สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การลดข้อมูลภาพ

IMAGE COMPRESSION

โดย

นายวัชรพัฐ การะพิงค์ รหัส 39013204

อาจารย์ที่ปรึกษา

รศ.ดร. มนัส สังวรศิลป์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่.....
เลขทะเบียน 34013
วัน, เดือน, ปี 29 มิ.ย. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การลดข้อมูลภาพ

ผู้จัดทำ

นายวัชรพัฐ การะพิงค์ รหัส 39013204

(รศ.ดร. มนัส สังวรศิลป์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลดข้อมูลภาพ

IMAGE COMPRESSION

ผู้จัดทำ

นายวัชรพัฐ การะพิงค์ รหัส 39013204

โครงงานได้รับการตรวจสอบแล้ว พร้อมที่จะทำการทดสอบ

(รศ.ดร. มนัส สังวรศิลป์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลดข้อมูลภาพ

นายวัชรพัฐ การะพิงค์ รหัส 39013204
อาจารย์ที่ปรึกษา รศ.ดร.มนัส สังวรศิลป์

บทคัดย่อ

ในโครงงานฉบับนี้ เป็นการศึกษาและทดลองพื้นฐานของการประมวลผลสัญญาณภาพ การแปลงสัญญาณภาพเป็นสัญญาณไฟฟ้า และการแปลงสัญญาณไฟฟ้าให้เป็นสัญญาณดิจิตอล ซึ่งเก็บไว้ในรูปแบบของตัวเลขและนำข้อมูลตัวเลขที่ได้มาเหล่านั้นมาทำการบีบอัดให้ได้ข้อมูลที่น้อยลง โดยคอมพิวเตอร์ซึ่งข้อมูลจะน้อยลงเท่าไรขึ้นอยู่กับวิธีการบีบอัดข้อมูลภาพ มีประโยชน์มากในการสื่อสารมูลภาพ ในปัจจุบันกำลังเป็นที่นิยมในการส่งข้อมูลทางอินเตอร์เน็ทและการใช้รูปภาพประกอบใน การทำเวปไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMAGE COMPRESSION

Mr.Watcharhapat Karraping Code 39013204

Advisor Asst.Prof.Dr.Manus Sungvomsilp

Academic Year 1998

Abstract

This project aims to present basic concept of image compression. Transform nalog signal from camara Is digital signal. We present the they of imagecompress by computor. The use of image compression is multimedia communication and data transmition Internet that expected to be popular in the future.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

ปัจจุบันคอมพิวเตอร์มีบทบาทสำคัญในชีวิตประจำวันของมนุษย์ ไม่ว่าจะเพื่อด้านความบันเทิงด้านการศึกษาทดลองหรือด้านธุรกิจและเป็นแหล่งใช้ติดต่อสื่อสารข้อมูล ภาพเป็นสิ่งสำคัญที่ใช้ในการสื่อสารทางคอมพิวเตอร์ จึงได้เกิดการประมวลผลสัญญาณภาพที่เป็น 2 มิติเรียกว่าดิจิตอลอิเมจโปรเซสซิ่ง

ดิจิตอลอิเมจโปรเซสซิ่ง จะเป็นกระบวนการใช้ประมวลผลภาพแล้วเก็บไว้ในรูปแบบของดิจิตอล(ตัวเลข) เช่น การหาขอบภาพ การทำภาพให้ภาพมีความคมชัดขึ้นและที่สำคัญอีกอย่างหนึ่งคือการลดข้อมูลภาพ ซึ่งปัจจุบันมีบทบาทมากในการส่งข้อมูล เพราะเมื่อข้อมูลภาพละเอียดมากทำให้เสียเวลาเสียค่าใช้จ่ายในการส่ง เช่น การส่งข้อมูลทางอินเตอร์เน็ท การส่งข้อมูลผ่านดาวเทียม

ในโครงงานนี้ การลดข้อมูลภาพ เป็นการประมวลผลสัญญาณภาพโดยเก็บข้อมูลภาพจากกล้อง(camara) เปลี่ยนสัญญาณภาพให้เป็นสัญญาณไฟฟ้าและเปลี่ยนสัญญาณไฟฟ้าเป็นสัญญาณดิจิตอลเพื่อจะนำมาประมวลผล โดยเป็นการลดข้อมูลภาพที่ได้มาจากกล้องเพื่อนำไปใช้ประโยชน์ต่อไป ขอขอบคุณ รศ.ดร.มนัส สังวรศิลป์ และรุ่นพี่ปริญญา โท ที่กรุณาให้คำแนะนำต่างๆ ทำให้เกิดความเข้าใจในโครงงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
คำนำ	III
บทที่ 1 บทนำ	1
1.1 ดิจิตอลอิมเมจโปรเซสซิ่ง	1
1.2 วัตถุประสงค์ของวิทยานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2 หลักการประมวลผลภาพ	2
2.1 หลักการพื้นฐานทางอิมเมจโปรเซสซิ่ง	2
2.2 ความหมายและนิยามของภาพในระบบดิจิตอล	4
2.3 การแทนรูปภาพด้วยระบบดิจิตอล	5
2.4 ระบบการประมวลผลทางดิจิตอล	5
2.5 การสุ่มแบบสม่ำเสมอและควอนไตเซชัน	7
2.6 เทคนิคต่าง ๆ ในการประมวลผลภาพ	8
2.6.1 อิเมจดิจิไตเซชัน (image digitization)	8
2.6.2 อิเมจเอนฮานเมนต์และรีสตอเรชัน	8
2.6.3 อิเมจเอ็นโค๊ดดิ้ง (image encoding)	9
2.6.4 อิเมจรีคอนสตรัคชัน (image reconstruction)	9
บทที่ 3 การลดข้อมูลภาพ	10
3.1 พรีดิกทีฟโค๊ดดิ้ง	10
3.2 ทรานส์ฟอร์มโค๊ดดิ้ง	11
3.3 ไฮบริดโค๊ดดิ้ง	12
3.4 การลดข้อมูลภาพด้วยวิธีรันเลงธ์	13
3.5 ความสามารถในการลดข้อมูลภาพด้วยรันเลนธ์	14
3.6 ข่าวสาร ปริมาณข่าวสารเฉลี่ย และส่วนที่ซ้ำ	14
3.7 ดิสครีตโคไซน์ทรานส์ฟอร์มโค๊ดดิ้ง	15
3.8 อัฟแมนโค๊ดดิ้ง	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 หลักการเบื้องต้นของการลดข้อมูลภาพ 22
4.1 Discrete Cosine Transform 22
4.2 การวัดความเหมือนจริงของภาพ 30
4.3 การควอนไทซ์ (Quantization) 34
บทที่ 5 การทำงานของการประมวลผลภาพ 38
การทำงานของโปรแกรม 39
บทที่ 6 การทดลองการลดข้อมูลภาพ 43
การทดลองที่ 6.1 43
การทอลองที่ 6.2 45
บทที่ 7 สรุปและวิจารณ์ผลการทดลอง 46
ปัญหาและอุปสรรค์ในโครงงาน
เอกสารอ้างอิง
กิตติกรรมประกาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้า
รูปที่ 2.1 ระบบการประมวลผลทางดิจิตอล	6
รูปที่ 3.1 ฮัฟแมนทรี เมื่อผ่านไป 2 รอบ	17
รูปที่ 3.2 แผนภาพแสดงแมนทรี	18
รูปที่ 3.3 ตัวอย่างการเข้ารหัสข้อมูล	19
รูปที่ 3.4 แสดงการถอดรหัสเมื่อได้รับรหัส 0	19
รูปที่ 3.5 แสดงการถอดรหัสเมื่อได้รับรหัส 1	20
รูปที่ 3.6 แสดงการถอดรหัสเมื่อได้รับรหัส 11	20
รูปที่ 3.7 แสดงการถอดรหัสเมื่อได้รับรหัส 111	21
รูปที่ 4.1 (ก) แสดงการแปลงแบบฟูริเยร์และการแปลงแบบดีสครีตโคไซน์ตามลำดับ	24
รูปที่ 4.1(ข) แสดงการแปลดีสครีตโคไซน์แบบ 2 มิติ	24
รูปที่ 4.1(ค) แสดงการแปลงดีสครีตโคไซน์ที่ทำในบล็อกขนาด 8 x 8	25
รูปที่ 4.1 แผนภาพแสดงตอน Forward Transform ของ DCT	26
รูปที่ 4.2 ตัวอย่างการกำหนดบิทให้กับสัมประสิทธิ์ในโดเมน ความถี่หลังการแปลงด้วย DCT กับบล็อคข้อมูลนาด 16*16 ซึ่งสามารถลดข้อมูลลงได้ จาก 8 บิทต่อจุดภาพให้เหลือ 1.5 บิทต่อจุดภาพ	28
รูปที่ 4.3 แสดงแผนภาพการแปลงกลับเมตริกซ์	29
รูปที่ 4.3 แสดงแผนภาพการแปลงกลับเมตริกซ์	33
รูปที่ 4.4 แสดงแผนภาพการแปลงกลับเมตริกซ์	33
รูปที่ 4.5 แสดงการกระจายของความน่าจะเป็นแบบยูนิฟอร์ม	34
รูปที่ 4.6 แสดงระดับของการควอนไทซ์	35
รูปที่ 4.7 แสดงภาพ (ก) และ (ข) แสดงการควอนไทซ์แบบมิดเทรดและแบบมิดไรซ์ตามลำดับ	37
รูปที่ 5.1 แสดงปลีอกไดอะแกรมการประมวลผลภาพ	38
รูปที่ 5.2 แสดงหน้อ จอ Video Previwe Record Mode	39
รูปที่ 5.3 เมนูหลัก Playback and Compression Mode	40
รูปที่ 5.4 เมนูหลัก Open bmp Mode	41
รูปที่ 5.5 เมนูการลดข้อมูลภาพนิ่ง	41
รูปที่ 6.1 ภาพความละเอียดภาพต่ำ	43
รูปที่ 6.2 ภาพความละเอียดภาพสูง	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.3 แสดงภาพที่บันทึกจากกล้อง 43
รูปที่ 6.4 ภาพบันทึกจากกล้องความละเอียดภาพต่ำ 45
รูปที่ 6.5 ภาพบันทึกจากกล้องความละเอียดสูง 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงจำนวนไบท์ที่ใช้ในการเก็บภาพ เมื่อ N และ M เปลี่ยนไป	8
ตารางที่ 6.1.1 แสดงผลการทดลองการลดข้อมูลแบบ Huffman	44
ตารางที่ 6.1.2 แสดงผลการทดลองการลดข้อมูลแบบ Runlength	44
ตารางที่ 6.2 แสดงการทดลองการลดข้อมูลภาพเคลื่อนไหว	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ดิจิตอลอิมเมจโปรเซสซิ่ง (Digital Image Processing)

ดิจิตอลอิมเมจโปรเซสซิ่ง โดยทั่วไปหมายถึง การประมวลผลของภาพใน 2 มิติ โดยคอมพิวเตอร์ หรือถ้าจะกล่าวในความหมายที่กว้างขึ้น จะหมายถึง การประมวลผลของข้อมูล 2 มิติใด

ดิจิตอลอิมเมจโปรเซสซิ่ง ได้มีการใช้งานอย่างกว้างขวาง เช่น การส่งข้อมูลผ่านดาวเทียมและยานอวกาศ , การส่งข้อมูลและเก็บข้อมูลภาพในธุรกิจ , เรดาร์ , โซนาร์ , การตรวจสอบชิ้นส่วนอัตโนมัติในโรงงาน

รูปที่ได้จากดาวเทียมมีประโยชน์ในการหาทรัพยากรธรรมชาติ , การทำแผนที่ , การทำนายผลิตผลทางการเกษตร , การเจริญเติบโตของเมือง , การพยากรณ์อากาศ และอื่น ๆ อีกมาก

การส่งเสริมและบันทึกข้อมูลภาพได้ใช้ในการออกอากาศทางโทรทัศน์ , การประชุมทางไกล (teleconference) , การรับ – ส่ง แฟกซ์ , การสื่อสารของระบบเครือข่ายคอมพิวเตอร์ , โทรทัศน์วงจรปิด , การสื่อสารทางทหาร ในทางการแพทย์ก็เช่น การประมวลผลภาพจาก x – ray และ ultrasonic scaning โดยภาพดังกล่าวสามารถนำมาวินิจฉัยโรคได้

เราสามารถสรุปได้ว่า เมื่อใดก็ตามที่มนุษย์หรืออุปกรณ์ใด ๆ ก็ตามได้รับข้อมูลและประมวลผลข้อมูลที่เป็น 2 มิติ เราก็จะเรียกได้ว่าเป็น อิมเมจโปรเซสซิ่ง

การประมวลผลสัญญาณภาพได้มีการประยุกต์ใช้งานอย่างกว้างขวาง ตัวอย่างเช่น อิมเมจเอนฮานเมนต์ (Image Enhancement) , อิมเอจรีสตอเรชั้น (Image restoration) ดังนั้นการใช้งานก็ต่างกันไป อย่างเช่น อิมเมจเอ็นฮานเมนต์ ใช้ปรับปรุงคุณภาพของภาพให้ดีขึ้น

1.2 วัตถุประสงค์ของปริญญานิพนธ์

เพื่อจะนำเสนอวิธีการลดข้อมูลภาพสำหรับลดพื้นที่ในการเก็บข้อมูล อีกทั้งยังประหยัดค่าใช้จ่ายและเวลาในการส่งข้อมูลเป็นระยะทางไกลๆ เช่น การส่งข้อมูลผ่านดาวเทียม การส่งข้อมูลในอินเทอร์เน็ต

1.3 ขอบเขตของปริญญานิพนธ์

เนื้อหาของปริญญานิพนธ์นี้จะเกี่ยวกับหลักการเบื้องต้นของอิเมจโปรเซสซิ่ง และการประยุกต์การใช้งาน รวมถึงการลดข้อมูลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการประมวลผลภาพ

2.1 หลักการพื้นฐานทางอิมเมจโปรเซสซิ่ง

อิมเมจโปรเซสซิ่ง (image processing) เป็นกระบวนการที่ใช้ในการจัดการข้อมูลที่เป็นรูปภาพต่างๆให้อยู่ในลักษณะของสัญญาณไฟฟ้าเพื่อที่จะได้นำข้อมูลที่เป็นสัญญาณไฟฟ้านั้นไปใช้ประโยชน์ทางอื่น เป็นต้นว่า การตกแต่ง , การส่งรูปภาพไปตามสายนำสัญญาณจากที่แหล่งหนึ่งไปยังอีกแห่งหนึ่ง (ซึ่งก็คือหลักการของโทรสาร) , การเก็บข้อมูลรูปภาพไว้ในหน่วยความจำเพื่อทำอัลบัมภาพอิเล็กทรอนิกส์เพื่อใช้ประโยชน์เป็นแฟ้มข้อมูลพนักงาน , แฟ้มอาชญากรรม เป็นต้น นอกเหนือไปจากนี้ยังสามารถนำไปใช้ในงานด้านการรักษาความปลอดภัย , ตรวจสอบลายนิ้วมือ ได้อีกด้วย

พิกเซล (pixel)

ภาพที่จะถูกแปลงเป็นสัญญาณไฟฟ้า จะได้รับการแบ่งรายละเอียดของภาพเป็นตารางเล็ก ๆ และตารางเล็ก ๆ นี้เองคือพิกเซล (pixel) และเมื่อมีการจัดเรียงพิกเซลก็จะเกิดพิกเซลที่เป็นแถว (row) และคอลัมน์ (column) โดยมีจำนวนแถวทางแนวนอนเป็น N แถว และมีจำนวนแถวทางแนวตั้ง M แถว ซึ่งในแต่ละตำแหน่งของพิกเซลจะแทนด้วย P(j, j) โดยที่ I และ j จะเป็นเลขจำนวนเต็มและที่เรียกการจัดเรียงของพิกเซลว่าพิกเซลเมตริกซ์ (pixel matrix)

เมื่อทราบตำแหน่งของพิกเซลแล้วก็จำเป็นต้องทราบว่าที่ตำแหน่งนั้น ๆ พิกเซลมีค่าเท่าไหร่ ซึ่งค่าที่ว่านี้ก็คือค่าเฉลี่ยความเข้มของแสงที่ตกกระทบบนตำแหน่งของแต่ละพิกเซลซึ่งมีค่าอยู่ระหว่าง 0 กับ 1 เมื่อทำเป็นภาพที่มีความเข้มสองระดับ

ลักษณะข้อมูลภาพ

1. ภาพ 2 ระดับ คือ มีแค่จุดขาวกับดำเท่านั้น โดยแต่และจุดเป็นข้อมูล 1 บิต

2. ภาพ 16 ระดับ ซึ่งในแต่ละจุดภาพจะเป็นข้อมูล 4 บิต ซึ่งทำให้สามารถแสดงภาพได้ 16 ระดับสี หรือ 16 ระดับสีเทา ขึ้นอยู่กับว่าภาพนั้นเป็นภาพสีหรือภาพขาว – ดำ

3. ภาพ 256 ระดับซึ่ง ในแต่ละจุดภาพจะเป็นข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงภาพได้ 256 ระดับสี หรือระดับสีเทา ขึ้นอยู่กับส่าภาพนั้นเป็นภาพสี หรือ ขาว – ดำ

4. ภาพ true Color ซึ่ง ในแต่ละจุดจะเป็นข้อมูลขนาด 24 บิต ทำให้สามารถแสดงผลภาพได้เหมือนภาพจริงที่สุด เพราะสามารถแสดงสีได้ถึง 16,777,216 สี ภาพ true color สามารถแสดงได้เฉพาะภาพสีเท่านั้น ไม่สามารถแสดงภาพขาว – ดำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงภาพนี้ใช้วิธี ตั้งค่าของแม่สีในตารางสี โดยอาจเลือกสีเป็นแบบ 16 สี จาก 64 สี หรือ 16 สี จาก 262,144 สี หรือ 256 สี จาก 262,144 สี ขึ้นอยู่กับโหมด การแสดงผล สำหรับ true color ไม่มีการเลือกสี แสดงผลโดยการส่งค่าสี RGB ผ่าน D/A สีละ 8 บิต ออกไปเลย ความแตกต่างของการแสดงภาพสีและภาพขาวดำคือ ภาพขาวดำจะต้องตั้งให้แม่สีทั้ง 3 สีมีค่าเท่ากัน เนื่องจาก VGA กำหนดให้แม่สีแต่ละสีใช้รีจิสเตอร์ (register) 6 บิต ทำให้แม่สีแต่ละแม่สีแสดงผลได้เพียง 34 ระดับเท่านั้น ยังผลให้เราแสดงภาพ 256 ระดับให้เห็นได้เพียง 64 ระดับเท่านั้น หากต้องการให้เห็นจริงทั้ง 256 ระดับต้องแสดงใน true color mode แล้วให้ RGB มีค่าเท่ากัน ซึ่งโหมดนี้ใช้รีจิสเตอร์ 8 บิต สำหรับแม่สีแต่ละสี

เกรย์สเกล (gray scale)

เกรย์สเกล หมายถึง ความแตกต่างของระดับความเข้มของแสง โดยเกรย์สเกลหนึ่ง ๆ อาจแบ่งเป็น 13 , 20 หรือ 9 ระดับ โดยระดับที่ว่านี้ก็คือ ระดับสีเทา ในภาพหนึ่ง ๆ ถ้าต้องการแบ่งระดับความเข้มแสงหรือระดับสีเทาให้มีหลาย ๆ ค่านั้น จำเป็นอย่างยิ่งที่จะต้องเพิ่มจำนวนบิต ที่แสดงค่าของพิกเซล ตัวอย่างเช่น ถ้าต้องการภาพที่มีระดับสีเทา 4 ระดับ ต้องแทนด้วยเลขฐานสองจำนวน 2 บิต ถ้าต้องการภาพที่มีระดับสีเทา 16 ระดับ ต้องแทนด้วยเลขฐานสองจำนวน 4 บิต และถ้าต้องการภาพที่มีระดับสีเทา 256 ระดับต้องแทนด้วยเลขฐานสองจำนวน 8 บิตเป็นต้น

จำนวนระดับสีเทาที่ต้องการนี้ก็คือค่าเลขยกกำลังของ 2 นั่นเอง ซึ่งค่าต่ำสุดหรือ 0 จะแทนสำดำหรือไม่มีความสว่างเลย และค่าที่มากที่สุดก็คือค่าที่น้อยกว่าจำนวนระดับสีเทาอยู่ 1 เช่น ค่า 15 ในระดับที่มีระดับสีเทา 16 ระดับก็จะเป็นสีขาวหรือสว่างมากที่สุด เป็นต้น

ในยุคแรก ๆ ระบบการมองเห็น (vision system) จะใช้ระบบเลขฐานสองเพราะสะดวกต่อการนำเซนเซอร์มาใช้ นอกจากนี้การรวบรวมข้อมูล การเก็บรักษาข้อมูลยังสามารถทำได้ง่ายอีกด้วย

ในปัจจุบัน ไมโครโปรเซสเซอร์ที่ใช้กันโดยทั่วไป ขนาดที่เล็กที่สุดก็คือ 8 บิต ดังนั้นเกรย์สเกลขนาด 8 ค่า , 16 ค่า และ 256 ค่า จึงไม่เป็นปัญหาในการประมวลผล

ความสามารถในการแบ่งแยกระดับความแตกต่างของสายตามนุษย์ โดยทั่ว ๆ ไปจะอยู่ระหว่าง 10 ถึง 15 ระดับ ดังนั้น เกรย์สเกลขนาด 16 ระดับ จึงถือได้ว่าใกล้เคียงกับสายตามนุษย์หรืออาจจะละเอียดน้อยกว่าสายตามนุษย์บ้างเล็กน้อย (ในบางคน) ในขณะที่เกรย์สเกลขนาด 64 หรือ 256 นั้นละเอียดเกินไปสำหรับมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮีสโตแกรม (Histogram)

ฮีสโตแกรมคือกราฟที่บอกให้ทราบถึงจำนวนของระดับสีเทาในภาพหนึ่ง ๆ โดยที่แกน x จะเป็นค่าของระดับสีเทา และแกน y เป็นจำนวนของพิกเซล ในฮีสโตแกรมหนึ่ง ๆ จะประกอบไปด้วย

1. จำนวนพิกเซลทั้งหมดของภาพ
2. จำนวนพิกเซลในแต่ละค่าของระดับสีเทา
3. กราฟที่แสดงจำนวนในแต่ละค่าของระดับสีเทา

กราฟที่ใช้ในฮีสโตแกรมจะเป็นกราฟแท่ง ซึ่งสามารถแสดงจำนวนพิกเซลในแต่ละค่าระดับสีเทาได้เป็นอย่างดี

รูปร่างหรือขนาดของฮีสโตแกรมจะเป็นข้อมูลที่แสดงคุณสมบัติของภาพว่า มีความคมชัด (contrast) มากน้อยเพียงใด ซึ่งข้อมูลนี้ก็คือประโยชน์ของฮีสโตแกรมที่จะใช้ในการกำหนดค่าเทรชโฮล (threshold) ค่าเทรชโฮลนี้จะใช้ในการแปลงรูปภาพให้กลายเป็นภาพที่มีระดับความเข้ม 2 ระดับ คือขาวกับดำ หรือ "0" กับ "1" ตามที่ได้กล่าวมาแล้วในเรื่องพิกเซลนั่นเอง

การสร้างฮีสโตแกรม

1. ต้องกำหนดก่อนว่าภาพที่จะนำมาสร้างฮีสโตแกรมนั้น จะแบ่งเป็นกี่พิกเซล
2. สร้างพิกเซลเมตริกซ์จากพิกเซลเล็ก ๆ
3. นำค่าของพิกเซลในพิกเซลเมตริกซ์ที่ได้จากข้อ 2 มาสร้างตารางความสัมพันธ์ระหว่างค่าระดับสีเทากับจำนวนของพิกเซลในแต่ละค่าระดับสีเทาว่ามีกี่พิกเซล
4. นำค่าที่ได้จากตารางในข้อ 3 มาพล็อตเป็นกราฟแท่ง โดยแกนทางแนวนอนเป็นค่าระดับสีเทา และแกนทางแนวตั้งเป็นแกนของจำนวนของพิกเซล และกราฟนี้คือฮีสโตแกรมนั่นเอง

2.2 ความหมายและนิยามของภาพในระบบดิจิตอล

ภาพ (Image) ในเชิงคณิตศาสตร์จะหมายถึง ฟังก์ชัน 2 มิติ f(x, y) โดย x และ y เป็นแกนพิกัดในระนาบ 2 มิติ ค่าฟังก์ชัน f(x, y) จะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพที่ตำแหน่ง x, y ซึ่งเราเรียกว่า ระดับสีเทา (Gray Scale) ซึ่งปกติเราจะให้จุดกำเนิดของแกนพิกัด (Coordinate) อยู่ทางมุมซ้ายของภาพ ภาพ 2 มิติที่แทนด้วยฟังก์ชัน f(x, y) โดย x และ y เป็นแกนในระนาบของภาพ ค่าของฟังก์ชันที่จุด (x, y) คือความเข้มแสงที่จุดนั้น เนื่องจากแสงเป็นพลังงานรูปหนึ่ง ดังนั้น f(x, y) ต้องไม่เป็นศูนย์ และมีค่า (finite) นั้นคือ

$$0 < f(x, y) < a \qquad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยธรรมชาติของแสง ซึ่งจะต้องมีแหล่งกำเนิดแสงและส่วนที่สะท้อนของแสง ดังนั้นเราสามารถแยกฟังก์ชัน f(x, y) ออกเป็นสองส่วนคือ อิลูมิเนชันคอมโพเนนท์ (illumination component) และรีเฟล็กแทนท์คอมโพเนนท์ (reflectant component) จะได้ว่า

$$f(x, y) = I(x, y) * r(x, y) \qquad (2.2)$$

เมื่อ

$$0 < I(x, y) < a \qquad (2.3)$$

และ

$$0 < r(x, y) < 1 \qquad (2.4)$$

สมการที่ 2.4 แสดงให้เห็นว่า ฟังก์ชันการสะท้อนถูกจำกัดขอบเขตระหว่าง 0 (ซึ่งหมายถึงการดูดซึมสมบูรณ์) และ 1 (ซึ่งหมายถึง การสะท้อนโดยสมบูรณ์) ธรรมชาติของ i (x, y) ขึ้นอยู่กับแหล่งกำเนิดแสง ในขณะที่ r (x, y) ขึ้นอยู่กับวัตถุที่สะท้อนแสงมาเข้าตา

ดังที่กล่าวมาแล้ว ความเข้มของภาพที่จุด (x, y) เราเรียกว่าระดับสีเทา (gray level) I จากสมการที่ (2.2) ถึง (2.4) จะเห็นว่า I อยู่ในช่วง

$$L_{min} < I < L_{min} \qquad (2.5)$$

ในทางทฤษฎี L ต้องมีค่าเป็นบวก ในขณะที่ L ต้องมีค่าน้อยกว่าอนันต์ ในทางปฏิบัติ L = Lr และ L = L r ช่วงของ (L , L) เราเรียกว่าช่วงของระดับสีเทา ในทางปฏิบัติโดยใช้หลักคณิตศาสตร์ เรานิยมปรับช่าง (L , L) ให้เป็นช่วง (O ,L) โดย L = O หมายถึง ดำสนิท และ L = 1 หมายถึง ขาว

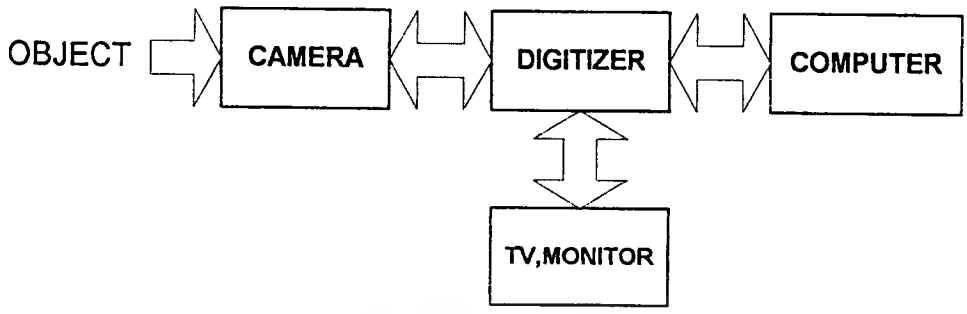
2.3 การแทนรูปภาพด้วยระบบดิจิตอล

ภาพดิจิตอล (digital image) เป็นภาพที่ถูกแปลงมาจากภาพอนาลอก อยู่ในรูปตัวเลข โดยภาพอนาลอกถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่าพิกเซล ในแต่ละพิกเซลจะถูกระบุตำแหน่งโดย (x, y) และค่าระดับสีเทาของพิกเซล นั้น คือค่าของ f(x, y)

2.4 ระบบการประมวลผลทางดิจิตอล

ระบบการประมวลผลทางดิจิตอลประกอบด้วย 3 ส่วนใหญ่ ๆ คือ ส่วนเปลี่ยนสัญญาณอนาลอก ให้เป็นสัญญาณดิจิตอล ซึ่งเรียกว่า ดิจิไตเซอร์ (Digitizer) ส่วนประมวลผล (Processing) และส่วนแสดงผล (Display) ดังแสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ระบบการประมวลผลทางดิจิตอล

จากรูปที่ 2.1 ส่วนแรกคือ ส่วนที่เปลี่ยนสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอล กล้อง (camera) เปรียบเสมือนดวงตาของมนุษย์ ทำหน้าที่เปลี่ยนภาพวัตถุมาเป็นสัญญาณทางไฟฟ้าและส่งให้ดิจิไตเซอร์ (Digitizer) ซึ่งทำหน้าที่เปลี่ยนสัญญาณไฟฟ้าให้เป็นสัญญาณดิจิตอล อุปกรณ์ส่วนนี้ได้แก่ กล้องโทรทัศน์ดิจิไตเซอร์ ซึ่งในภาพประกอบด้วยหลอดที่ทำหน้าที่เป็นสื่อไฟฟ้าทางแสง ภาพถูกโฟกัสลงบนผิวของหลอด และถูกเปลี่ยนให้เป็นสัญญาณไฟฟ้าที่สอดคล้องกับความสว่างของภาพในตำแหน่งนั้น ๆ จากนั้น ทำการควอนไตซิ่ง (quantizing) ข้อมูลภาพที่ได้เป็นสัญญาณดิจิตอล

ส่วนประมวลผล คือ คอมพิวเตอร์ ซึ่งเปรียบเสมือนสมอง ทำหน้าที่ประมวลผลและวิเคราะห์ข้อมูลภาพ

ส่วนแสดงผลทำหน้าที่เปลี่ยนข้อมูลตัวเลข (ซึ่งเป็นระดับสีเทา) ที่เก็บเป็น อะเรย์ (array) ในคอมพิวเตอร์ ให้อยู่ในรูปที่เหมาะสม และสื่อความหมายกับมนุษย์ได้ คือเป็นภาพที่ปกติทั่ว ๆ ไป อุปกรณ์ในส่วนนี้ได้แก่ จอ (monitor) ทีวี เครื่องพิมพ์ดีดที่สามารถผลในรูปกราฟฟิกส์ได้

ภาพ 1 ภาพ ที่ถูกเปลี่ยนจากสัญญาณดิจิตอล สำหรับคอมพิวเตอร์ที่มีขนาดใหญ่ขึ้นอยู่กับความละเอียดของภาพที่ต้องการ และจะมีผลทำให้ใช้เนื้อที่ในหน่วยความจำมากในการเก็บข้อมูลภาพ 1 ภาพ เช่น การเก็บภาพขนาด 256*256 จุด ที่มีความแตกต่างของระดับความเข้มของแต่ละจุดเท่ากับ 256 ระดับ จะต้องใช้เนื้อที่ในหน่วยความจำถึง 64,000 ไบท์ ในการเก็บภาพนี้ ดังนั้นในปัจจุบันนี้ได้มีการวิจัยหาวิธีที่จะเก็บภาพด้วยคอมพิวเตอร์ โดยใช้เนื้อที่หน่วยความจำให้น้อยที่สุดและยังคงรักษาความละเอียดของภาพตามการใช้งานได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การสุ่มแบบสม่ำเสมอและควอนไตเซชัน (Uniform sampling and Quantization)

เพื่อที่จะประมวลผลสัญญาณภาพด้วยระบบคอมพิวเตอร์ ฟังก์ชันของภาพ f(x, y) จะถูกทำให้เป็นสัญญาณไม่ต่อเนื่อง ทั้งระนาบของภาพ ซึ่งเราเรียกว่า การสุ่มภาพ (Image Sampling) ของฟังก์ชันที่ได้เรียกว่า การควอนไตเซชันระดับสีเทา (gray level quantization)

สมมติว่าสัญญาณภาพต่อเนื่อง f(x, y) ถูกดิจิไตซ์ ในระนาบ X Y เป็นช่วงเท่า ๆ กัน เราสามารถจัด f(x, y) ให้อยู่ในรูปเมตริกซ์ขนาด NxN ได้ดังสมการ

$$f(x, y) = \begin{pmatrix} f(0,0) & f(0,1) & \ldots\ldots\ldots\ldots & f(0,N-1) \\ f(0,1) & f(1,1) & \ldots\ldots\ldots\ldots & f(1,N-1) \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ f(N-1,0) & f(N-1,1) & \ldots\ldots & f(N-1,N-1) \end{pmatrix} \quad (2.6)$$

ทางขวามือของสมการ จะเรียกว่า ภาพดิจิตอล และทุก ๆ สมาชิกของเมตริกซ์จะเรียกว่า พิกเซล จากขบวนการสร้างภาพดิจิตอลข้างต้น จะเห็นว่า เราต้องทราบขนาดของความละเอียดของภาพ NxN พิกเซล และจำนวนระดับของ Gray level ในทางปฏิบัติการทำควอนไตซ์เซชันในระบบดิจิตอล จะเป็นค่าของ 2 ยกกำลังจำนวนเต็ม คือ

$$N = 2^n \quad (2.7)$$

$$G = 2^m \quad (2.8)$$

เมื่อ G คือ จำนวนระดับ Gray level ดังนั้นจำนวนบิท (bit) ที่ใช้ในการเก็บภาพหนึ่งภาพที่ถูกดิจิไตซ์ คือ

$$B = N \times N \times m \text{ บิท} \quad (2.9)$$

ดังตัวอย่างขนาด 128 x 128 pixels และระดับ Gray level ความจำขนาด 131,072 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงจำนวนไบท์ที่ใช้ในการเก็บภาพ เมื่อ N และ M เปลี่ยนไป

m / n	1	2	3	4	5	6	7	8
32	128	256	512	512	1024	1024	1024	1024
64	512	1024	2048	2048	4096	4096	4096	4096
128	2048	4096	8192	8192	16384	16384	16384	16384
256	8192	16384	32768	32768	65536	65536	65536	65536
512	32764	65536	131072	131072	262144	262144	262144	26144

2.6 เทคนิคต่าง ๆ ในการประมวลผลภาพ

เทคนิคต่าง ๆ สำหรับการประมวลผลภาพ แบ่งเป็น 4 พวกใหญ่ ๆ คือ

1. อิเมจดิจิไตเซชัน (image digitization)
2. อิเมจเอนฮานเมนต์และรีสตอเรชัน (image enhancement and restoration)
3. อิเมจเอ็นโด๊ดดิ้ง (image encoding)
4. อิเมจคอนสตรัคชัน (image reconstruction)

2.6.1 อิเมจดิจิไตเซชัน (image digitization)

ดังที่ได้กล่าวมาแล้วถึงความหมายของการ digitize ภาพ ซึ่งความละเอียดของภาพที่ได้ก็ขึ้นอยู่กับการจัดระดับภาพ ในปัจจุบันเครื่องมือที่ใช้ทำกระบวนการนี้เรียกว่า ดิจิไตเซอร์ (digitizer) ดิจิไตเซอร์สามารถเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอลได้ ดังนั้นข้อมูลที่ได้จึงเก็บเป็นเลขไบนารี โดยใช้ดิจิไตเซอร์เป็นตัวจัดการ

2.6.2 อิเมจเอนฮานเมนต์และรีสตอเรชัน (image enhancement and restoration)

อิเมจเอนฮานเมนต์เป็นการทำภาพให้อยู่ในรูปที่เหมาะสมขึ้น มีความคมชัดมากยิ่งขึ้น สำหรับการนำไปใช้งานเฉพาะอย่าง กล่าวคือ วิธีทำภาพ หรือปรับปรุงภาพ X-ray อาจจะไม่เป็นวิธีที่ดี เมื่อนำมาปรับปรุงภาพถ่ายดาวเคราะห์ ที่ส่งมาจากการสำรวจทางอวกาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีปรับปรุงคุณภาพของภาพ (enhancement) มีหลายวิธี ดังนี้

1. คอนทราสท์เอนฮานเมนท์ (Contrast enhancement) เป็นวิธีที่ทำให้ภาพคมชัดขึ้น โดยอาศัยวิธีฮีสโตแกรม อาจใช้แบบลีเนียร์สเทรท (linear stratch), พีซลีเนียร์สเทรท (piecewise linerar stretch) หรือ อีควอไลเซชัน (equalization)

2. เอจเอนฮานเมนต์ (Edge enhancement) เป็นการแยกความแตกต่างของจุดภาพที่ใกล้เคียงกัน เพื่อหาขอบเขตของภาพ

3. การประมวลผลภาพสีเทียม (Pseudo – color image processing) เป็นการใช้เทคนิคของการทำ density slicing และการใส่สีเทียมให้กับภาพขาว – ดำ ที่มีระดับ Gray level ต่าง ๆ กัน

4. การกรองภาพ (Filtering) เพื่อให้ภาพเรียบ (smoothing) หรือ คมชัด (sharpening) โดยใช้ตัวกรองความถี่ต่ำ (low pass filter) หรือ ตัวกรองความถี่สูง (high pass filter) ตามลำดับ

อิเมจรีสตอเรชัน (Image restoration)

เป็นขบวนการในการสร้างภาพกลับคืน โดยการหาค่าชดเชย และแก้ความคลาดเคลื่อนเนื่องมาจากข้อมูลในภาพผิดพลาดไป หรือเป็นขบวนการสร้างภาพกลับคืน จากภาพที่ถูกทำให้เสียไป เนื่องจากปรากฏการณ์ต่าง ๆ โดยใช้หลักการของพีชคณิตเชิงเส้น (linear algrbra)

2.6.3 อิเมจเอ็นโค็ดดิ้ง (image encoding)

เป็นการใช้เทคนิคต่าง ๆ เพื่อเข้ารหัสข้อมูล เนื่องจากข้อมูลภาพที่ได้จะถูกเก็บในลักษณะเป็นจำนวนไบท์ ดังตารางที่ 2.1 ซึ่งถ้าภาพมีขนาดใหญ่ ก็ต้องใช้พื้นที่ในการเก็บมาก ด้วยข้อจำกัดของไมโครคอมพิวเตอร์ ที่มีขนาดหน่วยความจำจำกัด

การเข้ารหัสข้อมูลจึงมีประโยชน์ ในด้านการลดพื้นที่ในการเก็บข้อมูลดังกล่าวมากผลของการเข้ารหัสข้อมูลนี้ เรียกว่าเป็นการลดข้อมูล (Data reduction หรือ Data compressing) ซึ่งเป็นเนื้อหาที่ทำโครงงานนี้ รายละเอียดของการลดข้อมูลภาพ ได้อธิบายในบทที่ 3 นอกจากนี้ การเข้ารหัสข้อมูลยังมีรหัสข้อมูลยังมีประโยชน์ในการช่วยลดปริมาณข้อมูลภาพ ที่ใช้ในระบบการสื่อสาร เช่น การส่งภาพถ่ายจากอวกาศมายังโลก การส่งข้อมูลภาพผ่านโมเด็ม (modem) เป็นต้น

2.6.4 อิเมจรีคอนสตรัคชัน (image reconstruction)

เป็นวิธีการสร้างภาพตัดขวางของวัตถุ โดยไม่ต้องผ่า หรือทำลายวัสดุ เพื่อประมวลผลโดยใช้คอมพิวเตอร์ เราเรียกการสร้างภาพตัดภาพตัวขวางด้วยคอมพิวเตอร์ว่า คอมพิวเตอร์โทโมกราฟฟี่ (Computer tomography)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การลดข้อมูลภาพ

สัญญาณภาพโดยธรรมชาติแล้วจะเป็นสัญญาณต่อเนื่อง (Analog) การนำข้อมูลภาพมาใช้ในการประมวลผลในด้านต่าง ๆ ที่มีเครื่องคอมพิวเตอร์เป็นตัวประมวลผลแล้วนั้น สัญญาณจะต้องแปลงให้เป็นสัญญาณที่เครื่องสามารถเข้าใจได้เสียก่อน นั่นคือต้องแปลงให้อยู่ในรูปของสัญญาณดิจิตอลซึ่งมีลักษณะที่ไม่ต่อเนื่อง (discrete siganl) โดยผ่านการสุ่มและจัดระดับ (sample and quantised) เมื่อรายละเอียดของภาพหนึ่ง ๆ ถูกแทนที่ด้วยข้อมูลแบบดิจิตอล ขนาดของข้อมูลภาพจึงมีจำนวนสูงมากเพื่อที่จะสามารถเก็บรายละเอียดของภาพได้เพียงพอ ทำให้ต้องใช้ขนาดของหน่วยความจำในการเก็บข้อมูลเหล่านี้มีขนาดใหญ่ตามไปด้วย ในกรณีที่มีการรับส่งข้อมูลภาพเหล่านี้ด้วยแล้วจะต้องเสียเวลารับส่งเป็นจำนวนมาก โดยเฉพาะเป็นการสื่อสารข้อมูลผ่านดาวเทียมด้วยแล้ว เวลาที่มากนั้นก็หมายถึงค่าใช้จ่ายที่สูงมากตามไปด้วย ดังนั้นจึงจำเป็นอย่างยิ่งที่จะต้องลดเวลาในการส่งข้อมูลให้น้อยที่สุดเท่าที่จะเป็นไปได้

การลดขนาดข้อมูลภาพ จึงถูกนำมาใช้ในการแก้ปัญหาเหล่านี้ เพื่อเป็นการประหยัดทางด้านเศรษฐกิจในส่วนของจำนวนหน่วยความจำที่ลดลง และในส่วนของเวลาที่ใช้ไปในการรับส่งใช้ว่าต้องการรายละเอียดมากน้อยเพียงใด เมื่อทำการลดข้อมูลแล้วจึงจะสามารถนำข้อมูลกลับมา (reconstruct) ได้อย่างเหมาะสม ซึ่งงานแต่ละอย่างมีความต้องการรายละเอียดของภาพที่ไม่เท่ากัน อย่างเช่นถ้านำไปใช้ในการตรวจสอบและจดจำรูปแบบของวัตถุต่าง ๆ โดยใช้คอมพิวเตอร์แล้ว ภาพที่ใช้จะเน้นเฉพาะส่วนของขอบวัตถุในภาพนั้น ๆ ก็เพียงพอ โดยไม่จำเป็นต้องมีรายละเอียดโครงสร้างภายในของภาพ ในขณะที่การลดข้อมูลภาพเพื่อการลดเวลาที่ใช้ไปในการรับส่งภาพผ่านช่องสัญญาณความเร็วต่ำอย่างเช่น ระบบโครงข่ายโทรทัศน์สาธารณะนั้น มีความจำเป็นอย่างยิ่งที่จะต้องเก็บรายละเอียดต่าง ๆ ของภาพให้ได้มากที่สุด หลักการพื้นฐานที่นิยมใช้ในการลอกข้อมูลภาพมีอยู่สามหลักการด้วยกัน คือ พรีดิกทีฟโค๊ดดิ้ง ซึ่งเป็นการเข้ารหัสใน โดเมนความถี่หลักการที่สอง คือ Transform Coding เป็นวิธีที่ประมวลผล (process) ในโดเมนของความถี่ส่วนหลักการสุดท้ายเป็นการรวบรวมเอาข้อดีต่าง ๆ จากสองหลักการแรกเข้าด้วยกัน เรียกว่า ไฮบริดโค๊ดดิ้ง

3.1 พรีดิกทีฟโค๊ดดิ้ง (predictive coding)

การลดขนาดข้อมูลภาพด้วยวิธี พรีดิกทีฟโค๊ดดิ้งนี้ เป็นการอาศัยคุณสมบัติของข้อมูลภาพที่มักจะมีค่าซ้ำ ๆ กัน และเมื่อข้อมูลอินพุทถูกกำหนดให้มีความเกี่ยวพันธ์กันนั้นก็คือจุดภาพที่มีตำแหน่งอยู่ใกล้ ๆ กัน มักจะมีค่าระดับแอมปลิจูดใกล้เคียวกันหรือเท่ากัน ดังนั้นจึงสามารถที่จะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของจุดภาพหนึ่งจุดหรือหลายๆ จุด ที่ผ่านมาใน Line นั้น Line ก่อนหน้านั้นหรือในเฟรมที่ผ่านมา เป็นตัวคาดคะเนหรือแทนค่าของจุดภาพปัจจุบัน ซึ่งโดยธรรมชาติทางสถิติของข้อมูลภาพเราสามารถที่จะคาดคะเนค่าของข้อมูลได้ไม่ผิดพลาดมากนัก จากค่าที่ได้จากการคาดคะเนนี้เอาไปลบกับค่าจริงของจุดภาพ จะได้เป็นค่าความแตกต่างระหว่างค่าจริงกับค่าที่เราคาดคะเนเอาไว้ ซึ่งค่านี้จะมีขนาดเล็ก และค่าผลต่างนี้จะถูกนำไปเข้ารหัสเพื่อจะเก็บไว้ใช้ในตอนถอดรหัส พร้อมกับค่าที่เราคาดคะเนเอาไว้ ดังนั้นในการที่จะเก็บในหน่วยความจำหรือต้องการส่งก็ใช้ค่าสองค่านี้ เมื่อถึงตอนที่จะนำภาพเดิมกลับมาหรือตอนถอดรหัส จะนำเอาค่าที่เราคาดคะเนไว้ในตอนแรกบวกกับค่าของผลต่างของจุดภาพนั้นกับค่าคาดคะเน ก็จะได้ค่าของจุดภาพนั้น ๆ ค่าผิดพลาดที่ได้ในตอนถอดรหัสเกิดขึ้นเพียงกรณีเดียวเท่านั้น ตอนจัดระดับ (quantised) ค่าความแตกต่างของการเข้ารหัสเท่านั้นวิธีนี้เป็นวิธีที่ง่ายแก่การสร้างระบบ และสามารถลดขนาดของภาพให้เหลือประมาณ 1 – 2 bit/element

3.2 ทรานส์ฟอร์มโค๊ดดิ้ง (Transform Coding)

การลดข้อมูลภาพด้วยวิธีของ ทรานส์ฟอร์มโค๊ดดิ้งนี้เป็นวิธีที่ซับซ้อน และมีขั้นตอนมากกว่าวิธีลดข้อมูลภาพโดยพรีดิกทีฟโค๊ดดิ้ง หลักการของวิธีทรานฟอร์มโค๊ดดิ้งจะทำการแปลงข้อมูลอินพุทที่อยู่ในรูปของโดเมนข้อมูล (data domain) ให้อยู่ในรูปของแถบความถี่ (spectral) หรือโดเมนความถี่ (frequency domain) โดยวิธีการแปลงแบบต่าง ๆ เช่น ฟูเรียทรานส์ฟอร์ม (Fourier Transform) จะเป็นการแปลงข้อมูลที่อยู่ในรูปของสเปตเชียลโดเมน (spatial domain) ให้อยู่ในรูปสัมประสิทธิ์ของพลังงานความถี่ โดยความถี่ต่ำ ๆ จะมีพลังงานสูง ที่ความถี่สูงพลังงานจะลดลงไป การเข้ารหัสจึงใช้จำนวนบิทสำหรับแต่ละช่วงความถี่ไม่เท่ากัน เมื่อต้องการอัตราบิทเรทสูง ๆ ค่าของพลังงานความถี่สูงจะถูกตัดทิ้งเป็นส่วนใหญ่ เป็นเหตุให้รายละเอียดส่วนที่เป็นรายละเอียดส่วนที่เป็นขอบภาพหายไป ทำให้ภาพที่ได้เบลอ ขาดความคมชัด

การแปลงที่นิยมใช้ในการลดข้อมูลภาพมีอยู่ด้วยกันหลายวิธี เช่น Fast Fourier Transform (FFT) , Fast – Hadamard Transform , Fast Slant Transform , Fast Discrete Cosing Transform , fast Discrete Sine Transform เป็นต้น ซึ่งในแต่ละวิธีก็มีข้อดีข้อเสียที่แตกต่างกันไปการแปลงที่นิยมใช้กันมากได้แก่ Discrete Cosine Transform เพราะเป็นวิธีที่สามารถคำนวณได้ง่ายเนื่องจากมีการคำนวณเฉพาะค่าจริงไม่ใช่ค่าจินตภาพ (imaginary)

วิธีทรานส์ฟอร์มโค๊ดดิ้ง ถึงแม้ว่าจะเป็นวิธีที่ยุ่งยากแต่ก็สามารถสร้างระบบได้ด้วยอุปกรณ์ทางฮาร์ดแวร์ (Hardware) ดิจิตอลความเร็วสูงได้ง่าย และเป็นระบบที่ยืดหยุ่น (adaptive system) สามารถที่จะลดขนาดของข้อมูลให้อยู่ในอัตราบิทเรทประมาณ 0.5 – 1 บิทต่อจุดภาพ ซึ่งเป็นช่วงอัตราการลดที่สามารถสร้างภาพเดิมกลับมาได้สมบูรณ์เพียงพอต่อการนำไปใช้งานต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ไฮบริดโค๊ดดิ้ง (Hybrid Coding)

ไฮบลิดโค๊ดดิ้งเป็นเทคนิคที่นำข้อดีของพรีดิกทีฟโค๊ดดิ้ง และ ทรานฟอร์มโค๊ดดิ้ง มาใช้ร่วมกัน ทั้งนี้เพราะบางครั้งการลดขนาดข้อมูลภาพด้วยทรานส์ฟอร์มโค๊ดดิ้ง ไม่อาจจะให้รายละเอียดของภาพเพียงพอ ดังนั้น การเพิ่มรายละเอียดของภาพจึงต้องใช้ความสัมพันธ์ของจุดภาพที่มีอยู่ทั้งทางแนวนอนและทางแนวตั้ง โดยการใช้แปลงแบบมิติเดียวในทิศทางแกนใดแกนหนึ่งของภาพ เช่น การแปลงกับข้อมูลภาพในทางแนวนอนต่อจากนั้นจึงใช้พรีดิกทีฟโค๊ดดิ้ง กับสัมประสิทธิ์ของการแปลงที่ได้ เพื่อที่จะใช้ในการคาดคะเนค่าของกลุ่มสัมประสิทธิ์ที่เหมือน ๆ กันซึ่งเหมือนกับการทำพรีดิกทีฟโค๊ดดิ้ง ใน นั่นเอง และในกรณีของการแปลงแบบ 2 มิติ สามารถทำได้เร็วขึ้นอาจจะใช้การทำพรีดิกทีฟโค๊ดดิ้ง กับค่าสัมประสิทธิ์ของการแปลงของภาพในบริเวณเดียวกันแต่เป็นเฟรมที่แตกต่างกันที่ตามมา การใช้ไฮบริดโค๊ดดิ้ง ในลักษณะนี้สามารถเป็นไปได้อีกลักษณะหนึ่งคือ ใช้การใช้พรีดิกทีฟโค๊ดดิ้ง ในขั้นตอนแรกก่อนแล้วจึงทำการแปลง แต่ในกรณีนี้ต้องการระบบที่ซับซ้อนมากกว่า

การลดข้อมูลภาพด้วยวิธีของไฮบริดโค๊ดดิ้งนี้ อาจจะพูดได้ว่ามีคุณสมบัติอยู่ระหว่างวิธีของพรีดิกทีฟโค๊ดดิ้ง และ ทรานส์ฟอร์มโค๊ดดิ้ง ดังนั้นอัตราบิดเรทต่ำสุดของวิธีนี้จึงไม้สามารถทำให้ต่ำกว่าวิธีของทรานส์ฟอร์มโค๊ดดิ้ง เพียงแต่สามารถสร้างได้ง่ายกว่า โดยอัตราบิทเรทของระบบจะมีค่าประมาณ 1 บิตต่อจุดภาพ ซึ่งอัตราบิทเรทที่สามารถสร้างภาพเดิมกลับมาใหม่ (reconstruction) ได้ภาพที่มีคุณภาพดีพอสมควร

3.4 การลดข้อมูลภาพด้วยวิธีรันเลงจ์ (Runlength Compression)

การลดข้อมูลด้วยวิธีนี้ อาศัยลักษณะทั่วไปของที่จะต้องมีส่วนของฉากหลัง (background) และพื้นหน้า (foreground) ในส่วนของ ฉากหลังจะมีรายละเอียดของภาพไม่มากนัก ส่วนนี้เองจะมีการเปลี่ยนแปลงของข้อมูลน้อย เมื่อเทียบกับส่วนของ foreground ซึ่งมีรายละเอียดและการเปลี่ยนแปลงของข้อมูลมาก ในส่วนที่มีการเปลี่ยนแปลงข้อมูลน้อยนี้เองที่เราสามารถนำการเข้ารหัสแบบรันเลนจ์มาประยุกต์ใช้ได้อย่างมีประสิทธิภาพ

การเข้ารหัสแบบนี้ จะจัดข้อมูลภาพเดิม ให้อยู่ในรูปของคู่ลำดับ (G, L) โดย G แทนระดับความเข้ม หรือระดับสีเทา L แทนความยาวของข้อมูลหรือ จำนวนจุดที่มีระดับสีเทา G การเข้ารหัสแบบนี้มีด้วยกัน 2 วิธีใหญ่ คือ

วิธีที่ 1 จะทำการอ่านข้อมูลเข้ามาโดยนับจำนวนข้อมูลที่ซ้ำกันกับข้อมูลนั้นเข้ามาด้วย แล้วแปลงข้อมูลเข้าไปเป็น 2 ไบท์ คือ ไบท์แรกจะเก็บจำนวนตัวข้อมูลที่ซ้ำกัน โดยจะมีไบท์ที่สองเก็บค่าของระดับสีที่ซ้ำกันนั้นเองไว้ โดยใน 1 ชุด ข้อมูลเอาท์พุท (2 ไบท์) จะนับจำนวน จุดที่ซ้ำกันได้ 256 จุดสี ตัวอย่างการเข้ารหัสข้อมูล เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเป็น (ค่าที่แสดงเป็นเลขฐาน 16)

AC AC AC AC 15 15 15 15 15 15 45 45 78 20 10 10 10 10

เข้ารหัสได้เป็น

04 AC 06 15 02 45 01 78 01 20 04 10

จะเห็นได้ว่าเราสามารถลดข้อมูลจาก 19 ไบท์ เหลือ 12 ไบท์ ซึ่งหากข้อมูลซ้ำกันถึง 256 จุด แล้วจะทำให้เราลดข้อมูลไปได้อย่างมหาศาล แต่ในขณะเดียวกันหากเกิดกรณีที่แย่ที่สุดของการใช้วิธีนี้ก็คือ กรณีที่ข้อมุลแต่ละตัวไม่ซ้ำกับจุดข้างเคียงเลย ซึ่งจะทำผลของการเข้ารหัสจะได้รหัสที่ยาวเป็น 2 เท่า ของข้อมุลอินพุท

วิธีที่ 2 จากความบกพร่องของวิธีการทำรันเลนจ์เอ็นโค๊ดดิ้ง วิธีแรก ตรงที่โปรแกรมจะทำการเข้ารหัสข้อมูล เมื่อนับจำนวนข้อมูลที่ได้ตั้งแต่ 1 – 255 จุด ดังนั้นถ้าเกิดข้อมูลของเรามี จุดสีที่อยู่โดด ๆ (ไม่ซ้ำกันจุดข้างเคียง) อยู่มากมายจะทำให้การเข้ารหัสไม่ทำให้ข้อมูลเล็กลงมากนัก จึงได้มีแนวคิดที่จะทำการแก้ไขปัญหาของการทำรันเลนจ์เอ็นโค๊ดดิ้ง วิธีแรกเกี่ยวกับการมีข้อมูลที่ไม่ซ้ำกันกับตัวข้างเคียง โดยมีหลักการอยู่ 2 ข้อดังนี้

1. จะไม่ทำการลดข้อมูลกับส่วนที่มีจำนวนซ้ำกันน้อยกว่า 3 ตัว

2. ให้มีการทำเครื่องหมายเพื่อที่จะแยกส่วนที่มีการลดข้อมูลกับส่วนที่ไม่มีการลดข้อมูลออกจากกัน โดยใช้บิท บิทหนึ่งเป็นตัวบอกว่าถูกลดขนาดหรือไม่

จากหลักการดังกล่าวนี้ จะใช้บิทที่ 7 ของไบท์บอกขนาดเป็นตัวบอกว่าข้อมูลที่ต่อจากนี้ไปมีการลดขนาดข้อมูลหรือไม่ ส่วนบิทที่เหลือเราก็ยังคงใช้บอกขนาดต่อไป เมื่อเรานับพิกเซลได้ซ้ำกันตั้งแต่ 3 ตัวขึ้นไป บิทที่ 7 ของไบท์บอกขนาดจะถูกเซทให้เป็น 1 หรือ แบะเราจะเริ่มนับ 1 ตั้งแต่พิกเซล ที่ซ้ำกันตั้งแต่ตัวที่ 4 เป็นต้นไป ทำให้บิทบอกขนาดมีค่าได้ตั้งแต่ 3 – 130 หรือใน 2 ไบท์นี้ เราอาจเก็บข้อมูลได้ถึง 130 ไบท์ ส่วนกรณีที่ไม่มีการลดขนาดบิทที่ 7 ของไบท์ บอกขนาดก็จะถูก set ให้เป็น 0 และ 7 บิท ที่เหลือจะบอกจำนวนของข้อมูลที่มีค่าไม่ซ้ำกันนั้นโดยค่าที่ตามมาจะเป็นข้อมูลที่ไม่ถูกลดขนาด ซึ่งจะมีค่าระหว่าง 0 ถึง 127 โดยจำนวนจริงจะมีค่าเท่ากับไบท์บอกขนาดบวกหนึ่ง ตัวอย่างเช่น

ข้อมูลที่เป็น (ค่าที่แสดงเป็นเลข 16)

AC AC AC AC 15 15 15 15 15 15 45 45 78 20 10 10 10 10

เข้ารหัสได้เป็น

81 AC 83 15 03 45 45 78 20 81 10

ความหมาย : 81 (ถูกลดขนาดลง 4 ไบท์มีค่า AC)

: 83 (ถูกลดขนาดลง 8 ไบท์มีค่า 15)

: 03 (ถูกลดขนาดลง 4 ไบท์มีค่า 45 45 78 20)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

: 81 (ถูกลดขนาดลง 4 ไบท์มีค่า 10)

โดยวิธีนี้ผลลัพธ์ที่ได้จะแย่ที่สุดในกรณีที่เกิดมีข้อมูลที่ไม่ซ้ำกันเลยมากกว่า 128 ไบท์ เมื่อทำการเข้ารหัสจะได้รหัสถึง 129 ไบท์ สำหรับทุก ๆ 128 ไบท์ของข้อมูลอินพุท และในกรณีที่ดีที่สุดก็คือเมื่อข้อมูลซ้ำกันถึง 128 ไบท์ เมื่อเข้ารหัสแล้วจะได้ข้อมุลที่ลดลงเหลือเพียง 2 ไบท์

3.5 ความสามารถในการลดข้อมูลภาพด้วยรันเลนจ์

ในกรณีที่ข้อมูลภาพมีความแตกต่างของระดับเทามากอัตราของข้อมูลต่อ 1 จุดภาพ (data bit rat) จะมากตามไปด้วย เนื่องจากมีความเป็นไปได้ที่ข้อมูลจะมีการเปลี่ยนแปลงสูง

การลดข้อมูลด้วยวิธีรันเลนจ์นี้จะไม่มีความคลาดเคลื่อนเกิดขึ้น หลังจากการถอดรหัส การใช้วิธีนี้ต้องพิจารณาถึงลักษณะของภาพต้นแบบว่ามีความเหมาะสมหรือไม่ ควรมีรายละเอียดของภาพน้อย เช่น ภาพเอกสาร ภาพทางด้านกราฟจากตาราง เราสามารถลดข้อมูลภาพได้ ดียิ่งขึ้นได้ โดยการปรับข้อมูลให้มีความแตกต่างของระดับสีเทาให้น้อยลง ด้วยวิธีของ gray scale transmission

3.6 ข่าวสาร ปริมาณข่าวสารเฉลี่ย และส่วนที่ซ้ำ

ข่าวสาร (information) ในความหมายของ ทฤษฎีข่าวสาร คือปริมาณความไม่แน่นอนของเหตุการณ์ (measure of uncertainly) เช่น สำหรับเหตุการณ์ (event) ที่เกิดค่อนข้างไม่แน่นอนเหตุการณ์นั้นมีข่าวสารน้อย สำหรับเหตุการณ์ที่เกิดไม่มากครั้ง แสดงว่าเหตุการณ์นั้นมีข่าวสารมาก

การวัดปริมาณข่าวสารของเหตุการณ์ คำนวณได้จาก

$$I_i = - \log_2 P_i \qquad \text{บิท}$$

โดยที่ P_i คือความน่าจะเกิด (probability) ของเหตุการณ์ I

ปริมาณข่าวสาร ของเหตุการ I บอกเราว่าควรจะใช้เนื้อที่ในการเก็บเหตุการณ์ I เท่าไร ตัวอย่าง เช่น เหตุการณ์ x มีความน่าจะเกิดเท่ากับ 1/8 แล้วแสดงว่า เนื้อที่ที่ใช้เก็บเหตุการณ์ x คือ 3 บิท

ปริมาณข่าวสารเฉลี่ยของเหตุการณ์ทั้งหมดหรือ entropy คำนวณได้จาก

$$H = - (P_i \log_2 P_1 + P_2 \log_2 P_2 + \ldots + P_m \log_2 P_m)$$

โดยที่ M คือ เหตุการณ์ที่เป็นไปได้ทั้งหมด

ปริมาณข่าวสารเฉลี่ย บอกเราว่าควรจะเก็บข้อมูลทั้งหมดด้วยอย่างน้อยที่สุดเท่าไร เช่น ข้อมูล 'e' มีความน่าจะเกิดเท่ากับ 1/16 ดังนั้นข้อมูล 'eeeee' ใช้เนื้อที่ในการเก็บเท่ากับ 20 บิท

ข่าวสารเฉลี่ย มีเงื่อนไขที่น่าสนใจ 2 กรณี คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เมื่อ $M = 1$ คือมีเหตุกาณ์เดียว ดังนั้น $P_i = 1$ ทำให้ $H = 0$ หมายความว่า ไม่มีข่าวสารเฉลี่ยของเหตุการณ์ที่ทราบแน่นอนแล้ว

2. เมื่อ $P_i = 0$ ทำให้ $H = 0$ หมายความว่า ไม่มีข่าวสารเฉลี่ยของเหตุการณ์ที่เป็นไปไม่ได้

นั่นคือ $H = -P\log_2 P = 0$ (เมื่อ P เข้าใกล้ 0)

ส่วนที่ซ้ำเกิดขึ้นเนื่องจาก การกระจายของข้อมูลไม่เท่ากัน คือความน่าจะเกิดของข้อมูลทุกตัวไม่เท่ากัน ถ้าการกระจายของข้อมูลเท่ากัน แล้ว $H = \log_2 M$ ซึ่ง H จะมีค่าสูงสุดดังนั้น $R = 0$ แต่ข้อมูลโดยทั่วไปมักกระจายไม่เท่ากัน

ดังนั้นเทคนิคการลดขนาดข้อมูลจะใช้ประโยชน์ จากการกระจายไม่เท่ากันของข้อมูลนี้เพื่อลดขนาดของข้อมูล โดยการเข้ารหัสข้อมูลด้วยรหัสที่เท่ากับ หรือ ต่างจาก entropy น้อยที่สุด ตัวอย่างของรหัสที่เข้าข้อมูล คือ Huffman หรือ Shannon Fano

3.7 ดิสครีตโคไซน์ทรานส์ฟอร์มโค๊ดดิ้ง (discrete Cosine Transform coding)

ดิสครีตโคไซน์ทรานส์ฟอร์มโค๊ดดิ้งเป็นวิธีการที่ใช้ในการลดข้อมูลภาพที่ได้รับความนิยมอย่างแพร่หลาย ทั้งนี้เพราะค่าสัมประสิทธิ์ในโดเมนของความถี่ที่ได้จะเป็นเทอมของค่าจริง (real time) เท่านั้น อีกทั้งยังสามารถคำนวณได้ไม่ยาก ในปัจจุบันหลักการของดิสครีตโคไซน์ทรานส์ฟอร์มโค๊ดดิ้ง ยังคงมีการวิจัยกันอยู่ต่อไปให้สามารถลดขนาดของข้อมูลให้ได้มากที่สุดพร้อมทั้งหาวิธีที่จะเพิ่มความเร็วในการคำนวณให้ได้เร็วขึ้นไปอีก

ดังนั้นในที่นี้จึงยกตัวอย่าง ระบบการลดข้อมูลภาพโดยใช้เทคนิคของดิสครีตโคไซน์ทรานส์ฟอร์มโค๊ดดิ้ง โดยมีโครงสร้างของระบบเป็นดังรูปที่ 3.1 จาก block diagram ของการแปลงภาพอินพุทที่เข้ามาจะถูกแยกออกเป็นบล็อกเล็ก ๆ โดยเราสามารถกำหนดขนาดของบล็อกได้ว่าให้เป็นเท่าไร ขนาดของบล็อกที่เหมาะสมจะเป็นตัวเพิ่มประสิทธิภาพในการรวมพลังงานของการแปลงซึ่งจะทำให้ภาพที่ได้มีรายละเอียดที่ดี ความเหมาะสมของขนาดของบล็อกค่าหนึ่ง ๆ จะเหมาะสมที่ค่าบิทเรทค่าหนึ่ง ๆ แต่อย่างไรก็ตามขนาดของบล็อกที่เหมาะสมสามารถคำนวณได้ด้วยคณิตศาสตร์ที่ยิ่งยากพอ ๆ กับการแปลง เลยทีเดียว โดยส่วนมากแล้วขนาดของบล็อกจะมีค่าเป็นเลขยกกำลังของสองอย่างเช่น 4, 8, 16 ซึ่งได้มาจาก $2^2, 2^3, 2^4$ เป็นต้น หลังจากการแยกข้อมูลออกเป็นบล็อกย่อย ๆ แล้ว ก็จะทำการแปลง ไปโดยอิสระของการแปลงแบบมิติเดียว และ สองมิติ ดังต่อไปนี้เป็น

$$F(k) = \sqrt{\frac{2}{N}}\alpha(k)\sum_{N-1}^{N-1} f(n)\cos\left[\frac{(2n+1)k\pi}{2N}\right]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $\alpha(0) = \sqrt{\frac{1}{2}}$ และ $\alpha(k) = 1$ เมื่อ k ไม่เท่ากับ 0

F (k) เป็นผลที่ได้จากการ Transform และ f(n) เป็นข้อมูลอินพุทตัวที่ n ส่วนมากสมการของการ Transform แบบสองมิติสามารถเขียนได้ดังนี้คือ

$$F(u, v) = \frac{2}{N}\alpha(u)\alpha(v)\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\cos\left[\frac{(2m+1)u\pi}{2N}\right]\cos\left[\frac{(2n+1)v\pi}{2N}\right]$$

เมื่อ

u, v เป็นตัวแปรของ discrete frequency มีค่าเป็น 0, 1, 2, ... N – 1

f (m, n) เป็นตำแหน่งของจุดภาพภายในบล็อกขนาด N x N (0, 1, 2, ..., N – 1)

f (u, v) เป็นผลจากการทำ discrete cosine Transform

$\alpha(0) = \sqrt{\frac{1}{2}}$ และ $\alpha(j) = 1$ เมื่อ j ไม่เท่ากับ 0

3.8 อัฟแมนโค๊ดดิ้ง (Huffman Coding)

ผู้ที่คิดวิธีการลดขนาดข้อมูลนี้คือ นาย D.A. Huffman ศิษย์เก่าจาก MIT เขาได้เสนอกระบวนการเข้ารหัส แบบนี้มาตั้งแต่ต้นทศวรรษ 1950 สำหรับหลักการสำคัญของการลดข้อมูลนี้คือการลดจำนวนบิทที่ใช้แทนตัวอักษรที่มีการใช้บ่อย ๆ และเพิ่มจำนวนบิทที่ใช้แทนตัวอักษรที่มีการใช้น้อย ๆ

การทำงานชนิดนี้อยู่บนพื้นฐานความจริงที่ว่า ในประโยคหรือไฟล์ข้อมูลที่เป็นภาษาอังกฤษหนึ่ง ๆ จะมีการใช้อักษรภาษาอังกฤษซ้ำไปซ้ำมาเสมอ และบางตัวก็ไม่ได้ใช้บ่อยสักเท่าไรถ้าภาษาอังกฤษตัวหนึ่งถูกแทนด้วยเลขฐานสองจำนวน 8 บิท แล้วเราพบว่าในไฟล์ข้อมูลมีตัวอักษร "a" ด้วยรหัส 4 บิท และให้อักษร "Z" แทนด้วยรหัส 14 บิท เป็นต้น

ปัจจุบันนี้ได้มีโปรแกรมแอพพลิเคชันหลาย ๆ ชนิดที่ได้ประยุกต์ใช้วิธีการลดขนาดของฮัฟแมนนี้ เช่น การลดขนาดแบบ MNP-5 ของโมเด็ม ซึ่งได้ใช้การลดข้อมูลฮัฟแมนแบบไดนามิก การเข้ารหัสแบบ Shannon – Fano การเข้ารหัสบางส่วนของโปรแกรม PKZIP การทำงานตอนท้ายการลดขนาดแบบ JPEG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การสร้างรหัสของฮัฟแมนก่อนอื่นอ่านข้อมูลหนึ่งรอบ เพื่อหาความถี่ของแต่ละข้อมูล จากนั้นเรียงลำดับของข้อมูล ที่มีความถี่มากอยู่ด้านหนึ่ง และข้อมูลที่มีความถี่น้อยอยู่อีกด้านหนึ่ง และกำหนดให้ข้อมูลแต่ละตัวคือโนด (Node) ใด ๆ ของทรี (tree) แล้วนำมาสร้าง ทรี ตามวิธีต่อไปนี้

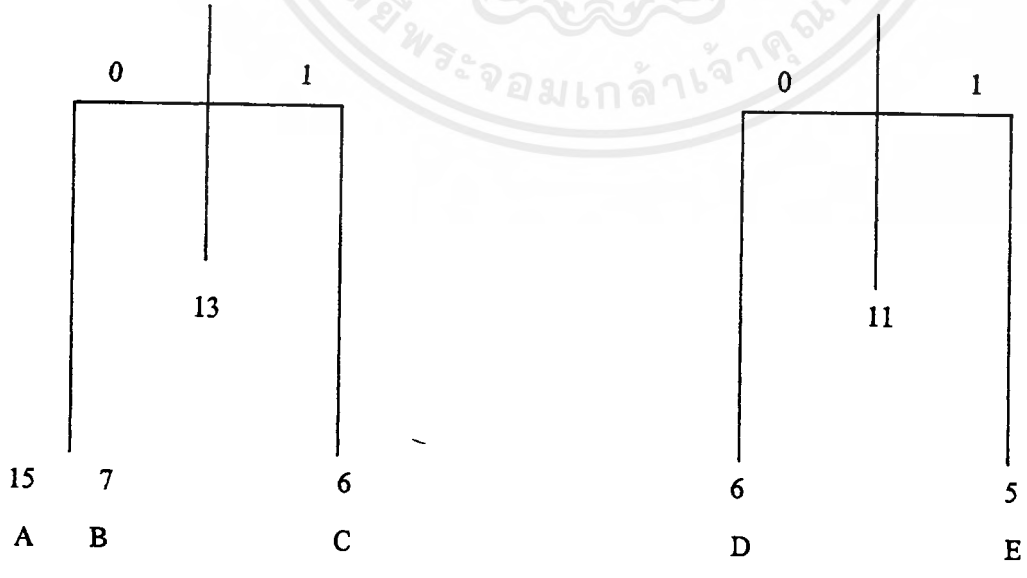
1. นำเอาข้อมูลหรือโนดที่มีความถี่น้อยที่สุดมาเป็น โนดลูก (Child Node) ของโนดพ่อ (Parent Node) ซึ่งความถี่ของโนดพ่อนี้จะเท่ากับผลรวมของความถี่ของโนดลูกทั้งสอง
2. โนดพ่อจะถูกนำไปรวมกับโนดอื่น ๆ ที่เหลือและโนดลูกที่ใช้แล้วทั้งสองจะไม่ถูกนำมาใช้อีก
3. โนดลูก หนึ่งจะถูกกำหนดให้เป็นบิท 0 และอีกโนดหนึ่งให้เป็นบิท 1
4. กระทำซ้ำตั้งแต่ 1 จนกระทั่งเหลือโนดเดียว เรียกว่า รากของทรี

ตัวอย่างเช่นข้อมูลมีลักษณะดังนี้

"AEDBAACBADCABECADACEDADACAEBABABEADABAD"

ความถี่	15	7	6	6	5
	A	B	C	D	E

โนดที่มีน้ำหนักน้อยที่สุดคือ D และ E ทั้งสองโนดจะถูกเชื่อมให้เป็นโนดพ่อซึ่งมีน้ำหนัก 11 แล้วเรากำหนดให้ โนด D มีค่าเป็น 0 และโนด E มีค่าเป็น 1 รอบที่สอง B และ C จะถูกนำมาเชื่อมเป็นโนดพ่อใหม่อีก ซึ่งมีน้ำหนักเป็น 13 ซึ่งจะเป็นดังรูปที่

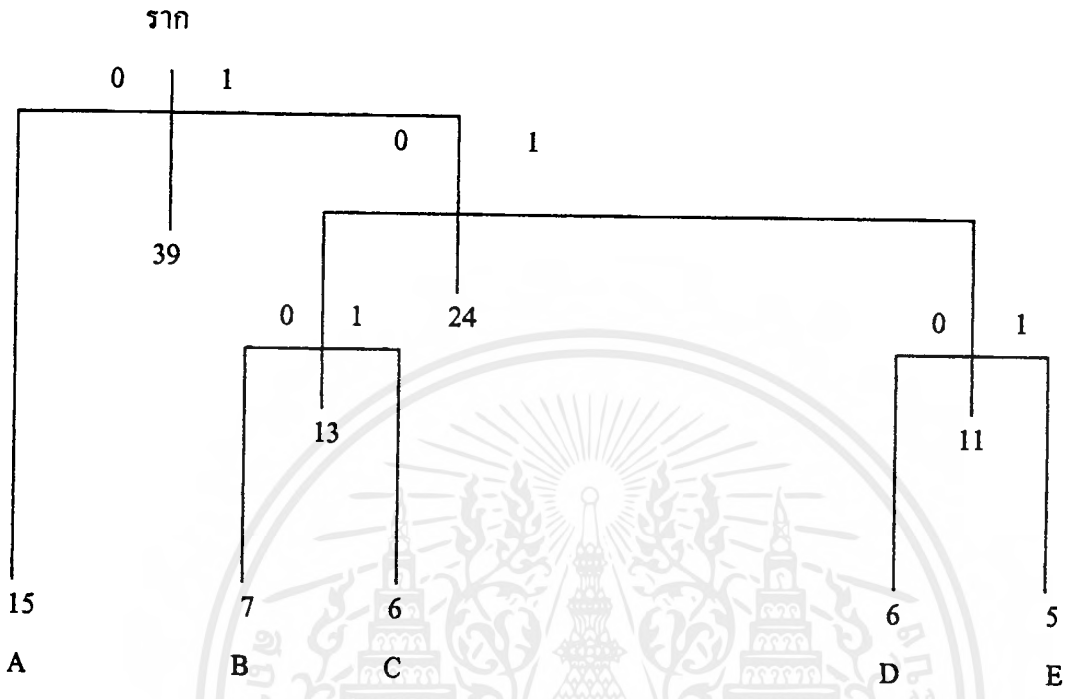


รูปที่ 3.1 ฮัฟแมนทรี เมื่อผ่านไป 2 รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

34013

รอบต่อมา 2 โนด ที่มีน้ำหนักน้อยที่สุด คือโนดพ่อ ของ B/C กับ D/E จะถูกเชื่อมเป็นโนดพ่อ หลังจากนั้นจะเหลือเพียง 2 โนด เมื่อนำมารวมกันจะได้รากของทรีซึ่งมีลักษระดังรูป 3.2



รูปที่ 3.2 แผนภาพแสดงแมนทรี

การหารหัสของข้อมูลทำได้โดยการ เดินทางไปยังเส้นทางที่ข้อมูลตัวนั้นผ่าน ผ่านโนดใดให้จำเอาไว้ รหัสของข้อมูลคือ ค่าของโนดที่จำไว้ตามลำดับ ดังนั้นรหัสของข้อมูลคือ

A = 0
B = 100
C = 101
D = 110
E = 111

หลังจากได้รหัสของข้อมูลแล้วจึงอ่านข้อมูลอีกรอบเพื่อลดขนาดข้อมูลทั้งหมดโดยการแทนข้อมูลด้วยรหัสของข้อมูล ดังนั้นจะได้รหัสของข้อมูลเรียงกันเป็นลักษณะเป็นลำดับเช่นเดียวกับลำดับของข้อมูล ดังรูปที่ 3.3

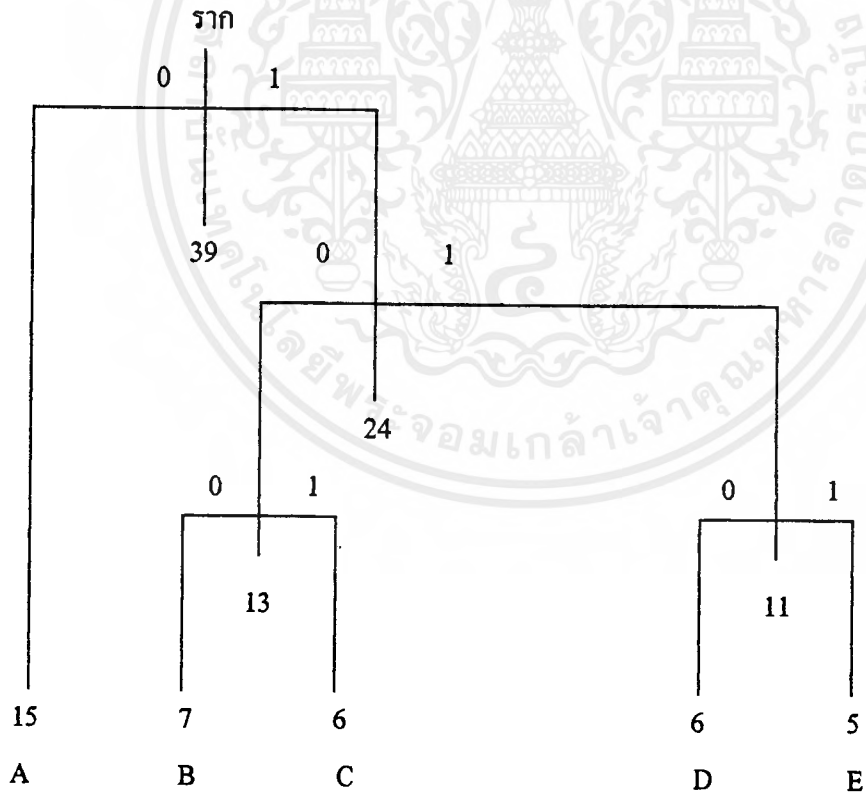
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"0 111 110 100 0 0 101...." รหัสของข้อมูลมีลักษณะเป็นบิท

"A E D B A A C....." ข้อมูลมีลักษณะเป็นไบท์

รูปที่ 3.3 ตัวอย่างการเข้ารหัสข้อมูล

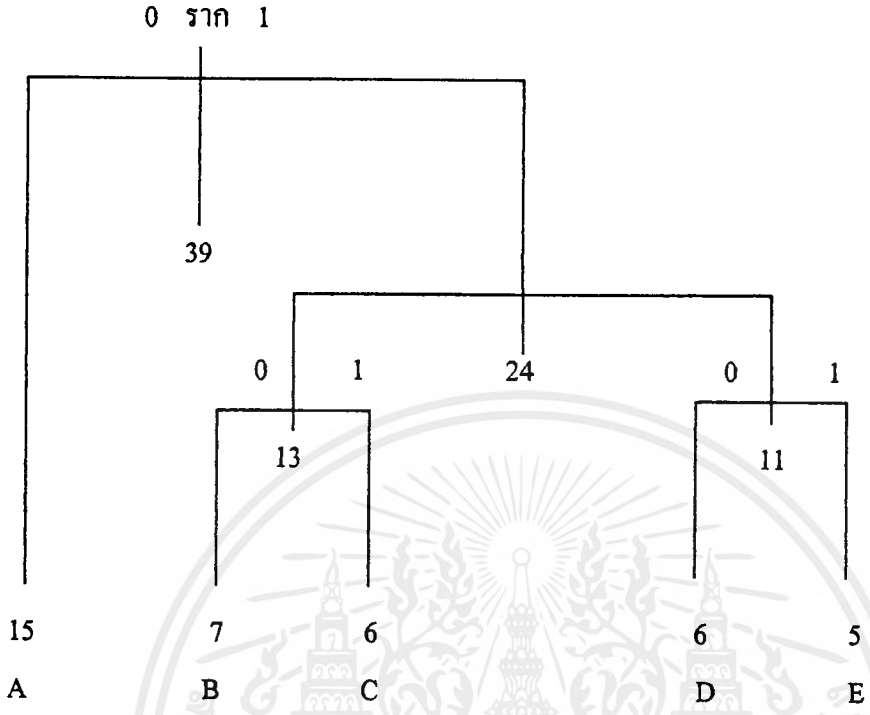
สำหรับการขยายข้อมูลกลับทำได้โดย การสร้างทรีขึ้นมาก่อนโดยทรีนี้คือทรีอันเดียวกับที่ใช้ในการลดขนาดข้อมูล โดยจะอ่านรหัสเข้ามาทีละบิทและเริ่มเดินทางจากรากไปยังกิ่งที่มีค่าเท่ากับรหัส 1 บิท ที่ป้อนเข้ามา และกระทำเช่นนี้ต่อไปจนกระทั่งไม่มี โดยย่อยลงไปอีกก็จะได้ข้อมูลคืนกลับมาหนึ่งตัว หลังจากนั้น เริ่มเดินทางจากรากใหม่ จนกระทั่งได้ข้อมูลกลับคืนมาจนครบหมด จากรหัสของข้อมูลข้างต้น บิทตัวแรกก็คือ 0 ดังนั้น สามารถถอดรหัสออกมาได้เป็น A ดังรูป 3.4



รูปที่ 3.4 แสดงการถอดรหัสเมื่อได้รับรหัส 0

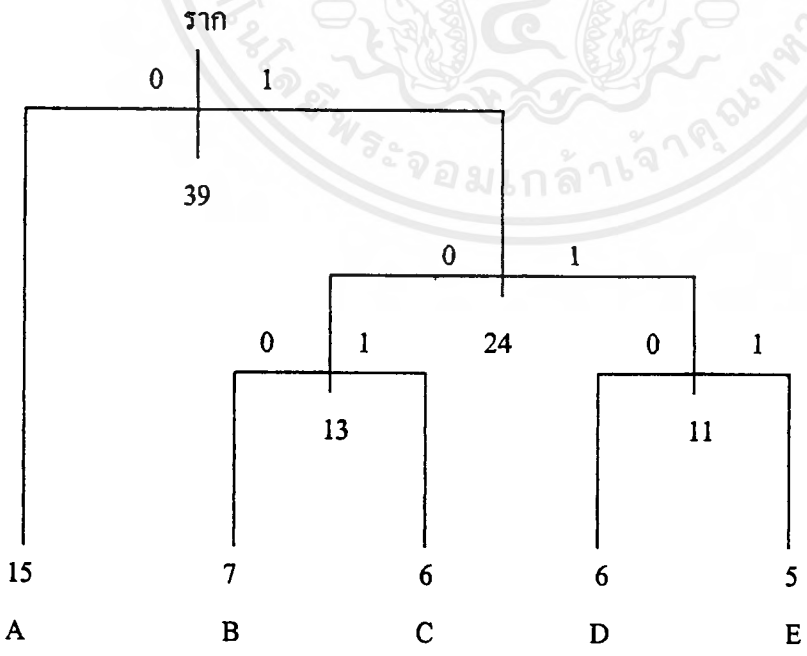
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นอ่านรหัสข้อมูลบิทที่ 2 คือ 1 เมื่อเดินทางจากรากไปยังทรีจะมีลักษณะดังรูป 3.5



รูปที่ 3.5 แสดงการถอดรหัสเมื่อได้รับรหัส 1

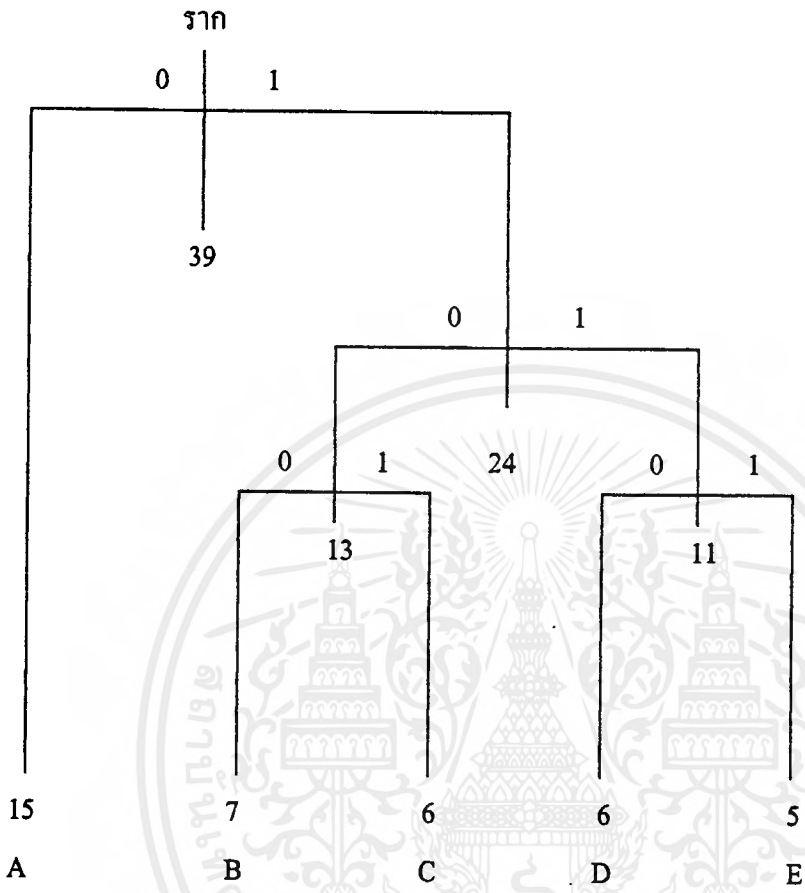
จากนั้นอ่านรหัสข้อมูลบิทที่ 3 คือ 1 เมื่อเดินทางจากรากไปยังทรีจะมีลักษณะดังรูป 3.6



รูปที่ 3.6 แสดงการถอดรหัสเมื่อได้รับรหัส 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นอ่านข้อมูลบิที่ 1 คือ 1 เมื่อเดินทางจากรากไปยังทรีจะมีลักษณะดังรูป 3.7



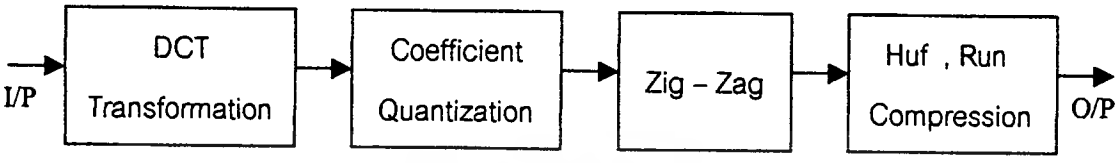
รูปที่ 3.7 แสดงการถอดรหัสเมื่อได้รับรหัส 111

ก็จะสามารถถอดรหัสได้เป็น A และจะกระทำเช่นนี้ต่อไปจนกระทั่งไม่มีข้อมูลเหลือ แล้วจะได้ข้อมูลกลับมาเหมือนเดิมทุกประการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
หลักการเบื้องต้นของการลดข้อมูลภาพ

Data Compression ลำดับขั้นตอนการทำงานของ Data Compression มี 4 ขั้นตอน ดังแสดงในรูป



ขั้นตอนทั้งสามของ Data Compression เป็นการ compress ที่มีประสิทธิภาพ และมีประโยชน์ ซึ่งมีการสูญเสียน้อยกว่า 10 เปอร์เซ็นต์ของข้อมูลจริง

4.1 Discrete Cosine Transform

Discrete Cosine Transform Coding เป็นวิธีการที่ใช้ในการลดข้อมูลภาพที่ได้รับความนิยมอย่างแพร่หลาย ทั้งนี้เพราะค่าสัมประสิทธิ์ในโดเมนความถี่ ที่ได้จะเป็นเทอมของค่าจริง (realtime) เท่านั้น อีกทั้งยังสามารถใช้งานจริงในลักษณะ real time โดยใช้ ฮาร์ดแวร์ได้ไม่ยาก ในปัจจุบันหลักการของ Discrete Cosine Transfer Coding ยังคงมีการวิจัยกันอยู่ต่อไป เพื่อให้สามารถลดข้อมูลได้มากที่สุดพร้อมทั้งหาวิธีที่จะเพิ่มวิธีในการคำนวณให้ได้เร็วขึ้นไปอีก

ดังนั้นในที่นี้จึงยกตัวอย่าง ระบบการลดข้อมูลภาพโดยใช้เทคนิคของ Discrete Cosine Transform Coding โดยมีโครงสร้างของระบบเป็นดังรูปที่ 4.1 จาก block diagram ของการ Transform coding ภาพอินพุทที่เข้ามาจะถูกแยกเป็นบล็อกเล็ก ๆ โดยเราสามารถกำหนดขนาดของบล็อกได้ว่าให้เป็นเท่าไร ขนาดของบล็อกที่เหมาะสมจะเป็นตัวเพิ่มประสิทธิภาพในการรวมพลังงานของการ Transform ซึ่งจะทำให้ภาพที่ได้มีรายละเอียดที่ดี ความเหมาะสมของขนาดบล็อกค่าหนึ่ง ๆ แต่อย่างไรก็ตามขนาดของบล็อกที่เหมาะสม สามารถคำนวณได้ด้วยคณิตศาสตร์ที่ยุ่งยากพอ ๆ กับการ Transform เลยทีเดียว โดยส่วนมากแล้วขนาดของบล็อกจะมีค่าเป็นเลขยกกำลังสองอย่างเช่น 4, 8, 16 ซึ่งได้มาจากสองยกกำลังสอง สองยกกำลังสาม สองยกกำลังสี่ ตามลำดับ หลังจากการแยกข้อมูลออกเป็นบล็อกย่อย ๆ แล้วจะทำการ Transform ไปโดยอิสระของแต่ละบล็อก โดยมีสมการของการ Transform มิติเดียวและสองมิติดังต่อไปนี้

$$f(k) = \sqrt{(2/N)}\,\alpha(k) \sum_{n=0}^{N-1} f(n) \cos\{(2n+1)\pi k / 2n\}$$

เมื่อ $k = 0, 1, \ldots\ldots, N-1$ ... (4.1.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ

$$\alpha\ (0) = \sqrt{(1/2)} \text{ และ } \alpha(k) = 1 \text{ เมื่อ } k \text{ ไม่เท่ากับ } 0 \qquad \ldots (4.1.2)$$

F (k) เป็นผลที่ได้จากการ Transform และ f (n) เป็นข้อมูลอินพุทตัวที่ n

ส่วนสมการของการ Transform แบบสองมิติสามารถเขียนได้ดังนี้ คือ

$$F(u, v) = \frac{2}{N}\alpha(u)\alpha(v)\sum_{m=0}^{N-1}\sum_{m=0}^{N-1} f(m.n)\cos\left\{\left(\frac{2m+1}{2N}\right)u\pi\right\}\cos\left\{\left(\frac{2n+1}{2N}\right)v\pi\right\}$$

เมื่อ

u, v เป็นตัวแปรของ discrete frequency มีค่าเป็น 0, 1, 2, ...N-1

f(m, n) เป็นตำแหน่งของจุดภาพภายในบล็อคขนาด N*N (0, 1, 2, ...N-1)

F(u, v) เป็นผลจากการทำ discrete cosine Transform

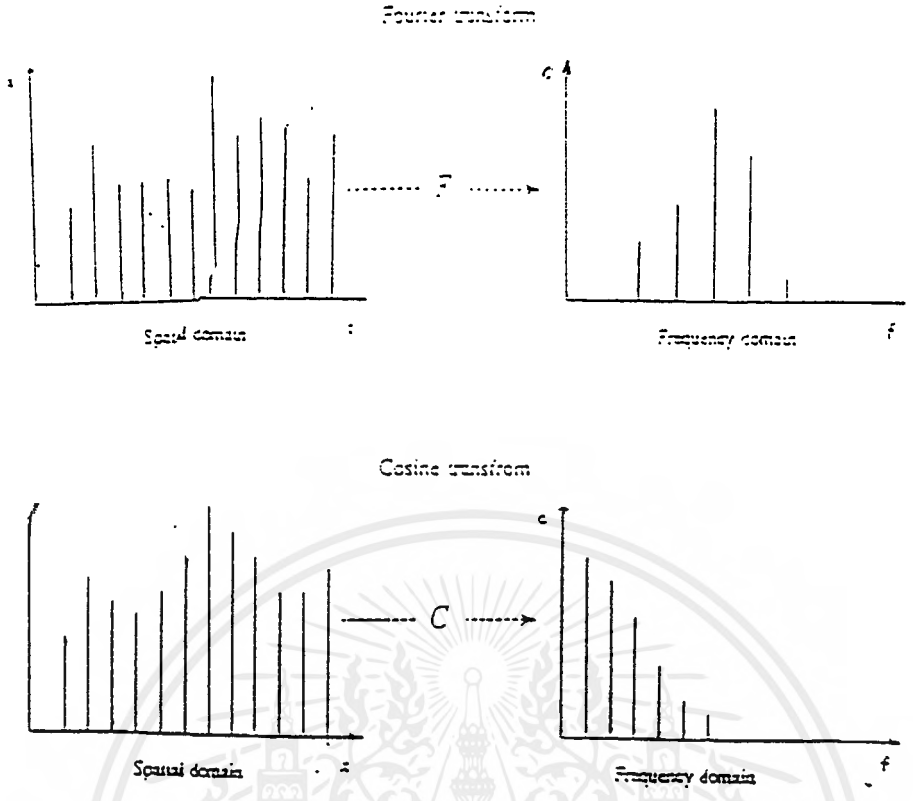
$\alpha(0) = \sqrt{(1/2)}$ และ $\alpha(j) = 1$ เมื่อ j ไม่เท่ากับ 0

ภาพที่ 4.1 ก เป็นการทำการแปลงข้อมูลให้อยู่ในโดเมนของความถี่ ซึ่งข้อมูลในโดเมนความถี่จะมีลักษณะเด่นอยู่ 3 ประการที่เห็นได้ชัด

1. ค่าความถี่ศูนย์ จะเป็นค่าของความเข้มเฉลี่ยของข้อมูล
2. ค่าความถี่สูงเป็นค่าที่บอกถึงข้อมูลที่มีการเปลี่ยนแปลงสูง
3. ค่าความถี่ต่ำ เป็นค่าที่บอกรายละเอียดโดยรวมของข้อมูล

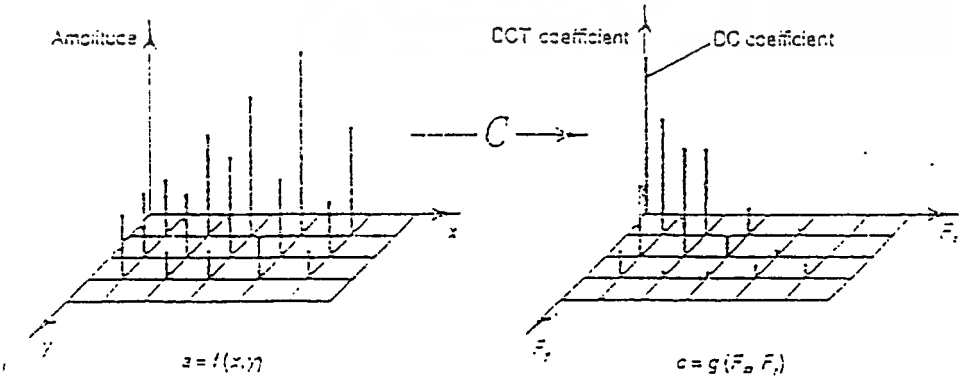
จากภาพที่ 4.1 ก จะสังเกตได้ว่าพลังงานที่ได้ในโนเมนของข้อความถี่ของการแปลงแบบฟูริเยร์นั้นจะมีลักษณะของการกระจายที่มากกว่าการแปลงแบบดีครีตโคไซน์ ซึ่งจะเห็นได้ชัดจากภาพที่แสดงการแปลงข้อมูล 2 มิติ โดย a เป็นข้อมูลของสัญญาณ และ c เป็นค่าขนาดของกลุ่มพลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 (ก) แสดงการแปลงแบบฟูริเยร์และการแปลงแบบดีสครีตโคไซน์ตามลำดับ

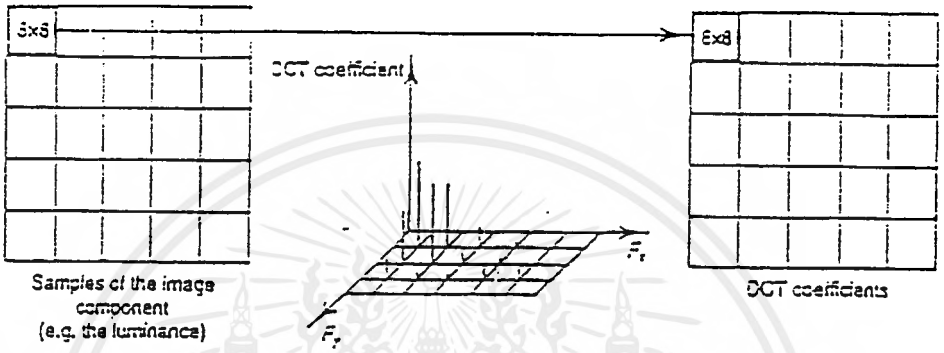
ในแต่ละบล็อกของ 8 x 8 นั้นจะมีด้วยกันทั้งหมด 64 ค่า โดยแต่ละค่าจะแสดงขนาด (Amplitude) ของแต่ละค่าที่เข้ามา ขนาดของสัญญาณนี้จะเป็นฟังก์ชั่นของจุดพิกัด 2 จุด โดยกำหนดให้ a = f(x, y) โดยที่ x, y จะเป็นมิติของสปาเชียลโดเมน ดังตัวอย่างในภาพลักษณะ 3 มิติในบางส่วนของบล็อก 8 x 8



รูปที่ 4.1(ข) แสดงการแปลดีสครีตโคไซน์แบบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากผ่านกระบวนการแปลงด้วยดีสรีตโคไซน์มาจะได้รูปแบบของฟังก์ชั่นที่อยู่ในโดเมนของความถี่โดยสมมติให้มีฟังก์ชั่นเป็น c = g (Fx, Fy) โดย c จะเป็นค่าของสัมประสิทธิ์ ส่วนค่าของ Fx และ Fy จะแสดงถึงค่าที่เกิดขึ้นในแกนของความถี่ในแต่ละทิศทาง โดยจากภาพที่ 4.1 ค เป็นการแสดงการแปลงรูปแบบดีสครีสโคไซน์

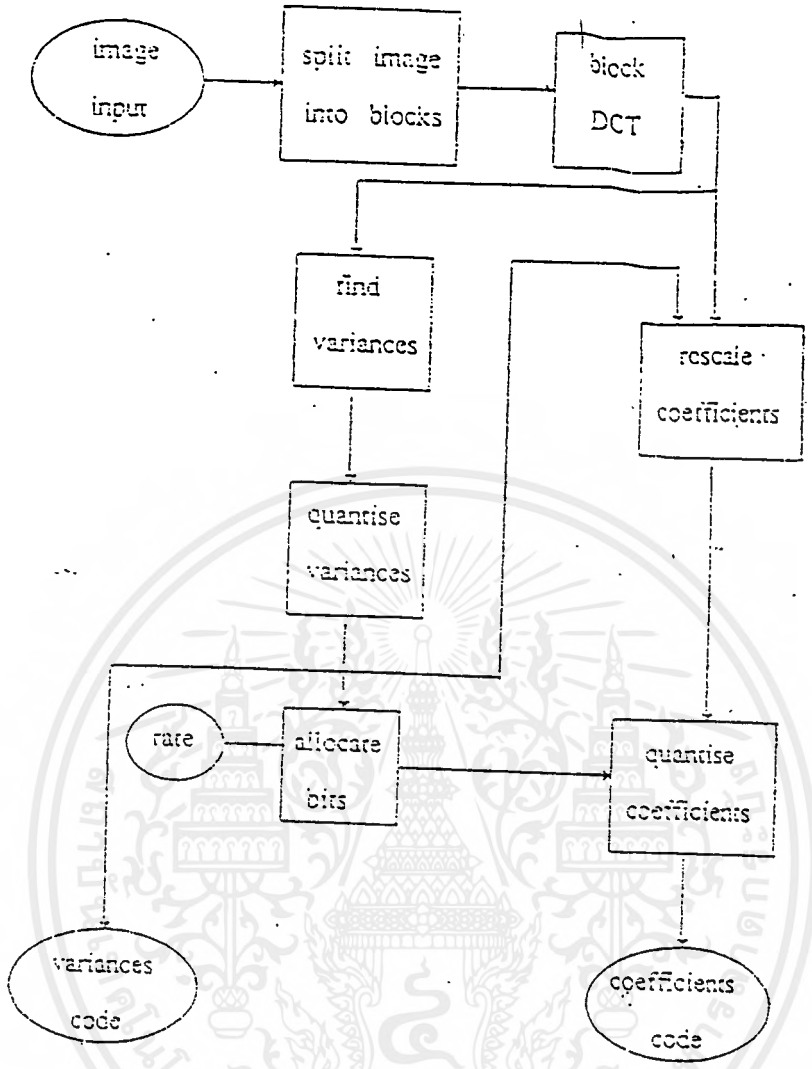


รูปที่ 4.1(ค) แสดงการแปลงดีสครีตโคไซน์ที่ทำในบล็อกขนาด 8 x 8

สัมประสิทธิ์ของ g (0, 0) นั้นจะเป็นสัมประสิทธิ์ที่ค่าความถี่ 0 ซึ่งจะเรียกว่าสัมประสิทธิ์ DC ซึ่งจะเป็นค่าเฉลี่ยของค่าในแต่ละบล็อก

ในแต่ละค่าภายในบล็อกของข้อมูลภาพ ซึ่งโดยปกติค่าของจุดภาพจะมีการเปลี่ยนแปลงน้อยจากจุดถึงจุด ดังนั้นขนาดของความถี่ที่ต่ำที่สุดจะสูงที่สุด ส่วนค่าของความถี่กลางและสูงจะมีค่าที่น้อยหรืออาจจะเป็นศูนย์ซึ่งในส่วนนี้จะไม่นำมาใช้อาจจะทอนลงไปดังแสดงในภาพที่ 4.2 ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แผนภาพแสดงตอน Forward Transform ของ DCT

หลังจากการผ่านการ Transform โดย Discrete Cosine Transform เพื่อเปลี่ยนข้อมูลให้อยู่ในรูปของ frequency domain พลังงานรวมส่วนใหญ่ของข้อมูลจะถูกเก็บ Z(pack) อยู่ในสัมประสิทธิ์เพียงไม่กี่ตัวของการ Transform จากสัมประสิทธิ์ที่ได้จะถูกนำมาหาค่า variances ของแต่ละบล็อค เพื่อที่จะนำค่า Standard deviation ไปคำนวณหา bit allocate สำหรับการจัดให้กับสัมประสิทธิ์แต่ละตัวในบล็อค และนอกจากนั้นยังส่งค่า Standard deviation ไปยังทางด้านรับด้วย เพื่อใช้ในตอน decode ดังนั้น จึงต้องทำการจัดระดับ (quantise) ค่านี้ก่อนที่จะเก็บหรือส่งไปยังด้านรับ และค่าสัมประสิทธิ์ตัวอื่น ๆ ก็จะถูกปรับระดับ (normalize) ด้วยค่าของ variances ที่ได้หลังจากการกำหนด bit allocate เพื่อปรับค่าของสัมประสิทธิ์ให้อยู่ในช่วงของค่าบิทเรทที่กำหนด ก่อนที่จะทำการ quantise ส่วนการกำหนดค่า bit allocate ของสัมประสิทธิ์ที่ตำแหน่ง (u, v) หลังการ Transform ในแต่ละบล็อค คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$b_{uv} = R + 0.5\log_2 \left\{ \sigma_{uv} 2 / \left(\prod_{k=0}^{n-1} \prod_{k=0}^{n=1} \sigma_{k1} 2 \right) 1/N \right\}$$

$$= \log_2 (\sigma_{uv}) + R - (1/N) \log_2 (\pi \ \pi \ \sigma_{k1}) \qquad \text{.....(4.1.4)}$$

เมื่อค่า σ_{uv} คือค่า Standard deviation ของสัมประสิทธิ์ ของการ Transform ที่ความถี่ (u, v) และ σ_{k1} เป็นค่า Standard deviation ของสัมประสิทธิ์ของการ Transform ที่ตำแหน่ง k1 โดย k และ 1 มีค่าเป็น 0 ถึง n – 1 เมื่อ n คือขนาดของบล็อก และ N คือจำนวนของสัมประสิทธิ์ในแต่ละบล็อค โดยที่ค่าคงที่ β กำหนดได้จาก

$$\beta = R - (1/N)\log \left[\prod_{k=0}^{n-1} \prod_{k=0}^{n-1} \sigma_{k1} \right] \qquad \text{.....(4.1.5)}$$

และค่าสูงสุดของบิทที่จะกำหนดให้กับ สัมประสิทธิ์แต่ละตัวจะกำหนดให้อยู่ได้ภาพค่า b_{max} $0 < b_{uv} < b_{max}$ และค่า b_{uv} เป็นจำนวนเต็ม

โดยที่ $\sum\sum_{u\ v} b_{uv} = B$ และ $B = R$ เมื่อ R คือ ค่าบิทเรทที่ต้องการ

หลังจากการคำนวณ bit allocation ค่าสัมประสิทธิ์ของการ Transform แต่ละตัวในบล็อกจะถูกปรับค่า (normalize) ให้อยู่ในช่วงขนาดของค่าไม่เกินจำนวนบิทที่กำหนด แล้วจึงทำการ quantise ค่าของสัมประสิทธิ์แต่ละตัว โดยอาศัยคุณสมบัติการกระจายพลังงานของสัมประสิทธิ์ของการ Transform ที่กล่าวไว้ว่า พลังงานส่วนใหญ่จะอยู่ในช่วงที่ความถี่ต่ำ ๆ หรืออาจนำรูปแบบการระจายความหนาแน่นแบบ Gaussian หรือ Laplacian มาใช้ก็ได้ ดังนั้นสัมประสิทธิ์ที่อยู่มุมซ้ายบนจะเป็นกลุ่มของความถี่ต่ำ ซึ่งเป็นส่วนที่สำคัญจะถูกกำหนดด้วยจำนวนบิทสูง ในขณะที่กลุ่มสัมประสิทธิ์ที่อยู่มุมขวาล่างจะเป็นกลุ่มของความถี่สูง ซึ่งถือว่าไม่สำคัญจะถูกกำหนดด้วยจำนวนต่ำๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ k = 0, 1, ..., N-1

เมื่อ f (n) เป็นผลที่ได้จากการทำ Inverse Transform กับ F (k) ซึ่งสัมประสิทธิ์จากการ Transform

กรณี แปลงกลับแบบ 2 มิติ (Two diminsional Inverse Transform)

$$f(m,n) = \frac{2}{N}\alpha(u)\alpha(v)\sum_{u=0}^{N-1}\sum_{u=0}^{N-1}F(u,v)\cos\left[\frac{(2m+1)u\pi}{2N}\right]\cos\left[\frac{(2N+1)v\pi}{2N}\right] \quad ...(4.1.7)$$

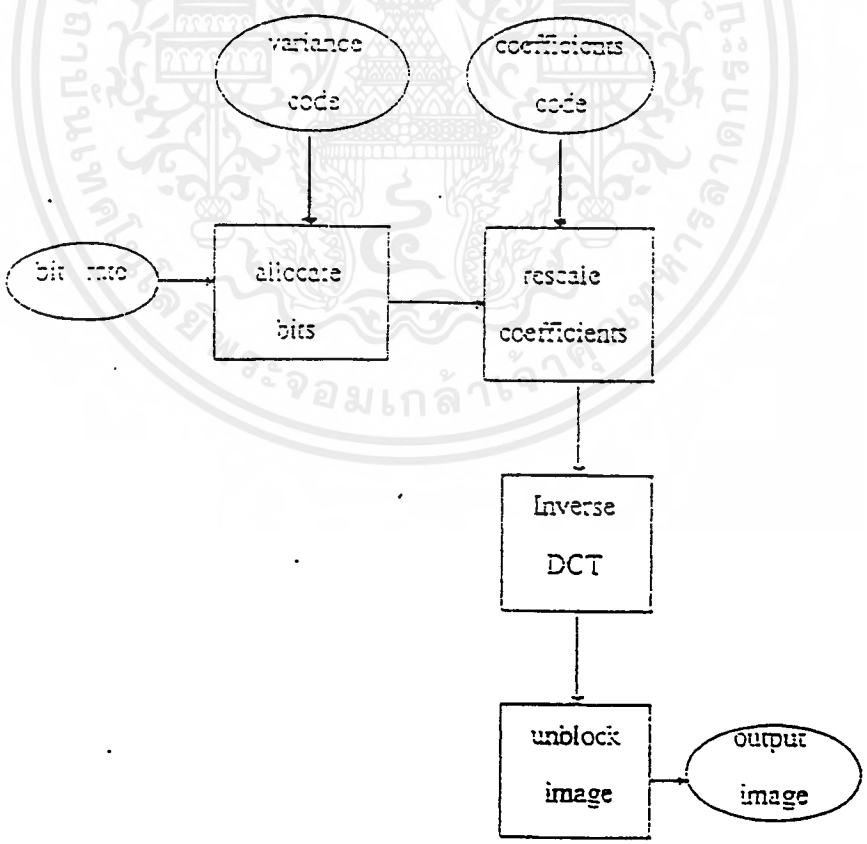
เมื่อ

$\alpha(0) = \sqrt{(1/2)}$ และ $\alpha(k) = 1$ เมื่อ k ไม่เท่ากับ 0

m, n เป็นตำแหน่งของจุดภาพ มีค่าตั้งแต่เป็น 0, 1, 2, ... N-1

F(u, v) เป็นค่าสัมประสิทธิ์ที่ได้จากการ Transform ภาพขนาด N*N

F(m, n) เป็นผลลัพธ์ของการทำ Inverse Transform



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงแม้ว่า การลดข้อมูลภาพโดยใช้หลักการของการ Transform จะเป็นที่ยอมรับ และได้รับความนิยมแพร่หลายมาจนทุกวันนี้ เพราะสามารถสร้างในลักษณะของ real time ได้ด้วย แต่ว่าของบิทเรทที่ใช้จะอยู่ในช่วง 1 ถึง 0.5 เพราะต่ำกว่านี้ คุณภาพของภาพที่ได้จะแย่มาก คือเป็นภาพที่ขาดรายละเอียด ส่วนของขอบภาพจะเบลอไม่มีความคมชัด ทั้งนี้เนื่องจากการแปลง data domain ไปอยู่ในรูปของ frequency domain นั้น ที่ช่วงความถี่สูง ๆ ซึ่งเป็นส่วนบริเวณขอบต่าง ๆ ในภาพมักจะถูกตัดทิ้งไปเพื่อให้ขนาดของข้อมูลลดลง จึงทำให้ขอบต่าง ๆ ในภาพไม่คมชัด และภาพที่ได้ยังมีลักษณะเป็นบล็อค ๆ ตามขนาดของบล็อคที่ถูกแบ่งในตอน Transform

4.2 การวัดความเหมือนจริงของภาพ (Image Fidelity)

ในการลดข้อมูลภาพนั้น จะมีส่วนหนึ่งที่เกิดผิดพลาดหรือสูญเสียไป ข้อผิดพลาดที่เกิดขึ้นนี้จะมีผลตอนที่สร้างภาพกลับคืนมา (reconstruction)และค่าความผิดพลาดนี้จะอยู่ในช่วงหนึ่งที่สามารถยอมรับได้ดังนั้นเกณฑ์การวัดความเหมือนจริงของภาพสามารถนำมาใช้ในการวัดประสิทธิภาพของภาพคือ ค่า root –mean – square (rms) ของ error ระหว่างข้อมูลภาพอินพุท และข้อมูลภาพเอาท์พุท (Signal – to – Noise Ratio) เมื่อกำหนดให้ข้อมูลภาพอินพุทประกอบด้วยอาเรย์ขนาด N*N ของจุดภาพ f(x, y) โดยที่ x, y มีค่าเป็น 0, 1, ... N-1 แต่ละจุดภาพมีค่าของระดับสีเทาที่เป็นไปได้ คือ 2 ยกกำลัง m เป็นจำนวนบิทของเลขฐานสอง

สำหรับทุกค่าของ x และ y ในช่วง 0, 1, ...N-1 ค่า error ระหว่างจุดภาพอินพุท และเอาท์พุทคือ

$$e(x, y) = g(x, y) - f(x, y) \qquad \text{.....(4.2.1)}$$

เมื่อ f(x, y) คือ input image ณ.จุด x, y ใด ๆ

เมื่อ g(x, y) คือ output image ณ.จุด x, y ใด ๆ

ค่าเฉลี่ยของ error1 กำลังสองของภาพ (mean square error) คือ

$$e2 = \left(1/N^2\right)\sum_{x=0}^{N-1}\sum_{x=0}^{N-1}\left\{e\left(x, y\right)\right\}^2$$

$$= \left(1/N^2\right)\sum_{x=0}^{N-1}\sum_{x=0}^{N-1}\left\{g\left(x, y\right) - f\left(x, y\right)\right\}^2 \qquad \text{.....(4.2.2)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น rms ของ error จึงสามารถเขียนได้ดังนี้คือ

$$e_{rms} = (e^2)^{1/2} \qquad \text{.....(4.2.3)}$$

ค่า root mean square error เป็นค่าที่ใช้ในการวัดความแตกต่างของข้อมูลอินพุทกับข้อมูลเอาท์พุท แต่เมื่อพิจารณาขนาดของข้อมูลเอาท์พุท ต่อขนาดสัญญาณรบกวน (noise) ก็จะได้เป็นค่า Siganl – to – Noise (SNR) เมื่อกำหนดได้สัญญาณภาพเอาท์พุทแต่ละจุดประกอบด้วยสัญญาณอินพุทบวกด้วยค่าสัญญาณรบกวน นั้นคือ

$$g(x, y) = f(x, y) + e(x, y) \qquad \text{.....(4.2.4)}$$

ดังนั้นค่า mean square Signal – to – Noise ของข้อมูลภาพเอาท์พุท สามารถหาได้โดยค่าเฉลี่ยของสัญญาณอินพุทกำลังสอง $f^2(x, y)$ หารด้วยค่าเฉลี่ยของสัญญาณรบกวนกำลังสอง $e^2(x, y)$ ของข้อมูลภาพทั้งหมด ซึ่งสามารถเขียนได้ดังนี้

$$(SNR) = \frac{(1/N^2)\sum_{x=0}^{N-1}\sum_{x=0}^{N-1} f^2(x, y)}{(1/N^2)\sum_{x=0}^{N-1}\sum_{x=0}^{N-1} e^2(x, y)} \qquad \text{.....(4.2.5)}$$

ค่า rms ของ SNR จึงสามารถเขียนได้ดังต่อไปนี้

$$(SNR) = \left\{(1/N^2)\sum_{X=0}^{N-1}\sum_{X=0}^{N-1} g^2(x, y)/(1/N^2)\sum_{X=0}^{N-1}\sum_{X=0}^{N-1} e^2(x, y)\right\}1/2 \qquad \text{.....(4.2.6)}$$

โดยที่เทอมส่วนของสมการข้างบน เป็นสมการของสัญญาณรบกวนที่อยู่ในรูปของผลต่างระหว่างข้อมูลอินพุทกับเอาท์พุท

หรือ เราอาจคิดเป็นค่า $(SNR)_{rms}$ dB ได้โดย

$$(SNR)rms\ dB = 20\ log(SND)rms \qquad \text{.....(4.2.7)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวิธีการที่ใช้ในการวัดความเหมือนจริงของภาพที่กล่าวมาข้างบนนี้ ไม่สามารถที่จะบ่งบอกหรือใช้เป็นเกณฑ์ในการพิจารณาได้แต่เพียงอย่างเดียว ในกรณีของภาพเอาท์พุทที่ได้จากการประมวลผล หรือรับส่งสัญญาณ โดยสายตาของมนุษย์เป็นตัวรับภาพจากจอภาพอีกที่หนึ่ง อย่างเช่น broadcast TV, ppicture phone , หรืองานต่าง ๆ ที่มีการใช้ Image Processing เป็นต้น ระบบการมองเห็นของมนุษย์จะมีคุณสมบัติพิเศษกล่าวคือ ระบบการมองเห็นจะมีคุณสมบัติพิเศษกล่าวคือ ระบบการมองเห็นของมนุษย์จะมีคุณสมบัติพิเศษกล่าวคือ ระบบการมองเห็นจะมีคุณสมบัติพิเศษกล่าวคือ ระบบการมองเห็นของตาจะไวต่อความเข้มของแสงในลักษณะของ logarithmic ดั้งนั้น error ในบริเวณที่เป็นที่มืดของภาพ จะเป็นได้ชัดกว่า error ที่อยู่ในบริเวณที่สว่าง และระบบการมองเห็นยังไวต่อการเปลี่ยนแปลงอย่างทันที่ทันใด ของระดับความเทาด้วย error ที่อยู่บนขอบหรือใกล้ ๆ ขอบของภาพจะเป็นได้ชัดกว่า error ที่อยู่ในโครงสร้างที่เป็นฉากหลังของภาพ ด้วยเหตุนี้ถึงแม้ว่าภาพสองภาพจะมีค่าของ rms, error เท่ากัน แต่อาจจะปรากฏความแตกต่างของคุณภาพของการมองเห็นที่แกตต่างกันไป

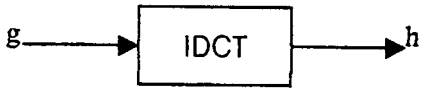
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ A^T คือ ของเมตริกซ์

และ Inverse Discrete Cosine Transform ของ g แสดงได้ดังสมการที่ (2)

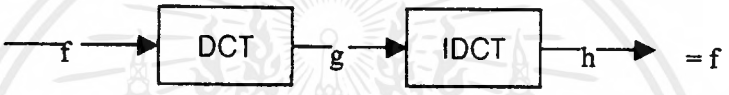
$$h = IDCT(g)$$

$$h = A^T gA \qquad \text{.....(4.3.2)}$$



รูปที่ 4.3 แสดงแผนภาพการแปลงกลับเมตริกซ์

เมื่อ f saturated จะได้



รูปที่ 4.4 แสดงแผนภาพการแปลงกลับเมตริกซ์

ในการแปลงเมตริกซ์ ขนาด 8 x 8 (N = 8) จะได้ค่า A ดังนี้

$$A = \begin{pmatrix} 0.354 & 0.49 & 0.462 & 0.416 & 0.354 & 0.278 & 0.191 & 0.098 \\ 0.354 & 0.416 & 0.191 & -0.098 & -0.354 & -0.49 & -0.462 & -0.278 \\ 0.354 & 0.278 & -0.191 & -0.49 & -0.354 & 0.098 & 0.462 & 0.416 \\ 0.354 & 0.098 & -0.462 & -0.278 & 0.354 & 0.416 & -0.191 & -0.49 \\ 03.54 & 0.098 & -0.462 & 0.278 & 0.354 & -0.416 & -0.191 & 0.49 \\ 0.354 & -0.278 & -0.191 & 0.49 & -0.354 & -0.098 & 0.462 & -0.416 \\ 0.354 & -0.416 & 0.191 & 0.098 & -0.354 & 0.49 & -0.462 & 0.278 \\ 0.354 & -0.49 & 0.462 & -0.416 & 0.354 & -0.278 & 0.191 & -0.098 \end{pmatrix}$$

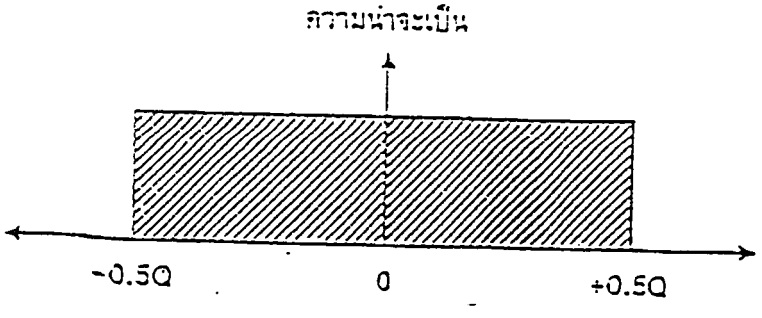
<u>โดยมีผลการทดลองดังนี้</u>

f คือข้อมูลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การควอนไทซ์ (Quantization)

องค์ประกอบอย่างหนึ่งที่มีผลกระทบต่อความเที่ยงตรงของการเปลี่ยนสัญญาณจากอนาลอกเป็นดิจิตอลก็คือการควอนไทซ์ เนื่องจากสัญญาณอินพุทเริ่มต้นนั้นเป็นสัญญาณที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง ดังนั้นในการวัดค่าของสัญญาณที่มีความเปลี่ยนแปลงอย่างต่อเนื่อง ดังนั้นในการวัดค่าของสัญญาณในขณะใดขณะหนึ่ง (Sample) โดยค่าที่ได้นั้นจะมีความเที่ยงตรงหรือไม่นั้นจะขึ้นอยู่กับอุปกรณ์หรือเครื่องมือที่นำมาใช้ในการวัด โดยตัวของการเปลี่ยนสัญญาณจากอนาลอกเป็นดิจิตอลนั้น จะทำการแบ่งระดับของสัญญาณอินพุทออกเป็นช่วง ๆ โดยช่วงเหล่านั้นจะแทนค่าที่เป็นตัวเลขค่าหนึ่ง ๆ สำหรับประโยชน์ของวิธีการนี้พบเห็นโดยทั่วไปก็คือการนำมาประยุกต์ใช้กับ การมอดูเลตแบบรหัสพัลส์ (Pulse Code Modulation) ในโทรศัพท์ โดยการแบ่งช่วงของสัญญาณอาจจะไม่เท่ากัน (Nonuniform) ตลอดทั้งช่วงของสัญญาณก็ได้ อย่างไรก็ตามในการเปลี่ยนให้เป็นสัญญาณดิจิตอลนั้นเราอาจจะเรียกการดิจิไตท์ของสัญญาณ (Digitize) โดยทั่วไปของการควอนไตซ์สัญญาณภาพจะแบ่งช่วงของการควอนไตซ์ออกเป็นออกเป็นช่วงที่เท่า ๆ กัน (Uniform quantization) สำหรับในการแทนค่าที่เป็นตัวเลขให้กับสัญญาณอินพุททำได้โดยการแบ่งระดับของการควอนไทซ์ออกเป็นช่วง ๆ ที่เท่า ๆ กัน (Uniform quantization) สำหรับในการแทนค่าที่เป็นตัวเลขให้กับสัญญาณอินพุททำได้โดยการแบ่งระดับในการควอนไทซ์ออกเป็นช่วง ๆ แล้วทำการเปรียบเทียบกับสัญญาณอินพุทที่เข้ามาตกอยู่ในช่วงที่แบ่งไว้แล้วแทนค่าของสัญญาณอินพุทตกอยู่ขณะนั้น และในขณะเดียวกันก็เกิดค่าผิดพลาดขึ้นมาด้วยแต่ค่าผิดพลาดนี้จะไม่เกินครึ่งหนึ่งของช่วงการควอนไทซ์แต่ละช่วง ซึ่งค่าผิดพลาดนี้จะอยู่ในระหว่าง 0.5 ถึง + 0.5 ของช่วงการควอนไทซ์แต่ละช่วง (Q) ดังแสดงในภาพที่ 4.2 ก ซึ่งจะเป็นการกระจายของความน่าจะเป็นแบบยูนิฟอร์มของการผิดพลาดเนื่องจากควอนไทซ์ (Probability distribution of quantize error)



รูปที่ 4.5 แสดงการกระจายของความน่าจะเป็นแบบยูนิฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุทก่อนที่จะผ่านการควอนไทซ์ ผลที่ได้นั้นเมื่อดูจากการทรานส์เฟอร์ฟังก์ชั่นแล้วจะมีความราบเรียบเป็นอย่างมาก และโดยทั่วไปแล้วรูปแบบของการควอนไทซ์นั้นสามารถที่จะแบ่งออกได้เป็น 2 แบบด้วยกันคือ

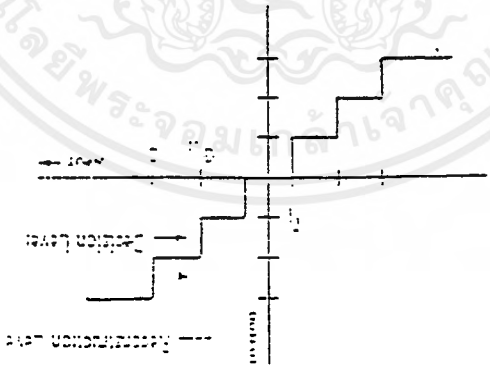
1. ตัวควอนไตแบบมิดเทรด (midtread uniform quantizer)

2. ตัวควอนไทซ์แบบมิดไรซ์ (Midrise quantizer)

เช่นเดียวกัน เมื่อพิจารณาที่การควอนไทซ์แบบยูนิฟอร์มแล้วสามารถที่จะแบ่งออกได้เป็น 2 แบบเหมือนกันคือ

1. ตัวควอนไทซ์แบบมิดเทรดยูนิฟอร์ม (Midtread uniform quantizer) ลักษณะของการควอนไตซ์แบบนี้คือค่าของสัญญาณอินพุทที่ในช่วงใกล้กับค่าที่เป็นศูนย์นั้นจะถูกแทนด้วยค่าของสัญญาณเอาท์พุทที่เป็นศูนย์ ดังแสดงในภาพที่ 4.3 ก

2. ตัวควอนไทซ์แบบมิดเทรดยูนิฟอร์ม (Midrise uniform quantizer) จากภาพที่ 4.3 ข จะสัญญณอินพุทสัญญาณที่อยู่ใกล้กับค่าที่เป็นศูนย์ในช่วงบวก (+) และ (-) โดยแทนค่าสัญญาณอินพุทให้ใกล้กับค่าที่เป็นศูนย์ทางด้านบวก ด้วยค่าของระดับที่เป็นค่าบวกให้กับสัญญาณเอาท์พุท และแทนค่าสัญญาณอินพุทใกล้กับค่าที่เป็นศูนย์กลางด้านลบ ด้วยค่าของระดับที่เป็นค่าลบให้กับสัญญาณเอ้าท์พุท



รูปที่ 4.6 แสดงระดับของการควอนไทซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

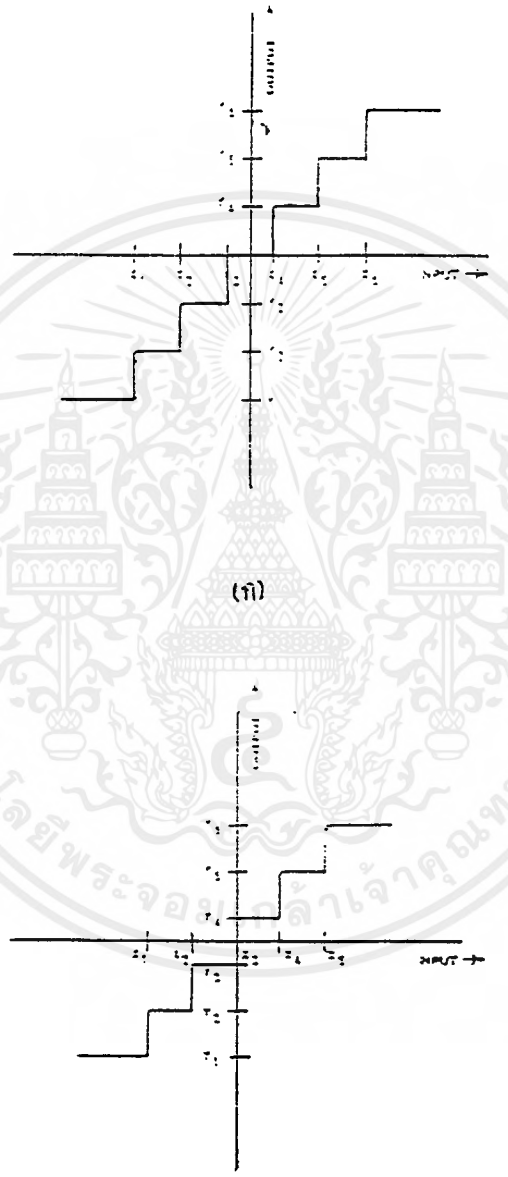
ในการประมวลผลของสัญญาณทางเชิงเลข จะแบ่งระดับหรือย่านของข้อมูลออกเป็นช่วง ๆ โดยที่แต่ละช่วงจะแทนระดับของข้อมูลที่เป็นค่าเดี่ยว ๆ ในที่นี้ก็คือการควอนไทซ์ ซึ่งสามารถแบ่งได้เป็น 2 ประเภท ได้แก่

1. แบบยูนิฟอร์ม (Uniform)
2. แบบนอน-ยูนิฟอร์ม (Non – Uniform)

ในที่นี้ได้เลือกการควอนไทซ์แบบยูนิฟอร์มดังแสดงในภาพที่ 4.3 ค ในการออกแบบตัวควอนไทซ์นั้นโดยจะมี Di : I=0, 1, 2,...L หมายถึงระดับของการตัดสินใจ (Decision Level) เมื่อมีข้อมูลอินพุทเข้ามาในช่วงของระดับของการตัดสินใจจะทำให้เกิด Ri ; I = ระดับของข้อมูลผลลัพธ์ (Reconstruction Level)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในการปฏิบัติแล้วนั้น ตัวอุปกรณ์ที่ใช้ในการเปลี่ยนสัญญาณจากอนาลอกเป็นดิจิตอล จะมีการชดเชยในส่วนของการเกิดความผิดพลาดจากการควอนไทซ์สัญญาณ โดยใช้เทคนิคที่เรียกว่าการ Dithering โดยเทคนิคนี้จะทำการบวกสัญญาณรบกวนเข้าไปในสัญญาณ



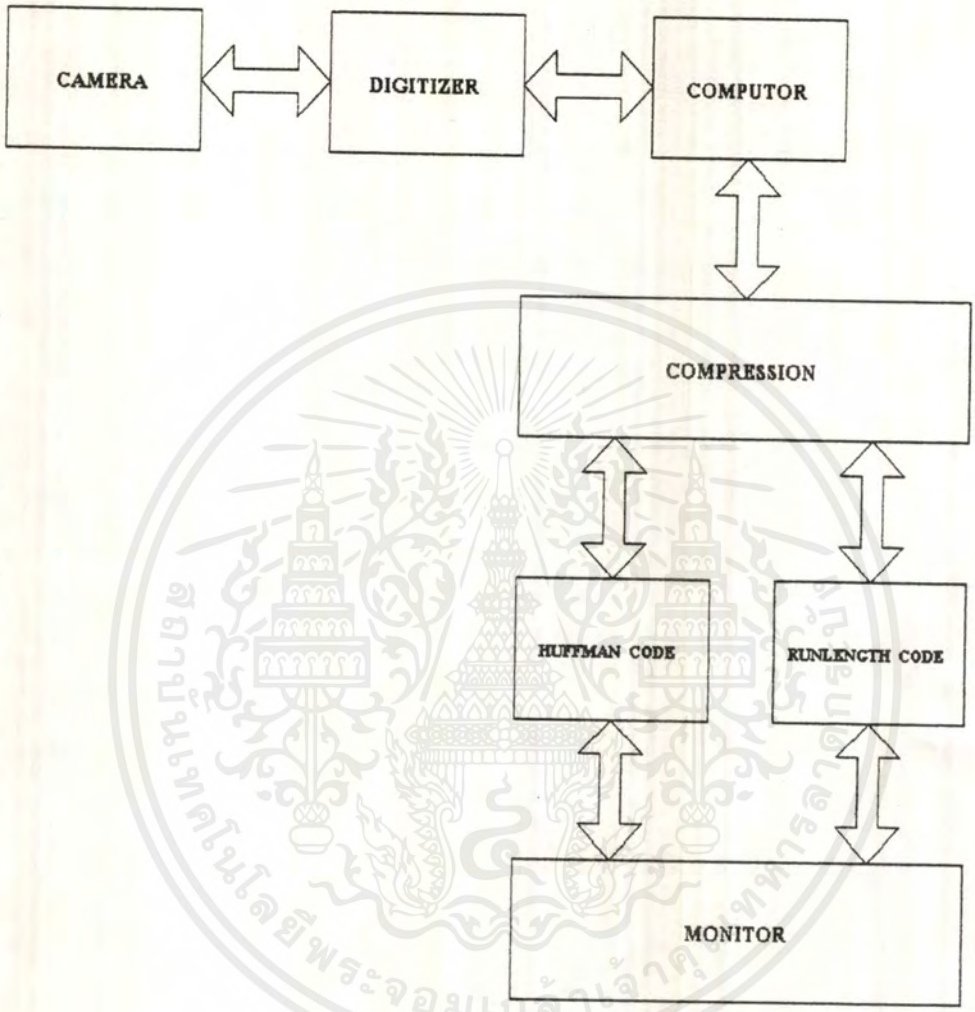
(ก)

(ข)

รูปที่ 4.7 **แสดงภาพ** (ก) **และ** (ข) **แสดงการควอนไทซ์แบบมิดเทรดและแบบมิดไรซ์ตามลำดับ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทำงานการประมวลผลภาพ



รูปที่ 5.1 แสดงปลือกไดอะแกรมการประมวลผลภาพ

ในปลือกแรกเป็นส่วนที่ใช้จับภาพภายนอกโดยกล้องดิจิตอลเป็น อินพุท ปลือกนี้จะทำหน้าที่เปลี่ยนสัญญาณภาพให้เป็นสัญญาณไฟฟ้าหรือที่เรียกว่าอนาลอก ปลือกที่สองจะรับสัญญาณอนาลอกจากส่วนแรกมาทำการเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอล ในส่วนนี้เรียกว่า ดิจิไตเซอร์ ส่วนที่สามเป็นส่วนประมวลผลคือคอมพิวเตอร์ซึ่งเปรียบเทียบเสมือนสมอง ทำหน้าที่ประมวลผลและวิเคราะห์ข้อมูลภาพ โดยในโครงงานนี้จะนำเอาคอมพิวเตอร์มาทำการลดข้อมูลภาพแสดงผลการทำ งานต่างๆ ที่จอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรม

รูปที่ 5.2 แสดงหน้อ จอ Video Previwe Record Mode

1.เมนูหลัก ประกอบไปด้วยหลายฟังก์ชั่น เช่น File, Sutup, Bmp Compress, Help และการทำงาน 3 โหมด คือ Video Previwe Record Mode , Playback and Compression Mode และ Open bmp Mode

1.1 Video Previwe Record Mode ดังที่แสดงในรูป 5.2 จะแสดงภาพ Realtime จากกล้อง ซึ่งจะมีฟังก์ชั่นสำหรับใช้งานเกี่ยวกับภาพ เช่น ปรับความคมชัดภาพ เลือกขนาดภาพและการบันทึกภาพจากกล้อง ซึ่งทำได้โดยการกดปุ่ม Recording จะมีหน้าต่างขึ้นมาให้ใส่ชื่อไฟล์ที่จะบันทึก ที่มีนามสกุลไฟล์เป็น (avi) จากนั้นกดปุ่ม OK โปรแกรมก็จะทำการบันทึกภาพจากกล้องทันที เมื่อต้องการหยุดการบันทึกภาพทำได้โดยกดปุ่ม Stop โปรแกรมจะหยุดการบันทึก ถ้าต้องการ บันทึกภาพเป็นภาพนิ่ง ก็กดปุ่ด Save bmp

1.2 Playback and Compress Mode ดังที่แสดงในรูป 5.3 ทำหน้าที่แสดงไฟล์ที่บันทึกไปจาก Video Previwe Record Mode หรือ ไฟล์ที่มีนามสกุล avi(*.avi) และหน้าที่สำคัญอีกอย่างหนึ่งคือการ Compression หรือ การลดข้อมูลภาพ ทำได้โดยใส่ไฟล์ที่ต้องการจะลดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

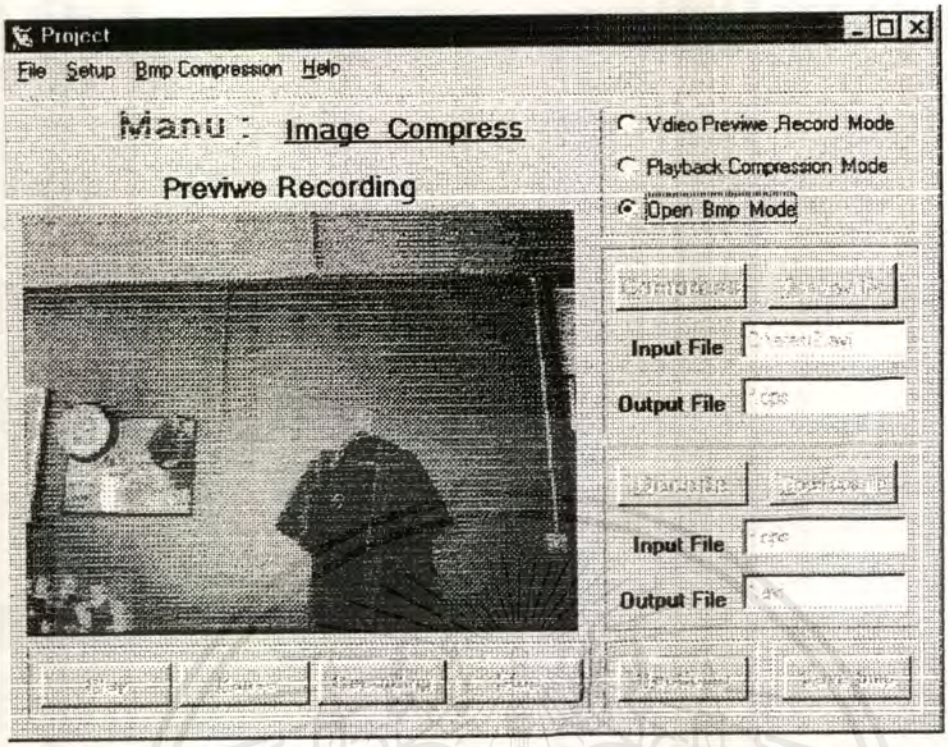
ภาพที่มีนามสกุล avi ในช่อง Input file หรือสามารถหาได้จากการกดปุ่ม Open avi file แล้วเลือกไฟล์ที่ต้องการจะลดข้อมูล ก็จะมีไฟล์ปรากฏที่ ช่อง input file จากนั้นตั้งชื่อไฟล์ที่ได้จากการลด

1.3 ข้อมูลภาพที่มีนามสกุล cps(*.cps) จากนั้นกดปุ่ม Compress โปรแกรมจะทำการลดข้อมูลภาพจะเสร็จเร็วหรือช้าขึ้นนั้นอยู่กับขนาดของไฟล์ที่นำมาลดข้อมูลถ้าไฟล์ Input มีขนาดใหญ่ก็นานกว่าไฟล์ขนาดเล็ก ส่วนการ Decode หรือแปลงข้อมูลกับก็ทำได้คล้ายๆ กันเพียงแต่ไฟล์ Input และ Output สลับกัน

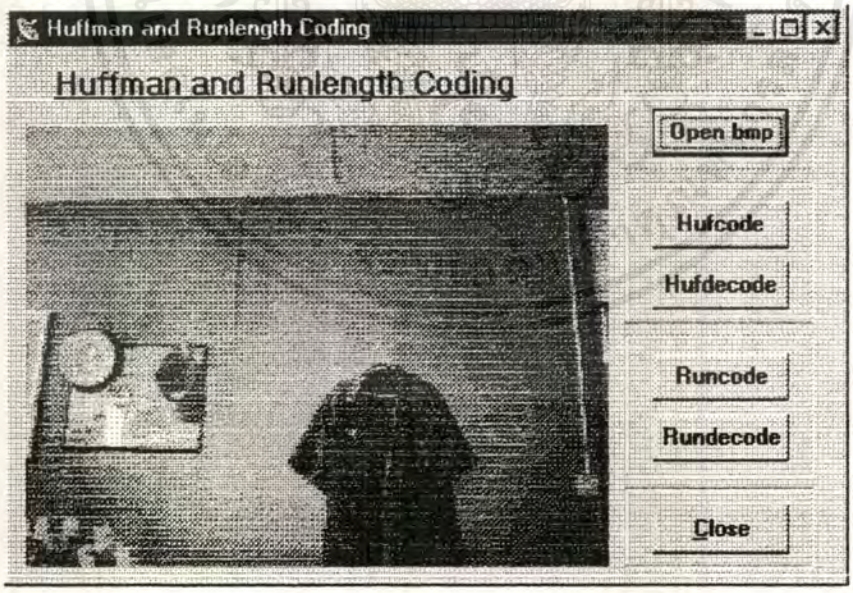
รูปที่ 5.3 เมนูหลัก Playback and Compression Mode

1.3 Open Bmp Mode ทำหน้าที่แสดงภาพที่ทำการบันทึกเป็นภาพนิ่ง ซึ่งมีนามสกุลว่า bmp (*.bmp)ดังแสดงในรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 เมนูหลัก Open bmp Mode



รูปที่ 5.5 เมนูการลดข้อมูลภาพนิ่ง

2. เมนูการลดข้อมูลภาพนิ่ง มี 2 แบบ คือการลดแบบ Huffman และแบบ Runlength ดังแสดงในรูปที่ 5.5 ซึ่งจะมี Input fil e เป็นชุด bmp การทำงานของฟังก์ชั่นปุ่มต่างๆ คล้ายกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะยกตัวอย่างการใช้งานโปรแกรมลดข้อมูลภาพนิ่งแบบ Huffman อย่างเดียว เริ่มจาก เลือกไฟล์ที่ต้องการลดข้อมูลซึ่งเป็นไฟล์จุด bmp(*.bmp) แล้วกดปุ่ม Hufcode จะมีหน้าต่างขึ้นมาให้ใส่ชื่อไฟล์หลังจากลดข้อมูลแล้วใส่ชื่อไฟล์แล้วก็กด Save โปรแกรมจะทำการลดข้อมูลภาพ เมื่อลดข้อมูลเสร็จก็จะ Message ขึ้นมาแสดงว่าการลดข้อมูลเสร็จสิ้น การแปลงภาพกลับก็ทำได้กดปุ่ม Hufdecode จะมีหน้าต่างขึ้นมาให้ใส่ไฟล์แปลงกลับจุด huf(*.huf) เสร็จแล้วกด OK ปรแกรมจะทำการแปลงข้อมูลภาพกลับ เมื่อแปลงข้อมูลภาพกลับเสร็จก็จะมี Message ขึ้นมาแสดงว่าการแปลงข้อมูลกับเสร็จสิ้น การลดข้อมูลแบบ Runlength ก็ทำคล้ายกันต่างกันตรงปุ่มใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

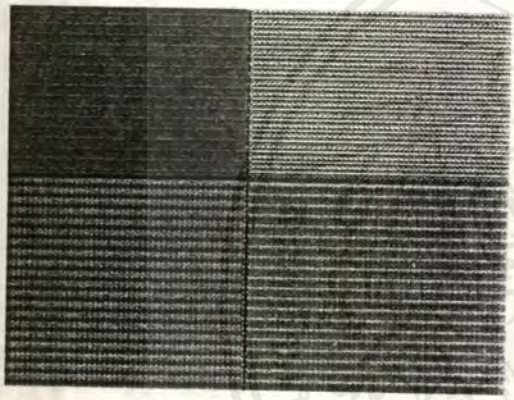
การทดลองและผลการทดลอง

การทดลองนี้ได้จัดทำขึ้นเพื่อแสดงให้เห็นว่าโครงงานนี้สามารถจะลดข้อมูลได้ที่จะลดข้อมูลได้จริง และทำการเปรียบเทียบผลของการลดข้อมูลของภาพที่มีความละเอียดของภาพแตกต่างกันได้ดังที่จะได้แสดงต่อไปนี้

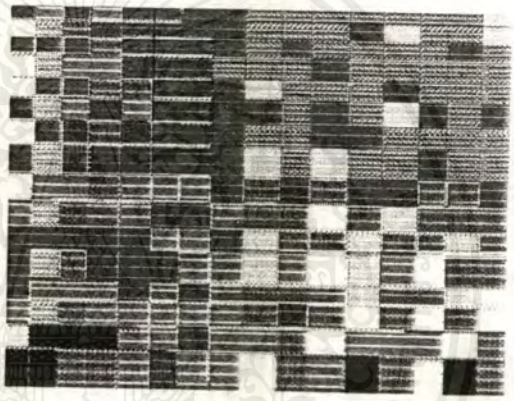
6.1 ขั้นตอนการทดลอง

ลำดับการทดลองเป็นดังนี้

1. ในขั้นแรกทดลองการลดข้อมูลภาพนิ่ง โดยใช้ภาพที่มีขนาดข้อมูลเท่ากัน แต่มีความละเอียดของภาพไม่เท่ากันดังแสดงในรูป 6.1 (test1.bmp)และ 6.2 (test2.bmp)

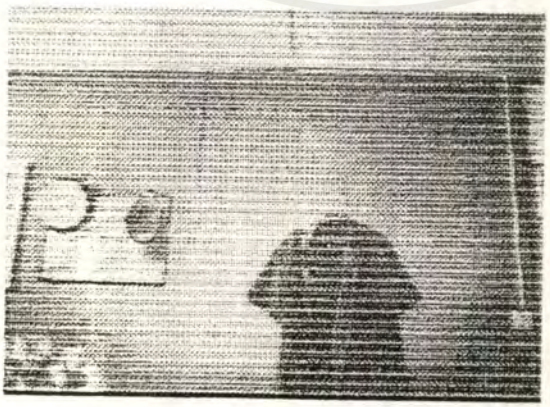


รูปที่ 6.1 ภาพความละเอียดภาพต่ำ (test1.bmp)



รูปที่ 6.2 ภาพความละเอียดภาพสูง(test2.bmp)

2. ทำการบันทึกภาพจากกล้องเก็บภาพเป็น (bmp file) ภาพที่ได้จากกล้องถือเป็นภาพที่มีความละเอียดของภาพมากที่สุด ดังแสดงในรูป (test3.bmp);



รูปที่ 6.3 แสดงภาพที่บันทึกจากกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ทำการลดข้อมูลภาพที่ละภาพโดยแยกตามวิธีการลดข้อมูลทั้งสองแบบ Huffman และ Runlength

ผลการทดลอง 6.1

เมื่อทำการทดลองแล้วจะได้ผลดังนี้

ตารางที่ 6.1.1 แสดงผลการทดลองการลดข้อมูลแบบ Huffman

File	Starting Size	Huffman Size	Ratio
Test1	230456	5180	97.75%
Test2	230456	10980	95.23%
Test3	153654	47600	69.02%

ตารางที่ 6.1.2 แสดงผลการทดลองการลดข้อมูลแบบ Runlength

File	Starting Size	Runlength Size	Ratio
Test1	230456	1078	99.53%
Test2	230456	15260	93.37%
Test3	153654	50810	66.93%

1. เมื่อทำการลดข้อมูลภาพที่มีความละเอียดน้อย ไฟล์ test1.bmp จากข้อมูลทั้งหมด 230456 ไบท์ลดข้อมูลภาพเป็นไฟล์ test1.run จะเหลือข้อมูลเพียง 1078 ไบท์ สามารถลดข้อมูลได้ถึง 229378 ไบท์และ test1.huf จะเหลือเพียง 5180 ไบท์ สามารถลดข้อมูลได้ถึง 219476 ไบท์

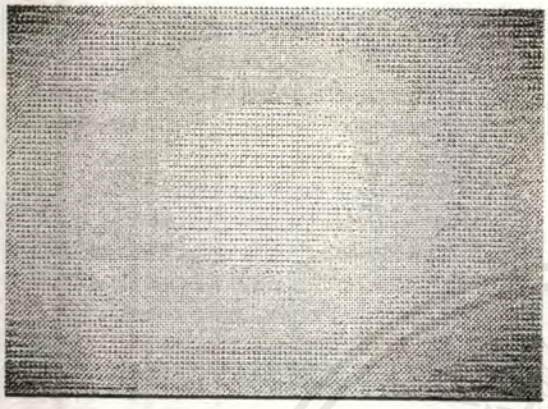
2. เมื่อทำการลดข้อมูลภาพที่มีความละเอียดสูง ไฟล์ test2.bmp จากข้อมูลทั้งหมด 230456 ไบท์ลดข้อมูลภาพเป็นไฟล์ test2.huf จะเหลือข้อมูลเพียง 10980 ไบท์ สามรถลดข้อมูลได้ถึง 219470 ไบท์และ test2.run จะเหลือข้อมูลเพียง 215196 ไบท์

3. เมื่อทำการลดข้อมูลภาพที่มีความละเอียดสูงสุดที่ได้จากกล้องไฟล์ test3.bmp จากข้อมูลทั้งหมด 153654 ไบท์ ลดข้อมูลภาพเป็นไฟล์ test3.huf จะเหลือข้อมูลภาพ 47000 ไบท์ สามรถลดข้อมูลได้ 106654 ไบท์ และ test3.run เหลือข้อมูลเพียง 50810 ไบท์ ลดข้อมูลได้ 102844 ไบท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.2

1. ทำการบันทึกภาพเคลื่อนไหวจากกล้องที่มีความละเอียดภาพน้อย ดังรูปที่ 6.4 (test4.avi)



รูปที่ 6.4 ภาพบันทึกจากกล้องความละเอียดภาพต่ำ

รูปที่ 6.5 ภาพบันทึกจากกล้องความละเอียดสูง

2. ทำการบันทึกภาพเคลื่อนไหวความละเอียดสูงภาพจากกล้อง ดังรูป 6.5(test5.avi)
3. ทำการลดข้อมูลภาพทั้งสองแบบแล้วเปรียบเทียบกัน

ตารางที่ 6.2 แสดงการทดลองการลดข้อมูลภาพเคลื่อนไหว

File	Starting Size	Huffman Size	Runlength Size	Ratio	
				huf	run
Test4	35488680	18382681	20476899	48.20%	42.30%
Test5	35719144	25700597	27092251	28.04%	24.15%

จากการทดลองลดข้อมูลภาพเคลื่อนไหว (test4) ที่มีความละเอียดของภาพน้อย โดยการลดข้อมูลทั้งสองแบบสามารถลดข้อมูลได้ใกล้เคียงกัน คือประมาณ 40-50 % และ (test5) ซึ่งที่มีความละเอียดภาพสูงกว่าหรือเป็นภาพที่ใช้งานจริงทั่วไปสามารถลดข้อมูลได้ 20-30 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปและวิจารณ์การทดลอง

จากการทดลองนี้สรุปได้ว่าโปรแกรมสามารถได้แสดงการนำเสนอการลดข้อมูลได้จริง โดยจะมีกรรมวิธี 2 แบบ คือ Huffman encode และ Runlength encode โดยจะสามารถลดข้อมูลได้เท่าไรนั้นขึ้นอยู่กับความละเอียดของภาพที่นำมาบีบอัดข้อมูลและคุณสมบัติของแต่ละวิธีของการบีบอัดข้อมูลภาพที่ได้เสนอไว้สองแบบข้างต้น

ในการทดลองที่ 6.1 จะเห็นว่าเมื่อนำภาพที่มีความละเอียดของภาพน้อยมาทำการบีดอีดข้อมูลจะสามารถลดข้อมูลของภาพได้มากกว่าภาพที่มีความละเอียดของภาพสูง โดยการบีบทั้งสองแบบจะให้ผลเหมือนกันแต่จะแตกต่างกันที่คุณสมบัติของแต่ละวิธีการคือ วิธีการลดข้อมูล แบบ Runlength codeที่ความละเอียดของภาพต่ำมากๆ จะมีความสามารถในการบีบข้อมูลได้ดีกว่า แบบ Huffman code แต่ที่ภาพมีความละเอียดของภาพสูงสามารถบีบข้อมูลได้น้อยกว่าแบบ Huffman code แต่ในการใช้งานจริงภาพที่ใช้งานในการสือสารต่างๆเป็นภาพที่มีความละเอียดสูง วิธี Huffman code จึงถูกใช้งานในขั้นตอนการบีบอัดข้อมูลต่างๆมากกว่าแบบ Runlength code เช่น ขั้นตอนสุดท้ายของการบีบอัดข้อมูลมาตรฐาน JPEG ในการทดลองที่ 6.2 เป็นการบีบข้อมูลภาพเคลื่อนไหวก็จะแสดงให้เห็นเช่นเดียวกันกับการทดลองที่ 6.1 คือที่ความละเอียดของภาพต่ำเปอร์เซนต์การบีบข้อมูลภาพจะสูงกว่าภาพที่มีความละเอียดสูง

สามารถสรุปได้ว่าโครงงานนี้สามารลดข้อมูลได้จริง ขึ้นอยู่กับความละเอียดของภาพซึ่งภาพที่ใช้งานเป็นภาพที่มีความละเอียดของภาพสูง ในโปรแกรมนี้ถือเป็นเพียงการศึกษาวิธีการลดข้อมูลภาพภาพเบื้องต้นเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปปัญหาและอุปสรรค์และแนวทางแก้ไข

ปัญหาและอุปสรรค์ในโครงงานส่วนใหญ่จะอยู่ในขั้นตอนการเขียนโปรแกรม ซึ่งเกิดจากการทดลองใช้โปรแกรมควบคุมกล้อง ซึ่งบางครั้งทำให้คอมพิวเตอร์ไม่สามารถทำงานต่อไปได้ ต้องทำการ Reset เครื่องใหม่ เมื่อทำการทดลองเขียนโปรแกรมใช้งานกล้องจนชำนาญก็เกิดความเข้าใจสามารถกำจัดปัญหานี้ไปได้และปัญหาสำคัญอีกอย่างหนึ่งคือ ในการบีบข้อมูลภาพที่มีข้อมูลมากการทำงานของโปรแกรมจะช้ามาก แต่อย่างไงก็ตามจุดประสงค์ของโครงงานเพื่อศึกษาการทำงานเบื้องต้นของการลดข้อมูลภาพ ถ้าจะให้สามารถใช้งานจริงควรมีการออกแบบฮาร์ดแวร์เพราะสามารถทำให้การทำงานเร็วกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. Image Processing theory algorithms & architecture,Maher.A.SidAhhmed McGraw-Hill International ,1992

2. Practical digital video with programming examples in C ,Phillip E. Mattisonn,John Wiley & Sons,Inc., 1994

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอกราบขอบพระคุณ คุณพ่อ และคุณแม่ ที่ให้การสนับสนุนด้านการศึกษาคอยดูแลเอาใจใส่และห่วงใยตลอดมาจนการทำงานครั้งนี้สำเร็จลุลวงไปได้ด้วยดี

ขอขอบพระคุณ รศ.ดร.มนัส สังวรศิลป์ อาจารย์ที่ปรึกษาของข้าพเจ้าและอาจารย์หลายๆ ท่านที่ให้ความอนุเคราะห์ทั้งทางด้านความรู้และอุปกรณ์ในการทำงาน รวมทั้งให้คำปรึกษาและแนะนำแนวทางการทำโครงงานวิทยานิพนธ์นี้

ขอขอบคุณพี่ๆ และเพื่อนๆ ทุกคนที่คอยให้กำลังใจและความช่วยเหลือ ในการทำงานครั้งนี้มาโดยตลอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Unit Video;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, extctrls,
  AviCapH;


type
 TAviCap = class(TCustomPanel)
 private
  { Private Declarations }
  RequestMicroSecPerFrame:dword;
  MakeUserHitOKToCapture:LongBool;
  PercentDropForError:Byte;
  Yield:bool;
  IndexSize:dword;
  ChunkGranularity:dword;
  UsingDOSMemory:Bool;
  NumVideoRequested:dword;
  CaptureAudio:Bool;
  NumAudioRequested:Byte;
  KeyAbort:Word;
  AbortLeftMouse:bool;
  AbortRightMouse:bool;
  LimitEnabled:bool;
  TimeLimit:Byte;
  MCIControl:bool;
  StepMCIDevice:bool;
  MCIStartTime:dword;
  MCIStopTime:dword;
  StepCaptureAt2x:bool;
  StepCaptureAverageFrames:Byte;
  AudioBufferSize:dword;
  DisableWriteCache:bool;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    StreamMaster:Word;
    FCapWndHandle: THandle;
    FConnected: boolean;
    FDriverDescripton: TStringList;
    FAviFileNAme:String;
    FOverlay:boolean;
    FScale:boolean;
    capturing:boolean;
    FCaptureParms:TCaptureParms;
    dummy:Boolean;
    protected
    { Protected Declarations }
    procedure Connect(b:Boolean);
    procedure SetAviFileName(S:String);
    procedure SetOverlay(b:boolean);
    procedure SetScale(b:boolean);
    procedure Disconnect;
    Procedure DlgSource(B:Boolean);
    Procedure DlgFormat(B:Boolean);
    Procedure DlgCompression(B:Boolean);
    Procedure DlgDisplay(B:Boolean);
  public
    { Public Declarations }
    DC:HDC;
    constructor Create(AOwner: TComponent); override;
    destructor Destroy; override;
    procedure paint;override;
    procedure StartCapture;
    procedure StopCApture;
    property cap_DriverDescripton: TStringList read FDriverDescripton;
    property cap_WndHandle: THandle read FCapWndHandle;
  published
    property cap_Connected: boolean read FConnected write connect;
    property cap_DlgVideoCompression: boolean read dummy write DlgCompression;
    property cap_DlgVideoSourece: boolean read dummy write DlgSource;
    property cap_DlgVideoDisplay: boolean read dummy write DlgDisplay;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
property cap_DlgVideoFormat: boolean read dummy write DlgFormat;
property cap_AviFileName:String read FAviFileNAme write SetAviFileNAme;
property cap_Overlay:Boolean read FOverlay write SetOverlay;
property cap_Scale:Boolean read FScale write SetScale;
property cap_OptRequestMicroSecPerFrame:DWORD read RequestMicroSecPerFrame
    write RequestMicroSecPerFrame;
property cap_OptMakeUserHitOKToCapture:Bool read MakeUserHitOKToCapture
    write MakeUserHitOKToCapture;
property cap_OptPercentDropForError:Byte read PercentDropForError
    write PercentDropForError;
property cap_OptYield:Bool read Yield
    write Yield;
property cap_OptIndexSize:dword read IndexSize
    write IndexSize;
property cap_OptChunkGranularity:dword read  ChunkGranularity
    write ChunkGranularity;
property cap_OptUsingDOSMemory:Bool read UsingDOSMemory
    write UsingDOSMemory;
property cap_OptNumVideoRequested:dword read NumVideoRequested
    write NumVideoRequested;
property cap_OptCaptureAudio:LongBool read CaptureAudio
    write CaptureAudio;
property cap_OptNumAudioRequested:Byte read NumAudioRequested
    write NumAudioRequested;
property cap_OptKeyAbort:Word read KeyAbort
    write KeyAbort;
property cap_OptAbortLeftMouse:Bool read AbortLeftMouse
    write AbortLeftMouse;
property cap_OptAbortRightMouse:Bool read AbortRightMouse
    write AbortRightMouse;
property cap_OptLimitEnabled:Bool read LimitEnabled
    write LimitEnabled;
property cap_OptTimeLimit:Byte read TimeLimit
    write TimeLimit;
property cap_OptMCIControl:Bool read MCIControl
    write MCIControl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    property cap_OptStepMCIDevice:Bool read StepMCIDevice
        write StepMCIDevice;
    property cap_OptMCIStartTime:dword read MCIStartTime
        write MCIStartTime;
    property cap_OptMCIStopTime:dword read MCIStopTime
        write MCIStopTime;
    property cap_OptStepCaptureAt2x:Bool read StepCaptureAt2x
        write StepCaptureAt2x;
    property cap_OptStepCaptureAverageFrames:Byte read StepCaptureAverageFrames
        write StepCaptureAverageFrames;
    property cap_OptAudioBufferSize:dword read AudioBufferSize
        write AudioBufferSize;
    property cap_OptDisableWriteCache:Bool read DisableWriteCache
        write DisableWriteCache;
    property cap_OptAVStreamMaster:Word read StreamMaster
        write StreamMaster;
  end;

procedure Register;

implementation
{$R *.DCR}

constructor TAviCap.Create(AOwner: TComponent);
var
a1,a2:ARRAY [0..127] of char;
b:Byte;
begin
 inherited Create(AOwner);
 parent:=AOwner as TWinCOntrol;
 capturing:=false;
 width:=260;
 height:=180;
 FDriverDescripton:=TStringList.Create;
 for b :=0  to 8 do
 if capGetDriverDescriptionA(b,a1,128,a2,128)then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    FDriverDescripton.Add(a1) else break;
    FCapWndHandle:=capCreateCaptureWindowA('',WS_VISIBLE+WS_CHILD+WS_BORDER,
                          0,0,0,0,handle,0);
    FAviFileNAMe:='C:\Capture.avi';
   capCaptureGetSetup (FCapWndHandle,WParam(SizeOf(TCaptureParms)),
          LParam(@(FCaptureParms)));
    with FCaptureParms do
    begin
       RequestMicroSecPerFrame:=dwRequestMicroSecPerFrame;
            MakeUserHitOKToCapture:=fMakeUserHitOKToCapture;
            PercentDropForError:=wPercentDropForError;
            Yield:=fYield;
            IndexSize:=dwIndexSize;
            ChunkGranularity:=wChunkGranularity;
            UsingDOSMemory:=fUsingDOSMemory;
            NumVideoRequested:=wNumVideoRequested;
            CaptureAudio:=fCaptureAudio;
            NumAudioRequested:=wNumAudioRequested;
            KeyAbort:=vKeyAbort;
            AbortLeftMouse:=fAbortLeftMouse;
            AbortRightMouse:=fAbortRightMouse;
            LimitEnabled:=fLimitEnabled;
      TimeLimit:=wTimeLimit;
            MCIControl:=fMCIControl;
            StepMCIDevice:=fStepMCIDevice;
            MCIStartTime:=dwMCIStartTime;
      MCIStopTime:=dwMCIStopTime;
            StepCaptureAt2x:=fStepCaptureAt2x;
            StepCaptureAverageFrames:=wStepCaptureAverageFrames;
            AudioBufferSize:=dwAudioBufferSize;
            DisableWriteCache:=fDisableWriteCache;
            StreamMaster:=AVStreamMaster;
  end;
      FOverlay:=True;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
destructor TAviCap.Destroy;
begin
 if FConnected then disconnect;
 if assigned(FDriverDescripton) then FDriverDescripton.free;
 DeleteDC(DC);
 CloseHandle(FCapWndHandle);
 inherited Destroy;
end;

procedure TAviCap.Connect(b:Boolean);
begin
if b then
 begin
 TextOut(dc,10,10,'Connecting',10);
 fconnected:=capDriverConnect (FCapWndHandle,  0);
 SetOverlay(FOverlay);
 end
 else disconnect;
end;

procedure TAviCap.Disconnect;
var
B:boolean;
begin
b:=FOverlay;
SetOverlay(False);
capDriverDisconnect(FCapWndHandle);
fconnected:=False;
SetOverlay(b);
end;

procedure TAviCap.Paint;
begin
 inherited Paint;
 SetWindowPos(FCapWndHandle,HWND_TOP,0,0,width,height,SWP_SHOWWINDOW);
 if dc=0 then dc:=Getdc(FCapWndHandle);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If csDesigning in ComponentState then TextOut(dc,10,10,PChar(NAme),Length(NAme));
  end;

   procedure TAviCap.SetAviFileName(S:String);
    begin
    capFileSetCaptureFile (FCapWndHandle, LParam(PChar(s)));
    FAVIFileNAme:=s;
    end;

    procedure TAviCap.SetOverlay(b:boolean);
   begin
   capOverlay (FCapWndHandle, WPARAM(b));
    FOverlay:=b;
   end;

    procedure TAviCap.SetScale(b:boolean);
  begin
   capPreviewScale (FCapWndHandle, WPARAM(b));
   FScale:=b;
  end;

procedure TAviCAp.StartCapture;
var
b:Boolean;
begin
 SetAviFileName(FAviFileNAme);
 b:=FOverlay;
SetOverlay(false);
capCaptureSetSetup (FCapWndHandle,WParam(SizeOf(FCaptureParms)),
        LParam(@(FCaptureParms)));
capPreview (FCapWndHandle,  WPARAM(True));
capCaptureSequence (FCapWndHandle);
setOverlay(b);
end;

procedure TAviCAp.StopCapture;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  begin
    capCaptureStop (FCapWndHandle);
  end;

  Procedure TAviCap.DlgSource(B:Boolean);
  begin
    capDlgVideoSource (FCapWndHandle);
  end;

  Procedure TAviCAp.DlgFormat(B:Boolean);
  begin
    capDlgVideoFormat (FCapWndHandle);
  end;

  Procedure TAviCap.DlgCompression(B:Boolean);
  begin
    capDlgVideoCompression (FCapWndHandle);
  end;

  Procedure TAviCap.DlgDisplay(B:Boolean);
  begin
    capDlgVideoDisplay (FCapWndHandle);
  end;

  procedure Register;
  begin
   RegisterComponents('Free', [TAviCap]);
  end;

  end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit Videoh;

interface

uses windows,messages;

const
avicap32='avicap32.dll';

WM_CAP_START=            WM_USER;

// start of unicode messages
WM_CAP_UNICODE_START=      WM_USER+100;

WM_CAP_GET_CAPSTREAMPTR=     (WM_CAP_START+ 1);

WM_CAP_SET_CALLBACK_ERROR=    (WM_CAP_START+ 2) ;
WM_CAP_SET_CALLBACK_STATUS=   (WM_CAP_START+ 3);

WM_CAP_SET_CALLBACK_YIELD=    (WM_CAP_START+ 4);
WM_CAP_SET_CALLBACK_FRAME=    (WM_CAP_START+ 5);
WM_CAP_SET_CALLBACK_VIDEOSTREAM= (WM_CAP_START+ 6);
WM_CAP_SET_CALLBACK_WAVESTREAM= (WM_CAP_START+ 7);
WM_CAP_GET_USER_DATA       =(WM_CAP_START+ 8);
WM_CAP_SET_USER_DATA       =(WM_CAP_START+ 9) ;

WM_CAP_DRIVER_CONNECT      =(WM_CAP_START+ 10);
WM_CAP_DRIVER_DISCONNECT    =(WM_CAP_START+ 11);

WM_CAP_DRIVER_GET_NAME     =(WM_CAP_START+ 12);
WM_CAP_DRIVER_GET_VERSION   =(WM_CAP_START+ 13);

WM_CAP_DRIVER_GET_CAPS     =(WM_CAP_START+ 14);

WM_CAP_FILE_SET_CAPTURE_FILE  =(WM_CAP_START+ 20);
WM_CAP_FILE_GET_CAPTURE_FILE  =(WM_CAP_START+ 21);
WM_CAP_FILE_SAVEAS        =(WM_CAP_START+ 23);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WM_CAP_FILE_SAVEDIB        =(WM_CAP_START+ 25);

// out of order to save on ifdefs
WM_CAP_FILE_ALLOCATE       =(WM_CAP_START+ 22);
WM_CAP_FILE_SET_INFOCHUNK  =(WM_CAP_START+ 24);

WM_CAP_EDIT_COPY           =(WM_CAP_START+ 30);

WM_CAP_SET_AUDIOFORMAT     =(WM_CAP_START+ 35);
WM_CAP_GET_AUDIOFORMAT     =(WM_CAP_START+ 36);

WM_CAP_DLG_VIDEOFORMAT     =(WM_CAP_START+ 41);
WM_CAP_DLG_VIDEOSOURCE     =(WM_CAP_START+ 42);
WM_CAP_DLG_VIDEODISPLAY    =(WM_CAP_START+ 43);
WM_CAP_GET_VIDEOFORMAT     =(WM_CAP_START+ 44);
WM_CAP_SET_VIDEOFORMAT     =(WM_CAP_START+ 45);
WM_CAP_DLG_VIDEOCOMPRESSION =(WM_CAP_START+ 46);

WM_CAP_SET_PREVIEW         =(WM_CAP_START+ 50);
WM_CAP_SET_OVERLAY         =(WM_CAP_START+ 51);
WM_CAP_SET_PREVIEWRATE     =(WM_CAP_START+ 52);
WM_CAP_SET_SCALE           =(WM_CAP_START+ 53);
WM_CAP_GET_STATUS          =(WM_CAP_START+ 54);
WM_CAP_SET_SCROLL          =(WM_CAP_START+ 55);

WM_CAP_GRAB_FRAME          =(WM_CAP_START+ 60);
WM_CAP_GRAB_FRAME_NOSTOP   =(WM_CAP_START+ 61);

WM_CAP_SEQUENCE            =(WM_CAP_START+ 62);
WM_CAP_SEQUENCE_NOFILE     =(WM_CAP_START+ 63);
WM_CAP_SET_SEQUENCE_SETUP  =(WM_CAP_START+ 64);
WM_CAP_GET_SEQUENCE_SETUP  =(WM_CAP_START+ 65);

WM_CAP_SET_MCI_DEVICE      =(WM_CAP_START+ 66);
WM_CAP_GET_MCI_DEVICE      =(WM_CAP_START+ 67);

WM_CAP_STOP                =(WM_CAP_START+ 68);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WM_CAP_ABORT              =(WM_CAP_START+ 69);

WM_CAP_SINGLE_FRAME_OPEN     =(WM_CAP_START+ 70);
WM_CAP_SINGLE_FRAME_CLOSE    =(WM_CAP_START+ 71);
WM_CAP_SINGLE_FRAME          =(WM_CAP_START+ 72);

WM_CAP_PAL_OPEN           =(WM_CAP_START+ 80);
WM_CAP_PAL_SAVE           =(WM_CAP_START+ 81);

WM_CAP_PAL_PASTE          =(WM_CAP_START+ 82);
WM_CAP_PAL_AUTOCREATE     =(WM_CAP_START+ 83);
WM_CAP_PAL_MANUALCREATE   =(WM_CAP_START+ 84);

WM_CAP_SET_CALLBACK_CAPCONTROL =(WM_CAP_START+ 85);

WM_CAP_END                =WM_CAP_SET_CALLBACK_CAPCONTROL;


function capCreateCaptureWindowA (
          lpszWindowName:PChar;
          dwStyle:DWORD;
          x,y,nWidth,nHeight:Integer;
          hwndParent:HWnd;nID:Integer):hwnd;stdcall; external avicap32;

function capGetDriverDescriptionA (
     wDriverIndex:UINT;
          lpszName:PChar;cbName:Integer;
          lpszVer:PChar;cbVer:Integer):Boolean;stdcall; external avicap32;

procedure capSetCallbackOnError(hCapWnd:hwnd; fpProc:Lparam);
procedure capSetCallbackOnStatus(hCapWnd:hwnd; fpProc:LParam);
procedure capSetCallbackOnYield(hCapWnd:hwnd; fpProc:LParam);
procedure capSetCallbackOnFrame(hCapWnd:hwnd; fpProc:LParam);
procedure capSetCallbackOnVideoStream(hCapWnd:hwnd; fpProc:LParam);
procedure capSetCallbackOnWaveStream(hCapWnd:hwnd; fpProc:LParam);
procedure capSetCallbackOnCapControl(hCapWnd:hwnd; fpProc:LParam);
procedure capSetUserData(hCapWnd:hwnd;lUser:LParam);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure capGetUserData(hCapWnd:hwnd);
function capDriverConnect(hCapWnd:hwnd; i:wParam):Boolean;
procedure capDriverDisconnect(hCapWnd:hwnd);
procedure capDriverGetName(hCapWnd:hwnd; szName:wParam; wSize:LParam);
procedure capDriverGetVersion(hCapWnd:hwnd; szVer:wParam; wSize:LParam);
procedure capDriverGetCaps(hCapWnd:hwnd; s:wParam; wSize:LParam);
procedure capFileSetCaptureFile(hCapWnd:hwnd; szName :LPARAM);
procedure capFileGetCaptureFile(hCapWnd:hwnd; szName :wPARAM; wSize :lPARAM);
procedure capFileAlloc(hCapWnd:hwnd; dwSize :LPARAM);
procedure capFileSaveAs(hCapWnd:hwnd; szName :LPARAM);
procedure capFileSetInfoChunk(hCapWnd:hwnd; lpInfoChunk :LPARAM);
procedure capFileSaveDIB(hCapWnd:hwnd; szName :LPARAM);
procedure capEditCopy(hCapWnd:hwnd);
procedure capSetAudioFormat(hCapWnd:hwnd; s :WPARAM; wSize :LPARAM);
procedure capGetAudioFormat(hCapWnd:hwnd; s :WPARAM; wSize :LPARAM);
procedure capGetAudioFormatSize(hCapWnd:hwnd);
procedure capDlgVideoFormat(hCapWnd:hwnd);
procedure capDlgVideoSource(hCapWnd:hwnd);
procedure capDlgVideoDisplay(hCapWnd:hwnd);
procedure capDlgVideoCompression(hCapWnd:hwnd);
procedure capGetVideoFormat(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
procedure capGetVideoFormatSize(hCapWnd:hwnd);
procedure capSetVideoFormat(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
procedure capPreview(hCapWnd:hwnd; f :WPARAM);
procedure capPreviewRate(hCapWnd:hwnd; wMS :WPARAM);
procedure capOverlay(hCapWnd:hwnd; f :WPARAM);
procedure capPreviewScale(hCapWnd:hwnd; f :WPARAM);
procedure capGetStatus(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
procedure capSetScrollPos(hCapWnd:hwnd; lpP :LPARAM);
procedure capGrabFrame(hCapWnd:hwnd);
procedure capGrabFrameNoStop(hCapWnd:hwnd);
procedure capCaptureSequence(hCapWnd:hwnd);
procedure capCaptureSequenceNoFile(hCapWnd:hwnd);
procedure capCaptureStop(hCapWnd:hwnd);
procedure capCaptureAbort(hCapWnd:hwnd);
procedure capCaptureSingleFrameOpen(hCapWnd:hwnd);
procedure capCaptureSingleFrameClose(hCapWnd:hwnd);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure capCaptureSingleFrame(hCapWnd:hwnd);
    procedure capCaptureGetSetup(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
    procedure capCaptureSetSetup(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
    procedure capSetMCIDeviceName(hCapWnd:hwnd; szName:LPARAM);
    procedure capGetMCIDeviceName(hCapWnd:hwnd; szName:LPARAM; wSize:WPARAM);
    procedure capPaletteOpen(hCapWnd:hwnd; szName:LPARAM);
    procedure capPaletteSave(hCapWnd:hwnd; szName:LPARAM);
    procedure capPalettePaste(hCapWnd:hwnd);
    procedure capPaletteAuto(hCapWnd:hwnd; iFrames:WPARAM; iColors :LPARAM);
    procedure capPaletteManual(hCapWnd:hwnd; fGrab :WPARAM; iColors :LPARAM);


    // ------------------------------------------------------
    // Structures
    // ------------------------------------------------------

    type
    TCapDriverCaps =Record
            wDeviceIndex:UINT;
            fHasOverlay:BOOL;
            fHasDlgVideoSource:BOOL;
            fHasDlgVideoFormat:BOOL;
            fHasDlgVideoDisplay:BOOL;
            fCaptureInitialized:BOOL;
            fDriverSuppliesPalettes:BOOL;

          hVideoIn:THANDLE;
          hVideoOut:THANDLE;
          hVideoExtIn:THANDLE;
          hVideoExtOut:THANDLE;
end;


type
TCapStatus = record
          uiImageWidth:UINT ;
          uiImageHeight:UINT ;
          fLiveWindow:BOOL ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fOverlayWindow:BOOL ;
        fScale:BOOL ;
        ptScroll:TPOINT ;
        fUsingDefaultPalette:BOOL ;
        fAudioHardware:BOOL ;
        fCapFileExists:BOOL ;
        dwCurrentVideoFrame:DWORD ;
        dwCurrentVideoFramesDropped:DWORD ;
        dwCurrentWaveSamples:DWORD ;
        dwCurrentTimeElapsedMS:DWORD ;
        hPalCurrent:HPALETTE ;
        fCapturingNow:BOOL ;
        dwReturn:DWORD ;
        wNumVideoAllocated:UINT ;
        wNumAudioAllocated:UINT ;
end;

type
TCaptureParms = Record
        dwRequestMicroSecPerFrame:DWORD ;
        fMakeUserHitOKToCapture:BOOL ;
        wPercentDropForError:UINT ;
        fYield:BOOL ;
        dwIndexSize:DWORD ;
        wChunkGranularity:UINT ;
        fUsingDOSMemory:BOOL ;
        wNumVideoRequested:UINT ;
        fCaptureAudio:BOOL ;
        wNumAudioRequested:UINT ;
        vKeyAbort:UINT ;
        fAbortLeftMouse:BOOL ;
        fAbortRightMouse:BOOL ;
        fLimitEnabled:BOOL ;
        wTimeLimit:UINT ;
        fMCIControl:BOOL ;
        fStepMCIDevice:BOOL ;
        dwMCIStartTime:DWORD ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        dwMCIStopTime:DWORD ;
        fStepCaptureAt2x:BOOL ;
        wStepCaptureAverageFrames:UINT ;
        dwAudioBufferSize:DWORD ;
        fDisableWriteCache:BOOL ;
        AVStreamMaster:DWORD;

  end;

  type
  TByteRec = record
   Lo, Hi: Byte;
  end;

  TWordRec = record
   Low, High: TByteRec;
  end;

 Type
 FOURCC=TWordRec;

 function mmioFOURCC(const c0,c1,c2,c3:Char):FOURCC;

 type
 TCapInfoChunk = record
      fccInfoID:FOURCC ;
      lpData:Pointer ;
      cbData:LONGINT ;
end;


const
 CONTROLCALLBACK_PREROLL =     1; //* Waiting to start capture */
 CONTROLCALLBACK_CAPTURING=    2; //* Now capturing */


implementation
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function mmioFOURCC(const c0,c1,c2,c3:Char):FOURCC;
begin
result.High.Hi:=Byte(c0);
result.High.Lo:=Byte(c1);
result.Low.Hi:=Byte(c2);
result.Low.Lo:=Byte(c3);
end;

procedure capSetCallbackOnError(hCapWnd:hwnd; fpProc:Lparam);
begin
SendMessage(hCapWnd, WM_CAP_SET_CALLBACK_ERROR, 0, fpProc)
end;

procedure capSetCallbackOnStatus(hCapWnd:hwnd; fpProc:LParam);
begin
SendMessage(hCapWnd, WM_CAP_SET_CALLBACK_Status, 0, fpProc)
end;

procedure capSetCallbackOnYield(hCapWnd:hwnd; fpProc:LParam);
begin
SendMessage(hCapWnd, WM_CAP_SET_CALLBACK_YIELD, 0, fpProc)
end;

procedure capSetCallbackOnFrame(hCapWnd:hwnd; fpProc:LParam);
begin
SendMessage(hCapWnd, WM_CAP_SET_CALLBACK_FRAME, 0, fpProc)
end;

procedure capSetCallbackOnVideoStream(hCapWnd:hwnd; fpProc:LParam);
begin
SendMessage(hCapWnd, WM_CAP_SET_CALLBACK_VideoSTREAM, 0, fpProc)
end;

procedure capSetCallbackOnWaveStream(hCapWnd:hwnd; fpProc:LParam);
begin
SendMessage(hCapWnd, WM_CAP_SET_CALLBACK_WAVESTREAM , 0, fpProc)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;

procedure capSetCallbackOnCapControl(hCapWnd:hwnd; fpProc:LParam);
begin
SendMessage(hCapWnd, WM_CAP_SET_CALLBACK_CapControl , 0, fpProc)
end;

procedure capGetUserData(hCapWnd:hwnd);
begin
SendMessage(hCapWnd, WM_CAP_GET_USER_DATA  , 0, 0)
end;

procedure capSetUserData(hCapWnd:hwnd;lUser:LParam);
begin
SendMessage(hCapWnd, WM_CAP_SET_USER_DATA  , 0, lUser)
end;

function capDriverConnect(hCapWnd:hwnd; i:wParam):Boolean;
begin
result:=SendMessage(hCapWnd, WM_CAP_DRIVER_CONNECT ,i  , 0)<>0;
end;

procedure capDriverDisconnect(hCapWnd:hwnd);
begin
SendMessage(hCapWnd, WM_CAP_DRIVER_DISCONNECT ,0  , 0)
end;

procedure capDriverGetName(hCapWnd:hwnd; szName:wParam; wSize:LParam);
begin
SendMessage(hCapWnd, WM_CAP_DRIVER_Get_Name ,szName  , wSize)
end;

procedure capDriverGetVersion(hCapWnd:hwnd; szVer:wParam; wSize:LParam);
begin
SendMessage(hCapWnd, WM_CAP_DRIVER_Get_VERSION ,szVer  , wSize)
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure capDriverGetCaps(hCapWnd:hwnd; s:wParam; wSize:LParam);
begin
SendMessage(hCapWnd, WM_CAP_DRIVER_Get_CAPS ,s  , wSize)
end;

procedure capFileSetCaptureFile(hCapWnd:hwnd; szName :LPARAM);
begin
SendMessage(hCapWnd, WM_CAP_FILE_SET_CAPTURE_FILE  ,0  , szName)
end;

procedure capFileGetCaptureFile(hCapWnd:hwnd; szName :wPARAM; wSize :lPARAM);
begin
SendMessage(hCapWnd, WM_CAP_FILE_GET_CAPTURE_FILE , szName,wSize)
end;

procedure capFileAlloc(hCapWnd:hwnd; dwSize :LPARAM);
begin
SendMessage(hCapWnd, WM_CAP_FILE_ALLOCATE , 0,dwSize)
end;

procedure capFileSaveAs(hCapWnd:hwnd; szName :LPARAM);
begin
SendMessage(hCapWnd, WM_CAP_FILE_SAVEAS , 0,szName)
end;

procedure capFileSetInfoChunk(hCapWnd:hwnd; lpInfoChunk :LPARAM);
begin
SendMessage(hCapWnd, WM_CAP_FILE_SET_INFOCHUNK , 0,lpInfoChunk)
end;

procedure capFileSaveDIB(hCapWnd:hwnd; szName :LPARAM);
begin
SendMessage(hCapWnd, WM_CAP_FILE_SAVEDIB , 0,szName)
end;

procedure capEditCopy(hCapWnd:hwnd);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SendMessage(hCapWnd, WM_CAP_EDIT_COPY , 0,0)
end;

procedure capSetAudioFormat(hCapWnd:hwnd; s :WPARAM; wSize :LPARAM);
begin
SendMessage(hCapWnd, WM_CAP_SET_AUDIOFORMAT , s,wSize)
end;

procedure capGetAudioFormat(hCapWnd:hwnd; s :WPARAM; wSize :LPARAM);
begin
SendMessage(hCapWnd, WM_CAP_GET_AUDIOFORMAT , s,wSize)
end;

procedure capGetAudioFormatSize(hCapWnd:hwnd);
begin
SendMessage(hCapWnd, WM_CAP_GET_AUDIOFORMAT , 0,0)
end;

procedure capDlgVideoFormat(hCapWnd:hwnd);
begin
SendMessage(hCapWnd, WM_CAP_DLG_VIDEOFORMAT , 0,0)
end;

procedure capDlgVideoSource(hCapWnd:hwnd);
begin
SendMessage(hCapWnd, WM_CAP_DLG_VIDEOSOURCE , 0,0)
end;

procedure capDlgVideoDisplay(hCapWnd:hwnd);
begin
SendMessage(hCapWnd, WM_CAP_DLG_VIDEODISPLAY , 0,0)
end;

procedure capDlgVideoCompression(hCapWnd:hwnd);
begin
SendMessage(hCapWnd, WM_CAP_DLG_VIDEOCOMPRESSION , 0,0)
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure capGetVideoFormat(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
begin
SendMessage(hCapWnd, WM_CAP_GET_VIDEOFORMAT , s,wSize)
end;

procedure capGetVideoFormatSize(hCapWnd:hwnd);
begin
SendMessage(hCapWnd, WM_CAP_GET_VIDEOFORMAT , 0,0)
end;

procedure capSetVideoFormat(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
begin
SendMessage(hCapWnd, WM_CAP_GET_VIDEOFORMAT , 0,0)
end;

procedure capPreview(hCapWnd:hwnd; f :WPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_SET_PREVIEW , f,0)
end;

procedure capPreviewRate(hCapWnd:hwnd; wMS :WPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_SET_PREVIEWRATE  , wMS,0)
end;

procedure capOverlay(hCapWnd:hwnd; f :WPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_SET_OVERLAY  , f,0)
end;

procedure capPreviewScale(hCapWnd:hwnd; f :WPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_SET_SCALE  , f,0)
end;

procedure capGetStatus(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
SendMessage(hCapWnd,  WM_CAP_GET_STATUS  , s,wSize)
end;

procedure capSetScrollPos(hCapWnd:hwnd; lpP :LPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_SET_SCROLL  , 0,lpP)
end;

procedure capGrabFrame(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_GRAB_FRAME  , 0,0)
end;

procedure capGrabFrameNoStop(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_GRAB_FRAME_NOSTOP  , 0,0)
end;

procedure capCaptureSequence(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_SEQUENCE  , 0,0)
end;

procedure capCaptureSequenceNoFile(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_SEQUENCE_NOFILE  , 0,0)
end;

procedure capCaptureStop(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_Stop  , 0,0)
end;

procedure capCaptureAbort(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_ABORT  , 0,0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;

procedure capCaptureSingleFrameOpen(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_SINGLE_FRAME_OPEN   , 0,0)
end;

procedure capCaptureSingleFrameClose(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_SINGLE_FRAME_CLOSE   , 0,0)
end;

procedure capCaptureSingleFrame(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_SINGLE_FRAME   , 0,0)
end;

procedure capCaptureGetSetup(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_GET_SEQUENCE_SETUP    , s,wSize)
end;

procedure capCaptureSetSetup(hCapWnd:hwnd; s:WPARAM; wSize:LPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_SET_SEQUENCE_SETUP    , s,wSize)
end;

procedure capSetMCIDeviceName(hCapWnd:hwnd; szName:LPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_SET_MCI_DEVICE    , 0,szName)
end;

procedure capGetMCIDeviceName(hCapWnd:hwnd; szName:LPARAM; wSize:WPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_GET_MCI_DEVICE    , wSize,szName)
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure capPaletteOpen(hCapWnd:hwnd; szName:LPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_PAL_OPEN , 0,szName)
end;

procedure capPaletteSave(hCapWnd:hwnd; szName:LPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_PAL_SAVE , 0,szName)
end;

procedure capPalettePaste(hCapWnd:hwnd);
begin
SendMessage(hCapWnd,  WM_CAP_PAL_PASTE , 0,0)
end;

procedure capPaletteAuto(hCapWnd:hwnd; iFrames:WPARAM; iColors :LPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_PAL_AUTOCREATE  , iFrames,iColors)
end;

procedure capPaletteManual(hCapWnd:hwnd; fGrab :WPARAM; iColors :LPARAM);
begin
SendMessage(hCapWnd,  WM_CAP_PAL_MANUALCREATE , fGrab,iColors)
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit Code;

interface

uses
 Wintypes, Winprocs, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
 StdCtrls, Compress, ExtCtrls, Compctrl;

type
 TEncode = class(TForm)
  Compbmp: TCompress;
  OpenDialog1: TOpenDialog;
  OpenDialog2: TOpenDialog;
  SaveDialog1: TSaveDialog;
  GroupBox1: TGroupBox;
  Image1: TImage;
  SaveDialog2: TSaveDialog;
  OpenDialog3: TOpenDialog;
  GroupBox2: TGroupBox;
  Button3: TButton;
  GroupBox3: TGroupBox;
  Button2: TButton;
  GroupBox4: TGroupBox;
  Button1: TButton;
  Button4: TButton;
  GroupBox5: TGroupBox;
  Button5: TButton;
  Button6: TButton;
  Label1: TLabel;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Compbmphuf(infilename, outfilename: string);
  procedure CompbmpRun(infilename, outfilename: string);
  procedure LoadABitmap(infilename,outfilename:string);
  procedure FormCreate(Sender: TObject);
  procedure Button3Click(Sender: TObject);
  procedure Button4Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  finally
   infile.free;
   outfile.free;
  end;
 end;

procedure TEncode.LoadABitmap(infilename,outfilename:string{bm: TBitmap;Resname: string});
var infile,outfile : Tfilestream;
begin
  infile := TfileStream.Create(infilename,fmOpenRead);
  outfile := Tfilestream.Create(outfilename,fmcreate);
  try
   compbmp.Expand(outfile,infile);
  finally
   infile.Free;
   outfile.free;
  end;
end;

procedure TEncode.Button1Click(Sender: TObject);
begin
  Savedialog1.FileName:='c:\*.huf';
  if savedialog1.Execute then
   begin
 CompbmpHuf(opendialog1.filename,savedialog1.filename);
 ShowMessage(savedialog1.filename+' File HufCode OK!');
 end;
 end;

procedure TEncode.Button2Click(Sender: TObject);
begin
 close;
 AAA.Enabled:=true;

 end;

procedure TEncode.FormCreate(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin

    button1.enabled:=false;
    button5.enabled:=false;
    end;

procedure TEncode.Button3Click(Sender: TObject);
begin
    opendialog1.filename:='c:\*.bmp';
    if opendialog1.Execute then
    begin
     with image1 do
      begin
        picture.LoadFromFile(opendialog1.filename);
        button1.enabled:=true;
        button5.enabled:=true;
       end;
       end;
end;

procedure TEncode.Button4Click(Sender: TObject);
begin
    opendialog2.filename := 'c:\*.huf';
    if opendialog2.Execute then
    begin
     loadAbitmap(opendialog2.filename,opendialog2.filename+'.bmp');
     ShowMessage(OPendialog2.filename+' File HufDecode OK!');
  end;
  end;

procedure TEncode.Button5Click(Sender: TObject);
begin
   savedialog2.FileName:='c:\*.run';
    if savedialog2.Execute then
    begin
    compbmpRun(opendialog1.filename,savedialog2.filename);
    ShowMessage(Savedialog1.filename+' File RunCode OK!');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
end;
procedure TEncode.Button6Click(Sender: TObject);
begin
   opendialog3.FileName := 'c:\*.run';
    if opendialog3.Execute then
     begin
    loadAbitmap(opendialog3.filename,opendialog3.filename+'.bmp');
    ShowMessage(Opendialog3.filename+' RunDecode OK!');
    end;

end;

procedure TEncode.FormClose(Sender: TObject; var Action: TCloseAction);
begin
   AAA.Enabled:=true;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit data;

interface

uses
  Wintypes, Winprocs, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Compress, ExtCtrls, Compctrl, MPlayer, FileCtrl, AviCap, Menus,
  EnhAVICap,Clipbrd,Jpeg;

type
  TAAA = class(TForm)
    OpenDialog1: TOpenDialog;
    OpenDialog2: TOpenDialog;
    SaveDialog1: TSaveDialog;
    SaveDialog2: TSaveDialog;
    OpenDialog3: TOpenDialog;
    GroupBox3: TGroupBox;
    Button2: TButton;
    GroupBox4: TGroupBox;
    Button1: TButton;
    Label1: TLabel;
    MediaPlayer1: TMediaPlayer;
    GroupBox2: TGroupBox;
    Button3: TButton;
    Button7: TButton;
    Button8: TButton;
    Button9: TButton;
    Panel1: TPanel;
    GroupBox1: TGroupBox;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    SaveDialog3: TSaveDialog;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Open1: TMenuItem;
    Exit1: TMenuItem;
    GroupBox5: TGroupBox;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Edit1: TEdit;
    Label2: TLabel;
    Edit2: TEdit;
    Label3: TLabel;
    GroupBox6: TGroupBox;
    Button4: TButton;
    Button6: TButton;
    Edit3: TEdit;
    Edit4: TEdit;
    Label4: TLabel;
    Label5: TLabel;
    Setup1: TMenuItem;
    Sou1: TMenuItem;
    Format1: TMenuItem;
    Display1: TMenuItem;
    About1: TMenuItem;
    About2: TMenuItem;
    comp: Tcompress;
    GroupBox7: TGroupBox;
    Button5: TButton;
    Video: TEnhAVICap;
    Image1: TImage;
    Openbmpfile1: TMenuItem;
    AviCap1: TAviCap;
    RadioButton3: TRadioButton;
    Panel2: TPanel;
    Image2: TImage;
    BmpCompression1: TMenuItem;
    RunlengthandHuffmancode1: TMenuItem;
    BmptoJpegFromat1: TMenuItem;
    Button10: TButton;
    OpenDialog4: TOpenDialog;
    Label6: TLabel;
    Label7: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Compresshuf(infilename, outfilename: string);
    procedure CompressRun(infilename, outfilename: string);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure LoadABitmap(infilename,outfilename:string);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure FileListBox1DblClick(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure Button8Click(Sender: TObject);
    procedure MediaPlayer1Notify(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Button9Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
    procedure RadioButton1Click(Sender: TObject);
    procedure Open1Click(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure Edit4Change(Sender: TObject);
    procedure Edit3Change(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Sou1Click(Sender: TObject);
    procedure Format1Click(Sender: TObject);
    procedure Display1Click(Sender: TObject);
    procedure Openbmpfile1Click(Sender: TObject);
    procedure RadioButton3Click(Sender: TObject);
    procedure About2Click(Sender: TObject);
    procedure RunlengthandHuffmancode1Click(Sender: TObject);
    procedure BmptoJpegFromat1Click(Sender: TObject);
    procedure Button10Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      comp.Expand(outfile,infile);

      finally
      infile.Free;
      outfile.free;
     end;
   end;

   procedure TAAA.Button1Click(Sender: TObject);
   begin
    Mediaplayer1.Close;
    CompressHuf(edit1.text,edit2.text);
    ShowMessage(edit2.Text+' Compress File OK!');
    Mediaplayer1.Open;
    Mediaplayer1.Play;
    edit3.Text:=edit2.text;
    end;



  procedure TAAA.Button4Click(Sender: TObject);
  begin
     opendialog2.filename := 'c:\*.huf';
     if opendialog2.Execute then
     begin
     edit3.Text:=opendialog2.FileName;
     edit4.Enabled:=true;
    end;
    end;

  procedure TAAA.Button5Click(Sender: TObject);
  begin
    if savedialog1.Execute then
    begin

    Avicap1.cap_Connected:=false;
    video.OpenVideo(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if video.ToClipboard then
     begin
     image1.picture.Assign(clipboard);
     image1.Picture.SaveToFile(Savedialog1.FileName);
     video.CloseVideo;
     Avicap1.cap_Connected:=true;
     ShowMessage(Savedialog1.FileName+' Save bmp OK!');
     form2.Image1.Picture.LoadFromFile(Savedialog1.FileName);
     form2.OpenDialog1.FileName:=Savedialog1.FileName;
     form2.button3.Enabled:=true;
     encode.Image1.Picture.LoadFromFile(Savedialog1.filename);
     encode.OpenDialog1.FileName:=Savedialog1.FileName;
     encode.Button1.Enabled:=true;
     encode.Button5.Enabled:=true;
     end;
    end;

end;
procedure TAAA.Button6Click(Sender: TObject);
begin

  loadAbitmap(edit3.text,edit4.text);
  ShowMessage(edit4.Text+' Decode File OK!');

end;

procedure TAAA.FileListBox1DblClick(Sender: TObject);
begin
  MediaPlayer1.FileName:=opendialog1.FileName;
  MediaPlayer1.open;
  Mediaplayer1.display:=GroupBox1;
  Mediaplayer1.Play;
  button1.Enabled:=true;
  button3.Enabled:=false;
  button7.Enabled:=true;
  button8.Enabled:=true;
  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TAAA.Button3Click(Sender: TObject);
begin
   Mediaplayer1.Play;
   button3.Enabled:=false;
   button7.Enabled:=true;
   button8.Enabled:=true;
end;

procedure TAAA.Button7Click(Sender: TObject);
begin
   Mediaplayer1.Pause;
   button3.Enabled:=true;
   button7.Enabled:=false;
   button8.Enabled:=true;
end;

procedure TAAA.Button8Click(Sender: TObject);
begin
  if Radiobutton1.Checked = true  then
  begin
    Avicap1.StopCApture;
    button9.Enabled:=true;
    button8.Enabled:=false;
  end;
  if Radiobutton1.Checked = false  then
  begin
  Mediaplayer1.Stop;
  button3.Enabled:=true;
  button7.Enabled:=false;
  button8.Enabled:=false;
  end;
  end;

procedure TAAA.MediaPlayer1Notify(Sender: TObject);
begin
   button3.Enabled:=true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    button7.Enabled:=false;
    button8.Enabled:=false;
  end;

  procedure TAAA.FormActivate(Sender: TObject);
  begin
    Avicap1.cap_Connected:=true;
    open1.Enabled:=false;
    openbmpfile1.Enabled:=false;
    end;

  procedure TAAA.Button9Click(Sender: TObject);
  begin
    savedialog3.FileName:='c:\*.avi';
    if Radiobutton1.Checked = true then
    begin
    if savedialog3.Execute then
    begin
     with Avicap1  do
      begin
       button8.Enabled:=true;
       button9.Enabled:=false;
       cap_Avifilename:=Savedialog3.FileName;
       Avicap1.StartCapture;

      end;
    end;
    end;
    end;

procedure TAAA.RadioButton2Click(Sender: TObject);
begin

  Avicap1.cap_Connected:=false;
  Mediaplayer1.Close;
  panel2.Visible:=true;
  panel2.Color:=clwindowText;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    radiobutton1.Checked:=false;
    radiobutton2.Checked:=true;
    radiobutton3.Checked:=false;
    Open1.Enabled:=true;
    button5.Enabled:=false;
    edit1.Enabled:=true;
    edit3.Enabled:=true;
    button9.Enabled:=false;
    button4.Enabled:=true;
    openbmpfile1.Enabled:=false;
    button10.Enabled:=true;
    button2.Enabled:=false;
    end;

procedure TAAA.RadioButton1Click(Sender: TObject);
begin
   panel2.Visible:=false;
   Avicap1.cap_Connected:=true;
   radioButton1.Checked:=true;
   radiobutton2.Checked:=false;
   radiobutton3.Checked:=false;
   button9.Enabled:=true;
   button3.Enabled:=false;
   button7.Enabled:=false;
   button8.Enabled:=false;
   open1.Enabled:=false;
   button1.Enabled:=false;
   edit1.Enabled:=false;
   edit2.Enabled:=false;
   edit3.Enabled:=false;
   edit4.Enabled:=false;
   button4.Enabled:=false;
   button6.Enabled:=false;
   button5.Enabled:=true;
   openbmpfile1.Enabled:=false;
   button10.Enabled:=false;
   button2.Enabled:=true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;

  procedure TAAA.Open1Click(Sender: TObject);
  begin
    opendialog1.FileName:='c:\*.avi';
    if opendialog1.Execute  then
     begin
       with mediaplayer1 do
        begin
         filename:=opendialog1.FileName;
         open;
         display := Panel2;
         play;
         button7.Enabled:=true;
         button8.Enabled:=true;
         edit1.Text:=opendialog1.FileName;
       end;
    end;
  end;


  procedure TAAA.Edit2Change(Sender: TObject);
  begin
    button1.Enabled:=true;
  end;

  procedure TAAA.Edit1Change(Sender: TObject);
  begin
    edit2.Enabled:=true;
  end;

  procedure TAAA.Edit4Change(Sender: TObject);
  begin
    button6.Enabled:=true;
  end;

  procedure TAAA.Edit3Change(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  begin
    edit4.Enabled:=true;
  end;

  procedure TAAA.Button2Click(Sender: TObject);
  begin
    savedialog2.FileName:='c:\*.cps';
    if savedialog2.Execute then
    begin
     button8.Enabled:=true;
     avicap1.cap_AviFileName:=savedialog2.FileName+'.avi';
     avicap1.StartCapture;
     compresshuff(Savedialog2.FileName+'.avi',savedialog2.filename);
     ShowMessage(Savedialog2.FileName+' Compress OK!');
     end;
     end;

 procedure TAAA.Exit1Click(Sender: TObject);
 begin
    close;
 end;

 procedure TAAA.Sou1Click(Sender: TObject);
 begin
   Avicap1.cap_DlgVideoSourece:=true;
 end;

 procedure TAAA.Format1Click(Sender: TObject);
 begin
  Avicap1.cap_DlgVideoFormat:=true;
 end;

 procedure TAAA.Display1Click(Sender: TObject);
 begin
   Avicap1.cap_DlgVideoDisplay:=true;
 end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TAAA.Openbmpfile1Click(Sender: TObject);
begin
   opendialog3.FileName:='c:\*.bmp';
   if opendialog3.Execute then
    begin

     image2.Picture.LoadFromFile(opendialog3.filename);
    end;
end;

procedure TAAA.RadioButton3Click(Sender: TObject);
begin
   Avicap1.cap_Connected:=false;
   Mediaplayer1.Close;
   button3.Enabled:=false;
   panel2.Visible:=false;
   panel2.Visible:=true;
   panel2.Color:=clwindowText;
   radiobutton1.Checked:=false;
   radiobutton2.Checked:=false;
   radiobutton3.Checked:=true;
   openbmpfile1.Enabled:=true;
   open1.Enabled:=false;
   button1.Enabled:=false;
   button4.Enabled:=false;
   button6.Enabled:=false;
   button7.Enabled:=false;
   button5.Enabled:=false;
   button8.Enabled:=false;
   button9.Enabled:=false;
   button10.Enabled:=false;
   button2.Enabled:=false;
   edit1.Enabled:=false;
   edit3.Enabled:=false;
   end;

procedure TAAA.About2Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 begin
    form3.Show;
    AAA.Enabled:=false;
 end;

 procedure TAAA.RunlengthandHuffmancode1Click(Sender: TObject);
 begin
    Encode.Show;
    AAA.Enabled:=false;
 end;

 procedure TAAA.BmptoJpegFromat1Click(Sender: TObject);
 begin
    Form2.show;
    AAA.Enabled:=false;
 end;

 procedure TAAA.Button10Click(Sender: TObject);
 begin
   opendialog4.FileName:='c:\*.avi';
   if opendialog4.Execute  then
     begin
       edit1.Text:=opendialog4.FileName;
       edit2.Enabled:=true;
   end;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้