การบีบอัดข้อมูลภาพโดยใช้การแปลงเวฟเล็ต

IMAGE COMPRESSION USING WAVELET TRANSFORM

T032600

โดย

นายจิตติ กลิ่นประภัศร

นายธราพงษ์ เพียซ้าย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่.....
เลขทะเบียน 32600
วัน, เดือน, ปี 18 พ.ค. 2542

ลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบีบอัดข้อมูลภาพโดยใช้การแปลงเวฟเล็ต

IMAGE COMPRESSION USING WAVELET TRANSFORM

โดย

นายจิตติ กลิ่นประภัศร 39013001

นายธราพงษ์ เพียซ้าย 39013013

อาจารย์ที่ปรึกษา

ดร.สุทธิชัย นพนาคีพงษ์

อาจารย์กฤษณ์ วงรุจิระ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การบีบอัดข้อมูลภาพโดยใช้การแปลงเวฟเล็ต**

IMAGE COMPRESSION USING WAVELET TRANSFORM

ผู้จัดทำ

1.นายจิตติ กลิ่นประภัศร 39013001

2.นายธราพงษ์ เพียซ้าย 39013013

อาจารย์ที่ปรึกษา

(ดร.สุทธิชัย นพนาคีพงษ์)

อาจารย์ที่ปรึกษา

(อ.กฤษณ์ วงรุจิระ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบีบอัดข้อมูลภาพโดยใช้การแปลงเวฟเล็ต

IMAGE COMPRESSION USING WAVELET TRANSFORM

โดย นายจิตติ กลิ่นประภัศร 39013001

นายธราพงษ์ เพียซ้าย 39013013

อาจารย์ที่ปรึกษา ดร.สุทธิชัย นพนาคีพงษ์

อาจารย์กฤษณ์ วงรุจิระ

บทคัดย่อ

ปัจจุบันได้มีการศึกษาพัฒนา และวิจัย ที่เกี่ยวกับการประมวลผลภาพกันอย่างกว้างขวาง ได้มีการนำเอาหลักการและทฤษฎีต่าง ๆ มาใช้ เพื่อที่จะหาวิธีในการประมวลผลภาพให้ง่ายขึ้น ลดขั้นตอนในการประมวลผลลง และช่วยให้ได้ผลลัพธ์เป็นที่น่าพอใจ ทฤษฎีของเวฟเล็ตเป็นอีกวิธีหนึ่งที่ได้ถูกนำมาประยุกต์ใช้งานที่เกี่ยวกับการประมวลผลภาพ

ดังนั้น ปริญญานิพนธ์นี้จึงนำเสนอ การลดขนาดข้อมูลภาพโดยใช้การแปลงเวฟเล็ต ซึ่งนำเอาทฤษฎีของการแปลงเวฟเล็ตมาประมวลผลภาพ เพื่อลดรายละเอียดของภาพบางประการ แล้วนำข้อมูลที่ได้ไปลดข้อมูลแบบไม่มีการสูญเสีย เพื่อที่จะได้จำนวนบิตที่ใช้ในการแสดงภาพลดลง

ABTRACT

Today researches and developments in image processing are rapidly growing. There were many theories used in order to shortening and simple the process without losing much of the picture quality. Wavelet theory was another theory used in image processing.

This project was present a way to compress image data via wavelet transform. The wavelet transform theory was used to reduce image detail in image processing. Afterward the reduced image was encoded by lossless compression algorithm to minimize storage data.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	**1**
บทที่ 2 ทฤษฎีและหลักการเบื้องต้น	**3**
2.1 พื้นฐานของภาพเชิงตัวเลข	3
2.1.1 โมเดลภาพ	3
2.1.2 การสุ่มและการจัดระดับ	3
2.1.3 ปริมาณข้อมูลภาพ	4
2.2 ทฤษฎีพื้นฐานของการแปลงเวฟเล็ต	5
2.2.1 ลักษณะพื้นฐานของเวฟเล็ต	5
2.2.2 ทฤษฎีพื้นฐานในการสเกลและเลื่อนตำแหน่งของเวฟเล็ต	7
2.2.3 ความเป็นมาของการแปลงเวฟเล็ต	10
2.2.4 ความสัมพันธ์ระหว่างการแปลงเวฟเล็ตและการแปลงฟูริเยร์ ในลักษณะของฟังก์ชันหน้าต่าง	14
2.3 การแปลงเวฟเล็ตแบบแบ่งช่วง	15
2.3.1 ปริภูมิเวกเตอร์ในการวิเคราะห์สัญญาณ	15
2.3.2 การวิเคราะห์แบบหลายระดับความละเอียดของมอร์เลต(Mallat)	16
2.3.3 การแปลงเวฟเล็ตโดยใช้หลักการของการวิเคราะห์ของฟิลเตอร์แบงค์	22
2.4 การแปลงเวฟเล็ตกับข้อมูลภาพ	31
2.4.1 การวิเคราะห์การแปลงด้วยฮาร์ฟังก์ชัน	33
2.4.2 การใช้ซีโร่ทรีของสัมประสิทธิ์เวฟเล็ตเข้ารหัสข้อมูลภาพ	36
2.5 หลักการเบื้องต้นของการลดข้อมูล	38
2.5.1 เอนโทรปี	38
2.5.2 การลดข้อมูลแบบที่ไม่ยอมให้มีการสูญเสีย	43
2.5.3 การลดข้อมูลแบบยอมให้มีการสูญเสียได้	48
2.5.4 เกณฑ์การวัดความเหมือนจริงของภาพ	52
2.5.5 อัตราบิต	53
2.5.6 อัตราการลดข้อมูล	54
บทที่ 3 การคำนวณและการสร้าง	**55**
3.1 การคำนวณการแปลงเวฟเล็ตกับข้อมูลภาพ	55
3.2 การกำหนดรหัสสัญลักษณ์ให้กับสัมประสิทธิ์เวฟเล็ต	67
3.3 การปรับปรุงวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี	74
3.4 การเข้ารหัสเลขคณิต	75
3.5 การส่งภาพแบบโปรเกรสซีฟโดยใช้การแปลงเวฟเล็ต	83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.6 การใช้งานโปรแกรมลดขนาดข้อมูลภาพโดยการแปลงเวฟเล็ตที่ใช้วิธีการของเอ็มเบดอิดซีโร่ทรีบน MATLAB	84
บทที่ 4 การทดลองและผลการทดลอง	**90**
การทดลองที่ 4.1	91
การทดลองที่ 4.2	100
การทดลองที่ 4.3	109
การทดลองที่ 4.4	111
การทดลองที่ 4.5	113
การทดลองที่ 4.6	117
การทดลองที่ 4.7	132
การทดลองที่ 4.8	136
บทที่ 5 บทวิจารณ์และบทสรุป	**141**
5.1 บทวิจารณ์	141
5.2 บทสรุป	142
ภาคผนวก	
กิตติกรรมประกาศ	
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

		หน้า
รูปที่ 2.1	เป็นตัวอย่างของเวฟเล็ตชนิดหนึ่งเรียกว่าเวฟเล็ตแบบ morlet $g(t) = e^{\frac{-t^2}{64}} \cos(t)$	6
รูปที่ 2.2	ลักษณะคุณสมบัติของเวฟเล็ต 2 เงื่อนไข	6
รูปที่ 2.3	ลักษณะของเวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่ง	7
รูปที่ 2.4	แสดงลักษณะของการแตกกระจายสัญญาณและรวมกับสัญญาณของเวฟเล็ต	8
รูปที่ 2.5	แสดงลักษณะของสัญญาณที่มีหลายระดับความถี่ที่ตำแหน่งเวลาต่างกัน	8
รูปที่ 2.6	พิจารณาเวฟเล็ตในแกน 2 มิติ โดยที่ให้แกนหนึ่งเป็นแกนเวลา อีกแกนหนึ่งเป็นแกนความถี่	9
รูปที่ 2.7	แสดงลักษณะของผลของการ correlation ระหว่าง $f(t), g(^{t-b}/_{a})$	11
รูปที่ 2.8	แสดงลักษณะความแตกต่างของสัญญาณเอาท์พุท ระหว่าง FT และ WT	11
รูปที่ 2.9	แสดงลักษณะของระนาบเวลาความถี่แบบคงที่	13
รูปที่ 2.10	แสดงลักษณะของระนาบเวลาเทียบกับความถี่และลักษณะการแตกกระจายสัญญาณของเวฟเล็ต	14
รูปที่ 2.11	แสดงกลุ่มของปริภูมิเวกเตอร์ที่ถูกแผ่คลุมโดยสเกลลิ่งฟังก์ชัน	18
รูปที่ 2.12	แสดงปริภูมิเวกเตอร์ของสเกลลิ่งฟังก์ชันและเวฟเล็ตฟังก์ชัน	19
รูปที่ 2.13	แสดงการลดอัตราการชักตัวอย่างลงครึ่งหนึ่ง	25
รูปที่ 2.14	แสดงการวิเคราะห์สัญญาณด้วย two-band	25
รูปที่ 2.15	แสดงการแตกกระจายสัญญาณจากระดับ $j+1$ ไปยังระดับ $j-1$	26
รูปที่ 2.16	แสดงการแตกกระจายสัญญาณใน V_3 ไปสู่สเกล $j=0$	26
รูปที่ 2.17	แสดงแถบความถี่จากการผ่านการกรองของ $h_0(n)$ และ $h_1(n)$	27
รูปที่ 2.18	แสดงการชักค่าการสุ่มตัวอย่างขึ้นสองเท่า	29
รูปที่ 2.19	แสดงการสังเคราะห์ด้วย two –band	30
รูปที่ 2.20	แสดงการสังเคราะห์สัญญาณจากสเกล $j-1$ ไปสู่ระดับ j	30
รูปที่ 2.21	แสดงการแปลงเวฟเล็ตของภาพ 1 ครั้ง	31
รูปที่ 2.22	แสดงแผนผังการทำการแปลงเวฟเล็ตกับภาพ 2 มิติ 3 ครั้ง	32
รูปที่ 2.23	แสดงรูปคลื่นจากสมการและเลื่อนตำแหน่ง	33
รูปที่ 2.24	แสดงลักษณะผลรวมของสเกลลิ่งฟังก์ชันพื้นฐาน	34
รูปที่ 2.25	แสดงวิธีการแปลงแบบมาตรฐานและวิธีการแปลงแบบไม่มาตรฐาน	35
รูปที่ 2.26	แสดงลักษณะของการสเกลและลักษณะในการสืบทอดของสัมประสิทธิ์ของเวฟเล็ต	36
รูปที่ 2.27	แสดงลักษณะการสแกนแบบตัวซี	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		หน้า
รูปที่ 2.28	แสดงการควอนไทซ์ค่าสัมประสิทธิ์และกำหนดสัญลักษณ์	38
รูปที่ 2.29	แสดงตัวอย่างการเข้ารหัสฮัฟแมน	44
รูปที่ 2.30	แสดงตัวอย่างการเข้ารหัสรันเลนจ์	44
รูปที่ 2.31	แสดงการแบ่งช่วงขอบเขตปัจจุบันบนพื้นฐานของความน่าจะเป็นของสัญลักษณ์ ที่ปรากฏขึ้น	45
รูปที่ 2.32	แสดงการแบ่งขอบเขตตามข้อมูลที่เพิ่มขึ้น	47
รูปที่ 2.33	แสดงระดับของการควอนไทซ์แบบยูนิฟอร์ม	49
รูปที่ 3.1	แสดงขั้นตอนการบีบอัดข้อมูลภาพโดยใช้การแปลงเวฟเล็ต	56
รูปที่ 3.2	แสดงสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ต	67
รูปที่ 3.3	ผลการทำกลับของลำดับชุดของข้อมูล	72
รูปที่ 3.4	หน้าจอเอดิเตอร์ สำหรับพิมพ์เรียกเมนูไดอะล็อกบ็อกซ์ สำหรับการลดขนาดข้อมูลภาพด้วยการแปลงเวฟเล็ต	84
รูปที่ 3.5	แสดงตัวอย่างเมนูการใช้งานเพื่อติดต่อกับผู้ใช้งาน	85
รูปที่ 3.6	แสดงไดอะล็อกบ็อกซ์ สำหรับการใส่อินพุตภาพเพื่อทำการลดขนาดข้อมูล	86
รูปที่ 3.7	แสดงภาพเมื่อคลิกเลือกภาพจากไดอะล็อกบ็อกซ์ อินพุตภาพ	86
รูปที่ 3.8	แสดงไดอะล็อกบ็อกซ์สำหรับกรอกอัตราการลดขนาดข้อมูล	87
รูปที่ 3.9	แสดงวินโดวส์แสดงระยะเวลาของการลดขนาดข้อมูลภาพ ขึ้นมาในขณะทำการลดขนาดข้อมูล	87
รูปที่ 3.10	แสดงไดอะล็อกบ็อกซ์สำหรับบันทึกภาพลงแฟ้มข้อมูล	88
รูปที่ 3.11	แสดงภาพที่ได้ภายหลังการผ่านวิธีการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเล็ต	88
รูปที่ 3.12	แสดงผลของการลดขนาดข้อมูล	89
รูปที่ 4.1	แสดงภาพต้นแบบที่ใช้ในการทดลอง (Lena ขนาด 512×512 จุดภาพ)	91
รูปที่ 4.2	แสดงสัมประสิทธิ์ของภาพภายหลังการแปลงเวฟเล็ตโดยวิธีมาตรฐาน	92
รูปที่ 4.3	แสดงภาพหลังจากการแปลงเวฟเล็ตโดยวิธีมาตรฐาน	93
รูปที่ 4.4	แสดงภาพที่สร้างกลับขึ้นมาจากสัมประสิทธิ์เวฟเล็ตในรูปที่ 4.2 โดยวิธีมาตรฐาน	94
รูปที่ 4.5	แสดงสัมประสิทธิ์ของภาพภายหลังการแปลงเวฟเล็ตโดยวิธีการแปลงภาพแบบไม่มาตรฐาน	95
รูปที่ 4.6	แสดงภาพภายหลังการแปลงเวฟเล็ต โดยวิธีการแปลงภาพแบบไม่มาตรฐาน	96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

		หน้า
รูปที่ 4.7	แสดงภาพที่สร้างกลับขึ้นมาจากสัมประสิทธิ์เวฟเล็ตในรูปที่ 4.5 โดยวิธีการแปลงภาพแบบไม่มาตรฐาน	97
รูปที่ 4.8	แสดงกราฟ 3 มิติเปรียบเทียบค่าความแตกต่างพลาดระหว่างจุดภาพของภาพต้นแบบกับภาพที่สร้างกลับขึ้นมาจากการแปลงเวฟเล็ตแบบไม่ต่อเนื่องด้วยวิธีการแปลงแบบมาตรฐาน	98
รูปที่ 4.9	แสดงกราฟ 3 มิติเปรียบเทียบค่าความแตกต่างพลาดระหว่างจุดภาพของภาพต้นแบบกับภาพที่สร้างกลับขึ้นมาจากการแปลงเวฟเล็ตแบบไม่ต่อเนื่องด้วยวิธีการแปลงแบบไม่มาตรฐาน	99
รูปที่ 4.10	แสดงผลของการปัดค่าเทรชโฮลในสัมประสิทธิ์เวฟเล็ตของภาพในรูปที่ 4.2 ที่ค่าเทรชโฮลเท่ากับ 0.1236	100
รูปที่ 4.11	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีมาตรฐานที่ถูกปัดค่าเทรชโฮลที่ 0.1236	101
รูปที่ 4.12	กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีมาตรฐานที่ถูกปัดเทรชโฮลที่ 0.1236	102
รูปที่ 4.13	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปัดค่าเทรชโฮลที่ 0.1236	103
รูปที่ 4.14	แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปัดค่าเทรชโฮลที่ 0.1236	104
รูปที่ 4.15	กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปัดเทรชโฮลที่ 0.1236	105
รูปที่ 4.16	แสดงผลของระดับเทรชโฮลที่ระดับต่างๆ ว่ามีผลต่อค่าความผิดพลาดของภาพและมีจำนวนศูนย์ในสัมประสิทธิ์เวฟเล็ตของภาพ ที่ได้จากการแปลงแบบมาตรฐาน	106
รูปที่ 4.17	แสดงผลของระดับเทรชโฮลที่ระดับต่างๆ ว่ามีผลต่อค่าความผิดพลาดของภาพและมีจำนวนศูนย์ในสัมประสิทธิ์เวฟเล็ตของภาพ ที่ได้จากการแปลงแบบไม่มาตรฐาน	107
รูปที่ 4.18	เปรียบเทียบระหว่างรูปที่ 4.16 และรูปที่ 4.17	108
รูปที่ 4.19	แสดงผลการลดขนาดข้อมูลจากการตั้งเทรชโฮลแล้วนำมาเข้ารหัสข้อมูล	109
รูปที่ 4.20	แสดงภาพ Lena ขนาด 256×256 จุดภาพ โดยมีขนาดข้อมูลภายหลังการลดขนาดเท่ากับ 1.101 bpp (ขนาดของข้อมูลเท่ากับ 9,016 ไบต์) ค่า MSE = 193.09 และ PSNR = 18.06	110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

		หน้า
รูปที่ 4.21	กราฟแสดงผลของการลดขนาดข้อมูลด้วยวิธีการของเอ็มเบดอิดซีโร่ทรี โดยใช้สัมประสิทธิ์ของฮาร์เป็นเวฟเล็ตฟังก์ชัน	111
รูปที่ 4.22	เป็นการแสดงภาพ Lena ขนาด 256×256 จุดภาพ ที่ได้จากการแปลงกลับที่มีขนาดของข้อมูลเท่ากับ 0.317 bpp (2596 byte) และมีค่าความผิดพลาด MSE= 183.715	112
รูปที่ 4.23	กราฟเปรียบเทียบวิธีการเข้ารหัสเอ็มเบดอิดซีโร่ทรี ของเวฟเล็ตฟังก์ชันต่างๆ	113
รูปที่ 4.24	แสดงผลของภาพที่ผ่านวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี ด้วย db1 ที่มีขนาดของข้อมูลเท่ากับ 0.12 bpp และค่า ค่า MSE = 344.05	114
รูปที่ 4.25	แสดงผลของภาพที่ผ่านวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี ด้วย db2 ที่มีขนาดของข้อมูลเท่ากับ 0.12 bpp และค่า MSE = 271.09	115
รูปที่ 4.26	แสดงผลของภาพที่ผ่านวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี ด้วย db5 ที่มีขนาดของข้อมูลเป็น 0.12 bpp และค่า MSE= 270.22	116
รูปที่ 4.27	แสดงผลของภาพที่ผ่านวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี ด้วย db7 ที่มีขนาดของข้อมูลเท่ากับ 0.12 bpp และค่า MSE = 280.13	117
รูปที่ 4.28	กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการของ EZW และ JPEG (ใช้ภาพ Lena ขนาด 256×256 จุดภาพ)	118
รูปที่ 4.29	ภาพที่ได้จากการลดขนาดข้อมูลด้วยวิธี EZW ที่ขนาดภาพเท่ากับ 0.19 บิตต่อจุดภาพ และค่าความผิดพลาด (MSE) = 238.59 (ใช้ภาพ Lena ขนาด 256×256 จุดภาพ)	119
รูปที่ 4.30	ภาพที่ได้จากการลดขนาดข้อมูลด้วยวิธี JPEG ที่ขนาดภาพเท่ากับ 0.18 บิตต่อจุดภาพ และค่าความผิดพลาด (MSE) = 346.11 (ใช้ภาพ Lena ขนาด 256×256 จุดภาพ)	120
รูปที่ 4.31	กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการของ EZW และ JPEG (ใช้ภาพ Lena ขนาด 512×512 จุดภาพ)	121
รูปที่ 4.32	แสดงภาพที่ผ่านการลดขนาดข้อมูลด้วยวิธีการ EZW ที่มีขนาดของข้อมูลเป็น 0.171 บิตต่อจุดภาพ (bpp) และค่าความผิดพลาด (MSE) เท่ากับ 78.71(ใช้ภาพ Lena ขนาด 512×512 จุดภาพ)	122
รูปที่ 4.33	แสดงภาพที่ผ่านการลดขนาดข้อมูลด้วยวิธีการ JPEGที่มีขนาดของข้อมูลเป็น 0.171 บิตต่อจุดภาพ (bpp) และค่าความผิดพลาด (MSE) เท่ากับ 117.72 (ใช้ภาพ Lena ขนาด 512×512 จุดภาพ)	123
รูปที่ 4.34	แสดงภาพ Brain ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบที่นำมาลดขนาดข้อมูล	124

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

		หน้า
รูปที่ 4.35	กราฟแสดงผลของการลดขนาดข้อมูลภาพระหว่างวิธีการ EZW และ JPEG (ภาพ Brain ขนาด 512×512 จุดภาพ)	125
รูปที่ 4.36	แสดงภาพที่มีขนาด 0.1835 bpp ที่ทำการลดขนาดข้อมูลภาพโดยวิธีการของ EZW (ภาพ Brain ขนาด 512×512 จุดภาพ)	126
รูปที่ 4.37	แสดงภาพที่มีขนาด 0.184 bpp ที่ทำการลดขนาดข้อมูลภาพโดยวิธีการของ JPEG	127
รูปที่ 4.38	แสดงภาพ Xray ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบที่นำมาลดขนาดข้อมูล	128
รูปที่ 4.39	กราฟแสดงผลการลดขนาดข้อมูลภาพด้วยวิธี EZW และ JPEG	129
รูปที่ 4.40	แสดงภาพที่มีขนาด 0.119 bpp ที่ทำการลดขนาดข้อมูลโดยวิธีการ EZW มีค่าความผิดพลาด (MES) = 52.34	130
รูปที่ 4.41	แสดงภาพที่มีขนาด 0.120 bpp ที่ทำการลดขนาดข้อมูลโดยวิธีการ JPEG มีค่าความผิดพลาด (MSE) = 162.49	131
รูปที่ 4.42	กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการ EZW(A) ,EZW(B) และ JPEG	132
รูปที่ 4.43	ภาพที่ได้จากการลดขนาดข้อมูลภาพด้วยวิธี EZW ที่ปรับปรุงวิธีการเข้ารหัสใหม่ มีขนาดเท่ากับ 0.19 bpp และค่าความผิดพลาด (MSE) = 172.15 (ใช้ภาพ Lena ขนาด 256×256 จุดภาพ)	133
รูปที่ 4.44	กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการ EZW(A) ,EZW(B) และ JPEG	134
รูปที่ 4.45	ภาพที่ได้จากการลดขนาดข้อมูลภาพด้วยวิธี EZW ที่ปรับปรุงวิธีการเข้ารหัสใหม่ มีขนาดเท่ากับ 0.116 bpp และค่าความผิดพลาด (MSE) = 50.57	135
รูปที่ 4.46	แสดงผลภาพแบบโปรเกรสซีฟที่ใช้วิธีการเอ็มเบดอิดซีโร่ทรีมาเข้ารหัสข้อมูล (ภาพ Lena ขนาด 256×256 จุดภาพ)	136
รูปที่ 4.47	แสดงผลภาพแบบโปรเกรสซีฟที่ใช้วิธีการเอ็มเบดอิดซีโร่ทรีมาเข้ารหัสข้อมูล (ภาพ Brain ขนาด 256×256 จุดภาพ)	138

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 2.1	แสดงจำนวนบิตที่ใช้ในการจัดเก็บข้อมูลภาพ ที่แปรตามค่า N และ m	5
ตารางที่ 2.2	แสดงจำนวนไบต์ที่ใช้ในการจัดเก็บข้อมูลภาพ ที่แปรตามค่า N และ m	5
ตารางที่ 2.3	แสดงการแบ่งขอบเขตตามความน่าจะเป็นสะสม	47
ตารางที่ 3.1	แสดงค่าของสัมประสิทธิ์ที่มีนัยสำคัญและไม่สำคัญในเอ็มเบดอิดซีโร่ทรี	68
ตารางที่ 3.2	แสดงค่าของสัมประสิทธิ์ที่มีค่านัยสำคัญ	68
ตารางที่ 3.3	แสดงค่าของสัมประสิทธิ์ที่มีค่านัยสำคัญ	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญผังงาน

		หน้า
ผังงานที่ 3.1	แสดงผังการทำงานการแปลงเวฟเล็ต	65
ผังงานที่ 3.2	ผังการทำงานการแปลงกลับเวฟเล็ต	66
ผังงานที่ 3.3	เข้ารหัสเอ็มเบดอิดซีโร่ทรี	71
ผังงานที่ 3.4	การทำงาน EZW ส่วนที่สอง	72
ผังงานที่ 3.5	แสดงการทำกลับของเอ็มเบดอิดซีโร่ทรี	73
ผังงานที่ 3.6	แสดงการทำกลับของเอ็มเบดอิดซีโร่ทรี(ต่อ)	74
ผังงานที่ 3.7	แสดงผังงานการเข้ารหัสเลขคณิต	79
ผังงานที่ 3.8	แสดงลูปการเข้ารหัสข้อมูลสัญลักษณ์	80
ผังงานที่ 3.9	แสดงผังงานในการถอดรหัสเลขคณิต	81
ผังงานที่ 3.10	แสดงผังงานในการนำบิตข้อมูลออกจากสตรีมของข้อมูล	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญคำสั่งจำลอง

หน้า

คำสั่งจำลองที่ 3.1 การนำภาพมาแปลงเวฟเล็ตด้วยวิธีมาตรฐาน 60

คำสั่งจำลองที่ 3.2 การนำสัมประสิทธิ์เวฟเล็ตของภาพที่ได้จากการแปลงเวฟเล็ตแบบมาตรฐานมาแปลงกลับ 61

คำสั่งจำลองที่ 3.3 การนำภาพมาแปลงเวฟเล็ตด้วยวิธีไม่มาตรฐาน 62

คำสั่งจำลองที่ 3.4 การนำสัมประสิทธิ์เวฟเล็ตของภาพที่ได้จากการแปลงเวฟเล็ตแบบไม่มาตรฐานมาแปลงกลับ 63

คำสั่งจำลองที่ 3.5 แสดงโปรแกรมลดขนาดข้อมูลภาพโดยการปัดค่าเทรชโฮลของสัมประสิทธิ์ภาพในการแปลงเวฟเล็ต 64

คำสั่งจำลองที่ 3.6 แสดงการเข้ารหัสเลขคณิตตามวิธีการของ วิทเทน, เนล, และเคลียรี่ 77

คำสั่งจำลองที่ 3.7 แสดงการถอดรหัสเลขคณิตตามวิธีการของ วิทเทน, เนล, และเคลียรี่ 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การลดขนาดข้อมูล (data compression) เป็นหัวใจสำคัญประการหนึ่งในระบบสื่อสาร เพื่อให้สามารถส่งข้อมูลจำนวนมากไปในช่องทางสื่อสารที่จำกัด และให้ถึงจุดหมายปลายทางในเวลาที่เหมาะสม การบีบอัดข้อมูลภาพ (image compression) เป็นหัวข้อหนึ่งในการลดขนาดข้อมูล โดยเป็นวิธีการลดปริมาณข้อมูล ที่ยังคงข่าวสาร (information) ที่กำลังสนใจไว้

ภาพเชิงตัวเลข (digital image) จะประกอบด้วยชุดของบิตข้อมูล ตัวอย่างเช่น ภาพระดับสีเทา (gray scale) ที่ประกอบด้วยอาร์เรย์ของข้อมูลที่ได้มาจากการสุ่มขนาด 256×256 ซึ่งในแต่ละการสุ่มจะประกอบด้วยค่าความเข้มจำนวน 8 บิต ดังนั้นจึงต้องการเนื้อที่สำหรับทำการเก็บข้อมูลจำนวน 254,288 บิต หรือ 65,536 ไบต์ ข้อมูลการสุ่มนี้เราเรียกง่ายว่าเป็นจุดภาพ (pixel) ซึ่งกรณีที่เป็นภาพสี ที่มีขนาดเดียวกันนี้ก็จะใช้หน่วยความจำในการจัดเก็บข้อมูลเพิ่มขึ้นเป็น 3 เท่า เนื่องจากความต้องการในการส่งข้อมูลภาพจะเป็นสัดส่วนกับจำนวนของบิตข้อมูล โดยในการส่งภาพเชิงตัวเลขตามปกตินั้นจะต้องใช้ความกว้างแถบ (bandwidth) ที่กว้างและใช้ระยะเวลานาน ยกตัวอย่างการส่งภาพจำนวน 524,288 บิต จะใช้เวลา 54.6 วินาที สำหรับการส่งข้อมูลผ่านโมเด็มที่ความเร็วในการส่ง 9,600 บิตต่อวินาที ซึ่งภาพเชิงตัวเลขอื่น ๆ อาจจะมีขนาดใหญ่กว่านี้ เช่น ภาพที่ได้จากรังสีเอ็กซ์ขนาด 2048×2048 จุดภาพ ที่ค่าความเข้ม 12 บิต ก็จะใช้เวลา 87 นาทีสำหรับการส่งที่ความเร็ว 9,600 บิตต่อวินาที

ค่าใช้จ่ายจะมีความสัมพันธ์กับการเก็บลงหน่วยความจำและการส่งข้อมูลภาพ จึงทำให้การลดข้อมูลมีความสำคัญ ในระบบสื่อสารข้อมูล ถ้ามีการใช้จำนวนข้อมูลมากเกินกว่าที่ควรจะเป็นสำหรับแทนข่าวสารทางคอมพิวเตอร์จะถือว่าเกิด ข้อมูลส่วนเกิน (data redundancy) โดยกระบวนการบีบอัดข้อมูลต้องทำการลดหรือกำจัดข้อมูลส่วนเกินเหล่านี้ออกไป การลดข้อมูลภาพเป็นกระบวนการลดจำนวนของบิตที่จะใช้ในการแสดงภาพ การลดข้อมูลภาพจะคำนึงถึงข้อมูลทางสถิติของภาพร่วมกับความเหมือนจริงที่ยอมรับได้ด้วยการสังเกตด้วยตาของมนุษย์ ประสิทธิภาพในการบีบอัดข้อมูลสามารถที่จะแสดงเป็นอัตราส่วนการบีบอัด (compression ratio) ซึ่งมีค่าเท่ากับ จำนวนบิตที่ใช้ในภาพต่อจำนวนพิกเซลในภาพ (บิตต่อพิกเซล) การลดข้อมูลยังสามารถออกแบบให้มีการส่งแบบโปรเกรสซีฟ (progressive) ซึ่งสามารถนำไปใช้ประโยชน์ได้เมื่อผู้สังเกตต้องการดูภาพที่มีรายละเอียดต่ำเร็วที่สุดเท่าที่จะเป็นไปได้เพื่อที่จะสามารถทำนายภาพที่จะถูกส่งต่อมาได้ เช่น การแสดงภาพในการสื่อสารอินเตอร์เน็ต

เทคนิคในการลดข้อมูลภาพสามารถแบ่งออกได้เป็น 2 กลุ่ม คือการลดข้อมูลแบบที่ไม่มีการสูญเสีย (lossless compression) และการลดที่มีการสูญเสีย (lossy compression) การลดข้อมูลแบบไม่มีการสูญเสียจะสามารถนำข้อมูลกลับมาได้ใหม่อย่างสมบูรณ์ บางครั้งเรียกว่า การเข้ารหัสแบบเอนโทรปี (entropy coding) การลดข้อมูลโดยปราศจากการสูญเสียนี้มีความจำเป็นสำหรับข้อมูลบางชนิด เช่น ข้อมูลทางการแพทย์และข้อมูลโปรแกรมคอมพิวเตอร์ เพราะถ้าแม้เพียงบิตเดียวของข้อมูลมีการเปลี่ยนแปลงไปในระหว่างกระบวนการลดข้อมูล ก็สามารถทำให้ข้อมูลเหล่านี้เกิดความเสียหายขึ้นอย่างมากได้ การลดข้อมูลประเภทนี้มีอัตราการลดข้อมูลที่ต่ำ ดังนั้นเพื่อให้ได้อัตราการลดข้อมูลที่สูงขึ้น จึงจำเป็นที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ต้องใช้การลดข้อมูลแบบที่ยอมให้มีการสูญเสียเกิดขึ้น การลดข้อมูลแบบที่มีการสูญเสียนี้จะทำให้เกิดความเพี้ยน (distortion) กับข้อมูลที่ได้ถูกสร้างกลับขึ้นมาใหม่ หัวใจในการออกแบบระบบการลดข้อมูลที่มีการสูญเสีย คือ การทำให้มีค่าความเพี้ยนเฉลี่ยต่ำที่สุดสำหรับอัตราบิต (bit rate) หรือในอัตราการลดข้อมูล (compression ratio) หนึ่งๆ นอกจากสองกลุ่มดังกล่าวข้างต้นแล้ว ในขบวนการของการลดขนาดข้อมูลอื่นๆ สามารถที่จะนำวิธีการของการลดขนาดข้อมูลแบบที่ยอมให้มีการสูญเสียและไม่มีการสูญเสียมารวมกันได้ โดยจะเรียกว่าเป็นการลดข้อมูลแบบผสม (hybrid compression) โดยในปริญญานิพนธ์นี้จะใช้วิธีการลดข้อมูลแบบผสม ซึ่งจะนำภาพมาแยกเป็นองค์ประกอบย่อยๆ ตามความถี่ของภาพที่ตำแหน่งต่างๆ บนภาพ โดยอาศัยการแปลงเวฟเล็ต (wavelet transform) โดยสัมประสิทธิ์ที่ได้หลังจากการแปลงเวฟเล็ตจะเป็นสัมประสิทธิ์ทางความถี่ นำค่าสัมประสิทธิ์ที่ได้มาผ่านวิธีการจัดระดับของสัมประสิทธิ์และจัดกลุ่มของข้อมูล แล้วนำข้อมูลที่ได้มาผ่านการเข้ารหัสแบบไม่มีการสูญเสีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการเบื้องต้น

2.1 พื้นฐานของภาพเชิงตัวเลข

2.1.1 โมเดลภาพ (Image model)

ภาพเป็นข้อมูลสองมิติ (two dimensions) สามารถเขียนให้อยู่ในรูปของฟังก์ชันทางคณิตศาสตร์ได้ด้วย $f(x,y)$ โดยที่ x และ y แสดงตำแหน่งของฟังก์ชันตามแนวแกนนอน และแนวแกนตั้งตามลำดับ ค่าของฟังก์ชัน $f(x,y)$ แสดงถึงความเข้มแสง (intensity) ในสปาเชียลโคออร์ดิเนตที่จุดนั้น ภาพที่มองเห็นโดยทั่วไปประกอบขึ้นด้วยแสงสะท้อนจากวัตถุ ตามธรรมชาติแล้วจะถูกกำหนดคุณสมบัติจากองค์ประกอบ 2 ประการ ประการแรกได้แก่ปริมาณของแสงที่ได้มาจากแหล่งกำเนิดมายังวัตถุที่กำลังมอง ซึ่งเรียกองค์ประกอบนี้ว่า ความสว่าง (illumination) แทนด้วย $i(x,y)$ อีกประการหนึ่งคือปริมาณแสงสะท้อนออกจากวัตถุ ซึ่งจะเรียกว่า การสะท้อน (reflection) แทนด้วย $r(x,y)$ โดยผลคูณของฟังก์ชัน $i(x,y)$ และ $r(x,y)$ จะได้เป็น $f(x,y)$ ดังสมการ (2.1)

$$f(x,y) = i(x,y) \times r(x,y) \tag{2.1}$$

โดยเมื่อ

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

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 $f(x,y)$ ต้องทำให้เป็นดิจิตอล ทั้งทางสปาเชียล และแอมปลิจูด การทำให้เป็นดิจิตอลของทาง สปาเชียลโคออร์ดิเนต (x,y) จะเรียกว่าการสุ่มตัวอย่างภาพ (image sampling) ในขณะที่การทำให้เป็นจำนวนเชิงเลขในแง่ของแอมปลิจูด จะเรียกว่าการควอนไทซ์

ภาพต่อเนื่อง $f(x,y)$ จะได้รับการประมาณค่าโดยการสุ่มตัวอย่าง แล้วนำมาเรียงกันในรูปแบบของ N×N อาร์เรย์ดังแสดงในสมการ 2.3 เมื่อแต่ละอิลิเมนต์ (element) ในอาร์เรย์เป็นจำนวนเต็มหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ทางด้านขวาของสมการ (2.3) จะเรียกว่าภาพเชิงตัวเลข หรือ ภาพดิจิตอล (digital image) ในขณะที่แต่ละอิลิเมนต์ของอาร์เรย์จะเรียกว่า องค์ประกอบของภาพ (image element, picture element หรือ pixel)

2.1.3 ปริมาณข้อมูลภาพ

การแปลงภาพจากสัญญาณต่อเนื่อง (analog) ให้เป็นสัญญาณดิจิตอลที่อยู่ในลักษณะของสัญญาณที่ไม่ต่อเนื่องหรือสัญญาณดิสครีต (discrete signal) ซึ่งจะต้องผ่านการสุ่มและจัดระดับ(sample and quantized) โดยในการสุ่มจะทำการสุ่มในแนวนอนจำนวน N ครั้ง และแนวตั้งจำนวน N ครั้ง เนื่องจากภาพมีลักษณะเป็น 2 มิติ ทำให้ได้ข้อมูลภาพเป็น $N \times N$ ข้อมูล ดังที่กล่าวมาแล้วในหัวข้อที่ 2.1.2 โดยในทางปฏิบัติไม่จำเป็นต้องเป็นเมตริกซ์ขนาด $N \times N$ เสมอไป อาจเป็น $N \times M$ ใด ๆ ก็ได้

ในการแปลงภาพจากอนาล็อกให้เป็นภาพดิจิตอล นอกจากจะพิจารณาจำนวนของการสุ่มแล้วยังต้องพิจารณาถึงจำนวนระดับสีที่มีได้ในแต่ละจุดของภาพ เช่นระดับสีเทา (gray scale)

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

และ

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

โดยที่ G เป็นจำนวนของระดับสีเทา โดยในหนึ่งจุดภาพหรือพิกเซลแทนด้วยความยาว m บิต

ดังนั้นเนื้อที่ในการเก็บภาพดิจิตอลทั้งหมดแสดงดังในสมการ (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 แสดงจำนวนไบต์กับค่าของ 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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	32768
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 แสดงจำนวนไบต์ที่ใช้ในการจัดเก็บข้อมูลภาพ ที่แปรตามค่า 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 การพิจารณาว่าภาพนั้น ๆ ดี หรือไม่ดี เป็นการยากที่จะนิยามเพราะว่าคุณภาพของภาพที่ต้องการนั้น ขึ้นอยู่กับงานที่จะนำไปใช้

2.2 ทฤษฎีพื้นฐานของการแปลงเวฟเล็ต

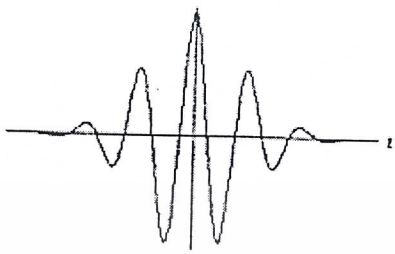
2.2.1 ลักษณะพื้นฐานของเวฟเล็ต

ทฤษฎีเวฟเล็ต (Wavelet Theory) เป็นคณิตศาสตร์ที่ใช้อธิบายการสร้างโมเดลของสัญญาณระบบหรือ กระบวนการทางฟิสิกส์ใด ๆ ซึ่งประกอบขึ้นด้วยเซตของสัญญาณเฉพาะหรือระบบย่อย ๆ มารวมกันเป็นสัญญาณหรือระบบนั้น ๆ สัญญาณเฉพาะนี้จะเป็นคลื่นเล็ก ๆ ซึ่งถูกเรียกว่า “เวฟเล็ต” (Wavelets) ลักษณะของเวฟเล็ตจะเป็นคลื่นที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง (oscillatary) และขนาดของคลื่นจะลดลงสู่ศูนย์อย่างรวดเร็วทั้งสองด้าน ดังแสดงในรูปที่ 2.1

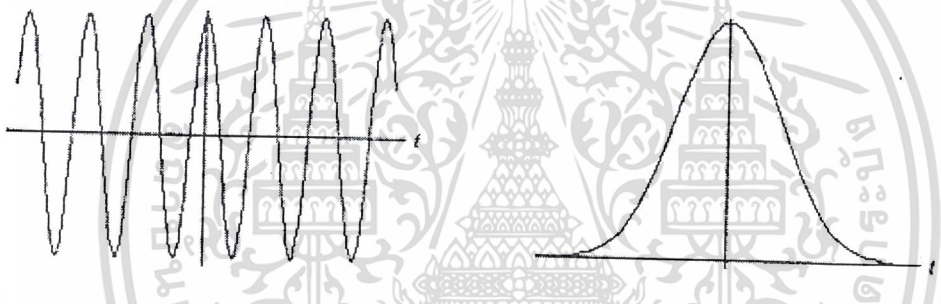
ในรูปที่ 2.1 เป็นตัวอย่างของเวฟเล็ตชนิดหนึ่งเรียกว่าเวฟเล็ตแบบมอร์เล็ต (Morlet) สังเกตว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีคุณสมบัติของเวฟเล็ตตามที่กล่าวมาทั้ง 2 เงื่อนไข คือ มีการออสซิเลตในลักษณะของสัญญาณรูปไซน์ (sinusoidal) ดังแสดงในรูปที่ 2.2 ก.) รวมกับเงื่อนไขของการลดลงอย่างรวดเร็วตามลักษณะฟังก์ชันหน้าต่าง (Windows Function) ดังแสดงในรูปที่ 2.2 ข)



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



ก) ลักษณะของสัญญาณซายน์ ; $\cos(t)$ ข) ลักษณะของฟังก์ชันหน้าต่าง ; $e^{\frac{-t^2}{64}}$

รูปที่ 2.2 แสดงลักษณะคุณสมบัติของเวฟเล็ต 2 เงื่อนไข

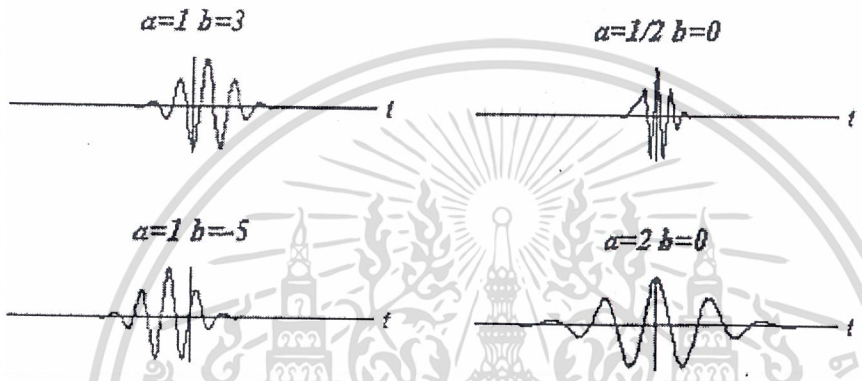
โดยเมื่อเกิดทั้งสองเงื่อนไขนี้พร้อมกัน ก็จะทำให้เกิดคลื่นในลักษณะเล็กๆ ที่เรียกว่า "เวฟเล็ต" โดยจากรูปที่ 2.1 จะเกิดจากการคูณกันของฟังก์ชันออสซิเลต(oscillate function) และฟังก์ชันการลดลง (decay function) ดังแสดงในรูปที่ 2.2 ก) และ ข) ตามลำดับ

การนำเวฟเล็ตหลาย ๆ อันมารวมกันเป็นเซต เพื่อจะใช้ในการอธิบายสัญญาณใด ๆ โดยที่คลื่นเวฟเล็ตแต่ละตัวจะมีโครงสร้างมาจากฟังก์ชันเดียวกัน ซึ่งฟังก์ชันนี้จะเป็นเวฟเล็ตต้นกำเนิดที่เรียกกว่าเวฟเล็ตแม่ (Mother Wavelets) คลื่นเวฟเล็ตแต่ละอันจะอยู่ภายในเซตของเวฟเล็ตนี้ โดยคลื่นแต่ละคลื่นจะเกิดจาก "การสเกล" (scaling : "a") และ "การเลื่อนตำแหน่ง" (translation or shifting : "b") ของเวฟเล็ตแม่ตัวนี้ การสเกลจะหมายถึง การหดเข้า (compressing) หรือ การขยายออก (dilation) ในที่นี้จะใช้ "a" แทนพารามิเตอร์ของการหดหรือขยาย ซึ่งจะเป็นการเปลี่ยนความถี่ของเวฟเล็ตแม่นั่นเอง ส่วนการเลื่อนตำแหน่งกำหนดโดยพารามิเตอร์ "b" จะหมายถึงการเลื่อนตำแหน่งการเกิดของคลื่นเวฟเล็ตบนแกนเวลา ในรูปที่ 2.3 จะแสดงให้เห็นถึงเวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่งไปที่ค่า "a" "b" ต่าง ๆ กัน โดยที่ยังคงอยู่ภายในเซตของเวฟเล็ตที่สัมพันธ์กับเวฟเล็ตแม่ ในรูปที่ 2.1 ซึ่งจะสังเกตเห็นว่า คลื่นเวฟเล็ตแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละคลื่นจะมีรูปร่างเหมือนเดิม โดยยังคงมีจำนวนไซเคิลของการออสซิเลตเท่าเดิม จะเปลี่ยนไปเฉพาะความถี่และตำแหน่งของการเกิดเท่านั้น นอกจากนี้เวฟเล็ตที่ถูกสเกลจะต้องทำการนอร์มอลไลซ์ (normalize) ด้วย $1/\sqrt{a}$ เสมอ เพื่อให้เวฟเล็ตที่ถูกสเกลไปแล้วยังคงมีพลังงานเช่นเดียวกับเวฟเล็ตแม่ ดังนั้นถ้าให้ $g(t)$ เป็น ฟังก์ชันเวฟเล็ตแม่ จะสามารถเขียนเป็นสมการทั่วไปของเวฟเล็ตที่ตำแหน่ง "a", "b" ใด ๆ ซึ่งสัมพันธ์กับเวฟเล็ตแม่ได้ดังนี้

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



รูปที่ 2.3 **ลักษณะของเวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่ง**

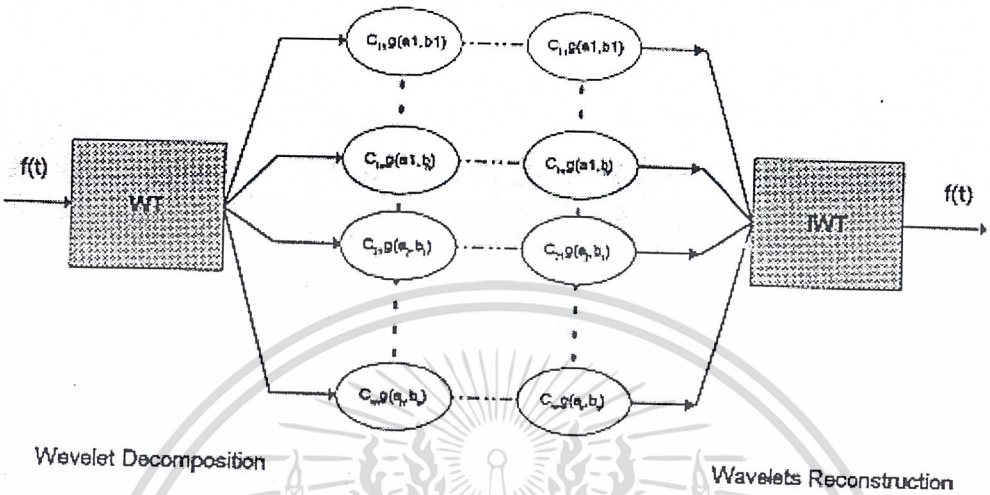
2.2.2 ทฤษฎีพื้นฐานในการสเกล และเลื่อนตำแหน่งของเวฟเล็ต

หากสามารถอธิบายลักษณะของสิ่งใดสิ่งหนึ่งได้จากลักษณะของสมาชิกเล็ก ๆ ที่เป็นองค์ประกอบแล้ว ทฤษฎีเวฟเล็ตก็เป็นอีกวิธีหนึ่งที่จะใช้ในการอธิบายลักษณะของสิ่งต่าง ๆ หรือระบบใด ๆ ได้เช่นเดียวกัน ในการที่จะใช้สมาชิกมาอธิบายสิ่งเหล่านี้ ก็จะมีวิธีการหาค่าสมาชิกเหล่านี้หลายวิธี การเลือกที่จะใช้วิธีใดก็ขึ้นอยู่กับว่าจะนำไปประยุกต์ใช้กับงานนั้นแล้วได้ประสิทธิภาพดีเพียงใด เช่น การใช้ระบบเลขฐานสิบจะใช้ในระบบบัญชีได้ เหมาะสมกว่าระบบเลขฐานสองหรือฐานสิบหก ในขณะเดียวกันระบบเลขฐานสองจะใช้อธิบายในระบบคอมพิวเตอร์ได้ดีกว่า ดังนั้นการใช้ทฤษฎีเวฟเล็ตให้ได้ดีก็จะขึ้นอยู่กับงานที่จะนำไปประยุกต์ใช้ด้วย

ทฤษฎีเวฟเล็ตจะใช้ในการอธิบายสิ่งใดสิ่งหนึ่ง เสมือนการแตกสิ่งเหล่านั้นออกเป็นส่วนประกอบเล็กๆ ที่สัมพันธ์กัน จะอยู่ในรูปของเวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่งและมีค่าถ่วงน้ำหนักต่างๆกันคูณอยู่กับเวฟเล็ตแต่ละตัว ดังนั้นจึงเปรียบเสมือนว่าสัญญาณใด ๆ จะสามารถสร้างขึ้นมาได้โดยมีคลื่นเวฟเล็ตเป็นฟังก์ชันพื้นฐาน (basis function : b.f.) หรือองค์ประกอบพื้นฐานกระบวนการแตกกระจายชิ้นส่วนนี้เรียกว่า "การแตกกระจายเวฟเล็ต" (Wavelet deconstruction) ซึ่งก็คือ การทำการแปลงเวฟเล็ต(Wavelet Transform: WT) นั่นเอง ในทำนองเดียวกัน การรวมกลับเวฟเล็ต (Wavelet reconstruction) ซึ่งเป็นการแปลงกลับเวฟเล็ต (Inverse Wavelet Transform : IWT) ก็จะเป็นการนำส่วน

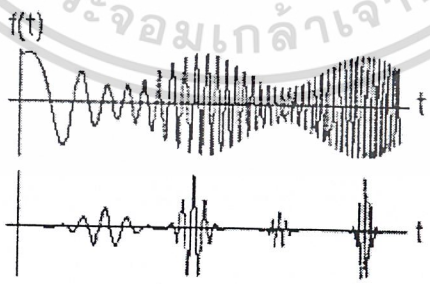
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบย่อยๆเหล่านี้มาเรียงต่อกันเพื่อประกอบเป็นสัญญาณเดิม ดังแสดงในรูปที่ 2.4 เป็นการแตกกระจายสัญญาณ $f(t)$ ออกเป็นคลื่นเวฟเล็ตที่ความถี่และตำแหน่งต่างกันจำนวน $N \times N$ ตัว



รูปที่ 2.4 แสดงลักษณะของการแตกกระจายสัญญาณและรวมกลับสัญญาณของเวฟเล็ต

ถ้าให้ $f(t)$ เป็นสัญญาณที่ประกอบด้วยคลื่นหลายความถี่ที่เกิดขึ้นบนตำแหน่งต่างๆบนแกนเวลา ดังแสดงในรูปที่ 2.4 ส่วนในภาพด้านล่างจะแสดงถึงการดึงสัญญาณ $f(t)$ ที่ตำแหน่งเวลาใด ๆ ออกมา



รูปที่ 2.5 แสดงลักษณะของสัญญาณที่มีหลายความถี่ที่ตำแหน่งเวลาต่างกัน

ถ้าทำการพิจารณาในแกน 2 มิติ โดยที่ให้แกนหนึ่งเป็นแกนเวลา ในอีกแกนหนึ่งเป็นแกนความถี่ จะสามารถทำการแยกสัญญาณที่ตำแหน่งเวลาและมีความถี่ใดๆ ได้แสดงในรูปที่ 2.6(ก) และในรูปที่ 2.6(ข) จะพิจารณาที่ค่าเวลาและค่าความถี่มีความต่อเนื่อง ซึ่งก็จะได้เป็นสัญญาณพื้นผิวที่ต่อเนื่อง

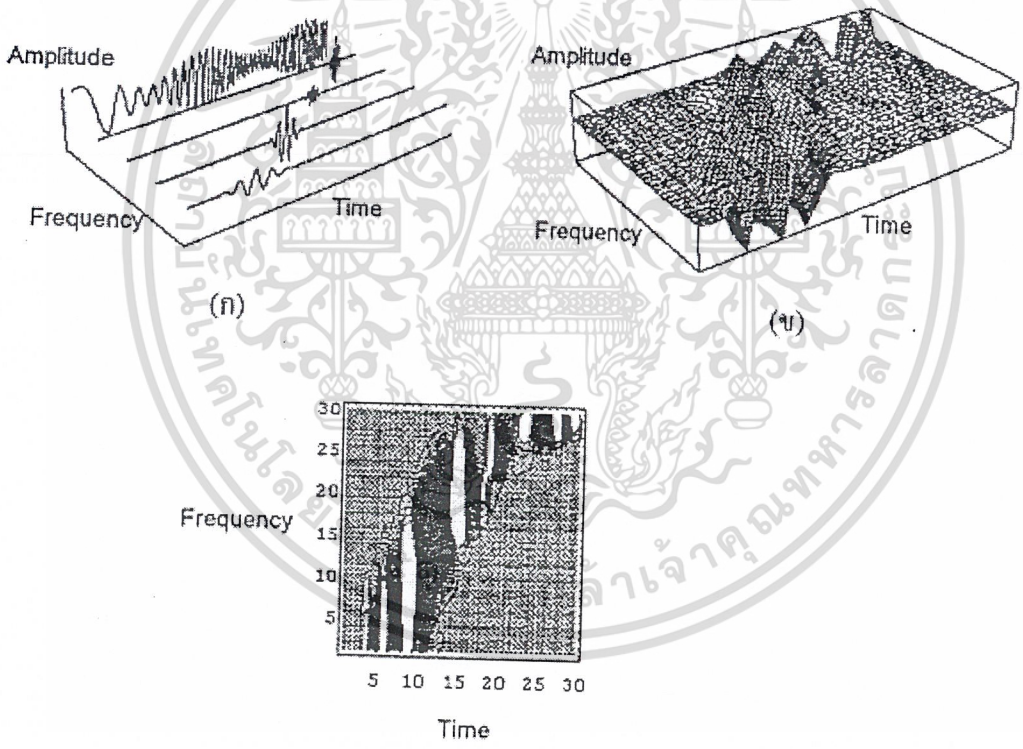
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากขึ้น ในรูปที่ 2.6(ค) จะเป็นการพิจารณามองจากด้านบนของรูปที่ 2.6(ข) ซึ่งจะเห็นว่าความเข้มจะแทนขนาดของสัญญาณ

การสเกลและการเลื่อนตำแหน่งจะสามารถเขียนเป็น สมการคณิตศาสตร์ในลักษณะการส่ง (mapping) ตัวแปรอิสระตัวใหม่ ดังแสดงในสมการที่ (2.8)

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

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



ก) แสดงการพิจารณาสัญญาณในลักษณะแบบ 2 มิติแบบแบ่งช่วง

ข) แสดงการพิจารณาสัญญาณในลักษณะแบบ 2 มิติแบบต่อเนื่อง

ค) แสดงลักษณะของสัญญาณเมื่อพิจารณาจากด้านบน

รูปที่ 2.6 พิจารณาเวฟเล็ตในแกน 2 มิติ โดยที่ให้แกนหนึ่งเป็นแกนเวลา อีกแกนหนึ่งเป็นแกนความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

โดย "*" หมายถึง คอมเพล็กคอนจูเกต (Complex Conjugate) "a" เป็นพารามิเตอร์ของการสเกล และ "b" จะเป็นพารามิเตอร์ของการเลื่อนตำแน่ง 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 Correlation) ระหว่างเวฟเล็ตแม่ที่ถูกสเกล (a) กับสัญญาณอินพุตที่ตำแหน่งเวลานั้น (b) ดังแสดงในรูปที่ 2.7

พิจารณารูปที่ 2.7 ก) เปรียบเทียบกับรูปที่ 2.7 ข) จะเห็นว่าในรูป ก) คลื่นเวฟเล็ตจะมีความถี่และเฟสใกล้เคียงกับสัญญาณอินพุตมาก ทำให้ปรากฏคลื่นเอาต์พุตออกมาที่ค่าสูง แต่ในภาพ ข) คลื่นอินพุตจะมีค่าความถี่สูงกว่าคลื่นเวฟเล็ตทำให้เอาต์พุตมีค่าต่ำกว่า

สามารถสรุปได้ว่า ถ้าคลื่นเวฟเล็ตมีความคล้ายกับสัญญาณอินพุตมากเท่าไรทั้งในด้านของความถี่ ลักษณะสัญญาณและตำแหน่งเวลาในการเปรียบเทียบ ก็จะทำให้ได้ค่าของสัมประสิทธิ์ของการทรานส์ฟอร์มที่มากนั้นเอง เพราะฉะนั้น การทำ การแปลงเวฟเล็ต จะเป็นการอธิบายสัญญาณ $f(t)$ ในรูปแบบของเวฟเล็ตโดเมน (Wavelet Domain) เทียบกับคลื่นเวฟเล็ตที่ค่า a, b ใด ๆ

ในรูปที่ 2.8 จะแสดงการเปรียบเทียบลักษณะผลลัพธ์เมื่อนำสัญญาณอินพุต 1 มิติ มาทำการแปลงฟูริเยร์ (Fourier Transform : FT) กับการใช้ การแปลงเวฟเล็ต จะเห็นว่า การแปลงฟูริเยร์ เป็นการเปลี่ยนสัญญาณจากในโดเมนเวลา (1มิติ) ไปเป็นโดเมนความถี่ (1มิติ) แต่ การแปลงเวฟเล็ต จะเป็นการเปลี่ยนไปเป็น "ระนาบเวลาและความถี่" (time-frequency plane) ซึ่งมี 2 มิติ

2.2.3 ความเป็นมาของการแปลงเวฟเล็ต

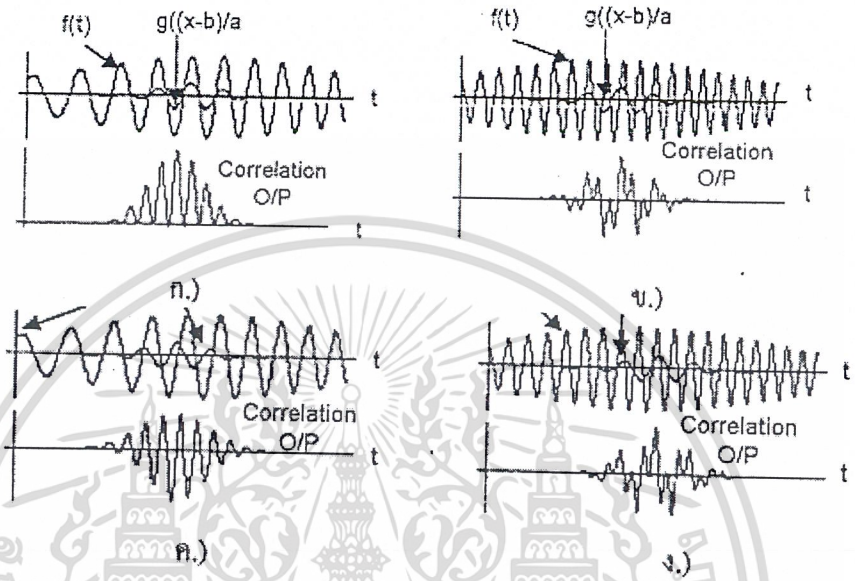
การแปลงฟูริเยร์ เป็นการทรานส์ฟอร์มที่นิยมใช้กันมาก ซึ่งเป็นการแปลงสัญญาณ $f(t)$ จากโดเมนเวลาไปเป็นสัญญาณ $F(\omega)$ ซึ่งอยู่ในโดเมนความถี่ หรือที่เรียกกันทั่วไปว่าเป็นการหาสเปกตรัมของสัญญาณ $f(t)$ นั่นเอง โดยมีสมการเป็น

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

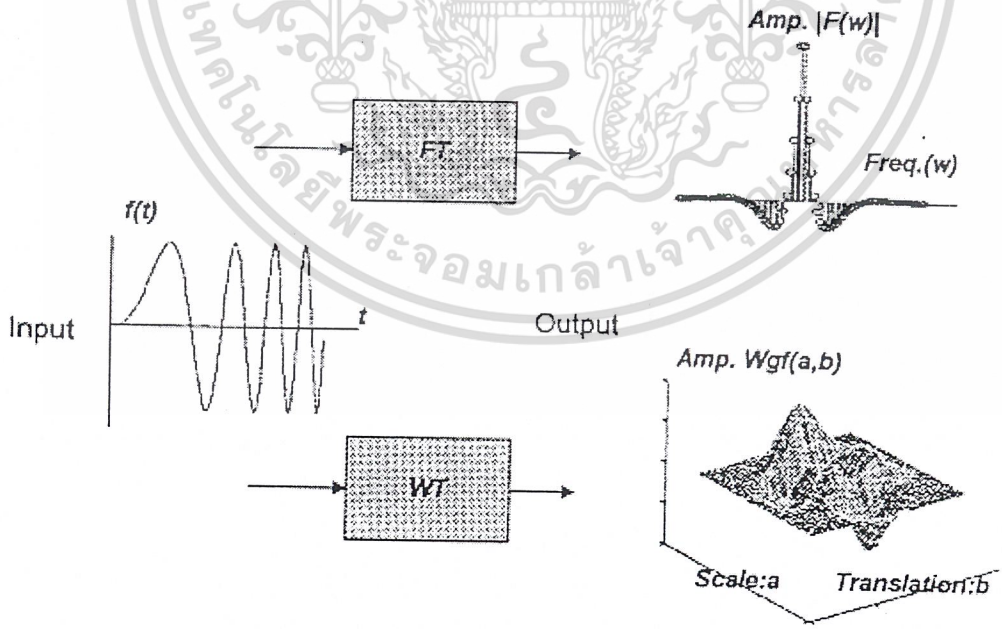
จะสังเกตพบว่าข้อมูลที่ได้จากการทรานส์ฟอร์มจะมีตลอดย่านความถี่ $(-\infty, \infty)$ ดังนั้นเมื่อเกิดการเปลี่ยนแปลงของสัญญาณอินพุตที่ขณะเวลาใด ๆ ก็จะส่งผลกระทบไปตลอดย่านของสเปกตรัม $F(\omega)$ ด้วย นอกจากนี้การใช้ การแปลงฟูริเยร์ ยังไม่สามารถเลือกทำการวิเคราะห์สัญญาณเฉพาะช่วงเวลา หรือ เฉพาะช่วงความถี่ใด ๆ ซึ่งบางครั้งต้องการวิเคราะห์สัญญาณเพียงแค่เฉพาะช่วงเวลาหรือ ช่วงความถี่เท่านั้น ดังนั้น การแปลงฟูริเยร์ จึงให้ข้อมูลที่มากเกินความจำเป็น นอกจากนั้นในทางปฏิบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณอินพุตจะมีการเปลี่ยนแปลงที่ไม่คงที่ (non-stationary signal) ซึ่งถ้าใช้ การแปลงฟูริเยร์ จะทำให้สิ้นเปลืองเวลาในการคำนวณ เนื่องจากมีข้อมูลมากเกินไปในการคำนวณใหม่แต่ละครั้ง ดังนั้น การแปลงฟูริเยร์ จึงเหมาะสำหรับสัญญาณที่มีการเปลี่ยนแปลงที่คงที่ (stationary signal)



รูปที่ 2.7 แสดงลักษณะของผลของการสหสัมพันธ์ (correlation) ระหว่าง $f(t), g(^{(t-b)}/_{a})$



รูปที่ 2.8 แสดงลักษณะความแตกต่างของสัญญาณเอาต์พุตระหว่าง
การแปลงฟูริเยร์ กับ การแปลงเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงกาบอร์ (Gabor Transform : GT) เป็นทรานส์ฟอร์มที่ได้พัฒนาขึ้นเป็นแบบแรก เพื่อแก้ปัญหาที่กล่าวมาข้างต้น โดยจะเป็นการทำ การแปลงฟูริเยร์ เฉพาะช่วงเวลา และ ช่วงความถี่ที่กำหนดผ่านฟังก์ชันหน้าต่าง (window function :w.f.) ซึ่งฟังก์ชันหน้าต่างนี้จะใช้เกาส์เซียนฟังก์ชัน (gaussian function) เท่านั้น

$$g_\alpha(t) = \frac{1}{2\sqrt{\pi\alpha}} e^{\frac{-t^2}{4\alpha}} \tag{2.11}$$

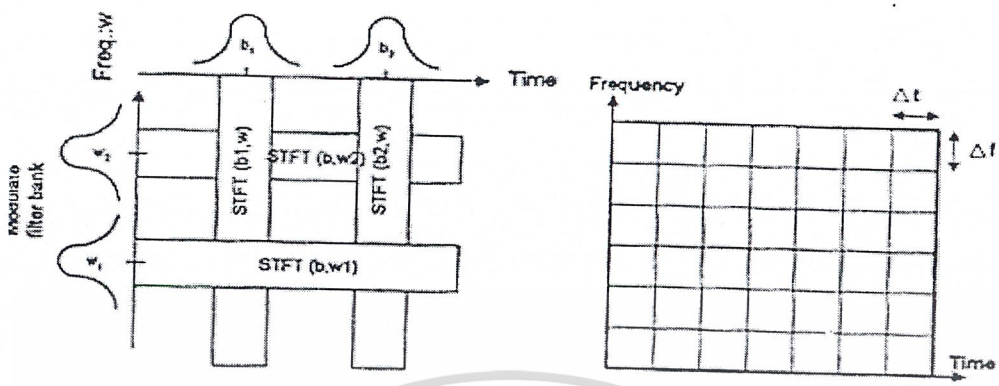
$$G_b^\alpha f(\omega) = \int_{-\infty}^{\infty} (e^{-i\omega t} f(t)) g_\alpha(t-b)dt \tag{2.12}$$

สมการที่ 2.11 เป็น ฟังก์ชันหน้าต่างเกาส์เซียน (gaussian window function) และสมการที่ 2.12 เป็นสมการของ การแปลงกาบอร์ จะเห็นว่าการทรานส์ฟอร์มแบบนี้จะสามารถเลือกตำแหน่งการวิเคราะห์สัญญาณ โดยกำหนดที่พารามิเตอร์ b และเลือกช่วงความถี่ด้วยพารามิเตอร์ α แต่เนื่องจากการแปลงกาบอร์ ใช้ ฟังก์ชันหน้าต่าง แบบเดียวซึ่งอาจไม่เหมาะสมกับการประยุกต์ใช้งานในด้านต่าง ๆ ดังนั้นจึงได้มีการพัฒนาต่อมาเป็นการแปลงฟูริเยร์ในช่วงเวลาสั้น (Short Time Fourier Transform :STFT) ซึ่งจะสามารถทำการเลือกใช้ให้เหมาะสมกับสัญญาณอินพุตที่จะวิเคราะห์ได้ดีขึ้น

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

สมการที่ 2.13 จะเป็นสมการการแปลงฟูริเยร์ในช่วงเวลาสั้น โดยที่ตำแหน่งเวลาในการวิเคราะห์จะกำหนดโดยค่า "b" และช่วงความถี่ในการวิเคราะห์กำหนดด้วยความถี่ของฟังก์ชันหน้าต่าง นั้นๆ แต่จะสังเกตเห็นว่า ทั้งการแปลงกาบอร์ และการแปลงฟูริเยร์ในช่วงเวลาสั้น จะมีลักษณะของช่วงการวิเคราะห์ที่คงที่ดังแสดงในรูปที่ 2.9 (ช่วงเวลา และช่วงความถี่คงที่ ; fixed resolution transform) ทั้งนี้เนื่องจากสัญญาณในทางปฏิบัตินั้น สัญญาณความถี่สูงจะมีช่วงคาบเวลาที่แคบในขณะที่สัญญาณความถี่ต่ำจะมีช่วงคาบเวลาที่กว้างกว่า ดังนั้น การแปลงกาบอร์ และการแปลงฟูริเยร์ในช่วงเวลาสั้น อาจจะให้ข้อมูลที่ได้จากการทรานส์ฟอร์มออกมามาก หรือ น้อยเกินไป (ต้องทำการทรานส์ฟอร์มหลายช่วงเวลามาประกอบกัน จึงจะได้ข้อมูลครบถ้วนในช่วงเวลาที่ต้องการ) ดังนั้นช่วงเวลาควรจะต้องมีการเปลี่ยนแปลงสัมพันธ์กับช่วงความถี่ที่จะทำการวิเคราะห์ โดยที่ช่วงความถี่ที่สูงขึ้นจะมีช่วงเวลาในการวิเคราะห์น้อยลง และในขณะที่เป็นการวิเคราะห์ช่วงความถี่ต่ำจะต้องใช้ช่วงเวลาที่ยาวกว่าในการวิเคราะห์สัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

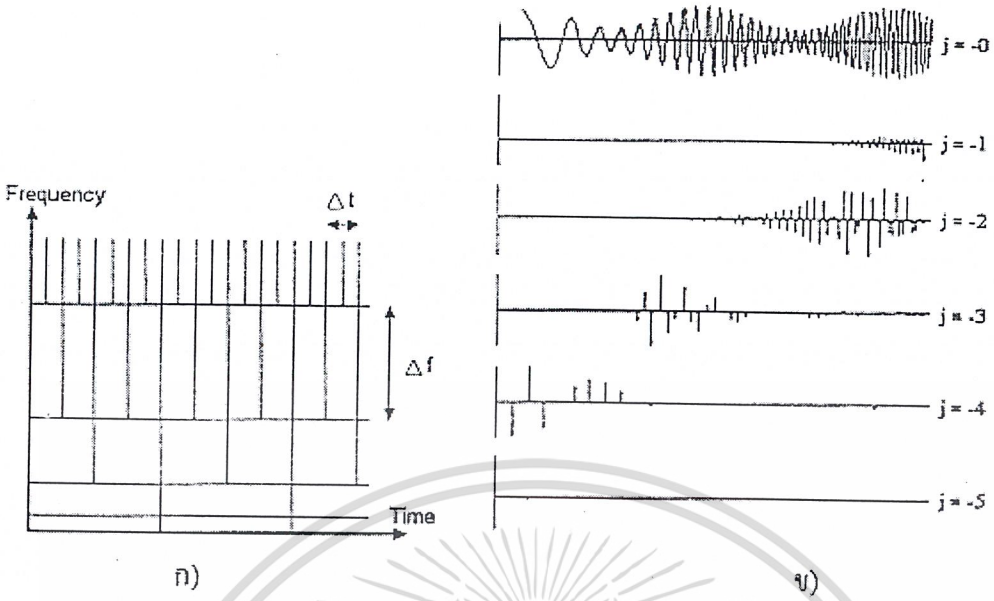


รูปที่ 2.9 แสดงลักษณะของระนาบเวลาความถี่แบบคงที่ (constant time-frequency plane)

จากเหตุผลดังกล่าวข้างต้นทำให้มีการพัฒนาเป็นการทรานส์ฟอร์มในรูปแบบใหม่ที่เรียก "การแปลงเวฟเล็ต" ซึ่งจะเป็นทรานส์ฟอร์มผ่านฟังก์ชันหน้าต่าง แต่ฟังก์ชันหน้าต่าง จะสามารถเลื่อนตำแหน่งและปรับเปลี่ยนช่วงเวลาให้เหมาะกับช่วงความถี่ที่จะวิเคราะห์ได้โดยเรียกฟังก์ชันหน้าต่างแบบนี้ว่า "ฟังก์ชันเวฟเล็ต" (Wavelet function)

$g(t)$ จะเป็นฟังก์ชันเวฟเล็ตแม่ ซึ่งจะถูกเลื่อนตำแหน่ง และ ถูกสเกล(เปลี่ยนความถี่) โดยพารามิเตอร์ "b" และ "α" ตามลำดับ โดยที่ช่วงเวลาและความถี่ในการทรานส์ฟอร์มจะสัมพันธ์กันในลักษณะที่กล่าวมาแล้วข้างต้น ดังแสดงในรูปที่ 2.10 ก.) ซึ่งจะเห็นว่า ระนาบเวลาและความถี่ จะมีลักษณะการเปลี่ยนแปลงช่วงเวลา และความถี่ที่สัมพันธ์กันคือ ในช่วงความถี่สูงจะใช้ช่วงเวลาในการวิเคราะห์เพียงสั้น ๆ (Δf สูง Δf ต่ำ) แต่ในช่วงความถี่ต่ำจะมีช่วงเวลาในการวิเคราะห์ที่ยาวนานขึ้น (Δf ต่ำ Δf สูง) ซึ่งจะเหมาะสมกับลักษณะสัญญาณในทางปฏิบัติ ส่วนในรูปที่ 2.10 ข.) จะเป็นการแสดงลักษณะการแตกกระจายสัญญาณออกไปในระดับความละเอียด (ความถี่) ที่ต่างกันออกไป ขอให้พิจารณาเปรียบเทียบกับรูปที่ 2.7 จะพบว่ามีความสัมพันธ์กัน ในระดับ $j=-1$ จะมีเฉพาะส่วนความถี่สูง และที่ระดับที่ j ต่ำลงสัญญาณจะมีค่าความถี่ที่ต่ำด้วยสอดคล้องกับภาพของ ระนาบเวลาและความถี่ นอกจากนี้ $g(t)$ ยังมีได้หลายรูปแบบ แต่ต้องเป็นไปตามคุณสมบัติของการแปลงเวฟเล็ต ดังนั้น จึงทำให้การแปลงเวฟเล็ตมีความยืดหยุ่นและสามารถเลือกใช้ให้เหมาะสมกับงานที่นำไปใช้ได้เป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก.)แสดงลักษณะของระนาบเวลาและความถี่สำหรับ การแปลงเวฟเล็ต

ข.)แสดงลักษณะการแตกกระจายสัญญาณไปยังระดับความละเอียดต่างๆ

รูปที่ 2.10 แสดงลักษณะของระนาบเวลาเทียบกับความถี่
และลักษณะการแตกกระจายสัญญาณของเวฟเล็ต

2.2.4 ความสัมพันธ์ระหว่างการแปลงเวฟเล็ต และการแปลงฟูริเยร์ในลักษณะของฟังก์ชันหน้าต่าง

จากที่ได้กล่าวมาบ้างแล้วในหัวข้อที่ผ่านมา ในหัวข้อนี้จะกล่าวเปรียบเทียบ การแปลงฟูริเยร์ และการแปลงเวฟเล็ต ในลักษณะ ของฟังก์ชันหน้าต่างให้ละเอียดขึ้น จากการที่การแปลงเวฟเล็ตมีความสัมพันธ์กับ การแปลงฟูริเยร์ และ อนุกรมฟูริเยร์ (Fourier Series) โดยที่ การแปลงฟูริเยร์ จะเป็นการอธิบายสัญญาณในลักษณะของผลรวมของการถ่วงน้ำหนักของฟังก์ชันเอ็กซ์โพเนลเชียลที่ความถี่ต่าง ๆ กัน น้ำหนักที่แต่ละความถี่จะเป็นค่าสัมประสิทธิ์ของการทำ การแปลงฟูริเยร์ แต่ในกรณีของ การแปลงเวฟเล็ต จะเป็นการนำเสนอฟังก์ชันในรูปแบบการประกอบกันของเวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่ง ดังนั้นเวฟเล็ตแม่จะเปรียบได้กับเอ็กซ์โพเนลเชียลฟังก์ชัน ส่วนการสเกลและการเลื่อนตำแหน่งจะเทียบได้กับการเลื่อนความถี่ (frequency shifting) ขณะที่พื้นผิว 2 มิติของสัมประสิทธิ์เวฟเล็ตจะใช้แทนสัมประสิทธิ์ของฟูริเยร์ที่เป็น 1 มิติดังนั้นถ้าให้เวฟเล็ตแม่เท่ากับ $g(t) = e^{jt}$ และให้ค่าการสเกล $a = \frac{1}{\omega}$ การเลื่อนตำแหน่ง $b = 0$ จะสามารถเขียนเป็นสมการ การแปลงเวฟเล็ต ได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$W_g f(a,b) = W_{e^{jt}} f(\tfrac{1}{\omega},0) \tag{2.14}$$

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

ซึ่งจะเป็นสมการของ การแปลงฟูริเยร์ นั่นเอง ดังนั้นจะเทียบได้ว่า การสเกลจะเป็นส่วนกลับของการเปลี่ยนความถี่ที่จะพิจารณานั้นเอง จากสมการ การแปลงฟูริเยร์ จะเห็นว่า $b = 0$ เสมอ และเนื่องจากฟังก์ชันหน้าต่างแบบเอ็กซ์โพเนลเชียลทำการเลื่อนตำแหน่งไม่ได้ เพราะเป็นสัญญาณที่ไม่มีจุดเริ่มต้น และจุดสิ้นสุด $(-\infty,\infty)$ จึงทำการเลื่อนตำแหน่งไม่ได้ต่างกับฟังก์ชันหน้าต่างของ การแปลงเวฟเล็ต ที่มีจุดเริ่มต้นและสิ้นสุด หรือมีขอบเขตที่แน่นอน (compact support) ดังนั้น จึงสามารถยกไปวางตำแหน่งใดก็ได้ทำให้สามารถเลือกตำแหน่งที่จะทำการพิจารณาสัญญาณได้

2.3 การแปลงเวฟเล็ตแบบแบ่งช่วง (Discrete wavelet transform :DWT)

ในเนื้อหาส่วนนี้จะกล่าวถึงการนำการแปลงเวฟเล็ต (Wavelet transform: WT) แบบแบ่งช่วง (discrete) มาวิเคราะห์สัญญาณอินพุตที่มีลักษณะเป็นข้อมูลการสุ่ม (sample data) โดยเริ่มจากคุณสมบัติที่สำคัญของการแปลงเวฟเล็ตที่เกี่ยวกับการวิเคราะห์สัญญาณที่สามารถเลือกระดับความละเอียดได้ (Multiresolution analysis : MRA) จากนั้นจะพิจารณาการแปลงเวฟเล็ตแบบแบ่งช่วง ในลักษณะฟิลเตอร์แบงค์ (filter banks) ที่จะกล่าวถึงการกำหนดการตอบสนองอิมพัลส์ ของตัวฟิลเตอร์ที่นำมาใช้ในกระบวนการ การแปลงเวฟเล็ตแบบแบ่งช่วง

2.3.1 ปริภูมิเวกเตอร์ในการวิเคราะห์สัญญาณ

พิจารณา a,b เป็นจำนวนจริงใด ๆ และ $\hat{i},\hat{j}$ เป็นเวกเตอร์หนึ่งหน่วย (Unit vector) ในปริภูมิเวกเตอร์ (vector space : V) ดังนั้น ถ้า $a\hat{i}+b\hat{j}\in V$ แล้ว จะได้ว่า เวกเตอร์ใด ๆ ที่ประกอบขึ้นจากเวกเตอร์หนึ่งหน่วยหนึ่งตัวหรือหลายตัว แล้วเวกเตอร์นั้นยังคงอยู่ในปริภูมิเวกเตอร์นี้ (V) ดังนั้นเวกเตอร์หนึ่งหน่วยดังกล่าวจะเรียกว่าเป็น เวกเตอร์พื้นฐาน (basis vector) แต่เราจะพิจารณาเวกเตอร์เป็นฟังก์ชันนั่นคือจะเรียกเวกเตอร์พื้นฐานนี้ใหม่ว่า ฟังก์ชันพื้นฐาน (basis function : b.f.) ถ้าฟังก์ชันพื้นฐาน ของเวกเตอร์ทำผลคูณภายใน (inner product) กัน เช่น $\hat{i},\hat{j}$ แล้วมีค่าเป็นศูนย์ จะได้ว่า ฟังก์ชันพื้นฐานดังกล่าวจะแสดงคุณสมบัติออร์ทอกอนัล (orthogonal) ซึ่งกันและกัน และถ้าฟังก์ชันพื้นฐานของเวกเตอร์ทำผลคูณภายในตัวเองแล้วมีค่าเป็นหนึ่ง ฟังก์ชันพื้นฐานที่มีคุณสมบัติทั้งสองก็จะเรียกว่าออร์ทอนอมอล (orthonormal :o.n.)

ถ้าพิจารณาในลักษณะของสัญญาณ อาจกล่าวได้ว่า ฟังก์ชันพื้นฐาน เป็นสัญญาณพื้นฐานเล็ก ๆ ที่สามารถนำมาประกอบกันเป็นปริภูมิของสัญญาณใด ๆ ดังนั้น ถ้าให้ U_i เป็นฟังก์ชันพื้นฐานที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออร์ทอนอมอล (orthonormal basis function :o.n.b.f.) ใด ๆ และ C_i เป็นค่าสัมประสิทธิ์ของ U_i จะได้ว่า สัญญาณ v ใด ๆ นั้น จะสามารถประกอบขึ้นจากฟังก์ชันพื้นฐานที่ออร์ทอนอมอลที่มีค่าขนาดต่าง ๆ มารวมกันดังสมการ

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

2.3.2 การวิเคราะห์แบบหลายระดับความละเอียดของมอร์เลต (Mallat)

สมมติให้ V_j เป็นปริภูมิเวกเตอร์ที่มี j แสดงถึงระดับความละเอียดและจำนวนของฟังก์ชันพื้นฐาน ที่ประกอบขึ้นเป็นฟังก์ชันนั้น จากหลักการของมอร์เลต ที่ได้อธิบายลักษณะคุณสมบัติของการวิเคราะห์สัญญาณที่ระดับความละเอียดต่าง ๆ ได้สรุปลักษณะคุณสมบัติได้ดังนี้

MRA 1) $V_{-\infty} \ldots \subset V_{-1} \subset V_0 \subset V_1 \ldots \subset V_{\infty}$

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) $V_i \oplus W_j = V_{j+1}$; $j \in ZZ$

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 ยังคงลู่เข้าสู่สัญญาณต้นกำเนิด นั่นคือ $f(t) \in L^2(IR)$ และในทางตรงกันข้าม ถ้าระดับความละเอียดที่ระดับ j น้อยลงเรื่อย ๆ จะทำให้สัญญาณที่ประมาณมีข่าวสารสูญหายไปมากขึ้น ตามลักษณะของ MRA 3)

พิจารณาปริภูมิเวกเตอร์ที่ระดับความละเอียด j จะมี ฟังก์ชันพื้นฐาน ที่จะประกอบกันเป็นสัญญาณการประมาณของ $f(t)$ เมื่อ $f(t) \in V_j$ ซึ่งเรียกฟังก์ชันพื้นฐาน เหล่านี้ว่า ฟังก์ชันสเกลลิ่ง (Scaling function) ซึ่ง ฟังก์ชันสเกลลิ่ง จัดเป็นสัญญาณพื้นฐานย่อย ๆ ที่จะเกิดที่ตำแหน่งที่เวลาต่าง ๆ กันภายใน V_j และมีความถี่ต่างกันภายใน V_j เดียวกัน ดังนั้นสามารถกำหนดเซตของฟังก์ชันสเกลลิ่ง ในเทอมของการเลื่อนตำแหน่งเป็นจำนวนเต็มของฟังก์ชันสเกลลิ่งพื้นฐานที่ปริภูมิ V_0 ได้คือ

$$\phi_k(t) = \phi(t-k) \tag{2.17}$$

เมื่อ $k \in ZZ$ และ $\phi \in L^2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ k จะเป็นตำแหน่งการเกิดของ $\phi(t)$ แต่ละตัว และปริภูมิย่อยของ $L^2(IR)$ ถูกแผ่คลุม (span) โดยฟังก์ชันสเกลลิ่งเหล่านี้ที่กำหนดโดย

$$v_0 = span\{\phi_k(t)\} \tag{2.18}$$

สำหรับทุกจำนวนเต็ม k จาก ∞ ถึง $-\infty$
ดังนั้น จะสามารถประมาณฟังก์ชันสัญญาณ $f(t)$ ภายในปริภูมิ V_0 ได้คือ

$$f(t) = \sum_k a_k \phi_k(t) \qquad ; f(t) \in V_0 \tag{2.19}$$

เนื่องจากความถี่จะเปลี่ยนแปลงขึ้นอยู่กับระดับความละเอียดของปริภูมิ j นั้น ๆ โดยอาศัยลักษณะคุณสมบัติที่ MRA 5) จะได้ว่า

$$f(x) \in V_0 \Leftrightarrow f(2x) \in V_1 \tag{2.20}$$
เมื่อ $j = 0$

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

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

const เป็นค่าคงที่ เพื่อทำการนอร์มอลไลซ์ฟังก์ชันสเกลลิ่งที่เกิดจากการสเกล (scaling) จากคุณสมบัติออร์ทอกอนัล ซึ่งกันและกันของฟังก์ชันสเกลลิ่ง ที่พิจารณาจาก

$$\left\langle \phi_{j,k}, \phi_{j,l} \right\rangle = \int \phi^*_{j,k}(t)\phi_{j,l}(t)dt = 0 \tag{2.22}$$
เมื่อ $k \neq l$

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ 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

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

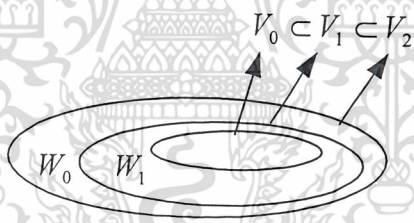
ดังนั้น

$$const = 2^{1/2}$$

จากสมการที่ (2.21) สามารถเขียนได้ใหม่คือ

$$\phi_{j,k}(t) = 2^{1/2}\phi(2^j t - k) \tag{2.24}$$

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$h(k)$ เป็นเซตสัมประสิทธิ์สเกลลิ่งฟังก์ชัน หาได้จาก $\left\langle \phi(t), \sqrt{2}\phi(2t-k) \right\rangle$ และสมการที่ (2.25) จะเรียกว่า สมการรีเคอร์ซีฟ (recursive equation) หรือบางครั้งอาจเรียกว่า สมการรีฟาร์ย์เมนต์(refinement equation) หรือ สมการไดเลชั่น (dilation equation)

จากรูปที่ 2.11 ภายในปริภูมิ V_1 จะประกอบด้วยปริภูมิ V_0 และปริภูมิที่เป็นส่วนประกอบออร์ทอกอนัล (orthogonal complement) ของ V_1 คือ W_0 นั่นคือ

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

เมื่อพิจารณาลักษณะคุณสมบัติตาม MRA 4) จะได้ว่า

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

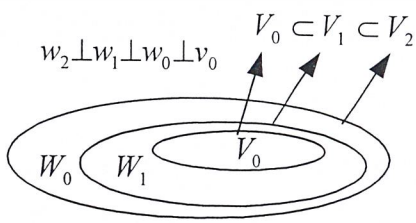
เมื่อนำสมการที่ (2.26) มาแทนใน สมการที่ (2.27)

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

ฉะนั้นโดยทั่วไปตามลักษณะคุณสมบัติ MRA 2) จะได้

$$L^2 = V_0 \oplus W_1 \oplus W_2 \oplus \ldots \tag{2.29}$$

นั่นคือ ภายในปริภูมิของ V_{j+1} จะประกอบด้วย ปริภูมิ V_j และปริภูมิ W_j โดยที่สมาชิกในปริภูมิ W_j ซึ่งจะออร์ทอกอนัล กับ สเกลลิ่งฟังก์ชัน ในปริภูมิ V_j เรียกว่า เวฟเล็ตฟังก์ชัน (Wavelet function : $\psi(t)$) ดังรูปที่ 2.12



รูปที่ 2.12 แสดงปริภูมิเวกเตอร์ของสเกลลิ่งฟังก์ชันและเวฟเล็ตฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

หรือระดับความละเอียดต่ำ เช่น $j=-5$

$$L^2 = V_{-5} \oplus W_{-5} \oplus W_{-4} \oplus \dots \tag{2.31}$$

หรือที่ $j=\infty$

$$L^2 = \dots \oplus W_{-2} \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus W_2 \oplus \dots \tag{2.32}$$

จากสมการที่ (2.32) เทียบกับสมการที่ (2.29) จะได้

$$W_{-\infty} \oplus \dots \oplus W_{-1} = V_0 \tag{2.33}$$

เนื่องจาก $f(t) \in L^2(IR)$ จะได้ว่า $f(t)$ จะประกอบด้วย เวฟเล็ตฟังก์ชันในปริภูมิ W_j นั่นคือสามารถประมาณฟังก์ชัน $f(t)$ ได้คือ

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

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

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

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 เพียงแต่จะเป็นส่วนประกอบออร์ทอกอนัลที่ประกอบอยู่ในปริภูมิ V_1 ดังนั้น ที่ระดับความละเอียดใด ๆ (j) และพิจารณาคุณสมบัติเหมือนสเกลลิ่งฟังก์ชัน จะได้สมการที่คล้ายกับสมการที่ (2.24) คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

เป็นเซตสัมประสิทธิ์เวฟเล็ตฟังก์ชัน โดยมีความสัมพันธ์กับเซตสัมประสิทธิ์สเกลลิ่งฟังก์ชัน ($h(k)$) จากลักษณะคุณสมบัติ MRA 4) จะได้ว่า

$$\left\langle \phi_{j,k}(t), \psi_{j,l}(t) \right\rangle = \int \phi^*_{j,k}(t)\psi_{j,l}(t)dt = 0 \tag{2.38}$$

เมื่อ $j,k,l \in ZZ$

จากสมการที่ (2.38) จะได้ความสัมพันธ์ของสัมประสิทธิ์เวฟเล็ต ที่ออร์ทอกอนัลกับสัมประสิทธิ์สเกลลิ่ง ดังสมการที่ (2.39)

$$h_1(k) = (-1)^k h(N-1-k) \tag{2.39}$$

เมื่อ N เป็นจำนวนข้อมูลของเซตสัมประสิทธิ์สเกลลิ่ง ($h(k)$)

จากสมการที่ (2.29) เราสามารถสร้างฟังก์ชันใด ๆ จากเซตของฟังก์ชัน $\phi_k(t)$ และ $\psi_{j,k}(t)$ แผ่คลุม $L^2(IR)$ ทั้งหมด สำหรับฟังก์ชันใด ๆ $g(t) \in L^2(IR)$ จะได้

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

ที่จะเหมือนกับการกระจายอนุกรมในเทอมของฟังก์ชันสเกลลิ่งและเวฟเล็ต จากสมการที่ (2.40) ในการกระจายอนุกรม $g(t)$ เทอมแรกของสมการ เป็นผลรวมแสดงถึงความละเอียดต่ำ หรือการประมาณที่หยาบ (coarse approximation) ของ $g(t)$ สำหรับแต่ละการเพิ่มของ j ในเทอมที่สองจะแสดงถึงความละเอียดที่สูงขึ้นและทำให้รายละเอียดของข้อมูล (detail) เพิ่มขึ้น เหมือนกับอนุกรมฟูริเยร์ โดยสมการที่ (2.40) ดังกล่าวจะเลือกปริภูมิสเกลลิ่งฟังก์ชันที่สเกล j ใดก็ได้ (สมการที่ (2.41) ใช้ $j = 0$ กำหนดกลุ่มของสเกลลิ่งฟังก์ชัน) คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

จากสมการที่ (2.41) สัมประสิทธิ์ทั้งสองเทอมสามารถคำนวณจากผลคูณภายใน คือ

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

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

สัมประสิทธิ์ในการกระจายของเวฟเล็ต จะถูกเรียกว่า ดิสครีตเวฟเล็ตทรานส์ฟอร์ม ของ $g(t)$

2.3.3 การแปลงเวฟเล็ตโดยใช้หลักการของการวิเคราะห์ฟิลเตอร์แบงค์

ในการประยุกต์ ตัวสัมประสิทธิ์ $h(n)$ และ $h_1(n)$ ที่กำหนดในสมการที่ (2.25) และ (2.37) และสัมประสิทธิ์ $c_j(k)$ และ $d_j(k)$ ในสมการที่ (2.42) และ (2.43) สามารถพิจารณามองเหมือนกับดิจิตอลฟิลเตอร์ (digital filter) และสัญญาณดิจิตอล (digital signal) ตามลำดับ ซึ่งทฤษฎีเวฟเล็ตนี้สามารถจะพัฒนาและศึกษาโดยการวิเคราะห์ใช้ฟิลเตอร์แบงค์ โดยลักษณะของฟิลเตอร์แบงค์ เปรียบเสมือนการแยกองค์ประกอบของสัญญาณ

<u>การวิเคราะห์จากระดับที่ความละเอียดสูง (fine scale) ไปยังระดับความละเอียดต่ำ (coarse scale)</u>
จากสมการรีเคอร์ซีฟ สมการที่ (2.25)

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

ถ้าทำการสเกลและเลื่อนตำแหน่งของตัวแปรเวลาจะได้ว่า

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณา V_j ในทำนองเดียวกับ สมการที่ (2.18) จะได้ว่า

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

ดังนั้น เมื่อ $f(t)\in V_{j+1}$ แล้ว

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

จากสมการที่ (2.48) พิจารณาที่สเกลของ $j+1$ และ $f(t)$ เกิดจากสเกลลิ่งฟังก์ชันอย่างเดียวไม่มีเวฟเล็ตฟังก์ชัน ดังนั้นที่สเกลความละเอียดต่ำ เวฟเล็ตฟังก์ชันจำเป็นสำหรับ

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

ถ้า $\phi_{j,k}(t)$ และ $\psi_{j,k}(t)$ เป็นออร์ทอกอร์นอร์ลกัน จะได้ว่าสัมประสิทธิ์สเกลลิ่งที่ระดับ j ($c_j(k)$) จะหาได้โดยผลคูณภายใน คือ

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

เมื่อนำสมการที่ (2.46) แทนลงในสมการที่ (2.50) จะได้

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

จากสมการที่ (2.50) เมื่อเทียบกับ สมการที่ (2.51)

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

ในทำนองเดียวกันกับกระบวนการตามสมการที่ (2.52) สำหรับสัมประสิทธิ์เวฟเล็ต คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

<u>การฟิลเตอร์ และการลดอัตราการชักค่าตัวอย่างลง (Down sampling)</u>

ในรูปแบบการประมวลผลสัญญาณดิจิตอล, การฟิลเตอร์สัญญาณที่เป็นลำดับ จะใช้การทำคอนโวลูชันระหว่างลำดับ (Sequence) กับเซตของจำนวนสัมประสิทธิ์ตัวฟิลเตอร์

เมื่อพิจารณาสำหรับ ลำดับของอินพุต (input sequence; $x(n)$) และสัมประสิทธิ์ฟิลเตอร์ (filter coefficient; $h(n)$) จะได้ ลำดับของเอาต์พุต (output sequence; $y(n)$)โดย

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

หรือเขียนในลักษณะการคอนโวลูชัน

$$y(n) = h(n) \otimes x(n) \qquad (2.55)$$

ดังนั้น จากสมการที่ (2.52) ทำให้อยู่ในลักษณะการคอนโวลูชัน ได้ดังสมการ (2.56)

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

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

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

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

$$c_j(k) = h(-2k) \otimes c_{j+1}(2k) \qquad (2.58)$$

หรือ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในทำนองเดียวกันกับสมการที่ (2.59) สามารถเขียนสมการที่ (2.53) ในลักษณะการคอนโวลูชันได้เป็น

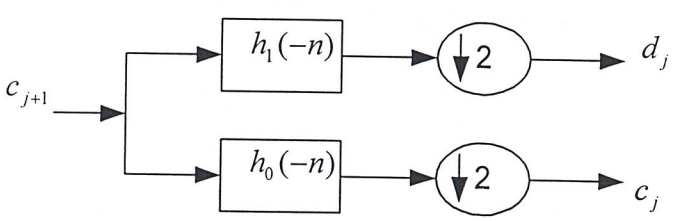
$$d_j(k) = h_1(-n) \otimes c_{j+1}(n) \qquad \text{เมื่อ } n = 2k \tag{2.60}$$

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

รูปที่ 2.13 แสดงการลดอัตราการชักค่าตัวอย่างลงครึ่งหนึ่ง

นอกจากการทำการลดค่าการชักค่าตัวอย่างจะทำให้เกิดการสูญหายของข้อมูลเนื่องจากครึ่งหนึ่งของข้อมูลต้องถูกตัดออกไป แต่ถ้าสัญญาณที่พิจารณาเป็นสัญญาณที่มีขอบเขตจำกัดจะไม่มีการสูญเสียของข่าวสารที่เกิดจากการลดค่าการชักค่าตัวอย่าง

ดังนั้นตัวกรองความถี่แบบดิจิตอล และการลดอัตราการชักค่าตัวอย่าง จะนำมาใช้กับสมการ และตามหลักการของฟิลเตอร์แบงค์ ของสัญญาณ ดังรูปที่ 2.14

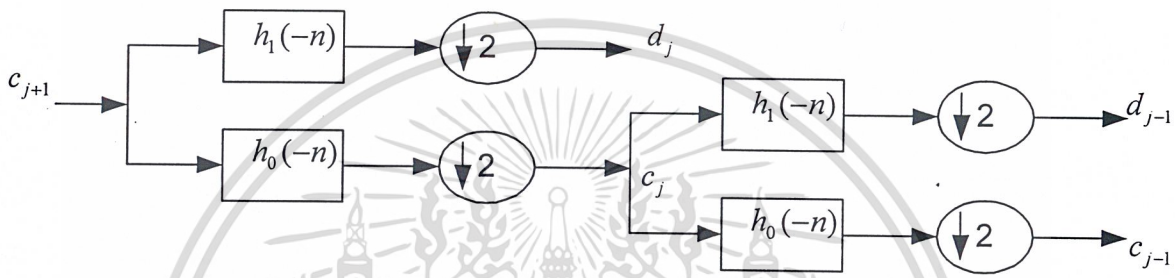


รูปที่ 2.14 แสดงการวิเคราะห์สัญญาณด้วย two-band

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

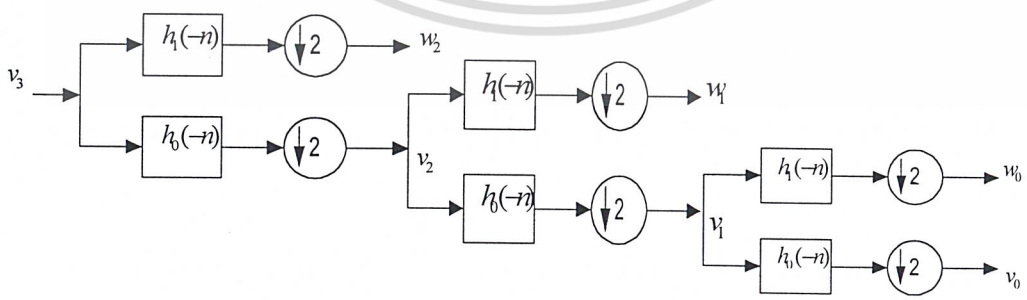
ถ้าจำนวนของตัวสัมประสิทธิ์ฟิลเตอร์มีจำนวนจำกัดขนาด N แล้ว ตัวฟิลเตอร์จะเรียกว่า การตอบสนองอิมพัลส์ที่จำกัด (finite impulse response; FIR) และจะเรียกว่า การตอบสนองอิมพัลส์ไม่จำกัด (infinite impulse response; IIR) ถ้าจำนวนของตัวสัมประสิทธิ์ฟิลเตอร์มีจำนวนไม่จำกัด

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



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

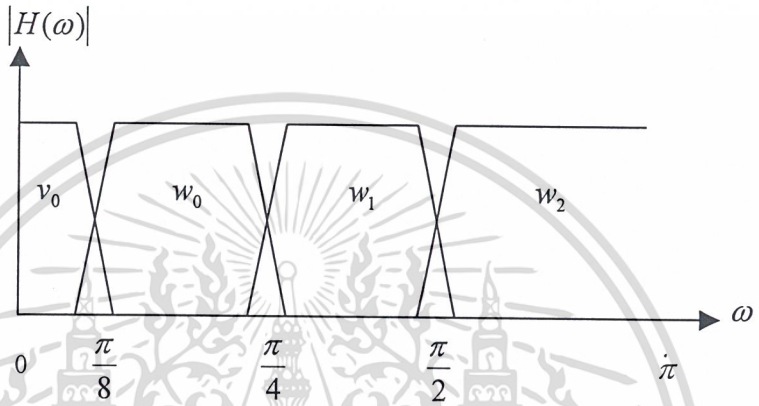
สำหรับในทางปฏิบัติสัญญาณใดที่มีช่วงของสัญญาณจำกัด จะมีระดับสูงสุดที่ $j=J$ ด้วยการเริ่มต้นอธิบายสัญญาณที่ระดับความละเอียดสูง (high resolution) ในเทอมของสัมประสิทธิ์ c_j และจะวิเคราะห์ลงไปสู่ระดับความละเอียดต่ำ (low resolution) $j=j_0$ ดังนั้นจะต้องใช้ขั้นตอนการแยกสัญญาณ $J-j_0$ ขั้นตอน เช่นในปริภูมิ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



รูปที่ 2.16 แสดงการแตกกระจายสัญญาณใน V_3 ไปสู่สเกล $j=0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากในปริภูมิ V_3 จะถูกแตกกระจายสัญญาณไปสู่ระดับ $j=2$ ทั้งในปริภูมิของสเกลลิ่ง (V_2) และปริภูมิของเวฟเล็ต (W_2) ซึ่งในปริภูมิเวกเตอร์ที่สเกล $j=2$ จะมีความละเอียดที่สูงกว่าปริภูมิที่สเกล $j=1$ เมื่อพิจารณาในลักษณะทางด้านความถี่ของสัญญาณ ระดับความละเอียดที่สูงเปรียบได้กับสัญญาณที่มีความถี่สูง ดังนั้น จากรูปที่ 2.16 ตัวฟิลเตอร์ของสัมประสิทธิ์ $h_1(n)$ จะกรองแยกสัญญาณความถี่สูง ที่ระดับ $j=2$ ออกมาจาก V_3 และตัวฟิลเตอร์ของสัมประสิทธิ์ $h_0(n)$ ก็จะกรองแยกสัญญาณความถี่ต่ำออกจาก V_3 ด้วย นั่นคือ $h_1(n)$ จะเป็น การกรองความถี่สูง (highpass filter) และ $h_0(n)$ เป็น การกรองความถี่ต่ำ (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

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

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

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

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

สมการที่ (2.64) เป็นบางส่วนของสมการที่ (2.41) ที่จำกัดของเขตของระดับ

การสังเคราะห์ที่ความละเอียดต่ำไปยังความละเอียดสูง

จากการแตกกระจายสัญญาณไปสู่ระดับต่าง ๆ ดังรูปที่ 2.16 เมื่อต้องการสร้างกลับคืน (reconstruction) ไปสู่สัญญาณที่ระดับที่ละเอียดเหมือนเดิม จะสามารถสร้างได้จากการรวม (combination) ของฟังก์ชันสเกลลิ่งและสัมประสิทธิ์เวฟเล็ตที่ความละเอียดต่ำ โดยการพิจารณาสัญญาณในปริภูมิของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันระดับ คือ $j+1$ ($f(t) \in V_{j+1}$) ซึ่งฟังก์ชันของสัญญาณสามารถเขียนในเทอมของสเกลลิ่งฟังก์ชันเหมือนสมการที่ (2.48)

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

หรือในเทอมที่ประกอบด้วยฟังก์ชันเวฟเล็ต เหมือนสมการที่ (2.49)

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

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

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

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

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

จากสมการที่ (2.67) จัดแทนตัวแปรใหม่ด้วยตัวแปร m คือ

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

จากสมการที่ (2.69) สามารถสร้างกลับคืนค่าสัมประสิทธิ์ $c_{j+1}(k)$ ได้เมื่อกำหนดให้ $2m + n = k$ นั่นคือ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

เมื่อเทียบสมการที่ (2.68) และ สมการที่ (2.71) แล้วจะได้ว่า

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

การฟิลเตอร์และการเพิ่มอัตราการชักค่าตัวอย่าง (Up-sampling)

เนื่องจากการวิเคราะห์สัญญาณด้วยการวิเคราะห์ความถี่โดยการแยกช่วงความถี่ออกจากกันทีละสองช่วง จะทำการแตกสัญญาณหรือกรองสัญญาณผ่านตัวฟิลเตอร์ $h_0(-n)$ และ $h_1(-n)$ แล้วทำการลดอัตราการชักค่าตัวอย่างลง ทำให้ได้สัญญาณที่ประกอบด้วยสเกลลิ่งฟังก์ชัน และเวฟเล็ตฟังก์ชัน ที่ระดับความละเอียดต่ำลงมา ดังนั้นการสังเคราะห์สร้างสัญญาณกลับคืนมา โดยใช้ลักษณะของการกรองความถี่แยกเป็นช่วง หรือเรียกว่าฟิลเตอร์แบงค์ ในการสังเคราะห์สัญญาณกลับ จะตรงกันข้ามกับการวิเคราะห์สัญญาณด้วยการแยกช่วงความถี่ออกจากกันทีละสองช่วง โดยทำการเพิ่มอัตราการชักค่าตัวอย่างให้มีค่าเป็นสองเท่าจากเดิมก่อน ซึ่งจะทำการแทรกค่าศูนย์ระหว่างเทอมของสัญญาณ ดังสมการที่ (2.73)

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

รูปที่ 2.18 แสดงการชักค่าการสุ่มตัวอย่างขึ้นสองเท่า

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

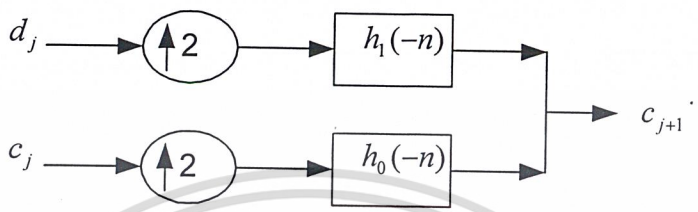
$$c_{j+1}(k) = c_j(k)\otimes h_0(k)\big|_{k=2m} + d_j(k)\otimes h_1(k)\big|_{k=2m} \qquad (2.74)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้ตัวแปร n ตัวแปรใหม่แทนตัวแปร k ในสมการที่ (2.74) จะได้

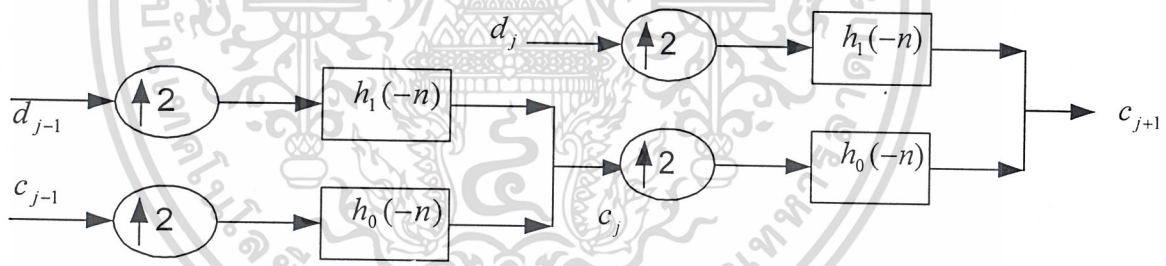
$$c_{j+1}(n) = c_j(n) \otimes h_0(n)\big|_{n=2m} + d_j(n) \otimes h_1(n)\big|_{n=2m} \qquad (2.75)$$

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

ในทำนองเดียวกันกับรูปที่ 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

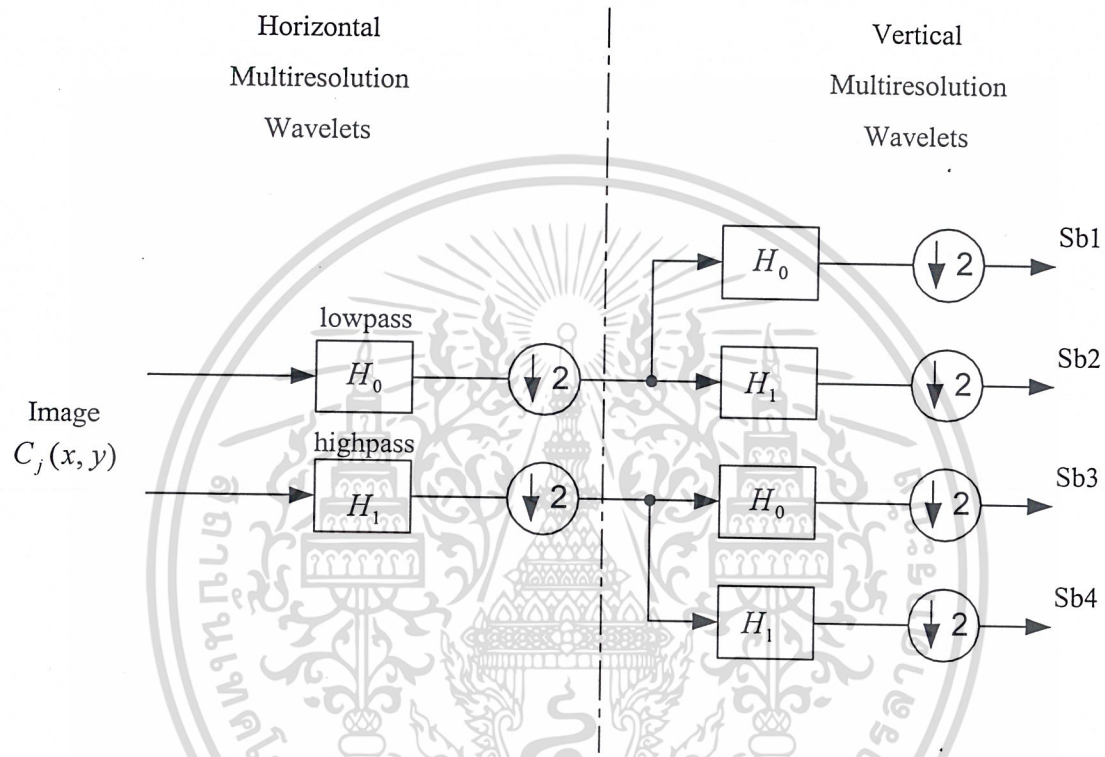
ลักษณะของกระบวนการตามรูปที่ 2.13 ถึง 2.25 จะเรียกว่า ***การแปลงเวฟเล็ตแบบไม่ต่อเนื่อง (discrete wavelet transform; DWT)*** และลักษณะของกระบวนการตามรูปที่ 2.18 และ 2.19 จะเรียกว่า ***การแปลงกลับเวฟเล็ตแบบไม่ต่อเนื่อง (inverse discrete wavelet transform; IDWT)***

เห็นได้ว่า $\phi(t)$ และ $\psi(t)$ จะเป็นฟังก์ชันที่มีช่วงเวลาจำกัด โดยจะมีค่าเท่ากับศูนย์ในกรณีที่ไม่อยู่ในช่วงของ $0 \le t \le N-1$ ดังนั้น $h_0(-n)$ และ $h_1(-n)$ ในสมการที่ (2.25) และ (2.37) จะมีลักษณะเป็นฟิลเตอร์ที่มีการตอบสนองอิมพัลส์ที่จำกัด (FIR Filter) ซึ่งจะเป็นผลให้เฟสของสัญญาณที่ผ่านการกรองความถี่มีลักษณะเชิงเส้นด้วย

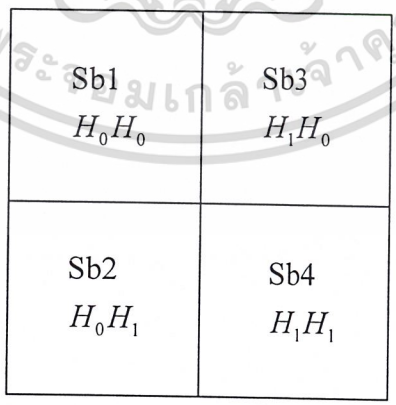
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การแปลงเวฟเล็ตกับข้อมูลภาพ

ข้อมูลภาพมีลักษณะเป็นข้อมูลสองมิติ ดังนั้นการทำการแปลงเวฟเล็ตกับภาพจะเป็นการทำ การแปลงเวฟเล็ตในทางแนวแกน x และแกน y สลับกันในแต่ละครั้ง ซึ่งสามารถเขียนเป็นแผนผัง แสดงได้ดังในรูปที่ 3.2



ก) แสดงลักษณะการแปลงโดยการใช้ Analysis filter banks

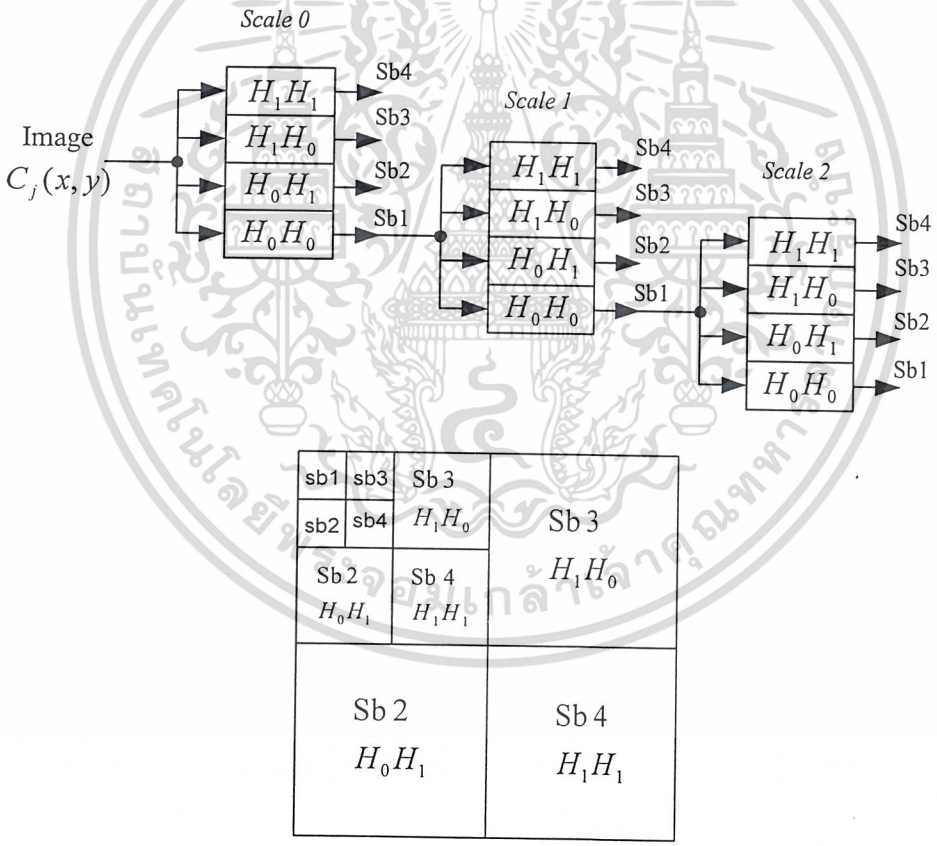


ข) ลักษณะของการแบ่งสับแบนด์ของภาพ

รูปที่ 2.21 แสดงการแปลงเวฟเล็ตของภาพ 1 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.21 สามารถอธิบายได้ว่าการทำการแปลงเวฟเล็ต 2 มิติ จะเป็นการทำการแปลงในแนวแกน x ตามจำนวนแถวของจุดภาพ ซึ่งทำให้ภาพถูกแบ่งเป็น 2 สับแบนด์ จากนั้นทำ การแปลงเวฟเล็ตในแนวแกน y ตามจำนวนคอลัมม์ของจุดภาพ ซึ่งทำให้ภาพถ่ายที่มีลักษณะเป็น 2 มิติที่ผ่านการทำการแปลงเวฟเล็ตหนึ่งครั้งจะถูกแบ่งเป็น 4 ส่วน (สับแบนด์) ตามที่แสดงดังรูปที่ 2.21 สับแบนด์แรก(Sb1) จะเป็นส่วนภาพที่ผ่านการกรองความถี่ต่ำผ่าน ทั้งสองแนวแกน สับแบนด์ที่ 2 (Sb2) จะเป็นส่วนภาพที่ผ่านการกรองความถี่สูงผ่าน ในแนวนอนและกรองความถี่ต่ำผ่าน ในแนวดิ่ง สับแบนด์ที่ 3 (Sb3) จะเป็นส่วนภาพที่ผ่านการกรองความถี่ต่ำผ่านในแนวนอน และการกรองความถี่สูงผ่านในแนวดิ่ง ในส่วนของสับแบนด์สุดท้าย (Sb4) จะเป็นส่วนภาพที่ผ่านการกรองความถี่สูงผ่านทั้งสองแนวแกน ดังนั้น ส่วนที่แสดงรายละเอียดของภาพจะเป็นส่วนข้อมูลในสับแบนด์ที่ 2, 3, และ 4 นอกจากนี้ เมื่อนำเอาสับแบนด์ที่ 1 ไปทำการแปลงเวฟเล็ตต่อไปเรื่อย ๆ ก็จะสามารถทำการแยกสับแบนด์ของภาพออกไปได้มากขึ้นด้วยดังแสดงในรูปที่ 2.22



รูปที่ 2.22 แสดงแผนผังการทำการแปลงเวฟเล็ตกับภาพ 2 มิติ 3 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การวิเคราะห์การแปลงด้วยฮาร์ฟังก์ชัน

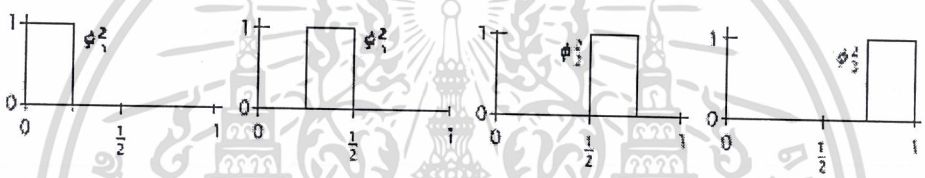
ฟังก์ชันที่ง่ายในการวิเคราะห์ คือ ฮาร์ฟังก์ชัน โดยมีสมการฟังก์ชันสเกลลิ่ง ดังสมการที่ (2.76)

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

สมการที่ (2.76) เมื่อมีการเลื่อนตำแหน่งและสเกล จะทำให้รูปคลื่นของฟังก์ชันมีการเปลี่ยนแปลงดังสมการที่ (2.77)

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

จากสมการที่ (2.77) เมื่อ $j = 2$ และ $k = 0,1,2,3$ จะสามารถแสดงรูปคลื่นได้ คือ



รูปที่ 2.23 แสดงรูปคลื่นจากสมการและเลื่อนตำแหน่ง

ในทำนองเดียวกันฟังก์ชันเวฟเล็ต แสดงดังสมการที่ (2.78)

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

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

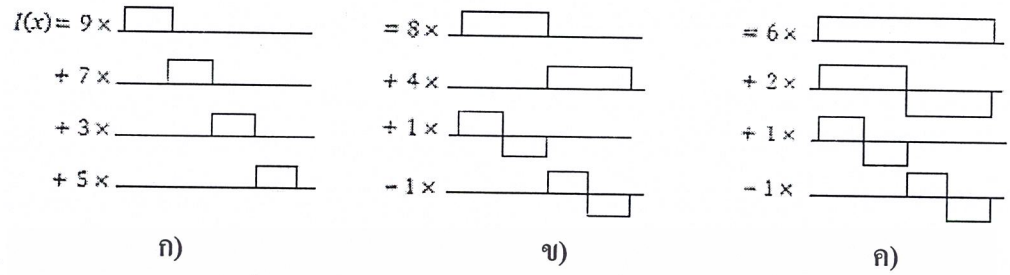
$$\psi_{j,k}(t) = \psi(2^j t - k) \qquad (2.79)$$

เพื่อให้เห็นถึงขั้นตอนการนำเวฟเล็ตมาใช้ในการแปลง ตัวอย่างเช่น สมมติมีข้อมูลภาพ 4 พิเซล คือ 9,7,3,5 ดังนั้นจำนวนระดับสเกล $j = 2, (2^2 = 4)$ เมื่อนำมาเขียนดังสมการที่ (2.19) จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

จะแสดงดังรูปที่ 2.24 ก)



ก) ข) ค)

รูปที่ 2.24 แสดงลักษณะผลรวมของสเกลลิ่งฟังก์ชันพื้นฐาน

จากสมการที่ (2.27) จะสามารถแตกกระจายสัมประสิทธิ์ ได้ คือ

$$V_2 = V_1 \oplus W_1 \qquad (2.81)$$

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

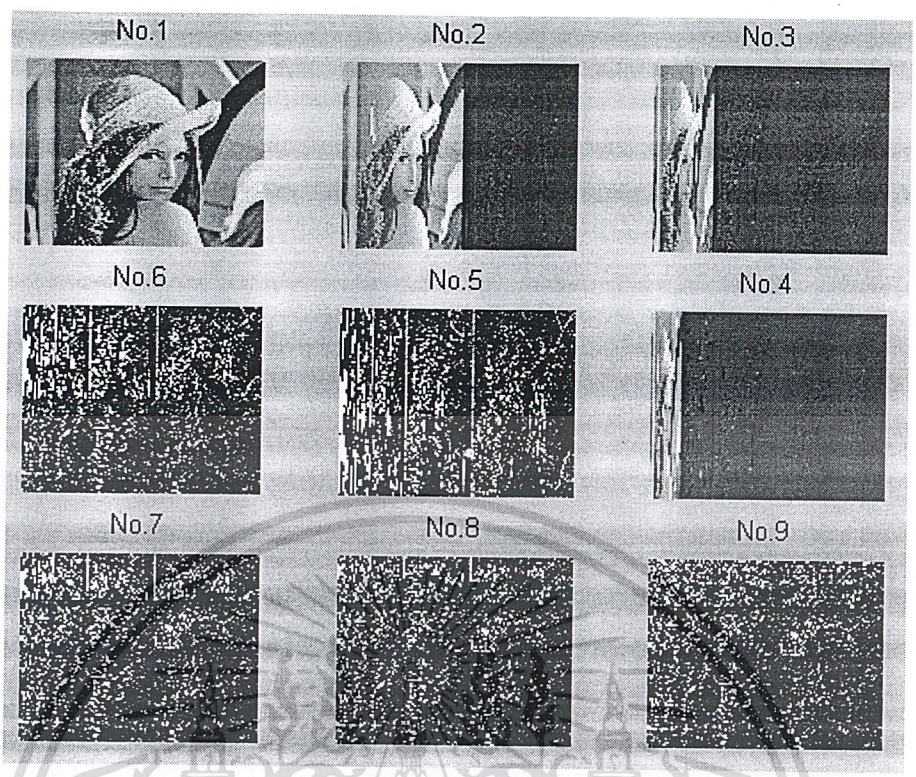
และเมื่อทำต่อจากนั้นอีกจะได้ $V_2 = V_0 \oplus W_0 \oplus W_1$ นั้นคือ

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

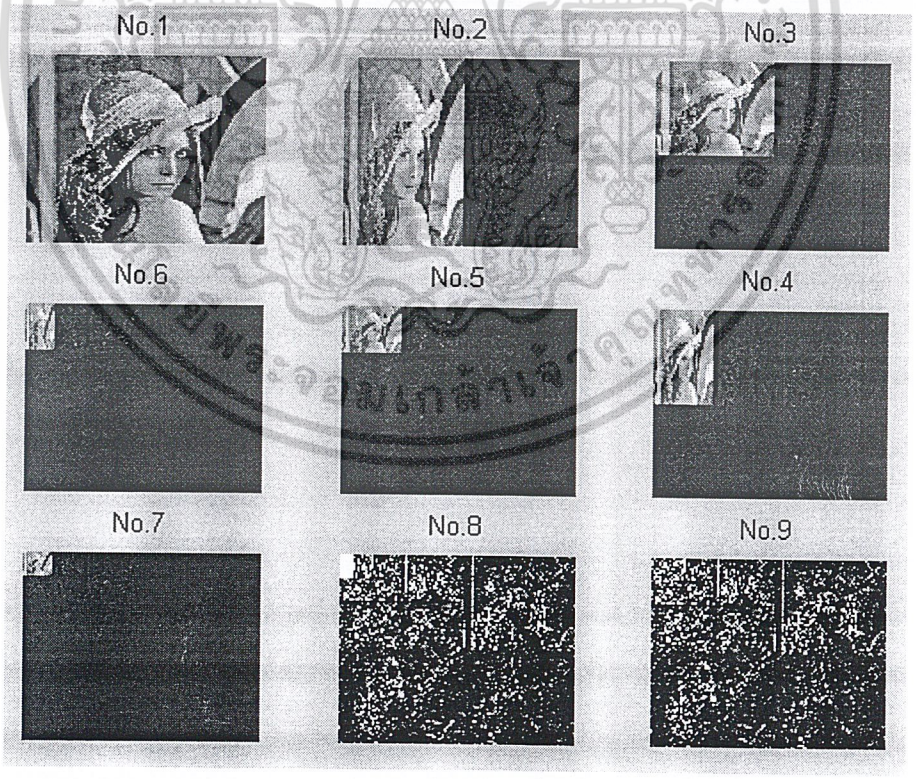
จะได้ผลสุดท้าย คือ การแปลงเวฟเล็ตโดยใช้ฮาร์ฟังก์ชัน คือ 6,2,1,-1

เนื่องจากสมการที่ (2.77) และ (2.79) ยังไม่ทำการนอร์มอลไลซ์ฟังก์ชัน ด้วย $2^{1/2}$ ดังนั้นเมื่อทำการนอร์มอลไลซ์ฟังก์ชันจากการแปลงสุดท้ายด้วย $2^{-1/2}$ ตามระดับสากล จะได้เป็น $6,2,1/\sqrt{2},-1/\sqrt{2}$ จากการพิจารณามาพบว่า ลักษณะของการนำฮาร์ฟังก์ชันมาใช้ในการกระจายสัมประสิทธิ์เหมือนกับการบวกและลบแล้วเฉลี่ยค่าเพียงเท่านั้น เนื่องจากข้อมูลในการวิเคราะห์ข้อมูลอยู่ในลักษณะของเมตริกซ์ ในการแปลงภาพโดยการแปลงเวฟเล็ตนั้นมีขั้นตอนการแปลงอยู่ 2 กระบวนการใหญ่ ๆ คือ วิธีการแปลงแบบมาตรฐาน (standard transform) และวิธีการแปลงแบบไม่มาตรฐาน (nonstandard) [7] ซึ่งมีการแปลงดังรูปที่ 2.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก.) แสดงลำดับการแปลงภาพแบบมาตราฐาน



ข.) แสดงลำดับการแปลงภาพแบบไม่มาตรฐาน

รูปที่ 2.25 แสดงวิธีการแปลงแบบมาตรฐาน (standard transform)
และวิธีการแปลงแบบไม่มาตรฐาน (nonstandard)

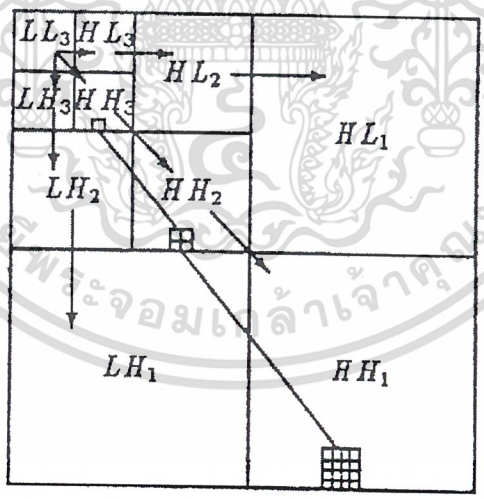
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การใช้ซีโร่ทรีของสัมประสิทธิ์เวฟเล็ตเข้ารหัสข้อมูลภาพ (Embeded Image Coding Using Zerotrees of Wavelet Coefficients)

สิ่งที่ได้หลังจากการแปลงเวฟเล็ตจะเรียกว่า สัมประสิทธิ์เวฟเล็ต สัมประสิทธิ์นี้จะถูกนำไปเข้ากำหนดรหัสและทำการลดรหัสข้อมูลลง โดย Jerome M. Shapiro จาก MIT ได้เสนอขั้นตอนขบวนการที่เหมาะสมกับการเข้ารหัสของสัมประสิทธิ์เวฟเล็ต คือ สัมประสิทธิ์ดังกล่าวจะถูกเปรียบเทียบกับค่าเทรชโฮลในระดับต่างๆ เพื่อกำหนดความสำคัญ (Significant) ของสัมประสิทธิ์แล้วนำมาจัดระดับข้อมูลด้วยการ ควอนไทซ์แบบการประมาณต่อเนื่อง (success – approximation quantization : SAQ) เมื่อกำหนดรหัสให้สัมประสิทธิ์แล้ว จะลดรหัสข้อมูลด้วยการเข้ารหัสเลขคณิต (Arithmetic Coding) ซึ่งเรียก ขบวนการดังกล่าวว่า เอ็มเบดอิด ซีโร่ทรี (Embedded Zerotrees Wavelet : EZW)

ก.) การกำหนดสัญลักษณ์ให้สัมประสิทธิ์เวฟเล็ต (Wavelet Coefficient Encoding)

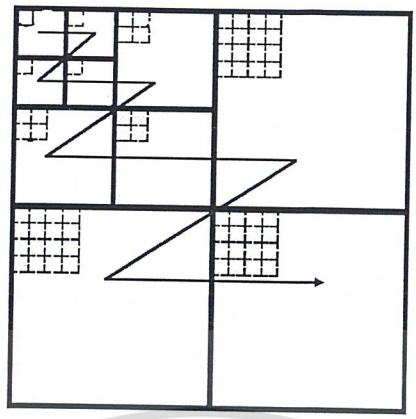
พิจารณารูปที่ 2.26 LL_3 จะถือว่าเป็นสับแบนด์สเกลที่หยาบสุด (Coarse Scale) และเรียกสัมประสิทธิ์ในสเกลนี้ว่าแหล่งกำเนิดหรือโหนดแม่ (parent) ส่วน HL_3 , HH_3 และ LH_3 เป็นสับแบนด์สเกลที่ละเอียดขึ้นมา เรียกสัมประสิทธิ์ในสเกลว่า โหนดลูก (Children) และสัมประสิทธิ์ในสเกลที่ละเอียดขึ้นมาอีก คือ HL_2 , HH_2 และ LH_2 เรียกว่า โหนดสืบทอด (descendant) โดยความสัมพันธ์ดังกล่าวจะเกิดไปจนกระทั่งถึงระดับสับแบนด์สเกลที่ละเอียดสูงสุด



รูปที่ 2.26 แสดงลักษณะของการสเกล และลักษณะในการสืบทอดของสัมประสิทธิ์เวฟเล็ต

พิจารณาสัมประสิทธิ์เวฟเล็ต X และค่าเทรชโฮล τ ถ้า $|X| > \tau$ สัมประสิทธิ์ในลักษณะนี้เรียกว่าสัมประสิทธิ์ที่มีนัยสำคัญสูง (significant coefficient) ภายเงื่อนไขของ $|X|$ นอกจากนั้นเรียกว่าสัมประสิทธิ์ตัวไม่สำคัญ (insignificant coefficient) โดยวิธีการนำสัมประสิทธิ์มาเปรียบเทียบกับค่าเทรช-โฮลซึ่งจะใช้การสแกนค่าของสัมประสิทธิ์แบบตัวซี (Z- SCAN) ดังรูปที่ 2.27

เอกสารนี้มีลิขสิทธิ์ ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 แสดงลักษณะการสแกนแบบตัวซี (Z)

สัมประสิทธิ์ตัวไม่สำคัญ ($|X| \le \tau$) จะถูกจัดค่าให้เป็นศูนย์ และถ้าสเกลสัมประสิทธิ์ทั้งหมดของ โหนดแม่ , โหนดลูก และ โหนดสืบทอด มีค่าของสัมประสิทธิ์น้อยกว่าค่าเทรชโฮล ก็จะจัดค่าสัมประสิทธิ์ทั้งหมดในสเกลให้เป็นศูนย์ทั้งหมด ลักษณะดังกล่าวจะเกิดเป็น กลุ่มของศูนย์หรือเรียกว่า ซีโร่ทรีรูท (Zerotrees Root : ZTR) แต่ถ้าในสเกลใดสเกลหนึ่งใน โหนดลูก และ โหนดสืบทอด มีสัมประสิทธิ์มากกว่าค่าเทรชโฮลเพียงหนึ่งตัว จะเกิดลักษณะที่เรียกว่า ไอโซเลทซีโร่ (Isolate Zero : IZ) ส่วนสัมประสิทธิ์ตัวสำคัญจะกำหนดเป็นสัมประสิทธิ์ที่เป็นบวก (Positive : POS) และเป็นลบ (Negative : NEG) ตามค่าของสัมประสิทธิ์นั้นๆ 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 ซึ่งความละเอียดของค่าสัมประสิทธิ์จากการควอนไทซ์จะสัมพันธ์กับค่าเทรชโฮลต่อการทำสแกนข้อมูลสัมประสิทธิ์หนึ่งครั้ง นั้นคือ

$$\tau_i = \frac{\tau_{i-1}}{2} \quad , i = 1,2,\ldots \tag{2.84}$$

เมื่อ τ เป็นค่าเทรชโฮลและ i. เป็นจำนวนครั้งการสแกนข้อมูลสัมประสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

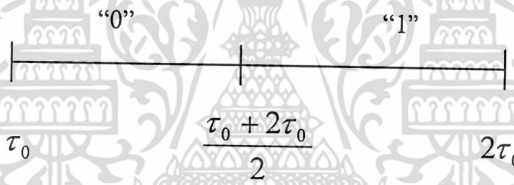
โดยค่าเทรชโฮลเริ่มต้น (τ_0) หาได้จากครึ่งหนึ่งของค่าสัมประสิทธิ์ที่สูงสุดในการแปลงเวฟเล็ต ($|x|_{\max} < 2\tau_0$) และค่าของ τ_0 ควรอยู่ในรูปของเลขยกกำลังสอง คือ

$$\tau_0 = M2^E \tag{2.85}$$

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 สับออร์ดิเนทจะใช้ตัวเลขฐานสอง คือ แทน "1" เพื่อชี้ค่าจริงของสัมประสิทธิ์ว่า มีค่าในช่วงบนจากค่ากลางระหว่างค่าเทรชโฮล และแทน "0" เพื่อชี้ค่าจริงของสัมประสิทธิ์ว่าอยู่ในช่วงล่างของค่ากลางระหว่างค่าเทรชโฮล ดังรูปที่ 28 ส่วนการถอดรหัสก็จะทำตรงกันข้ามกับการเข้ารหัส



รูปที่ 2.28 แสดงการควอนไทซ์ค่าสัมประสิทธิ์และกำหนดสัญลักษณ์

2.5 หลักการเบื้องต้นของการลดข้อมูล

การลดข้อมูลหรือการบีบอัดข้อมูล จะทำการลดหรือกำจัดข้อมูลส่วนเกินออก โดยข้อมูลส่วนเกินดังกล่าวจะแบ่งเป็น 3 ประเภทคือ ข้อมูลส่วนเกินจากการเข้ารหัสข้อมูล (coding redundancy) , ข้อมูลส่วนเกินที่เกิดจากสหสัมพันธ์ (correlation) ระหว่างพิกเซลของภาพ โดยค่าของพิกเซลใดๆ สามารถพยากรณ์ได้จากค่าของพิกเซลรอบข้างของตัวมัน เรียกว่า อินเตอร์พิกเซลรีดันเดซี (interpixel redundancy) ,และเกิดจากความสามารถในการรับรู้ทางประสาทตาของมนุษย์เรียกว่า ไซโครวิชวลรีดันเดซี (psychovisual redundancy)

2.5.1 เอนโทรปี (Entropy)

ชุดข้อมูล (data string) คือกลุ่มของสัญลักษณ์ (Symbol) และชุดของรหัส (Code) ก็คือชุดของเลขฐานสอง คือ "0" และ "1" โดยรหัสจะถูกจัดให้กับสัญลักษณ์แต่ละตัวที่อยู่ในชุดของข้อมูล ในขบวนการของการเข้ารหัสจะทำการนำสัญลักษณ์เหล่านี้มาแปลงเป็นรหัส ถ้าขบวนการเข้ารหัสนั้นดำเนินไปอย่างมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพ การบีบอัดข้อมูลก็จะมีประสิทธิภาพด้วย ผลก็คือ รหัสจะมีขนาดรวมทั้งหมดเล็กกว่า กลุ่มของสัญลักษณ์ ถ้าเราพิจารณถึงความซับซ้อน (redundancy) ของสัญลักษณ์ว่าเป็นการกระจายของความน่าจะเป็น หรือ การกระจายของความถี่ของมัน เช่น ในภาพที่ไม่มีความสัมพันธ์ของสัญลักษณ์แต่ละตัวของมันที่อยู่ใกล้กัน รูปแบบของสัญลักษณ์นี้จะถูกเรียกว่า ดิสครีตเมโมรี่เลสซอร์ต (Discrete Memoryless Source :DMS) หรือ ซีโร่ออเดอร์มาร์คอฟ (Zero – Order Markov Source) ในกรณีนี้วิธีเข้ารหัสที่ดีที่สุดที่สามารถอธิบายได้ด้วยคณิตศาสตร์ ซึ่งก็คือ ขบวนการการเข้ารหัสเลขคณิต ตามทฤษฎีของ ชานอล-เอนโทรปี (Shanon's Entropy)

ก. สัญลักษณ์และรหัส (Symbol and Code)

ข้อมูล คือ ข้อความ หรือ อินโฟเมชั่น (Information) ซึ่งต้องการที่จะส่ง, เก็บ, หรือแสดง ในรูปแบบของกลุ่มสัญลักษณ์ สัญลักษณ์จะเป็นสิ่งที่ใช้แทนข้อมูล ซึ่งสามารถจัดให้อยู่ในรูปแบบของข้อมูลที่เป็น ตัวอักษร , สัญลักษณ์ ,ตัวเลข หรือ อื่นๆ เพื่อความมีประสิทธิภาพ และการเก็บความลับ รหัสจะถูกกำหนดให้กับทุกๆ สัญลักษณ์ โดยทั่วไปก็คือ "0" และ "1" เพราะฉะนั้นแทนที่จะส่งข้อมูลในลักษณะที่เป็นสัญลักษณ์เราจะกำหนดสัญลักษณ์เหล่านั้นเป็นรหัส แล้วส่งรหัสดังกล่าวไปแทน โดยเราจะสามารถเรียกข้อมูลเดิมกลับมาจากรหัสได้ก็ต่อเมื่อเรารู้ความสัมพันธ์ระหว่างสัญลักษณ์ และ รหัส ซึ่งเรียกว่า ตัวเก็บรหัส (Codebook) เช่น ถ้าเราพิจารณาลำดับของสระดังนี้

i , o , a , e , e , i , a , e

เซตของสัญลักษณ์คือ S = { a , e , i , o , u } และข้อมูลคืออักษร 8 ตัว คือ D = i , o , a , e , e , i , a , e ซึ่งสมมติว่ามีค่าในตัวเก็บรหัสเป็น C = { 000 , 001 , 010 , 011 ,100 } ซึ่งจะเปลี่ยนเป็นชุดของข้อมูลดังนี้

010 011 000 001 001 010 000 001

ซึ่งจะได้ 24 บิต โดยเมื่อทำการเปลี่ยนค่าของรหัสในตัวเก็บรหัสเป็น C = {00 , 01 , 10 , 110 , 111 } ชุดของข้อมูลจะมีเพียง 17 บิต ดังนี้

10 110 100 100 10

แต่อย่างไรก็ตามชุดข้อมูลนี้ไม่สามารถทำการสร้างสัญลักษณ์กลับขึ้นมาได้ซึ่งข้อมูลที่ถูกสร้างกลับอาจถูกพิจารณาเป็น i o a e e i a e

ซึ่งจะมีคำถามว่า เมื่อใดที่ตัวเก็บรหัสชนิดไหนจะสามารถนำมาใช้ได้และเมื่อใดที่ตัวเก็บรหัสจะมีความสามารถในการบีบอัดที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามแรกจะสามารถตอบได้โดยไม่ยากนักและสามารถที่จะพิสูจน์ได้ คือ *"รหัสจะใช้ได้ดีก็ต่อเมื่อ ไม่มีรหัสใดเหมือนกับช่วงแรกของรหัสอื่นๆ ในตารางรหัส (ตัวเก็บรหัส) นี่คือรูปแบบของตัวเก็บรหัสที่ดี"* จากตัวอย่างที่ได้กล่าวมาแล้ว รหัส 0 ของ e คือ ช่วงแรกของรหัส 01 และ รหัส u ของ 0 คือ ช่วงแรกของรหัส 011 คือ u ซึ่งเห็นได้ว่าจะมีรหัสที่มีค่าของรหัสช่วงแรกเหมือนกับรหัสอื่น ๆ ในตัวเก็บรหัส

ส่วนคำถามที่สองนี้ถูกถามโดยฮัฟแมน (Huffman) โดยผลลัพธ์ได้ถูกจัดทำเป็นการเข้ารหัสแบบฮัฟแมน (The Huffman Coding)

ข. ทฤษฎีเอนโทรปีของชานอล (The Shannon Entropy Theorem)

ให้ $S=\{s_1,s_2,\ldots,s_n\}$ คือเซตของชุดสัญลักษณ์ N และ $D=\{d_1,d_2,\ldots d_l\}$ ชุดของสัญลักษณ์ l ในลำดับ (l คือความยาวของข้อมูล D) การกระจายของความน่าจะเป็นของสัญลักษณ์ S ในข้อมูล D เป็นกลุ่มของจำนวนบวก $P=\{p_1,p_2,\ldots,p_n\}$ ซึ่งแต่ละสัญลักษณ์กำหนดได้ดังนี้

$$P_i=\frac{\left|\{d_k\in D\,|\,d_k=s_i\}\right|}{l},\forall i=1,2,\ldots,n. \qquad (2.86)$$

ผลลัพธ์ คือ $\sum P_i=1$ ซึ่งถ้าความน่าจะเป็นในการกระจายตัวเป็นข้อมูลที่ซับซ้อนชนิดเดียวที่ถูกนำไปใช้คู่ของ (S , P) เรียก ซีโร่ออเดอร์มาร์คอฟ (Zero Order Markov Source) และลำดับข้อมูล D จะเรียกว่าลำดับของซีโร่ออเดอร์มาร์คอฟ (The Zero Order Markov Sequence)

จากข้างต้นที่กล่าวมา ซีโร่เอนโทรปี (Zero Entropy) ของลำดับข้อมูล D ถูกนำเสนอขึ้นมาโดย ครอส์ ชานอล (Claude Shannon) ในปี ค.ศ. 1940 ซึ่งเขาได้ยืมทฤษฎีของเทอร์โมไดนามิกมาใช้แสดง ดังนี้

$$e=e(D)=-\sum_{i=1}^{n}p_i\cdot\log_2 p_i \qquad (2.87)$$

กำหนดให้ $C=\{c_1,c_2,\ldots,c_n\}$ เป็นตัวเก็บรหัส ดังนั้นถ้าจะพิจารณา ความยาวเฉลี่ยของรหัสที่ใช้ในตัวเก็บรหัส C ที่แทนลำดับของข้อมูล D จะได้ดังสมการ (2.88)

$$e_C(D)=\sum_{i=1}^{n}p_i\cdot\left|c_i\right| \qquad (2.88)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎี สำหรับข้อมูลของสัญลักษณ์ในตัวเก็บรหัสของกลุ่มสัญลักษณ์ ความยาวเฉลี่ยของรหัสที่อยู่ในตัวเก็บรหัสที่ไว้สำหรับแสดงแทนลำดับข้อมูล จะมากกว่าหรือ เท่ากับซีโร่ออเดอร์เอนโทรปี (Zero Order Entropy) ของลำดับข้อมูลเสมอ

พิสูจน์

กำหนด $D=\{d_1,d_2,\ldots,d_l\}$ เป็น Zero Order Markav ลำดับของความยาวข้อมูล e ในสัญลักษณ์จากเซต $S=\{s_1,s_2,\ldots,s_n\}$ โดยค่าการกระจายของความน่าจะเป็น $P=\{p_1,p_2,\ldots,p_n\}$ และกำหนด $C=\{c_1,c_2,\ldots,c_n\}$ คือตัวเก็บรหัสของ S เพราะฉะนั้นจากทฤษฎีจะได้เท่ากับสมการ (2.89)

$$e_C(D)=\sum_{i=1}^{n}p_i\cdot|c_i|\geq e(D)=-\sum_{i=1}^{n}p_i\cdot\log_2 p_i \qquad (2.89)$$

ในกรณีที่ $n=1$ จะไม่พิจารณา

แต่กรณีที่ $n>1$ โดย C_0 เป็นเซตของรหัส ซึ่งมีรหัสแรกคือ 0 และ C_1 เป็นเซตของรหัสที่เริ่มด้วย 1 ถ้าเซตใดเซตหนึ่งเป็นเซตว่างเราสามารถที่ไม่ใช้รหัสตัวแรกของชุดรหัสนั้นได้ และได้รหัสใหม่ คือ C' โดยรหัสใหม่จะมีความยาวของมันน้อยกว่า C เราจึงแทน C ด้วย C' จะเห็นว่า $e_{C'}(D)<e_C(D)$ ซึ่งถ้า $e_{C'}(D)\geq e(D)$ จะสามารถพิสูจน์ได้ว่า $e_C(D)\geq e(D)$ ด้วย

ถ้าทั้ง C_0 และ C_1 ไม่ว่างเปล่า และ บรรจุจำนวนของรหัสน้อยกว่าในระดับที่ n เราให้ C_0^- และ C_1^- แทนชุดของรหัส C_0 และ C_1 ที่ไม่มีรหัสตัวแรก โดยแยกสัญลักษณ์ตามสมการ (2.90)

$$S_0=\{S_i\in S\mid C_i\in C_0\}\text{ และ}$$
$$S_1=\{S_i\in S\mid C_i\in C_1\} \qquad (2.90)$$

และเซตของข้อมูลทั้ง 2 D_0 และ D_1 ได้มาจากการกำหนด D ให้เป็นสับเซตของ S_0 และ S_1 ตามลำดับ โดย

$$q_0=\sum_{c_i\in C_0}p_i \text{ และ } q_1=\sum_{c_i\in C_1}p_i \qquad (2.91)$$

เห็นได้ว่า $q_0+q_1=1$ และ

$$P_0=\{\frac{P_i}{q_0}\in S\mid c_i\in C_0\}\text{ และ }P_1=\{\frac{P_i}{q_1}\in S\mid c_i\in C_1\} \qquad (2.92)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นความน่าจะเป็นของเซต D_0 และ D_1 ในสัญลักษณ์ S_0 และ S_1 ตามลำดับในการนำทฤษฎีมาใช้กับ (D_0, S_0, C_0^-, P_0) และ (D_1, S_1, C_1^-, P_1) เราจะได้

$$e_{C_k^-}(D_k) = \sum_{c_i \in C_k} \frac{p_i}{q_k} \cdot (|c_i| - 1) \geq e(D_k) = -\sum_{c_i \in C_k} \frac{p_i}{q_k} \cdot \log_2 \frac{p_i}{q_k} \qquad (2.93)$$

เมื่อ $k = 0,1$ และ

$$\sum_{c_i \in C_k} p_i \cdot |c_i| - \sum_{c_i \in C_k} p_i \geq -\sum_{c_i \in C_k} p_i \cdot \log_2 p_i + \sum_{c_i \in C_k} p_i \cdot \log_2 q_k \qquad (2.94)$$

เมื่อ $k = 0,1$ จะได้

$$\sum_{c_i \in C_k} p_i \cdot |c_i| - q_k \geq -\sum_{c_i \in C_k} p_i \cdot \log_2 p_i + q_k \log_2 q_k \qquad (2.95)$$

เมื่อบวกสมการทั้งสองกรณีสำหรับ $k = 0,1$ เข้าด้วยกัน จะได้ว่า

$$\begin{aligned} e_c(D) = \sum_{i=1}^{n} p_i \cdot |c_i| &\geq -\sum_{i=1}^{n} p_i \cdot \log_2 p_i + q_0 \cdot \log_2 q_0 + q_0 + q_1 \cdot \log_2 q_1 + q_1 \\ &= e(D) + \left(1 + q_0 \cdot \log_2 q_0 + (1 + q_0) \cdot \log_2 (1 - q_0)\right) \\ &\geq e(D) \end{aligned} \qquad (2.96)$$

เมื่อฟังก์ชัน $f(x) = x \log x + (1-x)\log(1-x)$ มีจุดตัดบนแกน x 1 จุดซึ่งจะเป็นจุดที่มีค่าน้อยที่สุด ตรง $x = \frac{1}{2}$ ตามทฤษฎีของแคลคูลัส (Calculus)

วิธีการเข้ารหัสของฮัฟแมน (The Huffman Coding Algorithm) จะพบตัวเก็บรหัส ซึ่งให้ค่าความยาวเฉลี่ยที่น้อยที่สุดเท่าที่จะเป็นได้ แต่ในส่วนของวิธีการเข้ารหัสแบบเลขาคณิต (The Arithmetic Coding Algorithm) จะได้ว่าความยาวเฉลี่ยของรหัสจะเท่ากับค่าของเอนโทรปีของลำดับข้อมูล P

เทคนิคในการลดข้อมูลสามารถแบ่งได้เป็น 2 ประเภท คือ การลดข้อมูลแบบไม่มีการสูญเสีย (lossless compression) และการลดข้อมูลแบบที่ยอมให้มีการสูญเสีย (lossy compression)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 การลดข้อมูลแบบที่ไม่มีการสูญเสีย (Lossless Compression)

การลดข้อมูลแบบที่ไม่มีการสูญเสียจะนำเอาประโยชน์ของส่วนเกินของข้อมูล (redundancy) มาใช้ ซึ่งวิธีการหนึ่งที่จะนำมาใช้วัดส่วนเกินของข้อมูล คือการหาความน่าจะเป็นของการกระจายของค่าความเข้มของจุดภาพ โดยที่เอนโทรปีอันดับที่หนึ่ง สามารถหาได้จาก

$$H = -\sum_{i} p(i)\log_2(p(i)) \qquad (2.97)$$

เมื่อ $p(i)$ คือความน่าจะเป็นของค่าความเข้ม i ของจุดภาพที่เกิดขึ้น

เอนโทรปีจะใช้อธิบายความไม่แน่นอนของข้อมูล และจะแสดงค่าเฉลี่ยของจำนวน บิตที่จะใช้แทนข้อมูล ภาพจะถูกลดข้อมูลโดยการเข้ารหัสค่าความเข้มของจุดภาพที่เกิดขึ้นบ่อยด้วยจำนวนบิตที่น้อย และเข้ารหัสค่าความเข้มของจุดภาพที่เกิดขึ้นน้อยกว่าด้วยจำนวนบิตที่มาก ซึ่งโดยเฉลี่ยแล้วจะเป็นการลดจำนวนบิตที่ใช้แทนข้อมูล

ตัวอย่างของการเข้ารหัสแบบที่ไม่มีการสูญเสีย

การเข้ารหัสฮัฟแมน (Huffman Coding)

การเข้ารหัสฮัฟแมนเป็นการลดข้อมูลแบบที่ไม่มีการสูญเสียวิธีหนึ่ง หลักการที่สำคัญของการรหัสฮัฟแมนคือ ข้อมูลที่มีความน่าจะเป็นสูงกว่าจะถูกเข้ารหัสด้วยจำนวนบิตที่น้อยกว่า

การเข้ารหัสฮัฟแมนสามารถอธิบายวิธีการโดยสรุป ได้ดังนี้

(1) ทำการนับความถี่เพื่อหาความน่าจะเป็นของข้อมูลทุกตัว และเรียงลำดับข้อมูลตามความน่าจะเป็นจากน้อยไปหามาก

(2) นำความน่าจะเป็นที่มีค่าต่ำสุด 2 ค่ามารวมกันเพื่อสร้างเป็นโหนด (node) ซึ่งถูกสร้างขึ้นทีละคู่ เริ่มต้นจากคู่ของข้อมูลที่มีความน่าจะเป็นน้อยที่สุดไปหาข้อมูลที่มีความน่าจะเป็นมากที่สุด แต่ถ้าหากข้อมูลตัวสุดท้ายไม่สามารถจับคู่ได้คือ เหลือเพียงตัวเดียวก็จะถูกแยกเป็นโหนดอิสระที่มีกิ่งเพียงกิ่งเดียว

(3) สร้างโหนดแม่ (parent node) ของโหนดคู่ที่ได้จากขั้นตอนแรกและจะมีค่าน้ำหนักเท่ากับผลรวมของค่าความน่าจะเป็นของโหนดลูก (child node) ทั้งสอง

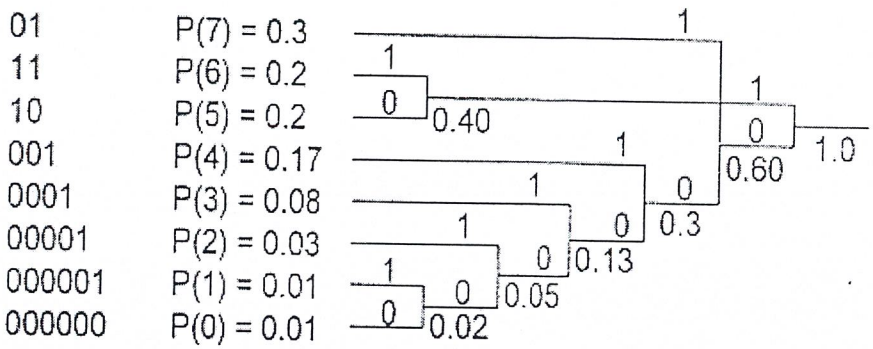
(4) โหนดแม่ที่ได้ก็จะถูกเพิ่มเข้ามาเป็นโหนดอิสระ และโหนดลูกก็จะถูกยกเลิก

(5) โหนดลูกที่ถูกกำหนด ให้เป็นเส้นทางของโหนดแม่โหนดหนึ่ง จะถูกกำหนดให้มีค่า เป็นไบนารี 0 ส่วนโหนดลูกที่เหลือก็จะถูกกำหนดให้มีค่าเป็น 1

(6) ทำซ้ำขั้นตอนทั้งหมดที่ผ่านมาจนกระทั่งเหลือโหนดอิสระเพียงโหนดเดียว ซึ่งโหนดที่เหลือนี้ก็จะถูกกำหนดเป็นโหนดราก (root node)

ตัวอย่างของการสร้างรหัสฮัฟแมน แสดงได้รูปที่ 2.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 แสดงตัวอย่างการเข้ารหัสฮัฟแมน

หลังจากที่ได้รหัสฮัฟแมนแล้ว การเข้ารหัสและการถอดรหัสสามารถทำได้โดยใช้การเปิดตาราง(look-up table) และรหัสที่ได้สามารถที่จะทำการถอดรหัสได้ทันทีโดยไม่ต้องอ้างถึงข้อมูลตัวอื่น ๆ

การเข้ารหัสรันเลนจ์ (Run-Length Coding)

เป็นวิธีการที่ง่ายที่สุดในการเข้ารหัสข้อมูลไบนารีที่มักจะเกิดกลุ่มของข้อมูล 0 และ 1 ดังเช่น ในข้อมูลภาพหรือข้อความสำหรับการส่งแฟกซ์ (facsimile) หรืออาจใช้กับภาพที่จุดภาพที่อยู่ใกล้กัน มีระดับการเปลี่ยนแปลงระดับความเข้มไม่มาก และมีการซ้ำเป็นจำนวนมาก โดยจะเก็บจำนวนของจุดภาพที่ซ้ำกันแทนที่จะเก็บค่าระดับความเข้มของจุดภาพเหล่านั้น

ตัวอย่างการเข้ารหัสรันเลนจ์ แสดงดังรูปที่ 2.30

INPUT :

0 0 0 0 5 5 4 7 7 7 7 7 7 0 0 0 0 0

ENCODING :

4 0 2 5 1 4 7 7 6 0

Result :

20 สัญลักษณ์ ลดลงเหลือ 10 สัญลักษณ์

(อัตราการลดขนาดข้อมูลเท่ากับ 2)

รูปที่ 2.30 แสดงตัวอย่างการเข้ารหัสรันเลนจ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้ารหัสเลขคณิต (Arithmetic Coding)

ในการเข้ารหัสเลขคณิตนั้นข่าวสารต้นแบบจะถูกแทนด้วยช่วงระยะห่างของจำนวนจริงระหว่าง 0 และ 1 โดยข้อมูลแต่ละตัวของข่าวสารจะทำให้ช่วงระยะห่างนั้นแคบลง ซึ่งระยะห่างที่เล็กลงนี้จะทำให้ใช้จำนวนของบิตมากขึ้น ตัวข้อมูลที่ต่อเนื่องกันของข่าวสารจะลดขนาดของระยะห่างตามความน่าจะเป็นของข้อมูล ข่าวสารที่มีความน่าจะเป็นสูงจะทำให้ระยะห่างถูกลดลงได้น้อยกว่าข่าวสารที่มีความน่าจะเป็นต่ำ ดังนั้นจึงทำให้ข่าวสารที่มีความน่าจะเป็นสูงใช้จำนวนบิตในการเข้ารหัสที่น้อยกว่า

วิธีการเข้ารหัสแบบพื้นฐานจะมีขั้นตอนดังนี้

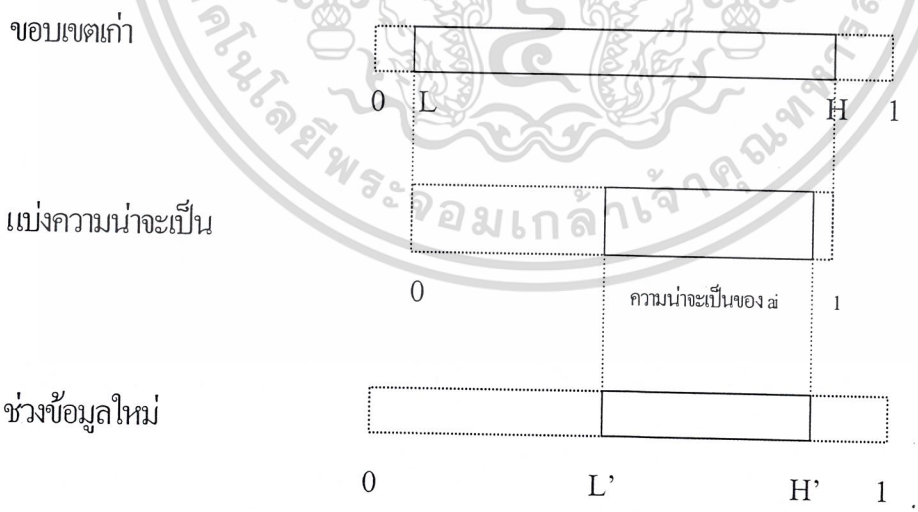
ขั้นตอนที่ 1 จะทำการกำหนดช่วงขอบเขตเริ่มต้น [L,H) ให้เป็น [0,1)

ขั้นตอนที่ 2 ในขั้นตอนนี้จะนำสัญลักษณ์ของข้อมูลมาพิจารณา โดยแบ่งเป็น 2 ขั้นตอนย่อย คือ (แสดงดังรูปที่ 2.31)

1. ทำการแบ่งช่วงขอบเขตปัจจุบันให้เป็นช่วงของขอบเขตย่อย ตามความน่าจะเป็นของสัญลักษณ์
2. เลือกช่วงของขอบเขตย่อยที่สอดคล้องกับสัญลักษณ์ที่พบในข้อมูลแล้วนำช่วงของขอบเขตดังกล่าวมาเป็นช่วงขอบเขตปัจจุบันแทน

กระทำในขั้นตอนที่สองจนกว่าจะสิ้นสุดชุดของข้อมูลที่จะนำเข้ามาเข้ารหัส หรือเมื่อพบอักษรที่แสดงถึงจุดสิ้นสุดของข้อมูล

ขั้นตอนที่ 3 เอาต์พุตที่ได้จากการเข้ารหัสก็ คือ ค่าของช่วงขอบเขตสุดท้ายที่ได้ในขั้นตอนที่ 2 โดยนำค่าขอบเขตดังกล่าวมาแปลงเป็นรหัสไบนารี



รูปที่ 2.31 แสดงการแบ่งช่วงของขอบเขตปัจจุบันบนพื้นฐานของความน่าจะเป็นของสัญลักษณ์ a_i ที่ปรากฏขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนบิตของรหัสไบนารีที่ใช้แทนช่วงขอบเขตสามารถพิจารณาได้จาก ผลคูณของค่าความน่าจะเป็นของแต่ละสัญลักษณ์ให้มีค่าเท่ากับ p ที่เกิดจากลำดับของสัญลักษณ์ข้อมูลจำนวนบิตของรหัสจะเท่ากับ $-\log_2 p$ 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

$$[L + P_C(H-L), L + P_N(H-L)) \qquad (2.98)$$

กระบวนการจะเริ่มจากข้อมูลต้นแบบที่ไม่ได้มีการเรียงลำดับความน่าจะเป็น จากนั้นจะทำการแบ่งช่วงระยะห่างระหว่าง 0 กับ 1 เป็นช่วงย่อๆ ตามความน่าจะเป็นสะสม (cumulative probability) อาทิ เช่น กำหนดให้ข้อมูลต้นแบบคือ 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 0.09 ลงเป็น [0.027,0.036)

4 D จะลดขอบเขตระยะห่างระหว่าง 0.027 กับ0.036 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# สามารถจะเข้ารหัสได้โดยการแทนด้วยช่วงระยะห่างหรือตัวเลขจำนวนใดๆที่อยู่ในช่วง [0.03483,0.0351) ค่าความน่าจะเป็นของการเข้ารหัสของข้อมูลนี้คือ $(0.3)^2 \times (0.1) \times (0.3) \times (0.1) = 0.00027$ จำนวนบิตของข้อมูลที่จะใช้แทนกลุ่มของสัญลักษณ์ข้อมูลนี้จะเท่ากับ $\lceil -\log_2(0.00027) \rceil = \lceil 11.854 \rceil = 12$ บิต

ดังนั้นเลือกข้อมูลในช่วง [0.03483,0.0351) มาแปลงเป็นไบนารี โดยเลือกค่ากลางในช่วงนี้จะได้เท่ากับ 0.034965 นำมาแปลงเป็นเลขไบนารีจะได้ $(0.000010001111 00110\ldots)_2$ แต่จะนำมาเพียง 12 บิตเท่านั้นที่ใช้แทนข้อมูลที่นำมาเข้ารหัส ดังนั้น จะได้รหัสที่แทนข้อมูล AABD# คือ 0000 1000 1111

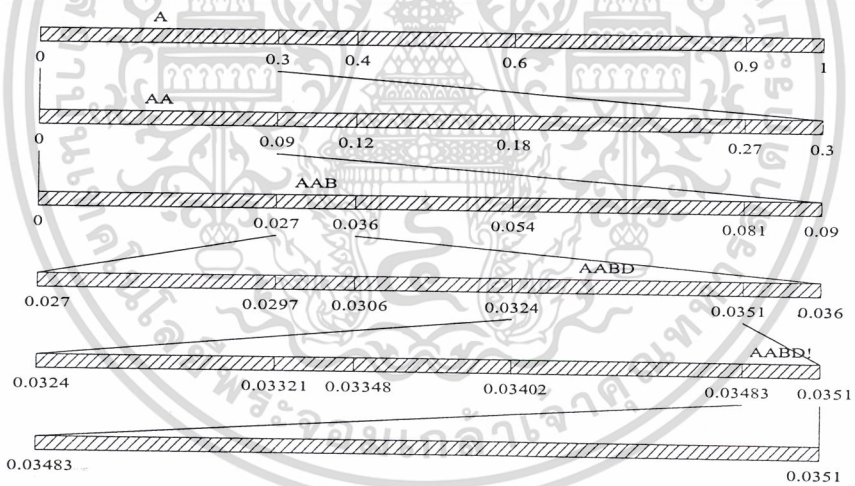
ถ้าเราจะพิจารณากำหนดรหัสให้กับสัญลักษณ์ S โดยกำหนดให้ใช้รหัสตัวละ 3 บิต (ความน่าจะเป็นของการเกิดแต่ละสัญลักษณ์เท่ากับ 1 ดังนั้นสัญลักษณ์จำนวน 5 ตัวจะสามารถกำหนดรหัสได้ต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่ากับ $\lceil -\log_2(1/5) \rceil = \lceil 2.321 \rceil = 3$ บิต) 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 2.32 แสดงการแบ่งขอบเขตตามข้อมูลที่เพิ่มขึ้น

สมมติว่าตัวถอดรหัสได้รับข้อมูลรหัสเป็น 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 [0,0.9) 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)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นช่วงของตัวอักษร B ดังนั้นจะได้อักษรตัวที่สามคือ B จากนั้นจะลดขอบเขตของ [0,0.09) ลงเป็น [0.027,0.036)

4. แบ่งช่วง [0.027,0.036) ตามความน่าจะเป็น แล้วพบว่า i = 0.0350 อยู่ในช่วง [0.0324,0.0351) ซึ่งเป็นช่วงของตัวอักษร D ดังนั้นจะได้อักษรตัวที่สามคือ D จากนั้นจะลดขอบเขตของ [0.027,0.036) ลงเป็น [0.0324,0.0351)

5. แบ่งช่วง [0.0324,0.0351) ตามความน่าจะเป็น แล้วพบว่า i = 0.0350 อยู่ในช่วง [0.03483,0.0351) ซึ่งเป็นช่วงของตัวอักษร # ดังนั้นจะได้อักษรตัวที่สามคือ # จากนั้นจะลดขอบเขตของ [0.0324,0.0351) ลงเป็น [0.03483,0.0351)

ในทางปฏิบัติแล้วมีอยู่หลายปัจจัยที่ทำให้การเข้ารหัสและถอดรหัสเลขคณิตมีความซับซ้อนกว่าการเข้ารหัสฮัฟแมน แต่ก็จะสามารถลดข้อมูลที่มากกว่า

การเข้ารหัสตามระดับของบิต (Bit Plane Coding)

สมมติว่าแต่ละค่าของจุดภาพในภาพระดับเทาภาพหนึ่งถูกแทนด้วยข้อมูล k บิต ดังนั้นเราสามารถสร้างภาพไบนารีจำนวน k ภาพ ซึ่งบิตแต่ละตำแหน่งจะสร้างได้ 1 ภาพ โดยภาพที่ได้มาจากบิตตำแหน่งที่มีนัยสำคัญสูงกว่าจะประกอบด้วยข่าวสารที่สำคัญกว่า และจะมีความซ้ำซ้อนของข้อมูลที่มากกว่า ดังนั้นจึงทำให้ง่ายต่อการลดข้อมูล วิธีการเข้ารหัสตามระดับของบิตจะอาศัยค่าระดับความเทาของภาพในการเพิ่มความสัมพันธ์ (correlation) ของจุดภาพ จากนั้นจึงทำการเข้ารหัสแบบรันเลนจ์หรือการเข้ารหัสเลขคณิตมาใช้ในการลดข้อมูล

2.5.3 การลดข้อมูลแบบยอมให้มีการสูญเสียได้

การลดข้อมูลแบบที่มีการสูญเสียจะทำให้เกิดความเพี้ยนของข้อมูลขึ้น แต่ก็มีอัตราการลดข้อมูลที่สูงกว่าการลดข้อมูลแบบที่ไม่มีการสูญเสียมาก ซึ่งมีอยู่หลายวิธีการดังเช่นต่อไปนี้

การควอนไทซ์แบบสเกลาร์ (Scalar Quantization , SQ)

การควอนไทซ์แบบสเกลาร์เป็นวิธีการที่ง่ายที่สุดของการลดข้อมูลแบบที่มีการสูญเสียโดยการควอนไทซ์จะทำการตรวจสอบข้อมูลอินพุตที่ได้รับและเลือกค่าประมาณที่ได้จากฐานข้อมูลที่กำหนดไว้ล่วงหน้าให้ใกล้เคียงที่สุด โดยปกติแล้วข้อมูลอินพุตจะอยู่ในรูปอนาลอกหรือสัญญาณต่อเนื่อง ส่วนเอาต์พุตที่ได้จะอยู่ในรูปดิจิตอล โดยสามารถนิยามการควอนไทซ์แบบ สเกลาร์ได้ว่าเป็นการส่งผ่าน (Mapping) $Q: R \to C$ เมื่อ R คือค่าจำนวนจริง และ $C = \{y_i; i = 1,2,\ldots,N\}$ เป็นชุดของข้อมูลเอาต์พุต y_i หรือตัวเก็บรหัส (codebook) ที่มีขนาดเป็น N

ในการสร้างตัวควอนไทซ์ขนาด N จุด จะต้องทำการแบ่งเส้นจริง R ออกเป็น N ส่วนเรียกว่า $R_i; i = 1,2,\ldots,N$ และค่าของส่วนที่ i สามารถหาได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_i = \{x \in R; Q(x) = y_i\} \equiv Q^{-1}(y_i) \tag{2.99}$$

โดยที่แต่ละส่วนจะมีคุณสมบัติดังนี้

$$\bigcup_{i=1}^{N} R_i = R \text{ และ } R_i \cap R_j = \phi \text{ เมื่อ } i \neq j \tag{2.100}$$

การควอนไทซ์จะแบ่งระดับหรือย่านของข้อมูลออกเป็นช่วง ๆ โดยที่แต่ละช่วงจะแทนระดับของข้อมูลที่เป็นค่าเดี่ยว ๆ ซึ่งสามารถแบ่งได้เป็น 2 ประเภท คือ

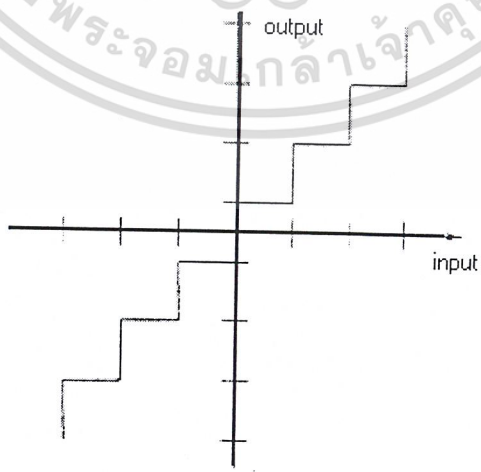
1. *การควอนไทซ์แบบยูนิฟอร์ม (Uniform Quantization)* เป็นการควอนไทซ์ที่มีรูปแบบที่ง่ายที่สุด โดยการแบ่งย่านของข้อมูลออกเป็นช่วงเท่า ๆ กัน ดังแสดงในรูปที่ 2.33 ในการออกแบบตัวควอนไทซ์จะมี d_i เป็นระดับการตัดสินใจ (decision level) และ r_i เป็นระดับของข้อมูลผลลัพธ์(reconstruction) เมื่อ $i = 0,1,2,\ldots,L$ โดยที่

$$r_i = \frac{d_i + d_{i+1}}{2} \tag{2.101}$$

และ จะได้ช่วงห่างของการควอนไทซ์ (Δ) เป็น

$$\Delta = d_{i+1} - d_i \tag{2.102}$$

ดังนั้น จะเห็นได้ว่าการควอนไทซ์แบบยูนิฟอร์มมีค่าของความแตกต่างที่เท่ากันตลอดย่านของข้อมูลอินพุต



รูปที่ 2.33 แสดงระดับของการควอนไทซ์แบบยูนิฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. *การควอนไทซ์แบบนอน-ยูนิฟอร์ม (Non - Uniform Quantization)* เป็นการออกแบบการควอนไทซ์ให้มีความเพี้ยนเกิดขึ้นน้อยที่สุด โดยอัลกอริทึม Lloyd-Max Quantizer ที่ได้ถูกพัฒนาขึ้นโดย Lloyd และ Max ซึ่งได้วิเคราะห์สัญญาณรบกวนจากการควอนไทซ์โดยใช้ความผิดพลาดกำลังสองเฉลี่ย (mean square error, MSE) และได้พยายามลดสัญญาณรบกวนนี้ลงโดยกำหนดความน่าจะเป็นของความหนาแน่นสัญญาณในช่วงหนึ่งมีค่าไม่คงที่ โดยที่ความเพี้ยนทั้งหมดที่เกิดขึ้นจากการควอนไทซ์หาได้จากสมการที่ (2.103)

$$E = \sum_{k=0}^{N} \int_{d_k}^{d_{k+1}} (y - r_k)^2 \, p(y) dy \qquad (2.103)$$

เมื่อ $p(y)$ คือฟังก์ชันความน่าจะเป็นของข้อมูล y
และกำหนดให้อนุพันธ์ของค่าความเพี้ยนเมื่อเทียบกับระดับ r_i และช่วง d_i เท่ากับศูนย์ทำให้ได้

$$d_i = \frac{r_i + r_{i-1}}{2} \qquad (2.104)$$

และ

$$r_i = \frac{\int_{d_i}^{d_{i+1}} y p(y) dy}{\int_{d_i}^{d_{i+1}} p(y) dy} \qquad (2.105)$$

การเข้ารหัสแบบทำนาย (Lossy Predictive Coding)

ในการเข้ารหัสแบบทำนายโดยทั่วไป ความสัมพันธ์ระหว่างจุดภาพที่อยู่ใกล้เคียงกันจะถูกนำมาใช้ในการทำนายค่าของแต่ละจุดภาพ วิธีหนึ่งในการลดข้อมูลที่นำเอาการเข้ารหัสแบบทำนายมาใช้คือการมอดูเลตแบบรหัสพัลส์เชิงอนุพันธ์ (differential pulse code modulation ,DPCM) โดยจะนำเอาค่าความผิดพลาดจากการทำนายที่ได้มาจากการลบค่าการทำนายออกจากค่าจริงของจุดภาพมาทำการควอนไทซ์แบบสเกลาร์ซึ่งจะทำให้เกิดความผิดเพี้ยนของข้อมูลขึ้น แต่ก็จะสามารถลดข้อมูลได้ในอัตราที่มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้ารหัสโดยการแปลง (Transform Coding)

การลดข้อมูลด้วยการเข้ารหัสการแปลงจะทำการแปลง ข้อมูลอินพุตที่อยู่ในรูปของสปาเชียลโดเมน ให้อยู่ในรูปของ สเปกตรัม หรือ โดเมนความถี่ โดยใช้วิธีการแปลงแบบต่าง ๆ เช่น การแปลงฟูริเยร์ (Fourier Transform) ในการแปลงข้อมูลที่อยู่ในรูปของ สปาเชียลโดเมน ให้อยู่ในรูปสัมประสิทธิ์ของพลังงานความถี่ จะได้ที่ความถี่ต่ำๆ จะมีพลังงานสูง และที่ความถี่สูง ๆ พลังงานจะลดลงไป สำหรับข้อมูลภาพโดยทั่ว ๆ ไปนั้นจะประกอบด้วยส่วนที่เป็นความถี่สูง คือส่วนที่เป็นรายละเอียดหรือขอบภายในภาพ และส่วนที่เป็นความถี่ต่ำภาพคือข้อมูลส่วนที่เป็นพื้นของภาพ ซึ่งจำนวนบิตที่ใช้ในการเข้ารหัสของแต่ละช่วงความถี่จะไม่เท่ากัน ดังนั้นเมื่อต้องการลดข้อมูลให้ได้มาก ค่าของพลังงานความถี่สูงจะถูกตัดทิ้งไปเป็นส่วนใหญ่ ทำให้รายละเอียดส่วนที่เป็นขอบในภาพขาดหายไป ภาพที่ได้จะเบลอ (Blur) ขาดความคมชัด

การแปลงที่ใช้ในการลดข้อมูลภาพมีอยู่หลายวิธี เช่น การแปลงฟูริเยร์แบบเร็ว (Fast Fourier Transform) , การแปลงวอล์ช - ฮาดามาร์ด (Walsh - Hadamard Transform) , การดัดแปลงดิสครีตโคซายน์ (Discrete Cosine Transform) , การแปลงดิสครีตซายน์ (Discrete Sine Transform) ,การแปลงฮาร์ (Haar Transform) และการแปลงคาร์ฮูเนนเลิฟ (Karhunen Loeve Transform) เป็นต้น ซึ่งแต่ละวิธีจะมีข้อดีและข้อเสียแตกต่างกันไป แต่ที่วิธีนิยมใช้ในการลดข้อมูลกันมากที่สุดคือการแปลงดิสครีตโคซายน์ (DCT) ตัวอย่างของการลดข้อมูลที่ใช้การแปลงดิสครีตโคซายน์ คือ การลดข้อมูลแบบ JPEG ซึ่งปัจจุบันวิธีการนี้เป็นวิธีการมาตรฐานที่ใช้ในการลดข้อมูล

การเข้ารหัสแบนด์ย่อย (Subband Coding)

ในการเข้ารหัสแบนด์ย่อย ภาพต้นแบบจะถูกกรองตามช่วงของความถี่เพื่อสร้างเป็นภาพ ภาพย่อย ซึ่งภาพเหล่านี้ถูกเรียกว่าภาพแบนด์ย่อย (subbands) เนื่องจากแต่ละภาพแบนด์ย่อยนี้ได้ถูกลดความกว้างแถบ (bandwidth) เมื่อเปรียบเทียบกับภาพต้นแบบ ทำให้สามารถที่จะทำการลดจำนวนการสุ่ม (downsampling) ลงได้ ซึ่งสามารถทำได้ด้วยกระบวนการในการกรอง (filtering) ส่วนการลดจำนวนการสุ่มนี้เรียกว่าเป็นขั้นตอนการวิเคราะห์ (analysis stage) หลังจากนั้นภาพแบนด์ย่อยที่ได้จะถูกเข้ารหัสด้วยตัวเข้ารหัสต่าง ๆ โดยที่อัตราบิต (bit rate) ต่าง ๆ หรือแม้แต่วิธีการเข้ารหัสต่าง ๆ กันก็สามารถนำมาใช้กับภาพแต่ละแบนด์ย่อยได้

ในการสร้างภาพกลับคืนมาอีกครั้งสามารถทำได้โดยการเพิ่มจำนวนการสุ่ม (up sampling) ของภาพแบนด์ย่อยที่ได้ถอดรหัสแล้ว จากนั้นนำมาผ่านตัวกรอง แล้วรวมภาพที่สร้างกลับคืนมาแต่ละแบนด์ย่อยเข้าด้วยกัน กระบวนการนี้เรียกว่าขั้นตอนการสังเคราะห์ (synthesis stage)

จะสังเกตได้ว่าการสร้างภาพแบนด์ย่อยจะไม่ทำให้เกิดการลดลงของข้อมูล เนื่องจากจำนวนของการสุ่ม (sample) ของทุกแบนด์ย่อยรวมกันแล้วยังคงเท่ากันกับภาพต้นแบบ แต่จะมีข้อดีในแง่ที่แต่ละภาพแบนด์ย่อยนั้นสามารถที่จะทำการเข้ารหัสได้ด้วยประสิทธิภาพที่ดีกว่าภาพต้นแบบเดิมซึ่งการเข้ารหัสแบนด์ย่อยอาจใช้การแปลงเวฟเล็ตเพื่อช่วยในการแยกแบนด์ย่อย ออกจากภาพต้นแบบได้ ซึ่งจะเป็นหลักการพื้นฐานสำหรับการแปลงเวฟเล็ต เพื่อแปลงสัมประสิทธิ์ในมิติหนึ่งสู่อีกมิติหนึ่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 เกณฑ์การวัดความเหมือนจริงของภาพ (Image Fidelity)

ในการลดข้อมูลภาพนั้น จะมีข้อมูลส่วนหนึ่งที่เกิดผิดพลาดหรือสูญเสียไป ข้อผิดพลาดที่เกิดขึ้นนี้จะมีผลในตอนที่สร้างภาพกลับคืนมา (reconstruction) และค่าความผิดพลาดนี้จะอยู่ในช่วงหนึ่งที่สามารถยอมรับได้ ดังนั้นเกณฑ์การวัดความเหมือนจริงของภาพสามารถนำมาใช้ในการวัดประสิทธิภาพของระบบได้ ตัวอย่างเกณฑ์ที่นิยมใช้ในการวัดคุณภาพของภาพคือ ค่า root-mean-square (rms) ของความผิดพลาดระหว่างข้อมูลภาพอินพุตและข้อมูลภาพเอาต์พุต นอกจากนี้ยังมีค่า rms ของอัตราส่วนของสัญญาณต่อสัญญาณรบกวนของภาพเอาต์พุต (Signal to Noise Ratio : SNR) เมื่อกำหนดให้ข้อมูลภาพอินพุตประกอบด้วยอาร์เรย์ขนาด N×N ของจุดภาพ $f(x,y)$ โดย x และ y มีค่าเป็น 0,1,...,N-1 และแต่ละจุดภาพมีค่าของระดับสีเทาที่เป็นไปได้คือ 2^m เมื่อ m เป็นจำนวนบิตของระดับสีเทา

สำหรับทุกค่าของ x และ y ในช่วง 0,1,...,N-1 ค่าความผิดพลาดระหว่างจุดภาพอินพุตและเอาต์-พุตคือ

$$e(x,y) = g(x,y) - f(x,y) \tag{2.106}$$

เมื่อ $e(x,y)$ คือ ค่าความผิดพลาดที่เกิดขึ้น ณ จุด x, y ใดๆ

$f(x,y)$ คือ ค่าภาพอินพุต ณ จุด x, y ใดๆ

$g(x,y)$ คือ ค่าภาพเอาต์พุต ณ จุด x, y ใดๆ

ค่าเฉลี่ยของความผิดพลาดกำลังสองของภาพ (mean square error) คือ

$$\begin{aligned} e_{ms} &= \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} e^2(x,y) \\ &= \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\left[g(x,y) - f(x,y)\right]^2 \end{aligned} \tag{2.107}$$

ดังนั้นค่า rms ของความผิดพลาดจึงสามารถเขียนได้ดังนี้

$$e_{rms} = [e_{rm}]^{1/2} \tag{2.108}$$

ค่า root mean square error เป็นค่าที่ใช้ในการวัดความแตกต่างของข้อมูลอินพุตกับข้อมูลเอาต์พุต แต่เมื่อพิจารณาขนาดของข้อมูลเอาต์พุตต่อขนาดของสัญญาณรบกวน (noise) ก็จะได้เป็นค่า SNR เมื่อกำหนดให้สัญญาณภาพเอาต์พุตแต่ละจุดประกอบด้วยสัญญาณอินพุตบวกด้วยค่าสัญญาณรบกวน นั้นคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$g(x,y) = f(x,y) + e(x,y) \tag{2.109}$$

ดังนั้น ค่าเฉลี่ยกำลังสองของอัตราส่วนของสัญญาณต่อสัญญาณรบกวนของข้อมูลภาพเอาต์พุต สามารถหาได้โดยค่าเฉลี่ยของสัญญาณอินพุตกำลังสอง หารด้วยค่าเฉลี่ยของสัญญาณรบกวนกำลังสอง ของข้อมูลภาพทั้งหมด ซึ่งสามารถเขียนได้ดังนี้

$$SNR_{ms} = \frac{\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} g^2(x,y)}{\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} e^2(x,y)} \tag{2.110}$$

ค่า rms ของ SNR จึงสามารถเขียนได้ดังต่อไปนี้

$$SNR_{rms} = \left(\frac{\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} g^2(x,y)}{\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} [g(x,y) - f(x,y)]^2} \right)^{1/2} \tag{2.111}$$

โดยที่เทอมส่วนของสมการข้างบนเป็นสมการของสัญญาณรบกวนที่อยู่ในรูปของผลต่างระหว่างข้อมูลอินพุตกับข้อมูลเอาต์พุต

จากวิธีการที่ใช้ในการวัดความเหมือนจริงของภาพที่กล่าวมาแล้วข้างต้น ไม่สามารถที่จะบ่งบอกหรือใช้เป็นเกณฑ์ในการพิจารณาได้แต่เพียงอย่างเดียว ในกรณีของภาพเอาต์พุตที่ได้จากการประมวลผลหรือรับส่งสัญญาณ โดยมีสายตาของมนุษย์เป็นตัวรับภาพ ซึ่งระบบการมองเห็นของสายตาจะไวต่อความเข้มแสงในลักษณะของล็อก (logarithmic) ดังนั้น ความผิดพลาดในบริเวณที่เป็นที่มืดของภาพจะเห็นได้ชัดเจนกว่าความผิดพลาดที่อยู่ในบริเวณที่สว่าง และระบบการมองเห็นยังไวต่อการเปลี่ยนแปลงอย่างทันทีทันใดของระดับสีเทาด้วย ความผิดพลาดที่อยู่บนขอบหรือใกล้ ๆ ขอบของวัตถุจะมีผลต่อการมองเห็นมากกว่าความผิดพลาดที่อยู่ในโครงสร้างที่เป็นฉากหลังของภาพ ด้วยเหตุนี้เอง ถึงแม้ว่าภาพจะมีค่าของ rms ในความผิดพลาดที่เท่ากัน แต่อาจจะปรากฏความแตกต่างของคุณภาพของการมองเห็นที่แตกต่างกันได้

2.5.5 อัตราบิต

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{bpp} = \frac{\text{จำนวนบิตทั้งหมดของภาพที่ลดข้อมูล}}{\text{จำนวนจุดภาพของภาพต้นแบบทั้งหมด}} \quad (2.112)$$

2.5.6 อัตราการลดข้อมูล

อัตราการลดข้อมูลคืออัตราส่วนระหว่างจำนวนบิตข้อมูลที่ใช้แทนภาพต้นแบบ และจำนวนบิตข้อมูลที่ใช้แทนภาพที่ถูกทำการลดข้อมูล โดยสามารถหาได้ดังสมการ

$$\text{อัตราการลดข้อมูล} = \frac{\text{จำนวนบิตทั้งหมดของภาพต้นแบบ}}{\text{จำนวนบิตทั้งหมดของภาพที่ลดข้อมูล}} \quad (2.113)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

ในการบีบอัดข้อมูลภาพโดยทั่วๆไปจะมีขบวนการที่คล้ายคลึงกัน โดยสามารถที่จะแยกเป็นสองแนวทางใหญ่ๆ คือ ดาต้าโมเดลลิ่ง (data modeling) และการเข้ารหัส (code packing) ในขั้นตอนแรกคือ ดาต้าโมเดลลิ่ง จะนำเอาข้อมูลข่าวสารมาเปลี่ยนรูปแบบใหม่ซึ่งมีลักษณะที่สัมพันธ์กับข้อมูลเดิม เพื่อที่จะสามารถใช้ลักษณะเฉพาะของโมเดลต่างๆ เพื่อกำจัด หรือ ลดความซ้ำซ้อนของข้อมูลบางประการลงได้ โดยอาศัยเทคนิคต่างๆ หลายรูปแบบดังเช่น รูปแบบการแปลงข้อมูลภาพ (image transform) ซึ่งอาจใช้การแปลงฟูเรียร์ , การแปลงของดิสครีตโคซายน์ (discrete cosine transform : DCT) ฯลฯ และอีกรูปแบบหนึ่งที่มีการนำมาใช้ คือ การแปลงเวฟเล็ต (wavelet transform) ซึ่งเป็นวิธีการแปลงข้อมูลภาพ ที่นำมาใช้ในการลดขนาดข้อมูลภาพของปริญญานิพนธ์นี้

ส่วนในขั้นตอนของการเข้ารหัส จะทำการนำข้อมูลที่ได้จากการทำโมเดลลิ่ง มาจัดระดับ และเข้ารหัสข้อมูลใหม่เพื่อให้ข้อมูลมีคุณลักษณะที่เหมาะสมในการนำไปใช้งาน โดยวิธีการโค้ดแพ็กกิ้งที่นำมาใช้ในปริญญานิพนธ์นี้จะแยกเป็นสองขั้นตอนการด้วยกัน คือขั้นตอนการแรก จะเป็นการนำโมเดลลิ่งที่ได้มาทำการจัดกลุ่มของสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ต เพื่อที่จะให้ได้สัญลักษณ์ของข้อมูล ที่มาแทนค่าสัมประสิทธิ์ที่ได้จากการทำโมเดลลิ่ง โดยอาศัยวิธีการจัดระดับประมาณค่าแบบซัคเซส์ซีฟ (successive approximation quantization :SAQ) ซึ่งเป็นขบวนการเปลี่ยนหรือแปลงสัมประสิทธิ์เวฟเล็ตไปเป็นชุดบิตข้อมูล โดยวิธีการนี้จะคล้ายกับการเข้ารหัสแบบบิตเพลน (bit plane encoding) ในหัวข้อที่ 2.5.2 สัมประสิทธิ์เวฟเล็ตถูกจัดค่าตามระดับบิต โดยระดับบิตจะเริ่มจากบิตที่มีนัยสำคัญก่อน โดยวิธีนี้ถูกคิดค้นขึ้นมาโดยเซฟิโร่ (Jerome M. Shapiro) และเรียกวิธีการดังกว่าว่า เอ็มเบดอิด ซีโร่ทรี (Embedded zerotree) ซึ่งผลที่ได้จะเป็นการจัดกลุ่มของข้อมูลที่เป็นค่าสัมประสิทธิ์ให้เป็นสัญลักษณ์ของข้อมูลเพื่อให้เหมาะสมที่จะนำไปเข้ารหัสลดขนาดข้อมูลแบบไม่มีการสูญเสียในขั้นตอนต่อไป

ขั้นตอนสุดท้ายของการเข้ารหัสคือ การนำเอาลำดับของข้อมูลที่ได้จากการจัดกลุ่มของสัมประสิทธิ์ในขั้นตอนแรกของการเข้ารหัส มาทำการ เข้ารหัสลดขนาดข้อมูลแบบไม่มีการสูญเสีย โดยจะนำเอาการเข้ารหัสเลขคณิต มาใช้งาน

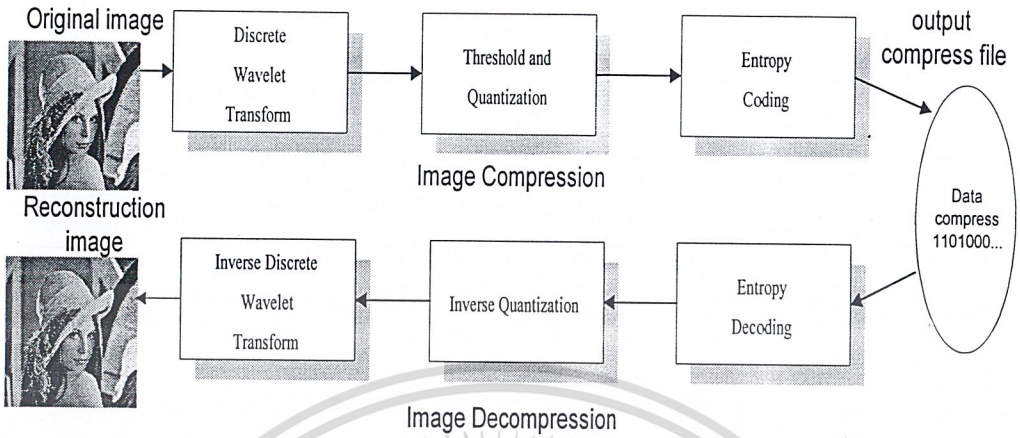
ขั้นตอนของการลดขนาดข้อมูลภาพด้วยวิธีการใช้การแปลงเวฟเล็ต จะแสดงในรูปที่ 3.1

3.1 การคำนวณการแปลงเวฟเล็ตกับข้อมูลภาพ

จากสมการ 2.52

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงขั้นตอนการบีบอัดข้อมูลภาพโดยใช้การแปลงเวฟเล็ต

และจากสมการ 2.53

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

นำทั้งสองสมการดังกล่าวมาใช้สำหรับการแปลงเวฟเล็ตแบบไม่ต่อเนื่องโดยค่า สัมประสิทธิ์ของการกรองความถี่ต่ำ (h_0) และการกรองความถี่สูง (h_1) คือสัมประสิทธิ์ของเวฟเล็ตฟังก์ชัน ซึ่งจะแสดงตัวอย่างการคำนวณดังตัวอย่างที่ 1 และ 2 โดยจะใช้ฟังก์ชันฮาร์เป็นเวฟเล็ตฟังก์ชันเพื่อความสะดวกในการคำนวณ

พิจารณา สัมประสิทธิ์ของการกรองความถี่ต่ำ (h_0) และการกรองความถี่สูง (h_1) ของฮาร์ฟังก์ชัน[7] คือ

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

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

จะได้ว่า

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

ดังนั้น

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

และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ในทำนองเดียวกัน

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

และ

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

ดังนั้นจะได้ผลการแปลงเวฟเล็ต คือ $\{\frac{16}{\sqrt{2}}, \frac{8}{\sqrt{2}}, \frac{2}{\sqrt{2}}, \frac{-2}{\sqrt{2}}\}$ ที่อยู่ในรูปแบบสมการ (2.29) ในบทที่ 2 และ เมื่อทำการแปลงเวฟเล็ต อีกจะได้ว่า

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

และ

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

ผลสุดท้ายจะได้สัมประสิทธิ์เวฟเล็ตคือ $\{12,4,\frac{2}{\sqrt{2}},-\frac{2}{\sqrt{2}}\}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 สมมติข้อมูลภาพมีขนาด 8 จุดภาพ คือ 1, 0, -3, 2, 1, 0, 1, และ 2 ข้อมูลภาพดังกล่าวสามารถแสดงการแปลงเวฟเล็ตแบบไม่ต่อเนื่องได้ดังนี้

c_3 | 1 | 0 | -3 | 2 | 1 | 0 | 1 | 2 |

d_2 | $\frac{1}{\sqrt{2}}$ | $-\frac{5}{\sqrt{2}}$ | $\frac{1}{\sqrt{2}}$ | $-\frac{1}{\sqrt{2}}$ |

c_2 | $\frac{1}{\sqrt{2}}$ | $-\frac{1}{\sqrt{2}}$ | $\frac{1}{\sqrt{2}}$ | $\frac{3}{\sqrt{2}}$ |

d_1 | 1 | -1 |

c_1 | 0 | 2 |

d_0 | $-\sqrt{2}$ |

c_0 | $\sqrt{2}$ |

จะได้สัมประสิทธิ์ของข้อมูลหลังการแปลงเวฟเล็ตคือ $\{\sqrt{2},-\sqrt{2},1,-1,\frac{1}{\sqrt{2}},-\frac{5}{\sqrt{2}},\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}\}$

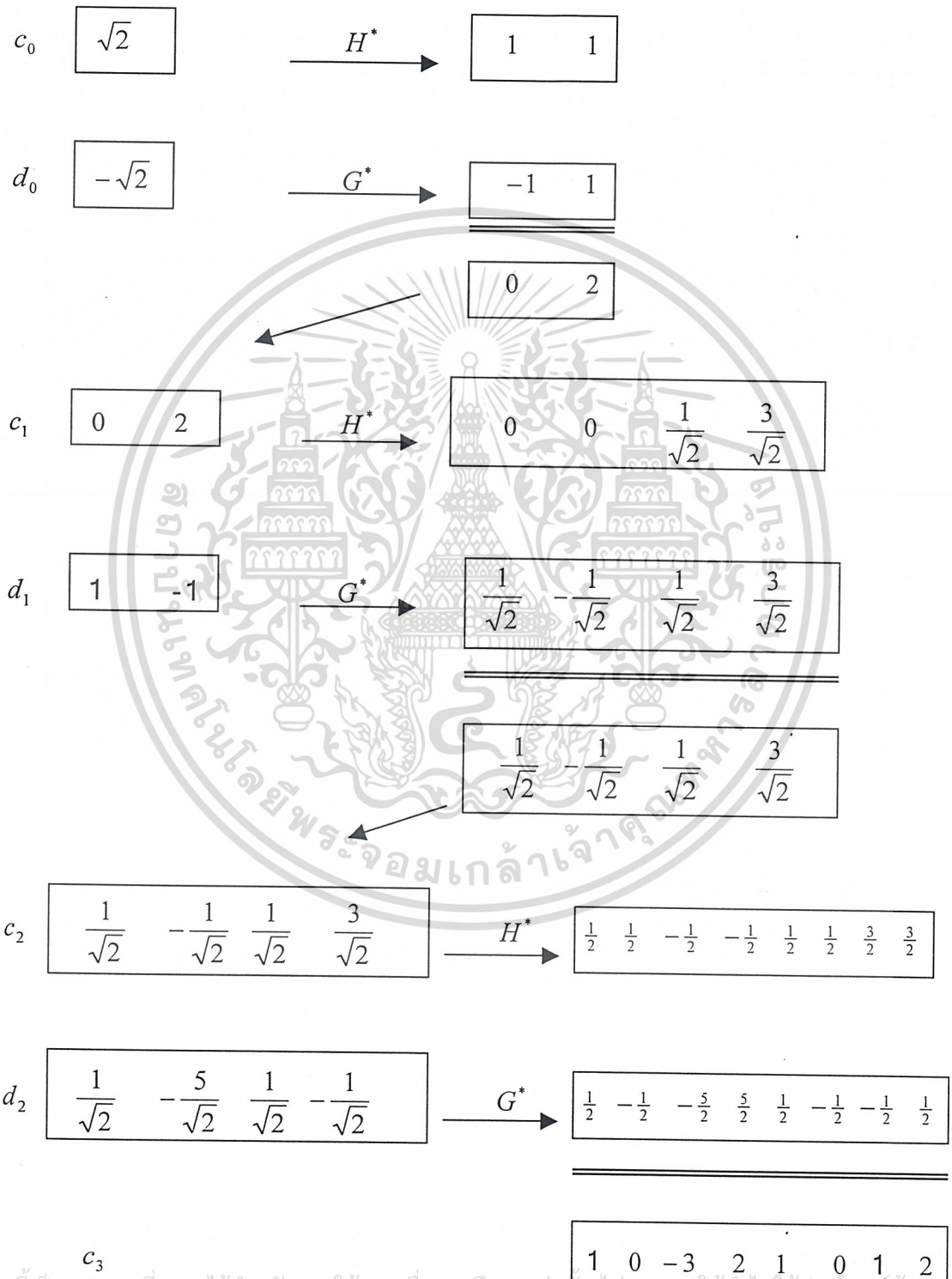
และจากสมการที่ 2.75 แสดงถึงลักษณะการสังเคราะห์สัญญาณกลับ จะตรงกันข้ามกับการวิเคราะห์สัญญาณด้วยการแยกช่วงความถี่ออกจากกันทีละสองช่วง โดยทำการเพิ่มอัตราการชักค่าตัวอย่างให้มีค่าเป็นสองเท่าจากเดิมก่อน ซึ่งจะทำการแทรกค่าศูนย์ระหว่างเทอมของสัญญาณ

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

สามารถแสดงตัวอย่างการสังเคราะห์สัญญาณกลับจากสัมประสิทธิ์ของข้อมูลที่ได้จากการแปลงเวฟเล็ตจากตัวอย่างที่ 2 ในตัวอย่างที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 3 แสดงการสังเคราะห์สัญญาณกลับจากสัมประสิทธิ์ของข้อมูลหลังการแปลงเวฟเล็ตคือ $\{\sqrt{2},-\sqrt{2},1,-1,\frac{1}{\sqrt{2}},-\frac{5}{\sqrt{2}},\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}\}$ ด้วยฮาร์ฟังก์ชัน



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบการแปลงเวฟเล็ต

จากการคำนวณการแปลงเวฟเล็ตที่กล่าวมา จะสามารถออกแบบโปรแกรมการคำนวณการแปลงเวฟเล็ตได้ดังคำสั่งจำลองที่ 3.1 และ 3.2 สำหรับการนำภาพมาแปลงเวฟเล็ต ด้วยวิธีมาตรฐาน และคำสั่งจำลองที่ 3.3 และ 3.4 สำหรับการนำภาพมาแปลงเวฟเล็ตด้วยวิธีไม่มาตรฐาน

สำหรับการนำเอาเวฟเล็ตมาแปลงข้อมูลภาพนั้นจะอาศัยวิธีการที่คล้ายกับการเข้ารหัสสับแบนด์ โดยจะมีลักษณะของการแปลงภาพที่เป็นแบบไม่มาตรฐาน[7] ดังนั้นในปริญญานิพนธ์นี้จะเลือกนำเอาหลักการแปลงข้อมูลภาพด้วยการแปลงเวฟเล็ตโดยใช้วิธีแบบไม่มาตรฐานเป็นหลัก และเขียนเป็นผังงานสำหรับการแปลงภาพแบบไม่มาตรฐานได้ใน ผังงานที่ 3.1 และ3.2

คำสั่งจำลองที่ 3.1 การนำภาพมาแปลงเวฟเล็ตด้วยวิธีมาตรฐาน

```
PROCEDURE StandardDecomposition
BEGIN
        FOR Row : 1 to h DO
                Decomposition( C[ Row , 1...w ]);
        END FOR
        FOR Col : 1 to w DO
                Decomposition( C[1...h , Col ]);
        END FOR
END.
PROCEDURE DecompositionStep
BEGIN
        FOR i : 1 to h/2 DO
                C'[i] = ( C[2i-1] + C[2i] ) / √2 ;
                C'[h/2 + i ] = ( C[2i-1] – C[2i] ) / √2 ;
        END FOR
        C = C';
END.
PROCEDURE Decomposition
BEGIN
        C = C / √h  ; %% normalize input coefficient
        WHILE h > 1 DO
                DecompositionStep( C[1...h]);
                h= h/2;
        END WHILE
END.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งจำลองที่ 3.2 การนำสัมประสิทธิ์เวฟเล็ตของภาพที่ได้จากการแปลงเวฟเล็ตแบบมาตรฐานมาแปลงกลับ

```
PROCEDURE InverseStandardDecomposition
BEGIN

        FOR Col : 1 to w DO
                InverseDecomposition( C[1...h, Col ]);
        END FOR

        FOR Row : 1 to h DO
                InverseDecomposition( C[Row,1...w]);
        END FOR
END.

PROCEDURE InverseDecompositionStep %% To same InverseStandardDecomposition
BEGIN
        FOR i = 1 to h/2 DO
                C' = C;
                C[2i-1] = (C'[i] + C'[h/2 + i]) / √2 ;
                C[2i] = ( C'[i] – C'[h/2 + i]) / √2 ;
        END FOR
END.

PROCEDURE InversDecomposition  %% To same InverseStandardDecomposition
BEGIN
        WHILE 1 < h DO
                InverseDecompositionStep( C[1...h]);
                h= 2h ;
        END WHILE
        C = √h · C
END.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งจำลองที่ 3.3 การนำภาพมาแปลงเวฟเล็ตด้วยวิธีไม่มาตรฐาน

```
PROCEDURE NonstandardDecomposition
BEGIN
        WHILE h>1 DO
                FOR Row : 1 to h DO
                        Decomposition( C[ Row , 1...h]);
                END FOR
                FOR Col : 1 to h DO
                        Decomposition( C[1...h , Col ]);
                END FOR
                h= h/2;
        END WHILE
END.

PROCEDURE DecompositionStep %%same in standard decomposition procedure
BEGIN
        FOR i : 1 to h/2 DO
                C'[i] = ( C[2i-1] + C[2i] ) / √2 ;
                C'[h/2 + i ] = ( C[2i-1] – C[2i] ) / √2 ;
        END FOR
        C = C';
END.

PROCEDURE Decomposition %%same in standard decomposition procedure

BEGIN
        C = C / √h ; %% normalize input coefficient
        WHILE h > 1 DO
                DecompositionStep( C[1...h]);
                h= h/2;
        END WHILE
END.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งจำลองที่ 3.4 การนำสัมประสิทธิ์เวฟเล็ตของภาพที่ได้จากการแปลงเวฟเล็ตแบบไม่มาตรฐานมาแปลงกลับ

```
PROCEDURE InverseNonstandardDecomposition
BEGIN
        WHILE 1 <h Do
                h= 2h;
                FOR Col : 1 to h DO
                        InverseDecomposition( C[1...h, Col]);
                END FOR
                FOR Row : 1 to h DO
                        InverseDecomposition( C[row,1...h]);
                END FOR
        END WHILE
        C = Ch;
END.

PROCEDURE InverseDecompositionStep
BEGIN
        FOR i = 1 to h/2 DO
                C' = C;
                C[2i-1] = (C'[i] + C'[h/2 + i] /√2  ;
                C[2i] = [ C'[i] – C'[h/2 + i]) /√2 ;
        END FOR
END.

PROCEDURE InversDecomposition
        WHILE 1 < h DO
                InverseDecompositionStep( C[1...h]);
                h= 2h ;
        END WHILE
        C = √h · C
END.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการเปลี่ยนฟังก์ชันแม่ของการแปลงเวฟเล็ตเราก็จะเปลี่ยนตรงสัมประสิทธิ์ของการกรองความถี่ต่ำ และความถี่สูง เท่านั้น ซึ่งลักษณะของการทำคำสั่งของโปรแกรมยังคงลักษณะเช่นเดิม

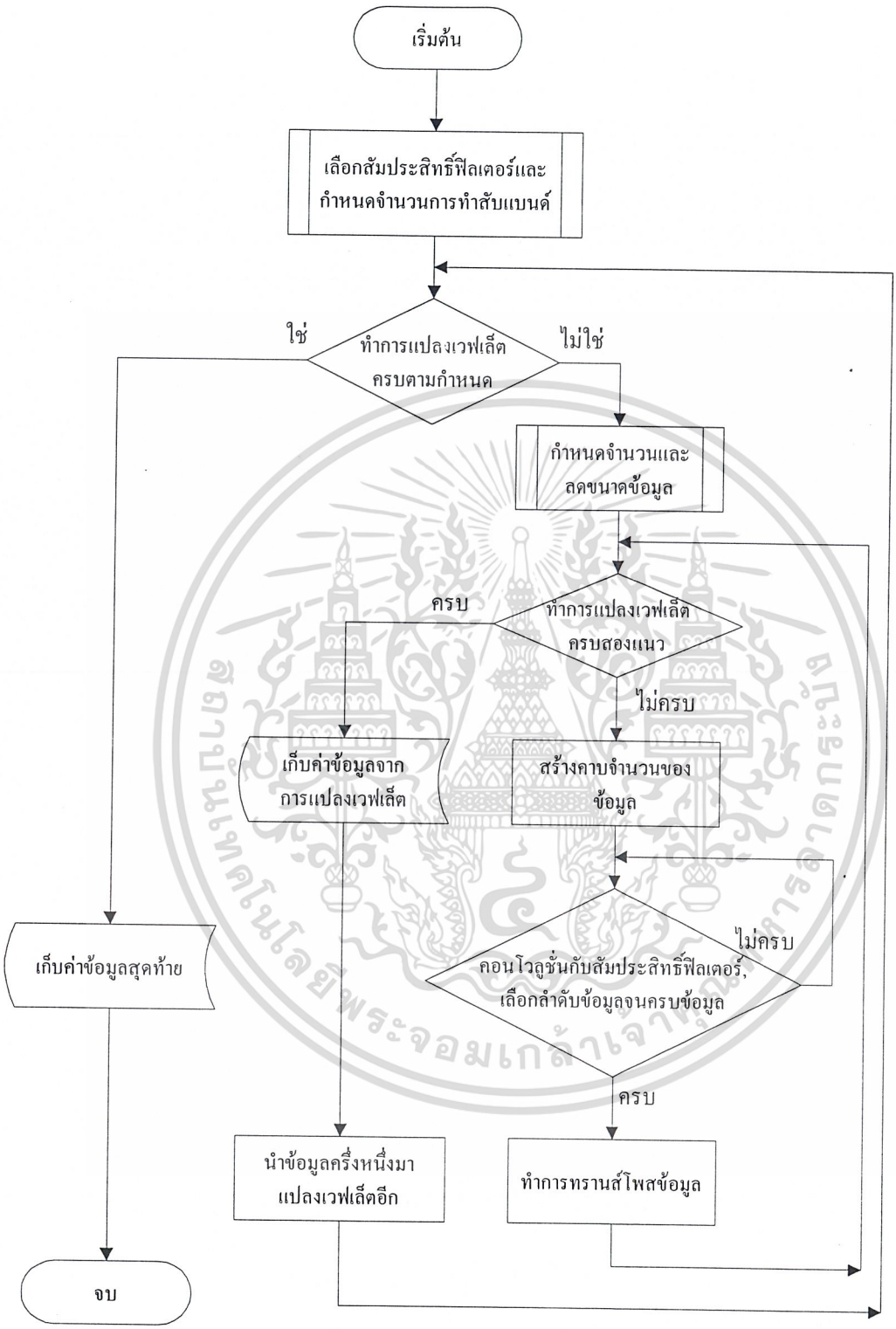
คำสั่งจำลองที่ 3.5 เป็นวิธีการปัดค่าเทรชโฮลของสัมประสิทธิ์เวฟเล็ตของภาพ เพื่อทำการปัดให้เกิดค่าความผิดพลาดให้ได้ตามที่ต้องการ ซึ่งเมื่อนำการปัดค่าเทรชโฮลแล้ว จะนำสัมประสิทธิ์ที่เหลืออยู่ไปผ่านการจัดระดับข้อมูลแล้วเข้ารหัสการลดขนาดข้อมูลที่ไม่มีการสูญเสียต่อไป โดยในการทดลองในบทที่ 4 ได้ทำการทดลองวิธีการดังกล่าวไว้ เพื่อนำมาเปรียบเทียบกับวิธีการของเอ็มเบดอิดซีโร่ทรี

จากผังงานจะนำไปเขียนเป็นโปรแกรมบนโปรแกรม MATLAB ได้ดังในภาพผนวก ก.

คำสั่งจำลองที่ 3.5 แสดงโปรแกรมลดขนาดข้อมูลภาพโดยการปัดค่าเทรชโฮลของสัมประสิทธิ์ภาพในการแปลงเวฟเล็ต

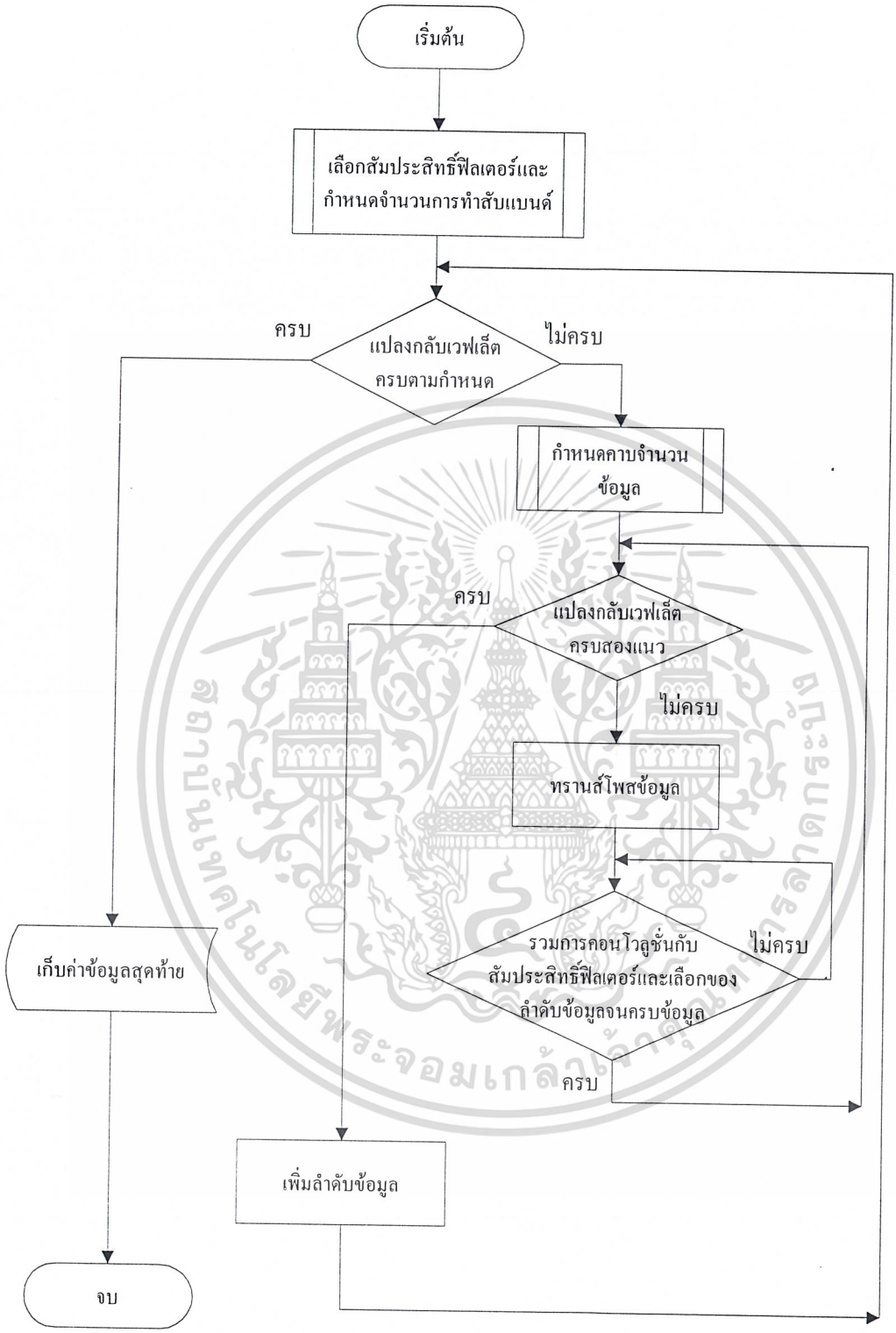
```
PROCEDURE Compress( C : array [1...m] of reals ; e : real )
BEGIN
        Tmin = min{ | C[i] | };
        Tmax = max{ |C[i] | };
        DO
                T = ( Tmin + Tmax) / 2;
                s= 0;
                FOR i : 1 to m DO
                        IF |C[i]| < T then s = s + (C[i])^2;
                END FOR
                IF s< e^2 then Tmin = T; else Tmax = T;
        UNTIL Tmin = Tmax;
        FOR i = 1 to m DO
                IF |C[i]| < T THEN C[i] = 0;
        END FOR
END.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผังงานที่ 3.1 แสดงผังการทำงานการแปลงเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผังงานที่ 3.2 ผังการทำงานการแปลงกลับเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การกำหนดรหัสสัญลักษณ์ให้กับสัมประสิทธิ์เวฟเล็ต

จากหัวข้อที่ 2.4.2 ในบทที่ 2 การเข้ารหัสแบบเอ็มเบดอิด ซีโร่ทรี (EZW) ที่มีกระบวนการจัดสัมประสิทธิ์เป็นโดมิแนนท์ และ สับออร์ดิเนท สามารถที่จะเขียนเป็นผังงานได้ดังผังงานที่ 3.3 จะสามารถตัวอย่างการนำข้อมูลสัมประสิทธิ์มาจัดการเข้ารหัสแบบเอ็มเบดอิด ซีโร่ทรี ได้ดังนี้

ตัวอย่าง

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 จะทำขบวนการโดมิแนนท์ก่อนคือ ทำการสแกนค่าสัมประสิทธิ์ตามรูปที่ 3.2 โดยทิศทางของการสแกนจะมีลักษณะเป็นแบบตัวซี (ดังในรูปที่ 2.27) ซึ่งจะพบค่าสัมประสิทธิ์ตัวแรกมีขนาดเท่ากับ 63 โดยมีค่าของสัมประสิทธิ์มากกว่าค่าเทรชโฮลเริ่มต้น ซึ่งจะถือว่าเป็นค่านัยสำคัญ และจะกำหนดสัญลักษณ์เป็น *POS* เพราะค่าสัมประสิทธิ์เป็นบวก เมื่อสแกนต่อไป จะพบสัมประสิทธิ์ที่มีค่าเท่ากับ –34 และมีค่าสัมบรูณ์ที่มากกว่าค่าเทรชโฮลเริ่มต้นแต่ค่าเป็นลบจะถูกกำหนดด้วยสัญลักษณ์เป็น *NEG* ถ้าสแกนต่อค่าสัมบรูณ์ของ –31 น้อยกว่าค่าเทรชโฮลเริ่มต้น แต่เมื่อมองในสับแบนด์สเกลของสัมประสิทธิ์ของทั้งโหนดลูก และ โหนดสืบทอด แล้วมีสัมประสิทธิ์บางตัวมากกว่าค่าเทรชโฮลเริ่มต้น ดังนั้นตำแหน่งนี้จะถูกกำหนดให้แทนด้วยสัญลักษณ์ คือ ไอโซเลทซีโร่ (*IZ*) จากนั้นค่า 23 ที่ถูกสแกนต่อมา มีค่าน้อยกว่าค่าเทรชโฮล รวมทั้งสัมประสิทธิ์ทั้งหมดของสับแบนด์ โหนดลูก และ โหนดสืบทอด มีค่าน้อยกว่าเทรชโฮล ตำแหน่งนี้ก็จะถูกกำหนดด้วย รากซีโร่ทรี (*ZTR*) ทำให้ไม่ต้องพิจารณาสัมประสิทธิ์ในโหนดลูก และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนดสืบทอดของโดมิแนนท์นี้อีก แล้วทำการสแกนต่อไปอีกโดยจะพิจารณาตามหลักการที่เหมือนกันกับหลักการที่กล่าวมาข้างต้น โดยสามารถสรุปเป็นตารางของขบวนการทำโดมิแนนท์ในการสแกนหนึ่งครั้ง ดังตารางที่ 3.1

ตารางที่ 3.1 แสดงค่าของสัมประสิทธิ์ที่มีนัยสำคัญและไม่สำคัญในเอ็มเบดอิดซีโร่ทรี

Subband	ค่าของสัมประสิทธ์	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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 และแทนค่าเป็นศูนย์ เพื่อให้เกิดกลุ่มข้อมูลที่เป็นศูนย์เพิ่มขึ้น ส่วนในขบวนการสับออร์ดิเนทครั้งที่สอง 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 จะนำสัญลักษณ์มาจัดเรียงตามลักษณะการสแกนแบบตัวซี (Z) คือ ขบวนการโดมิแนนท์จะได้ลำดับของสัญลักษณ์ดังนี้

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

และ ลำดับของสัญลักษณ์ในขบวนการของสับออร์ดิเนทจะได้ดังนี้

ลำดับของสัญลักษณ์ในสับออร์ดิเนท = { *1* , *0* , *1* , *0* }

เมื่อนำลำดับของสัญลักษณ์ทั้งโดมิแนนท์และสับออร์ดิเนทมารวมกัน จะได้ลำดับของสัญลักษณ์ใหม่ คือ

ลำดับของสัญลักษณ์ = { *POS* , *1* , *NEG* , *O* , *IZ* , *ZTR* , *POS* , *1* , *ZTR* , *ZTR* , *ZTR* , *ZTR* , *IZ* , *ZTR* , *ZTR* , *Z* , *Z* , *Z* , *Z* , *Z* , *POS* , *O* , *Z* , *Z* }

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยลำดับของชุดสัญลักษณ์นี้เกิดจากการทำบวนการของโดมิแนนท์และสับออร์ดิเนทหนึ่งครั้ง เพื่อให้ data stream มีรหัสสัญลักษณ์น้อยลงโดยการแทนสัญลักษณ์ {*POS,NEG,IZ,ZTR*} ด้วย { *01,10,11,00* } ตามลำดับ ดังนั้นจะได้ลำดับของสัญลักษณ์ใหม่ คือ

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 จะถูกนำมาจัดเรียงอีก คือ

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

ลำดับของสัญลักษณ์ในสับออร์ดิเนทครั้งที่ 2 = { *1,0* }

จากขบวนการโดมิแนนท์และสับออร์ดิเนทครั้งที่สอง รวมกันเป็นลำดับของสัญลักษณ์ ทั้งหมดคือ

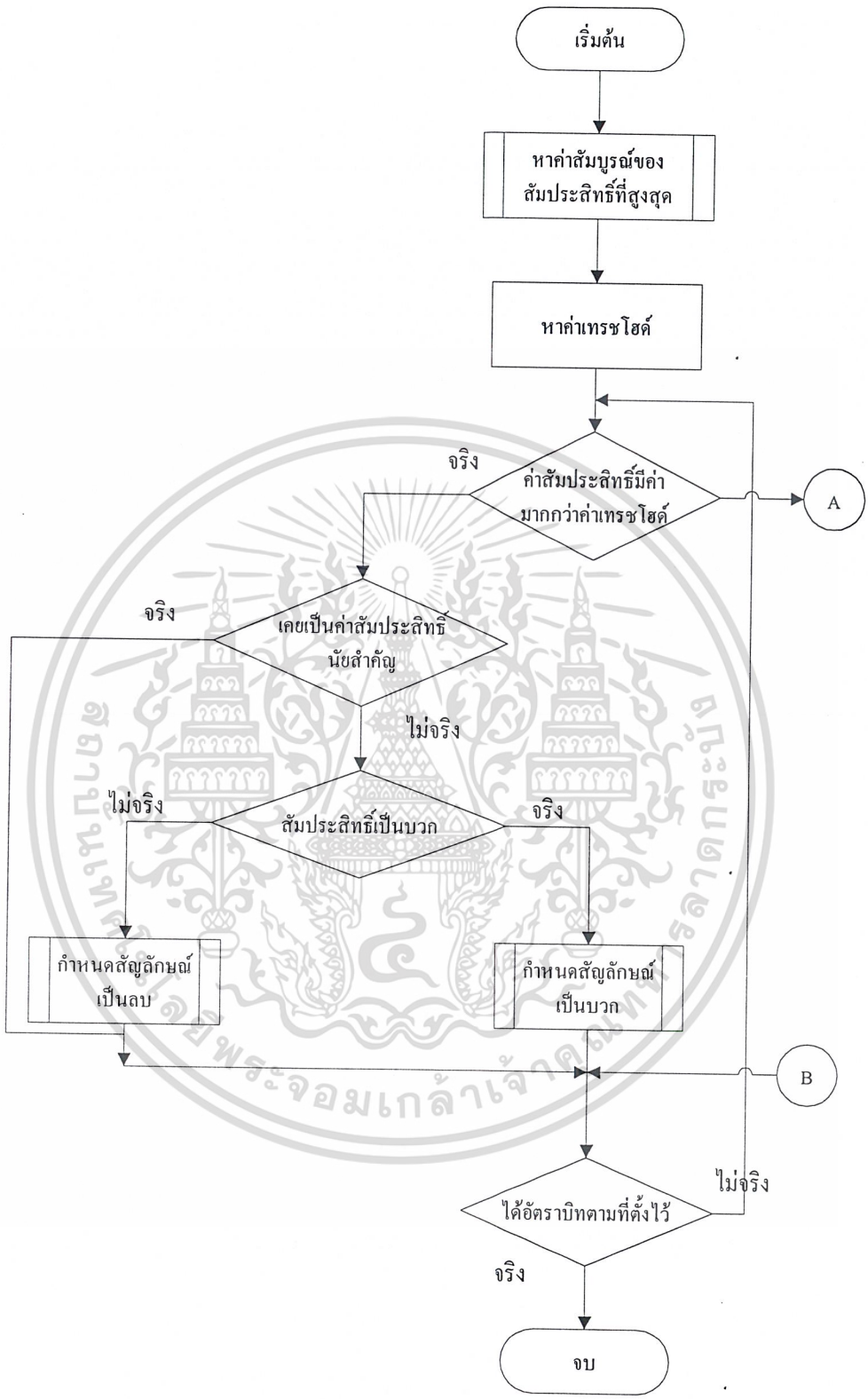
ลำดับของข้อมูล = { *รหัสสัญลักษณ์ในโดมิแนนท์และ รหัสสัญลักษณ์ในสับออร์ดิเนท ที่ 1 , รหัสสัญลักษณ์ในโดมิแนนท์และ รหัสสัญลักษณ์ในสับออร์ดิเนท ที่ 2* }

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* }

ลำดับของสัญลักษณ์ = { *01 , 11 , 10 , 00 , 11 , 00 , 01 , 10 , 00 , 00 , 00 , 00 , 11 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 01 , 01 , 00 , 00 , 10 , 1 , 01 , 0 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 00* }

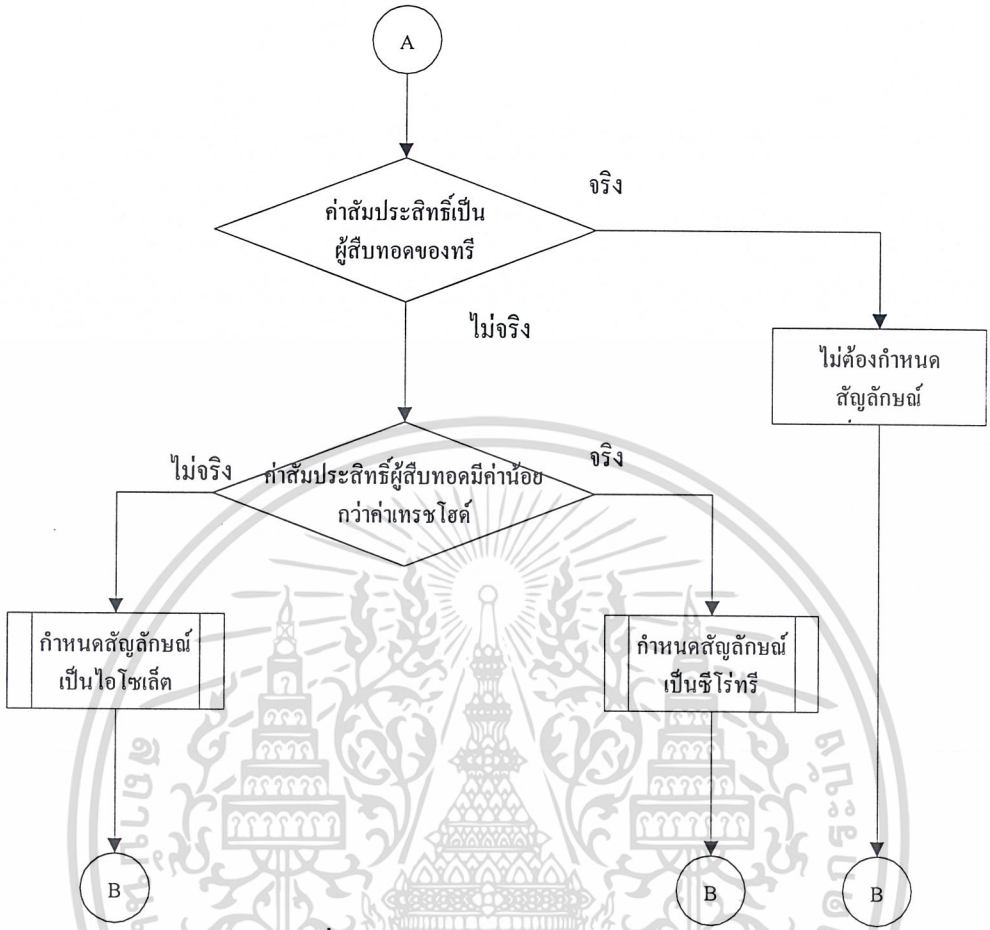
ในการทำกลับของเอ็มเบดอิดซีโร่ทรีจะมีขั้นตอนดังผังงานที่ 3.5 และผลจากการทำกลับของลำดับข้อมูล ที่ได้จากการสแกนสัมประสิทธิ์กลับจะแสดงในรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผังงานที่ 3.3 เข้ารหัส เอ็มเบดอิดซีโร่ทรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

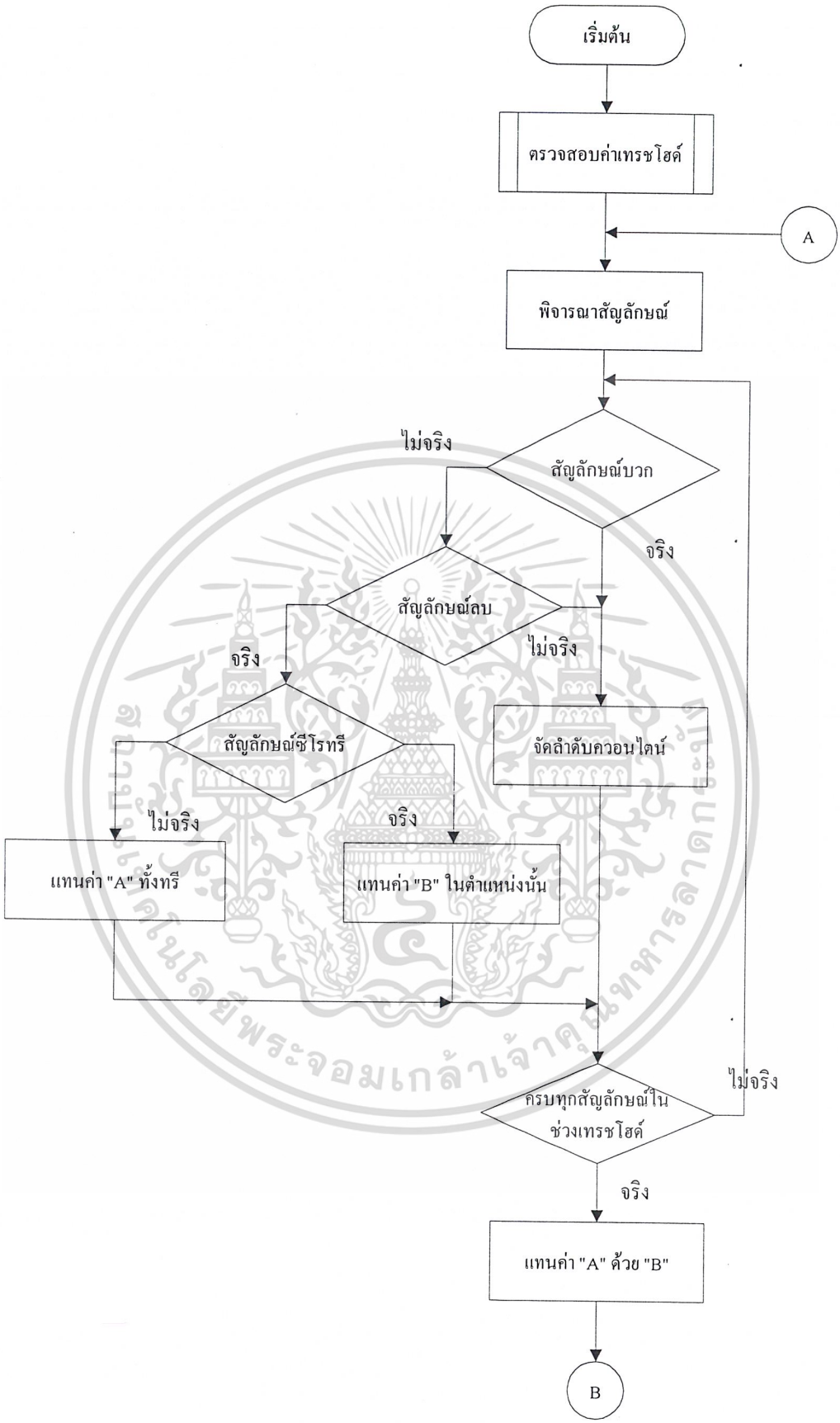


ผังงานที่ 3.4 การทำงานของ EZWส่วนที่สอง

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

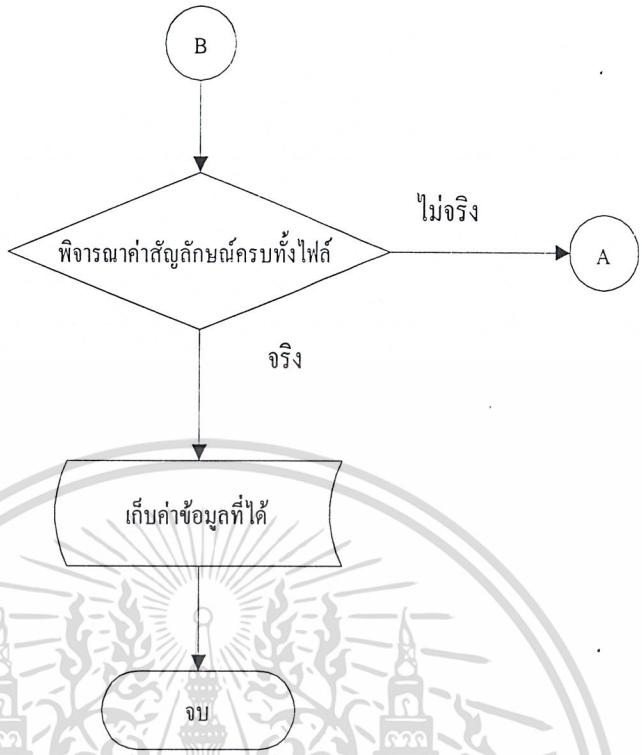
รูปที่ 3.3 ผลการทำกลับของ data stream

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

ผังงานที่ 3.5 แสดงการทำกลับขอเอ็มเบดอิ้ดซีโร่ทรี



ผังงานที่ 3.6 แสดงการทำกลับเอ็มเบดอิดซีโร่ทรี(ต่อ)

3.3 การปรับปรุงวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี

ในการนำสัมประสิทธิ์ของภาพภายหลังการแปลงเวฟเล็ตมาแทนด้วยรหัสข้อมูลนับว่ามีความสำคัญมากในการลดขนาดข้อมูล เพราะถ้าจำการจัดรหัสแทนค่าของสัมประสิทธิ์ได้เหมาะสมแล้ว ก็สามารถที่จะลดขนาดข้อมูลได้มากยิ่งขึ้น ดังนั้น จึงได้ทำการปรับปรุงวิธีการเข้ารหัสแบบเอ็มเบดอิด-ซีโร่ทรีใหม่ ดังนี้

พิจารณาการกำหนดสัญลักษณ์ในขบวนการโดมิแนนท์ตามลักษณะในผังงานที่ 3.3 และในวิธีการทำสับออร์ดิเนทแต่ละครั้งค่าสัมประสิทธิ์ที่มีค่ามากกว่าค่าระดับของแต่ละโดมิแนนท์จะถูกแทนด้วยข้อมูลไบนารีเพียงบิตเดียวเท่านั้น และในการทำโดมิแนนท์ครั้งที่หนึ่งจะไม่ทำการพิจารณาค่าของสับออร์-ดิเนท คือ

จากตัวอย่างที่ผ่านมา เมื่อนำมาเข้ารหัสด้วยวิธีการที่ปรับปรุงขึ้นมาใหม่จะได้ดังนี้

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

และ ลำดับของสัญลักษณ์ในขบวนการของสับออร์ดิเนทจะได้ดังนี้

ลำดับของสัญลักษณ์ในสับออร์ดิเนท = *ไม่ทำการพิจารณาในครั้งที่หนึ่ง*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ลำดับของสัญลักษณ์ในสับออร์ดิเนทครั้งที่ 2 = { *1 , 0 , 1 ,0* }

จากขบวนการโดมิแนนท์และสับออร์ดิเนทครั้งที่สอง รวมกันเป็นลำดับของสัญลักษณ์ ทั้งหมดคือ

ลำดับของสัญลักษณ์ = { *รหัสสัญลักษณ์ในโดมิแนนท์ ที่ 1 , รหัสสัญลักษณ์ในสับออร์ดิเนท ที่ 2 , รหัสสัญลักษณ์ในโดมิแนนท์ ที่ 2* }

ลำดับของสัญลักษณ์ = { *POS , NEG , IZ , ZTR , POS , ZTR , ZTR , ZTR , ZTR , IZ , ZTR , ZTR , Z , Z , Z , Z , Z , POS , Z , Z , 1 , 0 , 1, 0 , NEG , POS , ZTR , ZTR , ZTR , ZTR , ZTR , ZTR , ZTR , ZTR , ZTR , ZTR , ZTR* }

ลำดับของสัญลักษณ์ = { *01 , 10 , 11 , 00 , 01 , 00 , 00 , 00 , 00 , 11 , 00 , 00 , 00 , 11 , 11 , 11 , 11 , 11 , 01 , 11 , 11 , 1 , 0 , 1 ,0 , 10 , 01 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 00 , 00* }

และในวิธีการสร้างค่าสัมประสิทธิ์ขึ้นมาจากรหัสข้อมูล สามารถกระทำตามขบวนการในผังงานที่ 3.5 และ 3.6

3.4 การเข้ารหัสเลขคณิต

เนื่องจากวิธีการเข้ารหัสเลขคณิตจะอาศัยค่าในช่วงระยะขอบเขตของความน่าจะเป็นในการเกิดสัญลักษณ์ของข้อมูลแต่ละสัญลักษณ์ ซึ่งจะมีค่าเป็นจำนวนจริงที่มีช่วงอยู่ระหว่าง [0,1) พบว่าเมื่อเข้ารหัสกับลำดับข้อมูลที่มีขนาดยาว จะยิ่งทำให้ค่าช่วงระยะขอบเขตยิ่งมีค่าน้อยลงและจำนวนลำดับของรหัสยิ่งมากขึ้นด้วย ซึ่งจะเป็นปัญหาสำคัญในการเขียนโปรแกรมเพราะจะเกิดโฟลตติ้งพอยต์ (floating point) ขึ้นได้ วิธีการแก้ปัญหาในกรณีนี้ได้มีผู้คิดค้นวิธีขึ้นมา คือ วิทเทน (Witten) ,เนล (Neal) ,และ เคลียรี่ (Cleary) โดยเขาจะป้องกันการลดค่าของช่วงระยะขอบเขตลงได้ มีขั้นตอนที่เพิ่มขึ้นจากเดิมดังนั้นสามารถเขียนเป็นขั้นตอนได้ดังนี้

ขั้นตอนที่ 1 จะทำการกำหนดช่วงขอบเขตเริ่มต้น [*L,H*) ให้เป็น [0,1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 ในขั้นตอนนี้จะนำสัญลักษณ์ของข้อมูลมาพิจารณา โดยแบ่งเป็น 2 ขั้นตอนย่อย คือ (แสดงดังรูปที่ 2.31)

1. ทำการแบ่งช่วงขอบเขตปัจจุบันให้เป็นช่วงของขอบเขตย่อย ตามความน่าจะเป็นของสัญลักษณ์
2. เลือกช่วงของขอบเขตย่อยที่สอดคล้องกับสัญลักษณ์ที่พบในข้อมูลแล้วนำช่วงของขอบเขตดังกล่าวมาเป็นช่วงขอบเขตปัจจุบันแทน
   ทำซ้ำข้อมูลไปเรื่อยๆ
   1. ถ้าช่วงขอบเขตใหม่ไม่อยู่ในขอบเขตใด ขอบเขตหนึ่งใน $[0,\frac{1}{2})$ หรือ $[\frac{1}{4},\frac{3}{4})$ หรือ $[\frac{1}{2},1)$ จะหยุดการทำซ้ำในลูปนี้
   2. ถ้าช่วงขอบเขตใหม่อยู่ในช่วง $[0,\frac{1}{2})$ จะกำหนดให้เอาต์พุตเป็น 0 แล้วทำการเพิ่มขนาดระยะขอบเขตของข้อมูลปัจจุบันในช่วง $[0,\frac{1}{2})$ ให้เป็นสองเท่าแล้วขยายไปในช่วงทางด้านขวา
   3. ถ้าช่วงขอบเขตใหม่อยู่ในช่วง $[\frac{1}{2},1)$ จะกำหนดให้เอาต์พุตเป็น 1 แล้วทำการเพิ่มขนาดระยะขอบเขตของข้อมูลปัจจุบันในช่วง $[\frac{1}{2},1)$ ให้เป็นสองเท่าแล้วขยายไปในช่วงทางด้านซ้าย
   4. ถ้าช่วงขอบเขตอยู่ในช่วง $[\frac{1}{4},\frac{3}{4})$ จะไม่กำหนดให้มีเอาต์พุตเกิดขึ้น แต่จะกำหนดให้เกิดเอาต์พุตที่ตรงกันข้ามกับเอาต์พุตที่จะเกิดขึ้นในอนาคต แล้วทำการเพิ่มขนาดระยะขอบเขตของข้อมูลปัจจุบันในช่วง $[\frac{1}{4},\frac{3}{4})$ ให้เป็นสองเท่าแล้วขยายไปในสองข้างของจุดกึ่งกลาง

กระทำในขั้นตอนที่สองจนกว่าจะสิ้นสุดชุดของข้อมูลที่จะนำเข้ามาเข้ารหัส หรือเมื่อพบอักษรที่แสดงถึงจุดสิ้นสุดของข้อมูล

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

$$R = (H-L)+1 \tag{3.2}$$

$$H_{new} = L+R(Q[i])/Q_{total})-1 \tag{3.3}$$

$$L_{new} = L+R(Q[i-1])/Q_{total}) \tag{3.4}$$

โดย R คือระยะห่างของขอบเขต และ

H,L คือสองจุดของช่วงระยะในการเข้ารหัส

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยวิธีการเข้ารหัสเลขคณิตตามหลักการของ วิทเทน (Witten) ,เนล (Neal) ,และ เคลียรี่ (Cleary) สามารถเขียนเป็นคำสั่งจำลองได้ดังนี้

คำสั่งจำลองที่ 3.6 แสดงการเข้ารหัสเลขคณิตตามวิธีการของ วิทเทน (Witten) ,เนล (Neal) ,และ เคลียรี่ (Cleary)

```
PROCEDURE Encode
BEGIN
    Initialize model and register L,H,R;
    WHILE (symbol to 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 1 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;
        TERMINATE; }
    Update model;
}
END.
```

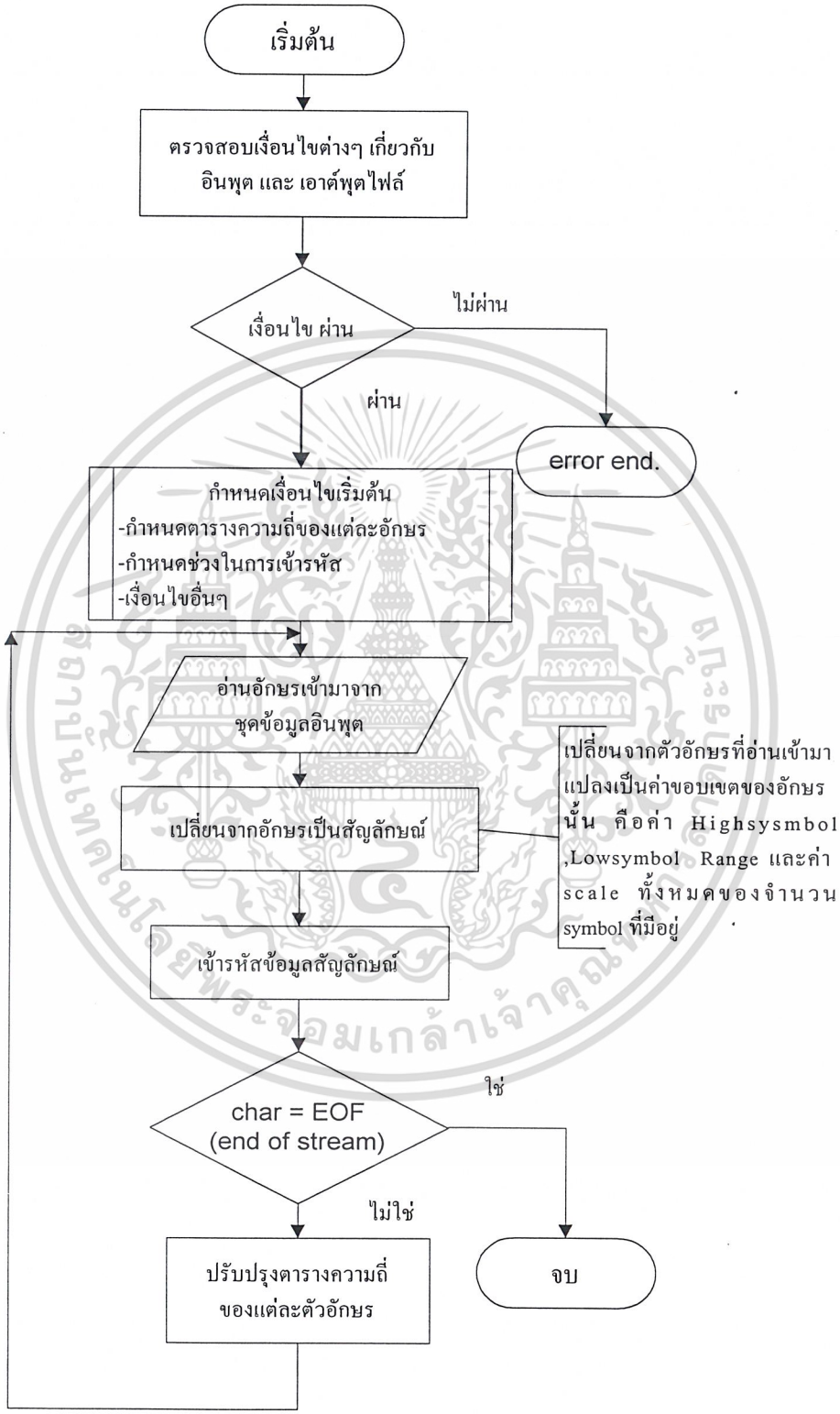
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งจำลองที่ 3.7 แสดงการเข้าถอดรหัสเลขคณิตตามวิธีการของ วิทเทน (Witten) ,เนล (Neal) ,และ เคลีย รี่ (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] ≤ V < Q[x];
        ALWAYS { %% Renormalize interval
            IF ( H ≤ Half)
                %% Do nothing here
            ELSEIF ( L ≤ Half ) {
                L = L – Half;
                H = H – Half;
                V = V – Half;}
            ELSEIF ( H < 3rd _qrt and L ≥ 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.
```

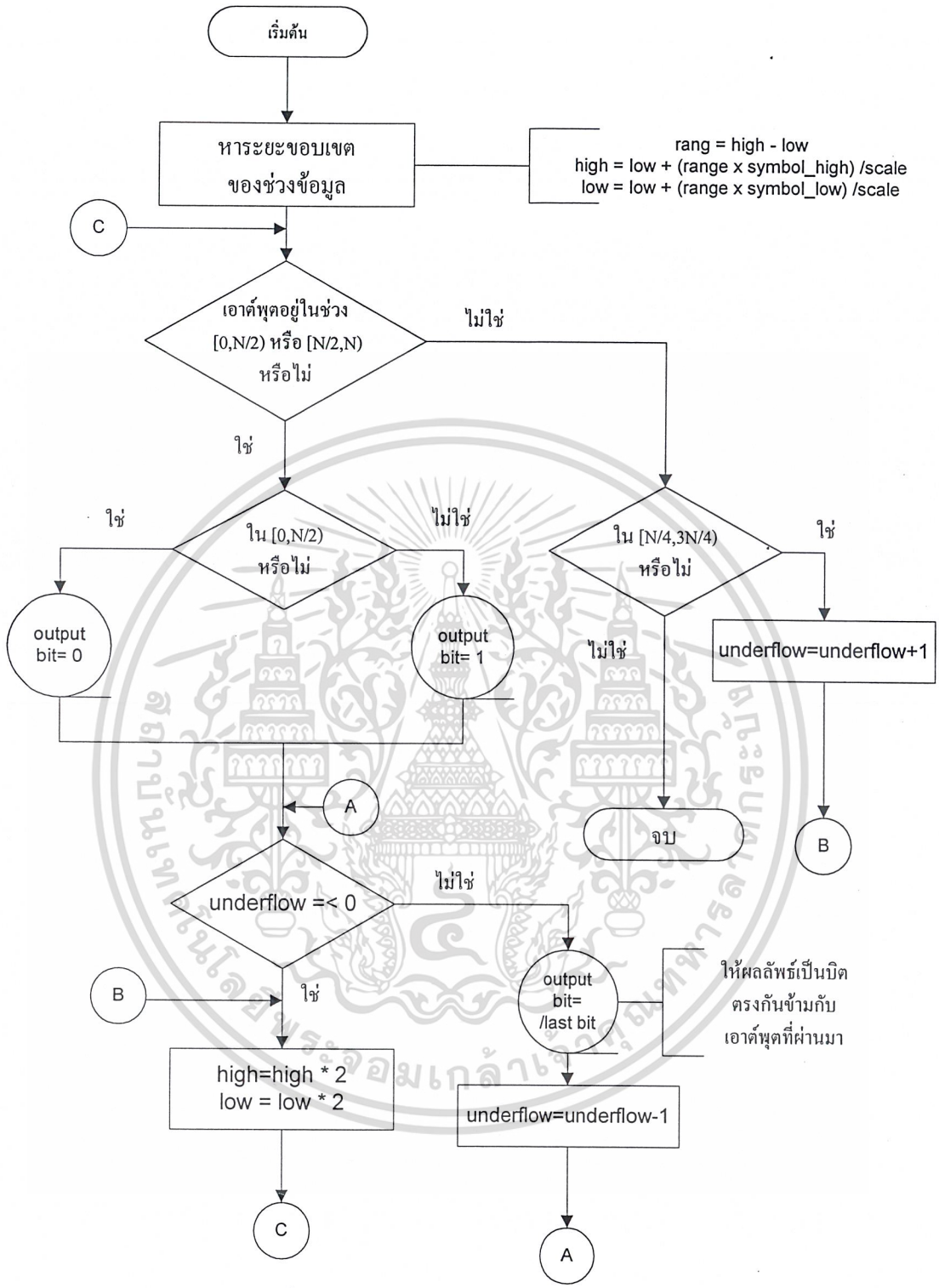
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผังการทำงานของโปรแกรมโดยสังเขป ดังนี้



ผังงานที่ 3.7 แสดงผังงานการเข้ารหัสเลขคณิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

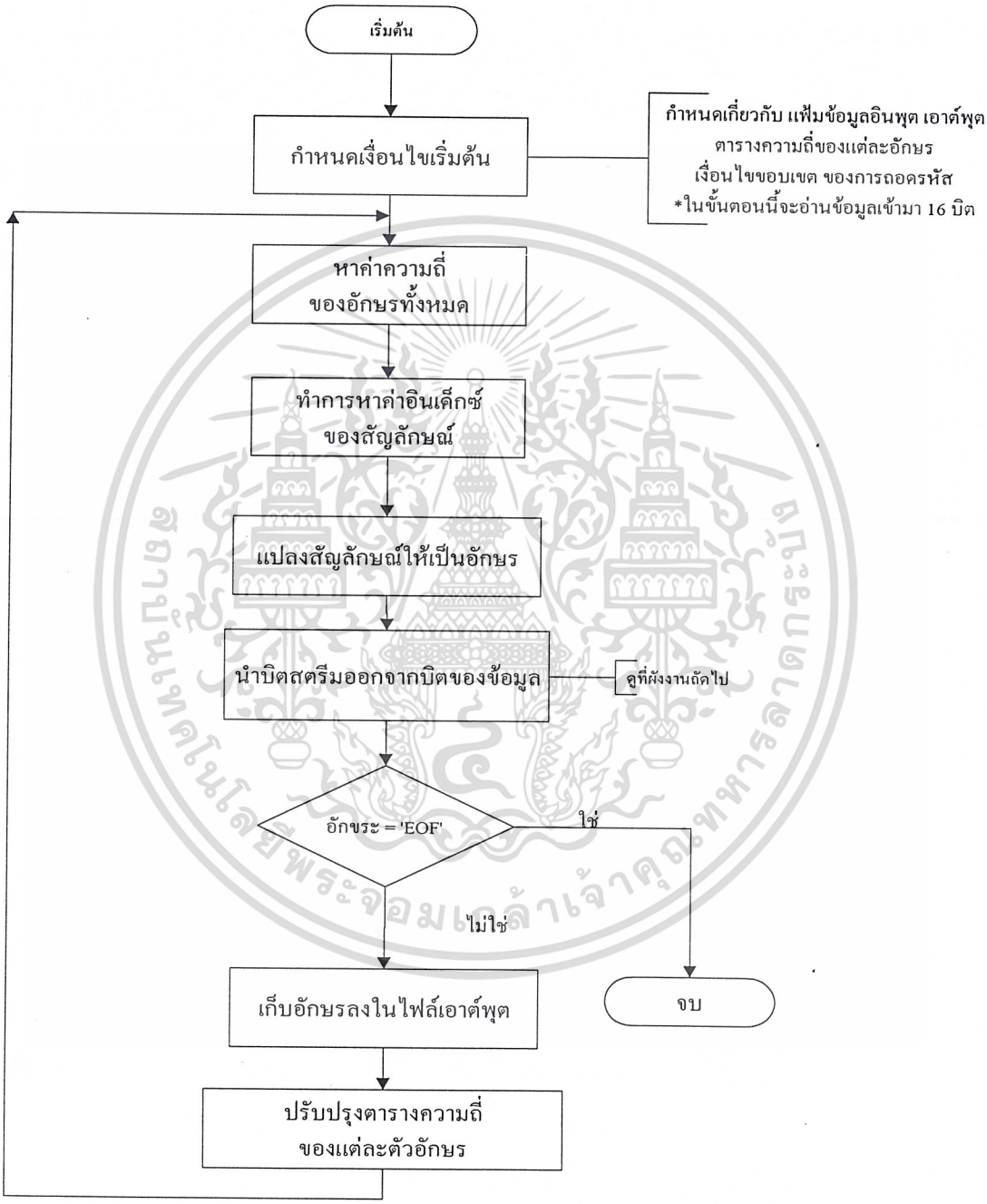


ผังงานที่ 3.8 แสดงลูปย่อการเข้ารหัสข้อมูลสัญลักษณ์

จากวิธีการข้างต้นนี้พบว่าจะสามารถป้องกันการลดช่วงระยะขอบเขตได้ระดับหนึ่ง แต่เมื่อนำมาใช้กับระยะขอบเขตเริ่มต้นที่ทำการกำหนดค่าเริ่มต้นเป็นจำนวนเต็มที่มีค่าสูงๆ ก็จะป้องกันการเกิดโฟลตติ้งพอยต์ได้ ซึ่งค่าจำนวนเต็มที่จะกำหนด เพื่อความสะดวกในการเขียนโปรแกรมเพื่อที่จะได้สะดวกต่อการดำเนินการทางบิต จึงควรกำหนดอยู่ในช่วงของค่า 2^n-1 ซึ่งในโปรแกรมที่อยู่ในภาคผนวก

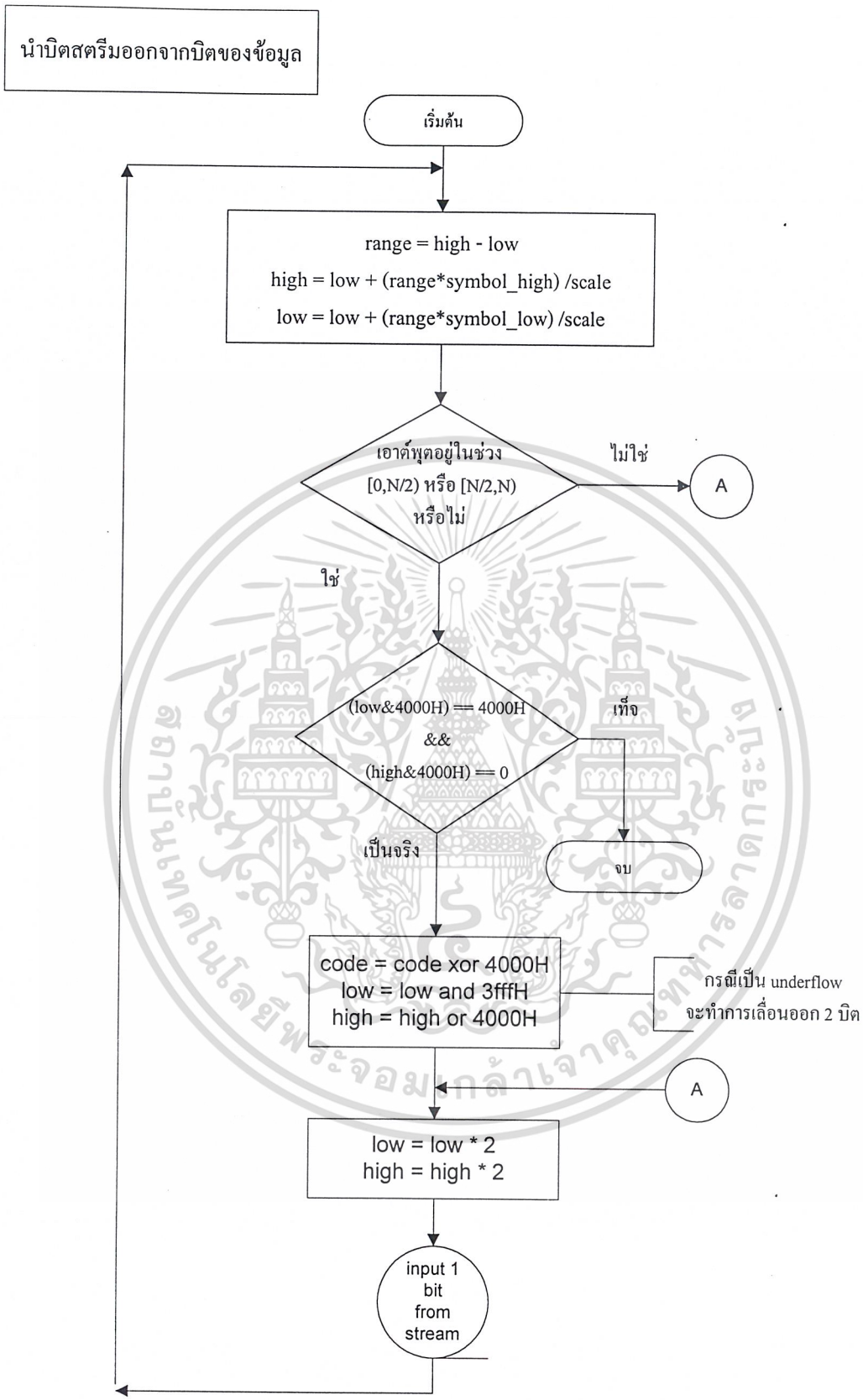
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ส่วนในการถอดรหัสดังที่ได้กล่าวมาแล้วในหัวข้อที่ 2.5.2 ในการนำมาเขียนเป็นโปรแกรมเพื่อใช้งานจะอาศัยหลักการของการเลื่อนบิตข้อมูลเข้ามาทีละบิตแล้วทำการนำมาเปรียบเทียบกับช่วงของโอกาสการเกิดขึ้นของสัญลักษณ์ในช่วงต่างๆ ว่าตรงกับช่วงใด โดยจะอาศัยวิธีการของ วิทเทน (Witten) ,เนล (Neal) ,และ เคลียรี่ (Cleary) ในการเขียนโปรแกรม ซึ่งสามารถเขียนเป็นผังงาน ได้ดังนี้



ผังงานที่ 3.9 แสดงผังงานในการถอดรหัสเลขคณิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผังงานที่ 3.10 แสดงผังงานในการนำบิตข้อมูลออกจากชุดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การส่งภาพแบบโปรเกรสซีฟโดยใช้การแปลงเวฟเล็ต

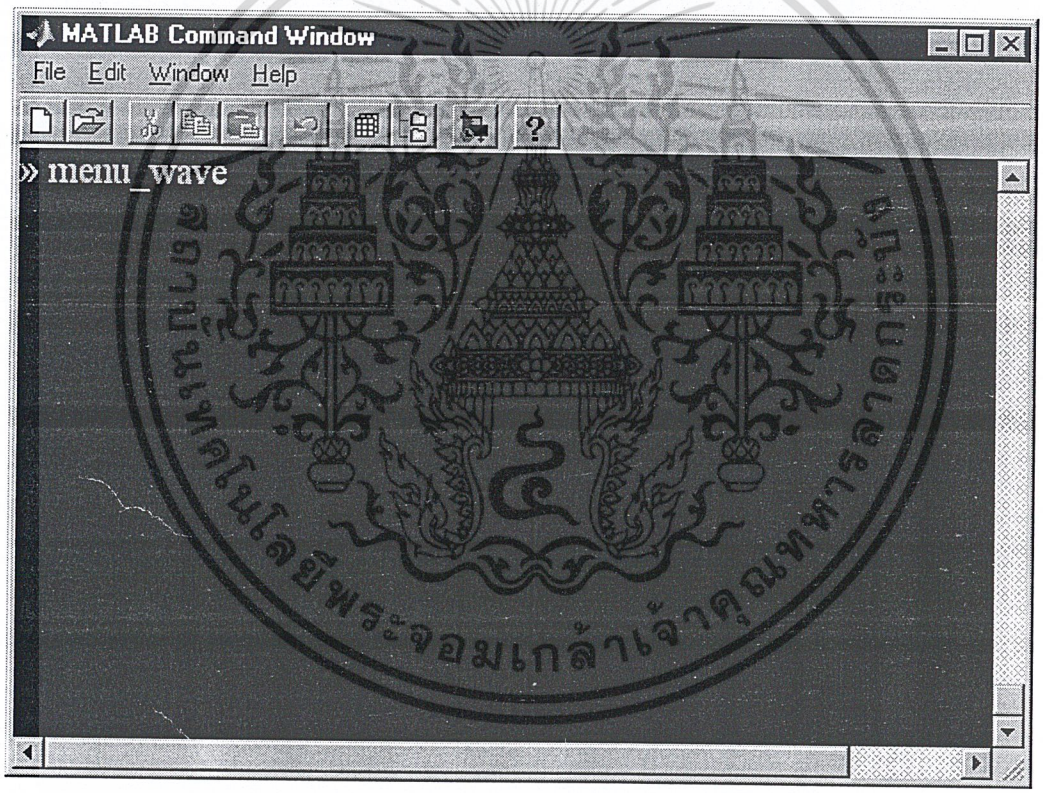
การส่งภาพแบบโปรเกรสซีฟเป็นการส่งภาพซ้อนกันไปเรื่อยๆ โดยในการส่งครั้งแรกๆ ภาพจะมีลักษณะหยาบและค่อยๆ มีความคมชัดมากขึ้นในการส่งครั้งต่อๆ ไป และเมื่อทำการพิจารณาภาพทั้งชุดแล้วพบว่าภาพในการส่งก่อนหน้าเหมือนกับเป็นภาพในการส่งครั้งต่อไปที่ผ่านตัวกรองความถี่ต่ำ ดังนั้นจึงสามารถทำการส่งภาพแบบโปรเกรสซีฟได้อีกวิธีหนึ่ง โดยนำการแปลงเวฟเล็ตมาใช้เป็นตัวกรอง เพราะการแปลงเวฟเล็ตนี้สามารถเทียบได้กับการทำการแบ่งแบนด์ความถี่ของภาพนั่นเอง

การส่งภาพแบบโปรเกรสซีฟจะทำการส่งข้อมูลภาพในส่วนที่สำคัญที่สุดของภาพออกไปก่อน ดังนั้นข้อมูลทั้งหมดต้องถูกตัดแบ่งออกเป็นส่วนๆ แล้วนำมาเข้ารหัสตามลำดับความสำคัญของข้อมูล ซึ่งการจัดเรียงในลักษณะของข้อมูลในโดเมนความถี่จะมีความเหมาะสมมากกว่า เพราะนอกจากจะถูกจัดเรียงตามลำดับความถี่ที่เพิ่มขึ้นแล้ว พลังงานส่วนใหญ่ยังรวมตัวอยู่ในย่านความถี่ต่ำของภาพอีกด้วย เนื่องจากข้อมูลที่ความถี่ต่ำแสดงถึงโครงร่างของภาพ ดังนั้น ถ้าเลือกเฉพาะข้อมูลส่วนนี้ก็จะสามารถเห็นภาพที่ไม่คมชัดเหมือนผ่านตัวกรองความถี่ต่ำนั่นเอง ส่วนข้อมูลที่ความถี่สูงทำให้เกิดภาพในส่วนที่เป็นขอบหรือรายละเอียดเหมือนผ่านตัวกรองความถี่สูง และเนื่องจากการแปลงข้อมูลภาพโดยอาศัยการแปลงเวฟเล็ตแบบไม่ต่อเนื่องเปรียบเสมือนการนำภาพมาผ่านการกรองความถี่เป็นช่วงๆ ทำให้ได้ภาพออกมาเป็นแบนด์ต่างๆ กัน และวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรีจะทำการเข้ารหัสสัมประสิทธิ์ที่มีค่าในแบนด์ของความถี่ต่ำก่อน แล้วพิจารณาไปยังแบนด์ของความถี่สูง โดยถ้านำเอารหัสข้อมูลของแต่ละครั้งที่ทำโดมิแนนท์มาเข้ารหัสเลขคณิตแล้วทำการส่งข้อมูลออกไป ก็จะเป็นการส่งภาพแบบโปรเกรสซีฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การใช้งานโปรแกรมลดขนาดข้อมูลภาพโดยการแปลงเวฟเล็ตที่ใช้วิธีการของเอ็มเบดอิดซีโร่ทรี บน MATLAB

ในการใช้งานตัวโปรแกรมเพื่อทำการทดสอบวิธีการลดขนาดข้อมูลภาพที่ใช้การแปลงเวฟเล็ต ได้ทำการเขียนโปรแกรมบน MATLAB เพื่อให้สามารถติดต่อกับผู้ใช้งานได้อย่างสะดวก ไม่ต้องพิมพ์คำสั่งต่างๆ ผ่านทางเอดิเตอร์บน MATLAB วิธีการเรียนใช้ไดอะล็อกบ็อกซ์สำหรับกระทำการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเล็ต ซึ่งใช้วิธีการของเอ็มเบดอิดซีโร่ทรี ได้โดยพิมพ์คำสั่ง menu_wave บนหน้าจอ เอดิเตอร์ ดังรูปที่ 3.4 เท่านั้นโดยผู้ใช้งานสามารถสั่งคำสั่งการลดขนาดข้อมูลผ่านเมาส์บนหน้าจอกราฟฟิกได้ซึ่งมีลักษณะของเมนูดังรูปที่ 3.5 ได้

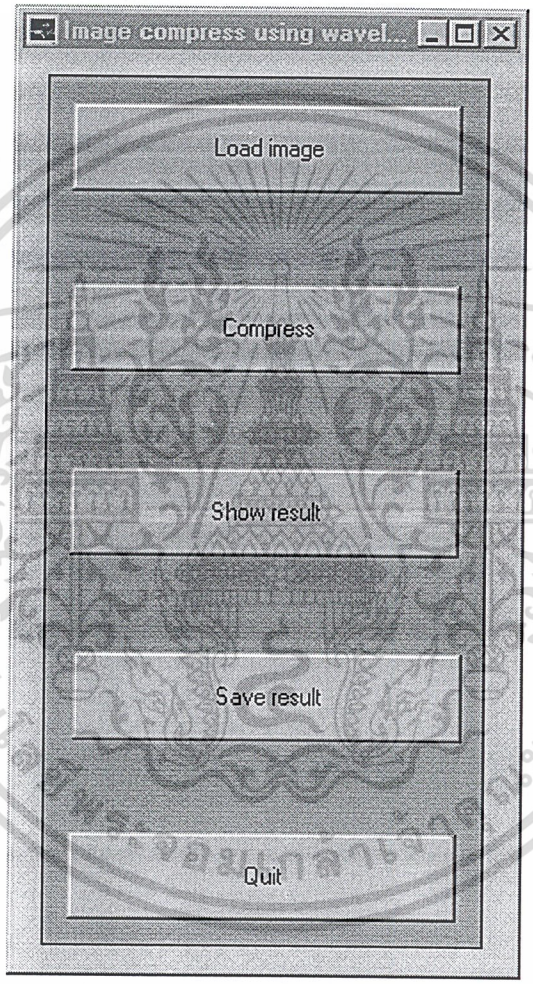


รูปที่ 3.4 หน้าจอเอดิเตอร์สำหรับพิมพ์เรียกเมนูไดอะล็อกบ็อกซ์
สำหรับการลดขนาดข้อมูลภาพด้วยการแปลงเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในเมนูจะประกอบด้วยกัน 5 เมนูหลักคือ

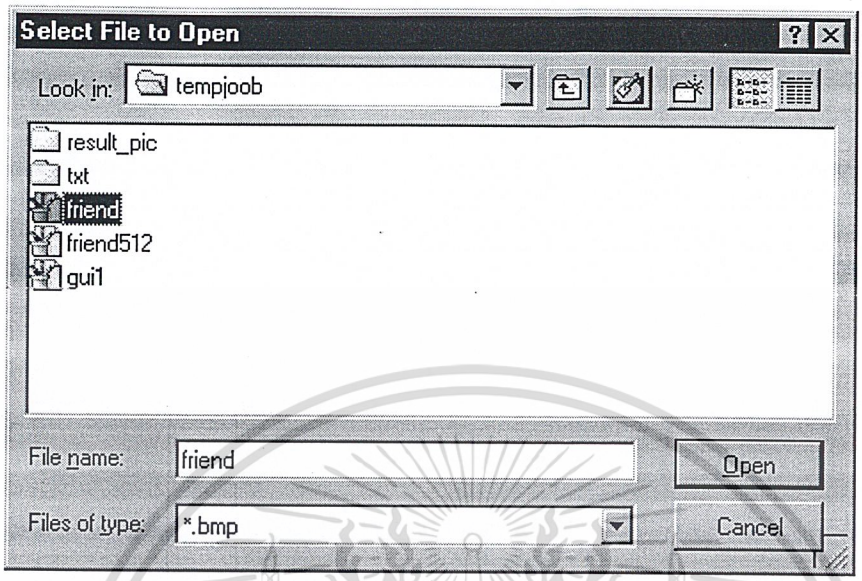
1. เมนูของการโหลดภาพอินพุตเข้ามา
2. เมนูสำหรับการลดขนาดข้อมูล
3. เมนูสำหรับการบันทึกผล
4. เมนูสำหรับการแสดงผล
5. เมนูสำหรับออกโปรแกรม



รูปที่ 3.5 แสดงตัวอย่างเมนูการใช้งานเพื่อติดต่อกับผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการคลิกใน เมนูโหลดภาพอินพุต จะมีไดอะล็อกบ็อกซ์ (dialog box) สำหรับเลือกอินพุตภาพขึ้นมา ดังรูปที่ 3.6



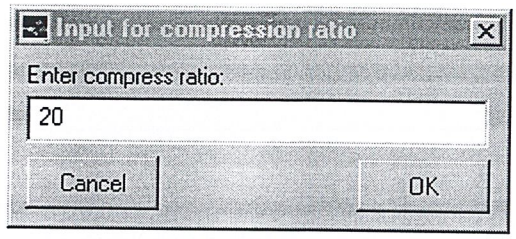
รูปที่ 3.6 แสดงไดอะล็อกบ็อกซ์ สำหรับการใส่อินพุตภาพเพื่อทำการลดขนาดข้อมูล

รูปที่ 3.7 แสดงภาพเมื่อคลิกเลือกภาพจากไดอะล็อกบ็อกซ์อินพุตภาพ

เมื่อทำการเลือกภาพแล้ว จะมีวินโดวส์แสดงภาพอินพุตที่เลือกไว้ขึ้นมาดังแสดงในรูปที่ 3.7

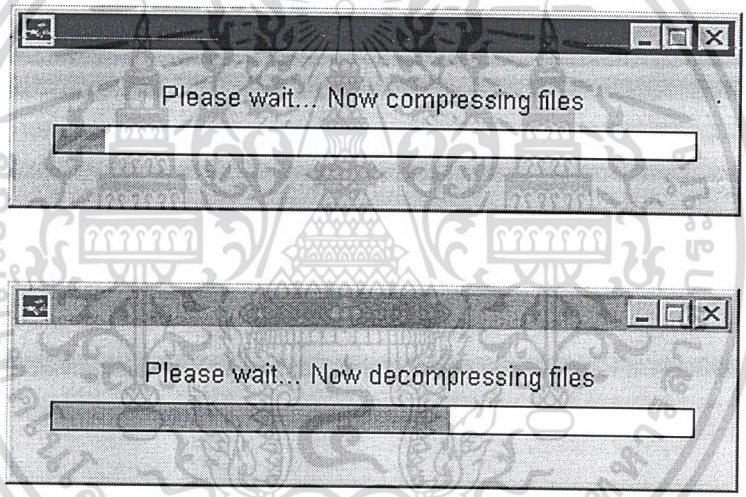
ในส่วนของเมนูของการลดขนาดข้อมูล จะมีไดอะล็อกบ็อกซ์ ซึ่งจะมีช่องว่างให้กรอกอัตราการลดขนาดข้อมูลขึ้นมา โดยสามารถเลือกอัตราการลดขนาดข้อมูลที่เทียบเป็น 1 ต่อ จำนวนที่ป้อนเข้าไป โดยมีค่าต่ำสุดที่ 20 (อัตราการลดขนาดข้อมูลทำได้ต่ำสุด 1 : 20) แสดงในรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงไดอะล็อกบ็อกซ์สำหรับกรอกอัตราการลดขนาดข้อมูล

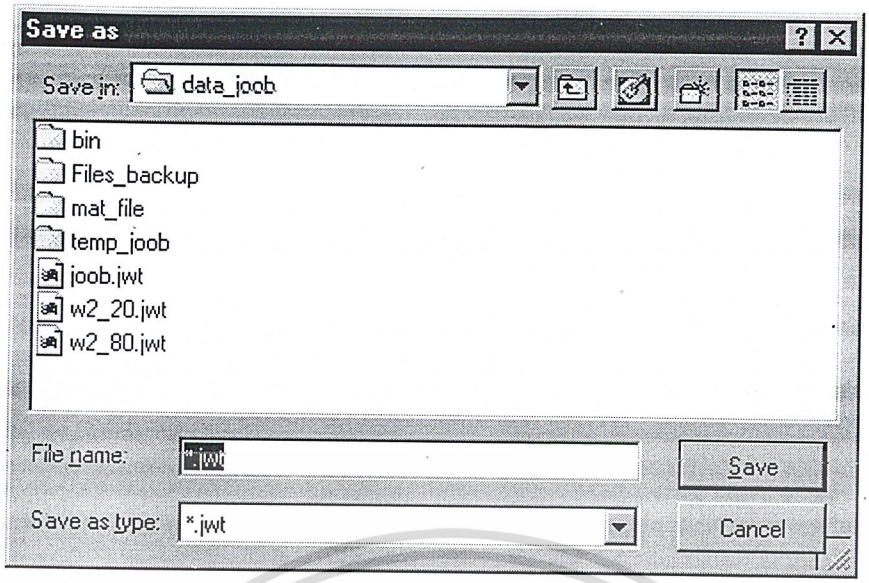
เมื่อคลิกปุ่ม OK โปรแกรมก็จะเริ่มทำการนำภาพที่ได้เลือกไว้แล้วมาผ่านวิธีการลดขนาดข้อมูลโดยการแปลงเวฟเล็ต ซึ่งในขั้นตอนนี้จะมี วินโดวส์ที่แจ้งระยะเวลาดำเนินการลดขนาดข้อมูล ขึ้นมา ดังรูปที่ 3.9



รูปที่ 3.9 แสดงวินโดวส์แสดงระยะเวลาของการลดขนาดข้อมูลภาพ ขึ้นมาในขณะทำการลดขนาดข้อมูล

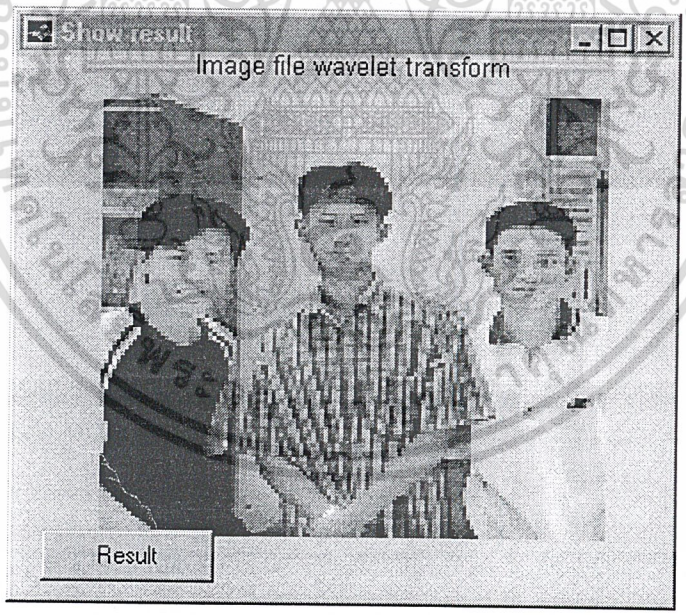
และเมื่อต้องการที่จะบันทึกขนาดของภาพไว้ จะมีไดอะล็อกบ๊อกซ์สำหรับบันทึกภาพลงแฟ้มข้อมูลขึ้นมา แสดงในรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงไดอะล็อกบ็อกซ์สำหรับบันทึกภาพลงแฟ้มข้อมูล

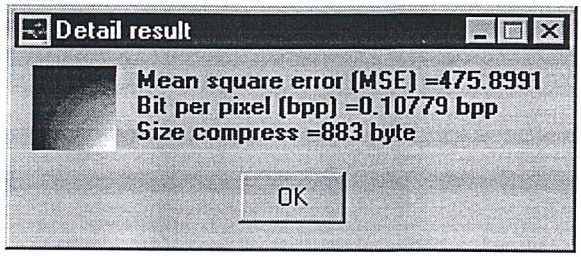
เมื่อต้องการที่จะดูข้อมูลภาพเอาต์พุตจากการลดขนาดข้อมูล สามารถกระทำได้โดยคลิกที่ปุ่ม Show result บนเมนูไดอะล็อกบ็อกซ์จะได้ผลดังรูปที่ 3.11



รูปที่ 3.11 แสดงภาพที่ได้ภายหลังการผ่านวิธีการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเล็ต

และเมื่อคลิกปุ่ม Result ในรูปที่ 3.11 จะมีไดอะล็อกบ็อกซ์แสดงผลของการลดขนาดข้อมูลขึ้นมา แสดงดังรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงผลของการลดขนาดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

การทดลองในบทนี้จะทำการนำอัลกอริทึมของการลดขนาดข้อมูลภาพด้วยวิธีการเอ็มเบดอิดซีโร่ทรี ที่จะทำการนำสัมประสิทธิ์ที่ได้จากการนำข้อมูลภาพมาผ่านการแปลงเป็นโมเดลของข้อมูล ด้วยวิธีการแปลงเวฟเล็ตแบบไม่ต่อเนื่อง เพื่อทดสอบผลของการลดขนาดข้อมูล

จากบทที่ 2 พบว่าฟังก์ชันเวฟเล็ตมีหลายรูปแบบ ซึ่งแต่ละรูปแบบจะให้ผลของการทำการแปลงเวฟเล็ตที่แตกต่างกัน การที่จะเลือกใช้ฟังก์ชันเวฟเล็ตชนิดใดนั้น จะต้องทำการพิจารณาเลือกใช้ฟังก์ชันที่เหมาะสมกับฟังก์ชันสเกลลิ่ง ที่จะนำมาใช้เป็นองค์ประกอบพื้นฐานของสัญญาณที่ต้องการวิเคราะห์ โดยจะขึ้นอยู่กับการที่จะนำไปใช้งาน โดยถ้าเป็นสัญญาณที่ไม่จำเป็นต้องพิจารณาถึงความต่อเนื่องของข้อมูลก็สามารถใช้เวฟเล็ตแบบฮาร์ ดังที่แสดงตัวอย่างในการออกแบบไว้ในบทที่ 3 ซึ่งง่ายต่อการคำนวณมาเป็นฟังก์ชันพื้นฐานในการแปลงเวฟเล็ต แต่ถ้าต้องการความต่อเนื่องของข้อมูลหลังจากการแปลงเวฟเล็ต ก็อาจใช้ฟังก์ชันเวฟเล็ตที่มีลักษณะที่ให้ความต่อเนื่องของข้อมูลมากขึ้น เช่น แบบดาร์บีชี (Daubechies)[3] เป็นต้น

ดังนั้นการทดลองที่ 4.1 จะทำการทดลองการแปลงเวฟเล็ตแบบไม่ต่อเนื่องที่มีลักษณะของการกรองความถี่เป็นช่วงๆ ดังที่กล่าวมาแล้วในบทที่ 2 ในการทดลองที่ 4.2 ทำการทดลองตั้งค่าเทรชโฮลเพื่อดูค่าของสัมประสิทธิ์และภาพที่สร้างกลับขึ้นมา ภายหลังการปัดค่าเทรชโฮล การทดลองที่ 4.3 จะนำสัมประสิทธิ์ที่เหลือจากการปัดค่าเทรชโฮลมาเข้ารหัสข้อมูลด้วยวิธีการรันเรนจ์ และฮัฟแมน การทดลองที่ 4.4 นำวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี มาเข้ารหัสสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ตแบบไม่ต่อเนื่อง การทดลองที่ 4.5 ทดลองเปลี่ยนเวฟเล็ตฟังก์ชัน แล้วนำมาเข้ารหัสข้อมูล เพื่อดูผลของการลดขนาดข้อมูล การทดลองที่ 4.6 นำภาพต่างมาทดลองกับวิธีการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเล็ตแบบไม่ต่อเนื่องแล้วนำสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ตมาเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี

ในปริญญานิพนธ์นี้ใช้โปรแกรม MATLAB รุ่น 5.1 ช่วยในการวิเคราะห์คำนวณผล โดยใช้วิธีการแปลงเวฟเล็ตตามผังงานที่ได้กล่าวมาแล้ว ในบทที่ 3 มาเขียนเป็นโปรแกรมคำสั่งบน MATLAB เพื่อใช้ในการวิเคราะห์ซึ่งจะได้โปรแกรมหลักสำหรับการแปลงข้อมูลภาพเป็นสัมประสิทธิ์ของเวฟเล็ตที่อยู่ในภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4.1 ทดลองโปรแกรมสำหรับการแปลงเวฟเล็ต กับข้อมูลภาพ โดยนำเอาภาพต้นฉบับมาทำการแปลงเวฟเล็ต ด้วยวิธีการแปลงแบบมาตรฐาน และไม่มาตรฐาน ที่ใช้สัมประสิทธิ์ของฮาร์เป็นเวฟเล็ตฟังก์ชัน แล้วทำการสร้างภาพกลับขึ้นมาจากสัมประสิทธิ์เวฟเล็ตของภาพนั้น ตามคำสั่งจำลองที่ 3.1ถึง 3.4 หรือโปรแกรมที่ 1 ถึงโปรแกรมที่ 4 ในภาคผนวก นำผลที่ได้จากการแปลงมาพิจารณาหาความผิดพลาดจากการแปลงเวฟเล็ต

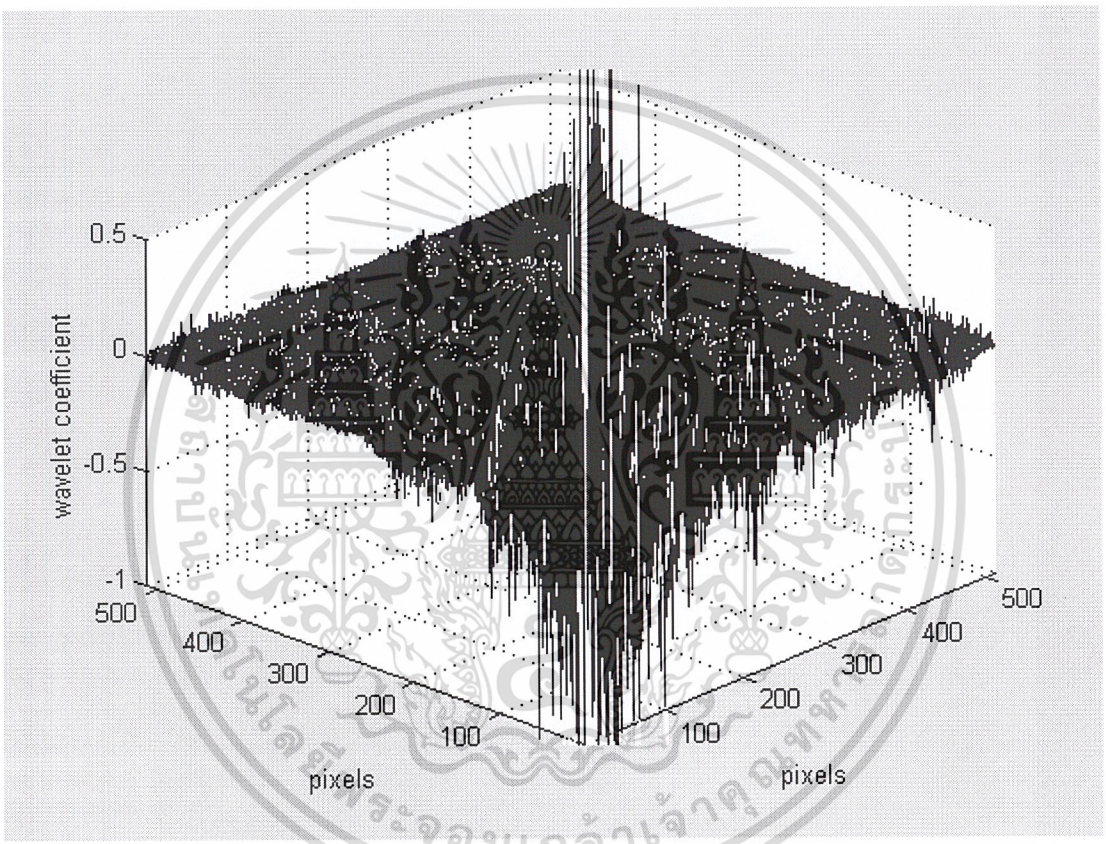
ในการทดลองจะใช้ภาพ Lena ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบ ดังแสดงในรูปที่ 4.1

รูปที่ 4.1 แสดงภาพต้นแบบที่ใช้ในการทดลอง (Lena ขนาด 512×512 จุดภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

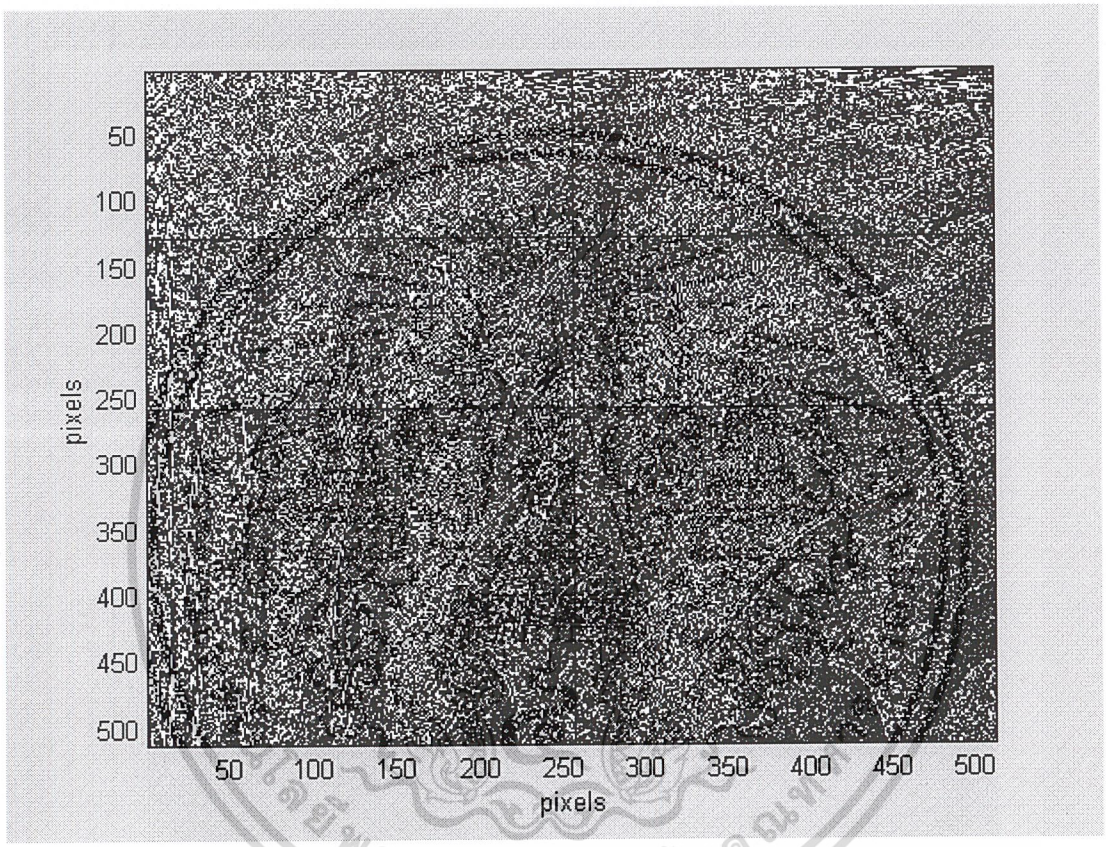
4.1.1) ทดลองแปลงภาพ โดยการใช้การแปลงเวฟเล็ต แบบมาตรฐาน จะได้สัมประสิทธิ์ของภาพแสดงเป็นกราฟ 3 มิติ ในรูปที่ 4.2 และแสดงภาพที่ถูกแบ่งเป็นสับแบนด์ในรูปที่ 4.3 และเมื่อนำสัมประสิทธิ์ เวฟเล็ตของภาพในรูปที่ 4.2 มาแปลงกลับ โดยวิธีการเดิมจะแสดงในรูปที่ 4.4

ผลการทดลอง 4.1.1



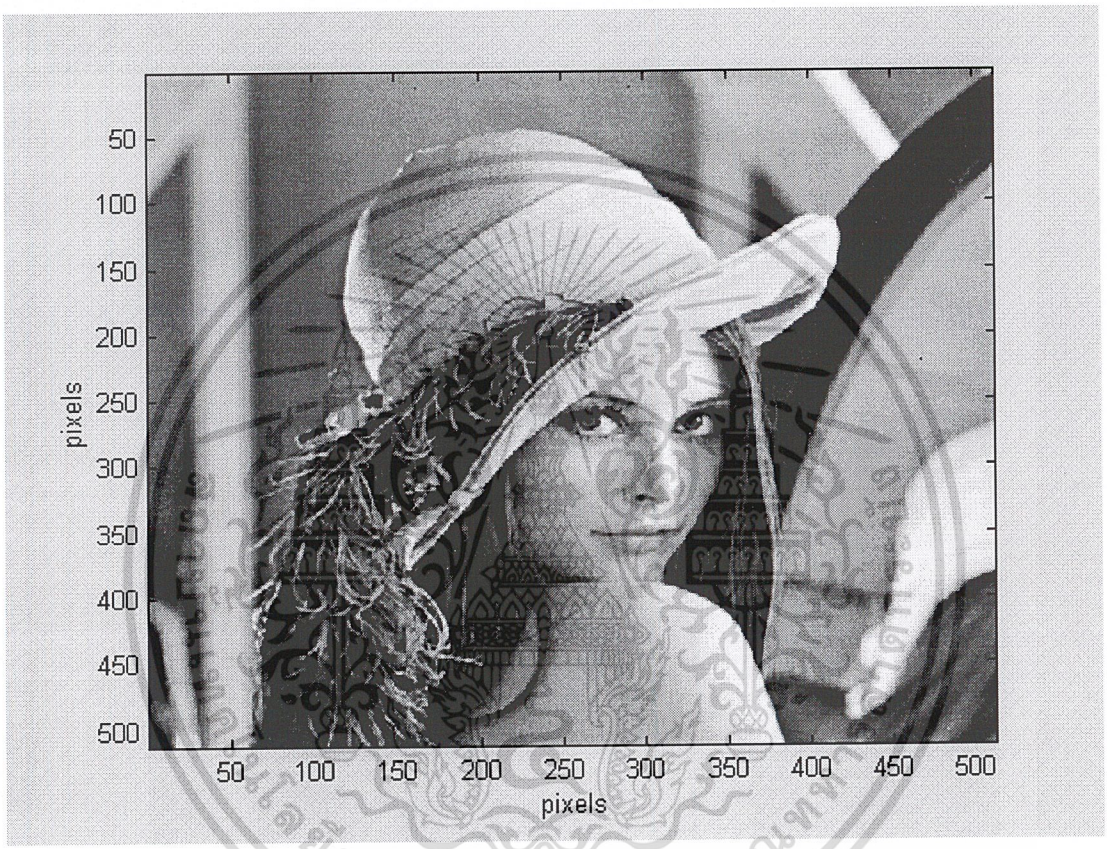
รูปที่ 4.2 แสดงสัมประสิทธิ์ของภาพภายหลังการแปลงเวฟเล็ตโดยวิธีมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงภาพภายหลังจากการแปลงเวฟเล็ต โดยวิธีมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

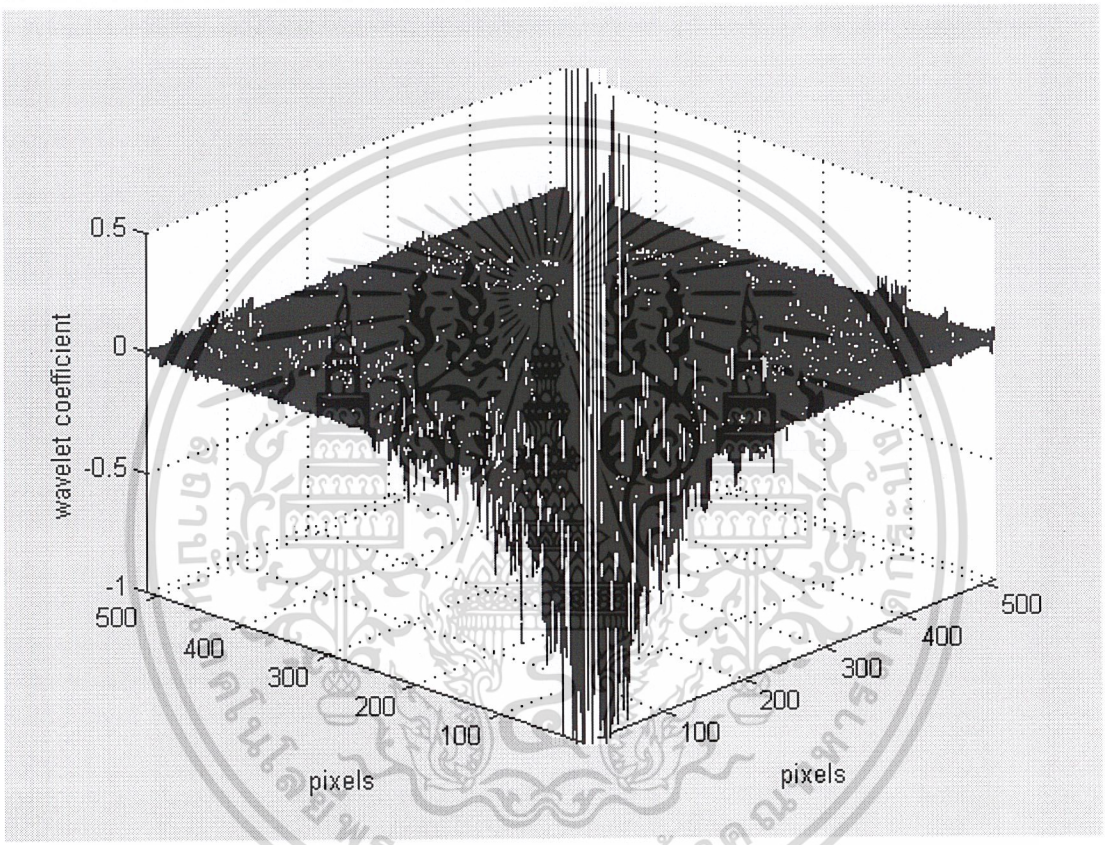


รูปที่ 4.4 แสดงภาพที่สร้างกลับขึ้นมาจากสัมประสิทธิ์เวฟเล็ตในรูปที่ 4.2 โดยวิธีมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

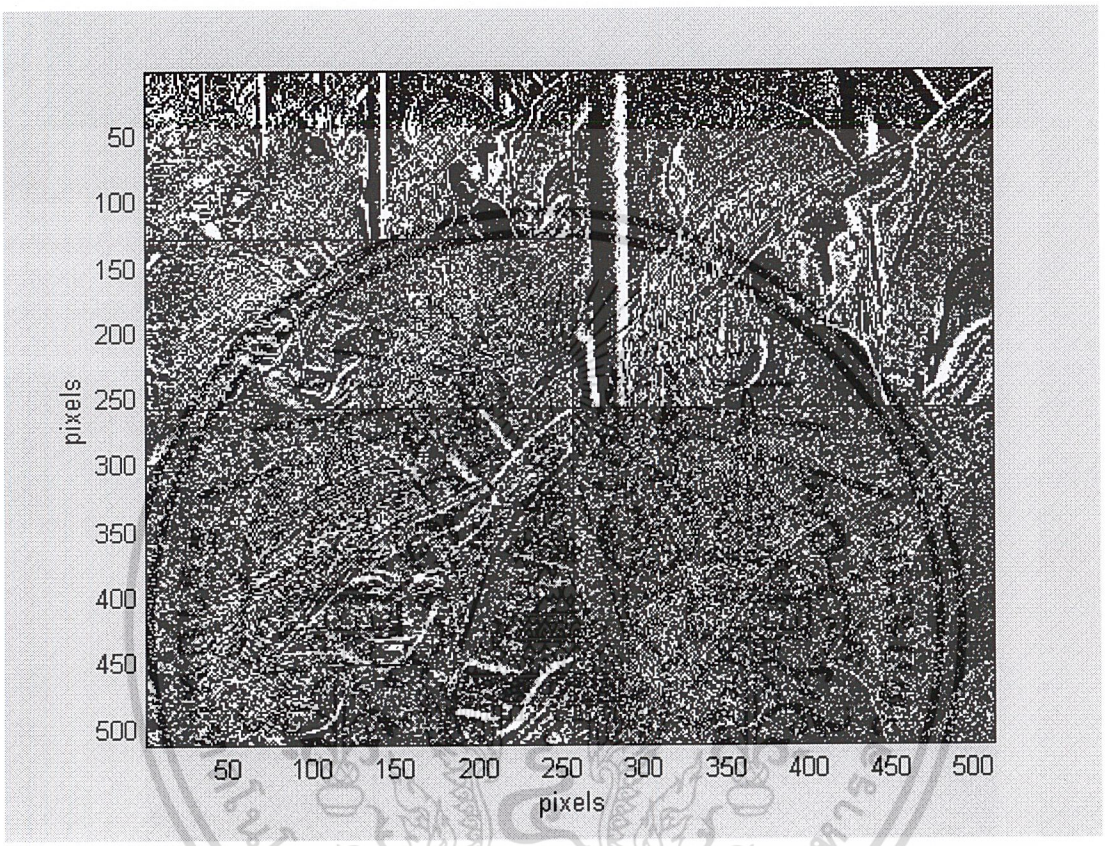
4.1.2) ทำการทดลองเหมือนในข้อที่ 4.1.1 แต่เปลี่ยนลักษณะการแปลงสัมประสิทธิ์ของภาพด้วยเวฟเล็ตเป็นแบบไม่มาตรฐาน จะได้สัมประสิทธิ์ของภาพแสดงเป็นกราฟ 3 มิติ ในรูปที่ 4.5 และแสดงภาพที่ถูกแบ่งเป็นสับแบนด์ในรูปที่ 4.6 และเมื่อนำสัมประสิทธิ์เวฟเล็ตของภาพในรูปที่ 4.5 มาแปลงกลับ โดยวิธีการเดิมจะได้รูปดังแสดงในรูปที่ 4.7

ผลการทดลอง 4.1.2



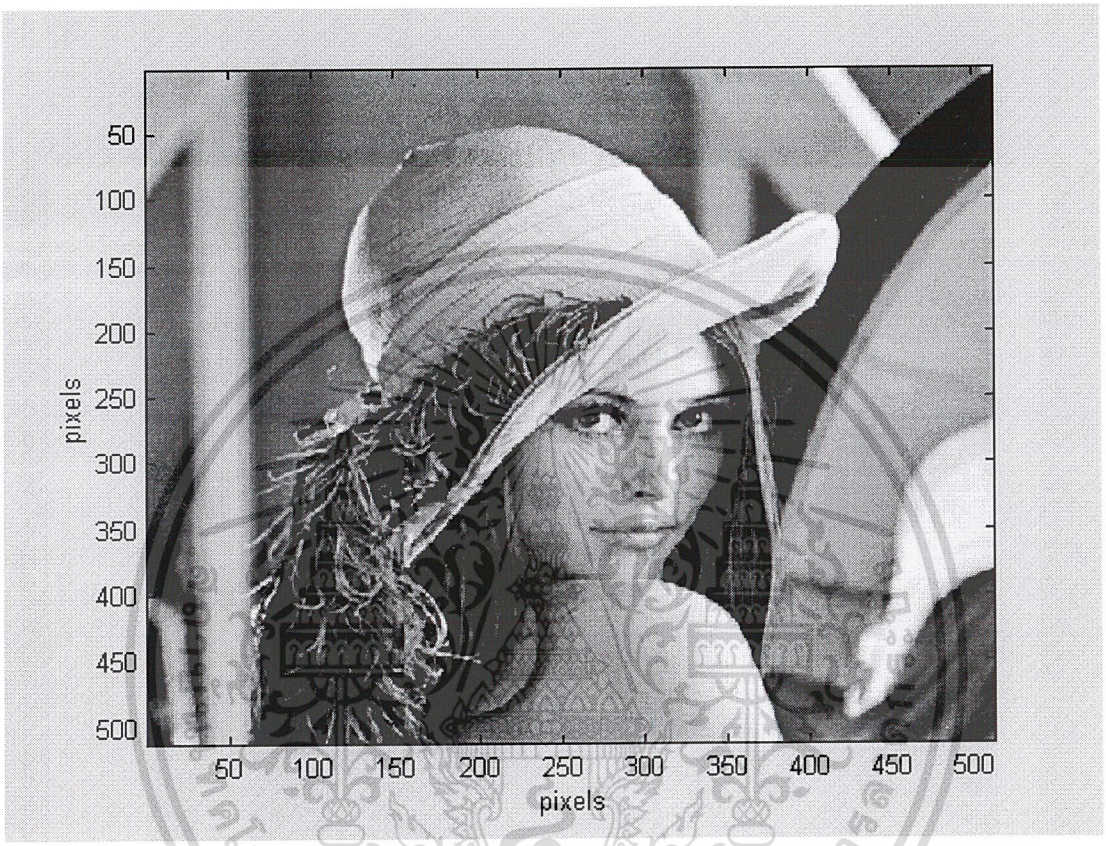
รูปที่ 4.5 แสดงสัมประสิทธิ์ของภาพภายหลังการแปลงเวฟเล็ตโดยวิธีการแปลงภาพแบบไม่มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงภาพภายหลังจากการแปลงเวฟเล็ต โดยวิธีการแปลงภาพแบบไม่มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

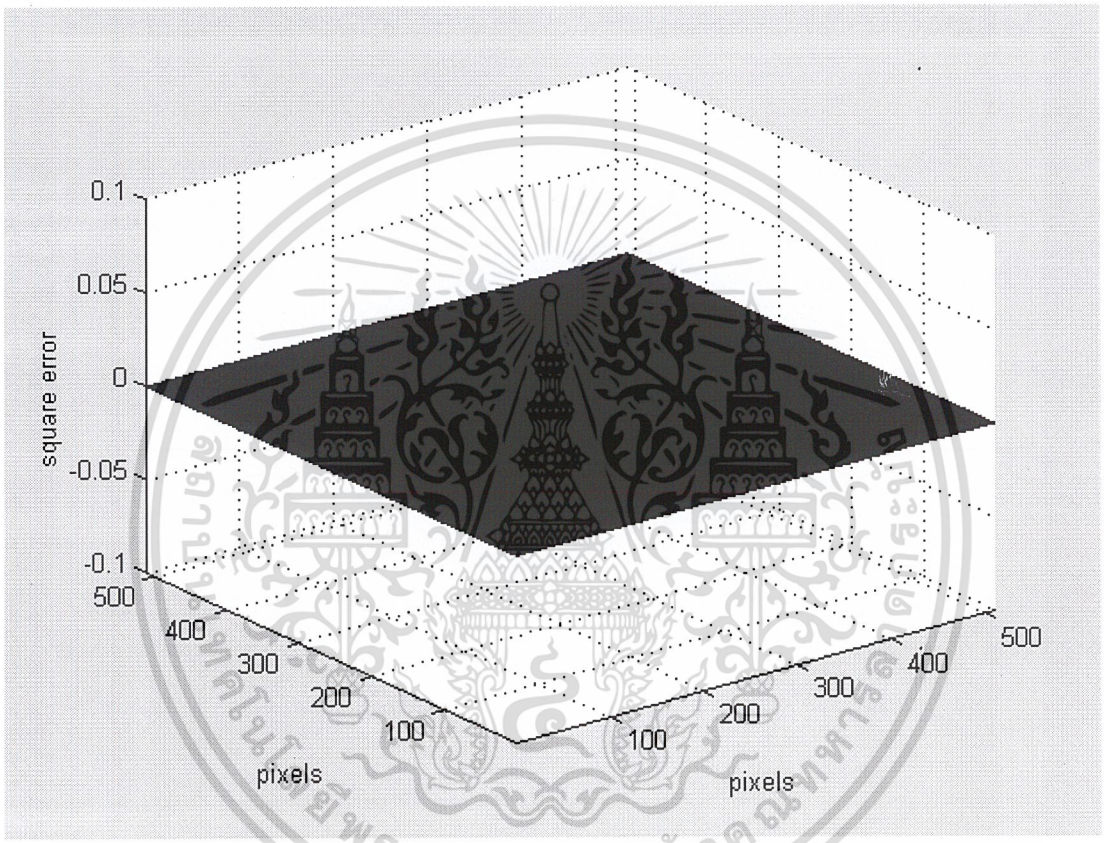


รูปที่ 4.7 แสดงภาพที่สร้างกลับขึ้นมาจากสัมประสิทธิ์เวฟเล็ตในรูปที่ 4.5 โดยวิธีการแปลงภาพแบบไม่มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

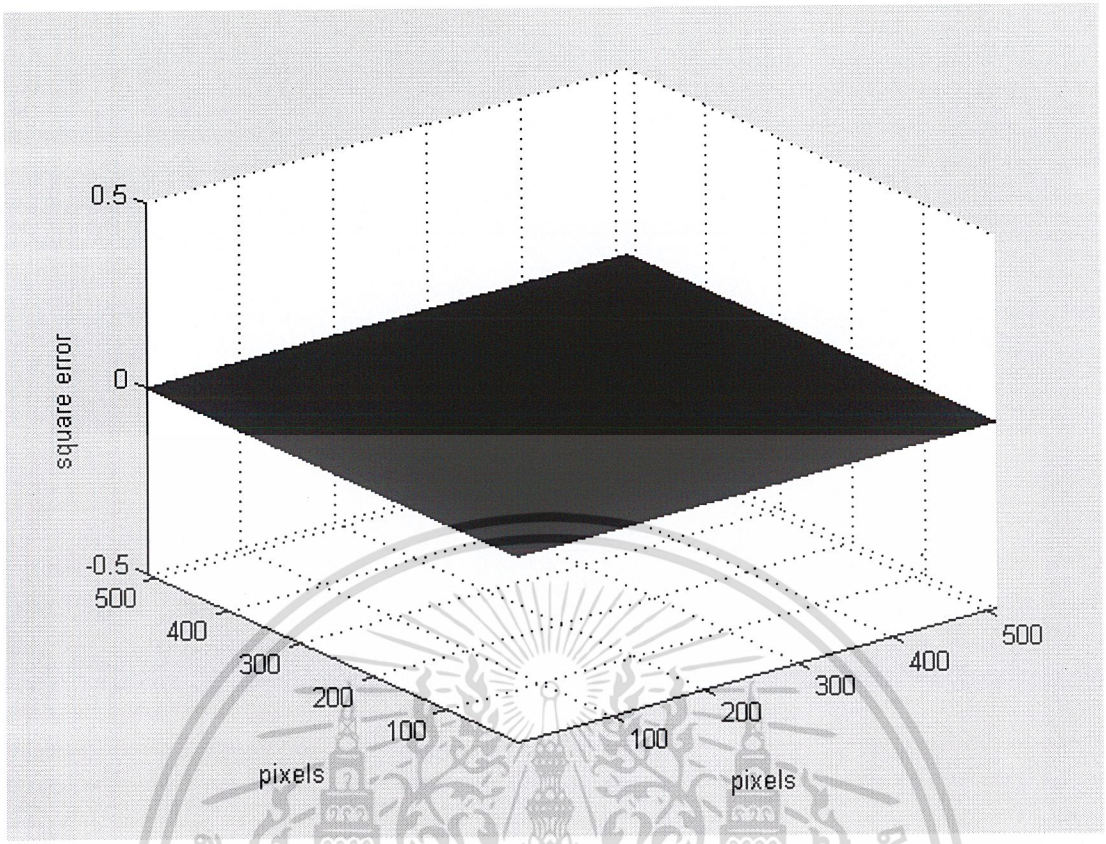
4.1.3) นำภาพที่ได้จากการแปลงกลับ แล้วนำมาหาค่าความผิดพลาดของข้อมูลจากการแปลงกลับ นำภาพต้นแบบเปรียบเทียบกับภาพที่สร้างกลับขึ้นมาจากสัมประสิทธิ์เวฟเล็ตแบบมาตรฐาน(รูปที่ 4.3) แสดงดังรูปที่ 4.8 และนำภาพต้นแบบมาเปรียบเทียบเช่นเดียวกันกับภาพที่สร้างกลับมาจากสัมประสิทธิ์เวฟเล็ตแบบไม่มาตรฐาน(รูปที่ 4.7) เพื่อตรวจสอบความผิดพลาดของข้อมูลภายหลังการแปลงเวฟเล็ต

ผลการทดลอง 4.1.3



รูปที่ 4.8 แสดงกราฟ 3 มิติ เปรียบเทียบค่าความแตกต่างระหว่างจุดภาพ ของภาพต้นแบบกับภาพที่สร้างกลับขึ้นมาจากการแปลงเวฟเล็ตแบบไม่ต่อเนื่องด้วยวิธีการแปลงแบบมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงกราฟ 3 มิติ เปรียบเทียบค่าความแตกต่างระหว่างจุดภาพ ของภาพต้นแบบกับภาพที่สร้างกลับขึ้นมาจากการแปลงเวฟเล็ตแบบไม่ต่อเนื่องด้วยวิธีการแปลงแบบไม่มาตรฐาน

เนื่องจากการนำข้อมูลภาพมาทำการแปลงเวฟเล็ตแบบไม่ต่อเนื่องที่มีลักษณะเหมือนกับการนำข้อมูลภาพมาผ่านการกรองความถี่ โดยการใช้สัมประสิทธิ์ของเวฟเล็ตฟังก์ชันที่นำมาใช้งานมาเป็นสัมประสิทธิ์ในการกรองความถี่ ซึ่งมีคุณสมบัติการตอบสนองอิมพัลส์ที่จำกัด (FIR Filter)[8] ซึ่งจะให้เฟสของสัญญาณเอาต์พุตที่ได้จากการกรองความถี่เป็นเชิงเส้น เมื่อนำความถี่ที่เป็นองค์ประกอบของสัญญาณที่พิจารณามารวมกัน ก็จะได้สัญญาณตัวเดิมกลับขึ้นมา

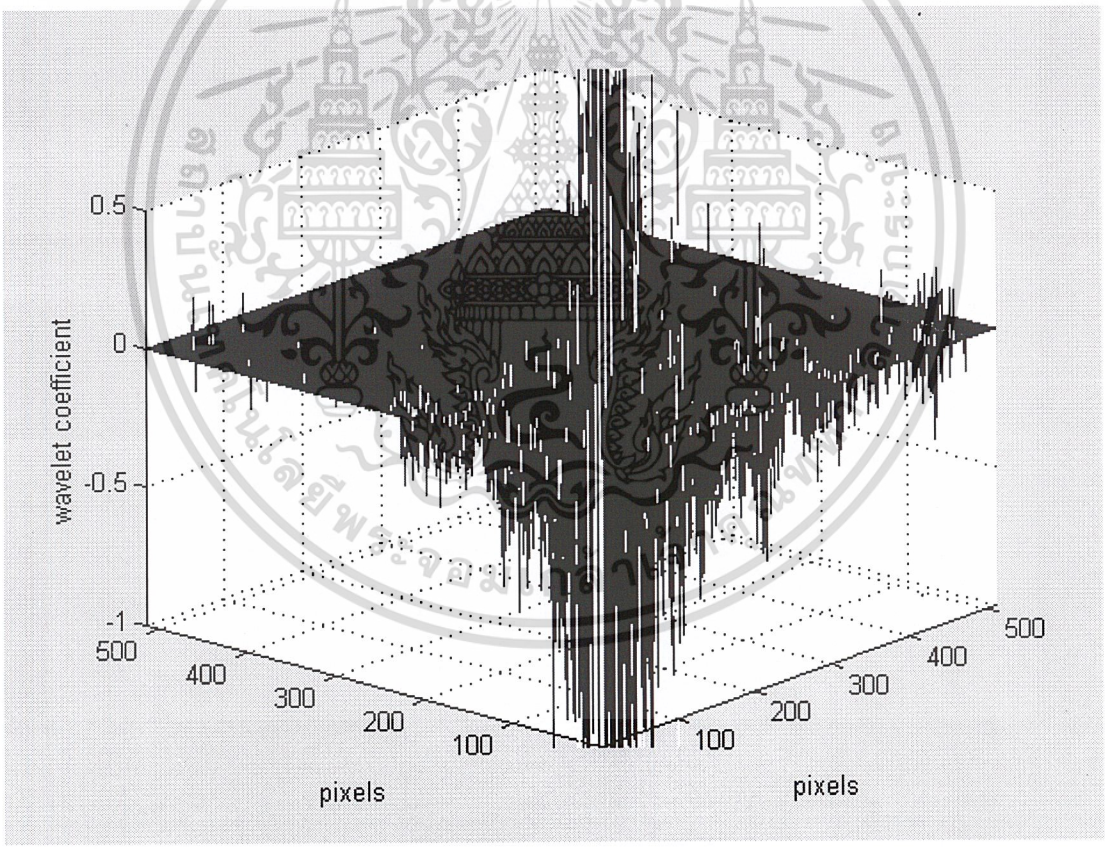
จากการทดลองรูปที่ 4.8 และรูปที่ 4.9 จะเห็นได้ว่าการแปลงสัมประสิทธิ์ของภาพ เป็นสัมประสิทธิ์ของเวฟเล็ต แล้วแปลงกลับจะได้ข้อมูลภาพที่เหมือนเดิม คือข้อมูลภาพไม่มีการสูญเสียเกิดขึ้นในขั้นตอนการแปลงเวฟเล็ต เป็นไปตามทฤษฎีของการแปลงเวฟเล็ต คือข้อมูลภาพต้นแบบถูกแปลงเป็นสัมประสิทธิ์ของการแปลงเวฟเล็ตแล้วเมื่อแปลงกลับจะไม่มีความผิดพลาดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4.2 ทดลองการปัดค่าระดับเทรชโฮลของสัมประสิทธิ์เวฟเล็ตของภาพ ที่ได้จากการแปลงเวฟเล็ตแบบมาตรฐานและไม่มาตรฐาน โดยใช้สัมประสิทธิ์ของฮาร์เป็นเวฟเล็ตฟังก์ชัน เมื่อทำการตั้งระดับเทรชโฮลที่เท่ากัน เพื่อดูผลเปรียบเทียบการปัดค่าสัมประสิทธิ์เวฟเล็ตของภาพที่ได้จากการแปลงแบบมาตรฐานและไม่มาตรฐาน ว่าเมื่อระดับเทรชโฮลเท่ากันแล้วจะมีผลต่อข้อมูลภาพที่ถูกสร้างกลับขึ้นมาอย่างไร และจำนวนศูนย์ในสัมประสิทธิ์ที่ได้หลังการปัดระดับเทรชโฮล

4.2.1) นำสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ตของภาพต้นแบบด้วยวิธีการแปลงแบบมาตรฐานโดยใช้สัมประสิทธิ์ของฮาร์ เป็นเวฟเล็ตฟังก์ชันในรูปที่ 4.2 มาผ่านการตั้งค่าเทรชโฮลที่ 0.1236 (หาได้จากโปรแกรมที่ 7 ในภาคผนวก หรือ คำสั่งจำลองที่ 3.5 ในบทที่ 3) โดยค่าสัมบูรณ์ตัวใดของสัมประสิทธิ์มีค่าน้อยกว่าค่าเทรชโฮลจะปัดให้มีค่าเป็นศูนย์ จะได้ผลดังรูปที่ 4.10

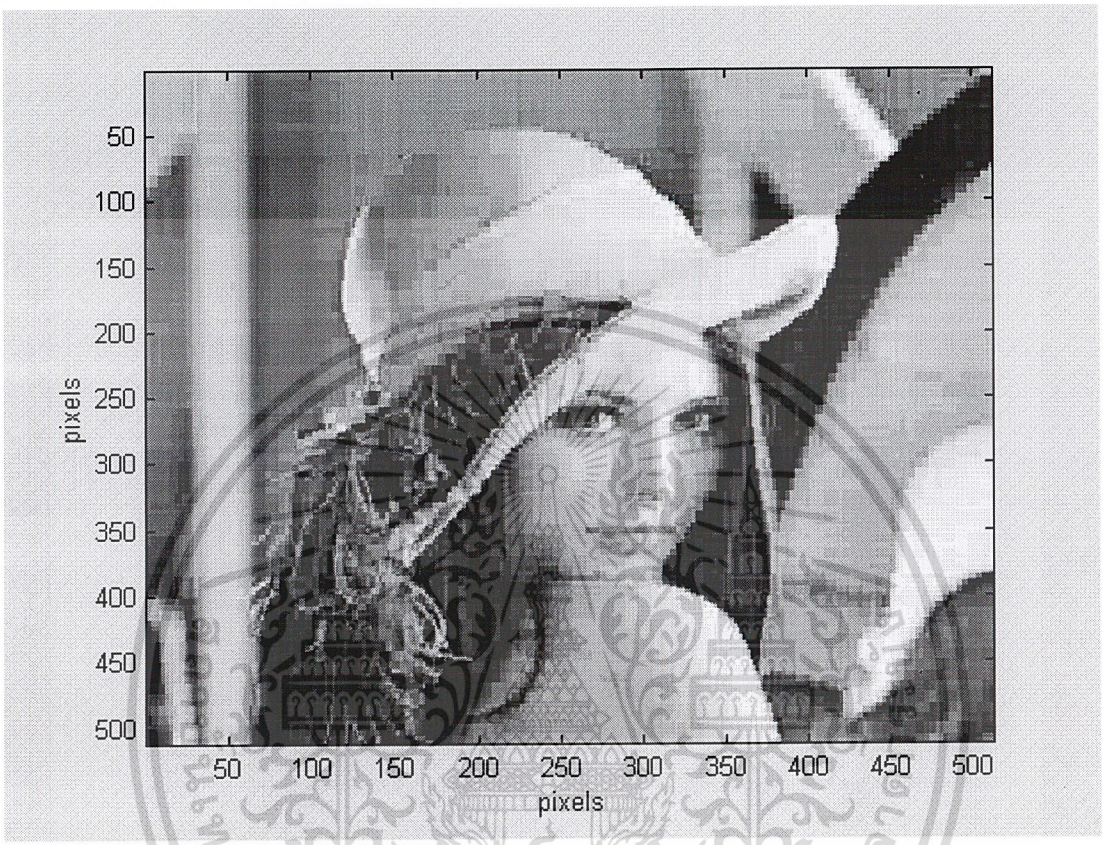
ผลการทดลอง 4.2.1



รูปที่ 4.10 แสดงผลของการปัดค่าเทรชโฮลในสัมประสิทธิ์เวฟเล็ต
ของภาพในรูปที่ 4.2 ที่ค่าเทรชโฮลเท่ากับ 0.1236

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำสัมประสิทธิ์ที่เหลือจากการปัดค่าเทรชโฮลมาแปลงกลับเป็นข้อมูลภาพในระดับสีเทา ตามเดิม จะได้ภาพที่แสดงในรูปที่ 4.11



รูปที่ 4.11 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพ
ที่ถูกแปลงด้วยวิธีมาตรฐานที่ถูกปัดค่าเทรชโฮลที่ 0.1236

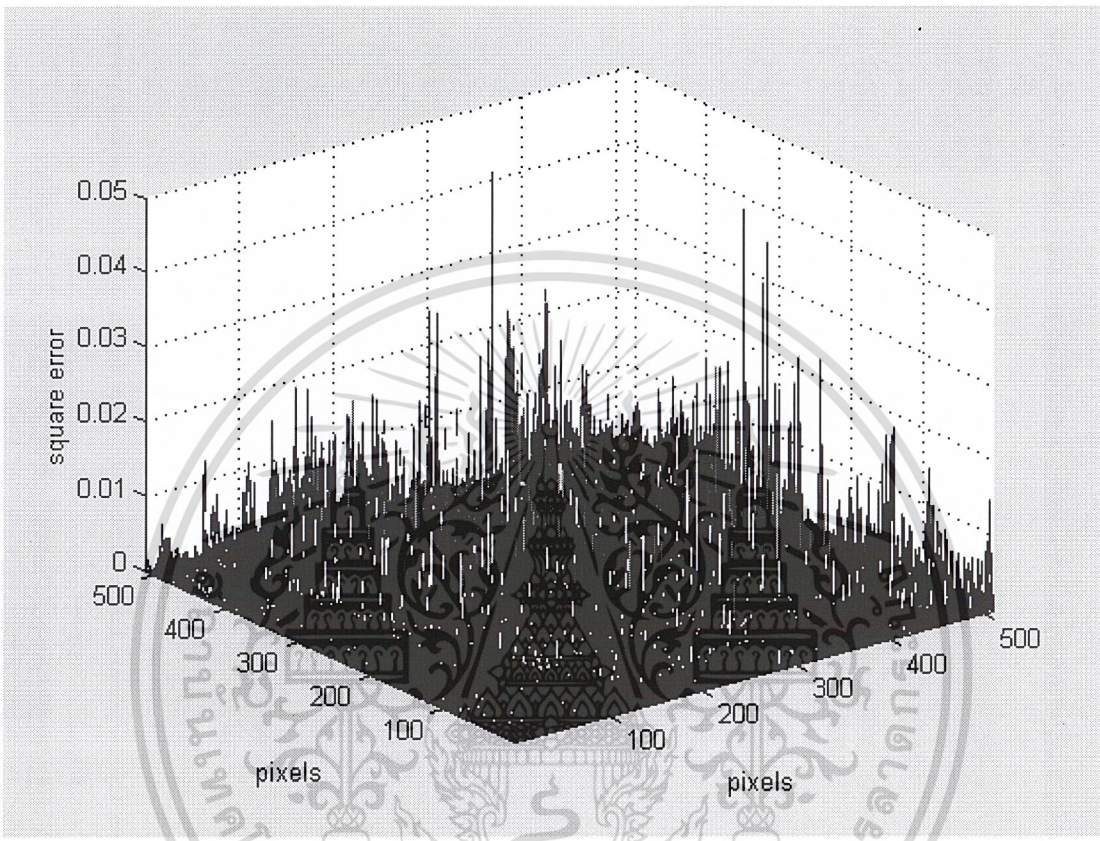
จากรูปที่ 4.11 นำมาหาค่าความผิดพลาดของภาพเทียบกับภาพต้นแบบ พบว่ามีค่าเฉลี่ยของความผิดพลาดกำลังสองของภาพ (mean square error:MSE) ตามหัวข้อที่ 2.5.4 จะได้

MSE = 100.04

จำนวนสัมประสิทธิ์ที่มีค่าเท่ากับศูนย์ = 257,069 ตัว

สามารถแสดงกราฟเปรียบเทียบจุดผิดพลาดระหว่างภาพต้นแบบและภาพภายหลังการสร้างกลับมาจากการแปลงเวฟเล็ต ซึ่งผ่านการตั้งค่าเทรชโฮลที่ 0.1236 แสดงในรูปที่ 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

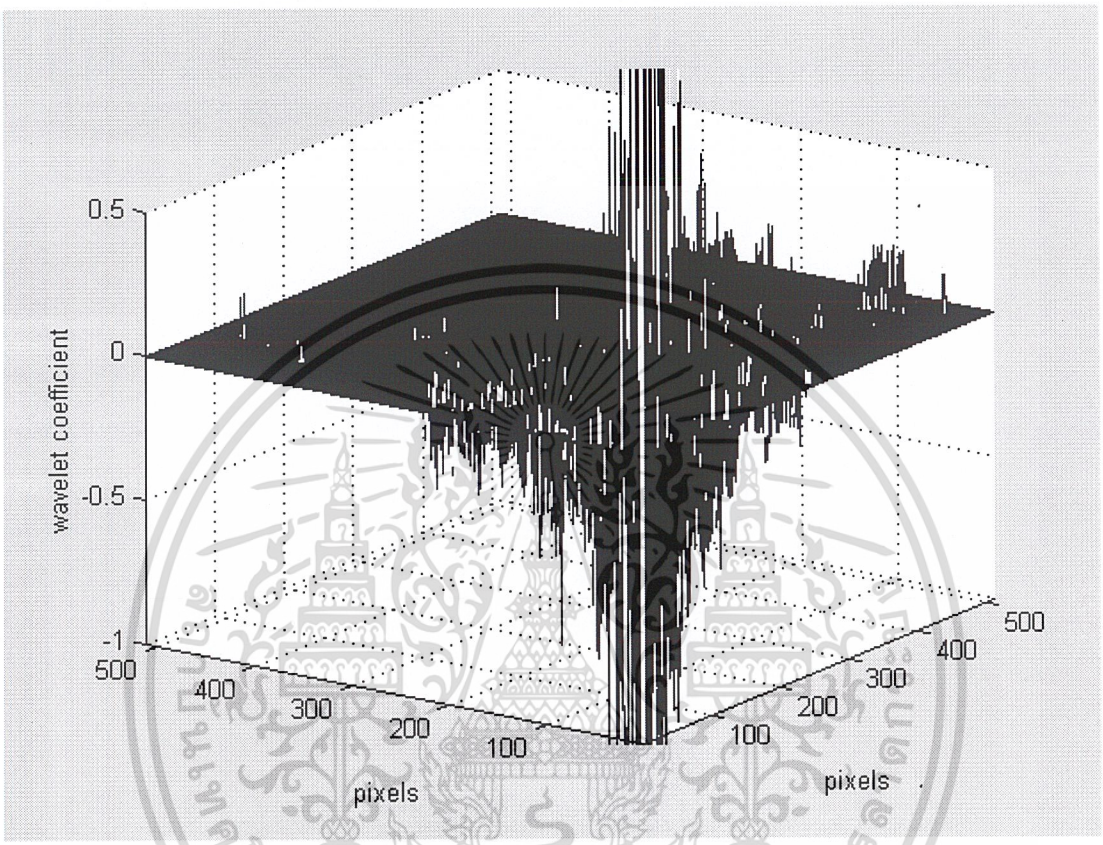


รูปที่ 4.12 กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจาก
สัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีมาตรฐานที่ถูกปัดค่าเทรชโฮลที่ 0.1236

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2) เปลี่ยนวิธีการแปลงเวฟเล็ต เป็นการแปลงแบบไม่มาตรฐาน แล้วทำการทดลองเหมือนข้อที่ 4.2.1

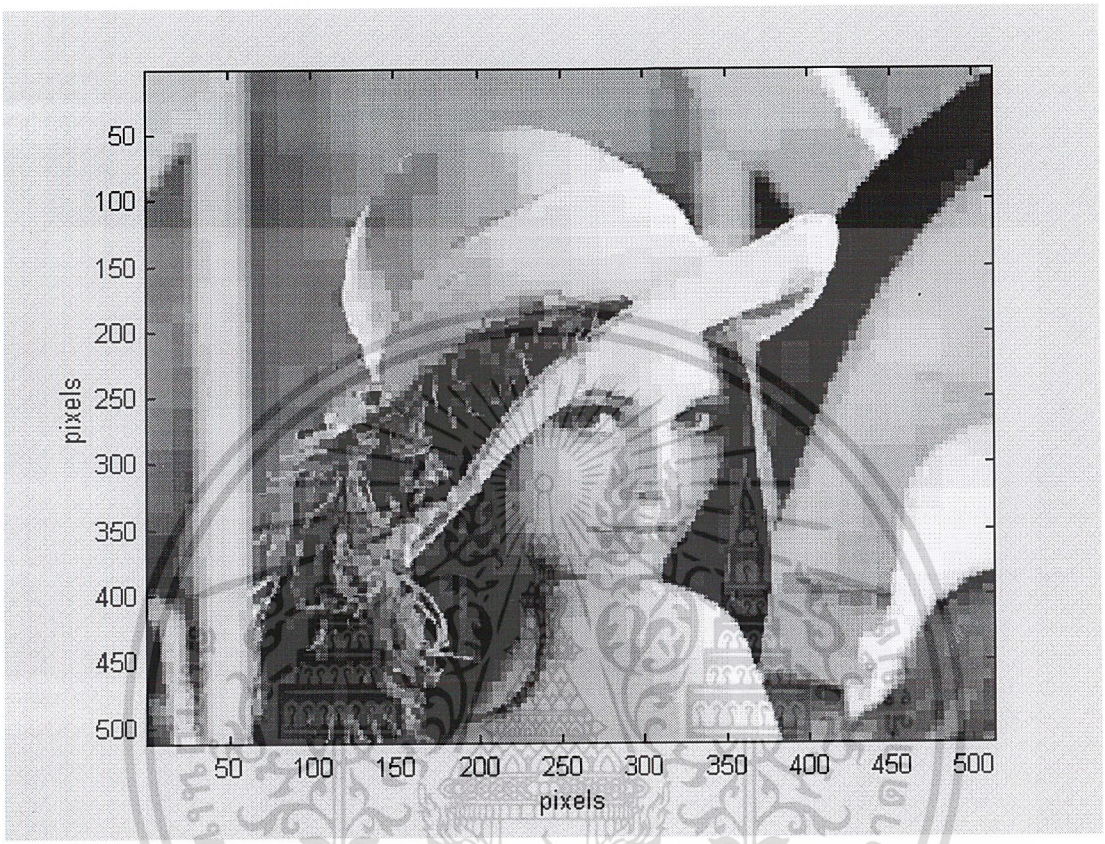
ผลการทดลอง 4.2.2



รูปที่ 4.13 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพ
ที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปัดค่าเทรชโฮลที่ 0.1236

นำสัมประสิทธิ์ที่เหลือจากการปัดค่าเทรชโฮลมาแปลงกลับเป็นข้อมูลภาพในระดับสีเทา ตามเดิม จะได้ภาพที่แสดงในรูปที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพ
ที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปัดค่าเทรชโฮลที่ 0.1236

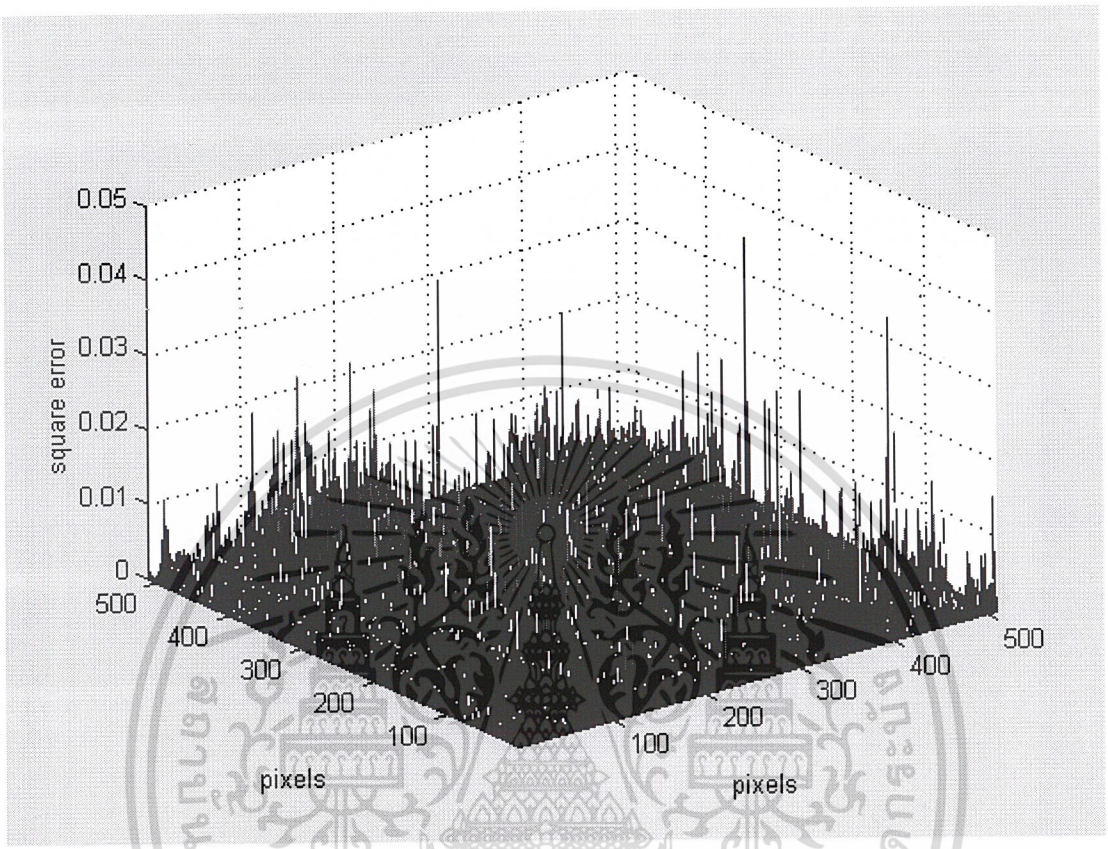
จากรูปที่ 4.14 นำมาหาค่าความผิดพลาดของภาพเทียบกับภาพต้นแบบ พบว่ามีค่าความผิดพลาดที่คิดค่าเฉลี่ยกำลังสองของความผิดพลาด (ตามหัวข้อที่ 2.5.4) จะได้

MSE = 97.73

จำนวนสัมประสิทธิ์ที่มีค่าเท่ากับศูนย์ = 257,250 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแสดงกราฟเปรียบเทียบจุดผิดพลาดระหว่างภาพต้นแบบและภาพภายหลังการสร้างกลับมาจากการแปลงเวฟเล็ต ซึ่งผ่านการตั้งค่าเทรชโฮลที่ 0.1236 แสดงในรูปที่ 4.15

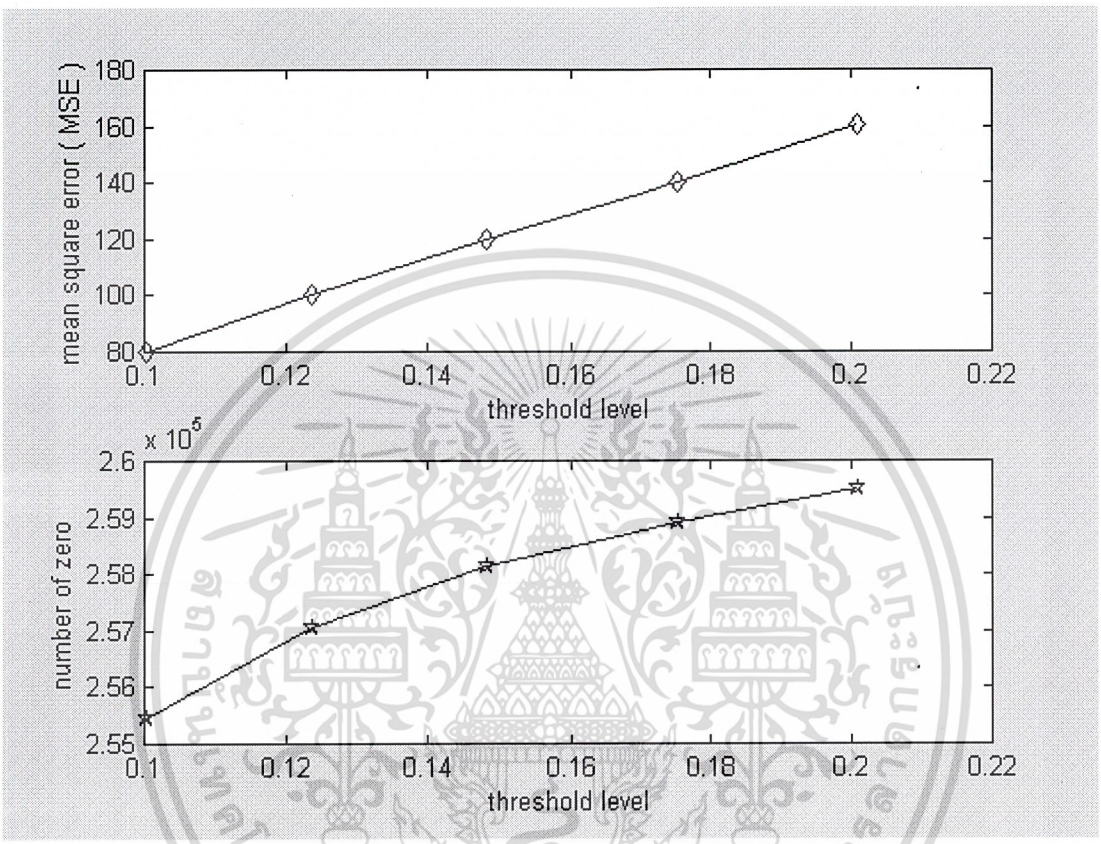


รูปที่ 4.15 กราฟแสดงค่าความผิดพลาดของภาพต้นแบบเทียบกับภาพที่สร้างจากสัมประสิทธิ์เวฟเล็ตของภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐานที่ถูกปัดค่าเทรชโฮลที่ 0.1236

จากการทดลองที่ 4.2 จะพบว่า เมื่อนำภาพมาผ่านการแปลงเวฟเล็ตแบบมาตรฐานและไม่มาตรฐาน แล้วผ่านการกำหนดค่าเทรชโฮลที่เท่ากันแล้วนั้น ผลของภาพเมื่อถูกปัดค่าเทรชโฮลแล้วนำมาแปลงกลับเป็นภาพขึ้นมาแล้วจะมีคุณลักษณะของภาพที่แตกต่างกัน คือจะมีค่าความผิดพลาดไม่เท่ากัน โดยจากการทดลองที่ผ่านมาที่ระดับเทรชโฮลที่ 0.1236 พบว่าภาพที่ถูกแปลงด้วยวิธีไม่มาตรฐาน จะมีค่าความผิดพลาดน้อยกว่าภาพที่ถูกแปลงด้วยวิธีมาตรฐาน และในภาพที่แปลงแบบไม่มาตรฐานจะมีจำนวนศูนย์เกิดขึ้นมากกว่าอีกด้วย โดยจำนวนศูนย์นี้สามารถที่จะบอกถึงความสามารถในการที่จะนำเอารหัสมาแทนสัมประสิทธิ์ที่เหลืออยู่ได้น้อยลงเมื่อเวลานำไปเข้ารหัสได้ ซึ่งถือได้ว่าเป็นการลดขนาดข้อมูลวิธีหนึ่ง

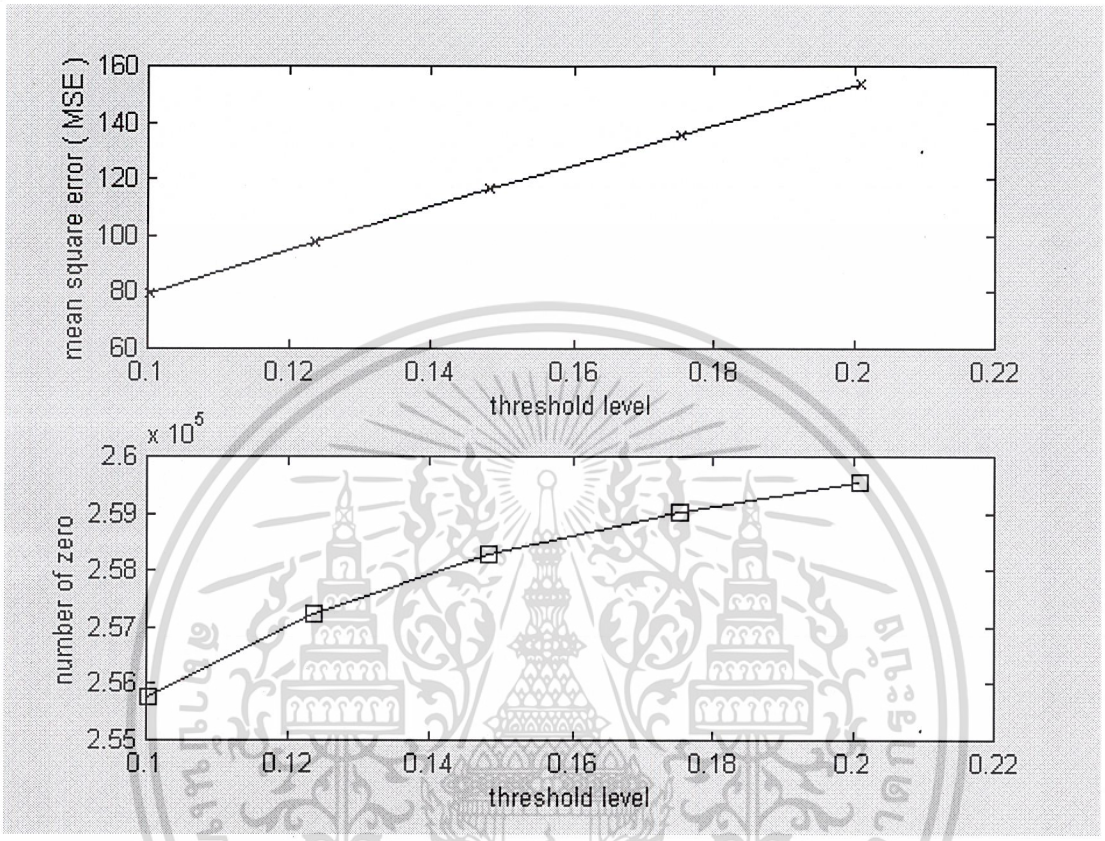
สามารถแสดงผลการเปรียบเทียบวิธีการแปลงแบบมาตรฐานและไม่มาตรฐานของภาพ Lena ขนาด 512×512 โดยเทียบระหว่างระดับเทรชโฮล กับค่าความผิดพลาดของภาพเมื่อสร้างกลับ และจำนวนศูนย์ที่เกิดขึ้นในสัมประสิทธิ์เวฟเล็ตของภาพ แสดงในรูปที่ 4.16 ถึง 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



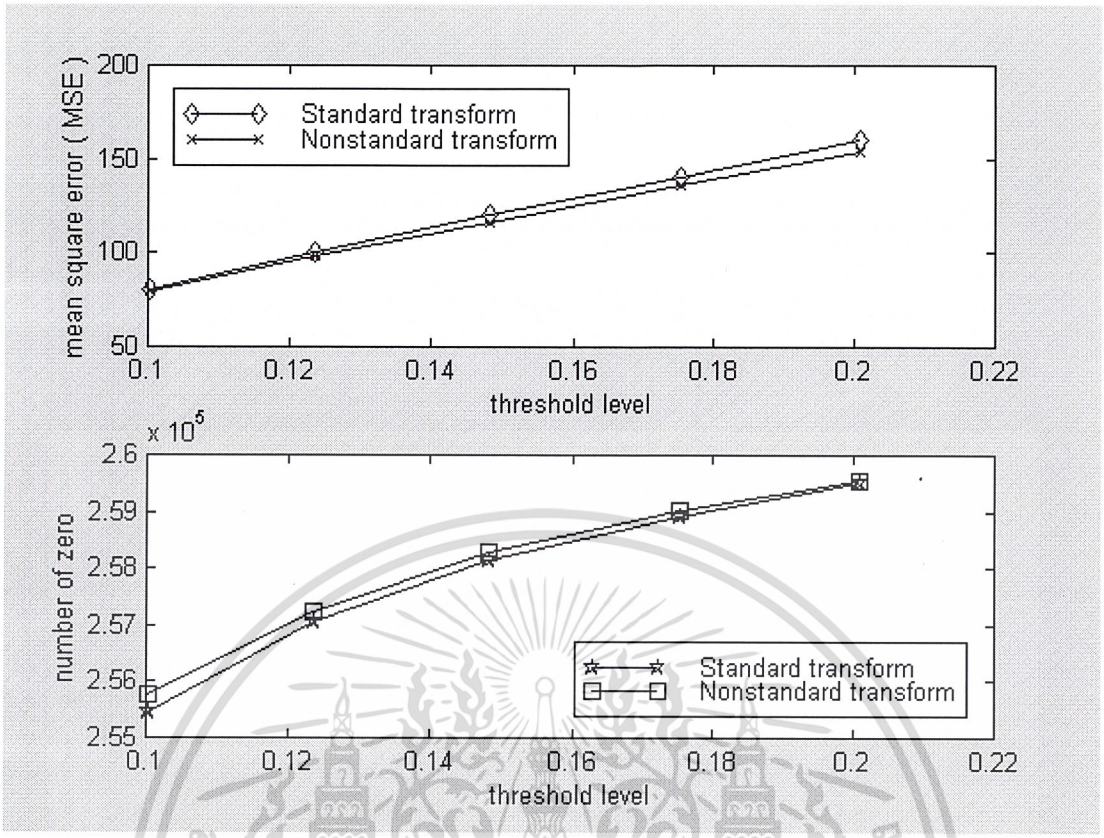
รูปที่ 4.16 แสดงผลของระดับเทรชโฮลที่ระดับต่างๆ ว่ามีผลต่อค่าความผิดพลาดของภาพ และมีจำนวนศูนย์ในสัมประสิทธิ์เวฟเล็ตของภาพที่ได้จากการแปลงแบบมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 แสดงผลของระดับเทรชโฮลที่ระดับต่างๆ ว่ามีผลต่อค่าความผิดพลาดของภาพ และมีจำนวนศูนย์ในสัมประสิทธิ์เวฟเล็ตของภาพที่ได้จากการแปลงแบบไม่มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 เปรียบเทียบระหว่าง รูปที่ 4.16 และรูปที่ 4.17

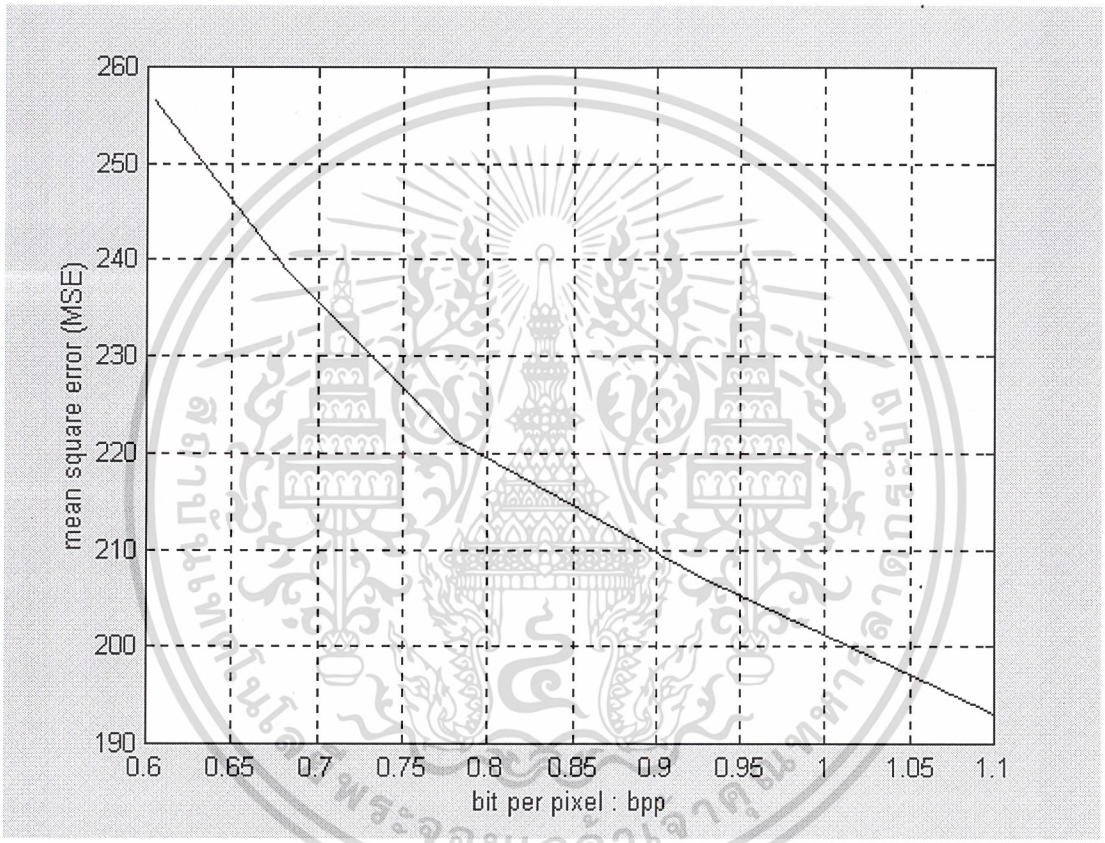
จากการทดลองที่ 4.2 จะเห็นได้ว่าการแปลงเวฟเล็ตแบบไม่มาตรฐานเหมาะสมที่จะนำมาใช้งานในการปัดค่าเทรชโฮลของสัมประสิทธิ์

ลักษณะของการแปลงแบบไม่มาตรฐานเปรียบเสมือนการฟิลเตอร์ภาพในแต่ละแบนด์ออกมา ทำให้เมื่อเวลานำภาพออกไปวิเคราะห์องค์ประกอบของสัญญาณแต่และแบนด์ความถี่จะสามารถกระทำได้ง่ายขึ้น ดังนั้นในการนำเวฟเล็ตมาใช้แปลงข้อมูลภาพ สำหรับการลดขนาดข้อมูลภาพ ที่นำมาใช้ในปริญญานิพนธ์นี้จะใช้วิธีการแปลงแบบไม่มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

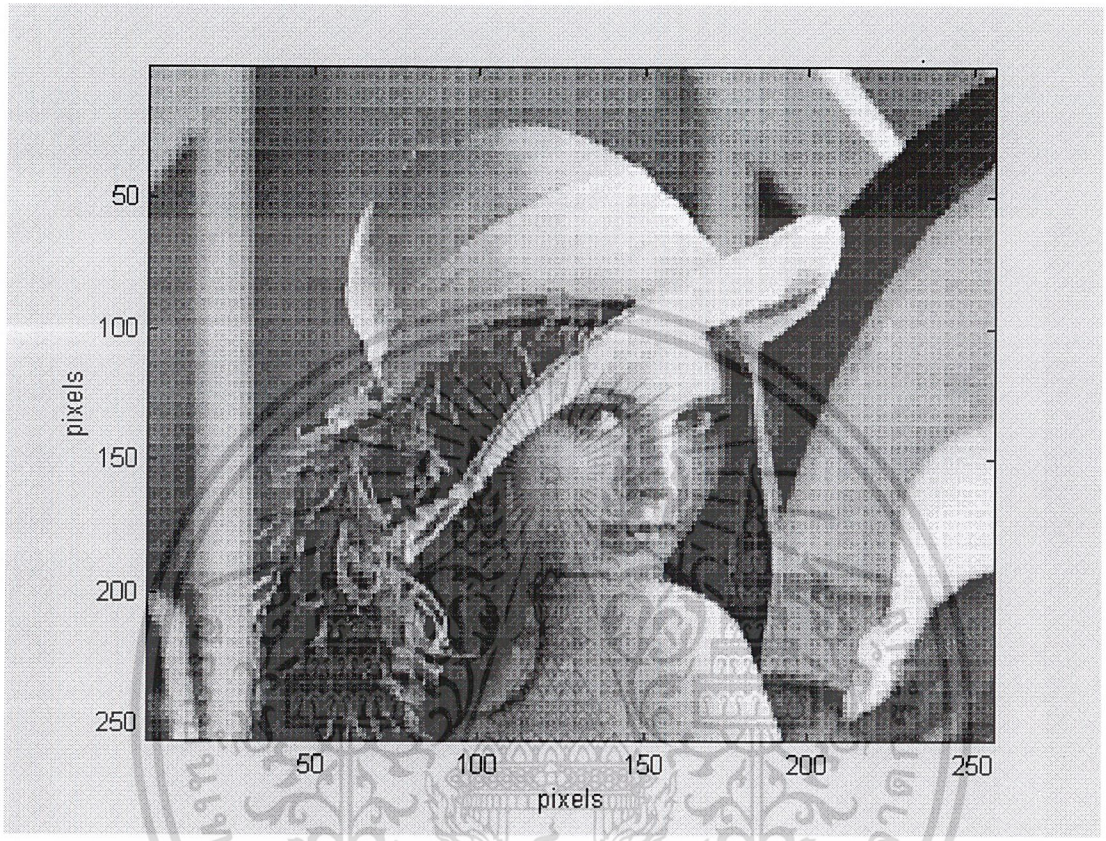
การทดลองที่ 4.3 นำเอาสัมประสิทธิ์ที่ได้จากการตั้งค่าเทรชโฮล ไปทำการจัดระดับข้อมูลแล้วผ่านการเข้ารหัสลดข้อมูลแบบไม่มีการสูญเสียคือเข้ารันเรนจ์ และเข้ารหัสฮัฟแมน ทำการบันทึกผลของข้อมูลที่ได้จากการลดขนาดข้อมูล เทียบค่าคามผิดพลาดที่เกิดขึ้นกับขนาดบิตต่อจุดภาพ (bpp) เพื่อดูผลของการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเล็ตกับข้อมูลภาพ โดยนำภาพ Lena ขนาด 256×256 จุดภาพ เป็นภาพต้นแบบ

ผลการทดลอง 4.3



รูปที่ 4.19 แสดงผลการลดขนาดข้อมูลจากการตั้งเทรชโฮลแล้วนำมาเข้ารหัสข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 แสดงภาพ Lena ขนาด 256×256 จุดภาพ โดยมีขนาดข้อมูลภายหลังการลดขนาดเท่ากับ 1.101 bpp (ขนาดของข้อมูลเท่ากับ 9,016 ไบต์) ค่า MSE = 193.09 และ PSNR = 18.06

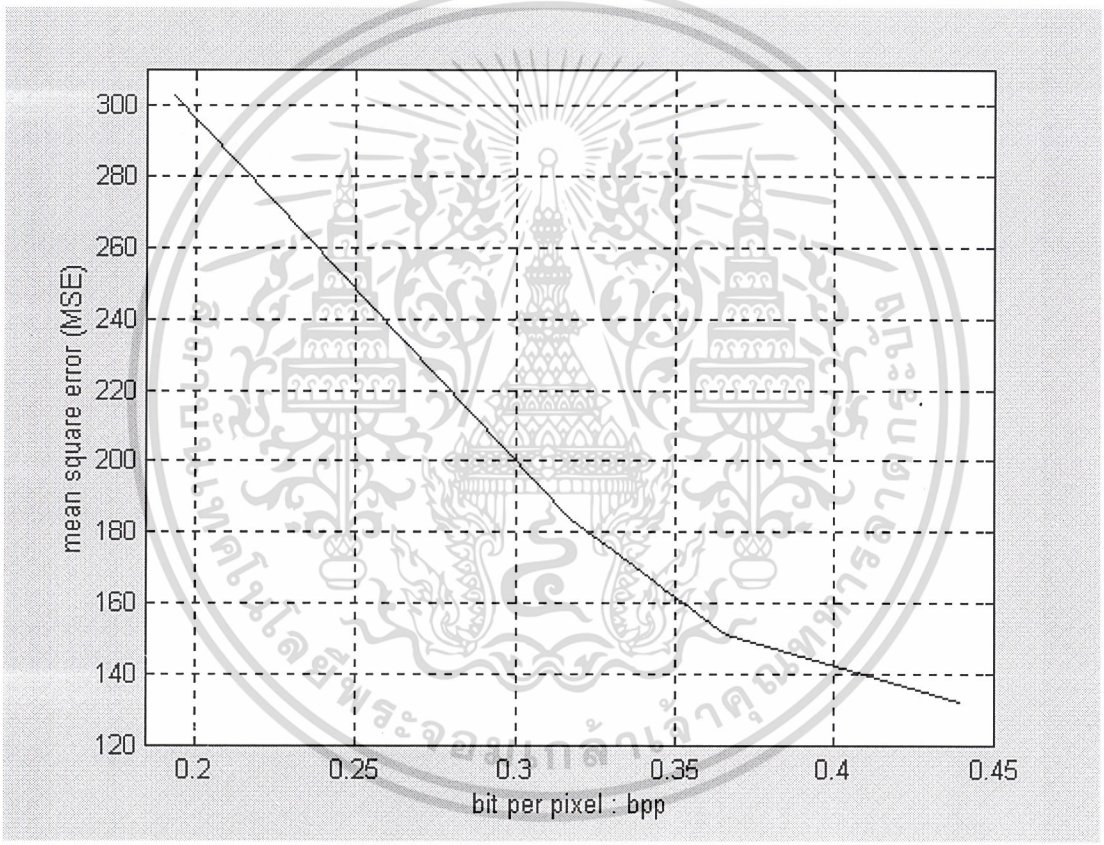
จากการทดลอง เมื่อขนาดของข้อมูลภาพที่ทำการลดขนาดข้อมูลมีอัตราของจำนวนบิตต่อจุดภาพที่ต่ำกว่า 1 บิตต่อจุดภาพ จะเกิดค่าความผิดพลาดของภาพที่ได้จากการลดขนาดข้อมูลมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4.4 นำเอาหลักการของ เอ็มเบดอิดซีโร่ทรี มาใช้ในการเข้ารหัสของสัมประสิทธิ์เวฟเล็ต โดยวิธีการจะนำเอาภาพที่ได้จากการแปลงเวฟเล็ตมาผ่านการจัดระดับด้วยวิธีการประมาณค่าแบบซัคเซสซีฟ โดยจะเปลี่ยนแปลงสัมประสิทธิ์เวฟเล็ตไปเป็นชุดบิตข้อมูล คล้ายกับวิธีการเข้ารหัสแบบบิตเพลน แล้วนำชุดบิตข้อมูลที่แทนสัมประสิทธิ์เวฟเล็ตดังกล่าวไปเข้ารหัสเลขคณิตเพื่อทำการลดขนาดข้อมูล โดยนำภาพ Lena ขนาด 256×256 จุดภาพ เป็นภาพต้นแบบ

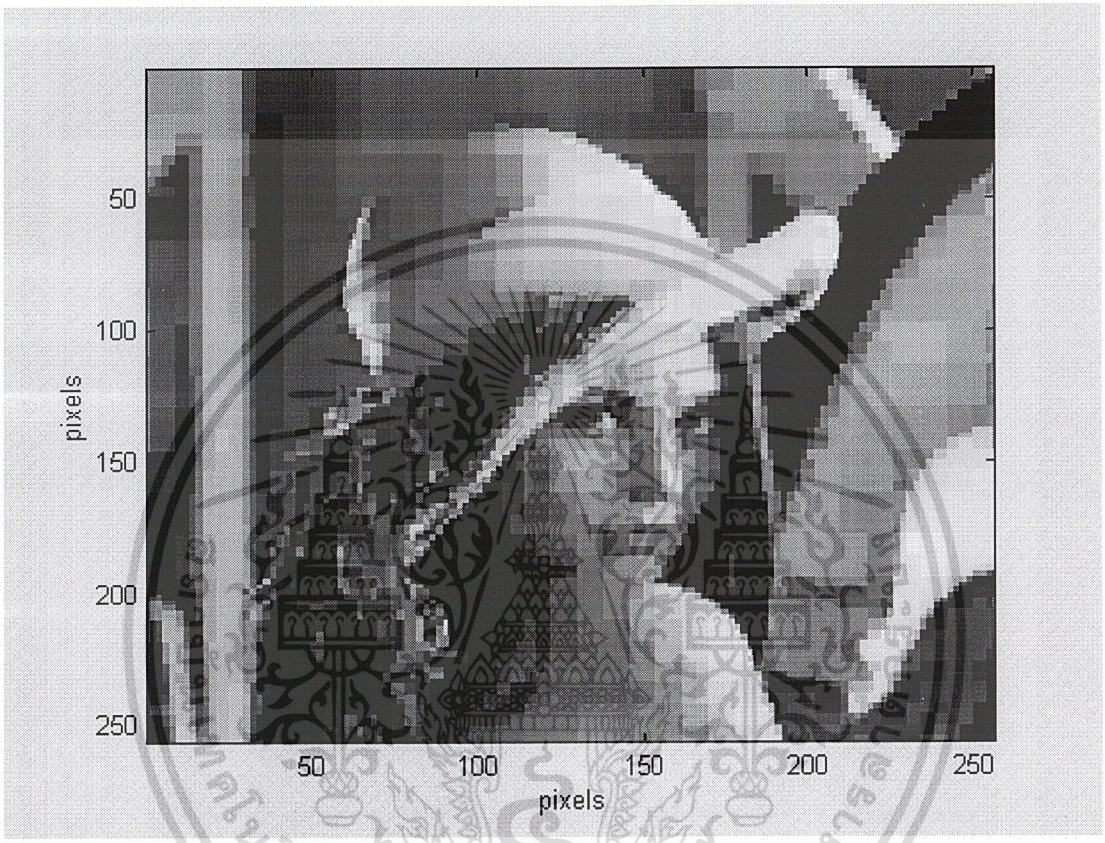
ผลการทดลอง 4.4

จากการนำเอาวิธีการเอ็มเบดอิดซีโร่ทรีมาใช้ในการเข้ารหัสของสัมประสิทธิ์เวฟเล็ตจะได้ผลแสดงเป็นกราฟดังรูป 4.21 และในรูปที่ 4.22 เป็นการแสดงภาพที่ได้จากการแปลงกลับ



รูปที่ 4.21 กราฟแสดงผลของการลดขนาดข้อมูลด้วยวิธีการของเอ็มเบดอิดซีโร่ทรี โดยใช้สัมประสิทธิ์ของฮาร์เป็นเวฟเล็ตฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 เป็นการแสดงภาพ Lena ขนาด 256×256 จุดภาพ ที่ได้จากการแปลงกลับที่มีขนาดของข้อมูลเท่ากับ 0.317 bpp (2596 byte) และมีค่าความผิดพลาด MSE= 183.715

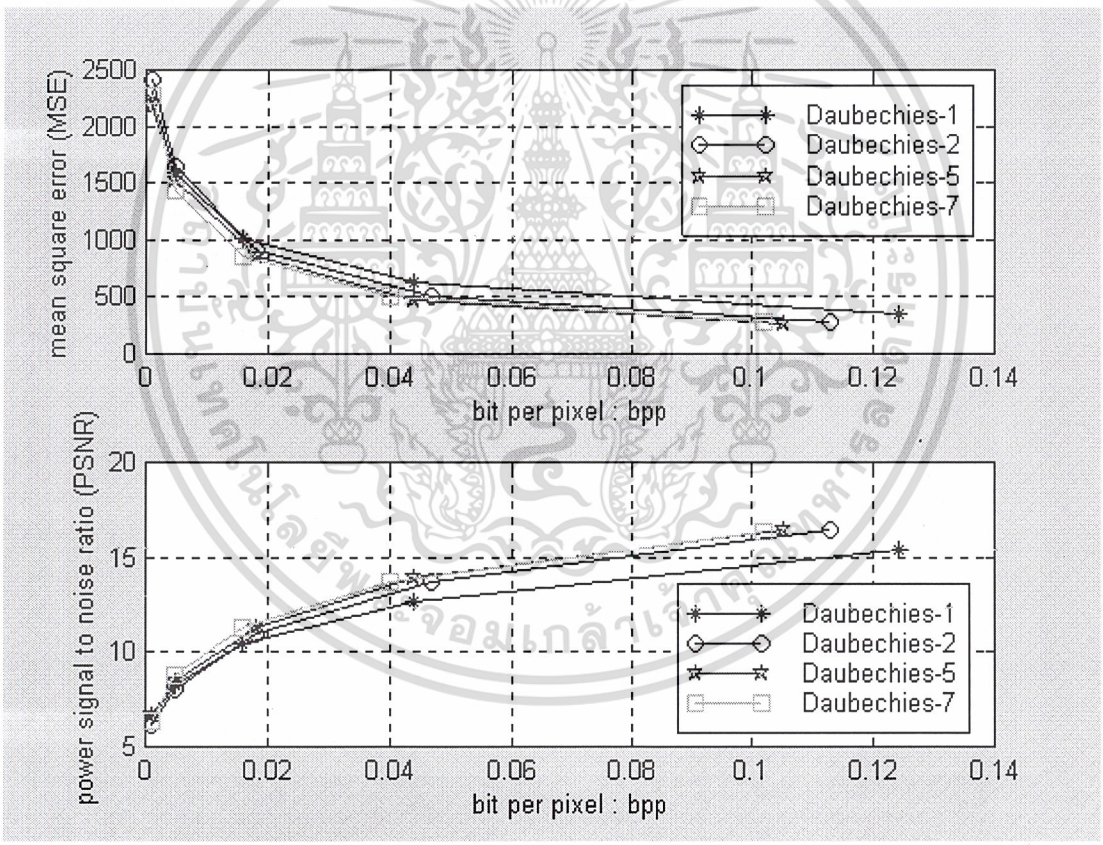
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4.5 จากในทฤษฎีของการแปลงเวฟเล็ต พบว่าเวฟเล็ตฟังก์ชัน ที่ใช้สำหรับการแปลงเวฟเล็ตจะมีหลายฟังก์ชันด้วยกัน การเลือกฟังก์ชันที่เหมาะสมกับวิธีการเข้ารหัสก็จะทำให้คุณภาพของการเข้ารหัสลดข้อมูลมากขึ้น และค่าความผิดพลาดของการจัดสัมประสิทธิ์ของภาพนั้นน้อยลง ดังนั้นการทดลองนี้จะเป็นการนำเอาเวฟเล็ตฟังก์ชันชนิดต่างๆ มาทดลองใช้กับวิธีการเอ็มเบดอิดซีโร่ทรี เพื่อหาเวฟเล็ตฟังก์ชันที่เหมาะสม

การทดลองจะเลือกใช้เวฟเล็ตฟังก์ชันของ ฮาร์ (Haar wavelet) หรือ ดาร์บีชี-1 (Daubechies-1 :db1) ,ดาร์บีชี-2 (db2), ดาร์บีชี-5 (db5),และ ดาร์บีชี –7 (db7) มาทดลองใช้เป็นเวฟเล็ตฟังก์ชันสำหรับการแปลงสัมประสิทธิ์ของภาพในการนำมาลดขนาดข้อมูลด้วยวิธีการของเอ็มเบดอิดซีโร่ทรี

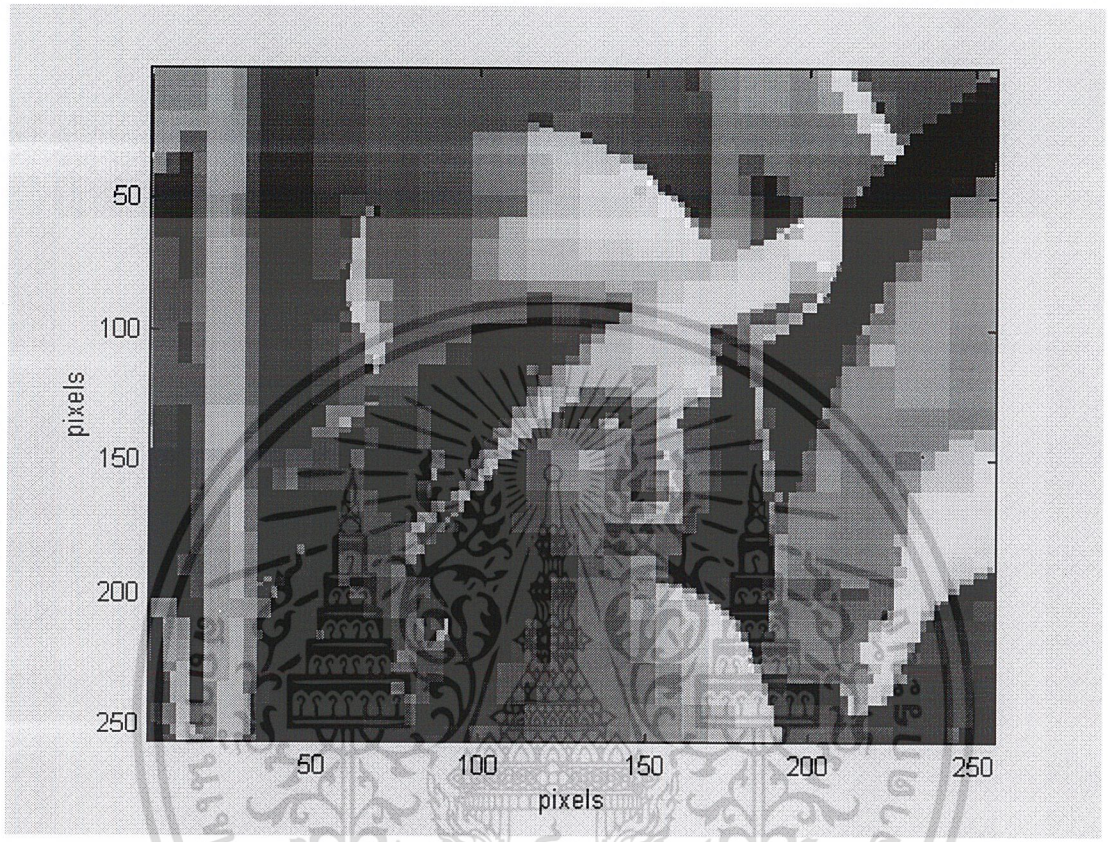
การทดลองนี้นำภาพ Lena ขนาด 256×256 จุดภาพ เป็นภาพต้นแบบ

ผลการทดลอง 4.5



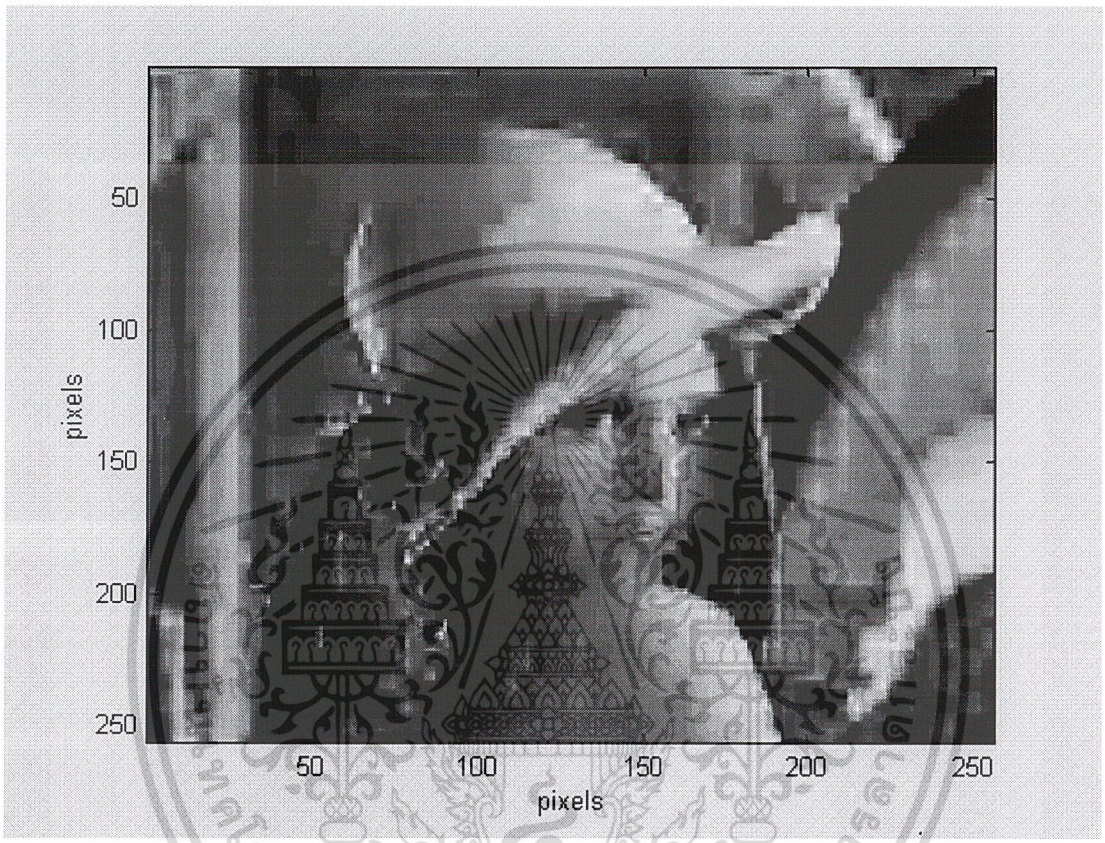
รูปที่ 4.23 กราฟเปรียบเทียบวิธีการเข้ารหัสเอ็มเบดอิดซีโร่ทรี ของเวฟเล็ตฟังก์ชันต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 แสดงผลของภาพที่ผ่านวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี ด้วย db1 ที่มีขนาดของข้อมูลเท่ากับ 0.12 bpp และค่า ค่า MSE = 344.05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

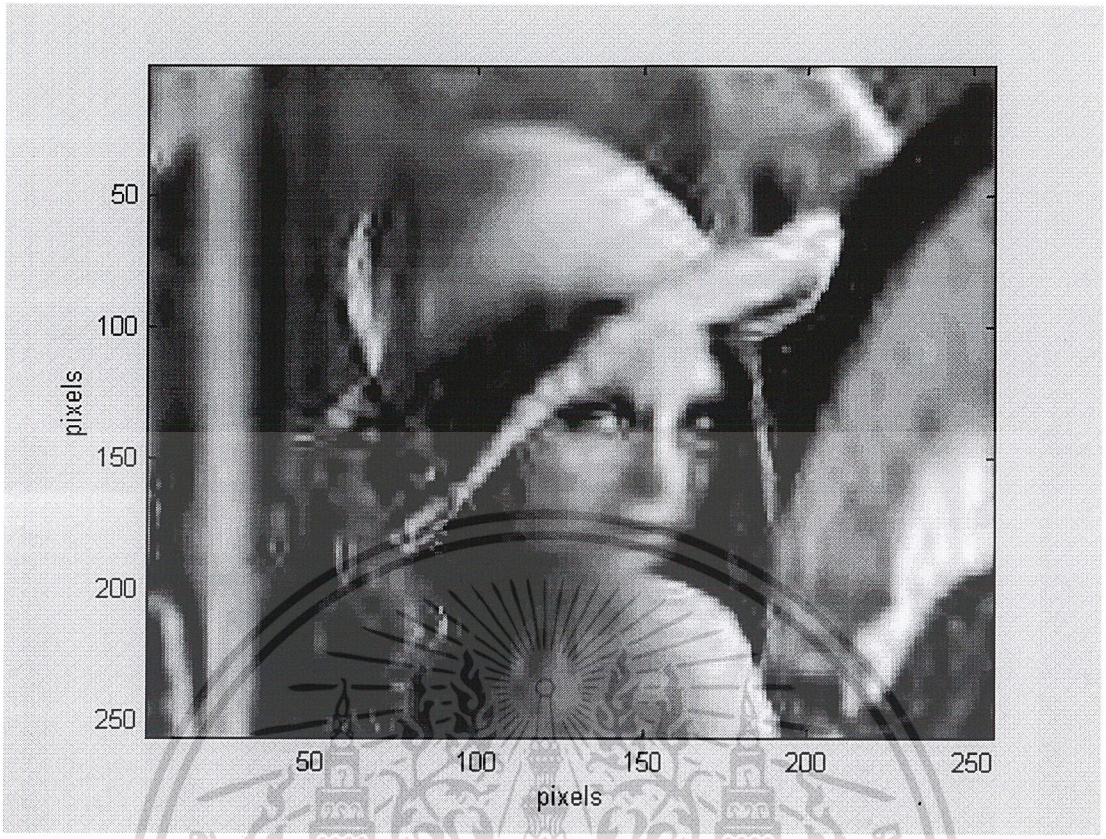


รูปที่ 4.25 แสดงผลของภาพที่ผ่านวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี ด้วย db2 ที่มีขนาดของข้อมูลเท่ากับ 0.12 bpp และค่า MSE = 271.09

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.26 แสดงผลของภาพที่ผ่านวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี ด้วย db5 ที่มีขนาดของข้อมูลเป็น 0.12 bpp และค่า MSE= 270.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 แสดงผลของภาพที่ผ่านวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี ด้วย db7 ที่มีขนาดของข้อมูลเท่ากับ 0.12 bpp และค่า MSE = 280.13

จากกราฟในรูปที่ 4.23 จะเห็นได้ว่าระดับที่จำนวนบิตที่ใช้แทนหนึ่งจุดภาพ (bpp) ต่ำๆ และระดับของสัญญาณต่อสัญญาณรบกวน ของฟังก์ชัน db5 และ db7 มีค่าสูง และระดับของค่าความผิดพลาดก็จะมีค่าน้อย และเมื่อนำจุดที่มีค่าระดับของ bpp ที่เท่าๆ กันมาแสดงภาพ ดังรูปที่ 4.24 ถึง รูปที่ 4.26 ปรากฏว่าในตำแหน่งจุดเดียวกันนี้ภาพที่ได้จากฟังก์ชัน db5 มีค่าต่ำที่สุด จากกราฟในรูปที่ 4.23 พบว่า เมื่อจำนวนบิตที่ใช้แสดงจุดภาพยิ่งมีอัตราที่ต่ำลงฟังก์ชัน db7 จะมีค่าความผิดพลาดที่ต่ำกว่า แต่ค่าความผิดพลาดในช่วง db7 ที่ต่ำกว่าฟังก์ชันอื่นๆ จะอยู่ในช่วงที่สูง

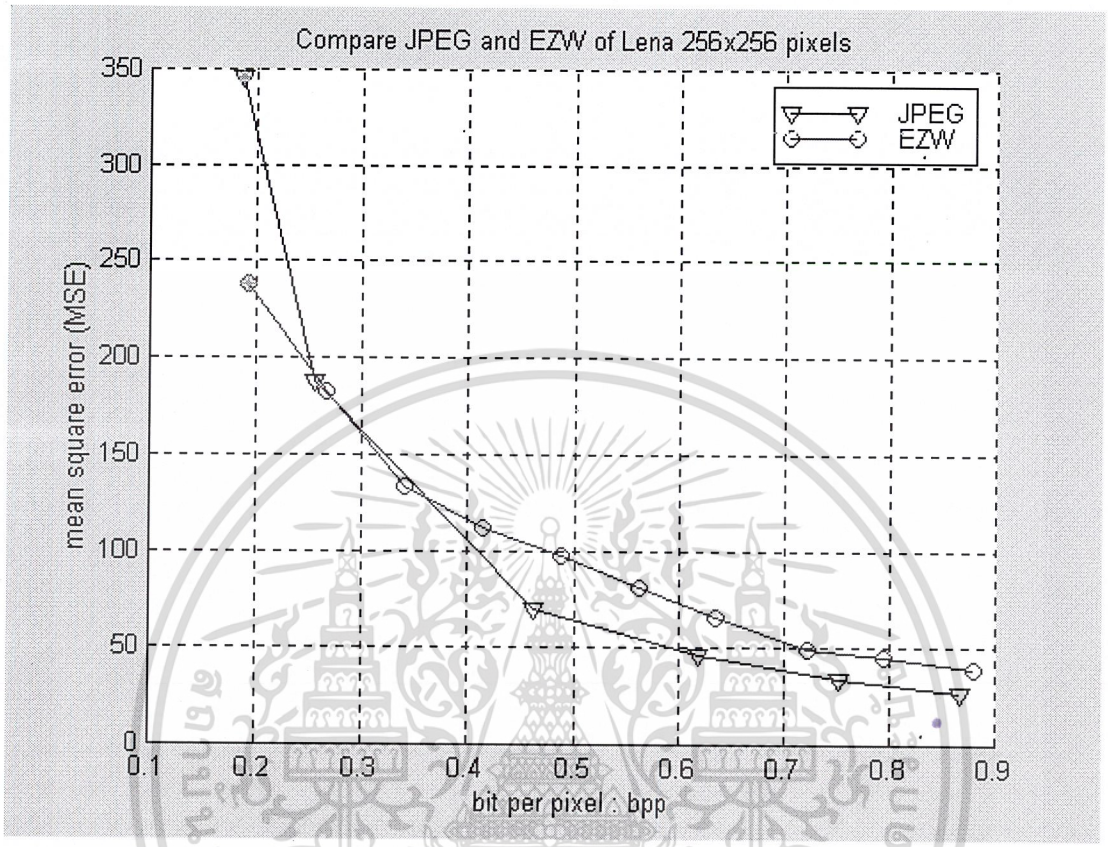
ดังนั้น ในการใช้วิธีของเอ็มเบดอิดซีโร่ทรีลดสัมประสิทธิ์ของเวฟเล็ต เราจะใช้ฟังก์ชัน db5 เป็นฟังก์ชันเวฟเล็ต สำหรับการแปลงโมเดลของภาพ

การทดลองที่ 4.6 ทดลองนำภาพต่างๆ มาทดลองวิธีการลดขนาดข้อมูลภาพด้วยวิธีการเอ็มเบดอิดซีโรทรี (EZW) โดยนำผลของการลดขนาดข้อมูลมาสร้างเป็นกราฟ และนำมาเทียบผลกับการลดขนาดข้อมูลภาพด้วยวิธีการเจเพก(JPEG) เพื่อดูค่าความผิดพลาดที่เกิดขึ้นเมื่อทำการลดขนานข้อมูลที่ต่ำกว่า 1 บิตต่อจุดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.1) นำภาพ Lena ขนาด 256×256 จุดภาพ เป็นภาพต้นแบบ

ผลการทดลอง 4.6.1



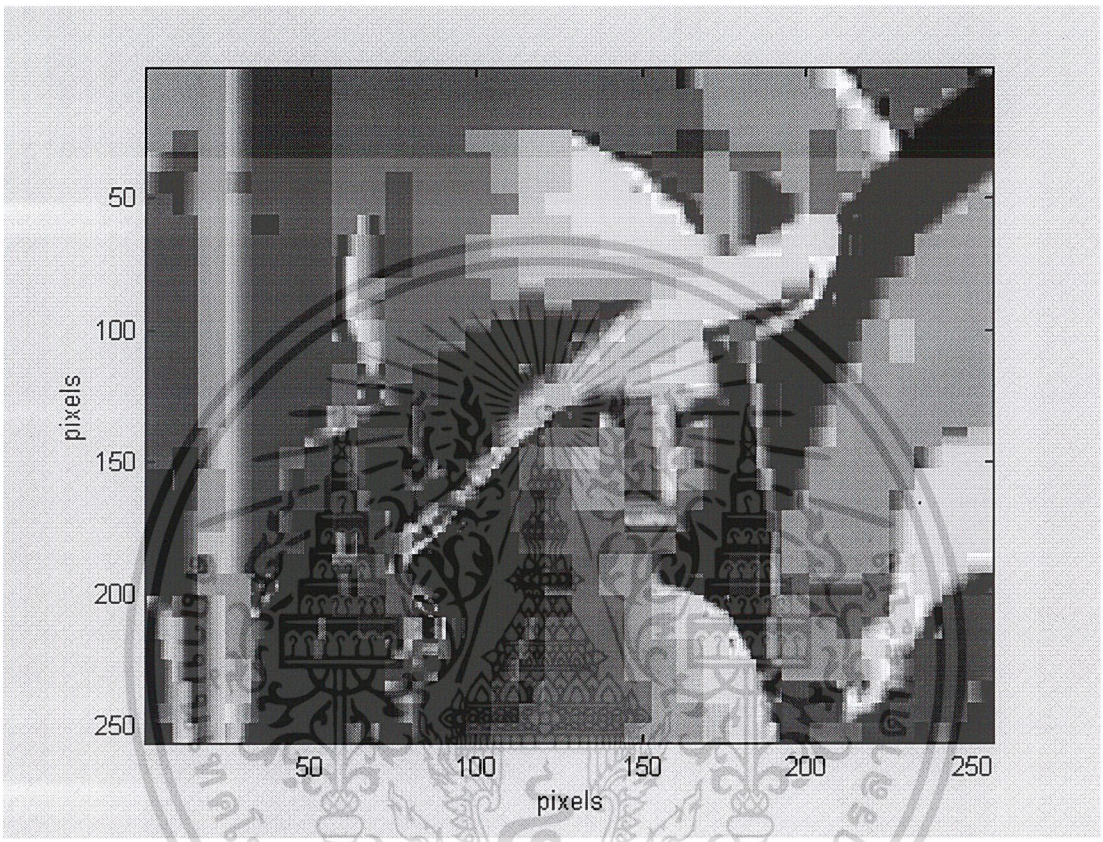
รูปที่ 4.28 กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการของ EZW และ JPEG (ใช้ภาพ Lena ขนาด 256×256 จุดภาพ)

จากกราฟในรูปที่ 4.28 จะเห็นได้ว่าวิธีการของ EZW ที่ระดับขนาด bpp ต่ำกว่าช่วง 0.35 bpp จะมีระดับของค่าความผิดพลาดที่ต่ำกว่า JPEG แต่ในช่วงที่มีค่า bpp ที่สูงกว่านั้น JPEG จะมีค่าของความผิดพลาดที่ต่ำกว่า ทำการเลือกภาพในช่วงที่ขนาด bpp ประมาณ 0.19 มาวิเคราะห์ดูภาพของวิธี EZW และ JPEG จะสามารถแสดงได้ดังรูปที่ 4.29 และ รูปที่ 4.30 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.29 ภาพที่ได้จากการลดขนาดข้อมูลด้วยวิธี EZW
ที่ขนาดภาพเท่ากับ 0.19 บิตต่อจุดภาพ และค่าความผิดพลาด (MSE) = 238.59
(ใช้ภาพ Lena ขนาด 256×256 จุดภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

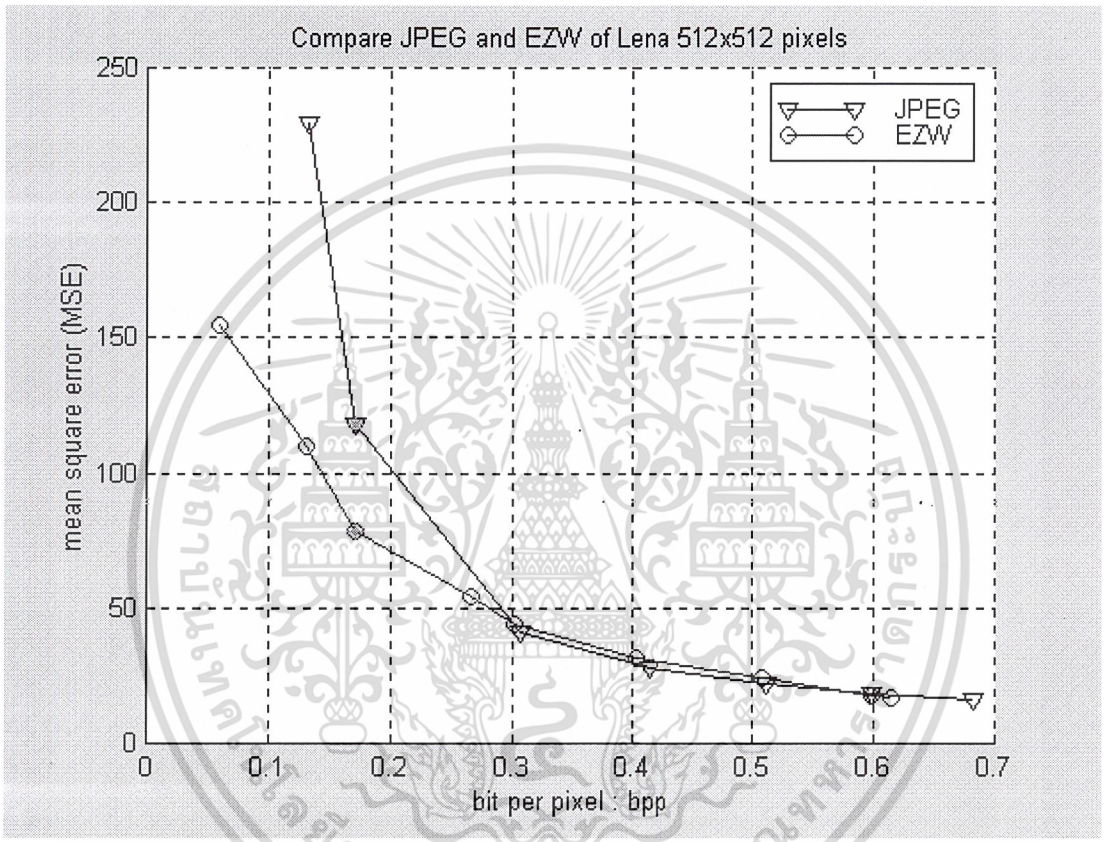


รูปที่ 4.30 ภาพที่ได้จากการลดขนาดข้อมูลด้วยวิธี JPEG
ที่ขนาดภาพเท่ากับ 0.18 บิตต่อจุดภาพ และค่าความผิดพลาด (MSE) = 346.11
(ใช้ภาพ Lena ขนาด 256×256 จุดภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.2) นำภาพ Lena ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบ

ผลการทดลอง 4.6.2

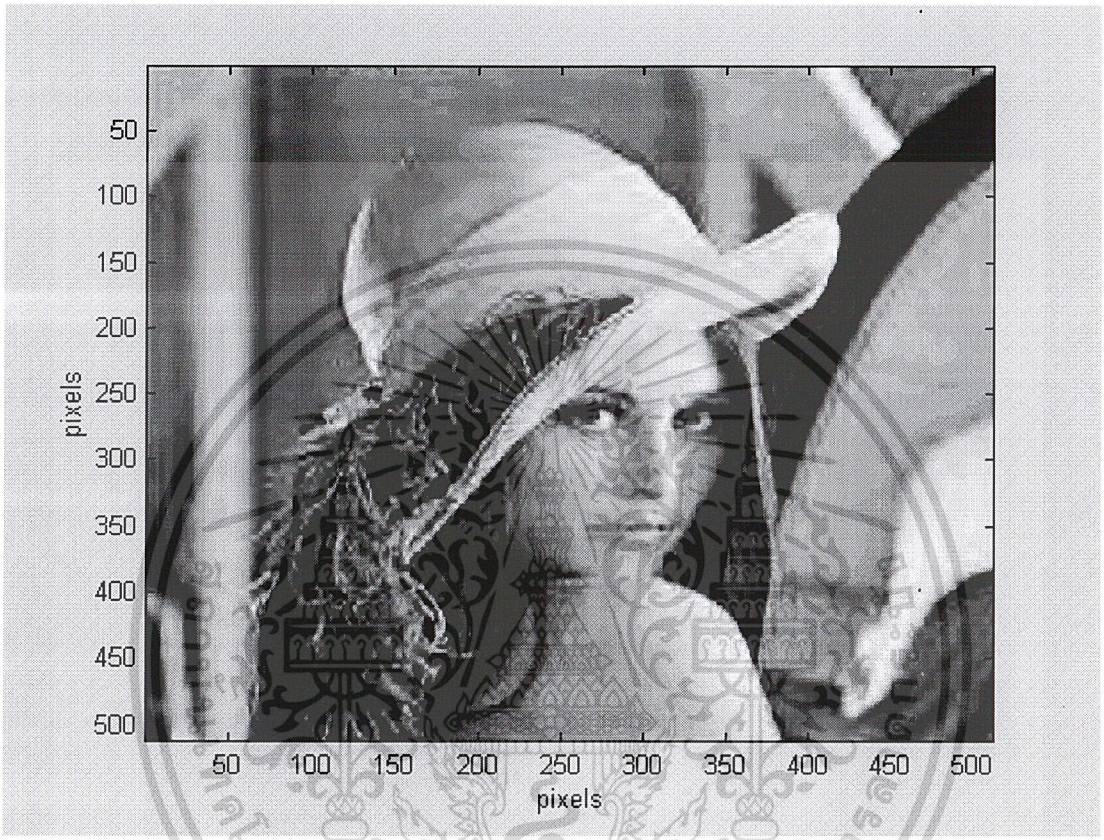


รูปที่ 4.31 กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการของ EZW และ JPEG (ใช้ภาพ Lena ขนาด 512×512 จุดภาพ)

จากรูปที่ 4.31 เห็นว่าในขนาดของภาพที่ต่ำกว่า 0.3 bpp วิธีการของ EZW จะมีค่าความผิดพลาดของการลดขนาดข้อมูลที่น้อยกว่าการลดขนาดข้อมูลภาพด้วยวิธีของเจเพก จากรูปที่ 4.28 ซึ่งได้ทดสอบรูปภาพชนิดเดียวกันแต่ขนาดแตกต่างกันพบว่าผลของการลดขนาดข้อมูลนั้นแตกต่างกันอย่างเห็นได้ ชัดเจน เนื่องด้วยในระดับภาพที่มีรายละเอียดที่สูงขึ้น จะมีองค์ประกอบของสัญญาณในภาพที่ละเอียดขึ้นตามไปด้วย ดังนั้นการวิเคราะห์ทางด้านความถี่ที่อาศัยการแยกแบนด์ของความถี่เป็นความถี่ย่อยๆ โดยใช้การแปลงเวฟเล็ตก็จะสามารถแบ่งแบนด์ของความถี่ได้ละเอียดมากขึ้นตามไปด้วย อาทิเช่น ภาพขนาด 256×256 จุดภาพ เมื่อทำการแปลงเวฟเล็ต จะถูกแบ่งแบนด์ของความถี่ออกเป็น 8 แบนด์ และในกรณีขนาดภาพเป็น 512×512 จุดภาพจะสามารถแบ่งแบนด์ของความถี่ได้เป็น 9 แบนด์ ซึ่งจะมีความ

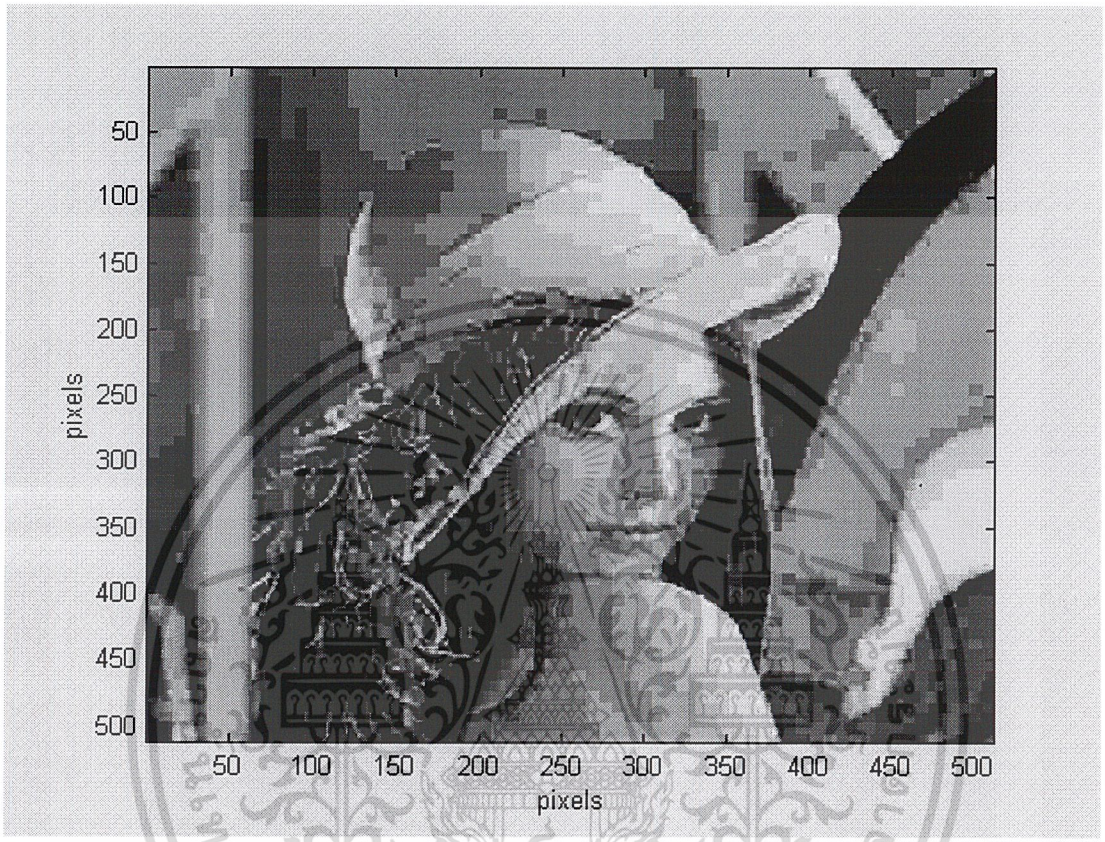
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ละเอียดในการแบ่งความถี่เพิ่มขึ้น ยิ่งในระดับภาพที่ละเอียดขึ้นไปลักษณะของการใช้การแปลงเวฟเล็ตก็จะยิ่งมี ประสิทธิภาพมากยิ่งขึ้น ในรูปที่ 4.35 และรูปที่ 4.36 ทำการแสดงภาพที่มีขนาดเท่ากับ 0.171 บิตต่อจุดภาพ ของวิธีการลดขนาดข้อมูลภาพแบบ EZW และ JPEG ตามลำดับ



รูปที่ 4.32 แสดงภาพที่ผ่านการลดขนาดข้อมูลด้วยวิธีการ EZW
ที่มีขนาดของข้อมูลเป็น 0.171 บิตต่อจุดภาพ (bpp) และค่าความผิดพลาด (MSE) เท่ากับ 78.71
(ใช้ภาพ Lena ขนาด 512×512 จุดภาพ)

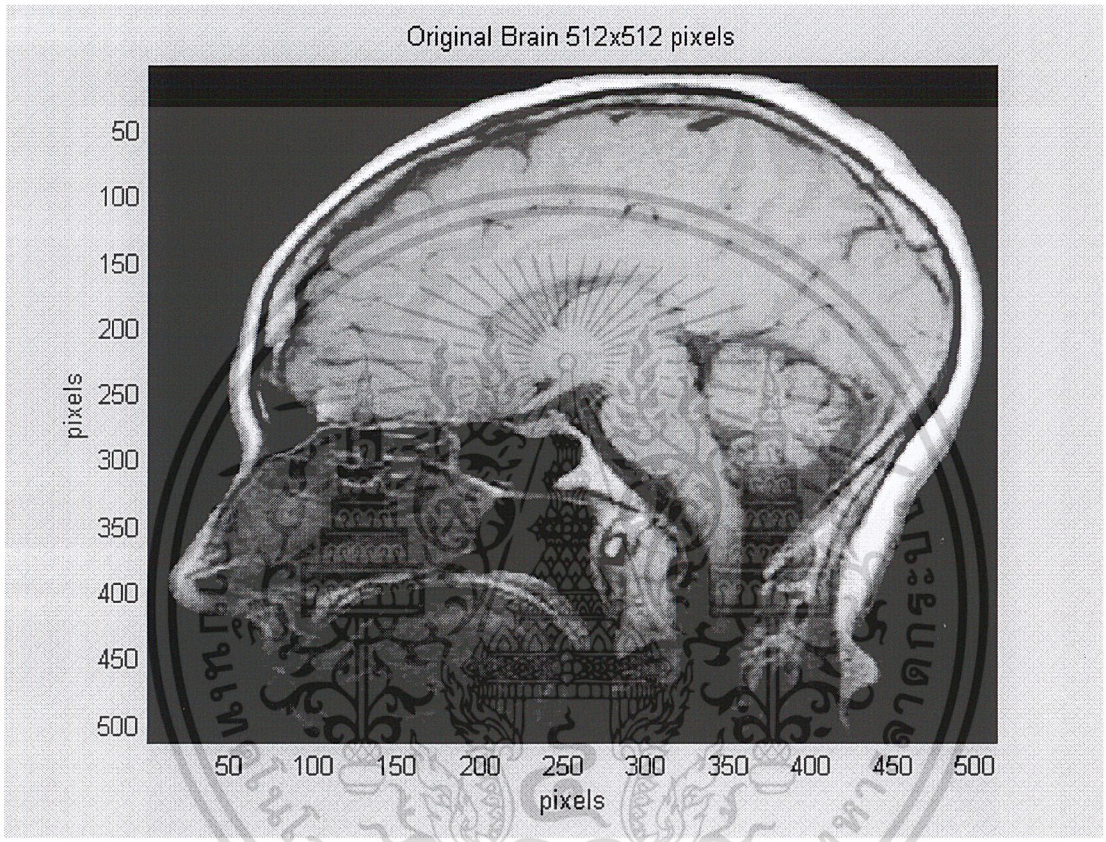
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 แสดงภาพที่ผ่านการลดขนาดข้อมูลด้วยวิธีการ JPEG ที่มีขนาดของข้อมูลเป็น 0.171 บิตต่อจุดภาพ (bpp) และค่าความผิดพลาด (MSE) เท่ากับ 117.72 (ใช้ภาพ Lena ขนาด 512×512 จุดภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

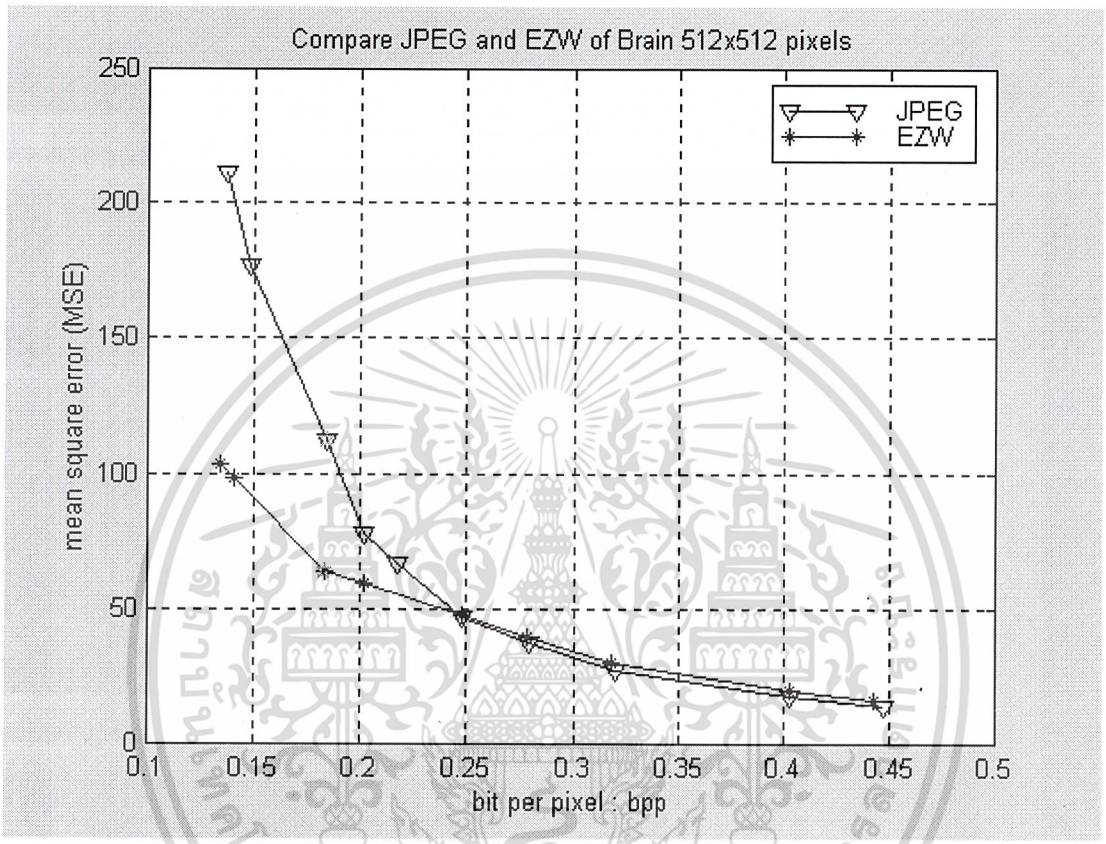
4.6.3) นำภาพ Brain ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบ



รูปที่ 4.34 แสดงภาพ Brain ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบที่นำมาลดขนาดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

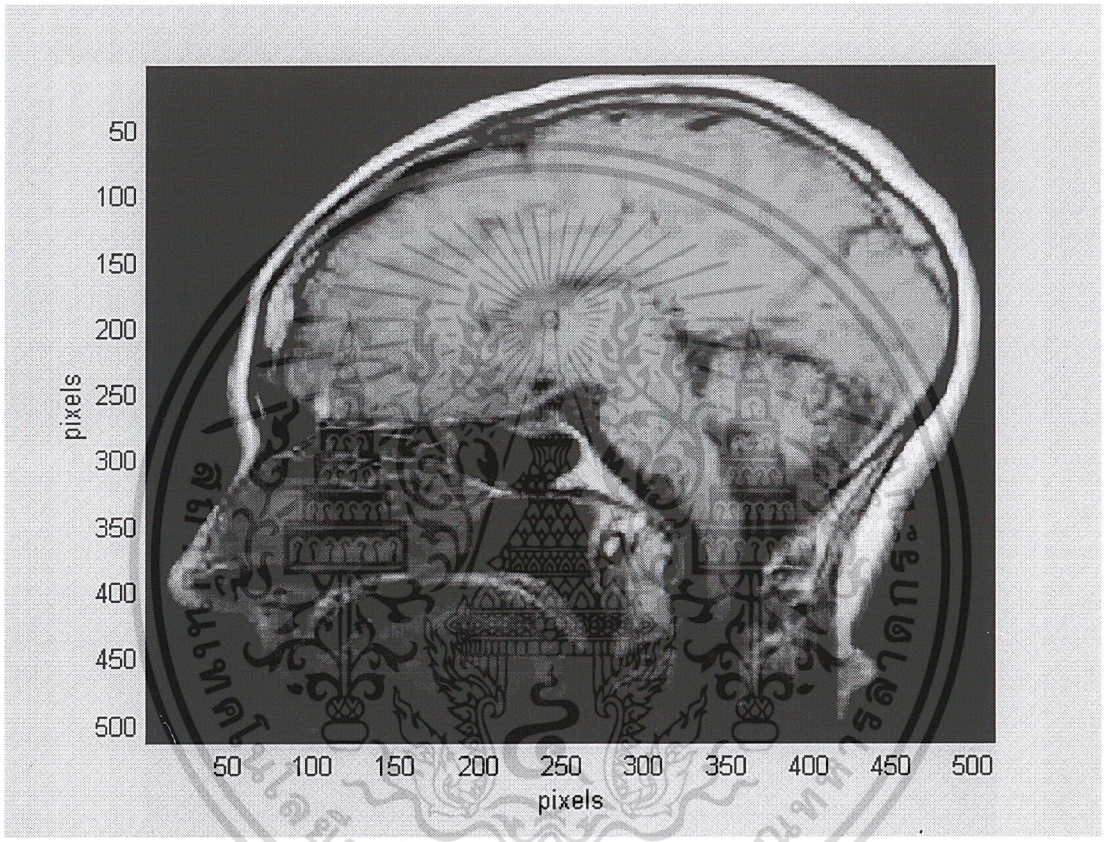
ผลการทดลอง 4.6.3



รูปที่ 4.35 กราฟแสดงผลของการลดขนาดข้อมูลภาพระหว่างวิธีการ EZW และ JPEG (ภาพ Brain ขนาด 512×512 จุดภาพ)

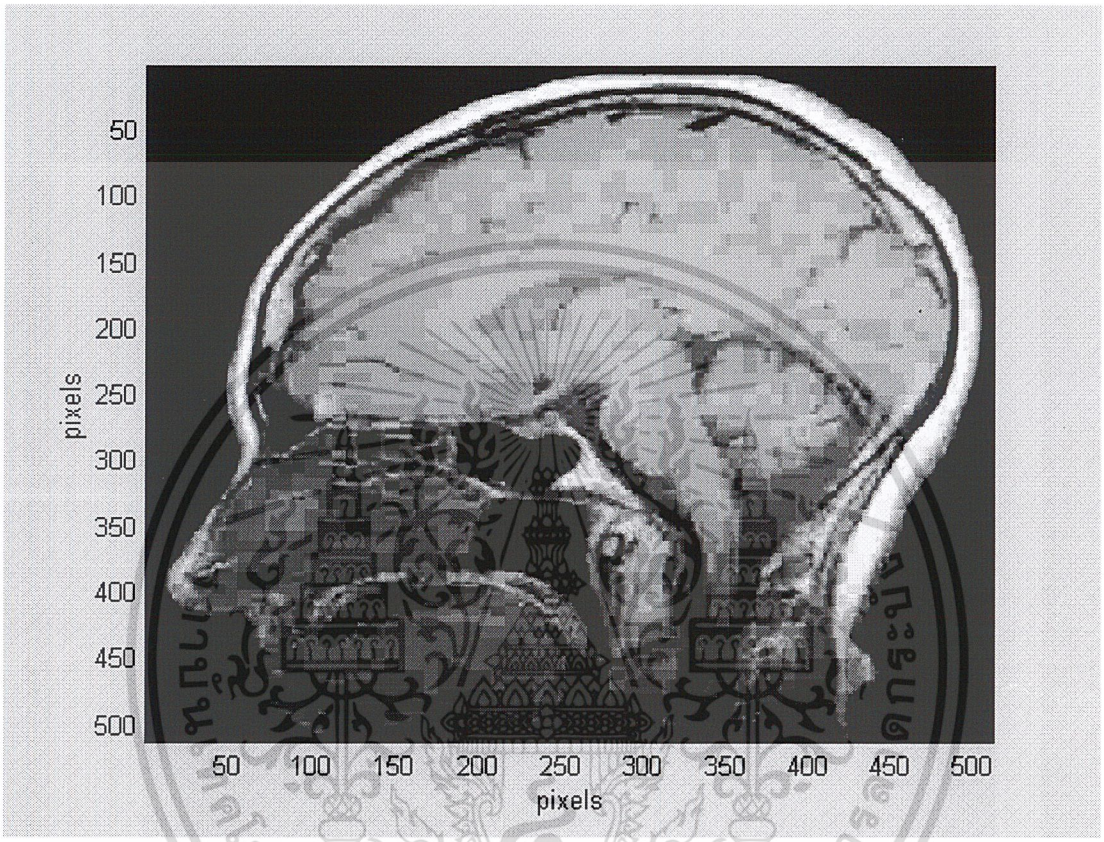
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงภาพที่มีอัตราของจำนวนบิตต่อจุดภาพที่ประมาณเท่ากับ 0.184 บิตต่อจุดภาพ ของวิธีลดขนาดข้อมูลด้วยวิธี EZW และ JPEG ได้ดังรูปที่ 4.36 และ 4.37 ตามลำดับ



รูปที่ 4.36 แสดงภาพที่มีขนาด 0.1835 bpp ที่ทำการลดขนาดข้อมูลภาพโดยวิธีการของ EZW (ภาพ Brain ขนาด 512×512 จุดภาพ)

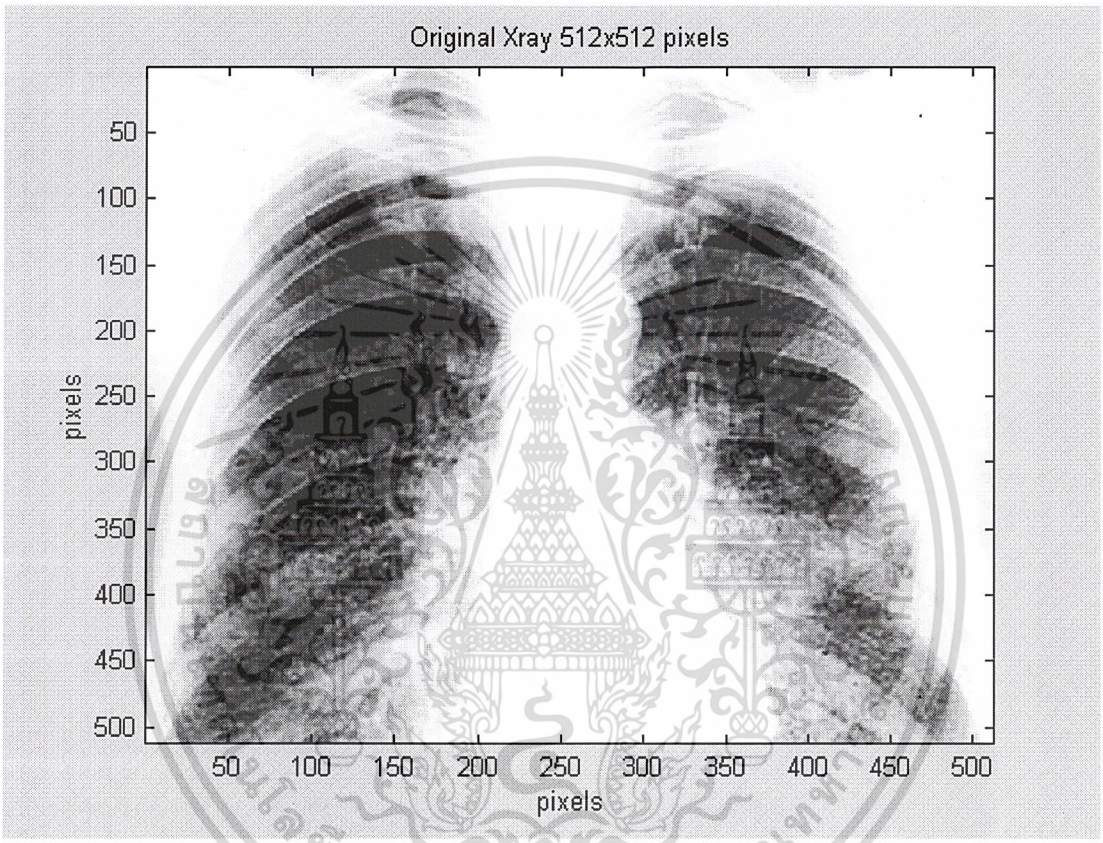
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.37 แสดงภาพที่มีขนาด 0.184 bpp ที่ทำการลดขนาดข้อมูลภาพโดยวิธีการของ JPEG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

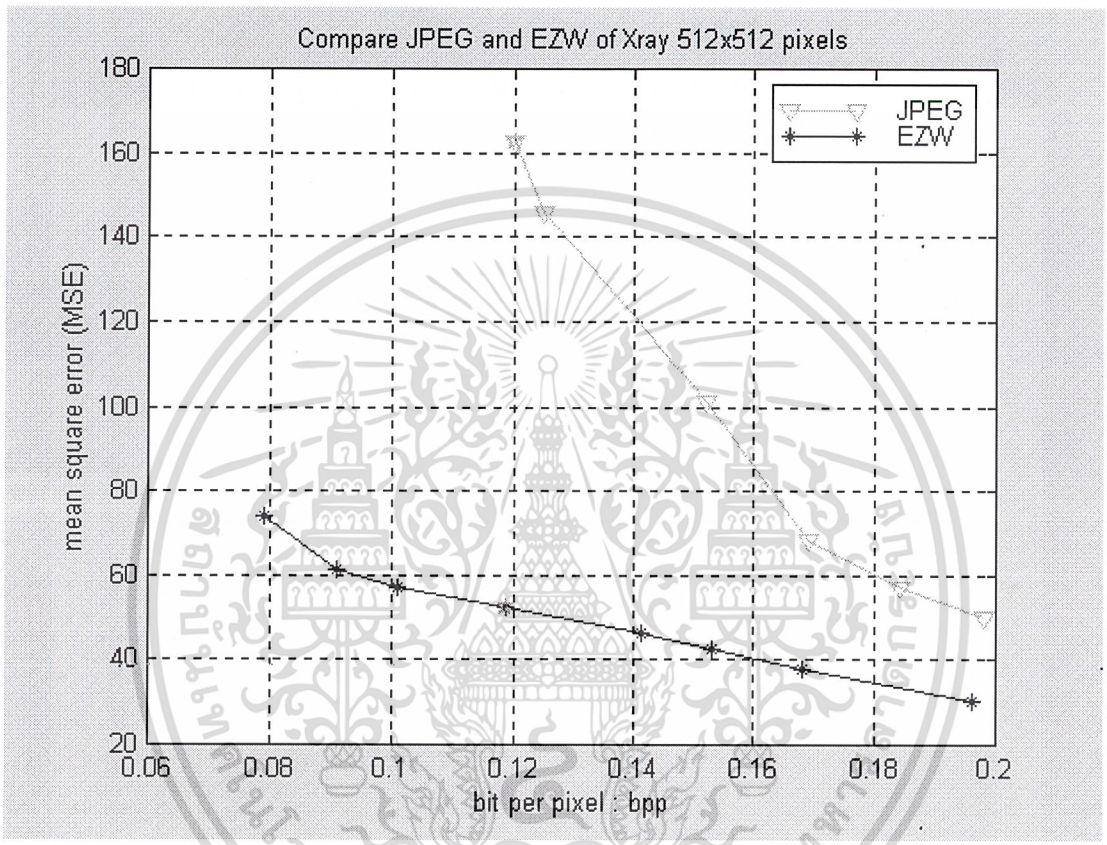
4.6.4) นำภาพ Xray ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบ



รูปที่ 4.38 แสดงภาพ Xray ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบที่นำมาลดขนาดข้อมูล

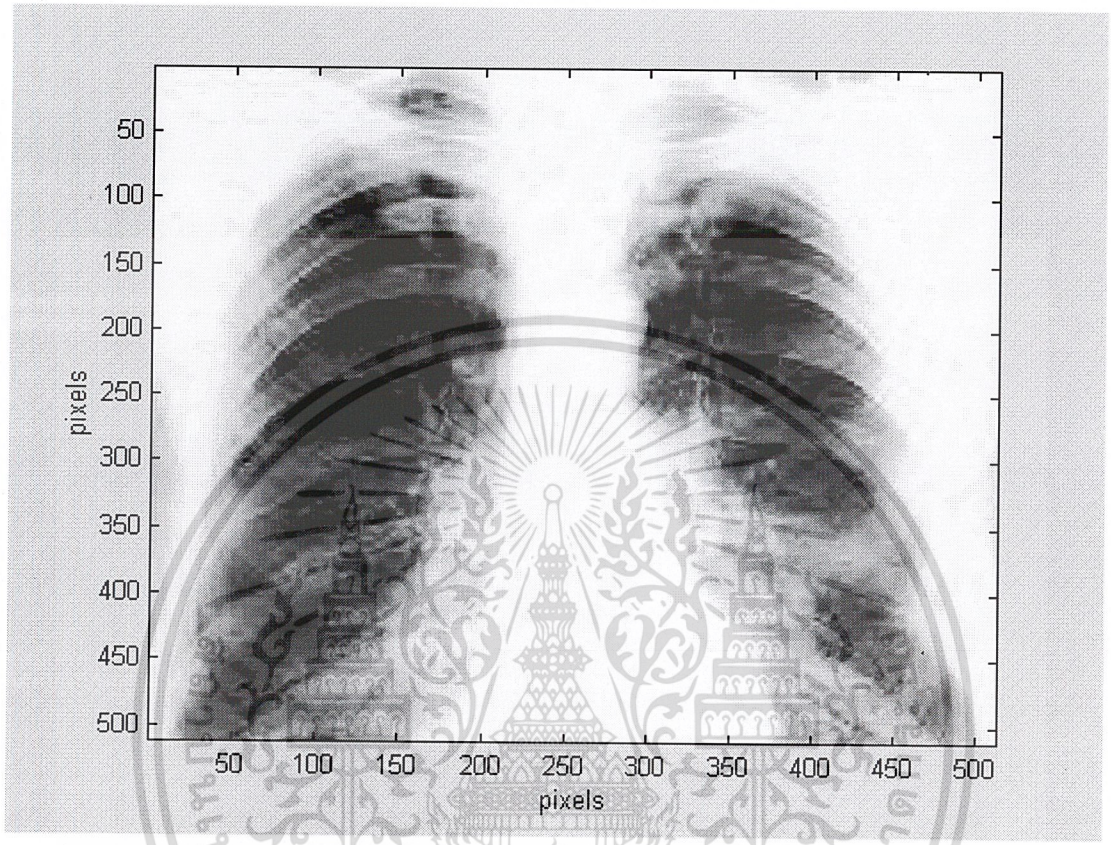
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง 4.6.4



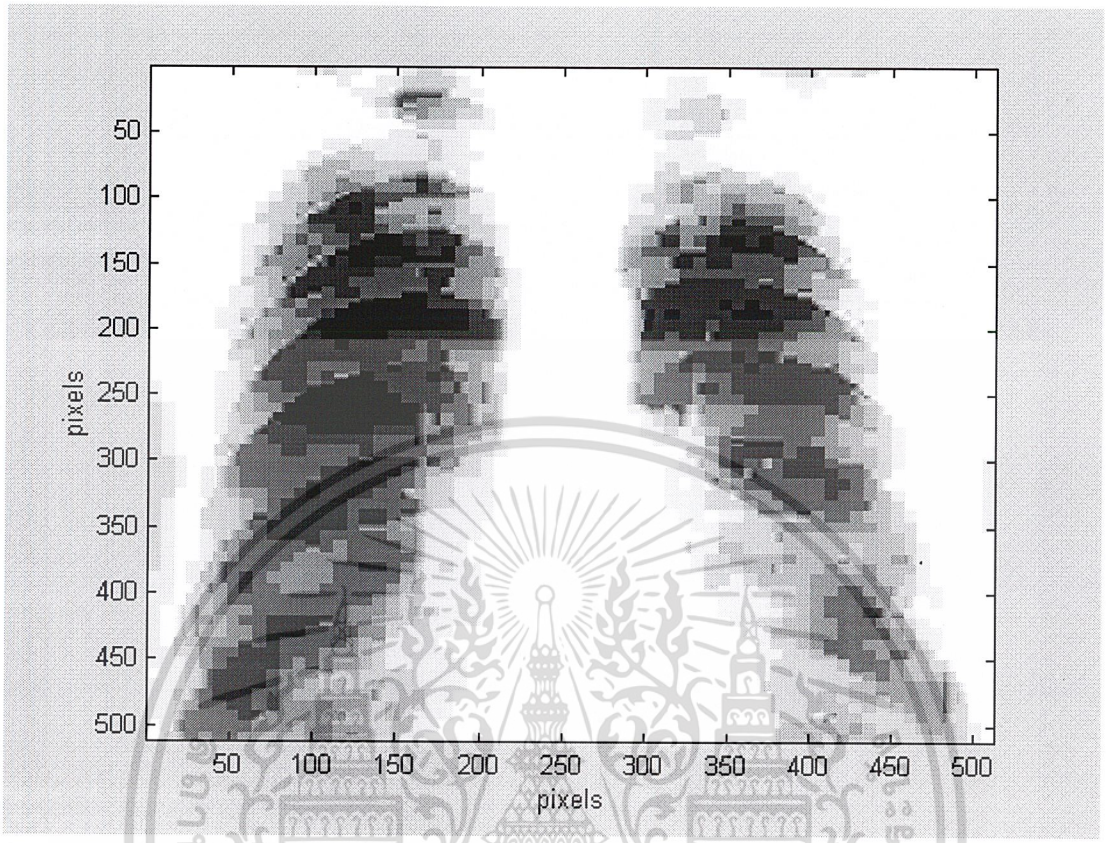
รูปที่ 4.39 กราฟแสดงผลการลดขนาดข้อมูลภาพด้วยวิธี EZW และ JPEG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.40 แสดงภาพที่มีขนาด 0.119 bpp ที่ทำการลดขนาดข้อมูลโดยวิธีการ EZW มีค่าความผิดพลาด (MES) = 52.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.41 แสดงภาพที่มีขนาด 0.120 bpp ที่ทำการลดขนาดข้อมูลโดยวิธีการ JPEG มีค่าความผิดพลาด (MSE) = 162.49

จากผลการทดลองในการทดลองที่ 4.6 เห็นได้ว่าการลดขนาดข้อมูลโดยการใช้การแปลงเวฟเล็ตแบบไม่ต่อเนื่องโดยการนำมาเข้ารหัส EZW สามารถที่จะลดข้อมูลภาพได้ต่ำกว่า 1 บิตต่อจุดภาพ โดยที่มีค่าความผิดพลาดของข้อมูลภาพต่ำกว่าวิธีการลดขนาดข้อมูลภาพด้วยวิธีการ JPEG โดยภาพที่ลดขนาดโดยวิธี EZW สามารถที่จะดูรายละเอียดภายในภาพได้ แต่ในขณะที่วิธีการของ JPEG นั้นภาพจะแตกออกเป็นบล็อกๆ แต่จะให้ขอบของภาพที่คมชัดกว่าวิธีการ EZW

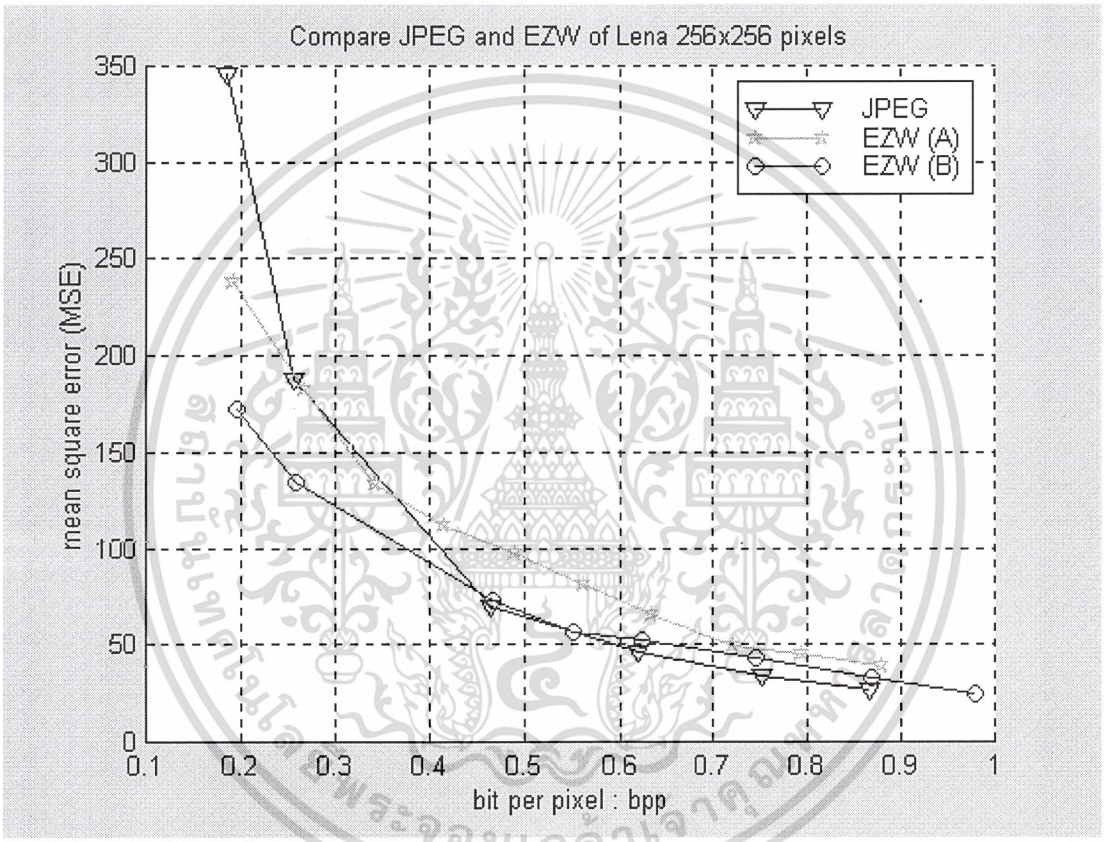
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4.7 ทดลองวิธีการเอ็มเบดอิดซีโร่ทรีที่ปรับปรุงการเข้ารหัสสัญลักษณ์ขึ้นมาใหม่ (กล่าวไว้ในบทที่ 3)

4.7.1) นำภาพ Lena ขนาด 256×256 จุดภาพ เป็นภาพต้นแบบ

ผลการทดลอง 4.7.1

ได้กราฟแสดงผลของการลดขนาดข้อมูลเปรียบเทียบค่าความผิดพลาดของการลดขนาดข้อมูลกับอัลกอริทึมเก่า และวิธีการเจเพก แสดงดังรูปที่ 4.42 โดยอัลกอริทึมของ EZW เดิมเป็น EZW(A) และวิธีการที่ปรับปรุงคือ EZW(B)



รูปที่ 4.42 กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการ EZW(A) ,EZW(B) และ JPEG

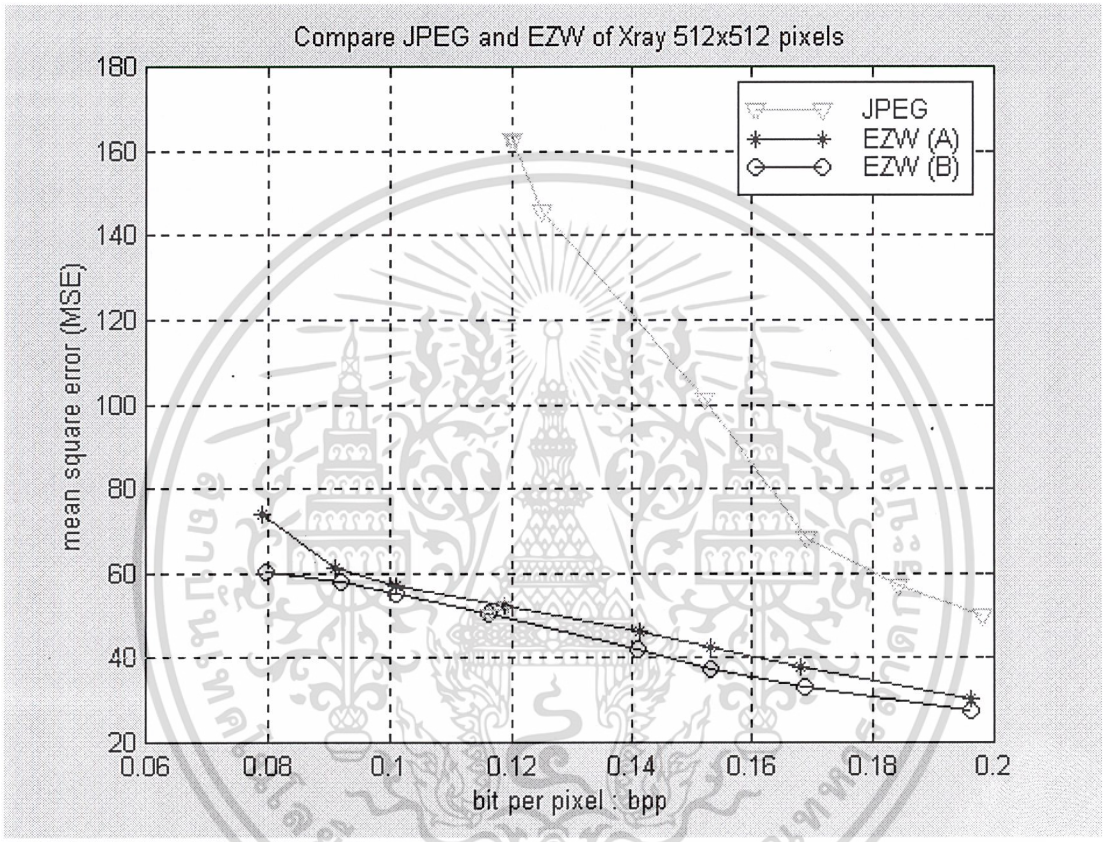
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.43 ภาพที่ได้จากการลดขนาดข้อมูลภาพด้วยวิธี EZW ที่ปรับปรุงวิธีการเข้ารหัสใหม่ มีขนาดเท่ากับ 0.19 bpp และค่าความผิดพลาด (MSE) = 172.15 (ใช้ภาพ Lena ขนาด 256×256 จุดภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

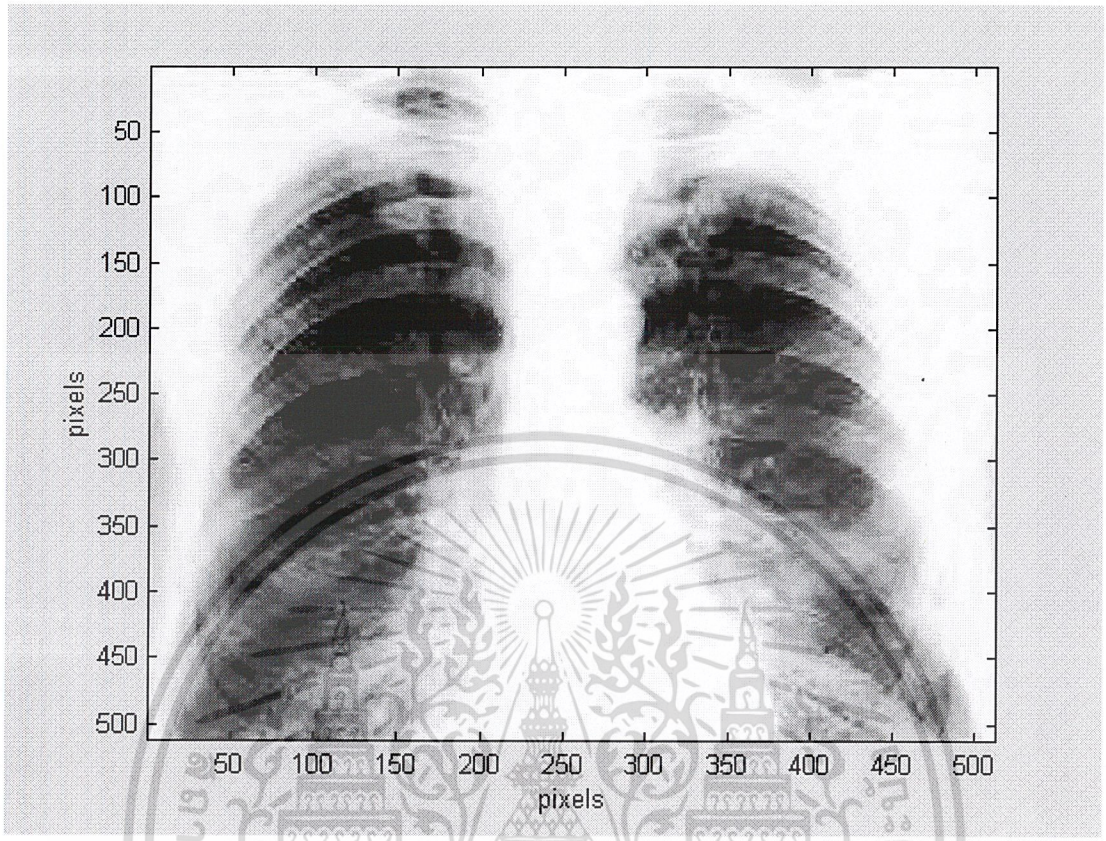
4.7.2) นำภาพ Xray ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบ

ผลการทดลอง 4.7.2



รูปที่ 4.44 กราฟแสดงผลของการลดขนาดข้อมูลภาพด้วยวิธีการ EZW(A) ,EZW(B) และ JPEG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.45 ภาพที่ได้จากการลดขนาดข้อมูลภาพด้วยวิธี EZW ที่ปรับปรุงวิธีการเข้ารหัสใหม่ มีขนาดเท่ากับ 0.116 bpp และค่าความผิดพลาด (MSE) = 50.57

จากการนำภาพ Lena ขนาด 256×256 จุดภาพ เป็นภาพต้นแบบมาทดลองการลดขนาดข้อมูลภาพเมื่อเปรียบเทียบกับผลในการทดลองที่ 4.6.1 ในรูปที่ 4.28 พบว่าวิธีการปรับปรุงการเข้ารหัสใหม่สามารถที่จะให้ค่าความผิดพลาดของภาพต่ำลง ในอัตราของจำนวนบิตต่อจุดภาพที่เท่าเดิม โดยภาพในรูปที่ 4.45 เป็นภาพที่มีขนาดของจำนวนบิตต่อจุดภาพที่เท่ากับภาพในรูปที่ 4.29 และ รูปที่ 4.30

และเมื่อนนำภาพ Xray ขนาด 512×512 จุดภาพ เป็นภาพต้นแบบ จะได้ผลในรูปที่ 4.44 และภาพในรูปที่ 4.45 ทำการแสดงผลของภาพที่มีอัตราจำนวนบิตต่อจุดภาพที่เท่ากับภาพในรูปที่ 4.40 และรูปที่ 4.41

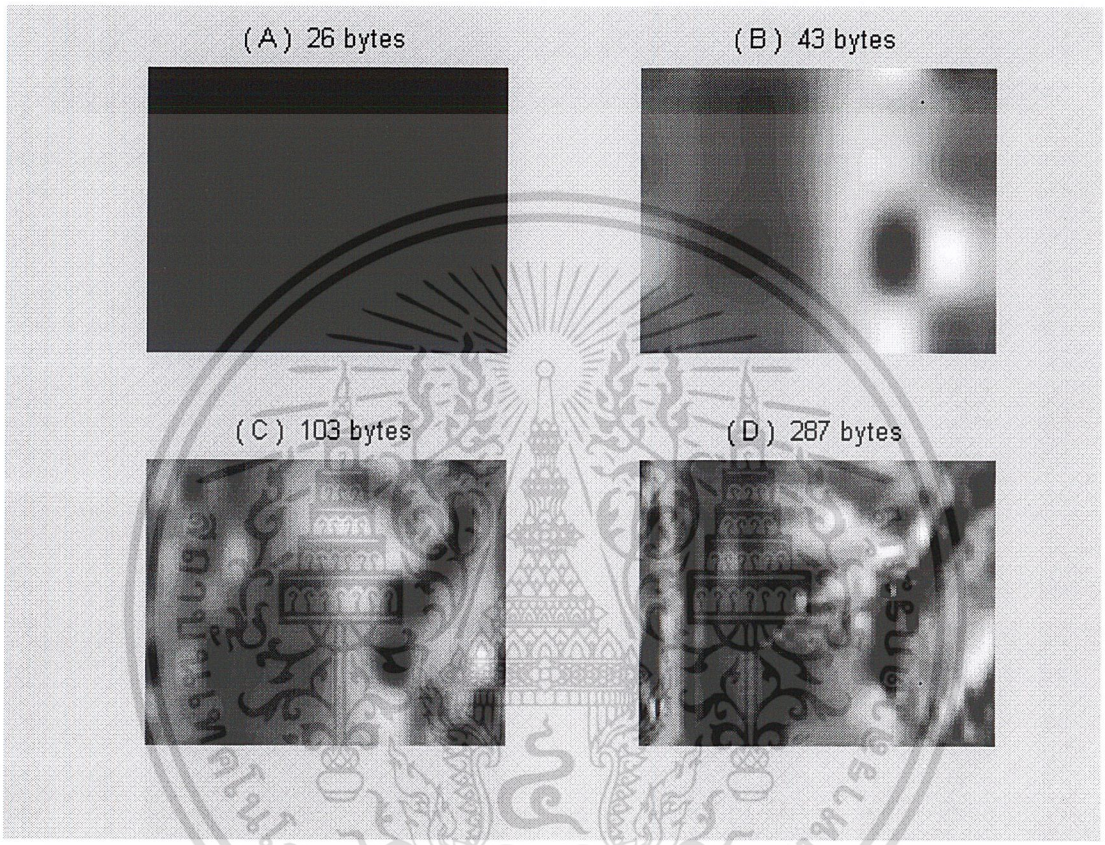
ดังนั้น การปรับปรุงวิธีการเข้ารหัสใหม่นี้จะให้ผลของการลดขนาดข้อมูลที่เป็นที่น่าพอใจกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4.8 ทดลองนำวิธีการเอ็มเบดอิดซีโร่ทรีมาทำการแสดงผลข้อมูลแบบโปรเกรสซีฟ

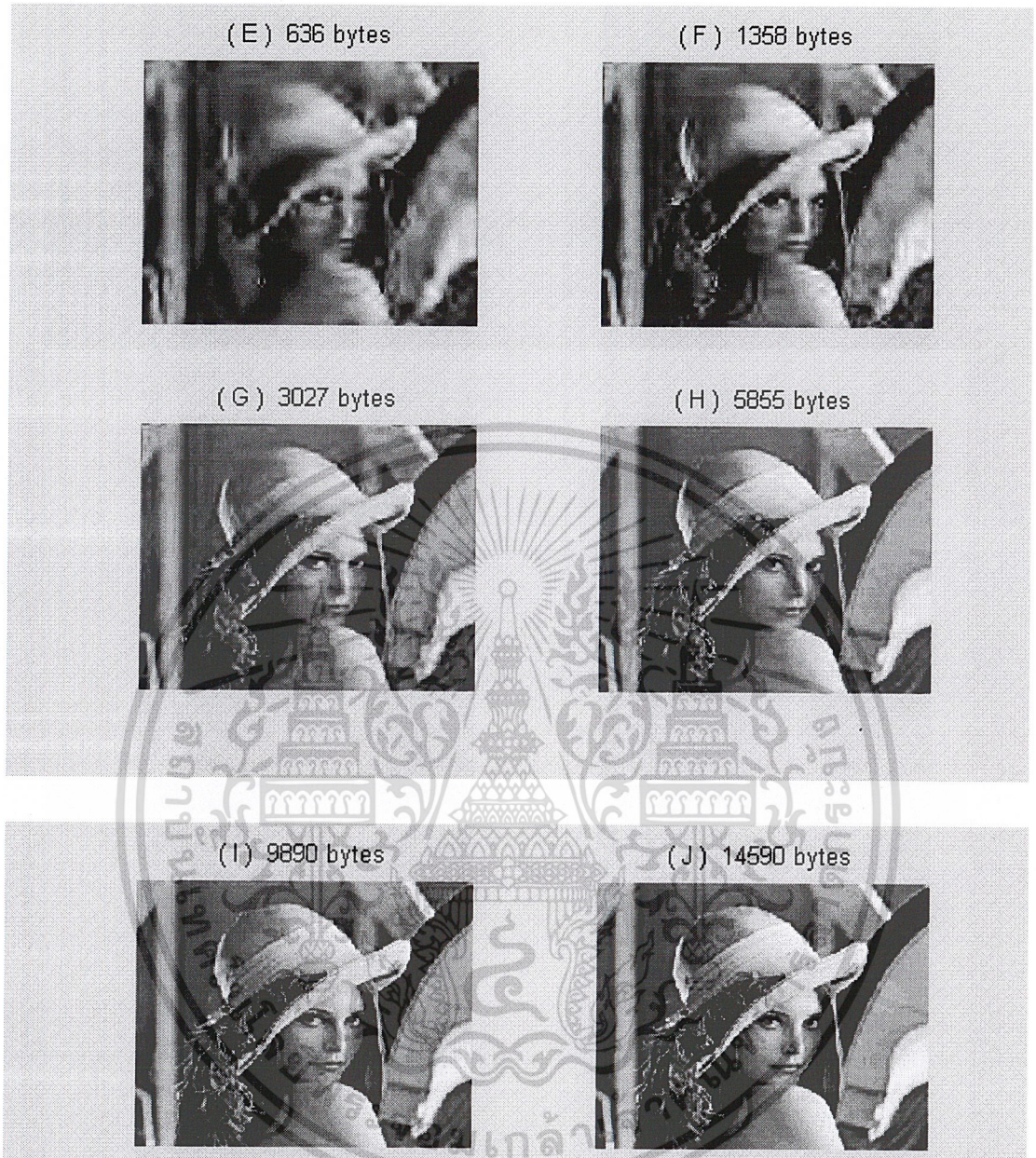
4.8.1) ทำการนำภาพ Lena ขนาด 256×256 จุดภาพ มาเข้ารหัสและแสดงผลแบบโปรเกรสซีฟ แสดงดังรูปที่ 4.46

ผลการทดลอง 4.8.1



รูปที่ 4.46 แสดงผลภาพแบบโปรเกรสซีฟที่ใช้วิธีการเอ็มเบดอิดซีโร่ทรีมาเข้ารหัสข้อมูล (ภาพ Lena ขนาด 256×256 จุดภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

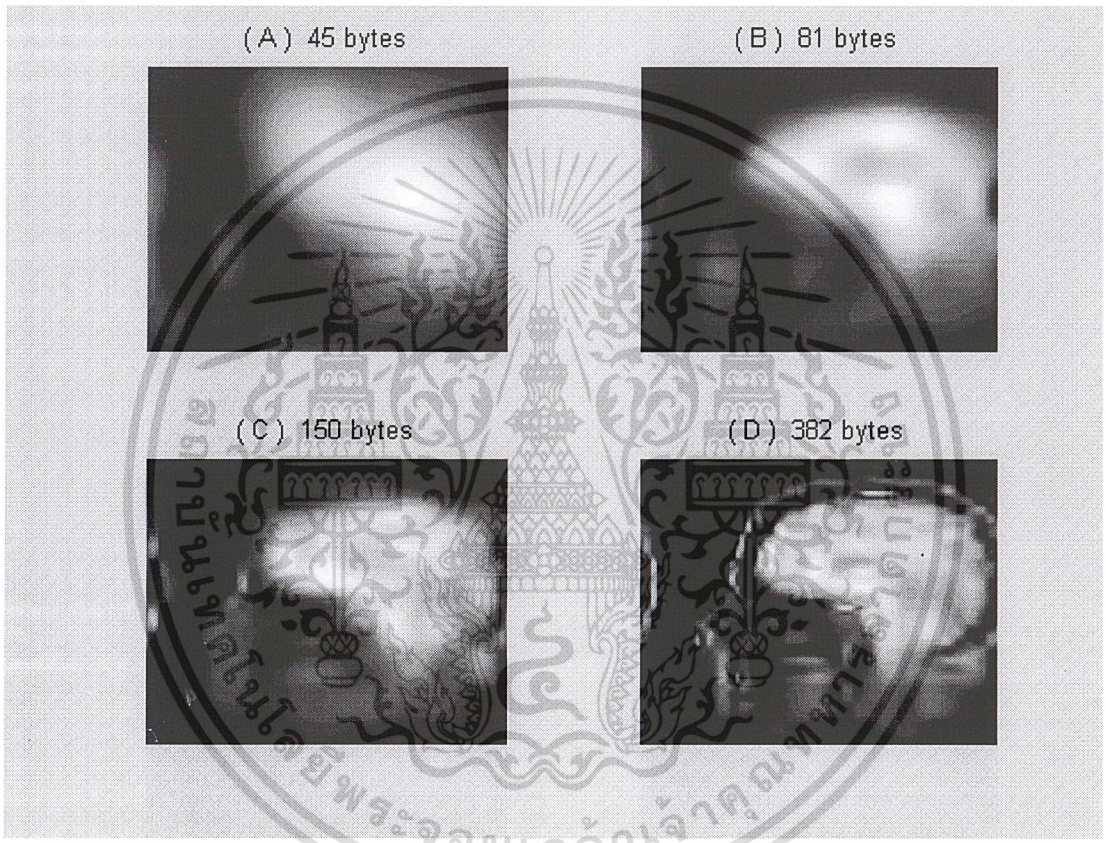


รูปที่ 4.46 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

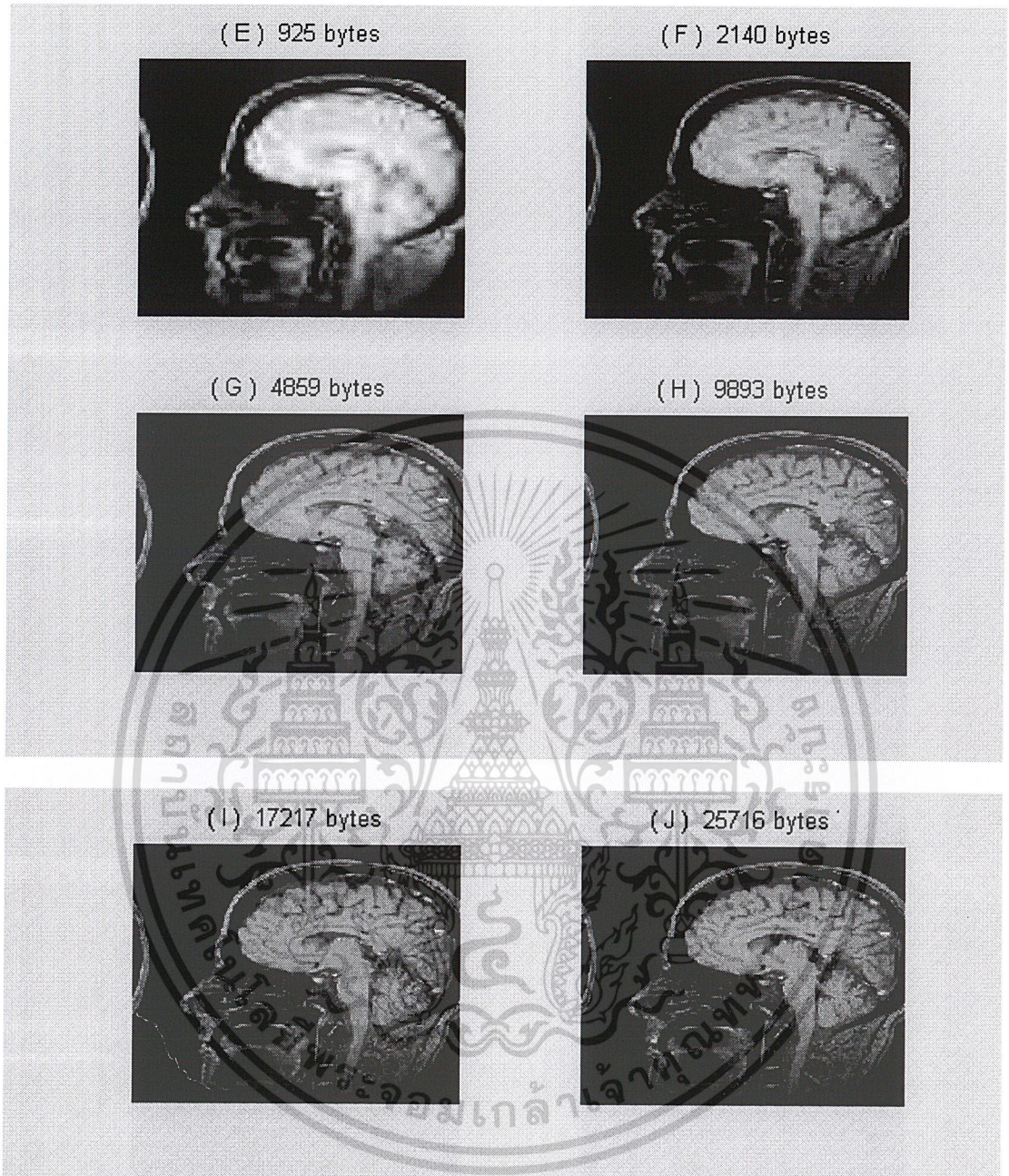
4.8.2) ทำการนำภาพ Brain ขนาด 256×256 จุดภาพ มาเข้ารหัสและแสดงผลแบบโปรเกรสซีฟ แสดงผลดังรูปที่ 4.47

ผลการทดลอง 4.8.2



รูปที่ 4.47 แสดงผลภาพแบบโปรเกรสซีฟที่ใช้วิธีการเอ็มเบดอิดซีโร่ทรีมาเข้ารหัสข้อมูล
(ภาพ Brain ขนาด 256×256 จุดภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.47 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองที่ 4.8 จะเห็นได้ว่า ในการแสดงผลครั้งแรกๆ ภาพจะมีลักษณะหยาบและค่อยๆ มีความคมชัดมากขึ้นในการแสดงผลครั้งต่อๆ ไป และเมื่อทำการพิจารณาภาพทั้งชุดแล้วพบว่าภาพในการแสดงผลก่อนหน้าเหมือนกับเป็นภาพในการแสดงผลครั้งต่อไปที่ผ่านตัวกรองความถี่ต่ำ ดั้งนั้นจึงสามารถทำการแสดงภาพแบบโปรเกรสซีฟได้อีกวิธีหนึ่ง โดยนำการแปลงเวฟเล็ตมาใช้เป็นตัวกรอง เพราะการแปลง เวฟเล็ตนี้สามารถเทียบได้กับการทำการแบ่งแบนด์ความถี่ของภาพนั่นเอง ดังนั้นถ้าทำการแบ่งข้อมูลเป็นเฟรมของข้อมูลทีละระดับของการแสดงผล ก็สามารถส่งข้อมูลเพื่อให้ปลายทางแสดงผลของภาพแบบโปรเกรสซีฟได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

5.1 บทวิจารณ์

การแปลงเวฟเล็ตเป็นการแปลงข้อมูลจากโดเมนของเวลา ให้ไปอยู่ในระนาบของเวลาและความถี่ โดยเหมือนกับการแยกสัญญาณที่ต้องการพิจารณาออกเป็นองค์ประกอบของสัญญาณย่อยๆ ที่เกิดขึ้นในช่วงเวลาต่างๆ การนำข้อมูลระดับเทาของภาพมาผ่านการแปลงเวฟเล็ตแบบไม่ต่อเนื่องให้เป็นองค์ประกอบของภาพในระนาบของเวลาและความถี่ โดยวิธีการแปลงเวฟเล็ตแบบไม่ต่อเนื่องด้วยวิธีไม่มาตรฐาน[7] จะเปรียบเสมือนการนำเอาภาพต้นแบบมาผ่านการกรองตามช่วงความถี่เพื่อสร้างเป็นภาพแบนด์ย่อย ซึ่งสัมประสิทธิ์ที่ใช้สำหรับการกรองความถี่นี้จะเป็นสัมประสิทธิ์ของเวฟเล็ตฟังก์ชัน โดยขึ้นอยู่กับเวฟเล็ตฟังก์ชันที่นำมาใช้ โดยสัมประสิทธิ์เหล่านี้จะมีคุณสมบัติเป็นตัวกรองความถี่ที่มีการตอบสนองอิมพัลส์ที่จำกัด[8] โดยจะให้เฟสของสัญญาณที่เป็นเชิงเส้น ทำให้การแยกองค์ประกอบของความถี่ แล้วนำความถี่เหล่านั้นมารวมกัน ก็จะได้สัญญาณตัวเดิมกลับขึ้นมา การเลือกฟังก์ชันเวฟเล็ตที่มีลักษณะคล้ายกับส่วนประกอบย่อยๆ ของสัญญาณอินพุต จะส่งผลให้การแยกองค์ประกอบของสัญญาณกระทำได้ดียิ่งขึ้น จากการทดลองมาพบว่าข้อมูลภาพจะมีความต่อเนื่องของสัญญาณอยู่มากในเนื้อของภาพ ดังนั้นถ้าทำการเลือกเวฟเล็ตฟังก์ชันที่ให้ความต่อเนื่องของข้อมูลก็จะทำให้ได้คุณภาพของเนื้อภาพที่ดี และในทางกลับกันขอบของภาพมักจะไม่มีความต่อเนื่องของข้อมูล ถ้าใช้เวฟเล็ตฟังก์ชันที่ไม่ให้ความต่อเนื่องของข้อมูลเช่น ฮาร์ฟังก์ชัน ก็จะทำให้ได้ขอบของภาพที่คมมากขึ้น แต่เนื้อภาพแย่ลงได้ ดังนั้นการเลือกใช้ฟังก์ชันเวฟเล็ตชนิดใดจะต้องพิจารณาเลือกให้เหมาะสมสำหรับการนำไปประยุกต์ใช้งาน

การนำสัมประสิทธิ์เวฟเล็ตมาเข้ารหัสข้อมูล เป็นขั้นตอนสำคัญในการลดขนาดข้อมูล โดยการลดขนาดข้อมูลภาพจะเกิดขึ้นในกระบวนการนี้ วิธีการที่นำเสนอในปริญญานิพนธ์นี้คือการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี โดยวิธีการนี้จะเป็นขั้นตอนที่ไม่สามารถย้อนกลับได้ เพราะขั้นตอนในการถอดรหัสไม่สามารถทำการระบุค่าของสัมประสิทธิ์ที่ตำแหน่งแน่นอน เป็นการประมาณค่าของข้อมูลต้นแบบ

การเข้ารหัสเอ็มเบดอิดซีโร่ทรีจะทำการจัดสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ตแบบไม่ต่อเนื่องเป็นชุดรหัสของข้อมูล โดยจำนวนของชุดรหัสนั้นขึ้นอยู่กับคุณภาพและอัตราการลดขนาดข้อมูลที่ต้องการนำชุดของรหัสดังกล่าวมาผ่านการเข้ารหัสเลขคณิตเพื่อทำการลดรหัสข้อมูลในขั้นตอนสุดท้าย

การนำเอาการแปลงเวฟเล็ตแบบไม่ต่อเนื่องมาใช้สำหรับการส่งข้อมูลภาพแบบโปรเกรสซีฟสามารถนำเอาการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรีมาใช้งานได้ เพราะวิธีการเข้ารหัสเอ็มเบดอิดซีโร่ทรีจะทำการเข้ารหัสในแบนด์ที่เป็นองค์ประกอบสำคัญของภาพก่อน นั่นคือองค์ประกอบทางความถี่ต่ำ และตามมาด้วยรายละเอียดของภาพที่เป็นองค์ประกอบทางความถี่สูง

จากการทดลองพบว่าเมื่อทดลองนำภาพ Lena ซึ่งเป็นภาพมาตรฐาน มาทำการทดลองการลดขนาดข้อมูลภาพโดยการแปลงเวฟเล็ต ที่นำสัมประสิทธิ์ของภาพที่ได้จากการแปลงเวฟเล็ตมาผ่านการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี พบว่าค่าของสัมประสิทธิ์ที่ได้เมื่อนำมาเข้ารหัสจะเกิดซีโร่ทรีที่น้อยเนื่องจากภาพ Lena จะมีรายละเอียดของภาพที่มาก แต่ในกรณีของภาพเอ็กซ์เรย์ ที่นำมาใช้ในการทดลองภาพจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นภาพที่เป็นระดับสีเดียวกันเป็นจำนวนมาก คือพื้นสีของฟิล์มเอ็กซ์เรย์ ดังนั้นเมื่อนำภาพนี้มาแปลงเวฟเล็ตแบบไม่ต่อเนื่องแล้วนำไปเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรีจะเกิดซีโร่ทรีเป็นจำนวนมากกว่าภาพของ Lena ซึ่งจะทำให้การลดขนาดข้อมูลภาพมีประสิทธิภาพได้มากกว่า

5.2 บทสรุป

การลดขนาดข้อมูลภาพ ด้วยวิธีการแปลงเวฟเล็ตแบบไม่ต่อเนื่องแล้วนำสัมประสิทธิ์ที่ได้จากการแปลงเวฟเล็ต ไปผ่านการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรี จากการทดลองสามารถลดขนาดข้อมูลภาพได้ต่ำกว่า 1 บิตต่อจุดภาพได้ โดยที่ค่าความผิดพลาดของภาพมีค่าที่อยู่ในระดับที่ทัดเทียมกับมาตรฐานของเจเพก และสามารถพอที่จะรับรู้รายละเอียดภายในภาพนั้นได้

แนวทางปรับปรุงและพัฒนา

การปรับปรุงการแปลงเวฟเล็ตแบบไม่ต่อเนื่อง

ควรทดลองหาเวฟเล็ตฟังก์ชันอื่นๆ มาใช้งานหรือทำการสร้างเวฟเล็ตฟังก์ชันที่เหมาะสมสำหรับข้อมูลภาพที่มีลักษณะตามคุณสมบัติของข้อมูลภาพ เพื่อการแปลงเวฟเล็ตที่เหมาะสมกับข้อมูลภาพที่สุด

การปรับปรุงการเข้ารหัสข้อมูล

เนื่องด้วยวิธีการเข้ารหัสแบบเอ็มเบดอิดซีโร่ทรีมีความซับซ้อน และช้า อันเนื่องมาจากวิธีการสแกนหาค่าสัมประสิทธิ์ ดังนั้นถ้าปรับปรุงวิธีการสแกนค่าสัมประสิทธิ์ให้รวดเร็วขึ้นก็จะทำให้น่าใช้งานมากยิ่งขึ้น

การกำหนดคุณภาพของข้อมูลภาพเอาต์พุต เนื่องจากในปริญญานิพนธ์นี้ได้เขียนโปรแกรมกำหนดเฉพาะขนาดเอาต์พุตที่ต้องการเท่านั้น มิได้กำหนดถึงคุณภาพของภาพเอาต์พุตที่ได้

การพัฒนา

ควรพัฒนาการเขียนโปรแกรมส่งข้อมูลแบบโปรเกรสซีฟโดยใช้การแปลงเวฟเล็ต เพื่อให้สามารถนำวิธีการลดขนาดข้อมูลภาพโดยการใช้การแปลงเวฟเล็ตไปประยุกต์ใช้งานได้จริง

การนำการแปลงเวฟเล็ตมาวิเคราะห์ข้อมูลภาพก่อนที่จะนำไปผ่านการเข้ารหัสลดขนาดข้อมูลภาพ อาทิเช่น อาทิเช่น ช่วยในการวิเคราะห์ภาพ เช่นการหาขอบ แล้วทำการลดข้อมูลในส่วนของพื้นผิวเพื่อในการสื่อสารอาจไม่จำเป็นในรายละเอียดในพื้นผิวมากนัก หรือทำการวิเคราะห์หาจุดสำคัญของภาพแล้วลดข้อมูลที่ไม่มีความสำคัญออกไป เช่นการนำมาประยุกต์ทางการแพทย์ในภาพเอ็กซ์เรย์ อาจต้องการวิเคราะห์เพียงบางส่วน บางจุดเท่านั้น ไม่จำเป็นต้องรักษารายละเอียดในส่วนอื่น ๆ นั้นก็คือ สามารถลดข้อมูลในส่วนที่ไม่จำเป็นลงได้ นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในภาคผนวก

		หน้า
โปรแกรมที่ 1.	โปรแกรมการแปลงภาพด้วยเวฟเล็ตด้วยวิธีมาตรฐาน	145
โปรแกรมที่ 2	โปรแกรมการแปลงภาพกลับด้วยวิธีมาตรฐาน	146
โปรแกรมที่ 3	โปรแกรมการแปลงภาพด้วยเวฟเล็ตด้วยวิธีไม่มาตรฐาน	146
โปรแกรมที่ 4	โปรแกรมแปลงกลับด้วยวิธีไม่มาตรฐาน	146
โปรแกรมที่ 5	โปแกรมการเข้ารหัสเอ็มเบดอิดซีโร่ทรี	147
โปรแกรมที่ 6	โปรแกรมการถอดรหัสเอ็มเบดอิดซีโร่ทรี	151
โปรแกรมที่ 7	โปแกรมการเข้ารหัสเอ็มเบดอิดซีโร่ทรี โดยพัฒนาวิธีการเข้ารหัส	155
โปรแกรมที่ 8	โปรแกรมการถอดรหัสเอ็มเบดอิดซีโร่ทรี โดยพัฒนาวิธีการถอดรหัส	159
โปรแกรมที่ 9	โปรแกรมหาค่าเทรชโฮลจากการตั้งค่าความผิดพลาด	162
โปรแกรมที่ 10	โปรแกรมปัดเทรชโฮล	162
โปรแกรมที่ 11	โปรแกรมหาค่าความผิดพลาดของภาพ	163
โปรแกรมที่ 12	โปรแกรมหาค่าเอ็นโทรปี	163
โปรแกรมที่ 13	โปรแกรมตรวจสอบขนาดของข้อมูลภาพ	164
โปรแกรมที่ 14	โปรแกรมเข้ารหัส 8 บิต	164
โปรแกรมที่ 15	โปรแกรมถอดรหัส 8 บิต	165
โปรแกรมที่ 16	โปรแกรมการแปลงเวฟเล็ต	166
โปรแกรมที่ 17	โปรแกรมการแปลงกลับเวฟเล็ต	166
โปรแกรมที่ 18	โปรแกรมแสดงผลภาพแบบโปรเกรสซีฟ	167
โปรแกรมที่ 19	โปรแกรมถอดรหัสจากการแบ่งข้อมูลสำหรับการแสดงผลแบบโปรเกรสซีฟ	169
โปรแกรมที่ 20	โปรแกรมการเข้ารหัส-ถอดรหัสเลขคณิต	175
โปรแกรมที่ 21	โปรแกรมเมนู	183
โปรแกรมที่ 22	โปรแกรมเรียกแฟ้มข้อมูลบนเมนู	184
โปรแกรมที่ 23	โปรแกรมป้อนค่าอัตราการบีบอัดข้อมูล	185
โปรแกรมที่ 24	โปรแกรมแสดงผลของภาพที่ทำการบีบอัดข้อมูล	186
โปรแกรมที่ 25	โปรแกรมจัดเก็บข้อมูลภาพ	186
โปรแกรมที่ 26	โปรแกรมแสดงภาพจากแฟ้มข้อมูล	187
โปรแกรมที่ 27	โปรแกรมแสดงภาพหน้าต่าง ผลของภาพที่ทำการบีบอัด	187
โปรแกรมที่ 28	โปรแกรมแสดงรายละเอียดของภาพที่ทำการบีบอัดข้อมูล	187
โปรแกรมที่ 29	โปรแกรมเมนูการเข้ารหัสเอ็มเบดอิดซีโร่ทรี	188
โปรแกรมที่ 30	โปรแกรมเมนูการถอดรหัสเอ็มเบดอิดซีโร่ทรี	191

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ โปรแกรมที่ 20 ทำงานบนระบบปฏิบัติการดอส โดยเขียนจากภาษาซี

PROGRAM NO.1

```
function g=stdn(c)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% standard transform
%%%%% c is input data (NxN)
%%%%% g is output data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d=[]; y=size(c);
for n=1:2
   c=c/sqrt(y(1));
   for row=1:y(1)
      w=y(2);
      while w>1
         for i=1:w/2
            d(i)     =( c(row,2*i-1) + c(row,2*i) )/sqrt(2);
            d(w/2+i)=( c(row,2*i-1) - c(row,2*i) )/sqrt(2);
         end
         c(row,1:w)=d(1:w);
         w=w/2;
      end
   end
   c=c';
end
g=c;
```

PROGRAM NO.2

```
function g=istdn(c)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% inverse standard transform
%%%%% c is input data (NxN)
%%%%% g is output data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d=[]; y=size(c);
for n=1:2
   c=c';
   for row=1:y(1)
      k=2;
      while k <=y(2)
         for i=1:k/2
            d(2*i-1) =( c(row,i) + c(row,k/2+i) )/sqrt(2);
            d(2*i)   =( c(row,i) - c(row,k/2+i) )/sqrt(2);
         end
         c(row,1:k)=d(1:k);
         k=2*k;
      end
   end
   c=c*sqrt(y(1));
end
g=round(c);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.3

```
function g=nonstdn(c)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% nonstandard transform
%%%%% c is input data (NxN)
%%%%% g is output data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d=[]; [h,w]=size(c); c=c/h;
while h>1
   for n=1:2
      for row=1:h
         for i=1:h/2
            d(i)    =( c(row,2*i-1) + c(row,2*i) )/sqrt(2);
            d(h/2+i)=( c(row,2*i-1) - c(row,2*i) )/sqrt(2);
         end
         c(row,1:h)=d(1:h);
      end
      c=c';
   end
   h=h/2;
end
g=c;
```

PROGRAM NO.4

```
function g=inonstdn(c)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% inverse nonstandard transform
%%%%% c is input data (NxN)
%%%%% g is output data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d=[]; y=size(c); NJ=log2(y(1));
for j=1:NJ
   for n=1:2
      c=c';
      for row=1:2^j
         k=2^j;
         for i=1:k/2
            d(2*i-1) =( c(row,i) + c(row,k/2+i) )/sqrt(2);
            d(2*i)   =( c(row,i) - c(row,k/2+i) )/sqrt(2);
         end
         c(row,1:k)=d(1:k);
      end
   end
end
g=round(c*y(1));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.5

```
function [header,symbols]=ezw_a(c,Nd)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%            FUNCTION [HEADER,SYMBOLS]=EZW_A(c,Nd)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  EZW_A  IS  EZW ENCODING TO SYMBOLS
%                      c  IS  FILE FOR ENCODE
%                     Nd  IS  NUMBER OF DOMINANTS
%                 HEADER  IS  HEADER FILE OF ENCODE
%                SYMBOLS  IS  SYMBOLS OF CODE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%   Initial value
dc=[]; ct=c; y=size(c); lv=log2(y(1));
hi_bi=ceil(log2(abs(max(c(:)))));
t0=2^(hi_bi)/2;          % initial threshold
mid=(2^(hi_bi)+t0)/2;    % medium value
header=[lv,t0,Nd];       % set header file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% LOOP EZW
for d=1:Nd
   t=t0/2^(d-1); half_t=mid/2^(d-1); Ns=Nd+1-d; d=ones(y(1));
   k3=find( ct<=2*t & ct>t );
   k4=find( ct>=-2*t & ct<-t );
   k5=find( abs(c) > 2*t );
   d(k3)=65;              % ' POS '
   d(k4)=66;              % ' NEG '
   d(k5)=0; ct(k5)=0;     % ' for skip '
   for j = 3:lv
      if j == 3
         for n = 1:4:5                    % n=1:2^(j-1):2^(j-1)+1
            for m = 1:4:5                 % m=1:2^(j-1):2^(j-1)+1
               for u = n:2:n+2            % u=n:2:n+2^(j-1)-2
                  for v = m:2:m+2         % v=m:2:m+2^(j-1)-2
                     for row = u:u+1
                        for col = v:v+1
                           td=d(row,col);
                           ab_ct=abs(ct(row,col));
                           rcmax=max([row col]);
                           if td == 0
                           else
                              if td > 64
                                 if td == 65
                                    dc=[dc 0 1]; % POS = 01
                                 else % td == 66
                                    dc=[dc 1 0]; % NEG = 10
                                 end
                                 ds=1; half=half_t;
                                 while ds <= Ns
                                    if ab_ct > half
                                       dc=[dc 1];
                                       half=half + t/2^(ds+1);
                                    else % if ab_ct < half
                                       dc=[dc 0];
                                       half=half - t/2^(ds+1);
                                    end
                                    ds=ds+1;
                                 end
                                 % SPACE
                              else % td == 1
                                 ls=lv; r=1; IZ=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    while ls > 1
                                       if rcmax <= 2^(r)
                                          row1=2^(ls-1)*row-...
                                             (2^(ls-1)-1):2^(ls-1)*row;
                                          col1=2^(ls-1)*col-(2^(ls-1)…
                                             -1):2^(ls-1)*col;
                                             ab_c=abs( ct(row1,col1) );
                                          if max(ab_c(:)) > t
                                             IZ=1; ls=1;
                                          end
                                       else
                                          r=r+1;
                                       end
                                       ls=ls-1;
                                    end
                                    if IZ == 1
                                       dc=[dc 1 1];     % IZ = 11
                                    else % if IZ == 0
                                       dc=[dc 0 0];     % ZTR = 00
                                       if rcmax > 2^(lv-1)
                                       else
                                          ls=lv; r=1;
                                          while ls >1
                                             if rcmax <= 2^(r)
                                                row1=2^(ls-1)*...
                                                   row-(2^(ls-1)-1)...
                                                   :2^(ls-1)*row;
                                                col1=2^(ls-1)*...
                                                   col-(2^(ls-1)-1)...
                                                   :2^(ls-1)*col;
                                                d(row1,col1) = 0;
                                             else
                                                r=r+1;
                                             end
                                             ls=ls-1;
                                          end
                                       end
                                    end
                                 end
                              end
                           end
                        end
                     end
                  end
               end
            end
         end
      end
   elseif (j >3)
      for n=1:2^(j-1):2^(j-1)+1
         for m=2^(j-1)+1:-2^(j-1):1
            if n ==1 & m ==1
               n=2^(j-1)+1;
            elseif n ==(2^(j-1)+1) & m ==1
               n=-1; m=-1; q=j; j=2;
            end
            for u=n:2:2^(j-1)+n-2
               for v=m:2:2^(j-1)+m-2
                  for row = u:u+1
                     for col = v:v+1
                        if (row <=0) & (col <=0)
                           j=q;
                        else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
td=d(row,col);
ab_ct=abs(ct(row,col));
rcmax=max([row col]);
if td == 0;
else
   if td > 64
      if td == 65
         dc=[dc 0 1]; % POS = 01
      else % td == 66
         dc=[dc 1 0]; % NEG = 10
      end
      ds=1; half=half_t;
      while ds <= Ns
         if ab_ct > half
            dc=[dc 1];
            half=half + t/2^(ds+1);
         else % if ab_ct < half
            dc=[dc 0];
            half=half - t/2^(ds+1);
         end
         ds=ds+1;
      end
      % SPACE
   else % td == 1
      ls=lv; r=1; IZ=0;
      while ls > 1
         if rcmax <= 2^(r)
            row1=2^(ls-1)*...
                 row-(2^(ls-1)-1)...
                 :2^(ls-1)*row;
            col1=2^(ls-1)*...
                 col-(2^(ls-1)-1)...
                 :2^(ls-1)*col;
            ab_c=abs( ct(row1,col1) );
            if max(ab_c(:)) > t
               IZ=1; ls=1;
            end
         else
            r=r+1;
         end
         ls=ls-1;
      end
      if IZ == 1
         dc=[dc 1 1];      % IZ = 11
      else % if IZ == 0
         dc=[dc 0 0];      % ZTR = 00
         if rcmax > 2^(lv-1)
         else
            ls=lv; r=1;
            while ls >1
               if rcmax <= 2^(r)
                  row1=2^(ls-1)*...
                       row-(2^(ls-...
                       1)-1):2^(ls-...
                       1)*row;
                  col1=2^(ls-1)*...
                       col-(2^(ls-...
                       1)-1):2^(ls-...
                       1)*col;
                  d(row1,col1) = 0;
               else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                                    r=r+1;
                                                end
                                                ls=ls-1;
                                            end
                                        end
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
symbols=dc;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.6

```
function g=dezw_a(header,symbols)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%            FUNCTION  g = DEZW_A(HEADER,SYMBOLS)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                DEZW_A  IS  EZW DECODING TO DATA
%                     g  IS  DATA OUT
%                HEADER  IS  HEADER FOR DECODE
%               SYMBOLS  IS  SYMBOLS FOR DECODE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%  Initial value  %%%%%
x=1; t0=header(2); mid=(2*t0+t0)/2; C=zeros(2^(header(1)));
Nd=header(3);   % get header
y=length(symbols); yt=y; y1=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% LOOP Inverse EZW
for d =1:Nd
   t=t0/2^(d-1); half_t=mid/2^(d-1); Ns=Nd+1-d;
   for j =3:header(1)
      if j == 3
         for n = 1:4:5              % n=1:2^(j-1):2^(j-1)+1
            for m = 1:4:5           % m=1:2^(j-1):2^(j-1)+1
               for u = n:2:n+2      % u=n:2:n+2^(j-1)-2
                  for v = m:2:m+2   % v=m:2:m+2^(j-1)-2
                     for row = u:u+1
                        for col = v:v+1
                           if abs(C(row,col)) >0
                           elseif (x <=y) & (yt-2 >=0)
                              half=half_t;
                              if symbols(x:x+1) == [0 1]; % POS
                                 x=x+2; ds=1;
                                 %%%%%%%%%%%%%%%%%%% check data
                                 y1=yt-2-Ns;
                                 if y1 >=0
                                    yt=y1;
                                 else
                                    ds=Ns+1; yt=y1;
                                 end
                                 %%%%%%%%%%%%%%%%%%%%
                                 while ds <= Ns
                                    if symbols(x) == 1
                                       half=half + t/2^(ds+1);
                                       x=x+1;
                                    else % if symbols(x) == 0
                                       half=half - t/2^(ds+1);
                                       x=x+1;
                                    end
                                    ds=ds+1;
                                 end
                                 C(row,col)= half ;
                              elseif symbols(x:x+1) == [1 0]; % NEG
                                 x=x+2; ds=1;
                                 %%%%%%%%%%%%%%%%%%% check data
                                 y1=yt-2-Ns;
                                 if y1 >=0
                                    yt=y1;
                                 else
                                    ds=Ns+1; yt=y1;
                                 end
                                 %%%%%%%%%%%%%%%%%%%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
                        while ds <= Ns
                            if symbols(x) == 1
                                half=half + t/2^(ds+1);
                                x=x+1;
                            else % if symbols(x) == 0
                                half=half - t/2^(ds+1);
                                x=x+1;
                            end
                            ds=ds+1;
                        end
                        C(row,col)= -half ;
                    elseif symbols(x:x+1) == [1 1]; % IZ
                        x=x+2; yt=yt-2;
                    elseif symbols(x:x+1) == [0 0]; % ZTR
                        x=x+2; yt=yt-2; ls=header(1); r=1;
                        while ls > 1
                            if max([row col]) <= 2^(r)
                                for h=(2^(ls-1))*row-...
                                    (2^(ls-1)-1):...
                                    (2^(ls-1))*row
                                  for g=(2^(ls-1))*...
                                    col-(2^(ls-1)-1):...
                                    (2^(ls-1))*col
                                  if abs(C(h,g)) >0
                                  else
                                    C(h,g) =1;
                                  end
                                end
                              end
                            else
                              r=r+1;
                            end
                            ls=ls-1;
                        end
                    end
                end
            end
        end
    end
elseif (j > 3)
    for n = 1:2^(j-1):2^(j-1)+1
        for m = 2^(j-1)+1:-(2^(j-1)):1
            if n ==1 & m ==1
                n=2^(j-1)+1;
            elseif n ==(2^(j-1)+1) & m ==1
                n=-1; m=-1; q=j; j=2;
            end
            for u = n:2:2^(j-1)+n-2
                for v = m:2:2^(j-1)+m-2
                    for row = u:u+1
                        for col = v:v+1
                            if (row <=0) & (col <=0)
                                j=q;
                            else
                                if abs(C(row,col)) >0
                                elseif (x <=y) & (yt-2 >=0)
                                    half=half_t;
                                    if symbols(x:x+1) == [0 1]; % POS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            x=x+2; ds=1;
            %%%%%%%%%%%%%%%%%% check data
            y1=yt-2-Ns;
            if y1 >=0
               yt=y1;
            else
               ds=Ns+1; yt=y1;
            end
            %%%%%%%%%%%%%%%%%%
            while ds <= Ns
               if symbols(x) == 1
                  half=half + t/2^(ds+1);
                  x=x+1;
               else % if symbols(x) == 0
                  half=half - t/2^(ds+1);
                  x=x+1;
               end
               ds=ds+1;
            end
            C(row,col)= half ;
         elseif symbols(x:x+1) == [1 0];% NEG
            x=x+2; ds=1;
            %%%%%%%%%%%%%%%%%% check data
            y1=yt-2-Ns;
            if y1 >=0
               yt=y1;
            else
               ds=Ns+1; yt=y1;
            end
            %%%%%%%%%%%%%%%%%%
            while ds <= Ns
               if symbols(x) == 1
                  half=half + t/2^(ds+1);
                  x=x+1;
               else % if symbols(x) == 0
                  half=half - t/2^(ds+1);
                  x=x+1;
               end
               ds=ds+1;
            end
            C(row,col)= -half ;
         elseif symbols(x:x+1) == [1 1]; % IZ
            x=x+2; yt=yt-2;
         elseif symbols(x:x+1) == [0 0];% ZTR
            x=x+2; yt=yt-2;ls=header(1); r=1;
            while ls > 1
               if max([row col]) <= 2^(r)
                  for h=(2^(ls-1))*row-...
                        (2^(ls-1)-1):...
                        (2^(ls-1))*row
                     for g=(2^(ls-1))*...
                         col-(2^(ls-1)-1):...
                         (2^(ls-1))*col
                        if abs(C(h,g)) >0
                        else
                           C(h,g) =1;
                        end
                     end
                  end
               else
                  r=r+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        end
                                        ls=ls-1;
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
    k=find( C == 1 ); C(k)=0;
end
g=C;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.7

```
function [header,symbols]=ezw_b(c,Nd)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%             FUNCTION [HEADER,SYMBOLS] = EZW_B(c,Nd)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  EZW_B  IS  EZW ENCODING TO SYMBOLS (TYPE B)
%                      c  IS  FILE FOR ENCODE
%                     Nd  IS  NUMBER OF DOMINANTS
%                 HEADER  IS  HEADER OF FILE
%                SYMBOLS  IS  SYMBOLS OF CODE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Initial value
ct=c; y=size(c); lv=log2(y(1));
hi_bi=ceil(log2(abs(max(c(:)))));
T_max=2^(hi_bi); T=T_max;
ht0=log2(T_max/2);  % initial threshold
header=[lv,ht0,Nd]; % set header file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DP =[]; % DP  = dominant pass
RV =[]; % RV  = reconstruction values
SL =[]; % SL  = subordinate list
%%%%% LOOP EZW_B

for nd=1:Nd
   T = T/2; d=ones(y(1));
   % find POS & NEG
   k3=find( ct <2*T & ct >=T );
   k4=find( ct >-2*T & ct<=-T );
   k5=find( abs(c) >= 2*T );
   d(k3)=65;              % ' POS '
   d(k4)=66;              % ' NEG '
   d(k5)=0; ct(k5)=0; % ' for skip '

   %%%%% quantized
   if nd ~= 1
      for N=1:length(SL)
         if SL(N) >= 2*T
            if SL(N) >= RV(N)
               DP=[DP 1];
               RV(N)=RV(N)+2*T/4;
            else
               DP=[DP 0];
               RV(N)=RV(N)-2*T/4;
            end
         end
      end
   end


   for j = 3:lv
      if j == 3
         for n = 1:4:5                    % n=1:2^(j-1):2^(j-1)+1
            for m = 1:4:5                 % m=1:2^(j-1):2^(j-1)+1
               for u = n:2:n+2            % u=n:2:n+2^(j-1)-2
                  for v = m:2:m+2         % v=m:2:m+2^(j-1)-2
                     for row = u:u+1
                        for col = v:v+1
                           td=d(row,col);
                           ab_ct=abs(ct(row,col));
                           rcmax=max([row col]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        if td == 0
                        else
                           if td > 64
                              if td == 65
                                 DP=[DP 0 1]; % POS = 01
                                 RV=[RV 1.5*T];
                                 SL=[SL ab_ct];
                              else % td == 66
                                 DP=[DP 1 0]; % NEG = 10
                                 RV=[RV 1.5*T];
                                 SL=[SL ab_ct];
                              end
                           else % td == 1
                              ls=lv; r=1; IZ=0;
                              while ls > 1
                                 if rcmax <= 2^(r)
                                    if max(max(abs(ct(2^(ls...
                                       -1)*row-(2^(ls-1)...
                                       -1) : 2^(ls-1) * ...
                                       row ,2^(ls-1)*col...
                                       -(2^(ls-1)-1) :2^...
                                       (ls-1)*col)))) > T
                                       IZ=1; ls=1;
                                    end
                                 else
                                    r=r+1;
                                 end
                                 ls=ls-1;
                              end
                              if (IZ == 1)|(rcmax > 2^(lv-1))
                                 DP=[DP 1 1]; % IZ = 11
                              else % IZ == 0
                                 DP=[DP 0 0]; % ZTR = 00
                                 ls=lv; r=1;
                                 while ls >1
                                    if rcmax <= 2^(r)
                                       d(2^(ls-1)*row-(2^...
                                       (ls-1)-1) :  2^...
                                       (ls-1)*row , 2^...
                                       (ls-1) *col-(2^...
                                       (ls-1)-1) :  2^...
                                       (ls-1)*col) = 0;
                                    else
                                       r=r+1;
                                    end
                                    ls=ls-1;
                                 end
                              end
                           end
                        end
                     end
                  end
               end
            end
         end
      end
   end
elseif (j >3)
   for n=1:2^(j-1):2^(j-1)+1
      for m=2^(j-1)+1:-2^(j-1):1
         if n ==1 & m ==1
            n=2^(j-1)+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
elseif n ==(2^(j-1)+1) & m ==1
   n=-1; m=-1; q=j; j=2;
end
for u=n:2:2^(j-1)+n-2
   for v=m:2:2^(j-1)+m-2
      for row = u:u+1
         for col = v:v+1
            if (row <=0) & (col <=0)
               j=q;
            else
               td=d(row,col);
               ab_ct=abs(ct(row,col));
               rcmax=max([row col]);
               if td == 0;
               else
                  if td > 64
                     if td == 65
                        DP=[DP 0 1]; % POS = 01
                        RV=[RV 1.5*T];
                        SL=[SL ab_ct];
                     else % td == 66
                        DP=[DP 1 0]; % NEG = 10
                        RV=[RV 1.5*T];
                        SL=[SL ab_ct];
                     end
                  else % td == 1
                     ls=lv; r=1; IZ=0;
                     while ls > 1
                        if rcmax <= 2^(r)
                           if max(max(abs(ct(2^...
                                 (ls-1)*row-(2^...
                                 (ls-1)-1) : 2^...
                                 (ls-1)*row ,2^...
                                 (ls-1)*col-(2^...
                                 (ls-1)-1) : 2^...
                                 (ls-1)*col))))> T
                              IZ=1; ls=1;
                           end
                        else
                           r=r+1;
                        end
                        ls=ls-1;
                     end
                     if (IZ == 1)|(rcmax > 2^(lv-1))
                        DP=[DP 1 1]; % IZ = 11
                     else % IZ == 0
                        DP=[DP 0 0]; % ZTR = 00
                        ls=lv; r=1;
                        while ls >1
                           if rcmax <= 2^(r)
                              d(2^(ls-1)*row...
                                 -( 2^(ls-1)...
                                 -1):  2^(ls...
                                 -1)*row ,2^...
                                 (ls-1)* col...
                                 -( 2^(ls-1)...
                                 -1):  2^(ls...
                                 -1)*col) = 0;
                           else
                              r=r+1;
                           end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        ls=ls-1;
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
end
symbols=DP;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.8

```
function g=dezw_b(header,symbols)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  FUNCTION g = DEZW_B(HEADER,SYMBOLS)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  DEZW_B  IS  EZW DECODING TO DATA (TYPE B)
%                       g  IS  DATA OUT
%                  HEADER  IS  HEADER FILE
%                  SYMBOL  IS  SYMBOLS FOR DECODE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
yt=length(symbols);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x=1; T=2^(header(2)+1); C=zeros(2^(header(1))); Nd=header(3);
ROW=[]; COL=[];
%%%%% LOOP DEZW_B
for nd =1:Nd
   T=T/2;
   for j =3:header(1)
      if yt > 0
        if j == 3
           for n = 1:4:5                    % n=1:2^(j-1):2^(j-1)+1
             for m = 1:4:5                  % m=1:2^(j-1):2^(j-1)+1
               for u = n:2:n+2              % u=n:2:n+2^(j-1)-2
                 for v = m:2:m+2            % v=m:2:m+2^(j-1)-2
                   for row = u:u+1
                     for col = v:v+1
                       if abs(C(row,col)) >0
                       else
                          if (yt-2) >= 0
                         half=1.5*T;
                         if symbols(x:x+1) == [0 1];
                          % POS symbol
                          x=x+2; yt=yt-2;
                          ROW=[ROW row];
                          COL=[COL col];
                          C(row,col)= half ;
                        elseif symbols(x:x+1)==[1 0];
                          % NEG symbol
                          x=x+2; yt=yt-2;
                          ROW=[ROW row];
                          COL=[COL col];
                          C(row,col)= -half ;
                        elseif symbols(x:x+1)==[1 1];
                          % IZ symbol
                          x=x+2; yt=yt-2;
                       elseif symbols(x:x+1)==[0 0];
                          % ZTR symbol
                          x=x+2; yt=yt-2;
                          ls=header(1); r=1;
                          while ls > 1
                             if max([row col])<=2^(r)
                                for h=(2^(ls-1))*...
                                      row-(2^(ls-...
                                      1)-1) : (2^...
                                      (ls-1))*row
                                   for g=(2^(ls-1))...
                                         *col-( 2^(...
                                         ls-1)-1):(...
                                         2^(ls-1)*col
                                      if abs(C(h,g))>0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปแสวงหาประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                                    else
                                                       C(h,g) =1;
                                                    end
                                                 end
                                              end
                                           else
                                              r=r+1;
                                           end
                                           ls=ls-1;
                                        end
                                     end
                                  else
                                     yt=yt-2;
                                  end
                               end
                            end
                         end
                      end
                   end
                end
             end
          end
       elseif (j > 3)
          for n = 1:2^(j-1):2^(j-1)+1
             for m = 2^(j-1)+1:-(2^(j-1)):1
                if n ==1 & m ==1
                   n=2^(j-1)+1;
                elseif n ==(2^(j-1)+1) & m ==1
                   n=-1; m=-1; q=j; j=2;
                end
                for u = n:2:2^(j-1)+n-2
                   for v = m:2:2^(j-1)+m-2
                      for row = u:u+1
                         for col = v:v+1
                            if (row <=0) & (col <=0)
                               j=q;
                            else
                               if abs(C(row,col)) >0
                               else
                                  if (yt-2) >= 0
                                  half=1.5*T;
                                  if symbols(x:x+1)==[0 1];
                                     % POS symbol
                                     x=x+2; yt=yt-2;
                                     ROW=[ROW row];
                                     COL=[COL col];
                                     C(row,col)= half ;
                                  elseif symbols(x:x+1)==[1 0];
                                     % NEG symbol
                                     x=x+2; yt=yt-2;
                                     ROW=[ROW row];
                                     COL=[COL col];
                                     C(row,col)= -half ;
                                  elseif symbols(x:x+1)==[1 1];
                                     % IZ symbol
                                     x=x+2; yt=yt-2;
                                  elseif symbols(x:x+1)==[0 0];
                                     % ZTR symbol
                                     x=x+2; yt=yt-2;
                                     ls=header(1); r=1;
                                     while ls > 1
                                        if max([row col])<=2^(r)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
                                                    for h=(2^(ls-1))*...
                                                          row-(2^(ls-...
                                                          1)-1) : (2^...
                                                          (ls-1))*row
                                                       for g=(2^(ls...
                                                          -1)) *col...
                                                          -(  2^(ls...
                                                          -1)-1):  ...
                                                          2^(ls-1))*col

                                                          if abs(C(h,g))...
                                                             >0

                                                          else
                                                             C(h,g) =1;
                                                          end
                                                       end
                                                    end
                                                 else
                                                    r=r+1;
                                                 end
                                                 ls=ls-1;
                                              end
                                           end
                                        else
                                           yt=yt-2;
                                        end
                                     end
                                  end
                               end
                            end
                         end
                      end
                   end
                end
             end
          end
       end
    end
end
k=find( C == 1 ); C(k)=0;
% Reconstruction values
if yt >= 1
   for N=1:length(ROW)
      if yt > 0
         if C(ROW(N),COL(N)) > 0
            if symbols(x)==1
               C(ROW(N),COL(N))=C(ROW(N),COL(N))+T/4;
            else
               C(ROW(N),COL(N))=C(ROW(N),COL(N))-T/4;
            end
            x=x+1; yt=yt-1;
         else % C(ROW(N),COL(N)) < 0
            if symbols(x)==1
               C(ROW(N),COL(N))= -( -C(ROW(N),COL(N))+T/4 );
            else
               C(ROW(N),COL(N))= -( -C(ROW(N),COL(N))-T/4 );
            end
            x=x+1; yt=yt-1;
         end
      end
   end
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end
g=C;
```

PROGRAM NO.9

```
function [Y,th]=compress(C,E)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% find threshold for error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [Y,th]=COMPRESS(C,E);
% C is Original image file
% E is Error between Original & Reconstruction image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[y1 y2]=size(C);
SC=sort(abs(C(:)'));
%%%%% Find threshold for Compress
tmin = min(SC);
tmax = max(SC);
while abs(tmax-tmin) > 0.0000001
   t=(tmin+tmax)/2; s=0;
   for n = (y1*y2):-1:1
      if abs(SC(1,n)) < t
         s = s+SC(1,n)^2;
      end
   end
   if s <= E^2
      tmin=t;
   else
      tmax=t;
   end
end
for row = 1:y1
   for col = 1:y2
      if abs(C(row,col)) <= t
         C(row,col) = 0;
      end
   end
end
Y=C; th=t;
```

PROGRAM NO.10

```
function g=threshold(C,t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% threshold value
%%%%% t is threshold
%%%%% C is input data (NxN)
%%%%% g is output data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y=size(C);
for row = 1:y(1)
   for col = 1:y(2)

      if abs(C(row,col)) <= t
         C(row,col) = 0;
      end
   end
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end
g=C;
```

PROGRAM NO.11

```
function [mse,snr,psnr]=err(P,C)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% err  is Measurement error
% mse  is Mean square error
% snr  is Signal to noise ratio
% psnr is Power signal to noise ratio
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Eo=C.^2; Eo=sum(Eo(:));
Er=(P-C).^2; Er=sum(Er(:)); y=size(P);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mse=Er/(y(1)^2);      % Mean square error
snr=Eo/Er;            % Signal to noise ratio
psnr=10*log10(snr);   % Power signal to noise ratio
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display
ResultError=sprintf('Mean Square Error(MSE) = %f\nSignal to …
Noise Ratio(SNR) = %f\nPower Signal to Noise Ratio(PSNR) = %f …
dB',mse,snr,psnr)
```

PROGRAM NO.12

```
function y = entropy(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    function y = entropy
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% find frequency
F=[]; u=[]; v=[];
k=max(x)+1;
for i=1:k;
   j=find(x==i-1);
   u(i)=length(j);
end
r=find(u>0);
v=u(r);
v=sort(v);
m=length(v);
for i=1:m;
   F(i)=v(m-i+1);
end
P=F/length(x);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%  calculation entropy
y=sum(-P.*log2(P));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.13

```
function g=cut_size(x,size_let)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% cut_size is function for find size compress
% size_let is size compress
%        x is data stream for find size
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
code_symbol=x; N=length(code_symbol);
n=floor(N/2); N_stop=n; k=2;
hmax=entropy(code_symbol); % find entropy max
size_max=hmax*length(code_symbol)/8;
hmid=entropy(code_symbol(1:n)); % find entropy mid
size_mid=hmid*length(code_symbol(1:n))/8;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if size_let < size_max
   while N_stop > 1
      if (size_let > size_mid) & (size_let < size_max)
         N_stop=floor(N/2^k); n=n+N_stop;
         hmid=entropy(code_symbol(1:n));
         size_mid=hmid*length(code_symbol(1:n))/8;
      else
         size_max=size_mid;
         N_stop=floor(N/2^k); n=n-N_stop;
         hmid=entropy(code_symbol(1:n));
         size_mid=hmid*length(code_symbol(1:n))/8;
      end
      k=k+1;
   end
   g=x(1:n);
else
   disp('MUST LESS THAN COMPRESSED SIZE')
end
```

PROGRAM NO.14

```
function store=bit_shift(bit_stream)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    bit_shift  is function for encoding
%    bit_stream is input for encoding
%    store      is output stream
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
l_stream =length(bit_stream);
mask =128; code =0; store =[];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:l_stream
   if bit_stream(i) ==1
      code =bitor(mask,code);
   end
   mask =bitshift(mask,-1);
   if mask ==0
      mask =128;
      store =[store code];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        code =0;
    end
end
```

PROGRAM NO.15

```
function out=bit_reverse(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  bit_reverse is function inverse bit_shift
%             x is input for decoding
%           out is output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
out=[]; l_x=length(x);
%%%%% Loop reverse shift bit
for n=1:l_x
   d2b=double(dec2bin(x(n)));
   if length(d2b)==1
      d2b=[48 48 48 48 48 48 48 d2b];
      out=[out d2b];
   elseif length(d2b)==2
      d2b=[48 48 48 48 48 48 d2b];
      out=[out d2b];
   elseif length(d2b)==3
      d2b=[48 48 48 48 48 d2b];
      out=[out d2b];
   elseif length(d2b)==4
      d2b=[48 48 48 48 d2b];
      out=[out d2b];
   elseif length(d2b)==5
      d2b=[48 48 48 d2b];
      out=[out d2b];
   elseif length(d2b)==6
      d2b=[48 48 d2b];
      out=[out d2b];
   elseif length(d2b)==7
      d2b=[48 d2b];
      out=[out d2b];
   else
      out=[out d2b];
   end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% replace data to stream output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a=find(out==48);
b=find(out==49);
out(a)=0;
out(b)=1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.16

```
function g=wtp2(f,arg,NJ)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% wtp2 is (Discrete)Wavelet transform 2-D
% f    is Images files
% arg  is Type Wavelet transform(filters coefficients)
%        ex. 'db1,'db2' etc.
% NJ   is Number of scale transform
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[h0,h1]=wfilters(arg,'d'); % Load wavelet coefficient from MATLAB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N=length(h0); y=size(f);
if nargin==2
   NJ=log2(y(2));
end
for j=1:NJ
   L=length(f); LJ=L/2;
   for xy=1:2
      c=[f(:,(mod((-(N-1):-1),L)+1)) f];     % Make periodic
      for m=1:y(1)
         dw=conv(c(m,:),h1); tw(m,LJ+1:2*LJ) =dw(N:2:(N+L-2));
         cw=conv(c(m,:),h0); tw(m,1:LJ)      =cw(N:2:(N+L-2));
      end
      f=tw(1:y(1),1:y(1))';
   end
   tw(1:y(1),1:y(1))=f; f=f(1:LJ,1:LJ); y(1)=y(1)/2;
end
g=tw;
```

PROGRAM NO.17

```
function f=iwtp2(g,arg,NJ)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% iwtp2 is Inverse (Discrete)wavelet transform 2-D
%     g is Images files
%   arg is Type wavelet transform(filters coefficients)
%          ex. 'db1','db2' etc.
%    NJ is Number of scale transform
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[g0,g1]=wfilters(arg,'r');  % Load wavelet coefficient from MATLAB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N=length(g0); L=length(g);
if nargin==2
   NJ=log2(L);
end
LJ=L/(2^NJ);
for j=1:NJ
   w=mod(0:N/2-1,LJ)+1;   % Make periodic
   for xy=1:2
      g=g';
      for m=1:2*LJ
         c=g(m,1:LJ); d=g(m,LJ+1:2*LJ);
         cu(1:2:2*LJ+N)=[c c(1,w)]; du(1:2:2*LJ+N)=[d d(1,w)];
         cw=conv(cu,g0)+conv(du,g1); g(m,1:2*LJ)=cw(N:N+2*LJ-1);
      end
   end
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   LJ=2*LJ;
end
f=round(abs(g));
```

PROGRAM NO.18

```
function progress(hd,sym,sub)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% hd  is header file
% sym is symbols dominant
% sub is symbols subordinate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[ns1,ns2,ns3,ns4,ns5,ns6,ns7,ns8,ns9,ns10,ns11,ns12,...
     sc1,sc2,sc3,sc4,sc5,sc6,sc7,sc8,sc9,sc10,sc11,sc12]=...
      partial_ds_12(sym,sub);
% dominant
[stn1,n1]=bit_shift_prog(ns1); stn1=[hd n1 stn1];
[stn2,n2]=bit_shift_prog(ns2); stn2=[n2 stn2];
[stn3,n3]=bit_shift_prog(ns3); stn3=[n3 stn3];
[stn4,n4]=bit_shift_prog(ns4); stn4=[n4 stn4];
[stn5,n5]=bit_shift_prog(ns5); stn5=[n5 stn5];
[stn6,n6]=bit_shift_prog(ns6); stn6=[n6 stn6];
[stn7,n7]=bit_shift_prog(ns7); stn7=[n7 stn7];
[stn8,n8]=bit_shift_prog(ns8); stn8=[n8 stn8];
[stn9,n9]=bit_shift_prog(ns9); stn9=[n9 stn9];
[stn10,n10]=bit_shift_prog(ns10); stn10=[n10 stn10];
[stn11,n11]=bit_shift_prog(ns11); stn11=[n11 stn11];
[stn12,n12]=bit_shift_prog(ns12); stn12=[n12 stn12];
% subordinate
[stc1,c1]=bit_shift_prog(sc1); stc1=[c1 stc1];
[stc2,c2]=bit_shift_prog(sc2); stc2=[c2 stc2];
[stc3,c3]=bit_shift_prog(sc3); stc3=[c3 stc3];
[stc4,c4]=bit_shift_prog(sc4); stc4=[c4 stc4];
[stc5,c5]=bit_shift_prog(sc5); stc5=[c5 stc5];
[stc6,c6]=bit_shift_prog(sc6); stc6=[c6 stc6];
[stc7,c7]=bit_shift_prog(sc7); stc7=[c7 stc7];
[stc8,c8]=bit_shift_prog(sc8); stc8=[c8 stc8];
[stc9,c9]=bit_shift_prog(sc9); stc9=[c9 stc9];
[stc10,c10]=bit_shift_prog(sc10); stc10=[c10 stc10];
[stc11,c11]=bit_shift_prog(sc11); stc11=[c11 stc11];
[stc12,c12]=bit_shift_prog(sc12); stc12=[c12 stc12];
% write files
fid=fopen('stn1.txt','w');fwrite(fid,stn1,'int8');fclose(fid);
fid=fopen('stn2.txt','w');fwrite(fid,stn2,'int8');fclose(fid);
fid=fopen('stn3.txt','w');fwrite(fid,stn3,'int8');fclose(fid);
fid=fopen('stn4.txt','w');fwrite(fid,stn4,'int8');fclose(fid);
fid=fopen('stn5.txt','w');fwrite(fid,stn5,'int8');fclose(fid);
fid=fopen('stn6.txt','w');fwrite(fid,stn6,'int8');fclose(fid);
fid=fopen('stn7.txt','w');fwrite(fid,stn7,'int8');fclose(fid);
fid=fopen('stn8.txt','w');fwrite(fid,stn8,'int8');fclose(fid);
fid=fopen('stn9.txt','w');fwrite(fid,stn9,'int8');fclose(fid);
fid=fopen('stn10.txt','w');fwrite(fid,stn10,'int8');fclose(fid);
fid=fopen('stn11.txt','w');fwrite(fid,stn11,'int8');fclose(fid);
fid=fopen('stn12.txt','w');fwrite(fid,stn12,'int8');fclose(fid);
%
fid=fopen('stc1.txt','w');fwrite(fid,stc1,'int8');fclose(fid);
fid=fopen('stc2.txt','w');fwrite(fid,stc2,'int8');fclose(fid);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fid=fopen('stc3.txt','w');fwrite(fid,stc3,'int8');fclose(fid);
fid=fopen('stc4.txt','w');fwrite(fid,stc4,'int8');fclose(fid);
fid=fopen('stc5.txt','w');fwrite(fid,stc5,'int8');fclose(fid);
fid=fopen('stc6.txt','w');fwrite(fid,stc6,'int8');fclose(fid);
fid=fopen('stc7.txt','w');fwrite(fid,stc7,'int8');fclose(fid);
fid=fopen('stc8.txt','w');fwrite(fid,stc8,'int8');fclose(fid);
fid=fopen('stc9.txt','w');fwrite(fid,stc9,'int8');fclose(fid);
fid=fopen('stc10.txt','w');fwrite(fid,stc10,'int8');fclose(fid);
fid=fopen('stc11.txt','w');fwrite(fid,stc11,'int8');fclose(fid);
fid=fopen('stc12.txt','w');fwrite(fid,stc12,'int8');fclose(fid);
% encoding
dos('arith e stn1.txt stn1.pgw');
dos('arith e stn2.txt stn2.pgw');
dos('arith e stn3.txt stn3.pgw');
dos('arith e stn4.txt stn4.pgw');
dos('arith e stn5.txt stn5.pgw');
dos('arith e stn6.txt stn6.pgw');
dos('arith e stn7.txt stn7.pgw');
dos('arith e stn8.txt stn8.pgw');
dos('arith e stn9.txt stn9.pgw');
dos('arith e stn10.txt stn10.pgw');
dos('arith e stn11.txt stn11.pgw');
dos('arith e stn12.txt stn12.pgw');
%
dos('arith e stc1.txt stc1.pgw');
dos('arith e stc2.txt stc2.pgw');
dos('arith e stc3.txt stc3.pgw');
dos('arith e stc4.txt stc4.pgw');
dos('arith e stc5.txt stc5.pgw');
dos('arith e stc6.txt stc6.pgw');
dos('arith e stc7.txt stc7.pgw');
dos('arith e stc8.txt stc8.pgw');
dos('arith e stc9.txt stc9.pgw');
dos('arith e stc10.txt stc10.pgw');
dos('arith e stc11.txt stc11.pgw');
dos('arith e stc12.txt stc12.pgw');
%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.19

```
function deprogess(X)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% X is original image file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
stop =0; k=0;
while stop ~= 1
   if k==0
      if (fopen('stn1.pgw','r') ~= -1) & …
         (fopen('stc1.pgw','r') ~= -1)
         % read size of files
         fid=fopen('stn1.pgw','r');
         [temp,szn1]=fread(fid);fclose(fid);
         fid=fopen('stc1.pgw','r');
         [temp,szc1]=fread(fid);fclose(fid);
         % read files
         dos('arith d stn1.pgw stn1.tmp');
         dos('arith d stc1.pgw stc1.tmp');
         fid=fopen('stn1.tmp','r');stn1=fread(fid)';fclose(fid);
         fid=fopen('stc1.tmp','r');stc1=fread(fid)';fclose(fid);
         k=1;
         %data
         hd=stn1(1:3);
         in1=stn1(4); istn1=stn1(5:length(stn1));
         ic1=stc1(1); istc1=stc1(2:length(stc1));
         osym1=bit_reverse_prog(istn1,in1);
         osub1=bit_reverse_prog(istc1,ic1);
         out1=total(hd(3),osym1,osub1);
         disp('   progress 1st time')
         tsz=szn1+szc1,%pause;
      end
   elseif k==1
      if (fopen('stn2.pgw','r') ~= -1) & …
         (fopen('stc2.pgw','r') ~= -1)
         % read size of files
         fid=fopen('stn2.pgw','r');
         [temp,szn2]=fread(fid);fclose(fid);
         fid=fopen('stc2.pgw','r');
         [temp,szc2]=fread(fid);fclose(fid);
         % read files
         dos('arith d stn2.pgw stn2.tmp');
         dos('arith d stc2.pgw stc2.tmp');
         fid=fopen('stn2.tmp','r');stn2=fread(fid)';fclose(fid);
         fid=fopen('stc2.tmp','r');stc2=fread(fid)';fclose(fid);
         k=2;
         %data

         in2=stn2(1); istn2=stn2(2:length(stn2));
         ic2=stc2(1); istc2=stc2(2:length(stc2));
         osym2=bit_reverse_prog(istn2,in2);
         osub2=bit_reverse_prog(istc2,ic2);
         out2=total(hd(3)-1,osym2,osub2);
         disp('   progress 2nd time')
         tsz=tsz+szn2+szc2,%pause;
      end
   elseif k==2
      if (fopen('stn3.pgw','r') ~= -1) & …
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        (fopen('stc3.pgw','r') ~= -1)
        % read size of files
        fid=fopen('stn3.pgw','r');
        [temp,szn3]=fread(fid);fclose(fid);
        fid=fopen('stc3.pgw','r');
        [temp,szc3]=fread(fid);fclose(fid);
        % read files
        dos('arith d stn3.pgw stn3.tmp');
        dos('arith d stc3.pgw stc3.tmp');
        fid=fopen('stn3.tmp','r');stn3=fread(fid)';fclose(fid);
        fid=fopen('stc3.tmp','r');stc3=fread(fid)';fclose(fid);
        k=3;
        %
        in3=stn3(1); istn3=stn3(2:length(stn3));
        ic3=stc3(1); istc3=stc3(2:length(stc3));
        osym3=bit_reverse_prog(istn3,in3);
        osub3=bit_reverse_prog(istc3,ic3);
        out3=total(hd(3)-2,osym3,osub3);
        disp('   progress 3rd time')
        %
        out=[out1 out2 out3];
        iout=dezw2datan(hd,out);
        i_out=iwtp2dbn(iout,'db5'); o3=i_out;figure
        imagesc(i_out);colormap(gray);
        er(X,i_out);
        tsz=tsz+szn3+szc3,%pause
    end
elseif k==3
    if (fopen('stn4.pgw','r') ~= -1) & ...
        (fopen('stc4.pgw','r') ~= -1)
        % read size of files
        fid=fopen('stn4.pgw','r');
        [temp,szn4]=fread(fid);fclose(fid);
        fid=fopen('stc4.pgw','r');
        [temp,szc4]=fread(fid);fclose(fid);
        % read files
        dos('arith d stn4.pgw stn4.tmp');
        dos('arith d stc4.pgw stc4.tmp');
        fid=fopen('stn4.tmp','r');stn4=fread(fid)';fclose(fid);
        fid=fopen('stc4.tmp','r');stc4=fread(fid)';fclose(fid);
        k=4;
        %
        in4=stn4(1); istn4=stn4(2:length(stn4));
        ic4=stc4(1); istc4=stc4(2:length(stc4));
        osym4=bit_reverse_prog(istn4,in4);
        osub4=bit_reverse_prog(istc4,ic4);
        out4=total(hd(3)-3,osym4,osub4);
        disp('   progress 4th time')
        out=[out out4];
        iout=dezw2datan(hd,out);
        i_out=iwtp2dbn(iout,'db5'); o4=i_out;figure
        imagesc(i_out);colormap(gray);
        er(X,i_out);
        tsz=tsz+szn4+szc4,%pause
    end
elseif k==4
    if (fopen('stn5.pgw','r') ~= -1) & ...
        (fopen('stc5.pgw','r') ~= -1)
        % read size of files
        fid=fopen('stn5.pgw','r');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        [temp,szn5]=fread(fid);fclose(fid);
        fid=fopen('stc5.pgw','r');
        [temp,szc5]=fread(fid);fclose(fid);
        % read files
        dos('arith d stn5.pgw stn5.tmp');
        dos('arith d stc5.pgw stc5.tmp');
        fid=fopen('stn5.tmp','r');stn5=fread(fid)';fclose(fid);
        fid=fopen('stc5.tmp','r');stc5=fread(fid)';fclose(fid);
        k=5;
        %
        in5=stn5(1); istn5=stn5(2:length(stn5));
        ic5=stc5(1); istc5=stc5(2:length(stc5));
        osym5=bit_reverse_prog(istn5,in5);
        osub5=bit_reverse_prog(istc5,ic5);
        out5=total(hd(3)-4,osym5,osub5);
        disp('   progress 5th time')
        out=[out out5];
        iout=dezw2datan(hd,out);
        i_out=iwtp2dbn(iout,'db5'); o5=i_out;figure
        imagesc(i_out);colormap(gray);
        er(X,i_out);
        tsz=tsz+szn5+szc5,%pause
    end
elseif k==5
    if (fopen('stn6.pgw','r') ~= -1) & …
        (fopen('stc6.pgw','r') ~= -1)
        % read size of files
        fid=fopen('stn6.pgw','r');
        [temp,szn6]=fread(fid);fclose(fid);
        fid=fopen('stc6.pgw','r');
        [temp,szc6]=fread(fid);fclose(fid);
        % read files
        dos('arith d stn6.pgw stn6.tmp');
        dos('arith d stc6.pgw stc6.tmp');
        fid=fopen('stn6.tmp','r');stn6=fread(fid)';fclose(fid);
        fid=fopen('stc6.tmp','r');stc6=fread(fid)';fclose(fid);
        k=6;
        %
        in6=stn6(1); istn6=stn6(2:length(stn6));
        ic6=stc6(1); istc6=stc6(2:length(stc6));
        osym6=bit_reverse_prog(istn6,in6);
        osub6=bit_reverse_prog(istc6,ic6);
        out6=total(hd(3)-5,osym6,osub6);
        disp('   progress 6th time')
        out=[out out6];
        iout=dezw2datan(hd,out);
        i_out=iwtp2dbn(iout,'db5'); o6=i_out;figure
        imagesc(i_out);colormap(gray);
        er(X,i_out);
        tsz=tsz+szn6+szc6,%pause
    end
elseif k==6
    if (fopen('stn7.pgw','r') ~= -1) & …
        (fopen('stc7.pgw','r') ~= -1)
        % read size of files
        fid=fopen('stn7.pgw','r');
        [temp,szn7]=fread(fid);fclose(fid);
        fid=fopen('stc7.pgw','r');
        [temp,szc7]=fread(fid);fclose(fid);
        % read files
        dos('arith d stn7.pgw stn7.tmp');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        dos('arith d stc7.pgw stc7.tmp');
        fid=fopen('stn7.tmp','r');stn7=fread(fid)';fclose(fid);
        fid=fopen('stc7.tmp','r');stc7=fread(fid)';fclose(fid);
        k=7;
        %
        in7=stn7(1); istn7=stn7(2:length(stn7));
        ic7=stc7(1); istc7=stc7(2:length(stc7));
        osym7=bit_reverse_prog(istn7,in7);
        osub7=bit_reverse_prog(istc7,ic7);
        out7=total(hd(3)-6,osym7,osub7);
        disp('   progress 7th time')
        out=[out out7];
        iout=dezw2datan(hd,out);
        i_out=iwtp2dbn(iout,'db5'); o7=i_out;figure
        imagesc(i_out);colormap(gray);
        er(X,i_out);
        tsz=tsz+szn7+szc7,%pause
    end
elseif k==7
    if (fopen('stn8.pgw','r') ~= -1) & …
       (fopen('stc8.pgw','r') ~= -1)
       % read size of files
       fid=fopen('stn8.pgw','r');
       [temp,szn8]=fread(fid);fclose(fid);
       fid=fopen('stc8.pgw','r');
       [temp,szc8]=fread(fid);fclose(fid);
       % read files
       dos('arith d stn8.pgw stn8.tmp');
       dos('arith d stc8.pgw stc8.tmp');
       fid=fopen('stn8.tmp','r');stn8=fread(fid)';fclose(fid);
       fid=fopen('stc8.tmp','r');stc8=fread(fid)';fclose(fid);
       k=8;
       %
       in8=stn8(1); istn8=stn8(2:length(stn8));
       ic8=stc8(1); istc8=stc8(2:length(stc8));
       osym8=bit_reverse_prog(istn8,in8);
       osub8=bit_reverse_prog(istc8,ic8);
       out8=total(hd(3)-7,osym8,osub8);
       disp('   progress 8th time')
       out=[out out8];
       iout=dezw2datan(hd,out);
       i_out=iwtp2dbn(iout,'db5'); o8=i_out;figure
       imagesc(i_out);colormap(gray);
       er(X,i_out);
       tsz=tsz+szn8+szc8,%pause
    end
elseif k==8
    if (fopen('stn9.pgw','r') ~= -1) & …
       (fopen('stc9.pgw','r') ~= -1)
       % read size of files
       fid=fopen('stn9.pgw','r');
       [temp,szn9]=fread(fid);fclose(fid);
       fid=fopen('stc9.pgw','r');
       [temp,szc9]=fread(fid);fclose(fid);
       % read files
       dos('arith d stn9.pgw stn9.tmp');
       dos('arith d stc9.pgw stc9.tmp');
       fid=fopen('stn9.tmp','r');stn9=fread(fid)';fclose(fid);
       fid=fopen('stc9.tmp','r');stc9=fread(fid)';fclose(fid);
       k=9;
       %
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        in9=stn9(1); istn9=stn9(2:length(stn9));
        ic9=stc9(1); istc9=stc9(2:length(stc9));
        osym9=bit_reverse_prog(istn9,in9);
        osub9=bit_reverse_prog(istc9,ic9);
        out9=total(hd(3)-8,osym9,osub9);
        disp('   progress 9th time')
        out=[out out9];
        iout=dezw2datan(hd,out);
        i_out=iwtp2dbn(iout,'db5'); o9=i_out;figure
        imagesc(i_out);colormap(gray);
        er(X,i_out);
        tsz=tsz+szn9+szc9,%pause
    end
elseif k==9
    if (fopen('stn10.pgw','r') ~= -1) & …
        (fopen('stc10.pgw','r') ~= -1)
        % read size of files
        fid=fopen('stn10.pgw','r');
        [temp,szn10]=fread(fid);fclose(fid);
        fid=fopen('stc10.pgw','r');
        [temp,szc10]=fread(fid);fclose(fid);
        % read files
        dos('arith d stn10.pgw stn10.tmp');
        dos('arith d stc10.pgw stc10.tmp');
        fid=fopen('stn10.tmp','r');stn10=fread(fid)';fclose(fid);
        fid=fopen('stc10.tmp','r');stc10=fread(fid)';fclose(fid);
        k=10;
        %
        in10=stn10(1); istn10=stn10(2:length(stn10));
        ic10=stc10(1); istc10=stc10(2:length(stc10));
        osym10=bit_reverse_prog(istn10,in10);
        osub10=bit_reverse_prog(istc10,ic10);
        out10=total(hd(3)-9,osym10,osub10);
        disp('   progress 10th time')
        out=[out out10];
        iout=dezw2datan(hd,out);
        i_out=iwtp2dbn(iout,'db5'); o10=i_out;figure
        imagesc(i_out);colormap(gray);
        er(X,i_out);
        tsz=tsz+szn10+szc10,%pause
    end
elseif k==10
    if (fopen('stn11.pgw','r') ~= -1) & …
        (fopen('stc11.pgw','r') ~= -1)
        % read size of files
        fid=fopen('stn11.pgw','r');
        [temp,szn11]=fread(fid);fclose(fid);
        fid=fopen('stc11.pgw','r');
        [temp,szc11]=fread(fid);fclose(fid);
        % read files
        dos('arith d stn11.pgw stn11.tmp');
        dos('arith d stc11.pgw stc11.tmp');
        fid=fopen('stn11.tmp','r');stn11=fread(fid)';fclose(fid);
        fid=fopen('stc11.tmp','r');stc11=fread(fid)';fclose(fid);
        k=11;
        %
        in11=stn11(1); istn11=stn11(2:length(stn11));
        ic11=stc11(1); istc11=stc11(2:length(stc11));
        osym11=bit_reverse_prog(istn11,in11);
        osub11=bit_reverse_prog(istc11,ic11);
        out11=total(hd(3)-10,osym11,osub11);
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            disp('   progress 11th time')
            out=[out out11];
            iout=dezw2datan(hd,out);
            i_out=iwtp2dbn(iout,'db5'); o11=i_out;figure
            imagesc(i_out);colormap(gray);
            er(X,i_out);
            tsz=tsz+szn11+szc11,%pause
         end
      elseif k==11
         if (fopen('stn12.pgw','r') ~= -1) & ...
            (fopen('stc12.pgw','r') ~= -1)
            % read size of files
            fid=fopen('stn12.pgw','r');
            [temp,szn12]=fread(fid);fclose(fid);
            fid=fopen('stc12.pgw','r');
            [temp,szc12]=fread(fid);fclose(fid);
            % read files
            dos('arith d stn12.pgw stn12.tmp');
            dos('arith d stc12.pgw stc12.tmp');
            fid=fopen('stn12.tmp','r');stn12=fread(fid)';fclose(fid);
            fid=fopen('stc12.tmp','r');stc12=fread(fid)';fclose(fid);
            k=12;
            %
            in12=stn12(1); istn12=stn12(2:length(stn12));
            ic12=stc12(1); istc12=stc12(2:length(stc12));
            osym12=bit_reverse_prog(istn12,in12);
            osub12=bit_reverse_prog(istc12,ic12);
            out12=total(hd(3)-11,osym12,osub12);
            disp('   progress 12th time')
            out=[out out12];
            iout=dezw2datan(hd,out);
            i_out=iwtp2dbn(iout,'db5'); o12=i_out;figure
            imagesc(i_out);colormap(gray);
            er(X,i_out);
            tsz=tsz+szn12+szc12,%pause
         end
      elseif k==12
         disp('COMPLETE FILES')
         stop=1;
      end
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.20

```
/*************************************************************
* file name Arithmetic coding                                *
* Use : arith [e|d] file1 file2                              *
*        e is Encode from file1 to file2.                    *
*        d is Decode from file1 to file2.                    *
* This program base on Implementtations of Arithmetic coding*
*                                                            *
*************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define MAXIMUM_SCALE   16383
#define BUFFER_SIZE 256

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


// Function prototypes.

short int get_current_count( SYMBOL *s );
static char buffer[ BUFFER_SIZE + 2 ];
static char *current_byte;
static int output_mask;
static int input_bytes_left;
static int input_bits_left;
static int past_eof;

/*
 * Eternal variable declarations.
 */
extern long underflow_bits;
extern int max_order;
extern int flushing_enabled;

static unsigned short int code; //Present input code
static unsigned short int low;  // Low current code range
static unsigned short int high; // High current code range

long underflow_bits; // Number of underflow bits pending

short int storage[ 258 ];
short int *totals = storage + 1; // to symbol index.

FILE *infile;
FILE *outfile;

void Encode( void );  // main Encode
void Expand( void );  // main Decode

void initarithdecode( FILE *stream );
void remove_bitstream( FILE *stream, SYMBOL *s );
void initarithencode( void );
void encode_symbol( FILE *stream, SYMBOL *s );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้นำไปดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void flushcode( FILE *stream );

short int input_bit( FILE *stream );
void initialize_output_bitstream( void );
void output_bit( FILE *stream, int bit );
void flushstreambit( FILE *stream );
void initialize_input_bitstream( void );
long bit_ftell_output( FILE *stream );
long bit_ftell_input( FILE *stream );

void initialize_model( void );
void update_model( int symbol );
int int2sym( int symbol, SYMBOL *s );
void get_symbol_scale( SYMBOL *s );
int sym2int( int count, SYMBOL *s );
void add_character_to_model( int c );
void flush_model( void );
// ********************************************************

// Routine for encode
// 1. Initialize model ,I/O ,Arithmetic coder.
// 2. Read char from file to convert to SYMBOL
// 3. Encode
// ********************************************************
void Encode(void)
{
    SYMBOL s;
    int c;

    initialize_model();
    initialize_output_bitstream();
    initarithencode();


    for ( ; ; ) // compression loop.
    {
        c = getc( infile );
                int2sym( c, &s );
                encode_symbol( outfile, &s );
        if ( c == EOF )
                        break;
                update_model( c );
    }

    flushcode( outfile );          //for clean some data in
    flushstreambit( outfile );     //register and memory
    fputc( '\n', stderr );
    fprintf( stderr,"\r %ld compress to %ld ",ftell(infile),
                    bit_ftell_output(outfile)  );
    exit( 0 );
}

// ********************************************************
// To decode .
// to do get binary stram from compression file to decode.

void Expand( void )
{
    SYMBOL s;
    int c,count;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    initialize_model();
    initialize_input_bitstream();
    initarithdecode( infile );
    for ( ; ; )
    {
      get_symbol_scale( &s );
      count = get_current_count( &s );
      c = sym2int( count, &s ); //□□□□ □ □□□□ symbol □□□ □□ □□ index
      remove_bitstream( infile, &s );
      if ( c == EOF )
            break;

             putc( (char) c, outfile );
      update_model( c );
    }
    fputc( '\n', stderr );
    fprintf( stderr,"\r %ld compress to %ld ",
                  bit_ftell_output(infile),ftell(outfile)  );

    exit( 0 );
}


// ********************************************************
void initialize_output_bitstream()
{
    current_byte = buffer;
    *current_byte = 0;
    output_mask = 0x80;
}

// ********************************************************
// for output bit compression use shift bit properties

void 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 flushstreambit( FILE *stream )
{
    fwrite( buffer, 1, (size_t)( current_byte - buffer ) + 1, stream
);
    current_byte = buffer;
}
// ********************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำ ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void initialize_input_bitstream()
{
    input_bits_left = 0;
    input_bytes_left = 1;
    past_eof = 0;
}

// ********************************************************

short int 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 bit_ftell_output( FILE *stream )
{
    long total;

    total = ftell( stream );
    total += current_byte - buffer;
    total += underflow_bits/8;
    return( total );
}

// ********************************************************

long bit_ftell_input( FILE *stream )
{
    return( ftell( stream ) - input_bytes_left + 1 );
}

// ********************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void initarithencode()
{
    low = 0;
    high = 0xffff;
    underflow_bits = 0;
}

// *********************************************************

void 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 flushcode( FILE *stream )
{
    output_bit( stream, low & 0x4000 );
    underflow_bits++;
    while ( underflow_bits-- > 0 )
        output_bit( stream, ~low & 0x4000 );
}

// *********************************************************
short int get_current_count( SYMBOL *s )
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    long range;
    short int count;

    range = (long) ( high - low ) + 1;
    count = (short int)
            ((((long) ( code - low ) + 1 ) * s->scale-1 ) / range );
    return( count );
}

// *******************************************************

void initarithdecode( FILE *stream )
{
    int i;

    code = 0;
    for ( i = 0 ; i < 16 ; i++ )
    {
        code <<= 1;
        code += input_bit( stream );
    }
    low = 0;
    high = 0xffff;
}

// *******************************************************

void remove_bitstream( 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 ) )
        {
        }
       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 initialize_model()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น มีการดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    short int i;

    for ( i = -1 ; i <= 256 ; i++ )
        totals[ i ] = i + 1;
}

// *******************************************************

void 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 int2sym( int c, SYMBOL *s )
{
    s->scale = totals[ 256 ];
    s->low_count = totals[ c ];
    s->high_count = totals[ c+1 ];
    return( 0 );
}

// *******************************************************

void get_symbol_scale( SYMBOL *s )
{
    s->scale = totals[ 256 ];
}

// *******************************************************

int 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 main( int argc, char *argv[] )
{
  char *text;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (argc != 4)
  { printf(" Arithmetic Coding and Decoding \n");
    printf(" useage :  arith [e|d]  <input> <output>\n"
           "                (e = Encode ,  d = Decode) \n");
         exit(-1);
  }

  if ( ( text =argv[1],text[1]||strpbrk(text,"DEde")==NULL)
    ||    ( text =argv[2],(infile=fopen(text,"rb"))==NULL)
    || ( text =argv[3],(outfile=fopen(text,"wb"))== NULL))
  {
    printf(" Error in %s \n",text);
    exit(-1);
  }

  if(toupper(*argv[1])=='E') Encode();
  else Expand();

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.21

```
function menu_wave()
% This is function menu_wave representation of image compression
% using wavelet transform
load menu_wave
a = figure('Units','points', ...
      'Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'MenuBar','none', ...

      'Name','Image compress using wavelet transform', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[30 58.5 188.25 343.5], ...
      'Tag','Fig2');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','img=load_bmp;', ...
      'Position',[20 291 147 33], ...
      'String','Load image', ...
      'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','[hd,isym,size_com,st1,st2,total_file]=…
          enter_ratio(img);', ...
      'Position',[20 223 147 33], ...
      'String','Compress', ...
      'Tag','Pushbutton2');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','saveimage(total_file);', ...
      'Position',[21 85 147 33], ...
      'String','Save result', ...
      'Tag','Pushbutton3');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','iwt=show_result1(hd,isym,st1,st2);', ...
      'Position',[20 154 147 33], ...
      'String','Show result', ...
      'Tag','Pushbutton4');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','close(gcbf)', ...
      'Position',[20 18 147 33], ...
      'String','Quit', ...
      'Tag','Pushbutton5');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.23 0.73 1], ...
      'Position',[9.75 8.25 167.25 326.25], ...
      'Style','frame', ...
      'Tag','Frame1');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.22

```
function InputImg = load_bmp
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is function InputImg = load_bmp
% for load gray_scale images files extension bmp
% InputImg is output files
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
t1 = [];
[t1 t2] = uigetfile('*.bmp'); % function load files of MATLAB V5.1
t3=t1;
if t1 ~= 0
   % check files extension jwt of wavelet transform or
   % extension bmp of input gray-scale images
   if t1(length(t1)-2:length(t1)) == char('jwt')
      t3=t1(1:length(t1)-3); t3=strcat(t3,'tmp');
      % decompress files
      dos(strcat([char('arith d') blanks(1) t1 blanks(1) t3]));
      %%%%%%%%%%%%%%%%%%%%%%
      fid = fopen(t3,'r');
      [data,size_com] = fread(fid); InputImg=data';
      fclose(fid);
      delete(t3);
      %%%%%%%%%% Decompressing %%%%%%%%%%%
      hd=InputImg(1:3);
      st1=InputImg(4:5);
      st2=InputImg(6:7);
      i_jwt=bit_reverse(InputImg(8:length(InputImg)));
      [id,h_dec]=dezw2data(hd,i_jwt);
      iwt=iwtp2dbn(id,'db5');
      close(h_dec);
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      % check size of images files
      if st1(2) ~=0 | st2(2) ~=0
         iwt=iwt(1:(256*st1(1)-st1(2)),1:(256*st2(1)-st2(2)));
      end
      show_result2,imagesc(iwt); colormap(gray);
      title('Load Image files Wavelet transform');
      msgbox( '  Load image file .jwt complete ... ','Gray-scale…
         load');
   elseif t1 ~= 0
      t1 = [t2 t1];
      InputImg = imread(t1, 'bmp');
      if length(size(InputImg)) == 3
         msgbox( '???  Image file is not gray-scale type',...
            'Error gray-scale load');
         InputImg = [];
      else
         InputImg = double(InputImg);
         y=size(InputImg);
         show_image % call show_image M-files
         imagesc(InputImg),colormap(gray);
         text_h=text(-1,0,strcat(upper(t3(1:length(t3)-4)),...
            ' ( original size',num2str(y(1)),'x',num2str(y(2)),...
            ' pixels )'));
         set(gca,'Title',text_h); axis off
         msgbox( '  Load image file complete ...  ','Gray-scale…
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            load');
      end
   end
else
   InputImg=0;
End
```

PROGRAM NO.23

```
function [hd,isym,size_com,st1,st2,total_file]=enter_ratio(img)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is function enter compress ratio
% img         is image for transform
% hd          is header file
% isym        is inverse symbols 8 bit to character
% size_com    is size compress
% st1         is matrix of size of image
% st2         is matrix of size of image
% sym         is symbols for compress
% total_file is [hd,st1,st2,sym]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
size_image=size(img);
% check size of image
if (log2(size_image(1)) ~= round(log2(size_image(1)))) & ...
      (log2(size_image(2)) ~= round(log2(size_image(2))))
   m=ceil(log2(max(size_image))); g=zeros(2^m);
   g(1:size_image(1),1:size_image(2))=g(1:size_image(1),1:…
      size_image(2))+img;
   img = g;
   st1=ceil(size_image(1)/256); st1=[st1,256*st1-size_image(1)];
   st2=ceil(size_image(2)/256); st2=[st2,256*st2-size_image(2)];
else
   st1=[size_image(1),0];st2=[size_image(2),0];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% menu for enter compress ratio
prompt={'Enter compress ratio:'};
def={'20'};
title='Input for compression ratio';
lineNo=1;
answer=str2num(char(inputdlg(prompt,title,lineNo,def)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if answer > 1
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   size_define=size_image(1)*size_image(2)/answer;
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % wavelet transform
   wt=wtp2dbn(img,'db5');
   [hd,sym,h_com]=ezw2symbol13(wt,size_define);
   sym=cut_size(sym,size_define); total_file=[hd,st1,st2,sym];
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %%%%%%%%%%%%%% read and write %%%%%%%%%%%%%%%%%
   fid = fopen('ftemp.txt','w'); % ftemp is file temporary
   fwrite(fid,total_file,'int8');
   fclose(fid);
   dos('arith e ftemp.txt ftemp.jwt');
   fid = fopen('ftemp.jwt','r');
   [data,size_com] = fread(fid);
   fclose(fid);
   delete('ftemp.txt');delete('ftemp.jwt');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    isym=bit_reverse(sym);
    close(h_com);
else
    handle_er=errordlg('Plase put new ratio again','Error','replace');
    hd=[]; isym=[]; size_com=[]; st1=[]; st2=[];
end
```

PROGRAM NO.24

```
function iwt=show_result1(hd,isym,st1,st2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is function show result by
% hd          is header file
% isym        is inverse symbols 8 bit to character
% st1         is matrix of size of image
% st2         is matrix of size of image
% iwt         is output image reconstructed
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% Decompressing %%%%%%%%%%%%
[id,h_dec]=dezw2data(hd,isym);
iwt=iwtp2dbn(id,'db5');
close(h_dec);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% check size of image
if st1(2) ~=0 | st2(2) ~=0
   iwt=iwt(1:(256*st1(1)-st1(2)),1:(256*st2(1)-st2(2)));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
show_result % call show_result M-file
imagesc(iwt); axis off; colormap(gray);
title('Image file wavelet transform');
```

PROGRAM NO.25

```
function saveimage(out)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is function save image files
% out  is file for save
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%function save files of MATLAB V5.1
[t1 t2]=uiputfile('*.jwt','Save as');
t3=t1(1:length(t1)-3); t3=strcat(t3,'txt');
fid = fopen(t3,'w');
fwrite(fid,out,'int8');
fclose(fid);
% compress files
dos(strcat([char('arith e') blanks(1) t3 blanks(1) t1]));
%%%%%%%%%%%%%%%%%%%%
delete(t3);
msgbox('    Save as file complete', 'Save complete');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.26

```
function show_image()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is function show image files
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load show_image
a = figure('Units','points', ...
      'Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'MenuBar','none', ...
      'Name','Show image', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[89.25 89.25 420 315], ...
      'Tag','Fig1');
```

PROGRAM NO.27

```
function show_result()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is function show result of images
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load show_result
a = figure('Units','points', ...
      'Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'MenuBar','none', ...
      'Name','Show result', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[89.25 89.25 420 315], ...
      'Tag','Fig1');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'Callback','detail_result(img,iwt,size_com);', ...
      'Position',[9.75 5.25 65.25 19.5], ...
      'String','Result', ...
      'Tag','Pushbutton1');
```

PROGRAM NO.28

```
function detail_result(img,iwt,size_com)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is function detail result by
% img         is image for transform
% iwt         is output image reconstructed
% size_com    is size compress
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mse=err(img,iwt); % calculate error
mse=strcat('Mean square error (MSE) = ',num2str(mse));
y=size(img); bpp=size_com*8/(y(1)*y(2));
bpp=strcat('Bit per pixel (bpp) = ',num2str(bpp),' bpp');
o_size=size_com;
o_size=strcat('Size compress = ',num2str(o_size),' byte');
%%%%% menu output display
Data=1:64;Data=(Data'*Data)/64;
h=msgbox(str2mat(mse,bpp,o_size),...
'Detail result','custom',Data,hot(64));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM NO.29

```
function [header,symbols,h_com]=ezw2symbol(c,size_define)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%            EZW2SYMBOL  IS  EZW ENCODING TO SYMBOLS
%                     c  IS  FILE FOR ENCODE
%                 H_COM  IS  HANDLE COMPRESS
%                HEADER  IS  HEADER FILE OF ENCODE
%               SYMBOLS  IS  SYMBOLS OF CODE
%          SIZE_DEFINE  IS  SIZE FOR COMPRESS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dc=[]; ct=c; y=size(c); Nd=13; size_compress=0;
lv=log2(y(1));
hi_bi=ceil(log2(abs(max(c(:))))); t0=2^(hi_bi)/2; ht0=log2(t0);
mid=(2^(hi_bi)+t0)/2;
header=[lv,ht0,Nd]; % set header file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h_com = waitbar(0,'Please wait... Now compressing files');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for d=1:Nd
   % computation waitbar here %
   waitbar(d/Nd)
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%
   t=t0/2^(d-1); half_t=mid/2^(d-1); Ns=Nd+1-d; d=ones(y(1));
   k3=find( ct<=2*t & ct>t );
   k4=find( ct>=-2*t & ct<-t );
   k5=find( abs(c) > 2*t );
   d(k3)=65; % ' POS '
   d(k4)=66; % ' NEG '
   d(k5)=0; ct(k5)=0; % ' for skip '
   for j = 3:lv
      if j == 3
         for n = 1:4:5                    % n=1:2^(j-1):2^(j-1)+1
            for m = 1:4:5                 % m=1:2^(j-1):2^(j-1)+1
               for u = n:2:n+2            % u=n:2:n+2^(j-1)-2
                  for v = m:2:m+2         % v=m:2:m+2^(j-1)-2
                     for row = u:u+1
                        for col = v:v+1
                           if size_compress < size_define
                              td=d(row,col);
                              ab_ct=abs(ct(row,col));
                              rcmax=max([row col]);
                              if td == 0
                              else
                                 if td > 64
                                    if td == 65
                                       dc=[dc 0 1]; % POS = 01
                                    else % td == 66
                                       dc=[dc 1 0]; % NEG = 10
                                    end
                                    ds=1; half=half_t;
                                    while ds <= Ns
                                       if ab_ct > half
                                          dc=[dc 1];
                                          half=half + t/2^(ds+1);
                                       else % if ab_ct < half
                                          dc=[dc 0];
                                          half=half - t/2^(ds+1);
                                       end
                                       ds=ds+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    end
                else % td == 1
                    ls=lv; r=1; IZ=0;
                    while ls > 1
                        if rcmax <= 2^(r)
                            row1=2^(ls-1)*row-...
                                (2^(ls-1)-1):2^...
                                (ls-1)*row;col1...
                                =2^(ls-1)*col-(...
                                2^(ls-1)-1):2^(...
                                ls-1)*col;
                            ab_c=abs(ct(row1,col1));
                            if max(ab_c(:)) > t
                                IZ=1; ls=1;
                            end
                        else
                            r=r+1;
                        end
                        ls=ls-1;
                    end
                    if IZ == 1
                        dc=[dc 1 1]; % IZ = 11
                    else % IZ == 0
                        dc=[dc 0 0]; % ZTR = 00
                        if rcmax > 2^(lv-1)
                        else
                            ls=lv; r=1;
                            while ls >1
                                if rcmax <= 2^(r)
                                row1=2^(ls-1)*...
                                    row-(2^(ls-...
                                    1)-1):2^(ls...
                                    -1)*row;
                                col1=2^(ls-1)*...
                                    col-(2^(ls-...
                                    1)-1):2^(ls...
                                    -1)*col;
                                d(row1,col1) = 0;
                            else
                                r=r+1;
                            end
                            ls=ls-1;
                        end
                    end
                end
            end
            % check size of image file
            data=bit_shift(dc);
            l_data=length(data);
            size_compress=...
                entropy(data)*l_data/8;
        end
    end
end
end
end
end
end
end
elseif (j >3)
for n=1:2^(j-1):2^(j-1)+1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for m=2^(j-1)+1:-2^(j-1):1
   if n ==1 & m ==1
      n=2^(j-1)+1;
   elseif n ==(2^(j-1)+1) & m ==1
      n=-1; m=-1; q=j; j=2;
   end
   for u=n:2:2^(j-1)+n-2
      for v=m:2:2^(j-1)+m-2
         for row = u:u+1
            for col = v:v+1
               if (row <=0) & (col <=0)
                  j=q;
               else
                  if size_compress < size_define
                     td=d(row,col);
                     ab_ct=abs(ct(row,col));
                     rcmax=max([row col]);
                     if td == 0;
                     else
                        if td > 64
                        if td == 65
                           dc=[dc 0 1]; % POS = 01
                        else % td == 66
                           dc=[dc 1 0]; % NEG = 10
                        end
                        ds=1; half=half_t;
                        while ds <= Ns
                           if ab_ct > half
                              dc=[dc 1];
                              half=half + t/2^(ds+1);
                           else % if ab_ct < half
                              dc=[dc 0];
                              half=half - t/2^(ds+1);
                           end
                           ds=ds+1;
                        end
                     else % td == 1
                        ls=lv; r=1; IZ=0;
                        while ls > 1
                           if rcmax <= 2^(r)
                              row1=2^(ls-1)*row-...
                                 (2^(ls-1)-1):2^...
                                 (ls-1)*row;
                              col1=2^(ls-1)*col-...
                                 (2^(ls-1)-1):2^...
                                 (ls-1)*col;
                              ab_c=abs(ct(row1,col1));
                              if max(ab_c(:)) > t
                                 IZ=1; ls=1;
                              end
                           else
                              r=r+1;
                           end
                           ls=ls-1;
                        end
                        if IZ == 1
                           dc=[dc 1 1]; % IZ = 11
                        else % IZ == 0
                           dc=[dc 0 0]; % ZTR = 00
                           if rcmax > 2^(lv-1)
                           else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        ls=lv; r=1;
                                        while ls >1
                                           if rcmax <= 2^(r)
                                              row1=2^(ls-1)*...
                                                 row-(2^(ls-...
                                                 1)-1):2^(ls...
                                                 -1)*row;
                                              col1=2^(ls-1)*...
                                                 col-(2^(ls-...
                                                 1)-1):2^(ls...
                                                 -1)*col;
                                              d(row1,col1) = 0;
                                           else
                                              r=r+1;
                                           end
                                           ls=ls-1;
                                        end
                                     end
                                  end
                                  end
                                  % check size of image file
                                  data=bit_shift(dc);
                                  l_data=length(data);
                                  size_compress=...
                                    entropy(data)*l_data/8;
                               end
                            end
                         end
                      end
                   end
                end
             end
          end
       end
    end
end
symbols=data;
```

PROGRAM NO.30

```
function [g,h_dec]=dezw2data(header,symbols)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    DEZW2DATA  IS  EZW DECODING TO DATA
%                             g  IS  DATA OUT
%                        HEADER  IS  HEADER FOR DECODE
%                       SYMBOLS  IS  SYMBOLS FOR DECODE
%                         H_DEC  IS  HANDLE DECOMPRESS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x=1; t0=2^header(2); mid=(2*t0+t0)/2;
C=zeros(2^(header(1))); Nd=header(3);
y=length(symbols); yt=y; y1=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h_dec = waitbar(0,'Please wait... Now decompressing files');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for d =1:Nd
   % computation waitbar here %
   waitbar(d/Nd)
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
t=t0/2^(d-1); half_t=mid/2^(d-1); Ns=Nd+1-d;
for j =3:header(1)
   if j == 3
      for n = 1:4:5                % n=1:2^(j-1):2^(j-1)+1
         for m = 1:4:5             % m=1:2^(j-1):2^(j-1)+1
            for u = n:2:n+2        % u=n:2:n+2^(j-1)-2
               for v = m:2:m+2     % v=m:2:m+2^(j-1)-2
                  for row = u:u+1
                     for col = v:v+1
                        if abs(C(row,col)) >0
                        elseif (x <=y) & (yt-2 >=0)
                           half=half_t;
                           if symbols(x:x+1)==[0 1];%POS
                              x=x+2; ds=1;
                              % check size of file
                              y1=yt-2-Ns;
                              if y1 >=0
                                 yt=y1;
                              else
                                 ds=Ns+1; yt=y1;
                              end
                              %%%%%%%%%%%%%%%%%%%%
                              while ds <= Ns
                                 if symbols(x) == 1
                                    half=half + t/2^(ds+1);
                                    x=x+1;
                                 else % if symbols(x) == 0
                                    half=half - t/2^(ds+1);
                                    x=x+1;
                                 end
                                 ds=ds+1;
                              end
                              C(row,col)= half ;
                           elseif symbols(x:x+1)==[1 0];%NEG
                              x=x+2; ds=1;
                              % check size of file
                              y1=yt-2-Ns;
                              if y1 >=0
                                 yt=y1;
                              else
                                 ds=Ns+1; yt=y1;
                              end
                              %%%%%%%%%%%%%%%%%%%%
                              while ds <= Ns
                                 if symbols(x) == 1
                                    half=half + t/2^(ds+1);
                                    x=x+1;
                                 else % if symbols(x) == 0
                                    half=half - t/2^(ds+1);
                                    x=x+1;
                                 end
                                 ds=ds+1;
                              end
                              C(row,col)= -half ;
                           elseif symbols(x:x+1)==[1 1];%IZ
                              x=x+2; yt=yt-2;
                           elseif symbols(x:x+1)==[0 0];%ZTR
                              x=x+2; yt=yt-2; ls=header(1); r=1;
                              while ls > 1
                                 if max([row col]) <= 2^(r)
                                 for h=(2^(ls-1))*row-(2^...
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    (ls-1)-1): (2^(ls-...
                                    1))*row
                                 for g=(2^(ls-1))*col-...
                                       (2^(ls-1)-1): (...
                                       2^(ls-1))*col
                                    if abs(C(h,g)) >0
                                    else
                                       C(h,g) =1;
                                    end
                                 end
                              end
                           else
                              r=r+1;
                           end
                           ls=ls-1;
                        end
                     end
                  end
               end
            end
         end
      end
   end
elseif (j > 3)
   for n = 1:2^(j-1):2^(j-1)+1
      for m = 2^(j-1)+1:-(2^(j-1)):1
         if n ==1 & m ==1
            n=2^(j-1)+1;
         elseif n ==(2^(j-1)+1) & m ==1
            n=-1; m=-1; q=j; j=2;
         end
         for u = n:2:2^(j-1)+n-2
            for v = m:2:2^(j-1)+m-2
               for row = u:u+1
                  for col = v:v+1
                     if (row <=0) & (col <=0)
                        j=q;
                     else
                        if abs(C(row,col)) >0
                        elseif (x <=y) & (yt-2 >=0)
                           half=half_t;
                           if symbols(x:x+1)==[0 1];%POS
                              x=x+2; ds=1;
                              % check size of file
                              y1=yt-2-Ns;
                              if y1 >=0
                                 yt=y1;
                              else
                                 ds=Ns+1; yt=y1;
                              end
                              %%%%%%%%%%%%%%%%%%%%%
                              while ds <= Ns
                                 if symbols(x) == 1
                                    half=half + t/2^(ds+1);
                                    x=x+1;
                                 else % if symbols(x) == 0
                                    half=half - t/2^(ds+1);
                                    x=x+1;
                                 end
                                 ds=ds+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                end
                C(row,col)= half ;
            elseif symbols(x:x+1)==[1 0];%NEG
                x=x+2; ds=1;
                % check size of file
                y1=yt-2-Ns;
                if y1 >=0
                   yt=y1;
                else
                   ds=Ns+1; yt=y1;
                end
                %%%%%%%%%%%%%%%%%%%%%
                while ds <= Ns
                   if symbols(x) == 1
                      half=half + t/2^(ds+1);
                      x=x+1;
                   else % if symbols(x) == 0
                      half=half - t/2^(ds+1);
                      x=x+1;
                   end
                   ds=ds+1;
                end
                C(row,col)= -half ;
            elseif symbols(x:x+1)==[1 1];%IZ
                x=x+2; yt=yt-2;
            elseif symbols(x:x+1)==[0 0];%ZTR
                x=x+2; yt=yt-2; ls=header(1);r=1;
                while ls > 1
                   if max([row col]) <= 2^(r)
                      for h=(2^(ls-1))*row-(2^...
                         (ls-1)-1):( 2^(ls-...
                         1))*row
                         for g=(2^(ls-1))*col-...
                            (2^(ls-1)-1) :(...
                            2^(ls-1))*col
                            if abs(C(h,g)) >0
                            else
                               C(h,g) =1;
                            end
                         end
                      end
                   else
                      r=r+1;
                   end
                   ls=ls-1;
                end
             end
          end
       end
    end
 end
 end
 end
 end
 end
 end
 end
 end
 k=find( C == 1 ); C(k)=0;
end
g=C;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถที่จะสำเร็จลุล่วงไปด้วยดีได้นั้น ต้องขอขอบพระคุณหลายๆ ฝ่ายที่คอยให้ความสนับสนุน และช่วยเหลือตลอดมา โดยเฉพาะอย่างยิ่งพระคุณของบิดามารดา ที่เป็นผู้ซึ่งให้โอกาสและให้การสนับสนุนอย่างไม่มีคำว่าย่อท้อ และที่จะขาดมิได้เลยคือ ท่านอาจารย์ที่ปรึกษา ดร.สุทธิชัย นพนาคีพงษ์ และท่านอาจารย์กฤษณ์ วงรุจิระ ที่คอยทุ่มเทเอาใจใส่ดูและ คอยให้คำปรึกษาแก่คณะผู้จัดทำมาโดยตลอด อย่างไม่มีคำว่าย่อท้อและเหน็ดเหนื่อยเช่นกัน ตลอดจนการจัดหาเครื่องมือเครื่องใช้ต่างๆ ที่จำเป็นให้ และอาจารย์อีกท่านหนึ่งที่คณะผู้จัดทำจะไม่กล่าวขอบพระคุณไว้ ณ ที่นี้เสียมิได้ อาจารย์นภัทร สระเอี่ยม ที่ให้คำแนะนำ ให้กำลังใจในการทำงาน และเปิดโอกาสให้คณะผู้จัดทำได้สามารถจัดทำปริญญานิพนธ์ฉบับนี้ได้อย่างเต็มที่

ทางคณะผู้จัดทำขอกราบขอบพระคุณไว้ ณ โอกาสนี้ด้วย

คณะผู้จัดทำ

นายจิตติ กลิ่นประภัศร

นายธราพงษ์ เพียซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ชัญชาญ มัคคุ้น., ***การแก้ไขลักษณะของลายเส้น โดยใช้เวฟเล็ตทรานส์ฟอร์ม***, วิทยานิพนธ์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง, พ.ศ. 2539

[2] ชินภัทร นันทจิวากรชัย,รศ.ดร. มนัส สังวรศิลป์, ดร.สุรพันธุ์ เอื้อไพบูลย์., ***"การลดขนาดข้อมูลภาพโดยใช้การแปลงเวฟเล็ต.,"*** วิศวสารลาดกระบัง ปีที่ 13 ฉบับที่2,หน้า 28-35,เมษายน พ.ศ. 2540

[3] C. Sidncy Burrus, Ramesh A. Gopinath, and Maitao Guo. ***Introduction to Wavelets and Wavelet Transforms***.,Prentice-Hall,1998

[4] Jerome M. Shapiro. "***Embendded Image coding using zerotree of wavelet coefficients***.," IEEE Trans. signal proc.,VOL.41 ,NO.12.pp3445-3462,DEC 1993.

[5] Shie Qian, Dapang Chen., ***Joint Time-Frequency Analysis Methods and applications***., NJ:Prentice-Hall,1996

[6] Charles K. Chui., ***An Introduction to Wavelets*** ., ACADEMIC PRESS .,1992

[7] Eric J. Stollnitz, Tony D. DeRose, David H. Salesin., "***Wavelets for Computer Graphics: A Primer, Part 1,***".IEEE Computer Grap. and App.,pp76-84.,MAY 1995

[8] Martin Vetterli, Cormac Herley., ***"Wavelets and Filter Banks : Theory and Design,".*** IEEE Trans. Signal proc.,VOL.40 ,NO.9. pp2207-2232,SEP 1992.

[9] Tilo Strutz, Heiko Schwarz, Erika Muller., ***"Predictive Image Coding With Adaptive Wavelet Transform.,"*** Proce. Of SPIE, VOL.3164, San Diego, CA, USA, JUY 1997

[10] Brani Vidakovic, Peter Muller., ***"WAVELETS FOR KIDS A Tutorial Introduction.,"*** Duke University, 1991

[11] Rafael C.Gonzalez,Richard E.Woods, ***Digital Image Processing***.,Addison-Wesley Publishing.,1992

[12] Ning Lu.,***Fractal Imaging***.,ACADEMIC PRESS.,1997

[13] Colm Mulcahy, "**Image compression using the Haar wavelet transform,**"Spelman Science and Math Journal.

[14] Jie Liang, ***"Highly Scalable Image Coding for Multimedia Applications.,"*** Media Technologies Lab, Digital Signal Processing R&D Center., Texas Instruments,Inc. (liang@ti.com : http://www.acm.org/sigmm/MM97/papers/liang/acm97.html#32)

[15] *Li-San Wang,* ***"Design and Implementation of an Adaptive Modeling Unit for Multialphabet Arithmetic Compression.,"*** National Taiwan University. (lswang@analog.ee.ntu.edu.tw : *http://dodger.ee.ntu.edu.tw/~lswang/arith/index.html*)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้