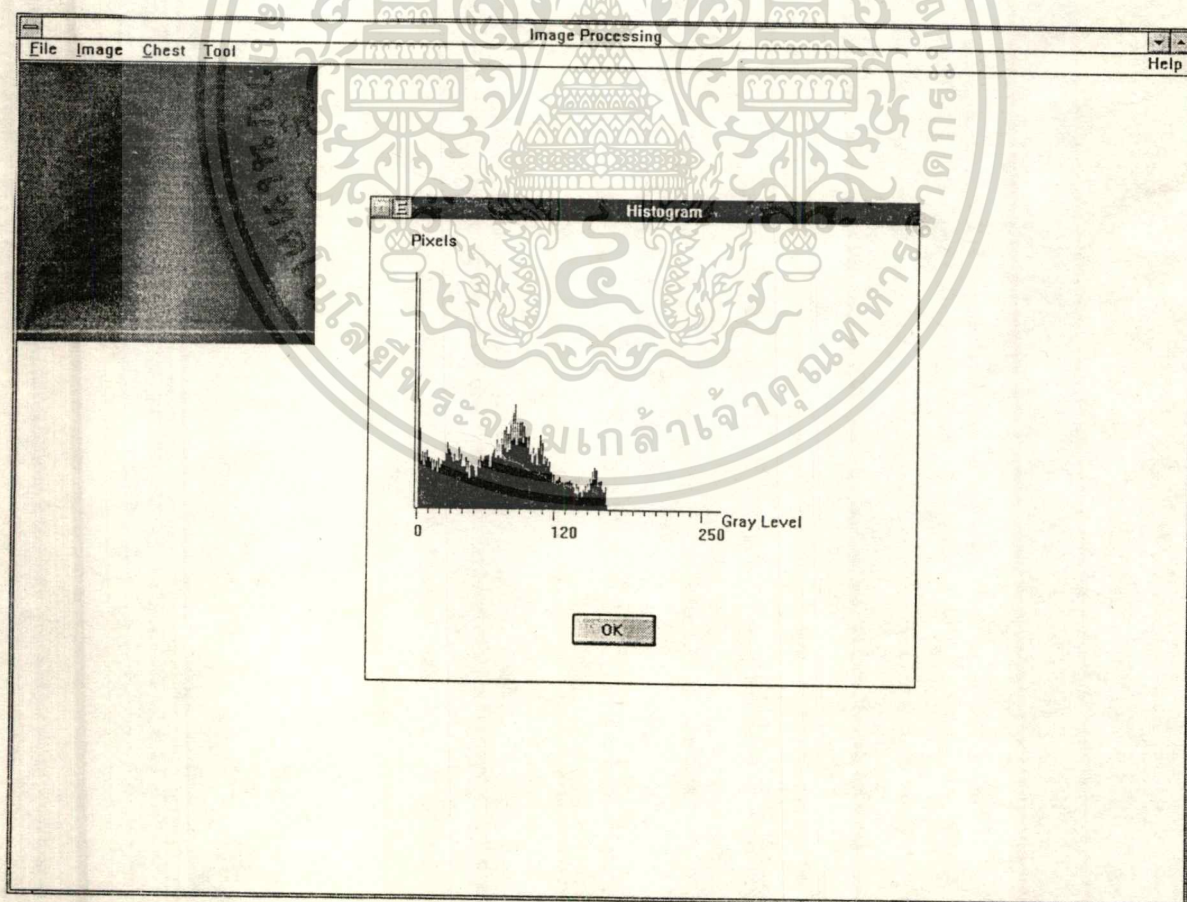
เอกสารนี้เป็น (a) วิธีการคำนวณโดยตรง (b) วิธีการเพิ่มความเร็วโดยการเปิดตาราง ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

วิธีการปรับปรุงภาพเพื่อช่วยในการวินิจฉัยภาพ

ในบทนี้จะกล่าวถึงอัลกอริทึมในการปรับปรุงภาพ ต่างๆ ที่นำมาใช้ในระบบงานโดยจะกล่าวถึงเป็นหัวข้อต่อไป สำหรับภาพที่เป็นภาพต้นแบบเป็นภาพเอ็กซเรย์ทรวงอก ขนาดภาพ 256x256 จุดภาพใหม่ของความละเอียดของจอภาพที่ใช้ 1024x768 จุดภาพ 256 สี โดยการเซตจากคอนโทรลพาเนล (Control Panel) ของระบบวินโดวส์ ให้ไปเรียกใช้ ไดรฟ์เวอร์ของการ์ดแสดงผลทางจอภาพ ซึ่งไดรฟ์เวอร์นี้จะมีพร้อมกับการตัวการ์ด ดังแสดงดังภาพข้างล่าง (ภาพที่แสดงจะแสดงฮิสโตแกรมของภาพที่ระดับเทาต่างในช่วง 0-255 พร้อมกันด้วย)

ภาพที่ 14



4.1 การขยายภาพ (ZOOMING)

บ่อยครั้งที่จำเป็นต้องการขยายภาพเพื่อดูรายละเอียดของภาพที่ใหญ่ขึ้นจะได้มองเห็นรายละเอียดเล็กๆได้ชัดเจนขึ้น การขยายภาพ([3] Anil K.Jain) นั้นมีอยู่ 2 วิธีด้วยกันคือ การทำซ้ำ และการแทรกแบบเชิงเส้น ซึ่งการขยายภาพส่วนใหญ่มักจะใช้วิธีการแทรกแบบเชิงเส้นรวมถึงโปรแกรมที่พัฒนาขึ้นนี้ด้วย โดยจะกล่าวเป็นหัวข้อดังนี้

4.1.1 การทำซ้ำ (Replication)

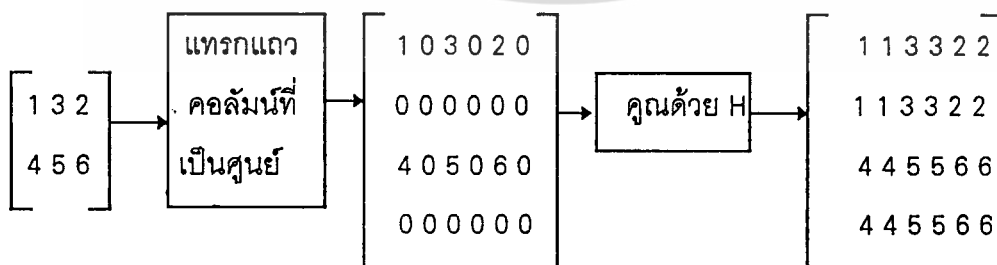
การทำซ้ำคือ เมื่อต้องการขยายภาพให้มีขนาดใหญ่ขึ้นก็จะแทรกแถวและคอลัมน์ที่เป็นศูนย์ไปทั้งแนวนอนและแนวตั้งของภาพ เช่น ถ้าต้องการขยายภาพขนาด $M \times N$ ถ้าแทรกแถวและคอลัมน์ที่เป็นศูนย์เข้าไปจะได้เมตริกซ์ขนาด $2M \times 2N$ และทำการคูณ (convolve) ด้วยเมตริก H ที่มีขนาด

$$H = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

ดังนั้นจะได้

$$g(m,n) = f(k,l), k = \lfloor m/2 \rfloor, L = \lfloor n/2 \rfloor \quad m,n=0,1,2,$$

โดยที่ $g(m,n)$ เป็นฟังก์ชันที่ทำการขยาย และ $f(k,l)$ เป็นฟังก์ชันเดิม จะแสดงตัวอย่างของการขยายภาพด้วยวิธีการทำซ้ำ ดังนี้



การขยายภาพด้วยวิธีนี้คือ การขยายขนาดของจุด ซึ่งภาพที่ได้จะมีลักษณะไม่เรียบ เป็นสี่เหลี่ยมของจุดติดกันไปทั้งภาพเพื่อที่จะแก้ปัญหานี้จะต้องใช้วิธีการแทรกแบบเชิงเส้นตรง (เนื่องจากวิธีการนี้เป็นวิธีที่มีปัญหาจึงเป็นอันถึงเพื่อใช้วิธีการต่อไป)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การแทรกแบบเชิงเส้นตรง (Linear Interpolation)

การทำการแทรกแบบเชิงเส้นตรงจะพิจารณาจุดตามแนวเส้นตรงที่จุดที่กำลังทำนั้น แทรกอยู่ระหว่างจุดในแถวหรือแนวตั้งจุดนั้นจะทำการอินเทอร์โพล์ขึ้นลักษณะตามแนวเส้นตรง ตัวอย่างมีเมตริกซ์ขนาด 2×2 การทำการแทรกเชิงเส้นตรงตามแนวนอน จะได้ดังนี้

$$g_1(m, 2n) = f(m, n) \quad , \quad 0 \leq m \leq M-1, 0 \leq n \leq N-1 \quad (4.1.2-1)$$

$$g_1(m, 2n+1) = 1/2[f(m, n) + f(m, n+1)] \quad , \quad 0 \leq m \leq M-1, 0 \leq n \leq N-1$$

การทำการแทรกเชิงเส้นตรงตามแนวตั้งจากผลลัพธ์ครั้งแรก

$$g(2m, n) = g_1(m, n) \quad (4.1.2-2)$$

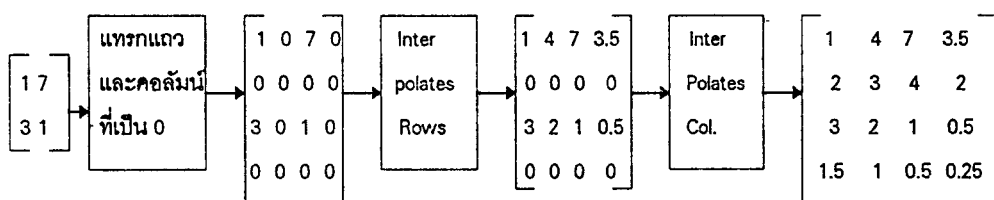
$$g(2m+1, n) = 1/2[g_1(m, n) + g_1(m+1, n)] \quad , \quad 0 \leq m \leq M-1, 0 \leq n \leq 2N-1$$

สมมติว่าภาพที่เริ่มต้นมีแถวและคอลัมน์รอบนอกเป็นศูนย์ขนาด $[0, M-1] \times [0, N-1]$ วิธีการที่กล่าวมาข้างต้นนั้นสามารถทำได้โดยคอนโวลูชันภาพขนาด $2M \times 2N$ ที่ทำการแทรกแถวและคอลัมน์ที่เป็นศูนย์จากภาพเดิมด้วยเมตริกซ์ โดยคำนวณเฉพาะตำแหน่งที่เป็น 0 แล้วนำค่าที่ได้ไว้ที่ภาพใหม่ (จะไม่ทำการแทนค่าไปที่ภาพที่กำลังคำนวณ) การคำนวณตำแหน่งอื่นๆ ทำจากภาพที่แทรกแถวและคอลัมน์ 0

$$H = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}$$

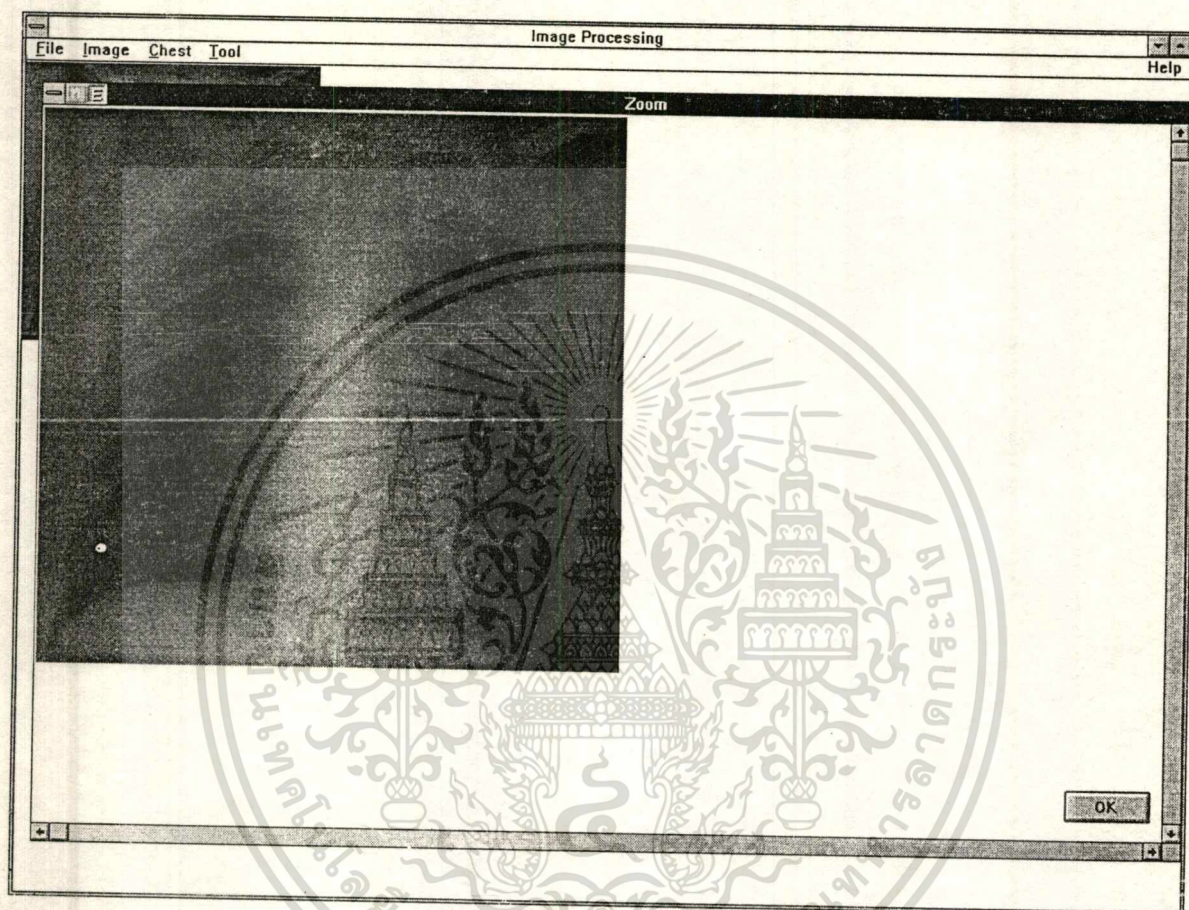
จุดศูนย์กลางของเมตริกซ์ H คือจุดที่กำลังพิจารณาในภาพ

ตัวอย่างการทำการแทรกเชิงเส้นตรง (Linear Interpolation) ด้วยเมตริกซ์ ขนาด 2×2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 15



ภาพที่ทำการขยายเป็น 2 เท่าจากภาพต้นแบบด้วยวิธีการแทรกเชิงเส้นตรง

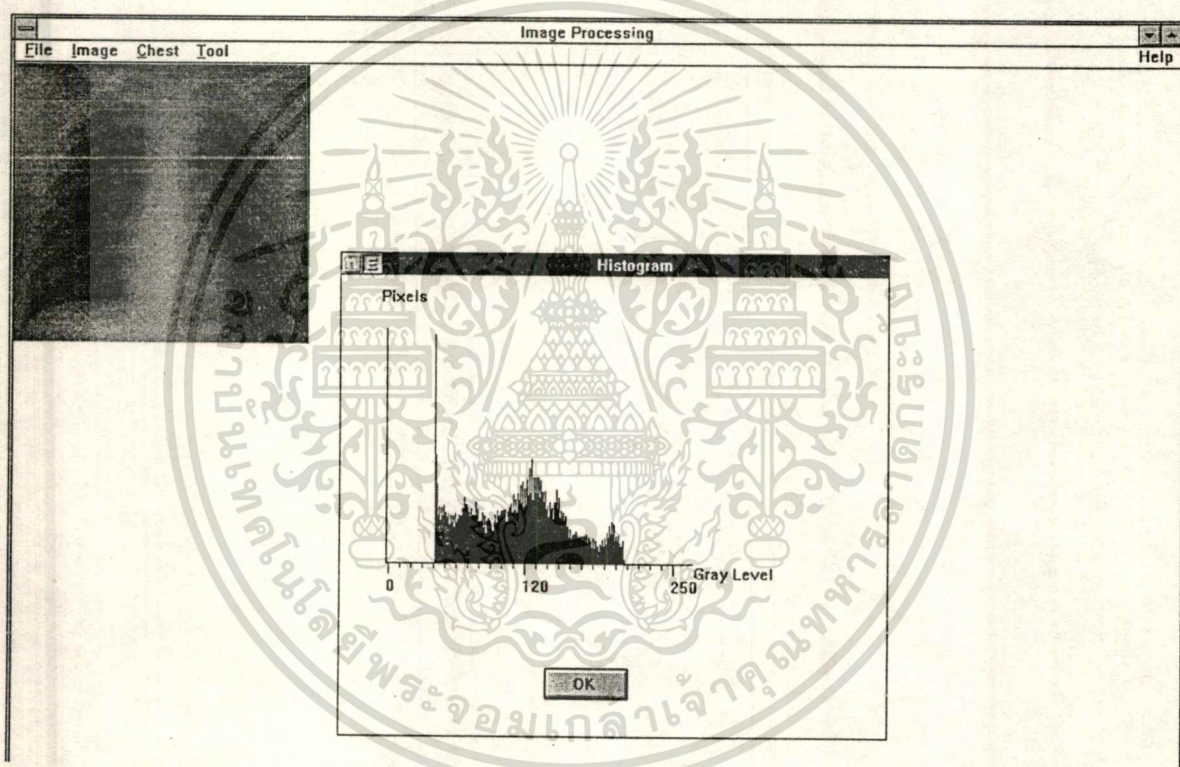
จากภาพที่ได้สามารถขยายภาพได้เป็น 2 เท่าและสามารถมองเห็นรายละเอียดได้ชัดเจนขึ้นและภาพที่ได้มีลักษณะเรียบเป็นภาพเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การปรับค่าความสว่างของภาพ (Brightness Adjust)

การปรับปรุงความสว่างของภาพ([2] Adrian Low)สามารถที่จะกระทำได้จากซอฟต์แวร์ โดยการบวกหรือลบด้วยค่าคงที่เข้าไปที่ภาพทั้งภาพเพื่อที่จะเพิ่มหรือลดความสว่าง ตามลำดับ (คือการขยับฮิสโตแกรมของภาพทั้งภาพไปทางบวกหรือลบ) ดังแสดงในภาพข้างล่าง

ภาพที่ 16



ภาพและฮิสโตแกรมที่ปรับความสว่างเพิ่มขึ้น

จากภาพที่ได้นั้นเมื่อปรับความสว่างของภาพเพิ่มขึ้น จะทำให้ส่วนของภาพที่อยู่ในระดับเทาต่ำ คือในส่วนของภาพที่มีดำนั้น ขยับระดับเทาออกมาในระดับเทาที่มากขึ้นตามปริมาณที่ปรับจากโปรแกรม ทำให้เห็นภาพในส่วนนี้ได้ดีขึ้น จากภาพและฮิสโตแกรมข้างบนระดับเทาของภาพจะเพิ่มขึ้นประมาณ 40 ระดับเทา

4.3 การปรับปรุงคอนทราสต์ของภาพ

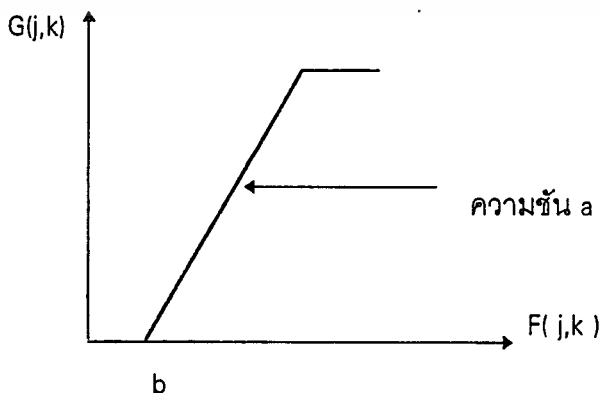
ในบางครั้งภาพที่ทำการพิจารณานั้นมีคอนทราสต์ของภาพต่ำคือภาพมีระดับเทาของจุดภาพใกล้เคียงกันตลอดทั้งภาพ ถ้ามองจากฮิสโตแกรมของภาพ จะพบว่าการกระจายของระดับเทาของจุดภาพจะรวมกันเป็นช่วงแคบๆทำให้ไม่สามารถมองเห็นรายละเอียดเล็กๆของภาพ ในการปรับปรุงคอนทราสต์ของภาพ โดยการเปลี่ยนลักษณะการกระจายฮิสโตแกรมของภาพใหม่ ให้ฮิสโตแกรมของภาพกว้างขึ้น และมีความต่างของค่าระดับสีเทาสูงขึ้นซึ่งจะทำให้สามารถมองเห็นรายละเอียดที่อยู่ในภาพเดิมที่ไม่สามารถมองเห็นได้ชัดเจนยิ่งขึ้น การปรับปรุงคอนทราสต์นั้นสามารถใช้ฟังก์ชันแบบเชิงเส้น และฟังก์ชันไม่เชิงเส้น อาจเป็นฟังก์ชันเส้นตรง, ฟังก์ชันกำลังสอง เป็นต้น เพื่อที่จะมาทำการกระจายฮิสโตรแกรมเสียใหม่

- การปรับปรุงคอนทราสต์โดยใช้ฟังก์ชันเส้นตรง ([9] อาโมทย์ สมบูรณ์แก้ว)

จากสมการที่ 4.3.1 ในเทอมสมการเส้นตรง $y = aF(j,k) + b$ นั้นจะมี a เป็นพารามิเตอร์เกนท์ (gain) ที่มีผลต่อความชัน (slope) ในการ mapping และมี b เป็นพารามิเตอร์ไบอัส (bias) เป็นค่าคงที่ที่บวกเข้าไปทุก ๆ จุดภาพ เมื่อนำมาใช้ในการปรับปรุงคอนทราสต์ของภาพต้องคำนึงถึงค่าที่ต่ำกว่าจุดตัดแกนเดิมที่ต้องกำหนดให้เป็นศูนย์ และค่าสูงที่เลยออกนอกย่านที่ต้องกำหนดให้เป็น 255 ดังแสดงในภาพที่ 17

$$G(j,k) = \begin{cases} 0 & \text{ถ้า } 0 < F(j,k) < \text{lower} \\ aF(j,k) + b & \text{ถ้า } \text{lower} < F(j,k) < \text{upper} \\ 255 & \text{ถ้า } \text{upper} < F(j,k) < 255 \end{cases} \quad (4.3.1)$$

ภาพที่ 17



แสดงฟังก์ชันในการดึงอย่างเชิงเส้นอย่างง่าย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ lower = กำหนดช่วงค่าของระดับเทาต่ำที่สุด ที่ต้องการปรับทำคอนทราส

upper = กำหนดช่วงค่าของระดับเทาสูงสุด ที่ต้องการทำปรับคอนทราส

b = ระดับเทาต่ำสุด

ในโปรแกรมจะกำหนดค่า lower = 0 และ upper = 255

จะเห็นว่าการกำหนดตัวแปรเกณฑ์ และไบอัสเพื่อให้ได้ฟังก์ชันเส้นตรงในการแปลงที่ต้องการนั้นกำหนดในลักษณะที่สื่อความหมายให้เข้าใจได้ยาก ดังนั้นในรูปที่กำหนดได้ง่ายเป็นรูปแบบที่นิยมใช้มากที่สุด ดังนี้

$$G(j,k) = \begin{cases} Y_{\min} & \text{ถ้า } 0 \leq F(j,k) \leq x_{\min} \\ F'(j,k) & \text{ถ้า } x_{\min} < F(j,k) < x_{\max} \\ Y_{\max} & \text{ถ้า } x_{\max} \leq F(j,k) \leq 255 \end{cases} \quad (4.3.2)$$

เมื่อมีการเพิ่มการกำหนดสมการเส้นตรงในรูปแบบที่ง่ายขึ้น ดังนี้

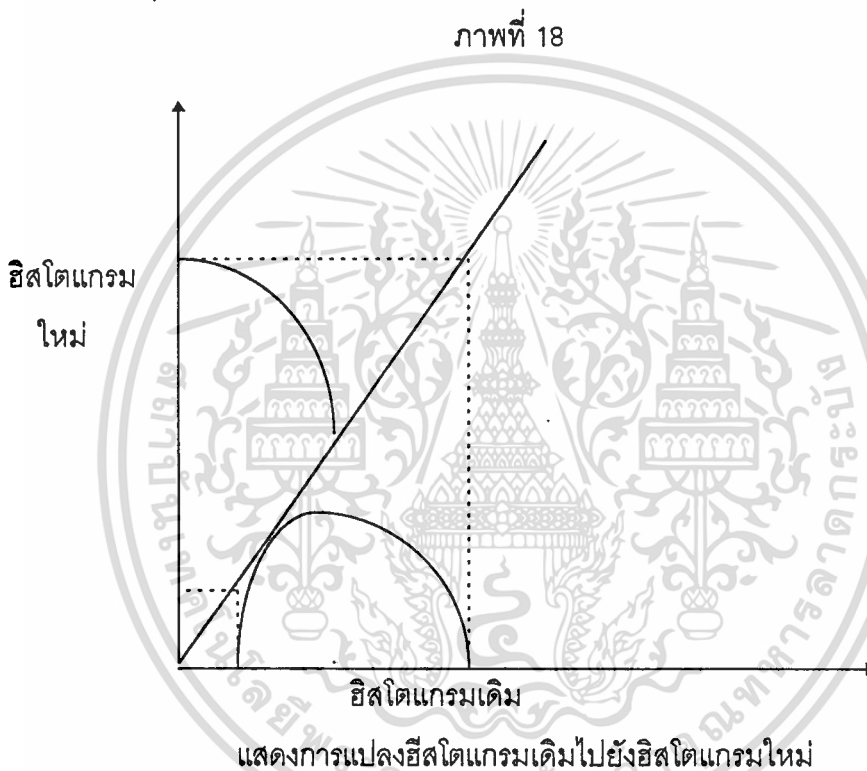
$$F'(j,k) = \frac{[(F(j,k) - x_{\min})(Y_{\max} - Y_{\min})]}{(x_{\max} - x_{\min})} + Y_{\min} \quad (4.3.3)$$

ในขณะที่วิธีการปกติในสมการที่ (4.3.1) ไม่สามารถทำได้นั้นสมการที่ (4.3.2) และ (4.3.3) มีข้อได้เปรียบอีกอย่างหนึ่งคือสามารถแปลงการกระจายไปสู่ทุกๆตำแหน่งบนฮิสโตแกรมใหม่ได้ทั้งหมดดังแสดงในรูปที่ 4.3.2 โดยสามารถกำหนดย่านบนแกนใหม่ได้ตามต้องการ

ในการประมวลผลสามารถนำวิธีการเปิดตารางดังที่ได้กล่าวมาแล้วในหัวข้อก่อนเพื่อช่วยเพิ่มความเร็วในการคำนวณได้ และนอกจากการให้ค่าสูงสุดและต่ำสุดโดยตรงแล้วยังสามารถใช้การคำนวณหาค่าสูงสุดและต่ำสุดของค่าระดับสีเทา แล้วทำการดึงอย่างอัตโนมัติ (automatic linear contrast stretching) ได้เช่นกันจึงมีประโยชน์อย่างมากในการนำไปใช้ดึงค่าที่เลยออกนอกย่านให้กลับเข้ามาอยู่ในย่าน 0 - 255

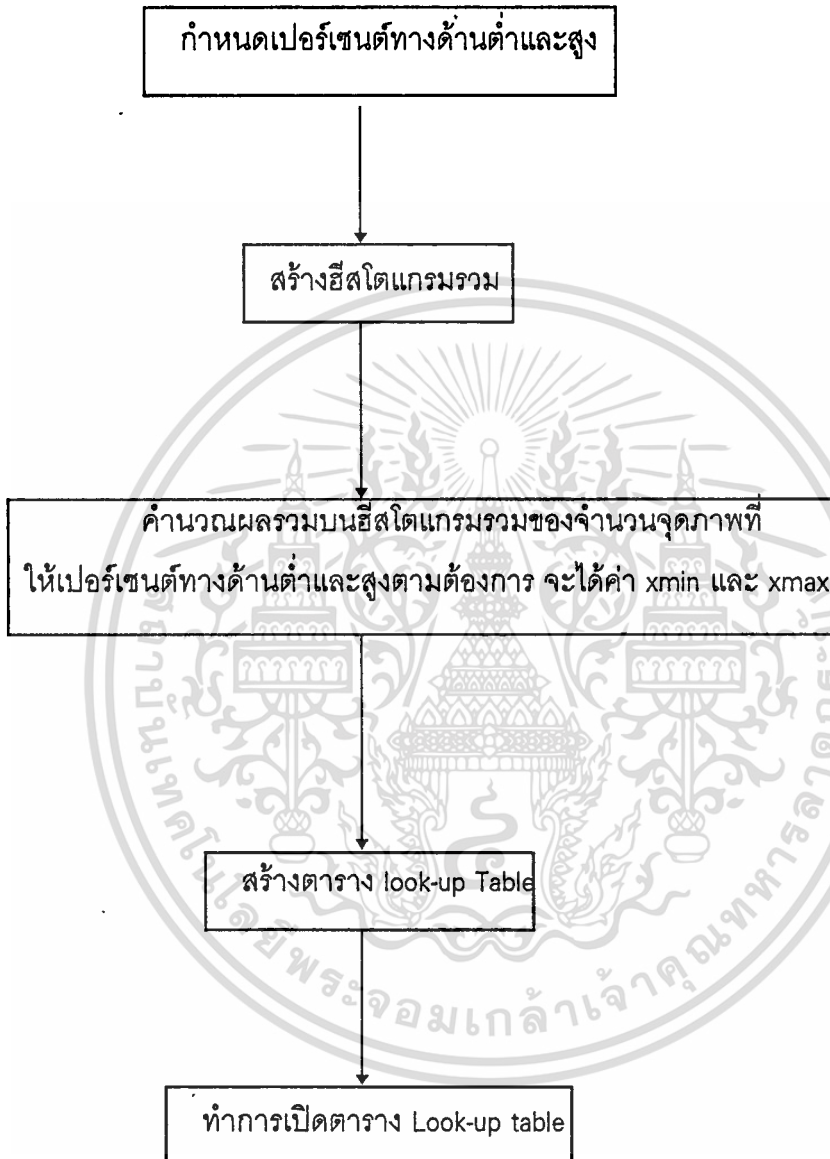
นอกจากนี้ในทางปฏิบัติจะพบภาพที่มีค่าระดับสีเทาเด่นออกมาจากกลุ่มค่าระดับสีเทาส่วนใหญ่ของภาพ ซึ่งค่าระดับสีเทานี้เมื่อเทียบกันแล้วจะเป็นเพียงจำนวนน้อย ๆ เท่านั้น ดังนั้นในการดึงอย่างอัตโนมัติจึงไม่อาจให้ค่าคอนทราสที่ดีออกมาได้ตัวอย่างเช่นเมื่อมีสัญญาณรบกวนไม่มากนักโดยที่ส่วนอื่นทั้งหมดมืดแต่ตรงกลาง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กวนที่มีค่าระดับสีเทาสูงอยู่ 1 จุดภาพที่เป็นค่าระดับสีเทาสูงสุดในภาพเมื่อทำการดึงอย่างอัตโนมัติค่าระดับสีเทาของสัญญาณรบกวนนี้จะเป็นค่า $x_{\max}$ ทำให้เกิดเกณฑ์ของการดึงกลุ่มข้อมูลหลักผิดไปโดยได้คอนทราสต์ที่น้อยกว่าที่ควรจะเป็น รูปที่ 4.3.4 แสดงฮิสโตแกรมของภาพผลลัพธ์ที่ได้จากการดึงอย่างอัตโนมัติด้วยค่าสูงสุดและต่ำสุดที่หาได้



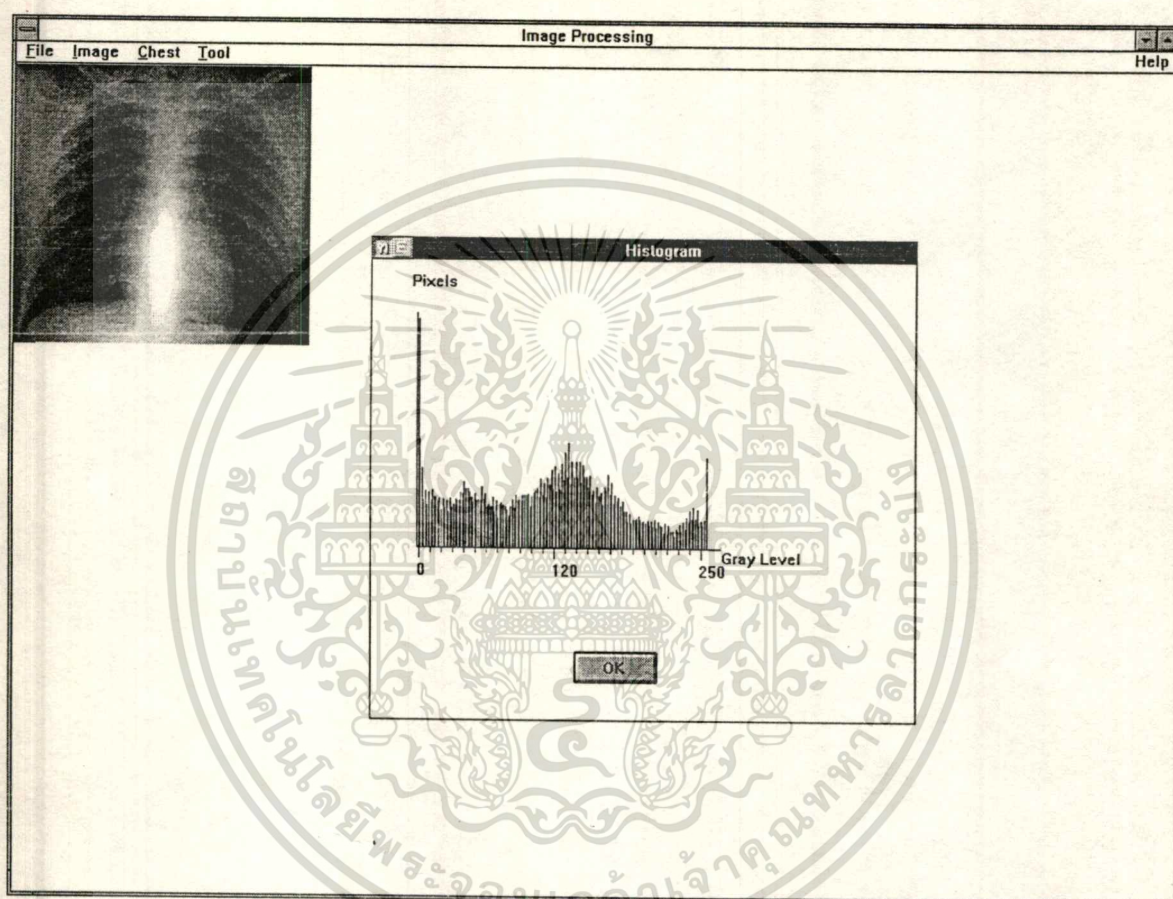
ในกรณีที่มีค่าแตกกลุ่มเป็นค่าเล็กๆจะทำให้ไม่สามารถให้ค่าคอนทราสต์ที่ออกมาได้ดังนั้น ในทางปฏิบัติวิธีการแก้ปัญหานี้ทั้งหมดนี้ให้หมดไป ทำได้โดยการนำเปอร์เซ็นต์ของจำนวนจุดภาพมาช่วยแก้ข้อบกพร่องของจุดภาพจำนวนน้อย ๆ เหล่านี้ทั้งทางด้านต่ำและทางด้านสูงของฮิสโตแกรมภาพผลลัพธ์ที่ได้จะมีคอนทราสต์ของกลุ่มข้อมูลส่วนใหญ่สูงตามที่ต้องการสำหรับภาพที่ 18 แสดงฮิสโตแกรมที่ทำการดึงอย่างอัตโนมัติ ด้วยค่าสูงสุดและต่ำสุด ที่นำค่าเปอร์เซ็นต์ของจำนวนจุดภาพทั้งทางด้านต่ำและทางด้านสูงมาใช้โดยข้อมูลทางด้านต่ำและทางด้านสูงใช้จำนวนจุดเท่ากับ 1 เปอร์เซ็นต์ของจำนวนจุดภาพทั้งหมด

ภาพที่ 19



ไฟล์ชาร์ตลำดับขั้นตอนการดึงเชิงเส้นอย่างอัตโนมัติด้วยค่าสูงสุดและ
ต่ำสุดที่หาได้จากการกำหนดเปอร์เซ็นต์ทางด้านต่ำและสูง

ภาพที่ 20



แสดงภาพและฮิสโตแกรมที่ทำการดึงอย่างอัตโนมัติ

จากภาพที่ได้ในภาพที่ 20 การกระจายของระดับเทาในภาพจะกระจาย ไปในทุก ๆ ระดับเทา 0 - 255 ระดับเทาเมื่อเทียบกับภาพต้นแบบในภาพที่ 14 ซึ่งภาพที่ได้นั้นมีฮิสโตแกรมของภาพรวมกันอยู่ ในช่วงระดับเทาต้นๆ หลังจากทำการปรับคอนทราสต์ของภาพด้วยการดึงเชิงเส้นแล้ว สามารถเห็นรายละเอียดของภาพได้ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การปรับปรุงฮิสโตแกรม (Histogram Modification Technique) ([4] William K. Pratt)

ฮิสโตแกรมของภาพสามารถที่จะแสดงถึงลักษณะโดยรวมของภาพได้วิธีการนี้จะเป็นการปรับปรุงภาพโดยทำการเปลี่ยนแปลงหรือแก้ไขฮิสโตแกรมภาพให้เป็นไปตามฮิสโตแกรมที่กำหนดให้ เมื่อเราให้ตัวแปร r แทนระดับความเข้มของจุดในภาพที่ต้องการปรับปรุง เพื่อให้ง่ายขึ้นเราจะแสดงค่า r ให้อยู่ในช่วง

$$0 \leq r \leq 1 \quad (4.4.1)$$

โดยที่ $r=0$ แทนสีดำ และ $r=1$ แทนสีขาวในระดับเทา

สำหรับ r ใด ๆ ที่มีค่าอยู่ในช่วง $[0,1]$ และเราจะสนใจรูปแบบการแปลงอยู่ในรูป

$$S = T(r) \quad (4.4.2)$$

ที่ค่าของระดับเทาในทุก ๆ S เกิดจากค่า r ในภาพเดิมซึ่งตั้งสมมุติฐานว่าฟังก์ชันในการแปลงเป็นดังสมการที่ (4.4.2) และจะต้องอยู่ในเงื่อนไข

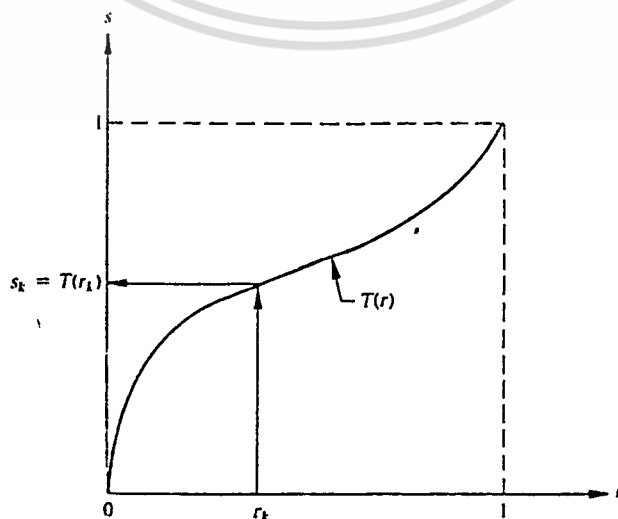
(1) $T(r)$ เป็นฟังก์ชันตัวแปลงเดียว (single-value) และเป็นฟังก์ชันเพิ่มจากจุดเริ่มต้นไปจุด

ปลายสุดทางเดียว (monotonically increasing) ในช่วง $0 \leq r \leq 1$

(2) $0 \leq T(r) \leq 1$ สำหรับ $0 \leq r \leq 1$

เงื่อนไขในข้อที่ 1 เพื่อที่จะป้องกันและรักษาลำดับของระดับเทาจากขาวไปดำในขณะที่เงื่อนไขที่ 2 เพื่อเป็นข้อยืนยันการแปลงค่า ค่าของระดับเทาของจุดจะอยู่ในช่วงที่กำหนด สำหรับตัวอย่างฟังก์ชันแปลงที่เป็นไปตามเงื่อนไขทั้ง 2 แสดงในภาพที่ 21

ภาพที่ 21

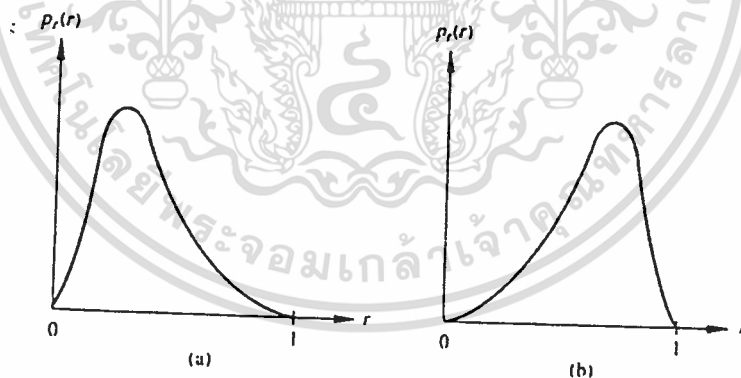


ในการทำฟังก์ชันแปลงอินเวอร์ส (inverse transform) จาก s กลับไป r แสดงดังข้างล่างนี้

$$r = T^{-1}(s) \quad 0 \leq s \leq 1 \quad (4.4.3)$$

โดยที่ $T^{-1}(s)$ ค่าของตัวแปร s จะต้องอยู่ภายใต้เงื่อนไขของสองข้อที่กล่าวข้างบนระดับเทา ในภาพจะอยู่กระจ่ายเป็นแบบสุ่ม (random) ในช่วง $[0,1]$ สมมติให้ตัวแปรนี้เป็นตัวแปรแบบต่อเนื่อง รูปภาพเดิมและฟังก์ชันแปลงของระดับเทาจะขึ้นอยู่กับฟังก์ชันความหนาแน่นของความน่าจะเป็น (probability density function) $P_r(r)$ และ $P_s(s)$ ตามลำดับ ซึ่งค่าฟังก์ชันความหนาแน่น (density function) ของระดับเทาสามารถบอกลักษณะทั่วไปของภาพได้ ตัวอย่างดังในภาพ 4.4.2(a) จากฟังก์ชันความหนาแน่นในภาพ ค่าของระดับเทาส่วนใหญ่จะอยู่ในช่วงเทาดำของระดับเทา แสดงว่าภาพนี้มีลักษณะค่อนข้างดำ หรือในภาพ 22(b) ลักษณะของภาพจะออกมาในโทนค่อนข้างสว่างหรือ สีขาว เพราะระดับเทาส่วนใหญ่อยู่ในโทนสีขาว

ภาพที่ 22



แสดงฟังก์ชันความหนาแน่นของระดับเทา (a)ภาพที่มืด (b)ภาพที่สว่าง

วิธีการปรับปรุงนี้จะเป็นการปรับค่าของฟังก์ชันของความน่าจะเป็นของระดับเทาโดยใช้ฟังก์ชันแปลง เป็นตัวกำหนดค่า

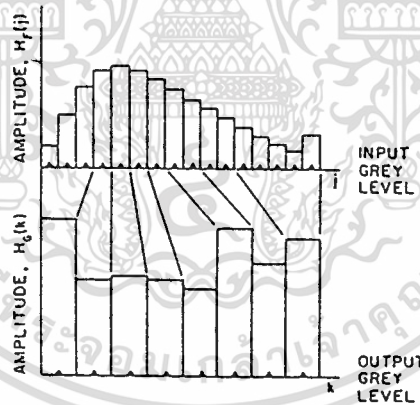
4.4.1 การปรับปรุงภาพโดยวิธีฮิสโตแกรมอีควอลไรเซชัน (Histogram Equalization)

จากภาพที่ 23 แสดงตัวอย่างของฮิสโตแกรมอีควอลไรเซชัน จากรูป $H_r(j)$ สำหรับ $j=1,2,\dots,J$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า แสดงสัดส่วนระหว่างขนาดของฮิสโตแกรมที่ระดับที่ j กับจำนวนจุดทั้งหมดของภาพวิธีการฮิสโตแกรมอีควอลไรเซชัน ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฮโดรเจนไอออน จะสร้างภาพผลลัพธ์ที่มีกระจายของระดับเทา G ให้มีขนาดเท่ากันทุก ๆ ระดับ $H_G(k)=1/k$ สำหรับ $k=1,2,...,K$ จากตัวอย่างในรูป 4.4.3 จำนวนระดับเทาของไฮโดรเจนไอออนผลลัพธ์จะกำหนดเพียงครั้งหนึ่งของจำนวนระดับเทาของไฮโดรเจนไอออนเริ่มต้น สำหรับวิธีการแปลงมีดังนี้ เริ่มจากระดับเทาต่ำสุดของไฮโดรเจนไอออนต้นแบบ จะทำการรวมไฮโดรเจนไอออนแต่ละระดับเทาให้มีค่าเข้าใกล้ค่าเฉลี่ย แล้วสร้างระดับเทาแรกในภาพใหม่เป็นระดับเทาเริ่มต้น วิธีการนี้จะทำต่อไปที่ค่าระดับเทาที่สูงขึ้นจนกระทั่งจบ ถ้าจำนวนของระดับเทาของภาพเริ่มต้นมีจำนวนมาก ค่าระดับเทาของภาพที่ปรับปรุง มักจะมีค่าระดับเทาน้อยกว่าภาพต้นแบบ ในการทำไฮโดรเจนไอออนสามารถกำหนดจำนวนระดับเทาของภาพต้นแบบให้เท่ากับภาพที่ปรับปรุงซึ่งในการปรับปรุงภาพจะจัดการกระจายของไฮโดรเจนไอออนในภาพใหม่

ภาพที่ 23



ตัวอย่างการใช้วิธีไฮโดรเจนไอออนด้วยระดับเทาที่ไม่เท่ากัน โดยประมาณ

การปรับปรุงไฮโดรเจนไอออนสามารถพิจารณาในรูปฟังก์ชันแปลง $g_k = T \{ f_j \}$ ขนาดของตัวแปร f_j เป็นดังนี้ $f_1 \leq f_j \leq f_j$ ที่แปลงไปยังตัวแปรผลลัพธ์ $g_1 \leq g_k \leq g_k$ โดยที่การแจกแจงความน่าจะเป็นของต้นแบบ (input probability) $P_R \{ f_j = a_j \}$ เป็นตัวกำหนดการแจกแจงความน่าจะเป็นผลลัพธ์ $P_R \{ g_k = b_k \}$ ที่ a_j และ b_k เป็นค่าของความน่าจะเป็นที่ระดับ j และ k ดังนั้นผลรวมการกระจายของความน่าจะเป็น (probability distribution) ของภาพต้นแบบและผลลัพธ์เป็นหนึ่ง

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\sum_{j=1}^J P_R \{ f_j = a_j \} = 1 \quad (4.4.1a)$$

$$\sum_{k=1}^K P_R \{ g_k = b_k \} = 1 \quad (4.4.1b)$$

นอกจากนั้นค่าของการแจกแจงสะสม (cumulative distribution) จะต้องมามีค่าเท่ากัน ที่ $b_k = T(a_j)$

$$\sum_{n=1}^k P_R \{ g_n = b_n \} = \sum_{m=1}^j P_R \{ f_m = a_m \} \quad (4.4.2)$$

ผลรวมของการแจกแจงความน่าจะเป็นสะสมทางด้านขวามือของสมการ (4.4.2) (ภาพต้นแบบ) สามารถแทนด้วยค่าฮิสโตแกรมสะสม (Cumulative Histogram)

$$\sum_{n=1}^k P_R \{ g_n = b_n \} = \sum_{m=1}^j H_F(m) \quad (4.4.3)$$

จากสมการที่ 4.4.3 จะต้องเปลี่ยนให้อยู่ในรูปของ g_k ที่หาได้จาก f_j ผลลัพธ์ของการทำคือ สร้างตารางสำหรับแต่ละระดับเทาผลลัพธ์จากระดับเทาต้นแบบ ฟังก์ชันการแปลงฮิสโตแกรมหาได้จาก การประมาณค่าการกระจายความน่าจะเป็นแบบไม่ต่อเนื่องจากการกระจายความน่าจะเป็นแบบต่อเนื่อง (continuous probability density) ผลลัพธ์ของการประมาณคือ

$$\int_{g_{\min}}^g p_g(g) dg = \int_{f_{\min}}^f p_f(f) df \quad (4.4.4)$$

ที่ $p_f(f)$ และ $p_g(g)$ เป็นความหนาแน่นความน่าจะเป็น f และ g ตามลำดับตัวแปรอินทิกรัล (integral) ทางด้านขวามือของสมการ (4.4.4) คือ ฟังก์ชันการกระจายความน่าจะเป็นสะสม ของภาพต้นแบบของตัวแปร f ดังนั้น

$$\int_{g_{\min}}^g p_g(g) dg = P_f(f) \quad (4.4.5)$$

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นสำหรับการใช้เรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $P_f(f) = \int_{f_{\min}}^f p_f(f) df$

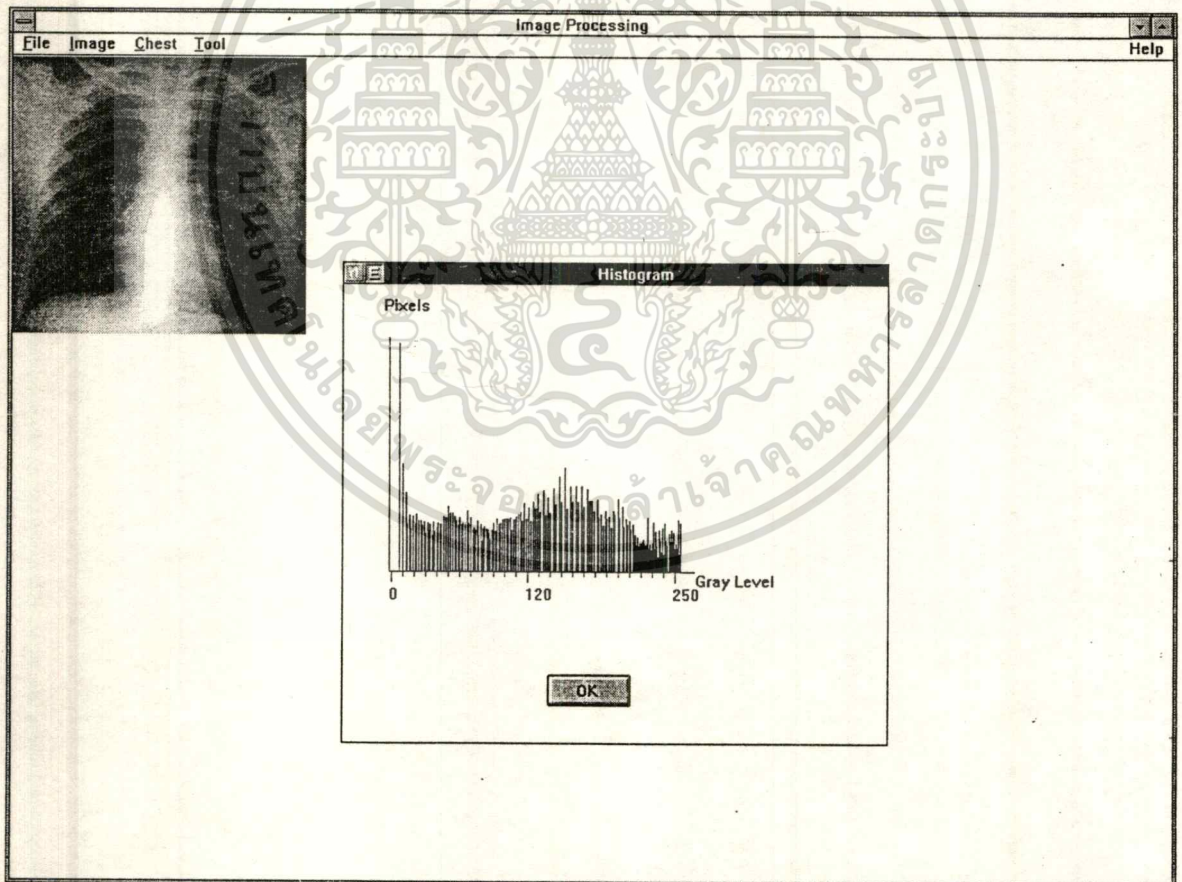
เนื่องจากต้องการผลลัพธ์ของฮิสโตแกรมที่มีการแจกแจงเท่ากัน ดังนั้นจะได้

$$P_g(g) = 1 / (g_{\max} - g_{\min}) \quad (4.4.6)$$

สำหรับ $g_{\min} < g < g_{\max}$ ฟังก์ชันแปลงของฮิสโตแกรมอิกวอไรเซชันจะได้ดังสมการข้างล่าง

$$g = [g_{\max} - g_{\min}] P_f(f) + g_{\min} \quad (4.4.7)$$

ภาพที่ 24



แสดงภาพการแปลงโดยฮิสโตแกรมอิกวอไรเซชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพและฮิสโตแกรมที่ได้หลังจากการทำวิธีการฮิสโตแกรมอิกวอไรเซชัน การกระจายของระดับเทาของภาพออกมาได้มากกว่า การกระจายของฮิสโตแกรมของภาพต้นแบบเดิม ทำให้ภาพมีการกระจายคอนทราสต์ที่ดีขึ้นจึงสามารถเห็นรายละเอียดในส่วนของภาพที่มีคอนทราสต์ที่ใกล้เคียงกันได้ดีขึ้น

ภาพที่ 25



4.5 การทำภาพให้ราบเรียบ (Image Smoothing) ([1] Rafael C.Gonzalez)

การทำภาพให้ราบเรียบนั้นเป็นการช่วยขจัดหรือลดสัญญาณรบกวน (noise) ที่สามารถเกิดขึ้นได้ในภาพดิจิทัล ซึ่งอาจเกิดมาจากระบบถ่ายภาพไม่ดีพอหรืออาจเกิดจากการสื่อสารข้อมูล การทำภาพให้ราบเรียบนั้นสามารถกระทำได้ทั้งสเปเชียลโดเมนและโดเมนความถี่ ในที่นี้จะกล่าวเฉพาะส่วนที่เป็นสเปเชียลโดเมนโดยใช้การทำคอนวูลูชันสำหรับวิธีการทำภาพให้ราบเรียบมีอยู่ด้วยกันหลายวิธี เช่น การเฉลี่ยรอบจุด, การเฉลี่ยถ่วงน้ำหนัก, การหากลาง เป็นต้น เนื่องจากการปรับภาพด้วยวิธีนี้จะไม่ค่อยได้ใช้เพราะภาพส่วนใหญ่จะไม่มีสัญญาณรบกวนในโปรแกรมจะใช้วิธีการเฉลี่ยรอบจุด

4.5.1 การเฉลี่ยรอบจุด (Neighborhood Averaging (Mean))

ให้ภาพขนาด $N \times N$ ของภาพ $f(x, y)$ จะทำให้กลมกลืน $g(x, y)$ จากการเฉลี่ยค่าระดับความเข้มที่ทุกจุด (x, y) จากบริเวณโดยรอบของจุด (x, y) ซึ่งเราจะเขียนได้ดังนี้

$$g(x, y) = \frac{1}{M} \sum_{(n, m) \in s} f(n, m) \quad (4.5.1)$$

โดย $x, y = 0, 1, \dots, N-1$ s เป็นเซตของคู่ลำดับที่อยู่ในบริเวณของจุด (x, y) ซึ่งรวม (x, y) ด้วย และ M คือจำนวนจุดที่อยู่ในบริเวณนั้น ซึ่งถ้าเป็นบริเวณ 3×3 เราจะสามารถเขียนอยู่ในรูปฟิลเตอร์เวท (filter weights) เป็นดังนี้

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

square window

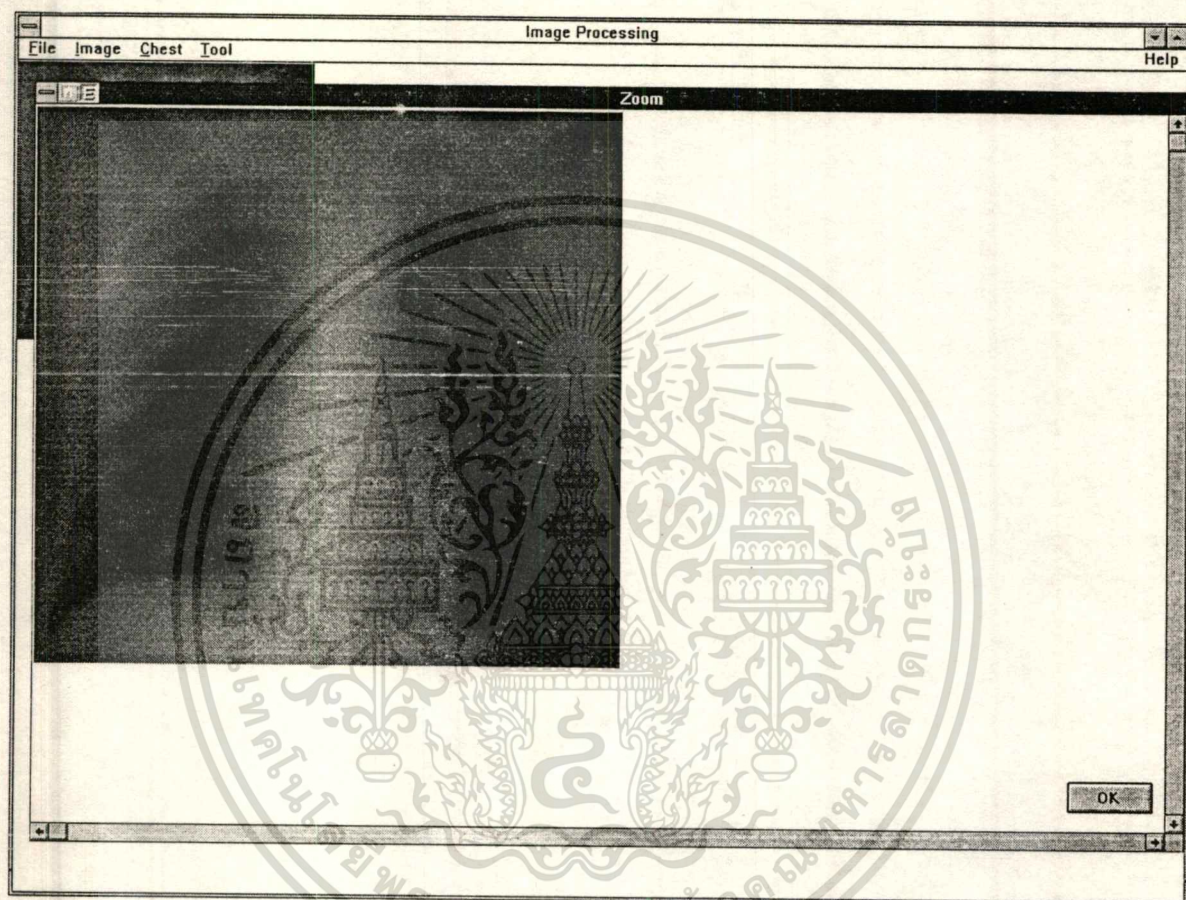
หรือในบางกรณีกำหนดให้ M อยู่ในรูปร่างเลขบวกเราก็จะได้มาส์กดังนี้

$$\begin{bmatrix} 1/5 & & \\ 1/5 & 1/5 & 1/5 \\ & 1/5 & \end{bmatrix}$$

plus shaped window

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 26



แสดงภาพที่ทำให้เรียบด้วยวิธีหาค่าเฉลี่ย

จากภาพที่ได้นั้น เมื่อเทียบกับภาพต้นแบบแล้ว ภาพจะมีลักษณะมัวไม่ชัดเจนเนื่องจากนำจุดบริเวณใกล้เคียงกันมาเฉลี่ย ในปัจจุบันนี้ภาพที่ได้จากอุปกรณ์ถ่ายภาพส่วนใหญ่ภาพที่ได้มักจะมีคุณภาพที่ดี ไม่ค่อยมีสัญญาณรบกวน ดังนั้นการใช้งานจริงวิธี การนี้จะไม่ได้ใช้งานเท่าใดนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การปรับปรุงภาพให้คมชัด (Image Sharpening) ([5] Ibrahim Sezan)

เป็นวิธีการที่ต้องการเพิ่มความคมชัดของขอบภาพ เพื่อที่จะสามารถเห็นรายละเอียดของภาพได้ดีขึ้น วิธีที่นิยมใช้กันมากคือ วิธีการทำอานชาร์ปมาส์กิง จะพยายามเน้นในส่วนที่เป็นขอบภาพโดยใช้พารามิเตอร์ในการเน้นขอบภาพ สามารถแบ่งเป็น 2 วิธีคือ การทำอานชาร์ปมาส์กิงแบบปรับไม่ได้ และการทำอานชาร์ปมาส์กิงแบบปรับได้

4.6.1 การทำอานชาร์ปมาส์กิงแบบปรับไม่ได้ (Non-Adaptive Unsharpmasking)

การทำอานชาร์ปมาส์กิงคำนวณได้จากการกรองภาพความถี่ต่ำ (low-pass filtering) จากภาพต้นแบบ แล้วนำค่านี้ไปลบจากภาพต้นฉบับเดิมภาพที่ได้จะเป็นภาพที่มีความถี่สูง (high frequency image) ซึ่งภาพความถี่สูงนี้จะเป็นบริเวณของขอบภาพหรือรายละเอียดของภาพหลังจากได้ภาพความถี่สูงแล้ว จะทำการเพิ่มหรือเน้นภาพนี้ด้วยการคูณพารามิเตอร์เข้าไปแล้วนำไปบวกกับภาพต้นแบบ จะแสดงสมการข้างล่างนี้

$$g(x,y) = f(x,y) + \beta [f(x,y) - \bar{f}(x,y)] \quad (4.6.1)$$

โดยที่ $g(x,y)$ = ค่าระดับเทาของภาพที่ตำแหน่ง (x,y)
 $f(x,y)$ = ค่าระดับเทาของภาพก่อนการทำอานชาร์ปมาส์กิง
 $\bar{f}(x,y)$ = ค่าระดับเทาของภาพความถี่ต่ำ ที่ตำแหน่ง x,y เป็นสัมประสิทธิ์ของภาพความถี่สูง (high frequency emphasis parameter) ที่ > 0
 β = พารามิเตอร์เน้น (Emphasis Parameter)

สำหรับภาพความถี่ต่ำ $\bar{f}(x,y)$ หาได้จากการทำให้เรียบ (smoothing) คือการหาค่าเฉลี่ยของจุดภาพต่าง ๆ เทมเพลตโดยการนำมาทำคอนโวลูชันกับข้อมูลภาพเดิมจะเขียนเทมเพลตเป็นเมตริกซ์ขนาด 3×3 ได้ดังนี้

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

หรือจะใช้เทมเพลตที่กล่าวมาแล้วในหัวข้อการทำภาพให้เรียบ ถ้าให้ค่า β มีค่าเท่ากับ 1 จากการมาส์กของอินซาร์ปมาส์กก็จะหาได้ดังนี้ สมมติให้จุด x_4 เป็นจุดที่กำลังประมวลผล แทนจุด $f(x,y)$

$$\bar{f}(x,y) = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} X_0 & X_1 & X_2 \\ X_3 & X_4 & X_5 \\ X_6 & X_7 & X_8 \end{bmatrix}$$

$$= \frac{X_0+X_1+X_2+X_3+X_4+X_5+X_6+X_7+X_8}{9}$$

การหาความถี่สูงในแต่ละตำแหน่งจุดภาพ X_4 ได้จาก

$$\begin{aligned} \bar{f}(x,y) - f(x,y) &= X_4 - \frac{X_0+X_1+X_2+X_3+X_4+X_5+X_6+X_7+X_8}{9} \\ &= \frac{9X_4 - X_0 - X_1 - X_2 - X_3 - X_4 - X_5 - X_6 - X_7 - X_8}{9} \end{aligned}$$

เมื่อทำการเน้นขอบจะได้ว่า

$$\begin{aligned} g(x,y) &= f(x,y) + [\bar{f}(x,y) - f(x,y)] = X_4 + \frac{9X_4 - X_0 - X_1 - X_2 - X_3 - X_4 - X_5 - X_6 - X_7 - X_8}{9} \\ &= \frac{9X_4 + 9X_4 - X_0 - X_1 - X_2 - X_3 - X_4 - X_5 - X_6 - X_7 - X_8}{9} \\ &= \frac{18X_4 - X_0 - X_1 - X_2 - X_3 - X_4 - X_5 - X_6 - X_7 - X_8}{9} \\ &= \frac{-X_0 - X_1 - X_2 - X_3 - 17X_4 - X_5 - X_6 - X_7 - X_8}{9} \end{aligned}$$

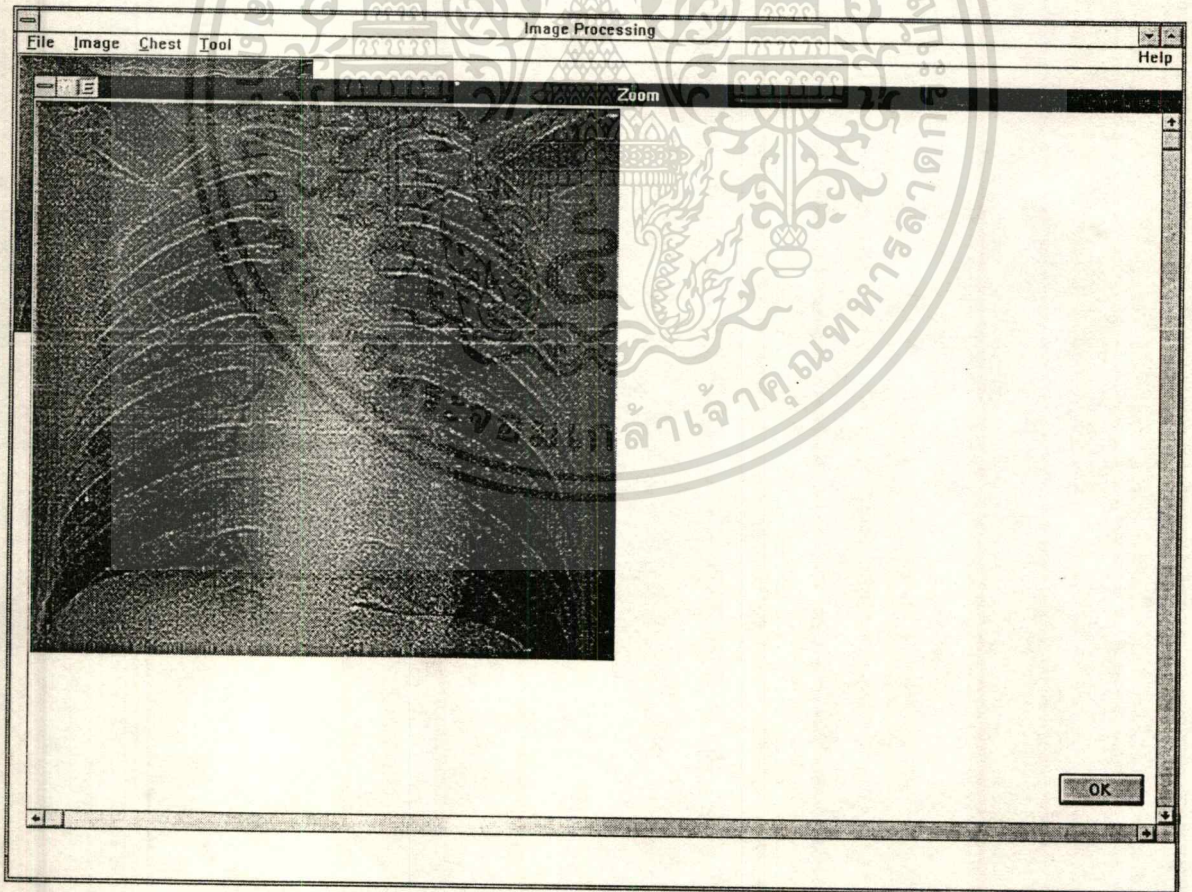
นั่นคือเมื่อนำมาเขียนในรูปเมตริกซ์ใหม่จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$g(x,y) = 1/9 \begin{bmatrix} -1 & -1 & -1 \\ -1 & 17 & -1 \\ -1 & -1 & -1 \end{bmatrix} \otimes \begin{bmatrix} X_0 & X_1 & X_2 \\ X_3 & X_4 & X_5 \\ X_6 & X_7 & X_8 \end{bmatrix}$$

ดังนั้น $\bar{f}(x,y)$ คือค่าเฉลี่ยเฉพาะบริเวณ (local mean) และ $f(x,y) - \bar{f}(x,y)$ เป็นค่าความแปรปรวนจากค่าเฉลี่ยเฉพาะบริเวณ (deviation from local mean) หรือเรียกว่าเป็นค่าคอนทราสต์เฉพาะส่วน (local contrast)

ภาพที่ 27



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 แสดงภาพที่ปรับความคมชัด (Unsharp Masking)
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ได้ในภาพที่ 27 ภาพที่ได้ในส่วนของภาพที่เป็นบริเวณของขอบภาพจะมีความชัดเจนขึ้น แต่ภาพที่ได้จะมีลักษณะหยาบ

4.6.2 การทำอันชาร์ปมาส์กิ้งแบบปรับได้ (Adaptive Unsharp Masking)

การทำอันชาร์ปมาส์กิ้งแบบปรับได้นั้นตัวแปรสัมประสิทธิ์ในการเน้นขอบภาพ (emphasis parameter) จะเปลี่ยนแปลงหรือแปรตามค่าของจุดที่ $f(x,y)$ หรือค่าเฉลี่ย เฉพาะบริเวณ (local mean) $\bar{f}(x,y)$ หรือแปรตามค่าคอนทราสต์เฉพาะส่วน (Local Contrast) $f(x,y) - \bar{f}(x,y)$ สามารถเขียนความสัมพันธ์ได้คือ

$\beta = \beta(f(x,y), \bar{f}(x,y), f(x,y) - \bar{f}(x,y))$ จากสมการที่ (4.6.1) สามารถเขียนได้เป็น

$$g(x,y) = f(x,y) + \beta(f(x,y), \bar{f}(x,y), f(x,y) - \bar{f}(x,y)) \cdot [f(x,y) - \bar{f}(x,y)] \quad (4.6.2-1)$$

วิธีการนี้จะนำไปใช้ในหัวข้อที่ 4.10 ต่อไป

4.7 การหาขอบภาพ (Edge Detection) ([1] Rafael C. Gozalez)

เพื่อช่วยในการวิเคราะห์และวินิจฉัยภาพ โปรแกรมสมารถที่จะทำการหาขอบภาพได้สำหรับวิธีการที่ใช้ในการหาขอบภาพนั้นยังใช้หลักการเดิมคือการมาส์กเป็นเมตริกซ์หรือการทำคอนวูลูชัน จะนิยามขอบภาพว่าเป็นบริเวณระหว่างพื้นที่ 2 ส่วนซึ่งมีระดับความเข้มต่างกันมาก ขอบภาพนั้นจะหาได้จากการหาอนุพันธ์ (Differential) ในการหาอนุพันธ์อันดับหนึ่งจะได้ความชันหรือ อัตราการเปลี่ยนแปลงของระดับความเข้ม เมื่อหาอนุพันธ์อันดับที่สองจะได้ค่าหรือบริเวณที่มีการเปลี่ยนแปลงมากหรือสูงสุด ดังนั้นบริเวณที่ว่าเป็นคือบริเวณที่เป็นขอบภาพวิธีการหาขอบภาพมีอยู่หลายวิธีด้วยกันวิธีการเหล่านี้ใช้การหาค่าเกรเดียน (Gradient Operation)

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix} \quad (4.7.1)$$

จากสมการข้างบนค่าเกรเดียนของภาพที่ $f(x,y)$ ที่ตำแหน่ง (x,y) กำหนดเป็นเวกเตอร์ 2 มิติ เป็นที่ทราบกันดีอยู่แล้วในเรื่องการวิเคราะห์เวกเตอร์ (Vector Analysis) ว่าเวกเตอร์ G แสดงทิศทางของอัตราการเปลี่ยนแปลงของ f ที่ตำแหน่ง (x,y) สำหรับการหาขอบภาพเราสนใจขนาดของเวกเตอร์ เราจะได้ค่าเกรเดียนเป็น

$$G[f(x,y)] = [G_x^2 + G_y^2]^{1/2} \quad (4.7.2)$$

ขนาดที่ได้นี้เป็นอัตราการเปลี่ยนแปลงของ $f(x,y)$ ต่อหน่วยระยะทางของ G เราสามารถประมาณโดยใช้ค่าสมบูรณ์

$$G[f(x,y)] = | G_x | + | G_y | \quad (4.7.3)$$

ถ้าเรามาสักเป็นเมตริกซ์ขนาด 3×3 ตามรูปที่แสดง จากสมการ (4.7.1) เราจะคำนวณหาค่า $\partial f / \partial x$ และ $\partial f / \partial y$ ที่ทุกๆจุดภาพ จะหาโดยใช้วิธีการใช้เมตริกซ์ขนาด 2×2

$$\begin{array}{cc} f(x,y) & f(x,y+1) \\ f(x+1,y) & f(x+1,y+1) \end{array}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
เมตริกซ์ขนาด 2×2 ที่ใช้คำนวณ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.วิธีการมาส์กแบบนี้เรียกว่า วิธีโรเบิร์ตเกรเดียน (Robert Gradient) เขียนเป็นสมการได้ดังข้างล่างนี้

$$G[f(x,y)] = | f(x,y) - f(x+1,y+1) | + | f(x+1,y) - f(x,y-1) |$$

จะเห็นได้ว่าการประมาณข้างต้น เป็นค่าของเกรเดียนที่เป็นสัดส่วนกับค่าความแตกต่างระดับเทาของจุดภาพบริเวณใกล้เคียงกัน ค่าเกรเดียนจะมีค่ามากบริเวณที่เป็นขอบภาพหรือบริเวณที่มีการเปลี่ยนแปลงระดับความเข้มมากและค่าเกรเดียนจะมีค่าน้อยบริเวณที่มีการเปลี่ยนแปลงระดับเทาน้อย ตารางข้างล่างนี้แสดงการมาส์กเป็นรูปแบบวิธีการต่างๆ โดยค่าที่อยู่ในกรอบสี่เหลี่ยมคือค่าที่กำลังพิจารณา

ถ้าเป็นเมตริกซ์ขนาด 3x3 ดังรูป ข้างล่าง

$$\begin{array}{ccc} X_1 & X_2 & X_3 \\ X_4 & X_5 & X_6 \\ X_7 & X_8 & X_9 \end{array}$$

$$G_x = (X_7 + 2X_8 + X_9) - (X_1 + 2X_2 + X_3)$$

$$G_y = (X_3 + 2X_6 + X_9) - (X_1 + 2X_4 + X_7)$$

H_1

H_2

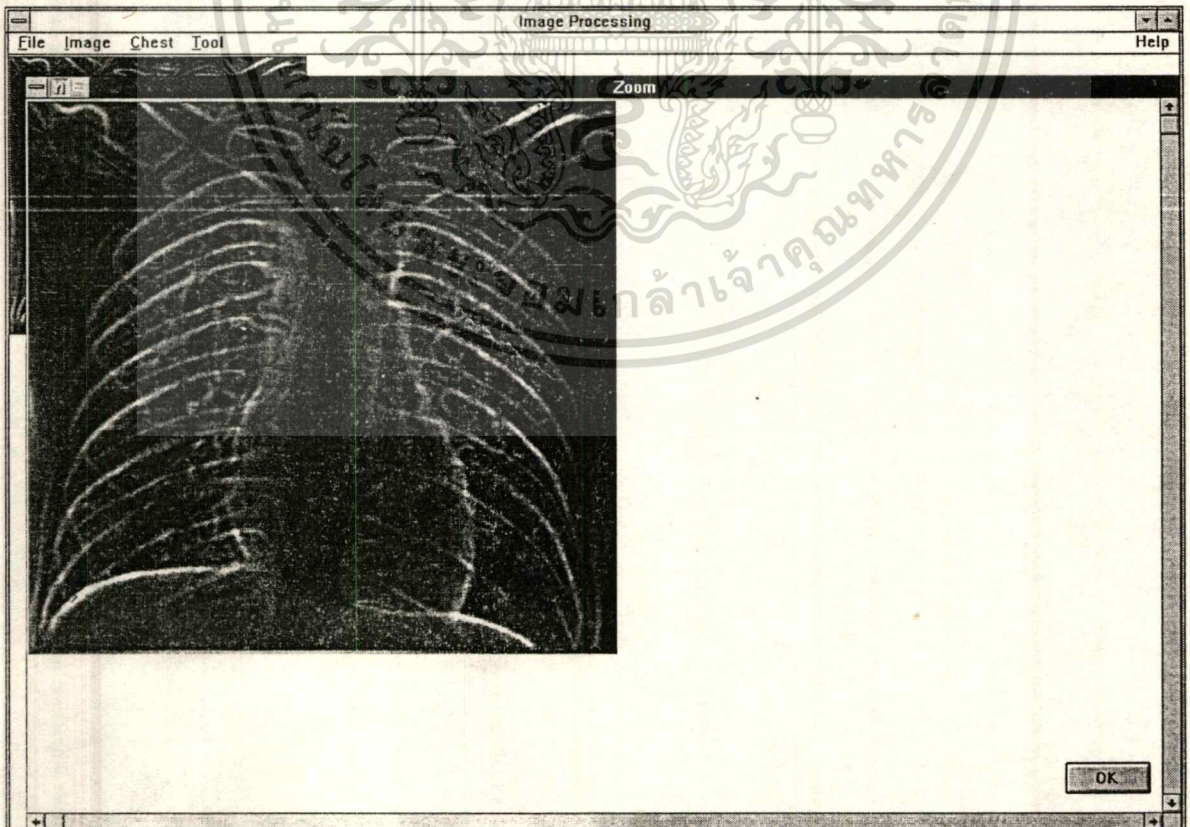
Robert

0	1	1	0
-1	0	0	-1

Sobel

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

ภาพที่ 28



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาพที่ทำการหาขอบภาพด้วยวิธี Sobel ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 การหาค่าเทรชโฮลด์ (Thresholding) ([1] Rafael C. Gonzalez)

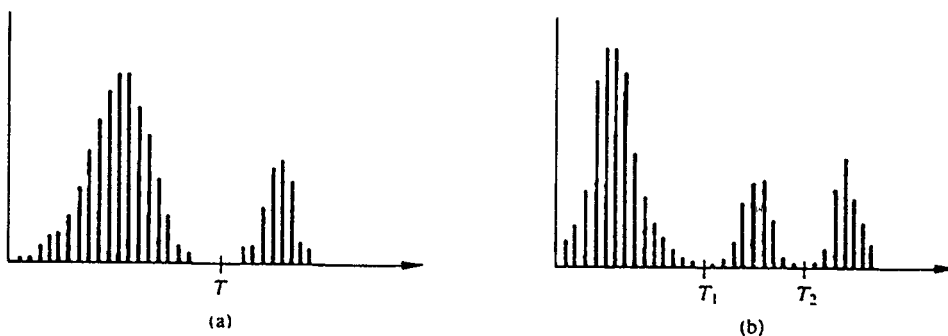
เทรชโฮลด์เป็นวิธีการหนึ่งที่สำคัญในการแยกส่วนภาพ (Segmentation) เทคนิคการหาค่าเทรชโฮลด์จะกล่าวต่อไปดังนี้ สมมุติค่าระดับเทาของฮิสโตแกรมดังรูปที่ 29(a) ของภาพ $f(x,y)$ ที่ประกอบด้วยวัตถุที่สว่างและฉากหลังที่มืด ที่ถูกแบ่งออกเป็นสองส่วนของระดับเทา วิธีการที่จะแยกสองส่วนนี้ออกจากกันโดยใช้ค่าเทรชโฮลด์ T สำหรับทุก ๆ จุด (x,y) ที่ $f(x,y) > T$ จะเรียกจุดเหล่านั้นว่าเป็นวัตถุ นอกจากนั้นเป็นส่วนของฉากหลัง แต่ค่าของฮิสโตแกรมที่มีรูปแบบเป็นดังรูป 29(b) จะแบ่งลักษณะของภาพได้เป็น 3 ส่วนตัวอย่าง สองส่วนที่เป็นวัตถุสว่าง บนฉากหลังที่มืด ดังนั้นสามารถใช้วิธีการเดียวกันในการจำแนกจุด (x,y) เป็นดังนี้ ถ้า $T_1 < f(x,y) \leq T_2$ ให้เป็นวัตถุหนึ่งและถ้า $f(x,y) > T_2$ เป็นอีกวัตถุหนึ่ง และส่วนที่เหลือคือ $f(x,y) \leq T_1$ เป็นส่วนของฉากหลัง ลักษณะแบบนี้เราจะเรียกว่าเทรชโฮลด์หลายระดับ (Multilevel Thresholding) การหาค่าเทรชโฮลด์แบบนี้จะทำให้ยากกว่าเทรชโฮลด์ตัวเดียวที่จะมาแยกวัตถุที่สนใจในที่จะกล่าวถึงเฉพาะค่าเทรชโฮลด์ตัวเดียว สำหรับหลักการในการหาจะเขียนเป็นสมการฟังก์ชัน T ที่มีรูปแบบดังข้างล่างนี้

$$T = T[x,y,P(x,y),f(x,y)] \quad (4.8.1)$$

ที่ $f(x,y)$ เป็นค่าระดับเทาที่จุด (x,y) และ $P(x,y)$ แสดงคุณสมบัติเฉพาะส่วนของจุดนั้นเช่นค่าเฉลี่ยของระดับเทารอบ ๆ จุด (x,y) เราสามารถสร้างภาพเทรชโฮลด์ $g(x,y)$ โดยนิยาม

$$g(x,y) = \begin{cases} 1 & \text{ถ้า } f(x,y) > T \\ 0 & \text{ถ้า } f(x,y) \leq T \end{cases} \quad (4.8.2)$$

ภาพที่ 29



เอกสารนี้เป็นเอกสารที่แสดงฮิสโตแกรมระดับเทา (a) เทรชโฮลด์เดียว (b) หลายเทรชโฮลด์ ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการข้างบนนี้ พบว่าค่าจุดที่ถูกแปลงเป็น 1 เป็นส่วนของวัตถุขณะที่ค่าระดับเทาของจุดที่ถูกแปลงเป็น 0 คือส่วนของภาพที่เป็นฉากหลัง เมื่อค่า T ขึ้นอยู่กับค่า $f(x,y)$ เพียงอย่างเดียวจะเรียกว่าโกลบอลเทรชโฮลด์ (global Threshold) ตัวอย่างเช่นในภาพ 29(a) ถ้าขึ้นอยู่กับ $f(x,y)$ และ $P(x,y)$ แล้วเรียกว่าค่าเทรชโฮลด์นั้นว่าเป็นโลคอลเทรชโฮลด์ (Local Threshold) ถ้าค่าเทรชโฮลด์ขึ้นอยู่กับตำแหน่งสเปเชียล (coordinates spatial) ของ x และ y เรียกว่าไดนามิกเทรชโฮลด์ (dynamic threshold)

4.8.1 วิธีการโกลบอลเทรชโฮลด์ (Global Thresholding Technique)

วิธีการง่าย ๆ ที่มักจะถูกใช้ในการแบ่งแยกส่วนของภาพคือ แบ่งค่าระดับเทาออกเป็นส่วนของระดับเทาและใช้ค่าเทรชโฮลด์เป็นตัวกำหนดขอบเขตของจุดภาพ ที่แสดงส่วนของภาพนั้น ๆ เทคนิคการทำวิธีการนี้คือ สมมติให้ค่าระดับเทาของภาพ $f(x,y)$ มีค่าฮิสโตแกรมแสดงดังรูป 30 (a) เราจะแยกฮิสโตแกรมที่มีจำนวนจุดมากใน $f(x,y)$ ที่มีติดออกจากส่วนที่เหลือที่มีกระจายของจำนวนจุดค่อนข้างจะเท่ากันออกจากกันฮิสโตแกรมดังรูปนี้เป็นฮิสโตแกรมที่มีฉากหลังค่อนข้างจะมีค่าระดับเทาไม่ต่างกันนักเราจะแบ่งภาพนี้ออกเป็น 2 ส่วน โดยใช้ค่าเทรชโฮลด์ T แสดงดังรูป 30 (b) ส่วนของ B_1 จะเป็นฉากหลังของภาพ ขณะที่ B_2 จะเป็นส่วนของวัตถุ ภาพจะถูกสแกนทั้งแนวตั้งและแนวนอนดูการเปลี่ยนแปลงจะระดับเทาของส่วนช่วงระดับเทาหนึ่งไปยังอีกช่วงระดับเทาหนึ่งเมื่อแยกระดับความเข้มออกเป็น 2 ส่วน คือ B_1 และ B_2 เราจะได้จาก

Pass 1 ในแต่ละแถวในภาพ $f(x,y)(x=0,1,...,N-1)$ สร้างแถวที่มีลักษณะเดียวกัน $g(x,y)$ โดยให้มีความสัมพันธ์กันสำหรับ $y=1,2,...,N-1$ ที่ L_E ระดับเทาที่ขอบ, L_B ระดับเทาฉากหลัง

$$g_1(x,y) = \begin{cases} L_E & \text{ถ้าระดับเทาของ } f(x,y) \text{ และ } f(x,y-1) \text{ มีค่าระดับเทาอยู่คนละช่วงความถี่ของระดับเทา } B_1 \text{ และ } B_2 \\ L_B & \text{ที่อื่น ๆ} \end{cases}$$

ที่ L_E และ L_B คือค่าระดับเทาของขอบภาพ และค่าระดับเทาของฉากหลังตามลำดับ

Pass 2 แต่ละหลัก (column) ใน $f(x,y)$ (i.e., $y=0,1,...,N-1$) สร้างคอลัมน์ที่มีลักษณะเดียวกันในภาพ $g_2(x,y)$ ให้มีความสัมพันธ์ดังนี้ $x=1,2,...,N-1$

$$g_2(x,y) = \begin{cases} L_E & \text{ถ้าระดับเทาของ } f(x-1,y) \text{ อยู่กันคนละช่วงความถี่ของระดับเทา } B_1 \text{ และ } B_2 \\ L_B & \text{ที่อื่น ๆ} \end{cases}$$

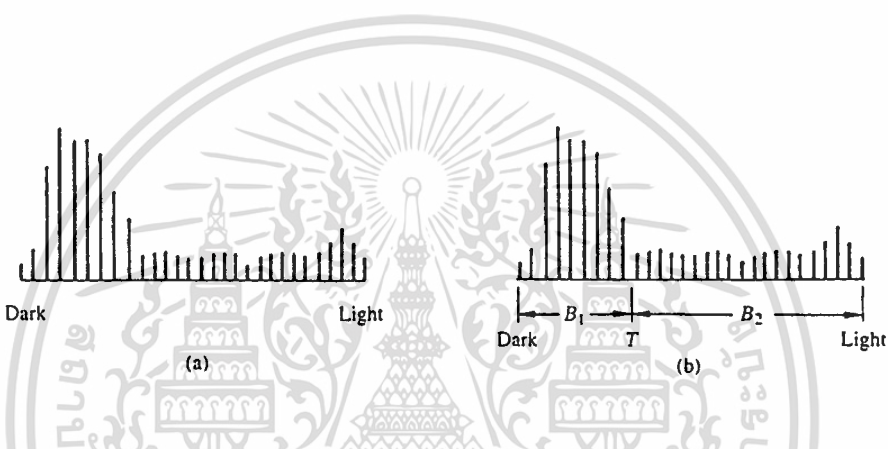
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ในการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะแยกจุดบนภาพที่เป็นขอบวัตถุออกจากฉากหลังโดยใช้สมการต่อไปนี้ สำหรับ

$x,y=0,1,...,N-1$

$$g(x,y) = \begin{cases} L_E & \text{ถ้า } g_1(x,y) \text{ หรือ } g_2(x,y) \text{ ค่าหนึ่งค่าใดมีค่าเท่ากับ } L_E \\ L_B & \text{ที่อื่น ๆ} \end{cases}$$

ภาพที่ 30



ฮิสโตแกรมเทอร์สไฮวด์

4.8.2 การหาค่าเทอร์สไฮวด์โดยอัตโนมัติ

เนื่องจากในหัวข้อต่อไปจะกล่าวถึงการปรับปรุงภาพด้วยการแยกภาพออกเป็น 2 ส่วนคือ วัตถุกับฉากหลัง ดังนั้นในหัวข้อนี้จะกล่าวถึงการแยกภาพด้วยการหาค่าเทอร์สไฮวด์โดยอัตโนมัติ มีอยู่ด้วยกันหลายวิธีด้วยกัน M.IBRAHIM SEZAN [5] ได้เสนอวิธีการหาค่าเทอร์สไฮวด์โดยการตรวจสอบค่าส่วนยอดของฮิสโตแกรม (Peak Detection Algorithm) เนื่องจากวิธีนี้ต้องการตัวแปรที่กำหนดขึ้นก่อนที่ได้มาจากการทดลองแทนค่าดูอยู่หลายตัวแปรจึงทำให้ไม่สะดวกในการใช้งาน ดังนั้นจึงทดลองใช้วิธีการหาค่าเทอร์สไฮวด์ของภาพระดับเทาโดยใช้ค่าเอนโทรปีของฮิสโตแกรม([6] Thierry PUN)แทน เพื่อนำผลไปใช้ในหัวข้อต่อไป สำหรับรายละเอียดของแต่ละวิธีจะกล่าวเป็นหัวข้อดังนี้

4.8.2.1 การหาค่าเทอร์สไฮวด์ของภาพระดับเทาโดยใช้ค่าเอนโทรปีของฮิสโตแกรม

(Grey Level Picture Thresholding using the entropy of the Histogram) ([6] Thierry PUN)

เป็นการหาค่าเทอร์สไฮวด์โดยอัตโนมัติในการแบ่งแยกภาพ ด้วยค่าเอนโทรปีของฮิสโตแกรม หลักการคือการหาค่าสูงสุดของเอนโทรปีก่อนแบ่งภาพเป็นขาวกับดำด้วยค่าเอนโทรปีหลัง

การแปลง ถ้าภาพถูกแยกส่วนออกเป็นวัตถุกับฉากหลังของฮิสโตแกรมที่แบ่งออกเป็นสองส่วน (bimodal) ค่าของเทรลไฮวต์จะอยู่ระหว่างค่าส่วนเงาของทั้งสองส่วนยอด (peak) ในภาพจริง ๆ แล้วการตรวจสอบส่วนเงาที่ต่ำที่สุดของฮิสโตแกรมของทั้งสองส่วนยอดสามารถทำได้ยาก เนื่องจากฮิสโตแกรมอาจจะมีหลายส่วนยอด (peak) หรือไม่มีค่าส่วนยอดเลยก็ได้ สำหรับวิธีการนี้ไม่จำเป็นต้องอาศัยความรู้เกี่ยวกับลักษณะของภาพอื่นนอกจาก ค่าฮิสโตแกรมของระดับเทา

- เอนโทรปีของฮิสโตแกรม

1) นิยาม ให้ค่าจุดในภาพที่มีระดับเทาเท่ากับ n ระดับ $[1, 2, \dots, n]$ N_i จะแสดงจำนวนของจุดที่ระดับเทา i ดังนั้นจำนวนจุดทั้งหมดแสดงโดย N ที่ $N = N_1 + N_2 + \dots + N_n$ ค่าความน่าจะเป็นที่ระดับเทา i คือ $P_i = N_i/N$ จะได้

$$P_i = N_i/N \geq 0, \quad \sum_{i=1}^n P_i = 1 \quad (4.8.2-1)$$

ค่าระดับเทาของฮิสโตแกรมสามารถพิจารณาด้วย n ระดับเทาเป็นอิสระจากภาพนั้นเราสามารถตั้งสมมุติฐานตัวแปรเหล่านี้จะมีค่าสถิติที่เป็นอิสระจากกัน หรือจะกล่าวอีกนัยหนึ่งว่าจำนวนของจุดที่ระดับเทาหนึ่งไม่มีความสัมพันธ์กับจำนวนของจุดของระดับเทาใกล้เคียงเพื่อที่จะทำการหาค่าเทรลไฮวต์ได้ง่ายขึ้น

ค่า information ที่ระดับเทา i ของฮิสโตแกรมเขียนได้ดังนี้

$$I_i = -\log_2(P_i) = -\lg(P_i) \quad (4.8.2-2)$$

และค่าเอนโทรปี ดังสมการข้างล่างนี้

$$H = E(I_i) = - \sum_{i=1}^n P_i \cdot \lg(P_i) \quad (4.8.2-3)$$

หลังจากการทำเทรลไฮวต์ภาพจะถูกแบ่งออกเป็นสองระดับคือขาว (w) กับดำ (b) จำนวนของจุดแต่ละระดับแสดงโดย N'_b และ N'_w ตามลำดับ และ P'_w กับ P'_b แสดงความน่าจะเป็นของระดับขาวกับดำ ซึ่งค่าของขาวกับดำเป็นอิสระต่อกัน ดังนั้นค่าเอนโทรปี ของภาพสองระดับจะเขียนได้ดังข้างล่างนี้

$$H' = -P'_w \lg(P'_w) - P'_b \lg(P'_b) \quad (4.8.2-4.)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เอนโทรปีก่อนและหลัง (A priori and posterior entropies)

ให้ s เป็นค่าของเทรสไฮวด์

$$N'_w = \sum_{i=1}^s N_i \quad , \quad N'_b = \sum_{i=s+1}^n N_i \quad (4.8.2-5)$$

และ

$$P'_w = N'_w / N \quad , \quad P'_b = N'_b / N \quad (4.8.2-6)$$

สามารถกำหนดเป็นค่าเอนโทรปีย่อย ๆ ที่มีสมการคณิตศาสตร์ดังข้างล่าง

$$H_w = - \sum_{i=1}^s P_i \lg(P_i) \quad (4.8.2-7)$$

$$H_b = - \sum_{i=s+1}^n P_i \lg(P_i) \quad (4.8.2-8)$$

$$H'_w = - P'_w \lg(P'_w) \quad (4.8.2-9)$$

$$H'_b = - P'_b \lg(P'_b) \quad (4.8.2-10)$$

สองสมการแรกให้ความหมายแตกต่างจากสองสมการหลังซึ่งใช้ในการวัดปริมาณของ information ก่อน (priori) ที่จะทำเทรสไฮวด์ของจุดที่จะเป็นขาวและดำ โดยใช้ค่าเทรสไฮวด์ s (priori หมายถึง จุดที่ยังไม่ได้มีการแปลงเป็นระดับขาวหรือดำ) สำหรับเอนโทรปี 2 ค่าสุดท้าย H'_w และ H'_b ใช้วัดค่า information ของจุดขาวและดำ หลังจากทำเทรสไฮวด์ จาก (4.8.2-3) และ (4.8.2-4) เราจะได้

$$H = H_w + H_b \quad , \quad H' = H'_w + H'_b \quad (4.8.2-11)$$

จากสมการไม่มีความสัมพันธ์กันโดยตรงระหว่าง H_w , H_b , H'_w , H'_b แต่มีความสัมพันธ์โดยเอกสารอ้างอิงจากสมการที่ (4.8.2-5) และสมการที่ (4.8.2-7) ถึง (4.8.2-10) แทนที่จะใช้ค่าความน่าจะเป็น ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางสถิติของรูปภาพ เราตั้งสมมุติฐานว่าฮิสโตแกรมสามารถของภาพอย่างเดียวยังพอที่จะใช้หาค่าเอนโทรปี

3) การเลือกค่าเอนโทรปีเอนโทรปี

3.1) หลักการ

ในการหาค่าเอนโทรปีที่จะทำภาพเป็นสองระดับเท่านั้น อันดับแรกจะต้องหาค่าเอนโทรปีหลังการแปลงสูงสุด H' โดยไม่คำนึงถึงความสัมพันธ์กับฮิสโตแกรมที่ยังไม่ได้ทำการเอนโทรปี

$$P'_b = P'_w \quad \text{ดังนั้น} \quad N'_b = N'_w \quad (4.8.2-12)$$

จากสมการ จำนวนจุดขาวและดำที่เท่ากันของภาพที่ทำการเอนโทรปี จะให้ค่าเอนโทรปีสูงสุด ภาพที่มีค่าขาวครึ่งหนึ่งและดำครึ่งหนึ่งจะมีค่าเอนโทรปีเหมือนกัน

ค่าเอนโทรปีก่อนทำการเอนโทรปี H สามารถหาได้จากฮิสโตแกรม เราพยายามที่จะกำหนดขอบเขตล่างสำหรับ H' ให้เป็นพารามิเตอร์ในฟังก์ชันของของระดับเทา s ของค่าเอนโทรปี และพยายามทำขอบเขตนี้ให้มีความมากที่สุดจากจุดนี้เราจะได้ผลลัพธ์ของเอนโทรปีก่อนการแปลงสูงสุดจากค่าเอนโทรปีหลังการแปลง H'

ในการทำเอนโทรปี จะแบ่งค่าเอนโทรปีก่อนเอนโทรปีออกเป็น 2 ส่วนคือ H_w และ H_b กำหนดให้ α เป็นค่าตัวแปร

$$H_w = \alpha H, \quad H_b = (1 - \alpha)H, \quad 0 \leq \alpha \leq 1 \quad (4.8.2-13)$$

แสดงอัตราส่วนปริมาณของ information ของภาพก่อนเอนโทรปีในจุดที่จะขาวและดำจากสมการ (4.8.2-7) และ (4.8.2-8) ให้

$$\begin{aligned} \sum_{i=1}^s P_i \lg(P_i) &= \alpha \sum_{i=1}^n P_i \lg(P_i) \\ &= -\alpha H \end{aligned} \quad (4.8.2-14)$$

ขอบเขตบนของสมการทางด้านซ้ายเขียนดังข้างล่าง
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 \sum_{i=1}^s P_i \lg(P_i) &\leq \sum_{i=1}^s P_i \lg(\max\{P_1, \dots, P_s\}) \\
 &= \lg(\max\{P_1, \dots, P_s\}) \cdot \left(\sum_{i=1}^s P_i\right)
 \end{aligned} \quad (4.8.2-15)$$

สำหรับทุก ๆ $i, P_i < 1$ ดังนั้น $\lg(\max P_1 \dots P_s)$ และส่วนซ้ายมือของสมการที่ 15 เป็นลบ เราจะ
ไม่รวมกรณีนี้ที่ $\lg(\max p_1 \dots p_s) = 0$ เพราะว่าจะเป็นการบอกว่าฮิสโตแกรมมีเพียงระดับเทาเดียวที่มี
ค่าเพราะ $\lg(1) = 0$ เมื่อมี P_i ตัวใดเป็น 1 ตัวที่เหลือเป็น 0 ดังนั้น

$$\begin{aligned}
 \sum_{i=1}^s P_i &\leq \frac{\sum_{i=1}^s P_i \lg(P_i)}{\lg(\max\{P_1, \dots, P_s\})} \\
 &= \frac{-\alpha H}{\lg(\max\{P_1, \dots, P_s\})}
 \end{aligned} \quad (4.8.2-16)$$

ในทางคล้าย ๆ กัน

$$\begin{aligned}
 \sum_{i=s+1}^n P_i &\leq \frac{\sum_{i=s+1}^n P_i \lg(P_i)}{\lg(\max\{P_{s+1}, \dots, P_n\})} \\
 &= \frac{-(1-\alpha)H}{\lg(\max\{P_{s+1}, \dots, P_n\})}
 \end{aligned} \quad (4.8.2-17)$$

จากนิยามสมการที่ 4 ได้ค่าเอนโทรปีหลัง (posteriori) เทรสไฮวด์

$$-H' = P'_w \lg(P'_w) + P'_b \lg(P'_b) \quad (4.8.2-18)$$

$$= \left(\sum_{i=1}^s P_i\right) \lg\left(\sum_{i=1}^s P_i\right) + \left(\sum_{i=s+1}^n P_i\right) \lg\left(\sum_{i=s+1}^n P_i\right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยค่าขอบเขตบนใน (4.8.2-16) และ (4.8.2-17) จะได้ว่า

$$H' \leq \frac{-\alpha H \cdot \lg(\sum_{i=1}^s P_i)}{\lg(\max\{P_1, \dots, P_s\})} + \frac{(1 - \alpha) H \cdot \lg(\sum_{i=s+1}^n P_i)}{\lg(\max\{P_{s+1}, \dots, P_n\})} \quad (4.8.2-19)$$

ดังนั้น

$$H' \geq H \left[\frac{-\alpha \cdot \lg(\sum_{i=1}^s P_i)}{\lg(\max\{P_1, \dots, P_s\})} + \frac{(1 - \alpha) \cdot \lg(\sum_{i=s+1}^n P_i)}{\lg(\max\{P_{s+1}, \dots, P_n\})} \right] \quad (4.8.2-20)$$

$$H' \geq Fe(\alpha) \cdot H \quad (4.8.2-21)$$

ที่ $Fe(\alpha)$ เป็นฟังก์ชันที่จะต้องหาค่าของ H' สำหรับอัลกอริทึมที่ใช้ในการหาค่า S ที่ดีที่สุด โดยใช้เอนโทรปีเทรสไฮวต์ จะกลาวดังข้างล่าง (พยายามหาค่าเอนโทรปีก่อนเทรสไฮวต์ เป็นตัวกำหนดเอนโทรปี H')

ภาพที่ 31

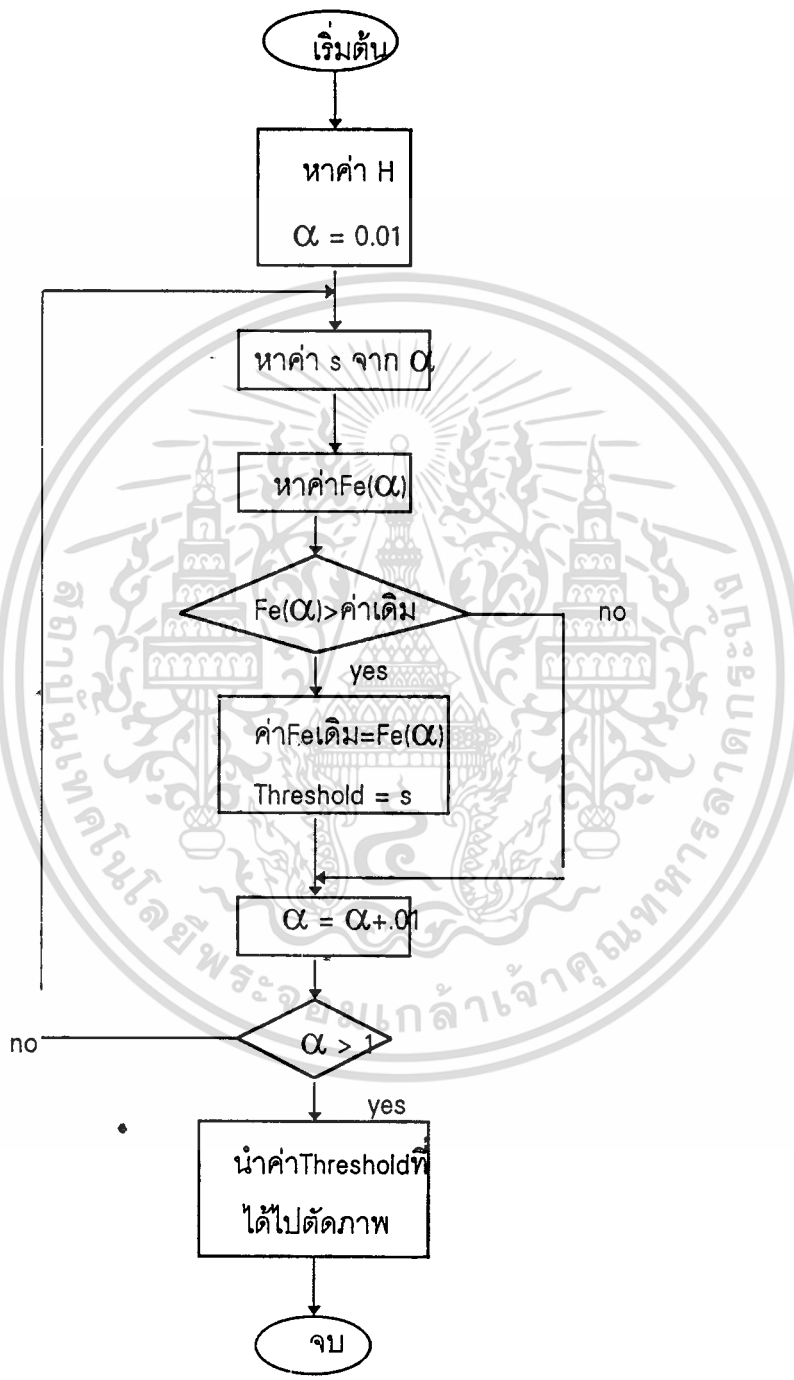


ภาพที่หาค่าเธรสโฮลด์โดยใช้เอนโทรปี

3.3 วิธีการเลือกเอนโทรปีเธรสโฮลด์ (กระทำจน α เข้าใกล้ 1 แล้วนำ $Fe(\alpha)$ ที่สูงที่สุดมาใช้)

- คำนวณค่า α โดยใช้ค่า α เริ่มใกล้ 0
- หาค่า s จากสมการที่ (4.8.2-14)
- คำนวณหาค่า $Fe(\alpha)$ ในสมการที่ (4.8.2-20)
- เพิ่มค่า α และไปที่ขั้นตอน (b) จนกว่า α เข้าใกล้ 1 ที่สุด
- หาค่า α ที่ให้ค่า $Fe(\alpha)$ สูงสุดค่าเอนโทรปีเธรสโฮลด์ (entropy threshold) จะคำนวณได้จากสมการที่ (4.8.2-14)

ภาพที่ 32



แสดงวิธีหาค่าเทรชโฮลด์ด้วยใช้เอนโทรปี

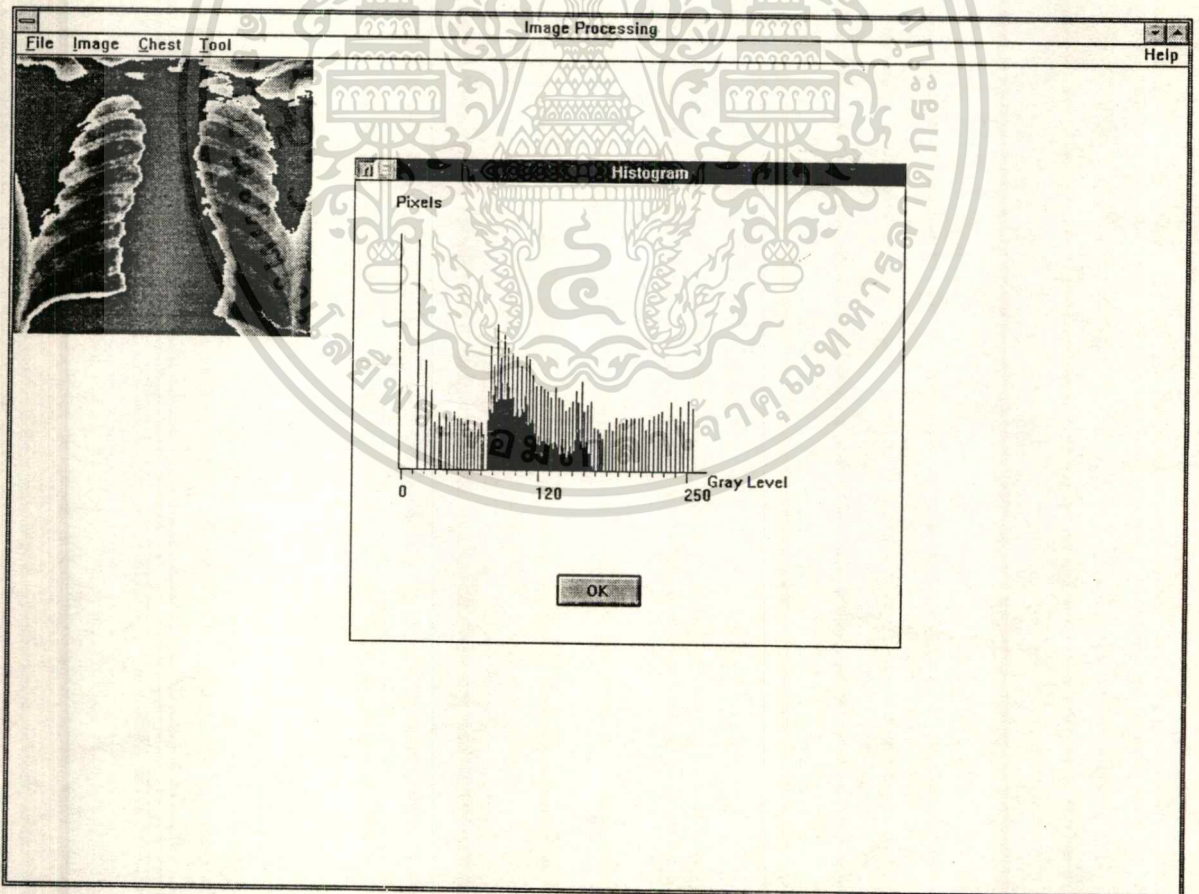
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 การปรับปรุงภาพด้วยวิธีฮิสโตแกรมอีควอไลเซชันแบบแยกส่วนภาพ

(([5] M.Ibrahim Sezan)

เมื่อได้ค่าเทรสโฮลด์แล้วจะทำการแบ่งภาพออกเป็น 2 ส่วน ในส่วนที่มีมืดและส่วนที่สว่างจากภาพที่ใช้ในการทำการทดลองส่วนที่เป็นบริเวณปอดจะเป็นภาพในส่วนที่มีมืดเราจะทำการปรับปรุงภาพในส่วนปอดนี้ด้วยวิธีฮิสโตแกรมอีควอไลเซชันในรูปที่ 33 จะเห็นได้ว่าฮิสโตแกรมของภาพในส่วนที่เป็นปอดจะสามารถกระจายระดับได้มากกว่าการคำนวณกับทั้งภาพ เนื่องจากไม่นำค่าฮิสโตแกรมในส่วนอื่นมาใช้คำนวณด้วย การปรับปรุงภาพในส่วนที่เป็นปอดนี้สามารถใช้วิธีการปรับขยายค่าคอนทราสต์ (Contrast Stretching) สำหรับภาพที่ทำการปรับปรุงแล้วแสดงดังภาพ 33

ภาพที่ 33



แสดงภาพที่ทำการปรับปรุงภาพในส่วนปอดโดยวิธีฮิสโตแกรมอีควอไลเซชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.10 การทำอันซาร์ปมาสก์แบบปรับได้แปรตามค่าระดับเทาของภาพ ([5] M.Ibrahim Sezan)

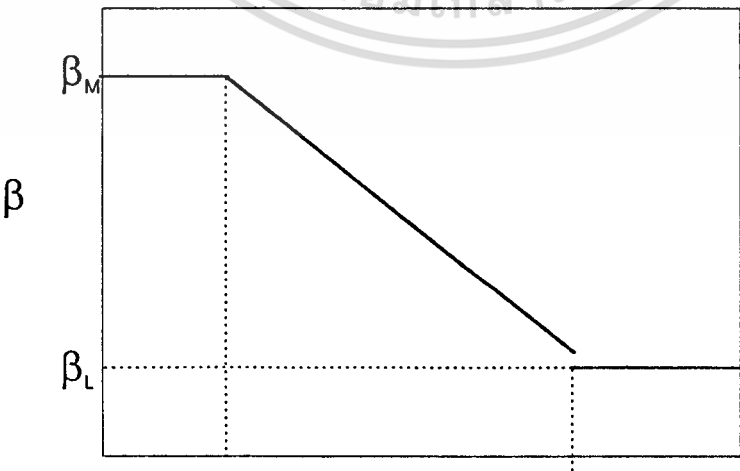
การนำวิธีอันซาร์ปมาสก์ (Unsharp Masking) ในการปรับปรุงภาพที่แยกส่วน นั้นจะใช้วิธีอันซาร์ปมาสก์แบบปรับได้ดังได้กล่าวมาแล้วในบทที่ 4 พารามิเตอร์เน้น (Emphasis Parameter) β จะแปรตามค่าระดับเทาของภาพ และนำค่าเทรสโวด์ที่ได้มาช่วยในการคำนวณ ดังในรูปที่ 34 สำหรับสมการการทำอันซาร์ปมาสก์แสดงในสมการที่ (4.10-1)

$$n'(i,j) = n(i,j) + \beta(n)[n(i,j) - \bar{n}(i,j)] \tag{ 4.10-1 }$$

- ที่ $n'(i,j)$ = ค่าระดับเทาที่จุด i,j หลังการคำนวณ
- $n(i,j)$ = ค่าระดับเทาที่จุด i,j
- $\beta(n)$ = ค่าพารามิเตอร์เน้นแปรตามค่าระดับเทาของภาพ
- $\bar{n}(i,j)$ = ค่าเฉลี่ยรอบจุด i,j

ในหัวข้อนี้จะนำค่าเทรสโวด์ที่ได้มาแยกส่วนอวัยวะส่วนที่เป็นปกติกับอวัยวะส่วนอื่น และจะนำค่านี้มาเป็นตัวกำหนดค่าของตัวแปร β ที่จะมาเน้นขอบภาพ จะกำหนดให้พารามิเตอร์นี้มีค่ามากในส่วนที่มีดของภาพเพื่อที่พยายามเห็นรายละเอียดภาพในส่วนนี้จึงต้องให้มีค่ามากสำหรับรูปแบบของการเปลี่ยนแปลงค่าพารามิเตอร์ $\beta(n)$ ในภาพที่ 34

ภาพที่ 34

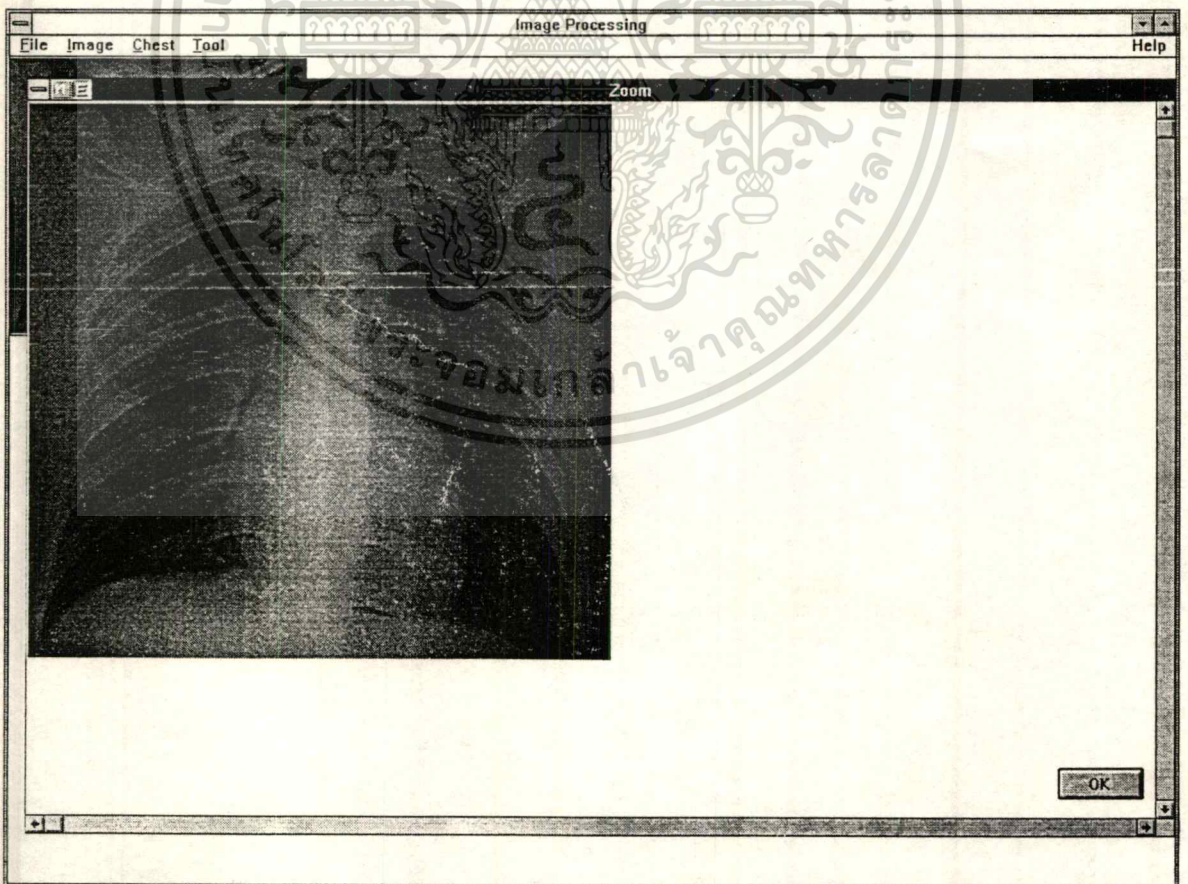


รูปแบบของพารามิเตอร์เน้น (Emphasis Parameter) จะมีค่ามากในส่วนที่มีมืด ($n \leq n_t$) และมีค่าน้อยในส่วนที่สว่าง ($n > n_t$) ดังนั้น $\beta_M > \beta_L$ พารามิเตอร์ a จะเป็นค่าระดับเทาที่ใช้ในการคำนวณหาค่า β

ค่า n_t คือค่าเทรชโฮลด์ที่ใช้แบ่งแยกภาพ ค่าตัวแปร a ($a < n_t$) เป็นระดับเทาที่ต้องการหา ค่า $\beta(n)$ ที่อยู่ในช่วง β_M และ β_L สามารถหาดังสมการ (4.10-2)

$$\beta(n) = \begin{cases} \beta_M & , 0 \leq n \leq a \\ -n(\beta_M - \beta_L)/(n_t - a) + (n_t - a)/(n_t - a) \beta_M - a \beta_L & , a \leq n \leq n_t \\ \beta_L & , n > n_t \end{cases} \quad (4.10-2)$$

ภาพที่ 35



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพในรูป 4.10-2 เมื่อเปรียบเทียบกับภาพจากวิธีการปรับภาพให้คมชัดในหัวข้อ
4.6.1 แล้วภาพที่ได้นี้จะมีลักษณะไม่หยابเหมือนกับในภาพที่ 27

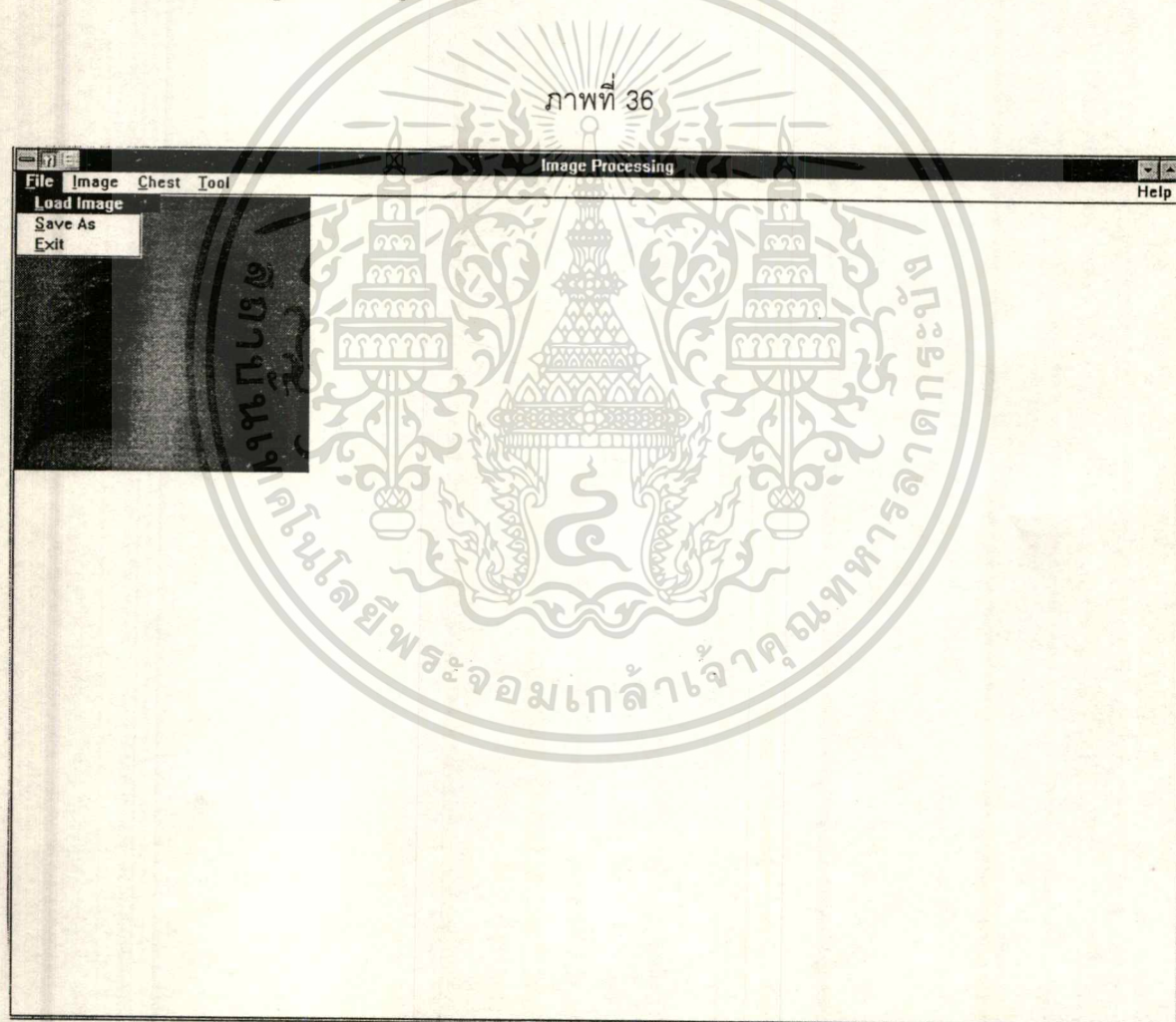


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วิธีการใช้งาน

จากระบบงานเดิมจะเข้าสู่โปรแกรมประมวลผลภาพจากจอภาพ ในภาพที่ 3 (b) แสดงหน้าจอที่จะเข้าสู่การประมวลผลข้อมูลภาพ โดยเลือกกดปุ่ม 'Process' ที่จอภาพระบบจะเข้าสู่การประมวลผลข้อมูลภาพ ดังรูปข้างล่าง



แสดงจอภาพการประมวลผลภาพ

จากภาพที่ 36 เมนูจะแบ่งออกเป็น 4 หัวข้อหลักคือ 'File' 'Image' 'Chest' 'Tool' ซึ่งจะ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
กล่าวถึงวิธีการใช้งานของเมนูแต่ละเมนูต่อไป
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

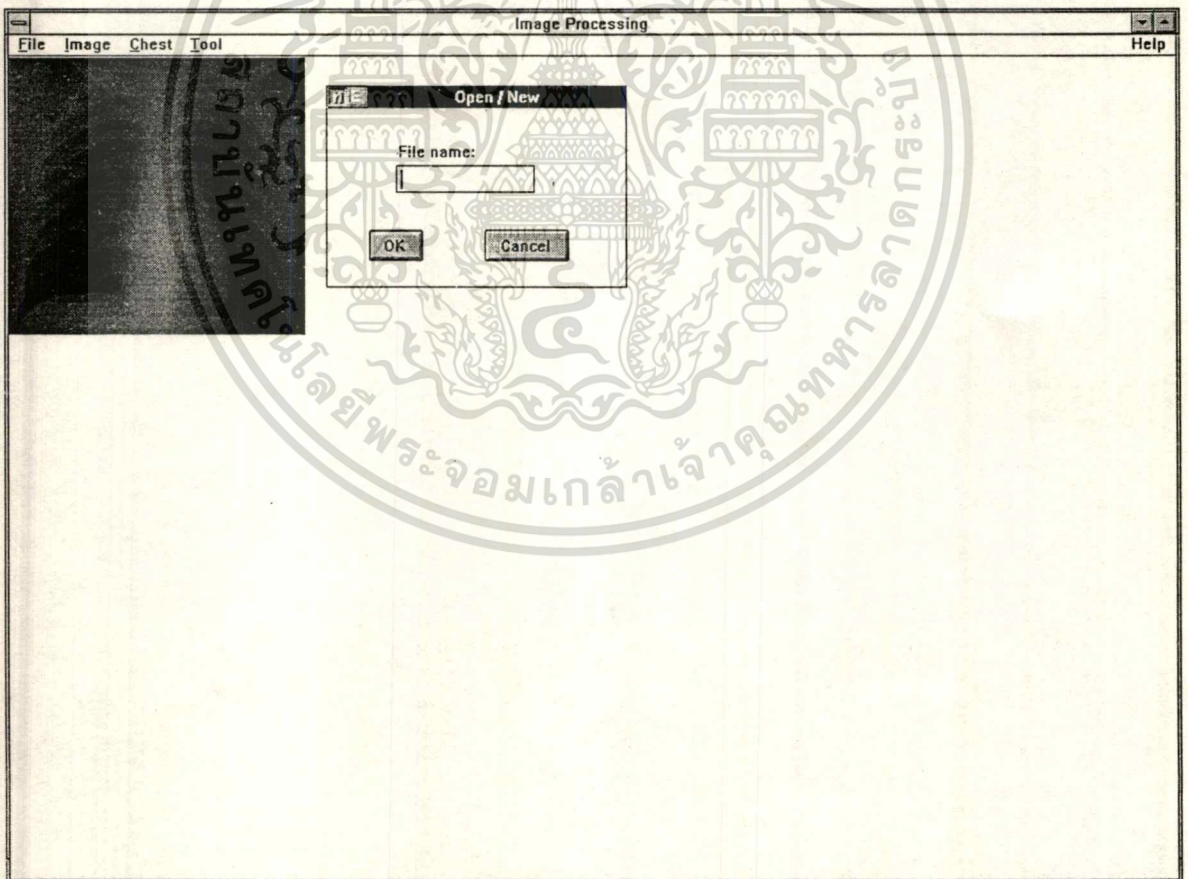
เมื่อผู้ใช้ระบบเลือกเมนูหลักในหัวข้อ 'File' ในภาพที่ 36 จะประกอบด้วยหัวข้อย่อย 3 หัวข้อด้วยกันดังจะกล่าวเป็นหัวข้อข้างล่างนี้

- 'Load Image' จะแสดงภาพต้นแบบก่อนที่จะมีการปรับปรุงภาพ เมื่อผู้ใช้ได้ทำการปรับปรุงภาพด้วยวิธีต่าง ๆ แล้ว ต้องการที่จะแสดงภาพเดิมก่อนปรับปรุง

- 'Save As' เมื่อผู้ใช้ต้องการบันทึกภาพที่แสดงในหน้าจอปัจจุบัน ออกมาเป็นแฟ้มข้อมูล .BMP โดยระบบจะให้ระบุชื่อแฟ้มข้อมูลที่ต้องการ ดังแสดงในภาพที่ 37

- 'Exit' ออกจากจอภาพการประมวลผลภาพ กลับไปที่หน้าจอในภาพที่ 3(b)

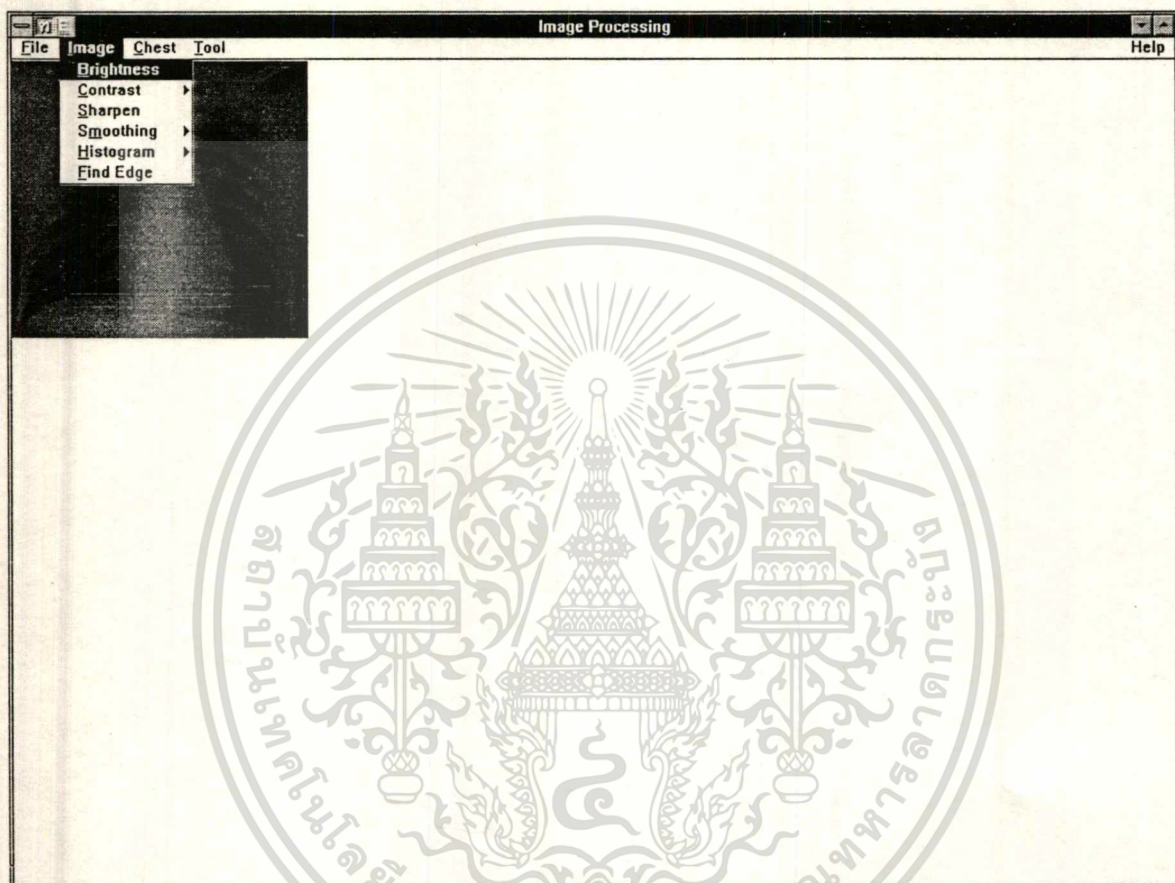
ภาพที่ 37



แสดงการบันทึกข้อมูลภาพจากจอภาพลงแฟ้มข้อมูล .BMP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 38



แสดงรายการจากเมนู 'Image'

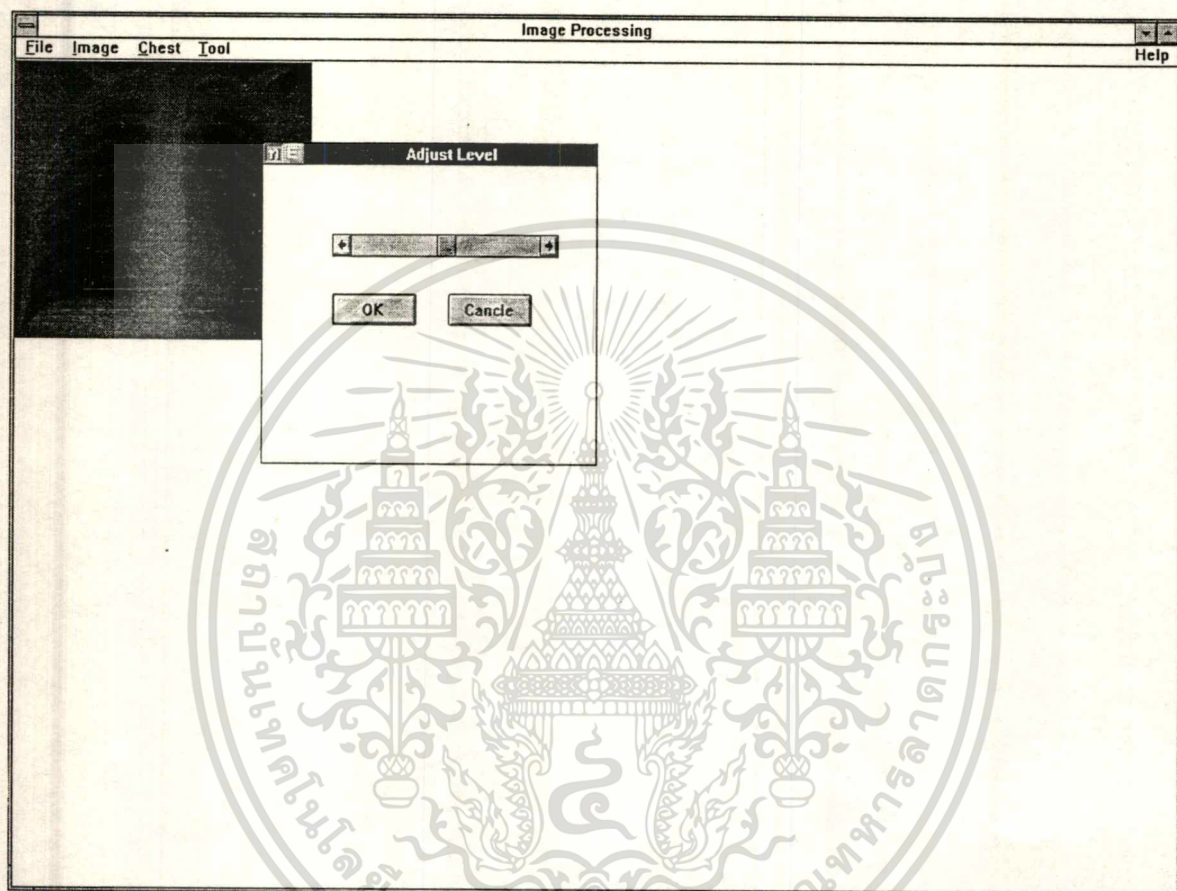
เมื่อผู้ใช้ระบบเลือกเมนู 'Image' จะมีหัวข้อย่อยในการทำงาน 5 หัวข้อดังนี้

- 'Brightness' เมื่อผู้ใช้ระบบต้องการปรับความสว่างของภาพ โดยสามารถกำหนดระดับการปรับได้ แสดงในภาพที่ 39 สำหรับผลลัพธ์ของวิธีนี้กล่าวไว้แล้วในหัวข้อ 4.2
- 'Contrast' เมื่อผู้ใช้ระบบต้องการปรับคอนทราสต์ของภาพ ผลลัพธ์ในหัวข้อ 4.3
- 'Sharpen' เมื่อผู้ใช้ระบบต้องการปรับความคมชัดของภาพ ผลลัพธ์ในหัวข้อ 4.6
- 'Smoothing' เมื่อผู้ใช้ระบบต้องการขจัดสัญญาณรบกวนของภาพ ผลลัพธ์ในหัวข้อ 4.5
- 'Histogram' เมื่อผู้ใช้ระบบต้องการปรับปรุงภาพด้วยวิธีฮิสโตแกรมมิกวไรเซชัน ผลลัพธ์ในหัวข้อ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

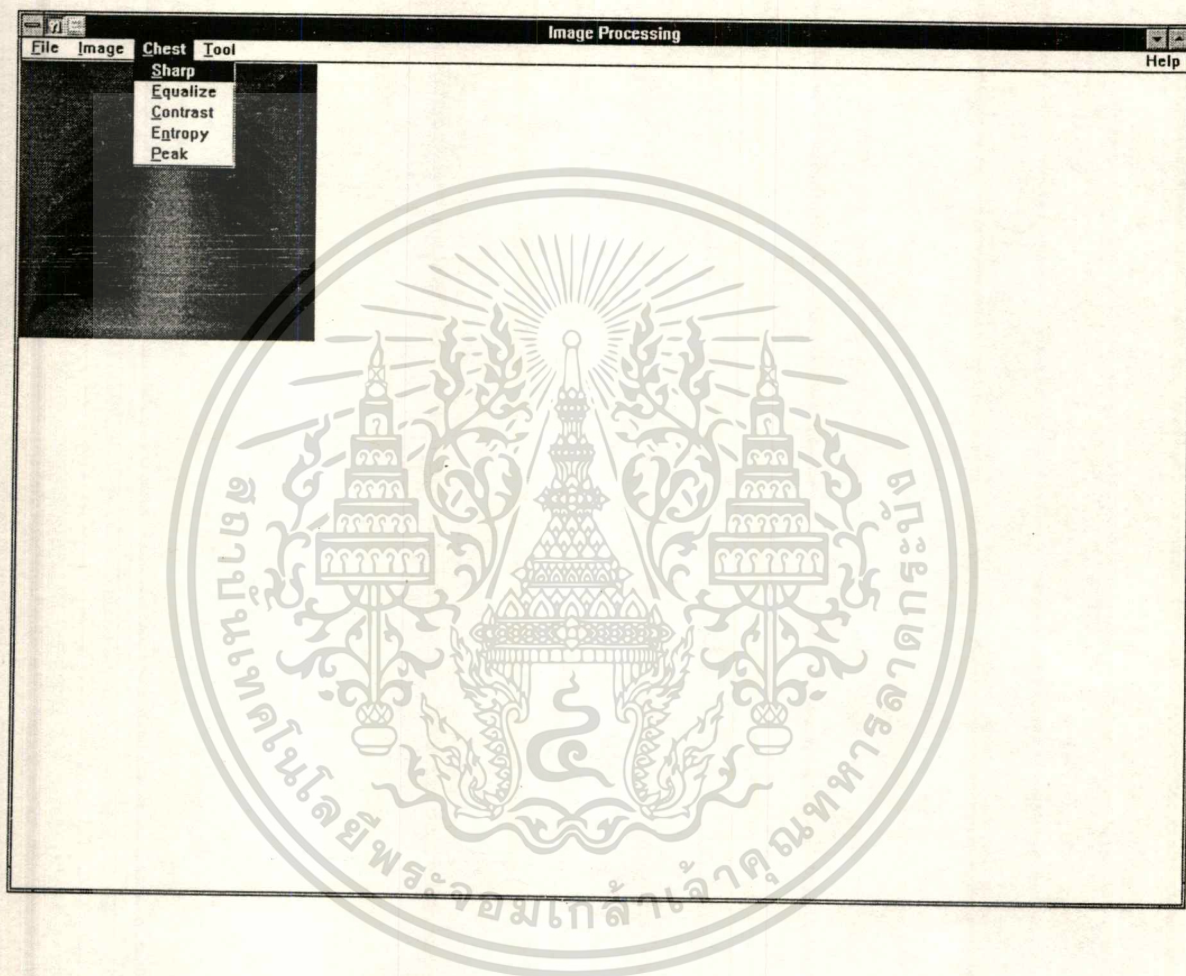
ภาพที่ 39



แสดงการปรับความสว่างของภาพ

จากภาพที่ 39 ผู้ใช้ระบบสามารถปรับระดับความสว่างของภาพให้ขึ้นลง เมื่อผู้ใช้เลือกการปรับความสว่างของภาพ จะปรากฏจอภาพเล็กๆ ขึ้นขึ้นมา ถ้าผู้ใช้ต้องการเพิ่มความสว่างของภาพก็เลื่อนแถบเลื่อนไปทางขวา ถ้าต้องการลดความสว่างของภาพก็เลื่อนแถบเลื่อนไปทางซ้าย

ภาพที่ 40



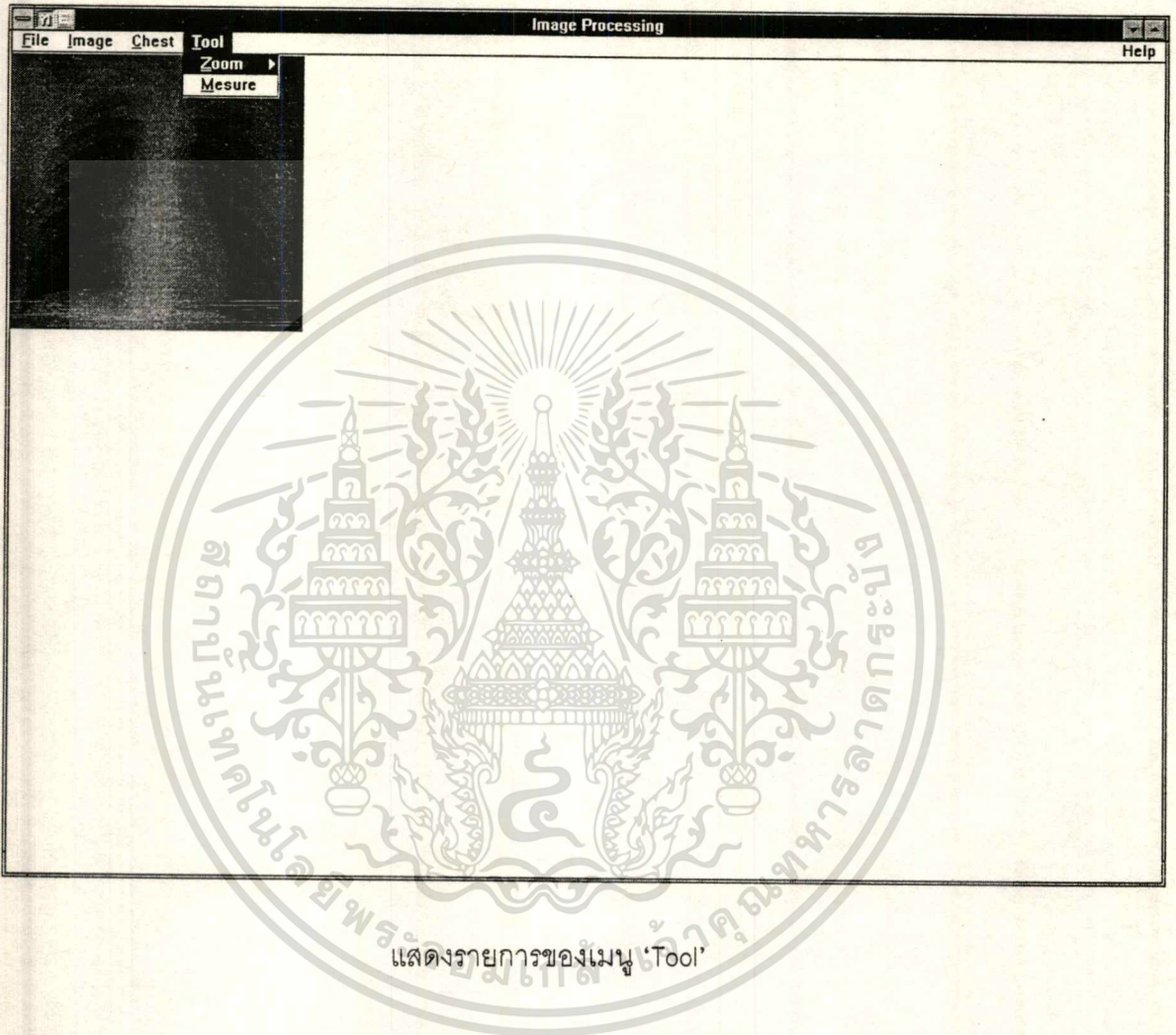
แสดงรายการเมนู 'Chest'

จากรูปที่ 40 เมื่อผู้ใช้เลือกหัวข้อ 'Chest' จะแสดงรายการหัวข้อต่างๆ ซึ่งหัวข้อนี้ใช้ในการปรับปรุงภาพเอ็กเรย์ทรวงอก

- 'Sharp' เป็นการปรับปรุงภาพให้คมชัด รายละเอียดและผลลัพธ์ กล่าวไว้ในหัวข้อ 4.10
- 'Equalize' เป็นการปรับปรุงฮิสโตแกรมของภาพ รายละเอียดและผลลัพธ์ กล่าวไว้ในหัวข้อ 4.9
- 'Entropy' เป็นการแยกภาพออกเป็น 2 ส่วนด้วยวิธีเอนโทรปี รายละเอียดและผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
กล่าวไว้ในหัวข้อ 4.8
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 41



สำหรับเมนู 'Tool' ในระบบจะมีการทำงานเพียงหัวข้อเดียว คือ การขยายภาพ (Zoom)
รายละเอียดและผลลัพธ์ของวิธีนี้แสดงในหัวข้อ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 สรุปงานวิจัย

ในการจัดทำโปรแกรมฐานข้อมูลภาพทางการแพทย์ที่สามารถนำเอาอัลกอริทึมต่างๆ ทางการประมวลผลข้อมูลภาพมาใช้ทำให้แพทย์ผู้ทำการวินิจฉัยภาพสามารถปรับปรุงภาพที่มีคุณสมบัติไม่ดีให้สามารถมองเห็นรายละเอียดส่วนใหญ่ของภาพได้ดีขึ้น ซึ่งการต้องการในการปรับปรุงภาพส่วนใหญ่แล้วภาพจะมีปัญหาทางด้านค่าคอนทราสต์ของภาพต่ำ ทำให้ไม่สามารถเห็นรายละเอียดของภาพได้ชัดเจนเท่าที่ควรหรือภาพที่ได้อยู่ในโทนที่ค่อนข้างมืดดังนั้นจุดประสงค์หลักคือพยายามที่จะหาอัลกอริทึมที่สามารถปรับปรุงภาพให้มีคอนทราสต์เพิ่มขึ้นและสามารถวินิจฉัยภาพได้ดีขึ้น ในบทที่ 4 นั้นได้กล่าวถึงการทดลองการทำฮิสโตแกรมอีควอไลเซชัน โดยแยกภาพออกเป็นสองส่วนด้วยค่าเทรชโฮลด์ และใช้ฮิสโตแกรมในแต่ละส่วนของภาพมาสร้างฟังก์ชันในการแปลงเพื่อที่จะปรับปรุงเฉพาะในส่วนที่เป็นปอดจากผลการทดลองในภาพที่ 33 จะทำการปรับปรุงเฉพาะในส่วนที่เป็นปอดเพื่อดูผลของวิธีการจะเห็นว่าในส่วนที่เป็นปอดจะเห็นรายละเอียดได้ดียิ่งขึ้นและจากฮิสโตแกรมของภาพจะเห็นได้ว่าค่าระดับเทาในส่วนที่เป็นปอดสามารถกระจายไปอยู่ในช่วง 0-255 ได้ทั้งหมด นอกจากนั้นยังได้ทำการทดลองใช้วิธีอันซาร์ปมาส์กแบบปรับได้ ภาพที่ได้จากวิธีการนี้จะเห็นรายละเอียดภาพในส่วนที่มืดได้ดีขึ้น

เนื่องจากในภาพรังสีเอ็กซเรย์บางชนิดมีความละเอียดสูงจึงไม่สามารถที่จะนำมาจัดเก็บได้ด้วยอย่างเช่น ภาพรังสีเอ็กซเรย์ทรวงอก แพทย์ผู้ทำการวินิจฉัยต้องการความละเอียดที่ 2048 x 2048 จุดภาพขึ้นไป ทำให้ไม่สามารถจัดหาอุปกรณ์การถ่ายภาพชนิดนี้ได้เพราะมีราคาค่อนข้างสูง ดังนั้นภาพที่ใช้ในบทที่ 4 จึงเป็นเพียงการนำตัวอย่างภาพมาทดลอง แต่ในปัจจุบันเทคโนโลยีทางด้านนี้ได้พัฒนาไปเร็วมาก ภายในเวลาอันใกล้นี้อุปกรณ์เหล่านี้จะมีราคาถูกลงและสามารถซื้อหามาใช้ได้

จากการนำเอาอัลกอริทึมต่างๆมาใช้ในการทำงาน สามารถที่จะให้ภาพที่มีคุณสมบัติดีขึ้นกว่าเดิม แต่ในการที่จะวัดว่าภาพที่ได้นั้นสามารถให้ความเที่ยงตรงในการวินิจฉัยของแพทย์นั้นยังทำได้ยาก อาจจะต้องอาศัยจากการทำการทดลองแล้วนำแพทย์ที่วินิจฉัยภาพหลายๆคน มาทดลองดูผลลัพธ์ที่ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

เอกสารอ้างอิง

- 1) Digital Image Processing , Rafael C. Gonzalez/Paul Wintz , Second Edition , Addison-Wesley Publishing Company,1987
- 2) Introduction Computer Vision Image Processing , Adrian Low ,McGrawHill,1991
- 3) Fundamental of Digital Image Processing , Anil K. Jain , Prentice Hall,1989
- 4) Digital Image Processing , William K. Pratt , Second Edition , A Wiley-Interscience Publication John Wiley & Son ,1991
- 5) Automic Anatomically Selective Image Enhancement in Digital Chest Radiography , M. Ibrahim Sezan,IEEE Transaction of Medical Imaging ,Vol. 8, No.2, June 1989 , p. 154-162
- 6) A New Method For Gray-Level Picture Thresholding Using The Entropy of Histogram , Thierry PUN , Signal Processing 2 (1980) , p 223-237
- 7) Regionally Adaptive Histogram Equalization of the Chest , Robert H. Sherrier and G. A. Johnson,IEEE Transaction of Medical Imaging ,Vol. MI-6 , No. 1, March 1987 , p 1-7
- 8) บริสุทธิ์ พันธมา , กวิน สนธิเพิ่มพูน, ดร. ไพรัช รัชชพงษ์, “การปรับปรุงภาพรังสีเอ็กซ (Enhancement of Radiography) “, วิทยานิพนธ์ สำหรับวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า ลาดกระบัง พ.ศ. 2532
- 9) อาโมทย์ สมบูรณ์แก้ว , ดร. พุศศักดิ์ ชิวสุวิทย์ , “การปรับปรุงรายละเอียดของขอบในภาพสี ด้วยการแปลงภาพผลลบที่ถูกทำให้เรียบ (Edge Enhancement in Color Image Using Subtracted Smoothing Image)”, วิทยานิพนธ์ สำหรับวิศวกรรมศาสตรมหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า ลาดกระบัง พ.ศ. 2537

ภาคผนวก ข.
บทความที่ตีพิมพ์ในวารสาร

- 1) มนัส นามะสนธิ , กวิน สนธิเพิ่มพูน , “ การสร้างและพัฒนาโปรแกรมการจัดการฐานข้อมูลและรูปภาพทางการแพทย์” ,วารสาร สาร NECTEC, ฉบับที่ 1 ปี พ.ศ. 2537



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*****

// PROGRAM : IMAGEPROC
*****

#include <windows.h>
#include <memory.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <io.h>
#include <conio.h>
#include <math.h>

#include "imagepro.h"

#define PALETTE_SIZE 256
#define RGB_RED 0
#define RGB_GREEN 1
#define RGB_BLUE 2
#define RGB_SIZE 3

typedef struct {
    int width,depth,bytes,bits;
    int background;
    unsigned char palette[768];
    int (*setup)();
    int (*closedown)();
} FILEINFO;

typedef struct {
    char id[2];
    long filesize;
    int reserved[2];
    long headersize;
    long infoSize;
    long width;
    long depth;
    int biPlanes;
    int bits;
    long biCompression;
    long biSizeImage;
    long biXPelsPerMeter;
    long biYPelsPerMeter;
    long biClrUsed;
    long biClrImportant;
} BMPHEAD;

FILEINFO fi;

```

```

HWND hInst;
HWND hScroll0;
FARPROC lpScroll0Standard;
BOOL FlagBright;
int test =0 ;
FILE *fp,*fp2;
LOGPALETTE *pPal;
HPALETTE hpal = NULL;
PAINTSTRUCT PtStr;
WORD nNumColors;
BMPHEAD bmp;
// Allocate Memory for Image
HGLOBAL hglb1,hglb2,hglb3,hglb4,hglb5;
unsigned char far *buffer1;
unsigned char far *buffer2;
unsigned char far *buffer3;
unsigned char far *buffer4;
unsigned char far *buffer5;
int RowPerPage,TotalPage,RowRemainder;
HGLOBAL handle1;
unsigned char far *image;
int Position;
// for scroll bar

int FirstFlag = 0;

long FAR PASCAL WndProc (HWND hWnd, WORD iMessage,
                        WORD wParam, LONG lParam);

long FAR PASCAL Scroll0Proc(HWND,WORD,WORD,WORD,WORD);
void colorpalette(HWND hWnd,unsigned char *p);
void ChangeFormat(HWND);
void WriteBmp(HWND hWnd);
void AllocMem(HWND);
void ReadMem(unsigned char far *,int,int);
void WriteMem(unsigned char far *,int,int);
void LoadImage(void);
void Brightness(int);
void Contrast(HWND,int,unsigned char *);
void AutoContrast(void);
void Sharp(void);
void Smooth(void);
void Weight(void);
void Edge(void);
void HistEqualize(void);
int Entropy(void);
void Write2Level(void);
int CalEmphasis(int gray,int Threshold);
void AnatomicChest(void);

```

```

void AnatomicEqualize(void);
BOOL FAR PASCAL OpenNewDlgProc(HWND,unsigned,WORD,LONG);
BOOL FAR PASCAL ScrollDlgProc(HWND,unsigned,WORD,LONG);
BOOL FAR PASCAL GraphDlgProc(HWND,unsigned,WORD,LONG);
BOOL FAR PASCAL ZoomDlgProc(HWND,unsigned,WORD,LONG);

static char szNameText[15];

int PASCAL WinMain (HANDLE hInstance, HANDLE hPrevInstance,
                    LPSTR lpszCmdParam, int nCmdShow)
{
    HWND hWnd;
    MSG Message;
    WNDCLASS WndClass;

    if (hPrevInstance)
    {
        memset(&WndClass,0x00,sizeof(WNDCLASS));
        WndClass.cbClsExtra = 0;
        WndClass.cbWndExtra = 0;
        WndClass.hbrBackground = GetStockObject(WHITE_BRUSH);
        WndClass.hCursor = LoadCursor(NULL, IDC_ARROW);
        WndClass.hIcon = LoadIcon (NULL, "END");
        WndClass.hInstance = hInstance;
        WndClass.lpfnWndProc = WndProc;
        WndClass.lpszClassName = "BRIGHT";
        WndClass.lpszMenuName = "MENU";
        WndClass.style = CS_HREDRAW | CS_VREDRAW;

        RegisterClass (&WndClass);
    }

    hInst = hInstance;
    hWnd = CreateWindow ("BRIGHT",          /* class name */
                        "Image Processing",  /* Caption */
                        WS_OVERLAPPEDWINDOW, /* Style */
                        0,                   /* x position */
                        0,                   /* y position */
                        1024,                /* cx - size */
                        768,                /* cy - size */
                        NULL,                /* Parent window */
                        NULL,                /* Menu */
                        hInstance,           /* Program Instance */
                        NULL);               /* Parameters */

    ShowWindow (hWnd, nCmdShow);

    while (GetMessage (&Message, 0, 0, 0))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        TranslateMessage(&Message);
        DispatchMessage(&Message);
    }

    return Message.wParam;
}

/*****
 *      Window Procedure: WndProc      */
*****/

long FAR PASCAL WndProc (HWND hWnd, WORD iMessage,
                        WORD wParam, LONG lParam)
{

    //PAINTSTRUCT PtStr;
    int i,x,y;

    BYTE red,green,blue;
    char str[40];

    char ch;
    int maxx,maxy,whchunk;
    double NumChunk;
    int xScreen,yScreen,bPos;
    // xms
    static FARPROC lpScrollProc;
    unsigned char color;
    float value;
    int j,n,alloc;
    long off; // offset of target

    FARPROC lpprocDia;
    //static HANDLE hInstance;

    switch (iMessage)
    {
        case WM_COMMAND:
            switch(wParam)
            {
                case ID_LOADIMAGE:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return 0;

    case ID_SAVE :
        memset(szNameText, '\0', 15);
        lpprocDia = MakeProcInstance(OpenNewDlgProc, hInst);
        if (DialogBox(hInst, "OPENNEW", hWnd, lpprocDia)) {
            WriteBmp(hWnd);
        }
        FreeProcInstance(lpprocDia);
        break;

    case ID_EXIT:
        SendMessage(hWnd, WM_CLOSE, 0, 0L);
        return 0;

    case ID_BRIGHT :
        lpprocDia = MakeProcInstance(ScrollDlgProc, hInst);
        if (DialogBox(hInst, "SCRLBOX", hWnd, lpprocDia)) {
            Brightness(Position);
        }
        return 0;

    case ID_CONTRAST1 :
        lpprocDia = MakeProcInstance(ScrollDlgProc, hInst);
        if (DialogBox(hInst, "SCRLBOX", hWnd, lpprocDia)) {
            Contrast(hWnd, Position, fi.palette);
        }
        return 0;

    case ID_CONTRAST2 :
        UpdateWindow(hWnd);
        AutoContrast();

return 0;

    case ID_GRAPH :
        lpprocDia = MakeProcInstance(GraphDlgProc, hInst);
        DialogBox(hInst, "GRAPH", hWnd, lpprocDia);

return 0;

    case ID_SHARP :
        UpdateWindow(hWnd);
        Sharp();
        return 0;

    case ID_AVERAGE :
        UpdateWindow(hWnd);
        Smooth();
        return 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case ID_WEIGHT :
    UpdateWindow(hWnd);
    Weight();
    return 0;

case ID_HISTEQ :
    UpdateWindow(hWnd);
    HistEqualize();
    return 0;

case ID_ADHIST :
    UpdateWindow(hWnd);

// AdaptiveHist();
return 0;

case ID_EDGE :
    UpdateWindow(hWnd);
    Edge();
    return 0;

case ID_ENTROPY :
    UpdateWindow(hWnd);
    Write2Level();
    return 0;

case ID_CHEST1 :
    UpdateWindow(hWnd);
    AnatomicChest();
    return 0;

case ID_CHEST2 :
    UpdateWindow(hWnd);
    AnatomicEqualize();
    return 0;

case ID_ZOOM2 ;
    lpprocDia = MakeProcInstance(ZoomDlgProc,hInst);
    DialogBox(hInst,"ZOOM",hWnd,lpprocDia);

return 0;

case ID_ABOUT :
    MessageBox(hWnd,
        "Sample Application".
        "Windows Graphics",
        MB_ICONINFORMATIONMB_OK);

return 0;
}

```

เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case WM_PAINT:
    hDC = BeginPaint(hWnd; &PtStr);

    ///

    if (FirstFlag ==0){
        ChangeFormat(hWnd);

FirstFlag = 1;
    }

//*****

    FlagBright = 0;

    pPal = (LOGPALETTE*) LocalAlloc(LPTR, sizeof(LOGPALETTE)+nNumColors*sizeof(PALETTEENTRY));
    if (!pPal){
        TextOut(hDC,10,10,"Allocate pPal Failed ",21);

exit(1);
    }

    pPal->palNumEntries = nNumColors;
pPal->palVersion = 0x300;
    for (i=0;i<nNumColors;i++){
        pPal->palPalEntry[i].peRed   = fi.palette[(i*3)];
        pPal->palPalEntry[i].peGreen = fi.palette[(i*3)+1];
        pPal->palPalEntry[i].peBlue  = fi.palette[(i*3)+2];
        pPal->palPalEntry[i].peFlags = (BYTE) 0;
    }
    hpal = CreatePalette(pPal);
    LocalFree((HLOCAL) pPal);
    hDC = GetDC(hWnd);
    SelectPalette(hDC,hpal,0);
    RealizePalette(hDC);

    for (y=0;y<fi.depth;y++){
ReadMem(image,fi.width,y);

        for (x=0;x<fi.width;x++){

            color = *(image+x);

            SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(color));

        }

    }

    EndPaint(hWnd, &PtStr);

    return 0;

case WM_DESTROY:
    PostQuitMessage(0);
    SetWindowLong(hScroll0,GWL_WNDPROC,(DWORD)lpScroll0Standard);
    FreeProcInstance(lpScroll0Proc);
    DeleteObject(hpal);
    GlobalUnlock(hglb1);
    GlobalFree(hglb1);

```

```

GlobalUnlock(hglb2);
GlobalFree(hglb2);
GlobalUnlock(hglb3);
GlobalFree(hglb3);
GlobalUnlock(handle1);
GlobalFree(handle1);

fclose(fp);
fclose(fp2);
return 0;

default:
return(DefWindowProc(hWnd, iMessage, wParam,
lParam));
}

//return 0;
}

void colorpalette(HWND hWnd,unsigned char *p)
{
int i,x,y;
unsigned char color;

for (i=0;i<256;i++){
pPal->palPalEntry[i].pcRed = *(p+i*3);
pPal->palPalEntry[i].pcGreen = *(p+i*3+1);
pPal->palPalEntry[i].pcBlue = *(p+i*3+2);
pPal->palPalEntry[i].pcFlags = (BYTE) 0;
}

hpal = CreatePalette(pPal);
LocalFree((HLOCAL) pPal);
hDC = GetDC(hWnd);
SelectPalette(hDC,hpal,0);
RealizePalette(hDC);
}

void ChangeFormat(HWND hWnd)
{
int x,y,i,j,n;
unsigned char color,color2,R,G,B,change[256];
unsigned char rgb[1024];

if((fp=fopen("temp.BMP","rb"))== NULL){
MessageBox(hWnd,"Error Message","Error open file temp.bmp",
MB_ICONINFORMATION|MB_OK);
}

```

```

    exit('F');
}

if((fp2=fopen("work.BMP","w+b"))== NULL){
    MessageBox(hWnd,"Error Message","Error open file work.bmp",
    MB_ICONINFORMATION|MB_OK);

    exit('F');
}

fseek(fp,77,0);
if(fread((char *)&brmp,1,sizeof(BMPHEAD),fp)==sizeof(BMPHEAD)){
    fi.width = (int)brmp.width;
    fi.depth = (int)brmp.depth;

    fi.bits = brmp.bits;
    if (fi.bits != 8)
    {
        MessageBox(hWnd,"Error Message","Error file not 8 bit image",
        MB_ICONINFORMATION|MB_OK);
        exit(1);
    }else{
        nNumColors = 256;
    }
    // get the palette
    n=1<<fi.bits;
    for(i=0;i<n;i++){
        {
            fi.palette[(i*RGB_SIZE)+RGB_RED] = fgetc(fp);
            fi.palette[(i*RGB_SIZE)+RGB_GREEN] = fgetc(fp);
            fi.palette[(i*RGB_SIZE)+RGB_BLUE] = fgetc(fp);
            fgetc(fp);
        }
    }else{
        TextOut(hDC,10,10,"Error reading Heading File ..",29);
        exit(1);
    }

    //***** set color palette *****

    //**** change PALETTE to all gray level ****//
    fseek(fp,77,0);
    for(i=0;i<256;i++){
        R = fi.palette[(i*RGB_SIZE)+RGB_RED];
        G = fi.palette[(i*RGB_SIZE)+RGB_GREEN];
        B = fi.palette[(i*RGB_SIZE)+RGB_BLUE];

        if((R==G)&&(R==B))
            change[i] = R;
        else
            change[i] = i;
    } // end for

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for(i=0;i<256;i++)
        {
            fi.palette[(i*RGB_SIZE)+RGB_RED] = i;
            fi.palette[(i*RGB_SIZE)+RGB_GREEN] = i;
            fi.palette[(i*RGB_SIZE)+RGB_BLUE] = i;
        }
    /***** write file *****/
        for(i=0;i<256;i++)
        {
            rgb[(i*4)+0] = i;
            rgb[(i*4)+1] = i;
            rgb[(i*4)+2] = i;
            rgb[(i*4)+3] = 0;
        }

        fwrite((char *)&brmp,sizeof(BMPHEAD),1,fp2);

        for(i=0;i<1024;i++)
            fputc(rgb[i],fp2);
    /*****
    //write to new file
        fseek(fp,1078,1); //size of BMP Header
        for (y=0;y<fi.depth;y++){
            for (x=0;x<fi.width;x++){
                color = fgetc(fp);
                color2 = change[color];
                fputc(color2,fp2);
            }
        }

    /***** Read to New file && Mem *****/
        fseek(fp2,0L,0);
        if(fread((char *)&brmp,1,sizeof(BMPHEAD),fp2)==sizeof(BMPHEAD)){
            fi.width = (int)brmp.width;
            fi.depth = (int)brmp.depth;
            fi.bits = brmp.bits;

            if (fi.bits != 8)
            {
                MessageBox(hWnd,"This File is not 8 bit Image","Error Message",
                MB_ICONINFORMATION|MB_OK);

                exit(1);
            }else{
                nNumColors = 256;
            }

            // get the pallette

```

เอกสารนี้เป็นเอกสารที่เผยแพร่สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        fi.palette[(i*RGB_SIZE)+RGB_BLUE] = fgetc(fp2);
        fi.palette[(i*RGB_SIZE)+RGB_GREEN] = fgetc(fp2);
        fi.palette[(i*RGB_SIZE)+RGB_RED] = fgetc(fp2);
        fgetc(fp2);
    }
}
else(
    TextOut(hDC,10,10,"Error reading Heading File ..",29);

    exit(1);
}

```

/***** set color palette *****/

```

AllocMem(hWnd);
handle1 = GlobalAlloc(GPTR,fi.width);
image = GlobalLock(handle1);

for (y=0;y<fi.depth;y++){
    for (x=0;x<fi.width;x++){
        color = fgetc(fp2);
        *(image+x) = color;
    }
    WriteMem(image,fi.width,y);
}
}

```

void WriteBmp(HWND hWnd)

```

{
    int i,j,x,y;
    FILE *fp1;
    HGLOBAL wHandle;
    unsigned char far *wBuf;
    unsigned char color;
    unsigned char rgb[1024];

```

```

    wHandle = GlobalAlloc(GPTR,fi.width);
    wBuf = GlobalLock(wHandle);

    if((fp1=fopen(szNameText,"w+b"))== NULL){
        MessageBox(hWnd,"Error Message","Error open file work.bmp",
            MB_ICONINFORMATIONMB_OK);

        exit("F");
    }

```

```

    for (i=0;i<256;i++){

```

```

        rgb[(i*4)+0] = i;

```

```

        rgb[(i*4)+1] = i;
        rgb[(i*4)+2] = i;
        rgb[(i*4)+3] = 0;
    }
    fwrite((char *)&brmp,sizeof(BMPHEAD),1,fp1);

    for(i=0;i<1024;i++)
        fputc(rgb[i],fp1);

    for (y=0;y<fi.depth;y++){
ReadMem(wBuf,fi.width,y);
        for (x=0;x<fi.width;x++){
            color = *(wBuf+x);
            fputc(color,fp1);
        }
    }
    GlobalUnlock(wHandle);
    GlobalFree(wHandle);
}

void AllocMem(HWND hWnd)
{
    long TotalByte;
    double temp;

    TotalByte = (long)fi.width*fi.depth;

    if(TotalByte > 65536L){
        RowPerPage = floor((long)65536L/fi.width);
        TotalPage = floor((long)fi.depth/RowPerPage);
        RowRemainder = fmod(fi.depth,RowPerPage);
        if (RowRemainder > 0)
            TotalPage += 1;
    }else{
        TotalPage = 1;
    }

    switch(TotalPage){
        case 1 : hglb1 = GlobalAlloc(GPTR,(long)fi.width*fi.depth);
        if (!hglb1){
            MessageBox(hWnd,"Allocate Memory Fail at hglb1","Error Message",
                MB_ICONINFORMATION|MB_OK);
            exit(1);
        }
        buffer1 = GlobalLock(hglb1);
        break;
        case 2 : if((hglb1 = GlobalAlloc(GPTR,(long)RowPerPage*fi.width))==NULL){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังไม่มีให้ตีแปลงเนื้อหา และต้องยังต้องสงวนลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MessageBox(hWnd,"Allocate Memory Fail at hglb1","Error Message",
        MB_ICONINFORMATION|MB_OK);

        exit(1);

    }

    buffer1 = GlobalLock(hglb1);

    if(RowRemainder == 0){

        if((hglb2 = GlobalAlloc(GPTR,(long)RowPerPage*fi.width))==NULL)

        {

            MessageBox(hWnd,"Allocate Memory Fail at hglb2","Error Message",
            MB_ICONINFORMATION|MB_OK);

            exit(1);

        }

    }else{

        hglb2 = GlobalAlloc(GPTR,(long)RowRemainder*fi.width);

        if (!hglb2){

            MessageBox(hWnd,"Allocate Memory Fail at hglb2","Error Message",
            MB_ICONINFORMATION|MB_OK);

            exit(1);

        }

        buffer2 = GlobalLock(hglb2);

        break;

        case 3 : if((hglb1 = GlobalAlloc(GPTR,(long)RowPerPage*fi.width))==NULL){

            MessageBox(hWnd,"Allocate Memory Fail at hglb1","Error Message",
            MB_ICONINFORMATION|MB_OK);

            exit(1);

        }

        buffer1 = GlobalLock(hglb1);

        ...

        if((hglb2 = GlobalAlloc(GPTR,(long)RowPerPage*fi.width))==NULL){

            MessageBox(hWnd,"Allocate Memory Fail at hglb2","Error Message",
            MB_ICONINFORMATION|MB_OK);

            exit(1);

        }

        buffer2 = GlobalLock(hglb2);

        if(RowRemainder == 0){

            if((hglb3 = GlobalAlloc(GPTR,(long)RowPerPage*fi.width))==NULL)

            {

                MessageBox(hWnd,"Allocate Memory Fail at hglb3","Error Message",
                MB_ICONINFORMATION|MB_OK);

                exit(1);

            }

        }else{

            if((hglb3 = GlobalAlloc(GPTR,(long)RowRemainder*fi.width))!=

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MessageBox(hWnd,"Allocate Memory Fail at hglb3","Error Message",
        MB_ICONINFORMATION|MB_OK);

        exit(1);
    }

    }

    buffer3 = GlobalLock(hglb3);
    break;
}

}

```

```

void ReadMem(unsigned char far *buffer,int BufSize,int RowOffset)

```

```

{
    int i,j;
    int BufHglb;
    int PageNo,RowAtPage;

    if(TotalPage == 1){
        for(i=0;i<BufSize;i++){
            *(buffer+i) = *(buffer1+fi.width*RowOffset+i); // not exceed 64K
        }
        PageNo = floor((long)RowOffset/RowPerPage);
        RowAtPage = (long)RowOffset - (long)RowPerPage*(PageNo-1);

        if(RowOffset<RowPerPage){
            for(i=0;i<BufSize;i++){
                *(buffer+i)=*(buffer1+BufSize*RowOffset+i);
            }
        }else{
            if((RowOffset>RowPerPage)&&(RowOffset<RowPerPage*2)){
                for(i=0;i<BufSize;i++){
                    *(buffer+i)= *(buffer2+BufSize*RowAtPage+i);
                }
            }else{
                if((RowOffset>RowPerPage*2)&&(RowOffset<RowPerPage*3))
                    for(i=0;i<BufSize;i++){
                        *(buffer+i) = *(buffer3+fi.width*RowAtPage-i);
                    }
            }
        }
    } // end total page
}

```

```

void WriteMem(unsigned char far *buffer,int BufSize,int RowOffset)

```

```

{
    int i,j;
    int BufHglb;
    int PageNo,RowAtPage;

```

```

    if(TotalPage == 1){

```

```

        for(i=0;i<BufSize;i++){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        *(buffer1+BufSize*RowOffset+i)= *(buffer+i);
    }

    }else{

        PageNo  = ceil((long)(RowOffset+1)/RowPerPage);
        RowAtPage = RowOffset - RowPerPage*(PageNo-1);

        if(RowOffset<RowPerPage){
            for(i=0;i<BufSize;i++)
                *(buffer1+BufSize*RowOffset+i) = *(buffer+i);
        }else{
            if((RowOffset>RowPerPage)&&(RowOffset<RowPerPage*2)){
                for(i=0;i<BufSize;i++)

                    *(buffer2+BufSize*RowAtPage+i) = *
(buffer+i);

            }else{
                if((RowOffset>RowPerPage*2)&&(RowOffset<RowPerPage*3))
                    for(i=0;i<BufSize;i++)

                        *(buffer3+fi.width*RowAtPage+i) = *
(buffer+i);
            }
        } //endif TotalPage
    }

}

BOOL FAR PASCAL OpenNewDlgProc(hDlg,message,wParam,lParam)
HWND hDlg;
unsigned message;
WORD wParam;
LONG lParam;
{
    static HWND hName;

    switch(message)
    {
        case WM_INITDIALOG :
            hName = GetDlgItem(hDlg,ID_NAME);
            SendMessage(hName,EM_LIMITTEXT,12,0L);
            SetFocus(hName);
            return(FALSE);

        case WM_COMMAND :
            switch(wParam)
            {
                case IDOK :
                    GetDlgItemText(hDlg,ID_NAME,szNameText,15);
                    EndDialog(hDlg,1);
                    return(TRUE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        case IDCANCEL :
            EndDialog(hDlg,0);
            return(TRUE);

        default :

            return(FALSE);
    }

    default :
        return(FALSE);
}

```

```

BOOL FAR PASCAL GraphDlgProc(hDlg,message,wParam,lParam)

```

```

HWND hDlg;

```

```

unsigned message;

```

```

WORD wParam;

```

```

LONG lParam;

```

```

{
    HWND hGraph;
    PAINTSTRUCT Gps;
    RECT Grect;
    HDC Ghdc;
    int StartX,StartY,StrLen;
    int x,y,i,j;
    float hist[256],ValueMax;
    unsigned int Percent[256];
    unsigned char color;
    HGLOBAL Hhandle;
    unsigned char far *hBuf;

```

```

    switch(message)

```

```

    {

```

```

        case WM_INITDIALOG :

```

```

            hGraph = GetDlgItem(hDlg,ID_PAINT);

```

```

            SetFocus(hGraph);

```

```

            return(FALSE);

```

```

        case WM_COMMAND :

```

```

            switch(wParam)

```

```

            {

```

```

                case IDOK :

```

```

                    EndDialog(hDlg,1);

```

```

                    return(TRUE);

```

```

                default :

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return(FALSE);
    }
    case WM_PAINT :
        Ghdc = BeginPaint(hDlg,&Gps);
        Hhandle = GlobalAlloc(GPTR,fi.width);
        hBuf = GlobalLock(Hhandle);

        for(i=0;i<256;i++)
            hist[i]=0;

        for (y=0;y<fi.depth;y++){
            ReadMem(hBuf,fi.width,y);
            for (x=0;x<fi.width;x++){
                color = *(hBuf+x);

                hist[color] += 1;
            }
        }

        ValueMax = 0;

        for(i=0;i<256;i++){ //search maximum value
            if(hist[i]>ValueMax)
                ValueMax = hist[i];
        }

        Percent[i] = (hist[i]*200)/ValueMax;

        StartX = 40;
        StartY = 250;
        MoveTo(Ghdc,StartX,StartY);
        LineTo(Ghdc,StartX+270,StartY);
        MoveTo(Ghdc,StartX,StartY);
        LineTo(Ghdc,StartX,StartY-205);
        TextOut(Ghdc,StartX-5,10,"Pixels",6);
        TextOut(Ghdc,StartX+270,StartY,"Gray Level",10);
        TextOut(Ghdc,StartX,StartY+12,"0",1);

        TextOut(Ghdc,StartX+120,StartY+12,"120",3);
        TextOut(Ghdc,StartX+250,StartY+12,"250",3);
        StartX += 2;
        for (i=0;i<=25;i++){ // Scale Graph
            MoveTo(Ghdc,StartX+(i*10),StartY);

            if(i==0 || i==12 || i==25)
                LineTo(Ghdc,StartX+(i*10),StartY+10);
            else
                LineTo(Ghdc,StartX+(i*10),StartY+5);
        }

        MoveTo(Ghdc,StartX,StartY);
        for(i=0;i<256;i++){
            MoveTo(Ghdc,StartX+i,StartY);

```

```
LineTo(Ghdc,StartX+i,StartY-Percent(i));
}

EndPoint(hDlg,&Gps);
GlobalUnlock(Hhandle);
GlobalFree(Hhandle);

return(FALSE);

default :
return(FALSE);
}

}
```

```
BOOL FAR PASCAL ZoomDlgProc(hDlg,message,wParam,lParam)
HWND hDlg;
unsigned message;
WORD wParam;
LONG lParam;
{
    HWND hZoom,hWnd;
    PAINTSTRUCT Gps;
    RECT Grect;
    HDC Ghdc;
    int x,y,i,j,count;
    float value,x1,x2,x3,x4,x5,x6,x7,x8,x9;
    unsigned char color;
    HGLOBAL Hhandle,lhandle1,lhandle2,lhandle3;
    unsigned char far *Row1;
    unsigned char far *Row2;
    unsigned char far *Row3;
    unsigned char far *Lbuf2;
    HGLOBAL Lhandle;
    unsigned char far *Lbuf;
    BOOL FlagRead = FALSE;
    BOOL FlagRead2 = FALSE;

    Lhandle = GlobalAlloc(GPTR,fi.width);
    Lbuf = GlobalLock(Lhandle);
    Hhandle = GlobalAlloc(GPTR,fi.width*2);
    Lbuf2 = GlobalLock(Hhandle);
    lhandle1 = GlobalAlloc(GPTR,fi.width*2);
    Row1 = GlobalLock(lhandle1);
    lhandle2 = GlobalAlloc(GPTR,fi.width*2);
    Row2 = GlobalLock(lhandle2);
    lhandle3 = GlobalAlloc(GPTR,fi.width*2);
    Row3 = GlobalLock(lhandle3);

    switch(message)
```

เอกสารนี้เป็นเอกสารทบทวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case WM_INITDIALOG :
    hZoom = GetDlgItem(hDlg,ID_PAINT);
    SetFocus(hZoom);
    return(FALSE);

case WM_COMMAND :
    switch(wParam)
    {
        case IDOK :
            EndDialog(hDlg,1);
            return(TRUE);

        default :
            return(FALSE);
    }

case WM_PAINT :
    Ghdc = BeginPaint(hDlg,&PtStr);

    pPal = (LOGPALETTE*) LocalAlloc(LPTR,sizeof(LOGPALETTE)+nNumColors*sizeof(PALETTEENTRY));
    if (!pPal){
        TextOut(Ghdc,10,10,"Allocate pPal Failed ",21);
        exit(1);
    }
    pPal->palNumEntries = nNumColors;
    pPal->palVersion = 0x300;
    for (i=0;i<nNumColors;i++){
        pPal->palPalEntry[i].peRed   = fi.palette[(i*3)];
        pPal->palPalEntry[i].peGreen = fi.palette[(i*3)+1];
        pPal->palPalEntry[i].peBlue  = fi.palette[(i*3)+2];
        pPal->palPalEntry[i].peFlags = (BYTE) 0;
    }
    hpal = CreatePalette(pPal);
    Ghdc = GetDC(hWnd);
    SelectPalette(Ghdc,hpal,0);
    RealizePalette(Ghdc);

    ReadMem(Lbuf,fi.width,0); //Read line 0

    for (i=0;i<fi.width*2;i++){ // init 3 Rows to Zoom
        if (FlagRead==FALSE){
            *(Row1+i) = *(Lbuf+(i/2));
            FlagRead = TRUE;
        }
        else{
            *(Row1+i) = 0;
            FlagRead = FALSE;
        }
        *(Row2+i) = 0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ReadMem(Lbuf,fi.width,1); // Read line 1
FlagRead = FALSE;
for (i=0;i<fi.width*2;i++){
if (FlagRead == FALSE){
*(Row3+i) = *(Lbuf+(i/2));
FlagRead = TRUE;
}else{
*(Row3+i) = 0;
FlagRead = FALSE;
}
}

for(x=1;x<(fi.width*2)-1;x++){ // Calculate first line
x4 = *(Row1+(x-1));
x5 = *(Row1+x);
x6 = *(Row1+(x+1));
value = x4/2 + x5 + x6/2;
SetPixel(Ghdc,x,fi.depth*2,PALETTEINDEX(value));
}
for(x=1;x<(fi.width*2)-1;x++){ //Calculate for second line
x1 = *(Row1+(x-1));
x2 = *(Row1+x);
x3 = *(Row1+(x+1));
x4 = *(Row2+(x-1));
x5 = *(Row2+x);
x6 = *(Row2+(x+1));
x7 = *(Row3+(x-1));
x8 = *(Row3+x);
x9 = *(Row3+(x+1));
value = x1/4 + x2/2 + x3/4 + x4/2 + x5 + x6/2 + x7/4 + x8/2 + x9/4;
if(value>255)
value = 255;
else
if(value<0)
value = 0;
SetPixel(Ghdc,x,(fi.depth*2)-1,PALETTEINDEX(value));
}

//***** Main Loop for Calculate *****
count = 1;
FlagRead = FALSE;

for(y=2;y<(fi.depth*2)-1;y++){ // Main Loop
if (FlagRead == FALSE){ // Read Memory Image & Insert Line 0
for(i=0;i<fi.width*2;i++)
*(Lbu2+i) = 0;

```

```

FlagRead = TRUE;

    }else{

        count += 1;
        FlagRead2 = FALSE;
        ReadMem(Lbuf,fi.width,count);
        for(i=0;i<fi.width*2;i++){

            if(FlagRead2 == FALSE){

                *(Lbuf2+i) = *(Lbuf+(i/2));
                FlagRead2 = TRUE;

            }else{

                *(Lbuf2+i) = 0;

            }

        }

        FlagRead2 = FALSE;

    }

}

FlagRead = FALSE;

}

for(x=0;x<fi.width*2;x++){ //Scroll Line up
    *(Row1+x) = *(Row2+x);
    *(Row2+x) = *(Row3+x);
    *(Row3+x) = *(Lbuf2+x);
}

for(x=1;x<(fi.width*2)-1;x++){ //Loop for putpixel 1 Line
    x1 = *(Row1+(x-1));
    x2 = *(Row1+x);
    x3 = *(Row1+(x+1));
    x4 = *(Row2+(x-1));
    x5 = *(Row2+x);
    x6 = *(Row2+(x+1));
    x7 = *(Row3+(x-1));
    x8 = *(Row3+x);
    x9 = *(Row3+(x+1));

    value = x1/4 + x2/2 + x3/4 + x4/2 + x5 + x6/2 + x7/4 + x8/2 + x9/4;
    if(value>255)

        value = 255;

    else

        if(value<0)

            value = 0;

    SetPixel(Ghdc.x,(fi.depth*2)-y,PALETTEINDEX(value));

    // SetPixel(Ghdc.x,(fi.depth*2)-y,PALETTEINDEX(x5));

}

}

```

```

        EndPaint(hDlg,&Gps);

        GlobalUnlock(Lhandle);
        GlobalFree(Lhandle);
        GlobalUnlock(Hhandle);
        GlobalFree(Hhandle);
        GlobalUnlock(Ihandle1);
        GlobalFree(Ihandle1);
        GlobalUnlock(Ihandle2);
        GlobalFree(Ihandle2);
        GlobalUnlock(Ihandle3);
        GlobalFree(Ihandle3);

    return(FALSE);

    default :
        return(FALSE);
    }
}

BOOL FAR PASCAL ScrollDlgProc(hDlg,message,wParam,lParam)
HWND hDlg;
unsigned message;
WORD wParam;
LONG lParam;
{
    HWND hScroll;

    switch(message)
    {
        case WM_INITDIALOG :
            hScroll = GetDlgItem(hDlg,ID_SCROLL);
            SetFocus(hScroll);
            SetScrollRange(hScroll,SB_CTL,0,200,TRUE);
            SetScrollPos(hScroll,SB_CTL,100,TRUE);
            return(FALSE);

        case WM_HSCROLL :
            SetFocus(HIWORD(lParam));
            if(wParam != SB_THUMBTRACK){
                Position = (BYTE)GetScrollPos(HIWORD(lParam),SB_CTL);
                switch(wParam)
                {

```

```

                    case SB_LINEDOWN :
                        Position += 10;
                        break;

                    case SB_LINEUP :

```

```

                        Position -= 10;
                        break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case SB_PAGEDOWN :
            Position += 10;
            break;
        case SB_PAGEUP:
            Position -= 10;
            break;
        case SB_THUMBPOSITION : // current position is specify by the nPos
            Position = (BYTE)LOWORD(lParam);
    }
    SetScrollPos(HIWORD(lParam),SB_CTL,Position,TRUE);
}

return(TRUE);
```

```
case WM_COMMAND :
{
    switch(wParam)
    {
        case IDOK :
            EndDialog(hDlg,1);
            return(TRUE);

        case IDCANCEL :
            EndDialog(hDlg,0);
            return(FALSE);

        default :
            return(FALSE);
    }
}

return(FALSE);
}
```

```
void LoadImage(void)
{
    HGLOBAL Lhandle;
    unsigned char far *Lbuf;
    unsigned char color;
    int x,y;

    Lhandle = GlobalAlloc(GPTR,fi.width);
    Lbuf = GlobalLock(Lhandle);

    fseck(fp2,1078,0);
```

```

        for (x=0;x<fi.width;x++){
            color    = fgetc(fp2);
            //SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(color));
            *(Lbuf+x) = color;
        }
        WriteMem(Lbuf,fi.width,y);
    }

    for (y=0;y<fi.depth;y++){
        ReadMem(Lbuf,fi.width,y);
        for (x=0;x<fi.width;x++){
            color = *(Lbuf+x);

            SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(color));
        }
    }

    GlobalUnlock(Lhandle);
    GlobalFree(Lhandle);
}

void Brightness(int count)
{
    int x,y,temp;
    unsigned char color;
    int array[768];
    HGLOBAL Bhandle,Dhandle;
    unsigned char far *bright;
    unsigned char far *MemBuf;
    char ch;
    int num;

    num = count - 100;
    Bhandle = GlobalAlloc(GPTR,fi.width);
    bright  = GlobalLock(Bhandle);
    Dhandle = GlobalAlloc(GPTR,fi.width);
    MemBuf  = GlobalLock(Dhandle);

    for (y=0;y<fi.depth;y++){
        ReadMem(bright,fi.width,y);

        for (x=0;x<fi.width;x++){
            color    = *(bright+x);
            temp      = color+num;
            temp      = max(temp,0.0);
            temp      = min(temp,255.0);
            *(MemBuf+x) = temp;
            SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(temp));
        }
        WriteMem(MemBuf,fi.width,y);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    GlobalUnlock(Bhandle);
    GlobalFree(Bhandle);
    GlobalUnlock(Dhandle);
    GlobalFree(Dhandle);
}

void Contrast(HWND hWnd,int count,unsigned char *p)
{
    int x,y,temp;
    int i,j,step;
    HGLOBAL Chandle,Dhandle;
    unsigned char far *Cbuf;
    unsigned char far *MemBuf;
    unsigned char color[768];
    int icolor;
    unsigned char tcolor;
    int array[768];
    char ch;
    float num=1;

    Chandle = GlobalAlloc(GPTR,fi.width);
    Cbuf = GlobalLock(Chandle);
    Dhandle = GlobalAlloc(GPTR,fi.width);
    MemBuf = GlobalLock(Dhandle);

    count = count/4;
    count = count - 25;    //- center scroll position
    if (count >= 0){
        for(i=0;i<count;i++)
            num *= 1.1;
    }else{
        step = abs(count);
        for(i=0;i<step;i++)
            num *= 0.9;
    }

    // MessageBox(hWnd,"00","Message",
    //
    MB_ICONINFORMATION|MB_OK);

    for(i=0;i<768;i++){
        array[i] = *(p++);

        for(ii=0;ii<768;ii++){
            array[ii] = (array[ii]-128)*num+128;

            temp = max(array[ii],0.0);

            temp = min(temp,255.0);

            color[ii] = temp;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for (y=0;y<fi.depth;y++){
ReadMem(Cbuf,fi.width,y);
        for (x=0;x<fi.width;x++){
            icolor    = *(Cbuf+x);
            tcolor    = color[icolor*3];

            *(MemBuf+x) = tcolor;

            SetPixel(hdc,x,fi.depth-y+1,PALETTEINDEX(tcolor));
        }
        WriteMem(MemBuf,fi.width,y);
    }

    GlobalUnlock(Chandle);
    GlobalFree(Chandle);
    GlobalUnlock(Dhandle);
    GlobalFree(Dhandle);
}

void AutoContrast(void)
{
    int i,j,x,y;
    float Xmin,Xmax,Ymin,Ymax;
    float TempMin,TempMax,hist[256];
    float Temp;
    double PercentCut;
    float Slope;
    unsigned char color;
    HGLOBAL Hhandle,Dhandle;
    unsigned char far *hBuf;
    unsigned char far *MemBuf;
    int LookTable[256];
    int ii;
    BOOL Flag=FALSE;

    Hhandle = GlobalAlloc(GPTR,fi.width);
    hBuf    = GlobalLock(Hhandle);
    Dhandle = GlobalAlloc(GPTR,fi.width);
    MemBuf  = GlobalLock(Dhandle);

    for(i=0;i<256;i++){
        hist[i]=0;

        LookTable[i]=0;
    }

    for (y=0;y<fi.depth;y++){
        ReadMem(hBuf,fi.width,y);
        for (x=0;x<fi.width;x++){
            color = *(hBuf+x);

            hist[color] += 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    TempMin = 0;
    TempMax = 0;
    Xmin = 0;
    Xmax = 255;
    Ymin = 0;
    Ymax = 255;
    PercentCut = fi.width*0.01*fi.depth; // set to 1%

    ii=0;

    do{
        TempMax += hist[255-ii];
        if(TempMax > PercentCut){
            Xmax = 255-ii;
        }
        Flag =TRUE;
        ii++;
    }while (Flag==FALSE);
    Flag=FALSE;
    ii=0;
    do{
        TempMin += hist[ii];
        if(TempMin > PercentCut){
            Xmin = ii;
            Flag = TRUE;
        }
        ii++;
    }while (Flag==FALSE);

    Slope = (Ymax-Ymin)/(Xmax-Xmin);

    for(i=0;i<256;i++){
        if( i < Xmin )
            LookTable[i] = 0;

    else

        if ((i >= Xmin)&&(i <= Xmax)){
            Temp = (i - Xmin)*Slope;
            if(Temp<0)
                Temp=0;

            if(Temp>255)
                Temp=255;

            LookTable[i] = Temp;
        }

    else

        if(i>Xmax)

            LookTable[i] =255 ;

        //IndexArr++;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for (y=0;y<fi.depth;y++){
ReadMem(hBuf,fi.width,y);
        for (x=0;x<fi.width;x++){
            color = *(hBuf+x);

            *(MemBuf+x)= LookTable[color];
            color = LookTable[color];
            SetPixel(hDC,x,fi.depth-y+1,PALETTEINDEX(color));
        }
        WriteMem(MemBuf,fi.width,y);
    }

    GlobalUnlock(Hhandle);
    GlobalFree(Hhandle);
    GlobalUnlock(Dhandle);
    GlobalFree(Dhandle);
}

void Sharp(void)
{
HGLOBAL hglb,Shandle,Dhandle;
unsigned char far *sharp;
unsigned char far *Sbuf;
unsigned char far *MemBuf;
int x,y,temp;
int i,j,step;
unsigned char color;
float value;
int icolor;
char ch;

    hglb = GlobalAlloc(GPTR,(long)fi.width*3); // allocate 3 row
    sharp = (unsigned char far *)GlobalLock(hglb);
    Shandle = GlobalAlloc(GPTR,fi.width);
    Sbuf = GlobalLock(Shandle);
    Dhandle = GlobalAlloc(GPTR,fi.width);
    MemBuf = GlobalLock(Dhandle);

    for(j=0;j<3;j++){
ReadMem(Sbuf,fi.width,j);
        for(i=0;i<fi.width;i++) // read first 3 rows using 3x3
            /*(sharp+i) = fgetc(fp2);

        *(sharp+fi.width*j+i) = *(Sbuf+i);
    }

```

```

for(x=0;x<fi.width;x++){      // write first line to screen
    SetPixel(hDC,x,fi.depth,PALETTEINDEX(*(sharp+x)));
    value =(*(sharp+fi.width*1+x))*5 - (*(sharp+fi.width*2+x)+
                                           (*(sharp+fi.width*0+x)) + (*(sharp+fi.width*1+x+1))+
                                           (*(sharp+fi.width*1+x-1)));
    SetPixel(hDC,x,fi.depth-1,PALETTEINDEX(value));
}

////////////////////////////////////

for(y=2;y<fi.depth-1;y++){    // Calculate Sharpening loop
ReadMem(Sbuf,fi.width,y);

    for(i=0;i<fi.width;i++){

        *(sharp+i)                                = *(sharp+fi.width*1+i); // row 0 = row 1
        *(sharp+fi.width*1+i) = *(sharp+fi.width*2+i); // row 1 = row 2
        /*(sharp+fi.width*2+i) = fgetc(fp2);
        *(sharp+fi.width*2+i) = *(Sbuf+i);
    }

    for(x=1;x<fi.width;x++){
        value =(*(sharp+fi.width*1+x))*5 - (*(sharp+fi.width*2+x)+
                                           (*(sharp+fi.width*0+x)) + (*(sharp+fi.width*1+x+1))+
                                           (*(sharp+fi.width*1+x-1)));
        if(value <0)
            value = 0;
        else if (value>255)
            value = 255;
        *(MemBuf+x) = value;
        SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(value));
    }
    WriteMem(MemBuf,fi.width,y);
}

//////////////////////////////////// Scroll Row up

GlobalUnlock(hglb);
GlobalFree(hglb);
GlobalUnlock(Shandle);
GlobalFree(Shandle);
GlobalUnlock(Dhandle);
GlobalFree(Dhandle);
}

void Smooth(void)
{
HGLOBAL hglb,hglb1,Smhandle,Dhandle;
unsigned char far *smooth;
unsigned char far *Smbuf;
unsigned char far *MemBuf;

```

```
int x,y,temp;
int i,j,stop;
unsigned char color;
```

```
float value;
int icolor;
char ch;
```

```
hglb   = GlobalAlloc(GPTR,(long)fi.width*3); // allocate 1 row
smooth = (unsigned char far *)GlobalLock(hglb);
Smhandle = GlobalAlloc(GPTR,fi.width);
Smbuf   = GlobalLock(Smhandle);
Dhandle = GlobalAlloc(GPTR,(long)fi.width);
MemBuf   = GlobalLock(Dhandle);

//fseek(fp2,sizeof(BMPHEAD)+1024,0);
for(j=0;j<3;j++){
ReadMem(Smbuf,fi.width*j);
    for(i=0;i<fi.width;i++) // read first 3 rows using 3x3
*(smooth+fi.width*j+i) = *(Smbuf+i);
        /*(smooth+i) = fgetc(fp2);
}

// write first 2 line
for(x=0;x<fi.width;x++){ // write first line to screen
SetPixel(hDC,x,fi.depth.PALETTEINDEX(*(smooth+x)));
value = (*(smooth+fi.width*1+x))*5 - (*(smooth+fi.width*2+x)+
        (*(smooth+fi.width*0+x)) + (*(smooth+fi.width*1+x+1))+
        (*(smooth+fi.width*1+x-1)));
SetPixel(hDC,x,fi.depth-1.PALETTEINDEX(value));
}

////////// using average //////////

for(y=2;y<fi.depth-1;y++){ // Calculate Smoothing loop
ReadMem(Smbuf,fi.width*y);
    for(i=0;i<fi.width;i++){
        *(smooth+i) = (*(smooth+fi.width*1+i); // row 0 = row 1
        *(smooth+fi.width*1+i) = *(smooth+fi.width*2+i); // row 1 = row 2
        *(smooth+fi.width*2+i) = *(Smbuf+i);
        /*(smooth+fi.width*2+i) = fgetc(fp2);
    }

    for(x=1;x<fi.width-1;x++){
        value = *(smooth+x-1) + *(smooth+x) +
        *(smooth+x+1) + *(smooth+fi.width+x-1)+
        *(smooth+fi.width+x) + *(smooth+fi.width+x+1)+
        *(smooth+fi.width*2+x-1) + *(smooth+fi.width*2+x)+
        *(smooth+fi.width*2+x+1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        value = value/9;
        if (value<0)
            value = 0;
        else if (value > 255)
            value = 255;

        *(MemBuf+x) = value;

        SetPixel(hdc,x,fi.depth-y,PALETTEINDEX(value));
    }

    WriteMem(MemBuf,fi.width,y);
}

//////////////////// Scroll Row up

GlobalUnlock(hglb);
GlobalFree(hglb);
GlobalUnlock(Smhandle);
GlobalFree(Smhandle);
GlobalUnlock(Dhandle);
GlobalFree(Dhandle);
}

void Weight(void)
{
    HGLOBAL hglb,hglb1,Smhandle,Dhandle;
    unsigned char far *smooth;
    unsigned char far *Smbuf;
    unsigned char far *MemBuf;
    int x,y,temp;
    int i,j,step;
    unsigned char color;

    float value;
    float x1,x2,x3,x4,x5,x6,x7,x8,x9;
    int icolor;
    char ch;

    hglb = GlobalAlloc(GPTR,(long)fi.width*3); // allocate 1 row
    smooth = (unsigned char far *)GlobalLock(hglb);
    Smhandle = GlobalAlloc(GPTR,fi.width);
    Smbuf = GlobalLock(Smhandle);
    Dhandle = GlobalAlloc(GPTR,(long)fi.width);
    MemBuf = GlobalLock(Dhandle);

    //fseek(fp2,sizeof(BMPHEAD)+1024,0);
    for(j=0;j<3;j++){
        ReadMem(Smbuf,fi.width,j);

        for(i=0;i<fi.width;i++) // read first 3 rows using 3x3
            *(smooth+fi.width*j+i) = *(Smbuf+i);
    }
}

```

```

}

// write first 2 line
for(x=0;x<fi.width;x++){          // write first line to screen
    SetPixel(hDC,x,fi.depth,PALETTEINDEX(*(smooth+x)));
    value = (*(smooth+fi.width*1+x))*5 - (*(smooth+fi.width*2+x)+
        (*(smooth+fi.width*0+x)) + (*(smooth+fi.width*1+x+1))+
        (*(smooth+fi.width*1+x-1)));
    SetPixel(hDC,x,fi.depth-1,PALETTEINDEX(value));
}

////////// using weight //////////

for(y=2;y<fi.depth-1;y++){          // Calculate Smoothing loop
    ReadMem(Smbuf,fi.width,y);

    for(i=0;i<fi.width;i++){
        *(smooth+i) = *(smooth+fi.width*1+i); // row 0 = row 1
        *(smooth+fi.width*1+i) = *(smooth+fi.width*2+i); // row 1 = row 2
        *(smooth+fi.width*2+i) = *(Smbuf+i);
    }

    for(x=1;x<fi.width-1;x++){
        x1 = *(smooth+x-1);
        x2 = *(smooth+x);
        x3 = *(smooth+x+1);
        x4 = *(smooth+fi.width+x-1);
        x5 = *(smooth+fi.width+x);
        x6 = *(smooth+fi.width+x+1);
        x7 = *(smooth+fi.width*2+x-1);
        x8 = *(smooth+fi.width*2+x);
        x9 = *(smooth+fi.width*2+x+1);
        value = (x1/16)+(x2/8)+(x3/16)+(x4/8)+(x5/4)+(x6/8)+(x7/16)+(x8/8)-(x9/16);

        if (value<0)
            value = 0;
        else if (value > 255)
            value = 255;

        *(MemBuf+x) = value;

        SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(value));
    }

    WriteMem(MemBuf,fi.width,y);
}

////////// Scroll Row up

GlobalUnlock(hgib);
GlobalFree(hgib);
GlobalUnlock(Smhandle);
GlobalFree(Smhandle);
GlobalUnlock(Dhandle);
GlobalFree(Dhandle);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Edge(void)
{
HGLOBAL hglb,Ehandle,Dhandle;

unsigned char far *edge;

unsigned char far *Ebuf;

unsigned char far *MemBuf;

int x,y,temp;

int i,j,step;

unsigned char color;

float value;

int icolor;

char ch;

```

```

hglb = GlobalAlloc(GPTR,(long)fi.width*3); // allocate 1 row
edge = (unsigned char far *)GlobalLock(hglb);
Ehandle = GlobalAlloc(GPTR,fi.width);
Ebuf = GlobalLock(Ehandle);
Dhandle = GlobalAlloc(GPTR,(long)fi.width);
MemBuf = GlobalLock(Dhandle);

for (j=0;j<3;j++){
ReadMem(Ebuf,fi.width,j);

for(i=0;i<fi.width;i++){ // read first 3 rows using 3x3
*(edge+fi.width*j+i) = *(Ebuf+i);
//*(edge+i) = fgetc(fp2);
}

// write first 2 line
for(x=0;x<fi.width;x++){ // write first line to screen
SetPixel(hDC,x,fi.depth-1,PALETTEINDEX(*(edge+x)));
value = (*(edge+fi.width*1+x))*5 - (*(edge+fi.width*2+x)+
*(edge+fi.width*0+x) + *(edge+fi.width*1+x-1))+
*(edge+fi.width*1+x-1));
SetPixel(hDC,x,fi.depth-1,PALETTEINDEX(value));
}

////////////////////////////////////

for(y=2;y<fi.depth-1;y++){
ReadMem(Ebuf,fi.width,y); // Calculate Edge loop
for(i=0;i<fi.width;i++){
*(edge+i) = *(edge+fi.width*1+i); // row 0 = row 1
*(edge+fi.width*1+i) = *(edge+fi.width*2+i); // row 1 = row 2
*(edge+fi.width*2+i) = *(Ebuf+i);
}

for(x=1;x<fi.width-1;x++){
value = abs(*(edge+x+1) + *(edge+fi.width+x+1)*2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

+ *(edge+fi.width*2+x+1) - *(edge+x-1)
- *(edge+fi.width*1+x-1)*2 - *(edge+fi.width*2+x-1))
+abs(*(edge+fi.width*2+x-1) + *(edge+fi.width*2+x)*2
+ *(edge+fi.width*2+x+1) - *(edge+x-1)
- *(edge+x)*2 - *(edge+x+1));

*(MemBuf+x) = value;
SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(value));
}

WriteMem(MemBuf,fi.width,y);
}

//////////////////// Scroll Row up

GlobalUnlock(hglb);
GlobalFree(hglb);
GlobalUnlock(Ehandle);
GlobalFree(Ehandle);
GlobalUnlock(Dhandle);
GlobalFree(Dhandle);
}

void HistEqualize(void)
{
    HGLOBAL hglb11,hglb22,hglb33,hglb55,Hhandle;
    unsigned char far *Hbuf;
    unsigned char far *MemBuf;
    int i,x,y;
    char ch;
    unsigned char color;
    long int far *hist;
    unsigned char far *table;
    long int far *Cumulative;
    int temp,temp2;

    double temp11>TotalPixel;
    float temp1;

    hglb11 = GlobalAlloc(GPTR,256*sizeof(long int));
    hist = (long int far *)GlobalLock(hglb11);
    hglb33 = GlobalAlloc(GPTR,256*sizeof(long int));
    Cumulative = (long int far *)GlobalLock(hglb33);
    hglb55 = GlobalAlloc(GPTR,256*sizeof(unsigned char));
    table = (unsigned char far *)GlobalLock(hglb55);
    Hhandle = GlobalAlloc(GPTR,fi.width);
    Hbuf = GlobalLock(Hhandle);
    hglb22 = GlobalAlloc(GPTR,fi.width);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MemBuf = GlobalLock(hglb22);

/***** HISTOGRAM EQUALIZATION *****/

TotalPixel = (long)fi.width*(long)fi.depth;
for(i=0;i<256;i++)
    *(hist+i) = 0;

for(y=0;y<fi.depth;y++){ // create histogram
    ReadMem(Hbuf,fi.width,y);
    for(x=0;x<fi.width;x++){
        temp    = *(Hbuf+x);
        hist[temp] += 1;
    }
}
temp11 = 0;
for(i=0;i<256;i++){
    temp11 += *(hist+i);
    *(Cumulative+i) = temp11;
}

for(i=0;i<256;i++){
    //temp1 = *(hist+i);
    temp11 = ceil(*(Cumulative+i)*256/TotalPixel)-1;
    if(temp11>255)
        temp11 = 255;
    else
        if (temp11<0)
            temp11 = 0;
    *(table+i) = temp11;
}

for(y=0;y<fi.depth;y++){
    ReadMem(Hbuf,fi.width,y);
    for(x=0;x<fi.width;x++){
        temp    = *(Hbuf+x);
        temp2 = *(table+temp);
        *(MemBuf+x) = temp2;
        SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(temp2));
    }
    WriteMem(MemBuf,fi.width,y);
}

GlobalUnlock(hglb11);
GlobalFree(hglb11);
GlobalUnlock(hglb22);
GlobalFree(hglb22);

```

```

GlobalUnlock(hglb33);
GlobalFree(hglb33);
GlobalUnlock(Hhandle);
GlobalFree(Hhandle);
GlobalUnlock(hglb55);
GlobalFree(hglb55);
}
/*
void AdaptiveHist(void)
{
HGLOBAL hglb11,hglb22,hglb33,hglb55,Hhandle;
unsigned char far *Hbuf;
unsigned char far *MemBuf;
int i,j,x,y,RectSizeX,RectSizeY,XStart,YStart;
char ch;
unsigned char color;
long int far *hist;
unsigned char far *table;
long int far *Cumulative;
int temp,temp2;

double temp11>TotalPixel;
float temp1;

Hhandle = GlobalAlloc(GPTR,fi.width);
Hbuf = GlobalLock(Hhandle);
hglb22 = GlobalAlloc(GPTR,fi.width);
MemBuf = GlobalLock(hglb22);
hglb11 = GlobalAlloc(GPTR,4096*sizeof(long int)); //256*4
hist = (long int far *)GlobalLock(hglb11);

hglb33 = GlobalAlloc(GPTR,256*sizeof(long int));
Cumulative = (long int far *)GlobalLock(hglb33);
hglb55 = GlobalAlloc(GPTR,256*sizeof(unsigned char));
table = (unsigned char far *)GlobalLock(hglb55);

//***** ADAPTIVE HISTOGRAM EQUALIZATION *****

TotalPixel = (long)fi.width*(long)fi.depth;
for(i=0;i<4096;i++)
    *(hist+i) = 0;

for(y=0;y<fi.depth;y++){ // create histogram
ReadMem(Hbuf,fi.width,y);
for(x=0;x<fi.width;x++){
temp = *(Hbuf+x);
hist[temp] += 1;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    temp11 = 0;
    for(i=0;i<256;i++){
        temp11 += *(hist+i);
        *(Cumulative+i) = temp11;
    }

    for(i=0;i<256;i++){
        //temp1 = *(hist+i);
        temp11 = ceil(*(Cumulative+i)*256/TotalPixel)-1;
        if(temp11>255)
            temp11 = 255;
        else
            if (temp11<0)
                temp11 = 0;
        *(table+i) = temp11;
    }

    for(y=0;y<fi.depth;y++){
        ReadMem(Hbuf,fi.width,y);
        for(x=0;x<fi.width;x++){
            temp = *(Hbuf+x);
            temp2 = *(table+temp);
            *(MemBuf+x) = temp2;
            SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(temp2));
        }
        WriteMem(MemBuf,fi.width,y);
    }

    GlobalUnlock(hglb11);
    GlobalFree(hglb11);
    GlobalUnlock(hglb22);
    GlobalFree(hglb22);
    GlobalUnlock(hglb33);
    GlobalFree(hglb33);
    GlobalUnlock(Hhandle);
    GlobalFree(Hhandle);
    GlobalUnlock(hglb55);
    GlobalFree(hglb55);

}
*/
int Entropy(void)
{
    double H=0,alpha,fe,alpha,max1pi,max2pi;
    double old_fe=0,lbmax1pi,lbmax2pi,lb1pi,lb2pi;
    double temp,temp2,temp3,temp4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double TotalPixel;

double Tprob,lg2;

int x,y,i,s,thr1;

unsigned char color;

int far *hist;

double far *Prob;

double far *lb;

HGLOBAL Hhandle,hglb11,Hprob,Hlb;

unsigned char far *hBuf;

Hhandle = GlobalAlloc(GPTR,fi.width);
hBuf = GlobalLock(Hhandle);

hglb11 = GlobalAlloc(GPTR,256*sizeof(int));
hist = (int far *)GlobalLock(hglb11);
Hprob = GlobalAlloc(GPTR,256*sizeof(double));
Prob = (double far *)GlobalLock(Hprob);
Hlb = GlobalAlloc(GPTR,256*sizeof(double));
lb = (double far *)GlobalLock(Hlb);

for(i=0;i<256;i++){
    *(hist+i)=0;
    *(Prob+i)=0;
    lb[i]=0;
}

for (y=0;y<fi.depth;y++){
    ReadMem(hBuf,fi.width,y);
    for (x=0;x<fi.width;x++){
        color = *(hBuf+x);
        hist[color] += 1;
    }
}

TotalPixel = (long)fi.width*(long)fi.depth;
lg2 = log10(2);

for(i=0;i<256;i++){
    *(Prob+i) = *(hist+i);
    *(Prob+i) = *(Prob+i)/TotalPixel;
    Tprob = *(Prob+i);

    if (Tprob > 0)
        *(lb+i) = log10(Tprob)/lg2;
    else
        *(lb+i) = 0;
    H += -(*(Prob+i)*(*(lb+i)));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//      for(alpha = 0.01;alpha<1.00001;alpha=alpha+0.01){
do{
    temp = -(alpha*H);
    temp2=0.0;
    for(i=0;i<256;i++){
        temp2 += *(Prob+i)*(*(lb+i));
        if(temp2<=temp){
            s = i;      // Find s
        }
    }
    temp3=0.0;
    max1pi = *(Prob+0);
    for(i=1;i<=s;i++){
        temp3 += *(Prob+i);
        if(*(Prob+i) > max1pi)
            max1pi = *(Prob+i);
    }
    temp4=0.0;
    max2pi = *(Prob+s+1);
    for(i=s+1;i<256;i++){
        temp4 += *(Prob+i);
        if(*(Prob+i)>max2pi)
            max2pi = *(Prob+i);
    }
    if(max1pi!=0 && max2pi!=0 && temp3!=0 && temp4!=0){
        lbmax1pi = log10(max1pi)/lg2;
        lbmax2pi = log10(max2pi)/lg2;
        lb1pi    = log10(temp3)/lg2;
        lb2pi    = log10(temp4)/lg2;
        fealpha  = ((alpha*lb1pi)/lbmax1pi)+(((1-alpha)*lb2pi)/lbmax2pi);
        if(fealpha >= old_fe){
            thr1 = s;
        }
    }
    old_fe = fealpha;
    alpha = alpha + 0.01;
}while(alpha <= 1.000);

// end loop alpha
GlobalUnlock(Hhandle);
GlobalFree(Hhandle);
GlobalUnlock(hglib1);
GlobalFree(hglib1);
GlobalUnlock(Hprob);
GlobalFree(Hprob);
GlobalUnlock(Hlb);
GlobalFree(Hlb);

return(thr1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void Write2Level(void)
{
    int x,y,Threshold;
    unsigned char color;
    HGLOBAL Hhandle,Dhandle;
    unsigned char far *hBuf;
    unsigned char far *MemBuf;

    Hhandle = GlobalAlloc(GPTR,fi.width);
    hBuf = GlobalLock(Hhandle);
    Dhandle = GlobalAlloc(GPTR,(long)fi.width);
    MemBuf = GlobalLock(Dhandle);

    Threshold = Entropy();
    for (y=0;y<fi.depth;y++){
        ReadMem(hBuf,fi.width,y);
        for (x=0;x<fi.width;x++){
            color = *(hBuf+x);
            if (color >= Threshold)
                color = 250;
            else
                color = 0;
            *(MemBuf+x) = color;
            SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(color));
        }
        WriteMem(MemBuf,fi.width,y);
    }
    GlobalUnlock(Hhandle);
    GlobalFree(Hhandle);
    GlobalUnlock(Dhandle);
    GlobalFree(Dhandle);
}

```

```

int CalEmphasis(int gray,int Threshold)
{

```

```

    float Beta,a,n,Bm,BI;

    int B;

    a = 5;
    Bm = 3;
    BI = 1;

    n = gray; // n=Grey Level at Pixel

```

```

    if(n >=0 && n <= a)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Beta = Bm;

    else

        if(gray > a && gray <= Threshold)

            Beta = -(n*(Bm-BI)/(Threshold - a))+((Threshold*Bm - a*BI)/(Threshold-a));

        else

            Beta = BI;

    B = Beta;

    return(B);

}

void AnatomicChest(void) //use adaptive unsharp masking
{
    int i,j,x,y,B,Point,Threshold;
    int far *Btable;
    float value,temp,temp1,temp2,temp3,temp4;
    unsigned char color;
    unsigned char far *sharp;
    HGLOBAL Hhandle,hglb,Dhandle,Bhandle;
    unsigned char far *hBuf;
    unsigned char far *MemBuf;

    Hhandle = GlobalAlloc(GPTR,fi.width);
    hBuf = GlobalLock(Hhandle);
    hglb = GlobalAlloc(GPTR,(long)fi.width*3); // allocate 3 row
    sharp = (unsigned char far *)GlobalLock(hglb);
    Dhandle = GlobalAlloc(GPTR,fi.width);
    MemBuf = GlobalLock(Dhandle);
    Bhandle = GlobalAlloc(GPTR,256*sizeof(int));
    Btable = (int far *)GlobalLock(Bhandle);

    Threshold = Entropy();

    for(i=0;i<256;i++)

        *(Btable+i) = CalEmphasis(i,Threshold);

    // Point = CalEmphasis(240,Threshold);

    for(j=0;j<3;j++){

        ReadMem(hBuf,fi.width*j);

        for(i=0;i<fi.width;i++) // read first 3 rows using 3x3

            *(sharp+fi.width*j+i) = *(hBuf+i);

    }

    // write first 2 line

    for(x=0;x<fi.width;x++){ // write first line to screen

        SetPixel(hDC,x,fi.depth.PALETTEINDEX(*(sharp+x)));// Line 1

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp = *(sharp+fi.width*1+x); // Center Point
temp1 = *(sharp+fi.width*2+x);
temp2 = *(sharp+fi.width*0+x);
temp3 = *(sharp+fi.width*1+x+1);
temp4 = *(sharp+fi.width*1+x-1);

Point = temp;

B = *(Btable+Point);
value = temp + B*(temp1+temp2+temp3+temp4)/4;
if(value <0)
    value = 0;
else if (value>255)
    value = 255;
SetPixel(hDC,x,fi.depth-1,PALETTEINDEX(value)); // Line 2
}

```

```

//*****//
for(y=2;y<fi.depth-1;y++){ // Calculate Sharpening loop
    ReadMem(hBuf,fi.width,y);
    for(i=0;i<fi.width;i++){
        *(sharp-i) = *(sharp+fi.width*1+i); // row 0 = row 1
        *(sharp-fi.width*1+i) = *(sharp+fi.width*2+i); // row 1 = row 2
        *(sharp+fi.width*2+i) = *(hBuf+i);
    }
    for(x=1;x<fi.width;x++){
        temp = *(sharp+fi.width*1+x); // Center Point
        temp1 = *(sharp+fi.width*2+x);
        temp2 = *(sharp+fi.width*0+x);
        temp3 = *(sharp+fi.width*1+x+1);
        temp4 = *(sharp+fi.width*1+x-1);

        Point = temp;
        B = *(Btable+Point);

        value = temp + B*(temp1+temp2+temp3+temp4)/4;
        if(value <0)
            value = 0;
        else if (value>255)
            value = 255;

        *(MemBuf+x) = value;
        SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(value));
    }
    WriteMem(MemBuf,fi.width,y);
}

//*****// Scroll Row up

```

GlobalUnlock(Hhandle);

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GlobalFree(Hhandle);
GlobalUnlock(hglb);
GlobalFree(hglb);
GlobalUnlock(Dhandle);
GlobalFree(Dhandle);
GlobalUnlock(Bhandle);
GlobalFree(Bhandle);
}

void AnatomicEqualize(void)
{
HGLOBAL hglb11,hglb22,hglb33,hglb55,Hhandle;
unsigned char far *Hbuf;
unsigned char far *MemBuf;
int i,x,y,NumPart2;
char ch;
unsigned char color;
long int far *hist;
unsigned char far *table;
long int far *Cumulative;
int temp,termp2,Threshold;
double temp11>TotalPixel>TotalPart1>TotalPart2;
float temp1;

hglb11 = GlobalAlloc(GPTR,256*sizeof(long int));
hist = (long int far *)GlobalLock(hglb11);
hglb33 = GlobalAlloc(GPTR,256*sizeof(long int));
Cumulative = (long int far *)GlobalLock(hglb33);
hglb55 = GlobalAlloc(GPTR,256*sizeof(unsigned char));
table = (unsigned char far *)GlobalLock(hglb55);
Hhandle = GlobalAlloc(GPTR,fi.width);
Hbuf = GlobalLock(Hhandle);
hglb22 = GlobalAlloc(GPTR,fi.width);
MemBuf = GlobalLock(hglb22);

/***** HISTOGRAM EQUALIZATION *****/
Threshold = Entropy();
TotalPixel = (long)fi.width*(long)fi.depth;
NumPart2 = 256-Threshold;

for(i=0;i<256;i++)
*(hist+i) = 0;

for(y=0;y<fi.depth;y++){ // create histogram
ReadMem(Hbuf,fi.width,y);
for(x=0;x<fi.width;x++){
temp = *(Hbuf+x);
hist[temp] += 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    }

    TotalPart1 = 0;
    TotalPart2 = 0;
    for(i=0;i<256;i++){
        if (i<Threshold){
            TotalPart1 += *(hist+i);
            *(Cumulative+i) = TotalPart1;
        }else{
            TotalPart2 += *(hist+i);
            *(Cumulative+i)= TotalPart2;
        }
    }

    for(i=0;i<256;i++){
        if(i<Threshold){
//
            temp11 = ceil(*(Cumulative+i)*Threshold/TotalPart1)-1;
            temp11 = ceil(*(Cumulative+i)*256/TotalPart1)-1;
            if(temp11>255)
                temp11 = 255;
            else
                if (temp11<0)
                    temp11 = 0;
            *(table+i) = temp11;
        }else{
//
            temp11 = ceil(*(Cumulative+i)*NumPart2/TotalPart2)-1;
            temp11 = ceil(*(Cumulative+i)*256/TotalPart2)-1;
            temp11 = i;
            if(temp11>255)
                temp11 = 255;
            else
                if (temp11<0)
                    temp11 = 0;
            *(table+i) = temp11;
        }
    }

    for(y=0;y<fi.depth;y++){
        ReadMem(Hbuf,fi.width,y);
        for(x=0;x<fi.width;x++){
            temp = *(Hbuf+x);
            temp2 = *(table+temp);
            *(MemBuf+x) = temp2;
            SetPixel(hDC,x,fi.depth-y,PALETTEINDEX(temp2));
        }
        WriteMem(MemBuf,fi.width,y);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GlobalUnlock(hglb11);
GlobalFree(hglb11);
GlobalUnlock(hglb22);
GlobalFree(hglb22);
GlobalUnlock(hglb33);
GlobalFree(hglb33);
GlobalUnlock(hhandle);
GlobalFree(hhandle);
GlobalUnlock(hglb55);
GlobalFree(hglb55);

}

/*****end program*****/

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายมนัส นามะสนธิ
วันเดือนปีเกิด	วันที่ 21 เมษายน พ.ศ. 2513
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตรบัณฑิต สาขาสถิติประยุกต์
สถานที่สำเร็จการศึกษา	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประสบการณ์การทำงาน	บริษัทไทยแมนเนจเม้นท์ไฮด์ (TMS) ตำแหน่งโปรแกรมเมอร์ เมื่อปี พ.ศ. 2536 ถึง 2537
อาชีพปัจจุบัน	บริษัทไทยแมนเนจเม้นท์ไฮด์ (TMS) ตำแหน่งโปรแกรมเมอร์

