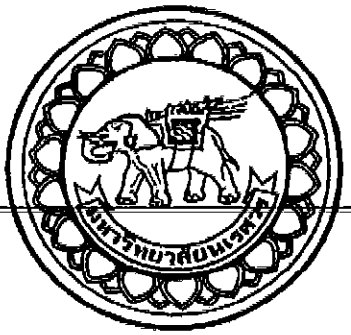


การบีบอัดข้อมูลภาพโดยการใช้วิธีซีโรทรี

IMAGE COMPRESSION BY USING ZEROTREE

นายวัชสัณห์ วรรณสอน รหัส 44370369

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25/ W.A. 2553/
เลขทะเบียน ร501654
เลขเรียกหนังสือ
มหาวิทยาลัยนเรศวร

ว
ว 384ก
2547

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2547

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	**การบีบอัดข้อมูลภาพโดยวิธีซีโรทรี**
ผู้เสนอโครงงาน	นายวัชสัณห์ วรรณสอน รหัส 44370369
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2547

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

.....กรรมการ

(ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ

(นายแสงชัย มังกรทอง)

หัวข้อโครงงาน	การบีบอัดข้อมูลภาพโดยวิธีซีโรทรี
ผู้ดำเนินโครงงาน	นายวัชสัณห์ วรรณสอน รหัส 44370369
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มแม่น
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2547

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและพัฒนาด้วยโปรแกรมแมทแล็ป สำหรับการบีบอัดข้อมูลภาพ 2 มิติ โดยใช้วิธีการแปลงรหัสแบบซีโรทรี ผลที่ได้จากโครงงานนี้ จะได้อัตราการบีบอัดข้อมูลภาพสูงสุด มีค่าเท่ากับ คือ 1.770 นอกจากนี้ค่า SNR และ PSNR มีค่าเข้าใกล้เลขอนันต์ทั้งสองกรณี นั้นคือการคลายการบีบอัดข้อมูลสมบูรณ์แบบเมื่อนำมาเปรียบเทียบกับข้อมูลต้นแบบของทั้งสองกรณี

Project Title	Image compression by using Zerotree
Name	Mr.Watchason Wannasorn ID 44370369
Project Advisor	Assistant Professor Suchart Yammen , Ph. D.
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic Year	2004

ABSTRACT

This project is to study and develop a Matlab program for compressing two-dimensional data by using the Zerotree coding method. Form the project results, the maximum compression ratio is 1.77 for the two-dimensional data. Furthermore, both the SNR and PSNR values approach infinity; that is, the decompressed data is perfectly recovered when compared with the original data in the two cases.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ สำเร็จได้ด้วยดีก็เนื่องด้วยความอนุเคราะห์จากอาจารย์ที่ปรึกษาโครงงานคือ ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น ที่คอยให้ความช่วยเหลือในทุก ๆ ด้าน ในโอกาสนี้ทางคณะผู้จัดทำโครงงานจึงขอขอบคุณทุก ๆ ท่านที่มีส่วนช่วยทำให้โครงงานนี้ประสบความสำเร็จได้ด้วยดี

นายวัชสัณห์ วรรณสอน

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช

บทที่ 1 บทนำ
1.1 หลักการและเหตุผล 1
1.2 วัตถุประสงค์ 1
1.3 ขอบข่ายของโครงงาน 1
1.4 ขั้นตอนการดำเนินงาน 2
1.5 ผลที่คาดว่าจะได้รับ 2
1.6 งบประมาณ 2

บทที่ 2 หลักการและทฤษฎี
2.1 การบีบอัดข้อมูล 3
2.2 ทฤษฎีเวฟเล็ต 3
2.2.1 พื้นฐานของการแปลงเวฟเล็ต 3
2.2.2 การวิเคราะห์สัญญาณแบบหลายระดับความละเอียด 6
2.2.3 การแปลงเวฟเล็ต 10
2.2.4 การแปลงกลับเวฟเล็ต 12
2.2.5 การประยุกต์ใช้กับข้อมูลภาพ 13
2.2.6 การประยุกต์ใช้เวฟเล็ตในการลดขนาดข้อมูล 14
2.3 การหาค่าความผิดพลาด (SNR และPSNR)
และการหาค่าอัตราส่วนการบีบอัดข้อมูล 15

สารบัญ (ต่อ)

หน้า

2.4 การเข้ารหัสแบบซีโรทรี 15
2.5 กระบวนการเข้ารหัสแบบซีโรทรี 16

บทที่3 วิธีการดำเนินโครงงาน
3.1 ศึกษาอัลกอริทึมการเข้ารหัสแบบซีโรทรีและขั้นตอนการออกแบบ 23
3.2 โปรแกรมการสร้างแผนผังรหัสซีโรทรี 25
3.3 โปรแกรมเข้ารหัสซีโรทรี 26
3.4 โปรแกรมการถอดรหัสซีโรทรี 26

บทที่ 4 การทดลอง
4.1 บทนำ 27
4.2 การเรียกใช้โปรแกรม 27
4.3 การทดลองเข้ารหัสและการถอดรหัสซีโรทรี 28
4.3.1 ผลการทดลองการบีบอัดข้อมูลภาพ 28

บทที่ 5 บทสรุป
5.1 สรุปผลการทดลอง 33

เอกสารอ้างอิง 34
ภาคผนวก 35
ประวัติผู้เขียนโครงงาน 51

สารบัญตาราง

ตารางที่		หน้า
2.1	ขั้นตอนการเข้ารหัสซีโรทรีขั้นตอนที่ 1	19
2.2	ขั้นตอนการเข้ารหัสซีโรทรีขั้นตอนที่ 2	21
2.3	ขั้นตอนการเข้ารหัสซีโรทรีขั้นตอนที่ 3	22
4.1	สรุปผลที่ได้จากการบีบอัดข้อมูลภาพโดยวิธีการแบบซีโรทรีทั้ง 4 กรณี	32

สารบัญรูป

รูปที่		หน้า
2.1	ลักษณะของเวฟเล็ตแม่ชนิด Daubechies20	4
2.2	แสดงลักษณะของเวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่งไปที่ค่า a,b ต่างๆกัน	5
2.3	ลักษณะของการแตกกระจายสัญญาณและการรวมกลับสัญญาณของเวฟเล็ต	5
2.4	ลักษณะการกระจาย (Decomposition) ของสเปซของเวกเตอร์	9
2.5	ลักษณะการกระจาย (Decomposition) สัญญาณไปยังระดับความละเอียดต่างๆ	10
2.6	Two-channel filter banks	11
2.7	Reconstruction Two-channel filter banks	11
2.8	ลักษณะของ DWT โดยใช้ Dyadic tree structure (Octave filter banks)	12
2.9	ลักษณะของ Reconstruction Two-channel filter banks	12
2.10	ลักษณะของ IDWT โดยการใช้ dyadic tree structure	12
2.11	แสดงการแปลงเวฟเล็ตของข้อมูลภาพออกเป็น 4 ส่วน คือ AGG1 aGH1	14
2.12	รูปแสดงขั้นตอนการลดขนาดข้อมูล	14
2.13	รูปแสดงขั้นตอนการทำ dominant pass	17
3.1	ขั้นตอนการดำเนินโครงการ	23
3.2	Flowchart โปรแกรมการบีบอัดข้อมูลภาพแบบซีโรทรี	24
3.3	Flowchart ฟังก์ชันสร้างแผนผังรหัสซีโรทรี	25
3.4	แผนภาพโปรแกรมการเข้ารหัสซีโรทรี	26
3.5	แผนภาพโปรแกรมถอดรหัสซีโรทรี	26
4.1	Flowchart โปรแกรมการบีบอัดข้อมูลภาพโดยวิธีการซีโรทรี	27
4.2	แสดงภาพ Lena ก่อนการบีบอัดและหลังการบีบอัดข้อมูลภาพด้วยซีโรทรี	28
4.3	แสดงภาพ Nemo ก่อนการบีบอัดและหลังการบีบอัดข้อมูลภาพด้วยซีโรทรี	29
4.4	แสดงภาพ dog ก่อนการบีบอัดและหลังการบีบอัดข้อมูลภาพด้วยซีโรทรี	30
4.5	แสดงภาพ Eye ก่อนการบีบอัดและหลังการบีบอัดข้อมูลภาพด้วยซีโรทรี	31

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

ในปัจจุบันการใช้คอมพิวเตอร์ถูกใช้ในการส่งข่าวสารกันมากขึ้น ซึ่งข้อมูลที่ถูกส่งจากเครื่องคอมพิวเตอร์ต้นทางไปยังเครื่องปลายทางหากมีขนาดข้อมูลที่ใหญ่แล้ว ก็จะทำให้การส่งข้อมูลเกิดความล่าช้า หากเป็นข้อมูลที่ต้องการเร่งด่วนแล้วก็จะทำให้เกิดผลเสียขึ้น อย่างเช่น ข้อมูลภาพของฟิล์ม x – ray ที่ใช้ในทางการแพทย์ ที่จำเป็นต้องมีข้อมูลที่ถูกต้องและแม่นยำ มีคุณภาพของภาพเหมือนกับภาพต้นแบบ หรือสูญเสียคุณสมบัติน้อยที่สุด

การแก้ปัญหาดังกล่าวสามารถทำได้โดยการบีบอัดข้อมูลภาพเพื่อลดขนาดของข้อมูลลง แต่ยังคงคุณสมบัติของภาพต้นฉบับไว้ โดยการเก็บไว้ในรูปของไฟล์ไบนารี่ ที่จะมีขนาดเล็กกว่าไฟล์ต้นฉบับ นอกจากนี้ขนาดข้อมูลที่เล็กลงยังเป็นการช่วยประหยัดเนื้อที่ของอุปกรณ์เก็บข้อมูลอีกด้วย

ในการบีบอัดข้อมูลภาพที่ดีนั้นต้องใช้อัตราส่วนการบีบอัด(Compression ratio) ที่เหมาะสมที่สุดเพื่อให้ภาพที่ได้มีความสมบูรณ์ของข้อมูลที่ต้องการ การบีบอัดข้อมูลรูปภาพนั้น วิธีการเข้ารหัสแบบซีโรทรี(EZW)เป็นทางเลือกหนึ่งที่ใช้ในการบีบอัดรูปภาพ โดยไม่ทำให้ภาพที่ผ่านการบีบอัดมีคุณภาพที่เปลี่ยนไป ขนาดของข้อมูลเล็กลง แต่ยังคงรักษาคุณภาพของภาพไว้ใกล้เคียงกับภาพเดิม การเข้ารหัสแบบซีโรทรีนี้ปัจจุบันมีการพัฒนาเป็นโปรแกรมในภาษาต่างๆมากมาย แต่ในโปรแกรม Matlab นั้นมีน้อยมากซึ่งหากมีการนำมาพัฒนาก็จะเป็นประโยชน์กับผู้สึกษาต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการเข้ารหัสด้วยวิธีซีโรทรี
2. เพื่อศึกษาการบีบอัดข้อมูลภาพด้วยวิธีเวฟเล็ตโดยใช้หลักการของซีโรทรี

1.3 ขอบข่ายของโครงงาน

1. ศึกษาวิธีการบีบอัดข้อมูลภาพโดยใช้วิธีซีโรทรี
2. ศึกษาการเขียนโปรแกรมที่เกี่ยวข้องกับการบีบอัดข้อมูลภาพโดยใช้ MATLAB

1.4 ขั้นตอนการดำเนินงาน

	2546		2547									
	พย	ธค	มค	กพ	มีค	เมย	พค	มิย	กค	สค	กย	ตค
ศึกษาค้นคว้าในเรื่องการบีบอัดข้อมูล	←					→						
ศึกษาค้นคว้าการใช้โปรแกรม Matlab				←		→						
ออกแบบการเขียนโปรแกรม						←		→				
พัฒนาทดสอบและแก้ไข								←		→		
สรุปและจัดทำรายงาน											←	→

1.5 ผลที่คาดว่าจะได้รับ

1. ลดจำนวนบิตในการจัดเก็บของข้อมูลภาพ โดยรักษาคุณภาพของข้อมูลภาพให้เหมือนภาพต้นแบบ
2. นำไปประยุกต์ใช้ในการส่งข้อมูลภาพได้เร็วขึ้น

1.6 งบประมาณ

1. ค่าวัสดุอุปกรณ์	500	บาท
2. ค่าถ่ายเอกสาร	500	บาท
รวมทั้งสิ้น	**1,000**	**บาท (หนึ่งพันบาทถ้วน)**

(หมายเหตุ) ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎี

2.1 การบีบอัดข้อมูลภาพ (Image Compression)

หลักในการบีบอัดข้อมูลภาพคือพยายามลดหรือกำจัดส่วนของข้อมูลที่เกินความจำเป็นหรือซ้ำซ้อนกัน (Data redundancy) โดยยังคงข่าวสารไว้เหมือนเดิม ซึ่งจะทำให้ข้อมูลภาพมีขนาดลดลงจากเดิมและสามารถนำภาพกลับมาแสดงภายหลังโดยผ่านกระบวนการคลาย (Decompression)

ประเภทของการบีบอัดข้อมูลภาพแบ่งได้เป็น 2 แบบคือ

2.1.1 การบีบอัดข้อมูลแบบไม่มีการสูญเสีย (Lossless compression หรือ Bit-preserving หรือ Versible compression)

2.1.2 การบีบอัดข้อมูลแบบมีการส่วนเสียบางส่วน (Lossy Compression หรือ Irreversible compression)

2.2 ทฤษฎีเวฟเล็ต (Wavelet transform)

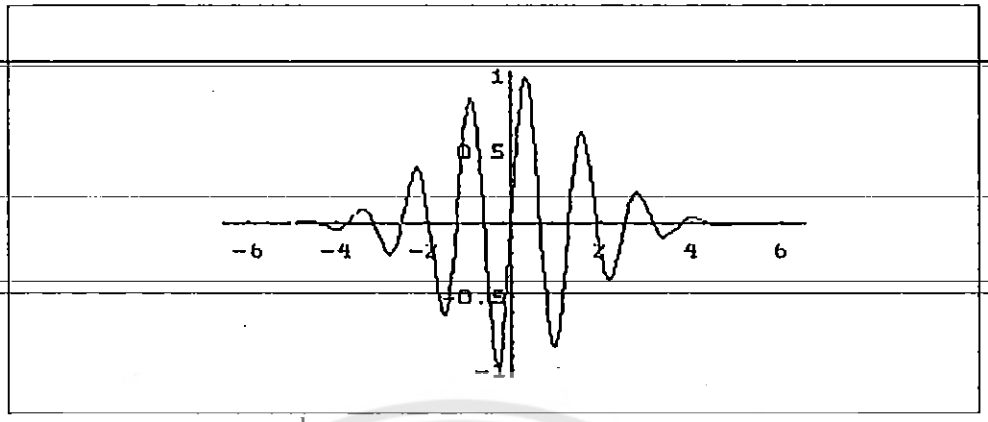
การแปลงเวฟเล็ต เป็นคณิตศาสตร์ที่ใช้ในการวิเคราะห์และสังเคราะห์ลักษณะของสัญญาณซึ่งมีประโยชน์มากในงานทางด้านการประมวลผลสัญญาณ (Signal processing) ทฤษฎีเวฟเล็ตสามารถนำมาประยุกต์เพื่ออธิบายลักษณะของสิ่งต่างๆหรือระบบใดๆได้ เช่น อธิบายการแก้ปัญหาสมการดิฟเฟอเรลเชียลที่จำลองระบบใดระบบหนึ่ง การเต้นของหัวใจ การไหลเวียนของเลือดผ่านร่างกาย การหาขอบภาพ การลดขนาดข้อมูล (ภาพ สัญญาณ) และใช้ในการแก้ปัญหาวิศวกรรมทางการแพทย์

2.2.1 พื้นฐานของการแปลงเวฟเล็ต

เมื่อใช้การแปลงฟูเรียร์ในการวิเคราะห์สัญญาณเฉพาะบางช่วงเวลาและความถี่เท่านั้นจะเสียเวลาในการคำนวณมากเพราะต้องคำนวณใหม่ตลอดย่าน ดังนั้นจึงได้มีการพัฒนาการแปลงที่สามารถวิเคราะห์สัญญาณได้เฉพาะช่วงเวลาและช่วงความถี่ที่สนใจเท่านั้น ทำให้สามารถพิจารณาผลกระทบของการเปลี่ยนแปลงของสัญญาณเฉพาะช่วงได้อย่างสะดวกและรวดเร็ว การแปลงที่พัฒนาขึ้นนี้เรียกว่า "การแปลงเวฟเล็ต"

การแปลงเวฟเล็ตจะใช้ในการอธิบายโครงสร้างของระบบสัญญาณที่ประกอบด้วยกลุ่มของสัญญาณเฉพาะมารวมกันเป็นสัญญาณหรือระบบนั้นๆ โดยสัญญาณเฉพาะนี้จะเป็นคลื่นเล็กๆ ที่เรียกว่าเวฟเล็ต ลักษณะของเวฟเล็ตจะเป็นคลื่นที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง (Oscillatory) และ

ขนาดของคลื่นจะลดลงสู่ศูนย์อย่างรวดเร็วทั้งสองด้าน ดังรูปที่ 2.1 ซึ่งเป็นเวฟเล็ตชนิดหนึ่งที่เรียกว่า เวฟเล็ตแบบ Daubechies20

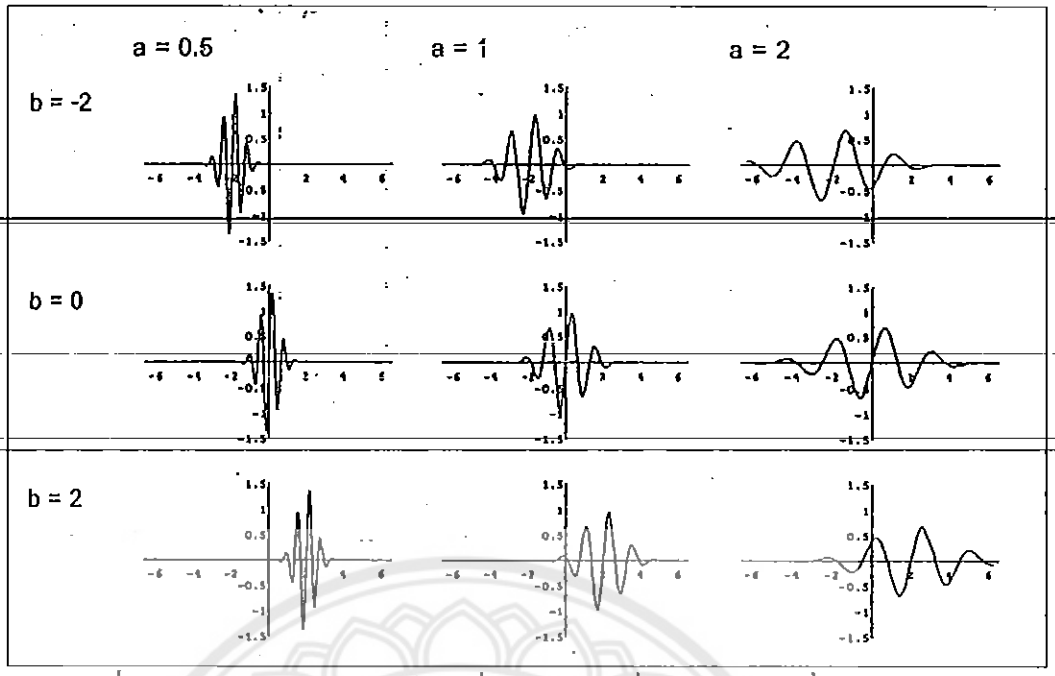


รูปที่ 2.1 ลักษณะของเวฟเล็ตแม่ชนิด Daubechies20

การนำเวฟเล็ตหลายๆอันมารวมกัน เป็นกลุ่มเพื่อใช้ในการอธิบายโครงสร้างของสัญญาณใดๆ โดยที่คลื่นเวฟเล็ตแต่ละตัวจะมีโครงสร้างมาจากฟังก์ชั่นเดียวกันซึ่งฟังก์ชั่นนี้จะเป็นเวฟเล็ตต้นกำเนิดที่เรียกว่า เวฟเล็ตแม่ (Mother Wavelets) คลื่นเวฟเล็ตแต่ละอันจะอยู่ภายในเซตของเวฟเล็ตนี้โดยแต่ละคลื่นจะเกิดจากการสเกล (Scaling :"a") และการเลื่อนตำแหน่ง (Translation :"b") ดังนั้นถ้าให้ W(t) เป็นฟังก์ชั่นเวฟเล็ตแม่ สามารถเขียนเป็นสมการทั่วไปของเวฟเล็ตที่ตำแหน่ง "a","b" ใดๆ ที่สัมพันธ์กันได้ดังนี้

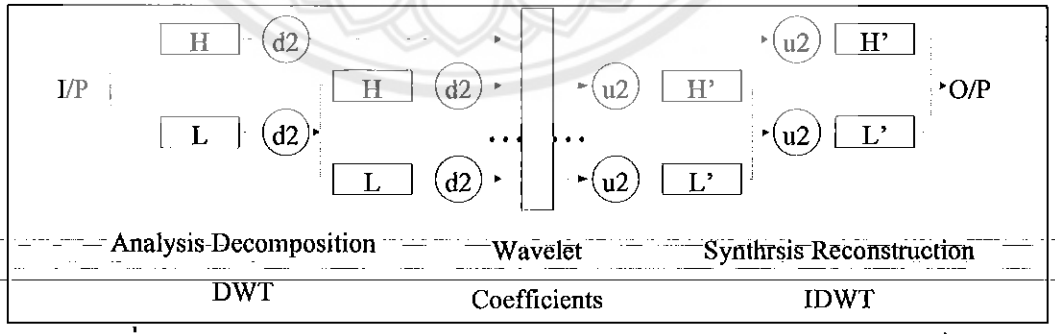
$$W_{b,a}(t) = \frac{1}{\sqrt{a}} W(\frac{t-b}{a}) \qquad (2.1)$$

W(t) จะเป็นฟังก์ชั่นเวฟเล็ตแม่ที่ถูกเลื่อนตำแหน่งและถูกสเกลโดยทารามิเตอร์ "a" และ "b" ตามลำดับ โดยที่ช่วงเวลาและความถี่ในการแปลงจะสัมพันธ์กันและเพื่อให้เวฟเล็ตที่ถูกสเกลไปแล้วมีพลังงานเท่ากับเวฟเล็ตแม่จึงต้องทำการนอมอร์ไลซ์ด้วย $1/\sqrt{a}$ เสมอ



รูปที่ 2.2 แสดงลักษณะของเวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่งไปที่ค่า a,b ต่างๆกัน

ทฤษฎีเวฟเล็ตจะใช้ในการอธิบายสิ่งๆสิ่งหนึ่งเสมือนการแตกสิ่งนั้นออกเป็นส่วนประกอบเล็กๆ สัมพันธ์กันโดยที่ชิ้นส่วนเหล่านี้จะอยู่ในรูปเวฟเล็ตที่ถูกสเกลและเลื่อนตำแหน่ง ดังนั้นจึงเปรียบเสมือนว่าสัญญาณใดๆ สามารถสร้างขึ้นมาได้โดยมีฟังก์ชั่นพื้นฐาน (Basis function) การแตกกระจายเวฟเล็ต(Wavelet Decomposition) ก็คือการทำการแปลงเวฟเล็ต (Wavelet Transform : WT)นั่นเองในทำนองเดียวกันการรวมกลับเวฟเล็ต(Wavelet Reconstruction) จะเป็นการแปลงกลับของเวฟเล็ต (Inverse Wavelet Transform : IWT) ซึ่งเป็นการนำส่วนประกอบย่อยๆ เหล่านี้มารวมกันเพื่อประกอบเป็นสัญญาณเดิมดังในรูปที่ 2.3



รูปที่2.3 ลักษณะของการแตกกระจายสัญญาณและการรวมกลับสัญญาณของเวฟเล็ต

2.2.2 การวิเคราะห์สัญญาณแบบหลายระดับความละเอียด

การวิเคราะห์สัญญาณแบบหลายระดับความละเอียด (Multiresolution Analysis : MRA) จะเป็นการวิเคราะห์สัญญาณที่สามารถเลือกระดับความละเอียดได้โดยการนำสัญญาณเล็กๆ ที่ระดับความละเอียด a ซึ่งมี b หลายๆตำแหน่งมารวมกันเกิดเป็นสัญญาณที่ระดับความละเอียดที่เลือกไว้ และเมื่อสัญญาณที่ทุกระดับความละเอียดมารวมกันจะเกิดเป็นสัญญาณอินพุทจริง

ก่อนที่จะศึกษาถึงการวิเคราะห์สัญญาณแบบMRAจะขออธิบายถึงทฤษฎีพื้นฐานของสเปซเวกเตอร์ (Vector space) ก่อนเนื่องจากมีความเกี่ยวข้องกับการวิเคราะห์แบบ MRA

สเปซของเวกเตอร์ในการวิเคราะห์สัญญาณ คือ สเปซหรือปริภูมิของสัญญาณใดๆ ที่เกิดจากการรวมกันของสัญญาณพื้นฐานย่อยๆ ที่เรียกว่า Basis function ถ้ากำหนดให้ a,b เป็นจำนวนจริงใดๆ และ $\hat{i},\hat{j}$ เป็นเวกเตอร์หนึ่งหน่วยในสเปซเวกเตอร์ V เสมอ ดังนั้นถ้าเวกเตอร์ใดประกอบขึ้นจากหลายเวกเตอร์หนึ่งหน่วยจะได้ว่าเวกเตอร์นั้นยังคงอยู่ในสเปซเวกเตอร์นี้ ดังนั้น $a\hat{i}+b\hat{j}+c\hat{k}+\ldots \in V$ โดยที่เวกเตอร์หนึ่งหน่วย $\hat{i},\hat{j}$ จะเป็นลักษณะเชิงตั้งฉาก (Orthogonal) ซึ่งกันและกัน ถ้าพิจารณาในลักษณะของสัญญาณอาจมองได้ว่า $\hat{i},\hat{j}$ คือ Basis function ที่เป็นสัญญาณเล็กๆ ที่นำมาประกอบกันเป็นสัญญาณใดๆ

สมมุติให้ V^j เป็นสเปซเวกเตอร์ที่มี j แสดงถึงระดับความละเอียดและจำนวนของ Basis function ที่ประกอบขึ้นเป็นฟังก์ชั่นนั้น ถ้า j มีค่าสูงขึ้นก็แสดงว่าที่ระดับความละเอียดสูงขึ้นจะมีจำนวน Basis function มากขึ้นทำให้สัญญาณที่เกิดจากการประกอบกันจาก Basis function มีความละเอียดมากขึ้นด้วย ดังนั้นอาจกล่าวได้ว่า j เป็นค่าแสดงถึงระดับความละเอียดของสัญญาณนั่นเอง จากข้อกำหนดเหล่านี้สามารถสรุปเป็นลักษณะสมบัติของการวิเคราะห์สัญญาณที่ระดับความละเอียดต่างๆ ได้ดังนี้

1) $V^{\infty}\ldots \subset V^{-1} \subset V^{1}\ldots \subset V^{\infty}$

2) Close $_{L^2}(\bigcup_{j\in ZZ} V^j) = L^2(IR)$: IR = เซตของจำนวนจริง

3) $(\bigcap_{j\in Z} V^j) = \{0\}$

4) $V^j + W^j = V^{j+1}$: $j\in$ ZZ : ZZ = เซตของจำนวนเต็ม

5) $f(x) \in V^j \longleftrightarrow f(2x) \in V^{j+1}$: $j \in$ ZZ: ZZ = เซตของจำนวนเต็ม

จากการที่ Basis function ประกอบกันเป็นสัญญาณการประมาณที่ระดับความละเอียด j ภายในสเปซ V^j จะเรียก Basis function เหล่านี้ว่า ฟังก์ชั่นสเกลลิ่ง (Scaling function : $\phi(t)$) สัญญาณเหล่านี้จะเกิดที่ตำแหน่งเวลาต่างๆกันของสเปซและมีความถี่เท่ากันภายในสเปซเดียวกัน

ฟังก์ชั่นสเกลลิ่งที่ระดับสเปซสูง (ระดับความละเอียดสูง) จะมีความถี่สูงและที่ระดับต่ำกว่าจะมีความต่ำกว่า ดังนั้นความสัมพันธ์กันระหว่างฟังก์ชั่นสเกลลิ่งของแต่ละสเปซจะเป็นดังนี้

$$f(x) \in V^j \quad \longleftrightarrow \quad f(2x) \in V^{j+1} \qquad : j \in ZZ \qquad (2.2)$$

ZZ = เซตของจำนวนเต็ม

จากสมการที่ 2.2 ทำให้สามารถเขียนเป็นความสัมพันธ์ระหว่างฟังก์ชั่นสเกลลิ่งภายในสเปซใดๆ ได้ดังนี้

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k) \qquad : j, k \in ZZ \qquad (2.3)$$

ZZ = เซตของจำนวนเต็ม

จากสมการที่ 2.3 จะพบว่าระดับความละเอียดต่ำลงมาหนึ่งระดับ Basis function จะมีความถี่ลดลงมาครั้งละสองเท่า อาศัยลักษณะคุณสมบัติ MRA จะทำให้สามารถทำการประมาณสัญญาณ $f(t) \in L^2(IR)$ ไปอยู่ในสเปซที่ระดับความละเอียด j ใดๆ ก็ได้ดังนี้

$$f_j(t) = \sum_k c_k^j \phi_{j,k}(t) \qquad (2.4)$$

โดยที่ C_k^j เป็นสัมประสิทธิ์หรือน้ำหนักที่คูณกับฟังก์ชั่นสเกลลิ่งที่ตำแหน่ง k ใดๆ แล้วประกอบขึ้นเป็น f(t) ที่ระดับความละเอียด j นั้นๆ

จากการวิเคราะห์สัญญาณที่ระดับความละเอียดต่ำลงมาจะทำให้พลังงานหรือสัญญาณบางส่วนหายไปอยู่ในสเปซอีกอันหนึ่งซึ่งจะเรียกว่า สเปซของเวกเตอร์เวฟเล็ต (Wavelet vector space : W^j) สเปซชนิดนี้จะคล้ายกับสเปซของเวกเตอร์ ดังนั้นสัญญาณภายใน W^j จะประกอบด้วย Basis function เช่นเดียวกันจะเรียกว่า ฟังก์ชั่นเวฟเล็ต (Wavelet function : $\psi(t)$) ดังนั้นสามารถเขียนสมการฟังก์ชั่นเวฟเล็ตที่ระดับความละเอียดใดๆ ได้ดังนี้

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \qquad : j, k \in ZZ \qquad (2.5)$$

ZZ = เซตของจำนวนเต็ม

จากสมการที่ 2.5 ซึ่งมีลักษณะเป็นสัญญาณที่มีการเลื่อนตำแหน่งและเปลี่ยนความถี่อย่างต่อเนื่องจะเปลี่ยนเป็นฟังก์ชั่นเวฟเล็ต ที่มีการเลื่อนตำแหน่งและเปลี่ยนความถี่แบบเต็มหน่วยโดยที่ $a = 2^{-j}$, $b = 2^{-j}k$

ถ้ากำหนดให้ $g_j(t)$ เป็นสัญญาณที่เกิดจาก Basis function และ $\psi_{j,k}(t)$ ภายในสเปซเดียวกันมารวมกันเป็นสัญญาณใดๆ จะได้ว่า

$$g_j(t) = \sum_k d_k^j \psi_{j,k}(t) \qquad (2.6)$$

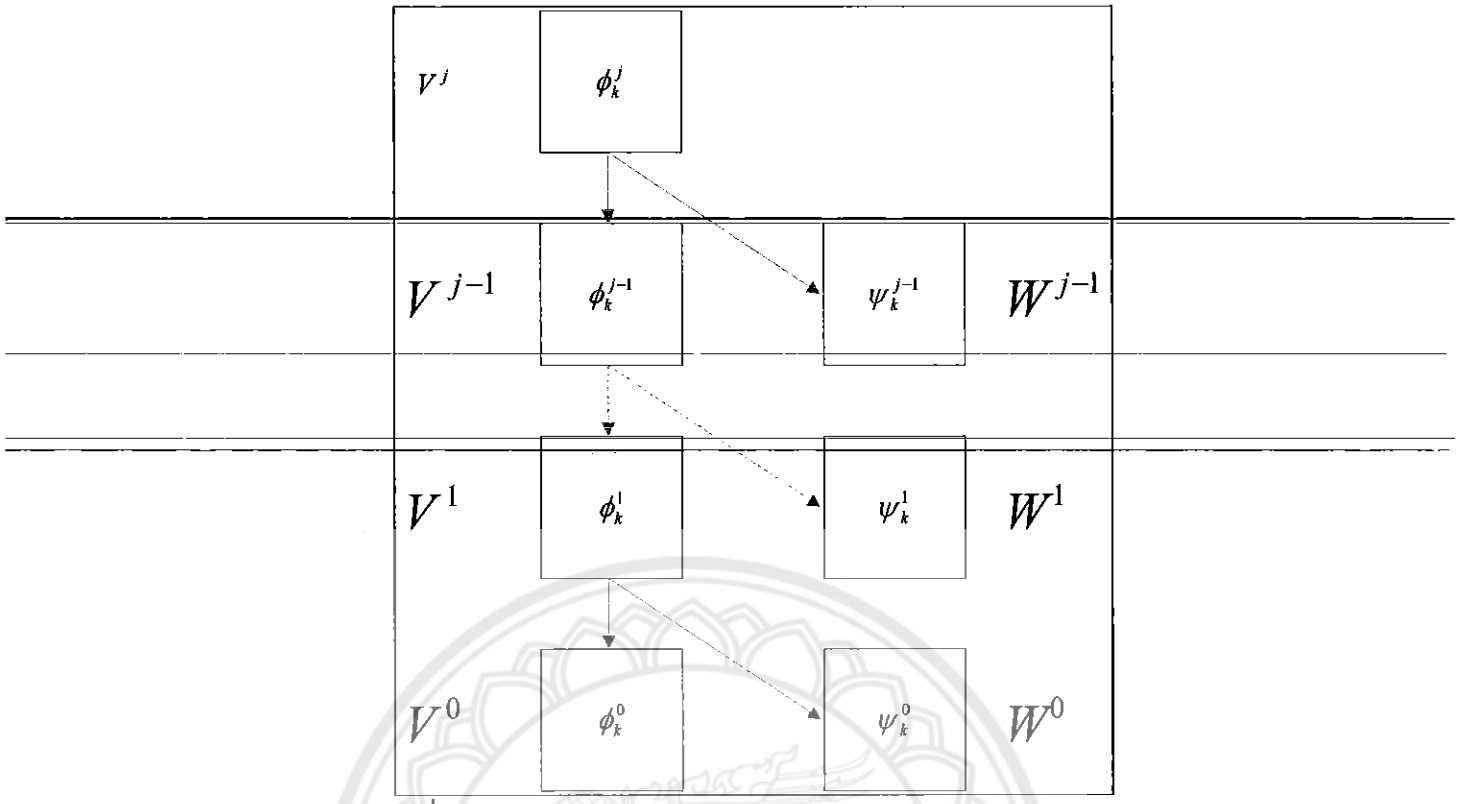
โดยที่ d_k^j เป็นสัมประสิทธิ์หรือค่าน้ำหนักที่คูณกับฟังก์ชั่นเวฟเล็ตที่ตำแหน่งนั้นๆ เพื่อเกิดเป็นสัญญาณ $g_j(t)$ ดังนั้นจากความสัมพันธ์ $V^j + W^j = V^{j+1}$ และจากสมการที่ 2.4 และ 2.6 จะได้ว่า

$$f_{j+1} = f_j + g_j \qquad (2.7)$$

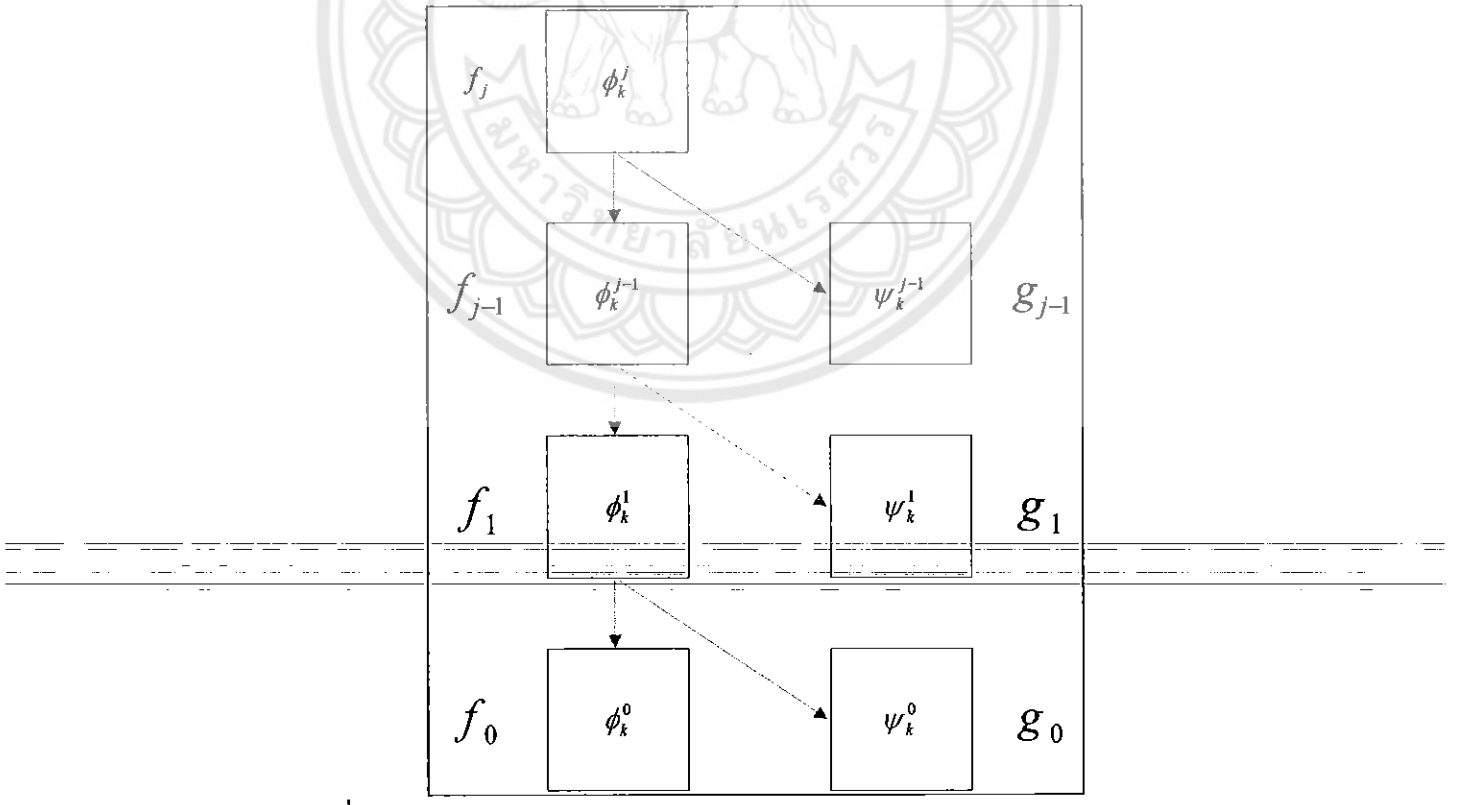
สมมุติให้ $f(t) \in V^{j+1}$ จะสามารถแตกกระจายให้ $f(t)$ ให้มีความละเอียดน้อยลงได้จากสมการ $V^j + W^j = V^{j+1}$ ซึ่งในขณะเดียวกัน V^j สามารถแตกต่อไปได้เรื่อยจนกระทั่ง $j = 0$ ดังนั้นจะได้เป็นความสัมพันธ์ว่า

$$V^{j+1} = V^0 + W^0 + W^1 + \ldots + W^j \qquad (2.8)$$

ในทำนองเดียวกัน f_{j+1} ก็สามารถแตกกระจายเป็น f_j และ g_j ซึ่งสามารถแสดงเป็นภาพการแตกกระจายสเปซและสัญญาณได้ดังรูปที่ 2.4 และ 2.5



รูปที่ 2.4 ลักษณะการกระจาย (Decomposition) ของสเปซของเวกเตอร์



รูปที่ 2.5 ลักษณะการกระจาย (Decomposition) สัญญาณไปยังระดับความละเอียดต่างๆ

f และ g ที่ลดระดับความละเอียดลงมาจะมีความถี่ของ Basis function ลดลงครั้งละสองเท่าเสมอและเราสามารถแจกแจงสัญญาณ f(t) ที่อยู่ในรูปแบบของฟังก์ชั่นสเกลลิ่งและฟังก์ชั่นเวฟเล็ตได้ดังนี้

$$f(t) = f_j + g_j + g_{j+1} + g_{j+2} + \ldots + g_\infty \qquad (2.9)$$

$$f(t) = \sum_m c_j(m)\phi_{j,m}(t) + \sum_{i=j}^{\infty}\sum_m d_i(m)\psi_{i,m}(t) \qquad (2.10)$$

ดังนั้นสัมประสิทธิ์เวฟเล็ตสามารถหาได้จากสมการการโปรเจคชั่น ดังสมการที่ 2.11 และ 2.12

$$c_j(m) = \left\langle f(t), \phi_{j,m}(t) \right\rangle \qquad (2.11)$$

$$d_j(m) = \left\langle f(t), \psi_{j,m}(t) \right\rangle \qquad (2.12)$$

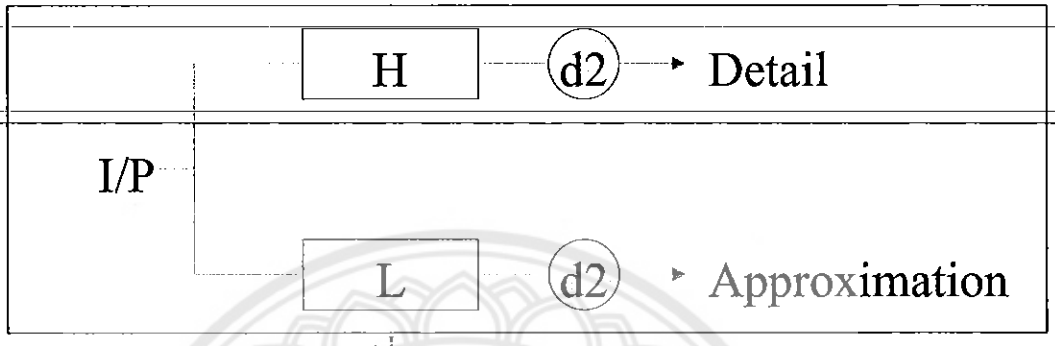
การแตกกระจายสัญญาณ f(t) ในสเปซ V^0 ไปจนถึงระดับความละเอียดที่ j ดังนั้นสัมประสิทธิ์ $c_0(m)$ จะถูกแตกเป็นสัมประสิทธิ์ในเซตของ $c_j(m)$ ในสเปซ V^j และกลุ่มของเซต $d_{-1}(m)$, $d_{-2}(m)$,..., $d_j(m)$ ซึ่งอยู่ในเวฟเล็ตสเปซที่ระดับความละเอียดต่างๆกัน จากขบวนการต่างๆที่กล่าวมาจะเป็นลักษณะของวิธีการแปลงเวฟเล็ตแบบเต็มหน่วย (Discrete Wavelet Transform : DWT)

2.2.3 การแปลงเวฟเล็ต

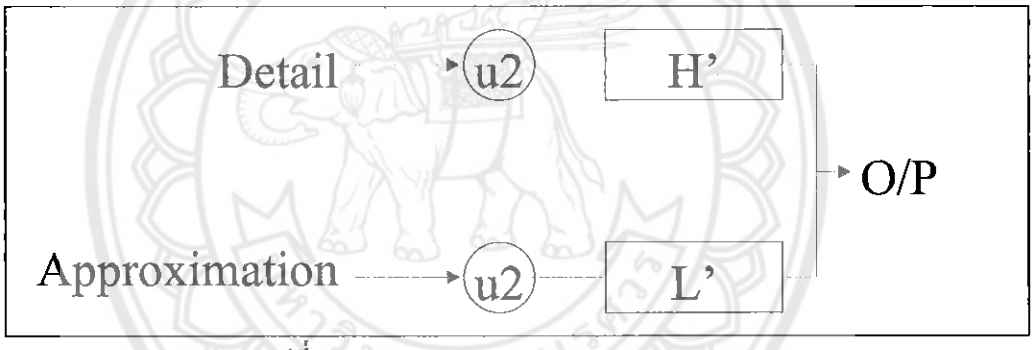
ขบวนการแปลงเวฟเล็ตจะมีลักษณะคล้ายกับขบวนการออคเทฟฟิลเตอร์แบงค์ (Octave filter banks) เนื่องจากการพิจารณาสัญญาณที่สามารถเปลี่ยนแปลงความถี่ในอัตราครั้งละสองเท่าซึ่งเปรียบเสมือนกับการนำสัญญาณอินพุทผ่านวงจรกรองความถี่ที่มีแบนวิธที่มีอัตราการลดลงสองเท่าเหมือนกับแบนวิธของฟังก์ชั่นหน้าต่าง (Window function) ในขณะนั้นนั่นเอง ดังนั้นจะสามารถนำเอาหลักการของฟิลเตอร์แบงค์ (Filter banks) มาใช้ในการแปลงเวฟเล็ตในทางปฏิบัติได้ ก่อนที่จะอธิบายการสร้างการแปลงเวฟเล็ตในลักษณะฟิลเตอร์แบงค์ จะขออธิบายหลักการพื้นฐานของฟิลเตอร์แบงค์ก่อน

ฟิลเตอร์แบงค์แบบสองช่องสัญญาณ (Two-channel filter banks) เป็นการแยกสัญญาณอินพุทออกเป็นสองส่วน โดยแบ่งเป็นส่วนของความถี่ต่ำและส่วนของความถี่สูง ดังนั้นฟิลเตอร์แบงค์แบบสองช่องสัญญาณ จึงประกอบด้วยส่วนที่เป็น Low pass filter :L และ Complementary

highpass filter : H ดังรูปที่ 2.6 ซึ่งมีลักษณะของการวิเคราะห์ฟิลเตอร์แบงค์แบบสองช่องสัญญาณและเป็นโครงสร้างที่กลับกัน กับการสังเคราะห์การสร้างกลับฟิลเตอร์แบงค์แบบสองช่องสัญญาณ (Reconstruction Two-channel filter banks) ความถี่ต่ำและความถี่สูง คือ L และ H กับตัวกรองการสร้างกลับความถี่ต่ำและความถี่สูงคือ L' , H' มีความสัมพันธ์กันในลักษณะที่เรียกว่า Quadrature mirror filters

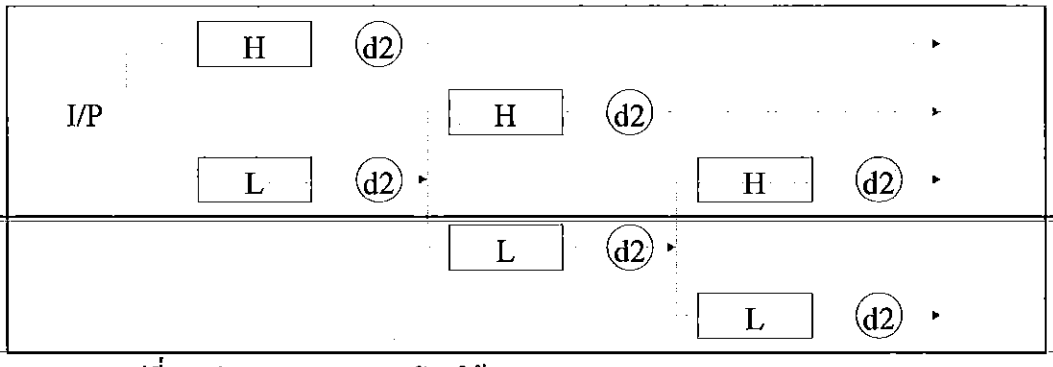


รูปที่ 2.6 Two-channel filter banks



รูปที่ 2.7 Reconstruction Two-channel filter banks

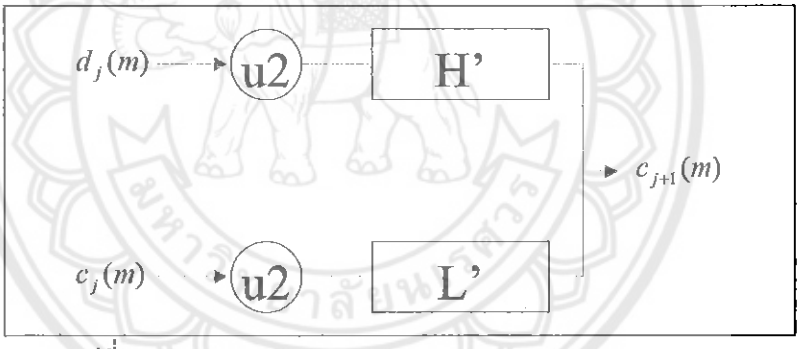
ลักษณะการวิเคราะห์ออคเทฟฟิลเตอร์แบงค์ จะเป็นโครงสร้างแบบต้นไม้ (Tree Structure) ซึ่งเป็นการสร้างเอาฟิลเตอร์แบงค์แบบสองช่องสัญญาณ มาเรียงต่อกันโดยใช้สัญญาณเอาท์พุทในส่วนที่เป็นความถี่ต่ำมาทำการแยกความถี่ออกอีกครั้งหนึ่ง ในกรณีที่ทำการแปลงเวฟเล็ตซ้ำในแนวของ Lowpass จะเป็นลักษณะของ Dyadic tree structure ดังรูปที่ 2.8 ซึ่งโครงสร้างในรูปนี้จะเป็นการแปลงแบบเต็มหน่วย (Discrete Wavelets Transform : DWT)



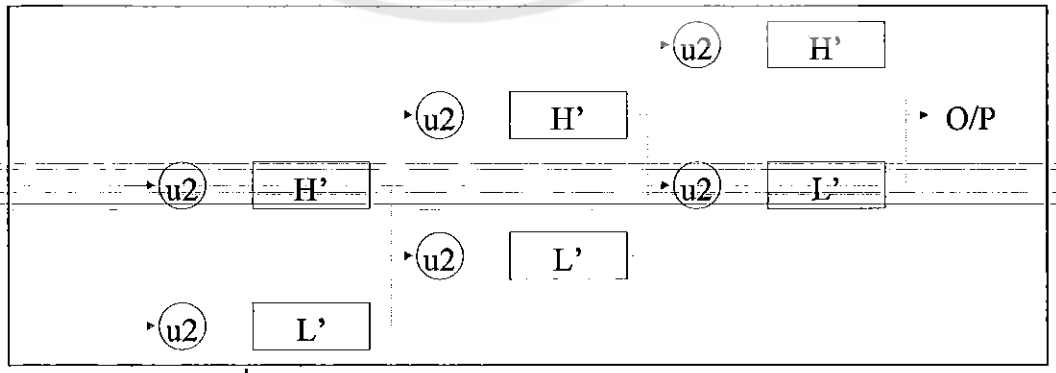
รูปที่ 2.8 ลักษณะของ DWT โดยใช้ Dyadic tree structure (Octave filter banks)

2.2.4 การแปลงกลับเวฟเล็ต

การแปลงแบบเต็มหน่วย (DWT) เป็นการแตกกระจาย (Decomposition) สัญญาณหรือการโปรเจคสัญญาณลงไปในสเปซของ V^{j} และ W^{j} กลับไปเป็นสเปซ V^{j+1} ก็จะเป็นการเพิ่มรายละเอียดของสัญญาณ f(t) จากระดับ j ไปเป็นระดับ j+1 วิธีการดังกล่าวนี้จะเหมือนกับขบวนการฟิลเตอร์แบงค์แบบสองช่องสัญญาณ ซึ่งเป็นการหาค่าของ $c_{j+1}(n)$ จากค่า $c_j(m)$ และ $d_j(m)$ ซึ่งสามารถอธิบายได้ดังรูปที่ 2.9



รูปที่ 2.9 ลักษณะของ Reconstruction Two-channel filter banks



รูปที่ 2.10 ลักษณะของ IDWT โดยการใช้ dyadic tree structure

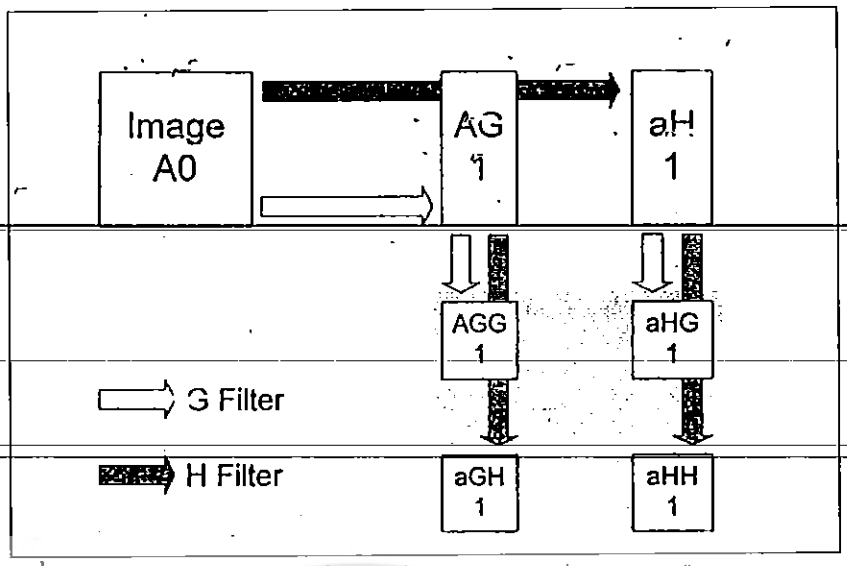
ลักษณะของ Tree structure filter banks เพื่อทำการรวบรวมสัมประสิทธิ์ $c_j(m)$ และ $d_j(m)$ กลับมาเป็น $c_{j+1}(n)$ อีกครั้ง ขบวนการนี้เรียกว่า การแปลงกลับเวฟเล็ต (Inverse Wavelet Transform : IWT) จากกระบวนการแปลงเวฟเล็ตและการแปลงกลับเวฟเล็ตจะสังเกตได้ว่าสัญญาณเอาท์พุทที่ได้จากการแปลงกลับจะมีค่าประมาณเท่ากับสัญญาณอินพุทของการแปลงเวฟเล็ตโดยที่รูปแบบของฟังก์ชั่นถ่ายโอน (Transfer function) จะมีในลักษณะการสร้างกลับอย่างสมบูรณ์ของฟิลเตอร์แบงค์ (Perfectly reconstructing filter banks) จากที่กล่าวมานี้จะใช้เฉพาะในกรณีของ Orthonormal wavelets หรือ Orhtonormal filter banks เท่านั้น

2.2.5 การประยุกต์ใช้กับข้อมูลภาพ

จากคุณสมบัติของเวฟเล็ต ที่เหมาะกับการนำมาประยุกต์ใช้กับการลดขนาดข้อมูลภาพ ในหัวข้อนี้จึงกล่าวถึงวิธีนำการแปลงเวฟเล็ตมาใช้งาน ขบวนการแปลงเวฟเล็ตที่นำมาใช้งานนี้จะเป็นวิธีการของ Multi resolution wavelet transform

จากข้อมูลภาพที่เป็นสัญญาณในสองมิตินี้ ขบวนการแปลงเวฟเล็ตที่ใช้จะต้องเป็นสองมิติด้วย แต่เพื่อความง่ายในการใช้งานข้อมูลภาพจะถูกแปลงสัญญาณเป็นหนึ่งมิติแทน เพื่อให้สามารถใช้การแปลงเวฟเล็ตที่กล่าวมาได้ โดยการจัดเรียงของแต่ละจุดภาพใหม่ แยกตามแนวแกนตั้ง และแกนนอน ข้อมูลที่ได้จากเส้นภาพในแนวแกนนอนแต่ละเส้นจะเป็นสัญญาณหนึ่งมิติ ซึ่งจะถูกกระทำด้วยการแปลงเวฟเล็ตจนครบทุกเส้นภาพ ผลที่ได้ออกมาจะเป็นค่าสัมประสิทธิ์ของเวฟเล็ตของสัญญาณในแต่ละเส้นภาพที่ไม่เกี่ยวข้องกันแล้วข้อมูลนี้จะถูกนำไปผ่านขบวนการแปลงเวฟเล็ตอีกครั้งตามแนวแกนตั้ง

จากการแปลงเวฟเล็ตแบบ Multiresolution ที่มีอัตราส่วนในการ Scaling เท่ากับสองผลลัพธ์ ที่ได้ จะแบ่งออกเป็นสองส่วนคือ High pass filter และ Low pass filter ดังนั้น เมื่อข้อมูลภาพถูกนำมาทำการแปลง ค่าสัมประสิทธิ์ที่ได้จึงแบ่งออกเป็นสี่ส่วน แสดงดังรูปที่ 2.11

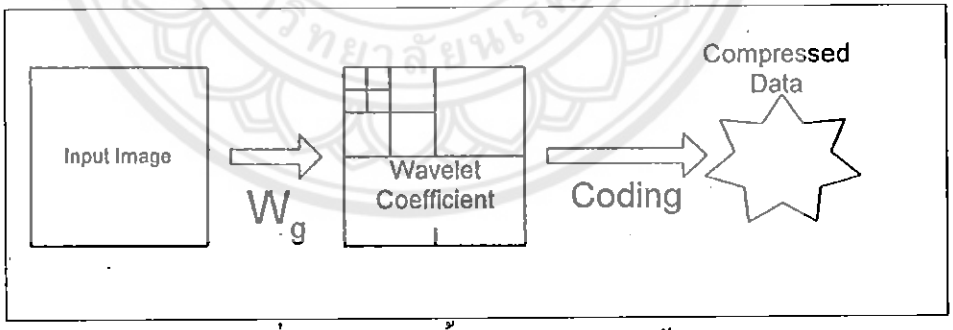


รูปที่ 2.11 แสดงการแปลงเวฟเล็ตของข้อมูลภาพออกเป็น 4 ส่วน คือ AGG1 aGH1 aHG1 aHH1 (ส่วนที่แรเงา) ซึ่งองค์ประกอบทั้งสี่นี้บังคงมีจำนวนจุดรวมกันทั้งหมดเท่ากับภาพต้นแบบ (AO)

จากนั้นส่วนของข้อมูลความถี่ต่ำที่เหลืออยู่ (AGG1) จะสามารถนำไปแตกกระจายต่อไปได้เช่นเดียวกันกับการแปลงเวฟเล็ตของสัญญาณหนึ่งมิติ

2.2.6 การประยุกต์ใช้เวฟเล็ตในการลดขนาดข้อมูล

ขบวนการลดขนาดข้อมูลจะกระทำกับค่าสัมประสิทธิ์เวฟเล็ตที่ได้ จากขบวนการวิเคราะห์ ดังรูปที่ 2.12



รูปที่ 2.12 รูปแสดงขั้นตอนการลดขนาดข้อมูล

ค่าสัมประสิทธิ์ที่ได้ ซึ่งเป็นส่วนประกอบย่อยของสัญญาณจะถูกนำมาผ่านขบวนการเข้ารหัส ในขั้นตอนของการเข้ารหัสนี้ อาจมีข้อมูลบางส่วนที่หายไปหรือเกินออกมา จากนั้นในขั้นตอนของการวิเคราะห์จะสามารถสร้างสัญญาณเอาท์พุท ที่มีลักษณะใกล้เคียงกับสัญญาณอินพุท ได้ทั้งนี้ก็ขึ้นอยู่กับว่าในขั้นตอนของการลดขนาดข้อมูลมีความผิดพลาดเกิดขึ้นเพียงใด

2.3 การหาค่าความผิดพลาด (SNR และPSNR) และการหาค่าอัตราส่วนการบีบอัดข้อมูล

อัตราส่วนของการบีบอัดข้อมูล (Compression Ratio : CR) คือ การเปรียบเทียบค่าระหว่างขนาดของข้อมูลต้นฉบับ (n1) กับข้อมูลที่ถูกบีบอัดข้อมูลแล้ว (n2) ดังสมการที่ 2.13

$$CR = \frac{n1}{n2} \qquad (2.13)$$

การหาความผิดพลาดของการบีบอัดข้อมูลนั้น เราสามารถใช้หลักการของ Signal to Noise Ratio (SNR) และ Peak Signal to Noise Ratio (PSNR) โดยจะเป็นตัววัดคุณภาพของการบีบอัดข้อมูลว่า เมื่อบีบอัดข้อมูลแล้วคุณภาพของภาพที่ได้ใหม่จะมีการสูญเสียมากน้อยเพียงไหน ดังสมการที่ 2.14 และสมการที่ 2.15

$$SNR_{(dB)} = 10\log_{10}\left\{\frac{\left[\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\left(s(m,n)\right)^2\right]}{\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\left(s(m,n)-\hat{s}(m,n)\right)^2}\right\} \qquad (2.14)$$

โดยที่ $s(m,n)$ และ $\hat{s}(m,n)$ คือ ค่าข้อมูลภาพต้นฉบับและภาพที่ถูกคลายหลังการบีบอัดแล้วตามลำดับ สำหรับ m และ n เป็นค่าดัชนีตามแนวนอนและแนวตั้งของภาพทั้งสอง

$$PSNR_{(dB)} = 10\log_{10}\left\{\frac{\max|s(m,n)|}{\frac{1}{MN}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\left(s(m,n)-\hat{s}(m,n)\right)^2}\right\} \qquad (2.15)$$

โดยที่ $\max|s(m,n)|$ คือ ค่าสูงสุดของข้อมูลภาพต้นแบบและค่า MN คือขนาดของข้อมูลภาพ

2.4 การเข้ารหัสแบบซีโรทรี (Zerotree)

กระบวนการนี้ถูกพัฒนาโดย Jerome M. Shapio เมื่อปี ค.ศ. 1993 การเข้ารหัสแบบซีโรทรีถูกออกแบบเป็นพิเศษโดยใช้ wavelet transforms ซึ่งอธิบายได้ว่าทำไมมันมีคำว่า wavelet ในชื่อของมัน การเข้ารหัสแบบซีโรทรีเดิมออกแบบมาเพื่อใช้กับรูปภาพ (2D-signals) แต่มันสามารถใช้บนสัญญาณที่มีมิติได้ด้วย (dimensional signals)

การเข้ารหัสแบบซีโรทรีอยู่บนพื้นฐานของการเข้ารหัสตามลำดับ บีบอัดรูปภาพให้กลายเป็น bitstream โดยเพิ่มรายละเอียด หมายถึงเมื่อบิทถูกเติมใน stream การถอดรหัสรูปภาพ

จะต้องบรรจุรายละเอียดต่างๆ ลักษณะคล้ายๆกับการเข้ารหัสรูปภาพของ JPEG มันเหมือนการแทนตัวเลขเช่นเดียวกับ π ทั้งหมดของตัวเลข เราเพิ่มความแม่นยำของตัวเลขที่ต้องการ ลำดับของการเข้ารหัสรู้จักกันในคำว่า embedded encoding หรือ E ใน EZW

การเข้ารหัสรูปภาพโดยใช้วิธีซีโรทรี ค่าที่ดีที่สุดของผลลัพธ์ที่ได้ เป็นผลน่าประหลาดใจของการบีบอัด ด้วยลักษณะการบีบอัดสายของข้อมูลที่สามารถกำหนดอัตราบิท(bit rate)ได้ อัตราบิทอาจเป็นได้ค่าเดียวถ้าข้อมูลสูญเสียที่ใดที่หนึ่ง ดังนั้นการบีบอัดนี้เป็น lossy อย่างไรก็ตาม การบีบอัดแบบ Lossless ก็อาจเป็นไปได้ในการเข้ารหัสแบบซีโรทรี

การเข้ารหัสแบบซีโรทรีนั้นอยู่บนพื้นฐานของ 2 สิ่งคือ

- ธรรมชาติของรูปโดยทั่วไปจะมีระยะคลื่นต่ำ (low pass spectrum) เมื่อรูปภาพถูกแปลงโดย wavelet พลังงานใน subband จะลดลงเช่นเดียวกับสเกล การลดลง (สเกลต่ำหมายถึงมีการกระจายสูง) ดังนั้นค่าสัมประสิทธิ์ของ wavelet ต้องอยู่ในค่าเฉลี่ย คือค่าเล็กในsubband ที่สูงและเล็กกว่าใน subband ที่ต่ำกว่า จะแสดงในลำดับการเข้ารหัสเป็นทางเลือกโดยธรรมชาติของการบีบอัดรูปภาพโดยการแปลง wavelet ขณะที่ subband ที่สูงกว่าจะถูกเพิ่มรายละเอียด
- ค่าสัมประสิทธิ์ของ wavelet ขนาดใหญ่จะสำคัญกว่าค่าสัมประสิทธิ์ที่มีขนาดเล็กกว่า

แนวความคิดของซีโรทรี

- discrete wavelet transform หรือ การกระจาย hierarchical subband
- การทำนายข้อมูลที่สูญหายของสเกลตรงข้าม จากความคล้ายคลึงกันของตัวมันเอง ของรูปภาพแต่กำเนิด
- Entropy ถูกเข้ารหัสต่อเนื่องจากการประมาณค่า และ
- การสูญเสียข้อมูลโดยทั่วไปจะได้จากการปรับ arithmetic coding

2.5 กระบวนการเข้ารหัสแบบซีโรทรี (Zerotree)

ในการทำการเข้ารหัสแบบซีโรทรีนั้นจะต้องกำหนดค่า Threshold เริ่มต้นก่อนซึ่งหาได้จากสูตร

$$T_0 = 2^{\lfloor \log_2(MAX(|\gamma(x,y)|)) \rfloor}$$

ซึ่ง MAX หมายถึงค่าที่มากที่สุดของสัมประสิทธิ์ในรูปภาพ และ $\gamma(x, y)$ คือค่าของสัมประสิทธิ์

จากนั้นจะถูกผ่านขั้นตอนการทำ dominant pass คือการสแกนภาพแล้วแทนด้วยสัญลักษณ์สำหรับทุกๆค่าสัมประสิทธิ์ ค่าสัมประสิทธิ์ที่มีค่ามากกว่า Threshold จะถูกแทนด้วย sp (significant coefficient) แทนด้วย sn กับค่าที่ติดลบ (negative significant) แทนด้วย zr สำหรับค่าสัมประสิทธิ์ที่มีขนาดน้อยกว่า Threshold และลูกหลานของมันมีค่าน้อยกว่า Threshold ดังนั้นค่าสัมประสิทธิ์นี้จึงถูกเรียกว่า zerotree root สุดท้ายแทนด้วย iz มันอาจจะเกิดขึ้นจนค่าสัมประสิทธิ์ของมันมีค่าน้อยกว่า Threshold แต่บางค่าของลูกหลานของมันมีค่ามากกว่า Threshold จึงถูกเรียกว่า isolated zero

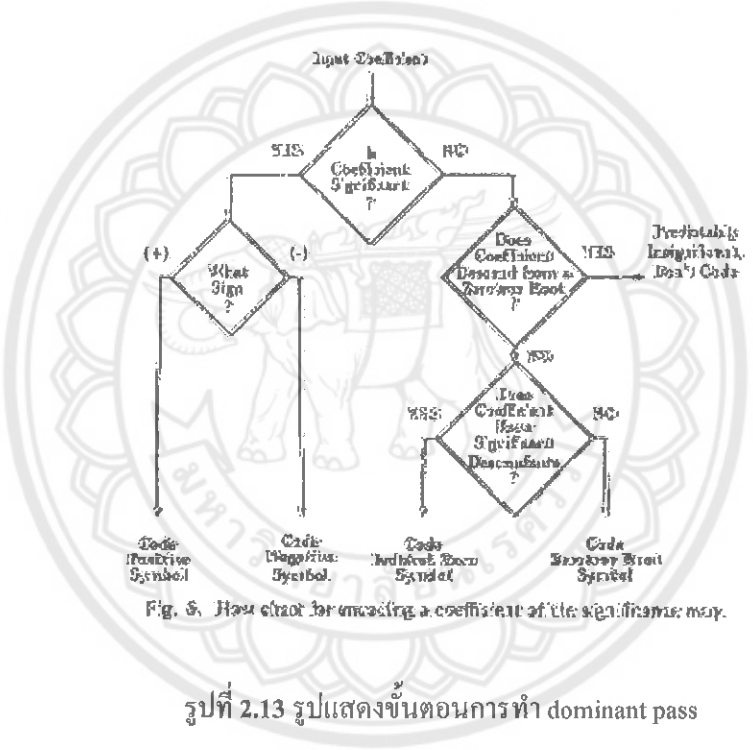


Fig. 5. Flow chart for encoding a coefficient of the significance map.

รูปที่ 2.13 รูปแสดงขั้นตอนการทำ dominant pass

หลังจากนั้นเข้าสู่การทำ Subordinate pass เป็นการผ่านที่ประณีตทำให้เกิดการลวงของระยะห่างที่ไม่แน่นอน แต่มันเด่นน้อยลงที่จะผลิต significant bit ถัดไป ของค่าสัมประสิทธิ์ทั้งหมดบน subordinate list ในลำดับของการจัดลำดับนี้เป็นหนทางที่ตัวถอดรหัสสามารถกระทำได้เช่นกัน เนื่องจากค่าสัมประสิทธิ์ขนาดใหญ่ที่สุดจะถูกส่งข้อมูลไปก่อนเป็นอำดับแรก แต่บนพื้นฐานเราจะไม่สนับสนุนการทำเช่นนี้

ตัวอย่างการเข้ารหัสซีโรทรี

การเข้ารหัสซีโรทรีของข้อมูลขนาด 4 x 4 สามารถเข้ารหัสซีโรทรีได้ดังต่อไปนี้

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

ขั้นตอนที่ 1

ทำการสร้างแผนผังการเข้ารหัสจากข้อมูลขนาด 4 x 4 ซึ่งสามารถจัดลำดับแผนผังการเข้ารหัสซีโรทรีได้ดังต่อไปนี้

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

ขั้นตอนที่ 2

ทำการหาค่า Threshold ตั้งต้น T_0 จากสมการ

$$T_0 = 2^{\lfloor \log_2(MAX(|\gamma(x,y)|)) \rfloor}$$

โดยที่ MAX หมายถึงค่าที่มากที่สุดของสัมประสิทธิ์ในรูปภาพ และ $\gamma(x, y)$ คือค่าของสัมประสิทธิ์ และได้ค่าสัมประสิทธิ์ที่มากที่สุดคือ 26

$$T_0 = 2^{\lfloor \log_2(|26|) \rfloor} = 16$$

ขั้นตอนที่ 3

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

ทำการหา dominant list โดยการแทนสัญลักษณ์สำหรับทุกๆค่าสัมประสิทธิ์ ตามลำดับ แผนผังการเข้ารหัสจะได้ดังต่อไปนี้

26 > 16 → sp

6 < 16 → มี descendants < 16 → zr

-7 < 16 → มี descendants < 16 → zr

7 < 16 → มี descendants < 16 → zr

จะได้สัญลักษณ์สำหรับส่งคือ sp zr zr zr

ขั้นตอนต่อไปทำการหา subordinate list ขั้นตอนนี้เป็นการกำหนดว่าค่า**สัมประสิทธิ์ควรถูกแทนค่าด้วยบิท '0' หรือบิท '1' จาก** $MAX(|\gamma(x,y)|$ ต้องอยู่ในช่วง $[T_0,2T_0)$**ดังนั้นจะได้ว่า** $MAX(|\gamma(x,y)|$ อยู่ในช่วง $[16,32)$

จากการหาข้างต้นนี้ จะได้ว่าค่าสัมประสิทธิ์ที่จะถูกแทนด้วยบิท '0' **จะอยู่ในช่วงของ** $[16,24)$ **หรือ** $[T_0,1.5T_0)$ และถูกแทนด้วยบิท '1'ในช่วงของ $[24,32)$ **หรือ** $[1.5T_0,2T_0)$ **ดังแสดงในตารางที่** 2.1

หมายเหตุ ค่าสัมประสิทธิ์ที่นำมาทำการหา subordinate list ต้องมีสัญลักษณ์เป็น sp หรือ sn เท่านั้น

ตารางที่ 2.1

ค่าสัมประสิทธิ์	symbol	reconstruct
26	1	28

หลังจากการเข้ารหัสซีโรทรีในรอบแรกแล้วให้ทำการหาค่า Threshold ใหม่โดยจะมีค่าเท่ากับ

$$T_N = T_{N-1} / 2 \qquad ; N = \text{ลำดับของ Threshold}$$

ซึ่งจะได้ $T_1 = T_0 / 2 = 8$

X	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

ขั้นตอนต่อไปทำการหา dominant list และ subordinate list ใหม่อีกครั้ง

ขั้นตอนการทำ dominant list

6 < 8 → มี descendants > 8 → iz
-7 < 8 → มี descendants < 8 → zr
7 < 8 → มี descendants < 8 → zr
13 ไม่มี descendants > 8 → sp
10 ไม่มี descendants > 8 → sp
6 ไม่มี descendants < 8 → iz
4 ไม่มี descendants < 8 → iz

จะได้สัญลักษณ์สำหรับส่งคือ iz zr zr sp sp iz iz

ขั้นตอนการทำ subordinate list

[8,16) [16,24) [24,32)
'0' : [8,12) [16,20) [24,32)
'1' : [12,16) [20,24) [28,32)

ผลของการหา subordinate list ในขั้นที่ 2 แสดงในตารางที่ 2.2

ตารางที่ 2.2

ค่าสัมประสิทธิ์	symbol	reconstruct
26	1 0	28 26
13	1	14
10	0	10

ต่อไปทำการหา Threshold ใหม่และทำการหา dominant list และ subordinate list ใหม่ รอบที่ 3

$$T_2 = T_1 / 2 = 4$$

X	6	X	X
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

ขั้นตอนการทำ dominant list

6 > 4 → sp

-7 < 4 มี descendants > 4 → sn

7 > 4 → sp

6 > 4 → sp

4 > 4 → sp

4 > 4 → sp

-4 ไม่มี descendants <4 → sn

2, -2 → iz

4 > 4 → sp

-3, -2, 0 → iz

ขั้นตอนการทำ subordinate list

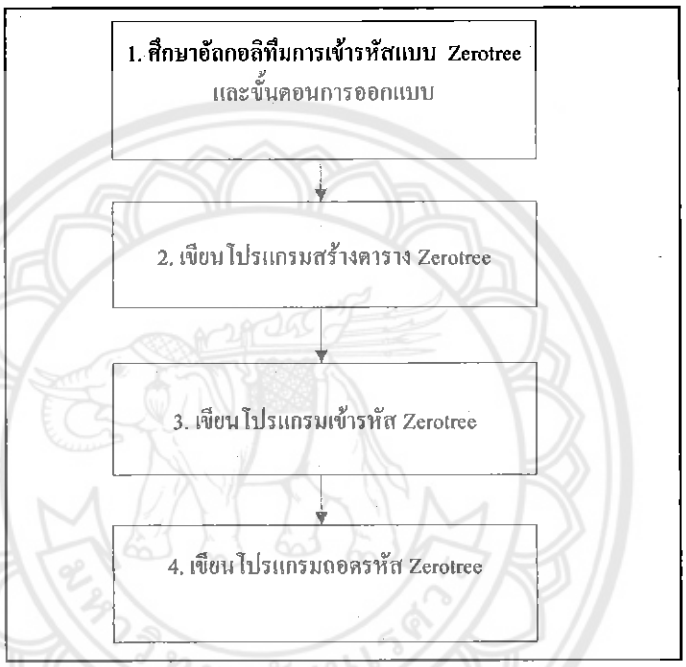
ผลของการหา subordinate list ในขั้นที่ 2 แสดงในตารางที่ 2.3

ตารางที่ 2.3

ค่าสัมประสิทธิ์	symbol	reconstruct
26	1 0	28 26
13	1	14 13
10	0	10 9
6	0	5
-7	1	-7
7	1	7
6	0	5
4	0	5
4	0	5
-4	0	-5
4	0	5

บทที่ 3
ขั้นตอนการดำเนินโครงงาน

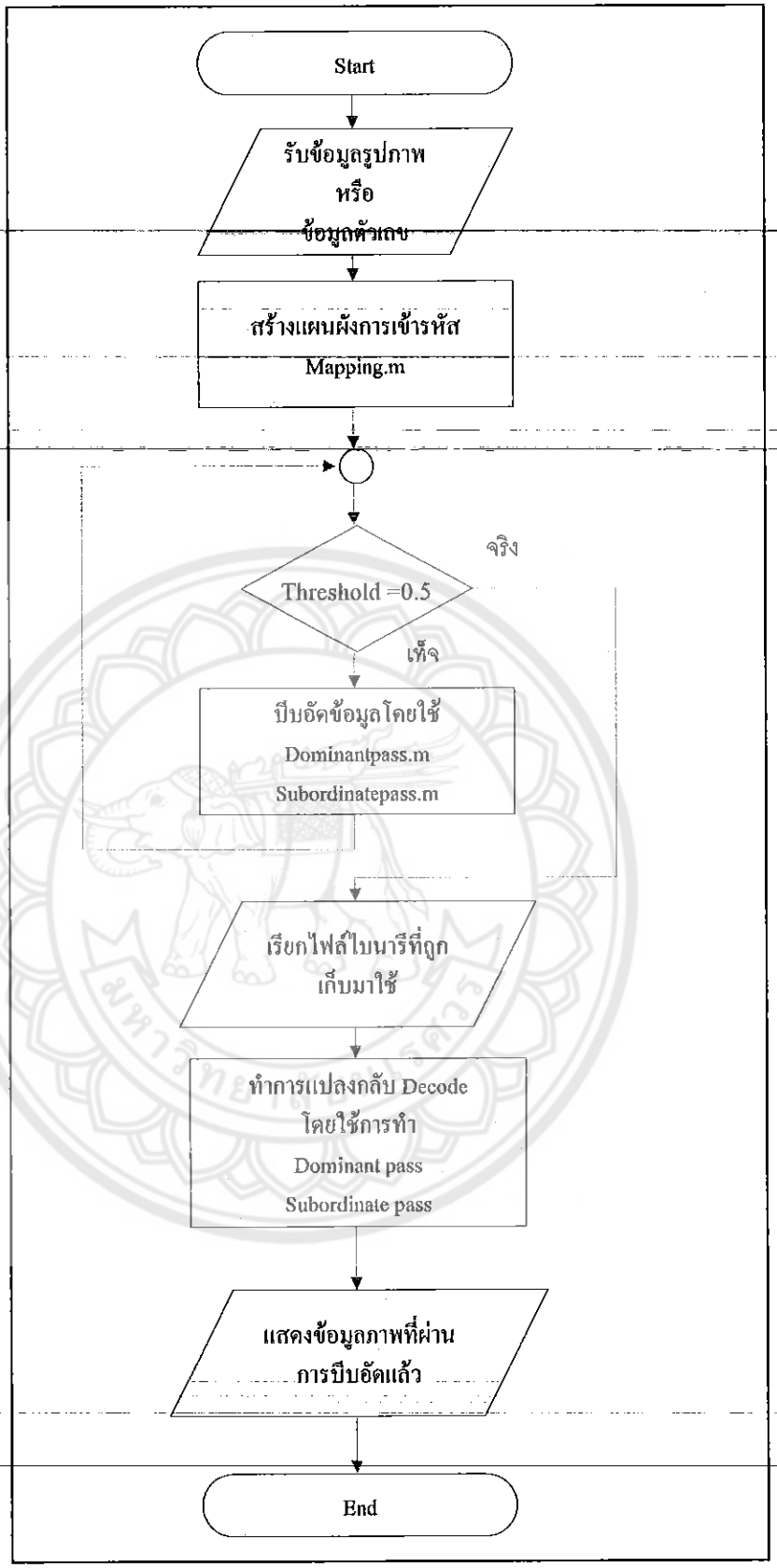
ในการบีบอัดข้อมูลภาพโดยวิธีซีโรทรี ได้แบ่งขั้นตอนการโครงงานออกเป็น 4 ขั้นตอน โดยเริ่มจากการศึกษาอัลกอริทึมการเข้ารหัสแบบ Zerotree และขั้นตอนการออกแบบ เขียนโปรแกรมสร้างแผนผังรหัสซีโรทรี เขียนโปรแกรมเข้ารหัสซีโรทรี และเขียนโปรแกรมถอดรหัสซีโรทรี แสดงดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการดำเนินโครงงาน

3.1 ศึกษาอัลกอริทึมการเข้ารหัสแบบซีโรทรีและขั้นตอนการออกแบบ

จากบทที่ 2 ศึกษาทฤษฎีและอัลกอลิทึมที่เกี่ยวข้อง เราสามารถนำทฤษฎีและอัลกอลิทึมที่ได้มาออกแบบโปรแกรมการบีบอัดข้อมูลภาพด้วยวิธีการซีโรทรีดังแสดงในรูปที่ 3.2 และนำมาเขียนโปรแกรมแผนผังรหัสซีโรทรี และ เขียนโปรแกรมการเข้ารหัสและถอดรหัสซีโรทรี ได้ซึ่งแสดงในหัวข้อต่อไป

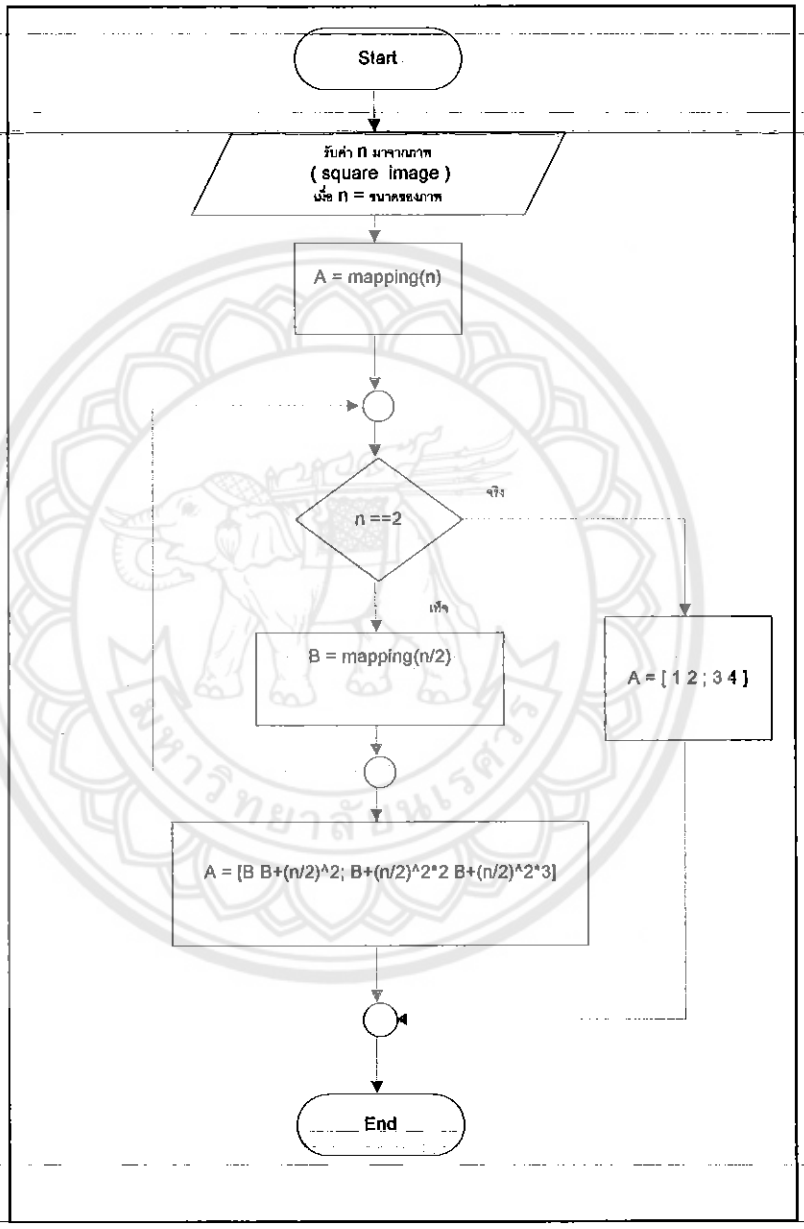


รูปที่ 3.2 Flowchart โปรแกรมการบีบอัดข้อมูลภาพแบบซีโรทรี

เลขหมู่ 621.64
ร15ร
ก387ก
2547

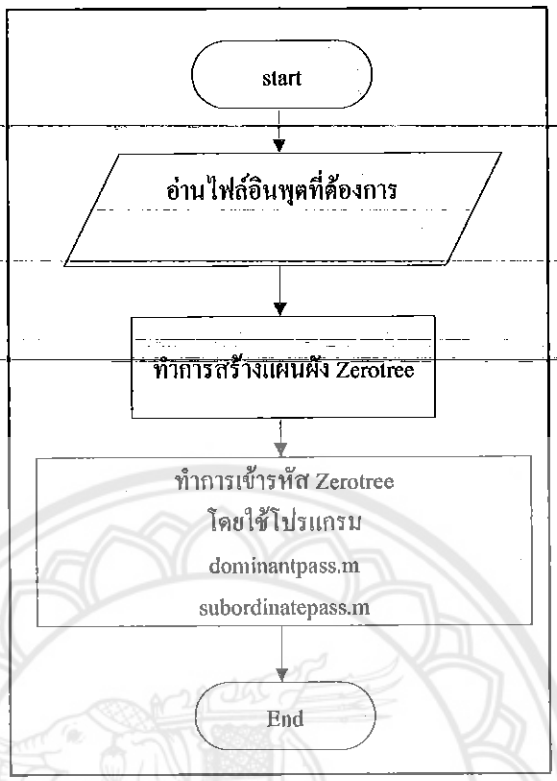
3.2 โปรแกรมการสร้างแผนผังรหัสซีโรทรี

การสร้างแผนผังรหัสซีโรทรีแสดงรูปที่ 3.3 และโปรแกรมการสร้างแผนผังรหัสซีโรทรีดูจากภาคผนวก จะใช้ค่าที่ได้นำมาสร้างแผนผังรหัสตามวิธีของการเข้ารหัสซีโรทรี โดยจะเริ่มจากการสร้างแผนผังของซีโรทรีก่อน ทำให้ได้โครงสร้างของแผนผังซีโรทรี โดยเก็บไว้ในตัวแปรอาเรย์เพื่อใช้ในการเข้ารหัสซีโรทรีต่อไป



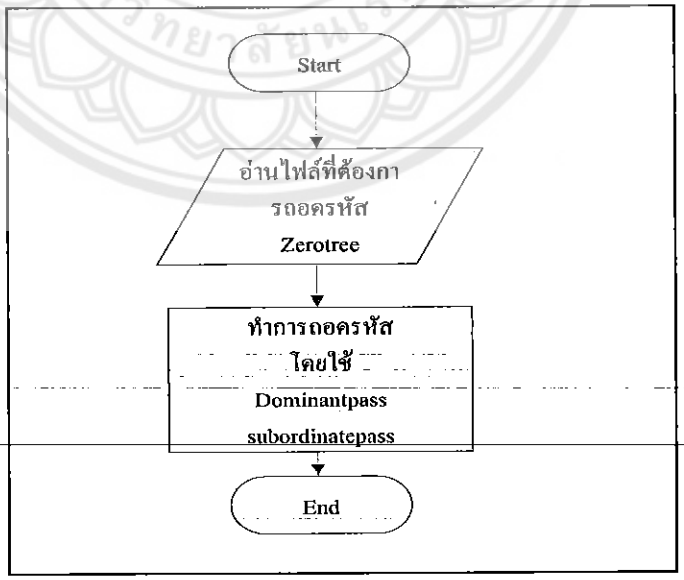
รูปที่ 3.3 Flowchart ฟังก์ชันสร้างแผนผังรหัสซีโรทรี

3.3 โปรแกรมเข้ารหัสซีโรทรี (Zerotree Encode)



รูปที่ 3.4 แผนภาพโปรแกรมการเข้ารหัสซีโรทรี (Zerotree Encode)

3.4 โปรแกรมการถอดรหัสซีโรทรี (Zerotree Decode)



รูปที่ 3.5 แผนภาพโปรแกรมถอดรหัสซีโรทรี (Zerotree Decode)

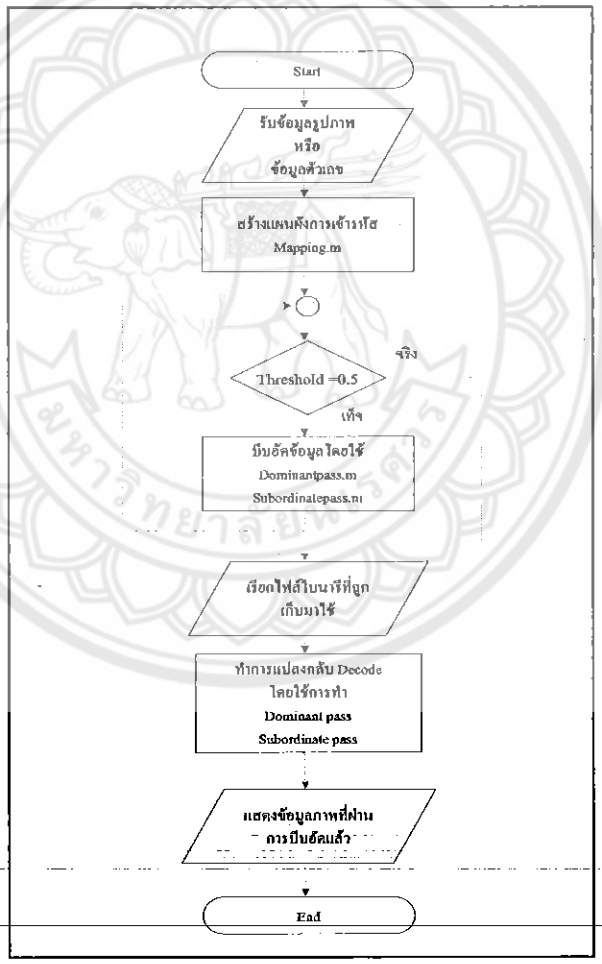
บทที่ 4
การทดลอง

4.1 บทนำ

ในบทนี้จะกล่าวถึงการนำหลักการทฤษฎีพื้นฐานและขั้นตอนการดำเนินการที่กล่าวมาแล้วในบทที่ 2 และ 3 มาใช้ในการทดลองลดขนาดข้อมูลภาพ โดยมีรายละเอียดของภาพและผลอัตราการบีบอัดข้อมูลภาพที่ได้ ค่า PSNR และ SNR ที่แสดงถึงคุณภาพของภาพที่ผ่านการลดขนาด

4.2 การเรียกใช้โปรแกรม

โปรแกรมการบีบอัดข้อมูลภาพโดยวิธีซีโรทรีเรียกใช้โปรแกรม test.m มีขั้นตอนการทำงานดังแสดงในรูปที่ 4.1 (สามารถดูโปรแกรมได้จากภาคผนวก)



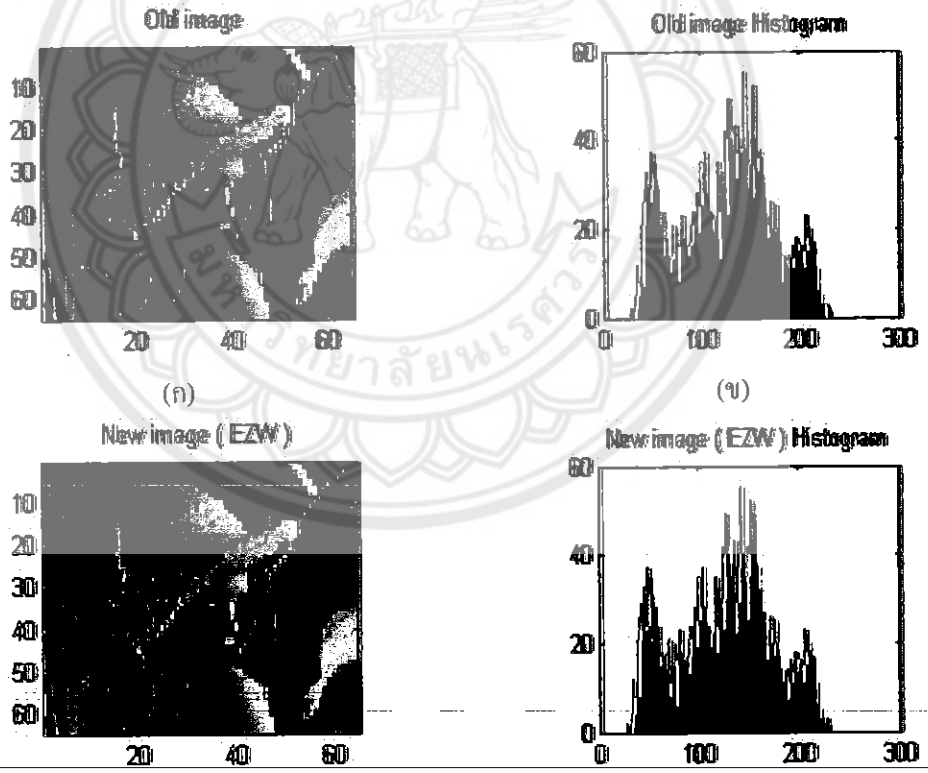
รูปที่ 4.1 Flowchart โปรแกรมการบีบอัดข้อมูลภาพโดยวิธีการซีโรทรี

4.3 การทดลองเข้ารหัสและการถอดรหัสซีโรทรี

การทดลองจะทำการเข้ารหัสและถอดรหัสซีโรทรี โดยใช้โปรแกรม MATLAB เปรียบเทียบกับการเข้ารหัสซีโรทรีโดยจะวัดจากอัตราการบีบอัดข้อมูลภาพ (Compression ratio) โดยไฟล์ตัวอย่างที่ใช้ในการทดลองเป็นไฟล์ที่เป็นไฟล์ภาพ ดังนี้

4.3.1 ผลการทดลองการบีบอัดข้อมูลภาพ

รูปที่ 4.2 (ก) แสดงภาพ Lena ก่อนการบีบอัดข้อมูลภาพซึ่งมีขนาดไฟล์ที่จัดเก็บ 5,757 bytes ความละเอียด 64X64 Pixel และรูปที่ 4.2 (ข) แสดงคุณลักษณะกราฟฮีสโตแกรมของภาพ ต่อจากนั้นเข้าการบีบอัดข้อมูลโดยใช้การเข้ารหัสซีโรทรี ได้ไฟล์การบีบอัดข้อมูลภาพซึ่งมีขนาดจัดเก็บ 3,251 bytes และได้คำนวณหาค่าอัตราส่วนการบีบอัดข้อมูลภาพเท่ากับ 1.770 เมื่อได้คลายการบีบอัดข้อมูลภาพจะได้ผลลัพธ์ของภาพแสดงไว้ในรูปที่ 4.2 (ค) ซึ่งได้แสดงผลกราฟฮีสโตแกรมไว้ในรูปที่ 4.2 (ง) ไว้ด้วย ทำการเปรียบเทียบภาพทั้งสองได้ค่า SNR และ PSNR เข้าสู่เลขอนันต์ นั้นแสดงให้เห็นว่าการบีบอัดและการคลายการบีบอัดข้อมูลภาพ Lena มีคุณลักษณะเหมือนกันทุกประการ



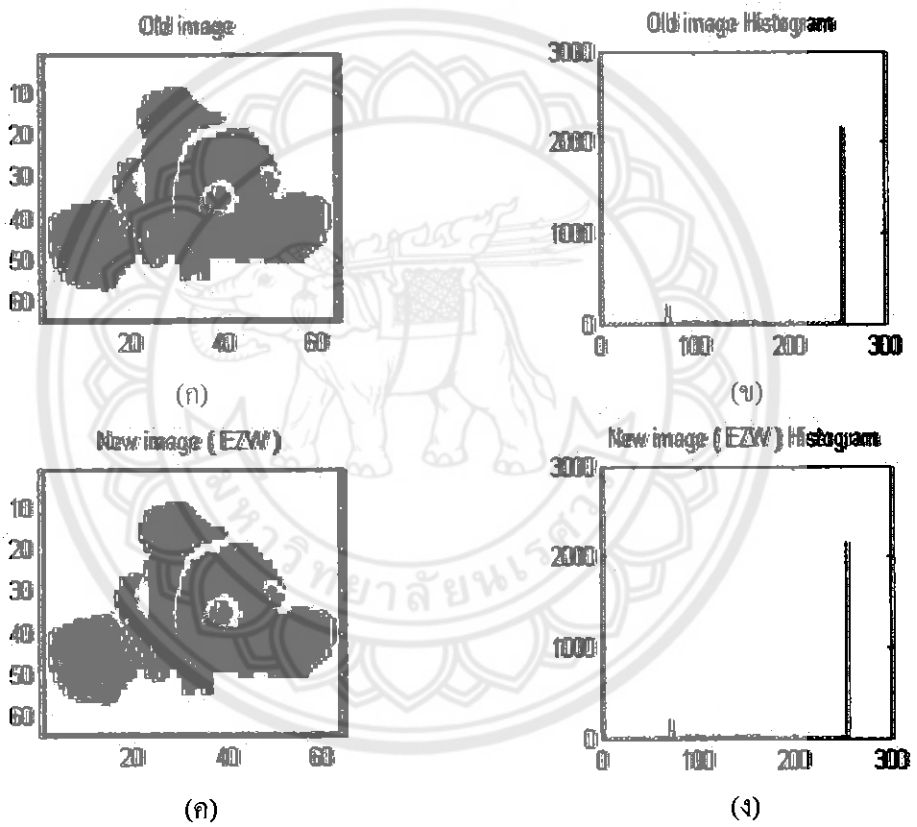
(ก) (ข)

(ค) (ง)

รูปที่ 4.2 แสดงภาพ Lena ก่อนการบีบอัดและหลังการบีบอัดข้อมูลภาพด้วยซีโรทรี

(CR = 1.770, SNR = ∞, PSNR = ∞)

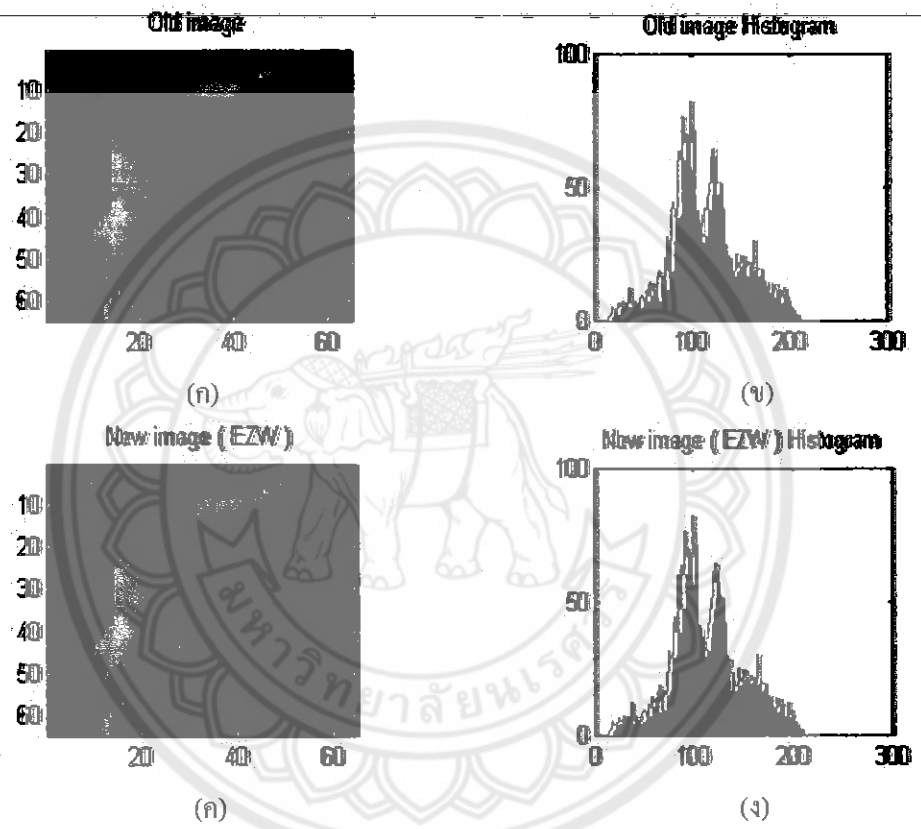
รูปที่ 4.3 (ก) แสดงภาพ Nemo ก่อนการบีบอัดข้อมูลภาพซึ่งมีขนาดไฟล์ที่จัดเก็บ 5,176 bytes ความละเอียด 64X64 Pixel และรูปที่ 4.3 (ข) แสดงคุณลักษณะกราฟฮีสโตแกรมของภาพ ต่อจากนั้นเข้าการบีบอัดข้อมูลโดยใช้การเข้ารหัสซีโรทรี ได้ไฟล์การบีบอัดข้อมูลภาพซึ่งมีขนาดจัดเก็บ 3,414 bytes และได้คำนวณหาค่าอัตราส่วนการบีบอัดข้อมูลภาพเท่ากับ 1.516 เมื่อได้คลายการบีบอัดข้อมูลภาพจะได้ผลลัพธ์ของภาพแสดงไว้ในรูปที่ 4.3 (ค) ซึ่งได้แสดงผลกราฟฮีสโตแกรมไว้ในรูปที่ 4.3 (ง) ไว้ด้วย ทำการเปรียบเทียบภาพทั้งสองได้ค่า SNR และ PSNR เข้าสู่เลขอนันต์ นั้นแสดงให้เห็นว่าการบีบอัดและการคลายการบีบอัดข้อมูลภาพ Nemo มีคุณลักษณะเหมือนกันทุกประการ



รูปที่ 4.3 แสดงภาพ Nemo ก่อนการบีบอัดและหลังการบีบอัดข้อมูลภาพด้วยซีโรทรี

(CR = 1.516, SNR = ∞, PSNR = ∞)

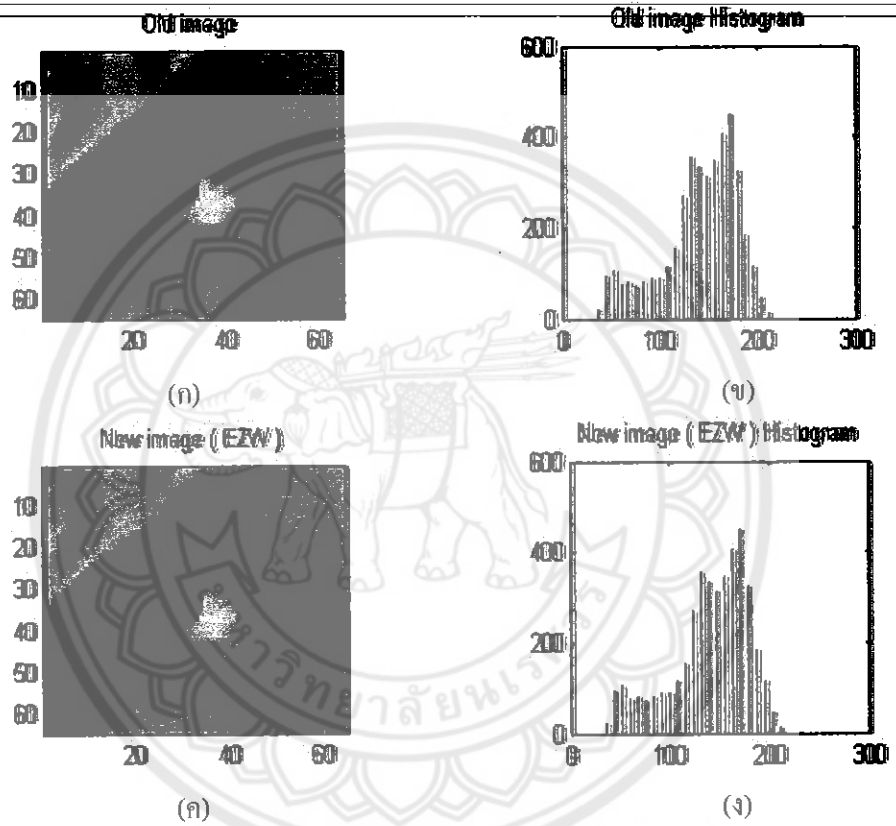
รูปที่ 4.4 (ก) แสดงภาพ dog ก่อนการบีบอัดข้อมูลภาพซึ่งมีขนาดไฟล์ที่จัดเก็บ 5,176 bytes ความละเอียด 64X64 Pixel และรูปที่ 4.4 (ข) แสดงคุณลักษณะกราฟฮีสโตแกรมของภาพต่อจากนั้นเข้าการบีบอัดข้อมูลโดยใช้การเข้ารหัสซีโรทรี ได้ไฟล์การบีบอัดข้อมูลภาพซึ่งมีขนาดจัดเก็บ 3,180 bytes และได้คำนวณหาค่าอัตราส่วนการบีบอัดข้อมูลภาพเท่ากับ 1.627 เมื่อได้คลายการบีบอัดข้อมูลภาพจะได้ผลลัพธ์ของภาพแสดงไว้ในรูปที่ 4.4 (ค) ซึ่งได้แสดงผลกราฟฮีสโตแกรมไว้ในรูปที่ 4.4 (ง) ไว้ด้วย ทำการเปรียบเทียบภาพทั้งสองได้ค่า SNR และ PSNR เข้าสู่เลขอนันต์ นั้นแสดงให้เห็นว่าการบีบอัดและการคลายการบีบอัดข้อมูลภาพ dog มีคุณลักษณะเหมือนกันทุกประการ



รูปที่ 4.4 แสดงภาพ dog ก่อนการบีบอัดและหลังการบีบอัดข้อมูลภาพด้วยซีโรทรี

(CR = 1.627, SNR = ∞, PSNR = ∞)

รูปที่ 4.5 (ก) แสดงภาพ Eye ก่อนการบีบอัดข้อมูลภาพซึ่งมีขนาดไฟล์ที่จัดเก็บ 5,486 bytes ความละเอียด 64X64 Pixel และรูปที่ 4.5 (ข) แสดงคุณลักษณะกราฟฮีสโตแกรมของภาพต่อจากนั้นเข้าการบีบอัดข้อมูลโดยใช้การเข้ารหัสซีโรทรี ได้ไฟล์การบีบอัดข้อมูลภาพซึ่งมีขนาดจัดเก็บ 3,395 bytes และได้คำนวณหาค่าอัตราส่วนการบีบอัดข้อมูลภาพเท่ากับ 1.615 เมื่อได้คลายการบีบอัดข้อมูลภาพจะได้ผลลัพธ์ของภาพแสดงไว้ในรูปที่ 4.5 (ค) ซึ่งได้แสดงผลกราฟฮีสโตแกรมไว้ในรูปที่ 4.5 (ง) ไว้ด้วย ทำการเปรียบเทียบภาพทั้งสองได้ค่า SNR และ PSNR เข้าสู่เลขอนันต์ นั้นแสดงให้เห็นว่าการบีบอัดและการคลายการบีบอัดข้อมูลภาพ Eye มีคุณลักษณะเหมือนกันทุกประการ



รูปที่ 4.5 แสดงภาพ Eye ก่อนการบีบอัดและหลังการบีบอัดข้อมูลภาพด้วยซีโรทรี

(CR = 1.615, SNR = ∞, PSNR = ∞)

จากผลการทดลองการเข้ารหัสและถอดรหัสไฟล์ภาพแบบซีโรทรี ดังกล่าวข้างต้นสามารถสรุปผลที่ได้จากการบีบอัดข้อมูลภาพโดยวิธีซีโรทรี ดังตารางที่ 4.1

ตารางที่ 4.1 สรุปผลที่ได้จากการบีบอัดข้อมูลภาพโดยวิธีการแบบซีโรทรีทั้ง 4 กรณี

ชื่อภาพ	ขนาด (พิกเซล)	ขนาดไฟล์ (bytes)	CR	SNR (dB)	PSNR (dB)
Lena	64 x 64	5,757	1.770	∞	∞
Nemo	64 x 64	5,176	1.516	∞	∞
dog	64 x 64	5,176	1.627	∞	∞
Eye	64 x 64	5,486	1.615	∞	∞

บทที่ 5
บทสรุป

จากการทดลองเพื่อลดขนาดข้อมูลภาพโดยใช้การเข้ารหัส และถอดรหัสแบบซีโรทรีกับภาพต้นแบบจำนวน 4 ภาพ คือ ภาพ Lena (lena.gif) , ภาพ Nemo (nemo.bmp) , ภาพ dog (dog.bmp) และภาพ Eye (eye1.tif) ซึ่งมีความละเอียด 64 x 64 พิกเซล โดยใช้วิธีซีโรทรีตามที่ได้อธิบายไว้ในวิธีการดำเนินการทดลองในบทที่ 3 พบว่าอัตราการบีบอัดข้อมูลของภาพ Lena เท่ากับ 1.770 ภาพ Nemo เท่ากับ 1.516 ภาพ dog เท่ากับ 1.627 และภาพ Eye เท่ากับ 1.615 นอกจากนี้พบว่าภาพทุกภาพเมื่อทำการคลายการบีบอัดภาพแล้วจะได้ค่า SNR และ PSNR เข้าสู่เลขอนันต์ นั้นแสดงให้เห็นว่าไม่มีการสูญเสียข้อมูลจากการบีบอัด และการคลายการบีบอัดข้อมูลภาพโดยใช้วิธีซีโรทรี

เอกสารอ้างอิง

[1] รศ.ดร.มนัส สังวรศิลป์ . **คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์** .กรุงเทพมหานคร : อินโฟเพรส, 2543

[2] A. S. Lewis and G. Knowies, **Image Compression Using the 2-D Wavelet Transform,** IEEE Trans Image Processing, Vol. 1, no. 2, 1992

[3] M. A. Sid-Ahmed, **Image Processing Theory, Algorithms, & Architectures.** McGraw-hill, Inc., 1995

[4] Randy K. Young, **Wavelet Theory and its Applications,** Kluwer Academic Publisherd, 1993

[5] Eric J. Stollnitz, Tony D. DeRose and David H. Selesin, **Wavelets for Computer Graphics: A Primer, Part1**, IEEE Computer Graphics and Applications, May 1995

[6] Jerome M. Shapiro , **Embedded Image Coding using Zerotrees of Wavelet Coefficients**, IEEE Transactions on signal Processing, December 1995

ภาคผนวก

โปรแกรมการบีบอัดข้อมูลภาพโดยวิธีซีโรทรี

SOURCE PROGRAM

1. โปรแกรมเรียกใช้การบีบอัดข้อมูลภาพ

```
% ==========================================
% (Test zerotree code)
% File name : test.m
% By :  Mr. Watchason  Wannasorn     ID 44370369
% ==========================================
clear all
close all

% This is a test program that tests the EZW encoder and decoder
% This function calls:
%    dominantpass.m
%    subordinatepass.m
%    checkdescents1.m
%    checkchildren.m
%    mapping.m

X=imread('eye1.tif');
X= double(X);
% ==========================Show=============================
figure(1)
map=gray(255);
subplot(2,2,1);
image(X),colormap(map)
title('Old image')

subplot(2,2,3);
[hx,hy]=histo(X);
bar(hx,hy,0)
title('Old image Histogram')
% ==========================================
```

```
X0=X;
Y0=max(X);

  Y1=max(Y0);
  for i=0:20;
    if 2^i<=Y1 & 2^i>0.5*Y1;
      threshold=2^i;                    % get initial threshold T0;
      initialthreshold=threshold;       % get initial threshold T0;
      laststeplevel=i+1;                % last step level
      break;
    end;
  end;



  sublist=[];
  sub_list=[];
  [xx,yy]=size(X);

A=mapping(xx);
[m,n]=size(A);
global N;                               % Let Morton scanorder vector as a global variable
 N=zeros(m*n,2);
   for i=1:m,
     for j=1:n,
         N(A(i,j),1)=i;
         N(A(i,j),2)=j;
     end
   end

order=1;
```

```
while threshold ~= 0.5,                % if threshold~=1, do dominantpass and subordinatepass.
threshold        ;

%Dominant Pass
[D,X,sublist,sub_list] = dominantpass(X,threshold,sublist,sub_list);
DD{order}=D     ;
significantlist{order}=sub_list;

%Subordinate pass
threshold=threshold/2;
if threshold ==0.5,
   break;
end

S = subordinatepass(sublist,threshold);
SS{order}=S      ;
order=order+1;
end

% ==========Save File==========
encode = 0;
for i=1:1:length(SS),
  encode = [SS{1,i} encode];
end

% Save Files
fid = fopen('zerostree.ezw','wb');                % open file in binary write mode
```

```
fwrite(fid,encode,'ubit1');                % write data to file
fclose(fid);                               % close file
[a b] = size(encode);
```

```
%read in the same file
fid  = fopen('zerostree.ezw','rb');        % open file
data = fread(fid,[a b],'ubit1');           % read in the data
fclose(fid);                               % close file
```

```
% Show size files
AD=dir('eye1.tif')
BD=dir('zerostree.ezw')
decode = 0 ;
for ii=length(SS):-1:1,
  decode = [encode(1:1:length(ii)) decode];
end
% ==========================================
%*********************EZW decoder**********************%
% ==========================================
 global N;
 [m,n]=size(N);                            % the size of initial image
                                           % m is the pixels of initial image

 XX=zeros(sqrt(m));                        % initialize the reconstructured image to zero;
 threshold=initialthreshold;               % initial theshold ;

 sublist=[];                               % sublist is the new position matrix
                                           % for all significant coefficients 'p' and 'n';
```

```
for level=1:laststeplevel,
        RR=zeros(size(XX));                % reference matrix RR;
      [a,b]=size(DD{level});
```

```
% dominant pass
   i=1; j=1;
   while j<=b,
     if RR(N(i,1),N(i,2))==0
        if DD{level}(j)=='p'
          if threshold==1
            XX(N(i,1),N(i,2))=threshold;
          else
            XX(N(i,1),N(i,2))=1.5*threshold;
          end
        end

        if DD{level}(j)=='n'
          if threshold==1
            XX(N(i,1),N(i,2))=-threshold;
          else
            XX(N(i,1),N(i,2))=-1.5*threshold;
          end
        end

        if DD{level}(j)=='t'& A(N(i,1),N(i,2))<=m/4
          RR=checkchildren(i,RR);          % all zerotree's descendants are set to 1.
        end

        RR(N(i,1),N(i,2))=1;               %reference matrix =1;
        i=i+1;
        j=j+1;

     else i=i+1;
     end
   end
```

```
% subordinate pass

[xx,yy]=size(significantlist{level});
threshold=threshold/2;
 for i=1:xx,
   if level==laststeplevel|threshold==0.5
     break
   end

  if SS{level}(i)==1
    if XX(sub_list(i,1),sub_list(i,2))>0;
      XX(sub_list(i,1),sub_list(i,2))= fix(XX(sub_list(i,1),sub_list(i,2))+ threshold/2);
    else
      XX(sub_list(i,1),sub_list(i,2))= fix(XX(sub_list(i,1),sub_list(i,2))-threshold/2);
    end
  end

  if SS{level}(i)==0
    if XX(sub_list(i,1),sub_list(i,2))>0;
      XX(sub_list(i,1),sub_list(i,2))= fix(XX(sub_list(i,1),sub_list(i,2))-threshold/2);
    else
      XX(sub_list(i,1),sub_list(i,2))= fix(XX(sub_list(i,1),sub_list(i,2))+threshold/2);
    end
  end
 end

% ==========Show Value=====================
   threshold ;
   level ;
   XX ;
end
```

```
initialiamge=X0;
reconstructedimage=XX;
difference=XX-X0;
% ==================Error=======================
% Check Error
[M N] = size(XX);
nsig = XX;
desig = X0;
ER = (nsig-desig);
ERR = ER^2;
nesig = nsig^2;
Error = (sum(ERR(:)))/M*N;
SNR = 10*log10((sum(nesig(:))/M*N)/Error)
PSNR = 10*log10(max(nesig(:))/Error)
% ================Show new image  EZW=============
% show  new image (ezw)
subplot(2,2,2);
image(XX),colormap(map)
title('New image ( EZW )')

subplot(2,2,4);
[hx_n,hy_n]=histo(XX);
bar(hx_n,hy_n,0)
title('New image ( EZW ) Histogram')
```

2. โปรแกรมฟังก์ชันที่ใช้ในการบีบอัด

```
%=========================================================
% function A = mapping(n)
% File name : mapping.m
% By :  Mr. Watchason  Wannasorn     ID 44370369
% =========================================================
function A = mapping(n)

if n == 2
  A = [1 2; 3 4];
else
  B = mapping(n/2);
  A = [B B+(n/2)^2; B+(n/2)^2*2 B+(n/2)^2*3];
end
```

```
%==========================================================
% function [D,X,sublist,sub_list] = dominantpass(X,threshold,sublist,sub_list)
% File name : dominantpass.m
% By :  Mr. Watchason  Wannasorn    ID 44370369
% ==========================================================
function [D,X,sublist,sub_list] = dominantpass(X,threshold,sublist,sub_list)
% Dominant pass function
D=[];
global N;
[m,n]=size(X);                  % X is the coefficients matrix
R=zeros(m);                     % matrix R is a reference matrix, same size as X; '0' means
                                %this coefficient is not a descendant from zerotree root;
[a,b]=size(N);

    if abs(X(1,1))>=threshold           % X(1,1) is DC coefficient
     sublist=[sublist, abs(X(1,1))];    % put significant coefficients's value to sublist
     sub_list=[sub_list;N(1,1),N(1,2)];% put the significant coefficients' position in sub_list
     if X(1,1)>0;
       D=[D,'p'];
     else D=[D,'n'];
     end
     X(1,1)=0;
    else D=[D,'z'];
    end

  for k=2:4,
     if abs(X(N(k,1),N(k,2)))>=threshold,
       sublist=[sublist, abs(X(N(k,1),N(k,2)))];
       % append this significant coefficient to the subordinate list;
```

```
sub_list=[sub_list;N(k,1),N(k,2)];

if X(N(k,1),N(k,2))>0                % determine the sign
  D=[D,'p']; % >0,assign a "p"
else D=[D,'n'];% <0,assign a "n"
end
X(N(k,1),N(k,2))=0;

% the significent coefficients is replaced by a '0' in the coefficients matrix
  else

   % 2,3,4 has no parents,just check its descendants.
      result = checkdescendants1( k,X,threshold,0);
      if result==1
        D=[D,'z'];
      else
        D=[D,'t'];
        R(N(k,1),N(k,2))=1;          % Zerotree, make all its descendants
        R=checkchildren(k,R);        % refference matrix component to 1.
      end

  end
end

for k=5:a,
   if abs(X(N(k,1),N(k,2)))>=threshold,
     sublist=[sublist, abs(X(N(k,1),N(k,2)))];
     sub_list=[sub_list;N(k,1),N(k,2)];
      if X(N(k,1),N(k,2))>0,         % determine the sign
        D=[D,'p'];                   % >0,assign a "p"
      else D=[D,'n'];                % <0,assign a "n"
      end
```

```
                X(N(k,1),N(k,2))=0;
            elseif R(N(k,1),N(k,2))==0
                result = checkdescendants1( k,X,threshold,0);
                                        % Check its has significant descendants?
                if result==1,
                  D=[D,'z']; % isolated zero
                else D=[D,'t'];% zerotree
                  R(N(k,1),N(k,2))=1;
                  R=checkchildren(k,R);
                                        % if zerotree, reference matrix coefficients=1
                end
            end
        end
```

```
%=========================================================
% function   result = checkdescendants1(j,X,threshold,result)
% File name :  checkdescendants1.m
% By :  Mr. Watchason  Wannasorn    ID 44370369
% =========================================================
function   result = checkdescendants1(j,X,threshold,result)
% initial set result=0
% if the result=1, means that a coefficient has at least
% 1 significant descendant.

global N
[m,n]=size(N);
for i=(4*j-3):4*j;
    if result==1 | i > m
      break;
    end;

   if abs(X(N(i,1),N(i,2)))>=threshold
     result=1;
     break;
   else
      result=checkdescendants1(i,X,threshold,result);
   end;
end;
```

```
%=========================================
% function   RR=checkchildren(j,RR)
% File name : checkchildren.m
% By :  Mr. Watchason  Wannasorn    ID 44370369
% =========================================
function   RR=checkchildren(j,RR)

% if a symbol 't' is encounted, then make all its descendants in reference
% matrix RR's components equal 1---ZEROTREES

global N
[m,n]=size(N);
for i=(4*j-3):4*j;
    if  i<=m,
      RR(N(i,1),N(i,2))=1;
      RR=checkchildren(j,RR);
    end
end;
```

3. โปรแกรมหาฮีสโตแกรมของข้อมูล

```
%=============================================
% function [l,h] = histo(xx);
% File name : histo.m
% By :  Mr. Watchason  Wannasorn     ID 44370369

% HISTO Display histogram of image data.
% L,H]=HISTO(I) displays a histogram with
% H(0),H(1),...,H(L),...,H(255)
% for the intensity image I of L =0,1,2,...,255.
% ==========================================
function [l,h] = histo(xx);
h = [];
for gl=0: 1: 255
   [indx,indy,val] =   find(xx==gl);
   h               =   [h sum(val)];
end
l = 0: 1: 255;
```

ประวัติผู้เขียนโครงงาน

ชื่อ นายวัชสัณห์ วรรณสอน

ภูมิลำเนา 16/7 หมู่ 2 ต.ขี้เหล็ก อ.แม่แตง จ.เชียงใหม่

ประวัติการศึกษา

- จบการศึกษาชั้นมัธยมศึกษาจากโรงเรียนนวมินทราชูทิศ พายัพ
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-Mail : jameslink11@hotmail.com