

การวิเคราะห์มัลติเรโซลูชันของข้อมูลภาพโดยใช้การแปลงเวฟเลต
MULTIRESOLUTION ANALYSIS OF IMAGE BY USING
WAVELET TRANSFORM



โดย

นายวริทธิ

สันติเดช

นางสาวอรนุช

เลิศสุวรรณกิจ

เลขที่.....
เลขทะเบียน..... 42131
วัน, เดือน, ปี..... 1.3 พ.ศ. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

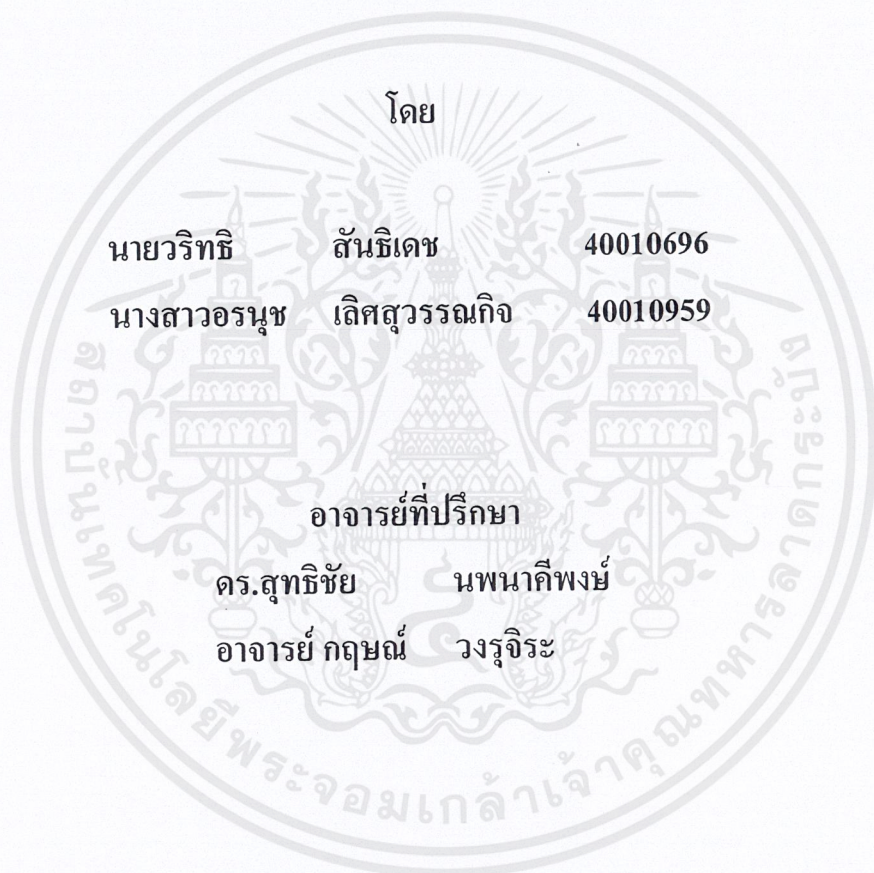
สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์มัลติเรโซลูชันของข้อมูลภาพโดยใช้การแปลงเวฟเล็ต
MULTIRESOLUTION ANALYSIS OF IMAGE BY USING
WAVELET TRANSFORM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2543

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวิเคราะห์มัลติเรโซลูชันของข้อมูลภาพโดยใช้การแปลงเวฟเล็ต

MULTIRESOLUTION ANALYSIS OF IMAGE BY USING WAVELET TRANSFORM

ผู้จัดทำ

1. นายวริทธิ สันติเดช 40010696
2. นางสาวอรนุช เลิศสุวรรณกิจ 40010959



(ดร.สุทธิชัย นพนาถิพงษ์)

อาจารย์ที่ปรึกษา



(อาจารย์ กฤษณ์ วงจริระ)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์มัลติเรโซลูชันของข้อมูลภาพโดยใช้การแปลงเวฟเลต
MULTIRESOLUTION ANALYSIS OF IMAGE BY USING
WAVELET TRANSFORM

โดย นายวริทธิ สันธิเดช 40010696
นางสาวอรนุช เลิศสุวรรณกิจ 40010959

อาจารย์ที่ปรึกษา คร.สุทธิชัย นพนาทิพงษ์
อาจารย์ กฤษณ์ วรจิวระ

บทคัดย่อ

ปัจจุบันได้มีการศึกษาวิจัยและพัฒนาเทคโนโลยีที่เกี่ยวกับการประมวลผลภาพอย่างกว้างขวางได้ มีการนำเอาหลักการและทฤษฎีต่างๆมาใช้ เพื่อที่จะทำให้วิธีประมวลผลภาพสามารถกระทำได้ง่ายขึ้น ทฤษฎีของการแปลงเวฟเลตก็เป็นอีกวิธีหนึ่งที่ได้ถูกนำมาประยุกต์ใช้งาน ที่เกี่ยวกับการประมวลผลภาพ ปริญญานิพนธ์นี้ได้ทำการศึกษาเกี่ยวกับการวิเคราะห์มัลติเรโซลูชันของภาพ เพื่อทำการส่งข้อมูล แบบความละเอียดหลายระดับ ทำให้สามารถลดข้อมูลในการส่งข้อมูลภาพได้เช่นกัน

ABSTRACT

Today there has been researching and developing rapidly in image processing, using principles and theories to assist the processing easier. Wavelet transform theory is one method that is used in image processing.

This thesis will study about the Multiresolution Analysis of image data to transfer data in progressive image method that can decrease the quality of transmitted data.

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการเบื้องต้น	2
2.1 พื้นฐานของภาพเชิงตัวเลข	2
2.1.1 โมเดลภาพ	2
2.1.2 การสุ่มและการจัดลำดับ	2
2.1.3 ปริมาณข้อมูลภาพ	3
2.2 ทฤษฎีพื้นฐานของการแปลงเวฟเล็ท	4
2.2.1 ลักษณะพื้นฐานของเวฟเล็ท	4
2.2.2 ทฤษฎีพื้นฐานในการสเกลและเลื่อนตำแหน่งของเวฟเล็ท	6
2.2.3 ความเป็นมาของการแปลงเวฟเล็ท	9
2.2.4 ความสัมพันธ์ระหว่างการแปลงเวฟเล็ทและการแปลงฟูรีเยร์ ในลักษณะของฟังก์ชันหน้าต่าง	13
2.3 การแปลงเวฟเล็ทแบบแบ่งช่วง	14
2.3.1 ปริภูมิเวกเตอร์ในการวิเคราะห์สัญญาณ	14
2.3.2 การวิเคราะห์แบบหลายระดับความละเอียดของมอร์เลต (Mallat)	15
2.3.3 การแปลงเวฟเล็ทโดยใช้หลักการของการวิเคราะห์ของพีวเดอร์เบงค์	21
2.4 การแปลงเวฟเล็ทกับข้อมูลภาพ	29
2.4.1 การวิเคราะห์การแปลงด้วยฮาร์ฟังก์ชัน	31
2.4.2 การใช้ซีโรทรี่ของสัมประสิทธิ์เวฟเล็ทเข้ารหัสข้อมูลภาพ	33
2.5 หลักการเบื้องต้นของการลดข้อมูล	36
2.6 เกณฑ์การวัดความเหมือนจริงของภาพ	40
2.6.1 อัตราบิด	42
2.6.2 อัตราการลดข้อมูล	42
บทที่ 3 การคำนวณและการสร้าง	43
3.1 ข้อมูลภาพบิตแมป (BMP)	43
3.1.1 หัวไฟล์ (BITMAP FILE HEADER)	44
3.1.2 หัวข้อมูล (BITMAPINFO)	44
3.2 การคำนวณการแปลงเวฟเล็ทกับข้อมูลภาพ	45
3.3 การกำหนดรหัสสัญลักษณ์ให้กับสัมประสิทธิ์เวฟเล็ท	51
3.4 การปรับปรุงวิธีการเข้ารหัสแบบเอ็มเบดซีโรทรี่	59
3.5 การเข้ารหัสเลขคณิต	60
3.6 การส่งภาพแบบโปรเกรสซีฟโดยใช้การแปลงเวฟเล็ท	68

	หน้า
3.7 การใช้งานโปรแกรมลดขนาดข้อมูลภาพโดยการแปลงเวฟเล็ต ที่ใช้วิธีการของเอ็มเบดซีโร้ทรีบน Vitual C++	69
บทที่ 4 การทดลองและผลการทดลอง	78
การทดลองที่ 4.1	79
การทดลองที่ 4.2	84
การทดลองที่ 4.3	85
การทดลองที่ 4.4	107
บทที่ 5 บทวิจารณ์และสรุป	109
5.1 บทวิจารณ์	109
5.2 บทสรุป	110



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

รูปที่ 2.1	เป็นตัวอย่างของเวฟเล็ดชนิดหนึ่งเรียกว่าเวฟเล็ดแบบมอเล็ด $g(t) = e^{-\frac{t^2}{64}} \cos(t)$	5
รูปที่ 2.2	ลักษณะคุณสมบัติของเวฟเล็ด 2 เงื่อนไข	5
รูปที่ 2.3	ลักษณะของเวฟเล็ดแม่ที่ถูกสเกลและเลื่อนตำแหน่ง	6
รูปที่ 2.4	แสดงลักษณะของการแตกกระจายสัญญาณและรวมกับสัญญาณของเวฟเล็ด	7
รูปที่ 2.5	แสดงลักษณะของสัญญาณที่มีหลายระดับความถี่ที่ตำแหน่งเวลาต่างกัน	7
รูปที่ 2.6	พิจารณาเวฟเล็ดในแกน 2 มิติ โดยที่ให้แกนหนึ่งเป็นแกนเวลา อีกแกนหนึ่งเป็นแกนความถี่	8
รูปที่ 2.7	แสดงลักษณะของผลของการสหสัมพันธ์(Correlation) ระหว่าง $f(t), g\left(\frac{t-b}{a}\right)$	10
รูปที่ 2.8	แสดงลักษณะความแตกต่างของสัญญาณเอาท์พุทระหว่าง การแปลงฟูริเยร์ กับการแปลงเวฟเล็ด	10
รูปที่ 2.9	แสดงลักษณะของระนาบเวลาความถี่แบบคงที่ (Constant Time-Frequency plane)	12
รูปที่ 2.10	แสดงลักษณะของระนาบเวลาเทียบกับความถี่และลักษณะการแตกกระจายสัญญาณของเวฟเล็ด	13
รูปที่ 2.11	แสดงกลุ่มของปริภูมิเวกเตอร์ที่ถูกแผ่คลุมโดยสเกลลิงฟังก์ชัน	17
รูปที่ 2.12	แสดงปริภูมิเวกเตอร์ของสเกลลิงฟังก์ชันและเวฟเล็ดฟังก์ชัน	18
รูปที่ 2.13	แสดงการลดอัตราการซีกตัวอย่างลงครึ่งหนึ่ง	24
รูปที่ 2.14	แสดงการวิเคราะห์สัญญาณด้วย two band	24
รูปที่ 2.15	แสดงการแตกกระจายสัญญาณจากระดับ $j+1$ ไปยังระดับ $j-1$	25
รูปที่ 2.16	แสดงการแตกกระจายสัญญาณใน V_j ไปสู่สเกล $j=0$	25
รูปที่ 2.17	แสดงแถบความถี่จากการผ่านการกรองของ $h_0(n)$ และ $h_1(n)$	26
รูปที่ 2.18	แสดงการซีกค่าการสุ่มตัวอย่างขึ้นสองเท่า	28
รูปที่ 2.19	แสดงการสังเคราะห์ด้วย two band	29
รูปที่ 2.20	แสดงการสังเคราะห์สัญญาณจากสเกล $j-1$ ไปสู่ระดับ j	29
รูปที่ 2.21	แสดงการแปลงเวฟเล็ดของภาพ 1 ครั้ง	30
รูปที่ 2.22	แสดงแผนผังการทำแปลงเวฟเล็ดกับภาพ 2 มิติ 3 ครั้ง	31
รูปที่ 2.23	แสดงรูปคลื่นจากสมการและเลื่อนตำแหน่ง	32
รูปที่ 2.24	แสดงลักษณะผลรวมของสเกลลิงฟังก์ชันพื้นฐาน	33
รูปที่ 2.25	แสดงลักษณะการสเกลและลักษณะในการสืบทอดของสัมประสิทธิ์ของเวฟเล็ด	34

	หน้า
รูปที่ 2.26	แสดงลักษณะการสแกนแบบตัวชี้(Z) 35
รูปที่ 2.27	แสดงการควอนไทซ์ค่าสัมประสิทธิ์และกำหนดสัญลักษณ์ 36
รูปที่ 2.28	แสดงการแบ่งช่องของขอบเขตปัจจุบันบนพื้นฐาน ของความน่าจะเป็นของสัญลักษณ์ a_i ที่ปรากฏขึ้น 37
รูปที่ 2.29	แสดงการแบ่งขอบเขตตามข้อมูลที่เพิ่มขึ้น 39
รูปที่ 3.1	แสดงขั้นตอนการบีบอัดข้อมูลภาพโดยใช้การแปลงเวฟเล็ต 45
รูปที่ 3.2	แสดงสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ต 51
รูปที่ 3.3	ผลการทำกลับของการสแกนครั้งที่ 1 56
รูปที่ 3.4	ผลการทำกลับของการสแกนครั้งที่ 2 57
รูปที่ 3.5	ผลการทำกลับของการสแกนครั้งที่ 3 57
รูปที่ 3.6	หน้าจอโคดะล็อกบ็อกซ์สำหรับโปรแกรมลดขนาดข้อมูลภาพ โดยการแปลงเวฟเล็ตที่ใช้วิธีการของเอ็มเบดซีโรทรี 69
รูปที่ 3.7	แสดงโคดะล็อกบ็อกซ์ สำหรับการใส่อินพุตภาพเพื่อทำการลดขนาดข้อมูล 70
รูปที่ 3.8	แสดงภาพเมื่อเลือกภาพจาก โคดะล็อกบ็อกซ์ 70
รูปที่ 3.9	แสดงภาพจากการแปลงเวฟเล็ตเมื่อเลือกระดับการแปลงเป็นสอง 71
รูปที่ 3.10	แสดงภาพจากการแปลงกลับเวฟเล็ตเมื่อเลือกระดับการแปลงเป็นสอง 72
รูปที่ 3.11	แสดงการเลือกค่าเฉลี่ยของจำนวนบิตต่อจุดภาพ หรือ ค่าเทรซ โสลเพื่อ เป็นตัวกำหนดระดับความละเอียดของภาพ 72
รูปที่ 3.12	แสดงโคดะล็อกบ็อกซ์ ที่ใช้ในการเก็บค่าจากการลดขนาดข้อมูล 73
รูปที่ 3.13	แสดงโคดะล็อกบ็อกซ์ ที่ใช้ในการเลือกข้อมูลโดยเลือกข้อมูล 73
รูปที่ 3.14	แสดงโคดะล็อกบ็อกซ์ที่ใช้ในการเลือกข้อมูลที่จะทำการเข้ารหัส โดยเลือกข้อมูล 74
รูปที่ 3.15	แสดงโคดะล็อกบ็อกซ์ที่ใช้ในการกำหนดชื่อข้อมูลหลังจากการเข้ารหัส 74
รูปที่ 3.16	แสดงผลที่ได้จากการเข้ารหัส 75
รูปที่ 3.17	แสดงผลที่ได้จากการถอดรหัส 75
รูปที่ 3.18	แสดงภาพที่ได้จากการแปลงกลับเอ็มเบดซีโรทรี 76
รูปที่ 3.19	แสดงแมสเสชบ็อกซ์ของค่าความผิดพลาด 76
รูปที่ 3.20	แสดงแมสเสชบ็อกซ์ของค่าอัตราเข้ารหัส 77
รูปที่ 4.1	แสดงภาพต้นแบบที่ใช้ในการทดลอง Lena ขนาด 512x512 จุดภาพ 79
รูปที่ 4.2	แสดงค่าสัมประสิทธิ์ของภาพ Lena ขนาด 256x256 จุดภาพ 80
รูปที่ 4.3	แสดงภาพหลังการการแปลงเวฟเล็ต 81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

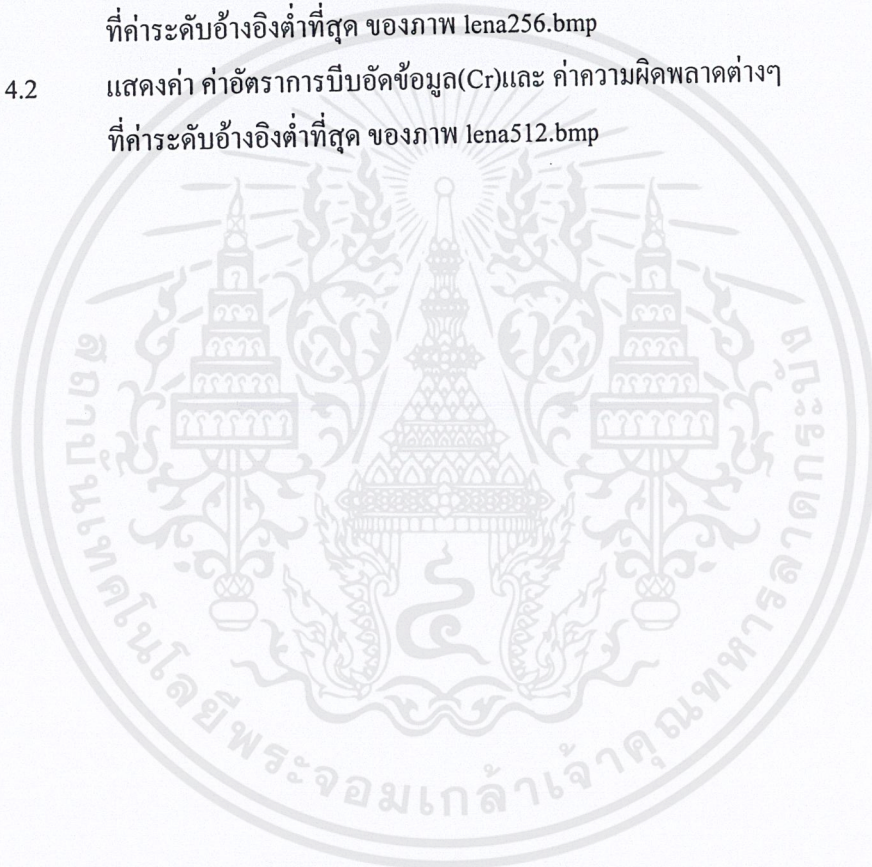
		หน้า
รูปที่ 4.4	แสดงค่าสัมประสิทธิ์ของภาพ Lena ขนาด 256x256 จุดภาพภายหลังจากการแปลงเวฟเล็ต	82
รูปที่ 4.5	แสดงภาพที่สร้างกลับขึ้นมาหลังจากการแปลงเวฟเล็ต	83
รูปที่ 4.6	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 1	85
รูปที่ 4.7	กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 1	85
รูปที่ 4.8	แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกปิดค่าระดับอ้างอิงที่ 1	86
รูปที่ 4.9	แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 1	86
รูปที่ 4.10	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.5	87
รูปที่ 4.11	กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.5	87
รูปที่ 4.12	แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกปิดค่าระดับอ้างอิงที่ 0.5	88
รูปที่ 4.13	แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.5	88
รูปที่ 4.14	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.25	89
รูปที่ 4.15	กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.25	89
รูปที่ 4.16	แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกปิดค่าระดับอ้างอิงที่ 0.25	90
รูปที่ 4.17	แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.25	90
รูปที่ 4.18	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.125	91

รูปที่ 4.19	กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ทของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.125	91
รูปที่ 4.20	แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกปิดค่าระดับอ้างอิงที่ 0.125	92
รูปที่ 4.21	แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.125	92
รูปที่ 4.22	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ทของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.0625	93
รูปที่ 4.23	กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ทของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.0625	93
รูปที่ 4.24	แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกปิดค่าระดับอ้างอิงที่ 0.0625	94
รูปที่ 4.25	แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.0625	94
รูปที่ 4.26	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ทของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.03125	95
รูปที่ 4.27	กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ทของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.03125	95
รูปที่ 4.28	แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกปิดค่าระดับอ้างอิงที่ 0.03125	96
รูปที่ 4.29	แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.03125	96
รูปที่ 4.30	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ทของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.015625	97
รูปที่ 4.31	กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ทของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.015625	97

	หน้า	
รูปที่ 4.32	แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกบีบอัดค่าระดับอ้างอิงที่ 0.015625	98
รูปที่ 4.33	แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกบีบอัดค่าระดับอ้างอิงที่ 0.015625	98
รูปที่ 4.34	แสดงค่าระดับอ้างอิงและค่าอัตราการบีบอัดข้อมูลของภาพ lena256.bmp	99
รูปที่ 4.35	แสดงค่าระดับอ้างอิงและค่าเฉลี่ยจำนวนบิตต่อจุดภาพของภาพ lena256.bmp	100
รูปที่ 4.36	แสดงค่าระดับอ้างอิงและค่า MSE ของภาพ lena256.bmp	100
รูปที่ 4.37	แสดงค่าระดับอ้างอิงและค่า PSNR ของภาพ lena256.bmp	101
รูปที่ 4.38	แสดงค่าระดับอ้างอิงและค่าอัตราการบีบอัดข้อมูลของภาพ lena256.bmp เทียบกับภาพ lena512.bmp	102
รูปที่ 4.39	แสดงค่าระดับอ้างอิงและค่าเฉลี่ยจำนวนบิตต่อจุดภาพของภาพ lena256.bmp เทียบกับภาพ lena512.bmp	102
รูปที่ 4.40	แสดงค่าระดับอ้างอิงและค่า MSE ของภาพ lena256.bmp เทียบกับภาพ lena512.bmp	103
รูปที่ 4.41	แสดงความละเอียดของภาพที่ bpp ค่าต่างๆ	104
รูปที่ 4.42	แสดงค่าเฉลี่ยจำนวนบิตต่อจุดภาพ และค่าอัตราการบีบอัดข้อมูลของภาพ lena256.bmp	105
รูปที่ 4.43	แสดงค่าเฉลี่ยจำนวนบิตต่อจุดภาพ และค่า MSE ของภาพ lena256.bmp	105
รูปที่ 4.44	แสดงค่าเฉลี่ยจำนวนบิตต่อจุดภาพ และค่า RMS ของภาพ lena256.bmp	106
รูปที่ 4.45	แสดงค่าเฉลี่ยจำนวนบิตต่อจุดภาพ และค่า PSNR ของภาพ lena256.bmp	106
รูปที่ 4.46	กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการของ EZW และ JPEG (ใช้ภาพ lena ขนาด 256x256 จุดภาพ)	107
รูปที่ 4.47	กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการของ EZW และ JPEG (ใช้ภาพ lena ขนาด 512x512 จุดภาพ)	108

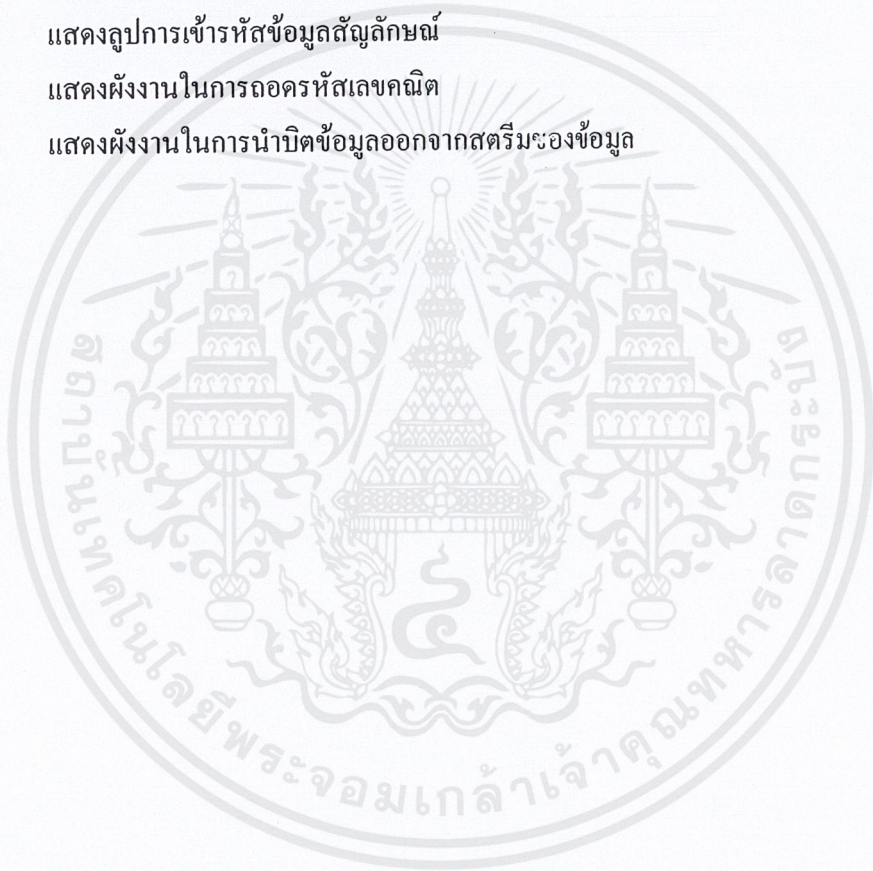
สารบัญตาราง

		หน้า
ตารางที่ 2.1	แสดงจำนวนบิตที่ใช้ในการจัดเก็บข้อมูลภาพ ที่แปรตามค่า N และ m	4
ตารางที่ 2.2	แสดงจำนวนไบต์ที่ใช้ในการจัดเก็บข้อมูลภาพ ที่แปรตามค่า N และ m	4
ตารางที่ 2.3	แสดงการแบ่งขอบเขตตามความน่าจะเป็นสะสม	39
ตารางที่ 3.1	แสดงค่าของสัมประสิทธิ์ที่มีนัยสำคัญและไม่สำคัญในเฮ็คเบดซีโรทรี	52
ตารางที่ 3.2	แสดงค่าของสัมประสิทธิ์ที่มีค่านัยสำคัญ	52
ตารางที่ 3.3	แสดงค่าของสัมประสิทธิ์ที่มีค่านัยสำคัญ	53
ตารางที่ 4.1	แสดงค่า ค่าอัตราการบีบอัดข้อมูลและ ค่าความผิดพลาดต่างๆ ที่ค่าระดับอ้างอิงต่ำที่สุด ของภาพ lena256.bmp	99
ตารางที่ 4.2	แสดงค่า ค่าอัตราการบีบอัดข้อมูล(Cr)และ ค่าความผิดพลาดต่างๆ ที่ค่าระดับอ้างอิงต่ำที่สุด ของภาพ lena512.bmp	101



สารบัญผ้งงาน

	หน้า	
ผ้งงานที่ 3.1	แสดงผ้งการทำงานการแปลงเวฟเล็ด	49
ผ้งงานที่ 3.2	ผ้งงานทำงานการแปลงกลับเวฟเล็ด	50
ผ้งงานที่ 3.3	เข้ารหัสเอ็มเบดซีโรทรี	55
ผ้งงานที่ 3.4	การทำงาน EZW ส่วนที่สอง	56
ผ้งงานที่ 3.5	แสดงการทำกลับของเอ็มเบดซีโรทรี	58
ผ้งงานที่ 3.6	แสดงการทำกลับของเอ็มเบดซีโรทรี(ต่อ)	59
ผ้งงานที่ 3.7	แสดงผ้งงานการเข้ารหัสเลขคณิต	64
ผ้งงานที่ 3.8	แสดงรูปการเข้ารหัสข้อมูลสัญลักษณ์	65
ผ้งงานที่ 3.9	แสดงผ้งงานในการถอดรหัสเลขคณิต	66
ผ้งงานที่ 3.10	แสดงผ้งงานในการนำบิตข้อมูลออกจากสตรีมระองข้อมูล	67



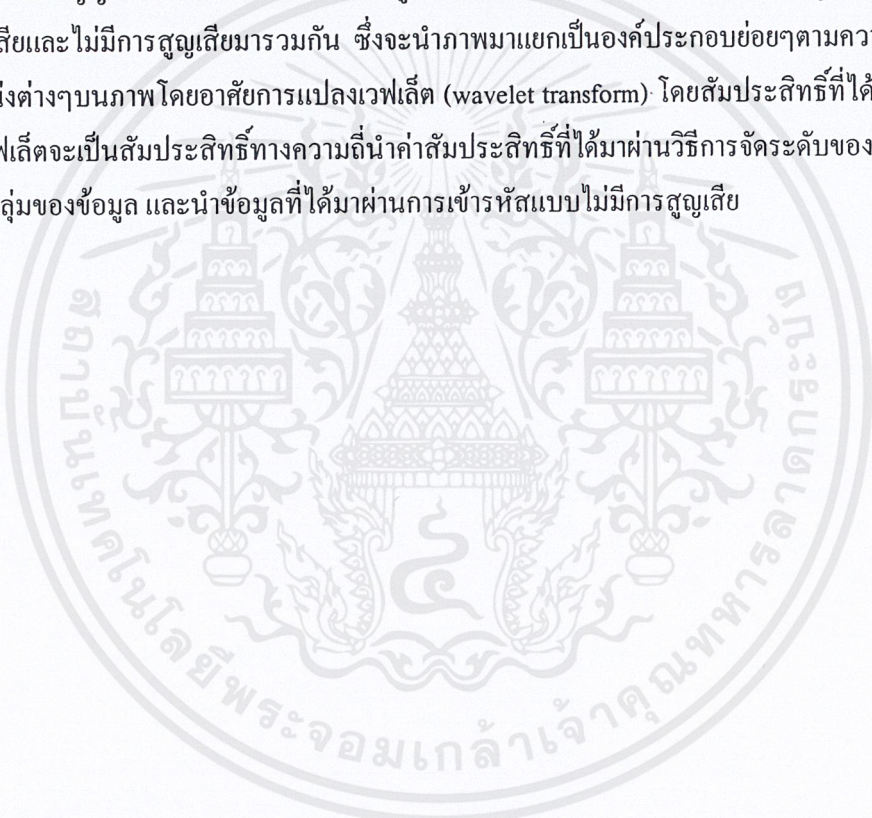
บทที่ 1

บทนำ

การลดขนาดข้อมูล (data compression) เป็นหัวใจสำคัญประการหนึ่งในระบบการสื่อสาร เพื่อให้สามารถส่งข้อมูลจำนวนมากไปในช่องทางสื่อสารที่จำกัด และให้ถึงจุดหมายปลายทางในเวลาที่เหมาะสม การบีบอัดข้อมูลภาพ (image compression) เป็นหัวข้อหนึ่งในการลดขนาดข้อมูล โดยเป็นวิธีการลดปริมาณข้อมูลที่ยังคงข่าวสาร (information) ที่กำลังสนใจไว้

เทคนิคในการลดขนาดข้อมูลภาพสามารถแบ่งออกได้เป็น 2 กลุ่ม คือการลดข้อมูลแบบที่มีการสูญเสีย (lossy compression) และการลดข้อมูลแบบไม่มีการสูญเสีย (lossless compression) สามารถนำข้อมูลกลับมาได้ใหม่อย่างสมบูรณ์ บางครั้งเรียกว่า การเข้ารหัสแบบเอนโทรปี (entropy coding)

ในปริิณญาณินพณัชนี จะใช้การลดข้อมูลแบบผสม คือการนำการลดขนาดข้อมูลแบบที่ยอมให้มีการสูญเสียและไม่มีการสูญเสียมารวมกัน ซึ่งจะนำภาพมาแยกเป็นองค์ประกอบย่อยๆตามความถี่ของภาพที่ตำแหน่งต่างๆบนภาพโดยอาศัยการแปลงเวฟเลต (wavelet transform) โดยสัมประสิทธิ์ที่ได้หลังจากการแปลงเวฟเลตจะเป็นสัมประสิทธิ์ทางความถี่นำค่าสัมประสิทธิ์ที่ได้มาผ่านวิธีการจัดระดับของสัมประสิทธิ์และจัดกลุ่มของข้อมูล และนำข้อมูลที่ได้มาผ่านการเข้ารหัสแบบไม่มีการสูญเสีย



บทที่ 2

ทฤษฎีและหลักการเบื้องต้น

2.1 พื้นฐานของภาพเชิงตัวเลข

2.1.1 โมเดลภาพ(Image model)

ภาพเป็นข้อมูลสองมิติ (two dimensions) สามารถเขียนให้อยู่ในรูปของฟังก์ชันทางคณิตศาสตร์ได้ด้วย $f(x, y)$ โดยที่ x และ y แสดงตำแหน่งของฟังก์ชันตามแนวแกนอน และแนวแกนตั้งตามลำดับค่าของฟังก์ชัน $f(x, y)$ แสดงถึงความเข้มแสง ในสเปซเชิงโคออร์ดิเนตที่จุดนั้น ภาพที่มองเห็นโดยทั่วไปประกอบขึ้นด้วยแสงสะท้อนจากวัตถุตามธรรมชาติแล้วจะถูกกำหนดคุณสมบัติจากองค์ประกอบ 2 ประการ ประการแรกได้แก่ ปริมาณของแสงที่ได้มาจากแหล่งกำเนิดมายังวัตถุที่กำลังมองซึ่งเรียกองค์ประกอบนี้ว่า ความสว่าง (illumination) แทนด้วย $i(x, y)$ อีกประการหนึ่งคือปริมาณแสงสะท้อนออกจากวัตถุซึ่งจะเรียกว่าการสะท้อน (reflection) แทนด้วย $r(x, y)$ โดยผลคูณของฟังก์ชัน $i(x, y)$ และ $r(x, y)$ จะได้เป็น $f(x, y)$ ดังสมการ (2.1)

$$f(x, y) = i(x, y) \times r(x, y) \quad (2.1)$$

โดยเมื่อ

$$\begin{aligned} 0 < i(x, y) < \infty \\ 0 < r(x, y) < 1 \end{aligned} \quad (2.2)$$

ความสว่าง $i(x, y)$ เป็นพลังงานรูปหนึ่ง ที่มีค่าไม่เป็นศูนย์และมีค่าจำกัดในช่วงหนึ่ง และการสะท้อน $r(x, y)$ จะมีค่าอยู่ระหว่าง 0 ซึ่งหมายถึง ดูดซับแสงทั้งหมด และเป็น 1 เมื่อสะท้อนแสงทั้งหมด ดังนั้นคุณลักษณะของฟังก์ชัน $i(x, y)$ กำหนดโดยแหล่งกำเนิดแสง ส่วนฟังก์ชัน $r(x, y)$ กำหนดโดยคุณสมบัติของวัตถุในการสะท้อนแสง

2.1.2 การสุ่มและการจัดระดับ (Sampling and Quantization)

การนำภาพมาประมวลผลโดยคอมพิวเตอร์จะต้องทำให้ภาพอยู่ในรูปแบบที่เหมาะสมกับคอมพิวเตอร์ ฟังก์ชันของภาพ $f(x, y)$ ต้องทำให้เป็นดิจิทัล ทั้งทางสเปซเชิง และ แอมพลิจูด การทำให้เป็นดิจิทัลของทางสเปซเชิงโคออดิเนต (x, y) จะเรียกว่าการสุ่มตัวอย่างภาพ (image sampling) ในขณะที่การทำให้เป็นจำนวนเชิงเลขในแง่ของแอมพลิจูด จะเรียกว่าการควอนไทซ์

ภาพต่อเนื่อง $f(x, y)$ จะได้รับการประมาณค่าโดยการสุ่มตัวอย่าง แล้วนำมาเรียงกันในรูปแบบของ $N \times N$ อาร์เรย์ดังแสดงในสมการ 2.3 เมื่อแต่ละอิเลเมนต์ (element) ในอาร์เรย์เป็นจำนวนเต็มหน่วย

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & \dots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(n-1, N-1) \end{bmatrix} \quad (2.3)$$

ทางด้านขวาของสมการ (2.3) จะเรียกว่าภาพเชิงตัวเลข หรือ ภาพดิจิทัล (digital image) ในขณะที่แต่ละ อิลิเมนต์ของอาร์เรย์จะเรียกว่า องค์ประกอบของภาพ (image element, picture element หรือ pixel)

2.1.3 ปริมาณข้อมูลภาพ

การแปลงภาพจากสัญญาณต่อเนื่อง(analog) ให้เป็นสัญญาณดิจิทัลที่อยู่ในลักษณะของสัญญาณที่ไม่ต่อเนื่องหรือสัญญาณคิสริต (discrete signal) ซึ่งจะต้องผ่านการสุ่มและจัดระดับ(sample and quantized) โดยการสุ่มจะทำการสุ่มในแนวนอนจำนวน N ครั้งและแนวตั้งจำนวน N ครั้ง เนื่องจากภาพมีลักษณะเป็น 2 มิติ ทำให้ได้ข้อมูลภาพเป็น $N \times N$ ข้อมูล ดังที่ได้กล่าวมาแล้วในหัวข้อที่ 2.1.2 โดยในทางปฏิบัติไม่จำเป็นต้องเป็นเมตริกซ์ขนาด $N \times N$ เสมอไป อาจเป็น $N \times M$ ใดๆก็ได้

ในการแปลงภาพจากอนาล็อกให้เป็นภาพดิจิทัล นอกจากพิจารณาจำนวนของการสุ่มแล้วยังต้องพิจารณาถึงจำนวนระดับสีที่มีได้ในแต่ละจุดของภาพ เช่นระดับสีเทา (gray scale)

$$N = 2^n \quad (2.4)$$

และ

$$G = 2^m \quad (2.5)$$

โดยที่ G เป็นจำนวนระดับสีเทา โดยในหนึ่งจุดภาพหรือพิกเซลแทนด้วยความยาว m บิต ดังนั้นเนื้อที่ในการเก็บภาพดิจิทัลทั้งหมดแสดงดังในสมการ (2.6)

$$b = N \times N \times m \quad (2.6)$$

ตัวอย่างเช่น หากภาพสีขนาด 128×128 พิกเซล โดยมีระดับของสีเป็น 64 ระดับ ดังนั้นจะต้องใช้เนื้อที่ในการเก็บข้อมูลทั้งหมด 98,304 บิต ตารางที่ 2.1 แสดงค่าของ b กับค่าของ N และ m และในตารางที่ 2.2 แสดงจำนวนไบต์ค่าของ N และ m

ปัจจัยที่เป็นตัวกำหนดขนาดของภาพที่เห็นได้ชัดเจนจากตารางที่ 2.1 และ 2.2 ก็คือค่าของ N และ m และทั้งสองค่านี้ยังเป็นตัวกำหนด ความละเอียด (resolution) ของภาพ การเพิ่มค่า N และ m ให้มากขึ้นจะทำให้ภาพมีความใกล้เคียงกับความเป็นจริงมากขึ้น นั่นคือมีความละเอียดของภาพมากขึ้น แต่จากสมการที่ 2.6 พบว่าถ้า N และ m มากขึ้นจะทำให้จำนวนบิตที่ใช้ในการเก็บข้อมูลจะเพิ่มขึ้นตาม

ตารางที่ 2.1 แสดงจำนวนบิตที่ใช้ในการจัดเก็บข้อมูลภาพ ที่แปรตามค่า N และ m

N/m	1	2	3	4	5	6	7	8
32	1024	2048	3027	4096	5120	6144	7816	8129
64	4096	8192	12288	16384	20480	24567	28672	327768
128	16384	32786	49152	65536	81920	98304	114686	131072
256	65536	113072	196608	262144	327680	393216	458725	524288
512	264144	524288	786432	1048576	1310720	1572864	1835008	2097152

ตารางที่ 2.2 แสดงจำนวนไบต์ที่ใช้ในการจัดเก็บข้อมูลภาพที่แปรตามค่า N และ m

N/m	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	32768	65536	131072	131072	262144	262144	262144	262144

ผลของ N และ m จะมีผลต่อคุณภาพของภาพ โดยถ้า N และ m มีค่าสูงนั้นหมายถึงภาพก็จะยิ่งมีรายละเอียดของภาพที่สูงขึ้นเช่นกัน การพิจารณาว่าภาพนั้นๆ ดี หรือไม่ดี เป็นการยากที่จะนิยามเพราะว่าคุณภาพของภาพที่ต้องการนั้น ขึ้นอยู่กับงานที่จะนำไปใช้

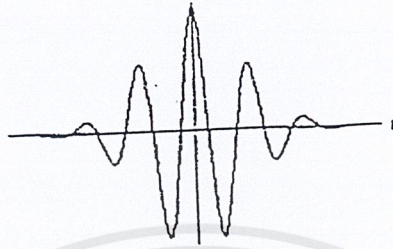
2.2 ทฤษฎีพื้นฐานของการแปลงเวฟเล็ต

2.2.1 ลักษณะพื้นฐานของเวฟเล็ต

ทฤษฎีเวฟเล็ต (Wavelet Theory) เป็นคณิตศาสตร์ที่ใช้อธิบายการสร้างโมเดลของสัญญาณระบบ หรือ กระบวนการทางฟิสิกส์ใดๆ ซึ่งประกอบขึ้นด้วยเซตของสัญญาณเฉพาะหรือระบบย่อยๆ มารวมกันเป็นสัญญาณระบบนั้นๆ สัญญาณเฉพาะนี้จะเป็นคลื่นเล็กๆ ซึ่งถูกเรียกว่า “เวฟเล็ต” (Wavelets) ลักษณะของเวฟเล็ตจะเป็นคลื่นที่มีการเปลี่ยนแปลงต่อเนื่อง (Oscillatory) และขนาดของคลื่นจะลดลงสู่ศูนย์อย่างรวดเร็วทั้งสองด้าน ดังแสดงในรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.1 เป็นตัวอย่างของเวฟเลตชนิดหนึ่งเรียกว่าเวฟเลตแบบมอเล็ต (Morlet) สังเกตว่ามีคุณสมบัติของเวฟเลตตามที่กล่าวมาทั้ง 2 เงื่อนไข คือ มีการออสซิลเลตในลักษณะของสัญญาณรูปไซน์ (sinusoidal) ดังแสดงในรูปที่ 2.2 ก.) รวมกับเงื่อนไขของการลดลงอย่างรวดเร็วตามลักษณะฟังก์ชันหน้าต่าง (Windows Function) ดังแสดงในรูปที่ 2.2 ข)



รูปที่ 2.1 เป็นตัวอย่างของเวฟเลตชนิดหนึ่งเรียกว่าเวฟเลตแบบมอเล็ต $g(t) = e^{-\frac{t^2}{64}} \cos(t)$



รูปที่ 2.2 แสดงลักษณะคุณสมบัติของเวฟเลต 2 เงื่อนไข

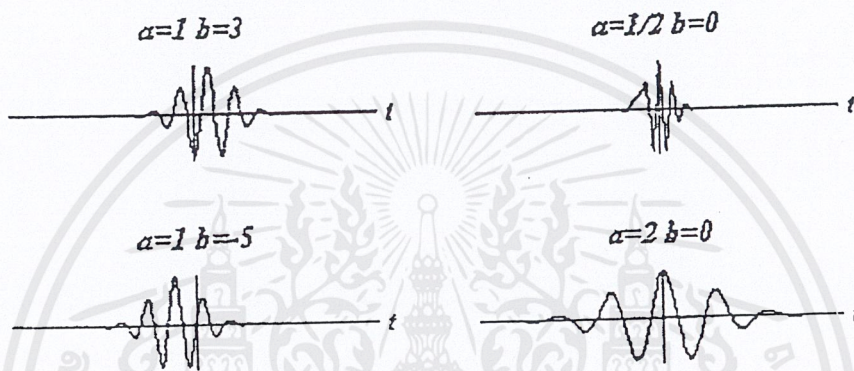
เมื่อทั้งสองเงื่อนไขเกิดขึ้นพร้อมกัน ก็จะทำให้เกิดคลื่นเล็กๆที่เรียกว่า “เวฟเลต” ดังนั้นการคูณกันของฟังก์ชันออสซิลเลต (Oscillate function) และฟังก์ชันการลดลง (Decay function) จะให้ผลเป็นคลื่นเวฟเลต ดังแสดงในรูปที่ 2.1

การนำเวฟเลตหลายๆอันมารวมกันเป็นเซต เพื่อจะใช้ในการอธิบายสัญญาณใดๆ โดยที่คลื่นเวฟเลตแต่ละตัวจะมีโครงสร้างมาจากฟังก์ชันเดียวกัน ซึ่งฟังก์ชันนี้จะเป็นเวฟเลตต้นกำเนิดที่เรียกว่าเวฟเลตแม่ (Mother Wavelets) คลื่นแต่ละอันจะอยู่ในเซตของเวฟเลตนี้ โดยคลื่นแต่ละคลื่นจะเกิดจาก “การสเกล” (Scaling : “a”) และ “การเลื่อนตำแหน่ง” (Translation or Shifting: “b”) ของเวฟเลตแม่ตัวนี้ การสเกลจะหมายถึง การหดเข้า (Compressing) หรือ การขยายออก (Dialation) ในที่นี้จะใช้ “a” แทนพารามิเตอร์ของการหดหรือขยาย ซึ่งจะเป็นการเปลี่ยนความถี่ของเวฟเลตแม่นั่นเอง ส่วนการเลื่อนตำแหน่งกำหนดโดยพารามิเตอร์ “b” จะหมายถึงการเลื่อนตำแหน่งการเกิดของคลื่นเวฟเลตบนแกนเวลา ในรูปที่ 2.3 จะแสดงให้เห็นถึงเวฟเลตที่ถูกสเกลและเลื่อนตำแหน่งไปที่ค่า “a” “b” ต่างๆกัน โดยที่ยังคงอยู่ในเซตของเวฟเลตที่สัมพันธ์กับเวฟเลตแม่ ในรูปที่ 2.1 ซึ่งจะสังเกตเห็นว่า คลื่นเวฟเลตแต่ละคลื่นจะมีรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ร่างเหมือนเดิม โดยยังคงมีจำนวนไซเคิลของการออสซิลเลตเท่าเดิม จะเปลี่ยนไปเฉพาะความถี่และตำแหน่งของกาเวตเท่านั้น นอกจากนี้เวฟเล็ตที่ถูกสเกลจะต้องทำการนอร์มอลไลซ์ (Normalize) ด้วย $1/\sqrt{a}$ เสมอ เพื่อให้เวฟเล็ตที่ถูกสเกลไปแล้วยังคงมีพลังงานเช่นเดียวกับเวฟเล็ตแม่ ดังนั้นถ้าให้ $g(t)$ เป็นฟังก์ชันเวฟเล็ตแม่ จะสามารถเขียนเป็นสมการทั่วไปของเวฟเล็ตที่ตำแหน่ง "a", "b" ใดๆซึ่งสัมพันธ์กับเวฟเล็ตแม่ได้ดังนี้

$$g_{b,a}(t) = \frac{1}{\sqrt{a}} g\left(\frac{t-b}{a}\right) \quad (2.7)$$



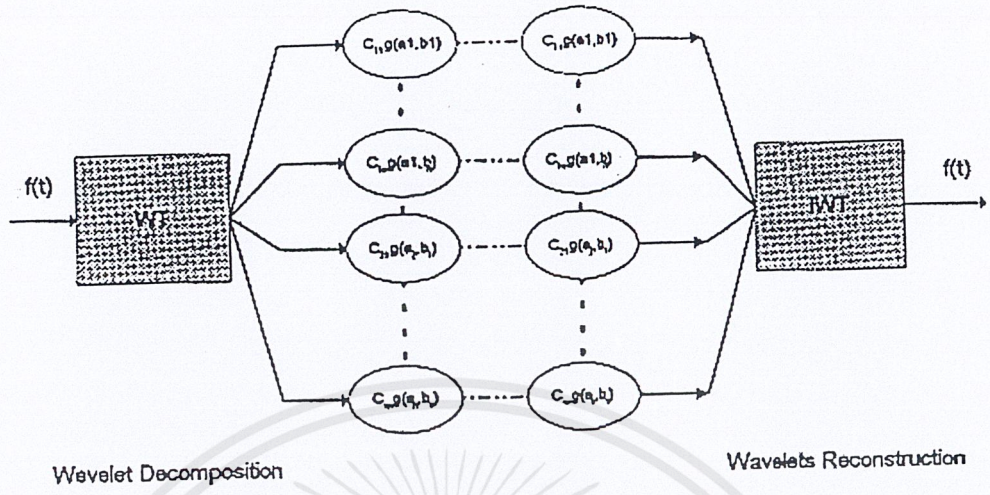
รูปที่ 2.3 ลักษณะของเวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่ง

2.2.2 ทฤษฎีพื้นฐานในการสเกล และเลื่อนตำแหน่งของเวฟเล็ต

หากสามารถอธิบายลักษณะของสิ่งใดสิ่งหนึ่งได้จากสมาชิกเล็กๆที่เป็นองค์ประกอบแล้ว ทฤษฎีเวฟเล็ตก็เป็นอีกวิธีหนึ่งที่จะใช้ในการอธิบายลักษณะของสิ่งต่างๆ หรือระบบใดๆ ได้เช่นเดียวกัน ในการที่จะใช้สมาชิกมาอธิบายสิ่งเหล่านี้ ก็จะเป็นการหาค่าสมาชิกเหล่านี้หลายวิธี การเลือกที่จะใช้วิธีใดก็ขึ้นอยู่กับว่าจะนำไปประยุกต์ใช้กับงานนั้นแล้วได้ประสิทธิภาพดีเพียงใด เช่น การใช้ระบบเลขฐานสิบจะใช้ในระบบบัญชีได้ เหมาะสมกว่าเลขฐานสองหรือฐานสิบหก ในขณะที่ระบบเลขฐานสองจะใช้อธิบายในระบบคอมพิวเตอร์ได้ดีกว่า ดังนั้นในการใช้ทฤษฎีเวฟเล็ตให้ได้ดีก็จะขึ้นอยู่กับงานที่จะนำไปประยุกต์ใช้ด้วย

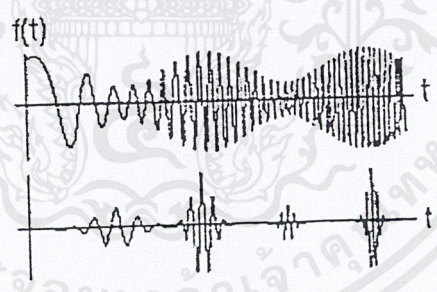
ทฤษฎีเวฟเล็ตจะใช้ในการอธิบายสิ่งใดสิ่งหนึ่งเหมือนการแตกสิ่งเหล่านั้นออกเป็นส่วนๆ ประกอบเล็กๆที่สัมพันธ์กันจะอยู่ในรูปของเวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่งและมีค่าถ่วงน้ำหนักต่างๆ กันคูณอยู่กับเวฟเล็ตแต่ละตัว ดังนั้นจึงเปรียบเสมือนว่าสัญญาณใดๆ จะสามารถสร้างขึ้นมาได้โดยมีคลื่นเวฟเล็ตเป็นฟังก์ชันพื้นฐาน (Basic function) หรือองค์ประกอบพื้นฐานกระบวนการแตกกระจายชิ้นส่วนนี้เรียกว่า “การแตกกระจายเวฟเล็ต” (Wavelet Transform : WT) นั่นเอง ในทำนองเดียวกันการรวมกลับเวฟเล็ต (Wavelet Reconstruction) ซึ่งเป็นการแปลงกลับเวฟเล็ต (Inverse Wavelets Transform : IWT) ก็จะเป็นการนำส่วนประกอบย่อยๆเหล่านี้มาเรียงตัวกันเพื่อประกอบเป็นสัญญาณเดิม ดังแสดงในรูปที่ 2.4 เป็นการแตกกระจายสัญญาณ $f(t)$ ออกเป็นคลื่นเวฟเล็ตที่ความถี่และตำแหน่งต่างกันจำนวน $N \times N$ ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงลักษณะของการแตกกระจายสัญญาณแตรวมกลับสัญญาณของเวฟเล็ต

ถ้าให้ $f(t)$ เป็นสัญญาณที่ประกอบด้วยคลื่นหลายถี่ที่เกิดขึ้นตำแหน่งต่างๆบนแกนเวลา ดังแสดงในรูปที่ 2.4 ส่วนในภาพด้านล่างจะแสดงถึงการดึงสัญญาณ $f(t)$ ที่ตำแหน่งเวลาใดๆออกมา



รูปที่ 2.5 แสดงลักษณะของสัญญาณที่มีหลายความถี่ที่ตำแหน่งเวลาต่างกัน

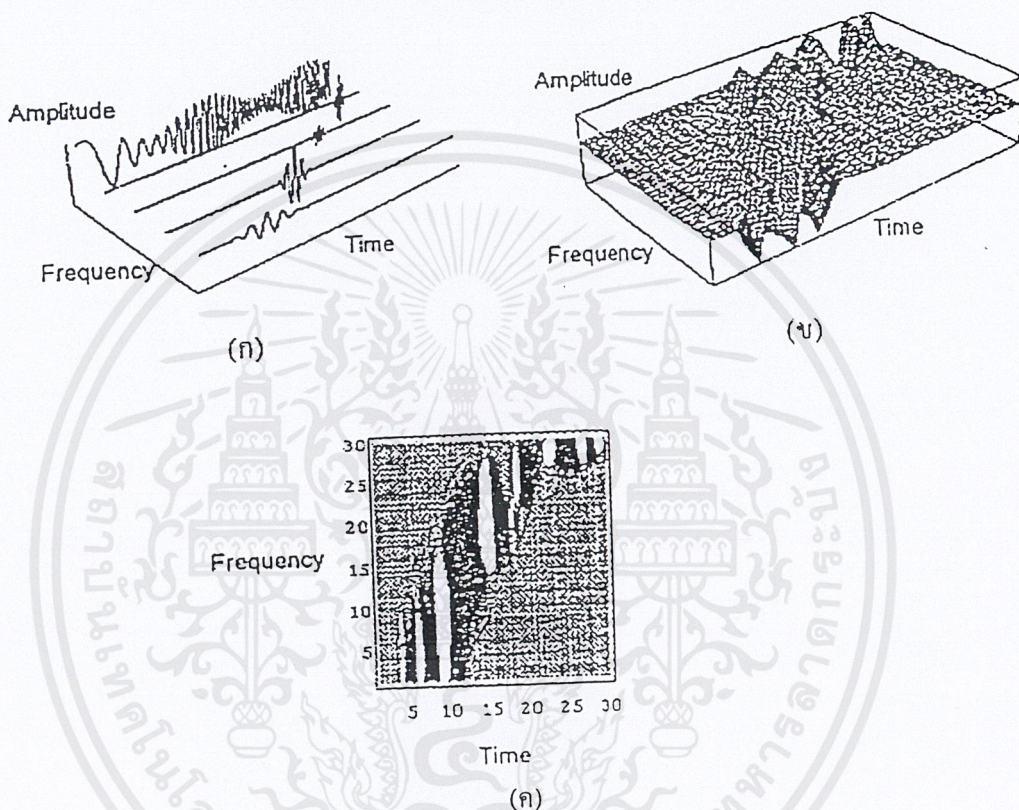
ถ้าทำการพิจารณาในแกน 2 มิติ โดยที่ให้แกนหนึ่งเป็นแกนเวลา ในอีกแกนหนึ่งเป็นแกนความถี่ จะสามารถทำการแยกสัญญาณที่ตำแหน่งเวลาและมีความถี่ใดๆ ได้แสดงในรูปที่ 2.6ก) และในรูปที่ 2.6ข) จะพิจารณาที่ค่าเวลาและค่าความถี่มีความต่อเนื่อง ซึ่งก็จะได้เป็นสัญญาณพื้นผิวที่ต่อเนื่องมากขึ้น ในรูปที่ 2.6ค) จะเป็นการพิจารณาองจกด้านบนของรูปที่ 2.6ข) ซึ่งจะเห็นว่าความเข้มจะแทนขนาดของสัญญาณ

การสเกลและการเลื่อนตำแหน่งจะสามารถเขียนเป็นสมการคณิตศาสตร์ในลักษณะการส่ง (Mapping) ตัวแปรอิสระตัวใหม่ ดังแสดงในสมการที่ (2.8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$t' = \frac{t-b}{a} \tag{2.8}$$

จากการที่สามารถทำการสร้างเซตของเวฟเล็ตได้จากการสเกลและการเลื่อนตำแหน่งของเวฟเล็ตแม่ ดังนั้นจะสามารถเขียนสมการสำหรับการแปลงเวฟเล็ตแบบต่อเนื่อง (Continuous Wavelets Transform :CWT) ที่ใช้ในการกระจาย (Decomposition) สัญญาณของฟังก์ชันใด $f(t)$ ด้วยเวฟเล็ตแม่ $g(t)$ เป็นดังนี้



- ก) แสดงการพิจารณาสัญญาณในลักษณะแบบ 2 มิติ แบบแบ่งช่วง
- ข) แสดงการพิจารณาสัญญาณในลักษณะแบบ 2 มิติ แบบต่อเนื่อง
- ค) แสดงลักษณะของสัญญาณเมื่อพิจารณาจากด้านบน

รูปที่ 2.6 พิจารณาเวฟเล็ตในแกน 2 มิติ โดยที่ให้แกนหนึ่งเป็นแกนเวลา อีกแกนหนึ่งเป็นแกนความถี่

$$W_g f(a,b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t) g^* \left(\frac{t-b}{a} \right) dt \tag{2.9}$$

โดย * หมายถึง คอมเพล็กซ์คอนจูเกต (Complex Conjugate) a เป็นพารามิเตอร์ของการสเกล และ b จะเป็นพารามิเตอร์ของการเลื่อนตำแหน่ง สัญลักษณ์ $W_g f(a,b)$ จะหมายถึง สัญญาณอินพุตที่ขึ้นอยู่กับตัวแปรอิสระ t จะถูกส่ง (Map) ไปเป็นฟังก์ชัน 2 มิติที่ขึ้นอยู่กับตัวแปรอิสระ a และ b โดยที่ $W_g f(a,b)$ จะเป็นค่าสัมประสิทธิ์ของเวฟเล็ต (Wavelet Coefficient) เวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่งจะใช้เป็นตัวแทนสัญญาณได้ สัมประสิทธิ์ของเวฟเล็ตที่ตำแหน่ง a,b จะเป็นค่าแสดงความเหมือนกัน (Degree of เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Correlation) ระหว่างเวฟเลตแม่ที่ถูกสเกล (a) กับสัญญาณอินพุตที่ตำแหน่งเวลานั้น (b) ดังแสดงในรูปที่ 2.7

พิจารณารูปที่ 2.7 ก) เปรียบเทียบกับรูปที่ 2.7 ข) จะเห็นว่าในรูป ก) คลื่นเวฟเลตจะมีความถี่และเฟสใกล้เคียงกับสัญญาณอินพุตมาก ทำให้ปรากฏคลื่นเอาต์พุตออกมาที่สูง แต่ในภาพ ข) คลื่นอินพุตจะมีค่าความถี่สูงกว่าคลื่นเวฟเลตทำให้เอาต์พุตมีค่าต่ำกว่า

สามารถสรุปได้ว่า ถ้าคลื่นเวฟเลตมีความคล้ายกันกับสัญญาณอินพุตเท่าไรทั้งในด้านของความถี่ ลักษณะสัญญาณและตำแหน่งเวลาในการเปรียบเทียบก็จะทำให้ได้ค่าของสัมประสิทธิ์ของการทรานส์ฟอร์มที่มากนั่นเอง เพราะฉะนั้น การทำการแปลงเวฟเลต จะเป็นการอธิบายสัญญาณ $f(t)$ ในรูปแบบของเวฟเลต โดเมน (Wavelet Domain) เทียบกับคลื่นเวฟเลตที่ค่า a, b ใดๆ

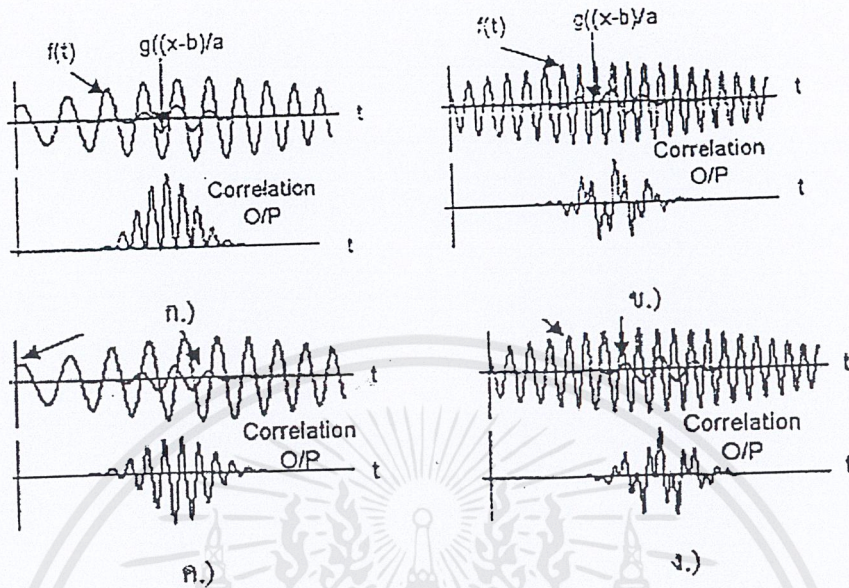
ในรูปที่ 2.8 จะแสดงการเปรียบเทียบลักษณะผลลัพธ์เมื่อนำสัญญาณอินพุต 1 มิติ มาทำการแปลงฟูริเยร์ (Fourier Transform: FT) ก็กับการใช้การแปลงเวฟเลต จะเห็นว่า การแปลงฟูริเยร์ เป็นการส่งสัญญาณจากโดเมนเวลา(1มิติ) ไปเป็นโดเมนความถี่ (1มิติ) แต่การแปลงเวฟเลตจะเป็นการส่งไปเป็น ระยะเวลาและความถี่ (Time-frequency plane) ซึ่งมี 2 มิติ

2.2.3 ความเป็นมาของการแปลงเวฟเลต

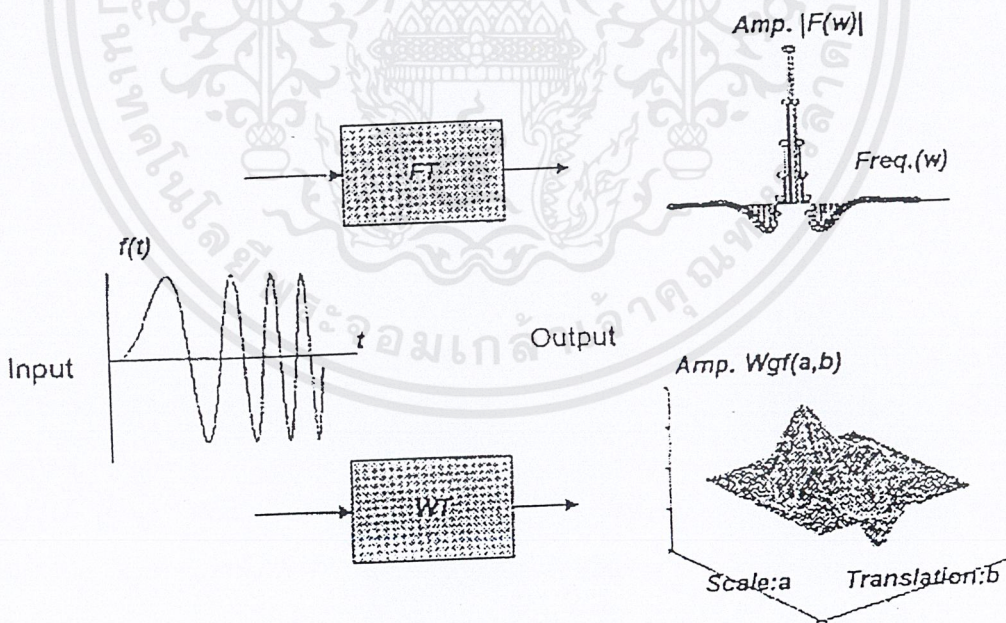
การแปลงฟูริเยร์ เป็นการทรานส์ฟอร์มที่นิยมใช้กันมาก ซึ่งเป็นการแปลงสัญญาณ $f(t)$ จากโดเมนเวลาไปเป็นสัญญาณ $F(\omega)$:ซึ่งอยู่ในโดเมนความถี่ หรือที่เรียกกันทั่วไปว่าเป็นการหาสเปกตรัมของสัญญาณ $f(t)$ นั้นเอง โดยมีสมการเป็น

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (2.10)$$

จะสังเกตพบว่าข้อมูลที่ได้จากการทรานส์ฟอร์มจะมีตลอดย่านความถี่ $(-\infty, \infty)$ ดังนั้นเมื่อเกิดการเปลี่ยนแปลงของสัญญาณอินพุตที่ขณะเวลาใดๆ ก็จะไปส่งผลกระทบต่อตลอดย่านของสเปกตรัม $F(\omega)$ ด้วย นอกจากนี้การใช้การแปลงฟูริเยร์ยังไม่สามารถเลือกทำการวิเคราะห์สัญญาณเฉพาะช่วงเวลา หรือเฉพาะช่วงความถี่ใดๆ ซึ่งบางครั้งต้องการวิเคราะห์สัญญาณเพียงแค่เฉพาะช่วงเวลาหรือ ช่วงความถี่เท่านั้น ดังนั้นการแปลงฟูริเยร์ จึงให้ข้อมูลที่มากเกินไป นอกจากนั้นในทางปฏิบัติสัญญาณอินพุตจะมีการเปลี่ยนแปลงที่ไม่คงที่ (Non-Stationary Signal) ซึ่งถ้าใช้ การแปลงฟูริเยร์ จะทำให้สัญญาณคลื่นเปลี่ยนเวลาในการคำนวณ เนื่องจากมีข้อมูลมากเกินไปในการคำนวณใหม่แต่ละครั้ง ดังนั้นการแปลงฟูริเยร์จึงเหมาะสำหรับสัญญาณที่มีการเปลี่ยนแปลงที่คงที่ (Stationary Signal)



รูปที่ 2.7 แสดงลักษณะของผลของการสหสัมพันธ์ (Correlation) ระหว่าง $f(t), g(\frac{t-b}{a})$



รูปที่ 2.8 แสดงลักษณะความแตกต่างของสัญญาณเอตพัตรระหว่างการแปลงฟูริเยร์ กับ การแปลงเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงกาบอร์ (Gabor Transform:GT) เป็นทรานส์ฟอร์มที่ได้พัฒนาขึ้นเป็นแบบแรก เพื่อแก้ปัญหาที่กล่าวมาข้างต้น โดยจะเป็นการทำ การแปลงฟูริเยร์ เฉพาะช่วงเวลา และช่วงความถี่ที่กำหนดผ่านฟังก์ชันหน้าต่างต่าง (Window function :w.f.) ซึ่งฟังก์ชันหน้าต่างนี้จะใช้เกาส์เซียนฟังก์ชัน (Gaussian function) เท่านั้น

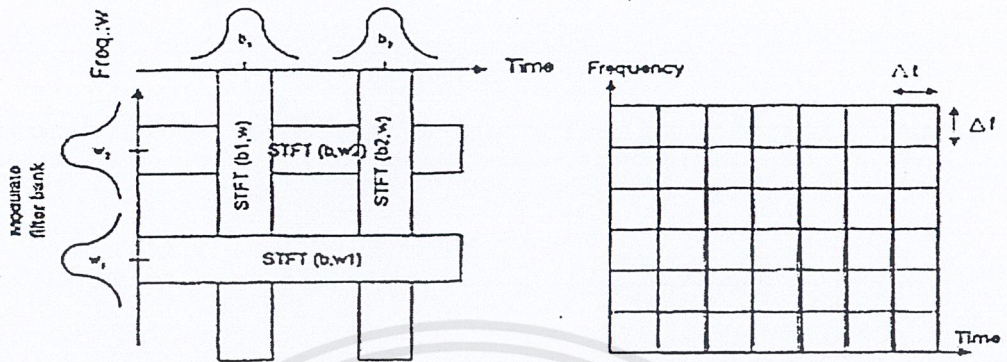
$$g_a(t) = \frac{1}{2\sqrt{\pi a}} e^{-\frac{t^2}{4a}} \quad (2.11)$$

$$G_b^a f(\omega) = \int_{-\infty}^{\infty} (e^{-i\omega t} f(t)) g_a(t-b) dt \quad (2.12)$$

สมการที่ 2.11 เป็นฟังก์ชันหน้าต่างเกาส์เซียน (Gaussian window function) และสมการที่ 2.12 เป็นสมการของการแปลงกาบอร์จะเห็นว่าทรานส์ฟอร์มแบบนี้จะสามารถเลือกตำแหน่งการวิเคราะห์สัญญาณ โดยกำหนดที่พารามิเตอร์ b และเลือกช่วงความถี่ด้วยพารามิเตอร์ a แต่เนื่องจากการแปลงกาบอร์ใช้ฟังก์ชันหน้าต่างแบบเดียวซึ่งอาจจะไม่เหมาะสมกับการประยุกต์ใช้งานในด้านต่าง ๆ ดังนั้นจึงได้มีการพัฒนาต่อมาเป็นการแปลงฟูริเยร์ในช่วงเวลาสั้น (Short Time Fourier Transform : STFT) ซึ่งจะสามารถทำการเลือกใช้ให้เหมาะสมกับสัญญาณอินพุตที่จะวิเคราะห์ได้ดีขึ้น

$$G_b^a f(\omega) = \int_{-\infty}^{\infty} (e^{-i\omega t} f(t) w(t-b) dt \quad (2.13)$$

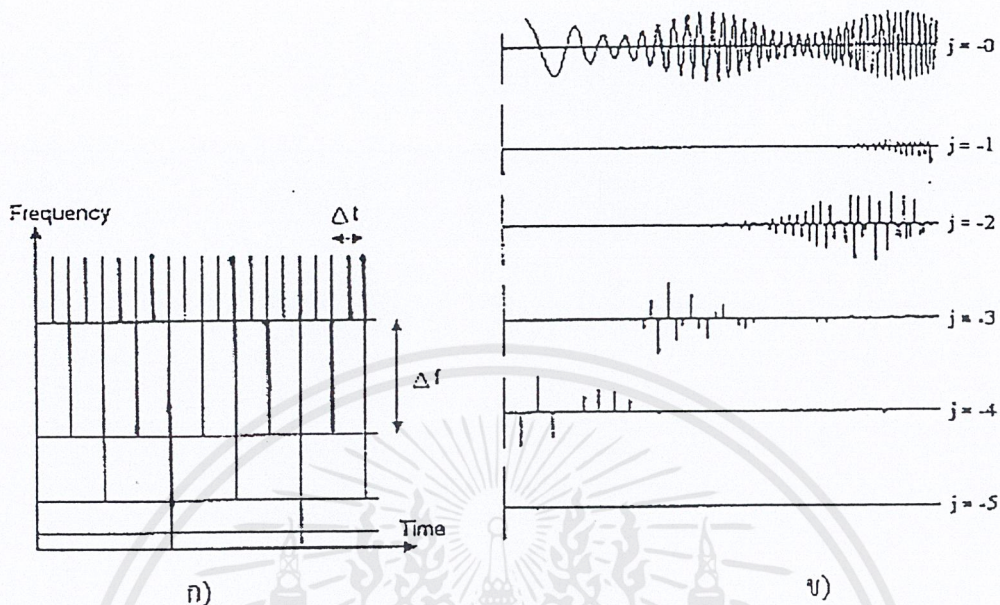
สมการที่ 2.13 จะเป็นสมการการแปลงฟูริเยร์ในช่วงเวลาสั้น โดยที่ตำแหน่งเวลาในการวิเคราะห์จะกำหนดโดยค่า “ b ” และช่วงความถี่ในการวิเคราะห์กำหนดด้วยความถี่ของฟังก์ชันหน้าต่างนั้นๆ แต่จะสังเกตเห็นว่า ทั้งการแปลงกาบอร์ และการแปลงฟูริเยร์ในช่วงเวลาสั้น จะมีลักษณะของช่วงการวิเคราะห์ที่คงที่ดังแสดงในรูปที่ 2.9 (ช่วงเวลา และช่วงความถี่คงที่ : Fixed resolution transform) ทั้งนี้เนื่องจากสัญญาณในทางปฏิบัติ นั้น สัญญาณความถี่สูงจะมีช่วงคาบเวลาที่แคบ ในขณะที่สัญญาณความถี่ต่ำจะมีช่วงคาบเวลาที่กว้างกว่า ดังนั้นการแปลงกาบอร์ และการแปลงฟูริเยร์ในช่วงเวลาสั้น อาจจะทำให้ข้อมูลที่ได้จากการทรานส์ฟอร์มออกมา มาก หรือ น้อยเกินไป (ต้องทำการทรานส์ฟอร์มหลายช่วงเวลามาประกอบกัน จึงจะได้ข้อมูลครบถ้วนในช่วงเวลาที่ต้องการ) ดังนั้นช่วงเวลาควรจะต้องมีการเปลี่ยนแปลงสัมพันธ์กับช่วงความถี่ที่จะทำการวิเคราะห์ โดยที่ช่วงความถี่ที่สูงขึ้นจะมีช่วงเวลาในการวิเคราะห์ที่น้อยลงและในขณะที่เป็นการวิเคราะห์ช่วงความถี่ต่ำจะต้องใช้ช่วงเวลาที่มากกว่าในการวิเคราะห์สัญญาณ



รูปที่ 2.9 แสดงลักษณะของระนาบเวลาความถี่แบบคงที่ (Constant Time-Frequency plane)

จากเหตุผลดังกล่าวข้างต้นทำให้มีการพัฒนาเป็นการทรานส์ฟอร์มในรูปแบบใหม่ที่เรียก “การแปลงเวฟเล็ต” ซึ่งจะเป็นทรานส์ฟอร์มผ่านฟังก์ชันหน้าต่าง แต่ฟังก์ชันหน้าต่างจะสามารถเลื่อนตำแหน่งและปรับเปลี่ยนช่วงเวลาให้เหมาะกับช่วงความถี่ที่จะวิเคราะห์ได้ โดยเรียกฟังก์ชันหน้าต่างแบบนี้ว่า “ฟังก์ชันเวฟเล็ต” (Wavelet function)

$g(t)$ จะเป็นฟังก์ชันเวฟเล็ตแม่ ซึ่งจะถูกลื่อนตำแหน่งและถูกสเกล(เปลี่ยนความถี่) โดยพารามิเตอร์ “ b ” และ “ a ” ตามลำดับ โดยที่ช่วงเวลาและความถี่ในการทรานส์ฟอร์มจะสัมพันธ์กันในลักษณะที่กล่าวมาข้างต้น ดังแสดงในรูปที่ 2.10 ก) ซึ่งจะเห็นว่าระนาบเวลาและความถี่ จะมีลักษณะการเปลี่ยนแปลงช่วงเวลา และความถี่ที่สัมพันธ์กันคือ ในช่วงความถี่สูงจะใช้ช่วงเวลาในการวิเคราะห์เพียงสั้นๆ (Δf สูง Δt ต่ำ) แต่ในช่วงความถี่ต่ำจะมีช่วงเวลาในการวิเคราะห์ที่ยาวนานขึ้น (Δf ต่ำ Δt สูง) ซึ่งจะเหมาะสมกับลักษณะสัญญาณในทางปฏิบัติ ส่วนในรูปที่ 2.10 ข) จะเป็นลักษณะการแตกกระจายสัญญาณออกไปในระดับความละเอียด(ความถี่) ที่ต่างกันออกไป ขอให้พิจารณาเปรียบเทียบกับรูปที่ 2.7 จะพบว่ามีความสัมพันธ์กัน ในระดับ $j = -1$ จะมีเฉพาะความถี่สูง และที่ระดับที่ j ต่ำลงสัญญาณจะมีค่าความถี่ที่ต่ำด้วยสอดคล้องกับภาพของ ระนาบเวลาและความถี่ นอกจากนี้ $g(t)$ ยังมีได้หลายรูปแบบ แต่ต้องเป็นไปตามคุณสมบัติของการแปลงเวฟเล็ต ดังนั้นจึงทำให้การแปลงเวฟเล็ตมีความยืดหยุ่นและสามารถเลือกใช้ให้เหมาะสมกับงานที่นำไปใช้ได้เป็นอย่างดี



ก.) แสดงลักษณะของระนาบเวลาและความถี่สำหรับการแปลงเวฟเล็ด
 ข.) แสดงลักษณะการแตกกระจายสัญญาณไปยังระดับความละเอียดต่างๆ
 รูปที่ 2.10 แสดงลักษณะของระนาบเวลาเทียบกับความถี่
 และลักษณะการแตกกระจายสัญญาณของเวฟเล็ด

2.2.4 ความสัมพันธ์ระหว่างการแปลงเวฟเล็ด และการแปลงฟูรีเยร์ในลักษณะของฟังก์ชัน หน้าต่าง

จากที่ได้กล่าวมาข้างแล้วในหัวข้อที่ผ่านมา ในหัวข้อนี้จะกล่าวเปรียบเทียบการแปลงฟูรีเยร์ และการแปลงเวฟเล็ด ในลักษณะของฟังก์ชันหน้าต่างให้ละเอียดขึ้น จากการที่การแปลงเวฟเล็ดมีความสัมพันธ์กับการแปลงฟูรีเยร์ และ อนุกรมฟูรีเยร์ (Fourier Series : FS) โดยที่การแปลงฟูรีเยร์จะเป็นการอธิบายสัญญาณในลักษณะของผลรวมของการถ่วงน้ำหนักของฟังก์ชันเอ็กซ์โพเนนเชียลที่ความถี่ต่างๆกัน น้ำหนักของแต่ละความถี่จะเป็นค่าสัมประสิทธิ์ของการทำการแปลงฟูรีเยร์แต่ในกรณีของการแปลงเวฟเล็ด จะเป็นการนำเสนอฟังก์ชันในรูปแบบการประกอบกันของเวฟเล็ดแม่ที่ถูกสเกลและเลื่อนตำแหน่ง ดังนั้นเวฟเล็ดแม่จะเปรียบได้กับเอ็กซ์โพเนนเชียลฟังก์ชัน ส่วนการสเกลและการเลื่อนตำแหน่งจะเทียบได้กับการเลื่อนความถี่ (Frequency shifting) ขณะที่พื้นผิว 2 มิติของสัมประสิทธิ์เวฟเล็ดจะใช้แทนสัมประสิทธิ์ของฟูรีเยร์ที่เป็น 1 มิติ ดังนั้นถ้าให้เวฟเล็ดแม่เท่ากับ $g(t) = e^{it}$ และให้ค่าการสเกล $a = 1/\omega$ การเลื่อนตำแหน่ง $b = 0$ จะสามารถเขียนเป็นสมการการแปลงเวฟเล็ดได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$W_g f(a, b) = W_{e^{i\omega}} f\left(\frac{1}{\omega}, a\right) \quad (2.14)$$

$$W_g f(a, b) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt = F(\omega) \quad (2.15)$$

ซึ่งจะเป็นสมการของการแปลงฟูรีเยร์นั่นเอง ดังนั้นจะเทียบได้ว่าการสเกลจะเป็นส่วนกลับของการเปลี่ยนความถี่ที่จะพิจารณานั้นเอง จากสมการการแปลงฟูรีเยร์จะเห็นว่า $b = 0$ เสมอ และเนื่องจากฟังก์ชันหน้าต่างแบบเอ็กซ์โพเนนเชียลทำการเลื่อนตำแหน่งไม่ได้ เพราะเป็นสัญญาณที่ไม่มีจุดเริ่มต้นและจุดสิ้นสุด $(-\infty, \infty)$ จึงทำการเลื่อนตำแหน่งไม่ได้ต่างกับฟังก์ชันหน้าต่างของการแปลงเวฟเล็ต ที่มีจุดเริ่มต้นและสิ้นสุด หรือมีขอบเขตที่แน่นอน (Compact support) ดังนั้น จึงสามารถยกไปวางตำแหน่งใดก็ได้ ทำให้สามารถเลือกตำแหน่งที่จะทำการพิจารณาสัญญาณได้

2.3 การแปลงเวฟเล็ตแบบแบ่งช่วง (Discrete wavelet transform : DWT)

ในเนื้อหาส่วนนี้จะกล่าวถึงการนำการแปลงเวฟเล็ต (Wavelet transform : WT) แบบแบ่งช่วง (Discrete) มาวิเคราะห์สัญญาณอินพุตที่มีลักษณะเป็นข้อมูลการสุ่ม (Sample data) โดยเริ่มจากคุณสมบัติที่สำคัญของการแปลงเวฟเล็ตที่เกี่ยวกับการวิเคราะห์สัญญาณที่สามารถสื่อระดับความละเอียดได้ (Multiresolution analysis : MRA) จากนั้นจะพิจารณาการแปลงเวฟเล็ตแบบแบ่งช่วง ในลักษณะฟิลเตอร์แบงก์ (filter banks) ที่จะกล่าวถึงการกำหนดการตอบสนองอิมพัลส์ ของตัวฟิลเตอร์ที่นำมาใช้ในกระบวนการแปลงเวฟเล็ตแบบแบ่งช่วง

2.3.1 ปริภูมิเวกเตอร์ในการวิเคราะห์สัญญาณ

พิจารณา a, b เป็นจำนวนจริงใด ๆ และ \hat{i}, \hat{j} เป็นเวกเตอร์หนึ่งหน่วย (Unit vector) ในปริภูมิเวกเตอร์ (Vector space : V) ดังนั้น ถ้า $a\hat{i} + b\hat{j} \in V$ แล้วจะได้ว่า เวกเตอร์ใด ๆ ที่ประกอบขึ้นจากเวกเตอร์หนึ่งหน่วยหนึ่งตัวหรือหลายตัว แล้วเวกเตอร์นั้นยังคงอยู่ในปริภูมิเวกเตอร์นี้ (V) ดังนั้นเวกเตอร์หนึ่งหน่วยดังกล่าวจะเรียกว่าเป็น เวกเตอร์พื้นฐาน (Basis vector) แต่เราจะพิจารณาเวกเตอร์เป็นฟังก์ชัน นั่นคือจะเรียกเวกเตอร์พื้นฐานนี้ใหม่ว่า ฟังก์ชันพื้นฐาน (Basis function : b.f.) ถ้าฟังก์ชันพื้นฐานของเวกเตอร์ทำผลคูณภายใน (inner product) กัน เช่น \hat{i}, \hat{j} แล้วมีค่าเป็นศูนย์ จะได้ว่าฟังก์ชันพื้นฐานดังกล่าวจะแสดงคุณสมบัติออร์ทอโกนัล (orthogonal) ซึ่งกันและกัน และถ้าฟังก์ชันพื้นฐานของเวกเตอร์ทำผลคูณภายในตัวเองแล้วมีค่าเป็นหนึ่งฟังก์ชันพื้นฐานที่มีคุณสมบัติทั้งสองก็จะเรียกว่าออร์ทอนอโมล (orthonormal : o.n.)

ถ้าพิจารณาในลักษณะของสัญญาณ อาจกล่าวได้ว่า ฟังก์ชันพื้นฐานเป็นสัญญาณพื้นฐานเล็ก ๆ ที่สามารถนำมาประกอบกันเป็นปริภูมิของสัญญาณใด ๆ ดังนั้น ถ้าให้ U_i เป็นฟังก์ชันพื้นฐานที่ออร์ทอนอโมล (orthonormal basis function : o.n.b.f.) ใด ๆ และ C_i เป็นค่าสัมประสิทธิ์ของ U_i จะได้ว่า สัญญาณ v

ใด ๆ นั้น จะสามารถประกอบขึ้นจากฟังก์ชันพื้นฐานที่ออร์ทอนอโมลที่มีค่าขนาดต่าง ๆ มารวมกันดังสมการ

$$v = \sum_{i=-\infty}^{\infty} C_i U_i \quad v \in V \quad (2.16)$$

2.3.2 การวิเคราะห์แบบหลายระดับความละเอียดของมอร์เลต (Mallat)

สมมติให้ V_j เป็นปริภูมิเวกเตอร์ที่มี j แสดงถึงระดับความละเอียดและจำนวนของฟังก์ชันพื้นฐานที่ประกอบขึ้นเป็นฟังก์ชันนั้น จากหลักการของมอร์เลต ที่ได้อธิบายลักษณะคุณสมบัติของการวิเคราะห์สัญญาณที่ระดับความละเอียดต่าง ๆ ได้สรุปลักษณะคุณสมบัติได้ดังนี้

- MRA 1) $V_{-\infty} \dots \subset V_{-1} \subset V_0 \subset V_1 \dots \subset V_{\infty}$
- MRA 2) $Close_{L^2}(\bigcup_{j \in ZZ} V_j) = L^2(IR)$; $j \in ZZ$ (เซตของจำนวนเต็ม),
 IR เป็นเซตของจำนวนจริง
- MRA 3) $(\bigcap_{j \in ZZ} V_j) = \{0\}$
- MRA 4) $V_j * W_j = V_{j+1}$; $j \in ZZ$
- MRA 5) $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$; $j \in ZZ$

จากลักษณะคุณสมบัติทั้ง 5 ข้อ ถ้ากำหนดให้ $f(t) \in S$ เมื่อ S เป็นปริภูมิเวกเตอร์ของสัญญาณ เมื่อพิจารณาเทียบกับ MRA 2) จะได้ว่า สัญญาณ $f(t)$ สามารถประมาณให้อยู่ในปริภูมิเวกเตอร์ที่ระดับความละเอียด j ใด ๆ ได้โดยถ้าระดับความละเอียดเพิ่มขึ้นถึงอนันต์สัญญาณที่ประมาณนั้น ยังคงอยู่เข้าสู่สัญญาณต้นกำเนิด นั่นคือ $f(t) \in L^2(IR)$ และในทางตรงกันข้าม ถ้าระดับความละเอียดที่ระดับ j น้อยลงเรื่อย ๆ จะทำให้สัญญาณที่ประมาณมีข่าวสารสูญหายไปมากขึ้น ตามลักษณะของ MRA 3)

พิจารณาปริภูมิเวกเตอร์ที่ระดับความละเอียด j จะมีฟังก์ชันพื้นฐานที่จะประกอบกันเป็นสัญญาณการประมาณของ $f(t)$ เมื่อ $f(t) \in V_j$ ซึ่งเรียกฟังก์ชันพื้นฐานเหล่านี้ว่า ฟังก์ชันสเกลลิ่ง (Scaling function) ซึ่งฟังก์ชันสเกลลิ่ง จัดเป็นสัญญาณพื้นฐานย่อย ๆ ที่จะเกิดที่ตำแหน่งที่เวลาต่าง ๆ กันภายใน V_j และมีความถี่ต่างกันภายใน V_j เดียวกัน ดังนั้นสามารถกำหนดเซตของฟังก์ชันสเกลลิ่ง ในเทอมของการเลื่อนตำแหน่งเป็นจำนวนเต็มของฟังก์ชันสเกลลิ่งพื้นฐานที่ปริภูมิ V_0 ได้คือ

$$\phi_k(t) = \phi(t - k) \quad (2.17)$$

เมื่อ $k \in ZZ$ และ $\phi \in L^2$

โดยที่ k จะเป็นตำแหน่งการเกิดของ $\phi(t)$ แต่ละตัว และปริภูมิย่อยของ $L^2(\mathbb{R})$ ถูกแผ่คลุม (span) โดยฟังก์ชันสเกลลิ่งเหล่านี้ที่กำหนดโดย

$$V_0 = \text{span}\{\phi_k(t)\} \quad (2.18)$$

สำหรับทุกจำนวนเต็ม k จาก ∞ ถึง $-\infty$

ดังนั้น จะสามารถประมาณฟังก์ชันสัญญาณ $f(t)$ ภายในปริภูมิ V_0 ได้คือ

$$f(t) = \sum_k a_k \phi_k(t) \quad ; f(t) \in V_0 \quad (2.19)$$

เนื่องจากความถี่จะเปลี่ยนแปลงขึ้นอยู่กับระดับความละเอียดของปริภูมิ j นั้น ๆ โดยอาศัยลักษณะคุณสมบัติที่ MRA 5) จะได้ว่า

$$f(x) \in V_0 \Leftrightarrow f(2x) \in V_1 \quad (2.20)$$

เมื่อ $j = 0$

นั่นคือที่ระดับ V_1 มีความถี่เป็นสองเท่าของความถี่ที่ระดับ V_0 ลักษณะดังกล่าวก็คือการสเกลทางเวลาของสเกลลิ่งฟังก์ชัน ดังนั้นเซตของสเกลลิ่งฟังก์ชันที่เกิดจากการทำสเกล และเลื่อนตำแหน่ง ของฟังก์ชันสเกลลิ่งพื้นฐาน ที่ระดับ j และ $j+1$ ใด ๆ เขียนเป็นสมการได้คือ

$$\phi_{j,k}(t) = \text{const} \times \phi(2^j t - k) \quad (2.21)$$

const เป็นค่าคงที่ เพื่อทำการนอร์มอลไลซ์ฟังก์ชันสเกลลิ่งที่เกิดจากการสเกล (scaling) จากคุณสมบัติออร์ทอกอนัล ซึ่งกันและกันของฟังก์ชันสเกลลิ่ง ที่พิจารณาจาก

$$\langle \phi_{j,k}, \phi_{j,l} \rangle = \int \phi_{j,k}^*(t) \phi_{j,l}(t) dt = 0 \quad (2.22)$$

เมื่อ $k \neq l$

จากสมการที่ (2.22) เมื่อ $k = l$ ก็จะเป็นการหาขนาดหรือนอร์มอลไลซ์ ของตัวเอง ถ้าขนาดเท่ากับหนึ่งจะได้ว่า

$$\int \{\phi_{j,k}(t)\}^2 dt = 1 \quad (2.23)$$

จากสมการที่ (2.21) นำมาแทนในสมการที่ (2.23) เพื่อหาค่า const . จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\int const^2 \phi^2(2^j t - k) dt = 1$$

$$2^{-j} const^2 \int \phi^2(2^j t - k) d(2^j t) = 1$$

เนื่องจากสมการที่ (2.23) จะได้

$$2^{-j} const^2 = 1$$

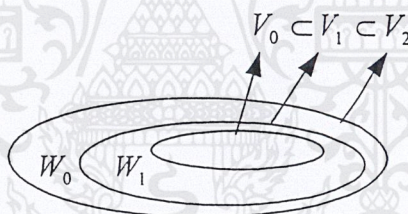
ดังนั้น

$$const = 2^{j/2}$$

จากสมการที่ (2.21) สามารถเขียนได้ใหม่คือ

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k) \tag{2.24}$$

จากลักษณะคุณสมบัติ MRA 1) สามารถแสดงกลุ่มของปริภูมิเวกเตอร์ได้ดังรูปที่ 2.11



รูปที่ 2.11 แสดงกลุ่มของปริภูมิเวกเตอร์ที่ถูกแผ่คลุมโดยสเกลลิ่งฟังก์ชัน

จากรูปที่ 2.11 $V_0 \subset V_1$ จะได้ว่าสเกลลิ่งฟังก์ชันในปริภูมิ V_0 เป็นสับเซตของสเกลลิ่งฟังก์ชันในปริภูมิ V_1 ดังนั้น สเกลลิ่งฟังก์ชันในปริภูมิ V_0 สามารถเขียนได้เหมือนกับการรวมเชิงเส้นของฟังก์ชันพื้นฐาน $\sqrt{2}\phi(2t - k)$ ของ $V_1; j = 1$ นั่นคือ

$$\phi(t) = \sum_k h(k) \sqrt{2}\phi(2t - k) \tag{2.25}$$

$h(k)$ เป็นเซตสัมประสิทธิ์สเกลลิ่งฟังก์ชัน หาได้จาก $\langle \phi(t), \sqrt{2}\phi(2t - k) \rangle$ และสมการที่ (2.25) จะเรียกว่า สมการรีเคอร์ซีฟ (recursive equation) หรือบางครั้งอาจเรียกว่า สมการรีไฟน์เมนต์ (refinement equation) หรือสมการไดเลชัน (dilation equation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.11 ภายในปริภูมิ V_1 จะประกอบด้วยปริภูมิ V_0 และปริภูมิที่เป็นส่วนประกอบออร์ทอกอนัล (orthogonal complement) ของ V_0 นั่นคือ

$$V_1 = V_0 * W_0 \tag{2.26}$$

เมื่อพิจารณาลักษณะคุณสมบัติตาม MRA 4) จะได้ว่า

$$V_2 = V_1 * W_1 \tag{2.27}$$

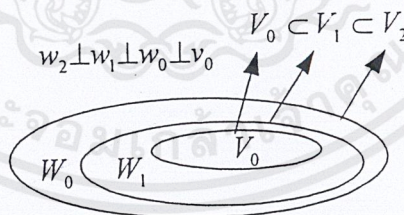
เมื่อนำสมการที่ (2.26) มาแทนในสมการที่ (2.27)

$$V_2 = V_0 * W_0 * W_1 \tag{2.28}$$

ฉะนั้นโดยทั่วไปตามลักษณะคุณสมบัติ MRA 2) จะได้ว่า

$$L^2 = V_0 * W_0 * W_1 * \dots \tag{2.29}$$

นั่นคือ ภายในปริภูมิของ V_{j+1} จะประกอบด้วยปริภูมิ V_j และปริภูมิ W_j โดยที่สมาชิกในปริภูมิ W_j ซึ่งจะออร์ทอกอนัลกับสเกลลิงฟังก์ชันในปริภูมิ V_j เรียกว่าเวฟเลตฟังก์ชัน (Wavelet function : $\psi(t)$)
 ดังรูปที่ 2.12



รูปที่ 2.12 แสดงปริภูมิเวกเตอร์ของสเกลลิงฟังก์ชันและเวฟเลตฟังก์ชัน

สมการที่ (2.29) จะเห็นว่า V_0 เป็นปริภูมิเริ่มต้นที่ถูกแผ่คลุมด้วยสเกลลิงฟังก์ชัน $\phi(t - k)$ ดังนั้น เมื่อพิจารณาร่วมกับ MRA 1) สเกลของปริภูมิเริ่มต้นอาจจะเริ่มที่หรือเลือกที่ระดับความละเอียดที่สูง เช่น $j = 10$

$$L^2 = V_{10} * W_{10} * W_{11} * \dots \tag{2.30}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือระดับความละเอียดต่ำ เช่น $j = -5$

$$L^2 = V_{-5} * W_{-5} * W_{-4} * \dots \quad (2.31)$$

หรือที่ $j = \infty$

$$L^2 = \dots * W_{-2} * W_{-1} * \dots * W_0 * W_1 * W_2 * \dots \quad (2.32)$$

จากสมการที่ (2.32) เทียบกับสมการที่ (2.29) จะได้

$$W_{-\infty} * \dots * W_{-1} = V_0 \quad (2.33)$$

เนื่องจาก $f(t) \in L^2(\mathbb{R})$ จะได้ว่า $f(t)$ จะประกอบด้วยเวฟเล็ตฟังก์ชันในปริภูมิ W_j นั่นคือสามารถประมาณฟังก์ชัน $f(t)$ ได้คือ

$$f(t) = \sum_{j,k} a_{j,k} \psi_{j,k}(t) \quad (2.34)$$

โดย $a_{j,k}$ เป็นสัมประสิทธิ์ที่จะเรียกว่า ดิสครีตเวฟเล็ตทรานส์ฟอร์ม (Discrete wavelet transform : DWT) ของ $f(t)$ ที่สามารถคำนวณหาได้จากการใช้ ผลคูณภายใน คือ

$$a_{j,k} = \langle \psi_{j,k}(t), f(t) \rangle \quad (2.35)$$

ส่วน $\psi_{j,k}(t)$ คือ เวฟเล็ตฟังก์ชัน ที่ได้จากการสเกล j และการเลื่อนตำแหน่ง k ของเวฟเล็ตแม่ (Mother wavelet : $\psi(t)$)

เมื่อพิจารณาจากสมการ ปริภูมิ W_0 จะมีคุณสมบัติต่าง ๆ เหมือนกับ ปริภูมิ V_0 เพียงแต่จะเป็นส่วนประกอบย่อยที่ประกอบอยู่ในปริภูมิ V_1 ดังนั้น ที่ระดับความละเอียดใด ๆ (j) และพิจารณาคุณสมบัติเหมือนสเกลลิงฟังก์ชัน จะได้สมการที่คล้ายกับสมการที่ (2.24) คือ

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad (2.36)$$

เนื่องจาก เวฟเล็ตเหล่านี้จะอยู่ในปริภูมิที่แผ่คลุมโดยสเกลลิงฟังก์ชันถัดไป นั่นคือ $W_0 \subset V_1$ และในทำนองเดียวกันกับสมการที่ (2.25) และสมการที่ (2.36) สามารถเขียนเวฟเล็ตฟังก์ชันเป็นลักษณะผลบวกเชิงเส้นของสเกลลิงฟังก์ชันพื้นฐาน (basis scaling function) ของปริภูมิ V_1 คือ

$$\psi(t) = \sum_k h_1(k) \sqrt{2} \phi(2t - k) \quad ; k \in \mathbb{Z} \quad (2.37)$$

เป็นเซตสัมประสิทธิ์เวฟเลตฟังก์ชัน โดยมีความสัมพันธ์กับเซตสัมประสิทธิ์สเกลลิงฟังก์ชัน ($h(k)$) จากลักษณะคุณสมบัติ MRA 4) จะได้ว่า

$$\langle \phi_{j,k}(t), \psi_{j,l}(t) \rangle = \int \phi_{j,k}^*(t) \psi_{j,l}(t) dt = 0 \quad (2.38)$$

เมื่อ $j, k, l \in \mathbb{Z}$

จากสมการที่ (2.38) จะได้ความสัมพันธ์ของสัมประสิทธิ์เวฟเลต ที่ออร์ทอกอนัลกับสัมประสิทธิ์สเกลลิง ดังสมการที่ (2.39)

$$h_1(k) = (-1)^k h(N-1-k) \quad (2.39)$$

เมื่อ N เป็นจำนวนข้อมูลของเซตสัมประสิทธิ์สเกลลิง ($h(k)$)

จากสมการที่ (2.29) เราสามารถสร้างฟังก์ชันใด ๆ จากเซตของฟังก์ชัน $\phi_k(t)$ และ $\psi_{j,k}(t)$ แต่ละกลุ่ม $L^2(\mathbb{R})$ ทั้งหมด สำหรับฟังก์ชันใดๆ $g(t) \in L^2(\mathbb{R})$ จะได้ว่า

$$g(t) = \sum_{k=-\infty}^{\infty} c(k) \phi_k(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d(j,k) \psi_{j,k}(t) \quad (2.40)$$

ที่จะเหมือนกับการกระจายอนุกรมในเทอมของฟังก์ชันสเกลลิงและเวฟเลตจากสมการที่(2.40)ในการกระจายอนุกรม $g(t)$ เทอมแรกของสมการ เป็นผลรวมแสดงถึงความละเอียดต่ำ หรือการประมาณที่หยาบ (coarse approximation) ของ $g(t)$ สำหรับแต่ละการเพิ่มของ j ในเทอมที่สองจะแสดงถึงความละเอียดที่สูงขึ้นและทำให้รายละเอียดของข้อมูล (detail) เพิ่มขึ้น เหมือนกับอนุกรมฟูรีเยร์ โดยสมการที่ (2.40) ดังกล่าวจะเลือกปริภูมิสเกลลิงฟังก์ชันที่สเกล j ใดก็ได้ (สมการที่ (2.41) ใช้ $j = 0$ กำหนดกลุ่มของสเกลลิงฟังก์ชัน) คือ

$$g(t) = \sum_k c_{j_0}(k) \phi_{j_0,k}(t) + \sum_k \sum_{j=j_0}^{\infty} d_j(k) \psi_{j,k}(t) \quad (2.41)$$

จากสมการที่ (2.41) สัมประสิทธิ์ทั้งสองเทอมสามารถคำนวณจากผลคูณภายใน คือ

$$c_j(k) = \langle g(t), \phi_{j,k}(t) \rangle \quad (2.42)$$

$$d_j(k) = \langle g(t), \psi_{j,k}(t) \rangle \quad (2.43)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัมประสิทธิ์ในการกระจายของเวฟเล็ต จะถูกเรียกว่า ดิสครีตเวฟเล็ตทรานส์ฟอร์ม ของ $g(t)$

2.3.3 การแปลงเวฟเล็ตโดยใช้หลักการของการวิเคราะห์ฟิลเตอร์แบงก์

ในการประยุกต์ ตัวสัมประสิทธิ์ $h(n)$ และ $h_1(n)$ ที่กำหนดในสมการที่ (2.25) และ (2.27) และสัมประสิทธิ์ $c_j(k)$ และ $d_j(k)$ ในสมการที่ (2.42) และ (2.43) สามารถพิจารณามองเหมือนกับดิจิทัลฟิลเตอร์ (digital filter) และสัญญาณดิจิทัล (digital signal) ตามลำดับ ซึ่งทฤษฎีเวฟเล็ตนี้สามารถจะพัฒนาและศึกษาโดยการวิเคราะห์ใช้ฟิลเตอร์แบงก์ โดยลักษณะของฟิลเตอร์แบงก์ อาจเปรียบเสมือนการแยกองค์ประกอบของสัญญาณ

การวิเคราะห์จากระดับที่ความละเอียดสูง (fine scale) ไปยังความละเอียดต่ำ (coarse scale)
จากสมการรีเคอร์ซีฟ สมการที่ (2.25)

$$\phi(t) = \sum_n h(n) \sqrt{2} \phi(2t - n) \quad (2.44)$$

ถ้าทำการสเกลและเลื่อนตำแหน่งของตัวแปรเวลาจะได้ว่า

$$\begin{aligned} \phi(2^j t - k) &= \sum_n h(n) \sqrt{2} \phi(2^j t - n) \\ &= \sum_n h(n) \sqrt{2} \phi(2^{j+1} t - 2k - n) \end{aligned} \quad (2.45)$$

จากสมการที่ (2.45) ทำการเปลี่ยนตัวแปรโดยให้ $m = 2k + n$ จะได้

$$\phi(2^j t - k) = \sum_n h(m - 2k) \sqrt{2} \phi(2^{j+1} t - m) \quad (2.46)$$

เมื่อพิจารณา V_j ในทำนองเดียวกับ สมการที่ (2.18) จะได้ว่า

$$V_j = \text{span} \left\{ 2^{j/2} \phi(2^j t - k) \right\} \quad (2.47)$$

ดังนั้น เมื่อ $f(t) \in V_{j+1}$ แล้ว

$$f(t) = \sum_k c_{j+1}(k) 2^{(j+1)/2} \phi(2^{j+1} t - k) \quad (2.48)$$

จากสมการที่ (2.48) พิจารณาที่สเกลของ $j+1$ และ $f(t)$ เกิดจากสเกลลิงฟังก์ชันอย่างเดียวกันไม่มีเวฟเล็ทฟังก์ชัน ดังนั้นที่สเกลความละเอียดค่าเวฟเล็ทฟังก์ชันจำเป็นสำหรับ

$$f(t) = \sum_k c_{j+1}(k) 2^{j/2} \phi(2^j t - k) + \sum_k d_j(k) 2^{j/2} \psi(2^j t - k) \quad (2.49)$$

ถ้า $\phi_{j,k}(t)$ และ $\psi_{j,k}(t)$ เป็นออร์ทอกรีนอรัลกัน จะได้ว่าสัมประสิทธิ์สเกลลิงที่ระดับ j ($c_j(k)$) จะหาได้โดยผลคูณภายในคือ

$$c_j(k) = \langle f(t), \phi_{j,k}(t) \rangle \quad (2.50)$$

เมื่อนำสมการที่ (2.46) แทนลงในสมการที่ (2.50) จะได้

$$\begin{aligned} c_j(k) &= \langle f(t), 2^{j/2} \phi(2^j t - k) \rangle \\ &= \langle f(t), 2^{j/2} \sum_m h(m-2k) \sqrt{2} \phi(2^{j+1} t - m) \rangle \\ &= \sum_m h(m-2k) \langle f(t), 2^{j+1/2} \phi(2^{j+1} t - m) \rangle \end{aligned} \quad (2.51)$$

จากสมการที่ (2.50) เมื่อเทียบกับสมการที่ (2.51)

$$c_j(k) = \sum_m h(m-2k) c_{j+1}(m) \quad (2.52)$$

ในการทำงานเดียวกันกับกระบวนการตามสมการที่ (2.52) สำหรับสัมประสิทธิ์เวฟเล็ทคือ

$$d_j(k) = \sum_m h_1(m-2k) c_{j+1}(m) \quad (2.53)$$

การฟิลเตอร์ และการลดอัตราการซีกค่าตัวอย่างลง (Down Sampling)

ในรูปแบบการประมวลผลสัญญาณดิจิทัล, การฟิลเตอร์สัญญาณที่เป็นลำดับ จะใช้การทำคอนโวลูชันระหว่างลำดับ (Sequence) กับเซตของจำนวนสัมประสิทธิ์ตัวฟิลเตอร์

เมื่อพิจารณาสำหรับลำดับของอินพุต (input sequence; $x(n)$) และสัมประสิทธิ์ฟิลเตอร์ (filter coefficient; $h(n)$) จะได้ลำดับของเอาต์พุต (output sequence; $y(n)$) โดย

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (2.54)$$

หรือเขียนในลักษณะการคอนโวลูชัน

$$y(n) = h(n) * x(n) \quad (2.55)$$

ดังนั้น จากสมการที่ (2.52) ทำให้อยู่ในลักษณะการคอนโวลูชันได้ดังสมการ (2.56)

$$c_j(k) = \sum_m h(m-2k)c_{j+1}(m) \quad (2.56)$$

ให้ $m - 2k = -x$ ดังนั้น $m = 2k - x$ แทนค่าในสมการที่ (2.56)

$$c_j(k) = \sum_m h(-x)c_{j+1}(2k-x) \quad (2.57)$$

จากสมการที่ (2.54) และ (2.55) ดังนั้นจัดรูปสมการที่ (2.57) จะได้

$$c_j(k) = \sum_m h(-2k) * c_{j+1}(2k) \quad (2.58)$$

หรือ

$$c_j(k) = h(-n) * c_{j+1}(n) \quad \text{เมื่อ} \quad n=2k \quad (2.59)$$

และในทำนองเดียวกันกับสมการที่ (2.59) สามารถเขียนสมการที่ (2.53) ในลักษณะการคอนโวลูชันได้เป็น

$$d_j(k) = h(-n) * c_{j+1}(n) \quad \text{เมื่อ} \quad n=2k \quad (2.60)$$

จากสมการที่ (2.59) และ (2.60) สัมประสิทธิ์ $c_j(k)$ และ $d_j(k)$ จะได้จากสัมประสิทธิ์ $c_{j+1}(2k)$ ดังนั้นเมื่อพิจารณาเทียบกับหลักการฟิลเตอร์เบงก์ กับการฟิลเตอร์สัญญาณดิจิทัลแล้ว เซตสัมประสิทธิ์ของ $h(-n)$ และ $h_1(-n)$ เทียบได้กับ สัมประสิทธิ์ของตัวฟิลเตอร์ทางดิจิทัล และตัวชี้ (index) ที่แสดงถึงลำดับของสัญญาณ ของ $c_j(k)$ และ $d_j(k)$ ได้จากการคอนโวลูชันกับ $h(-n)$ และ $h_1(-n)$ แล้วชักค่าตัวอย่างลงสองเท่า เรียกว่าการลดค่าการชักค่าตัวอย่าง หรือ อาจเรียกว่าดีซิเมเตอร์ (Decimator) ดังรูปที่ 2.13

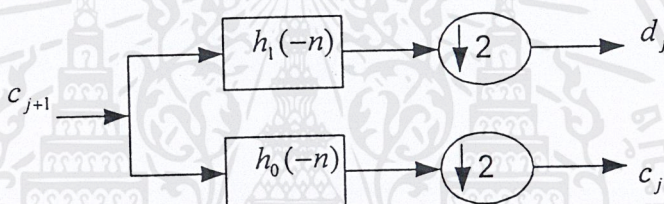
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงการลดอัตราการซีกค่าตัวอย่างลงครึ่งหนึ่ง

นอกจากการทำกรลดค่าการซีกค่าตัวอย่างจะทำให้เกิดการสูญหายของข้อมูลเนื่องจากครึ่งหนึ่งของข้อมูลต้องถูกตัดออกไป แต่ถ้าสัญญาณที่พิจารณาเป็นสัญญาณที่มีขอบเขตจำกัดจะไม่มีกรสูญเสียบของข่าวสารที่เกิดจากการลดค่าการซีกค่าตัวอย่าง

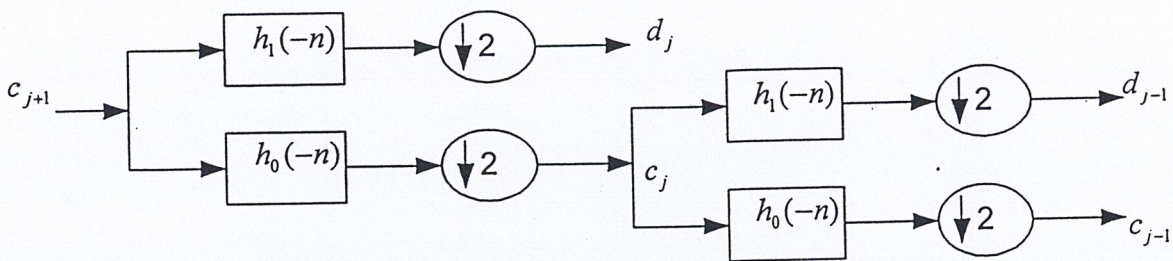
ดังนั้น ตัวกรองความถี่แบบดิจิทัล และการลดอัตราการซีกค่าตัวอย่าง จะนำมาใช้กับสมการและตามหลักการของฟิลเตอร์เบงค์ ของสัญญาณ ดังรูปที่ 2.14



รูปที่ 2.14 แสดงการวิเคราะห์สัญญาณด้วย two band

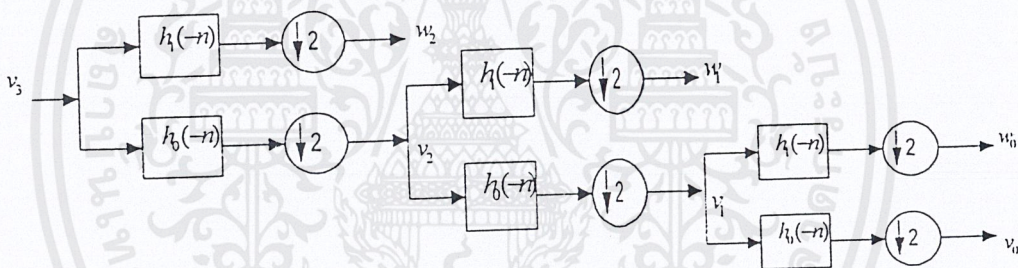
ถ้าจำนวนของตัวสัมพันธ์ฟิลเตอร์มีจำนวนจำกัดขนาด N แล้ว ตัวฟิลเตอร์จะเรียกว่า กรตอบสนองอิมพัลส์ที่จำกัด (finite impulse response ; FIR) และจะเรียกว่า กรตอบสนองอิมพัลส์ไม่จำกัด (infinite impulse response ; IIR) ถ้าจำนวนของตัวสัมพันธ์ฟิลเตอร์มีจำนวนไม่จำกัด

จากรูปที่ 2.14 แสดงการกรองสัญญาณหรือแยกสัญญาณ (Decomposition) c_{j+1} ที่ระดับ $j + 1$ ไปสู่สัมพันธ์ c_j และ d_j ที่ระดับ j ดังนั้น ถ้านำฟิลเตอร์เบงค์ดังกล่าวมาแยกสัญญาณ (สัมพันธ์สเกลลง) อีก เพื่อให้ได้สเกลที่หยาบอีก เช่น สเกล $j - 1$ ดังรูปที่ 2.15



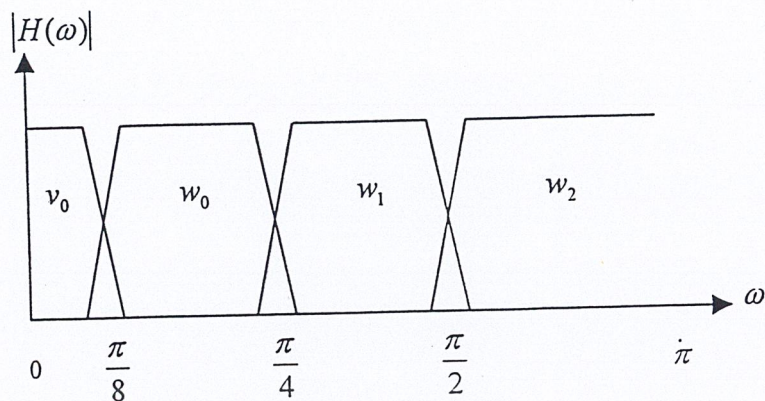
รูปที่ 2.15 แสดงการแตกกระจายสัญญาณจากระดับ $j+1$ ไปยังระดับ $j-1$

สำหรับในทางปฏิบัติสัญญาณใดที่มีช่วงของสัญญาณจำกัด จะมีระดับสูงสุดที่ $j = J$ ด้วยการเริ่มต้นอธิบายสัญญาณที่ระดับความละเอียดสูง (high resolution) ในเทอมของสัมประสิทธิ์ c_j และจะวิเคราะห์หลังไปสู่ระดับความละเอียดต่ำ (low resolution) $j = j_0$ ดังนั้นจะต้องใช้ขั้นตอนการแยกสัญญาณ $J - j_0$ ขั้นตอน เช่นในปริภูมิ V_3



รูปที่ 2.16 แสดงการแตกกระจายสัญญาณใน V_3 ไปสู่สเกล $j = 0$

เนื่องจากในปริภูมิ V_3 จะถูกแตกกระจายสัญญาณไปสู่ระดับ $j = 2$ ทั้งในปริภูมิของสเกลสูง (V_2) และปริภูมิของเวฟเลต (W_2) ซึ่งในปริภูมิเวกเตอร์ที่สเกล $j = 2$ จะมีความละเอียดที่สูงกว่าปริภูมิที่สเกล $j = 1$ เมื่อพิจารณาในลักษณะทางด้านความถี่ของสัญญาณ ระดับความละเอียดที่สูงเปรียบได้กับสัญญาณที่มีความถี่สูง ดังนั้น จากรูปที่ 2.16 ตัวฟิลเตอร์ของสัมประสิทธิ์ $h_1(n)$ จะกรองแยกสัญญาณความถี่สูง ที่ระดับ $j = 2$ ออกจาก V_3 และตัวฟิลเตอร์ของสัมประสิทธิ์ $h_0(n)$ ก็จะกรองแยกสัญญาณความถี่ต่ำออกจาก V_3 ด้วย นั่นคือ $h_1(n)$ จะเป็นการกรองความถี่สูง (highpass filter) และ $h_0(n)$ เป็นการกรองความถี่ต่ำ (lowpass filter) จากรูปที่ 2.16 เมื่อแสดงในลักษณะการวิเคราะห์ทางด้านความถี่ จะแสดงได้ดังรูปที่ 2.17



รูปที่ 2.17 แสดงแถบความถี่จากการผ่านการกรองของ $h_0(n)$ และ $h_1(n)$

ดังนั้นสำหรับ $f(t) \in V_J$ และจากสมการ จะได้ว่า

$$f(t) = \sum_k c_J(k) \phi_{J,k}(t) \quad (2.61)$$

$$f(t) = \sum_k c_{J-1}(k) \phi_{j-1,k}(t) + \sum_k d_{J-1}(k) \psi_{J-1,k}(t) \quad (2.62)$$

$$f(t) = \sum_k c_{J-2}(k) \phi_{j-2,k}(t) + \sum_k \sum_{j=J-2}^{J-1} d_{j-2}(k) \psi_{j,k}(t) \quad (2.63)$$

$$f(t) = \sum_k c_{j_0}(k) \phi_{j_0,k}(t) + \sum_k \sum_{j=j_0}^{J-1} d_{j_0}(k) \psi_{j,k}(t) \quad (2.64)$$

สมการที่ (2.64) เป็นบางส่วนของสมการที่ (2.41) ที่จำกัดของเขตของระดับ

การสังเคราะห์ที่ความละเอียดต่ำไปยังความละเอียดสูง

การแตกกระจายสัญญาณไปสู่ระดับต่างๆ ดังรูปที่ 2.16 เมื่อต้องการสร้างกลับคืน (reconstruction) ไปสู่สัญญาณที่ระดับที่ละเอียดเหมือนเดิม จะสามารถสร้างได้จากการรวม (combination) ของฟังก์ชันสเกลลิงและสัมประสิทธิ์เวฟเล็ตที่ความละเอียดต่ำโดยการพิจารณาสัญญาณในปริภูมิของฟังก์ชันระดับคือ $j+1$ ($f(t) \in V_{j+1}$) ซึ่งฟังก์ชันของสัญญาณสามารถเขียนในเทอมของสเกลลิงฟังก์ชันเหมือนสมการที่ (2.48)

$$f(t) = \sum_k c_{j+1}(k) 2^{j+1/2} \phi(2^{j+1}t - k) \quad (2.65)$$

หรือในเทอมที่ประกอบด้วยฟังก์ชันเวฟเล็ต เหมือนสมการที่ (2.49)

$$f(t) = \sum_k c_{j+1}(k) 2^{j/2} \phi(2^j t - k) + \sum_k d_j(k) 2^{j/2} \psi(2^j t - k) \quad (2.66)$$

โดยการแทนสมการที่ (2.45) และสมการที่ (2.37) ลงในสมการที่ (2.66)

$$\begin{aligned} f(t) &= \sum_k c_j(k) \sum_n h_0(n) 2^{j+1/2} \phi(2^{j+1} t - 2k - n) \\ &\quad + \sum_k d_j(k) \sum_n h_1(n) 2^{j+1/2} \psi(2^{j+1} t - 2k - n) \end{aligned} \quad (2.67)$$

จากสมการที่ (2.65) สามารถหาค่า $c_{j+1}(k)$ จากการคูณด้วย $2^{j+1/2} \phi(2^{j+1} t - k)$

$$c_{j+1}(k) = \langle f(t), \phi_{j+1,k}(t) \rangle \quad (2.68)$$

จากสมการที่ (2.67) จัดแทนตัวแปรใหม่ด้วยตัวแปร m คือ

$$\begin{aligned} f(t) &= \sum_m c_j(m) \sum_n h_0(n) 2^{j+1/2} \phi(2^{j+1} t - 2m - n) \\ &\quad + \sum_m d_j(m) \sum_n h_1(n) 2^{j+1/2} \psi(2^{j+1} t - 2m - n) \end{aligned} \quad (2.69)$$

จากการสมการที่ (2.69) สามารถสร้างกลับคืนค่าสัมประสิทธิ์ $c_{j+1}(k)$ ได้เมื่อกำหนดให้ $2m + n = k$ นั้น
คือ

$$\begin{aligned} f(t) &= \sum_m c_j(m) \sum_k h_0(k - 2m) 2^{j+1/2} \phi(2^{j+1} t - k) \\ &\quad + \sum_m d_j(m) \sum_n h_1(k - 2m) 2^{j+1/2} \psi(2^{j+1} t - k) \end{aligned} \quad (2.70)$$

เมื่อพิจารณาผลคูณภายในของสมการที่ (2.70) กับ $2^{j+1/2} \phi(2^{j+1} t - k)$ จะได้ว่า

$$\langle f(t), \phi_{j+1,k}(t) \rangle = \sum_m c_j(m) h_0(k - 2m) + \sum_m d_j(m) h_1(k - 2m) \quad (2.71)$$

เมื่อเทียบสมการที่ (2.68) และ สมการที่ (2.71) แล้วจะได้ว่า

$$c_{j+1}(k) = \sum_m c_j(m) h_0(k - 2m) + \sum_m d_j(m) h_1(k - 2m) \quad (2.72)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การฟิลเตอร์และการเพิ่มอัตราการซีกค่าตัวอย่าง (Up-sampling)

เนื่องจากการวิเคราะห์สัญญาณด้วยการวิเคราะห์ความถี่โดยการแยกช่วงความถี่ออกจากกันทีละสองช่วง จะทำการแตกสัญญาณหรือกรองสัญญาณผ่านตัวฟิลเตอร์ $h_0(-n)$ และ $h_1(-n)$ แล้วทำการลดอัตราการซีกค่าตัวอย่างลง ทำให้ได้สัญญาณที่ประกอบด้วยสเกลลิงฟังก์ชัน และเวฟเล็ตฟังก์ชัน ที่ระดับความละเอียดต่ำลงมา ดังนั้นการสังเคราะห์สร้างสัญญาณกลับคืนมา โดยใช้ลักษณะของการกรองความถี่แยกเป็นช่วง หรือเรียกว่าฟิลเตอร์แบงค์ ในการสังเคราะห์สัญญาณกลับ จะตรงกันข้ามกับการวิเคราะห์สัญญาณด้วยการแยกช่วงความถี่ออกจากกันทีละสองช่วง โดยทำการเพิ่มอัตราการซีกค่าตัวอย่างให้มีค่าเป็นสองเท่าจากเดิมก่อน ซึ่งจะทำการแทรกค่าศูนย์ระหว่างเทอมของสัญญาณ ดังสมการที่ (2.73)

$$\begin{aligned} Y(2n) &= X(n) \\ Y(2n+1) &= 0 \end{aligned} \quad (2.73)$$



รูปที่ 2.18 แสดงการซีกค่าการสุ่มตัวอย่างขึ้นสองเท่า

จากสมการที่ (2.72) พิจารณาผลรวมของแต่ละเทอมในรูปแบบการคอนโวลูชัน ร่วมกับสมการที่ (2.44) และ (2.45) เขียนได้ใหม่เป็น

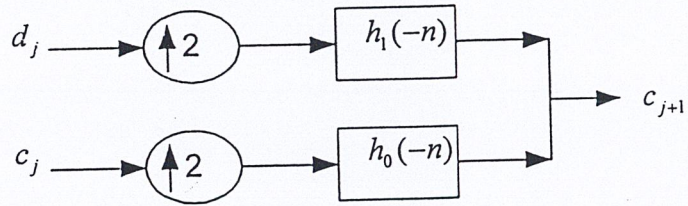
$$c_{j+1}(k) = c_j(k) * h_0(k)|_{k=2m} + d_j(k) * h_1(k)|_{k=2m} \quad (2.74)$$

โดยใช้ตัวแปร n ตัวแปรใหม่แทนตัว k ในสมการที่ (2.74) จะได้

$$c_{j+1}(n) = c_j(n) * h_0(n)|_{n=2m} + d_j(n) * h_1(n)|_{n=2m} \quad (2.75)$$

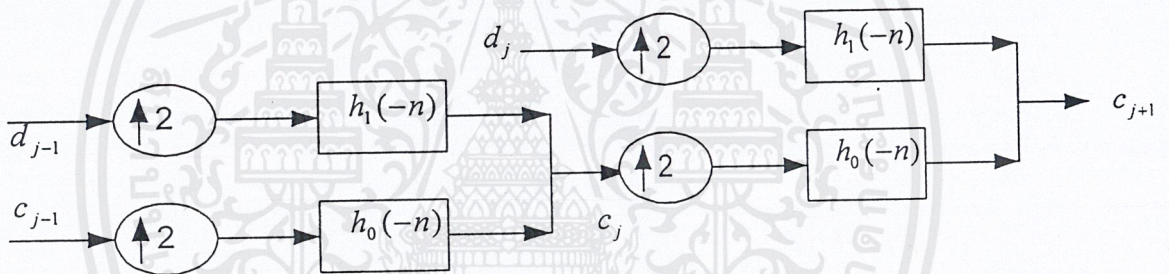
จากสมการที่ (2.75) นำมาสร้างในลักษณะฟิลเตอร์แบงค์ จะได้ดังรูปที่ 2.19 (เมื่อ $g_0(n) = h_0(n)$,

$$g_1(n) = h_1(n))$$



รูปที่ 2.19 แสดงการสังเคราะห์ด้วย Two-band

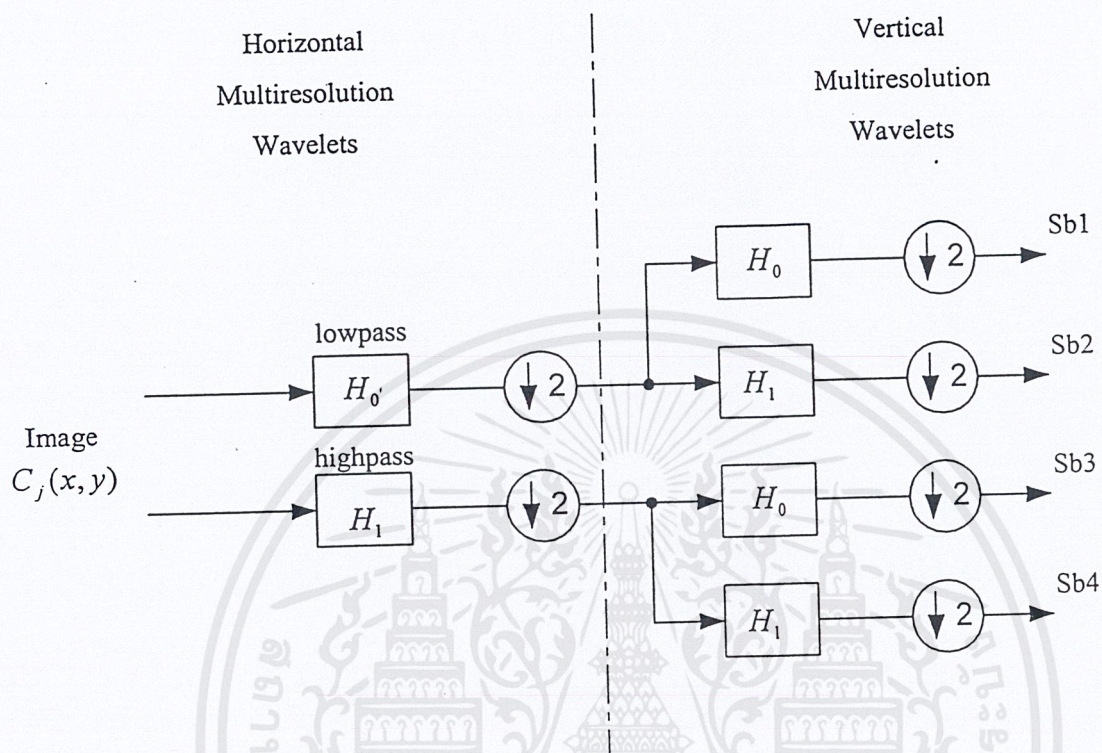
ในทำนองเดียวกันรูปที่ 2.19 สามารถทำการสังเคราะห์สัญญาณจากระดับ $j-1$ ไปสู่ระดับ j ดังรูปที่ 2.20 (เมื่อ $g_0(n) = h_0(n)$, $g_1(n) = h_1(n)$)

รูปที่ 2.20 แสดงการสังเคราะห์สัญญาณจากสเกล $j-1$ ไปสู่ระดับ j

ลักษณะของกระบวนการตามรูปที่ 2.13 ถึง 2.25 จะเรียกว่า การแปลงเวฟเลตแบบไม่ต่อเนื่อง (discrete wavelet transform ; DWT) และลักษณะของกระบวนการตามรูปที่ 2.18 และ 2.19 จะเรียกว่า การแปลงกลับเวฟเลตแบบแบ่งช่วง (inverse discrete wavelet transform ; IDWT)

2.4 การแปลงเวฟเลตกับข้อมูลภาพ

ข้อมูลภาพมีลักษณะเป็นข้อมูลภาพสองมิติ ดังนั้นการทำการแปลงเวฟเลตกับภาพจะเป็นการทำการแปลงเวฟเลตในทางแนวแกน x และแกน y สลับกันในแต่ละครั้ง ซึ่งสามารถเขียนเป็นแผนผัง แสดงได้ดังในรูปที่ 3.2



ก) แสดงลักษณะการแปลงโดยใช้ Analysis filter banks

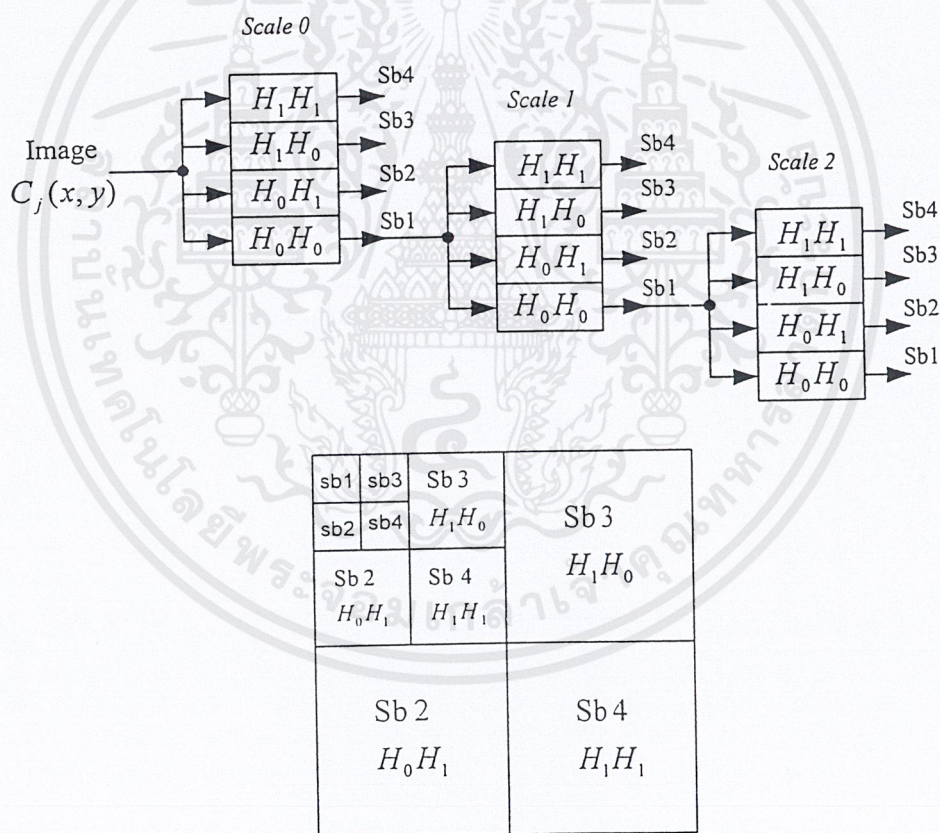
Sb1 H_0H_0	Sb3 H_1H_0
Sb2 H_0H_1	Sb4 H_1H_1

ข) ลักษณะการแบ่งสับแบนด์ของภาพ

รูปที่ 2.21 แสดงการแปลงเวฟเล็ตของภาพ 1 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.21 สามารถอธิบายได้ว่าการทำการแปลงเวฟเล็ต 2 มิติ จะเป็นการทำการแปลงในแนวแกน x ตามจำนวนแถวของจุดภาพ ซึ่งทำให้ภาพถูกแบ่งเป็น 2 สับแบนด์ จากนั้นทำการแปลงเวฟเล็ตในแนวแกน y ตามจำนวนคอลัมน์ของจุดภาพ ซึ่งทำให้ภาพถ่ายที่มีลักษณะเป็น 2 มิติ ที่ผ่านการทำการแปลงเวฟเล็ตหนึ่ง ครั้ง จะถูกแบ่งเป็น 4 ส่วน (สับแบนด์) ตามที่แสดงดังรูปที่ 2.21 สับแบนด์แรก (Sb1) จะเป็นส่วนภาพที่ผ่านการกรองความถี่ต่ำผ่านทั้งสองแนวแกน สับแบนด์ที่ 2 (Sb2) จะเป็นส่วนภาพที่ผ่านการกรองความถี่สูงผ่านในแนวอน และกรองความถี่ต่ำผ่านในแนวตั้ง สับแบนด์ที่ 3 (Sb3) จะเป็นส่วนภาพที่ผ่านการกรองความถี่ต่ำผ่านในแนวอน และการกรองความถี่สูงผ่านในแนวตั้ง ในส่วนของสับแบนด์สุดท้าย (Sb4) จะเป็นส่วนภาพที่ผ่านการกรองความถี่สูงผ่านทั้งสองแนวแกน ดังนั้น ส่วนที่แสดงรายละเอียดของภาพจะเป็นส่วนข้อมูลในสับแบนด์ที่ 2, 3, และ 4 นอกจากนี้ เมื่อนำเอาสับแบนด์ที่ 1 ไปทำการแปลงเวฟเล็ตต่อไปเรื่อย ๆ ก็จะสามารถทำการแยกสับแบนด์ของภาพออกไปได้มากขึ้นด้วย ดังแสดงในรูปที่ 2.22



รูปที่ 2.22 แสดงแผนผังการทำการแปลงเวฟเล็ตกับภาพ 2 มิติ 3 ครั้ง

2.4.1 การวิเคราะห์การแปลงด้วยฮาร์ฟิงก์ชัน

ฟังก์ชันที่ง่ายในการวิเคราะห์ คือ ฮาร์ฟิงก์ชัน โดยมีสมการฟังก์ชันสเกลลิง ดังสมการที่ (2.76)

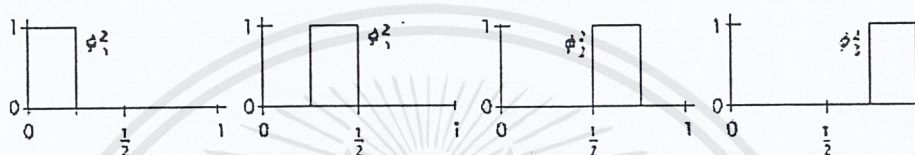
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\phi(t) = \begin{cases} 1 & \text{for } 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.76)$$

สมการที่ (2.76) เมื่อมีการเลื่อนตำแหน่งและสเกล จะทำให้รูปคลื่นของฟังก์ชันมีการเปลี่ยนแปลงดังสมการที่ (2.77)

$$\phi_{j,k}(t) = \phi(2^j t - k) \quad (2.77)$$

จากการสมการที่ (2.77) เมื่อ $j = 2$ และ $k = 0, 1, 2, 3$ จะสามารถแสดงรูปคลื่นได้คือ



รูปที่ 2.23 แสดงรูปคลื่นจากสมการและเลื่อนตำแหน่ง

ในทำนองเดียวกันฟังก์ชันเวฟเล็ต แสดงดังสมการที่ (2.78)

$$\psi(t) = \begin{cases} 1 & ; 0 \leq t < \frac{1}{2} \\ 0 & ; \text{otherwise} \\ -1 & ; \frac{1}{2} \leq t < 1 \end{cases} \quad (2.78)$$

สมการที่ (2.78) เรียกว่า เวฟเล็ตแม่ (mother wavelet) ดังนั้น เมื่อมีการสเกลและเลื่อนตำแหน่งของเวฟเล็ตแม่จะมีสมการที่ (2.79)

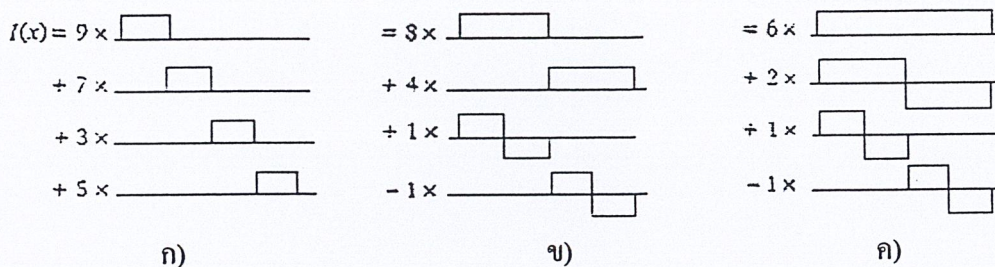
$$\psi_{j,k}(t) = \psi(2^j t - k) \quad (2.79)$$

เพื่อให้เห็นถึงขั้นตอนการนำเวฟเล็ตมาใช้ในการแปลง ตัวอย่างเช่น สมมติมีข้อมูลภาพ 4 พิกเซล คือ 9,7,2,5 ดังนั้น จำนวนระดับสเกล $j = 2$, ($2^2 = 4$) เมื่อนำมาเขียนดังสมการที่ (2.19) จะได้ว่า

$$I(x) = c_0^2 \phi_0^2(x) + c_1^2 \phi_1^2(x) + c_2^2 \phi_2^2(x) + c_3^2 \phi_3^2(x) \quad (2.80)$$

จะแสดงดังรูปที่ 2.24 ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 แสดงลักษณะผลรวมของสเกลลิงฟังก์ชันพื้นฐาน

จากสมการที่ (2.27) จะสามารถแตกกระจายสัมประสิทธิ์ได้ คือ

$$V_2 = V_1 * W_2 \quad (2.81)$$

$$I(x) = c_0^1 \phi_0^1(x) + c_1^1 \phi_1^1(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x) \quad (2.82)$$

และเมื่อทำต่อจากนั้นอีกจะได้ $V_2 = V_0 * W_0 * W_1$ นั่นคือ

$$I(x) = c_0^0 \phi_0^0(x) + c_1^0 \phi_1^0(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x) \quad (2.83)$$

จะได้ผลสุดท้าย คือ การแปลงเวฟเล็ตโดยใช้ฮาร์ฟังก์ชัน คือ 6,2,1,-1

เนื่องจากสมการที่ (2.77) และ (2.79) ยังไม่ทำการนอร์มอลไลซ์ฟังก์ชัน ด้วย $2^{-1/2}$ ดังนั้นเมื่อทำการนอร์มอลไลซ์ฟังก์ชันจากการแปลงสุดท้ายด้วย $2^{-1/2}$ ตามระดับสเกล จะได้เป็น 6,2, $1/\sqrt{2}$, $-1/\sqrt{2}$ จากการพิจารณามาพบว่า ลักษณะของการนำฮาร์ฟังก์ชันมาใช้ในการกระจายสัมประสิทธิ์เหมือนกับการบวกและลบแล้วเฉลี่ยค่าเพียงเท่านั้น เนื่องจากข้อมูลในการวิเคราะห์ข้อมูลอยู่ในลักษณะของเมตริกซ์ ในการแปลงภาพโดยการแปลงเวฟเล็ตนั้นมีขั้นตอนการแปลงอยู่ 2 กระบวนการใหญ่ ๆ คือ วิธีการแปลงแบบมาตรฐาน (standard transform) และวิธีการแปลงแบบไม่มาตรฐาน (nonstandard)

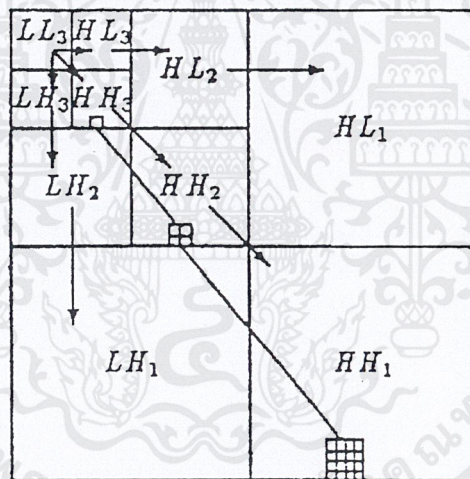
2.4.2 การใช้ซีโรทรีของสัมประสิทธิ์เวฟเล็ตเข้ารหัสข้อมูลภาพ

(Embedded Image Coding Using Zerotrees of Wavelet Coefficients)

สิ่งที่ได้หลังจากการแปลงเวฟเล็ตจะเรียกว่า สัมประสิทธิ์เวฟเล็ต สัมประสิทธิ์นี้จะถูกนำไปเข้ากำหนดรหัสและทำการลดรหัสข้อมูลลง โดย Jerome M. Shapiro จาก MIT ได้เสนอขั้นตอนขบวนการที่เหมาะสมกับการเข้ารหัสของสัมประสิทธิ์เวฟเล็ต คือ สัมประสิทธิ์ดังกล่าวจะถูกเปรียบเทียบกับค่าเทรชโฮลในระดับต่าง ๆ เพื่อกำหนดความสำคัญ (Significant) ของสัมประสิทธิ์แล้วนำมาจัดระดับข้อมูลด้วยการควอนไทซ์แบบการประมาณต่อเนื่อง (success approximation quantization : SAQ) เมื่อกำหนดรหัสให้สัมประสิทธิ์แล้วจะลดรหัสข้อมูลด้วยการเข้ารหัสเลขคณิต (Arithmetic Coding) ซึ่งเรียกขบวนการดังกล่าวว่า เอ็มเบด ซีโรทรี (Embedded Zerotrees Wavelet : EZW)

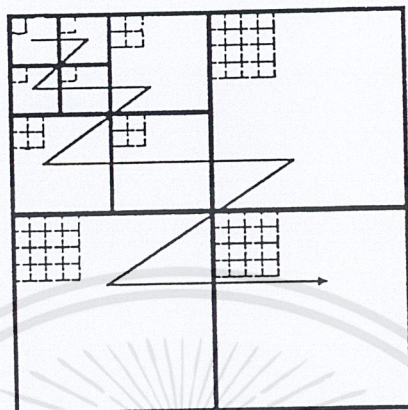
ก. การกำหนดสัญลักษณ์ให้สัมประสิทธิ์เวฟเล็ต (Wavelet Coefficient Encoding)

พิจารณารูปที่ 2.25 LL_3 จะถือว่าเป็นสับแบนด์สเกลที่หยาบสุด (Coarse Scale) และเรียกสัมประสิทธิ์ในสเกลนี้ว่าแหล่งกำเนิดหรือโหนดแม่ (parent) ส่วน HL_3 , HH_3 และ LH_3 เป็นสับแบนด์สเกลที่ละเอียดขึ้นมาอีก คือ HL_2 , HH_2 และ LH_2 เรียกว่า โหนดสืบทอด (descendant) โดยความสัมพันธ์ดังกล่าวจะเกิดไปจนกระทั่งถึงระดับสับแบนด์สเกลที่ละเอียดสูงสุด



รูปที่ 2.25 แสดงลักษณะของการสเกล และลักษณะในการสืบทอดของสัมประสิทธิ์เวฟเล็ต

พิจารณาสัมประสิทธิ์เวฟเล็ต X และค่าเทรชโฮล τ ถ้า $|X| > \tau$ สัมประสิทธิ์ในลักษณะนี้เรียกว่า สัมประสิทธิ์ที่มีนัยสำคัญสูง (significant coefficient) ภายในเงื่อนไขของ $|X|$ นอกจากนั้นเรียกว่า สัมประสิทธิ์ตัวไม่สำคัญ (insignificant coefficient) โดยวิธีการนำสัมประสิทธิ์มาเปรียบเทียบกับค่าเทรชโฮลซึ่งจะใช้การสแกนค่าของสัมประสิทธิ์แบบตัวซี (Z-SCAN) ดังรูปที่ 2.26



รูปที่ 2.26 แสดงลักษณะการสแกนแบบตัวซี (z)

สัมประสิทธิ์ตัวไม่สำคัญ ($|x| \leq r$) จะถูกจัดค่าให้เป็นศูนย์ และถ้าสเกลสัมประสิทธิ์ทั้งหมดของโหนดแม่, โหนดลูก และ โหนดสืบทอด มีค่าของสัมประสิทธิ์น้อยกว่าค่าเทรชโฮล ก็จะจัดค่าสัมประสิทธิ์ทั้งหมดในสเกลให้เป็นศูนย์ทั้งหมด ลักษณะดังกล่าวจะเกิดเป็น กลุ่มของศูนย์หรือเรียกว่า ซีโรทรีรูท (Zerotrees Root : ZTR) แต่ถ้าในสเกลใดสเกลหนึ่งใน โหนดลูก และ โหนดสืบทอด มีสัมประสิทธิ์มากกว่าค่าเทรชโฮลเพียงหนึ่งตัว จะเกิดลักษณะที่เรียกว่า ไอโซเลตซีโร (Isolate Zero : IZ) ส่วนสัมประสิทธิ์ตัวสำคัญจะกำหนดเป็นสัมประสิทธิ์ที่เป็นบวก (Positive : POS) และเป็นลบ (Negative : NEG) ตามค่าของสัมประสิทธิ์นั้น ๆ ซึ่งจะมีสัญลักษณ์ที่ใช้ในการเข้ารหัสเพียงสี่ตัวเท่านั้น คือ

$$\text{สัญลักษณ์} = \{POS, NEG, IZ, ZTR\}$$

ข.) การจัดระดับค่าแบบการประมาณต่อเนื่อง

(Successive Approximation Quantization : SAQ)

เนื่องจากการทำควอนไทซ์ของสัมประสิทธิ์ตัวสำคัญ (POS และ NEG) แบบค่าประมาณต่อเนื่อง (SAQ) จะมีความสัมพันธ์กับการเข้ารหัสแบบบิต (bit plane coding) กับขนาดค่าของสัมประสิทธิ์ ซึ่งความละเอียดของค่าสัมประสิทธิ์จากการควอนไทซ์จะสัมพันธ์กับค่าเทรชโฮลต่อการทำสแกนข้อมูลสัมประสิทธิ์หนึ่งครั้งนั้นคือ

$$\tau_i = \frac{\tau_{i-1}}{2}, i = 1, 2, \dots \quad (2.84)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ τ เป็นค่าเทรชโฮลและ i เป็นจำนวนครั้งการสแกนข้อมูลสัมประสิทธิ์

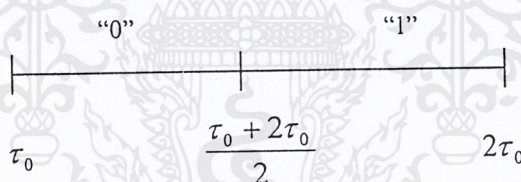
โดยค่าเทรชโฮลเริ่มต้น (τ_0) หาได้จากครึ่งหนึ่งของค่าสัมประสิทธิ์ที่สูงสุดในการแปลงเวฟเล็ต ($|X|_{\max} < 2\tau_0$) และค่าของ τ_0 ควรอยู่ในรูปของเลขยกกำลังสอง คือ

$$\tau_0 = M2^E \quad (2.85)$$

เมื่อ M เป็นค่าคงที่ และ E เป็นเลขจำนวนเต็ม

ค.) การเข้ารหัสสัญลักษณ์และถอดรหัสสัญลักษณ์ของสัมประสิทธิ์เวฟเล็ต

การจัดการกับสัมประสิทธิ์เวฟเล็ตถูกแยกเป็นสองส่วน คือ ขบวนการโดมิแนนท์ (dominant pass) และขบวนการ สับออร์ดิเนท (subordinate pass) โดยเริ่มจากขบวนการโดมิแนนท์ ทำการกำหนดสัญลักษณ์ $\{POS, NEG, IZ, ZTR\}$ ให้กับสัมประสิทธิ์ที่ได้มาจากการทำ Z สแกนสัมประสิทธิ์ที่เป็นสัญลักษณ์ POS หรือ NEG จะถูกนำมาทำต่อด้วยขบวนการสับออร์ดิเนท ซึ่งสัญลักษณ์ที่ใช้เข้ารหัสใน สับออร์ดิเนท จะใช้ตัวเลขฐานสอง คือ แทน "1" เพื่อชี้ค่าจริงของสัมประสิทธิ์ว่า มีค่าในช่วงบนจากค่ากลางระหว่างค่าเทรชโฮล และแทน "0" เพื่อชี้ค่าจริงของสัมประสิทธิ์ว่าอยู่ในช่วงล่างของค่ากลางระหว่างค่าเทรชโฮล ดังรูปที่ 2.27 ส่วนการถอดรหัสก็จะทำตรงกันข้ามกับการเข้ารหัส



รูปที่ 2.27 แสดงการควอนไทซ์ค่าสัมประสิทธิ์และกำหนดสัญลักษณ์

2.5 หลักการเบื้องต้นของการลดข้อมูล

การลดข้อมูลหรือการบีบอัดข้อมูล จะทำการลดหรือกำจัดข้อมูลส่วนเกินออก โดยข้อมูลส่วนเกินดังกล่าวจะแบ่งเป็น 3 ประเภท คือ ข้อมูลส่วนเกินจากการเข้ารหัสข้อมูล (coding redundancy) , ข้อมูลส่วนเกินที่เกิดจากสหสัมพันธ์ (correlation) ระหว่างพิกเซลของภาพ โดยค่าของพิกเซลใดๆ สามารถพยากรณ์ได้จากค่าของพิกเซลรอบข้างของตัวมัน เรียกว่า อินเตอร์พิกเซลดันเดซี (interpixel redundancy) , และเกิดจากความสามารถในการรับรู้ทางประสาทตาของมนุษย์เรียกว่า ไซโครวิชวลรีดันเดซี (psychovisual redundancy) ซึ่งในปริปัญญานี้จะใช้วิธีการลดข้อมูลแบบ การเข้ารหัสเลขคณิต

การเข้ารหัสเลขคณิต (Arithmetic Coding)

ในการเข้ารหัสเลขคณิตนั้นข่าวสารต้นแบบจะถูกแทนด้วยช่วงระยะห่างของจำนวนจริงระหว่าง 0 และ 1 โดยข้อมูลแต่ละตัวของข่าวสารจะทำให้ช่วงระยะห่างนั้นแคบลง ซึ่งระยะห่างที่เล็กลงนี้จะทำให้ใช้จำนวนของบิตมากขึ้น ตัวข้อมูลที่ต่อเนื่องกันของข่าวสารจะลดขนาดของระยะห่างตามความน่าจะเป็นของข้อมูล ข่าวสารที่มีความน่าจะเป็นสูงจะทำให้ระยะห่างถูกลดลงได้น้อยกว่าข่าวสารที่มีความน่าจะเป็นต่ำ ดังนั้นจึงทำให้ข่าวสารที่มีความน่าจะเป็นสูงใช้จำนวนบิตในการเข้ารหัสที่น้อยกว่า

วิธีการเข้ารหัสแบบพื้นฐานจะมีขั้นตอนดังนี้

ขั้นตอนที่ 1 จะทำการกำหนดช่วงขอบเขตเริ่มต้น $[L, H]$ ให้เป็น $[0, 1]$

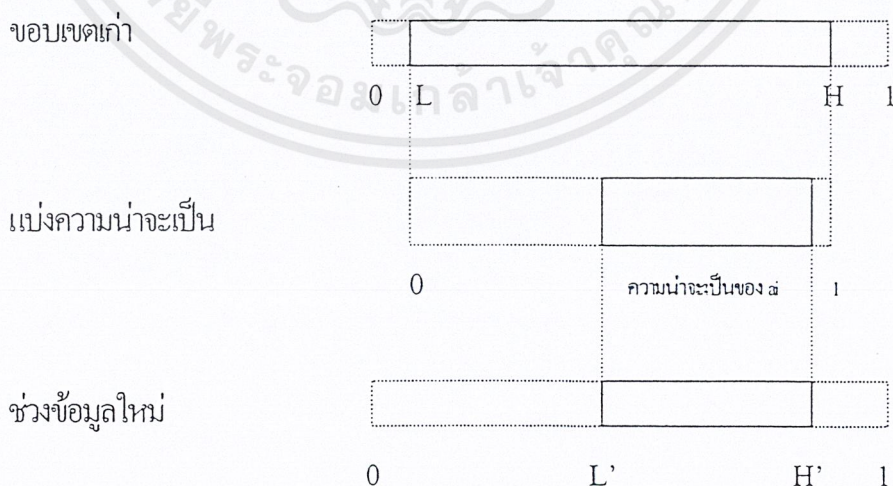
ขั้นตอนที่ 2 ในขั้นตอนนี้จะนำสัญลักษณ์ของข้อมูลมาพิจารณา โดยแบ่งเป็น 2 ขั้นตอนย่อย คือ (แสดงดังรูปที่ 2.29)

1. ทำการแบ่งช่วงขอบเขตปัจจุบันให้เป็นช่วงของขอบเขตย่อย ตามความน่าจะเป็นของสัญลักษณ์

2. เลือกช่วงของขอบเขตย่อยที่สอดคล้องกับสัญลักษณ์ที่พบในข้อมูลแล้วนำช่วงของขอบเขตดังกล่าวมาเป็นช่วงขอบเขตปัจจุบันแทน

กระทำในขั้นตอนที่สองจนกว่าจะสิ้นสุดชุดของข้อมูลที่จะนำเข้ามาเข้ารหัส หรือเมื่อพบอักษรที่แสดงถึงจุดสิ้นสุดของข้อมูล

ขั้นตอนที่ 3 เอาต์พุตที่ได้จากการเข้ารหัสก็คือ ค่าของช่วงขอบเขตสุดท้ายที่ได้ในขั้นตอนที่ 2 โดยนำค่าขอบเขตดังกล่าวมาแปลงเป็นรหัสไบนารี



รูปที่ 2.28 แสดงการแบ่งช่องของขอบเขตปัจจุบันบนพื้นฐาน

ของความน่าจะเป็นของสัญลักษณ์ a_i ที่ปรากฏขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนบิตของรหัสไบนารีที่ใช้แทนช่วงขอบเขตสามารถพิจารณาได้จาก ผลคูณของค่าความน่าจะเป็นของแต่ละสัญลักษณ์ให้มีค่าเท่ากับ P ที่เกิดจากลำดับของสัญลักษณ์ข้อมูลจำนวนบิตของรหัสจะเท่ากับ $-10 \log_2 P$ บิต

ในขั้นตอนที่ 2 จะทำการคำนวณขนาดของขอบเขตของสัญลักษณ์ a_i ที่ปรากฏขึ้นโดยทำการหาค่าความน่าจะเป็นสะสม 2 ตัว คือ $P_c = \sum_{k=1}^{i-1} p_k$ และ $P_n = \sum_{k=1}^i p_k$ และนำมาคำนวณหาช่วงขอบเขตใหม่ได้ดังนี้

$$[L + P_c(H - L), L + P_n(H - L)] \quad (2.86)$$

กระบวนการจะเริ่มจากข้อมูลต้นแบบที่ไม่ได้มีการเรียงลำดับความน่าจะเป็นจากนั้นจะทำการแบ่งช่วงระยะห่างระหว่าง 0 กับ 1 เป็นช่วงย่อยๆ ตามความน่าจะเป็นสะสม (cumulative probability) อาทิ เช่น กำหนดให้ข้อมูลต้นแบบ คือ $S = \{A, B, C, D, \#\}$ มีความน่าจะเป็นเป็น $P = \{0.3, 0.1, 0.2, 0.3, 0.1\}$ ตามลำดับการแบ่งช่วงระยะห่างระหว่าง 0 กับ 1 แสดงไว้ในตารางที่ 2.3

ตัวอย่างการเข้ารหัส AABD# แสดงไว้ในรูปที่ 2.32 โดยมีขั้นตอนดังต่อไปนี้

1. A ตัวแรกจะลดขอบเขตของระยะห่างระหว่าง 0 กับ 1 ลงเป็น $[0, 0.3]$
2. A ตัวที่สองจะลดขอบเขตของระยะห่างระหว่าง 0 กับ 0.3 ลงเป็น $[0, 0.09]$
3. B จะลดขอบเขตระยะห่างระหว่าง 0 กับ 0.3 ลงเป็น $[0.027, 0.036]$
4. D จะลดขอบเขตระยะห่างระหว่าง 0.027 กับ 0.036 ลงเป็น $[0.0324, 0.0351]$
5. # จะลดขอบเขตระยะห่างระหว่าง 0.0324 กับ 0.0351 ลงเป็น $[0.0348, 0.0351]$

โดยที่ $[0, 0.3]$ หมายถึงระยะห่างที่อยู่ในช่วงมากกว่าหรือเท่ากับ 0 และน้อยกว่า 0.3 เป็นต้น

ดังนั้นข้อมูล AABD# สามารถจะเข้ารหัสได้โดยการแทนด้วยช่วงระยะห่างหรือตัวเลขจำนวนใดๆ ที่อยู่ในช่วง $[0.0348, 0.0351]$ ค่าความน่าจะเป็นของการเข้ารหัสของข้อมูลนี้คือ

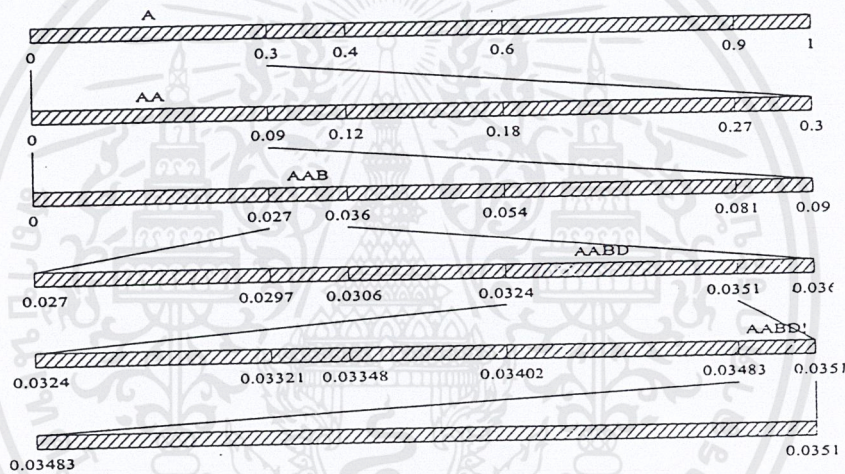
$(0.3)^2 \times (0.1) \times (0.3) \times (0.1) = 0.00027$ จำนวนบิตของข้อมูลที่จะใช้แทนกลุ่มของสัญลักษณ์ข้อมูลนี้จะเท่ากับ $[-\log_2(0.00027)] = [11.854] = 12$ บิต

ดังนั้นเลือกข้อมูลในช่วง $[0.0348, 0.0351]$ มาแปลงเป็นไบนารี โดยเลือกค่ากลางในช่วงนี้จะได้เท่ากับ 0.034965 นำมาแปลงเป็นเลขไบนารีจะได้ $(0.000100011100110\dots)_2$ แต่จะนำมาเพียง 12 บิตเท่านั้นที่ใช้แทนข้อมูลที่นำมาเข้ารหัส ดังนั้น จะได้รับรหัสที่แทนข้อมูล AABD# คือ 0000 1000 1111

ถ้าเราจะพิจารณากำหนดรหัสให้กับสัญลักษณ์ S โดยกำหนดให้ใช้รหัสตัวเลข 3 บิต (ความน่าจะเป็นของการเกิดแต่ละสัญลักษณ์เท่ากับ 1 ดังนั้นสัญลักษณ์จำนวน 5 ตัว จะสามารถกำหนดรหัสได้ต่ำสุดเท่ากับ $[-\log_2(1/5)] = [2.321]$ จะได้ขนาดของข้อมูล AABD# เป็น 15 บิต (3×5) จากตัวอย่างเห็นว่าข้อมูลจะถูกลดลงมา 3 บิต

ตารางที่ 2.3 แสดงการแบ่งขอบเขตตามความน่าจะเป็นสะสม

ข้อมูลต้นแบบ	ความน่าจะเป็น (Probability)	ความน่าจะเป็นสะสม (Cumulative probability)	ขอบเขต (Range)
A	0.3	0.3	[0,0.3)
B	0.1	0.4	[0.3,0.4)
C	0.2	0.6	[0.4,0.6)
D	0.3	0.9	[0.6,0.9)
#	0.1	1.0	[0.9,1)



รูปที่ 2.29 แสดงการแบ่งขอบเขตตามข้อมูลที่เพิ่มขึ้น

สมมติว่าตัวอักษรที่ i ได้รับข้อมูลรหัสเป็น $i = 0.0350$ ดังนั้นกระบวนการถอดรหัสจะมีขั้นตอนดังต่อไปนี้

1. เริ่มจากแบ่งช่วง $[0, 1)$ ตามความน่าจะเป็น แล้วพบว่า $i = 0.0350$ อยู่ในช่วง $[0, 0.3)$ ซึ่งเป็นช่วงของตัวอักษร A ดังนั้นจะได้อักษรตัวแรกคือ A จากนั้นจะลดขอบเขตของ $[0, 1)$ ลงเป็น $[0, 0.3)$
2. แบ่งช่วง $[0, 0.3)$ ตามความน่าจะเป็นแล้วพบว่า $i = 0.0350$ อยู่ในช่วง $[0, 0.09)$ ซึ่งเป็นช่วงของตัวอักษร A ดังนั้นจะได้อักษรตัวที่สองคือ A จากนั้นจะลดขอบเขตของ $[0, 0.3)$ ลงเป็น $[0, 0.09)$
3. แบ่งช่วง $[0, 0.09)$ ตามความน่าจะเป็น แล้วพบว่า $i = 0.0350$ อยู่ในช่วง $[0.027, 0.036)$ ซึ่งเป็นช่วงของตัวอักษร B ดังนั้นจะได้อักษรตัวที่สามคือ B จากนั้นจะลดขอบเขตของ $[0, 0.09)$ ลงเป็น $[0.027, 0.036)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แบ่งช่วง $[0.027, 0.036]$ ตามความน่าจะเป็น แล้วพบว่า $i = 0.0350$ อยู่ในช่วง $[0.0324, 0.0351]$ ซึ่งเป็นช่วงของตัวอักษร D ดังนั้นจะได้อักษรตัวที่สามคือ D จากนั้นจะลดขอบเขตของ $[0.0324, 0.0351]$ ลงเป็น $[0.0324, 0.0351]$
5. แบ่งช่วง $[0.0324, 0.0351]$ ตามความน่าจะเป็น แล้วพบว่า $i = 0.0350$ อยู่ในช่วง $[0.03483, 0.0351]$ ซึ่งเป็นช่วงของตัวอักษร # ดังนั้นจะได้อักษรตัวที่สาม คือ # จากนั้น จะลดขอบเขตของ $[0.0324, 0.0351]$ ลงเป็น $[0.03483, 0.0351]$

2.6 เกณฑ์การวัดความเหมือนจริงของภาพ (Image Fidelity)

ในการลดข้อมูลภาพนั้น จะมีข้อมูลส่วนหนึ่งที่เกิดผิดพลาดหรือสูญหายไป ข้อผิดพลาดนี้เกิดขึ้นนี้จะมีผลในตอนที่สร้างภาพกลับคืนมา (reconstruction) และค่าความผิดพลาดนี้จะอยู่ในช่วงหนึ่งที่สามารถยอมรับได้ ดังนั้นเกณฑ์การวัดความเหมือนจริงของภาพสามารถนำมาใช้ในการวัดประสิทธิภาพของระบบได้ ตัวอย่างเกณฑ์ที่นิยมใช้ในการวัดคุณภาพของภาพคือ ค่า root-mean-square (rms) ของความผิดพลาดระหว่างข้อมูลภาพอินพุตและข้อมูลภาพเอาต์พุต นอกจากนี้ยังมีค่า rms ของอัตราส่วนของสัญญาณต่อสัญญาณรบกวนของภาพเอาต์พุต (Signal to Noise Ratio) เมื่อกำหนดให้ข้อมูลภาพอินพุตประกอบด้วยอาร์เรย์ขนาด $N \times N$ ของจุดภาพ $f(x, y)$ และ x และ y มีค่าเป็น $0, 1, \dots, N-1$ และแต่ละจุดภาพมีค่าของระดับสีเทาที่เป็นไปได้คือ 2^m เมื่อ m เป็นจำนวนบิตของระดับสีเทา

สำหรับทุกค่าของ x และ y ในช่วง $0, 1, \dots, N-1$ ค่าความผิดพลาดระหว่างจุดภาพอินพุตและเอาต์พุตคือ

$$e(x, y) = g(x, y) - f(x, y) \quad (2.87)$$

เมื่อ $e(x, y)$ คือค่าความผิดพลาดที่เกิดขึ้น ณ จุด x, y ใด ๆ
 $f(x, y)$ คือค่าภาพอินพุต ณ จุด x, y ใด ๆ
 $g(x, y)$ คือค่าภาพเอาต์พุต ณ จุด x, y ใด ๆ

ค่าเฉลี่ยของความผิดพลาดทั้งสองของภาพ (mean square error) คือ

$$\begin{aligned} e_{ms} &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^2(x, y) \\ &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [g(x, y) - f(x, y)]^2 \end{aligned} \quad (2.88)$$

ดังนั้นค่า rms ของความผิดพลาดจึงสามารถเขียนได้ดังนี้

$$e_{rms} = [e_{rms}]^{1/2} \quad (2.89)$$

ค่า root mean square error เป็นค่าที่ใช้ในการวัดความแตกต่างของข้อมูลอินพุตกับข้อมูลเอาต์พุต แต่เมื่อพิจารณาขนาดของข้อมูลเอาต์พุตต่อขนาดของสัญญาณรบกวน (noise) ก็จะได้เป็นค่า Signal-to-Noise Ratio (SNR) เมื่อกำหนดให้สัญญาณภาพเอาต์พุตแต่ละจุดประกอบด้วยสัญญาณอินพุตบวกด้วยค่าสัญญาณรบกวน นั่นคือ

$$g(x, y) = f(x, y) + e(x, y) \quad (2.90)$$

ดังนั้นค่าเฉลี่ยกำลังสองของอัตราส่วนของสัญญาณต่อสัญญาณรบกวนของข้อมูลภาพเอาต์พุต สามารถได้ โดยค่าเฉลี่ยของสัญญาณอินพุตกำลังสองหารด้วยค่าเฉลี่ยของสัญญาณรบกวนกำลังสอง ของข้อมูลภาพทั้งหมด ซึ่งสามารถเขียนได้ดังนี้

$$SNR_{rms} = \frac{\frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g^2(x, y)}{\frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^2(x, y)} \quad (2.91)$$

ค่า rms ของ SNR จึงสามารถเขียนได้ดังต่อไปนี้

$$SNR_{rms} = \left(\frac{\frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g^2(x, y)}{\frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [g(x, y) - f(x, y)]^2} \right)^{1/2} \quad (2.92)$$

โดยที่เทอมส่วนของสมการข้างบนเป็นสมการของสัญญาณรบกวนที่อยู่ในรูปของผลต่างระหว่างข้อมูลอินพุตกับข้อมูลเอาต์พุต

จากวิธีการที่ใช้ในการวัดความเหมือนจริงของภาพที่กล่าวมาข้างต้นไม่สามารถที่จะบ่งบอกหรือใช้เป็นเกณฑ์ในการพิจารณาได้แต่เพียงอย่างเดียว ในกรณีของภาพเอาต์พุตที่ได้จากการประมวลผลหรือรับส่งสัญญาณ โดยมีสายตาของมนุษย์เป็นตัวรับภาพ ซึ่งระบบการมองเห็นของสายตาคงไวต่อความเข้มแสงในลักษณะของล็อก (logarithmic) ดังนั้น ความผิดพลาดในบริเวณที่เป็นที่มืดของภาพจะเห็นได้ชัดเจนกว่าความผิดพลาดที่อยู่ในบริเวณที่สว่าง และระบบการมองเห็นยังไวต่อการเปลี่ยนแปลงอย่างทันทีทันใดของระดับสีเทาด้วย ความผิดพลาดที่อยู่บนขอบหรือใกล้ๆ ขอบของวัตถุจะมีผลต่อการมองเห็นมากกว่าความผิดพลาดที่อยู่ในโครงสร้างที่เป็นฉากหลังของภาพ ด้วยเหตุนี้เอง ถึงแม้ว่าจะมีค่าของ rms

ในความผิดพลาดที่เท่ากัน แต่อาจจะปรากฏความแตกต่างของคุณภาพของการมองเห็นที่แตกต่างของคุณภาพของการมองเห็นที่แตกต่างกันได้

2.6.1 อัตราบิต

อัตราบิต (bit rate) คือค่าเฉลี่ยของจำนวนบิตต่อจุดภาพ (bits per pixel, bpp) ของภาพที่โดนลดขนาดข้อมูล โดยสามารถคำนวณได้จากอัตราส่วนระหว่างจำนวนบิตข้อมูลทั้งหมดของภาพที่ถูกลดข้อมูลต่อจำนวนจุดภาพทั้งหมดของภาพต้นแบบ ดังสมการต่อไปนี้

$$\text{bpp} = \frac{\text{จำนวนบิตทั้งหมดของภาพที่ลดข้อมูล}}{\text{จำนวนจุดภาพของภาพต้นแบบทั้งหมด}} \quad (2.93)$$

2.6.2 อัตราการลดข้อมูล

อัตราการลดข้อมูลคืออัตราส่วนระหว่างจำนวนบิตข้อมูลที่ใช้แทนภาพต้นแบบ และจำนวนบิตข้อมูลที่ใช้แทนภาพที่ถูกทำการลดข้อมูลโดยสามารถหาได้ดังสมการ

$$\text{อัตราการลดข้อมูล} = \frac{\text{จำนวนบิตทั้งหมดของภาพต้นแบบ}}{\text{จำนวนบิตทั้งหมดของภาพที่ลดข้อมูล}} \quad (2.94)$$

บทที่ 3

การคำนวณและการสร้าง

ในการบีบอัดข้อมูลภาพโดยทั่ว ๆ ไปจะมีขบวนการที่คล้ายคลึงกัน โดยสามารถที่จะแยกเป็นสองแนวทางใหญ่ ๆ คือ คาด้าโมเดลลิ่ง (data modeling) และการเข้ารหัส (code packing) ในขั้นตอนแรกคือ คาด้าโมเดลลิ่ง จะนำเอาข้อมูลข่าวสารมาเปลี่ยนรูปแบบใหม่ซึ่งมีลักษณะที่สัมพันธ์กับข้อมูลเดิม เพื่อที่จำสามารถใช้ลักษณะเฉพาะของโมเดลต่าง ๆ เพื่อกำจัด หรือ ลดความซ้ำซ้อนของข้อมูลบางประการลงได้ โดยอาศัยเทคนิคต่าง ๆ หลายรูปแบบดังเช่น รูปแบบการแปลงข้อมูลภาพ (image transform) ซึ่งอาจใช้การแปลงฟูเรียร์, การแปลงของดิสครีตโคไซน์ (discrete cosine transform : DCT) ฯลฯ และอีกรูปแบบหนึ่งที่มีการนำมาใช้ คือ การแปลงเวฟเลต (wavelet transform) ซึ่งเป็นวิธีการแปลงข้อมูลภาพ ที่นำมาใช้ในการลดขนาดข้อมูลภาพของปริญญาานิพนธ์นี้

ส่วนในขั้นตอนของการเข้ารหัส จะทำการนำข้อมูลที่ได้จากการทำโมเดลลิ่ง มาจัดระดับ และเข้ารหัสข้อมูลใหม่เพื่อให้ข้อมูลมีคุณลักษณะที่เหมาะสมในการนำไปใช้งาน โดยวิธีการโค้ดแพ็คเกจที่นำมาใช้ในปริญญาานิพนธ์นี้จะแยกเป็นสองขั้นตอนด้วยกัน คือ ขั้นตอนแรก จะเป็นการนำโมเดลลิ่งที่ได้มาทำการจัดกลุ่มของสัมประสิทธิ์ที่ได้จากการแปลงเวฟเลต เพื่อที่จะให้ได้สัญลักษณ์ของข้อมูล ที่มาแทนค่าสัมประสิทธิ์ที่ได้จากการทำโมเดลลิ่ง โดยอาศัยวิธีการจัดระดับประมาณค่าแบบซัคเซสซีฟ (successive approximation quantization : SAQ) ซึ่งเป็นขบวนการเปลี่ยนหรือแปลงสัมประสิทธิ์เวฟเลตไปเป็นชุดบิตข้อมูล โดยวิธีการนี้จะคล้ายกับการเข้ารหัสแบบบิตเพลน (bit plane encoding) ในบทที่ 2 สัมประสิทธิ์เวฟเลตถูกจัดค่าตามระดับบิต โดยระดับบิตจะเริ่มจากบิตที่มีนัยสำคัญก่อน โดยวิธีนี้ถูกคิดค้นขึ้นมาโดยเซฟิโร (Jerome M. Shapiro) และเรียกวินวิธีการดังกล่าวว่า เอ็มเบดซีโรทรี (Embedded zerotree) ซึ่งผลที่ได้จะเป็นการจัดกลุ่มของข้อมูลที่เป็นค่าสัมประสิทธิ์ให้เป็นสัญลักษณ์ของข้อมูลเพื่อให้เหมาะสมที่จะนำไปเข้ารหัสลดขนาดข้อมูลแบบไม่มีการสูญเสียในขั้นตอนต่อไป

ขั้นตอนสุดท้ายของการเข้ารหัสคือการนำเอาลำดับของข้อมูลที่ได้จากการจัดกลุ่มของสัมประสิทธิ์ในขั้นตอนแรกของการเข้ารหัส มาทำการเข้ารหัสลดขนาดข้อมูลแบบไม่มีการสูญเสีย โดยจะนำเอาการเข้ารหัสเลขคณิตมาใช้งาน

ขั้นตอนของการลดขนาดข้อมูลภาพด้วยวิธีการใช้การแปลงเวฟเลต จะแสดงในรูปที่ 3.1

3.1 ข้อมูลภาพบิตแมป(BMP)

ข้อมูลภาพในรูปแบบบิตแมปประกอบด้วย 3 ส่วน คือ

- หัวไฟล์(BITMAPFILEHEADER)
- หัวข้อมูล(BITMAPINFO) ซึ่งรวมทั้งข้อมูลต่างๆของพาเลตสี(color palette)
- ข้อมูลภาพจริง

3.1.1 หัวไฟล์(BITMAPFILEHEADER)

หัวของไฟล์(BITMAPFILEHEADER) จะประกอบด้วยข้อมูลเกี่ยวกับชนิดของ รูปแบบข้อมูลภาพขนาดของภาพและโครงสร้างแสดงดังนี้

- bfType เป็นชนิดของรูปแบบข้อมูลภาพ สำหรับบิตแมปคือ "BM"
- bfSize เป็นขนาดของไฟล์ทั้งหมด
- bfReserbed1,2 ไม่ได้ใช้งาน ปกติกำหนดค่าเป็นศูนย์
- bfOffbits เป็นขนาดของหัวไฟล์ทั้งหมด ใช้เพื่อข้ามไปจุดเริ่มต้นของข้อมูลภาพจริง

3.1.2 หัวข้อมูล (BITMAPINFO)

หัวข้อมูลจะเป็นตัวกำหนดขนาดต่างๆและข้อมูลสีแสดงดังนี้

- biSize เป็นขนาดของส่วนหัวข้อมูล(BITMAPINFOHEADER)มีหน่วยเป็นไบต์
- biWidth เป็นความกว้างของภาพ(มีหน่วยเป็นจุดภาพต่อเส้น)
- biHeight เป็นความสูงของภาพ(มีหน่วยเป็นจุดภาพต่อเส้น)
- biPlanes เป็นจำนวนหน้าของสี (color plane) สำหรับอุปกรณ์เป้าหมาย (targetdevice) ปกติกำหนดให้เป็น 1 เสมอ
- biBitCount จำนวนบิตต่อจุดภาพ (1,4,8,24)
- biCompression ชนิดของการบีบอัด
- biSizeImage เป็นขนาดของภาพ(มีหน่วยเป็นไบต์)
- biXPelsPerMeterความละเอียดสำหรับอุปกรณ์เป้าหมายในแนวนอนต่อหนึ่งเมตร
- biYPelsPerMeterความละเอียดสำหรับอุปกรณ์เป้าหมายในแนวตั้งต่อหนึ่งเมตร
- biClrUsed จำนวนดัชนีสี (color index) ในตารางสี (color table) มีค่าได้ 3 กรณี
- biClrImportant จำนวนความสำคัญของดัชนีสี (color index) ในการแสดงผล ถ้าเป็นศูนย์ทุกสี จะมีความสำคัญเท่ากันหมด

ข้อมูลภาพจริง (Actual Image Information)

ส่วนของข้อมูลบิตแมปจริงจะมีการเก็บที่แตกต่างกันตามจำนวนบิตข้อมูลภาพ(biBitCount)และลักษณะการบีบอัด bitCompressionแต่โดยทั่วไปแล้วข้อมูลในรูปแบบบิตแมปปกติที่นิยมในปัจจุบันในการส่งข้ามระหว่างโปรแกรมประยุกต์ต่าง ๆ นั้นจะไม่ใช้การบีบอัดข้อมูล ดังนั้นจึงแบ่งรูปแบบข้อมูลในลักษณะที่นิยมใช้กันในปัจจุบัน ได้ดังนี้

กรณีที่ 1 ถ้าเป็น 8 บิตต่อภาพ หมายความว่า มีดัชนีสี bmiColors ที่เป็นอาร์เรย์จำนวน 256 อาร์เรย์คือมีสี 256 สี โดยข้อมูลแต่ละ 8 บิตจะแทนหนึ่งจุดภาพ

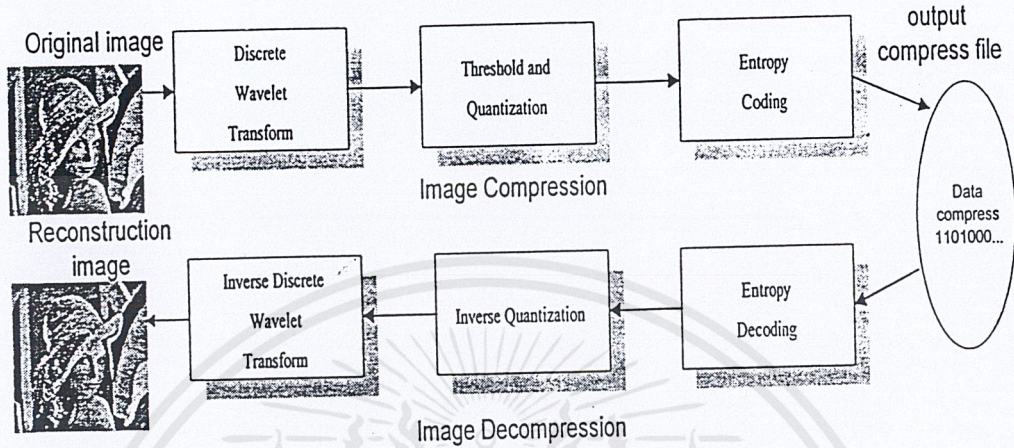
กรณีที่ 2 ถ้าเป็น 24 บิตต่อจุดภาพ หมายความว่า จะสามารถแสดงสีได้พร้อมๆกันเท่ากับ 2^{24} คือ 16.7 ล้านสีที่สามารถผสมสีได้โดยตรง ไม่ต้องมีการใช้ดัชนีสี ดังนั้นในกรณีนี้ bmiColours เท่ากับ NULL ข้อมูลแต่ละ 3 ไบต์ที่เรียงกันจะแสดง relative intensity ของสีฟ้า สีเขียว สีแดง (BGR) ตามลำดับ โดยข้อมูลแต่ละ 24 บิตจะแทน 1 จุดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การคำนวณการแปลงเวฟเล็ตกับข้อมูลภาพ

จากสมการ 2.52

$$c_j(k) = \sum_m h_0(m-2k)c_{j+1}(m)$$



รูปที่ 3.1 แสดงขั้นตอนการบีบอัดข้อมูลภาพโดยการใช้การแปลงเวฟเล็ต

และจากสมการ 2.53

$$d_j(k) = \sum_m h_1(m-2k)c_{j+1}(m)$$

นำทั้งสองสมการดังกล่าวมาใช้สำหรับการแปลงเวฟเล็ตแบบไม่ต่อเนื่องโดยค่าสัมประสิทธิ์ของการกรองความถี่ต่ำ (h_0) และการกรองความถี่สูง (h_1) คือสัมประสิทธิ์ของเวฟเล็ตฟังก์ชัน ซึ่งจะแสดงตัวอย่างการคำนวณดังตัวอย่างที่ 1 และ 2 โดยจะใช้ฟังก์ชันฮาร์เป็นเวฟเล็ตฟังก์ชันเพื่อความสะดวกในการคำนวณ

พิจารณา สัมประสิทธิ์ของการกรองความถี่ต่ำ (h_0) และการกรองความถี่สูง (h_1) ของฮาร์ฟังก์ชันคือ

$$h_0 = \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\} \text{ และ } h_1 = \left\{ \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right\} \quad (3.1)$$

ตัวอย่างที่ 1 ข้อมูลภาพมีขนาด 4 พิกเซล คือ 9, 7, 3 และ 5

จะได้ว่า

$$c_1(k) = \sum_{m=0}^3 h_0(m-2k)c_2(m)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น

$$\begin{aligned} c_1(0) &= h_0(0)c_2(0) + h_0(1)c_2(1) + h_0(2)c_2(2) + h_0(3)c_2(3) \\ &= \frac{9}{\sqrt{2}} + \frac{7}{\sqrt{2}} = \frac{16}{\sqrt{2}} \end{aligned}$$

และ

$$\begin{aligned} c_1(1) &= h_0(-2)c_2(0) + h_0(-1)c_2(1) + h_0(0)c_2(2) + h_0(1)c_2(3) \\ &= \frac{3}{\sqrt{2}} + \frac{5}{\sqrt{2}} = \frac{8}{\sqrt{2}} \end{aligned}$$

ในทำนองเดียวกัน

$$\begin{aligned} d_1(k) &= \sum_{m=0}^3 h_1(m-2k)c_2(m) \\ d_1(1) &= h_1(0)c_2(0) + h_1(1)c_2(1) + h_1(2)c_2(2) + h_1(3)c_2(3) \\ &= \frac{9}{\sqrt{2}} - \frac{7}{\sqrt{2}} = \frac{2}{\sqrt{2}} \end{aligned}$$

และ

$$\begin{aligned} d_1(1) &= h_1(-2)c_2(0) + h_1(-1)c_2(1) + h_1(0)c_2(2) + h_1(1)c_2(3) \\ &= \frac{3}{\sqrt{2}} - \frac{5}{\sqrt{2}} = \frac{-2}{\sqrt{2}} \end{aligned}$$

ดังนั้นจะได้ผลการแปลงเวฟเล็ต คือ $\left\{ \frac{16}{\sqrt{2}}, \frac{8}{\sqrt{2}}, \frac{2}{\sqrt{2}}, \frac{-2}{\sqrt{2}} \right\}$ ที่อยู่ในรูปแบบสมการ (2.29) ในบทที่ 2

และเมื่อทำการแปลงเวฟเล็ต อีกจะได้ว่า

$$\begin{aligned} c_0(k) &= \sum_{m=0}^1 h_0(m-2k)c_1(m) \\ c_0(0) &= h_0(0)c_1(0) + h_0(1)c_1(1) \\ &= \frac{16}{2} + \frac{8}{2} = 12 \end{aligned}$$

และ

$$\begin{aligned} d_0(k) &= \sum_{m=0}^1 h_1(m-2k)c_1(m) \\ d_0(0) &= h_1(0)c_1(0) + h_1(1)c_1(1) \\ &= \frac{16}{2} - \frac{8}{2} = 4 \end{aligned}$$

ผลสุดท้ายจะได้สัมประสิทธิ์เวฟเล็ต คือ $\left\{ 12, 4, \frac{2}{\sqrt{2}}, \frac{-2}{\sqrt{2}} \right\}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 สมมติข้อมูลภาพมีขนาด 8 จุดภาพ คือ 1, 0, 3, 2, 1, 0, 1 และ 2 ข้อมูลภาพดังกล่าวสามารถแสดงการแปลงเวฟเลตแบบไม่ต่อเนื่องได้ดังนี้

$$\begin{array}{l}
 c_3 \quad \boxed{1 \quad 0 \quad -3 \quad 2 \quad 1 \quad 0 \quad 1 \quad 2} \\
 d_2 \quad \boxed{\frac{1}{\sqrt{2}} \quad -\frac{5}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}}} \\
 c_2 \quad \boxed{\frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{3}{\sqrt{2}}} \\
 d_1 \quad \boxed{1 \quad -1} \\
 c_1 \quad \boxed{0 \quad 2} \\
 d_0 \quad \boxed{-\sqrt{2}} \\
 c_0 \quad \boxed{\sqrt{2}}
 \end{array}$$

จะได้สัมประสิทธิ์ของข้อมูลหลังการแปลงเวฟเลต คือ $\left\{ \sqrt{2}, -\sqrt{2}, 1, -1, \frac{1}{\sqrt{2}}, -\frac{5}{\sqrt{2}}, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right\}$

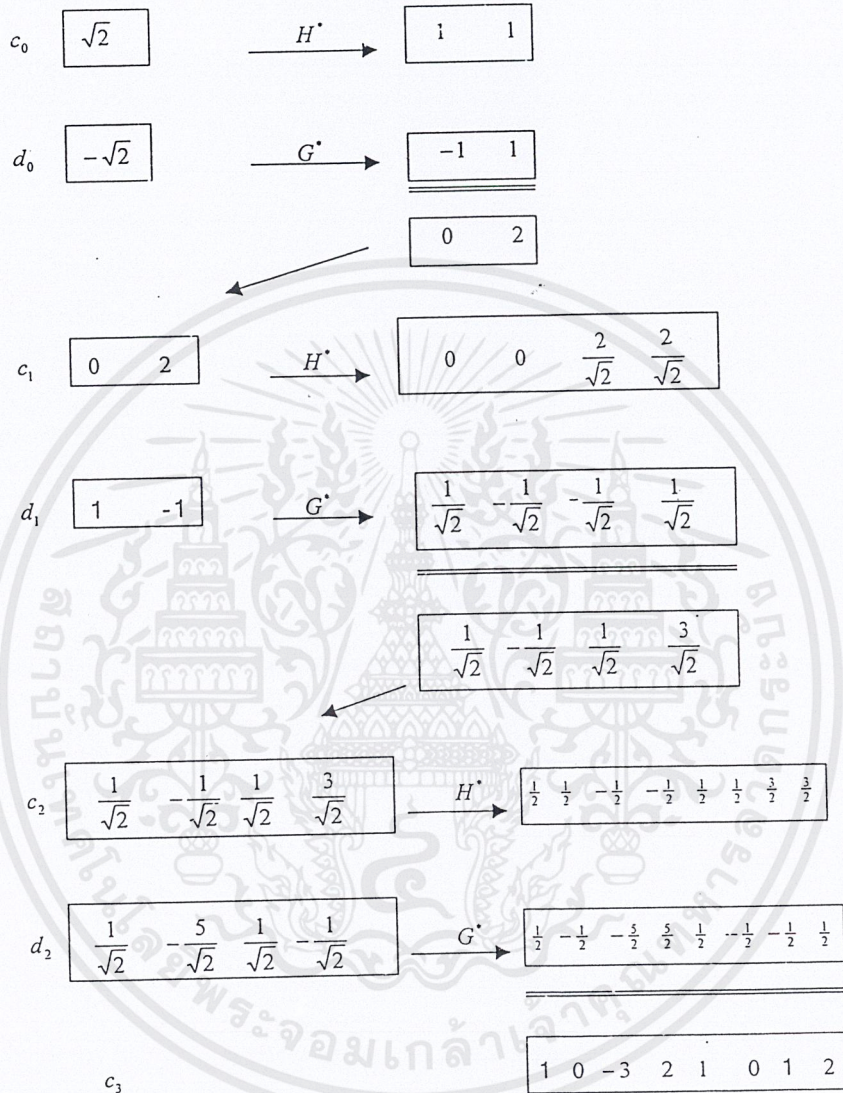
และจากสมการที่ 2.75 แสดงถึงลักษณะการสังเคราะห์สัญญาณกลับ จะตรงกันข้ามกับการวิเคราะห์สัญญาณด้วยการแยกช่วงความถี่ออกจากกันที่ละสองช่อง โดยทำการเพิ่มอัตราการซัดค่าตัวอย่างให้มีค่าเป็นสองเท่าจากเดิมก่อน ซึ่งจะทำการแทรกค่าศูนย์ระหว่างเทอมของสัญญาณ

$$c_{j+1}(n) = c_j(n) * h_0(n)|_{n=2m} + d_j(n) * h_1(n)|_{n=2m}$$

สามารถแสดงดังตัวอย่างการสังเคราะห์สัญญาณกลับจากสัมประสิทธิ์ของข้อมูลที่ได้จากการแปลงเวฟเลตจากตัวอย่างที่ 2 และตัวอย่างที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

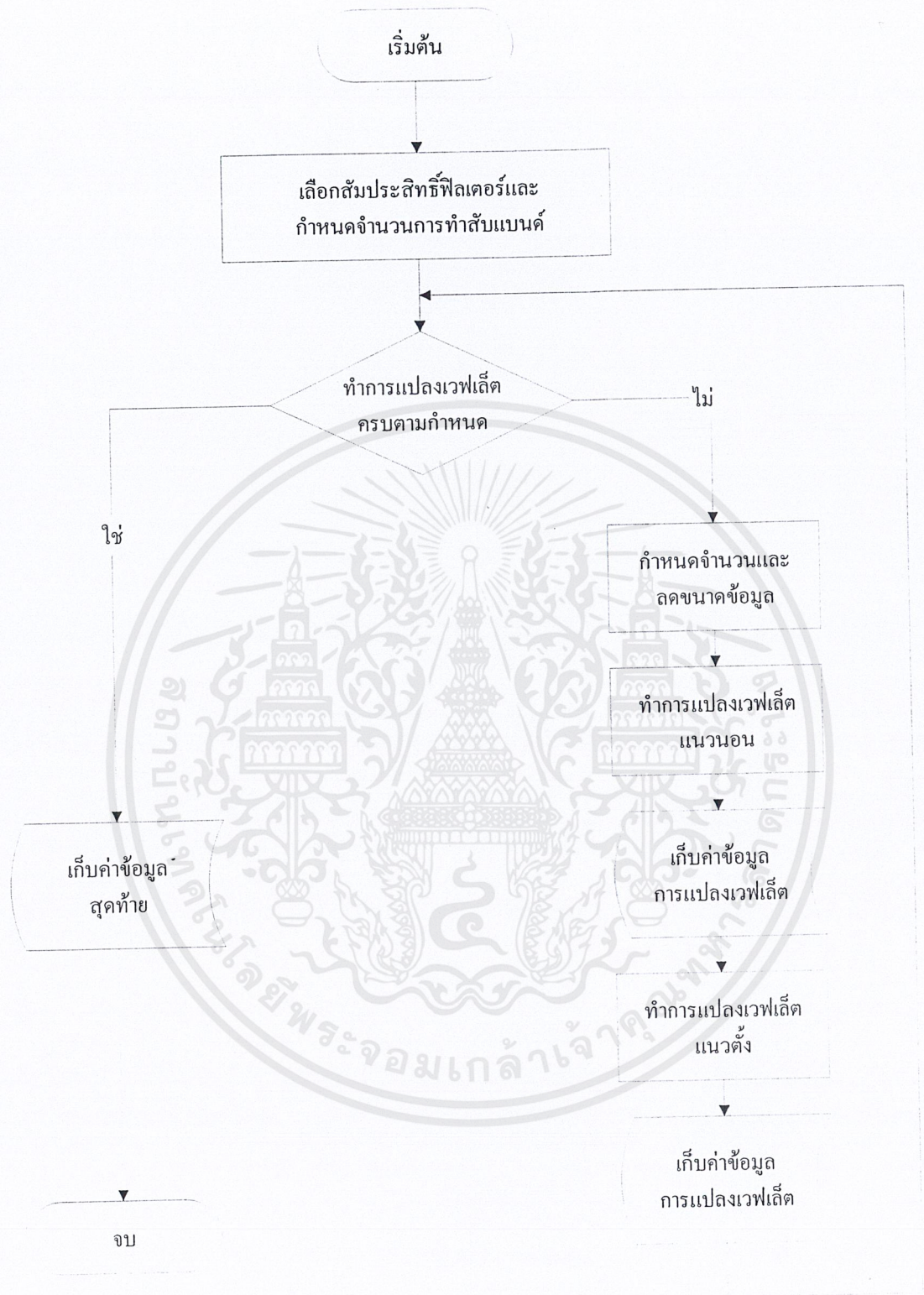
ตัวอย่างที่ 3 แสดงการสังเคราะห์สัญญาณกลับจากสัมประสิทธิ์ของข้อมูลหลังการแปลงเวฟเล็ตคือ $\left\{ \sqrt{2}, -\sqrt{2}, 1, -1, \frac{1}{\sqrt{2}}, -\frac{5}{\sqrt{2}}, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right\}$ ด้วยฮาร์ฟิงก์ชัน



จะได้ข้อมูลเดิมกลับมาคือ 1, 0, -3, 2, 1, 0, 1 และ 2

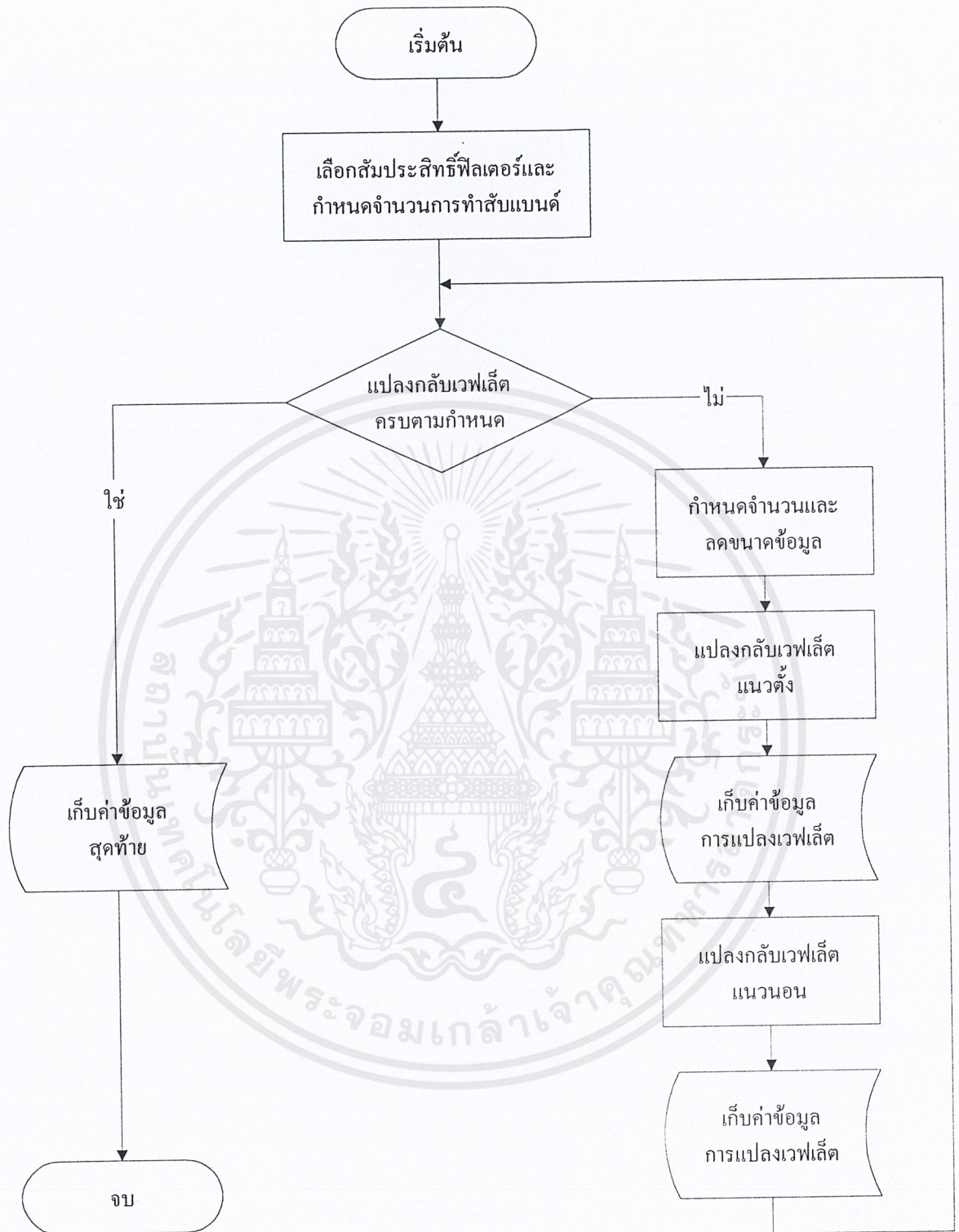
การออกแบบการแปลงเวฟเล็ต

จากการคำนวณการแปลงเวฟเล็ตที่กล่าวมา จะสามารถออกแบบโปรแกรมการคำนวณการแปลงเวฟเล็ตได้ดังโปรแกรมการแปลงเวฟเล็ตทั้งไปและกลับในภาคผนวก และมีผังการทำงานดังผังการทำงานที่ 3.1 และ ผังงานที่ 3.2 ตามลำดับ



ผังงานที่ 3.1 แสดงผังการทำงานการแปลงเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผังงานที่ 3.2 ผังการทำงานการแปลงกลับเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การกำหนดรหัสสัญลักษณ์ให้กับสัมประสิทธิ์เวฟเล็ต

จากหัวข้อที่ 2.4.2 ในบทที่ 2 การเข้ารหัสแบบเอ็มเบดซีโรทรี (EZW) ที่มีกระบวนการจัดสัมประสิทธิ์เป็นโดมิแนนท์ และ สับออร์ดิเนท สามารถที่จะเขียนเป็นผังงานได้ดังผังงานที่ 3.3 จะสามารถตัวอย่างการนำข้อมูลสัมประสิทธิ์มาจัดการเข้ารหัสแบบเอ็มเบดซีโรทรี ได้ดังนี้

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

รูปที่ 3.2 แสดงสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ต

พิจารณารูปที่ 3.2 ที่เกิดจากการแปลงเวฟเล็ตของภาพขนาด 8×8 เป็น 3 ระดับค่าสัมประสิทธิ์จากการแปลงที่มีค่าสูงสุด คือ 63 เลือกค่าเทรชโฮลเริ่มต้น (τ_0) จากครึ่งหนึ่งของค่าสูงสุด คือ $\tau_0 = 63/2 = 31.5$ เลือกค่าเทรชโฮลในรูปสองยกกำลัง นั่นคือ $\tau_0 = 2^5 = 32$ เริ่มต้น จะทำวนการโดมิแนนท์ก่อนคือ ทำการสแกนค่าสัมประสิทธิ์ตามรูปที่ 3.2 โดยทิศทางของการสแกนจะมีลักษณะเป็นแบบตัวซี (ดังในรูปที่ 2.26) ซึ่งจะพบค่าสัมประสิทธิ์ตัวแรกมีขนาดเท่ากับ 63 โดยมีค่าของสัมประสิทธิ์มากกว่าค่าเทรชโฮลเริ่มต้น ซึ่งจะถือว่าเป็นค่านัยสำคัญ และจะกำหนดสัญลักษณ์เป็น POS เพราะค่าสัมประสิทธิ์เป็นบวก เมื่อสแกนต่อไป จะพบสัมประสิทธิ์มีค่าเท่ากับ -34 และมีค่าสัมบูรณ์ที่มากกว่าค่าเทรชโฮลเริ่มต้นแต่ค่าเป็นลบจะถูกกำหนดด้วยสัญลักษณ์เป็น NEG ถ้าสแกนต่อค่าสัมบูรณ์ของ -31 น้อยกว่าค่าเทรชโฮลเริ่มต้น แต่เมื่อมองในสับแบนด์สเกลของสัมประสิทธิ์ของทั้ง โหนดลูกและ โหนดสืบทอด แล้วมีสัมประสิทธิ์บางตัวมากกว่าค่าเทรชโฮลเริ่มต้น ดังนั้นตำแหน่งนี้จะถูกกำหนดให้แทนด้วยสัญลักษณ์ คือ ไอโซเลทซีโร (IZ) จากนั้นค่า 23 ที่ถูกสแกนต่อมา มีค่าน้อยกว่าค่าเทรชโฮล รวมทั้งสัมประสิทธิ์ทั้งหมดของสับแบนด์ โหนดลูก และ โหนดสืบทอดมีค่าน้อยกว่าค่าเทรชโฮลตำแหน่งนี้ก็จะถูกกำหนดด้วย รากซีโรทรี (ZTR) ทำให้ไม่ต้องพิจารณาสัมประสิทธิ์ใน โหนดลูก และ โหนดสืบทอดของโดมิแนนท์นี้อีก แล้วทำการสแกนต่อไปอีกโดยจะพิจารณาตามหลักการที่เหมือนกันกับหลักการที่กล่าวมาข้างต้น โดยสามารถสรุปเป็นตารางของขบวนการทำโดมิแนนท์ในการสแกนหนึ่งครั้ง ดังตารางที่ 3.1

ตารางที่ 3.1 แสดงค่าของสัมประสิทธิ์ที่มีนัยสำคัญและไม่สำคัญในเอ็มเบดซีโรทรี

Subband	ค่าของ สัมประสิทธิ์	Symbol	ค่าที่สร้างกลับ
LL3	63	POS	48
HL3	-34	NEG	-48
LH3	-31	IZ	0
HH3	23	ZTR	0
HL2	49	POS	48
HL2	10	ZTR	0
HL2	14	ZTR	0
HL2	-13	ZTR	0
LH2	15	ZTR	0
LH2	14	IZ	0
LH2	-9	ZTR	0
LH2	-7	ZTR	0
HL1	7	Z	0
HL1	13	Z	0
HL1	3	Z	0
HL1	4	Z	0
LH1	-1	Z	0
LH1	47	POS	48
LH1	-3	Z	0
LH1	-2	Z	0

ตารางที่ 3.2 แสดงค่าของสัมประสิทธิ์ที่มีค่านัยสำคัญ

ขนาดของสัมประสิทธิ์	Symbol	ขนาดสัมประสิทธิ์ที่สร้างกลับ
63	1	56
34	0	40
49	1	56
47	0	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 3.1 สัมประสิทธิ์ที่เป็นตัวนัยสำคัญ (*POS* และ *NEG*) จะมาเข้าขบวนการสับออร์ดิเนทจะนำมาทำการควอนไทซ์ โดยหาค่ากลางจาก $\frac{\tau_0 + 2\tau_0}{2} = \frac{32 + 64}{2} = 48$ ดังนั้นค่า 63 จะมากกว่าค่ากลางคือ 48 และถูกกำหนดสัญลักษณ์ด้วย “ 1 .” และค่าสัมบูรณ์ของ -34 จะน้อยกว่า 48 ถูกกำหนดสัญลักษณ์ด้วย “0” ดังตารางที่ 3.2

เมื่อทำการสแกนครั้งที่สอง ในขบวนการโดมิแนนท์ ค่าเทรชโฮล จะถูกกำหนดใหม่จากสมการ 1 คือ $\tau_1 = \frac{\tau_0}{2} = 16$ สัมประสิทธิ์ตัวที่ถูกกำหนดสัญลักษณ์ *POS* หรือ *NEG* และจะถูกสแกนข้าม และแทนค่าเป็นศูนย์เพื่อให้เกิดกลุ่มข้อมูลที่เป็นศูนย์เพิ่มขึ้นส่วนในขบวนการสับออร์ดิเนทครั้งที่สอง สัมประสิทธิ์ที่มากกว่า ค่า τ_1 ก็จะถูกนำมาจัดค่าระดับใหม่ จะกระทำเช่นนี้จนกระทั่งได้ เงื่อนไขตามต้องการจึงจะหยุดการทำสแกนข้อมูล ผลที่ได้จากตัวอย่างข้างต้น สามารถแสดงได้ในตารางที่ 3.3

ตารางที่ 3.3 แสดงค่าของสัมประสิทธิ์ที่มีค่านัยสำคัญ

ขนาดของสัมประสิทธิ์	Symbol	ขนาดสัมประสิทธิ์ที่สร้างกลับ
63	11	60
34	00	36
49	10	52
47	01	44
31	1	28
23	0	20

จากผลในตารางที่ 3.1 และ 3.2 จะนำสัญลักษณ์มาจัดเรียงตามลักษณะการสแกนแบบตัวชี้ (*Z*) คือ ขบวนการโดมิแนนท์จะได้ลำดับของสัญลักษณ์ดังนี้

$$\text{ลำดับของสัญลักษณ์ในโดมิแนนท์} = \{ POS, NEG, IZ, ZTR, POS, ZTR, ZTR, ZTR, ZTR, ZTR, IZ, ZTR, ZTR, Z, Z, Z, Z, Z, POS, Z, Z \}$$

และลำดับของสัญลักษณ์ในขบวนการของสับออร์ดิเนทจะได้ดังนี้

$$\text{ลำดับของสัญลักษณ์ในสับออร์ดิเนท} = \{ 1, 0, 1, 0 \}$$

เมื่อนำลำดับของสัญลักษณ์ทั้งโดมิแนนท์และสับออร์ดิเนทมารวมกัน จะได้ลำดับของสัญลักษณ์ใหม่ คือ

$$\text{ลำดับของสัญลักษณ์} = \{ POS, 1, NEG, 0, IZ, ZTR, POS, 1, ZTR, ZTR, ZTR, ZTR, IZ, ZTR, ZTR, Z, Z, Z, Z, Z, POS, 0, Z, Z \}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และลำดับของชุดสัญลักษณ์นี้เกิดจากการทำขบวนการของโดมิแนนท์และสับออร์ดิเนทหนึ่งครั้ง เพื่อให้ data stream มีรหัสสัญลักษณ์น้อยลง โดยการแทนสัญลักษณ์ $\{POS, NEG, IZ, ZTR\}$ ด้วย $\{01, 10, 11, 00\}$ ตามลำดับ ดังนั้นจะได้ลำดับของสัญลักษณ์ใหม่ คือ

$$\text{ลำดับของสัญลักษณ์} = \{01, 1, 10, 0, 11, 00, 01, 1, 00, 00, 00, 00, 11, 00, 00, 00, 00, 00, 00, 00, 01, 0, 11, 11\}$$

และเมื่อการทำขบวนการของโดมิแนนท์และสับออร์ดิเนทครั้งที่สอง ผลที่ได้ในตารางที่ 1.3 จะถูกนำมาจัดเรียงอีกคือ

$$\text{ลำดับของสัญลักษณ์ในโดมิแนนท์ (ครั้งที่ 2)} = \{NEG, POS, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR\}$$

$$\text{ลำดับของสัญลักษณ์ในสับออร์ดิเนทครั้งที่ 2} = \{1, 0\}$$

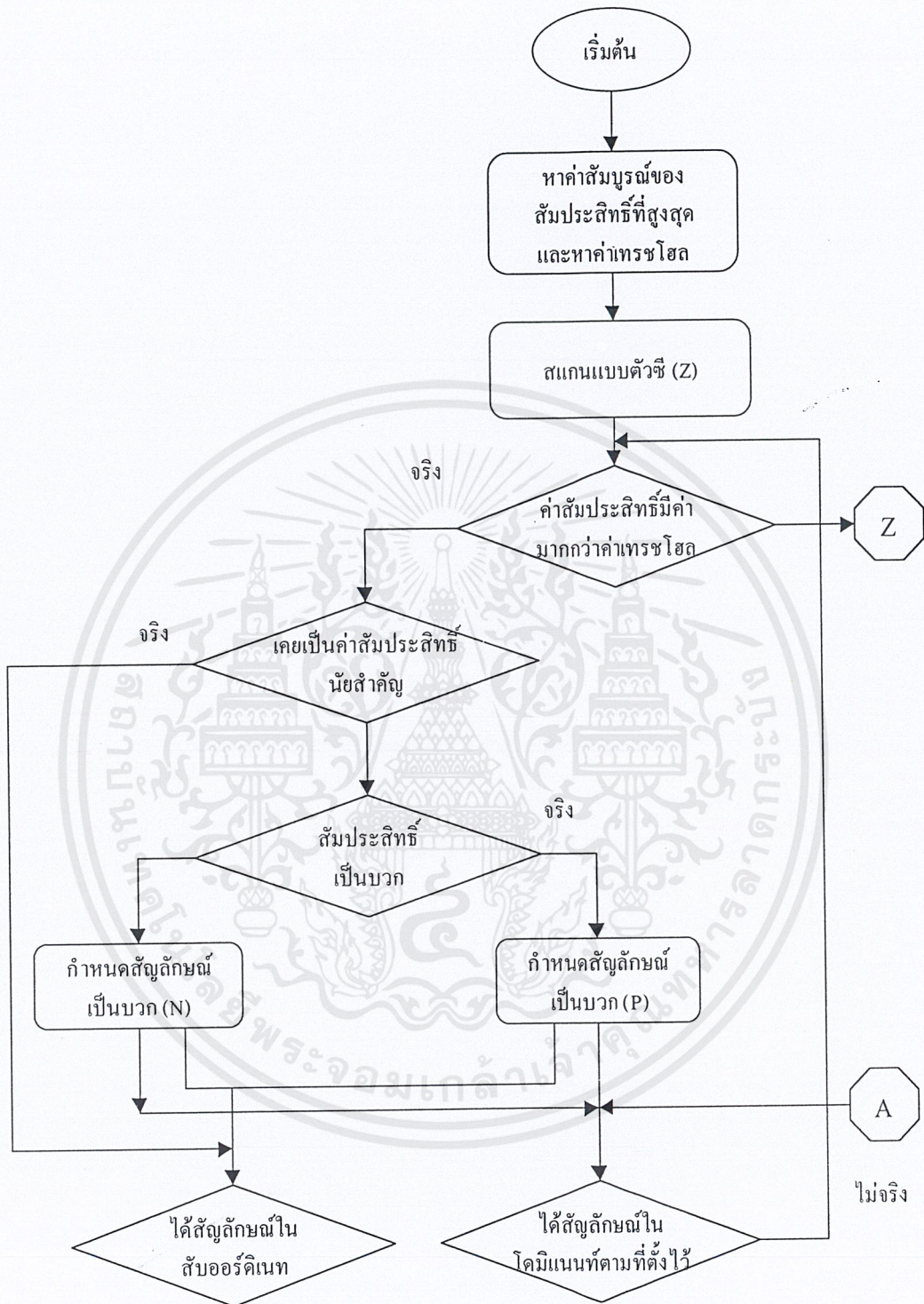
จากขบวนการโดมิแนนท์และสับออร์ดิเนทครั้งที่สอง รวมกันเป็นลำดับของสัญลักษณ์ ทั้งหมดคือ

$$\text{ลำดับของข้อมูล} = \{\text{รหัสสัญลักษณ์ในโดมิแนนท์และ รหัสสัญลักษณ์ในสับออร์ดิเนทที่ 1, รหัสสัญลักษณ์ในโดมิแนนท์และรหัสสัญลักษณ์ในสับออร์ดิเนทที่ 2}\}$$

$$\text{ลำดับของสัญลักษณ์} = \{POS, 11, NEG, 00, IZ, ZTR, POS, 11, ZTR, ZTR, ZTR, ZTR, IZ, ZTR, ZTR, Z, Z, Z, Z, Z, POS, 01, Z, Z, NEG, 1, POS, 0, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR\}$$

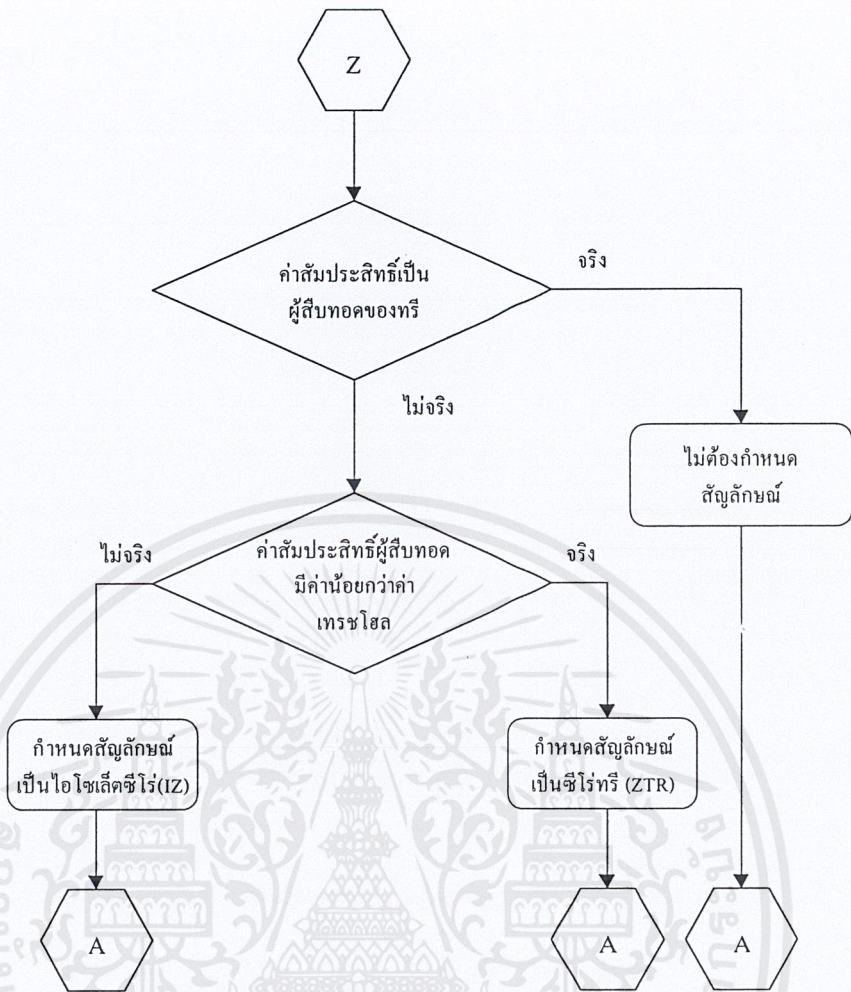
$$\text{ลำดับของสัญลักษณ์} = \{01, 11, 10, 00, 11, 00, 01, 10, 00, 00, 00, 00, 11, 00, 00, 00, 00, 00, 00, 01, 01, 00, 00, 10, 1, 01, 0, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00\}$$

ในการทำกลับของเอ็มเบดซีโรทรีจะมีขั้นตอนดังผังงานที่ 3.5 และผลจากการทำกลับของลำดับข้อมูล ที่ได้จากการสแกนสัมประสิทธิ์กลับจะแสดงในรูปที่ 3.3, 3.4 และ 3.5



ผังงานที่ 3.3 เซารหัสเอ็มเบดซีโรตรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผังงานที่ 3.4 การทำงานของ EZW ส่วนที่สอง

56	-40	56	0	0	0	x	x
0	0	x	x	0	0	x	x
0	0	0	0	x	x	x	x
x	x	x	x	x	x	x	x
x	x	0	40	x	x	x	x
x	x	0	0	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x

รูปที่ 3.3 ผลการทำกลับของการสแกนครั้งที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

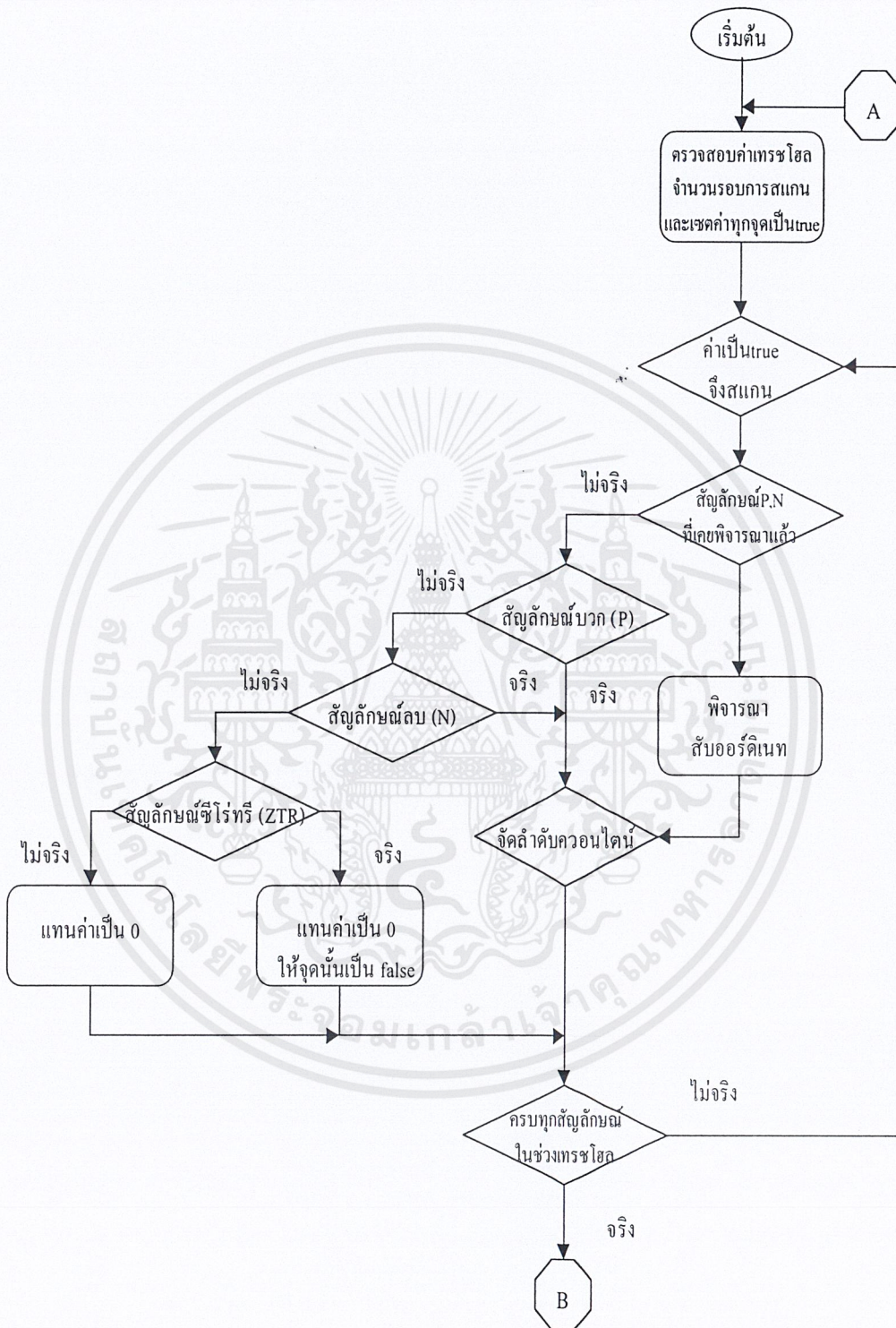
60	-36	52	0	0	0	x	x
-28	20	x	x	0	0	x	x
0	0	0	0	x	x	x	x
0	0	0	0	x	x	x	x
x	x	0	44	x	x	x	x
x	x	0	0	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x

รูปที่ 3.4 ผลการทำกลับของการสแกนครั้งที่ 2

62	-34	50	10	0	10	-14	0
-28	20	14	-14	0	0	0	0
14	14	0	-10	0	0	0	10
-14	0	-14	14	0	0	0	0
0	10	0	46	x	x	0	0
0	0	0	0	x	x	0	0
0	0	x	x	0	0	0	0
0	10	x	x	0	0	0	0

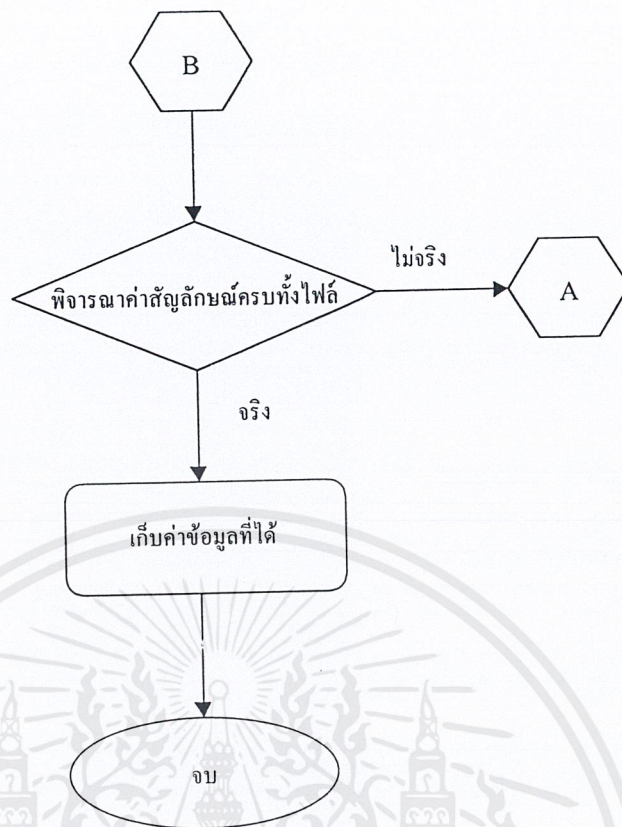
รูปที่ 3.5 ผลการทำกลับของการสแกนครั้งที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผังงานที่ 3.5 แสดงการทำกลับของเอ็มเบดซี โรทรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผังงานที่ 3.6 แสดงการทำกลับเอ็มเบดซีโรทรี (ต่อ)

3.4 การปรับปรุงวิธีการเข้ารหัสแบบเอ็มเบดซีโรทรี

ในการนำสัมประสิทธิ์ของภาพภายหลังการแปลงเวฟเล็ดมาแทนด้วยรหัสข้อมูลนั้นว่ามีความสำคัญมากในการลดขนาดข้อมูล เพราะถ้าจัดการรหัสแทนค่าของสัมประสิทธิ์ได้เหมาะสมแล้ว ก็จะสามารถที่จะลดขนาดข้อมูลได้มากยิ่งขึ้น ดังนั้น จึงได้ทำการปรับปรุงวิธีการเข้ารหัสแบบเอ็มเบดซีโรทรีใหม่ ดังนี้

พิจารณาการกำหนดสัญลักษณ์ในขบวนการโดมิแนนท์ตามลักษณะในผังงานที่ 3.3 และในวิธีการทำสับออร์ดิเนท แต่ละครั้งค่าสัมประสิทธิ์ที่มีค่ามากกว่าระดับของแต่ละโดมิแนนท์จะถูกแทนด้วยข้อมูลไบนารีเพียงบิตเดียวเท่านั้น และในการทำโดมิแนนท์ครั้งหนึ่งจะไม่ทำการพิจารณาค่าของสับออร์ดิเนท คือ

จากตัวอย่างที่ผ่านมาเมื่อนำมาเข้ารหัสด้วยวิธีการที่ปรับปรุงขึ้นมาใหม่จะได้ดังนี้

ลำดับของสัญลักษณ์ในโดมิแนนท์ = { POS , NEG , IZ , ZTR , POS , ZTR , ZTR , ZTR , ZTR ,
ZTR , IZ , ZTR , ZTR , Z , Z , Z , Z , POS , Z , Z }

และลำดับของสัญลักษณ์ในขบวนการของสับออร์ดิเนทจะได้ดังนี้

ลำดับของสัญลักษณ์ในสับออร์ดิเนท = ไม่ทำการพิจารณาในครั้งที่หนึ่ง

ลำดับของสัญลักษณ์ในโดมิแนนท์ (ครั้งที่ 2) = { NEG , POS , ZTR , ZTR , ZTR , ZTR , ZTR ,
ZTR , ZTR , ZTR , ZTR , ZTR , ZTR }

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

1. ถ้าช่วงขอบเขตใหม่ไม่อยู่ในขอบเขตใด ขอบเขตหนึ่งในรูป $\left[0, \frac{1}{2}\right)$ หรือ

$\left[\frac{1}{4}, \frac{3}{4}\right)$ หรือ จะหยุดการทำซ้ำในรูปนี้

2. ถ้าช่วงขอบเขตใหม่อยู่ในช่วง $\left[0, \frac{1}{2}\right)$ จะกำหนดให้เอาต์พุตเป็น 0 แล้วทำการเพิ่มขนาดระยะขอบเขตของข้อมูลปัจจุบันในช่วง $\left[0, \frac{1}{2}\right)$ ให้เป็นสองเท่าแล้วขยายไปในช่วงทางด้านขวา

3. ถ้าช่วงขอบเขตใหม่อยู่ในช่วง $\left[\frac{1}{2}, 1\right)$ จะกำหนดให้เอาต์พุตเป็น 1 แล้วทำการเพิ่มขนาดระยะขอบเขตของข้อมูลปัจจุบันในช่วง $\left[\frac{1}{2}, 1\right)$ ให้เป็นสองเท่าแล้วขยายไปในช่วงทางด้านซ้าย

4. ถ้าช่วงขอบเขตอยู่ในช่วง $\left[\frac{1}{4}, \frac{3}{4}\right)$ จะไม่กำหนดให้มีเอาต์พุตเกิดขึ้น แต่จะกำหนดให้เกิดเอาต์พุตที่ตรงกันข้ามกับเอาต์พุตที่จะเกิดขึ้นในอนาคต แล้วทำการเพิ่มขนาดระยะขอบเขตของข้อมูลปัจจุบันในช่วง $\left[\frac{1}{4}, \frac{3}{4}\right)$ ให้เป็นสองเท่าแล้วขยายไปในสองข้างของจุดกึ่งกลาง

กระทำในขั้นตอนที่สองจนกว่าจะสิ้นสุดชุดของข้อมูลที่จะนำเข้ามาเข้ารหัส หรือเมื่อพบอักษรที่แสดงถึงจุดสิ้นสุดของข้อมูล

ในหลักการของ วิทเทน (Witten) เนล (Neal), และ เคลียร์ (Cleary) ได้กำหนดช่วงของการเข้ารหัส $[L, H)$ ไว้ในรีจิสเตอร์เพื่อการเข้ารหัส และหลักการที่นำมาใช้สำหรับแบ่งช่วงระยะขอบเขต เป็นไปตามสมการ 3.2 ถึง 3.4

$$R = (H-L) + 1 \quad (3.2)$$

$$H_{\text{new}} = L + R \left(\frac{Q[i]}{Q_{\text{total}}} \right) - 1 \quad (3.3)$$

$$L_{\text{new}} = L + R \left(\frac{Q[i-1]}{Q_{\text{total}}} \right) \quad (3.4)$$

โดย R คือ ระยะห่างของขอบเขต และ

H, L คือสองจุดของช่วงระยะในการเข้ารหัส

ซึ่งค่า R, L, และ H ทุกค่าจะถูกทำการปรับระดับจาก 1 จนเป็นค่า $2^r - 1$ โดย r ก็จะแสดงถึงความละเอียดของตัวแปร R, L, และ H ซึ่งเป็นการใช้การเข้ารหัสแบบใช้จำนวนเต็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยวิธีการเข้ารหัสเลขคณิตตามหลักการของ วิทเทน (Witten) เนล (Neal) และ เคลียร์รี (Cleary) สามารถเขียนเป็นคำสั่งจำลองได้ ดังนี้

คำสั่งจำลองที่ 3.1 แสดงการเข้ารหัสเลขคณิตตามวิธีการของ วิทเทน (Witten) เนล (Neal) และ เคลียร์รี (Cleary)

```

PROCEDURE Encode
BEGIN
    Initialize model and register L , H , R ;
    WHILE ( symbol o encode ){
        Get next symbol i ;
        R=( H - L )+1;
        H=L+( R * Q[I] / Qtotal ) - 1;
        L=L+( R * Q[I-1] / Qtotal );
        ALWAYS{%% Renormalize interval
            IF( H<Half ) {
                Output bit 0 and FOLW bits of 1;
            }
            ELSEIF ( L≥Half ) {
                Output bit 0 and FOLW bits of 0;
                L = L - Half ;
                H = H - Half ; }
            ELSEIF ( H<3rd _qrt and L≥1st _qrt ) {
                L = L-1st _qrt;
                H = H-1st _qrt;
                FOLW = FOLW+1 ; }
            ELSE BREAK;%%renormalize over
            L = L * 2 ;
            H = H * 2 + 1 ;
        }
    }
    IF ( End of file ) {
        Output rest bits in coder register;
        TEAMINATE ;;}
    Update model ;
}

END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งจำลองที่ 3.2 แสดงการเข้ารหัสเลขคณิตตามวิธีการของ วิทเทน (Witten) , เนล (Neal) , และ เคลียร์ (Cleary)

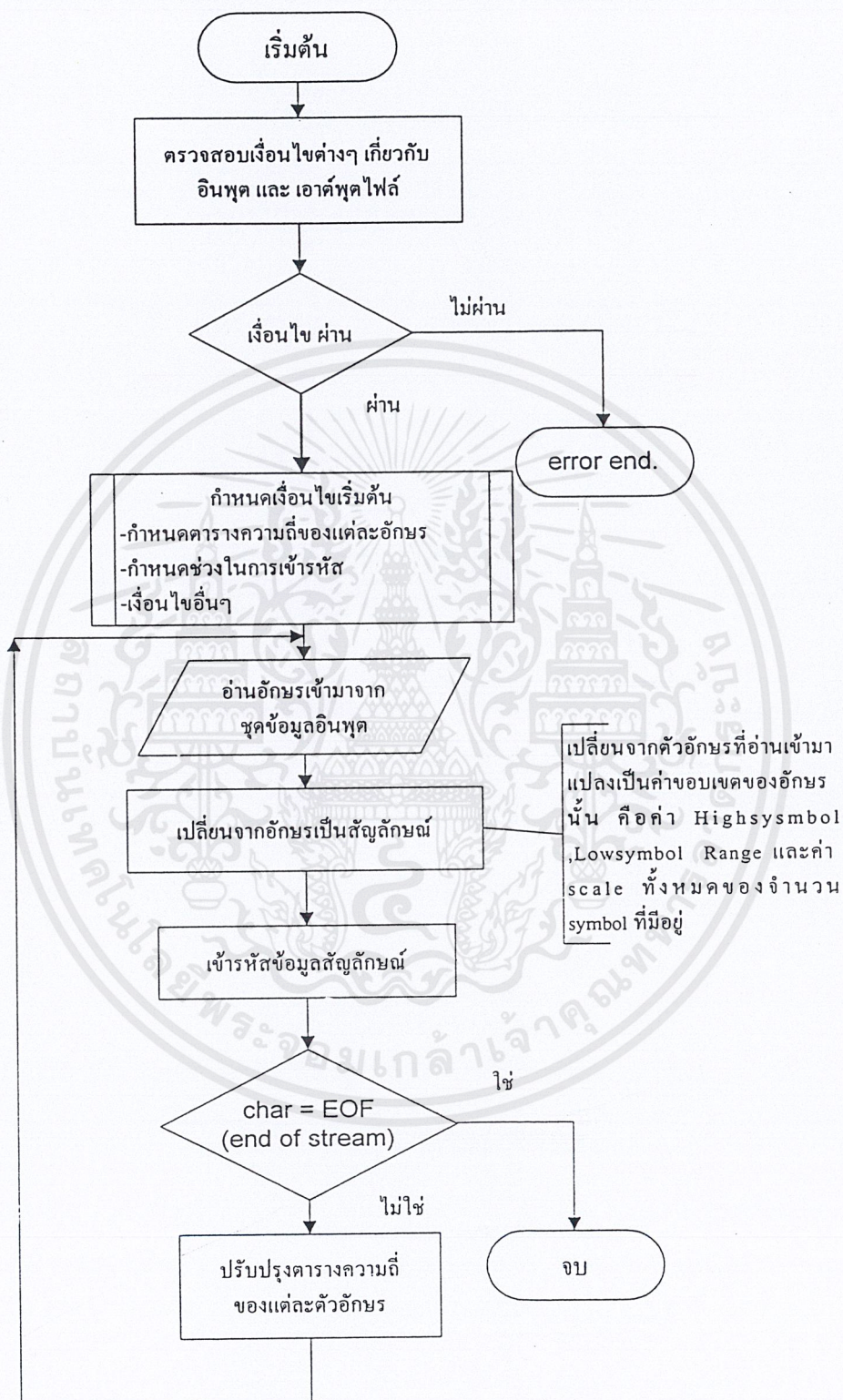
```

PROCEDURE Decode
BEGIN
    Initialize model and register L,H,R,C;
    ALWAYS{
        R = (H - L) + 1;
        V = ((C - L + 1) / R) * Q[0];
        Find symbol x such that  $Q[x-1] \leq V < Q[x]$ ;
        ALWAYS{%% Renormalize interval
            IF (H  $\leq$  Half)
                %% Do nothing here
            ELSEIF (L  $\leq$  Half) {
                L = L - Half;
                H = H - Half;
                V = V - Half;}
            ELSEIF (H  $<$  3rd _qrt and L  $\geq$  1st _qrt) {
                L = L - 1st _qrt;
                H = H - 1st _qrt;
                V = V * 2 + input_bit();}
            }
        Output symbol x;
        IF (End of file) TERMINATE;
        Update model;
    }
END.

```

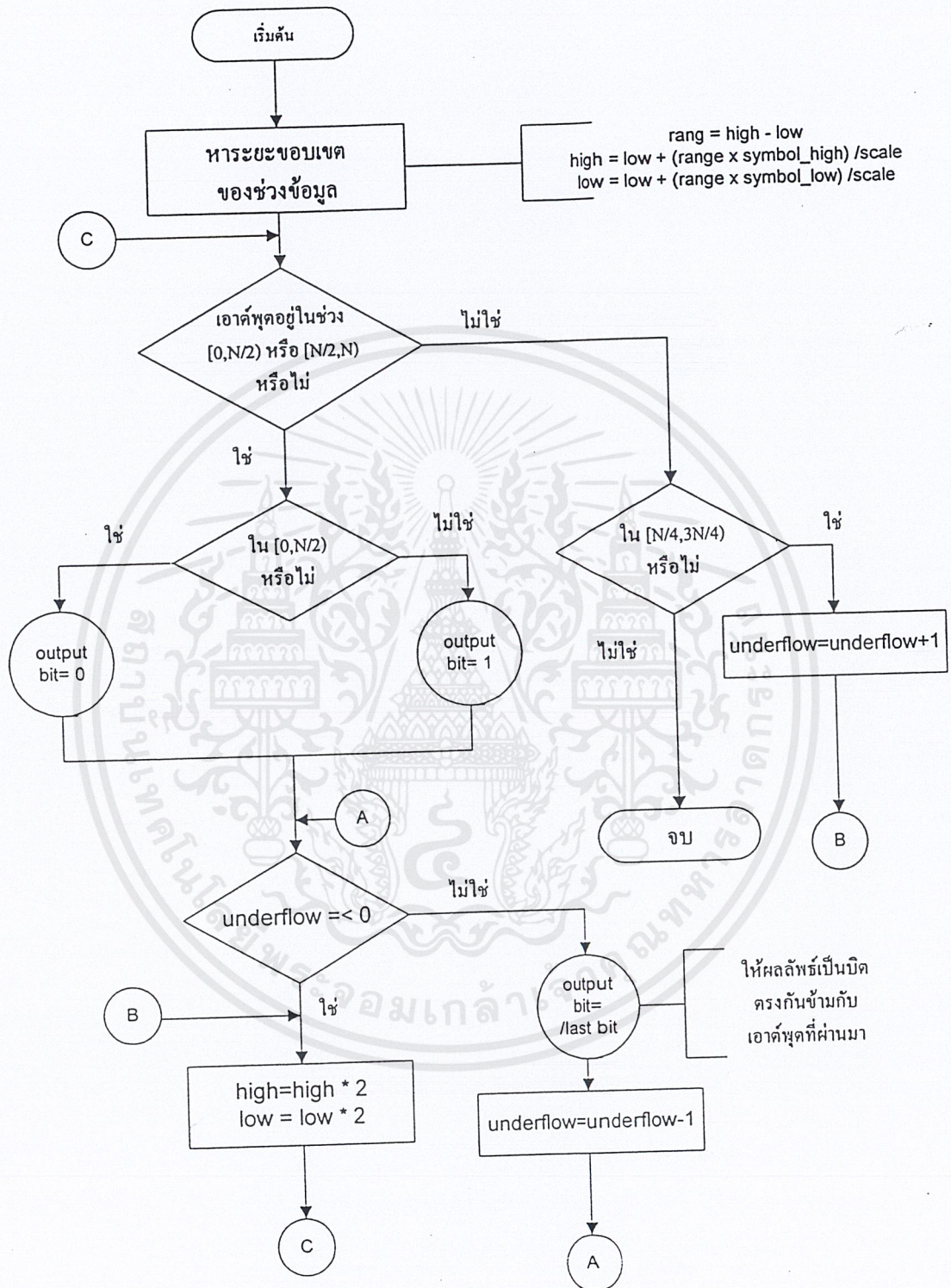
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผังการทำงานของโปรแกรม โดยสังเขปดังนี้



ผังงานที่ 3.7 แสดงผังงานการเข้ารหัสเลขคณิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



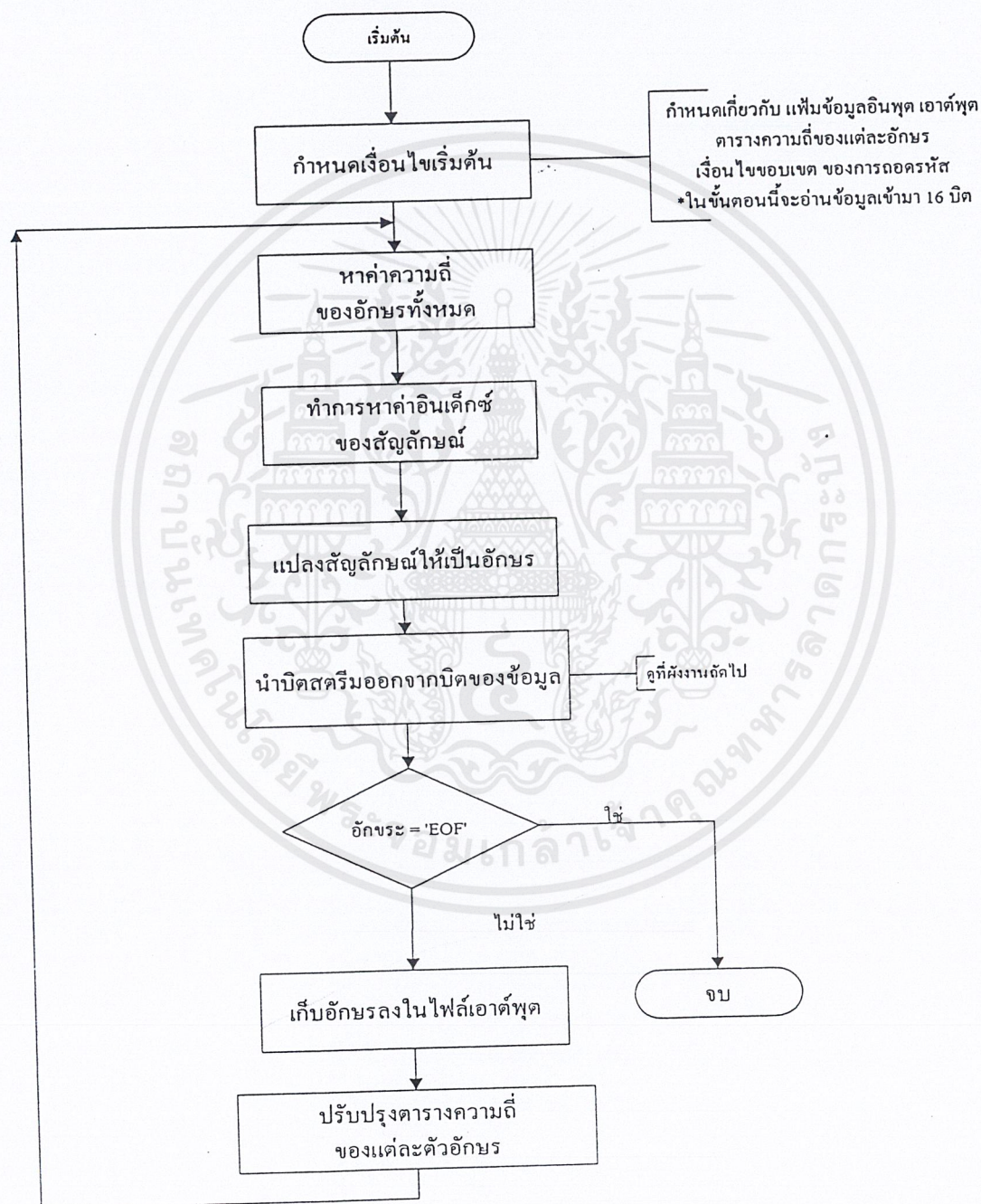
ผังงานที่ 3.8 แสดงรูปย่อการเข้ารหัสข้อมูลสัญลักษณ์

วิธีการข้างต้นนี้พบว่าจะสามารถป้องกันการลัดช่วงระยะขอบเขตได้ระดับหนึ่ง แต่เมื่อนำมาใช้กับระยะขอบเขตเริ่มต้นที่ทำการกำหนดค่าเริ่มต้นเป็นจำนวนเต็มที่มีค่าสูงๆก็จะป้องกันการเกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

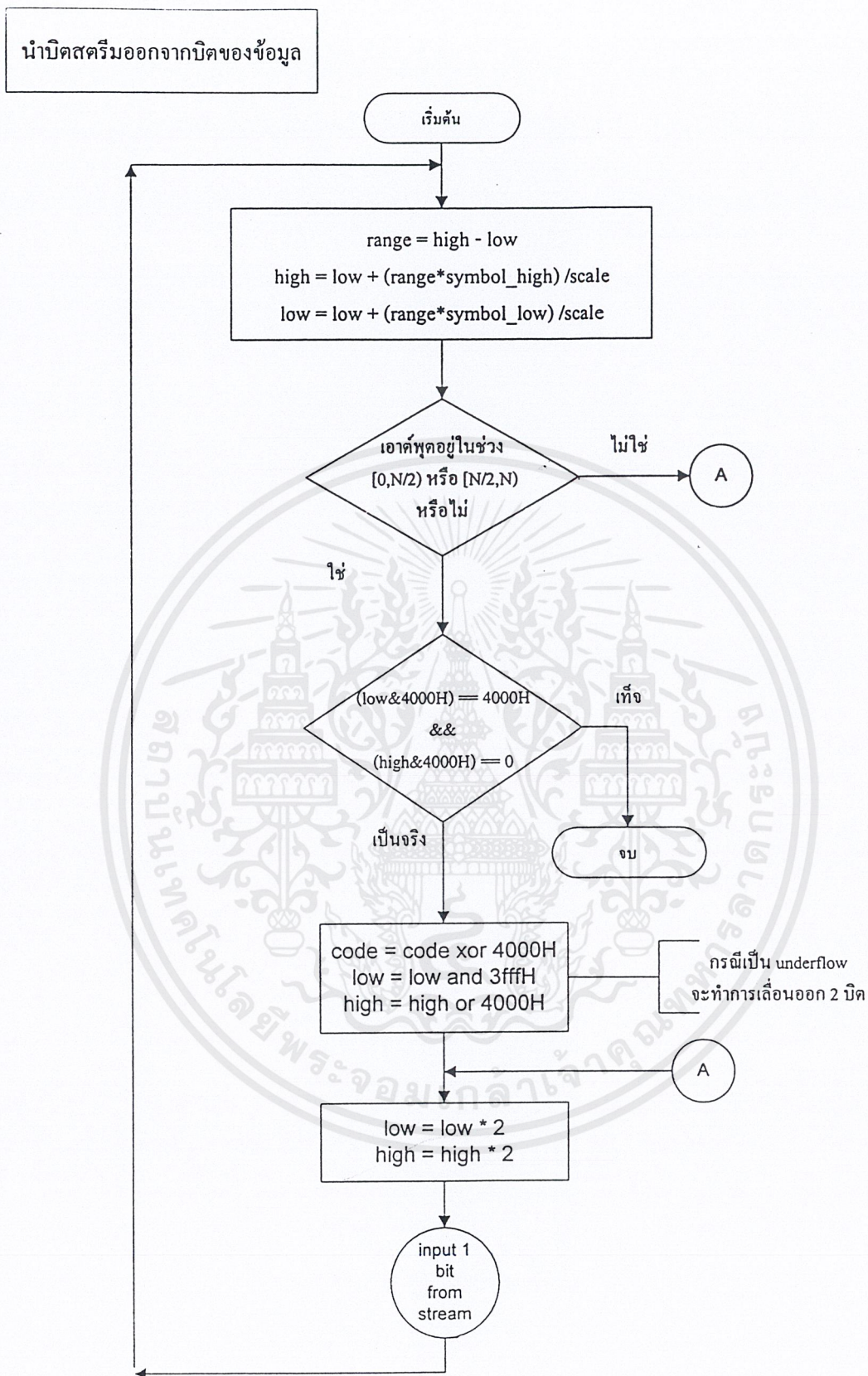
โพลตติ้งพอยต์ได้ ซึ่งค่าจำนวนเต็มที่จะกำหนด เพื่อความสะดวกในการเขียนโปรแกรมเพื่อที่จะได้สะดวกต่อการดำเนินการทางบิต จึงควรกำหนดอยู่ในช่วงของค่า 2ⁿ - 1 ซึ่งในโปรแกรมที่อยู่ในภาคผนวก

ส่วนในการถอดรหัสดังที่ได้กล่าวมาแล้วในหัวข้อที่ 2.5.2 ในการนำมาเขียนเป็นโปรแกรมเพื่อใช้งานจะอยู่อาศัยหลักการของการเคลื่อนบิตข้อมูลเข้ามาทีละบิตแล้วทำการนำมาเปรียบเทียบกับช่วงของโอกาส การเกิดขึ้นของสัญลักษณ์ในช่วงต่างๆ ว่าตรงกับช่วงใด โดยจะอาศัยวิธีการของ วิทเทน (Witten) เนล (Neal) และ เคลียร์ (Cleary) ในการเขียนโปรแกรม ซึ่งสามารถเขียนเป็นผังงาน ได้ดังนี้



ผังงานที่ 3.9 แสดงผังงานในการถอดรหัสเลขคณิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผังงานที่ 3.10 แสดงผังงานในการนำบิตข้อมูลออกจากชุดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การส่งภาพแบบโปรเกรสซีฟโดยใช้การแปลงเวฟเล็ต

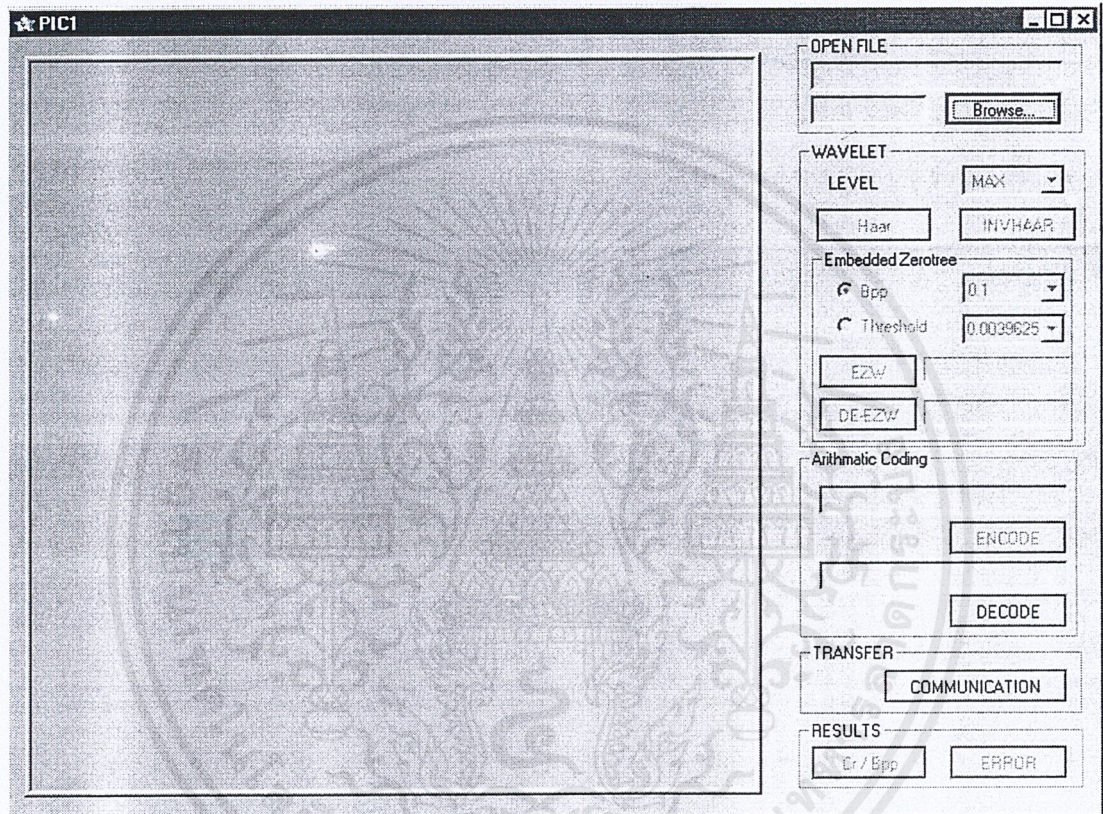
การส่งภาพแบบโปรเกรสซีฟเป็นการส่งภาพซ้อนกันไปเรื่อยๆ โดยในการส่งครั้งแรกๆ ภาพจะมีลักษณะหยาบและค่อยๆ มีความคมชัดมากขึ้นในการส่งครั้งต่อไป และเมื่อทำการพิจารณาภาพทั้งหมดแล้วพบว่าภาพในการส่งออกหน้าเหมือนกับเป็นภาพในการส่งครั้งต่อไปที่ผ่านตัวกรองความถี่ต่ำ ดังนั้นจึงสามารถทำการส่งภาพแบบโปรเกรสซีฟได้อีกวิธีหนึ่งโดยนำการแปลงเวฟเล็ตมาใช้เป็นตัวกรอง เพราะการแปลงเวฟเล็ตนี้สามารถเทียบได้กับการทำการแบ่งแบนด์ความถี่ของภาพนั่นเอง

การส่งภาพแบบโปรเกรสซีฟจะทำการส่งข้อมูลภาพในส่วนที่สำคัญที่สุดของภาพออกไปก่อน ดังนั้นข้อมูลทั้งหมดต้องถูกตัดแบ่งออกเป็นส่วนๆ แล้วนำมาเข้ารหัสตามลำดับความสำคัญของข้อมูล ซึ่งการจัดเรียงในลักษณะของข้อมูลในโดเมนความถี่จะมีความเหมาะสมมากกว่า เพราะนอกจากจะถูกจัดเรียงตามลำดับความถี่ต่ำแสดงถึงโครงร่างของภาพ ดังนั้น ถ้าเลือกเฉพาะข้อมูลทำให้เกิดภาพในส่วนที่เป็นขอบหรือรายละเอียดเหมือนผ่านตัวกรองความถี่สูง และเนื่องจากการแปลงข้อมูลภาพโดยอาศัยภาพออกมาเป็นแบนด์ต่างๆ กัน และวิธีการเข้ารหัสแบบเอ็มเบดดิโรว์ทรีจะทำการเข้ารหัสสัมประสิทธิ์ที่มีค่าในแบนด์ของความถี่ต่ำก่อน แล้วพิจารณาไปยังแบนด์ของความถี่สูง โดยนำเอารหัสข้อมูลของแต่ละครั้งที่ทำโคมิแนนท์มาเข้ารหัสเลขคณิตแล้วทำการส่งข้อมูลออกไป ก็จะเป็นการส่งภาพแบบโปรเกรสซีฟ

3.6 การใช้งานโปรแกรมลดขนาดข้อมูลภาพโดยการแปลงเวฟเลตที่ใช้วิธีการของเอ็มเบดซีโรทรีบน

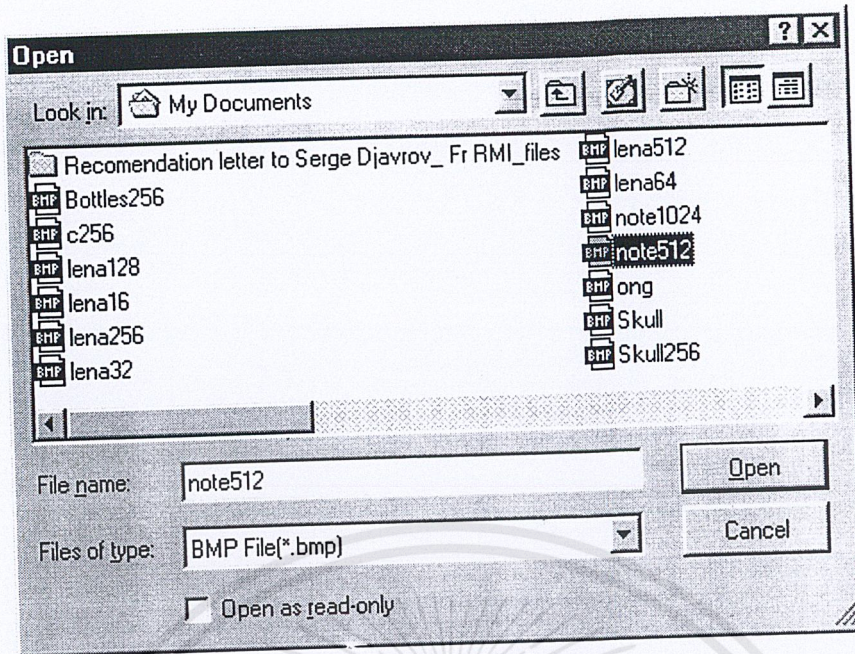
Vitual C++

ในการใช้งานตัวโปรแกรมเพื่อทำการลดขนาดข้อมูลภาพที่ใช้ในการแปลงเวฟเลต ได้ทำการเขียนโปรแกรมลงบน Vitual C++ โดยใช้ไดอะล็อกบ็อกซ์ (dialog box) สำหรับการกระทำกระบวนการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเลต ซึ่งใช้วิธีการของเอ็มเบดซีโรทรี โดยเมื่อทำการเปิดโปรแกรมจะแสดงหน้าจอไดอะล็อกบ็อกซ์ดังภาพที่ 3.6



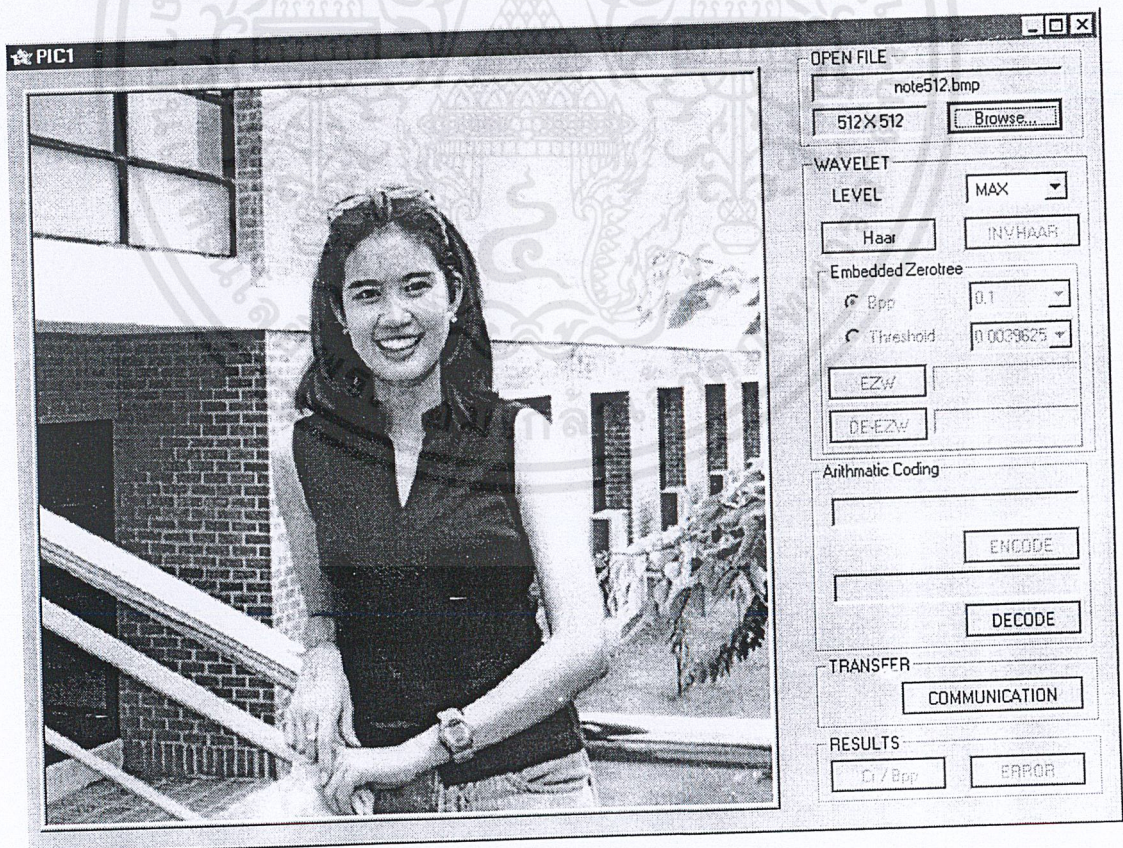
รูปที่ 3.6 หน้าจอไดอะล็อกบ็อกซ์สำหรับโปรแกรมลดขนาดข้อมูลภาพโดยการแปลงเวฟเลตที่ใช้วิธีการของเอ็มเบดซีโรทรี

เมื่อทำการกดปุ่มเลือกภาพ (Browse) ก็จะปรากฏไดอะล็อกบ็อกซ์ สำหรับเลือกอินพุตภาพขึ้นมา ดังรูป 3.7



รูปที่ 3.7 แสดงไดอะล็อกบ็อกซ์ สำหรับการใส่อินพุตภาพเพื่อทำการลดขนาดข้อมูล

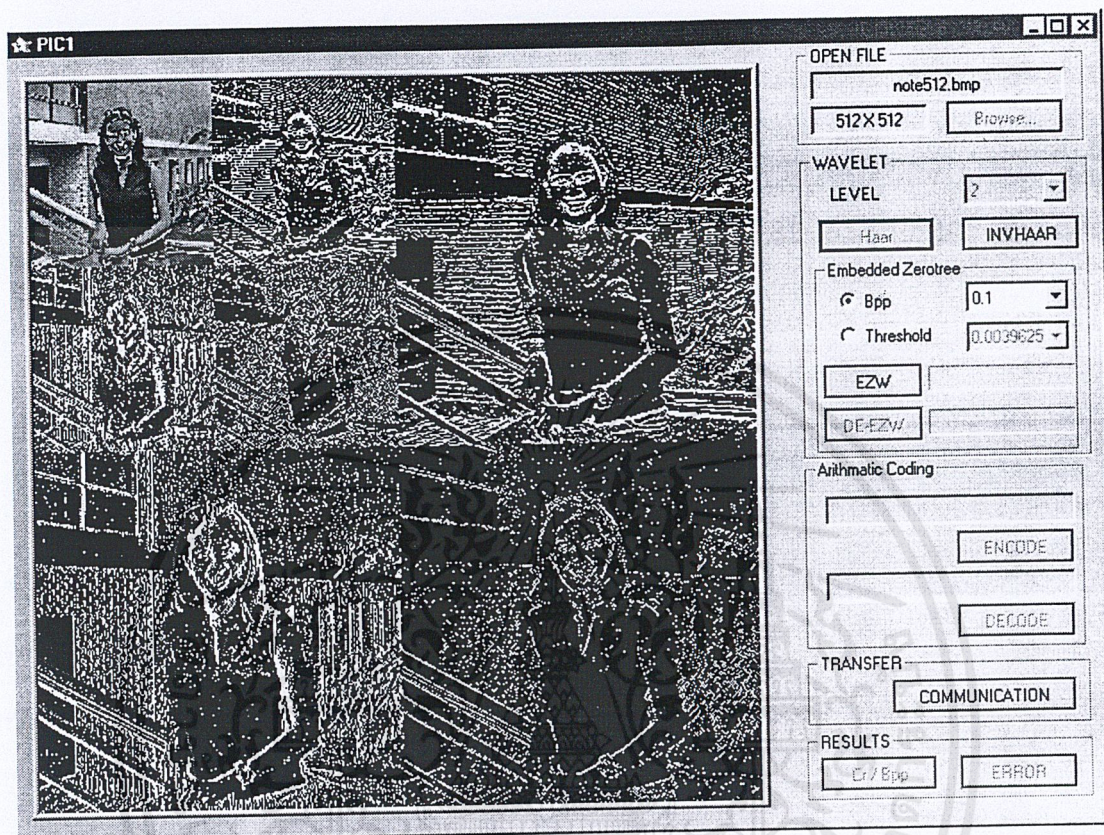
เมื่อทำการเปิดภาพแล้ว ไดอะล็อกบ็อกซ์จะแสดงภาพที่เลือกขึ้นมา ดังภาพ 3.8



รูปที่ 3.8 แสดงภาพเมื่อเลือกภาพจากไดอะล็อกบ็อกซ์

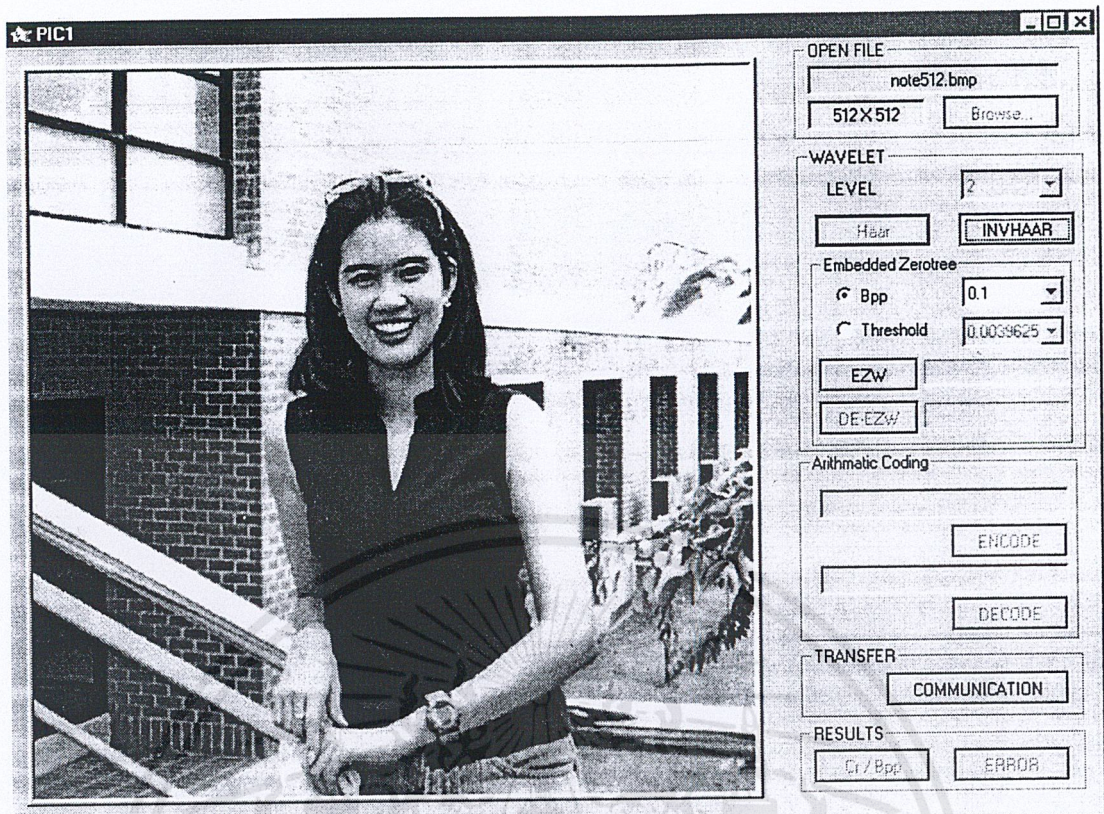
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการแปลงเวฟเล็ต เราสามารถเลือกระดับการแปลงได้ ซึ่งจะแสดงภาพที่ได้จากการแปลงเวฟเล็ตโดยค่า 1-9 จะเป็นจำนวนระดับการแปลงเวฟเล็ต ส่วนค่า 0 จะเป็นการแปลงเวฟเล็ตจนถึงระดับสุดท้าย โดยจะคำนวณจากค่าของขนาดภาพอินพุต ในภาพ 3.9 จะใช้ 2 ระดับ



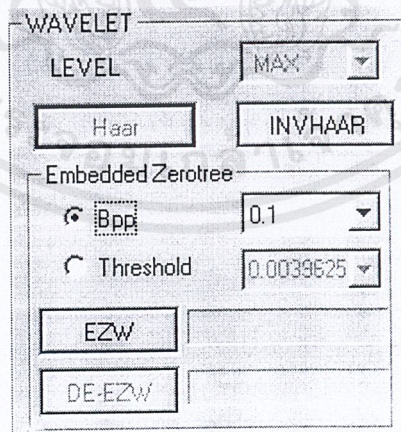
รูปที่ 3.9 แสดงภาพจากการแปลงเวฟเล็ตเมื่อเลือกระดับการแปลงเป็น สอง

และหากต้องการภาพเดิมกลับมาก็ต้องเลือกระดับการแปลงให้เหมือนกับค่าระดับการแปลงเวฟเล็ต ในการทำการแปลงกลับเวฟเล็ต จะแสดงภาพดังรูปที่ 3.10



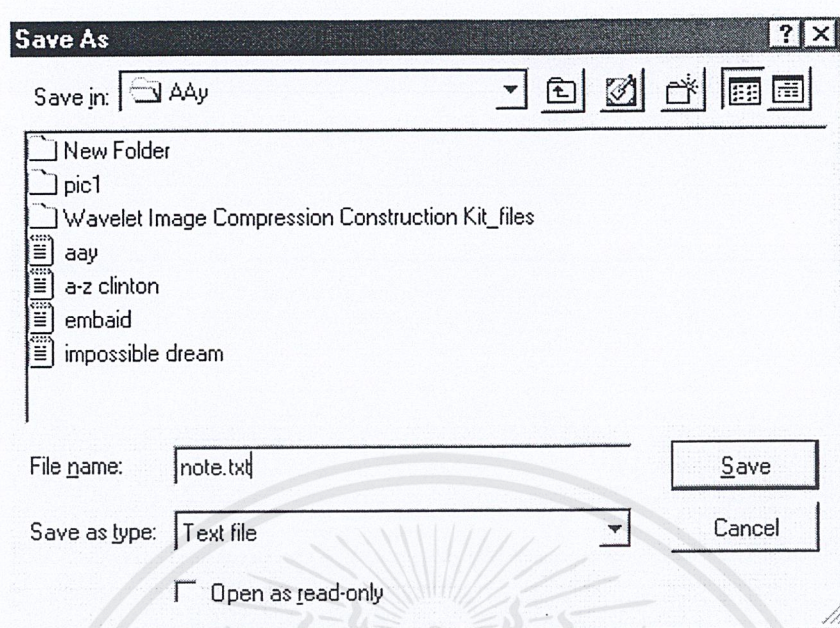
รูปที่ 3.10 แสดงภาพจากการแปลงกลับเวฟเล็ตเมื่อเลือกระดับการแปลงเป็น สอง

ในส่วนของการทำเอ็มเบดซีโรทรี จะต้องเลือกว่าจะใช้ ค่าเฉลี่ยของจำนวนบิตต่อจุดภาพ หรือ ค่าเทรชโฮล เป็นตัวกำหนดระดับความละเอียดของภาพ ดังรูปที่ 3.11

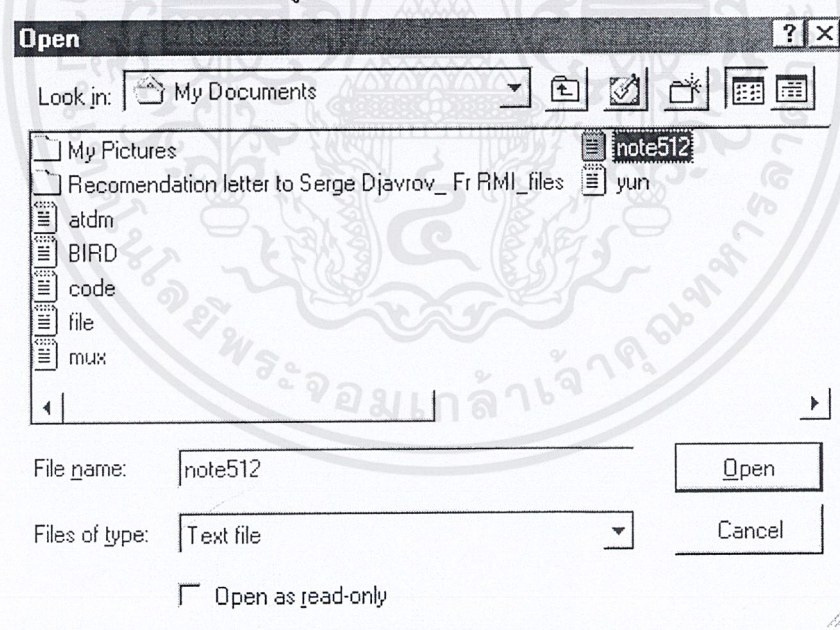


รูปที่ 3.11 แสดงการเลือกค่าเฉลี่ยของจำนวนบิตต่อจุดภาพ หรือ ค่าเทรชโฮลเพื่อ เป็นตัวกำหนดระดับความละเอียดของภาพ

ในส่วนของการลดขนาดข้อมูลที่ใช้วิธีการของเอ็มเบดซีโรทรี ทำได้โดยกดปุ่ม EZW ก็จะมี ไดอะล็อกบ็อกซ์ สำหรับกำหนดชื่อของข้อมูลที่ต้องการจะเก็บไว้ ดังรูปที่ 3.12
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



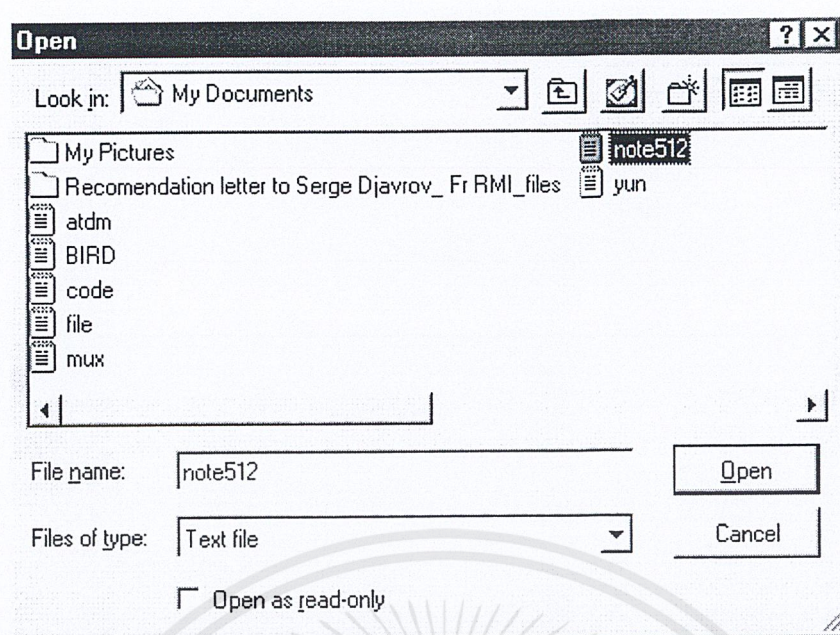
รูปที่ 3.12 แสดงไดอะล็อกบ็อกซ์ ที่ใช้ในการเก็บค่าจากการลดขนาดข้อมูลโดยเก็บไว้ที่ note.txt และหากต้องการนำข้อมูลเดิมกลับมา ก็กดปุ่ม DEEZW ก็จะมีไดอะล็อกบ็อกซ์ สำหรับใช้เลือกข้อมูล ที่ต้องการจะแปลงกลับมา ดังรูป 3.13



รูปที่ 3.13 แสดงไดอะล็อกบ็อกซ์ ที่ใช้ในการเลือกข้อมูล โดยเลือกข้อมูลที่เก็บไว้ที่ note.txt

ในส่วนของการเข้ารหัส เมื่อกดปุ่ม ENCODE ก็จะมีไดอะล็อกบ็อกซ์เพื่อใช้ในการเลือก ข้อมูลที่จะเข้ารหัส ขึ้นมา ดังรูป 3.14

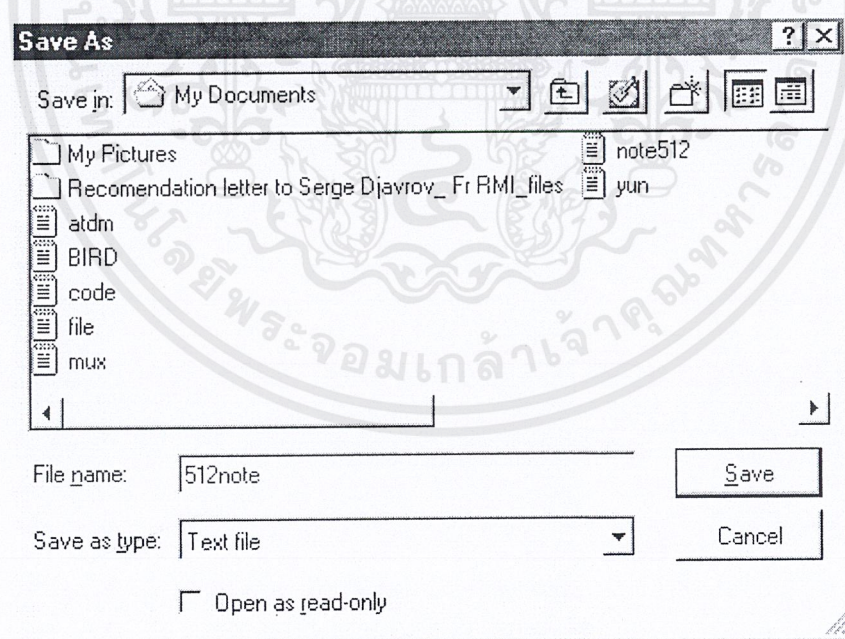
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงไดอะล็อกบ็อกซ์ที่ใช้ในการเลือกข้อมูลที่จะทำการเข้ารหัสโดยเลือกข้อมูลที่เก็บไว้ที่

note.txt

หลังจากนั้นก็จะมีไดอะล็อกบ็อกซ์ที่ใช้สำหรับกำหนดชื่อของข้อมูลที่ต้องการจะเก็บไว้หลังการเข้ารหัสขึ้นมา ดังรูปที่ 3.15

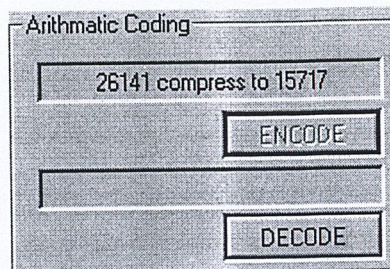


รูปที่ 3.15 แสดงไดอะล็อกบ็อกซ์ที่ใช้ในการกำหนดชื่อข้อมูลหลังจากการเข้ารหัสโดยเลือกข้อมูลมาเก็บ

ไว้ที่ 512note.txt

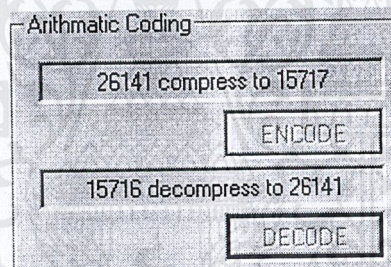
และจะมีช่องแสดงผลที่ได้จากการเข้ารหัส ดังรูป 3.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



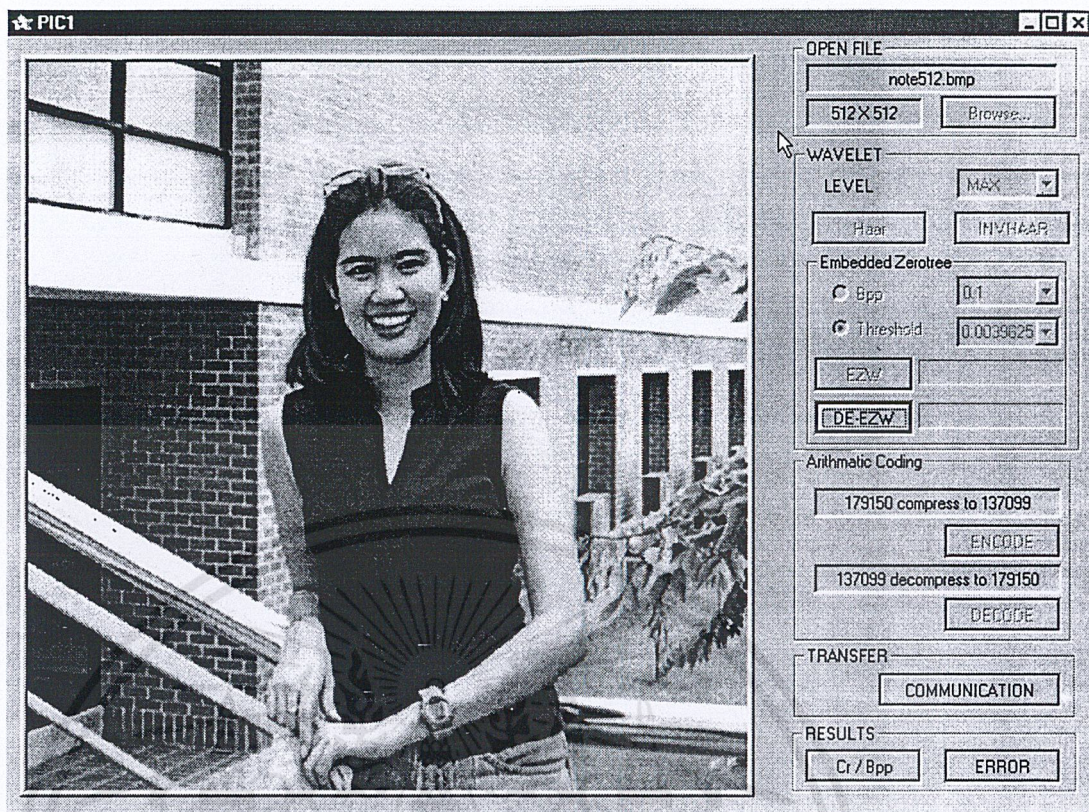
รูปที่ 3.16 แสดงผลที่ได้จากการเข้ารหัส

เมื่อกดปุ่ม DECODE ก็จะมีไดอะล็อกบ็อกซ์เพื่อใช้ในการเลือก ข้อมูลที่จะถอดรหัส ขึ้นมา หลังจากเลือกข้อมูลแล้ว ก็จะมีไดอะล็อกบ็อกซ์สำหรับการกำหนดชื่อข้อมูลที่จะเก็บหลังการถอดรหัส ซึ่งจะมีช่องแสดงผลหลังจากการถอดรหัส ดังรูป 3.17



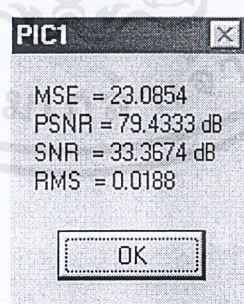
รูปที่ 3.17 แสดงผลที่ได้จากการถอดรหัส

หลังจากการทำกรแปลงกลับเอ็มเบดซีโรทรี จะได้ผลดังรูปที่ 3.18



รูปที่ 3.18 แสดงภาพที่ได้จากการแปลงกลับเอ็มเบดซีโรทรี

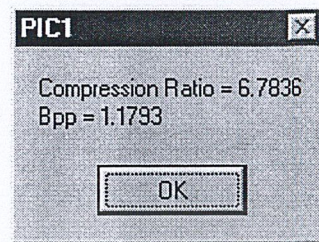
เมื่อต้องการตรวจสอบค่าความผิดพลาดของภาพที่ได้หลังการแปลงกลับเอ็มเบดซีโรทรี สามารถทำได้โดยกดปุ่ม ERROR แล้วทำการเลือกภาพต้นแบบเพื่อทำการเปรียบเทียบจะได้ผลดังรูปที่ 3.19



รูปที่ 3.19 แสดงเมตริกซ์ของค่าความผิดพลาด

เมื่อต้องการตรวจสอบค่าอัตราเข้ารหัสที่ได้หลังการแปลงกลับเอ็มเบดซีโรทรี สามารถทำได้โดยกดปุ่ม Cr/Bpp แล้วทำการเลือกภาพต้นแบบเพื่อทำการเปรียบเทียบและทำการเลือกไฟล์ที่เข้ารหัสแล้วจะได้ผลดังรูปที่ 3.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 แสดงเมตเสชบ็อกซ์ของค่าอัตราเข้ารหัส



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

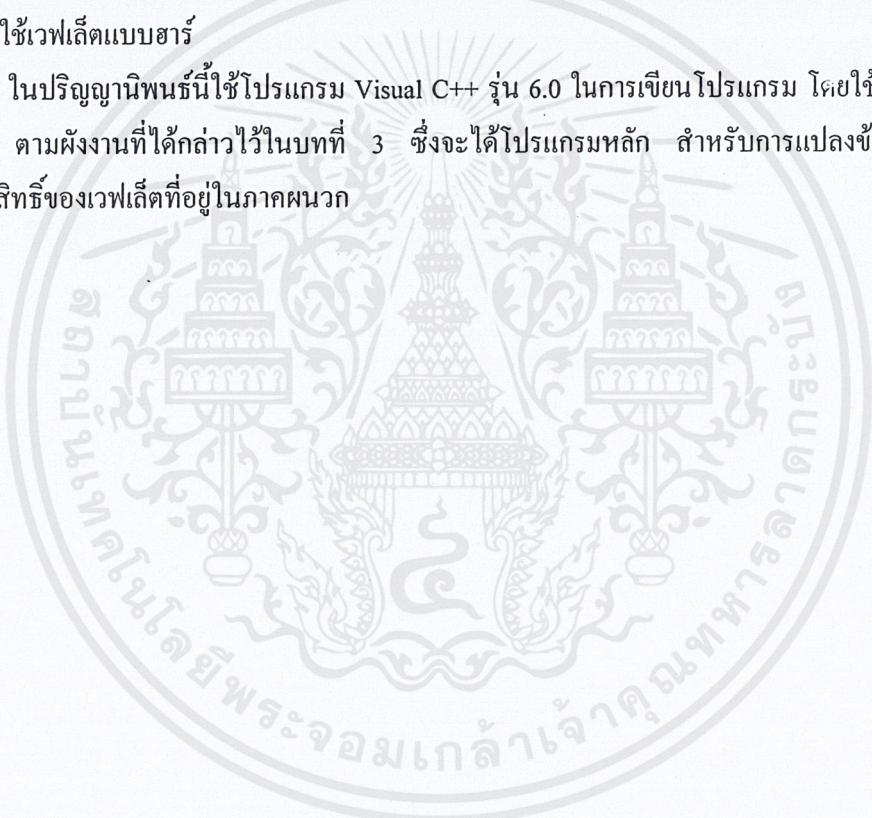
บทที่ 4

การทดลองและผลการทดลอง

การทดลองในบทนี้จะทำการลดขนาดข้อมูลภาพด้วยวิธีการเอ็มเบดซีโรว์ที่จะทำการนำสัมประสิทธิ์ที่ได้จากการนำเอาข้อมูลภาพมาผ่านการแปลงเป็นโมเดลของข้อมูล ด้วยวิธีการแปลงเวฟเล็ต เพื่อทดสอบผลของการลดขนาดข้อมูล

จากบทที่ 2 พบว่าฟังก์ชันเวฟเล็ตมีหลายรูปแบบ ซึ่งแต่ละรูปแบบจะให้ผลของการแปลงเวฟเล็ตที่แตกต่างกัน การที่จะเลือกใช้ฟังก์ชันเวฟเล็ตชนิดใดนั้น จะต้องทำการพิจารณาเลือกใช้ฟังก์ชันที่เหมาะสมกับฟังก์ชันสเกลลิง ที่จะนำมาใช้เป็นองค์ประกอบพื้นฐานของสัญญาณที่ต้องการวิเคราะห์ โดยจะขึ้นอยู่กับการที่จะนำไปใช้งาน โดยถ้าเป็นสัญญาณที่ไม่จำเป็นต้องพิจารณาถึงความต่อเนื่องของข้อมูลก็สามารถใช้เวฟเล็ตแบบฮาร์

ในปริญญานิพนธ์นี้ใช้โปรแกรม Visual C++ รุ่น 6.0 ในการเขียนโปรแกรม โดยใช้วิธีการแปลงเวฟเล็ต ตามผังงานที่ได้กล่าวไว้ในบทที่ 3 ซึ่งจะได้โปรแกรมหลัก สำหรับการแปลงข้อมูลภาพเป็นสัมประสิทธิ์ของเวฟเล็ตที่อยู่ในภาคผนวก



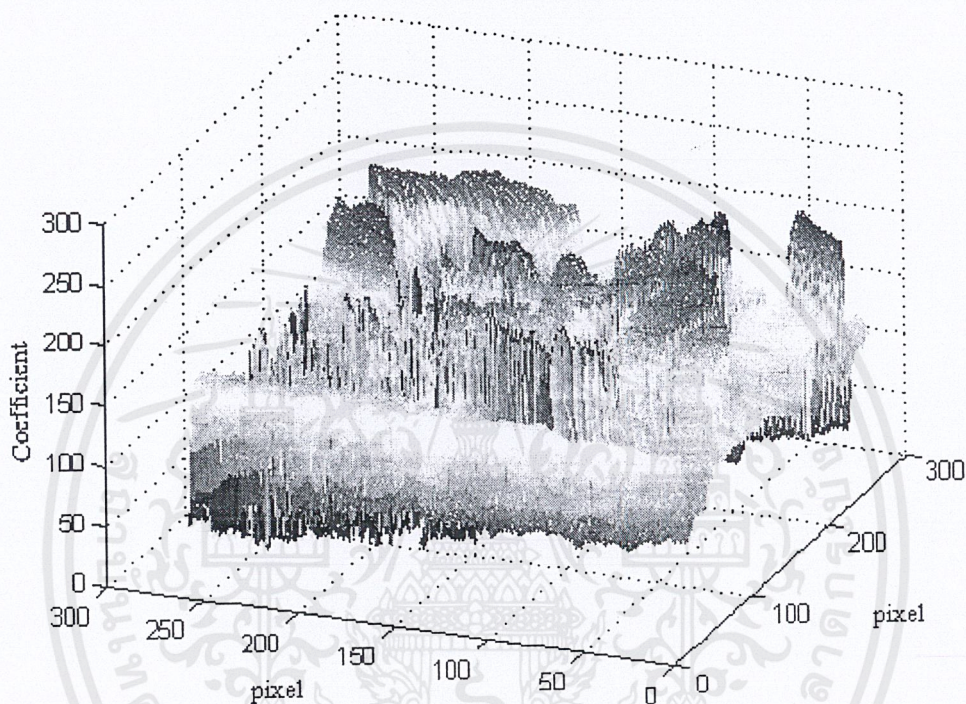
การทดลองที่ 4.1 ทดลองโปรแกรมสำหรับการแปลงเวฟเล็ต กับข้อมูลภาพ โดยนำเอาภาพต้นฉบับมาทำการแปลงเวฟเล็ต ที่ใช้สัมประสิทธิ์ของฮาร์เป็นเวฟเล็ตฟังก์ชัน แล้วทำการสร้างภาพกลับขึ้นมาจากสัมประสิทธิ์เวฟเล็ตของภาพนั้น นำผลที่ได้จากการแปลงมาพิจารณาหาความผิดพลาดจากการแปลงเวฟเล็ต

ในการทดลองจะใช้ภาพ Lena ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบ ดังแสดงในรูปที่ 4.1



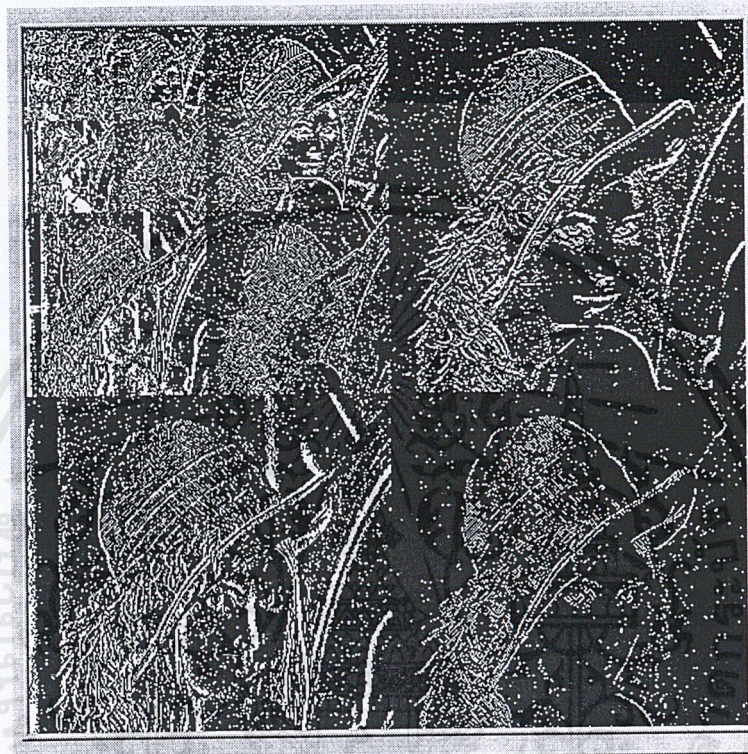
รูปที่ 4.1 แสดงภาพต้นแบบที่ใช้ในการทดลอง (Lena ขนาด 512x512 จุดภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



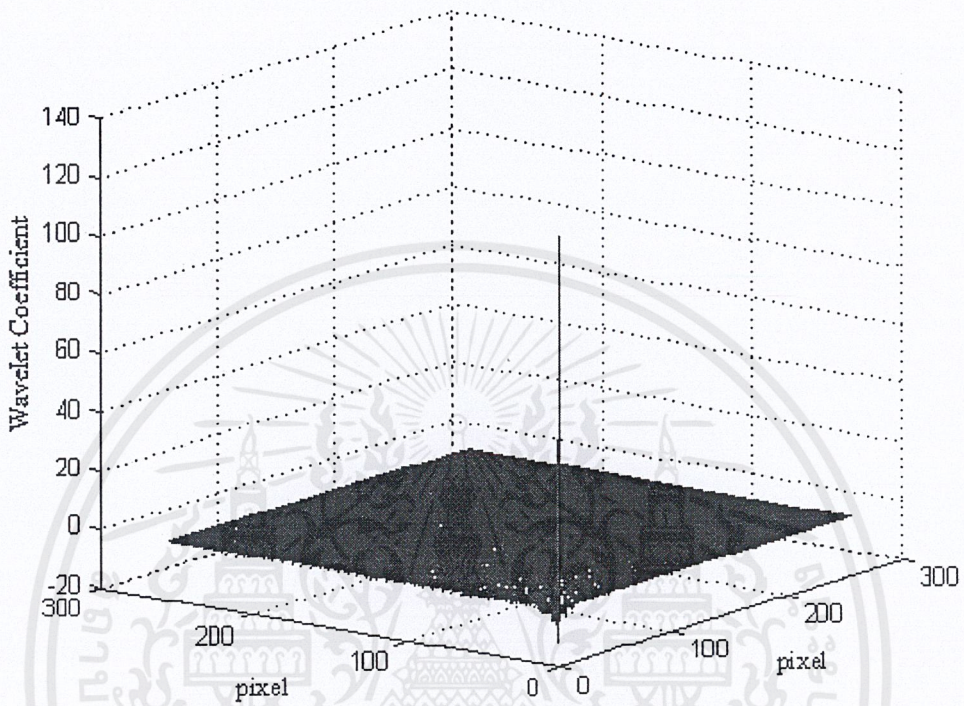
รูปที่ 4.2 แสดงค่าสัมประสิทธิ์ของภาพLena ขนาด256x256จุดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงภาพภายหลังจากการแปลงเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงค่าสัมประสิทธิ์ของภาพLena ขนาด 256x256จุดภาพหลังจากการแปลงเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงภาพที่สร้างกลับขึ้นมาหลังการแปลงเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4.2 ในการทดลองนี้จะแสดงวิธีการหาค่าระดับอ้างอิงเริ่มต้น ให้กับเอ็มเบดซีโรทรี ซึ่งจำลองโดยเปิดภาพ Iena 8x8 พิกเซลโดยมีค่าแต่ละจุดภาพดังนี้

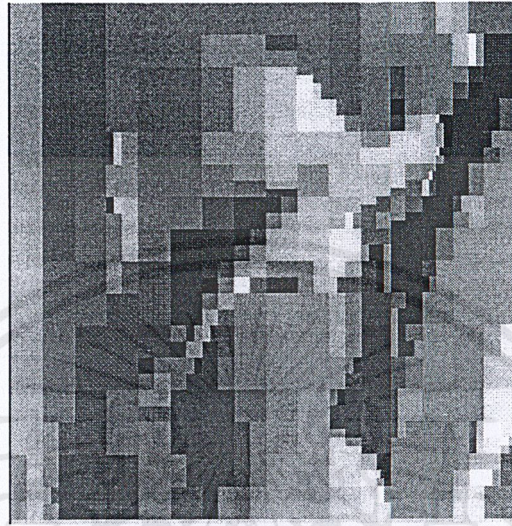
95.0000 129.0000 132.0000 179.0000 123.0000 148.0000 142.0000 53.0000
 85.0000 115.0000 141.0000 189.0000 207.0000 155.0000 179.0000 54.0000
 86.0000 176.0000 146.0000 127.0000 153.0000 199.0000 43.0000 155.0000
 82.0000 107.0000 87.0000 47.0000 169.0000 54.0000 56.0000 162.0000
 91.0000 63.0000 51.0000 147.0000 134.0000 48.0000 158.0000 143.0000
 80.0000 91.0000 145.0000 132.0000 166.0000 87.0000 149.0000 207.0000
 82.0000 85.0000 52.0000 100.0000 157.0000 79.0000 107.0000 210.0000
 62.0000 43.0000 67.0000 124.0000 147.0000 213.0000 134.0000 99.0000

และค่าสัมประสิทธิ์หลังจากการแปลงเวฟเล็ด โดยใช้ฮาร์ฟังก์ชันดังนี้

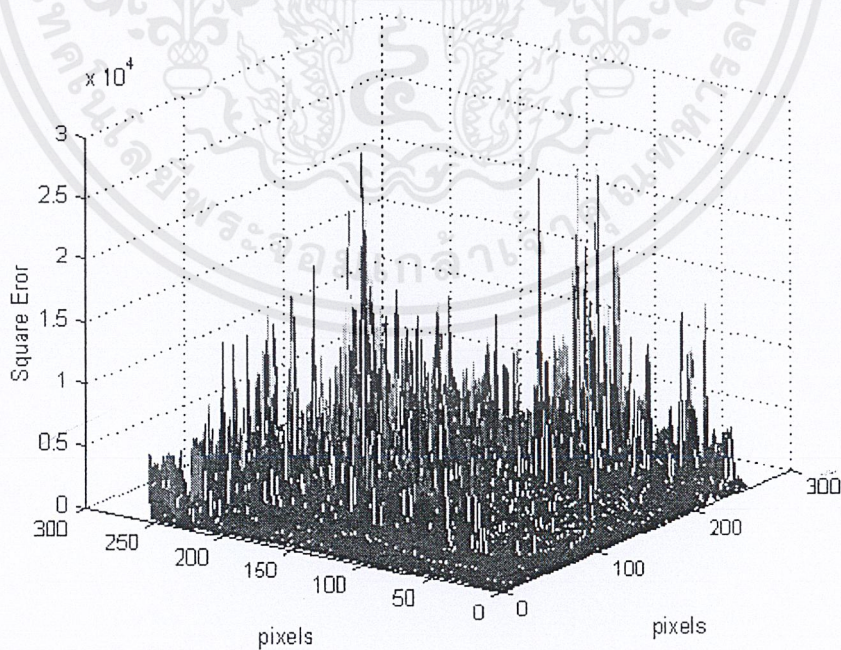
119.1875 5.0313 6.4688 2.1875 1.5000 -1.1875 -5.6875 -2.3750
 -14.8750 10.8438 5.7813 -1.6875 4.5625 8.6875 8.0625 -1.2500
 -5.4063 11.3750 -8.1563 1.4375 -1.0625 -4.9375 -4.4375 -3.4375
 -6.9063 -5.5000 -2.4688 -8.3750 3.8750 -2.4375 -7.7500 5.2500
 -4.0000 -5.9375 1.6875 13.3750 -0.2500 0.0625 -4.8125 -2.2500
 -7.1875 3.6875 4.3125 -13.6250 -4.0625 -1.3125 -10.0625 -0.3750
 1.0625 -5.1875 10.3125 -2.6875 2.4375 -6.8125 0.4375 4.5625
 1.0000 -6.5625 0.7500 -4.2500 -1.3750 0.5625 9.0000 -8.6250

ตัวโปรแกรมจะทำการคำนวณค่าระดับอ้างอิงเริ่มต้นจากการหาค่าสัมประสิทธิ์ที่มีค่ามากที่สุด จากตัวอย่างนี้ค่าสัมประสิทธิ์ที่มีค่ามากที่สุดคือ 119.1875 จะทำการหารด้วย 2 ซึ่งจะเหลือ 59.59 และทำการเลือกค่าระดับอ้างอิงในรูปเลขสองยกกำลัง จะได้เป็นค่า 64

การทดลองที่ 4.3 ทดลองการปิดค่าระดับเทรซโฮลของสัมประสิทธิ์เวฟเล็ตของภาพ ที่ได้จากการแปลงเวฟเล็ตแบบไม่มาตรฐาน โดยใช้สัมประสิทธิ์ของฮาร์เป็นเวฟเล็ตฟังก์ชัน เมื่อทำการตั้งค่าระดับอ้างอิงที่ต่างกัน นำผลที่ได้จากการแปลงมาพิจารณาหาความผิดพลาดจากการแปลงเวฟเล็ต เมื่อทำการใช้ค่าระดับอ้างอิงที่ 1 ภาพที่ได้จะเป็นดังรูปที่ 4.6



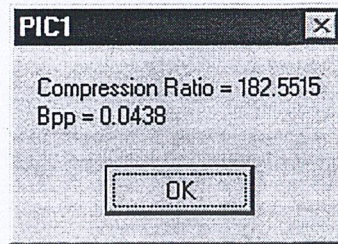
รูปที่ 4.6 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 1



รูปที่ 4.7 กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 1

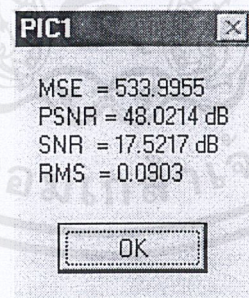
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสเป็น 182.5515 และค่าเฉลี่ยจำนวนบิตต่อจุดภาพเป็น 0.0438



รูปที่ 4.8 แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกปิดค่าระดับอ้างอิงที่ 1

จะได้ ค่าMean Square Error(MSE) ของภาพเป็น 533.9955
 ค่าPeak Signal to Noise Ratio(PSNR) ของภาพเป็น 48.0214 dB
 ค่าSignal to Noise Ratio (SNR) ของภาพเป็น 17.5217 dB
 ค่าRoot Mean Square (RMS) ของภาพเป็น 0.0903
 ดังรูปที่ 4.9



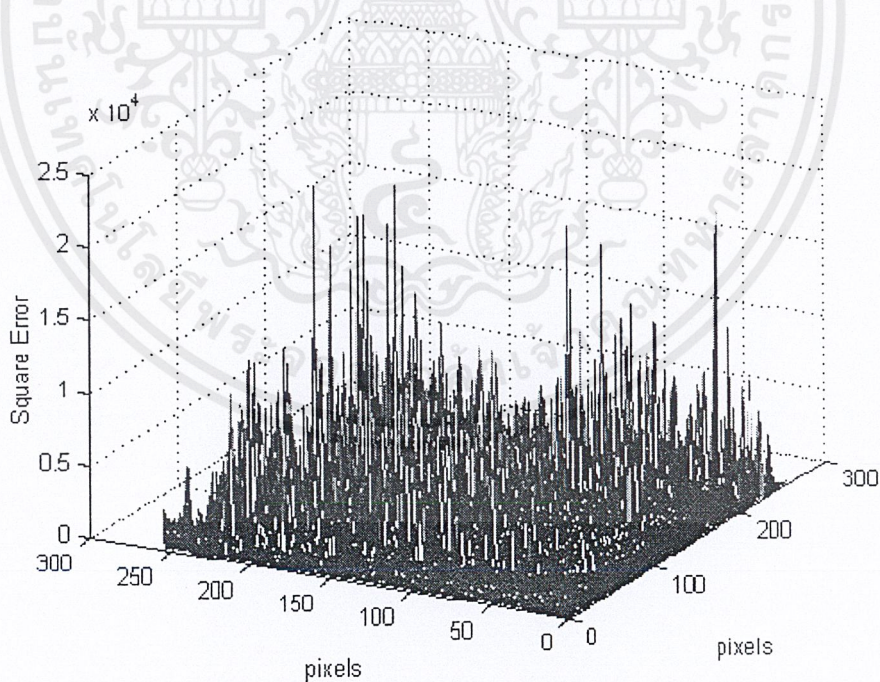
รูปที่ 4.9 แสดงค่า MSE , PSNR , SNR และ RMSที่ถูกปิดค่าระดับอ้างอิงที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการใช้ค่าระดับอ้างอิงที่ 0.5 ภาพที่ได้จะเป็นดังรูปที่ 4.10



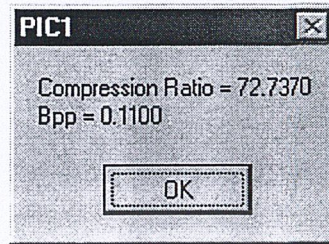
รูปที่ 4.10 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ดของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.5



รูปที่ 4.11 กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ดของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

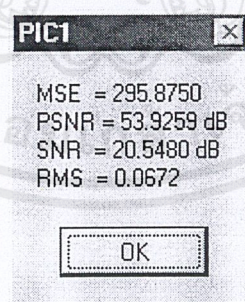
จะได้ค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสเป็น 72.7370 และค่าเฉลี่ยจำนวนบิตต่อจุดภาพเป็น 0.1100



รูปที่ 4.12 แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกปิดค่าระดับอ้างอิงที่ 0.5

จะได้ ค่าMean Square Error(MSE) ของภาพเป็น 295.8750
 ค่าPeak Signal to Noise Ratio(PSNR) ของภาพเป็น 53.9259 dB
 ค่าSignal to Noise Ratio (SNR) ของภาพเป็น 20.5480 dB
 ค่าRoot Mean Square (RMS) ของภาพเป็น 0.0672

ดังรูปที่ 4.13



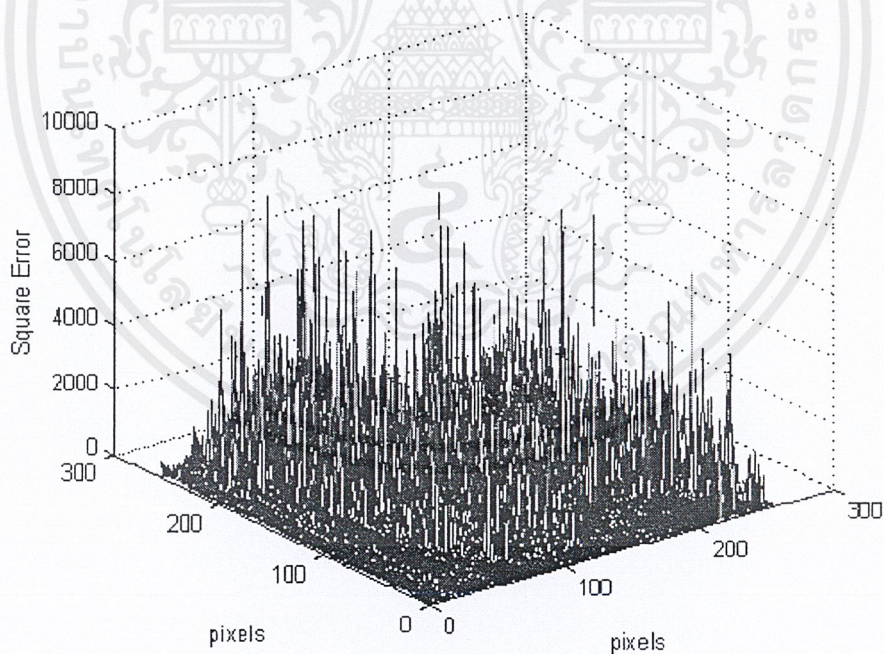
รูปที่ 4.13 แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการใช้ค่าระดับอ้างอิงที่ 0.25 ภาพที่ได้จะเป็นดังรูปที่ 4.14



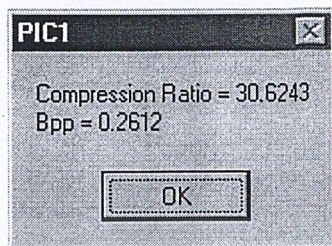
รูปที่ 4.14 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.25



รูปที่ 4.15 กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

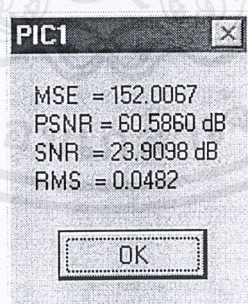
จะได้ค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสเป็น 30.6243 และค่าเฉลี่ยจำนวนบิตต่อจุดภาพเป็น 0.2612



รูปที่ 4.16 แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกบีบอัดค่าระดับอ้างอิงที่ 0.25

จะได้ ค่าMean Square Error(MSE) ของภาพเป็น 152.0067
 ค่าPeak Signal to Noise Ratio(PSNR) ของภาพเป็น 60.5860 dB
 ค่าSignal to Noise Ratio (SNR) ของภาพเป็น 23.9098 dB
 ค่าRoot Mean Square (RMS) ของภาพเป็น 0.0482

ดังรูปที่ 4.17



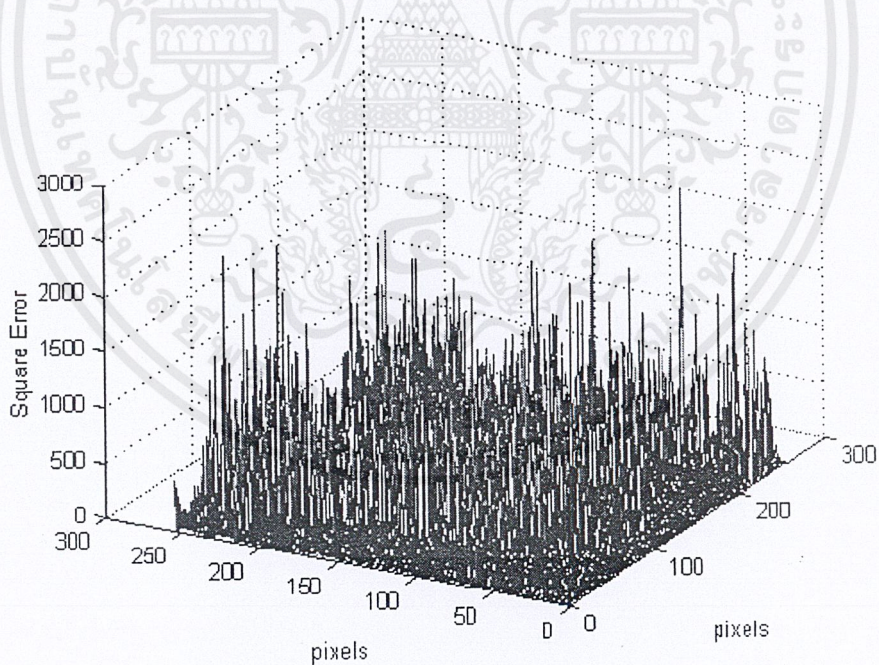
รูปที่ 4.17 แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกบีบอัดค่าระดับอ้างอิงที่ 0.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการใช้ค่าระดับอ้างอิงที่ 0.125 ภาพที่ได้จะเป็นดังรูปที่ 4.18



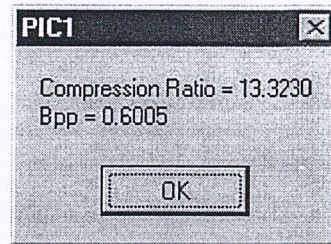
รูปที่ 4.18 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.125



รูปที่ 4.19 กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.125

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสเป็น 13.3230 และค่าเฉลี่ยจำนวนบิตต่อจุดภาพเป็น 0.6005



รูปที่ 4.20 แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพ ที่ถูกปิดค่าระดับอ้างอิงที่ 0.125

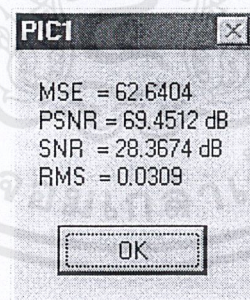
จะได้ ค่าMean Square Error(MSE) ของภาพเป็น 62.6404

ค่าPeak Signal to Noise Ratio(PSNR) ของภาพเป็น 69.4512 dB

ค่าSignal to Noise Ratio (SNR) ของภาพเป็น 28.3674 dB

ค่าRoot Mean Square (RMS) ของภาพเป็น 0.0309

ดังรูปที่ 4.21



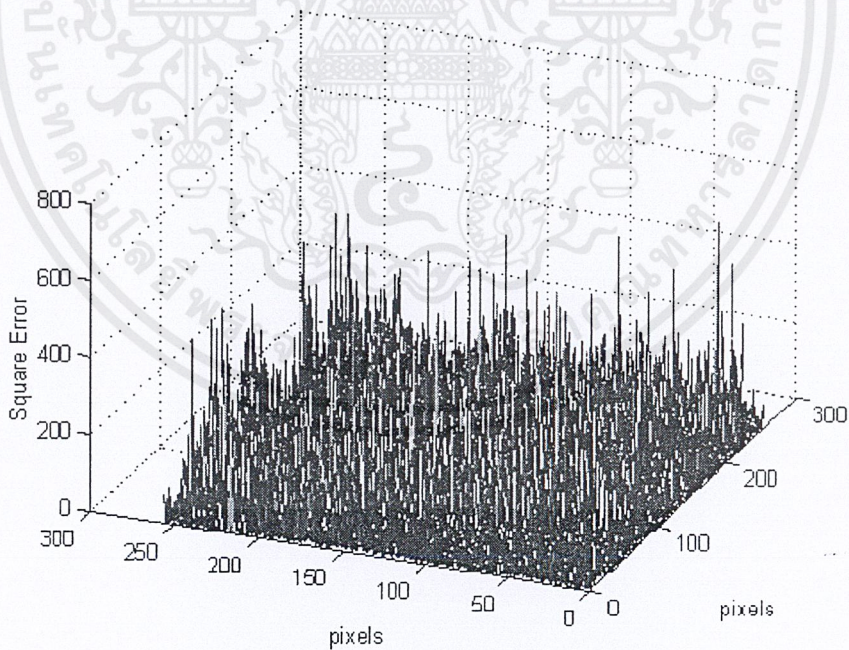
รูปที่ 4.21 แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.125

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการใช้ค่าระดับอ้างอิงที่ 0.0625 ภาพที่ได้จะเป็นดังรูปที่ 4.22



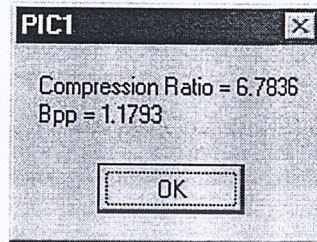
รูปที่ 4.22 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.0625



รูปที่ 4.23 กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.0625

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสเป็น 6.7836 และค่าเฉลี่ยจำนวนบิตต่อจุดภาพเป็น 1.1793



รูปที่ 4.24 แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพที่ถูกปิดค่าระดับอ้างอิงที่ 0.0625

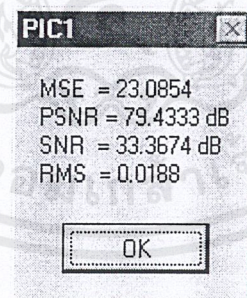
จะได้ ค่าMean Square Error(MSE) ของภาพเป็น 23.0854

ค่าPeak Signal to Noise Ratio(PSNR) ของภาพเป็น 79.4333 dB

ค่าSignal to Noise Ratio (SNR) ของภาพเป็น 33.3674 dB

ค่าRoot Mean Square (RMS) ของภาพเป็น 0.0188

ดังรูปที่ 4.25



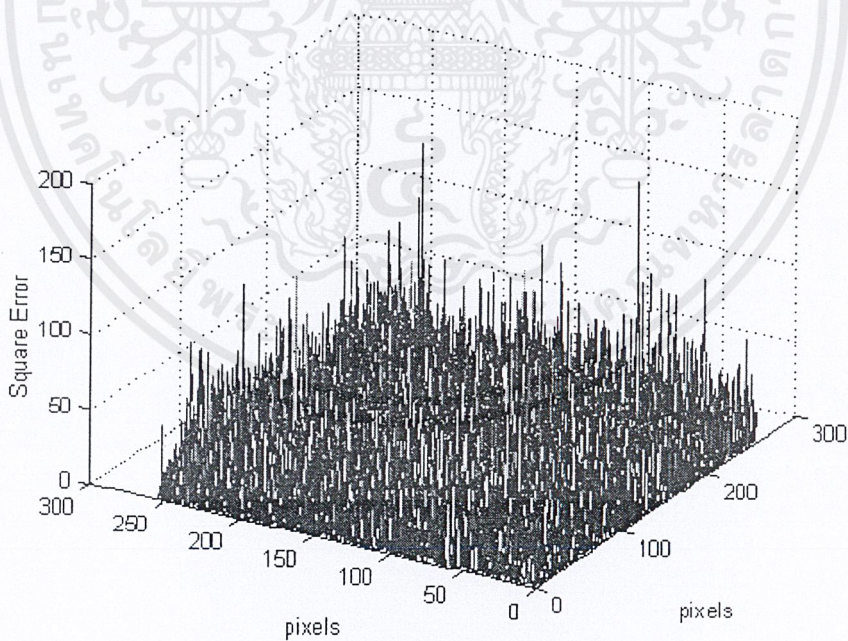
รูปที่ 4.25 แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.0625

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการใช้ค่าระดับอ้างอิงที่ 0.03125 ภาพที่ได้จะเป็นดังรูปที่ 4.26



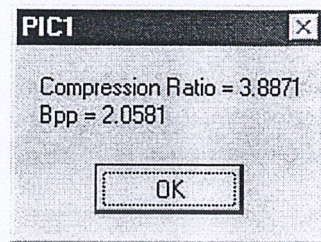
รูปที่ 4.26 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.03125



รูปที่ 4.27 กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.03125

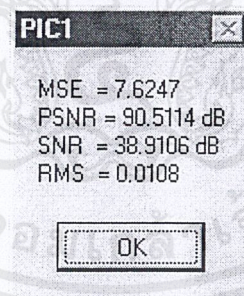
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสเป็น 3.8871 และค่าเฉลี่ยจำนวนบิตต่อจุดภาพเป็น 2.0581



รูปที่ 4.28 แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพ ที่ถูกปิดค่าระดับอ้างอิงที่ 0.03125

จะได้ ค่าMean Square Error(MSE) ของภาพเป็น 7.6247
 ค่าPeak Signal to Noise Ratio(PSNR) ของภาพเป็น 90.5114 dB
 ค่าSignal to Noise Ratio (SNR) ของภาพเป็น 38.9106 dB
 ค่าRoot Mean Square (RMS) ของภาพเป็น 0.0108
 ดังรูปที่ 4.29



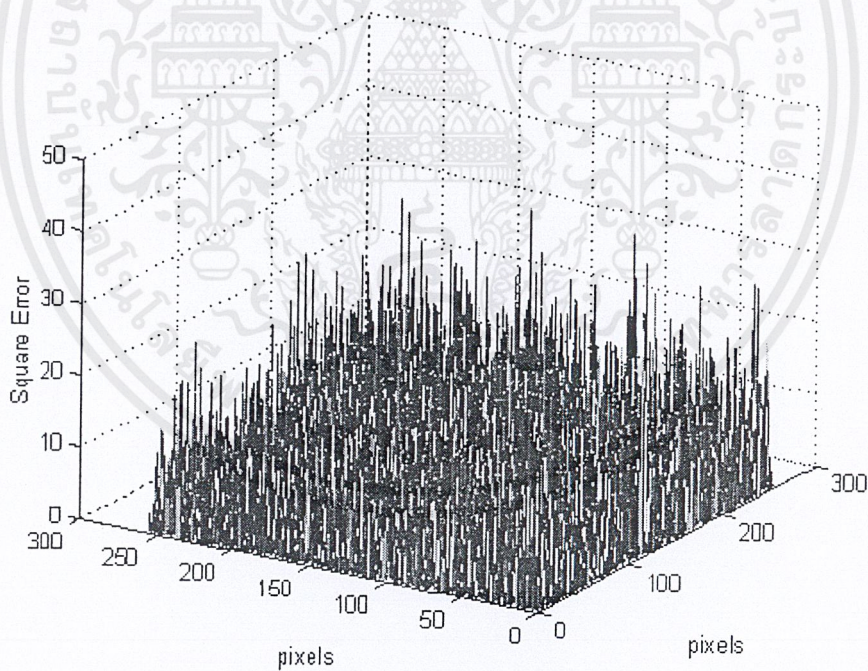
รูปที่ 4.29 แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.03125

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการใช้ค่าระดับอ้างอิงที่ 0.015625 ภาพที่ได้จะเป็นดังรูปที่ 4.30



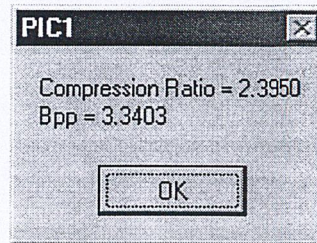
รูปที่ 4.30 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.015625



รูปที่ 4.31 กราฟแสดงค่าความผิดพลาดของภาพต้นแบบที่เทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปิดค่าระดับอ้างอิงที่ 0.015625

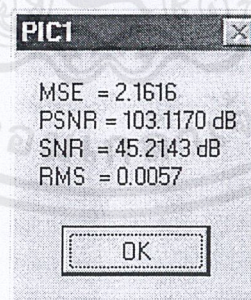
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสเป็น 2.3950 และค่าเฉลี่ยจำนวนบิตต่อจุดภาพเป็น 3.3403



รูปที่ 4.32 แสดงค่าอัตราการบีบอัดข้อมูลระหว่างข้อมูลภาพจริงกับข้อมูลที่ได้หลังจากที่ถูกเข้ารหัสและค่าเฉลี่ยจำนวนบิตต่อจุดภาพ ที่ถูกปิดค่าระดับอ้างอิงที่ 0.015625

จะได้ ค่าMean Square Error(MSE) ของภาพเป็น 2.1616
 ค่าPeak Signal to Noise Ratio(PSNR) ของภาพเป็น 103.1170 dB
 ค่าSignal to Noise Ratio (SNR) ของภาพเป็น 45.2143 dB
 ค่าRoot Mean Square (RMS) ของภาพเป็น 0.0057
 ดังรูปที่ 4.33



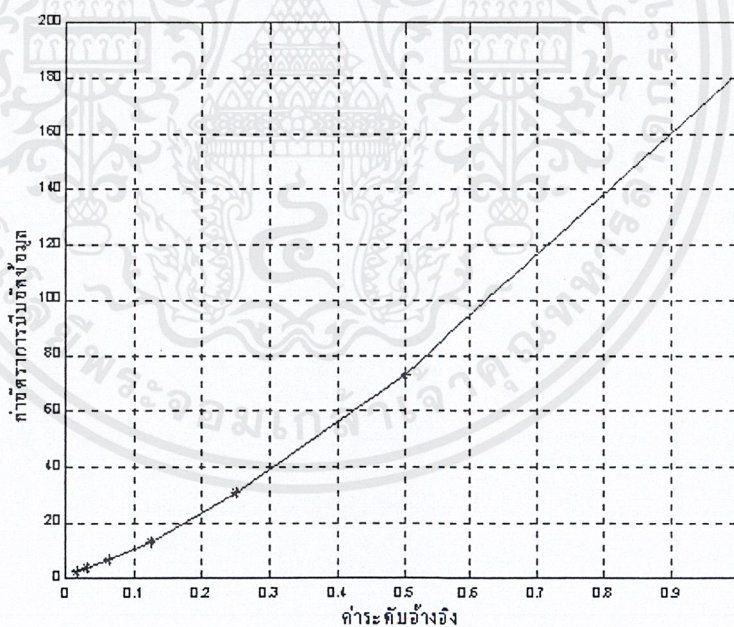
รูปที่ 4.33 แสดงค่า MSE , PSNR , SNR และ RMS ที่ถูกปิดค่าระดับอ้างอิงที่ 0.015625

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแสดงผลการเปรียบเทียบข้อมูลที่ได้จากภาพ lena256.bmp กับค่าอัตราการบีบอัดข้อมูลและ ค่าความผิดพลาดต่างๆ ที่ค่าระดับอ้างอิงต่ำที่สุด ได้ดังนี้

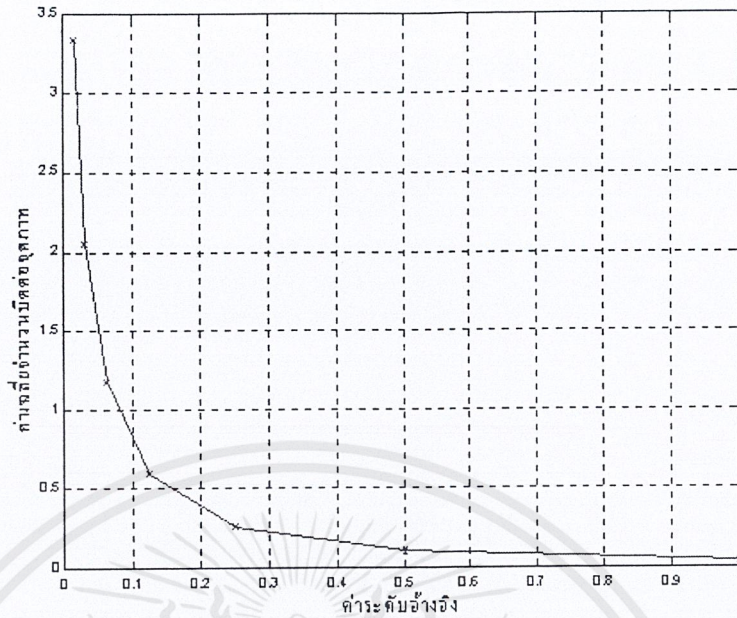
THRESHOLD	Cr	Bpp	MSE	RMS	PSWR	SNR
1	182.5515	0.0438	533.995	0.0903	48.0214	17.5217
0.5	72.737	0.11	295.875	0.0672	53.9259	20.548
0.25	30.6243	0.2612	152.0067	0.0482	60.586	23.9098
0.125	13.323	0.6005	62.6404	0.0309	69.4512	28.3674
0.0625	6.7836	1.1793	23.0854	0.0188	79.4333	33.3674
0.03125	3.8871	2.0581	7.6247	0.0108	90.5114	38.9106
0.015625	2.395	3.3403	2.1616	0.0057	103.117	45.2143

ตารางที่ 4.1 แสดงค่า ค่าอัตราการบีบอัดข้อมูลและ ค่าความผิดพลาดต่างๆ ที่ค่าระดับอ้างอิงต่ำที่สุด ของภาพ lena256.bmp

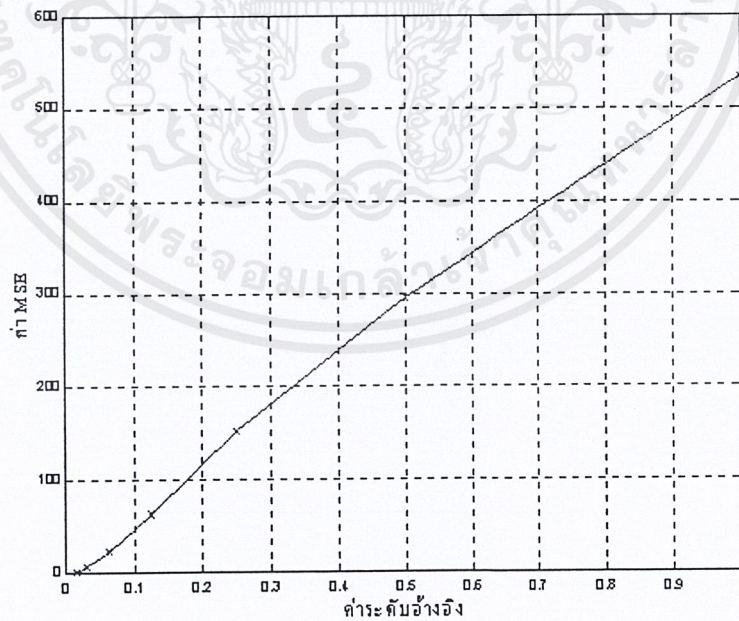


รูปที่ 4.34 แสดงค่าระดับอ้างอิง(Threshold)และค่าอัตราการบีบอัดข้อมูล(Cr) ของภาพ lena256.bmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

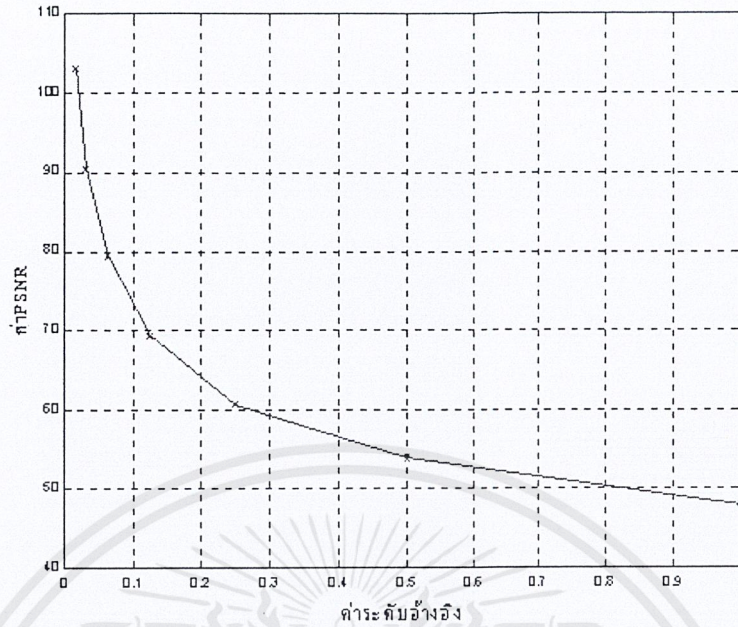


รูปที่ 4.35 แสดงค่าระดับอ้างอิง(Threshold)และค่าเฉลี่ยจำนวนบิตต่อจุดภาพ (Cr) ของภาพ lena256.bmp



รูปที่ 4.36 แสดงค่าระดับอ้างอิง (Threshold)และค่า MSE ของภาพ lena256.bmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



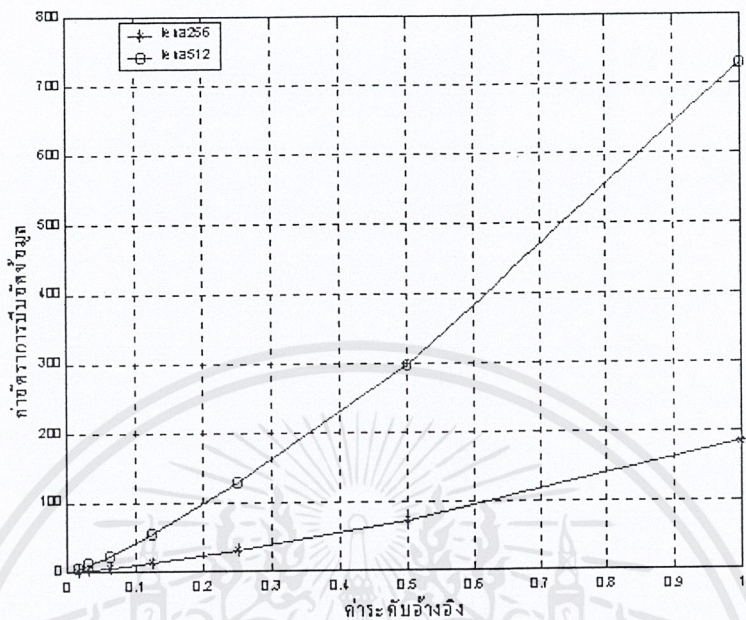
รูปที่ 4.37 แสดงค่าระดับอ้างอิง (Threshold)และค่าPSNR ของภาพ lena256.bmp

สามารถแสดงผลการเปรียบเทียบข้อมูลที่ได้จากภาพ lena512.bmpกับค่าอัตราการบีบอัดข้อมูล และ ค่าความผิดพลาดต่างๆ ที่ค่าระดับอ้างอิงต่ำที่สุด ได้ดังนี้

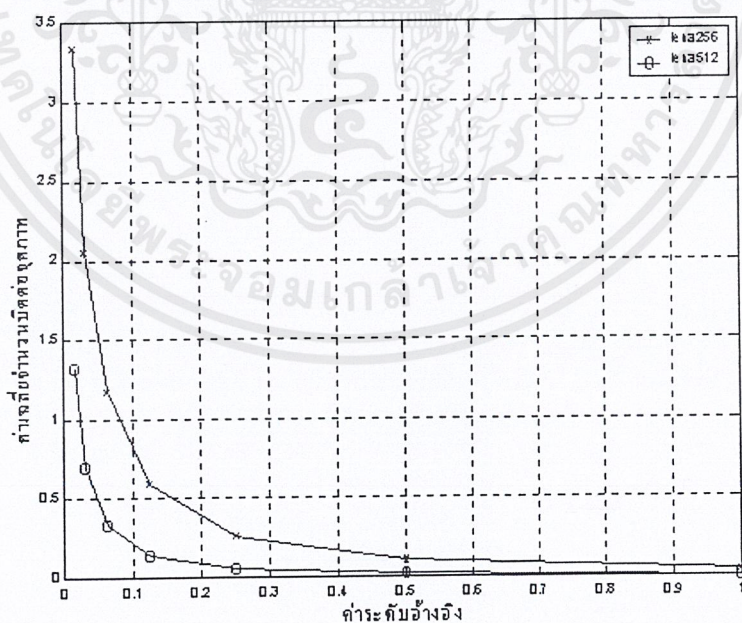
THRESHOLD	Cr	Bpp	MSE	RMS	PSNR	SNR
1	730.2061	0.011	543.0269	0.0455	47.8537	17.4342
0.5	296.543	0.027	308.3282	0.0343	53.5136	20.3352
0.25	127.378	0.0628	175.1926	0.0259	59.1664	23.1914
0.125	53.7401	0.1489	91.7142	0.0187	65.6385	26.4518
0.0625	23.7859	0.3363	42.8716	0.0128	73.2432	30.2657
0.03125	11.4679	0.6976	17.9562	0.0083	81.9459	34.6238
0.015625	6.0405	1.3244	6.7584	0.0051	91.7174	39.5121

ตารางที่ 4.2 แสดงค่า ค่าอัตราการบีบอัดข้อมูล(Cr)และ ค่าความผิดพลาดต่างๆ ที่ค่าระดับอ้างอิงต่ำที่สุด ของภาพ lena512.bmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

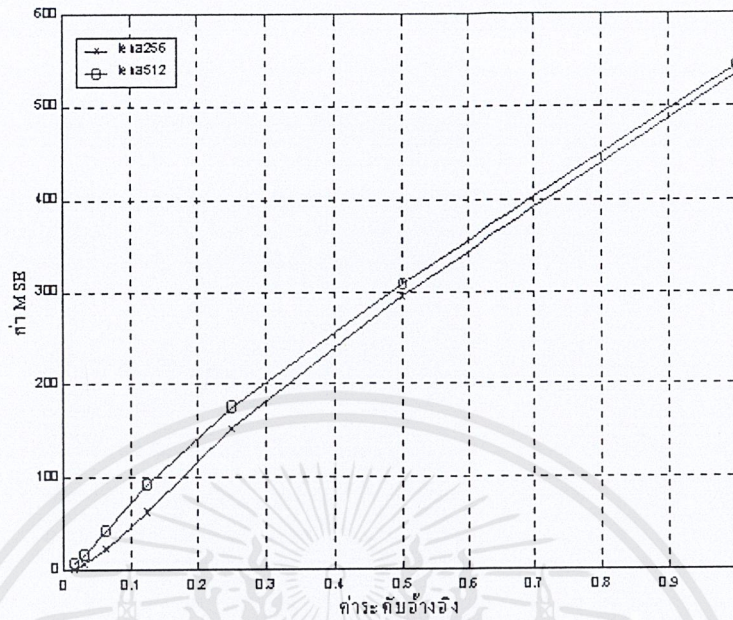


รูปที่ 4.38 แสดงค่าระดับอ้างอิง(Threshold) และค่าอัตราการบีบอัดข้อมูล(Cr) ของภาพ lena256.bmp เทียบกับภาพ lena512.bmp



รูปที่ 4.39 แสดงค่าระดับอ้างอิง(Threshold)และค่าเฉลี่ยจำนวนบิตต่อจุดภาพ(Cr) ของภาพ lena256.bmp เทียบกับภาพ lena512.bmp

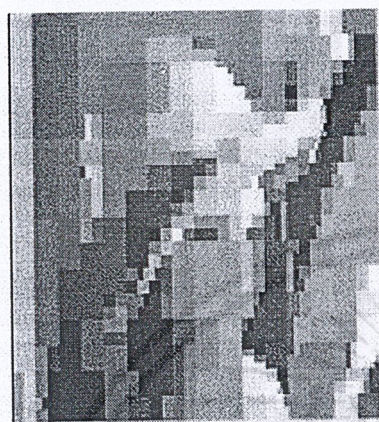
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.40 แสดงค่าระดับอ้างอิง (Threshold) และค่า MSE ของภาพ lena256.bmp เทียบกับภาพ lena512.bmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ใช้ค่าเฉลี่ยจำนวนบิตต่อจุดภาพ เป็นตัวอ้างอิงจะได้ความละเอียดของภาพดังรูป 4.41



bpp=0.0438



bpp=0.1100



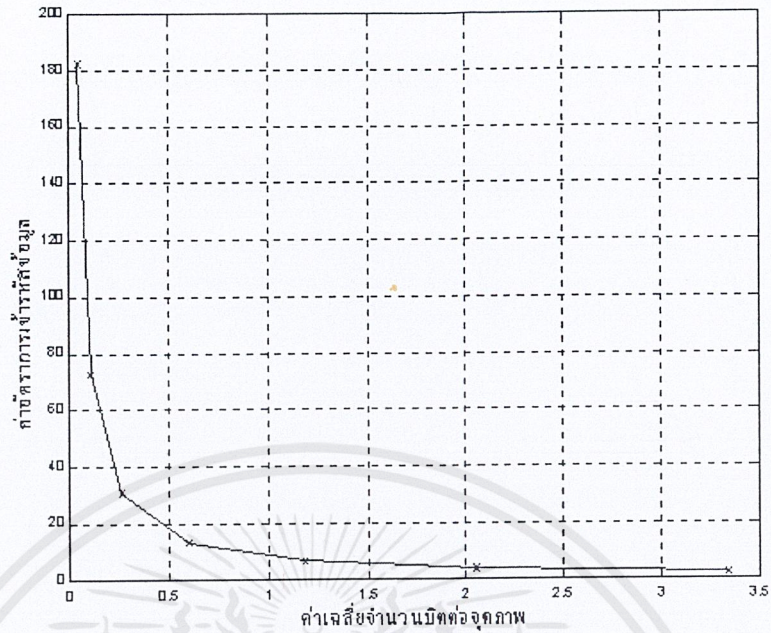
bpp=0.2612



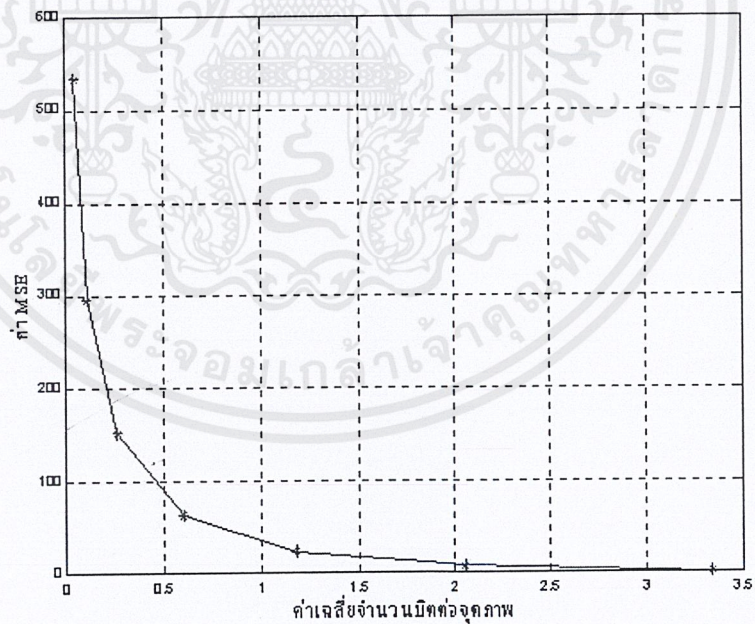
bpp=0.6005

รูปที่ 4.41 แสดงความละเอียดของภาพ ที่ bpp ค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

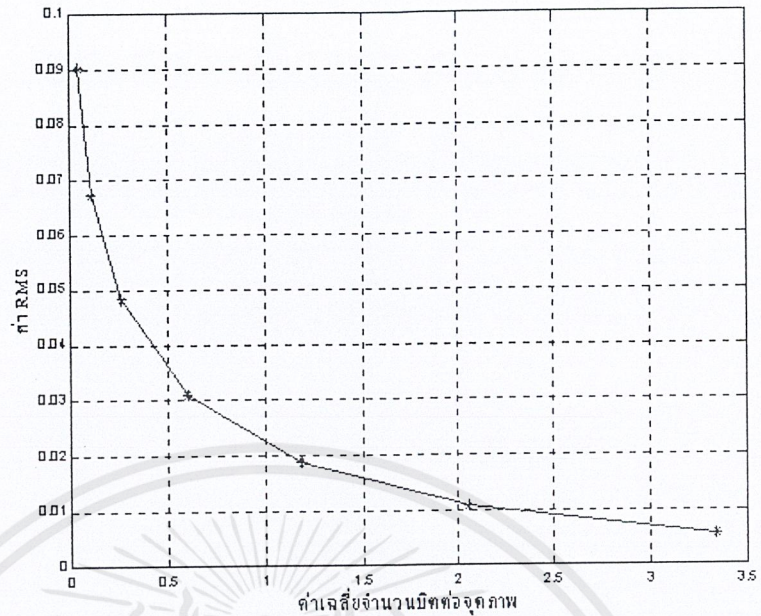


รูปที่ 4.42 แสดงค่าเฉลี่ยจำนวนบิตต่อจุดภาพ (bpp) และค่าอัตราการบีบอัดข้อมูล (Cr) ของภาพ lena256.bmp

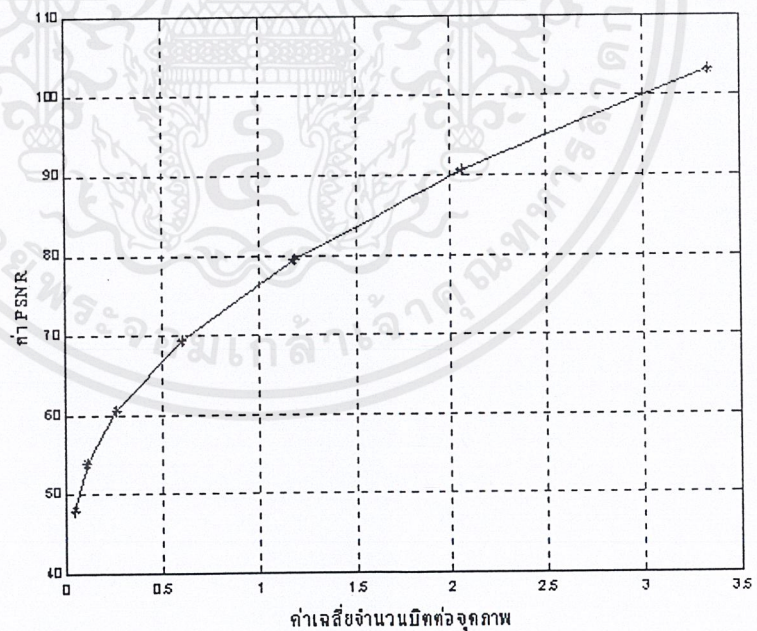


รูปที่ 4.43 แสดงค่าเฉลี่ยจำนวนบิตต่อจุดภาพ (bpp) และค่า MSE ของภาพ lena256.bmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.44 แสดงค่าเฉลี่ยจำนวนบิตต่อจุดภาพ(bpp) และค่า RMS ของภาพ lena256.bmp

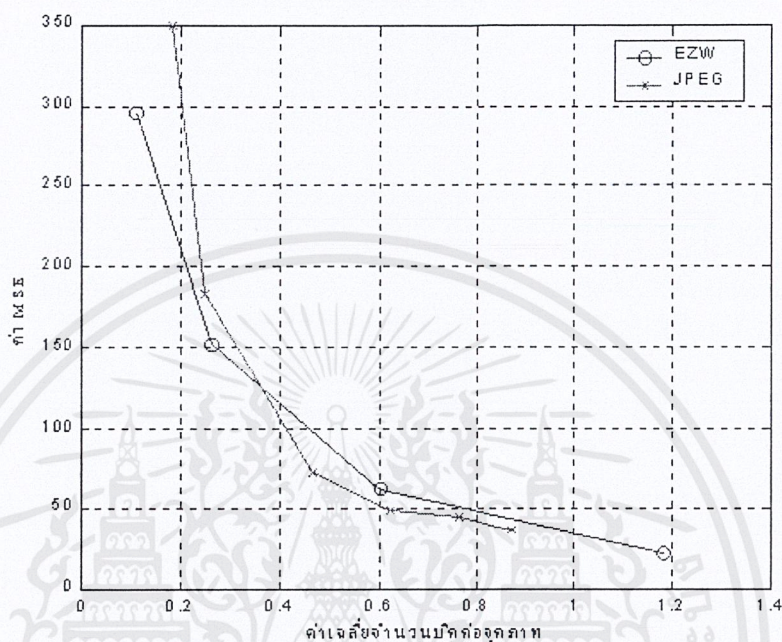


รูปที่ 4.45 แสดงค่าเฉลี่ยจำนวนบิตต่อจุดภาพ(bpp) และค่า PSNR ของภาพ lena256.bmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4.4 ทดลองเปรียบเทียบผลที่ได้จากการลดขนาดข้อมูลรูปภาพ ระหว่างวิธีการของวิธีการของเอ็มเบดซีโรทีรี และวิธีการของเจเพก(JPEG) เพื่อดูค่าความผิดพลาดที่เกิดขึ้น

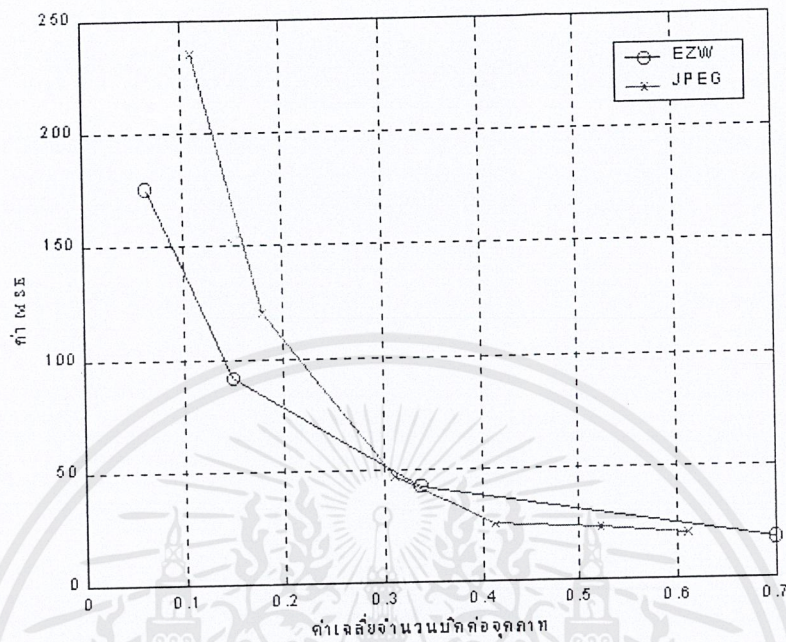
1. รูปภาพ Lena ขนาด 256x256 จุดภาพ เป็นภาพต้นแบบ



รูปที่ 4.46 กราฟแสดงผลของการลดขนาดข้อมูลรูปภาพด้วยวิธีการของ EZW และ JPEG (ใช้ภาพ lena ขนาด 256x256 จุดภาพ)

จากกราฟในรูปที่ 4.46 จะเห็นได้ว่าวิธีการของ EZW ที่ระดับขนาด bpp ต่ำกว่าช่วง 0.35 bpp จะมีระดับของค่าความผิดพลาดที่ต่ำกว่า JPEG แต่ในช่วงที่มีค่า bpp ที่สูงกว่านั้น JPEG จะมีค่าความผิดพลาดที่ต่ำกว่า

2. รูปภาพ Lena ขนาด 512x512 จุดภาพ เป็นภาพต้นแบบ



รูปที่ 4.47 กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการของ EZW และ JPEG (ใช้ภาพ Lena ขนาด 512x512 จุดภาพ)

จากกราฟในรูปที่ 4.47 จะเห็นได้ว่าวิธีการของ EZW ที่ระดับขนาด bpp ต่ำกว่าช่วง 0.30 bpp จะมีระดับของค่าความผิดพลาดที่ต่ำกว่า JPEG แต่ในช่วงที่มีค่า bpp ที่สูงกว่านั้น JPEG จะมีค่าความผิดพลาดที่ต่ำกว่าจากรูปที่ 4.46 ซึ่งได้ทดสอบรูปภาพชนิดเดียวกันแต่ขนาดต่างกันพบว่า ผลของการลดขนาดข้อมูลนั้นแตกต่างกันอย่างเห็นได้ชัดเจน เนื่องจากในระดับภาพที่มีรายละเอียดที่สูงขึ้น จะมีองค์ประกอบของสัญญาณในภาพที่ละเอียดขึ้นตามไปด้วย ดังนั้นการวิเคราะห์ทางด้านความถี่ที่อาศัยการแยกแบนด์ของความถี่เป็นความถี่ย่อยๆ โดยใช้การแปลงเวฟเล็ต ก็จะสามารถแบ่งแบนด์ของความถี่ได้ละเอียดมากขึ้นตามไปด้วย ยกตัวอย่างเช่น ภาพขนาด 256x256 จุดภาพ เมื่อทำการแปลงเวฟเล็ต จะถูกแบ่งแบนด์ของความถี่ได้เป็น 8 แบนด์ และในกรณี ภาพขนาด 512x512 จุดภาพ เมื่อทำการแปลงเวฟเล็ต จะถูกแบ่งแบนด์ของความถี่ได้เป็น 9 แบนด์ ซึ่งจะมีความละเอียดในการแบ่งความถี่เพิ่มขึ้น

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทวิจารณ์

การแปลงเวฟเล็ตเป็นการแปลงข้อมูลจากโดเมนเวลา ให้ไปอยู่ในระนาบของเวลาและความถี่ โดยเหมือนกับการแยกสัญญาณที่ต้องการพิจารณาเป็นองค์ประกอบของสัญญาณย่อยๆ ที่เกิดขึ้นในช่วงเวลาต่างๆ การนำข้อมูลระดับเทาของภาพมาผ่านการแปลงเวฟเล็ต ให้เป็นองค์ประกอบของภาพในระนาบเวลาและความถี่ จะเปรียบเสมือนการนำเอาภาพต้นแบบมาผ่านการกรองตามช่วงความถี่เพื่อสร้างเป็นภาพแบนด์ย่อย ซึ่งสัมพันธ์ที่ใช้สำหรับการกรองความถี่นี้ จะเป็นสัมพันธ์ของเวฟเล็ตฟังก์ชัน โดยขึ้นอยู่กับเวฟเล็ตฟังก์ชันที่นำมาใช้ โดยสัมพันธ์เหล่านี้จะมีคุณสมบัติเป็นตัวกรองความถี่ตอบสนองอิมพัลส์ที่จำกัด โดยจะให้เฟสของสัญญาณที่เป็นเชิงเส้น ทำให้การแยกองค์ประกอบของความถี่ แล้วนำความถี่เหล่านั้นมารวมกัน ก็จะได้สัญญาณตัวเดิมกลับมา การเลือกฟังก์ชันเวฟเล็ตที่มีลักษณะคล้ายกับส่วนประกอบย่อยๆของสัญญาณอินพุต จะส่งผลให้การแยกองค์ประกอบของสัญญาณกระทำได้ดีขึ้น จากการทดลองพบว่าฟังก์ชันที่มีความต่อเนื่องจะทำให้ได้คุณภาพเนื้อภาพที่ดี ถ้าใช้เวฟเล็ตฟังก์ชันที่ไม่ให้ความต่อเนื่องของข้อมูล เช่น ฮาร์ฟังก์ชัน ก็จะทำให้ได้ขอบภาพที่คมมากขึ้น แต่เนื้อภาพจะแย่ง ดังนั้น การเลือกฟังก์ชันชนิดใดก็ขึ้นอยู่กับการประยุกต์ใช้งาน

การนำสัมพันธ์เวฟเล็ตมาเข้ารหัสข้อมูล เป็นขั้นตอนสำคัญในการลดขนาดข้อมูล โดยการลดขนาดข้อมูลภาพจะเกิดขึ้นในกระบวนการนี้ วิธีการนำเสนอในปริภูมิพิกเซลคือการเข้ารหัสข้อมูลแบบเอ็มเบดซีโรที โดยวิธีการนี้เป็นขั้นตอนที่ไม่สามารถย้อนกลับได้ เพราะขั้นตอนในการถอดรหัสไม่สามารถทำการระบุค่าของสัมพันธ์ที่ตำแหน่งแน่นอน เป็นการประมาณค่าของข้อมูลต้นแบบ

การเข้ารหัสเอ็มเบดซีโรทีจะทำการจัดสัมพันธ์ที่ได้จากการแปลงเวฟเล็ตเป็นชุดรหัสของข้อมูล โดยจำนวนของชุดรหัสนั้นขึ้นอยู่กับคุณภาพและอัตราการลดขนาดข้อมูลที่ต้องการนำชุดของรหัสดังกล่าวมาผ่านการเข้ารหัสเลขคณิตเพื่อทำการลดรหัสข้อมูลในขั้นตอนสุดท้าย

การนำเอาการแปลงเวฟเล็ตมาใช้สำหรับการส่งข้อมูลภาพแบบโปรเกรสซีฟสามารถนำเอาการเข้ารหัสแบบเอ็มเบดซีโรทีมาใช้ได้ เพราะวิธีการเข้ารหัสแบบเอ็มเบดซีโรทีจะทำการเข้ารหัสในแบบด์ที่เป็นองค์ประกอบสำคัญของภาพก่อน นั่นคือองค์ประกอบทางความถี่ต่ำ และตามมาด้วยรายละเอียดของภาพที่เป็นองค์ประกอบทางความถี่สูง

จากการทดลองพบว่าเมื่อทดลองนำภาพ Lena ซึ่งเป็นภาพมาตรฐาน มาทำการทดลองลดขนาดข้อมูลภาพโดยการแปลงเวฟเล็ต ที่นำสัมพันธ์ของภาพที่ได้จากการแปลงเวฟเล็ตมาผ่านการเข้ารหัสแบบเอ็มเบดซีโรที พบว่าค่าของสัมพันธ์ที่ได้เมื่อนำมาเข้ารหัสจะเกิดซีโรทีที่น้อยเนื่องจากภาพ Lena จะมีรายละเอียดของภาพที่มาก

5.2 บทสรุป

ในการลดขนาดข้อมูลภาพ ด้วยวิธีการแปลงเวฟเล็ต แล้วนำสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ต ไปผ่านการเข้ารหัสแบบเอ็มเบดซีโรทรี จากการทดลองพบว่า เมื่อทำการปรับค่าเทรชโฮลที่มีค่าน้อยลง จะได้ภาพที่มีความละเอียดสูงขึ้น แต่ค่าอัตราการบีบอัดข้อมูลจะลดลง

แนวทางปรับปรุงและพัฒนา

ควรทดลองห้ เฟังก์ชันอื่นๆ มาใช้งานให้เหมาะสมกับข้อมูลภาพ เพื่อจะได้เหมาะสมกับการแปลงข้อมูลมากที่สุด

โปรแกรมการเข้ารหัสแบบเอ็มเบดซีโรทรี มีความซับซ้อน และช้า อันเนื่องมาจากการสแกนหาค่าสัมประสิทธิ์ ถ้าปรับปรุงวิธีการสแกนค่าสัมประสิทธิ์ให้รวดเร็วขึ้นแล้วก็จะมีความน่าใช้ขึ้น



ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนโปรแกรมใช้งาน

```

// pic1Dlg.cpp : implementation file
//

#include "stdafx.h"
#include "pic1.h"
#include "pic1Dlg.h"
#include "math.h"
#include "IMOPEN.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "fcntl.h"
#include "io.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

// CPic1Dlg dialog
static char buffer[256+2];
static char *current_byte;
static int output_mask;
static int input_bytes_left;
static int input_bits_left;
static int past_eof;
static unsigned short int code;
static unsigned short int low;
static unsigned short int high;
int level;
int size, leve;
int chksizel, chksize2;
char *output;
BYTE *In;
float *pica;
double *indicator, *data, *colour, *pic_inv, *pic_haar, *pic_decode;
long int u;
int v, ongY, ongX, allndnum=-1, allsnum=-1;
BYTE result[500000];
//*****VC++ create*****//

CPic1Dlg::CPic1Dlg(CWnd* pParent /*=NULL*/)
: CDialog(CPic1Dlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CPic1Dlg)
    m_lev = -1;
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDI_STAR1);
}

void CPic1Dlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CPic1Dlg)
    DDX_Control(pDX, IDC_BPP, m_bpp);
}

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DDX_Control(pDX, IDC_BROWSE, m_browsebutt);
DDX_Control(pDX, IDC_LEVEL, m_ctrlllevel);
DDX_Control(pDX, IDC_INVHAAR, m_invhaarbutt);
DDX_Control(pDX, IDC_RMS, m_rmsbutt);
DDX_Control(pDX, IDC_RESULTS, m_resultsbutt);
DDX_Control(pDX, IDC_COMM, m_commbutt);
DDX_Control(pDX, IDC_DECODE, m_decodebutt);
DDX_Control(pDX, IDC_ENCODE, m_encodebutt);
DDX_Control(pDX, IDC_DEEZW, m_deezwbutt);
DDX_Control(pDX, IDC_EZW, m_ezwbutt);
DDX_Control(pDX, IDC_HAAR, m_haarbutt);
DDX_Control(pDX, IDC_PIC, m_picframe);
DDX_Control(pDX, IDC_THRESHOLD, m_threshold);
DDX_Control(pDX, IDC_PRODEEZW, m_prodeezw);
DDX_Control(pDX, IDC_PROEZW, m_proezw);
DDX_CBIndex(pDX, IDC_LEVEL, m_lev);
//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CPic1Dlg, CDialog)
//{{AFX_MSG_MAP(CPic1Dlg)
ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_BN_CLICKED(IDC_BROWSE, OnBrowse)
ON_BN_CLICKED(IDC_HAAR, OnHaar)
ON_BN_CLICKED(IDC_ENCODE, OnEncode)
ON_BN_CLICKED(IDC_DECODE, OnDecode)
ON_BN_CLICKED(IDC_EZW, OnEzw)
ON_CBN_SELCHANGE(IDC_LEVEL, OnLevel)
ON_BN_CLICKED(IDC_DEEZW, OnDeezw)
ON_BN_CLICKED(IDC_COMM, OnComm)
ON_BN_CLICKED(IDC_RESULTS, OnResults)
ON_BN_CLICKED(IDC_RMS, OnRms)
ON_BN_CLICKED(IDC_INVHAAR, OnInvhaar)
ON_BN_CLICKED(IDC_RADBPP, OnRadbpp)
ON_BN_CLICKED(IDC_RADTHRES, OnRadthres)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

// CPic1Dlg message handlers
BOOL CPic1Dlg::OnInitDialog()
{
    x.Create(NULL, WS_VISIBLE|SS_BITMAP, CRect(0,0,600,600), this, IDC_STAT
    IC);
    CDialog::OnInitDialog();

    // Set the icon for this dialog. The framework does this
    automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon
    //sh = x.GetDC();
    sh = m_picframe.GetDC();
    //totals=new short int;
    Cresult=new char[100];
    totals=storage+1;
    load = false;

    m_invhaarbutt.EnableWindow(FALSE);
    m_ezwbutt.EnableWindow(FALSE);
    m_encodebutt.EnableWindow(FALSE);

```

```

m_commbutt.EnableWindow(TRUE);
m_resultsbutt.EnableWindow(FALSE);
m_rmsbutt.EnableWindow(FALSE);
m_deezwbutt.EnableWindow(FALSE);
m_haarbutt.EnableWindow(FALSE);
m_ctrllevel.EnableWindow(FALSE);
m_ctrllevel.SetCurSel(0);
m_threshold.EnableWindow(FALSE);
m_threshold.SetCurSel(0);
m_bpp.EnableWindow(FALSE);
m_bpp.SetCurSel(0);
CheckDlgButton(IDC_RADBPP, 2);
GetDlgItem(IDC_RADBPP)->EnableWindow(FALSE);
GetDlgItem(IDC_RADTHRES)->EnableWindow(FALSE);
level = 0;
m_proezw.SetRange(0, 30);
m_proezw.SetStep(2);
m_proezw.SetPos(0);
m_prodeezw.SetRange(0, 30);
m_prodeezw.SetStep(2);
m_prodeezw.SetPos(0);
// TODO: Add extra initialization here return TRUE;
// return TRUE unless you set the focus to a control
}

// If you add a minimize button to your dialog, you will need the
code below
// to draw the icon. For MFC applications using the document/view
model,
// this is automatically done for you by the framework.

void CPic1Dlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;
        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the
user drags
// the minimized window.
HCURSOR CPic1Dlg::OnQueryDragIcon()
{
    return (HCURSOR)m_hIcon;
}

```

เอกสารนี้เป็นเอกสารที่ **return (HCURSOR)m_hIcon;** เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
//*****End of part that VC++ create*****//
void CPic1Dlg::AddDlgItemText(int nID,LPCTSTR str)
{ CString g;
  GetDlgItemText(nID,g);
  g=g+str;
  SetDlgItemText(nID,g);
}

unsigned int CPic1Dlg::calpos(unsigned int y, unsigned int x)
{
  unsigned int result;
  result = (y*WIDTH)+x;
  return result;
}

double CPic1Dlg::abscon(double x) //absolute
{ double ans;
  ans = x;
  if (x < 0) ans = x*(-1);
  return ans;
}

void CPic1Dlg::showim(double *im,unsigned int y,unsigned int x,int z)
{ //for showing picture//
  int iy,ix;
  float hue1;
  COLORREF hue2;
  HDC hdc;
  if (z==0)
  { for (iy=0;iy<y;iy++)
    {
      for(ix=0;ix<x;ix++)
      { hue1 = 1*im[calpos(iy,ix)];/////normal
        hue2 = RGB((int)(hue1+0.5),(int)(hue1+0.5),(int)(hue1+0.5));
        sh->SetPixel(ix,iy,hue2);
      }
    }
  }
  else
  { for (iy=0;iy<y;iy++)
    {
      for(ix=0;ix<x;ix++)
      { hue1 = 255*im[calpos(iy,ix)];/////Haar
        hue2 = RGB((int)(hue1+0.5),(int)(hue1+0.5),(int)(hue1+0.5));
        sh->SetPixel(ix,iy,hue2);
      }
    }
  }
  UpdateWindow();
}

void CPic1Dlg::OnBrowse() //Create Browse Dialog//
{
  char temp[10];
  int ab,ix,iy,bc,y,z;
  long H, W;
  CString tem,zero;
  char *tem1 = new char[5];
  BYTE *tp = new BYTE[1000],*tp1;

```

```

CFileDialog fd(TRUE,NULL,NULL,NULL,"BMP File (*.bmp)|*.bmp|All File
(*.*)|*.*||"); //Show BMP File or All File//
if(fd.DoModal()==IDOK)
{ BeginWaitCursor();
  filename = fd.GetFileName();
  SetDlgItemText(IDC_GETPICTURE,filename); //Show name of file
  that you choose//
  fp = fopen(filename,"rb+");
  //_setmode(fp,_O_BINARY);
  if (fp != NULL)
  { fread( &bmfh, sizeof(BITMAPFILEHEADER), 1, fp);
    //ab = GetLastError();
    fread( &bmih, sizeof(BITMAPINFOHEADER), 1, fp);
    sprintf(temp, "%d X %d", bmih.biWidth, bmih.biHeight);
    SetDlgItemText(IDC_HEADER, temp);
    //Show width and height of picture//
    H = 512; W = 512;
    if (bmih.biHeight < 512) H = bmih.biHeight;
    //if height<512 then height=real height//
    if (bmih.biWidth < 512) W = bmih.biWidth;
    //if width<512 then width=real width//
    WIDTH=bmih.biWidth;//if width>512 then show width=512//
    HEIGHT=bmih.biHeight;//if height>512 then show height=512//

    float nx=(float)bmih.biWidth ,ny=(float)bmih.biHeight;
    size = WIDTH*HEIGHT;
    pic = new double[size];

    if (bmih.biBitCount != 8) //8 bit//
    { MessageBox("Not 256 color Bitmap.");
      return;
    }
    hpicture=(HBITMAP) LoadImage(NULL,filename,IMAGE_BITMAP
,W,H,LR_LOADFROMFILE);
    if(hpicture==NULL)
    { MessageBox("Error not BMP");
      return;
    }
    ab = bmih.biWidth % 4;
    bc=4-ab;
    fseek(fp, bmfh.bfOffBits, SEEK_SET);
    for (iy=bmih.biHeight-1;iy>=0;iy--)
    {
      y=(int)bmih.biWidth/128;
      z=bmih.biWidth%128;
      tp1=tp;
      if (y>0)
      { fread(tp,128,y,fp);
        tp1=tp+(y*128);
      }
      if (z>0)

      fread(tp1,z,1,fp);
      for (ix=0;ix<bmih.biWidth;ix++)
      pic[calpos(iy,ix)] = *(tp+ix);
      if ( (iy!=0) && (ab!=0) )
      fseek(fp,bc, SEEK_CUR);
    }
    lvl=log(bmih.biWidth)/log(2);
    if (level == 0) level = lvl;
    load = true;
  }
}

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        OnLevel();
        fclose(fp);
        delete tem1;
        delete tp;
        EndWaitCursor();

        showim(pic,bmih.biHeight,bmih.biWidth,0);//Show picture//
        m_haarbutt.EnableWindow(TRUE);
        m_ctrllevel.EnableWindow(TRUE);
    }
}

void CPic1Dlg::OnLevel()
{ UpdateData();
  if (m_ctrllevel.GetCurSel() == 0) level = lvl;
  else level = m_lev;
}
//*****Haar Function*****//
void CPic1Dlg::OnHaar()
{
    int iy,ix,lev;
    float r2, nx=(float)bmih.biWidth ,ny=(float)bmih.biHeight,wx,wy;
    double *picb, *picc, *picd;
    pic_haar = new double[size];
    picb = new double[size];
    picc = new double[size];
    picd = new double[size];
    m_decodebutt.EnableWindow(FALSE);
    for(iy=0;iy<ny;iy++) //divide by coefficient//
        for(ix=0;ix<nx;ix++)
            { pic_haar[calpos(iy,ix)]=pic[calpos(iy,ix)]/sqrt(ny);
              pic_haar[calpos(iy,ix)]=pic_haar[calpos(iy,ix)]/sqrt(nx);
            }
    r2= sqrt(2);
    wx=nx;wy=ny;
    for (lev=0;lev<level;lev++)//number of level//
    { for(iy=0;iy<wy;iy++)
      for(ix=0;ix<wx;ix++)
        picb[calpos(iy,ix)]=pic_haar[calpos(iy,ix)];
      for(iy=0;iy<wy;iy++) //horizontal wavelet transform//
      { for(ix=0;ix<wx/2;ix++)
        {
            picc[calpos(iy,ix)]=(((picb[calpos(iy,2*ix)])
            +(picb[calpos(iy,2*ix+1)]))/r2);
            picc[calpos(iy,(int)wx/2+ix)]=(((picb[calpos(iy,2*ix)])
            -(picb[calpos(iy,2*ix+1)]))/r2);
        }

        for(ix=0;ix<wx;ix++)
            picb[calpos(iy,ix)]=picc[calpos(iy,ix)];
    }
    for(ix=0;ix<wx;ix++) //vertical wavelet transform//
    { for(iy=0;iy<wy/2;iy++)
      {
            picc[calpos(iy,ix)]=(((picb[calpos(2*iy,ix)])
            +(picb[calpos(2*iy+1,ix)]))/r2);
            picc[(int)calpos(wx/2+iy,ix)]=(((picb[calpos(2*iy,ix)])
            -(picb[calpos(2*iy+1,ix)]))/r2);
        }
        for(iy=0;iy<wy;iy++)

```

```

        picb[calpos(iy,ix)]=picc[calpos(iy,ix)];
    }
    for(iy=0;iy<wy;iy++)
        for(ix=0;ix<wx;ix++)
            pic_haar[calpos(iy,ix)]=picb[calpos(iy,ix)];
    wx=wx/2;wy=wy/2;
    for(iy=0;iy<wy;iy++)
        { for(ix=0;ix<wx;ix++)
            {
                picd[calpos(iy,ix)] = pic_haar[calpos(iy,wx+ix)];
                pic_haar[calpos(iy,wx+ix)] = pic_haar[calpos(wy+iy,ix)];
                pic_haar[calpos(wy+iy,ix)] = picd[calpos(iy,ix)] ;
            }
        }
    }
    showim(pic_haar, bmih.biHeight, bmih.biWidth,1);
    indicator = pic_haar;
    m_invhaarbutt.EnableWindow(TRUE);
    m_ezwbutt.EnableWindow(TRUE);
    m_bpp.EnableWindow(TRUE);
    m_browsebutt.EnableWindow(FALSE);
    GetDlgItem(IDC_RADBPP)->EnableWindow(TRUE);
    GetDlgItem(IDC_RADTHRES)->EnableWindow(TRUE);
    m_haarbutt.EnableWindow(FALSE);
    m_ctrllevel.EnableWindow(FALSE);
    free(picb);
    free(picc);
    free(picd);
}

void CPic1Dlg::callInvwavelet()
{ indicator = pic_decode;
  OnInvwavelet();
}

void CPic1Dlg::OnInvhaar()
{ indicator = pic_haar;
  leve = level;
  OnInvwavelet();
}

//*****Invert Wavelet Transform*****//
void CPic1Dlg::OnInvwavelet()
{ int iy,ix,lev,i;
  float r2, nx=(float)WIDTH ,ny=(float)HEIGHT,wx,wy;
  double *picx, *picy, *picz;
  pic_inv = new double[size];
  picx = new double[size];
  picy = new double[size];
  picz = new double[size];
  r2= sqrt(2);
  wx=nx;wy=ny;
  for (ix=0;ix<wx;ix++)
    for(iy=0;iy<wy;iy++)
      picx[calpos(iy,ix)]=indicator[calpos(iy,ix)];
    for (i=0;i<leve;i++)
      { wy = wy/2;
        wx = wx/2;
      }
    for (lev=0;lev<leve;lev++)
      {
        for(iy=0;iy<wy;iy++)

```

```

    { for(ix=0;ix<wx;ix++)
      {
        picz[calpos(iy,ix)] = picx[calpos(iy,wx+ix)];
        picx[calpos(iy,wx+ix)] = picx[calpos(wy+iy,ix)];
        picx[calpos(wy+iy,ix)] = picz[calpos(iy,ix)] ;
      }
    }
  }
  //////////////////////////////////
  wx=2*wx;wy=2*wy;
  for(ix=0;ix<wx;ix++)//horizontal invert wavelet transform//
  { for(iy=0;iy<wy/2;iy++)
    {
      picy[calpos(2*iy,ix)]=(((picx[calpos(iy,ix)])
      + (picx[calpos(wx/2+iy,ix)]))/r2);
      picy[calpos((int)2*iy+1,ix)]=(((picx[calpos(iy,ix)])
      - (picx[calpos(wx/2+iy,ix)]))/r2);
    }
    for(iy=0;iy<wy;iy++)
      picx[calpos(iy,ix)]=picy[calpos(iy,ix)];
  }
  for(iy=0;iy<wy;iy++)//vertical invert wavelet transform//
  { for(ix=0;ix<wx/2;ix++)
    {
      picy[calpos(iy,2*ix)]=(((picx[calpos(iy,ix)])
      + (picx[calpos(iy,wx/2+ix)]))/r2);
      picy[calpos(iy,(int)2*ix+1)]=(((picx[calpos(iy,ix)])
      - (picx[calpos(iy,wx/2+ix)]))/r2);
    }

    for(ix=0;ix<wx;ix++)
      picx[calpos(iy,ix)]=picy[calpos(iy,ix)];
  }
}
for(iy=0;iy<ny;iy++)//multiplied coefficient//
for(ix=0;ix<nx;ix++)
{
  pic_inv[calpos(iy,ix)]=picx[calpos(iy,ix)]*sqrt(ny);
  pic_inv[calpos(iy,ix)]=pic_inv[calpos(iy,ix)]*sqrt(nx);
}

showim(pic_inv,HEIGHT,WIDTH,0);
free(picx);
free(picy);
free(picz);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//*****Arithmetic Coding*****//
void CPic1Dlg::Encode(CString ininame, CString outfname)
{
    CPic1Dlg::SYMBOL s;
    int c;
    initialize_model();
    initialize_output_bitstream();
    initarithencode();
    if (ininame == "")
    {
        MessageBox("Please Specify File Name To Encode");
        CFileDialog fs(TRUE,NULL,NULL,NULL,"Text file|*.txt|All File (*.*)|*.*||"); //show text file and all file//
        if(fs.DoModal()!=IDOK) return;
        ininame=fs.GetPathName();
    }
    else ininame = ininame;
    infile=fopen(ininame,"rb");
    if (outfname == "")
    {
        MessageBox("Please Specify Recorded File Name");
        CFileDialog fe(FALSE,NULL,NULL,NULL,"Text file|*.txt|All File (*.*)|*.*||"); //show text file and all file//
        if(fe.DoModal()!=IDOK) return;
        outname=fe.GetPathName();
    }
    else outname = outfname;
    outfile=fopen(outname,"wb");
    for (; ;)
    {
        c=getc(infile);
        int2sym(c,&s);
        encode_symbol(outfile,&s);
        if (c==EOF) break;
        update_model(c);
    }
    flushcode(outfile);
    flushstreambit(outfile);
    fputc('\n',stderr);
    sprintf(Cresult,"%ld compress to %ld",ftell(infile),bit_ftell_output(outfile));
    if (ininame == "" && outfname == "")
    SetDlgItemText(IDC_ARCODEIN,Cresult);
    fclose(infile);
    fclose(outfile);
    m_decodebutt.EnableWindow(TRUE);
    m_encodebutt.EnableWindow(FALSE);
}
//*****//
void CPic1Dlg::Expand(void)
{
    CPic1Dlg::SYMBOL s;
    CFileDialog fs(TRUE,NULL,NULL,NULL,"Text file|*.txt|All File (*.*)|*.*||");
    CFileDialog fe(FALSE,NULL,NULL,NULL,"Text file|*.txt|All File (*.*)|*.*||");
    MessageBox("Please Specify File Name To Decode ");
    if(fs.DoModal()!=IDOK) return;
    ininame=fs.GetPathName();
    infile=fopen(ininame,"rb");
    MessageBox("Please Specify File Name To Record");

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(fe.DoModal()!=IDOK) return;
outname=fe.GetPathName();
outfile=fopen(outname,"wb");
int c,count;
initialize_model();
initialize_input_bitstream();
initarithdecode(infile);
for(;;)
{
    get_symbol_scale(&s);
    count=get_current_count(&s);
    c=sym2int(count,&s);
    remove_bitstream(infile,&s);
    if(c==EOF)
        break;
    putc((char)c,outfile);
    update_model(c);
}
fputc('\n',stderr);
sprintf(Cresult,"%ld decompress to %ld",
bit_ftell_input(infile),ftell(outfile));

SetDlgItemText(IDC_ARCODEOUT,Cresult);
fclose(infile);
fclose(outfile);
m_decodebutt.EnableWindow(FALSE);
}
//*****//
void CPic1Dlg::initialize_output_bitstream()
{
    current_byte=buffer;
    *current_byte=0;
    output_mask=0x80;
}
//*****//
void CPic1Dlg::output_bit(FILE *stream,int bit)
{
    if (bit)
        *current_byte|=output_mask;
    output_mask>>=1;
    if (output_mask==0)
    {
        output_mask=0x80;
        current_byte++;
        if(current_byte==(buffer+BUFFER_SIZE))
        {
            fwrite(buffer,1,BUFFER_SIZE,stream);
            current_byte=buffer;
        }
        *current_byte=0;
    }
}
//*****//
void CPic1Dlg::flushstreambit(FILE *stream)
{
    fwrite(buffer,1,(size_t)(current_byte-buffer)+1,stream);
    current_byte=buffer;
}
//*****//
void CPic1Dlg::initialize_input_bitstream()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

input_bits_left=0;
input_bytes_left=1;
past_eof=0;
}
//*****//
short int CPic1Dlg::input_bit(FILE *stream)
{
    if(input_bits_left==0)
    {
        current_byte++;
        input_bytes_left--;
        input_bits_left=8;
        if(input_bytes_left==0)
        {
            input_bytes_left=fread(buffer,1,BUFFER_SIZE,stream);
            if(input_bytes_left==0)
            {
                if(past_eof)
                {
                    fprintf(stderr,"Bad input file\n");
                    exit(-1);
                }
                else
                {
                    past_eof=1;
                    input_bytes_left=2;
                }
            }
            current_byte=buffer;
        }
        input_bits_left--;
        return((*current_byte>>input_bits_left)&1);
    }
//*****//
long CPic1Dlg::bit_ftell_output(FILE *stream)
{
    long total;
    total=ftell(stream);
    total+=current_byte-buffer;
    total+=underflow_bits/8;
    return(total);
}
//*****//
long CPic1Dlg::bit_ftell_input(FILE *stream)
{
    return(ftell(stream)-input_bytes_left+1);
}
//*****//
void CPic1Dlg::initarithencode()
{
    low=0;
    high=0xffff;
    underflow_bits=0;
}
//*****//
void CPic1Dlg::encode_symbol(FILE *stream,SYMBOL *s)
{
    long range;
    range=(long) (high-low)+1;
    high=low+(unsigned short int)

```

```

    ((range*s->high_count)/s->scale-1);
    low=low+(unsigned short int)
    ((range *s->low_count)/s->scale);
    for(;;)
    {
        if((high&0x8000)==(low&0x8000))
        {
            output_bit(stream,high&0x8000);
            while(underflow_bits>0)
            {
                output_bit(stream,~high&0x8000);
                underflow_bits--;
            }
        }
        else if((low&0x4000)&&!(high&0x4000))
        {
            underflow_bits+=1;
            low&=0x3fff;
            high|=0x4000;
        }
        else
            return;
        low<<=1;
        high<<=1;
        high|=1;
    }
}
//*****//
void CPic1Dlg::flushcode(FILE *stream)
{
    output_bit(stream,low&0x4000);
    underflow_bits++;
    while(underflow_bits-->0)
        output_bit(stream,~low&0x4000);
}
//*****//
short int CPic1Dlg::get_current_count(SYMBOL *s)
{
    long range;
    short int count;
    range=(long)(high - low)+1;
    count=(short int)
        (((long)(code-low)+1)*s->scale-1)/range;
    return(count);
}
//*****//
void CPic1Dlg::initarithdecode(FILE *stream)
{
    int i;
    code=0;
    for(i=0;i<16;i++)
    {
        code<<=1;
        code+=input_bit(stream);
    }
    low=0;
    high=0xffff;
}
//*****//
void CPic1Dlg::remove_bitstream(FILE *stream,SYMBOL *s)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

long range;
range=(long) (high-low)+1;
high =low+(unsigned short int)
((range*s->high_count)/s->scale-1);
low=low+(unsigned short int)
((range *s->low_count)/s->scale);
for(;;)
{
    if((high&0x8000)==(low&0x8000))
    {
    }
    else
    if((low & 0x4000)==0x4000&&(high&0x4000)==0)
    {
        code^=0x4000;
        low &=0x3fff;
        high|=0x4000;
    }
    else
    return;
    low<<=1;
    high<<=1;
    high|=1;
    code<<=1;
    code+=input_bit(stream);
}
}
//*****//
void CPic1Dlg::initialize_model()
{
    short int i;
    for(i=-1;i<=256;i++)
        totals[i]=i+1;
}
//*****//
void CPic1Dlg::update_model(int symbol)
{
    int i;
    for(symbol++;symbol<=256;symbol++)
        totals[symbol]++;
    if(totals[256]==MAXIMUM_SCALE)
    {
        for(i=0;i<=256;i++)
        {
            totals[i]/=2;
            if(totals[i]<=totals[i-1])
                totals[i]=totals[i-1]+1;
        }
    }
}
//*****//
int CPic1Dlg::int2sym(int c,SYMBOL *s)
{
    s->scale=totals[256];
    s->low_count=totals[c];
    s->high_count=totals[c+1];
    return(0);
}
//*****//
void CPic1Dlg::get_symbol_scale(SYMBOL *s)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    s->scale=totals[256];
}
//*****//
int CPic1Dlg::sym2int(int count,SYMBOL *s)
{
    int c;
    for (c=255;count<totals[c];c--);
    s->high_count=totals[c+1];
    s->low_count=totals[c];
    return(c);
}
//*****//
void CPic1Dlg::OnEncode() //function main of encode
{ Encode("", ""); //call function encode//
}
//*****//
void CPic1Dlg::OnDecode() //function main of decode
{
    Expand(); //call function expand//

    m_deezwbutt.EnableWindow(TRUE);
    m_encodebutt.EnableWindow(FALSE);
    m_haarbutt.EnableWindow(FALSE);
    m_invhaarbutt.EnableWindow(FALSE);
}

//*****End of Arithmetic Coding*****//
//*****Embedded Zerotree*****//
int CPic1Dlg::qwidth(int level) // find width of quadrant in this
                                level
{ return WIDTH/pow(2, level);
}
//*****//
int CPic1Dlg::qheight(int level) // find height of quadrant in
                                this level
{ return HEIGHT/pow(2, level);
}
//*****//
int CPic1Dlg::startx(int quad, int level)// start of x
{ int tmp;
  tmp = ( (quad+1) % 2 ) * qwidth(level);
  return tmp;
}
//*****//
int CPic1Dlg::starty(int quad, int level)// start of y
{ int tmp;
  tmp = ( (quad) / 3 ) * qheight(level);
  return tmp;
}
//*****//
void CPic1Dlg::setbuff(int y, int x, int val) // Set symbol
{
  buff[calpos(y,x)] = val;
}
//*****//
int CPic1Dlg::chkval(int y, int x) // check one pixel only P, N and Z
will return
{ int val;
  if (data[calpos(y,x)] >=T) val = P;
  else if (data[calpos(y,x)] <= -T) val=N;
  else val = Z;
}

```

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return val;
}
//*****//
int CPic1Dlg::isgroupiz(int nexty, int nextx) // check 4 pixel
nearest for P or N
{ int yes = 0;
  if (chkval(nexty, nextx) != Z && check[calpos(nexty,nextx)]==false)
    yes = 1;
  if (chkval(nexty, nextx+1) != Z &&
    check[calpos(nexty,nextx+1)]==false) yes = 1;
  if (chkval(nexty+1, nextx) != Z &&
    check[calpos(nexty+1,nextx)]==false) yes = 1;
  if (chkval(nexty+1, nextx+1) != Z &&
    check[calpos(nexty+1,nextx+1)]==false) yes = 1;
  return yes;
}
//*****//
int CPic1Dlg::find(int y, int x, int quad, int level)//check ZT or IZ
{ int val, ans, diffx, diffy, nextx, nexty;
  int ans1, ans2, ans3, ans4;
  val = chkval(y, x);
  if ( (val == P) || (val == N) ) ans = val;
  else if (level == 1) ans = Z;
  else
  { diffx = x - startx(quad, level);
    diffy = y - starty(quad, level);
    nextx = startx(quad, level-1) + (2 * diffx);
    nexty = starty(quad, level-1) + (2 * diffy);
    if (isgroupiz(nexty, nextx) == 1) ans = IZ;
    // for faster calculation
    else
    { if (check[calpos(nexty,nextx)]==true) ans1=ZT;
      else ans1 = find(nexty, nextx, quad, level-1);
      if (check[calpos(nexty,nextx+1)]==true) ans2=ZT;
      else ans2 = find(nexty, nextx+1, quad, level-1);
      if (check[calpos(nexty+1,nextx)]==true) ans3=ZT;
      else ans3 = find(nexty+1, nextx, quad, level-1);
      if (check[calpos(nexty+1,nextx+1)]==true) ans4=ZT;
      else ans4 = find(nexty+1, nextx+1, quad, level-1);
      ans = ZT;
      if (ans1 == P || ans1 == N || ans1 == IZ) ans = IZ;
      else if (ans2 == P || ans2 == N || ans2 == IZ) ans = IZ;
      else if (ans3 == P || ans3 == N || ans3 == IZ) ans = IZ;
      else if (ans4 == P || ans4 == N || ans4 == IZ) ans = IZ;
    }
  }
  return ans;
}
//*****//
int CPic1Dlg::parentx(int y,int x,int quad, int level)
//check parent of that pixel
{ int diffx, px = 0;
  if (level <= MAXLEVEL)
  { diffx=x-startx(quad, level);
    px=startx(quad, level+1)+((diffx)/2);
  }
  return px;
}
//*****//
int CPic1Dlg::parenty(int y,int x,int quad, int level)
//check parent of that pixel

```

เอกสารนี้ //check parent of that pixel เป็นทรัพย์สินของมหาวิทยาลัยสุโขทัย มิได้อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ int diffy, py = 0;
  if (level <= MAXLEVEL)
    { diffy=y-starty(quad,level);
      py=starty(quad,level+1)+((diffy)/2);
    }
  return py;
}

//*****//
int CPic1Dlg::parent(int y,int x,int quad, int level)
//check parent of that pixel
{ int px, py;

  px=parentx(y, x, quad,level);
  py=parenty(y, x, quad,level);
  return buff[calpos(py,px)];
}
//*****//

void CPic1Dlg::OnEzw() //function main of Embedded Zerotree
{ int j,k,p,sub,dom,Ans4,even,even1,part,count,test;
  int Scan,Level, QHeight, QWidth, Quad ,Sy,Sx, X, Y, A;
  int Ans,Ans2,parent_1,outcount,y,ix,iy,t2,t0,t3, radiobutt;
  int dnum,snum,px, py, upscan=0,alldnum=-1,allnum=-1;
  int fhandle;
  long fsize;
  float nx=(float)bmih.biWidth ,ny=(float)bmih.biHeight,wide,dx;
  float t1,bppnow,bppsel,diff;
  CString result,Symbol;
  CString tmpinfilename, tmpoutfilename;

  chksize1 = 0;
  chksize2 = 0;
  tmpinfilename = "~tmp~in.tmp";
  tmpoutfilename = "~tmp~out.tmp";
  output = new char[1500000];
  store = new BYTE[size];
  symb = new BYTE[size];
  buff = new char[size];
  check = new bool[size];
  data = new double[size];

  CFileDialog ft(FALSE,NULL,NULL,OFN_OVERWRITEPROMPT,"Text
file|*.txt|Document file|*.doc|ALL FILES(*.*)|*.*||");
  if (ft.DoModal()==IDOK)
    OutputFileName = ft.GetPathName();
  else return;
  if((wp=fopen(OutputFileName,"wb")) == NULL)
  { MessageBox("Can not open write file.");
    return;
  }

  // Temporary File to calculate BPP
  tmpin=fopen(tmpinfilename,"wb");

  // Additional Scan Loop
  upscan = m_threshold.GetCount() - m_threshold.GetCurSel();

  // Get Data from Haar
  for (ix=0;ix<nx;ix++)
    for (iy=0;iy<ny;iy++)

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

data[calpos(iy,ix)]=pic_haar[calpos(iy,ix)];

// Initial Threshold
t1 = ceil((data[calpos(0,0)])/2);
t2 = ceil(log(t1)/log(2));
t0 = pow(2,t2);

// Initial Data
T=t0;
t2 =t2+upscan;
t3=2*t0;
outcount=0;
u=0;
v=0;
count=0;
radiobutt = GetCheckedRadioButton(IDC_RADBPP, IDC_RADTHRES);
bppsel = (m_bpp.GetCurSel()+1)*0.1;

// Write header to file
fwrite(&size,sizeof(int),1,wp); // Number of pixel
fwrite(&t0,sizeof(int),1,wp); // Initial Threshold
fwrite(&t2,sizeof(int),1,wp); // Scan number
fwrite(&level,sizeof(int),1,wp); // Level wavelet number
chksize2 += 4*sizeof(int);

// Write to Temporary file
fwrite(&size,sizeof(int),1,tmpin); // Number of pixel
fwrite(&t0,sizeof(int),1,tmpin); // Initial Threshold
fwrite(&t2,sizeof(int),1,tmpin); // Scan number
fwrite(&level,sizeof(int),1,tmpin); // Level wavelet number
chksize1 += 4*sizeof(int);

setbuff(0,0, P); // for first value
// MAXLEVEL is end inner level (1 pixel/Quadrant)
MAXLEVEL = ceil(log(WIDTH)/log(2));
BeginWaitCursor();
output[outcount++]=P; //get 1 into output

// Reset Data
for (Y=0;Y<HEIGHT;Y++)
  for (X=0;X<WIDTH;X++)
    { check[calpos(Y,X)] = false;
      store[calpos(Y,X)] = 0;
    }
for (Scan = 1;Scan<=t2; Scan++)
{ if (Scan==1) dnum=1;
  else dnum=0;
  snum=0;
  check[calpos(0,0)] = true; //don't scan this pixel again
  count=(2*count)+2;

  for (Level = MAXLEVEL; Level >= 1; Level--)
  { QHeight = qheight(Level);
    QWidth = qwidth(Level);
    for (Quad = 2; Quad <= 4; Quad++)
    { Sx = startx(Quad, Level);
      Sy = starty(Quad, Level);
      for (Y = Sy; Y <= Sy+QHeight-1; Y++)
        for (X = Sx; X <= Sx+QWidth-1; X++)
          {

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภายใต้การดูแลของศูนย์คอมพิวเตอร์และระบบสารสนเทศของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        setbuff(Y, X, Ans); //call function setbuff

if (check[calpos(Y,X)]==false)
{ if(Level==MAXLEVEL && Scan==1)
  {
    output[outcount++]=(char)Ans;
    dnum=dnum+1;
    if(Ans==P||Ans==N) check[calpos(Y,X)]=true; //for don't
    scan that pixel again
  }
  else
  { parent_1=parent(Y,X,Quad,Level); //call function parent
    if (parent_1 == P || parent_1== N || parent_1 == IZ)
    { px = parentx(Y, X, Quad, Level);
      //call function parentx
      py = parenty(Y, X, Quad, Level);
      //call function parenty
      if (check[calpos(Y,X)] == false)
      //pixel isn't scanned
      {
        output[outcount++]=(char)Ans; //keep dominant
        dnum=dnum+1; //count dominant
        if(Ans==P||Ans==N)
        { check[calpos(Y,X)]=true;
          }
        }
      }
    }
  } //X
} //Quad
} //Level
//****finding first subordinate****//
wide=(t3-T)/count;
dx=abscon(data[calpos(0,0)]-T;
part=floor(dx/wide);
even1 = part % 2;
if (even1==0);
  Ans4=0; //subordinate=0
if (even1==1)
  Ans4=1; //subordinate=1
output[outcount++] = Ans4; //keep first subordinate
snum=snum+1; //count subordinate
//****finding the next subordinate****//
for(j=1; j<=Scan; j++)
{ for (Level = MAXLEVEL; Level >= 1; Level--)
  { QHeight = qheight(Level);
    QWidth = qwidth(Level);
    for (Quad = 2; Quad <= 4; Quad++)
    { Sx = startx(Quad, Level);
      Sy = starty(Quad, Level);
      for (Y = Sy; Y <= Sy+QHeight-1; Y++)
      for (X = Sx; X <= Sx+QWidth-1; X++)
      { if (buff[calpos(Y,X)]==P || buff[calpos(Y,X)]==N)
        { if (store[calpos(Y,X)]==0) store[calpos(Y,X)]=Scan;
          wide=(t3-T)/count;
          dx=abscon(data[calpos(Y,X)]-T;
          part=floor(dx/wide);
          even = part % 2;
          if (even==0)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

}

void CPic1Dlg::makehead(int x,int y)
{ // Beware index exceed of (output and result)
  int ix,bit;
  BYTE temp,domend,subend;
  fwrite(&x,sizeof(int),1,wp);
  fwrite(&y,sizeof(int),1,wp);

  // Write Temporary file
  fwrite(&x,sizeof(int),1,tmpin);
  fwrite(&y,sizeof(int),1,tmpin);

  bit=3;
  for ( ix=1;ix<=x;ix++)
  { if (bit==3) temp=0;
    domend=output[u++];
    temp = temp | (domend<<((bit--)*2));
    if( (bit == -1)|| (ix==x) )
    { bit=3;
      result[v++]=temp;
      fwrite(&temp,1,1,wp);
      chksize2++;
      // Write Temporary file
      fwrite(&temp,1,1,tmpin);
      chksize1++;
    }
  }
  bit=7;
  temp=0;
  for ( ix=1;ix<=y;ix++)
  { if (bit==7) temp=0;
    subend=output[u++];
    temp = temp | (subend<<(bit--));
    if( (bit == -1)|| (ix==y) )
    { bit=7;
      result[v++]=temp;
      fwrite(&temp,1,1,wp);
      chksize2++;
      // Write Temporary file
      fwrite(&temp,1,1,tmpin);
      chksize1++;
    }
  }
}

//*****Invert Embedded Zerotree*****//
void CPic1Dlg::OnDeezw()
{ int t0,t2,nxny,ix,iy,Scan,bit;
  int x,y,dompos=-1, subpos=-1;
  //int v=0;
  v=0;
  float wid;
  int *dom = new int[1000000], *sub = new int[1000000];
  BYTE flag,temp;
  In = new BYTE[2000000];
  CFileDialog fr(TRUE,NULL,NULL,OFN_OVERWRITEPROMPT,"Text
file|*.txt|Document file|*.doc|ALL FILES(*.*)|*.*||");
  //choose file for invert embedded zerotree
  if (fr.DoModal()==IDOK)
  {InputFileName = fr.GetFileName();}
}

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{ return;}

if((rp=fopen(InputFileName,"rb+")) == NULL)
{
    MessageBox("Can not open read file.");
    return;
}

fread(&nxny, sizeof(int),1,rp );
fread(&t0, sizeof(int),1,rp );//threshold
fread(&t2, sizeof(int),1,rp );//scan
fread(&leve, sizeof(int),1,rp );
wid =sqrt(nxny);
WIDTH=wid;
HEIGHT=wid;

for (Scan=1;Scan<=t2;Scan++)
{ fread(&x, sizeof(int),1,rp );
  fread(&y, sizeof(int),1,rp );
  bit=3;flag=3;
  for(ix=1;ix<=x;ix++)
  { if (bit==3)
    { fread(&In[v], 1,1,rp );
      ////////////////////////////////////
      //if (In[v]!=result[v])
      //{ return;
      //}
      temp=In[v++];
    }
    dom[++dompos]=temp>>((bit--)*2)&flag;
    if (bit==--1)
      bit=3;
  }
  bit=7;flag=1;
  for(iy=1;iy<=y;iy++)
  { if (bit==7)
    { fread(&In[v], 1,1,rp );
      //if (In[v]!=result[v])
      //{ return;
      //}
      temp=In[v++];
    }
    sub[++subpos]=temp>>(bit--)&flag;

    if (bit==--1)
      bit=7;
  }
}
fclose(rp);

int Level, QHeight, QWidth, Quad,Sx, Sy, X,Y,i,domin,subor,mid,SCAN;
int px, py;
int *scan;
double coef;

scan = new int[nxny];
colour = new double[nxny];
pic_decode = new double[nxny];
check = new bool[nxny];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

colour[calpos(0,0)]=1; //set first pixel has subordinate=1

MAXLEVEL =leve;

T=t0;

for (Y=0;Y<HEIGHT;Y++)
{ for (X=0;X<WIDTH;X++)
  { scan[calpos(Y,X)]=0;
    colour[calpos(Y,X)]=0;//initial all value in colour
  }
}
dompos = -1;
subpos = -1;
domin = dom[++dompos];//read next dominant

BeginWaitCursor();

for (SCAN=1;SCAN<=t2;SCAN++)
{
  subor=sub[++subpos];//read next subordinate
  mid = 3*T/2;
  if (SCAN == 1) coef = T + T/2 + ((2*subor)-1) * (T/4);
  //find approximate coefficient if SCAN=1
  else coef = colour[calpos(0,0)] + ((2*subor)-1) * (T/4);
  //find approximate coefficient in other SCAN
  colour[calpos(0,0)]=coef;//keep coef in matrix colour
  pic_decode[calpos(0,0)]=coef;
  for (i=1;i<=SCAN;i++)
  { for (Y=0;Y<HEIGHT;Y++)
    { for (X=0;X<WIDTH;X++)
      check[calpos(Y,X)] = true;
    }

    for (Level = MAXLEVEL; Level >= 1; Level--)
    { QHeight = qheight(Level);
      QWidth = qwidth(Level);
      for (Quad = 2; Quad <= 4; Quad++)
      { Sx = startx(Quad, Level);
        Sy = starty(Quad, Level);
        for (Y = Sy; Y <= Sy+QHeight-1; Y++)
          for (X = Sx; X <= Sx+QWidth-1; X++)
            { px = parentx(Y, X, Quad, Level);
              py = parenty(Y, X, Quad, Level);

              if (check[calpos(py,px)]==false) // in case Zero-Tree
                { if (colour[calpos(Y,X)] == 0)
                  { check[calpos(Y,X)]=false;//
                }

                else if (scan[calpos(Y,X)]==i)
                { subor=sub[++subpos];//read next subordinate
                  coef = ( abscon(colour[calpos(Y,X)])+(((2*subor)-1)
                    (T/4)) ) * ((colour[calpos(Y,X)]
                    /abscon(colour[calpos(Y,X)]));
                  //find approximate coefficient

                  colour[calpos(Y,X)]=coef;
                }
              }
            }
          }
        }
      }
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{ if (scan[calpos(Y,X)]==i)
  { subor=sub[+subpos]; //read next subordinate
    coef = ( abscon(colour[calpos(Y,X)])+((2*subor)-1)
      * (T/4)) ) * ((colour[calpos(Y,X)])
      /abscon(colour[calpos(Y,X)]));
    //find approximate coefficient
    colour[calpos(Y,X)]=coef;
  }
else if (scan[calpos(Y,X)]==0 && i == SCAN)
{ domin=dom[+dompos]; //read next dominant
  if (domin==ZT ||domin==Z)
  { check[calpos(Y,X)]=false;
    colour[calpos(Y,X)]=0;
  }
else if (domin==IZ) colour[calpos(Y,X)]=0;
else //in case Positive and Negative
{ subor=sub[+subpos]; //read next subordinate
  if (subor == 1) coef = 7*T/4; //if read subordinate=1
  else coef = 5*T/4; ; //if read subordinate=0
  if (domin==N) coef=-coef; //in case Negative
  colour[calpos(Y,X)]=coef;
  scan[calpos(Y,X)]=SCAN;
}
}
}
pic_decode[calpos(Y,X)]=colour[calpos(Y,X)];
} // for X
} // for Quad
} // for Level
} // for i
T=T/2;
size = nxny;
callInvwavelet();
m_prodeezw.StepIt();
} // for SCAN

EndWaitCursor();
m_prodeezw.SetPos(30);
m_prodeezw.SetPos(0);
m_rmsbutt.EnableWindow(TRUE);

CFileDialog ff(FALSE,NULL,NULL,OFN_OVERWRITEPROMPT,"Text
file|*.txt|Document file|*.doc|ALL FILES(*.*)|*.*||");
if (ff.DoModal()==IDOK)
{
  OutputFileName = ff.GetPathName();
}
else
{
  return;
}
if((xx=fopen(OutputFileName,"wb")) == NULL){
  MessageBox("Can not open write file.");
  return;
}
float x5;
for(iy=0;iy<HEIGHT;iy++)
{ for(ix=0;ix<WIDTH;ix++)
  { x5= pic_inv[calpos(iy,ix)];
    fprintf(xx,"%4.6f\n",x5);
  }
}

```

เอกสารนี้เป็นเอกสารที่พิมพ์โดยโปรแกรมอัตโนมัติ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    fprintf(xx, "\r\n");
}
//////////

free(dom);
free(sub);
free(In);
free(scan);
free(colour);
free(pic_decode);
free(check);
m_deezwbutt.EnableWindow(FALSE);
}
//*****End of Invert Embedded Zerotree*****//

void CPic1Dlg::OnComm() //communication using hyperterminal//
{
    WinExec("Hypertrm",WM_SIZE);
}

void CPic1Dlg::OnResults() //calculate Cr Bpp//
{
    FILE *il;
    float MAX1,MAX2,cr,handle,bpp;
    char msg[200];
    BITMAPFILEHEADER bmf;
    BITMAPINFOHEADER bmi;
    int width,height;
    //open dialog//
    CFileDialog f1(TRUE,NULL,NULL,NULL,"bitmap file|.bmp|Text
file|.txt|All File(*.*)|*.*||");
    CFileDialog f2(TRUE,NULL,NULL,NULL,"Text file|.txt|bitmap
file|.bmp|All File(*.*)|*.*||");
    if(f1.DoModal()==IDOK) //Open original file//
    {
        file1=f1.GetPathName();
    }
    MessageBox("Please identify the file#2.,"OPEN
FILE#2",MB_OK|MB_ICONINFORMATION);
    if(f2.DoModal()!=IDOK) return; //open compress file //
    file2=f2.GetPathName();
    if((il=fopen(file1,"rb")) == NULL)
    {
        MessageBox("Can not open file#1.");
        return;
    }
    fread( &bmf, sizeof(BITMAPFILEHEADER), 1, il);
    fread( &bmi, sizeof(BITMAPINFOHEADER), 1, il);
    width = bmi.biWidth;
    height = bmi.biHeight;
    MAX1=width*height;//number of pixels//

    handle=_open(file2,_O_RDONLY);
    if( handle == -1 )
    {
        MessageBox( "Open failed on read file." );
        return;
    }
    MAX2= _filelength(handle);//number of compress(byte) //
    fclose(handle);
}

```

เอกสารนี้เป็น close(handle); ทรัพยากรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cr=MAX1/MAX2; //calculate Cr //
bpp=(8*MAX2)/MAX1; //calculate Bpp //
sprintf(msg,"Compression Ratio = %4.4f\r\nBpp = %4.4f",cr,bpp);
MessageBox(msg);
}
//Calculate MSE RMS PSNR SNR //
void CPic1Dlg::OnRms()
{
FILE *il;
float RMS,MSE,psnr,SNR,PSNR,snr,pinv2;
char msg[200];
BYTE *tp = new BYTE[1000],*tp1, *original;
int i,j,redundant,more,count,onemore;
int width,height,pixnum;
double sum=0,diff,sumout=0,sum2=0,diff2;
BITMAPFILEHEADER bmf;
BITMAPINFOHEADER bmi;
CString filename;
//open dialog//
CFileDialog fl(TRUE,NULL,NULL,NULL,"bitmap file|*.bmp|All File
(*.*)|*.*||");

if(fl.DoModal()!=IDOK) return;//open original file//
else filename = fl.GetPathName();
il = fopen(filename,"rb+");
if (il == NULL) return;
fread( &bmf, sizeof(BITMAPFILEHEADER), 1, il);
fread( &bmi, sizeof(BITMAPINFOHEADER), 1, il);

width=bmi.biWidth;
height=bmi.biHeight;
original = new BYTE[width*height];
if (bmi.biBitCount != 8)
{ MessageBox("Not 256 color Bitmap.");
return;
}
redundant = bmi.biWidth % 4;
more=4-redundant;
fseek(il, bmf.bfOffBits, SEEK_SET);
tp1 = tp;
for (j=bmi.biHeight-1; j>=0; j--)
{ count=(int)bmi.biWidth/128;
onemore=bmi.biWidth%128;

if (count>0)
{ fread(tp,128,count,il);
tp1=tp+(count*128);
}
if (onemore>0) fread(tp1,onemore,1,il);
for (i=0;i<bmi.biWidth;i++)
original[calpos(j,i)] = *(tp+i);
if ( (j!=0) && (redundant!=0) ) fseek(il,more, SEEK_CUR);
}
fclose(il);

for (j=0; j<height; j++)
for (i=0; i<width; i++)
{ diff = (double)original[calpos(j,i)] - (pic_inv[calpos(j,i)]);
diff2 = diff*diff;
pinv2 = pic_inv[calpos(j,i)]*pic_iny[calpos(j,i)];
sum = sum+diff2;
}
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        sumout = sumout+pinv2;
    }
    pixnum = width*height;
    MSE= sum/pixnum;
    RMS= abscon(sqrt(sum)/pixnum);
    psnr= (255*255)/MSE;
    PSNR=10*log(psnr);
    snr=sumout/sum;
    SNR= 5*log(snr);
    sprintf(msg, "MSE = %5.4f\r\nPSNR = %5.4f dB\r\nSNR = %5.4f
dB\r\nRMS = %5.4f",MSE, PSNR, SNR, RMS);
    MessageBox(msg);
}
// clear Memory//
void CPic1Dlg::clearmem()
{
    clearpicmem(pic_haar);
    clearpicmem(pic_inv);
    clearpicmem(pic_decode);
    clearpicmem(data);
}
//Clear variable of picture//
void CPic1Dlg::clearpicmem(double *thispic)
{int y, x;

    if (pic != NULL)
        for (y=0; y<HEIGHT; y++)
            for (x=0; x<WIDTH; x++)
                thispic[calpos(y,x)] = 0;
}

void CPic1Dlg::OnRadbpp()
{m_threshold.EnableWindow(FALSE);
 m_bpp.EnableWindow(TRUE);
}

void CPic1Dlg::OnRadthres()
{m_bpp.EnableWindow(FALSE);
 m_threshold.EnableWindow(TRUE);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนโปรแกรม header

```

//*****VC++ create*****//
// pic1Dlg.h : header file

#if
!defined(AFX_PIC1DLG_H__93C88008_BFBA_11D4_8158_DAAF4C008638__INCLUDE
D_)
#define
AFX_PIC1DLG_H__93C88008_BFBA_11D4_8158_DAAF4C008638__INCLUDED_
#define MAXIMUM_SCALE 16383
#define BUFFER_SIZE 256

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
//*****end of VC++ create*****//
#define Z      3 // Zero for Level = 1      -T <= X => T
#define ZT     3 // ZeroTree
#define IZ     2 // IsolateZero
#define P      0 // Positive                X > T
#define N      1 // Negative
//*****VC++ create*****//
// CPic1Dlg dialog

class CPic1Dlg:public CDialog
{
// Construction
public:
//*****end of VC++ create*****//
    HBITMAP hpicture;
    CBitmap Picture;
    CStatic x;
    CPic1Dlg(CWnd* pParent = NULL); // standard constructor
    BITMAPFILEHEADER bmfh;
    BITMAPINFOHEADER bmih;
    nx=(float)bmih.biWidth,ny=(float)bmih.biHeight,w,wx,wy,k;
    double *pic;
    CDC *sh;
    CString filename;
    FILE *fp;
//*****VC++ create*****//

typedef struct {
    unsigned short int low_count;
    unsigned short int high_count;
    unsigned short int scale;
    }SYMBOL;

//function prototypes
/*Eternal variable declarations*/
long underflow_bits;
int max_order;
int flushing_enabled;
//long underflow_bits;
//*****end of VC++ create*****//

short int storage[258];
short int *totals;
CString inname,outname;
char *Cresult;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FILE *infile,*outfile;
void Encode(CString infile, CString outfile);
void Expand(void);
void initialize_output_bitstream();
void output_bit(FILE *stream,int bit);
unsigned int calpos(unsigned int y, unsigned int x);
void flushstreambit(FILE *stream);
void initialize_input_bitstream();
short int input_bit(FILE *stream);
long bit_ftell_output(FILE *stream);
long bit_ftell_input(FILE *stream);
void showim(double *im,unsigned int y, unsigned int x,int z);
void initarithencode();
void encode_symbol(FILE *stream,SYMBOL *s);
void flushcode(FILE *stream);
short int get_current_count(SYMBOL *s);
void initarithdecode(FILE *stream);
void remove_bitstream(FILE *stream,SYMBOL *s);
void initialize_model();
void update_model(int symbol);
int int2sym(int c,SYMBOL *s);
void get_symbol_scale(SYMBOL *s);
int sym2int(int count,SYMBOL *s);
char *buff;
int MAXLEVEL;
int qwidth(int level);
int qheight(int level);
int startx(int quad, int level);
int starty(int quad, int level);
void setbuff(int y, int x, int val);
void makehead(int x,int y);
int chkval(int y, int x);
int isgroupiz(int nexty, int nextx);
int find(int y, int x, int quad, int level);
int parentx(int y, int x, int quad, int level);
int parenty(int y, int x, int quad, int level);
int parent(int y,int x,int quad, int level);
double abscon(double x);
void clearmem();
void clearpicmem(double *thispic);
double T;//Threshold
CString file1, file2, OutputFileName, InputFileName;
FILE *rp, *wp, *xx, *tmpin;
void AddDlgItemText(int nID,LPCTSTR str);
bool *check, load;
float lvl;
BYTE *store, *symb;
int WIDTH,HEIGHT;
//*****VC++ create*****//
// Dialog Data
//{{AFX_DATA(CPic1Dlg)
enum { IDD = IDD_PIC1_DIALOG };
CComboBox m_bpp;
CButtonm_browsebutt;
CComboBox m_ctrllevel;
CButtonm_invhaarbutt;
CButtonm_rmsbutt;
CButtonm_resultsbutt;
CButtonm_commbutt;
CButtonm_decodebutt;
CButtonm_encodebutt;

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CButtonm_deezwbutt;
CButtonm_ezwbutt;
CButtonm_haarbutt;
CStaticm_picframe;
CComboBox m_threshold;
CProgressCtrl m_prodeezw;
CProgressCtrl m_proezw;
int m_lev;
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CPic1Dlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
public:
void callInvwavelet();

protected:
HICON m_hIcon;

// Generated message map functions
//{{AFX_MSG(CPic1Dlg)
virtual BOOL OnInitDialog();
afx_msg void OnPaint();
afx_msg HCURSOR OnQueryDragIcon();
afx_msg void OnBrowse();
afx_msg void OnHaar();
afx_msg void OnChangeEdit();
afx_msg void OnEncode();
afx_msg void OnDecode();
afx_msg void OnInvwavelet();
afx_msg void OnEzw();
afx_msg void OnLevel();
afx_msg void OnDeezw();
afx_msg void OnComm();
afx_msg void OnResults();
afx_msg void OnRms();
afx_msg void OnInvhaar();
afx_msg void OnBpp();
afx_msg void OnRadbpp();
afx_msg void OnRadthres();
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations
immediately before the previous line.

#endif //
!defined(AFX_PIC1DLG_H__93C88008_BFBA_11D4_8158_DAAF4C008638__INCLUDE
D_)
//*****end of VC++ create*****//

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาโทฉบับนี้ สำเร็จลุล่วงได้ด้วยดี ด้วยความช่วยเหลือจากบุคคลหลายๆท่าน ทั้งในด้านความรู้ ความช่วยเหลือ เครื่องมือและอุปกรณ์ต่างๆ ตลอดจนถึงการสั่งสอน ให้คำแนะนำ อีกทั้งยังคอยเป็นกำลังใจในปริญญาโทฉบับนี้ ซึ่งต้องกล่าวถึง

บิดา มารดา ที่คอยเป็นกำลังใจและสนับสนุนในเรื่องต่างๆเกี่ยวกับการทำปริญญาโทฉบับนี้ อาจารย์กฤษณ์ วรุจิระ และดร.สุทธิชัย นพนาถิพงษ์ อาจารย์ที่ปรึกษาในการทำปริญญาโทฉบับนี้ คอยให้คำแนะนำ ความช่วยเหลือ ให้คำปรึกษาในเรื่องต่างๆ และยังให้ความสนับสนุนในทุกๆด้าน

นายเศรษฐกร กาเมือง ที่ให้คำแนะนำต่างๆ ให้คำปรึกษาในเรื่องข้อมูลทางด้านทฤษฎีที่เป็นประโยชน์

นายชนินทร์ มหารักษ์ ที่ให้คำแนะนำเกี่ยวกับVisual C++ และข้อเสนอแนะในการแก้ปัญหาต่างๆ เสมอมา นายสงคราม สุนทรานู นายวรวิทย์ ธนสถิตย์ชัย กับคำแนะนำที่เป็นประโยชน์ นายวิกรม ราชวัตร น.ส.เจ จันทร์ศุกฤษณ์ ที่อนุเคราะห์เครื่องพิมพ์ นายอุษิษฐ์ หิตะวัฒนกุล นายปัฐวิกร พลอยประเสริฐ น.ส.ธีรดา จันทร์ศรี น.ส. ลัดดาวัลย์ ศรีวัฒนานุกุลกิจ น.ส.รติภักดิ์ แจ้จอรุณ นายต่อลาภ วินิจมงคลสิน นายวฤธ คูสุวรรณ นายปรัชญา บัณฑิตอาภรณ์ น.ส. ภริณี พิทยาธิคุณ นายชัยวัฒน์ รักอู่ นายภาสกร แคนยุกต์ น.ส. อังคณา โชติเวชกุล นายวรุฒม์ บุญธรรม นายสุทธิชัย แก้วแก้วดวง นายสนธิชัย กังวาลสกุลทอง นายณัฐวุฒิ จักรวาลวิบูลย์ นายเฉลิมศักดิ์ ศรีพันธุ์ นายคชาพลฤทธิ์ บุรัมย์ยากร น.ส. พัชรินี เทียนตระกูล และเพื่อนๆทุกคนที่คอยช่วยเหลือและเป็นกำลังใจให้เสมอมา

คณะผู้จัดทำ

15 มีนาคม 2544

บรรณานุกรม

- [1] ชินภัทร นันทจิวาร,รศ.ดร.มนัส ตั้งวรศิลป์,ดร.สุรพันธุ์ เอื้อไพบูลย์, “ การลดขนาดข้อมูลภาพโดยใช้การแปลงเวฟเลต,” วิศวกรรมลาดกระบัง ปีที่ 13 ฉบับที่2,หน้า 28-35,เมษายน พ.ศ. 2540
- [2] Jerome M.Shapiro. “*Embended Image coding using zerotree of wavelet coefficients.*,” IEEE Trans,signal proc.,VOL 41,NO.12.pp3445-3462,DEC 1993
- [3] Brani Vidakovic,Peter Muller., “*WAVELET FOR KIDS A Tutorial Introduction.*,” Duke University,1991
- [4] C.Sidncy Burrux,Ramesh A.Gopinath,and Maitao Guo.*Introduction to Wavelets and Wavelet Transforms.*,Prentice-Hall,1998
- [5] Charles K. Chui,*An Introduction to Wavelet.*, ACADEMIC PRESS.,1992
- [6] Eric J.Stollnitz,Tony D. DeRose,David H. Salesin., “*Wavelets for Computer Graphics:A Primer,Part1,*” IEEE Computer Grap. And App.,pp76-84.,MAY 1995
- [7] Colm Mulcahy, “*Image compression using the Haar wavelet Transform,*” Spelman Science and Math Journal.
- [8] Ghassan Al-Regib,2000, “*Embedded Zerotree Wavelet Encoding (EZW) Based on Shapiro's paper*” ,Geogia Institute of Technology Atlanta,GA,gregib@ece.gatech.edu